

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA - NÚCLEO DE INTELIGÊNCIA
ARTIFICIAL**

TATIANA OLIVEIRA LIMA

**PRODUÇÃO DE PÁGINAS JORNALÍSTICAS DINÂMICAS DIRETO NA NUVEM,
COM HIPERTEXTOS, MULTIMÍDIA E TEXTMINING, MULTIPLATAFORMA,
MULTIDEVICES, UTILIZANDO PARADIGMA FUNCIONAL LISP COM EMACS
ORG-MODE**

**UBERLÂNDIA
2016**

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA - NÚCLEO DE INTELIGÊNCIA
ARTIFICIAL**

TATIANA OLIVEIRA LIMA

**PRODUÇÃO DE PÁGINAS JORNALÍSTICAS DINÂMICAS DIRETO NA NUVEM,
COM HIPERTEXTOS, MULTIMÍDIA E TEXTMINING, MULTIPLATAFORMA,
MULTIDEVICES, UTILIZANDO PARADIGMA FUNCIONAL LISP COM EMACS
ORG-MODE**

Dissertação apresentada ao núcleo de Inteligência Artificial, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, como requisito parcial para a obtenção do título de Mestre em Ciências.

Orientador: Keiji Yamanaka
Coordenador: Darizon A. Andrade

**UBERLÂNDIA
2016**

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

L732p
2016

Lima, Tatiana Oliveira, 1989-

Produção de páginas jornalísticas dinâmicas direto na nuvem, com hipertextos, multimídia e textmining, multiplataforma, multidevices, utilizando paradigma funcional LISP com EMACS ORG-MODE [recurso eletrônico] / Tatiana Oliveira Lima. - 2016.

Orientador: Keiji Yamanaka.

Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.

Modo de acesso: Internet.

Disponível em: <http://doi.org/10.14393/ufu.di.2020.3004>

Inclui bibliografia.

Inclui ilustrações.

1. Engenharia elétrica. 2. Computação em nuvem. 3. Jornalismo eletrônico. 4. Multimídia (Computação). I. Yamanaka, Keiji, 1956-, (Orient.). II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.3

TERMO DE APROVAÇÃO

TATIANA OLIVEIRA LIMA

PRODUÇÃO DE PÁGINAS JORNALÍSTICAS DINÂMICAS DIRETO NA NUVEM,
COM HIPERTEXTOS, MULTIMÍDIA E TEXTMINING, MULTIPLATAFORMA,
MULTIDEVICES, UTILIZANDO PARADIGMA FUNCIONAL LISP COM EMACS
ORG-MODE

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Núcleo de Pós-Graduação em Inteligência Artificial, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, pela seguinte banca examinadora:

Prof. Dr. Keiji Yamanaka
Orientador
FEELT - UFU

Prof. Dr. Luciano Martins Neto
FEELT - UFU

Prof. Dr. Reny Cury Filho
PMU

Uberlândia, 08 de Fevereiro de 2016.



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Ata da defesa de DISSERTAÇÃO DE MESTRADO junto ao Programa de Pós-graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia.

Defesa de Dissertação de Mestrado Acadêmico, número 622/2016/PPGEE

Data: 08 de fevereiro de 2016.

Discente: TATIANA OLIVEIRA LIMA

Número de matrícula: 11412EEL015

Título do Trabalho: Produção de páginas jornalísticas dinâmicas direto na nuvem, com hipertextos multimídia e textmining multiplataforma, multidevices, utilizando paradigma funcional lisp com Emacs org-mode.

Área de concentração: Processamento da Informação.

Linha de pesquisa: Inteligência Artificial.

As 14:00 horas do dia 08 de fevereiro de 2016 na Sala de Defesas da Faculdade de Engenharia Elétrica, Campus Santa Mônica da Universidade Federal de Uberlândia, reuniu-se a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta:

Luciano Martins Neto

CPF: 538.578.288-20

Reny Cury Filho

CPF: 288.228.896-49

e Keiji Yamanaka

CPF: 005.225.308-26, orientador do candidato.

Iniciando os trabalhos o presidente da mesa Prof. Dr. Keiji Yamanaka apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu a Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir a candidata. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu os conceitos finais.

Em face do resultado obtido, a Banca Examinadora considerou a candidata A provada.

Esta defesa de Dissertação de Mestrado Acadêmico é parte dos requisitos necessários à obtenção do título de Mestre. O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar, foram encerrados os trabalhos às 16 horas e 10 minutos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Prof. Dr. Luciano Martins Neto
UFU

Prof. Dr. Reny Cury Filho
Prefeitura Municipal de Uberlândia

Prof. Dr. Keiji Yamanaka
UFU

Ao meu amado pai, amigo e exemplo.

AGRADECIMENTOS

Ser um bom profissional está além de fazer uma boa faculdade, é um processo de constante atualização e aperfeiçoamento. Sempre busquei formas não convencionais de ser uma jornalista diferenciada, e por vezes me esbarrei com a tecnologia. Muito me surpreende que vários colegas meus sejam quase leigos quando se trata de computadores, então pouco depois de me formar, me veio a pergunta: como eu posso contribuir para melhorar esse cenário?

Um desafio grande que sem algumas pessoas não poderia se tornar possível. Deus sempre me deu forças para continuar, abrindo as portas certas e fechando as erradas. Sempre apontando soluções, até mesmo colocando as pessoas certas no meu caminho. Impossível dizer que um trabalho desses seria mérito meu, é resultado de um conjunto de pessoas e um trabalho árduo.

Agradeço primeiramente, e meu querido orientador que me perdoe, a meu pai, sempre ao meu lado me suportando, ajudando, e me guiando, como uma espécie de anjo da guarda por quem eu tenho o maior dos amores.

Agora sim eu agradeço ao meu orientador, Keiji. Sem seu apoio eu não estaria aqui hoje podendo concluir o trabalho.

Mãe, seu imenso amor me deu forças e sossego, sempre nas horas certas. Thales e Daniel, meus queridos irmãos, qualquer agradecimento pela cumplicidade e companheirismo é pouco.

Vítor e Ana Augusta, os estagiários que foram além de suas funções para contribuir e somar, meu muito obrigada. A Ana no começo do trabalho, com toda a sua alegria e dedicação, e ao Vítor por estar sempre pronto com diferentes soluções, mas nunca perdendo a calma, independente do dia e horário.

Aos meus, por acreditarem em mim, sempre.

Jornalismo é publicar aquilo que alguém não quer que se publique.

Todo o resto é publicidade.

George Orwell

SUMÁRIO

RESUMO.....	XIV
1 INTRODUÇÃO	16
1.1 ESTADO DA ARTE	16
1.2 O DESAFIO DOS JORNAIS DA ERA DA INTERNET.....	18
2 OBJETIVO	20
2.1 GERAL	20
2.2 ESPECÍFICOS	20
3 JUSTIFICATIVA.....	21
4 METODOLOGIA	25
4.1 TIPO DE MÉTODO DE ABORDAGEM	25
4.2 PESQUISA.....	25
4.3 O DELINEAMENTO DA PESQUISA DA DISSERTAÇÃO.....	26
4.3.1 Recursos Necessários À Pesquisa:	26
4.4 CRONOGRAMA REALIZADO.....	27
5 CONCEITOS E TÉCNICAS SOBRE JORNAIS IMPRESSOS E DIGITAIS (WEB JOURNALS).....	28
5.1 BREVE HISTÓRIA DO JORNALISMO.....	28
5.2 COMUNICAÇÃO VISUAL NOS JORNAIS	30
5.3 ESTRUTURA DO JORNALISMO NA WEB.....	31
5.4 A TECNOLOGIA E O JORNALISMO	31
5.5 OS NATIVOS DIGITAIS E O JORNALISMO.....	32
5.6 A PROPAGANDA E OS JORNAIS.....	34
5.7 ESTRUTURA DE PRODUÇÃO DE UM JORNAL	36
6 OS JORNAIS IMPRESSOS E DIGITAIS NO BRASIL – UMA ANÁLISE COMPARATIVA.....	39
6.1 CIRCULAÇÃO	39
6.2 ONLINE/IMPRESSO	40
6.3 DIÁRIO/TEMPO REAL.....	40
6.4 CANAIS DE INTERAÇÃO	41
6.5 PADRÃO DE CORES.....	41
6.6 NÚMERO DE COLUNAS	42
6.7 EDITORIAS	42
6.8 FORMA DE CONTATO.....	43

7	A ESCOLHA DA LINGUAGEM PARA PROGRAMAÇÃO, IMPLEMENTAÇÃO SAP (SIMPLE AS POSSIBLE) DE PÁGINAS DINÂMICAS DE NOTÍCIAS NA WEB, FERRAMENTAS E CONCEITOS.....	44
7.1	VISÃO GERAL DAS LINGUAGENS COMPATÍVEIS COM O DOMÍNIO DAS FERRAMENTAS A SEREM UTILIZADAS E DESENVOLVIDAS NESTE TRABALHO	44
7.1.1	Programação Direta Na Nuvem Sem A Censura, Limitações Impostas Pelo Jailshell Do Servidor Da Hospedagem.....	44
7.2	A ESCOLHA DAS LINGUAGENS	47
7.2.1	Emacs – Org-Mode	47
7.2.1.1	<u>Multiplataforma</u>	54
7.2.1.2	<u>Instalando o Emacs no Windows</u>	55
7.2.1.3	<u>Baixando o Emacs para Windows</u>	56
7.2.1.4	<u>Abrindo, executando finalmente o Emacs</u>	63
7.3	SSH – SECURE SHELL – O CAMINHO SEGURO PARA SE PROGRAMAR DIRETO NA NUVEM E PARA COMUNICAR, TROCAR DADOS ENTRE A MESMA E SEU COMPUTADOR.....	68
7.3.1	O Que É Ssh?	68
7.3.2	Vantagens do Uso do Protocolo SSH	69
7.3.3	Dados Fundamentais Para se Poder Implementar Páginas Direto na Nuvem.....	70
7.3.3.1	<u>Quatro parâmetros, informações necessárias para abrir um terminal putty no windows (bem como em qualquer outro sistema operacional), fazer transmissão de dados segura entre desktop e nuvem usando scp...</u>	70
7.3.4	Terminal SSH no Windows Com o Putty.....	72
7.3.4.1	<u>Abrindo um terminal SSH com o Putty</u>	72
7.3.4.2	<u>Usando o terminal SSH do Putty: comandos básicos necessários à esta pesquisa</u>	75
7.4	EMACS-LISP.....	79
7.4.1	Conceitos Básicos De Elisp, Uma Linguagem De Paradigma Funcional.	79
7.4.2	Funções, Domínio, Imagem - Características Básicas, Formalização:	79
7.4.3	Sintaxe	81
7.4.4	Criando Uma Função Em Elisp:	82
7.4.5	Criando Uma Função No Elisp	85
7.4.5.1	<u>Funções com eventos</u>	86
7.4.6	Outras Características, Conceitos E Funções Em Elisp: Manipulando Strings (Texto) E Listas.....	87
7.4.7	Expressões Regulares Em Elisp	89
7.4.8	Manipulando Listas	90

7.4.9	Lendo Arquivos	91
7.4.9.1	<u>Lendo o arquivo e retornando uma string</u>	91
7.4.9.2	<u>Lendo o arquivo e retornando uma lista de string dado um separador ou uma expressão regular como separador</u>	92
7.4.9.3	<u>Limpendo caracteres, palavras indesejadas do texto</u>	94
7.4.9.4	<u>Criando uma função para ser lembrada mais facilmente</u>	94
7.5	CONCLUSÃO.....	95
8	CONCEITOS E ELEMENTOS DE PÁGINAS DE INTERNET	96
8.1	A PÁGINA DINÂMICA.....	96
8.2	BROWSERS, LINGUAGENS: LIMITES TECNOLÓGICOS	96
8.3	PRODUÇÃO DE PÁGINAS MULTIMÍDIA E COM HIPERTEXTO EM HTML... 97	
8.4	COMENÇANDO PELO FIM	98
8.4.1	Separando A Query String Da Url	101
8.5	ELEMENTOS BÁSICOS DE UMA PÁGINA COM CODIFICAÇÃO HTML, PASSO A PASSO	101
8.6	CODIFICAÇÃO HTML BÁSICA NA MONTAGEM DOS 8 ITENS, ELEMENTOS UTILIZADOS NO JORNAL.....	102
8.7	ELEMENTOS DAS PÁGINAS DO JORNAL.....	103
8.7.1	Título, Subtítulo E Texto:.....	103
8.7.2	Links.....	106
8.7.2.1	<u>Individualizado:</u>	106
8.7.2.2	<u>Inserido um link no meio de um texto:</u>	107
8.7.3	Figuras E Imagens	107
8.7.3.1	<u>Imagem estática local</u>	107
8.7.3.2	<u>Imagem vinda de um link</u>	108
8.7.3.3	<u>Imagem com link ativo</u>	109
8.7.4	Sons, Músicas, Áudio Em Geral.....	110
8.7.5	Vídeo.....	111
8.7.5.1	<u>Colocando um vídeo local</u>	111
8.7.5.2	<u>Colocando um link (código de incorporação) de um video da internet</u>	111
8.7.6	TEXT MINING	113
	Como o mesmo exige programação em linguagem de alto nível, este item será apresentado no próximo capítulo.	113
8.8	ESTRUTURA FINAL DE UM ARQUIVO BEM FORMATADO EM HTML.....	113
8.9	CONCLUSÃO:.....	116

9	PROGRAMANDO, CRIANDO AS PÁGINAS DINÂMICAS EM DESKTOP E DIRETO NA NUVEM UTILIZANDO O EDITOR EMACS-LISP (NA NUVEM) E EMACS NO MODO ORG (ORG-MODE NO DESKTOP)	118
9.1	FAZENDO UMA PÁGINA DINÂMICA USANDO EMACS-ORG-MODE NO DESKTOP	120
9.1.1	Criando Um Diretório Para A Página.....	121
9.1.2	Instalando Emacs No Windows.....	121
9.1.3	Instalando O Emacs Automaticamente Pelo Windows.....	122
9.2	ELEMENTOS DA PÁGINA:.....	124
9.3	EMACS – ORG-MODE.....	124
9.3.1	Iniciando Uma Página No Org-Mode.....	125
9.3.2	EXEMPLOS DE CÓDIGOS NO EMACS E PÁGINA HTML CORRESPONDENTE.	129
9.3.2.1	<u>Com índice</u>	129
9.3.2.2	<u>Sem índice</u>	131
9.3.2.3	<u>Colocando o nome do autor na página</u>	132
9.3.2.4	<u>Colocando uma imagem utilizando o link da mesma na internet</u>	133
9.3.2.5	<u>Colocando uma imagem na página que está em um diretório local (no C:jornal taty) e não um link. No caso, a imagem deve estar no mesmo diretório do arquivo .org.</u>	135
9.3.2.6	<u>Colocando um link ativo dentro de um texto.</u>	136
9.3.2.7	<u>Colocando imagem seguida de link na mesma linha.</u>	137
9.3.2.8	<u>Colocando tabelas dinâmicas, com fórmulas por campo, no org-mode.</u> ..	139
9.3.2.9	<u>Inserindo um video direto na página através de um link com código de incorporação</u>	143
9.3.2.10	<u>Inserindo áudio com o player de áudio direto na página de um arquivo que está na pasta da página html</u>	145
9.3.2.11	<u>Inserindo um video com o player de video direto na página de um arquivo que está na pasta da página html</u>	145
9.3.2.12	<u>Inserindo um link em uma imagem: usando uma imagem como botão</u>	146
9.3.2.13	<u>Inserindo programação nas páginas dinâmicas.</u>	148
9.4	PUBLICANDO A PÁGINA NA NUVEM	156
9.4.1	Transferindo Arquivos Para A Nuvem, Para A Hospedagem, Utilizando O Windows.....	157
9.4.2	Instalando Os Programas Do Putty	157
9.4.3	Transferindo Finalmente Os Arquivos Da Página Dinâmica Criada Para O Diretório Public_Html.....	160
9.4.3.1	<u>Usando o terminal com o PSCP do Putty para transferir os arquivos</u>	161

9.4.3.2	<u>O comando de transferência de arquivos entre o Windows e a nuvem na hospedagem desejada</u>	164
9.5	CRIANDO UMA PÁGINA MAIS DINÂMICA INSERINDO PROGRAMAÇÃO E COMPILAÇÃO JUST IN TIME.	166
9.6	O PROGRAMA: A PÁGINA DINÂMICA PROGRAMADA.....	168
9.6.1	Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional Windows E Org-Mod.	169
9.6.1.1	<u>Emacs na nuvem com Windows:</u>	169
9.6.2	Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Linux, Ubuntu , E Org-Mod.	174
9.6.3	Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional Android E Org-Mod.	176
9.6.3.1	<u>Acessando o servidor via SSH em dispositivos Android</u>	177
9.6.3.2	<u>Baixando da Play Store o server auditor que transforma seu celular ou tablet em um cliente SSH</u>	177
	Smartphones:	177
9.6.4	Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional Apple E Org-Mod.	186
9.6.4.1	<u>Acessando o servidor via SSH em dispositivos Apple</u>	186
9.6.4.2	<u>Baixando da Apple Store o server auditor que transforma seu celular ou tablet em um cliente SSH:</u>	186
9.7	CRIANDO UMA PÁGINA PROGRAMADA, COM EMACS E LISP, COM UMA PROVA DE CONCEITO DE TEXT MINING, INSERINDO PROGRAMAÇÃO E COMPILAÇÃO JUST IN TIME.....	192
9.8	A PÁGINA PROVA DO CONCEITO.....	194
9.8.1	Arquivos Executáveis Pelos Browsers	196
9.8.2	CRIANDO O PROGRAMA.	197
9.8.2.1	<u>Agora é só iniciar o programa.</u>	199
9.8.2.2	<u>Código html da interface</u>	202
9.8.2.2.1	<i>Cabeçalho do arquivo HTML</i>	202
9.8.2.2.2	<i>Corpo do arquivo HTML</i>	202
9.8.3	Subindo Os Arquivos Para A Nuvem.	204
9.9	CONCLUSÃO.....	206
	CONSIDERAÇÕES FINAIS	207
	TRABALHOS FUTUROS	209

REFERÊNCIAS.....	210
ANEXOS	ERRO! INDICADOR NÃO DEFINIDO.
ANEXO 1 INSTALANDO O EMACS NO WINDOWS	212
ANEXO 2 – DADOS DA HOSPEDAGEM NA NUVEM	226
ANEXO 3 – SSH	230
ANEXO 4 – TERMINAL SSH NO WINDOWS COM O PUTTY	231
ANEXO 5 – SERVIDOR E CLIENTE	236
ANEXO 6 TRANSFERIR ARQUIVOS ATRAVÉS DE UM TERMINAL PELO COMANDO "SCP" NO LINUX.....	237
ANEXO 7 EMACS LISP	241
ANEXO 8- ORG-MODE – MANUAL	252
ANEXO 9 – ATALHOS DO EMACS	342

RESUMO

Estamos na Era da Informação, onde qualquer pessoa com um smarthphone em mãos é capaz de registrar um momento e publicá-lo na internet. Isso gera um problema grave aos jornais: eles estão publicando notícias que ao irem ao ar, já estão velhas. O presente trabalho apresenta um panorama geral da situação dos jornais impressos e digitais no país, bem como propõe uma solução eficiente para capacitar o jornalista na produção de páginas jornalísticas dinâmicas direto na nuvem, com hipertextos, multimídia e textmining, multiplataforma, multidevices, utilizando paradigma funcional com emacs org-mode, e-lisp.

Palavras-chave: Jornalismo Multimídia; Programação na Nuvem; Textmining; Paradigma Funcional; Capacitação.

ABSTRACT

We're in the Information Age, where anyone with a smartphone it's capable to covers a moment, register and publish it on the internet. This creates a serious problem to newspapers: when the news go to air, are already old. This paper presents an overview of the situation of press and digital journals in the country, and proposes an efficient solution to enable the journalist to produce dynamic news pages directly in the cloud, with hypertext, multimedia and textmining, multiplatform, multidevices, using functional paradigm with emacs org -mode, e-lisp.

Key words: Multimedia Journalism; Programming in the Cloud; Texmining; Functional paradigm; Training

1 INTRODUÇÃO

1.1 ESTADO DA ARTE

Atualmente a situação dos jornais impressos é preocupante. Na última década, grandes jornais deixaram a versão impressa para existirem apenas na internet ou simplesmente se extinguiram. O Jornal do Brasil, um dos maiores jornais brasileiros foi fundado em 1881, deixou de ser distribuído no papel e passou a ser produzido apenas para a versão digital em 2010 (FERREIRA; SOUSA, 2014). Segundo as autoras, o JB foi o primeiro jornal brasileiro a criar portal na internet, e, por causa da concorrência com O Globo e altos custos para manter a publicação física, também foi o primeiro a deixar de ser impresso.

A circulação do jornal em 2008 mantinha-se em torno de 95 mil exemplares. Essa boa fase durou pouco tempo, o periódico entrou em uma nova rota de queda: em 2010, a tiragem diária caíra para apenas 20 mil exemplares. O JB possuía um gasto mensal de 3 milhões de reais por mês, além de um acúmulo de passivo estimado em 100 milhões de reais em dívidas. (FERREIRA; SOUSA, 2014, p.156).

Em 2013, o The New York Times deixa de lado a distribuição impressa no Brasil (SÃO PAULO, 2013). No fim de 2015, também para de ser distribuído na Tailândia (REDAÇÃO PORTAL IMPRENSA, 2015). As notícias que mostram a decisão do jornal novaiorquino apontam um único motivo: o alto custo da operação impressa.

Agner *et. al.* (2015), afirma que a queda da leitura do impresso é preocupante e se agrava gradativamente. Ele indica que conforme a previsão de Philip Mteyer de que em 2043, pelo *andar da carruagem*¹, os jornais impressos diários nos Estados Unidos da América podem deixar de existir. A baixa tiragem, resultante da redução de leitores, faz com que a mídia impressa seja menos interessante para os anunciantes. Melech (2012) é mais pessimista. Para ele “As projeções para o fim do jornal impresso indicam que nos Estados Unidos ocorrerá em 2017, depois na Inglaterra em 2019, em seguida Canadá e Noruega, em 2020 e em 2027 no Brasil.” (MELECH, 2012, p.57).

¹ Expressão popular que indica: pela forma que as coisas vão.

Um círculo vicioso: os jornais estão em crise por falta de publicidade, diminuem a tiragem para reduzir os custos e perdem novamente patrocínio, como mostra Barsotti *apud* Agner (2015).

Segundo Barsotti (2014), a indústria jornalística está em crise e a circulação dos jornais impressos, em queda. Nos mercados maduros, a circulação dos veículos impressos vem caindo, juntamente com as receitas publicitárias. Com uma audiência menor, os jornais já não são mais atraentes para o mercado anunciante. Nos EUA, o total de publicidade online já superou a de jornais impressos, segundo o relatório *State of the Media*, citado pela autora. No mundo, a Internet ultrapassou as revistas, tornando-se o terceiro maior meio em faturamento. No Brasil, houve pequeno crescimento da indústria jornalística em função da expansão do PIB e da entrada de títulos populares, mas esta tendência já foi interrompida. (AGNER, 2015, p.3).

De acordo com a Pesquisa Brasileira de Mídia de 2015, dos usuários que usava internet, 67% o fazem para se informar (BRASIL, 2014). O número de leitores de jornais digitais aumentou em 20% em relação ao ano passado, mas a maior parte deles consome informação pelas redes sociais. O que é alarmante, já que o que é publicado nas redes sociais não passa por um crivo de veracidade e/ou de qualidade.

A pesquisa aponta internet como o meio de comunicação que mais cresce entre os brasileiros, com acesso de um quarto da população diariamente. A intensidade média dos usuários na rede é de 3h39min, mais tempo do que os telespectadores passam em frente à TV. A mesma pesquisa mostra que 75% dos entrevistados afirmaram que não leem mais jornais impressos. A mídia mudou de forma veloz, radical e, aparentemente, irreversível (BRASIL, 2014).

Quando a pergunta é sobre os endereços eletrônicos mais acessados, o Facebook lidera com 63,6% de segunda a sexta-feira e 67,1% nos fins de semana. O ranking dos cinco mais acessados é completado pelos sites Globo.com (7%), G1 (5,6%), Yahoo (5%) e Youtube (4,9%). (VICTOR, 2014, s.p.).

Segundo Alcadipani, os jornais impressos estão entre os produtos mais perecíveis do mercado. Para o autor, a vida útil do impresso nas prateleiras é de, no máximo, 12 horas. Uma das maiores máximas dessa indústria é “ninguém paga pelas notícias de ontem”. Nas palavras de Alcadipani, “Os jornais não estão perdendo leitores para outros jornais seus concorrentes, mas sim para outras mídias.” (ALCADIPANI, 2007, p.33).

1.2 O DESAFIO DOS JORNAIS DA ERA DA INTERNET

Esta década inicia com um volume imenso de informação sendo divulgada totalmente sem critério ou formato, onde cada um quer ter a primazia da notícia sem se preocupar com a forma, ética e estética da apresentação. Assim surgiram os blogs em larga escala, por serem ágeis, sendo logo após atropelados pelo Twitter, com uma limitação de publicação de 140 caracteres.

Como em qualquer área do conhecimento, às vezes há surpresas com a criatividade com que as pessoas superam obstáculos e se adaptam às restrições. Com tanta confusão e volume de informação, a criação, a adequação dos jornais impressos para jornais digitais foi um acontecimento, de certa forma, previsível.

Os jornais migraram para a rede e tiveram que se adaptar à nova mídia. A leitura seqüencial perdeu sentido. Foi necessário colocar hipertextos² (proporcionando leituras não lineares), estimular a interatividade com o público, pôr à disposição arquivos, inserir recursos de som e imagens fixas e animadas. A publicação online requer, principalmente, a velocidade, personalização, interatividade e a individualidade. (LARANJEIRA, 1988, p. 169).

Uma das formas de fazer o público participar da produção da notícia, foi usá-lo como produtor de conteúdo, principalmente com a popularização dos smartphones e dispositivos móveis.

Além de prover acesso cada vez mais instantâneo à informação, em praticamente qualquer lugar, os suportes móveis possibilitaram ao leitor seu lugar como produtor de conteúdo. Pode-se afirmar que um novo ciclo de inovação vem sendo gerado a partir da utilização dos dispositivos móveis e pela criação e utilização de aplicativos jornalísticos. (AGNER, 2015, p.3).

A maioria dos grandes jornais hoje já vendem mais de 70% de suas edições em mídia digital (LARANJEIRA, 1988), o que é benéfico quando se fala em meio ambiente. Infelizmente, qualquer um com um celular e um blog já se julgam verdadeiros jornalistas acarretando graves danos à uma arte milenar e respeitada. Mesmo assim, é necessária e desejada a participação dos nativos digitais como jornalistas cidadãos. Segundo Zanella (2008), é interessante que o leitor participe, mas também é essencial que se descubra formas de que isso aconteça sem comprometer a credibilidade do jornal ao longo dos seus 90 anos de existência.

² De acordo com Landow (1992), o hipertexto pode ser utilizado em forma de links, torna o texto , colaborativo, e em constante produção e transformação.

A era digital e a notícia em tempo real geram a necessidade de uma adaptação tecnológica por parte dos jornalistas clássicos. Os novos “jornalistas tecnológicos” e suas notícias instantâneas carecem de conceitos, muitas vezes não tendo o compromisso com a pesquisa, com a fonte, a ética, bom senso e com a verdade. Terceiro (1997, p.184) afirma que “Os jornalistas terão de se aplicar mais à tarefa de informar bem, proporcionando notícias contextualizadas, dissecadas e valorizadas”. Isso devido à proximidade entre o consumidor e o produtor da notícia. Da mesma forma em que a internet facilita a divulgação de informações pelo público, ela também permite que a verificação dos fatos seja mais acessível e precisa.

Por outro lado, as pessoas que estavam acostumadas com um jornal tradicional vêm sentindo uma carência de formato e conteúdo que antes tinham nos jornais impressos, os quais, por perda significativa de leitores, veem algumas editorias, ou mesmo jornais inteiros, serem cortados.

Então surgem algumas questões: Os jornais digitais conseguem suprir a necessidade de consumo de notícia do leitor? As tecnologias utilizadas são apropriadas para a leitura do jornal? As tecnologias existentes são apropriadas para a escrita, formatação adequada do jornal aos diferentes leitores eletrônicos? As tecnologias utilizadas facilitam o processo de produção de notícias? As tecnologias existentes são suficientes e eficientes no processo de produção de notícias para o mundo atual globalizado?

2 OBJETIVO

2.1 GERAL

Fazer um guia definitivo, pra jornalistas e profissionais da comunicação, de construção de páginas dinâmicas e inteligentes que possam ser postadas automaticamente e rapidamente na internet utilizando qualquer dispositivo e sistema operacional que tenha acesso à nuvem (seja Windows, Linux, Macintosh, android, ios ou tablet).

2.2 ESPECÍFICOS

- Analisar comparativamente os elementos gráficos e de conteúdo dos jornais impressos e digitais no Brasil.
- Fazer um levantamento teórico e histórico sobre a transição tecnológica do jornalismo impresso para o digital.
- Realizar prova de conceito que mostre o funcionamento do sistema de publicação on-line multidevice e em tempo real.

3 JUSTIFICATIVA

Essa dissertação tem como motivação a baixa venda dos jornais impressos, devido à evasão de anunciantes. Só que isso é movido por diversos fatores, dentre eles o fato de, com a instantaneidade da publicação da informação na internet, quando os jornais colocam a notícia no ar, a informação já é considerada velha e foi “furada”³ por um internauta qualquer.

Como isso pode acontecer? Hoje qualquer pessoa que tenha um smarthphone com uma câmera é capaz de registrar um acontecimento e logo publicá-lo em um blog, instagram, facebook, etc.. O único problema é que o fato é divulgado por esse internauta sem quaisquer compromissos com a verdade, legislação, ética e com o bom português. Não há preocupação com as regras básicas do jornalismo ou com os critérios de noticiabilidade apontados por Wolf (2003)⁴. O fato é transmitido de acordo com a vontade do próprio internauta, em primeira mão, e, muitas vezes, sem a devida qualidade.

Nessa situação, uma coisa não pode ser evitada: o fato já foi publicado e o jornal irá divulgar posteriormente uma notícia melhor estruturada e apurada, mas sem o ineditismo. Qual seria então o motivo de as pessoas continuarem a consumir as notícias dos jornais se elas encontram toda a informação devida no feed de notícias das redes sociais? Somente a credibilidade já não basta ao jornal. Depois de divulgada uma informação, poucos irão atrás de aprofundamento dela. Mesmo que seja inverídico, muitos usuários da internet compartilharão o post, basta que seja interessante o suficiente para fazê-lo.

O verdadeiro problema encontrado para jornalismo na era digital, é que por mais que os jornais online denominem-se em “tempo real”, são passados para trás pelos usuários de smarthphones que estão 100% do tempo conectados. Então qual seria a solução? Uma delas seria a velocidade. É muito importante para um jornal que ele consiga, no mesmo intervalo de tempo que uma pessoa qualquer divulgar uma notícia. É preciso que o jornalista seja mais rápido do que o usuário comum da internet para noticiar um fato, e, para isso, ele precisa saber utilizar os

3 “Furo” no jornalismo é o jargão usado para quem noticiou o fato em primeira mão, antes de qualquer veículo ou pessoa.

4 “A noticiabilidade é constituída pelo complexo de requisitos que se exigem para os eventos – do ponto de vista da estrutura do trabalho nos aparatos informativos e do ponto de vista do profissionalismo dos jornalistas –, para adquirir a existência pública de notícia.” (WOLF, 2003, p.195).

artifícios que a rede mundial de computadores criada por Lee⁵ possui, com a tecnologia que tiver em mãos. O processo deve ser rápido e simples, sem que seja necessária a mudança de aparelho tecnológico. Jornais são estruturas complexas, com servidores programados em Macintosh e Linux, sendo que os próprios jornalistas desconhecem a possibilidade desses sistemas. As faculdades brasileiras de jornalismo muitas vezes não possuem o recurso necessário para capacitar o estudante a utilizar a tecnologia, tornando o jornalista obsoleto em sua própria formação.

O que move essa dissertação é justamente capacitar o jornalista, os editores e todos os profissionais dos jornais responsáveis por postar a notícia, publicar a notícia na web, publicar a notícia no meio digital, a fazê-lo na mesma velocidade e facilidade que um internauta faça um post. Só que vem um problema, veja a situação: o jornalista está em grande parte do tempo na rua, e o que tem em mãos é o próprio celular. Muitas vezes o aparelho fica sem bateria e o jornalista precisa conseguir outro aparelho, seja emprestado ou em algum recinto próximo, para não perder o furo. O hall de aparelhos que o jornalista pode conseguir é imenso: android, iphone, Windows Phone, ou qualquer notebook (Windows ou Linux), cada um com sua particularidade de uso. Nesse momento o profissional precisa ser capaz de usar o que estiver à disposição, com a melhor qualidade e o mais rápido possível. Para que isso possa ser feito, uma tecnologia que é de conhecimento de todo jornalista, desde que a imprensa foi criada na Grécia, é a escrita.

Desde o Ensino Fundamental, aprende-se a escrever. Na faculdade, o jornalista passa a ter conhecimento de formalismo e técnicas de escrita em pirâmide invertida⁶, entre outras. Durante as disciplinas, é ensinado ao profissional ética, história, sociologia, e diversas teorias consideradas importantes para a formação de um bom jornalista, independente do veículo que for trabalhar. O que acontece é que os jornais impressos, de TV ou rádio estão concorrendo com o mercado digital, e as próprias faculdades não conseguem acompanhar a evolução tecnológica para formar jornalistas voltados ao mercado online. Tentar concorrer com o mundo e com

⁵ A World Wide Web (W.W.W.) foi lançada em 1989 por Tim Bernes Lee e foi o marco inicial de popularização da internet. (FERRARI, 2004).

⁶ “A técnica da pirâmide invertida pode resumir-se em poucas palavras: a redacção de uma notícia começa pelos dados mais importantes – a resposta às perguntas O quê, quem, onde, como, quando e por quê – seguido de informações complementares organizadas em blocos decrescentes de interesse.” (CANAVILHAS, 2006, p.5).

o mercado digital sem estar preparado, é suicídio. Com os jornais caminhando cada vez mais integralmente para a internet, é necessário saber utilizar essa tecnologia e dotar qualquer jornalista, desde o ingresso na faculdade à funcionários de jornais, a usarem uma tecnologia simples de postagem, no caso da dissertação, utilizando o texto. O texto redigido pelo jornalista vira uma página HTML com todas as mídias (imagens, vídeos, sons, etc.), hipertextos, melhor aproveitamento da multimídia e até mesmo text mining.

O que seria o text mining para o jornalismo? Seria o jornalista, dentro da própria notícia, construir um programa que possa buscar dentro do texto uma informação que seria trabalhosa de se encontrar a olho nu. Isso se chama mineração de dados, no caso de um texto chamado text mining. Veja o exemplo: O jornalista escreve uma matéria sobre os celulares mais utilizados pelos jovens em 2015. Então, ele precisa criar uma tabela que organize por tamanho os aparelhos. Lembrando que o jornalista hoje está em uma corrida contra o tempo, ele precisa fazer isso o mais rápido possível. São centenas de aparelhos, e um jornalista só. Dotado do conhecimento apresentado nesta dissertação, basta que crie, em text mining, um programa que faça a busca e já coloque a informação no local desejado. Esse diferencial, nem o internauta que está plugado 24h/dia consegue com facilidade.

Além de capacitar e concorrer com a notícia estática, o jornalista tem, com esse trabalho, a possibilidade de páginas dinâmicas. A própria página consegue extrair informações, buscar informações para completar a notícia numa velocidade muito maior do que o usuário comum.

É o que intenta o presente trabalho: facilitar e agilizar o processo de postagem de notícias, em qualquer local e com qualquer tecnologia. A ideia é simples, se o repórter tiver acesso às informações de hospedagem do jornal, ele já consegue colocar a notícia no ar, caso não tenha, já deixa a matéria pronta para ser publicada pelo editor. Tudo fazendo uso da estrutura a ser apresentada posteriormente na presente dissertação.

Então o trabalho vai além de tentar fazer com que o jornal possa vender mais e ter anunciantes. Caso ele consiga publicar a notícia em primeira mão, com mais qualidade, mais informação, automaticamente a notícia dele vai ser mais lida, os anunciantes vão voltar a patrocinar esse jornal, e as vendas e receita do veículo aumentarão.

Como sempre, a educação é o primórdio de toda a ciência. Se a pessoa é bem educada e tem conhecimento, o avanço social, financeiro, são consequências. Tudo começa pela educação, uma boa formação. O trabalho pretende construir esse tipo de conhecimento.

4 METODOLOGIA

A contextualização no domínio do conhecimento no qual este trabalho está centrado é fundamental para a capacitação conceitual e prática na aquisição das habilidades e competências necessárias para o desenvolvimento e conclusão dos objetivos.

O trabalho, pela complexidade de encontrar um profissional do jornalismo que tenha conhecimento tecnológico e de programação suficiente para que não dependa de técnicos de informática para publicar suas notícias, bem como a dificuldade dos jornais se adaptarem e estarem à frente dos internautas na divulgação de fatos, carece de uma estrutura metodológica e de pesquisa que permita uma visão qualitativa dos porquês, bem como quando, da atuação de tecnologias que permitam, de forma simples e rápida, ao jornalista publicar e alterar a página em tempo real a partir de qualquer dispositivo nestes domínios. Assim como estrutura metodológica quantitativa no sentido de se avaliar e determinar o onde, como e o quanto atuar de forma parcimônia e ortogonal⁷, para que os resultados obtidos das ações efetivadas não sejam arbitrários, que surta o efeito desejado.

4.1 TIPO DE MÉTODO DE ABORDAGEM

Conforme descrito anteriormente, o tipo de abordagem desta pesquisa, desde a coleta, análise e interpretação dos dados será Qualitativa e Qualitativa.

4.2 PESQUISA

A pesquisa será inicialmente exploratória na busca dos saberes e dos agentes que compõem o cenário, o vasto domínio onde o tema está centrado, passando a descritiva após análise e interpretação dos dados e descritiva pelas conclusões das ações que se deve implementar para concretização dos objetivos.

⁷ https://en.wikipedia.org/wiki/Occam%27s_razor#cite_ref-24 **Navalha de Ockham**: *Entia non sunt multiplicanda praeter necessitatem*. Essa *lex payrsimoniae*, quando efetivada, permite minimizar e até eliminar o risco de resultados arbitrários, inesperados e indesejáveis.

4.3 O DELINEAMENTO DA PESQUISA DA DISSERTAÇÃO

Inicialmente o delineamento da pesquisa da dissertação acontece através de coleta de dados documental e bibliográfico, com estudos de casos reais obtido pelo estudo de 40 jornais impressos e online no Brasil entre os estados de São Paulo, Rio de Janeiro, Minas Gerais, Distrito Federal e Goiás. Com posterior tratamento de dados de acordo com circulação, se existem as versões online e impressa, se são diário/tempo real, os canais de interação, padrão de cores, número de colunas, editorias e forma de contato. Seguido de estudo e análise de linguagem ideal para publicação e programação na nuvem *just in time*.

Essa primeira etapa documental tem por objetivo a organização e interpretação que culminem por apresentar soluções, métodos e ações que minimizem uma possível arbitrariedade na análise dos dados, focando em decisões ortogonais⁸ e com *lex parsimoniae*.

4.3.1 Recursos Necessários À Pesquisa:

Os recursos disponibilizados pela Universidade Federal de Uberlândia, detentora de uma completa estrutura composta por um corpo docente especializado e acesso a equipamentos de primeira linha, foram suficientes e potencialmente capazes e competentes para suprir conceitualmente e tecnicamente as pesquisas necessárias para efetivação deste projeto de pesquisa, já elencadas no corpo deste documento.

Em complemento ao suporte existente na Universidade Federal de Uberlândia, a autora desta dissertação já possuía experiência no domínio do trabalho, assim como contatos especializados na área que corroboraram para viabilizar a efetivação da dissertação de mestrado. Assim, nenhum recurso adicional foi necessário para a efetivação da pesquisa, além dos supracitados.

⁸ Análise e uso de soluções que não sejam ambíguas ou conflitantes. Da matemática, a ortogonalidade implica em se possuir axiomas cujas regras de inferência não permitem deduzir nenhum deles a partir dos outros, e, com isto, elimina-se a ambiguidade e arbitrariedade nos resultados.

5 CONCEITOS E TÉCNICAS SOBRE JORNAIS IMPRESSOS E DIGITAIS (WEB JOURNALS)

5.1 BREVE HISTÓRIA DO JORNALISMO

A mídia impressa é um meio de comunicação tradicional que há séculos é utilizado para divulgar notícias e informações para as massas. Em Roma, cerca de 59 A.C., existe o relato do jornal mais antigo do mundo, o *CaesarActa Diurna* (BARBER, 2002). Júlio César divulgava todos os eventos programados nas principais cidades, escritos em placas brancas e expostos em locais públicos e populares. Homens levavam as placas de maneira à cavalo, e, muitas vezes, quando chegavam aos locais já eram notícia velha.

Na China do século VIII, os primeiros jornais surgiram em Pequim sob a forma de boletins escritos à mão (JORNAIS, s.d.). O processo era lento, demorado e sem possibilidade de ser produzido em larga escala, por ser manuscrito.

Até que em 1447 Johann Gutenberg modernizou a imprensa. A prensa de papel, construída com base na tecnologia dos tipos móveis e também da prensa de vinho permitiu que Gutenberg criasse uma nova indústria (BARBER, 2002). A máquina de Gutenberg possibilitou o livre intercâmbio de ideias e a disseminação do conhecimento através de boletins informativos, que levavam a uma classe cada vez maior de comerciantes notícias de interesse sobre o mercado.

A invenção de Gutenberg é apontada por alguns autores como revolucionária, marco que segundo eles tirou o mundo de vez da Idade Média, levando-o para a Era da Renascença, com o despertar definitivo da ciência e do jornalismo profissional (LINDOO, 1998). O primeiro livro a ser produzido, lançado e vendido em escala industrial através da prensa de Gutenberg foi a Bíblia de Gutenberg⁹, que antes era acessível a poucos nos mosteiros – antes da criação da prensa, o processo de reprodução da bíblia era lento, pois era todo feito a mão em mosteiros.

Outra tecnologia que transformou a imprensa escrita foi a invenção do telégrafo, em 1844. O processo de transmissão de informações torna-se mais ágil,

⁹ A Bíblia de Gutenberg foi o primeiro livro impresso da Europa, com tiragem de 200 cópias, que hoje são raridades (só restam 49, das quais apenas 21 estão completas). A obra foi digitalizada integralmente e está disponível no website da Universidade do Texas em Austin: <http://www.hrc.utexas.edu/exhibitions/permanent/gutenberg/> (14/12/15).

as notícias são então passadas em questão de minutos, o que permitiu relatos mais atuais e relevantes (BARBER, 2002).

Segundo Barber (2002), em meados do século XIX, os jornais se tornaram o principal veículo de divulgação e recebimento de informações. Os jornais também ajudaram na divulgação de propaganda evolucionária.

A imprensa viveu seus anos dourados entre 1890 e 1920, até o surgimento do rádio, que explodiu no cenário da mídia nos anos 20 (BARBER, 2002). Os jornais foram obrigados a reavaliar seu papel como principal fonte de informação da sociedade, e correram o risco de desaparecer.

A conversa de que “o jornal impresso vai morrer” é antiga. Nessa época, como o rádio era uma mídia mais barata e uma alternativa de transmitir informações, temia-se que ele substituísse o jornal (JORNAIS, s.d.). Porém os jornais reagiram à concorrência, renovaram os formatos e conteúdos para torna-los mais atraentes. Por muito tempo o jornal impresso pautou os noticiários dos rádios, que só liam no ar as últimas notícias publicadas no diário da manhã.

Assim que os jornais conseguiram se adaptar à novidade do rádio, viram-se obrigados a fazer uma auto avaliação à luz de um novo e poderoso veículo: a televisão. Entre 1940 e 1990, a circulação de jornais nos EUA caiu quase pela metade. Apesar da queda brusca, a televisão não tornou o jornal obsoleto. A resposta aos avanços tecnológicos foi usar a cor na impressão e mediante artigos curtos, rápidos e objetivos, adaptados do modelo televisivo (BARBER, 2002).

A atual revolução tecnológica gera novos desafios e oportunidades para a mídia tradicional. As informações estão disponíveis em tempo real para um número um grande número de pessoas.

No final dos anos 90, havia cerca de 700 sites na Internet, hoje se contam aos milhares (ARNT, 2002). O volume e a atualização de informações na Internet não têm paralelo, mas isso não decretou o fim da relevância dos jornais. Isso faz com que os jornais impressos tenham que se adaptar rapidamente com a nova tecnologia, dessa vez sim podendo se tornar obsoletos, já que as notícias publicadas nos jornais já se tornam velhas no fechamento de cada edição.

Segundo Arnt (2002), sociólogo e pesquisador do Centro Nacional de Pesquisa Científica da França (CNRS),

No primeiro momento, as novas tecnologias serviram tão somente para modernizar o processo industrial e dinamizar as redações (pela substituição

de velhas máquinas de escrever por computadores), numa segunda etapa, a tecnologia vai facilitar a comunicação interna, entre os diversos setores do jornal. Quando os jornais começaram a fazer edições online não sabiam para onde iam, nem por que o faziam, mas tinham a intuição de que se não fizessem acabariam por desaparecer. Hoje, pode-se falar de um jornalismo digital, que amplia, redobra, multiplica o potencial do jornalismo impresso. (ARNT, 2002, p.1)

5.2 COMUNICAÇÃO VISUAL NOS JORNAIS

Desde os primeiros impressos as artes gráficas já estão presentes. A Bíblia de Gutemberg, por exemplo possui páginas cuidadosamente diagramadas e ilustradas. Com os avanços das técnicas de impressão, as letras receberam melhores acabamentos, enquanto o design gráfico começava a influenciar o conteúdo.

Com a correria diária, a linguagem visual passou a ser indispensável, por agilizar a troca de informações. “O jornal impresso transforma-se em nossa época em um festival de signos e ícones, buscando atrair e estimular a atenção dos consumidores” (MARSHALL, 2003, p. 49). O jornalismo online segue o mesmo padrão, sendo mais gráfico e menos textual, sem comprometer a qualidade da informação noticiada.

A contrariar a ideia de Gutemberg, os primeiros jornais impressos não demonstravam muitas preocupações quanto ao design e o planejamento gráfico. Em grande maioria, a diagramação contava com grande quantidade de letras em fontes pequenas, blocos compactados no decorrer das páginas e separados por linhas muito próximas ao texto (MARSHALL, 2003).

A tecnologia foi avançando e o custo de impressão diminuindo. Isso fez com que os jornais aumentassem o número de páginas e utilizassem imagens com mais frequência. A organização se tornou agradável, a linha de divisão foi substituída pelo espaço em branco e os textos passaram a utilizar fontes e blocos maiores.

Atualmente, os poucos jornais impressos já são coloridos e procuram utilizar constantemente recursos gráficos, em tabelas e infográficos, que contribuem para a compreensão de determinadas informações. Na internet a informação se complementa, com infográficos interativos, News games e diversos recursos que buscam atrair a atenção da nova geração.

Para não perder leitores, jornais como O Globo e Folha de S. Paulo construíram na década de 90 novos parques gráficos para acompanhar as tecnologias internacionais (FILHO, 2006). Isso obrigou os jornalistas a aperfeiçoar os conhecimentos em artes gráficas e nas plataformas tecnológicas de publicação, para atender um público leitor apressado e exigente.

A preocupação de tornar a leitura mais rápida foi constante na década de 90. As imagens passaram a substituir extensos relatórios e tabelas por gráficos de fácil interpretação, fornecendo instrumentos eficazes que permitem aos jornalistas publicarem mais informações e – consequentemente – de melhor qualidade.

5.3 ESTRUTURA DO JORNALISMO NA WEB

O estilo de texto para a internet deve ser curto, na ordem direta, com palavras-chave destacadas, em bloco de cerca de cem palavras, no máximo. A diagramação é linkada e multimidiática. Uso extensivo de hiperlinks, listas com bolinhas e entretítulos (FILHO, 2006).

O internauta lê de forma diferente. Ele apenas passa os olhos pelas telas à procura de palavras-chave. Não possui paciência para textos elaborados ou longos. Uso extensivo de imagens, infográficos e cores (FILHO, 2006). No caso do G1, as editorias do portal de notícias são separadas por cores. Como exemplo, a editoria de política é destacada com a cor vermelha, entretenimento com azul ou amarelo e esporte em verde. Isso facilita ao internauta localizar a informação.

Alguns jornais digitais possuem um padrão semelhante ao impresso, e acabam perdendo o mercado para os que se adaptam à linguagem online, tanto visual quanto escrita.

5.4 A TECNOLOGIA E O JORNALISMO

A inovação tecnológica não é vilã dos meios de comunicação. A internet banda larga permitiu a criação de novas ferramentas de publicação e de estruturação dos webjornais. O diferencial da rede é a interação entre o leitor e meio de comunicação. Hoje, além de comentários, envio de e-mail, o usuário tem uma diversidade de opções. Moherdau (2008) cita algumas:

Adicionar a sites de bookmarks (favoritos), escolher o melhor conteúdo, eleger critérios de credibilidade por meio de votação, RSS (Really Simple Syndication), newsletter, verificar estatísticas de acesso, postagem de conteúdo via blog, fotolog e sistemas de open source, como Wikipedia, por exemplo, e melhorar o desempenho da busca por meio de palavras-chave inseridas no texto, entre outros. (Moherdau, 2008, p.2)

Para a autora, utilizar geradores de conteúdo com interfaces amigáveis, com ferramentas de edição de áudio, vídeo, imagem, flash, slide show, permite realizar todo o processo de produção em uma redação online. O celular também pode ser visto como recurso de produção, aprimorando a estrutura da notícia.

Mesmo com a tecnologia em mãos, como tablets e smartphones, a produção da notícia muitas vezes fica refém da estrutura da redação. Com isso, os jornalistas que desconhecem as plataformas móveis ficam com notícias “velhas”, principalmente com o imediatismo da informação na internet. A cobertura noticiosa passa a ser em tempo real, e quem não o faz fica obsoleto.

Para Moherdau (2008, p.7), os “formatos de mídia não são apenas canais de informação, e o designer tem de enxergar o computador como uma nova mídia que permite transformar a experiência multimídia.”.

5.5 OS NATIVOS DIGITAIS E O JORNALISMO

Pedro Pinciroli Jr, engenheiro e diretor industrial da Folha de S. Paulo na década de 90, não vislumbrou a grandiosidade que aguardava o jornalismo com o surgimento da internet. Na época a única preocupação era tornar a informação mais visual, para otimizar o consumo de notícia. Segundo ele:

Esse uso crescente de gráficos tem uma explicação lógica: leitores do mundo inteiro têm o tempo cada vez mais curto. Todo leitor é impaciente e quer diminuir o tempo que ele emprega na leitura. Então, para diminuir esta impaciência, nós temos de dar informação visual, complementando a escrita. Não podemos nos esquecer que estamos na geração que nasceu vendo televisão (PINCIROLI JR Apud VIANNA, 1992, p. 38).

Isso porque Pinciroli concedeu essa entrevista em um período anterior à explosão da internet. Hoje temos uma geração de nativos digitais que estão acostumados com uma grande quantidade de informações visuais e pouco texto. O desafio para o jornalista nas mídias digitais atuais ultrapassa as barreiras da competição gráfica, e entra na instantaneidade da publicação e concorrência com um imenso hall de sites e blogs para não perder o “furo”.

No período de Pinciroli, esperava-se que a geração Y seguisse os padrões da geração X. Ávidos por informação e sem tempo. Segundo Chiovato, a geração atual é dos nativos digitais.

a geração Y e outras vislumbradas como o futuro da humanidade se resumem atualmente nesta especificação: Nativos Digitais. O que muda é que, já desde bebê, o novo ser humano, moldado pela sociedade e respectiva tecnologia, desde o berço passa a ter contato com sistemas computacionais, onde se encontra nos berços, não adaptado, mas nativo, um suporte para tablet, a primeira babá. Assim, já nascem os bebês em contato com uma metáfora diferente do que é um livro e de como manuseá-lo. Os personagens ganham vida e voz e respondem a alguns comandos, interagem. Desde cedo, portanto, passam a ter contato com interfaces computacionais interativas e multimídia. (CHIOVATO, 2015, p.16).

Na internet, o jornal precisa ser mais atrativo. É necessário que esteja reinventando-se e adaptando-se ao novo consumidor pertencente à geração dos nativos digitais. Além de ser visualmente agradáveis, as informações precisam ser multimidiáticas e suprir a necessidade de conhecimento do leitor. O desafio está em ser comercializável e não comprometer a essência da produção jornalística.

Filho, 2004, estabeleceu alguns critérios da produção jornalística, que apesar de terem sido publicados há mais de dez anos ainda são atuais. Ele se baseou segundo os critérios utilizados pela Society for News Design para avaliar e premiar os jornais de melhor aspecto visual no mundo, estabelecidos por Jim Jennings e publicados no Brasil por Ricardo Noblat no Correio Braziliense de 03/08/99. De acordo a conclusão de Filho:

O conteúdo constitui a parte fundamental de um jornal. Embora o design desempenhe um papel importante para impulsionar as vendas, podemos considerá-lo apenas como embalagem e não como produto.
Toda publicação atende a um determinado segmento do mercado. Sua produção deve ser direcionada a este público, visando criar uma identidade que agrade sua principal fatia de consumidores.
Todas as páginas devem possuir um ritmo coerente com os padrões da publicação. O design das páginas internas também merece atenção, assim como todos os cadernos do jornal.
Fotos sem conteúdo não agregam valor no jornalismo, sendo consideradas perda de espaço. As fotos precisam estar coerentes com as notícias que acompanham e ajudar o leitor compreendê-las.
A participação do leitor no processo jornalístico é fundamental. A redação deve estudar formas de entrar em contato com os leitores e procurar atender suas expectativas.
É necessário produzir material diferenciado, sem medo de correr riscos ou fugir dos padrões. A criatividade pode fazer diferença nos pequenos detalhes da publicação, seja em títulos, legendas ou imagens. (FILHO, 2004, p.95).

Os jornais digitais de hoje estão constantemente fazendo alterações no layout dos sites, sem comprometer os princípios supra estabelecidos por Filho. Como exemplo, tem-se o portal de notícias pertencente a um dos grandes grupos de mídia do país, o G1. Esse portal sofreu diversas alterações, tanto em layout como na forma de passar o conteúdo. Foi criado no final de 2015 o “Minuto G1”. Trata-se de um espaço de um minuto na programação de TV da rede Globo onde um jornalista mostra as principais manchetes do mundo, publicadas no G1. Para se aproximar do novo público de nativos digitais, o programa quebra a figura do jornalista formal de terno e gravata e substitui por jovens vestidos com as últimas tendências da moda para apresentar a “pílula” de notícias.

5.6 A PROPAGANDA E OS JORNAIS

A produção de notícia já esteve concentrada em grandes grupos midiáticos. Basicamente, quatro “famílias” nacionais ditavam as regras do mercado de opinião - Globo, Abril, Folha e Estado. Estes grupos também controlavam a maior parte da verba publicitária. Grande parte da fatia da propaganda nacional era destinada aos veículos que proporcionassem mais audiência. As TVs abertas eram as mais beneficiadas, por chegar a quase 100% dos lares brasileiros.

Com o surgimento da internet, os meios de comunicação passaram a migrar-se para o universo online. Devido à instantaneidade da comunicação, a forma que os leitores consomem notícias mudou. Passa a ser mais atrativo entrar em um site ou blog com informações em tempo real do que esperar o jornal do dia seguinte nas bancas. A produção “instantânea” de notícias passa a tomar o mercado do jornalismo tradicional, o que afeta também o investimento publicitário.

Segundo o Ibope Media, são 94,2 milhões de internautas, sendo que 50,7 milhões acessam regularmente a internet. Diariamente são 38% conectadas à rede mundial de computadores, e o Brasil ocupa a 5ª posição no país mais conectado.

O maior anunciante nos primeiros seis meses de 2014, a Unilever, aplicou este ano menos R\$ 528 milhões em anúncios (um corte de 25% corrigindo os valores pelo IGP-M). A Nestlé cortou R\$ 194 milhões (menos 37,3%). As duas maiores cervejarias, cortaram juntas R\$ 579 milhões (cortes de 30,5% e 41% respectivamente). Três grandes bancos que estão na lista dos 30 maiores anunciantes (Caixa Econômica Federal, Itaú e Bradesco) cortaram R\$ 495 milhões. A lista segue, com cortes significativos (e contundentes) de Petrobras, Volkswagen, GM, Fiat, Tim, Pão de Açúcar.... (Stephanowitz , 2015, s.p.)

Isso mostra que o setor passa por mudança de época e de hábitos. TVs abertas, jornais, revistas e rádios perderam fatias do mercado publicitário para a internet e para mídias mais segmentadas. Ibope Media ressalva que o meio internet registra um crescimento de até 32,9% no último ano.

Para Abel Reis, presidente da Agência Click, que só trabalha no meio digital, essa é uma mídia "popular no sentido mais genuíno e atinge parcelas da população que não estão na pirâmide social e econômica". Ele calcula que muitas marcas hoje destinam até 50% do orçamento publicitário para as mídias digitais.

O número de pessoas que usam o smartphone para ter acesso à internet continuou em crescimento no Brasil e chegou a 72,4 milhões no segundo trimestre de 2015. [...] A partir do total de usuários de smartphones apurado no segundo trimestre de 2015, a Nielsen IBOPE realizou no mês de julho uma pesquisa para identificar quais os aplicativos campeões de uso entre os brasileiros. As redes sociais e os aplicativos para comunicação continuaram predominando entre os apps mais populares. (ABERJE, 2015, s.p.).

Esses números revelam um público alvo para a publicidade muito maior que dos grandes veículos de comunicação. Com os smartphones, as pessoas estão 100% do tempo conectadas à internet, e recebendo informações novas a todo instante. As agências de propaganda começam então a investir no mercado online, onde consegue mensurar a eficácia de sua propaganda. Com a mídia tradicional, não era possível determinar se o comercial foi assistido, e nem se a propaganda de fato foi convertida em venda. A internet veio para mudar e aprimorar as formas de se fazer propaganda.

Uma pesquisa de mercado sobre investimentos publicitários do Ibope Media mostra que o faturamento com anúncios nos meios TVs abertas, jornais, revistas e rádios somados caiu 8,5% no primeiro semestre de 2015 em relação ao mesmo período de 2014. O meio mais atingido foi o de revistas, com queda de 20,9%. Novamente mostrando o impacto na mídia impressa.

Em uma participação na Conferência Internacional News Media Association 2015, o diretor da Casa Editorial El Tiempo, maior jornal da Colômbia, disse que quem não evoluir vai morrer. Em entrevista ao jornal Valor Econômico, os irmãos Roberto Irineu Marinho (presidente), João Roberto Marinho (vice-presidente) e José Roberto Marinho (vice-presidente), donos do Grupo Globo, afirmaram que o jornal impresso vai morrer e que todos os jornais serão digitais.

Os próprios diários brasileiros sofrem com a redução de suas receitas. Começam a encolher as publicações, demitir em massa, despertando dúvidas crescentes sobre sua capacidade de entregar notícias, análises e opiniões relevantes.

O marketing também sofre alterações, principalmente com o novo público alvo da geração Y e dos nativos digitais. França, 2015, coloca em questão o desafio de direcionar a publicidade em uma rede globalizada, e traz um novo termo, a “glocalização”.

A globalização não é mais uma novidade, contudo, outro processo que tem uma relação intrínseca com o primeiro é a “glocalização” esta é a palavra que tem causado mudanças profundas no modo de agir e pensar do mercado. Trabalhar com pensamento global, porém com foco naquilo que acontece localmente, no território de atuação. (FRANÇA, 2015, p.1).

Quando se conhece bem o cliente e lhe oferece um produto/serviço específico e único em termos de atender a suas necessidades e desejos mais distintos, pouco aproveitados pela concorrência, oferece-lhe conveniência, uma ótima relação custo-benefício e um relacionamento tão mais profundo que dificilmente a concorrência migra para o mesmo campo e dificilmente o cliente se sente disposto a mudar de marca. O cliente pode até pagar mais (KOTLER, 1998). O que ele quer é justamente isso: sentir-se único e realmente importante para as marcas, que lhe dão exatamente o que buscam e precisam.

5.7 *ESTRUTURA DE PRODUÇÃO DE UM JORNAL*

Um jornal, seja ele online ou impresso, é composto por diversas funções, podendo ser compostas por um jornalista ou não. Limitando-se à parte de produção jornalística, sem levar em consideração os trâmites empresariais convencionais como o RH, Financeiro e comercial, Moraes apud Bueno 2011 divide a atividade jornalística nos seguintes objetivos: Informar ao público; Iniciar o processo de informação; Coletar informação; Registrar informação; Qualificar informação; Atualizar as informações; Comunicar-se; Demonstrar competências pessoais.

Bueno descreve os jornalistas como profissionais que

Recolhem, redigem, registram através de imagens e de sons, interpretam e organizam informações e notícias a serem difundidas, expondo, analisando e comentando os acontecimentos. Fazem seleção, revisão e preparo definitivo das matérias jornalísticas a serem divulgadas em jornais, revistas,

televisão, rádio, internet, assessorias de imprensa e quaisquer outros meios de comunicação com o público. (BUENO, 2011, s.p.)

A produção jornalística é dividida em etapas, mais comumente definidas como elaboração de pauta, redação e edição. A grande tendência é que o repórter se torna cada vez mais responsável por todas as etapas de produção da notícia. De acordo com Caldas, 2002, isso deve-se à revolução tecnológica que tornou os processos mais rápidos, fazendo as notícias serem publicadas praticamente em tempo real.

Funções foram extintas, atividades reagrupadas e redefinidas, o tempo da notícia mudou. [...] o repórter se desdobra em três ou quatro, podendo chegar ao infamante provedor de conteúdo, editores e subs desmembram-se em pauteiros, fechadores. [...] Sob a pressão do tempo, esse inimigo mortal, é preciso fechar cada vez mais cedo. (CALDAS, 2002, p.14).

Voltando às etapas de produção de notícias, tudo tem início na pauta. Ela determina quais os assuntos serão tratados na edição (número) do produto em questão. Também mostra a importância do tema e dá apoio ao jornalista em relação a abordagem do tema, contendo dados, fontes e perguntas. Nas redações tradicionais da década de 90, existia um profissional responsável por isso, o pauteiro. Hoje algumas redações possuem produtores que elaboram as pautas, mas a grande maioria acumula a função do repórter com pauteiro.

Após receber ou produzir a pauta, o repórter faz a checagem dos dados, etapa denominada de “apuração”. Nessa fase o jornalista entrevista as fontes, pesquisa e analisa o tema para transmitir a informação de uma forma precisa e correta.

A redação é o que o próprio nome da etapa diz: transformar as informações colhidas e apuradas em texto. Tudo feito de acordo com a linguagem jornalística. Hoje não há mais espaço para o redator nos jornais. O próprio jornalista também fica encarregado dessa função.

Cabe ao editor aprovar, reescrever, cortar textos, legendar, classificar a editoria e importância e mandar um feedback ao jornalista. Não é o editor quem distribui os trabalhos aos repórteres, essa é a função do chefe de reportagem. Em alguns jornais, ele seleciona e distribui as pautas.

Outra função existente nas redações atuais e antigas é o planejador gráfico. Ele produz todo o material imagético dos jornais (infográficos, fios, selos, ilustrações, disposição de fotos, tamanho de texto e fontes). Um profissional com

funções semelhantes é o diagramador. Em pequenos jornais, o último faz o papel dos dois.

O setor artístico do jornal também possui um responsável: o editor de arte. Ele cuida da apresentação, mantém o projeto gráfico do jornal, escolhe junto ao editor de opinião as charges do dia e vê onde e se as imagens produzidas pelo planejador gráfico irão entrar na edição.

Já o editor de opinião, como o próprio nome diz, é responsável pelos textos opinativos no jornal. Em pequenos jornais, o editor chefe faz a função do de arte e do de opinião também.

Os jornais também possuem colaboradores diários ou eventuais, que podem ou não serem jornalistas. Eles são denominados colunistas ou articulistas, e normalmente são pautados pelo editor de opinião.

6 OS JORNAIS IMPRESSOS E DIGITAIS NO BRASIL – UMA ANÁLISE COMPARATIVA

A pesquisa serve de subsídio para a realização do projeto de aplicação o qual foi materializado, como prova de conceito. Antes de gerar qualquer produto, foi necessário primeiramente realizar uma pesquisa dos jornais existentes em versões impressas e digitais, a fim de conhecer o estado da arte.

Um fator relevante a ser verificado em tais veículos de comunicação é a periodicidade das publicações, sendo que um maior foco será dado às publicações digitais que ocorrem em frequências maiores que as impressas, quase na velocidade com que os fatos ocorrem.

Foi analisado, junto aos jornais, os recursos existentes e quais as possibilidades de modificação e personalização, seja automática ou manual, on line e off line que cada jornal tem disponível, permitindo levantar o tempo de atualização e atuação do jornal no universo online, onde o leitor está plugado e se atualizando em tempo real, o dia todo.

Foi realizado um levantamento dos formatos de jornais impressos e digitais existentes e de expressão, veiculação comprovada, formando um banco de dados formatados de forma a se poder extrair informações úteis da padronização de postagem dos jornais online e a adaptação do impresso para a internet.

Com base no banco de dados montado com as pesquisas nos diversos jornais é possível relatar certas características gerais e específicas desses meios de comunicação. Ao todo, foram analisados 40 jornais online entre os estados de São Paulo, Rio de Janeiro, Minas Gérias, Distrito Federal e Goiás, e observados os aspectos de circulação, se existem as versões online e impressa, se são diário/tempo real, os canais de interação, padrão de cores, número de colunas, editorias e forma de contato.

6.1 CIRCULAÇÃO

No aspecto da circulação o observado é que no estado de São Paulo, os 17 jornais analisados se dividem em circulação nacional, estadual e municipal. Os jornais Folha de S. Paulo, Estadão, Valor Econômico, Brasil Econômico, DCI (Diário, Comércio, Indústria & Serviços), DC Diário do Comércio, Brasil De Fato e Jornal Do

Dia, tem circulação nacional. Os jornais Diário de S. Paulo e Agora São Paulo tem circulação estadual, e os jornais Metrô News, Jornal Estação Free São Paulo, Diário Lance!, Meio & Mensagem e Diário de Notícias, são de circulação municipal.

No estado do Rio de Janeiro foram analisados 7 jornais, sendo todos eles (Jornal O Globo, O Dia, Correio do Brasil, Jornal Do Brasil, Extra, Jornal do Commercio, Jornal Corporativo) de circulação nacional. Em Minas Gerais também foram observados sete jornais, que se dividem em circulação nacional, estadual e regional. Os jornais Estado de Minas e Hoje Em Dia, são de circulação nacional, já O Tempo e Diário do Comércio são estaduais e os jornais Correio de Uberlândia, Jornal de Uberaba e Jornal da Manhã, tem circulação regional.

No Distrito Federal todos os jornais analisados (Jornal Alô Brasília, Correio Braziliense, Jornal de Brasília, Brasília Agora, Jornal Correio do Planalto) são de circulação regional, mesma característica apresenta pelos quatro jornais (Diário Da Manhã, Jornal O Popular, Tribuna do Planalto, Jornal O Hoje) analisados no estado de Goiás.

6.2 ONLINE/IMPRESSO

O objetivo aqui era observar se os jornais analisados possuem versão impresso e digital, somente impressa ou somente digital. No estado de São Paulo três jornais, dos 17 analisados, apresentam somente versão digital, são eles o jornal DC Diário do Comércio, Diário Lance! e Jornal Do Dia. No estado do Rio de Janeiro somente um jornal, dos sete observados, apresenta exclusivamente versão digital, o Jornal do Brasil. Em Minas Gerai todos os jornais analisados possuem versão digital e impressa, bem como os jornais do Distrito Federal e também do estado de Goiás.

6.3 DIÁRIO/TEMPO REAL

A característica da periodicidade dos jornais foi geral, todos os jornais analisados - 40 -, mesmo tendo duas versões ou somente uma, são diários e apresentam notícias também em tempo real.

6.4 CANAIS DE INTERAÇÃO

Quando se pensa em canais de interação, o foco são as formas como o leitor que acessa o portal de um jornal na Internet, tem de interagir com o veículo de comunicação, seja por meio de redes sociais, e-mail e comentários. Nos 40 jornais analisados, apenas um (Jornal do Dia) não apresentou canais de interação.

O Facebook, Twitter e G+ (rede social do Google), são os mais presentes canais de interação, o Facebook, por exemplo, estão ausentes somente no Jornal do Dia e no Diário de Notícias, ambos do estado de São Paulo. A presença dos comentários nas páginas das notícias nos portais também é notável, e é a forma mais simples e eficaz de interação. Outro canal muito presente são os e-mails, foi raro um portal não apresentar a opção de seus leitores enviarem alguma mensagem.

Ainda encontra-se a presença de outras redes sociais, como o Instagram (Jornal Estação Free São Paulo, Diário do Comércio de Minas Gerais, Jornal de Brasília e Jornal O Hoje do estado de Goiás), o YouTube (DCI - Diário, Comércio, Indústria & Serviços - de São Paulo, Brasil de Fato também de São Paulo, Jornal de Brasília, Diário da Manhã e Jornal O Hoje, ambos de Goiás), o LinkedIn (Estadão, Valor Econômico, DCI - Diário, Comércio, Indústria & Serviços -, DC Diário do Comércio, Jornal Estação Free São Paulo, Meio & Mensagem, Jornal Corporativo, Diário do Comércio de Minas Gerais, Diário da Manhã e Jornal O Hoje) e também redes como Flickr, diHITT, Delicious, Digg e as opções de assinar a Newsletter do jornal ou ainda opção como Sugestão de Pauta, como presente no Jornal Estação Free São Paulo.

6.5 PADRÃO DE CORES

O padrão das cores dos portais é bem característico, além da cor azul ser quase unânime - por senso comum representa a cor da Internet -, observa-se também que portais que possuem editorias de entretenimento dão mais liberdade à cartela de cores e investem em cores que chamam mais a atenção.

De forma geral, as cores azul, branco e preto dominam os portais, azul em detalhes, branco no fundo e preto para a fonte, porém, não raro aparecem cores como vermelho e cinza. Outras cores presentes são o amarelo, alaranjado e verde. No portal do Jornal de Brasília, por exemplo, nota-se uma cartela de cores bem

variada, o que confere até certa descontração para o jornal, já no Estadão o uso somente do branco e do azul atribuem seriedade.

O que nota-se, entretanto, é que, mesmo os jornais que optam por cores além do branco, azul e preto, as distribuem para formar uma estética agradável e que não canse a leitura, por isso as fontes são sempre em preto, cinza, azul escuro ou verde escuro, o fundo branco, e os detalhes da página ou links adicionais são os que seguram as cores mais fortes, como vermelho, amarelo e alaranjado.

6.6 NÚMERO DE COLUNAS

Os 40 jornais analisados apresentaram seus portais divididos em colunas, elas facilitam a navegação do leitor e conseguem organizar vários conteúdos em uma só página. O número de colunas foi de duas à cinco, sendo três colunas a quantidade mais presente. Dos 40 jornais, 31 optam pela divisão em três colunas, três jornais optam por duas colunas, 3 optam por quatro colunas e mais três que optam por cinco colunas.

6.7 EDITORIAS

As editorias dos jornais analisados são os itens que mais se diferem, uma vez que cada jornal cria as editorias que lhes favorecem. Porém, algumas editorias possuem presença quase unânime nos jornais, como as editorias de política, notícias sobre o Brasil e o mundo, economia e esporte.

No levantamento dos dados observou-se que o jornal Diário de Notícias, do estado de São Paulo, e o Jornal com o maior número de editorias, são 16. Característica peculiar, uma vez que a circulação do jornal é municipal. Já o Jornal Correio do Planalto, do Distrito Federal, é o jornal com o menor número de editorias, são três (Dicas, Eventos e Mais Notícias) e dentre elas, política, por exemplo, não é uma editoria presente, mesmo sendo um jornal da capital do Brasil.

A questão das editorias é algo bem particular de cada jornal, por isso as diferenças entre elas são justificáveis. Outros pontos a serem relatados é que nos jornais de circulação nacional e de grande importância para o país, as editorias se parecem, e tem uma certa generalidade para o número total de editorias, por exemplo, o Jornal Folha de S. Paulo e o Estadão, possuem seis editorias que se

assemelham: política (em ambas), mundo/internacional, economia (em ambas), cotidiano/São Paulo, esportes (em ambas) e cultura (em ambas), e o número total de editorias não passa de dez.

6.8 FORMA DE CONTATO

O objetivo ao analisar a forma de alguém ter contato com o jornal é de observar quais são os meios mais utilizados para tal função. Nas análises foram observadas três formas de contato que predominam: formulário de contato, que é quando o jornal disponibiliza uma página em seu portal para que o interessado deixe suas informações e a mensagem, telefones para contato, sejam eles normais ou fax, e também o e-mail de contato do jornal, que também uma maneira prática e eficaz.

Dos 40 jornais analisados, sete (Agora São Paulo, Jornal Estação Free São Paulo, Jornal do Dia, Correio do Brasil, Jornal do Commercio, Diário do Comércio, Jornal O Popular) apresentam somente uma das formas de contato, um jornal não apresenta nenhuma forma de contato (Diário da Manhã, do estado de Goiás), e o jornal Valor Econômico, do estado de São Paulo, além do formulário de contato e do telefone para contato, ainda apresenta a opção do chat online.

7 A ESCOLHA DA LINGUAGEM PARA PROGRAMAÇÃO, IMPLEMENTAÇÃO SAP (SIMPLE AS POSSIBLE) DE PÁGINAS DINÂMICAS DE NOTÍCIAS NA WEB, FERRAMENTAS E CONCEITOS.

7.1 VISÃO GERAL DAS LINGUAGENS COMPATÍVEIS COM O DOMÍNIO DAS FERRAMENTAS A SEREM UTILIZADAS E DESENVOLVIDAS NESTE TRABALHO

A escolha de linguagem de programação para prova dos conceitos derivados dos objetivos desta dissertação, bem como para geração da estrutura básica de suporte à efetivação dos objetivos específicos e gerais desta dissertação, é de fundamental importância, devendo atender todos os requisitos já elencados, de uma forma macro, a saber:

7.1.1 Programação Direta Na Nuvem Sem A Censura, Limitações Impostas Pelo Jailshell¹⁰ Do Servidor Da Hospedagem.

A linguagem deverá possuir recursos nativos para manipulação de informações multimídia e com hipertexto, direto na nuvem, evitando queda na eficiência das ações em tempo real necessárias ao produto aqui proposto, ou seja: Paradigmas e linguagens para implementação de páginas jornalísticas hipermídia interativas, para edição on-line, direta na nuvem, gerando código executável ou compilação *just in time*¹¹ na nuvem, independente de sistema operacional, browsers e dispositivos.

Vários paradigmas e respectivas linguagens foram testados no laboratório de Inteligência Artificial da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, buscando atender estes requisitos fundamentais.

As linguagens Haskell e C (inclusive Gambit C, devido a velocidade) são duas linguagens populares e eficientes que bem representam os paradigmas Funcional e Procedural, respectivamente. Assim, elas foram testadas para programação direta na nuvem, tendo sido descartadas principalmente pela

¹⁰ jailShell –limitações de ações que se pode implementar e realizar em uma hospedagem, no caso, tais como não poder gerar código executável, limitações de portas, limitações no SSH, outros. <http://forums.glowhost.com/knowledge-base/faq-what-jailshell-2433.html>, https://en.wikipedia.org/wiki/Restricted_shell, http://wiki.lunarpages.com/Jailshell_Disabled_Commands

¹¹ Compilação Just In Time – interpreta e executa quando o script é lido pelo browser

incompatibilidade na biblioteca **libC**¹² dos servidores na nuvem com a mesma biblioteca utilizada nos desktops e laptops. A constante modificações, atualizações desta libC, de forma não padronizada e não temporalmente sincronizadas, fazem com que os programas que se deseja atualizar em tempo real, como desta dissertação, acabem não rodando e/ou acarretando erros na atualização das páginas estáticas, bem como nas dinâmicas¹³.

Outras linguagens, de alto nível, alta eficiência, foram testadas com benchmarks focados no sistema deste trabalho, de onde se concluiu que:

- a. **CLEAN**, uma linguagem de paradigma funcional moderna, funcionou perfeitamente, sem restrições na programação, bastando apenas renomear o arquivo .exe para .cgi e fazer o upload para a nuvem. O problema com esta linguagem é que não é possível fazer os updates e upgrades diretos na nuvem sem um servidor dedicado. Um servidor dedicado custa caro (em torno de 45 vezes mais) em comparação com os gerais disponibilizados pelos sites de hospedagem¹⁴. Deve-se, com esta linguagem, fazer a compilação na máquina (desktop ou laptop) e, depois, fazer o upload do(s) arquivo(s) gerado(s). Uma grande vantagem desta linguagem é independer da libC.
- b. **Ciao Prolog** – linguagem de paradigma Lógico, a qual funcionou bem na nuvem, de forma similar ao CLEAN, independentemente, também, da libC.
- c. **Bigloo** – linguagem com paradigma funcional, a qual também foi capaz de realizar as funções necessárias direto na nuvem, mas precisa de ajustes não triviais nas diretivas de compilação. Estes ajustes, solução encontrada, não são nativos. Foram obtidos pela consulta e suporte do criador da linguagem (Manuel Serrano), cuja solução fornecida foi a de compilar com: **-static-all-bigloo e -copt -static**.
- d. **Racket**: Paradigma funcional, mas que permite também programação de forma procedural e lógica. Racket é uma linguagem de programação de espectro total (*full-spectrum programming language*). Ela vai além do Lisp e do Scheme, aceita dialetos que suportam objetos, tipos, avaliação lazy, e

¹² **libC** - biblioteca padrão do C na maioria das distribuições Linux, sistema operacional mais utilizado na atualidade (2015) para implementação de servidores e programação em nuvem.

¹³ **Páginas dinâmicas**: Interação com o usuário

¹⁴ **Custo de hospedagem**: como exemplo, um site não dedicado, na HostGator (hospedagem eficiente e segura), custa a partir de R\$9,99 (nove reais e noventa e nove centavos) por mês, enquanto um dedicado custa a partir R\$450,00 (quatrocentos e cinquenta reais) por mês, portanto, 45 vezes mais caro(www.hostgator.com.br).

outros recursos. Funciona na nuvem sem problemas. Consegue-se, com esta linguagem, coloca-la completamente em um servidor não dedicado da Hostgator. Esta característica potencializa, permite a utilização de todos os recursos da linguagem, com compilação direta na nuvem, sem que os uploads sejam recusados, barrados, pela hospedagem, funcionando como uma hospedagem dedicada de baixo custo, permitindo que as alterações nas páginas estáticas e dinâmicas sejam efetivadas até mesmo por soluções mobiles como celulares e tablets.

- e. **SBCL Lisp**: Um grande problema encontrado com esta linguagem, a mais eficiente e versátil da atualidade, foi que o Lisp aloca mais memória do que permitido na rede, gerando mensagens de erro que não indicam tal problema. Soluções foram implementadas para usar o SBCL no projeto, tal como compilar com a linha de comando no script: **sbcl - -dynamic-space-size 256**. Um grande problema nesta linguagem é a capacitação de um profissional para trabalhar com a mesma, mesmo sem ser direto na nuvem. Ela possui uma alta relação custo de aprendizado/benefício, para o presente trabalho.
- f. **Emacs com emacs-lisp** – Excelente linguagem para programação com paradigma textual, e, devido a isto, não sofre os efeitos do controle, da censura pela hospedagem. Para tanto, deve-se programar utilizando o **ORG MODE**¹⁵. A princípio, um paradigma aparentemente complexo de se aprender, com uma grande quantidade de comandos para se conhecer e decorar (para agilizar o processo de produção), mas aderente, no caso de um jornal, ao paradigma utilizado pelos jornalistas e editores no dia a dia. É uma solução gratuita e de excelente custo benefício. Uma grande vantagem desta linguagem é a independência de recursos visuais, os quais são fortemente datados, perecíveis tecnologicamente. Todo arquivo é gerado textualmente e compilado, posteriormente, para praticamente todos os formatos, entre eles, o de páginas WEB, aceitando inclusive imagens no formato moderno SVG¹⁶ aceito por todos os navegadores modernos.

¹⁵ **ORG MODE** - http://orgmode.org/worg/org-tutorials/orgtutorial_dto.html

¹⁶ **SVG** - <https://pt.wikipedia.org/wiki/SVG> - O mesmo descreve de forma vetorial desenhos e gráficos bidimensionais, quer de forma estática, quer dinâmica ou animada.

7.2 A ESCOLHA DAS LINGUAGENS

Pelos motivos citados, a escolha recairia sobre dois paradigmas, duas linguagens e ambientes distintos: **DrRacket** e **Emacs com emasc-Lisp no modo ORG**.

Pela simplicidade, agilidade na implementação das páginas, por independência da censura da **jailshell**, por ser padrão em todos os sistemas operacionais da atualidade, por ser nativo nas hospedagens da nuvem e por ser uma linguagem multiparadigma, além do melhor editor para tais propósitos, definiu-se pelo **emacs Org-Mode**. Observando-se o seguinte:

- a. No caso de se montar um jornal para grande circulação e que precisa de um servidor, banco de dados e recursos que os servidores padrões, genéricos das hospedagens não atenderiam em velocidade, recursos e liberdade de acessos e buscas de informações. Neste caso, **DrRacket** é a escolha pelos motivos já citados, o que não é o caso desta dissertação.
- b. O profissional jornalista, com pouca fluência em programação, acostumado a edição de textos em editores com linguagens de markup¹⁷ (principalmente), mas que desejam recurso de geração de páginas dinâmicas WEB multimídia, com hipertexto e que rodem em tecnologia mobile e desktop com precisão, fidelidade ao texto gerado, onde os recursos dos servidores genéricos, como o Apache¹⁸, são suficientes. Neste caso, a opção **Emacs-orgmode-emas-lisp** é a mais aderente e vai de encontro a esta pesquisa.

7.2.1 Emacs – Org-Mode

- Permite ser usado tanto em desktop, mobile e direto na nuvem com o popular servidor Apache¹⁹. Assim, nenhuma barreira ou restrição serão impostas às páginas criadas nesta plataforma.
- O **Emacs** foi implementado em Lisp: Emacs é Lisp. Lisp pertence a um paradigma funcional, matemático, ela é transparente referencialmente, e sem arbitrariedades. A matemática livra-se de decisões arbitrárias

¹⁷ Markup- https://en.wikipedia.org/wiki/Markup_language

¹⁸ Apache - https://en.wikipedia.org/wiki/Apache_HTTP_Server

¹⁹ Servidor Apache - é o mais bem sucedido servidor web livre. Foi criado em 1995 por Rob McCool, então funcionário do NCSA (National Center for Supercomputing Applications). https://en.wikipedia.org/wiki/Apache_HTTP_Server

porque os sistemas matemáticos possuem dois componentes: um conjunto de axiomas ortogonais e um conjunto de regras de inferência. Os axiomas são ortogonais porque as regras de inferência não permitem deduzir nenhum deles a partir dos outros. Outra exigência da matemática é a **navalha de Ockham**²⁰ (*Entia non sunt multiplicanda praeter necessitatem* - As entidades não devem ser multiplicadas além da necessidade). Essa *lex parsimoniae* diz que o número de regras de inferência somado com o número de axioma deve ser mínimo. Isso elimina a arbitrariedade na matemática.

- A semântica do Lisp, baseada no Cálculo Lambda, é matemática pura. Por definição, possui três axiomas apenas (alpha conversion, beta reduction and eta conversion) e uma única regra de inferência e é *Lazy*²¹ (*lex parsimoniae*). Isto não impede que se utilizem avaliações *Eager*²² (não *lex parsimoniae*). Como já afirmado, Racket permite a implementação de qualquer gramática e sintaxe, dialetos, que se deseje, inclusive de linguagens procedimentais. Assim, **Emacs** une todas as qualificações, ferramentas e recursos necessários para implementação das páginas dinâmicas objeto desta dissertação, herdando, em um editor, todas as vantagens e recursos da melhor e mais completa linguagem de programação funcional há mais de 30 anos.

Um grande atrativo é que ao usuário do Emacs esta característica é transparente, gerando a linguagem de mark-up mais intuitiva e com a estrutura mais limpa que se possa desejar, onde, em poucas linhas de código, cria-se páginas multidevices, multiplataforma onde outras, como **HTML** gastar-se-ia muito mais.

EXEMPLO: Como exemplo, eis uma página contendo:

- Título

²⁰ **Navalha de Ockham** - O princípio afirma que a explicação para qualquer fenômeno deve assumir apenas as premissas estritamente necessárias à explicação do mesmo e eliminar todas as que não causariam qualquer diferença aparente nas predições da hipótese ou teoria.

²¹ **Lazy** – apesar da tradução corrente chamar de avaliação preguiçosa, pelo contrário, é uma avaliação eficiente, *lex parsimoniae*, ou seja: avaliação das funções e argumentos somente quando necessário e solicitado explicitamente.

²² **Eager** – argumentos e funções são sempre avaliados por completo, sem que seja solicitado explicitamente, como é feito nas linguagens procedurais.

- Um índice
- Uma imagem local
- Uma música local
- Um link para um vídeo na internet

A página deverá ser implementada de forma a rodar em qualquer dispositivo, mobile ou desktop, em qualquer browser.

A PÁGINA:



PÁGINA DE AVALIAÇÃC PÁGINA DE AVALIA X + - □ X

← → ↻ | file:///C:/jornal_taty/ta | ☆ | ≡ | 🔍 | ...

PÁGINA DE AVALIAÇÃO DE CÓDIGO

Table of Contents

- [1. Imagem da Natalie Portman](#)
- [2. Música Royals](#)

1 Imagem da Natalie Portman

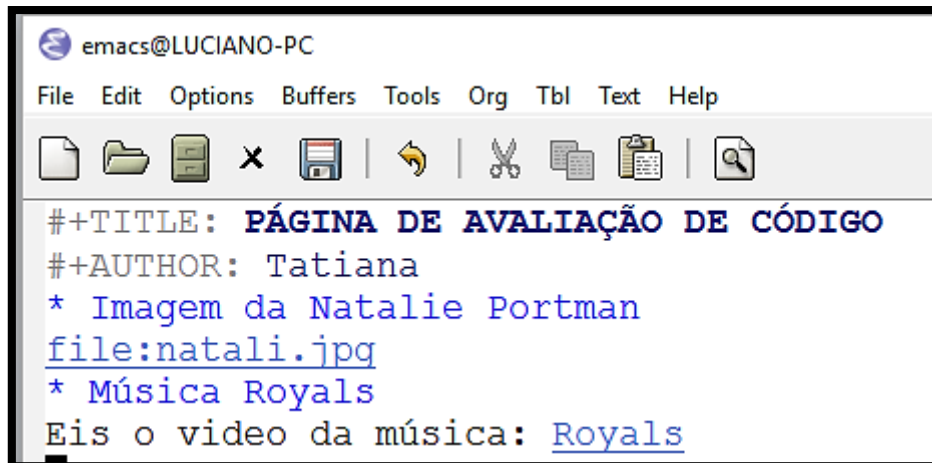


2 Música Royals

Eis o video da música: [Royals](#)

Author: Tatiana
Created: 2016-01-14 qui 03:38
[Emacs](#) 24.5.1 ([Org](#) mode 8.2.10)
[Validate](#)

A PÁGINA EM EMACS ORG-MODE: 6 LINHAS DE CÓDIGO

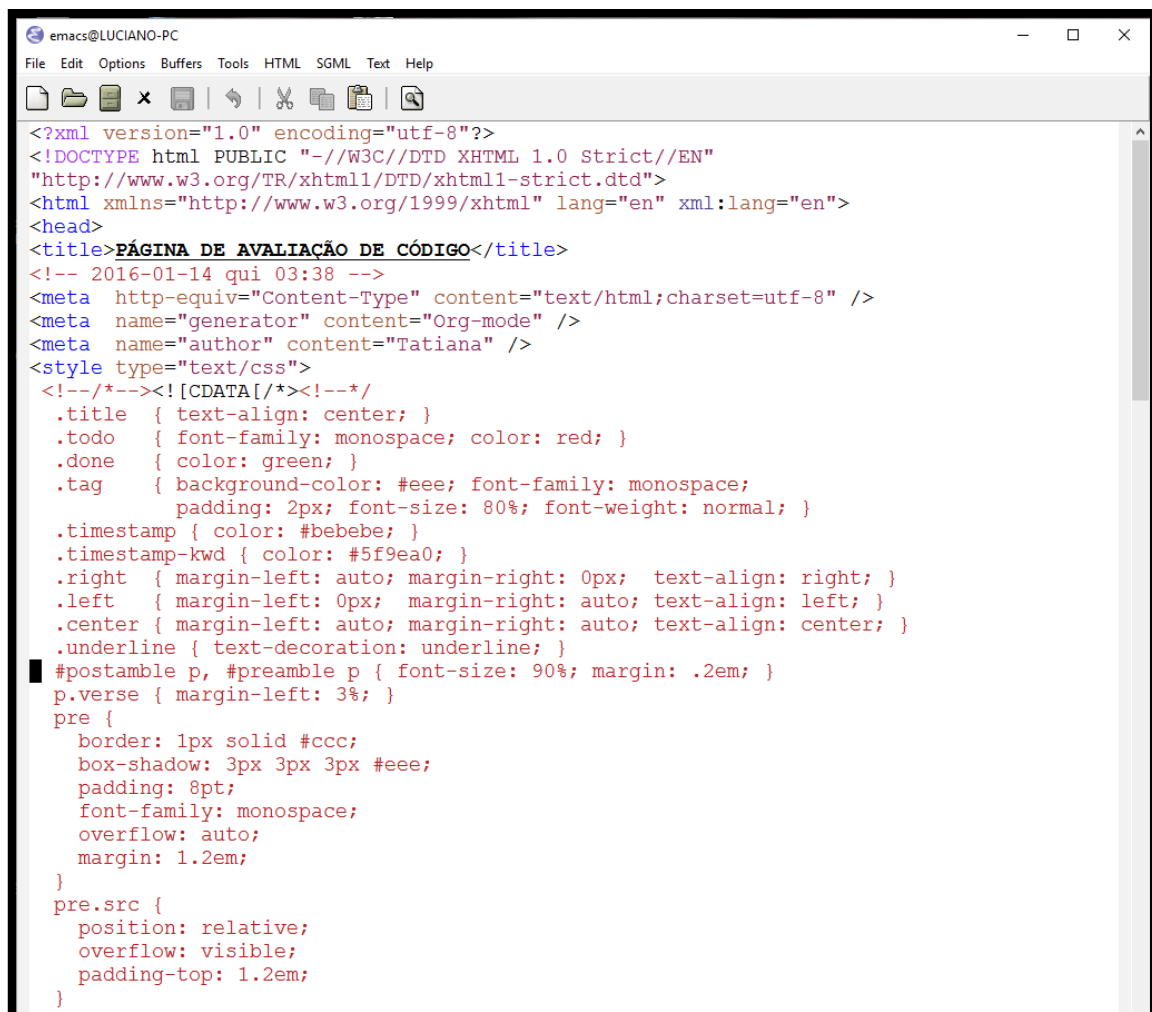


```

emacs@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
# +TITLE: PÁGINA DE AVALIAÇÃO DE CÓDIGO
# +AUTHOR: Tatiana
* Imagem da Natalie Portman
file:natali.jpg
* Música Royals
Eis o video da música: Royals

```

A PÁGINA EM HTML – 152 LINHAS – 25 VEZES MAIS LINHAS DE CÓDIGO



```

emacs@LUCIANO-PC
File Edit Options Buffers Tools HTML SGML Text Help
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>PÁGINA DE AVALIAÇÃO DE CÓDIGO</title>
<!-- 2016-01-14 qui 03:38 -->
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
<meta name="generator" content="Org-mode" />
<meta name="author" content="Tatiana" />
<style type="text/css">
<!--/*--><![CDATA[/*><!--*/
.title { text-align: center; }
.todo { font-family: monospace; color: red; }
.done { color: green; }
.tag { background-color: #eee; font-family: monospace;
padding: 2px; font-size: 80%; font-weight: normal; }
.timestamp { color: #bebebe; }
.timestamp-kwd { color: #5f9ea0; }
.right { margin-left: auto; margin-right: 0px; text-align: right; }
.left { margin-left: 0px; margin-right: auto; text-align: left; }
.center { margin-left: auto; margin-right: auto; text-align: center; }
.underline { text-decoration: underline; }
#postamble p, #preamble p { font-size: 90%; margin: .2em; }
p.verse { margin-left: 3%; }
pre {
border: 1px solid #ccc;
box-shadow: 3px 3px 3px #eee;
padding: 8pt;
font-family: monospace;
overflow: auto;
margin: 1.2em;
}
pre.src {
position: relative;
overflow: visible;
padding-top: 1.2em;
}

```

CONTINUA

```

padding-top: 1.2em;
}
pre.src:before {
display: none;
position: absolute;
background-color: white;
top: -10px;
right: 10px;
padding: 3px;
border: 1px solid black;
}
pre.src:hover:before { display: inline;}
pre.src-sh:before { content: 'sh'; }
pre.src-bash:before { content: 'sh'; }
pre.src-emacs-lisp:before { content: 'Emacs Lisp'; }
pre.src-R:before { content: 'R'; }
pre.src-perl:before { content: 'Perl'; }
pre.src-java:before { content: 'Java'; }
pre.src-sql:before { content: 'SQL'; }

table { border-collapse:collapse; }
caption.t-above { caption-side: top; }
caption.t-bottom { caption-side: bottom; }
td, th { vertical-align:top; }
th.right { text-align: center; }
th.left { text-align: center; }
th.center { text-align: center; }
td.right { text-align: right; }
td.left { text-align: left; }
td.center { text-align: center; }
dt { font-weight: bold; }
.footpara:nth-child(2) { display: inline; }
.footpara { display: block; }
.footdef { margin-bottom: 1em; }
.figure { padding: 1em; }
.figure p { text-align: center; }
.inlinetask {
padding: 10px;
border: 2px solid gray;
margin: 10px;
background: #ffffcc;
}

```

CONTINUA

```

#org-div-home-and-up
{ text-align: right; font-size: 70%; white-space: nowrap; }
textarea { overflow-x: auto; }
.linetr { font-size: smaller }
.code-highlighted { background-color: #ffff00; }
.org-info-js_info-navigation { border-style: none; }
#org-info-js_console-label
{ font-size: 10px; font-weight: bold; white-space: nowrap; }
.org-info-js_search-highlight
{ background-color: #ffff00; color: #000000; font-weight: bold; }
/*]]>*/-->
</style>
<script type="text/javascript">

```

CONTINUA

```

<!--/*--><![CDATA[/*><!--*/
function CodeHighlightOn(elem, id)
{
  var target = document.getElementById(id);
  if(null != target) {
    elem.cacheClassElem = elem.className;
    elem.cacheClassTarget = target.className;
    target.className = "code-highlighted";
    elem.className   = "code-highlighted";
  }
}
function CodeHighlightOff(elem, id)
{
  var target = document.getElementById(id);
  if(elem.cacheClassElem)
    elem.className = elem.cacheClassElem;
  if(elem.cacheClassTarget)
    target.className = elem.cacheClassTarget;
}
/*]]>*///-->
</script>
</head>
<body>
<div id="content">
<h1 class="title">PÁGINA DE AVALIAÇÃO DE CÓDIGO</h1>
<div id="table-of-contents">
<h2>Table of Contents</h2>
<div id="text-table-of-contents">
<ul>
<li><a href="#sec-1">1. Imagem da Natalie Portman</a></li>
<li><a href="#sec-2">2. Música Royals</a></li>
</ul>
</div>
</div>
</div>

```

CONTINUA

```

<div id="outline-container-sec-1" class="outline-2">
<h2 id="sec-1"><span class="section-number-2">1</span> Imagem da Natalie
Portman</h2>
<div class="outline-text-2" id="text-1">
<div class="figure">
<p>
</p>
</div>
</div>
<div id="outline-container-sec-2" class="outline-2">
<h2 id="sec-2"><span class="section-number-2">2</span> Música Royals</h2>
<div class="outline-text-2" id="text-2">
<p>
Eis o video da música: <a
href="https://www.youtube.com/watch?v=nlcIKh6sBtc">Royals</a>
</p>
</div>
</div>
<div id="postamble" class="status">
<p class="author">Author: Tatiana</p>
<p class="date">Created: 2016-01-14 qui 03:38</p>
<p class="creator"><a href="http://www.gnu.org/software/emacs/">Emacs</a>
24.5.1 (<a href="http://orgmode.org">Org</a> mode 8.2.10)</p>
<p class="validation"><a
href="http://validator.w3.org/check?uri=referer">Validate</a></p>
</div>
</body>
</html>

```

FIM

Observa-se, portanto, o quanto de agilidade o repórter, o jornalista, o editor de um jornal ganha em relação à quem for codificar, mesmo usando programas gráficos.

O grau de personalização e recursos de interação dinâmica permitidos pelo **emacs**, de uma forma simples e intuitiva, com poucos comandos a decorar, a assimilar, com um baixíssimo custo/benefício faz desta uma excelente escolha para os propósitos desta dissertação.

No próximo capítulo tais recursos serão detalhados passo a passo, tornando segura e confiável a utilização do **emacs org-mode** na produção de páginas dinâmicas.

7.2.1.1 Multiplataforma

Não se cansa de dizer o quanto trabalhar com Windows para trabalhar direto na nuvem, comunicar com servidores é uma tarefa discriminada pelos profissionais de computação.

O que ocorre é que ainda em várias Universidades, Faculdades, escolas e empresas (principalmente as MPEs²³), usam Windows. Como computador pessoal, no Brasil, é ainda o sistema operacional mais em uso após o Android nos Mobile Phones.

Assim, é de fundamental importância desmistificar este preconceito, já que, conforme será apresentado neste trabalho, trabalhar com o Windows é tão simples quanto em qualquer outro sistema operacional, seja para geração de páginas off line ou on line na nuvem, direto na hospedagem, sem perder a segurança que o protocolo **SSH** disponibiliza.

Dito isto, o tópico a seguir mostra com detalhes como instalar o Emacs no Windows: o primeiro passo. É de suma importância que isto faça parte do corpo da dissertação e não de um anexo, já que é algo ainda de grande desconhecimento por parte dos usuários alvo desta deste trabalho (jornalistas, estudantes, editores, correspondentes, etc.), mesmo os de programação.

As informações e conceitos aqui obtidos são de fundamental importância para as implementações que serão efetivadas no próximo capítulo.

²³ **MPE- Médias e pequenas empresas** – geralmente fundamentalmente familiares. São a maioria das empresas no Brasil atualmente.

7.2.1.2 Instalando o Emacs no Windows

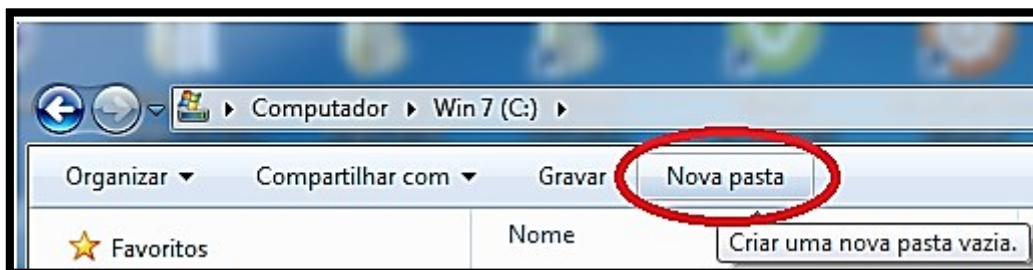
É importante padronizar onde instalar o programa. Saber onde está instalado. Não deixar à cargo do sistema operacional, no caso, o Windows, escolher instalar onde bem desejar. Saber onde tudo está instalado é fundamental para um profissional poder fazer alterações, configurações e outras ações que se façam necessárias, como é este, o caso. Usar o Emacs no Linux é simples. Tudo foi criado para estar no local certo. Tudo funciona em uma pasta chamada **home**, pasta esta que, no Windows, ninguém sabe dizer onde está, mas, seguindo os passos que serão mostrados, este problema deixa de existir e pode-se trabalhar no Windows com o Emacs sem nenhum problema, como um bom profissional trabalharia.

- INSTALANDO O EMACS NA RAIZ DO DISCO C DO WINDOWS, DENTRO DE UMA PASTA CHAMADA **emacs**, OU SEJA:

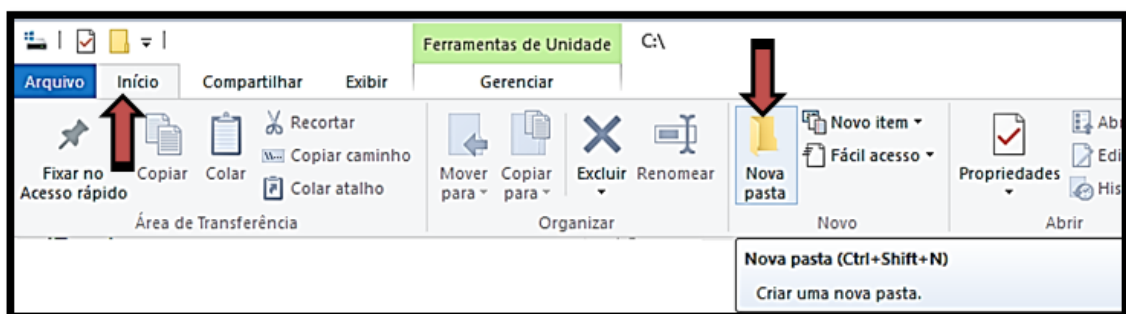
- Instalar o **emacs** em **C:\emacs**.

Para tanto, crie um diretório na pasta C: chamada emacs. Em cada versão do Windows isto é feito de uma forma diferente.

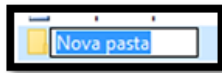
No Windows 7, estando no gerenciador de arquivos, entra-se no disco C: e clica-se, na barra de menus, em **Nova pasta**



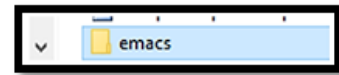
No Windows 10, estando no gerenciador de arquivos, entra-se no disco C:, clique no menu **Início** e em **Nova Pasta**



Feito isto, em qualquer um das versões citadas, ou mesmo mais antigas, uma pasta é aberta:



Nomeie a mesma com o nome emacs...



7.2.1.3 Baixando o Emacs para Windows

Existem vários repositórios de onde você pode baixar o emacs.

Dois deles são:

- http://sourceforge.net/projects/emacsbinw64/?source=typ_redirect
- <http://mirror.nbtelecom.com.br/gnu/emacs/windows/>

O **Windows 7** geralmente abre uma janela para que o usuário escolha onde quer baixar o arquivo que vai fazer o download (baixar), já no **Windows 10** o mesmo normalmente baixa para o diretório Downloads (e aí começam os problemas, a falta de padronização).

Se o Windows perguntar onde deseja baixar (Abrir), já se sabe: No **disco C**, dentro do diretório **emacs**.

Se o mesmo for baixado direto para o diretório **Downloads**, sem problemas, é só depois transferir o arquivo baixado para a pasta **emacs** no **disco C**.

IMPORTANTE: ESCOLHA A ÚLTIMA VERSÃO COM EXTENSÃO **.zip** E NÃO A **.sig**²⁴.

- DOWNLOAD AUTOMÁTICO PARA A PASTA DOWNLOAD, OU OUTRA PASTA PRÉ PROGRAMADA (é possível escolher na configuração)

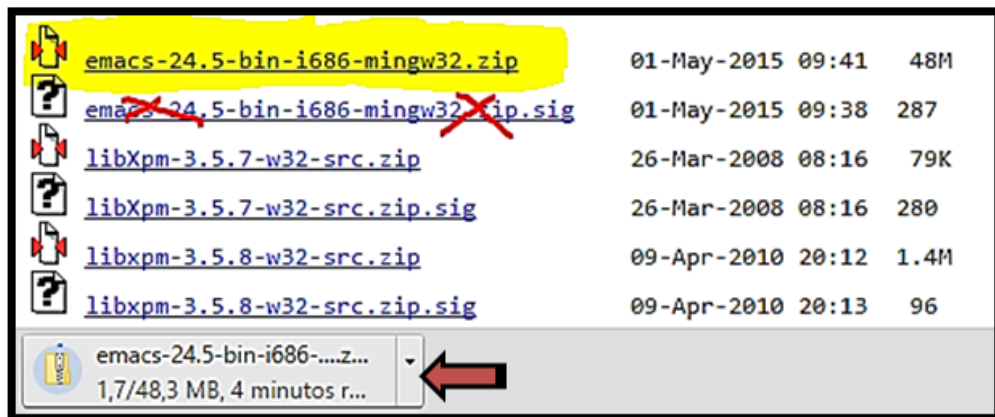
Suponha a escolha para baixar ter sido no endereço:

<http://mirror.nbtelecom.com.br/gnu/emacs/windows/>

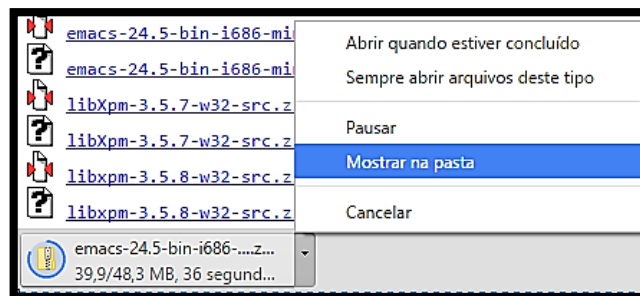
No caso de baixar automático, na parte inferior da página onde estiver dando o download, aparece a mensagem do tipo:

²⁴ **ARQUIVOS .sig – sig vem de signature: assinatura.** Os mesmo são arquivos textuais que são lidos de forma diferentes por softwares diferentes. O mesmo não vem, neste caso, com o emacs. O certo é o arquivo **.zip**. Observe os tamanhos de cada um:

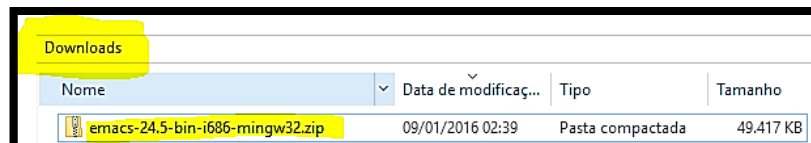
emacs-24.5-bin-i686-mingw32.zip	09/01/2016 02:39	Pasta compactada	49.417 KB
emacs-24.5-bin-i686-mingw32.zip.sig	09/01/2016 01:53	Arquivo SIG	1 KB



Após o download terminar, clique na setinha ao lado e clique em **Mostrar na pasta**. O sistema operacional abrirá a pasta onde baixou, mesmo que não saiba onde é:



No qual está o arquivo zipado do **emacs**:

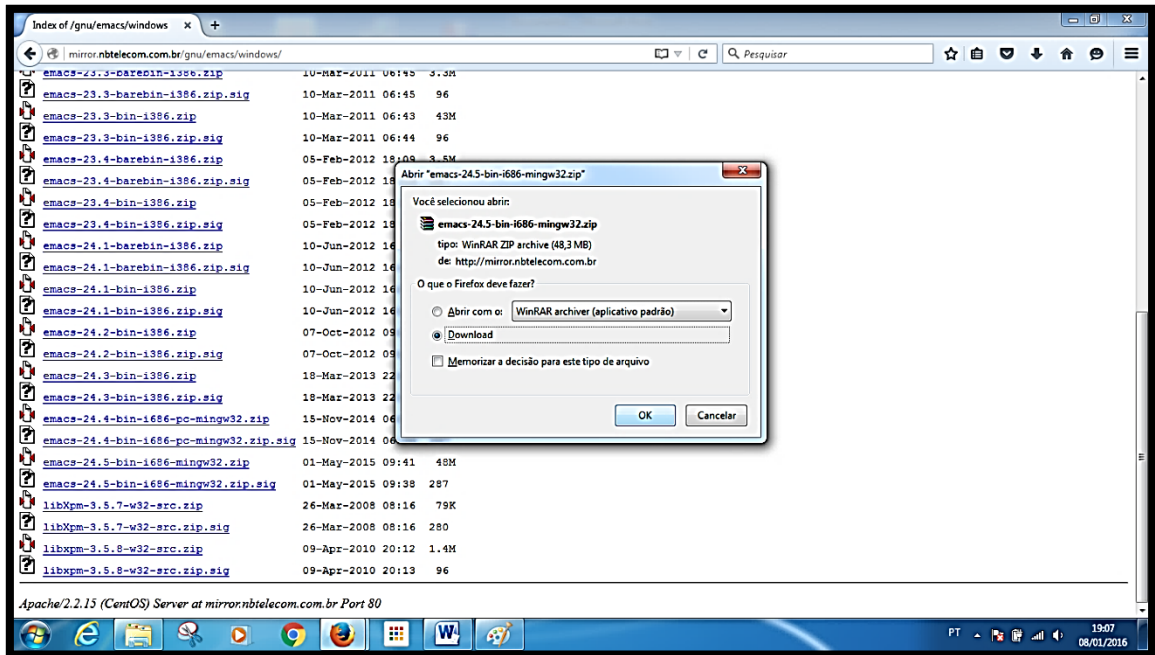


O mesmo deverá ser transferido ou copiado para a pasta **C:\emacs**

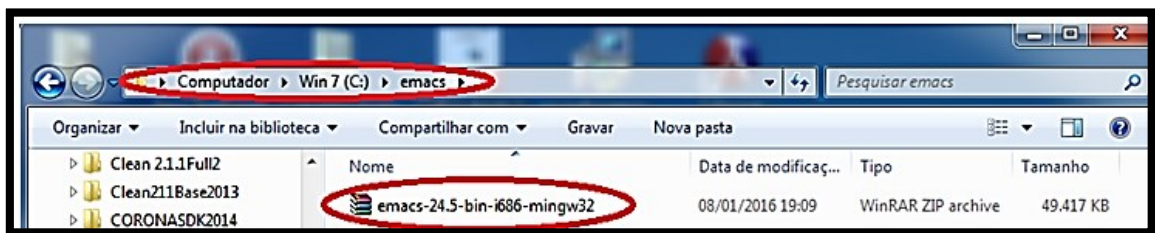


- DOWNLOAD COM SOLICITAÇÃO DE EM QUE PASTA BAIXAR.

Nestes casos, como é o padrão do Windows 7 (caso você não tenha alterado) normalmente a seguinte tela aparecerá:

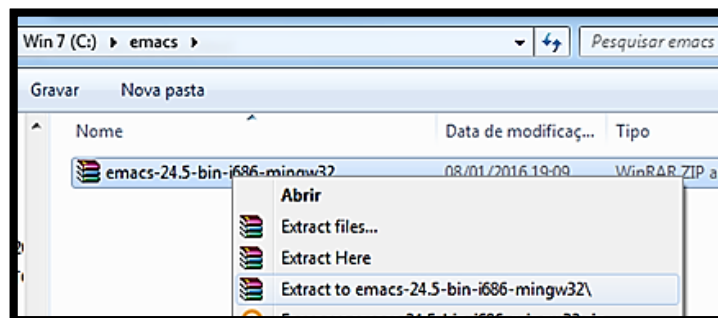


Escolha abrir na pasta `C:\emacs`

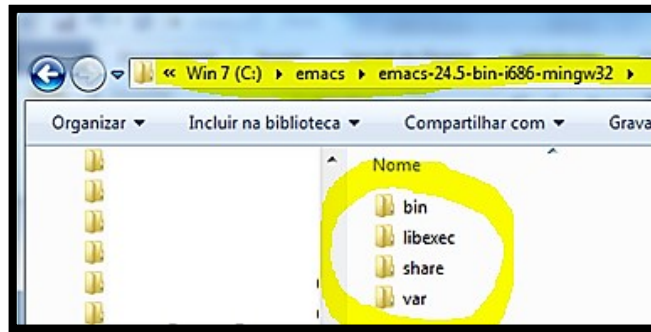


- DESCOMPACTANDO O ARQUIVO NA PASTA emacs

Descompacte os arquivos na pasta `C:\emacs`



Feito isto, tem-se uma pasta chamada `emacs-24.5-bin-i686-mingw32`. Dentro dela está a pasta `bin`, `libexec`, `share` e `var`.



- VARIÁVEL DE AMBIENTE²⁵ **HOME**

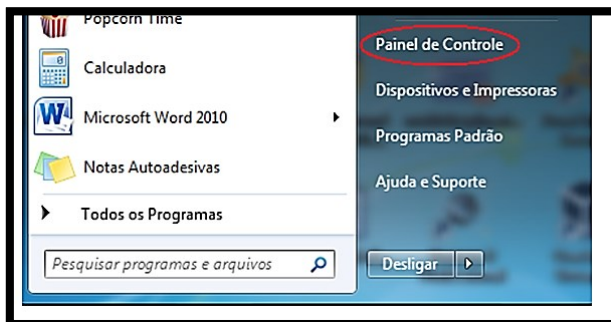
Este é o grande "segredo", por assim dizer, para que se possa trabalhar com o emacs como um profissional do Linux trabalha.

Criando a variável HOME:

- NO WINDOWS 7 -> Clique com o botão **esquerdo** do mouse no ícone do menu **Iniciar**



Clique em **Painel de Controle**



CLIQUE no item **Sistema**

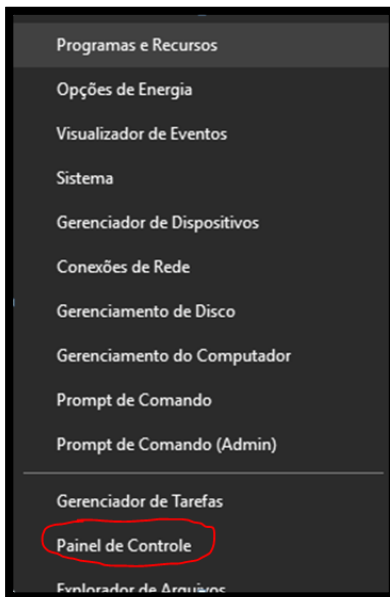


- NO WINDOWS 10 -> Clique com o botão **direito** do mouse no ícone do menu **Iniciar**



Clique em **Painel de Controle**

²⁵ **Variável de ambiente:** Uma variável de ambiente é uma variável, um local no computador, que é criada no sistema operacional que poderá ser utilizada por qualquer programa instalado no sistema operacional, uma variável que não pertence apenas ao programa instalado.

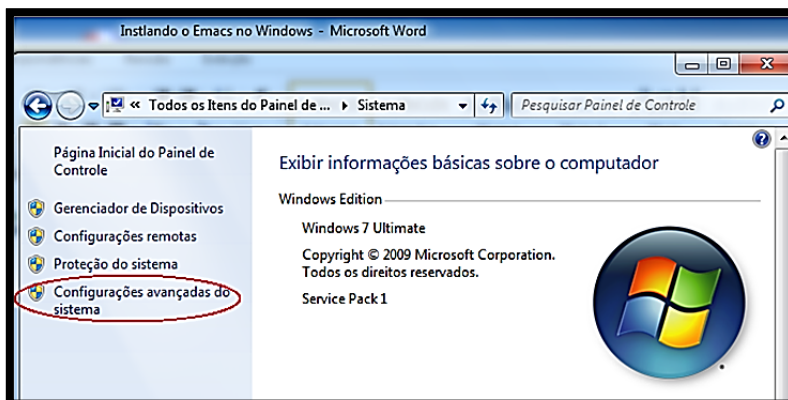


CLIQUE no item **Sistema**



Ao entrar em Sistema, Clique em **Configurações Avançadas do Sistema**

No Windows 7

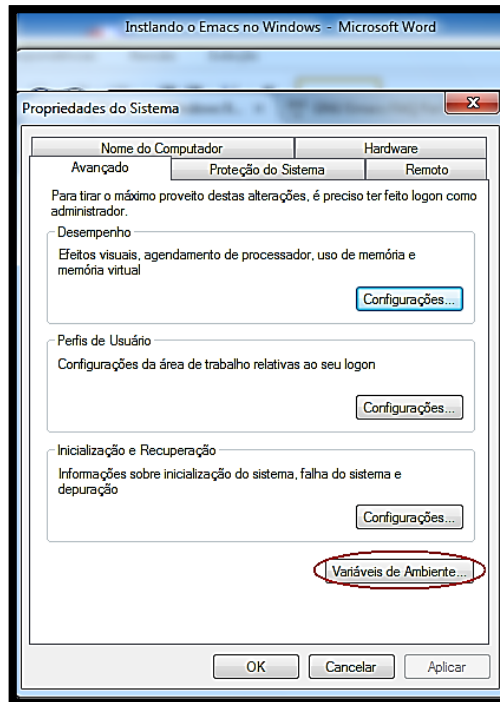


- No Windows 10

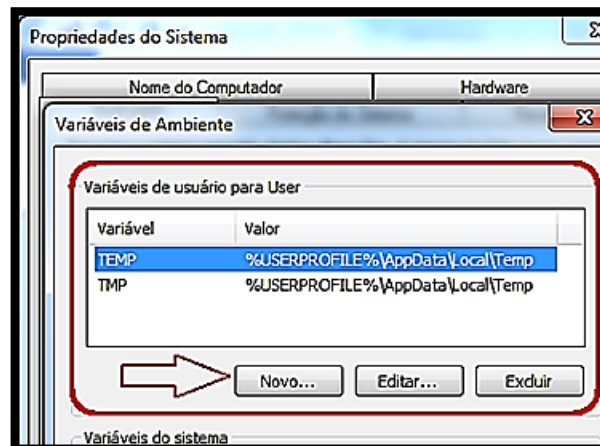


Clique em **Variáveis de Ambiente...** onde você vai criar a variável **HOME**, necessária para um trabalho profissional no **emacs**, quando for necessário

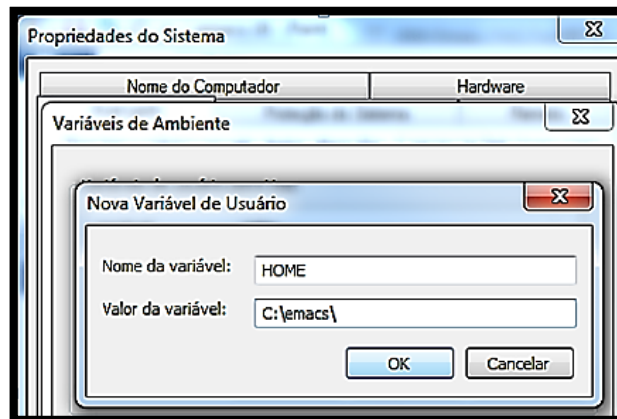
personaliza-lo, configurá-lo de forma diferente da padrão com que ele vem (o que geralmente é necessário).



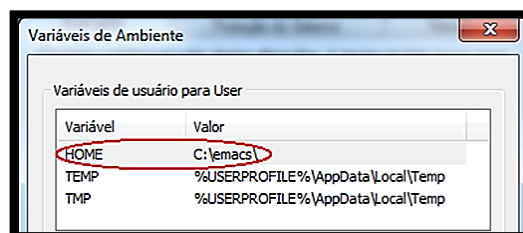
Em **Variáveis de Ambiente** crie uma nova variável chamada **HOME** com o valor **C:\emacs**. Clique em **Novo...** e preencha os campos como as imagens a seguir



Entre com a variável e o valor:



Clique em OK e veja que a variável foi criada.



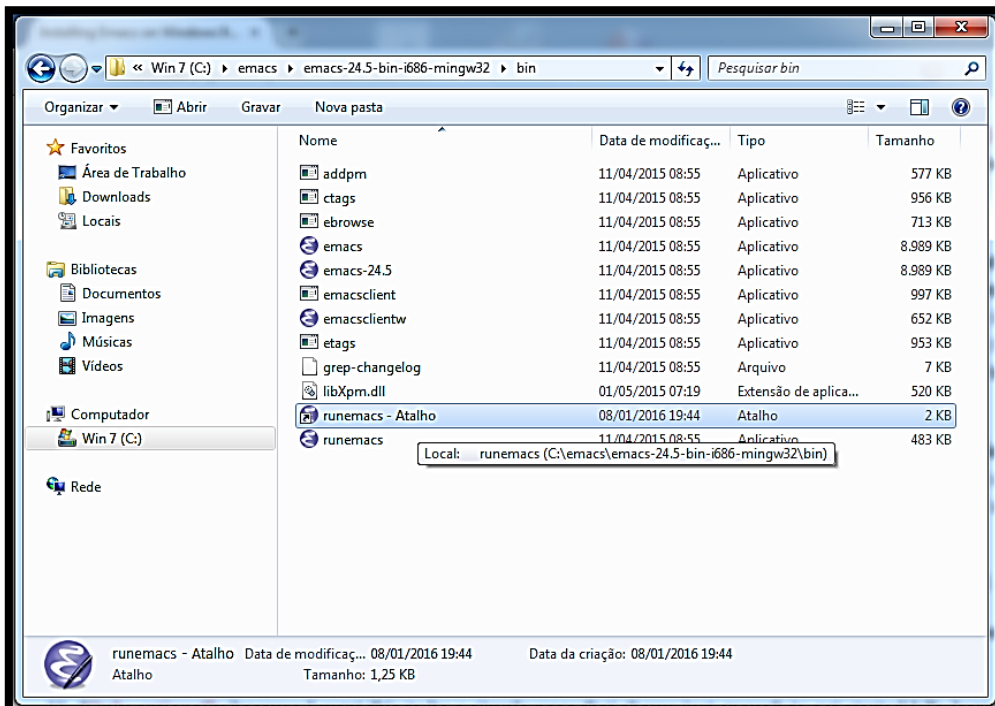
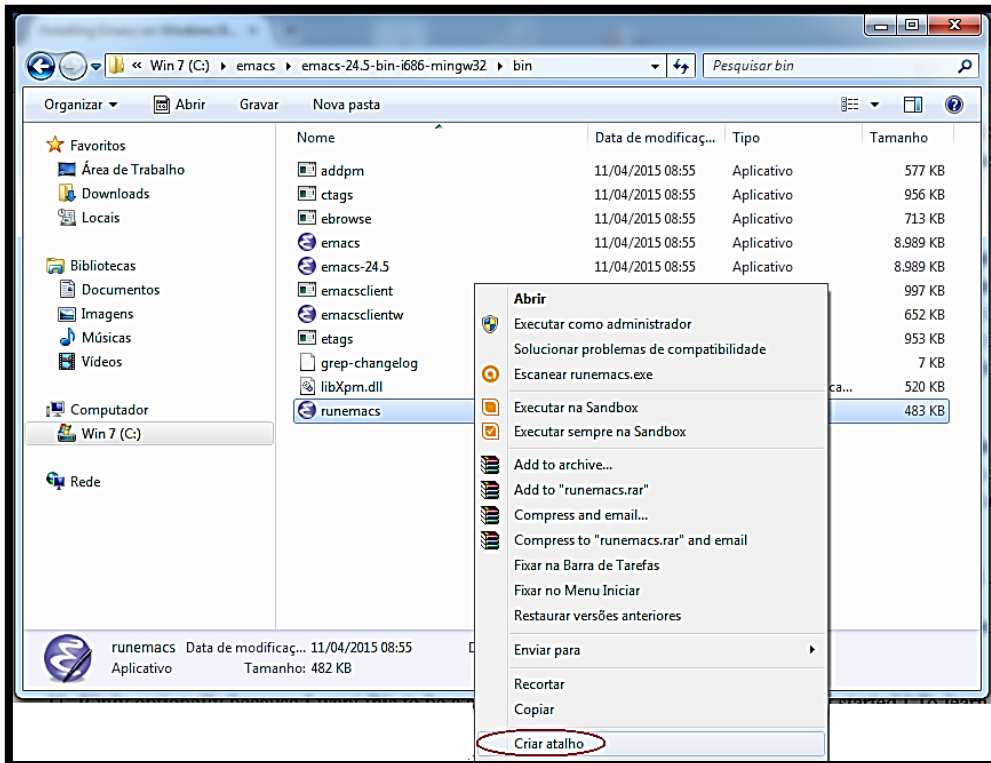
Clique em OK para fechar a janela.

IMPORTANTE!

PARA QUE A INSTALAÇÃO TENHA EFEITO E PARA QUE O SISTEMA OPERACIONAL ASSOCIE A VARIÁVEL HOME AO EMACS, DEVE-SE REINICIAR O COMPUTADOR ANTES DE USAR O EMACS.

Após reiniciar o Windows:

Quando ligar o computador novamente, vá até a pasta **c:\emacs\ emacs-24.5-bin-i686-mingw32\bin**, procure pelo arquivo **runemacs**. Crie um atalho e mova-o para a área de trabalho para acessar facilmente.



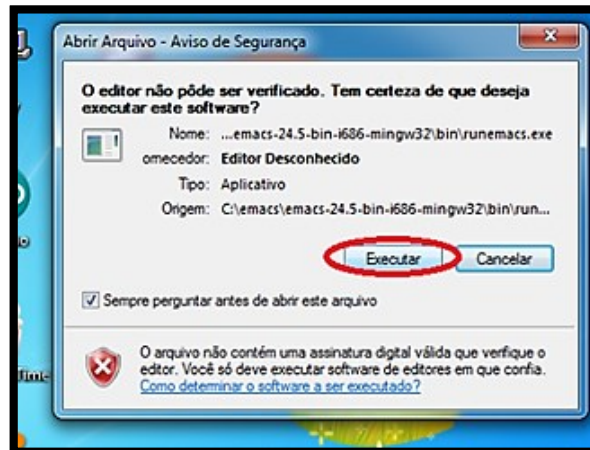
7.2.1.4 Abriendo, executando finalmente o Emacs

Seguidas as instruções, o **Emacs** está pronto para ser corretamente executado.

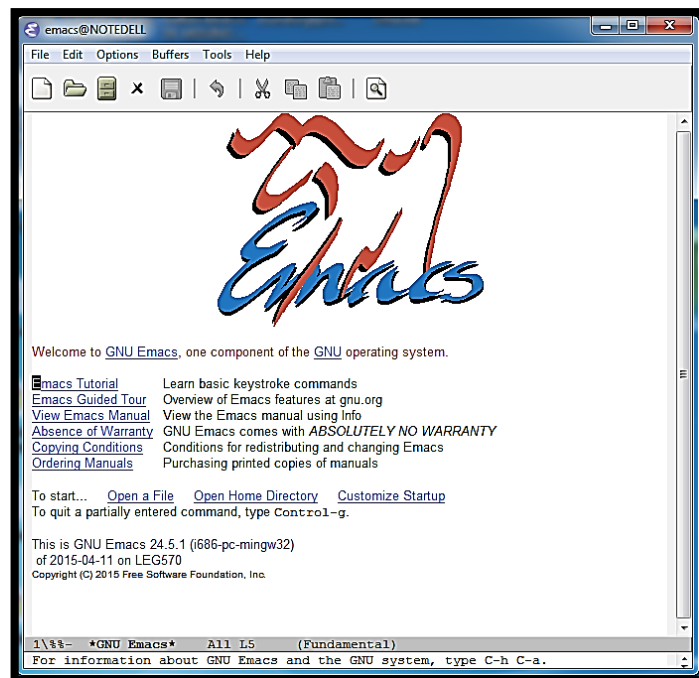
Vá até a **Área de Trabalho(desktop)** e execute o **Emacs**



No **Windows 7** a seguinte informação aparece. Clique em **Executar**.



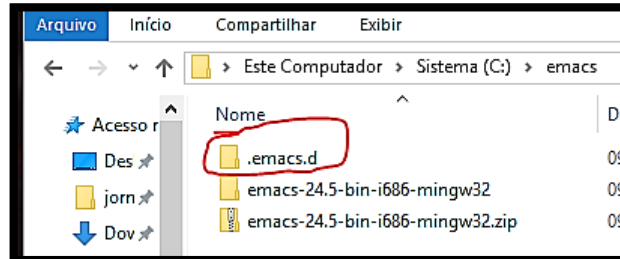
O Emacs é aberto, comprovando que a instalação teve sucesso.



IMPORTANTE: Todos os passos até aqui foram relevantes para que a pasta de configuração do EMACS fosse criada, o que no Linux é algo convencional e normal. Sem isto, a pasta **.emacs.d** não é criada e não se consegue personalizar o emacs para algumas inserções necessárias na criação de páginas multimídia interativas.

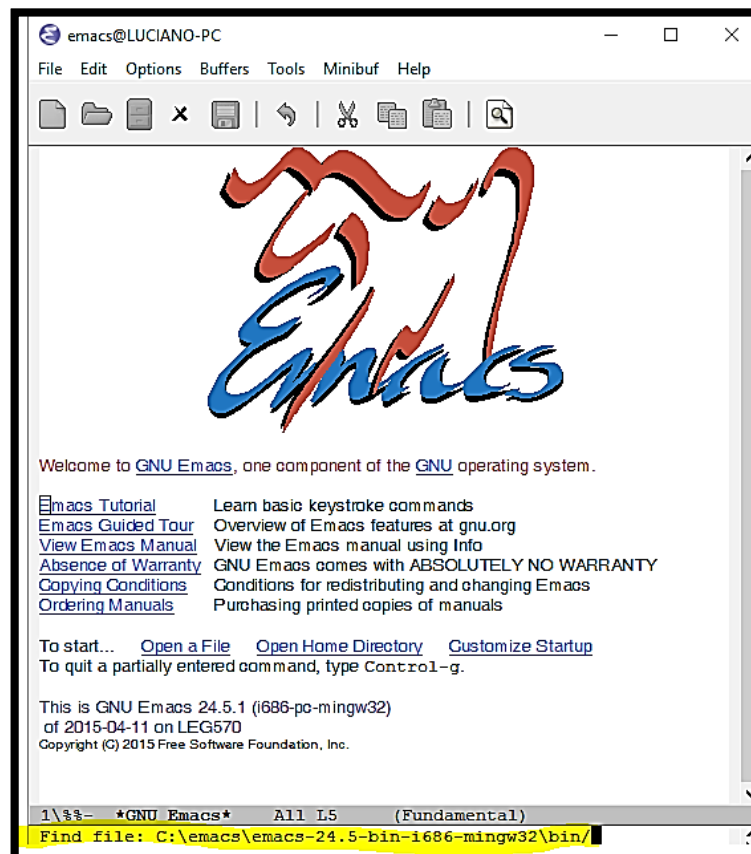
- VERIFICANDO QUE A PASTA **.emacs.d** FOI CRIADA

1. Feche o **emacs** e volte na pasta **c:\emacs**.
2. Verifique que a pasta **.emacs.d** foi criada.



Dentro dela será criado o arquivo de configuração do **emacs**.

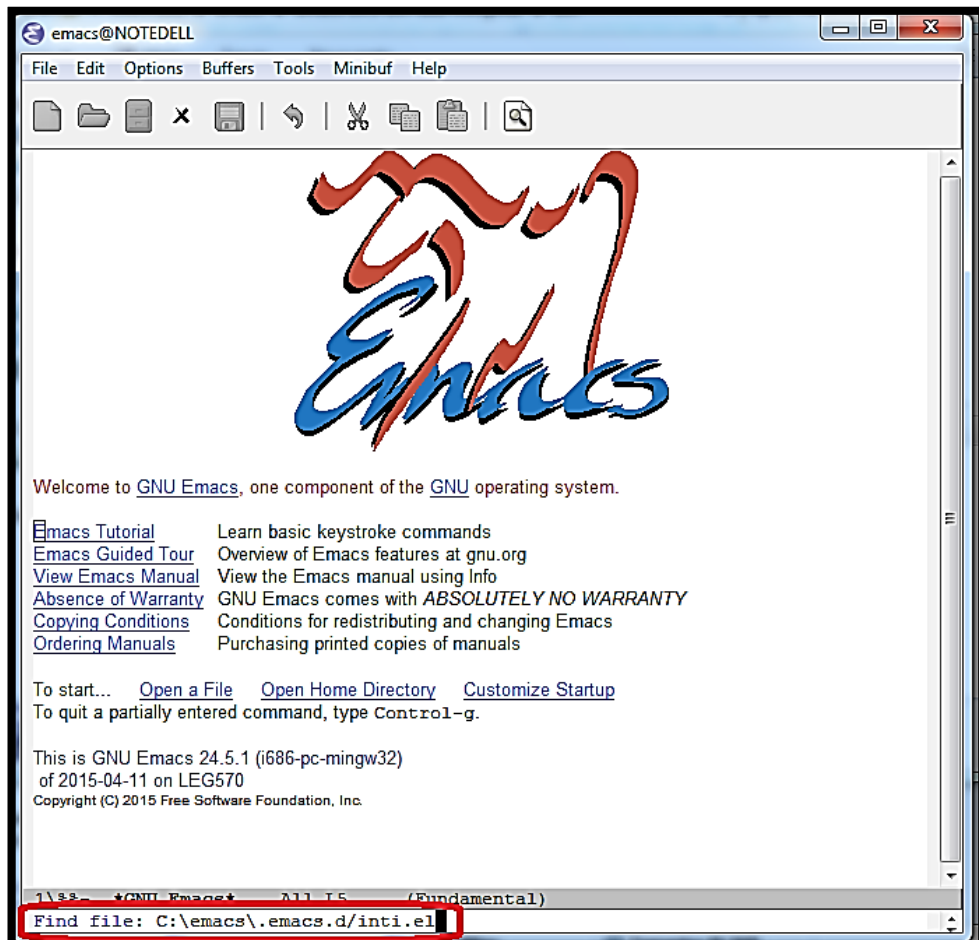
3- Abra o emacs, digite **C-x**²⁶. **C-f**²⁷ (onde **C-** é a tecla **Ctrl** – control) A linha de comando é ativada na parte inferior da janela do emacs.



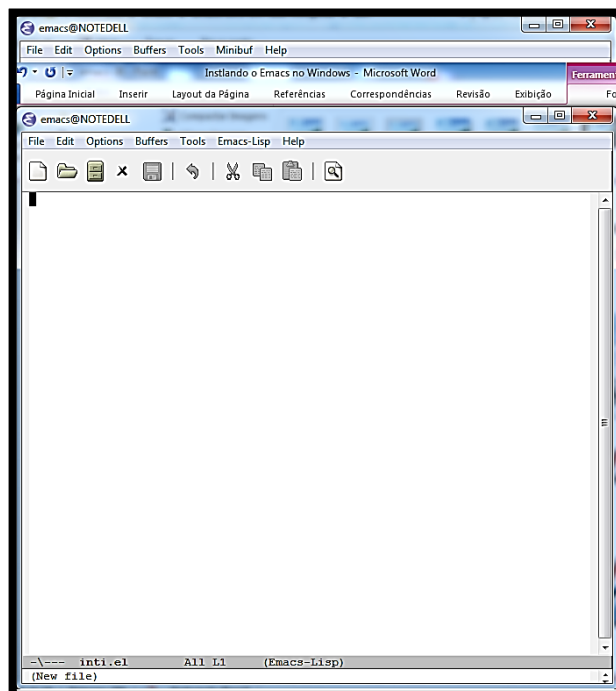
4- Apague a linha de comando e digite nela: **c:/emacs/.emacs.d/init.el** e aperte **ENTER**.

²⁶ **C-x** Aperte a tecla **Ctrl** e depois a tecla **X**. São só duas teclas, não tem o traço (-).

²⁷ **C-f** Aperte a tecla **Ctrl** e depois a tecla **f**. São só duas teclas, não tem o traço (-).



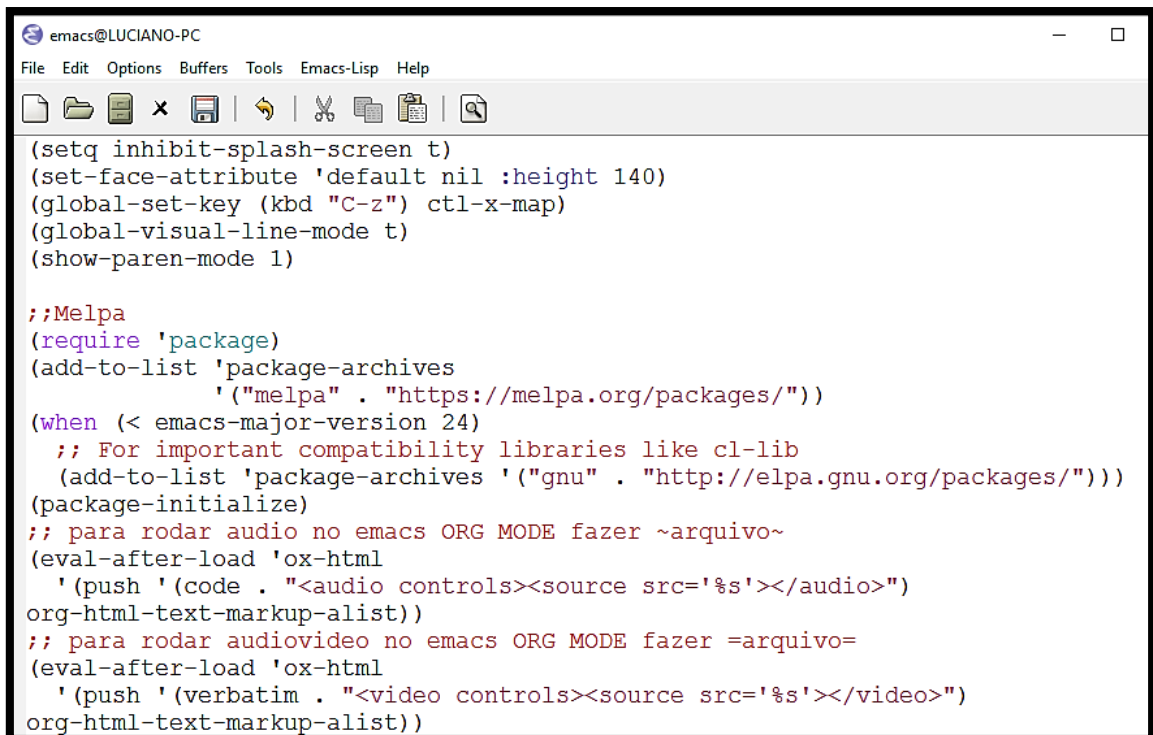
Uma página em branco no **emacs** é iniciada.



Nesta linha deve ser colocada uma configuração padrão, que pode ser sempre esta, sem modificações, portanto, não se precisa entender os comandos inseridos nela. Na mesma está especificando o tamanho da janela, que é para aceitar os caracteres especiais, checagem automática de parênteses e importação de arquivos.

Existem configurações diferentes e que serem para qualquer coisa. Algumas pessoas até postam orgulhosos que configuraram o emacs completamente em apenas 20 minutos. Como já foi dito, o **emacs** não é apenas um editor de texto, é um ambiente de programação multiparadigma completo,

Digite o texto que segue para colocar as configurações básicas e suficientes para os trabalhos a que esta dissertação se propõe.



```

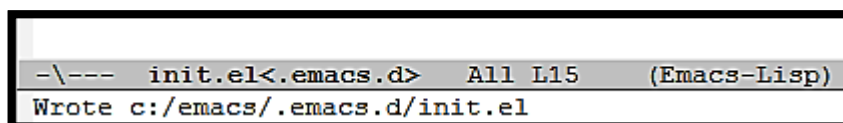
emacs@LUCIANO-PC
File Edit Options Buffers Tools Emacs-Lisp Help
[Icons]
(setq inhibit-splash-screen t)
(set-face-attribute 'default nil :height 140)
(global-set-key (kbd "C-z") ctl-x-map)
(global-visual-line-mode t)
(show-paren-mode 1)

;;Melpa
(require 'package)
(add-to-list 'package-archives
  ("melpa" . "https://melpa.org/packages/"))
(when (< emacs-major-version 24)
  ;; For important compatibility libraries like cl-lib
  (add-to-list 'package-archives ("gnu" . "http://elpa.gnu.org/packages/")))
(package-initialize)
;; para rodar audio no emacs ORG MODE fazer ~arquivo~
(eval-after-load 'ox-html
  '(push '(code . "<audio controls><source src='%s'></audio>")
    org-html-text-markup-alist))
;; para rodar audiovideo no emacs ORG MODE fazer =arquivo=
(eval-after-load 'ox-html
  '(push '(verbatim . "<video controls><source src='%s'></video>")
    org-html-text-markup-alist))

```

Feito isto, salve o arquivo digitando **C-x C-s**,

Vai aparecer a seguinte mensagem na barra de comando:



```

- \ ---  init.el<.emacs.d>  All L15  (Emacs-Lisp)
Wrote c:/emacs/.emacs.d/init.el

```

Feche o **emacs** e abra-o novamente.

Observe que a tela de entrada do emacs mudou.

```

emacs@LUCIANO-PC
File Edit Options Buffers Tools Lisp-Interaction Help File Edit Options Buffers Tools Lisp-Interaction Help
;; This buffer is for notes you don't want to save, and for Lisp
evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.

1\--- *scratch* All L5 (Lisp Interaction Wrap)
For information about GNU Emacs and the GNU system, type C-h C-a.

```

O arquivo **init.el** foi criado

Nome	Data de modificaç...	Tipo	Tamanho
init.el	09/01/2016 16:40	Arquivo EL	1 KB

Assim, o emacs está configurado e pronto para uso.

7.3 SSH – SECURE SHELL – O CAMINHO SEGURO PARA SE PROGRAMAR DIRETO NA NUVEM E PARA COMUNICAR, TROCAR DADOS ENTRE A MESMA E SEU COMPUTADOR

7.3.1 O Que É Ssh?

SSH significa: **SECURE SHELL**²⁸ -> execução de comandos à distância com segurança.

²⁸ **Shell** – Um programa, um interpretador, responsável pela comunicação entre o usuário e o computador. O mesmo geralmente, e de preferência, aceita comandos vindos de uma interface de

O **SSH** é um protocolo²⁹ de comunicação muito seguro (mesmo em redes inseguras) para se conectar remotamente a um servidor Linux (sim, nativamente Linux), através da internet ou rede local (intranet). Isto significa que você poderá se comunicar com outros computadores, com a nuvem, fazer downloads e uploads com segurança, já que todos os dados trafegados neste protocolo são criptografados com uma criptografia segura.

WINDOWS - No caso do **Putty (Anexo 4)**, permite-se realizar esta comunicação **SSH** na plataforma Windows, ainda bastante utilizada no Brasil em todos os domínios da informação, seja em trabalhos acadêmicos, profissionais e principalmente em atividades informais.

7.3.2 Vantagens do Uso do Protocolo SSH

SSH é sinônimo de SEGURANÇA.

Com o protocolo **SSH** você poderá:

- acessar **remotamente, com segurança**, o servidor através de computadores, notebooks, tablets e celulares;
- acessar o servidor, **remotamente, com segurança**, utilizando a internet ou intranet (rede doméstica);
- acessar o servidor, **remotamente, com segurança**, utilizando modems com fio, wireless, 3G e 4G;
- transferir arquivos, tanto download como upload, **com segurança**, entre servidor e cliente (**Anexo 3**);
- acessar **remotamente, com segurança**, o seu provedor de hospedagem de sites;
- executar **remotamente, com segurança**, aplicações no servidor;
- executar **remotamente, com segurança**, comandos bash³⁰ no servidor.

O SSH garante a segurança dos dados transmitidos entre cliente e servidor. Todos os dados são criptografados, inclusive senhas, e, desta forma,

linha de comando (CLI), como no DOS (sistema operacional Windows antes das interfaces visuais, hoje chamado de **prompt de comando – uma janelinha preta** - somente com comandos textuais).

²⁹ **PROTOCOLO**: Um conjunto organizado, estruturado de informações, instruções, que garantem que as mesmas serão executadas, interpretadas corretamente, em qualquer lugar onde forem utilizadas.

³⁰ **Comandos bash** - O *bash* é um interpretador de **comandos**, uma espécie de tradutor entre o sistema operacional e o usuário, normalmente conhecido como shell, no Linux.

mesmo que alguém consiga interceptá-los, a criptografia protege o acesso ao conteúdo da informação.

Cita-se a criptografia do protocolo **SSH** como vantagem, devido aos outros serviços de troca de informação, download e upload, tais como: FTP, Telnet, rLogin, outros, não criptografarem as informações entre cliente e servidor. A criptografia do SSH é segura o bastante para se confiar nas transferências de informação/dados.

7.3.3 Dados Fundamentais Para se Poder Implementar Páginas Direto na Nuvem

7.3.3.1 Quatro parâmetros, informações necessárias para abrir um terminal putty no windows (bem como em qualquer outro sistema operacional), fazer transmissão de dados segura entre desktop e nuvem usando scp.

Antes de executar o **PUTTY** (assim como no Mac e no Linux) é necessário se conhecer 4 (quatro) coisas.

- 1- O **Host Name – domínio** – é o nome do domínio utilizado para contratar sua hospedagem na Web (no caso deste exemplo, é: **coronasdkgames.com.br**)
- 2- Conhecer o **login** da sua hospedagem
- 3- Conhecer o **password**, a **senha** de administrador da sua hospedagem.
- 4- A **porta** que a sua hospedagem libera para comunicação **SSH**. No caso conforme já mostrado para a HostGator, a porta é a **2222**.

1- HOST NAME – o host name é o domínio da hospedagem (no caso do exemplo é **coronasdkgames.com.br**).

2 e 3- LOGIN (nome do usuário) e PASSORD (senha)

Primeiro, quando você contrata uma hospedagem, como na HostGator, por exemplo, você recebe um e-mail, após o pagamento ser aprovado, recebendo duas informações importantes imprescindíveis para acessar a nuvem, na sua hospedagem:

- O **login** – usuário de acesso ao **cPanel** (painel de controle) e **FTP**³¹
- O **password** - a senha de acesso ao **cPanel** e **FTP**

O E-mail: Assunto ->

Bem vindo à HostGator Brasil - Dados do seu Plano

CONTEÚDO DO EMAIL :

Bem vindo à Hostgator [REDACTED]

Você adquiriu um plano **M3**.

É muito importante que antes de utilizar o plano de hospedagem, você leia os **Termos de Serviço** e **Política de E-mails**. O não cumprimento dos termos, pode acarretar na suspensão temporária ou definitiva do seu plano de hospedagem.

Termos de Serviço: <http://www.hostgator.com.br/tos.shtml>

Políticas de Emails: <http://www.hostgator.com.br/emails.shtml>

Lembrando que para planos compartilhados (P, M, G & Revendas) existe um limite no envio de e-mails, onde você poderá realizar no máximo 500 disparos por hora, com delay de no mínimo 6 segundos, totalizando no máximo o envio de 5.000 mensagens por dia. O não cumprimento poderá acarretar em suspensão temporária ou definitiva da conta.

Prezando pelo seu conforto, a HostGator disponibilizou algumas informações importantes para ajudá-lo na configuração e utilização do seu plano de hospedagem:

<http://hostgator.com.br/primeirospassos.shtml>

- Informações básicas sobre seu plano de hospedagem:

O domínio principal é: coronasdkgames.com.br

Usuário de acesso ao cPanel e FTP: [REDACTED] **login**

Senha de acesso ao cPanel e FTP: [REDACTED] **password**

- Informações sobre as configurações de DNS:

DNS Master: ns64.hostgator.com.br IP Master: 50.97.97.24
 DNS Slave: ns65.hostgator.com.br IP Slave: 50.97.97.25

- O registro de domínio e propagação de DNS pode levar até 24 horas. Neste caso, enquanto o domínio não estiver acessível, navegue pelo painel administrativo por um link provisório que pode ser encontrado abaixo:

Painel Administrativo (cPanel): <http://50.97.97.24/cpanel>
 Visualização do Website:: <http://50.97.97.24/~coron759/>

- Após o período de propagação, utilize o acesso direto:

Painel Administrativo (cPanel): <http://coronasdkgames.com.br/cpanel>
 Visualização do Website: <http://coronasdkgames.com.br>

4- PORTA PARA UTILIZAR A COMUNICAÇÃO SEGURA NA NUVEM UTILIZANDO SSH

Novamente, utilizando o Hostgator como exemplo, e o domínio coronasdkgames.com, o procedimento é o seguinte.

- 1- No email da figura anterior, tem a informação do endereço do Painel Administrativo (cPanel), no caso:

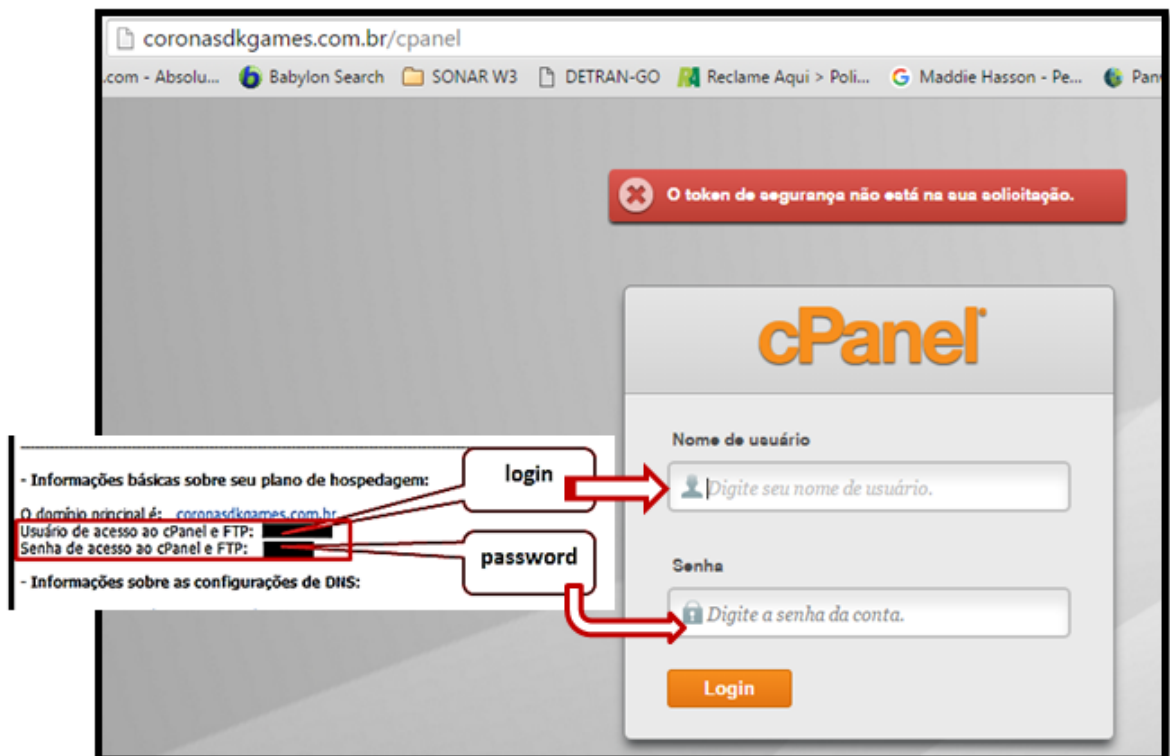
³¹ **FTP** ou **File Transfer Protocol** (em português, Protocolo de Transferência de Arquivos) é uma forma bastante rápida e versátil de transferir arquivos, sendo uma das mais usadas na Internet.

Painel Administrativo (cPanel): <http://coronasdkgames.com.br/cpanel>

- 2- Entre com este endereço no browser de sua escolha

<http://coronasdkgames.com.br/cpanel>

- 3- A janela de login do cPanel é aberta. Entre com Nome do usuário que é o login, o usuário de acesso ao Cpanel mostrado anteriormente e com a senha de acesso ao Cpanel



- 4- O Painel de Controle é aberto. Clique em Ativar/Desativar SSH. Se na sua hospedagem não tiver algo parecido, no campo de busca, digite SSH para localizar onde entrar. Os dados que você precisa estão, no HostGator, em HostGator Links.

7.3.4 Terminal SSH no Windows Com o Putty

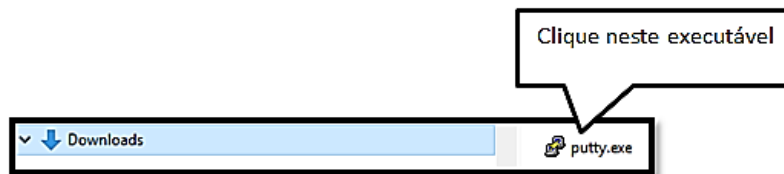
7.3.4.1 Abrindo um terminal SSH com o Putty

O **PUTTY** não precisa de instalação, como já foi dito anteriormente neste capítulo, assim, após clicar no executável baixado, um terminal **SSH** já será aberto para você trabalhar.

De posse das 4 (quatro) informações descritas anteriormente (se não tiver, nem continue, procure-as com quem contratou pra você a hospedagem), você está pronto para iniciar, abrir um terminal **SSH** no Windows, utilizando o **PUTTY**.

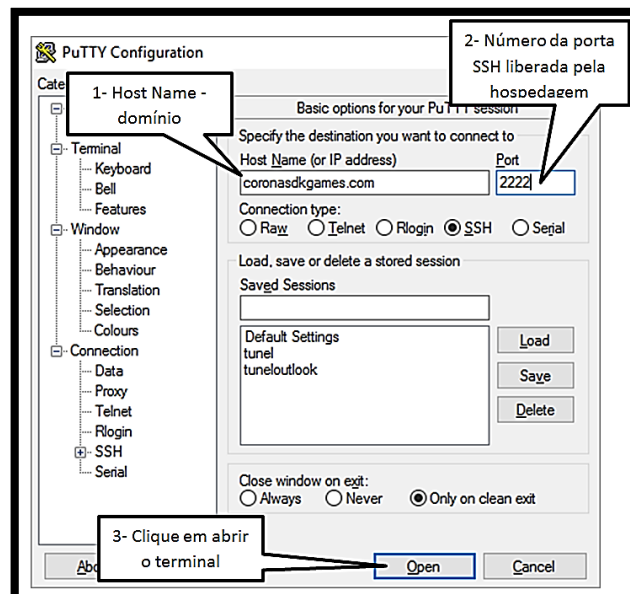
A única coisa difícil, por assim dizer, para utilizar o **PUTTY** e abrir um terminal **SSH** com ele, é conhecer estas 4(quatro) informações: o **hostname (domínio)**, a **porta SSH**, o **login** e o **password**.

Ao clicar no executável do **PUTTY**, que normalmente estará na pasta Download (ou onde você tiver escolhido salvar),



a página de configuração do **PUTTY** é aberta.

Entre com o **Host Name** (o **nome do domínio** de registro na hospedagem) e a porta para comunicação **SSH** liberada pelo servidor.



Ao fazer isto, o terminal é aberto, onde serão solicitadas as outras duas informações:

- primeiro, o **login**,
- depois, o **password**.

Dando tudo certo, o terminal é aberto com o prompt formado da seguinte forma:

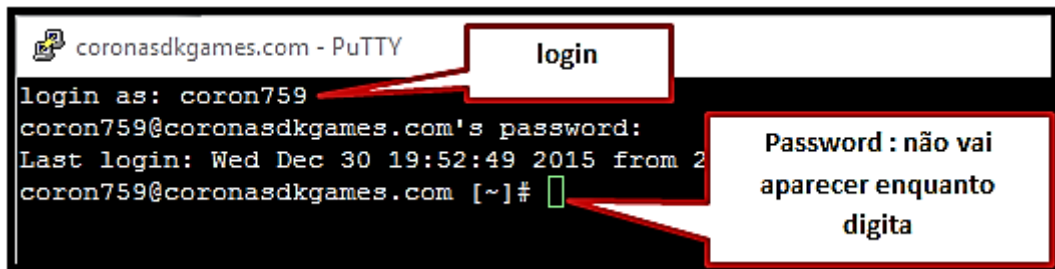
login@domínio [~] #

No caso do exemplo, o login é **coron759** e o domínio **coronasdkgames.com** (já informado ao **PUTTY** na janela de configuração).

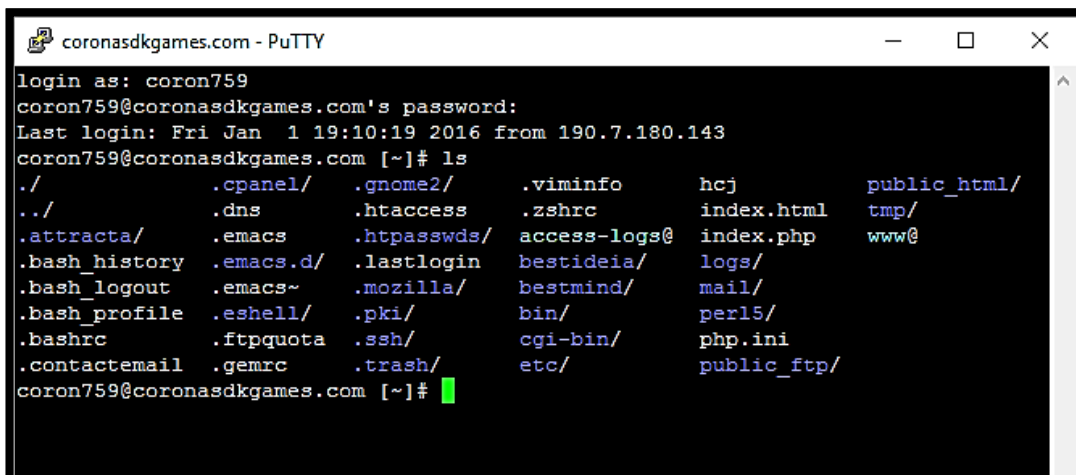
Assim, o prompt de comando deverá ficar:

corona759@coronasdkgames.com.br [~]#

Se algo der errado, vier uma mensagem de erro e não esse prompt, feche a janela do terminal e execute o **PUTTY** quantas vezes forem necessárias até que você entre corretamente o login e o password (dê **ENTER** após entrar).



Digite **ls** (vem de lista, listar) e clique no botão **ENTER**. Este comando lista todo o conteúdo do que está na **coronasdkgames.com.br** (no caso do exemplo).



Isto ocorrendo, é mais uma confirmação que o terminal está funcionando e que você já está conectado à nuvem, no local de sua hospedagem. Neste momento, não faça nada que possa destruir o conteúdo da mesma, caso saiba executar os comandos do **bach** (**shell**). Nesta linha, você tem direito de fazer tudo. Apagar, renomear, etc... etc...

7.3.4.2 Usando o terminal SSH do Putty: comandos básicos necessários à esta pesquisa.

Limpando a janela do terminal: basta digitar o comando **clear** e apertar a tecla **ENTER**

```

coronasdkgames.com - PuTTY
login as: coron759
coron759@coronasdkgames.com's password:
Last login: Fri Jan  8 11:40:59 2016 from 189.40.67.143
coron759@coronasdkgames.com [~]# ls
./                .cpanel/         .gnome2/         .viminfo         hcj              php.ini
../              .dns             .htaccess        .zshrc           index.html       public_ftp/
.attracta/       .emacs          .htpasswd/       access-logs@    index.php        public_html/
.bash_history    .emacs.d/       .lastlogin       bestideia/       jornal/          tmp/
.bash_logout     .emacs~         .mozilla/        bestmind/        logs/            www@
.bash_profile    .eshell/        .pki/            bin/             luciano/
.bashrc          .ftpquota       .ssh/            cgi-bin/         mail/
.contactemail   .gemrc          .trash/          etc/             perl5/
coron759@coronasdkgames.com [~]# clear

```

ENTER

```

coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]#

```

Entrando no diretório **public_html**. Para isto, utilize o comando **cd** de mudar (change) de diretório(directoy).

Digite: **cd public_html/** e aperte a tecla **ENTER**.

```

coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/

```

ENTER

```

coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]#

```

Para testar, crie(make) um diretório (**directory**) de nome **mestrado** dentro de **public_html**. O comando é **mkdir** **mestrado** e aperte a tecla **ENTER**.

```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]# mkdir mestrado
```

ENTER

```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]# mkdir mestrado
coron759@coronasdkgames.com [~/public_html]#
```

Observe que não retorna nenhuma mensagem de sucesso ou de erro, apenas o prompt de comando [coron759@coronasdkgames.com \[~/public_html\]#](#), o que significa que a operação foi um sucesso.

Para comprovar que tudo deu certo, digite o comando **ls** e aperte a tecla **ENTER**. Verá que o diretório **mestrado** foi criado.

```
coronasdkgames.com - PuTTY
accountinggateway.php  helcio.html          productimage.php
addons/                 hominispress/        products.php
admin/                  includes/             productupdates.php
ajax/                   index.php             programasc.rar
alltut.pdf              init.php              racket/
b.jpg                   internet.html         remote.php
bestideia/              javascript/           rkt/
bestmind/               jornal/               rss.php
biblioteca/             language/             scriptc.html
bottom.php              lib/                  scriptc2.html
brands.php              login.php             search.php
business/               loja2/                sitemap.php
cache/                  loja3/                subscribe.php
camargo.html            mapas2n/              tags.php
captcha.php             mega/                 templates/
cart.php                mestrado/             top.php
```

Entre na pasta **mestrado**, digite o comando **ls** e verá que a pasta está criada, mas vazia.

```
compare.php             mypage.k*            wishlist.php
config/                 natali.jpg           xml.php
convertMIDI.rar         news.php             xmlsitemap.php
debussy.jpg             noradi/
coron759@coronasdkgames.com [~/public_html]# cd me
mega/                   mestrado/
coron759@coronasdkgames.com [~/public_html]# cd mestrado/
coron759@coronasdkgames.com [~/public_html/mestrado]# ls
./ ../
coron759@coronasdkgames.com [~/public_html/mestrado]#
```

Para transitar, mover, entre diretórios, o comando básico, como já visto, é o **cd**.

O comando **cd** pode ser acrescido de outros comandos para realizar algumas tarefas básicas. Como exemplo, parta do diretório do exemplo anterior: o diretório **mestrado**. A partir dele, os comandos para mudar de diretório são:

- Digite **cd /** e aperte a tecla **ENTER** para ir ao diretório raiz.

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd /
coron759@coronasdkgames.com [/]#
```

- Digite **cd** e aperte a tecla **ENTER** para ir ao seu diretório pessoal.

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd
coron759@coronasdkgames.com [~]#
```

- Digite **cd ..** e aperte a tecla **ENTER** para acessar um diretório de nível acima do atual.

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd ..
coron759@coronasdkgames.com [~/public_html]#
```

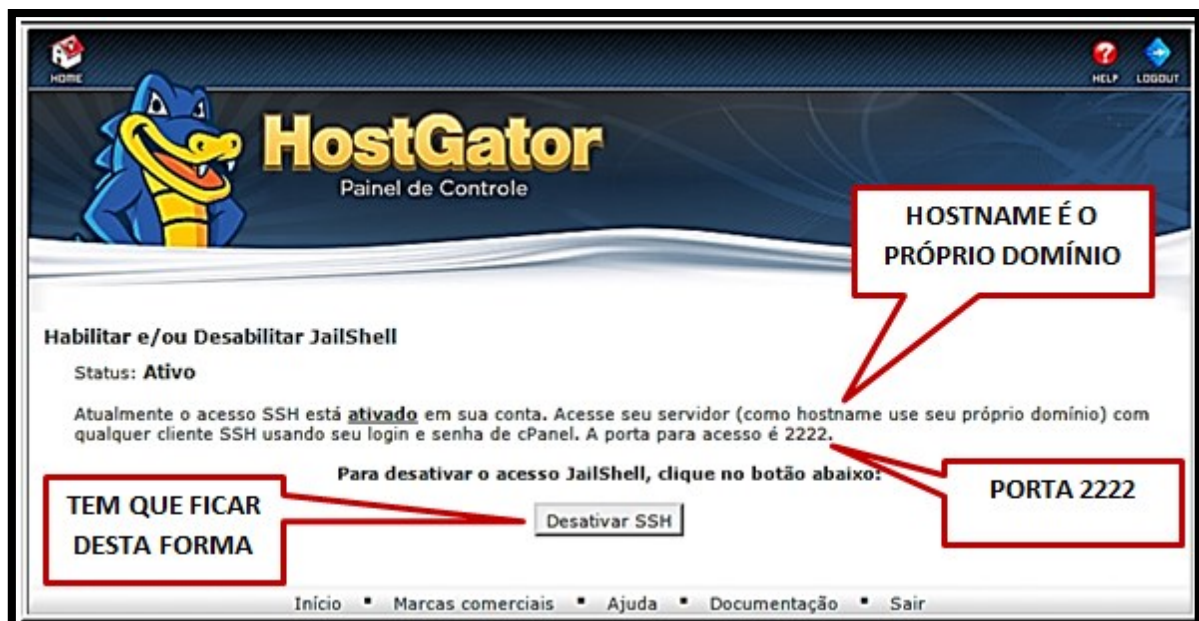
- Digite **cd -** e aperte a tecla **ENTER** para voltar ao diretório anterior

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd -
/home/coron759/public_html
coron759@coronasdkgames.com [~/public_html]#
```

OBS. não se pode esquecer de dar um espaço entre o comando **cd** e o respectivo complemento (**/** ou **..** ou **-** ou **~**).



Ao clicar em Ativar/Desativar SSH, a janela de ativação do SSH é aberta contendo a informação do número da porta disponibilizada pela Hostgator, ou sua hospedagem, para acessar a porta de comunicação segura SSH, no caso, a 2222.



7.4 EMACS-LISP

7.5 CONCEITOS BÁSICOS DE ELISP, UMA LINGUAGEM DE PARADIGMA FUNCIONAL.

Antes de entrar nos conceitos e programação básica em **ELISP**, é importante que alguns conceitos sejam revistos, principalmente pelo público alvo dos objetivos que são os jornalistas e afins no respectivo domínio.

7.5.1 Funções, Domínio, Imagem - Características Básicas, Formalização:

A linguagem **LISP (ELISP)** possui paradigma funcional. A mesma representa, modela os eventos, comandos, objetos e morfismos com o paradigma matemático de funções.

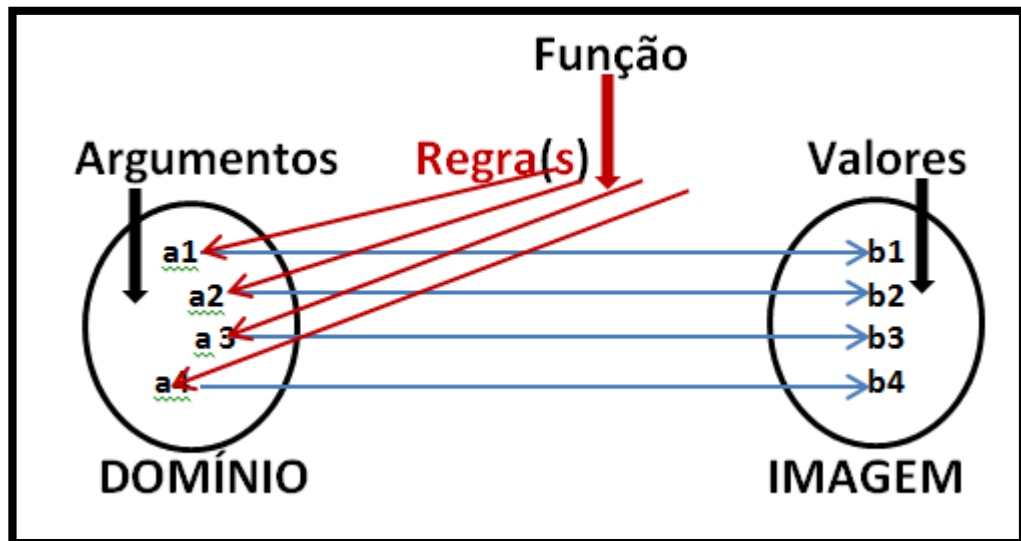
A Teoria Moderna dos Conjuntos foi formalizada por Georg Cantor e Richard Dedekind em 1870.

Desta teoria gerou-se a possibilidade da modelagem dos objetos (conjuntos de elementos com mesma lei de formação) e das ações sobre e entre eles (os morfismos).

As funções, ou morfismos, possuem regras de inferência capazes de inferir valor (o resultado a aplicação da função a um ou mais elementos (argumentos) de um conjunto chamado de Domínio).

O conjunto formado por todos os valores inferidos é chamado de Imagem.

Desta teoria axiomática de conjuntos, Zermelo em 1908 e Abraham Fraenkel em 1922 descreveram formalmente o conjunto Imagem, também denominado por conjunto solução, já que o mesmo contém todos os valores inferidos pelas regras de inferências aplicadas a todos os argumentos do domínio.



O elemento do domínio é chamado argumento da função e o elemento do contradomínio associado ao argumento é chamado valor.

No começo do século, matemáticos e filósofos se preocupavam com métodos que, a partir do argumento, pudessem calcular o valor de uma função. Assim, dado um elemento do domínio, eles buscavam descobrir, desenvolver um processo mecânico para se avaliar, encontrar o correspondente elemento do contradomínio.

Este processo mecânico deveria ser constituído por uma sucessão de aplicações de regras de inferências, as quais determinariam o valor da função. Os filósofos da matemática deram, a este processo, o nome de computação do valor da função. Foi Alonzo Church (Church, 1941) quem propôs o primeiro sistema formal de computação.

A Modelagem de objetos, comandos e eventos diretamente através de funções somente é possível quando uma ação (regra) for aplicada a um elemento (argumento), só se obtém uma resposta. Sempre que se aplicar esta ação (regra) ao mesmo elemento, a resposta deverá ser a mesma (transparência referencial). Quando isto não ocorre, tem-se que fazer algoritmos recursivos para encontrar mais de uma solução, ou modelar através de linguagens lógicas que implementem relações sem necessidade de algoritmos recursivos.

Exemplo:

<p>Função</p> <p>seno 30° - Uma só resposta:</p> <p>→ 0,5</p>

<p>Relação</p> <p>irmão de Tatiana – duas respostas:</p> <p>→ Daniel</p> <p>→ Thales</p>
--

Assim, uma função possui portanto:

- Nome
- Argumento(s)
- Regra(s) de inferência

Com o seguinte formato:

nome arg(s) = regra

onde: - o símbolo '=' é um separador, o qual separa a chamada da função da regra.

- **regra** é a ação que vai ser aplicada ao argumento, inferindo um valor.

Exemplo de chamada de uma função:

seno 30°, onde: **seno** é o nome da função e **30°** é o argumento.

Observe que o nome da função vem primeiro e o argumento depois.

Assim, classicamente, o operador, a função, é pré-fixado.

7.5.2 Sintaxe

No LISP e outras linguagens, os operadores, as funções são prefixas, como na definição matemática.

Assim, uma função que soma dois argumentos deveria possuir uma chamada da seguinte forma:

Matematicamente: soma x y

Em LISP, os operadores, as chamadas de qualquer função, comando ou evento, são pré-fixadas, com a diferença da matemática que sempre deve-se colocar entre parênteses: **(soma x y)**

Matematicamente, comumente a função soma seria especificada da seguinte forma:

$$\text{soma } x \ y = x + y$$

Deve-se observar que se tem uma chamada prefixada seguida de uma regra infixa. Isto causa, ao aprendiz em matemática, uma dificuldade lógica em adquirir tal conhecimento. Com o tempo esta estrutura é assimilada pelo cérebro e nem mais se percebe.

Em Racket, todas as definições e chamadas são prefixadas, o que facilita o aprendizado, seguindo a regra da ortogonalidade matemática e evitando a ambiguidade na formalização das funções.

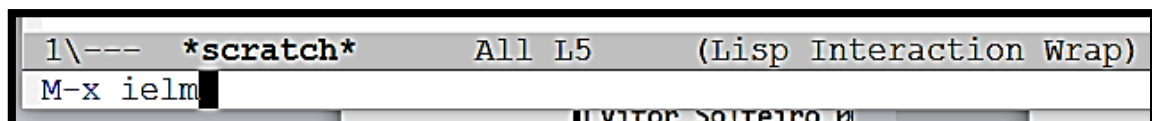
7.6 CRIANDO UMA FUNÇÃO EM ELISP:

Para se testar os conceitos, na prática, é necessário ter um ambiente de programação para se avaliar os resultados dos mesmos. Como já foi dito, **Emacs** é muito mais que um editor, é um ambiente de programação multiparadigma que possui como base principal o funcional, já tendo nativo o **ELISP: Emacs LISP**.

Assim, antes de continuar, começar a testar a criação de funções e outros recursos, é importante que se saiba como usar o Emacs como ambiente de programação do **ELISP**³².

Para testar o Elisp, você pode utilizar o próprio Emacs.

Para tanto, estando no Emacs, para entrar no modo de programação do **ELISP**. Para testar as funções e estrutura da linguagem, digite **M-x** (onde **M** significa **Alt**). ao fazer isto, a linha de comando na parte inferior da janela do emacs é aberta. Digite **ielm**

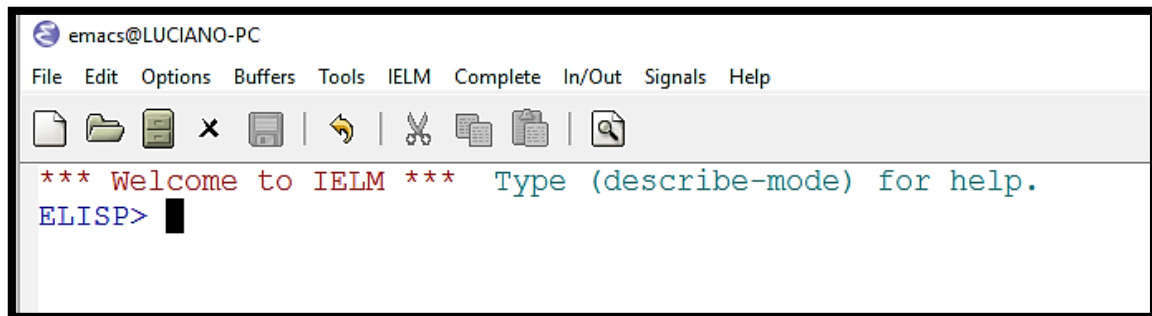


```

1\--- *scratch* All L5 (Lisp Interaction Wrap)
M-x ielm
  
```

Aperte **ENTER**. O Emacs ficará no modo **ELISP**.

³² Detalhes de uso do **Emacs** com os respectivos atalhos em linha de comando, bem como da programação em **Org-Mode** e conceituação de cliente-servidor são apresentadas nos **ANEXOS 1, 2 3 e 4**.



```
emacs@LUCIANO-PC
File Edit Options Buffers Tools IELM Complete In/Out Signals Help
*** Welcome to IELM *** Type (describe-mode) for help.
ELISP> █
```

O **ANEXO 5** apresenta o **HELP** do emacs – **ELISP**. Ou, se preferir, digite **(help)** no prompt do **ELISP** e aperte **ENTER**, para abrir o ambiente em dois e mostrar o **HELP**.



```
emacs@LUCIANO-PC
File Edit Options Buffers Tools IELM Complete In/Out Signals Help
*** Welcome to IELM *** Type (describe-mode) for help.
ELISP> (help)█
```

The screenshot shows the Emacs window titled 'emacs@LUCIANO-PC'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Help-Mode', and 'Help'. The toolbar contains icons for file operations and editing. The main text area displays the following content:

```

*** Welcome to IELM *** Type (describe-mode) for help.
ELISP> (help)
|
1\*- *ielm* All L3 (IELM:run on *ielm* Wrap)
You have typed , the help character. Type a Help option:
(Use SPC or DEL to scroll through this text. Type q to exit the Help command.)

a PATTERN Show commands whose name matches the PATTERN (a list of words
or a regexp). See also the `apropos' command.
b Display all key bindings.
c KEYS Display the command name run by the given key sequence.
C CODING Describe the given coding system, or RET for current ones.
d PATTERN Show a list of functions, variables, and other items whose
documentation matches the PATTERN (a list of words or a regexp).
e Go to the *Messages* buffer which logs echo-area messages.
f FUNCTION Display documentation for the given function.
F COMMAND Show the Emacs manual's section that describes the command.
g Display information about the GNU project.
h Display the HELLO file which illustrates various scripts.
i Start the Info documentation reader: read included manuals.
1\%*- *Metahelp* Top L1 (Help Wrap)
Type one of the options listed, or SPACE or DEL to scroll:

```

Apertando a tecla **q** do teclado do computador, volta-se ao prompt do **ELISP**.

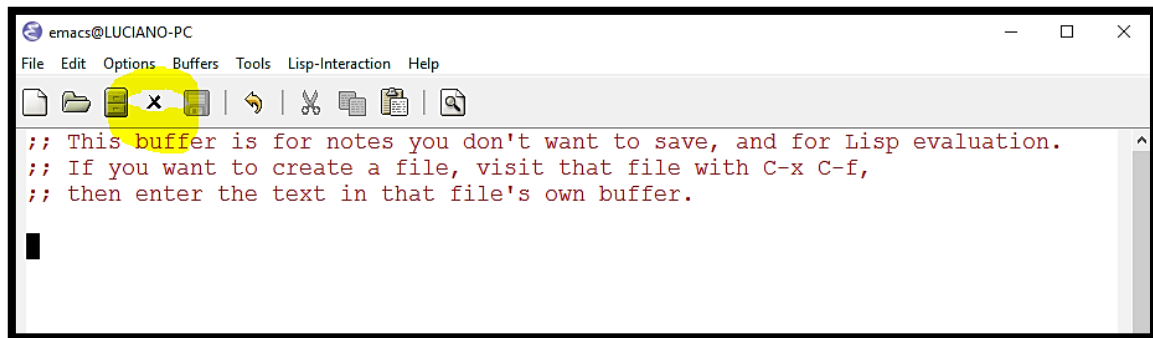
The screenshot shows the Emacs window titled 'emacs@LUCIANO-PC'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'IELM', 'Complete', 'In/Out', 'Signals', and 'Help'. The main text area displays the following content:

```

*** Welcome to IELM *** Type (describe-mode) for help.
ELISP> (help)
nil
ELISP>

```

Para fechar o ambiente de programação do **ELISP**, basta clicar no botão **X** do menu da barra superior do **EMACS**.



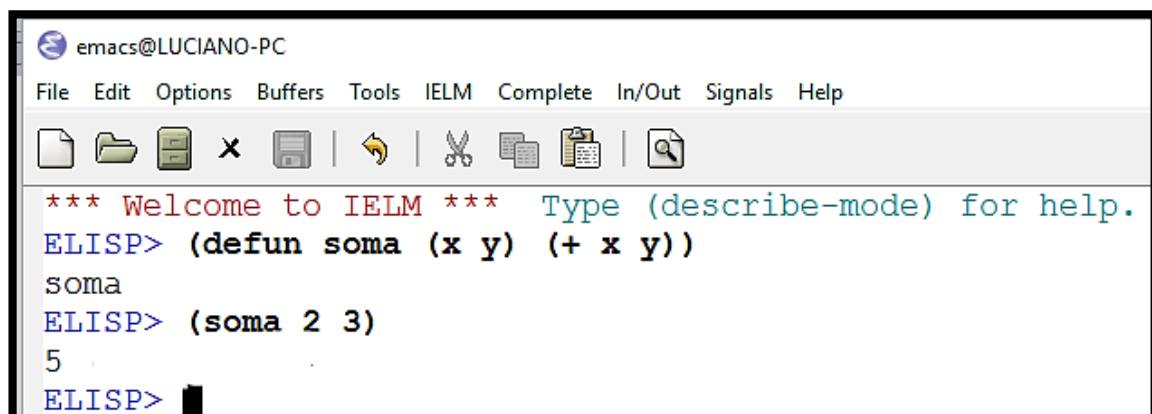
Pronto, com estas instruções básicas já se pode aprender um pouco do LISP do **Emacs**, o qual possui grande parte do **Common Lisp**³³ compatível com o uso na hospedagem na nuvem sem que a mesma bloqueie os recursos inseridos nas páginas que serão criadas com ele.

7.6.1 Criando Uma Função No Elisp

Para se criar uma função, que infere um valor apenas para cada argumento, utiliza-se a primitiva **defun**, seguida da chamada da função (entre parênteses e prefixada), seguida da(s) regra(s) de inferência (também prefixada).

```
( defun nome_função ( argumento(s) ) ( regra ) )
```

Eis o exemplo da implementação em **ELISP** da função **soma** com dois argumentos, seguido do teste com os argumentos 2 e 3, resultando no valor 5.



³³ **Common Lisp** - O LISP mais completo e de melhor performance entre todos os dialetos LISP, além de ser gratuito. A melhor versão atual do Common Lisp é o **SBCL**, só disponível para computadores pessoais mais comuns a menos de uma década. https://en.wikipedia.org/wiki/Common_Lisp

Observe que é simples e intuitivo, totalmente aderente à definição matemática, onde o uso dos parênteses eliminam o separador (símbolo ‘=’) entre chamada da função e regra, evitando a sobrecarga do símbolo ‘=’ que gera ambiguidade na interpretação que pode ser interpretado erroneamente como sendo um operador de comparação ou atribuição.

7.6.1.1 Funções com eventos

Nem sempre uma função executa um comando diretamente, como somar dois valores. Na vida temos duas situações que modelamos na computação:

- **comandos**
- **eventos**

Um **comando** é executado imediatamente quando chamado, como no caso da função **soma**.

Já um **evento** caracteriza algo que pode ou vai acontecer, e, caso acontecendo, um ou mais comandos deverão ser executados, dependendo do que ocorrer no evento. Diz-se que existem condições para se decidir que comando(s) deve(m) ser executado(s) quando determinado evento ocorrer.

Um exemplo disto é:

Se ocorrer o evento, a condição do argumento da função **executa** ser maior que 5, deve-se executar o comando (a regra) de retornar o dobro do valor, mas, se o argumento da função for menor ou igual a 5, deve-se executar o comando (a regra) de retornar o triplo do valor.

A condição **se**, em **ELISP**, é substituída pela palavra **if**

A estrutura de testar um evento condicional é:

```
(if (condição) (regra se for verdade) (regra se for falso) )
```

A implementação em **ELISP** fica:

```

ELISP> (defun executa (x)
         (if (> x 5) (* 2 x) (* 3 x)))
executa
ELISP> (executa 4)
12
ELISP> (executa 6)
12

```

Onde:

- condição = (> x 5)
- regra se for verdade = (* 2 x)
- regra se for falso = (* 3 x)

7.6.2 Outras Características, Conceitos E Funções Em Elisp: Manipulando Strings (Texto) E Listas

Nesta dissertação, o foco é trabalhar com textos e dados (que podem ser listas de textos, por exemplo).

Textos são agrupamentos (vetor) de caracteres formando palavras e frases. Em computação chamamos a este tipo de dados de **String**.

Uma **String** em **ELISP** é formalizada por um conjunto de caracteres delimitados por aspas.

Exemplo:

"Dissertação de Mestrado"

Existem várias funções para manipular **strings** no **ELISP** que são bastante úteis para extrair informações, modificar ou formatar textos.

Por exemplo: - Separar a **string "Dissertação de Mestrado"** nas palavras contidas nela. Nesta frase, o separador é o caractere de espaço. Mas, em outras frases, podem ser: ",", ":", ";", "-" e outros sinais, inclusive os de pontuação.

O **ELISP** possui uma função para separar(**split**) strings de várias formas. Uma delas é simplesmente separar as strings pelo caractere de espaço. A mesma possui o nome de **Split-string**.

O formato de uso da mesma é:


```
(Split-string string caractere_separação)
```

OBSERVE: O separador utilizado foi o de espaço, ou seja: " "

```
ELISP> (split-string "Dissertação de Mestrado" " ")
("Dissertação" "de" "Mestrado")
```

Neste caso especial, onde o separador é o espaço, pode-se suprimir o uso do mesmo:

```
ELISP> (split-string "Dissertação de Mestrado")
("Dissertação" "de" "Mestrado")
```

A função **Split-string** criou-se uma lista com três substrings, palavras do texto, o que é bastante útil em análise de textos.

Imagine, agora, uma string, um texto, contendo duas frases, onde as mesmas são separadas pelo caractere ".".

TEXTO: "Hoje está um dia lindo. A previsão do tempo é tempo bom o dia todo."

Em vez de se separar o texto em palavras, pode-se desejar identificar as frases do texto. Neste caso, explicita-se na função **split-string** o caractere de separação, ou seja, o caractere ".".

```
ELISP> (split-string "Hoje está um dia lindo. A previsão do tempo é tempo bom o dia todo." "\\.")
("Hoje está um dia lindo" " A previsão do tempo é tempo bom o dia todo" "")
```

Se houver mais de um separador junto, esta função devolve strings vazias nas redundâncias, o que pode comprometer a análise do texto.

```
ELISP> (split-string "Hoje está um dia lindo... A previsão do tempo é tempo bom o dia todo." "\\.")
("Hoje está um dia lindo" "" "" " A previsão do tempo é tempo bom o dia todo" "")
```

Para evitar que isto ocorra, basta colocar o caractere **t** (true) no fim da chamada da função, antes do último parêntese.

```
ELISP> (split-string "Hoje está um dia lindo. A previsão do tempo é tempo bom o dia todo." "\\.") t)
("Hoje está um dia lindo" " A previsão do tempo é tempo bom o dia todo")
```

No caso, não só no **ELISP**, como na maioria das linguagens de computação, quando um caractere, um símbolo, já é utilizado na linguagem com uma função, uma ação pré determinada, para que o mesmo seja visto pelo programa como um caractere, o mesmo deve vir precedido de duas contra-barras "\". Assim, o caractere "." deve ser utilizado na seguinte forma: "\.", conforme exemplo.

Observe o quando uma simples função já dá uma certa "inteligência" a um sistema que a utilizar, como será feito no próximo capítulo com um exemplo simples de **text mining** de uma busca de informações em um arquivo texto.

Uma das ferramentas computacionais modernas mais poderosas para manipular, extrair informações de texto são as **EXPRESSÕES REGULARES (REGEXPs – Regular Expressions)**, as quais são objeto do próximo item.

7.6.3 Expressões Regulares Em Elisp

Por ser o LISP, ELISP é uma linguagem muito rica em recursos, colocar neste documento todas as possibilidades e combinações de expressões regulares disponibilizadas por esta linguagem demandariam várias páginas, poluindo o documento. Assim, seguem links que tratam diversos tipos de expressões regulares:

- 1- Usando contra-barra (backslash) em expressões regulares:

https://www.gnu.org/software/emacs/manual/html_node/elisp/Regexp-Backslash.html#Regexp-Backslash

Exemplo de alteração: '|' significa: uma coisa ou outra. Como explicado para o caractere ponto, quando um caractere já é utilizado na linguagem para outra função, deve-se acrescentar um backslach antes. Assim, a alteração fica assim: \|

Usando o exemplo da última **split-string**, colocando como separador o caractere '.' (\.), onde já usamos o backslash para utilizá-lo, acrescentando mais uma frase, separando com o sinal de ! (\!), tem-se que o texto será separado ou pelo caractere '.' ou o caractere '!'.

```
ELISP> (split-string "Hoje está um dia lindo. A previsão do tempo é tempo
bom o dia todo! Chove." "\\.\|\\|!" t)
("Hoje está um dia lindo" " A previsão do tempo é tempo bom o dia todo" "
Chove")
```

Usando o caractere espaço em vez do caractere !, observe que o texto será todo separado em palavras e que a exclamação (!) permanece grudada na palavra todo, já que não foi usada como separador.

```
ELISP> (split-string "Hoje está um dia lindo. A previsão do tempo é tempo
bom o dia todo! Chove." "\\.|\\| " t)
("Hoje" "está" "um" "dia" "lindo" "A" "previsão" "do" "tempo" "é" "tempo"
"bom" "o" "dia" "todo!" "Chove")
```

7.6.4 Manipulando Listas

Pegar elementos em uma lista, em ELISP é simples.

Antes, deve-se conhecer a estrutura de dados LISTA.

De uma forma macro, uma lista, em ELISP(ou outro LISP qualquer) é um conjunto de elementos entre parêntes, iniciado por um apóstrofo (coate), separados por um caractere de espaço.

Exemplos:

- '(1 2 3 4)
- ("casa" 234)

Uma lista possui uma **cabeça** e uma **cauda**:

CABEÇA DE UMA LISTA: **É o primeiro elemento da lista**

A função para pegar a cabeça de uma lista em ELISP é: **car**

Exemplo:

```
ELISP> (car '(1 2 3 4))
1
ELISP> (car '("casa" 234))
"casa"
```

CAUDA DE UMA LISTA: **É a lista sem o primeiro elemento, ou seja, a cauda de uma lista sempre será uma lista.**

A função para pegar a cauda de uma lista em ELISP é: **cdr**

```

ELISP> (cdr '(1 2 3 4))
(2 3 4)

ELISP> (cdr '("casa" 234))
(234)

```

Para facilitar pegar outros elementos, pode-se usar contrações de **car** e **cdr**.

Por exemplo: **cadr** pega o **car** do **cdr** de uma lista, ou seja: a cabeça da cauda. Observe, tendo como base as respostas do exemplo anterior:

```

ELISP> (cadr '(1 2 3 4))
2
ELISP> (cadr '("casa" 234))
234

```

7.6.5 Lendo Arquivos

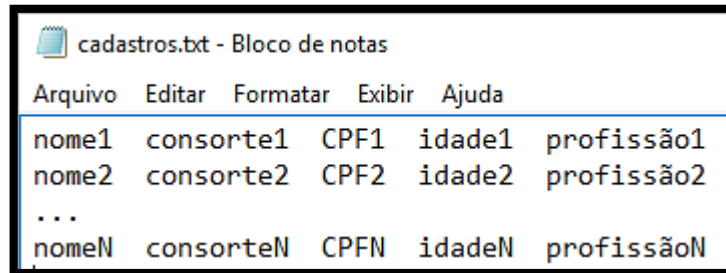
No exemplo do próximo capítulo, um arquivo colocado na nuvem será utilizado como um banco de dados para que seja realizada uma busca (text mining) no mesmo. Para tanto, é necessário se saber como ler um arquivo em **ELISP**.

Ler arquivo não é um processo simples em nenhuma linguagem, caso se deseje ter um controle do que se está lendo e como se deseja especificar o formato de como se deseja retornar o que for lido.

7.6.5.1 Lendo o arquivo e retornando uma string.

A seguir, é apresentada uma função que retorne uma string com o conteúdo do arquivo, mantendo todos os caracteres da mesma.

Como exemplo, dado o arquivo texto localizado em **c:/jornal_taty_cadastros.txt:**

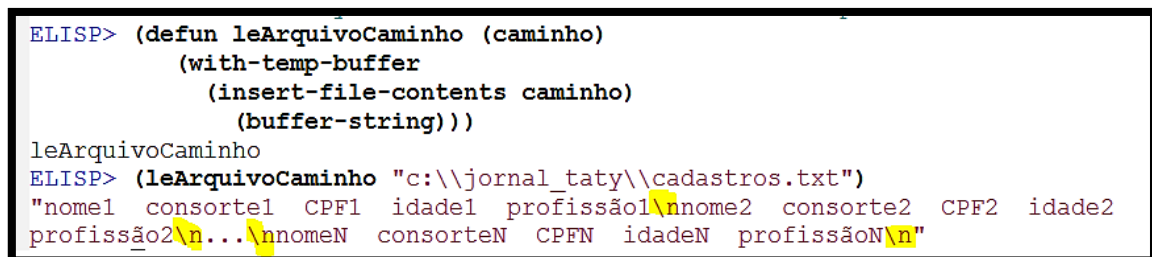


```

nome1 consorte1 CPF1 idade1 profissao1
nome2 consorte2 CPF2 idade2 profissao2
...
nomeN consorteN CPFN idadeN profissaoN

```

A função **leArquivoCaminho**, abaixo definida, faz esta leitura e devolve o arquivo texto como uma string. Observe que na string tem-se o comando `\n` que significa **saltar linha** que foi preservado pela função.



```

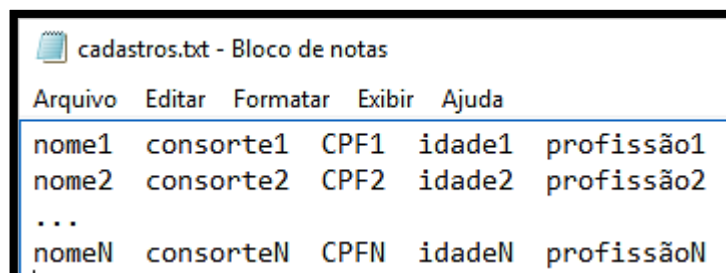
ELISP> (defun leArquivoCaminho (caminho)
  (with-temp-buffer
    (insert-file-contents caminho)
    (buffer-string)))
leArquivoCaminho
ELISP> (leArquivoCaminho "c:\\journal_taty\\cadastros.txt")
"nome1 consorte1 CPF1 idade1 profissao1\nnome2 consorte2 CPF2 idade2
profissao2\n...\nnomeN consorteN CPFN idadeN profissaoN\n"

```

7.6.5.2 Lendo o arquivo e retornando uma lista de string dado um separador ou uma expressão regular como separador.

Esta é uma função importante para este trabalho quando se deseja **analisar**, fazer text mining em frases de um arquivo texto, por exemplo.

Como exemplo, dado o mesmo arquivo texto localizado em `c:/journal_taty_cadastros.txt`:



```

nome1 consorte1 CPF1 idade1 profissao1
nome2 consorte2 CPF2 idade2 profissao2
...
nomeN consorteN CPFN idadeN profissaoN

```

Para que seja feita a leitura e separar o arquivo texto por linhas, basta usar a função de leitura de arquivo criada **leArquivoCaminho**, e, para separar o arquivo, utilizar a função já conhecida: **split-string** tendo como caractere de separação o `"\n"`, ou uma expressão regular com a regra desejada.

Observe que na criação da função foi utilizado após o separador, o operador `t`, eu evita leituras redundantes(no caso, dois ou mais salta linhas seguidos) no separador, evitando strings vazias no resultado.

```

ELISP> (defun separaTextoPorLinha (caminho separador)
        (split-string (leArquivoCaminho caminho) separador t))
separaTextoPorLinha
ELISP> (separaTextoPorLinha "c:\\jornal_taty\\cadastros.txt" "\n")
("nome1 consorte1 CPF1 idade1 profissão1" "nome2 consorte2 CPF2
idade2 profissão2" "... " "nomeN consorteN CPFN idadeN profissãoN")

```

Este foi um exemplo de texto bem comportado, onde em cada linha tinha-se uma frase, um contexto completo.

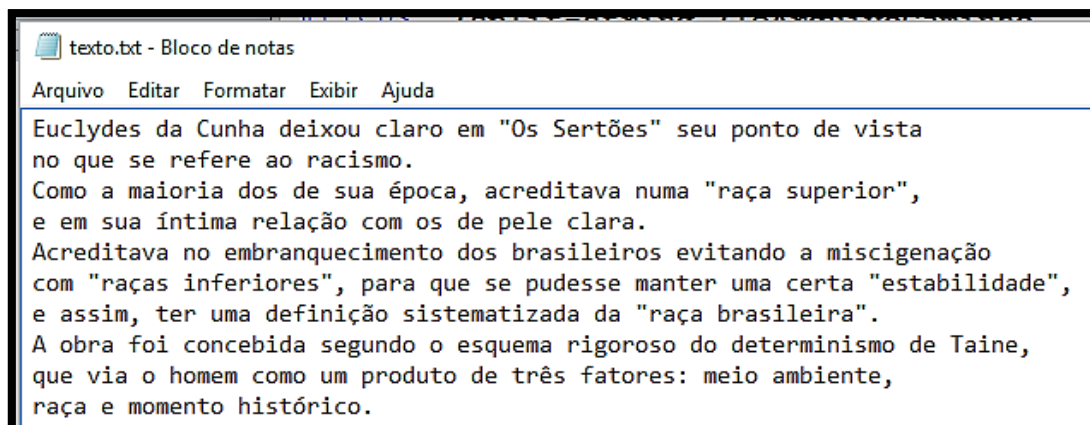
Mas isto nem sempre ocorre em edições de texto. O saltar linha nem sempre pode ser utilizado como separador de frases. Aliás, não é. Os separadores de frases são os caracteres: `.` `!` e `?`.

Assim, deve-se criar uma função que separe as frases do texto para que, posteriormente, se possa analisa-las (fazer um text mining).

Esta função já está criada, já que deixou-se, como argumento, uma variável para assumir o caractere ou expressão regular de separação.

O texto que será analisado, neste novo exemplo, está em: `c:/jornal_taty/texto.txt`, com um trecho de análise de Euclides da Cunha que criou e publicou o primeiro livro reportagem no Brasil.: **OS SERTÕES**. Neste arquivo, como pode ser visto a seguir, tem-se vários saltar linha (`\n`) para que as frases fiquem visíveis no espaço reservado.

Desta forma, separar pelo saltar linha (`\n`) não separaria frases. O caractere `!`, portanto, neste texto, é o adequado.



```

texto.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
Euclides da Cunha deixou claro em "Os Sertões" seu ponto de vista
no que se refere ao racismo.
Como a maioria dos de sua época, acreditava numa "raça superior",
e em sua íntima relação com os de pele clara.
Acreditava no embranquecimento dos brasileiros evitando a miscigenação
com "raças inferiores", para que se pudesse manter uma certa "estabilidade",
e assim, ter uma definição sistematizada da "raça brasileira".
A obra foi concebida segundo o esquema rigoroso do determinismo de Taine,
que via o homem como um produto de três fatores: meio ambiente,
raça e momento histórico.

```

O resultado é o texto agora separado pelo caractere '.', mantendo todos os demais, inclusive o saltar linha (`\n`).

```
ELISP> (separaTextoPorLinha "c:\\jornal_taty\\texto.txt" "\\.")
("Euclides da Cunha deixou claro em \"Os Sertões\" seu ponto de vista \nno
que se refere ao racismo" " \nComo a maioria dos de sua época, acreditava
numa \"raça superior\", \ne em sua íntima relação com os de pele clara" "
\nAcreditava no embranquecimento dos brasileiros evitando a miscigenação
\ncom \"raças inferiores\", para que se pudesse manter uma certa
\n\"estabilidade\", \ne assim, ter uma definição sistematizada da \"raça
brasileira\"" " \nA obra foi concebida segundo o esquema rigoroso do
determinismo de Taine, \nque via o homem como um produto de três fatores:
meio ambiente, \nraça e momento histórico" "\n\n\n\n")
```

7.6.5.3 Limpendo caracteres, palavras indesejadas do texto

No exemplo anterior, o texto foi separado como se desejava, ou seja, em frases, mas trouxe frases poluídas com, por exemplo, caracteres de saltar linha, destacados na figura anterior em amarelo (para quem tem uma cópia colorida, quem não tiver, está destacado em cinza claro).

Para limpar estes caracteres indesejáveis, ou trocar alguma palavra ou caractere do texto, pode-se utilizar a função **replace-regexp-in-string**.

O formato da mesma é:

```
(replace-regexp-in-string padrão_substituto o_que_se_deseja_alterar texto)
```

Exemplo com parte do texto extraído na função anterior

```
ELISP> (replace-regexp-in-string "\n" "" "racismo \nComo a maioria ")
"racismo Como a maioria "
```

Onde "\n" foi substituído pela string vazia, ou seja, foi apagado.

7.6.5.4 Criando uma função para ser lembrada mais facilmente

```
ELISP> (defun limpaTexto (padrao_novo padrao_tirar texto)
      (replace-regexp-in-string padrao_novo padrao_tirar texto))
limpaTexto
ELISP> (limpaTexto "\n" "" "racismo \nComo a maioria ")
"racismo Como a maioria "
```

Outra solução é usar funções de alta ordem, como **mapconcat 'identity**. Esta função quebra um texto por um padrão utilizado pela função **split-string**(já

explicado anteriormente), substitui o padrão de quebra (separador) por outro padrão e junta a string (o texto) novamente.

Como exemplo, será utilizado a quebra pelo padrão ponto(.), depois pelo padrão salta linha (\n) e, como o último padrão de quebra foi o \n, o mesmo será substituído pelo caractere de espaço, para que a última palavra da linha não fique ligada à primeira palavra da linha seguinte.

```
ELISP> (mapconcat 'identity (separaTextoPorLinha (separaTextoPorLinha
"c:\\jornal_taty\\texto.txt" "\\.") "\n") " ")
*** Eval error *** Wrong type argument: stringp, ("Euclides da Cunha deixou claro em "Os
Sertões" seu ponto de vista
no que se refere ao racismo"
Como a maioria dos de sua época, acreditava numa "raça superior",
e em sua íntima relação com os de pele clara"
Acreditava no embranquecimento dos brasileiros evitando a miscigenação
com "raças inferiores", para que se pudesse manter uma certa "estabilidade",
e assim, ter uma definição sistematizada da "raça brasileira"
A obra foi concebida segundo o esquema rigoroso do determinismo de Taine,
que via o homem como um produto de três fatores: meio ambiente,
raça e momento histórico"
"
```

Compare com a figura abaixo e veja que os padrões \n foram retirados. Observe, também, que, com apenas uma chamada de função separou-se as frases e eliminou-se os padrões desejados.

```
ELISP> (separaTextoPorLinha "c:\\jornal_taty\\texto.txt" "\\.")
("Euclides da Cunha deixou claro em "Os Sertões" seu ponto de vista no
que se refere ao racismo"
"
Como a maioria dos de sua época, acreditava
numa "raça superior",
e em sua íntima relação com os de pele clara"
"
Acreditava no embranquecimento dos brasileiros evitando a miscigenação
com "raças inferiores", para que se pudesse manter uma certa
"estabilidade",
e assim, ter uma definição sistematizada da "raça
brasileira"
"
A obra foi concebida segundo o esquema rigoroso do
determinismo de Taine,
que via o homem como um produto de três fatores:
meio ambiente,
raça e momento histórico"
"")
```

7.7 CONCLUSÃO

Este capítulo se encarregou de apresentar as potencialidades do paradigma e ferramentas escolhidas para a implementação dos objetivos e trabalhos desta dissertação, conceituando e formalizando os recursos necessário ao próximo capítulo nas implementações das metas pretendidas, tal como viabilizar se trabalhar com o sistema operacional Windows em aplicações diretas na nuvem, com transporte de dados entre cliente (a máquina) e servidor (a nuvem) mediadas por terminal de comunicação bidirecional **SSH** seguro e efetuar programas típicos de manipulação de texto para text mining.

8 CONCEITOS E ELEMENTOS DE PÁGINAS DE INTERNET

8.1 A PÁGINA DINÂMICA

O objetivo deste trabalho, sob uma visão macro, é viabilizar recursos conceituais e tecnológicos para se implementar páginas estáticas e dinâmicas localmente e na nuvem, onde o público alvo é um jornalista, ou afim, que deseja colocar no ar uma página de notícias bem formatadas, estruturadas mas de uma forma ágil, rápida, podendo concorrer com os internautas e respectivos celulares. Apesar da forma tradicional de um jornal, principalmente impresso, ser estática³⁴, ou seja, após a montagem e publicação do jornal as páginas do mesmo não mudam, sejam na forma impressa quanto eletrônica, a interação com o usuário é algo que tem se tornado uma necessidade para sobrevivência deste meio. A interação com o usuário e entre os próprios elementos das páginas é que dá origem aos jornais, o que gera a necessidade de se implementar páginas dinâmicas.

- **Páginas Estáticas:** As páginas estáticas possuem código HTML com um layout fixo, registrando informações textuais, vídeos, imagens e links pré estabelecidos pelo Web Designer, tal como as páginas da Wikipedia, por exemplo.
- **Páginas Dinâmicas:** Possuem código HTML com uma interface projetada e implementada por um Web Designer (faz a parte gráfica e estrutural do site), contendo bancos de dados, possuindo interação com o usuário gerenciada por um programa de computador, elaborado pelo Programador Web (programa a interface tornando ativos e funcionais os elementos estruturados pelo Web Designer). O conteúdo pode mudar de acordo com as interações. Uma necessidade que surgiu com o e-commerce, e-business, e-learning e as mídias sociais (face book, outros).

8.2 BROWSERS, LINGUAGENS: LIMITES TECNOLÓGICOS

A imaginação não é mais o limite do que, quando e onde a humanidade pode almejar. O limite é o conhecimento e domínio da tecnologia já existente. A tecnologia superou, e muito, a imaginação. Resta a nós imaginar o que fazer com tanta

³⁴ **Páginas estáticas e dinâmicas** - <http://tecnologia.culturamix.com/tecnologias/paginas-estaticas-e-paginas-dinamicas>

tecnologia: um paradoxo? Mas, infelizmente, para muitos profissionais e sistemas/empresas o conhecimento da tecnologia ainda é escasso, bem como o acesso aos recursos para adquiri-la ou implementá-la.

No caso dos jornais e revistas eletrônicas, e o Brasil é um caso típico atualmente (12/2015). Carece-se, mesmo primariamente, do conhecimento sobre ***Feature Detection for Newspaper Publishing***³⁵ e sobre ***Text Mining***³⁶.

Fazer páginas dinâmicas para jornais multi-browsers, multi-devices, com tecnologia mobile ou desktop, é um problema generalizado neste momento de transição tecnológica entre as mídia impressa e mídia na WEB, onde se pensa que se pode fazer de tudo, para qualquer dispositivo, o que, na prática, não tem sido verdade. Pode-se citar, por exemplo, nesta data, 12/2015, que a Folha de São Paulo, a revista Época e Veja não resolveram este problema e que o jornal **O Globo** procurou contorná-lo criando uma assinatura diferente para cada máquina-device (mobile ou desktop), o que não é desejável.

A linguagem HTML sofreu modificações tecnológicas para tentar unificar os recursos de hipermídia, evitando soluções proprietárias e pontuais antes efetivadas por Flash e Java. Desta forma, continua-se esperando que os browsers possam ser padronizados para a leitura e reprodução de tais recursos, fazendo com que grandes empresas, como o Jornal Globo, tenha que editar mais de um jornal para atender vários tipos de equipamentos (leitores - devices) e browsers.

Aconselha-se como leitura complementar a esta dissertação, uma coletânea de contribuições revisadas e registradas sobre este problema, recursos e ***Feature Detection*** sobre HTML5 em <http://diveintohtml5.info/index.html> , e, principalmente, em <http://diveintohtml5.info/detect.html> .

8.3 PRODUÇÃO DE PÁGINAS MULTIMÍDIA E COM HIPERTEXTO EM HTML

É importante que se conheça as edições, montagens padrões de cada um destes item para que todos os browsers, seja para desktops ou tecnologia mobile, independente do sistema operacional, consigam reproduzir tais páginas criadas, sejam estáticas, sejam dinâmicas.

³⁵ **Featuring Detection** - [https://en.wikipedia.org/wiki/Feature_detection_\(web_development\)](https://en.wikipedia.org/wiki/Feature_detection_(web_development)) , <https://modernizr.com/> , https://developer.mozilla.org/en-US/docs/Browser_Feature_Detection , o livro completo: HTML5: Up and Running pode ser adquirido em <http://shop.oreilly.com/product/9780596806033.do>

³⁶ **Text Mining** - https://en.wikipedia.org/wiki/Text_mining

Conhecer as limitações, também, de onde está hospedando seu servidor Web é também de suma importância para que uma página criada possa rodar sem travar, sem omitir qualquer das mídias citadas, por falta de recurso.

Novamente, como é o fundamento deste trabalho, adotar-se-á a **Navalha de Ockhan: a *Lex Parsinoniae***.

Neste capítulo, portanto, serão abordados os conceitos para implementação eficiente, genérica, de uma página, contendo os itens mais comuns em um jornal Web, a saber:

- 1- Títulos, Subtítulos e Textos
- 2- Links ativos
- 3- Figuras, imagens locais ou na Web
- 4- Sons, músicas, áudio em geral, locais ou na Web
- 5- Vídeos, locais ou na Web
- 6- Campos de textos para o leitor postar a opinião deles.
- 7- Text Mining

8.4 COMENÇANDO PELO FIM

É interessante iniciar este estudo entendendo como se dá a interação do usuário com uma página dinâmica já publicada na WEB.

Assim, será iniciada esta abordagem apresentando uma página com um, depois dois, depois três campos de textos, e, finalizando, uma página com mais alguns elementos como figuras, links, sons e vídeos.

Através deste conhecimento, tornar-se-á claro o que se deve fazer ao criar uma página, entendendo como os dados e ações interagem e que não dependem do projeto e implementação visual da interface com o usuário.

Uma página dinâmica de internet, portanto, possui duas estruturas:

1. Interface Visual (IHM- IHC ou HCI³⁷)
2. Programação e componentes dinâmicos

Interface Visual- HCI: O projeto e implementação da parte visual da interface é de competência do Web Designer³⁸. O mesmo distribui os componentes de uma página de forma harmoniosa e funcional.

³⁷ **HCI – Human Computer Interaction** - https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction
<http://interaccaohomemaquina.blogspot.com.br/>

Programação: É realizada pelo programador Web³⁹. A função do mesmo é codificar cada elemento da interface para que se possa interagir com o mesmo, bem como atribuir ações associadas a cada elemento.

Toda as informações, ações interativas realizadas pelos usuários nas páginas são encaminhadas para um campo de texto da página web o qual contém a **URL**⁴⁰ (Uniform Resource Locator- endereço da página), mais um complemento denominado de **Query String**⁴¹. A Query String é a string subsequente à URL da página, onde estão registradas todas as interações do usuário com a página.

URL – Eis um exemplo típico de uma URL:

`http://www.radzero.com.br/index.html`

Onde se tem inicialmente o tipo de protocolo de comunicação utilizando: (`http`), o hostname⁴² (`www.radzero.com.br`) e o file name (nome do arquivo) (`index.html`).

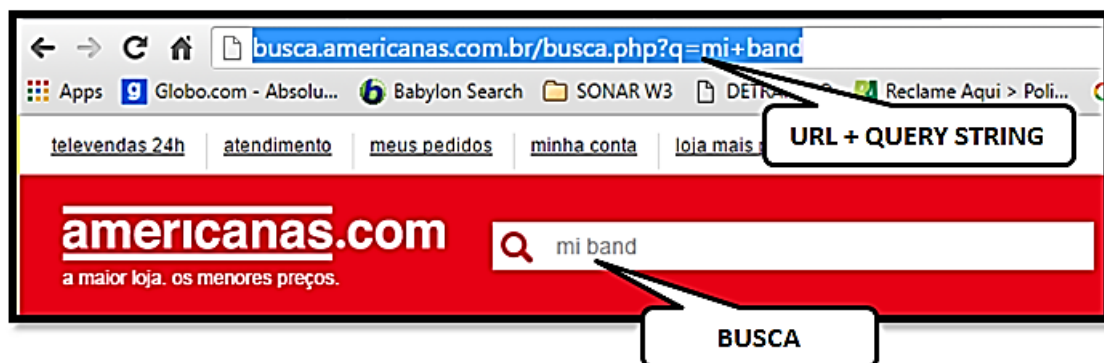
Outro exemplo poderia ser o da `americanas.com.br`

`http://www.americanas.com.br`

A qual apresenta apenas o protocolo (`http`), e o domínio `www.americanas.com.br`.

Quando ocorre uma interação em uma página, é acrescentada, a esta **URL**, mais textos que comporão a **Query String**.

Observe o exemplo de uma busca na **americanas.com.br** :



³⁸Web Designer - https://en.wikipedia.org/wiki/Web_design
<http://cursosonline.uol.com.br/assinatura/artigos/formacao-profissional/o-que-faz-um-web-designer/>

³⁹Programador Web - https://en.wikipedia.org/wiki/Web_development
<http://metablog.xcake.com.br/o-que-faz-um-programador-web>

⁴⁰URL - https://en.wikipedia.org/wiki/Uniform_Resource_Locator

⁴¹Query String - https://en.wikipedia.org/wiki/Query_string

⁴²hostname – domínio da hospedagem

Obtêm-se, na linha de edição do browser, a **URL + Query String** resultante da busca de uma pesquisa, a saber:

<http://busca.americanas.com.br/busca.php?q=mi+band>

Na maioria das vezes, o **http://** é omitido da linha de edição dos Browsers, mas, se capturar a linha, o mesmo, quando colado em um editor de texto, aparece.

Neste caso, temos:

- **URL** = <http://busca.americanas.com.br/>
- **Query String** = [busca.php?q=mi+band](#)

Observe que, para o Sistema de análise interativa da página dinâmica, implementado pelo programador web, o que interessa é a Query String ([busca.php?q=mi+band](#)), a qual contém a informação da interação, ou seja: uma busca ([busca.php?](#)) foi efetuada no código php (uma linguagem de programação de internet) e a pergunta ([q=mi+band](#) - q de query) foi sobre **mi band** (uma pulseira inteligente).

Assim, todas as interações, como será visto logo a seguir, estão registradas nesta Query String, seja criptografada ou não⁴³. Tudo passa pela Query String e o banco de dados do sistema implementado. É através da informação contida na Query String que se determina o tipo de busca e o que se deseja retornar em pesquisas ou registros nos bancos de dados, ou, até mesmo, em outros sites, como é o foco de vários sites concentradores de **feeds**⁴⁴ de temas específicos, como, por exemplo: sobre restaurantes, sobre vendas de carros e até sites famosos de produtos generalizados como buscapé e outros.

⁴³ Como curiosidade, há alguns anos atrás, quando este tipo de informação não era muito conhecido pelos usuários de internet, alguns hackers capturavam esta linha do browser contendo a Query string e mudavam a mesma para conseguir descontos maiores em produtos, entrar com cupons, etc. Era só mudar a Query String e dar um refresh, atualização, na página. Mesmo grandes empresas como [americanas.com](#) e [submarino](#) foram grandemente lesadas até descobrir esta falha na programação.

⁴⁴ **Feed**, também é conhecido como "**RSS Feeds**". **Rich Site Summary**, ou seja, uma forma simplificada de apresentar o conteúdo de um site. Na prática, **Feeds** são usados para que um usuário de internet possa acompanhar os novos artigos e demais conteúdo de um site ou blog sem que precise visitar o site em si. Sempre que um novo conteúdo for publicado em determinado site, o "assinante" do feed poderá ler. <https://pt.wikipedia.org/wiki/Feed> , <http://blog.planalto.gov.br/o-que-sao-feeds/>

8.4.1 Separando A Query String Da Url

No capítulo 5 será mostrado como pegar os dados de consulta em uma Query String e fazer uma busca em arquivo texto sobre a consulta, utilizando os conceitos vistos neste capítulo.

8.5 ELEMENTOS BÁSICOS DE UMA PÁGINA COM CODIFICAÇÃO HTML, PASSO A PASSO

Como foi supra citado, 7(sete) serão os itens, elementos que serão inseridos em uma página jornalística multimídia com hipertexto básica. Também foi citado que serão utilizados recursos que possam ser executados corretamente pelos browsers e leitores digitais. A solução para isto é utilizar código HTML⁴⁵ puro que, quando bem formado⁴⁶, é aceito por todos os browsers e dispositivos.

O motivo de se apresentar neste capítulo alguns elementos citados para uma página HTML é que o editor EMACs, ORG MODE, que vai ser utilizado para gerar as páginas multiplataformas, multidevices, permite que seja inserido codificação HTML no código das mesmas, caso desejado for.

Nestes caso, como vai ser apresentado no capítulo 5, basta colocar o código HTML entre o seguinte cabeçalho e terminador:

#+BEGIN_HTML

Codificação HTML

#+END_HTML

⁴⁵ **Curso HTML puro**– Hyper Text Markup Language(Linguagem de Formatação de Hipertexto) http://pt.slideshare.net/richard_romancini/curso-html-9817393 , <https://www.youtube.com/watch?v=YnWuJV23GU&list=PLOyO77tk1fqUB3pAnXDFrG85tQnzFLYVv>

⁴⁶ **Validador – W3C Markup Validation Service** - <https://validator.w3.org/> , também em : <http://w3c.github.io/developers/tools/> e <https://validator.w3.org/nu/>

8.6 CODIFICAÇÃO HTML BÁSICA NA MONTAGEM DOS 8 ITENS, ELEMENTOS UTILIZADOS NO JORNAL

Antes de inserir elementos: antes de iniciar a montar a páginas com os elementos do jornal, toda página, por mais simples, deve ter um padrão para iniciar e terminar. No caso, deve-se criar a página com um título, mesmo que a ela fique em branco, sem elementos.

Uma página é criada pelo abrir e fechar de uma tag dizendo a mesma ser HTML e com um título que aparecerá no arcabouço da página. Neste momento, não se pode confundir o título da página com o título das matérias que se deseja colocar no jornal.

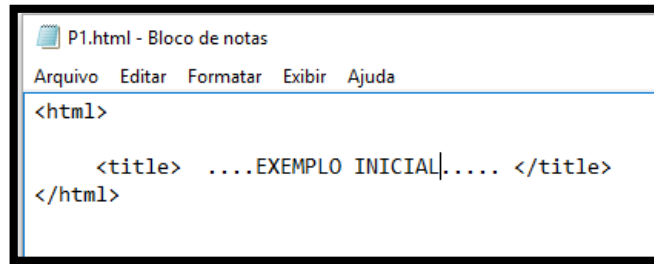
Assim, uma página em branco, sem elementos simples, pode ser criada como segue, onde a **tag** para criar a página é `<html> ...</html>`, e, para criar o título do arcabouço é `<title> título </title>`, ou seja:

```
<html>
    <title> .... EXEMPLO INICIAL ..... </title>
</html>
```

Os comandos em **HTML**, são delimitados por um abrir e fechar de tags⁴⁷.

Páginas **HTML** podem ser editadas em editores de texto comum, como, por exemplo, o bloco de notas. Eis um exemplo de criação de uma página em branco, sem elementos, contendo apenas o título da mesma no arcabouço:

⁴⁷ **TAG** – É um rótulo que indica o que o browser deve fazer com a informação que está delimitada pelo abrir e fechar da mesma. Uma tag em HTML inicia pelo abri e fechar dos símbolos `<tag>` e finaliza pela mesma estrutura acrescentando o símbolo de fechamento `/`, ou seja: `</tag>`. <http://pt-br.html.net/tutorials/html/lesson3.php>



```

P1.html - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<html>
    <title> ....EXEMPLO INICIAL|. .... </title>
</html>

```

Rodando esta página no browser, tem-se a página em branco com o Título:



8.7 ELEMENTOS DAS PÁGINAS DO JORNAL

A seguir serão apresentadas as formatações em HTML puro para produção dos elementos desejados inicialmente no jornal, utilizando em todos o princípio da *lex parsinoniae*.

- 1- Títulos, Subtítulos e Textos
- 2- Links ativos
- 3- Figuras, imagens
- 4- Sons, músicas, áudio em geral
- 5- Vídeos
- 6- Text mining.

8.7.1 Título, Subtítulo E Texto:

Um Título, Subtítulo ou texto é formatado da mesma forma em HTML, variando em tamanho, disposição e cor.

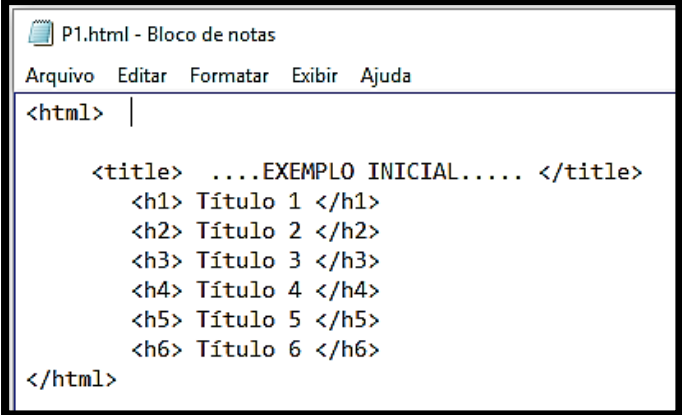
São seis os tamanhos básicos para texto em HTML, começando pelo tamanho 1 (o maior) até o tamanho 6, (o menor).

As tags numeradas de 1 a 6, indicam o tamanho do texto. Pode-se colocar outras tags dentro de uma tag, como, no caso, align (alinhamento do texto), style (estilo da letra: cor, tipagem). A cor, tipo de letra são argumentos destas tags.

A tag é formatada como segue:

<hn> ... </hn>, onde **n** é o tamanho da letra.

Exemplo:



```

P1.html - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<html> |
    <title> ...EXEMPLO INICIAL..... </title>
    <h1> Título 1 </h1>
    <h2> Título 2 </h2>
    <h3> Título 3 </h3>
    <h4> Título 4 </h4>
    <h5> Título 5 </h5>
    <h6> Título 6 </h6>
</html>
  
```

A página gerada fica:



As tags para texto, que serão utilizadas inicialmente são:

COR: **<hn style="color: cor " > texto </hn>**

FONTE: **<hn style= font-family:fonte1, fonte2> texto"**

EXEMPLO:

```

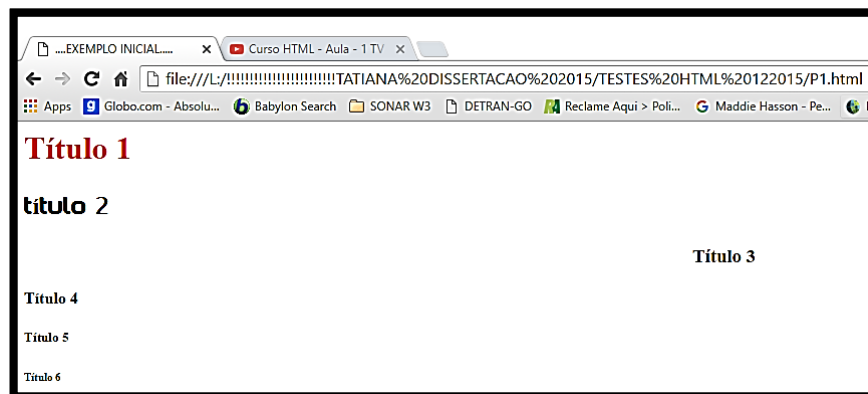
P1.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>

  <title> ....EXEMPLO INICIAL..... </title>
  <h1 style= "color:red"> Título 1 </h1>
  <h2 style= "font-family:paola"> Título 2 </h2>
  <h3 align= "center"> Título 3 </h3>
  <h4> Título 4 </h4>
  <h5> Título 5 </h5>
  <h6> Título 6 </h6>

</html>

```

A página gerada fica:



Pode-se agrupar todos os recursos em um item de texto apenas.

```

html3.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>

  <title> ....EXEMPLO INICIAL..... </title>
  <h1 align= "center" style= "color:red; font-family:paola"> Título 1 </h1>
  <h2> Título 2 </h2>
  <h3> Título 3 </h3>
  <h4> Título 4 </h4>
  <h5> Título 5 </h5>
  <h6> Título 6 </h6>

</html>

```

Exemplo:



São estes os recursos básicos para montagem de Títulos, Subtítulos e Textos.

8.7.2 Links

No caso do jornal, objetiva-se a inserção de links em duas estruturas distintas.

- a. Individualizado: abaixo de um elemento da página.
- b. Inserido em um texto: tornando parte do texto com um link ativo.

8.7.2.1 Individualizado:

A tag para colocar um link na página é:

` texto que servirá como link `

Exemplo

```
P1.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>
  <title> ...EXEMPLO INICIAL.... </title>
  <h1 align = "left" style= "color:red; font-family:paola"> COLOCANDO UM LINK ATIVO! </h1>
  <h2>Clique no link a seguir:</h2>
  <a href="http://www.mapasdeconhecimento.com.br/">Saiba tudo sobre os MCEs</a>
</HTML>
```

8.7.2.2 Inserido um link no meio de um texto:

Utiliza-se a mesma tag, mas, como não se quer saltar linha no texto, ou seja, o link faz parte do texto, a tag com o link deverá estar na mesma tag do texto.

Exemplo:

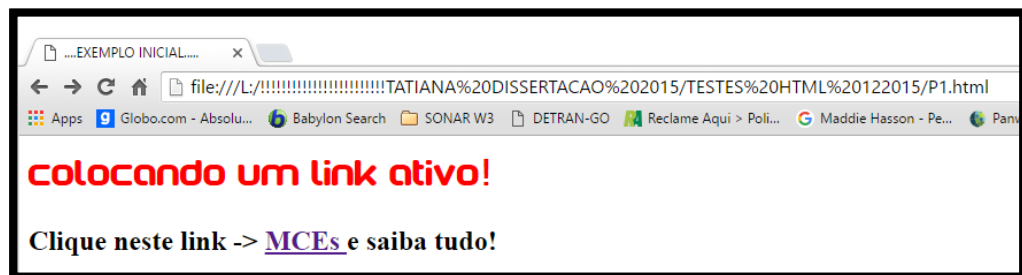


```
P1.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>

<title> ...EXEMPLO INICIAL... </title>
<h1 align = "left" style= "color:red; font-family:paola"> COLOCANDO UM LINK ATIVO! </h1>
<h2>Clique neste link -> <a href="http://www.mapasdeconhecimento.com.br/">MCEs </a> e saiba tudo! </h2>

</HTML>
```

A página gerada fica:



8.7.3 Figuras E Imagens

Podem ser de 3 (três) tipos

- imagem estática local
- imagem vinda de um link
- imagem como link (como se fosse um botão)

8.7.3.1 Imagem estática local

A figura tem que estar no mesmo diretório da página html

Exemplo:

```

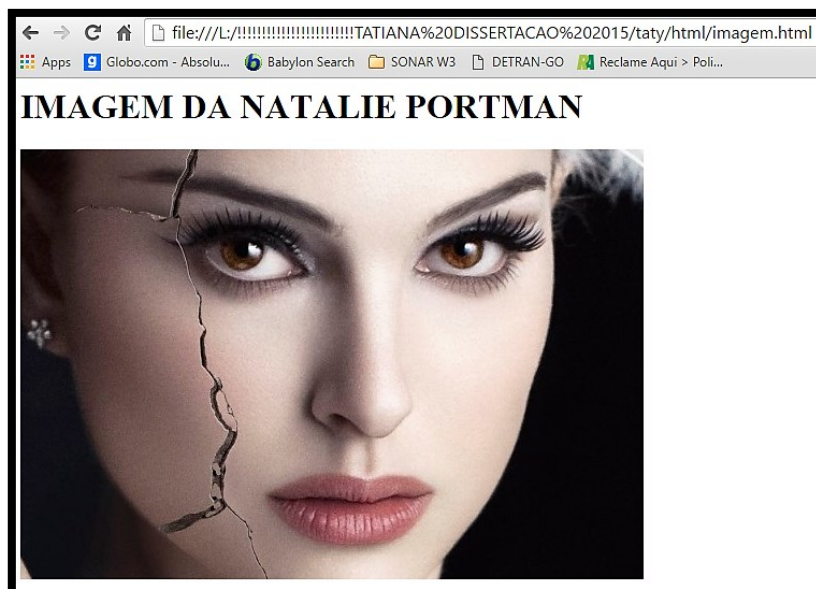
imagem.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>

    <title> ....FIGURAS E IMAGENS LOCAIS..... </title>
    <h1> IMAGEM DA NATALIE PORTMAN</h1>
    

</html>

```

A página gerada fica:



8.7.3.2 Imagem vinda de um link

Exemplo:

```

IMAGEMIncorporação.TXT - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>

    <title> IMAGEM VIA LINK</title>
    <h1> DE LONGE, IMPERA A PURA BELEZA! </h1>
    

</html>

```

A página gerada fica assim:



8.7.3.3 Imagem com link ativo

Exemplo:

```

imagemBotao.html - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<html>

    <title> ....FIGURAS E IMAGENS LOCAIS..... </title>
    <h1> IMAGEM DA NATALIE PORTMAN COM LINK ATIVO</h1>
    <a href="http://www.mapasdeconhecimento.com.br">
        <img src= natali.jpg border="none" /></a>

</html>

```

A página gerada fica:



Ao clicar na imagem, vai para a página com o endereço <http://www.mapasdeconhecimento.com.br>

8.7.4 Sons, Músicas, Áudio Em Geral

Com o objetivo de colocar um áudio padrão, escolha o formato MP3 (.mp3)

A codificação HTML para um player de áudio é a seguinte:

```

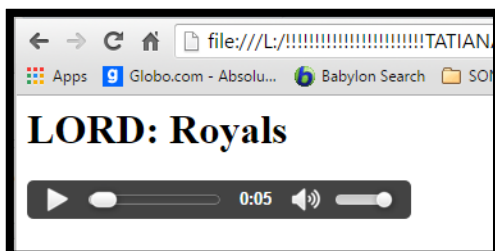
audioL.html - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
<html>

    <title> ...ÁUDIO..... </title>
    <h1> LORD: Royals</h1>
    <audio src="Royals.mp3" controls></audio>

</html>

```

A página gera fica:



8.7.5 Vídeo

Você pode colocar um vídeo local, na mesma pasta que colocar a página html, ou colocar um link de vídeo da internet.

8.7.5.1 Colocando um vídeo local

A codificação fica assim:

```

video.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<html>
    <title> VÍDEO</title>
    <h1> CELULARES E CÂNCER</h1>
    <video src="radiacao.mp4" controls></video>
</html>

```

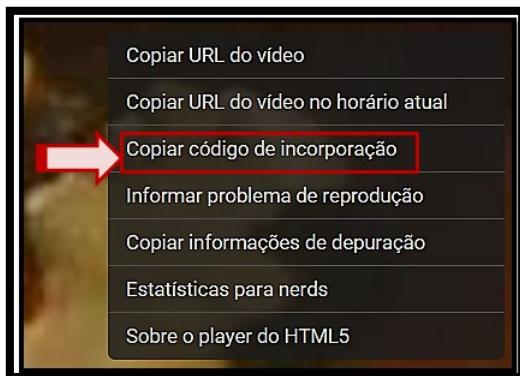
A página fica assim:



8.7.5.2 Colocando um link (código de incorporação) de um vídeo da internet

Quando se deparar com um vídeo na internet que deseje colocar na página de notícias, e não apenas um link para abrir a página com o vídeo, você pode colocar o código de incorporação do vídeo direto na página html.

Faça o seguinte. Coloque o cursor do mouse em cima do vídeo que deseja incorporar, e clique com o botão direito do mouse. Um menu de opções vai abrir. Clique em Copiar código de incorporação.



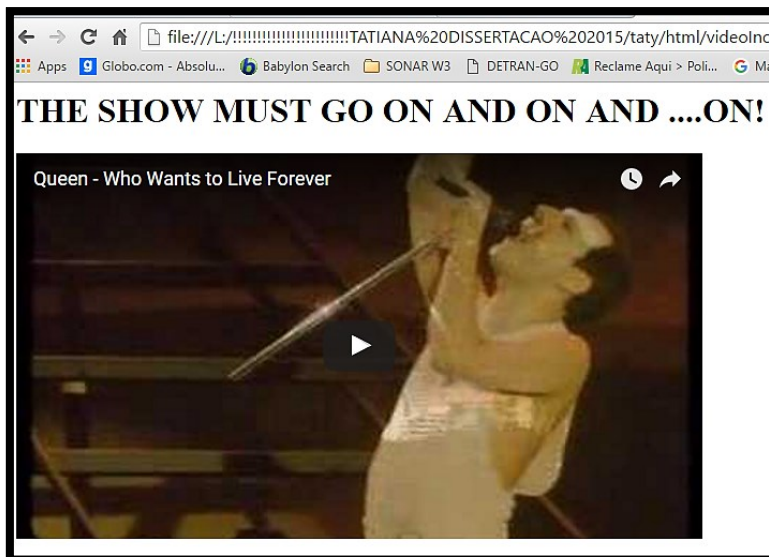
Exemplo:

```

videoincorporação.html - Bloco de not. s
Arquivo Editar Formatar Exibir Ajuda
<html>
  <title> VÍDEO INCORPORAÇÃO - SHOW MUST GO ON!</title>
  <h1> THE SHOW MUST GC ON AND ON AND ...ON!</h1>
  <iframe width="640" height="360" src="http://www.youtube.com/embed/pC4Z0xpu2rs"
    frameborder="0" allowfullscreen></iframe>
</html>

```

A página fica assim:



8.7.6 TEXT MINING

Como o mesmo exige programação em linguagem de alto nível, este item será apresentado no próximo capítulo.

8.8 ESTRUTURA FINAL DE UM ARQUIVO BEM FORMATADO EM HTML

Para organizar uma página em HTML, é bom ter uma estrutura.

Apesar de não precisar, se todos usassem, facilitaria fazer text mining nas mesmas. Assim, separa-se o arquivo HTML entre cabeçalho (HEAD) e corpo (BODY).

A estrutura de uma página HTML com o princípio da *lex parsimoniae* fica assim:

```
<HTML>

  <HEAD>
    <TITLE> TÍTULO DA PÁGINA </TITLE>
  </HEAD>

  <BODY>
    CONTEÚDO DA PÁGINA
  </BODY>

</HTML>
```

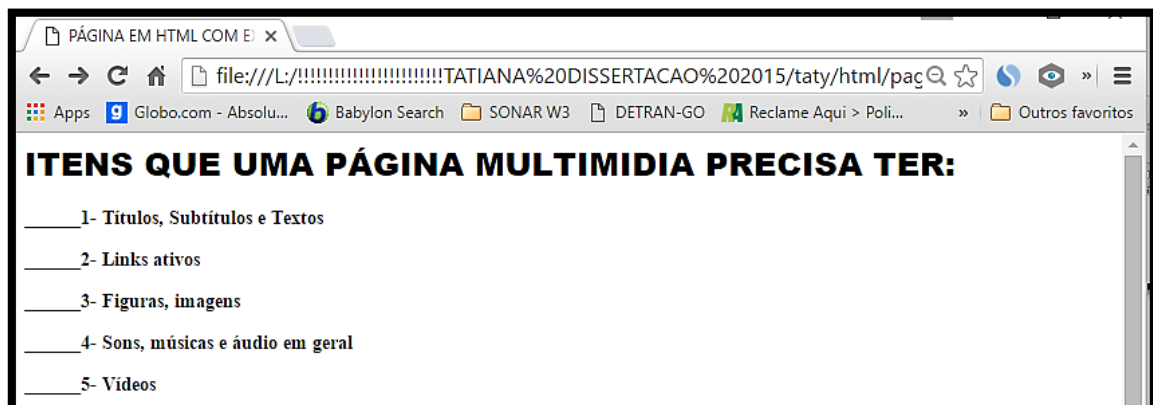
Montando uma página com todos os 5 recursos apresentados, tem-se:

```

pagina.html - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<HEAD>
    <title> PÁGINA EM HTML COM EXEMPLO DE MÍDIAS E LINKS </title>
</HEAD>
<BODY>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        ITENS QUE UMA PÁGINA MULTIMÍDIA PRECISA TER:</h1>
    <h3>_____1- Títulos, Subtítulos e Textos</h3></p>
    <h3>_____2- Links ativos</h3>
    <h3>_____3- Figuras, imagens </h3>
    <h3>_____4- Sons, músicas e áudio em geral </h3>
    <h3>_____5- Vídeos </h3>
</br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        UM MÉTODO QUE OS EDUCADORES PRECISAM CONHECER!</h1>
    <h2>Clique neste link -> <a href="http://www.mapasdeconhecimento.com.br/"> MCE </a> e saiba tudo!</h2>
</br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        BELEZA IMORTALIZADA PELA ARTE</h1>
        
    </br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        DE LONGE: PURA BELEZA!</h1>
    
    </br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        ROMANTISMO COM UM RETOQUE DE ARISTOCRACIA</h1>
        <a href="http://www.kboing.com.br/claude-debussy/1-1088551/" target="parent">
        </a>
    </br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        OS RITMOS MUSICAIS ESTÃO SEMPRE SURPREENDENDO</h1>
        <audio src="Royals.mp3" controls></audio>
    </br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        UM BOM CLIP NUNCA DEVE SER ESQUECIDO!</h1>
        <iframe width="640" height="360" src="http://www.youtube.com/embed/pC4Z0xpu2rs"
        frameborder="0" allowfullscreen></iframe>
    </br>
    <h1 align="left" style="color:black; font-family:ARIAL BLACK">
        NEM TUDO É BELEZA: CELULARES: CAI A CENSURA!</h1>
        <video src="radiacao.mp4" controls></video>
    <br>
    <h4>_____</h4>
    <h4>AUTORA: Tatiana Oliveira Lima</h4>
    <h4> hominisPress</h4>
</BODY>
</HTML>

```

A página gerada fica:



CONTINUA

UM MÉTODO QUE OS EDUCADORES PRECISAM CONHECER!

Clique neste link -> [MCE](#) e saiba tudo!

BELEZA IMORTALIZADA PELA ARTE

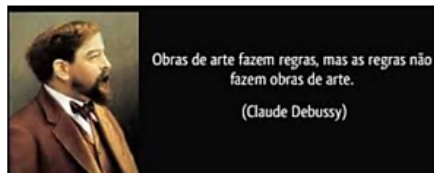


DE LONGE: PURA BELEZA!



CONTINUA

ROMANTISMO COM UM RETOQUE DE ARISTOCRACIA



OS RITMOS MÚSICAIS ESTÃO SEMPRE SURPREENDENDO



CONTINUA

UM BOM CLIP NUNCA DEVE SER ESQUECIDO!



Queen - Who Wants to Live Forever

NEM TUDO É BELEZA: CELULARES: CAI A CENSURA!



AUTORA: Tatiana Oliveira Lima

hominisPress

8.9 CONCLUSÃO:

Implementar uma página em HTML é algo simples, mas não é fácil. Demanda muita atenção. Qualquer espaço, aspas, enfim, qualquer detalhe que for esquecido ou acrescentado compromete a página toda.

O arquivo ficou até pequeno, 48 linhas, quando comparado ao mostrado no início deste capítulo. O motivo é que na programação do mesmo não se preocupou com **Feature Detection**, ou seja, não se preocupou em colocar na programação se a cada recurso inserido o Browser onde for ler consegue utilizar, apresentar corretamente, o que pode impedir da página ir ao ar em um dispositivo ou browser sem o recurso.

No próximo capítulo será visto como montar páginas com mais simplicidade, recursos ativos, e com menos código, facilitando a leitura e a correção de erros, caso houverem. Também será utilizado um editor textual, e **EMACS**, como aqui foi utilizado o **Bloco de Notas(Note Pad)** do Windows.

9 PROGRAMANDO, CRIANDO AS PÁGINAS DINÂMICAS EM DESKTOP E DIRETO NA NUVEM UTILIZANDO O EDITOR EMACS-LISP (NA NUVEM) E EMACS NO MODO ORG⁴⁸ (ORG-MODE NO DESKTOP)

No Capítulo 7 e 8 foi visto como implementar os elementos básicos de uma página de Jornal utilizando programação com **DrRacket**, seguindo a estrutura básica especificada nesta dissertação. O mesmo foi dedicado a quem deseja implementar um servidor para construir um jornal Web, com um banco de dados robusto, para grande quantidade de acessos simultâneos, o que a linguagem escolhida suporta.

A linguagem escolhida possui um rigoroso formalismo matemático de forma a não se tornar obsoleta, como ocorre com várias linguagens datadas, que mudam de acordo com que as respectivas bibliotecas e dlls mudam com o tempo, como foram os casos do Flash da Adobe que foi substituído pela disputa atual entre **V8**⁴⁹ da Google e o **spiderMonkey**⁵⁰ da Mozilla, e da **jvm**⁵¹ pela entrada em vigência da **llvm**⁵² e **V8**, a partir do fim de 2014.

Este capítulo é dedicado mais especificamente ao Jornalista, ao estudante de jornalismo que precisa correr contra o tempo para publicar com primazia uma notícia onde a concorrência com os bilhões de internautas, com seus mobiles com câmeras fotográficas e sites de relacionamento, publicam, informalmente, sem compromisso com a verdade, legislação e linguagem elaborada, as notícias em primeira mão, levando até mesmo bons jornais a sofrerem com a

⁴⁸ **ORG-MODE** - <https://en.wikipedia.org/wiki/Org-mode> , <http://www.emacswiki.org/emacs/OrgMode>

⁴⁹ **V8 – Máquina virtual JavaScript** da Google (não confundir com JVM – Máquina Virtual Java). [https://en.wikipedia.org/wiki/V8_\(JavaScript_engine\)](https://en.wikipedia.org/wiki/V8_(JavaScript_engine))

⁵⁰ **Spider monkey** - [https://en.wikipedia.org/wiki/SpiderMonkey_\(software\)](https://en.wikipedia.org/wiki/SpiderMonkey_(software)) - **Máquina Virtual JavaScript** da Mozilla Firefox e desktop GNOME 3, concorrente da **V8** .

⁵¹ **Jvm – Java Virtual Machine** - https://en.wikipedia.org/wiki/Java_virtual_machine - é um programa que carrega e executa os aplicativos Java, convertendo os bytecodes em código executável de máquina. Os programas escritos em Java podem funcionar em qualquer plataforma de hardware e software que possua uma versão da JVM, que, infelizmente não se tornou padrão para todos os dispositivos. <http://www.handersonfrota.com.br/introducao-a-jvm-java-virtual-machine/>

⁵² **llvm – Low Level Virtual Machine** – antes, apenas privilégio da plataforma Apple, dos Macs e iPx, é hoje disponível para qualquer plataforma e dispositivos móveis. O LLVM pode prover camadas intermediárias de um compilador, lendo a [representação intermediária](#) de um compilador e retornando outra representação otimizada, que pode ser então convertida e ligada em [código de montagem](#) para determinada plataforma. **llvm** realmente é multiplataforma, multidevices, multiparadigma e é aceita por todos os browsers. https://pt.wikipedia.org/wiki/Low_Level_Virtual_Machine, <https://en.wikipedia.org/wiki/LLVM>

crise de investidores pelo desinteresse em propaganda devido postarem notícias de segunda mão que ninguém interessa em ler.

Um jornalista, que não é especialista em computação, internet, servidores, etc., necessita conhecer passo a passo todas as informações para poder criar pátinas dinâmicas, seja direto na nuvem ou por upload de uma máquina pessoal (já que nem sempre o mesmo possui a senha do servidor do jornal). Necessita saber sobre **hostname**, **login**, **password** da **hospedagem/cPanel**, porta **SSH**, ... , seja para acesso através do **Linux**, **Mac**, **Windows**, **Apple (iPx)** e **Android**. Assim, fecha-se um domínio. Um jornalista, para ser rápido, ter a primazia da notícia, tem que usar o que tiver em mãos ou o que conseguir emprestado.

De posse de tais conhecimentos, o jornalista, o editor poderão competir com os concorrentes, com qualquer dispositivo que tenha em mãos, seja um notebook, um desktop, um celular ou um tablet, independentemente do dispositivo e sistema operacional.

Existe uma mistificação que programar, editar direto na nuvem, até mesmo em desktop, é uma tarefa para poucos, somente para pessoas muito capacitadas. Esta dissertação, este capítulo, busca desmistificar isto com provas de conceitos de que, quando se sabe, e não se precisa muito, não é difícil de fazer. Por outro lado, não se pode confundir algo fácil com algo simples. Como citou Einstein:

"Everything should be made as simple as possible, but not simpler"

As ações exigem um rigor de atenção a detalhes, onde a falta de um símbolo, de um espaço a mais ou a menos fazem a diferença em acontecer e nada acontecer.

Assim, neste capítulo será mostrado passo a passo, utilizando a metodologia dos **MCE**⁵³[**REFERÊNCIA**: LIMA, L. V.; FERREIRA, D. C. M.; CAMARGO JÚNIOR, H. SCHIOVATO, N. S. C. [Mapas de Conhecimento Estruturado: proposta de uma nova abordagem metodológica de ensino e aprendizagem](#). Publicado pela revista Educere et Educare, Vol. 9 Ed. Especial. (2014)], todos os conceitos e conhecimentos necessários ao entendimento de como efetivar as objetivadas páginas dinâmicas, tornando fácil, não simples, tais implementações. Desta forma, o que pode parecer, a princípio, poder ser relegado a

⁵³ **MCE – Mapas de Conhecimento Estruturado** - www.mapasdeconhecimento.com.br -

um anexo nesta dissertação, não pode ser excluído do texto principal para não prejudicar a compreensão e a capacitação em se repetir os conceitos registrados.

Uma significativa contribuição deste trabalho é capacitar ao leitor, jornalista ou outro profissional, a conseguir repetir com segurança os conceitos aqui registrados, o que dificultaria se fossem os detalhes colocados em um ou mais anexos.

9.1 FAZENDO UMA PÁGINA DINÂMICA USANDO EMACS-ORG-MODE NO DESKTOP

Para se trabalhar direto no computador e criar uma página de notícias para fazer o upload para a hospedagem, primeiro tem que baixar a versão adequada ao seu computador, à plataforma, sistema operacional utilizado pelo mesmo.

Linux e Mac geralmente já vêm com uma versão do **Emacs**⁵⁴. Como esta ferramenta foi implementada pensando nos profissionais da computação, que trabalham principalmente em Linux e algumas máquinas, como os Macs que possuem um Linux limitado, trabalhar com a mesma em Windows não foi preocupação dos implementadores. Mesmo assim, foi feita uma implementação para o pessoal de Windows, mas a documentação de como contornar os problemas do Windows em aplicações profissionais no emacs, inexistem mesmo na internet, ou, quando existem, são incompletas não funcionam. Isto se dá principalmente porque o usuário do Windows está acostumado a deixar o Windows fazer tudo para ele, instalar os programas onde achar melhor, etc.

Assim, como este capítulo trata de fazer aplicações profissionais em um Windows que não foi feito para isto (trabalhar com nuvem, servidores, SSH, ...), todos os passos aqui registrados devem ser seguidos à risca para que se tenha o controle completo da ferramenta (o emacs) e para que tudo funcione conforme desejado. O que parece ser um alerta a uma dificuldade complexa, na realidade é o dia a dia, uma rotina pra os programadores e usuários Linux.

⁵⁴ **Emacs** – Mais que um editor, mas uma plataforma de programação e desenvolvimento multiparadigma. <https://en.wikipedia.org/wiki/Emacs> , <http://www.emacswiki.org/>

9.1.1 Criando Um Diretório Para A Página

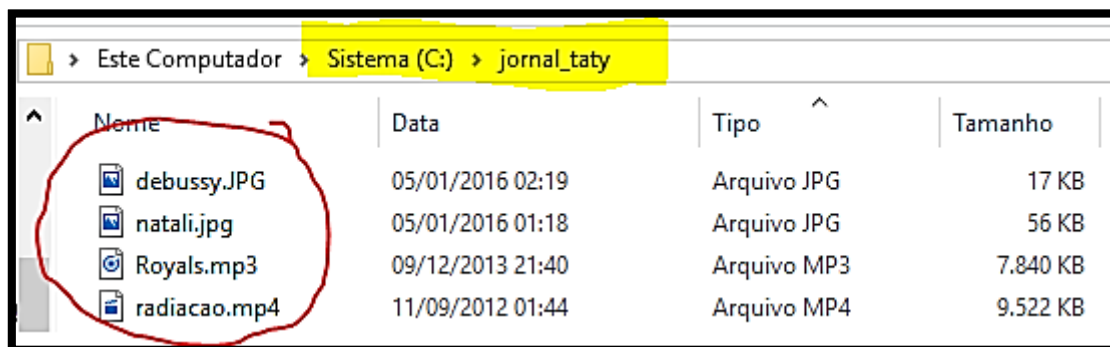
É importante organizar o trabalho de construção de uma página, criando um diretório específico para ela, de tal forma que não se esqueça de algum arquivo que for utilizado pela mesma na hora de subir a página para a hospedagem.

Como sugestão para esta dissertação, uma pasta denominada **jornal_taty** deverá ser criada no disco **C** (poderia ser em qualquer outro lugar). No **ANEXO EMACS_CONFIGURANDO** é explicado como criar uma pasta no **Windows 7** e no **Windows 10**, para que se guarde os arquivos e a página **html** gerada.

Na mesma serão colocados, inicialmente, 4 (quatro) arquivos de mídia local:

- 1- **debussy.jpg** - imagem
- 2- **natali.jpg** - imagem
- 3- **Royals.mp3** - áudio mp3
- 4- **radiacao.mp4** - vídeo mp4

Os mesmos serão utilizados na montagem da página.



Neste diretório serão armazenados outros arquivos necessários ao desenvolvimento e à página **html final**.

9.1.2 Instalando Emacs No Windows

Na nuvem não se precisa instalar o Emacs, as hospedagens já possuem uma versão, normalmente o **emacs-23.1**⁵⁵, já, no desktop, principalmente em plataforma Windows, você terá que baixar a versão adequada à sua plataforma, seu sistema operacional. O **ANEXO EMACS_CONFIGURANDO** mostra todos os

⁵⁵ **emacs-23.1** - <http://lists.gnu.org/archive/html/info-gnu-emacs/2009-07/msg00000.html>

⁵⁵ <http://www.emacswiki.org/emacs/EmacsLisp> , <http://emacs-doctor.com/learn-emacs-lisp-in-15-minutes.html>

detalhes, passo a passo de como baixar o Emacs em um local de fácil acesso e conhecido e como configurá-lo corretamente e criar um arquivo de inicialização para que novos recursos possam ser associados ao mesmo, já que, como dito, o Emacs é um ambiente de programação multiparadigma orientado a macros⁵⁶, completo, e não apenas um bom editor para programação.

Apesar do anexo citado apresentar com detalhes como configurar o emacs, pode-se trabalhar com o mesmo da forma com que é baixado. A única limitação se dá quando se desejar personaliza-lo e utilizar recursos não padrões diferentes dos que são disponibilizados para o programador, o usuário comum (padrão).

Assim, recomenda-se seguir os passos do anexo supra citado, mas, para mostrar que não existe problema para usar o **emacs** na forma que vem distribuído, segue uma instalação padrão Windows, onde você não tem controle da mesma, o Windows decide onde e o que instalar. Para um profissional em Linux, isto é um caos, mas, para os usuários do Windows, denominam de conforto.

9.1.3 Instalando O Emacs Automaticamente Pelo Windows

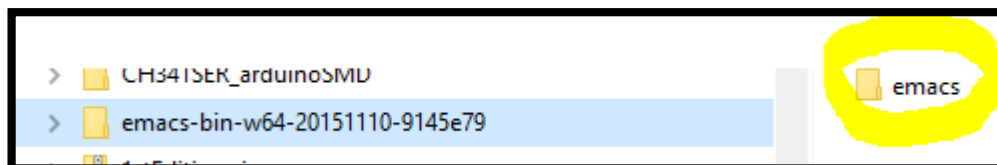
Tomando como exemplo uma plataforma Windows 64, tem-se uma boa distribuição em:

http://sourceforge.net/projects/emacsbinw64/?source=typ_redirect

1- Ao baixar, descompacte o arquivo

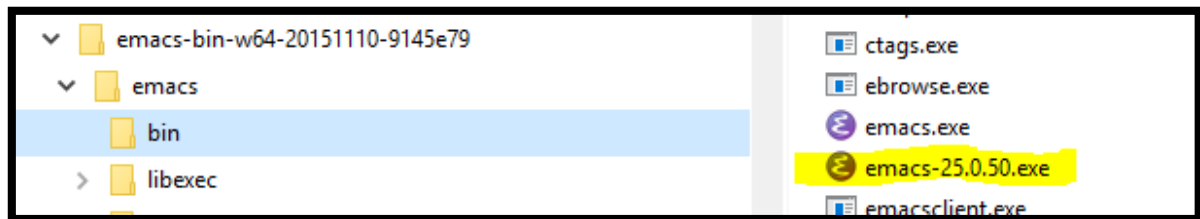


2- Entre no diretório do **emacs**

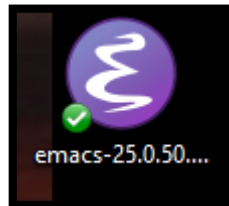


3- Execute o arquivo, no caso, **emacs-25.0.50.exe**

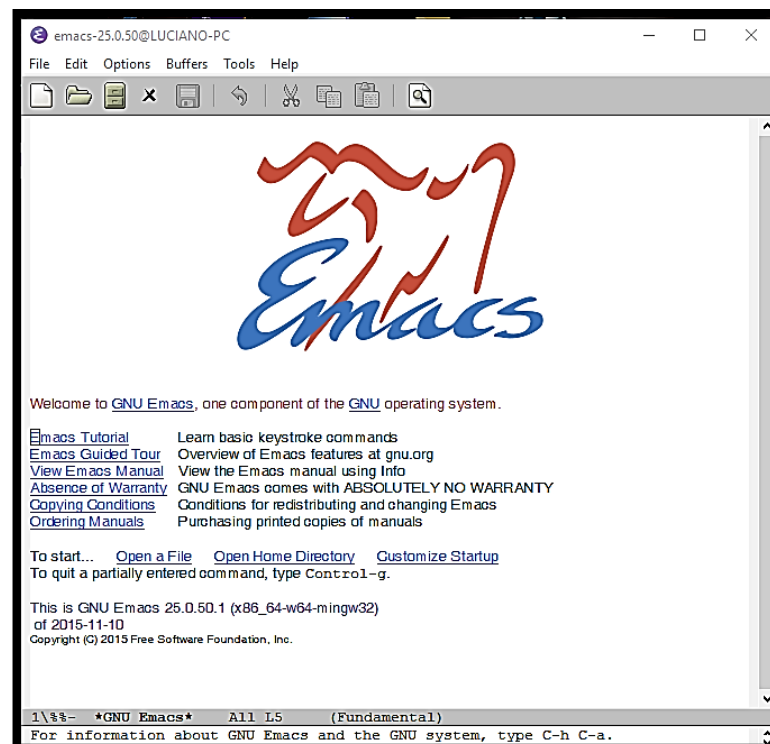
⁵⁶ **Orientada a macros:** você configura a sintaxe e cria qualquer estrutura multiparadigma que deseje, inclusive define linguagens específicas em contextos desejados.



- 4- Crie um atalho para ele na área de trabalho (desktop), conforme explicado no **ANEXO EMACS_CONFIGURANDO**



- 5- Clique no mesmo (ou direto no local de instalação – item 3, anterior)
- 6- Ao fazer isto, a interface, o ambiente de programação/edição do **emacs** é aberto.



O que interessa a esta dissertação, com o **Emacs**, é montar uma página de notícias, e, portanto, apenas os recursos necessários a este objetivo. O Emacs é um editor e um ambiente de programação que não carece de recursos suplementares para efetivação de qualquer projeto de edição ou programação. O

mesmo foi desenvolvido em **LISP**, paradigma utilizado⁵⁷ nesta pesquisa como sendo o ideal e mais aderente aos objetivos traçados, como já visto anteriormente.

9.2 ELEMENTOS DA PÁGINA:

Conforme já foi objeto de estudo no Capítulo 8, a prova de conceito de uma página deverá conter;

- 1- TÍTULO
- 2- SUBTÍTULO
- 3- TEXTO
- 4- TEXTO COM LINK INTERNO
- 5- TABELA ATIVA
- 6- LINK FORA DO TEXTO (página, audio, vídeo)
- 7- IMAGEM
- 8- VIDEO LOCAL
- 9- ÁUDIO LOCAL
- 10-UM PROGRAMA PARA INTERAGIR COM O LEITOR/USUÁRIO, UTILIZANDO OS CONCEITOS JÁ VISTOS (data mining, text mining, manipulação de dados, etc.).

9.3 EMACS – ORG-MODE

ORG-MODE é a ferramenta de **markup**⁵⁸ mais utilizada no mundo para formalizações de manuais, trabalhos técnicos e páginas, onde toda a edição é feita de forma textual, e, posteriormente, pode ser compilada para qualquer formato desejado, inclusive outras linguagens de **markup**, tais como: **Latex**, **HTML**, **PDF**, **DOC** e outros.

A mesma permite a criação de tabelas com fórmulas da mesma forma (mas mais segura e eficiente) que faz o Excel, por exemplo, inserção de programas em diversas linguagens, como: java, C, Lisp e outras. Tudo em formato texto, de tal forma que o arquivo não fica datado, ou seja, não dependa de dlls, libs, etc...

⁵⁷ Lembrando que DrRacket é Scheme e Scheme é um dialeto do LISP.

⁵⁸ **Markup language:** linguagem de anotação com tags, marcações, que tornam as partes do textos distinguíveis sintaticamente umas das outras. https://en.wikipedia.org/wiki/Markup_language

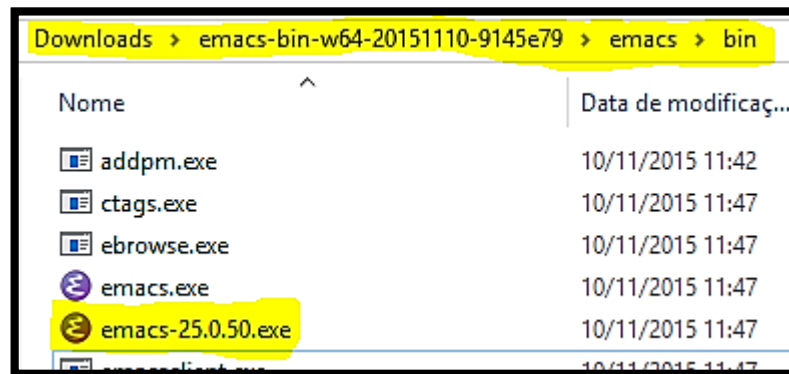
Formato texto sempre vai ser possível de ser lido, já que é um vetor de caracteres (uma string).

No endereço http://orgmode.org/worg/org-tutorials/orgtutorial_dto.html são apresentadas mais características e potencialidades do ORG-MODE, além das necessárias, aqui, para a prova de conceito da geração de uma página dinâmica.

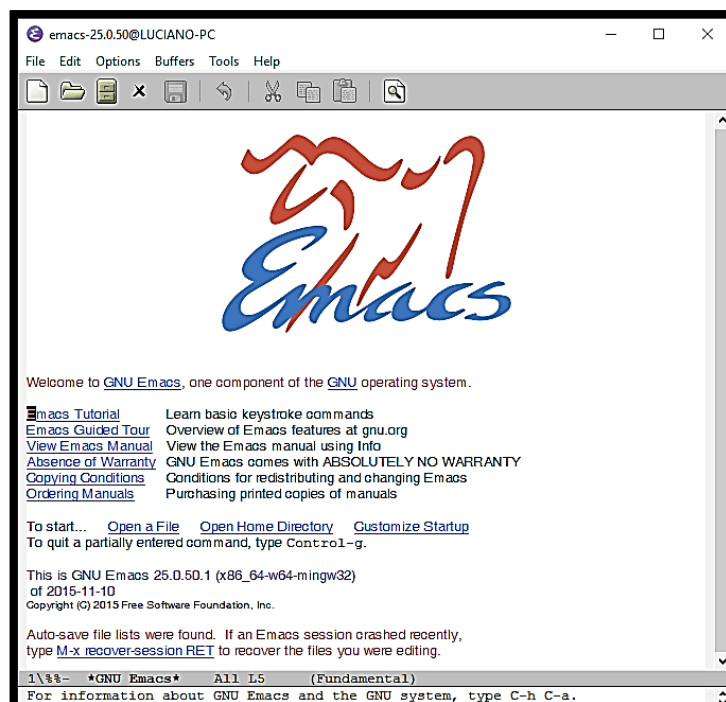
9.3.1 Iniciando Uma Página No Org-Mode.

- 1- Abra o Emacs clicando no arquivo **emacs-25.0.50.exe** (a versão do exemplo) no local instalado.

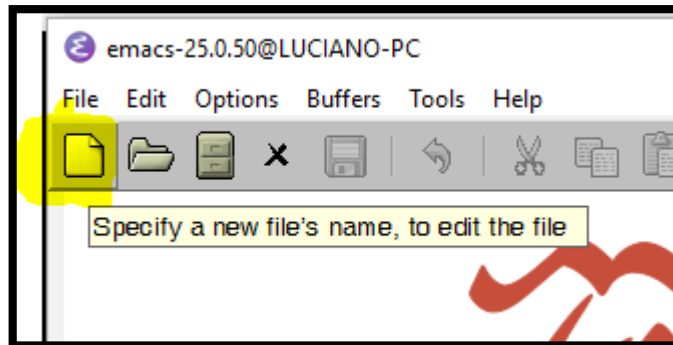
Downloads\emacs-bin-w64-20151110-9145e79\emacs\bin\emacs-25.0.50.exe



- 2- Ao fazer isto a interface, o ambiente do Emacs é aberto.

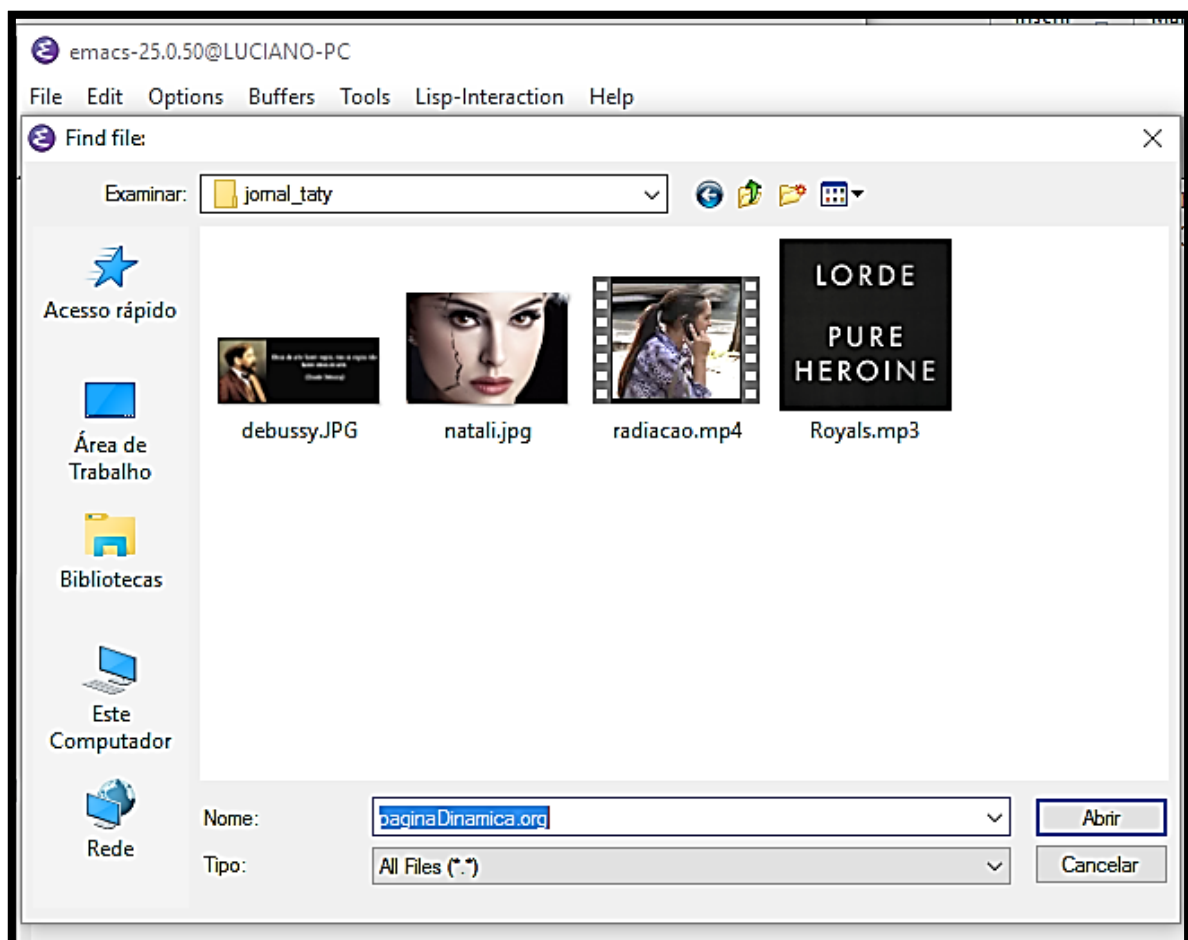


3- Abra um arquivo novo: **File->**

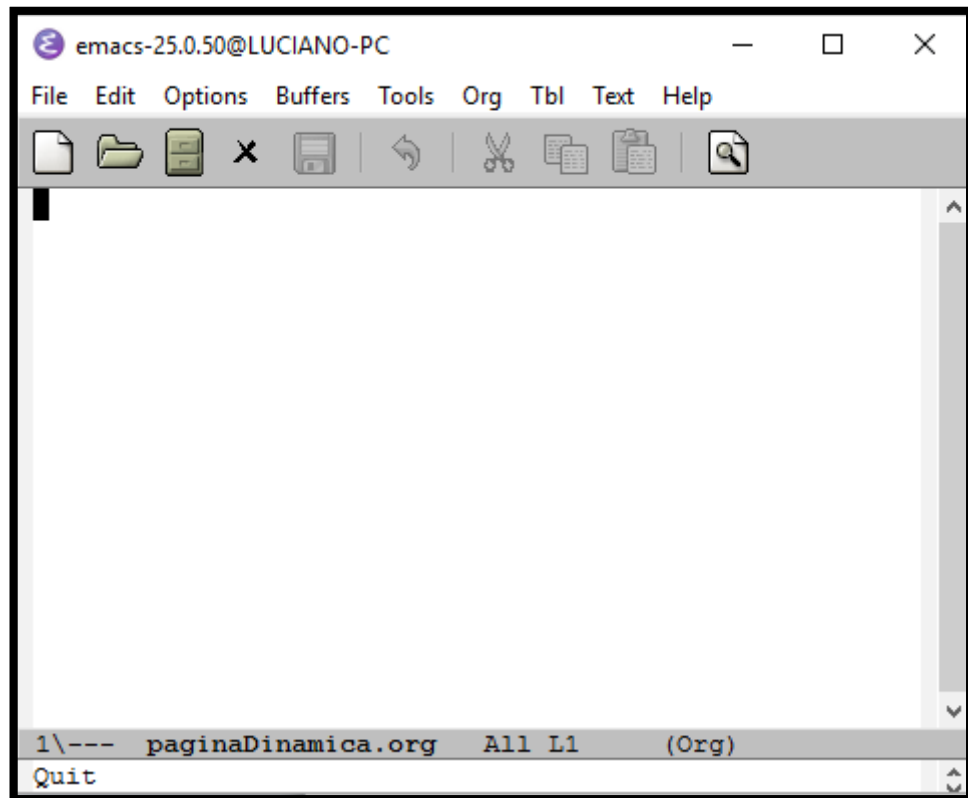


Obs. Poder-se-ia ter utilizado os comandos **C-x C-f**, com a diferença que a linha de comando na parte inferior da janela do Emacs seria ativada para entrada textual do caminho completo e nome do arquivo a ser salvo.

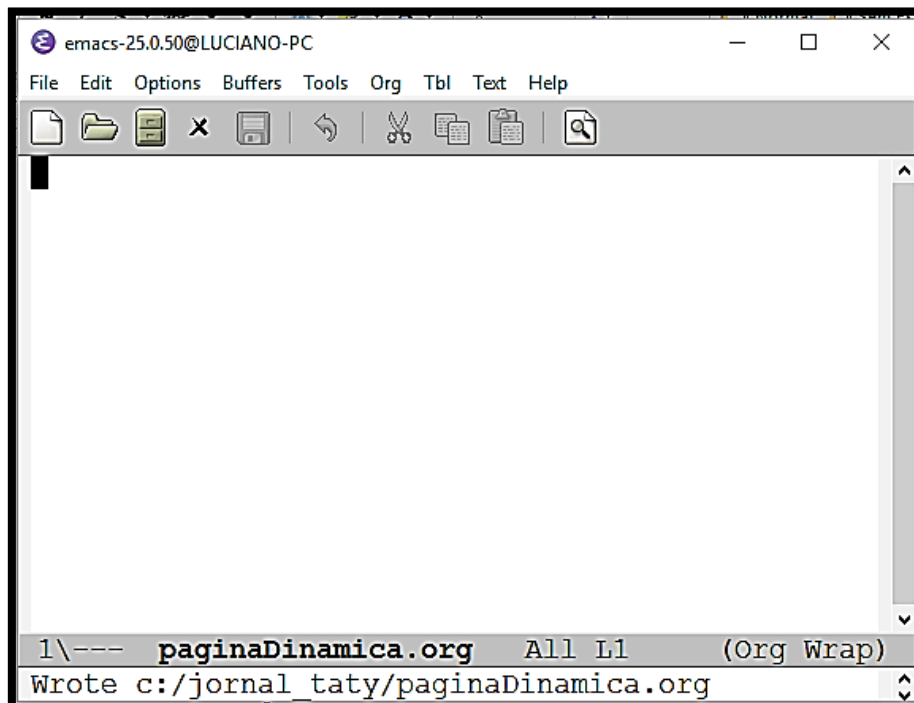
4- Dê um nome ao arquivo, por exemplo: `paginaDinamica.org` e salve a mesma em um diretório de trabalho, como, por exemplo, o `c:\jornal_taty`



5- Clique em Abrir e uma página nova é criada para eu desenvolvimento.



6- Digite **C-x C-s** para salvar a página



7- OK, já se pode iniciar os trabalhos. Após criar a página e começar a programar, você pode salvar normalmente pelo menu **File** ou pelo atalho **C-x C-s** (onde **C** é a tecla **Ctrl** do teclado). Quando você salva o arquivo

com a extensão **.org**, automaticamente o **emacs** reconhece todos os comandos do **org-mod** internos do arquivo salvo.

- 8- **A primeira coisa que será feito, no passo a passo, é o cabeçalho padrão** do para gerar uma página **HTML** simples, tal como: **nome da página, um título, dois subtítulos e/não um índice**

No cabeçalho padrão, pode-se desejar, ou não, se ter um índice e ter ou não o autor da página.

- 1- Para o título da página, o comando é :

#+TITLE: seguido do Título da página

- 2- Para incluir o nome do Autor, opcional, o comando é:

#+AUTHOR: seguido do autor

- 3- Para escolher se vai o não ter índice, o comando é:

#+OPTIONS: toc: n, onde **n** é a quantidade de sub-títulos no corpo de texto que aparecerão na tabela de conteúdo (sumário). Se **n** for igual a **nil**, isto significa que não será montado um sumário, se for **1** ou **2**, indica a quantidade de sub-títulos (headlines) que o sumário conterà por título. Se não se colocar **#+OPTIONS:**, o padrão é ter índice com dois subtítulos.

- 4- **Um título do texto** é iniciado por um símbolo ***** seguido de espaço, seguido de texto.

- 5- **Um sub-título do texto** é iniciado por dois ou mais símbolos ***** (ex. ******, *******, ...) seguido de espaço, seguido de texto.

Exemplos:

Título: * Título 1

Subtítulo: ** Subtítulo 1

Subtítulo do subtítulo: *** SubSubtítulo 1 (neste caso, o **toc** tem que ser **3**, pelo menos)

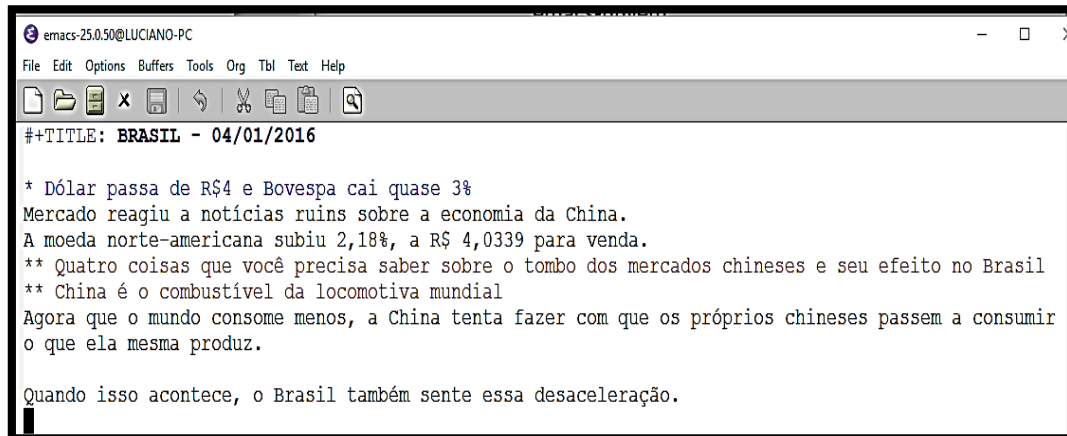
- 6- Linhas grandes não precisam ser quebradas, o código gerado faz com que cada linha se adapte ao tamanho da tela.

- 7- Quando se deseja saltar ou quebrar de propósito uma linha, deve-se dar dois **ENTERs** no ponto de quebra desejado.

9.3.2 EXEMPLOS DE CÓDIGOS NO EMACS E PÁGINA HTML CORRESPONDENTE.

9.3.2.1 Com índice

Arquivo **emacs – org-mode** sem o uso de **#+OPTIONS:** no cabeçalho.



```

emacs-25.0.50@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
#+TITLE: BRASIL - 04/01/2016

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir
o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

```

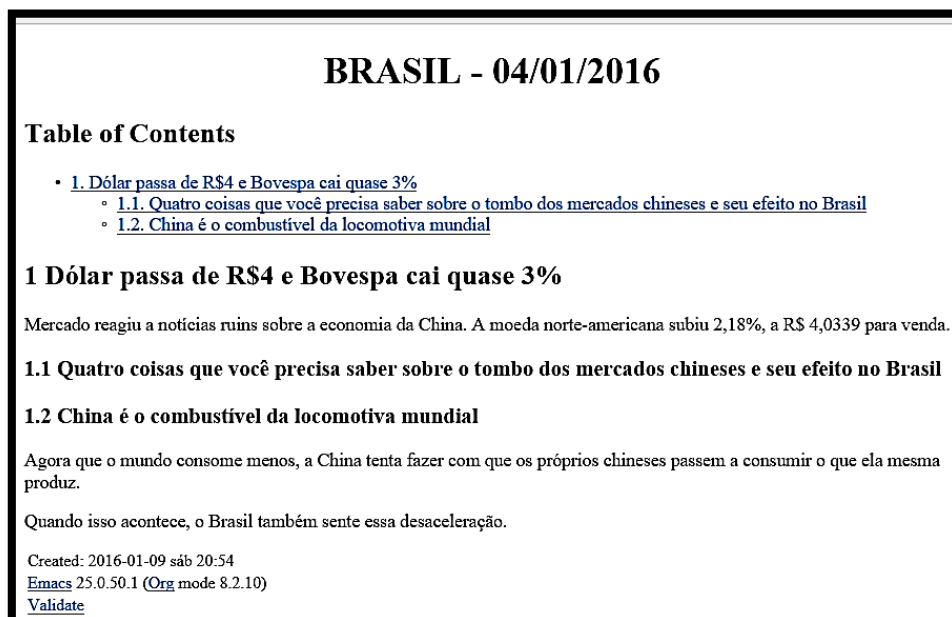
Para Salvar o arquivo, digita-se **C-x C-s**

Para gerar a página HTML correspondente à codificação no **org-mode**, o sequencia de comandos é: **C-c C-e h o** . Com esta sequencia uma página **html** será gerada e aberta. Caso se deseje apenas criar a página, a sequência é: **C-c C-e h h** .

Será utilizado em toda esta dissertação, este capítulo, a sequência:

C-c C-e h o

Utilizando esta sequência, a página gerada fica assim:



```

BRASIL - 04/01/2016

Table of Contents

  • 1. Dólar passa de R\$4 e Bovespa cai quase 3%
    • 1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
    • 1.2. China é o combustível da locomotiva mundial

1 Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China. A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.

1.1 Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil

1.2 China é o combustível da locomotiva mundial

Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

Created: 2016-01-09 sáb 20:54
Emacs 25.0.50.1 (Org mode 8.2.10)
Validate

```

Observe que no trecho do **emacs**

```
* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
```

Tinha-se duas linhas no texto sobre o dólar passar de 4 reais, e no HTML gerado, não:

```
1 Dólar passa de R$4 e Bovespa cai quase 3%
```

```
Mercado reagiu a notícias ruins sobre a economia da China. A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
```

Já, neste outro trecho no **emacs**

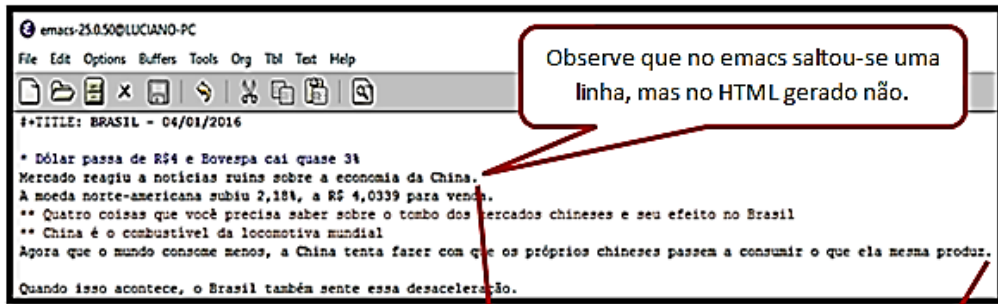
```
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.
Quando isso acontece, o Brasil também sente essa desaceleração.
```

No **HTML** saltou-se uma linha.

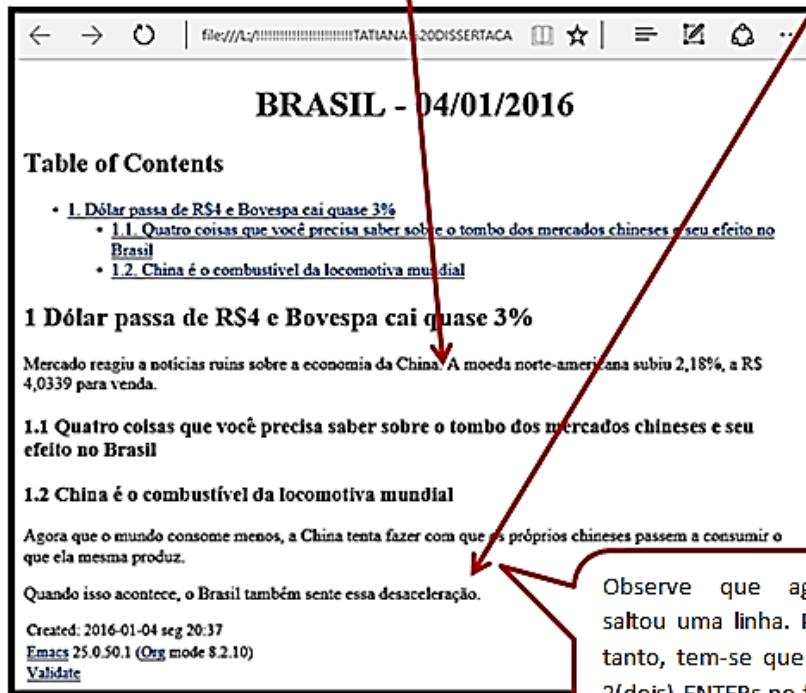
```
1.2 China é o combustível da locomotiva mundial
```

```
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.
Quando isso acontece, o Brasil também sente essa desaceleração.
```

Quando se deseja que se quebre uma linha, tem-se que saltar duas linhas no ponto de quebra (dois **ENTERs**)

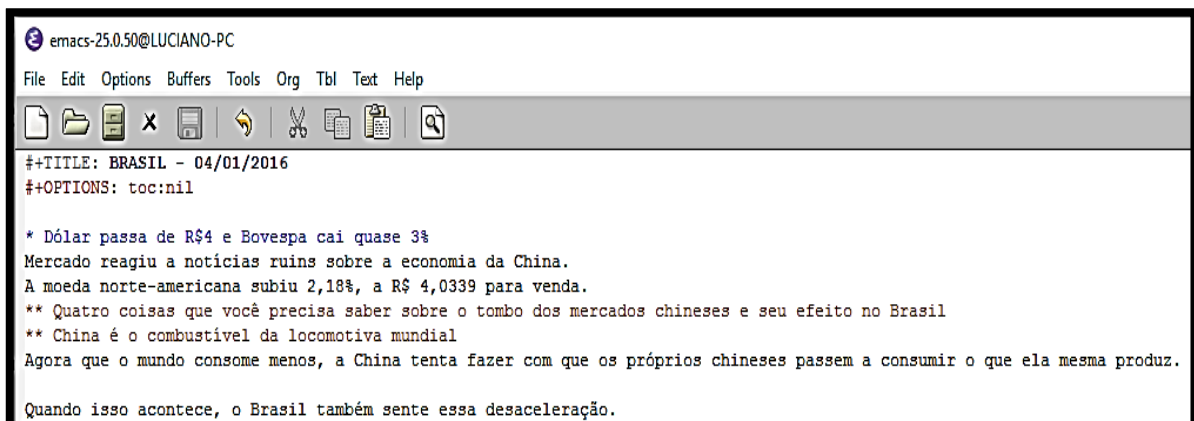


Arquivo HTML com índice, já que não se usou no cabeçalho o comando **#+OPTIONS:**



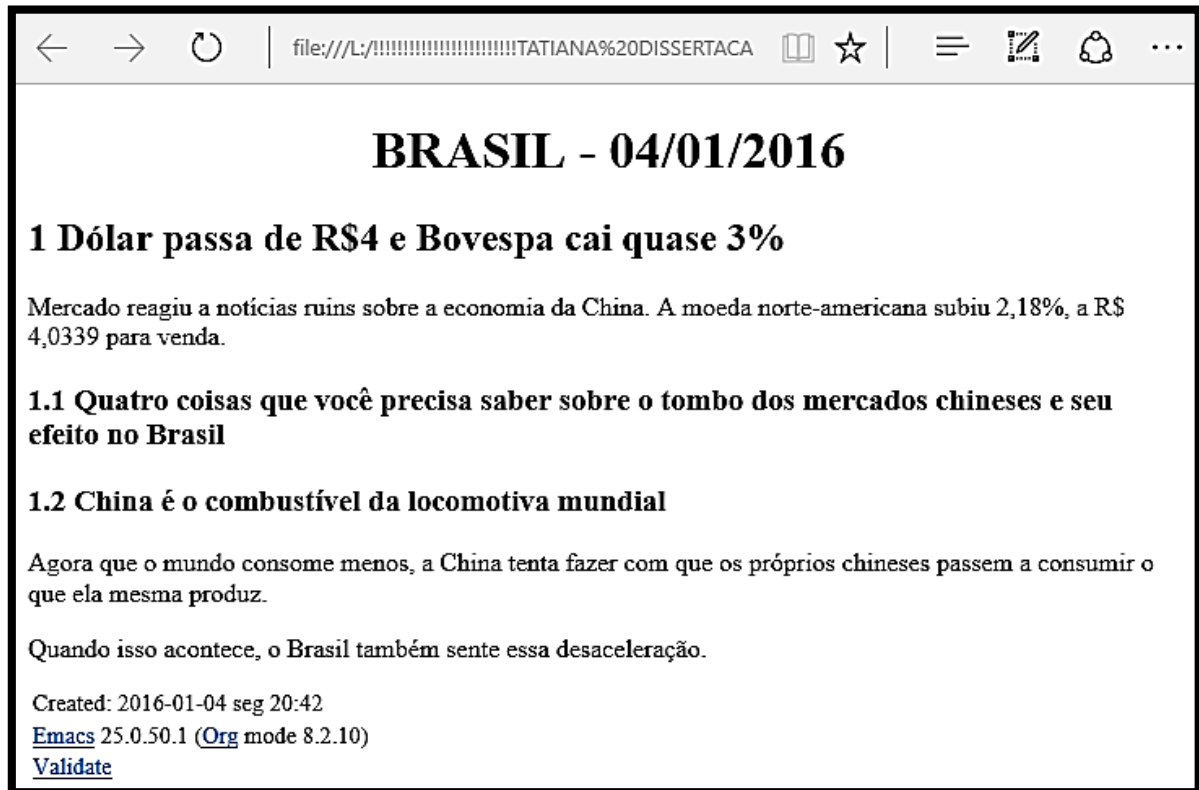
9.3.2.2 Sem índice

Arquivo emacs – org-mode com o uso de **#+OPTIONS:** no cabeçalho.



Gerando a página da notícia HTML (C-c C-e h o), sem índice, fica:

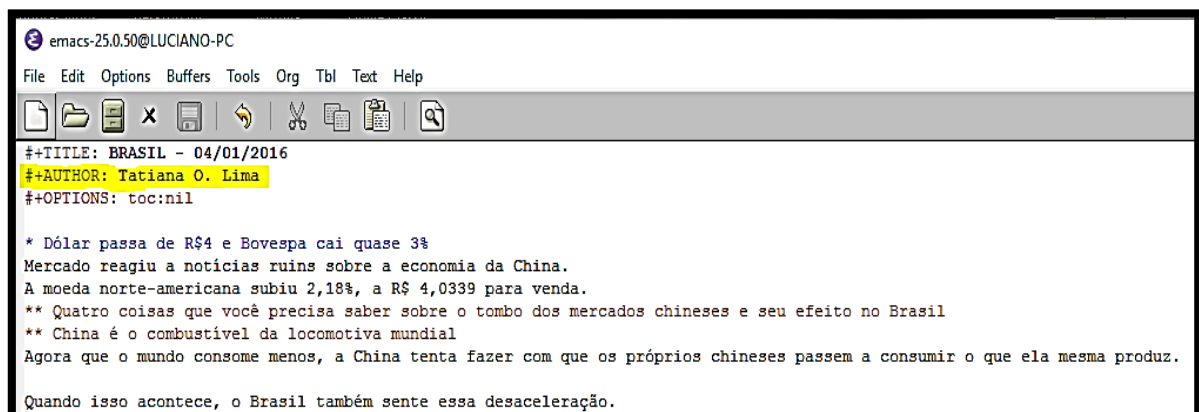
#+OPTIONS: toc:nil



Observe que automaticamente a página é criada, com índice, itens e sub-itens:

9.3.2.3 Colocando o nome do autor na página

Basta colocar o comando **#+AUTHOR:** seguido do nome do autor



< PÁG | pagl | Pági | Pági | BRA | BRA' | I X | + - □ X

← → ↻ | file:///L:/!!TATIANA%20DISSERTACA | ☆ | ≡ | 🔔 | ⋮

BRASIL - 04/01/2016

1 Dólar passa de R\$4 e Bovespa cai quase 3%

Mercado reagiu a notícias ruins sobre a economia da China. A moeda norte-americana subiu 2,18%, a R\$ 4,0339 para venda.

1.1 Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil

1.2 China é o combustível da locomotiva mundial

Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

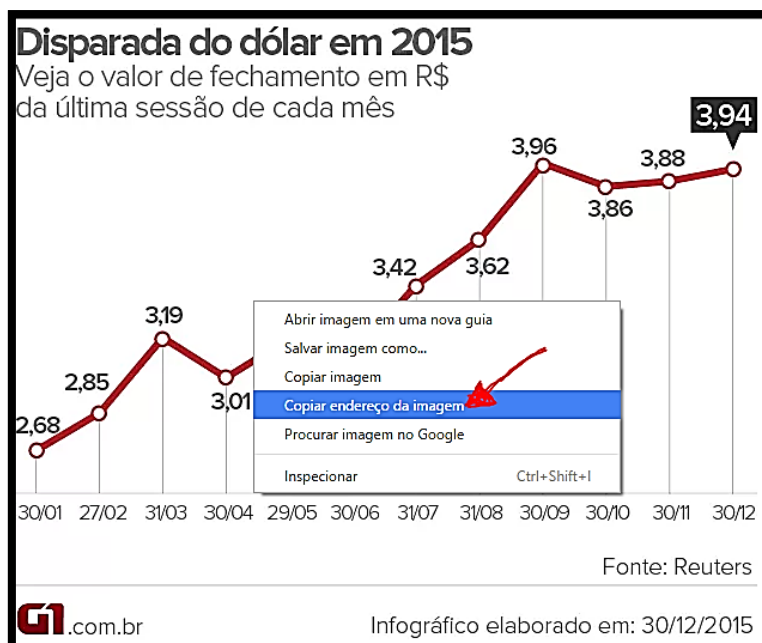
Author: Tatiana O. Lima
 Created: 2016-01-04 seg 21:49
 Emacs 25.0.50.1 (Org mode 8.2.10)
[Validate](#)

9.3.2.4 Colocando uma imagem utilizando o link da mesma na internet

Basta colocar o link da IMAGEM, capturando o mesmo na internet.

Basta localizar a imagem que deseja, colocar o cursor do mouse em cima dela e clicar com o botão direito do mouse. Um conjunto de opções é aberto.

Escolha **Copiar endereço da imagem**



Se for colocar a imagem logo após um texto, para que a imagem não fique grudada após o texto, salte duas linhas (dois ENTERs) antes de colocar o link no texto. Se for após um título ou subtítulo, não precisa.

#+OPTIONS: toc:2 significa que será colocado no índice até 2(dois) subníveis (subíndices) de índice.

```

emacs-25.0.50@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
#+TITLE: BRASIL - 04/01/2016
#+AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:2

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.glbimg.com/-3yFTcECSEmwXCRn1ElCsqrdw=/s.glbimg.com/jo/gl/f/original/2015/12/30/dolar-30.12\_v2.jpg

```

← → ↻ | file:///L:/!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!TATIANA%20DISSERTACAO%202015/ta

BRASIL - 04/01/2016

Table of Contents

- [1. Dólar passa de R\\$4 e Bovespa cai quase 3%](#)
 - [1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil](#)
 - [1.2. China é o combustível da locomotiva mundial](#)
- [2. VEJA A ESCALADA DO DÓLAR EM 2015](#)

1 Dólar passa de R\$4 e Bovespa cai quase 3%

Mercado reagiu a notícias ruins sobre a economia da China. A moeda norte-americana subiu 2,18%, a R\$ 4,0339 para venda.

1.1 Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil

1.2 China é o combustível da locomotiva mundial

Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

2 VEJA A ESCALADA DO DÓLAR EM 2015

Disparada do dólar em 2015

Veja o valor de fechamento em R\$ da última sessão de cada mês

Data	Valor (R\$)
30/01	2,68
27/02	2,85
31/03	3,19
30/04	3,01
29/05	3,18
30/06	3,10
31/07	3,42
31/08	3,62
30/09	3,96
30/10	3,86
30/11	3,88
30/12	3,94

Fonte: Reuters

G1.com.br Infográfico elaborado em: 30/12/2015

Autor: Tatiana O. Lima
 Created: 2016-01-04 seg 22:17
 Emacs 25.0.50.1 (Org mode 8.2.10)
[Validate](#)

9.3.2.5 Colocando uma imagem na página que está em um diretório local (no C:\jornal taty) e não um link. No caso, a imagem deve estar no mesmo diretório do arquivo .org.

Basta colocar o seguinte formato: **file:nome do arquivo**. Não precisa colocar sublinhado, o emacs coloca automaticamente após escrever **file:**.

```

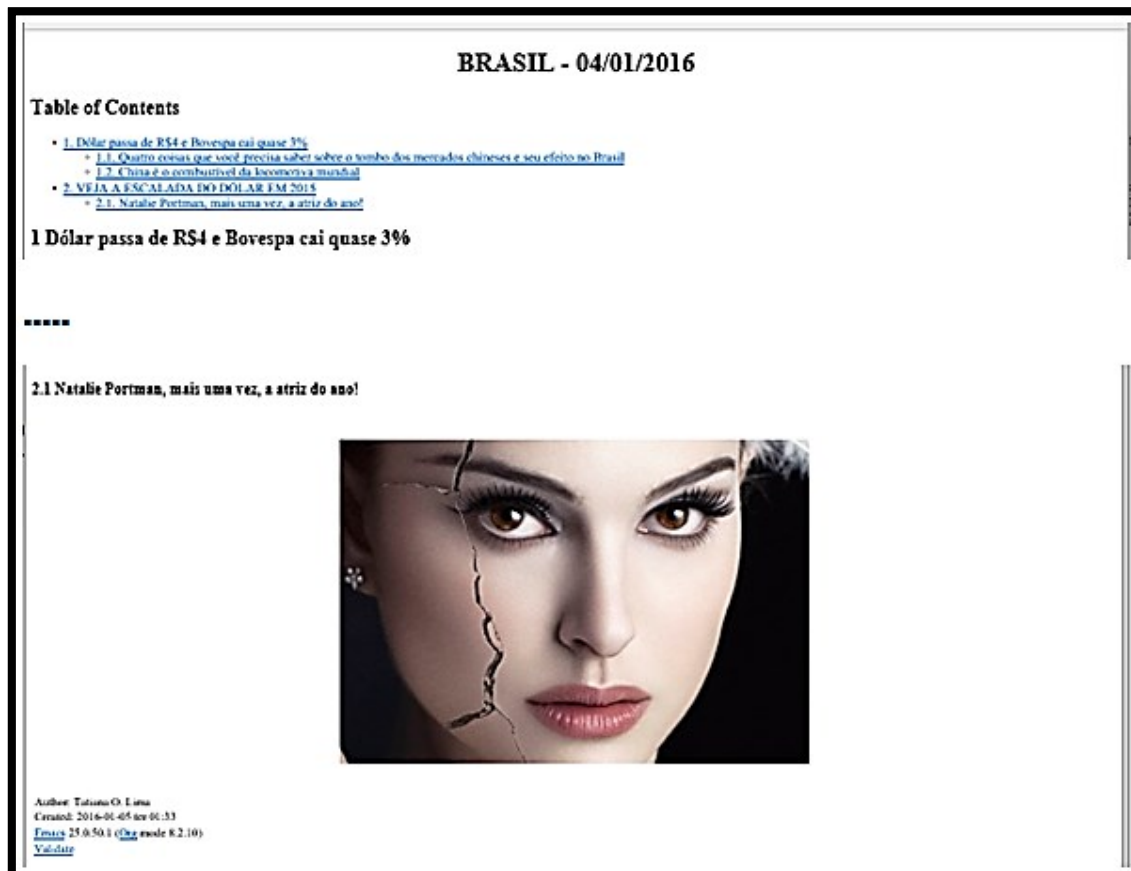
emacs-25.0.50@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
[Icons]
#+TITLE: BRASIL - 04/01/2016
#+AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:2

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.glbimg.com/-3yfTcECSEmwdXCRn1ElCsqrdDw=/s.glbimg.com/jo/g1/f/original/2015/12/30/dolar-30.12\_v2.jpg
** Natalie Portman, mais uma vez, a atriz do ano!
file:natali.jpg

```



9.3.2.6 Colocando um link ativo dentro de um texto.

Basta colocar no seguinte formato: **[[link] [palavra link]]**. No caso **[[https://www.youtube.com/watch?v=jFPpwM_iS7c][Ataque dos Clones]]**

Quando acaba de digitar esta estrutura, ela muda para um link: **Ataque dos Clones**

```
** Natalie Portman, mais uma vez, a atriz do ano!  
file:natali.jpg
```


```
É interessante ver a atuação da Natalie Portman no filme Ataque dos Clones de 2002.
```

BRASIL - 04/01/2016

Table of Contents

- [1. Dólar passa de R\\$4 e Bovespa cai quase 3%](#)
 - [1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil](#)
 - [1.2. China é o combustível da locomotiva mundial](#)
- [2. VEJA A ESCALADA DO DOLAR EM 2015](#)
 - [2.1. Natalie Portman, mais uma vez, a atriz do ano!](#)

2.1 Natalie Portman, mais uma vez, a atriz do ano!



É interessante ver a atuação da Natalie Portman no filme [Ataque dos Clones](#) de 2002.

Author: Tatiana O. Lima
Created: 2016-01-05 10:01:46

OBS. Este link pode ser para qualquer site ou mídia na internet. Somente será ativado após clicado. É diferente do link para figura que é aberto assim que encontrado. Assim, pode-se colocar um link de um vídeo ou de música que, quando clicado nele, no link, o mesmo dispara a mídia adequada à reprodução.

9.3.2.7 Colocando imagem seguida de link na mesma linha.

Isto é uma combinação do que já foi visto anteriormente. Como os recursos já vistos pode-se formatar qualquer notícia padrão de um jornal, onde, na Web, a maioria das informações são extraídas dela mesma, ou seja, tem-se praticamente links para tudo que se pesquisa, tais como: Imagens, vídeos, músicas, sons, textos, outros.

No caso deste item 7, o que desejamos fazer é colocar um título referenciando a uma música de Claude Debussy e na linha seguinte colocar a figura de Claude Debussy seguido de um link para a música Claire the Lune.

Fica assim a codificação no Emacs

Título:

*** OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU,
COMPOSTA POR CLAUDE DEBUSSY.**

Figura:

file:debussy.jpg

Link da música:

**[[<http://www.kboing.com.br/claude-debussy/1-1088551/>][CLAIR THE
LUNE]]**

Texto na linha abaixo da figura:

... salta duas linhas (dois ENTER)

**TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E
PAZ NO CORAÇÃO!**

OBS: O link deve vir logo após a codificação de inserção da figura para ficar um depois do outro. Enquanto não se coloca 2 (dois) ENTER (salta linha) após um elemento, a linha continua. Somente no caso de títulos e subtítulos que sempre se inicia uma linha nova automaticamente.

O Arquivo Emacs fica assim:

```

emacs-25.0.50@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
[Icons]
#+AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:2

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz

Quando isso acontece, o Brasil também sente essa desaceleração.

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.glbimg.com/-3yftcEC5EmwdXCRn1ElCsqRDw/s.glbimg.com/jo/g1/f/original/2015/12/30/dolar-30.12\_v2.jpg

** Natalie Portman, mais uma vez, a atriz do ano!
file:natali.jpg

É interessante ver a atuação da Natalie Portman no filme Ataque dos Clones de 2002.
* OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.
file:debussy.jpg CLAIR THE LUNE

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

```

A página da notícia HTML fica:

BRASIL - 04/01/2016

Table of Contents

- [1. Dólar passa de R\\$4 e Bovespa cai quase 3%](#)
 - [1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil](#)
 - [1.2. China é o combustível da locomotiva mundial](#)
- [2. VEJA A ESCALADA DO DÓLAR EM 2015](#)
 - [2.1. Natalie Portman, mais uma vez, a atriz do ano!](#)
- [3. OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.](#)



É interessante ver a atuação da Natalie Portman no filme [Ataque dos Clones](#) de 2002.

3 OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.



Obras de arte fazem regras, mas as regras não fazem obras de arte.

(Claude Debussy)

[CLAIR THE LUNE](#)

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

Author: Tatiana O. Lima

Com apenas 17 linhas textuais obtém-se uma página HTML bem formatada e rica de mídias normalmente desejadas em uma notícia de um jornal.

O jornalista se preocupa com o texto e o **emacs-org-mode** se encarrega de formatar adequadamente, segundo as normas mais aceitas e vigentes.

9.3.2.8 Colocando tabelas dinâmicas, com fórmulas por campo, no org-mode.

Tabelas são elementos muito importantes para o jornalista, bem como para todas as áreas do conhecimento, principalmente as ativas onde um campo da tabela pode ser utilizado para inferir outro campo, tal como totalizações, transformações ou outro tipos de inferências dada uma função pré-determinada. O **org-mode** permite que se construa tais tabelas textualmente, e, uma vez compiladas, elas são geradas graficamente no formato que forem salvas (HTML, PDF, outros).

Para implementar uma tabela, utiliza-se o símbolo '|' como separador de campos.

- 1- Entre com o nome da tabela

#+NAME: TABELA DE CONVERSÃO DE TAMANHO DE CELULARES EM POLEGADAS PARA CENTÍMETRO

- 2- Entre com a primeira linha. No último elemento da tabela, em vez de colocar o símbolo '|', aperte a tecla **Tab** para fechar a linha e formatar a tabela, iniciando uma nova linha.

O diagrama mostra uma tabela em org-mode com três colunas: 'MODELO', 'POLEGADAS' e 'CENTÍMETRO'. Cada coluna tem um espaço reservado para um valor, representado por uma barra vertical. Uma caixa de texto vermelha aponta para o final da primeira linha da tabela, contendo o texto: 'Clicar em Tab para fechar a linha da coluna e abrir uma nova linha.'.

MODELO	POLEGADAS	CENTÍMETRO

- 3- Insira os elementos das colunas, clicando em tab a cada elemento entrado para ir para o próximo campo da tabela.

```
#+NAME: CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       |              |
| Galaxy S6  | 5.65      |              |
| lumia 920  | 5.13      |              |
```

4- Colocando uma fórmula na tabela (TBLFM).

Ao colocar o cursor na coluna onde se deseja inserir uma fórmula, digite:

C-c = . Ao fazer isto, uma linha de comando é aberta na parte inferior do ambiente do Emacs para você entrar com a fórmula. A fórmula, a função, pegará o segundo elemento da linha, segunda coluna, e converterá o valor em polegadas para metro. O comando **\$** seleciona uma coluna da linha. No caso, o cursor está parado na coluna 3, ou seja: **\$3**.

```
#+NAME: CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       |              |
| Galaxy S6  | 5.65      |              |
| lumia 920  | 5.13      |              |
| Coolpad Halo | 7.2835    |              |

1\*- pagDin3.org 27% L40 (Org)
Column formula $3=
```

Conversão: 1 polegada = 2.54 centímetros

A fórmula de conversão fica assim: $\$3=(2.54*\$2)$

Digitada a fórmula, clique em ENTER. A fórmula aparecerá na parte inferior da tabela e o valor já convertido será automaticamente inserido na coluna selecionada na tabela.

```
#+NAME: CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       | 13.97       |
| Galaxy S6  | 5.65      |              |
| lumia 920  | 5.13      |              |
| Coolpad Halo | 7.2835    |              |
#+TBLFM: $3=(2.54*$2)
```

Para inserir a fórmula em um outro campo, por exemplo, na coluna abaixo, basta colocar o cursor nela e digitar novamente C-c = que a fórmula anterior aparece na linha de comando na parte inferior do ambiente do Emacs.

```
#+NAME:CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       | 13.97       |
| Galaxy S6  | 5.65      |              |
| lumia 920  | 5.13      |              |
| Coolpad Halo | 7.2835   |              |
#+TBLFM: $3=(2.54*$2)

1\*- pagDin3.org 27% L41 (Org)
Column formula $3=(2.54*$2)
```

Desejando colocar a mesma, tecla ENTER, se não, modifique e tecla ENTER. Repita o procedimento para as demais colunas.

```
#+NAME:CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       | 13.97       |
| Galaxy S6  | 5.65      | 14.351      |
| lumia 920  | 5.13      | 13.0302     |
| Coolpad Halo | 7.2835   | 18.50009    |
#+TBLFM: $3=(2.54*$2)
```

O código completo no emacs-org-mode fica assim:

```
emacs-25.0.50@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
[Icons]
#+TITLE: BRASIL - 04/01/2016
#+AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:4

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
*** Atividade industrial mais fraca
*** Implantação do 'circuit breaker'
*** Yuan mais fraco
*** Venda de ações...
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.glbimg.com/-3yftcEC5EmwdXCRn1E1CsgdRDw=/s.glbimg.com/jo/g1/f/original/2015/12/30/dolar-30.12\_v2.jpg

** Natalie Portman, mais uma vez, a atriz do ano!...
* OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.
file:debussy.jpg CLAIR THE LUNE

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

** MÚSICA É TUDO DE BOM!...
* VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO
#+NAME:CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       | 13.97       |
| Galaxy S6  | 5.65      | 14.351      |
| lumia 920  | 5.13      | 13.0302     |
| Coolpad Halo | 7.2835   | 18.50009    |
#+TBLFM: $3=(2.54*$2)
```

A página HTML fica:

BRASIL - 04/01/2016

Table of Contents

- [1. Dólar passa de R\\$4 e Bovespa cai quase 3%](#)
 - [1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil](#)
 - [1.1.1. Atividade industrial mais fraca](#)
 - [1.1.2. Implantação do 'circuit breaker'](#)
 - [1.1.3. Yuan mais fraco](#)
 - [1.1.4. Venda de ações](#)
 - [1.2. China é o combustível da locomotiva mundial](#)
- [2. VEJA A ESCALADA DO DOLAR EM 2015](#)
 - [2.1. Natalie Portman, mais uma vez, a atriz do ano!](#)
- [3. OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.](#)
 - [3.1. MÚSICA É TUDO DE BOM!](#)
- [4. VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO](#)
 - [4.1. RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!](#)
- [5. EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.](#)

1 Dólar passa de R\$4 e Bovespa cai quase 3%

...



É interessante ver a atuação da Natalie Portman no filme [Ataque dos Clones](#) de 2002.

3 OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.



[CLAIR THE LUNE](#)

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

3.1 MÚSICA É TUDO DE BOM!



4 VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO

MARCA	POLEGADAS	CENTÍMETROS
iPhone 6	5.5	13.97
Galaxy S6	5.65	14.351
lumia 920	5.13	13.0302
Coolpad Halo	7.2835	18.50009

Estes tipos de tabelas agilizam o processo de edição de uma notícia.

9.3.2.9 Inserindo um vídeo direto na página através de um link com código de incorporação

Quando se deparar com um vídeo na internet que deseje colocar na página de notícias, e não apenas um link para abrir a página com o vídeo, você pode colocar o código de incorporação direto no org-mode. O código vem em HTML, mas você não precisa entender nada sobre ele, apenas copiar e colar na estrutura do org-mode.

Faça o seguinte. Coloque o cursor do mouse em cima do vídeo que deseja incorporar, e clique com o botão direito do mouse. Um menu de opções vai abrir. Clique em **Copiar código de incorporação**.

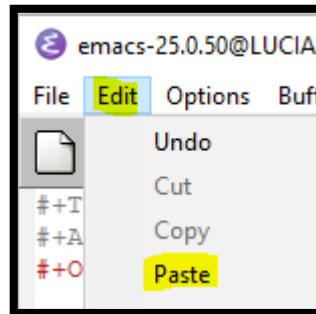


Feito isto, digite os seguintes comandos no org-mode, saltando uma linha entre o **BEGIN** e o **END**:

#+BEGIN_HTML

#+END_HTM

Coloque o cursor do mouse na linha entre os comandos, clique no menu Edit -> opção Paste na janela do emacs para colocar o código



#+BEGIN_HTML

```
<iframe width=".." height=".." src="http://www..." frameborder="0"
allowfullscreen></iframe>
```

#+END_HTML

Eis um exemplo no emacs, org-mode:

```
** RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!
#+BEGIN_HTML
<iframe width="640" height="360" src="https://www.youtube.com/embed/pC4Z0xpu2rs" frameborder="0" allowfullscreen></iframe>
#+END_HTML
```

Este trecho de código fica assim na página HTML:



9.3.2.10 Inserindo áudio com o player de áudio direto na página de um arquivo que está na pasta da página html

Em vez de colocar um link para a página com o áudio, você pode colocar o player de áudio na página, que, ao clicar em tocar, um áudio especificado por você será executado.

Novamente, um código HTML será inserido entre os comandos já vistos. Neste caso, você terá que decorar o código HTML para isto. É a forma mais fácil. As que independem de decorar o código HTML exigem recursos de programação avançado que se tornam mais difíceis do que decorar um simples código, como segue.

#+BEGIN_HTML

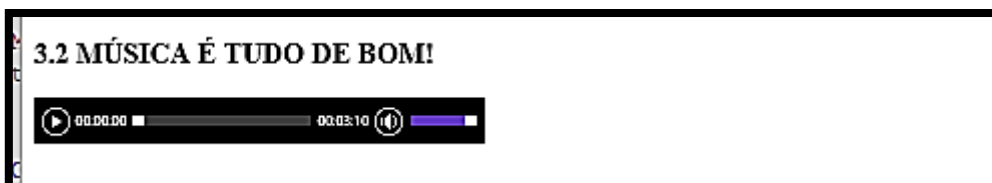
```
<audio controls> <source src="audio.mp3"> </audio>
```

#+END_HTML

Eis um exemplo no emacs, org-mode:

```
** MÚSICA É TUDO DE BOM!
#+BEGIN_HTML
<audio controls><source src="Royals.mp3"></audio>
#+END_HTML
```

Este trecho de código fica assim na página HTML:



Observe que o nome da música não aparece. Caso deseje, acrescente a mesma como texto. Os controles disponibilizados é o de executar, mudo e de volume.

9.3.2.11 Inserindo um vídeo com o player de vídeo direto na página de um arquivo que está na pasta da página html

A solução é a mesma sugerida para o áudio, mudando apenas a palavra **áudio** para **vídeo**. Para que o vídeo rode em qualquer browser, deve-se optar pelo

formato mp4. Se o seu vídeo for em outro formato, converta. Existem vários programas de conversão⁵⁹ (como o [Format Factory](#), por exemplo).

#+BEGIN_HTML

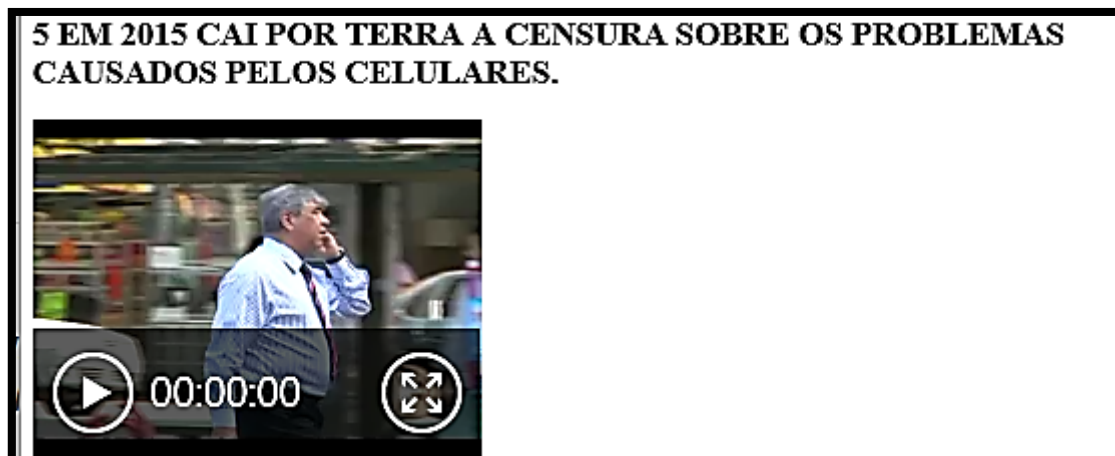
```
<video controls> <source src="video.mp4"> </video>
```

#+END_HTML

Eis um exemplo no emacs, org-mode:

```
* EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.
#+BEGIN_HTML
<video controls><source src="radiacao.mp4"></video>
#+END_HTML
```

Este trecho de código fica assim na página HTML:



9.3.2.12 Inserindo um link em uma imagem: usando uma imagem como botão

Em muitos aplicativos, é interessante que se crie botões com ícones, figuras. Melhor, criar gadgets. O grande problema é que isto requer recursos, features que são datadas, ou seja, mudam de acordo com bibliotecas existentes nos browsers. Se estiver utilizando um recurso e o mesmo for alterado em uma versão do browser ou sistema operacional, sua página fica comprometida.

Assim, nada melhor que usar os recursos básicos que o **HTML** padrão permite.

⁵⁹ Conversores gratuitos: <http://www.techtudo.com.br/artigos/noticia/2012/04/top-5-conversores-de-video.html>

Fazer isto no emacs ORG MODE é simples. Usa-se a estrutura do item anterior, ou seja:

[[link para onde se quer ir][file:imagemBotao.jpg]

Exemplo:

[[https://studio.code.org/c/139619831][file:jogo.jpg]]

Que, ao fechar o último colchete, fica assim no emacs org-mode:

```
* CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 200.000
INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOUR OF CODE" AO MESMO TEMPO
Veja um jogo criado em 1 minuto e 30 segundos em CODE

[[https://studio.code.org/c/139619831][file:jogo.jpg]]
```

```
* CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 200.000
INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOUR OF CODE" AO MESMO TEMPO
Veja um jogo criado em 1 minuto e 30 segundos em CODE

file:jogo.jpg
```

No Html fica assim



Acrescentou-se às mídias que devem estar no mesmo diretório que a página html que será gerada, uma figura: **jogo.jpg**.

Na mesma serão colocados, inicialmente, 4 (quatro) arquivos de mídia local:

- 1- debussy.jpg - imagem
- 2- 2-natali.jpg - imagem
- 3- Royals.mp3 - áudio mp3
- 4- radiacao.mp4 - vídeo mp4
- 5- jogo.jpg - imagem

9.3.2.13 Inserindo programação nas páginas dinâmicas.

Muitas vezes é necessário ao jornalista, e outros profissionais, realizarem cálculos, buscas, consulta a banco de dados, outros, para inserir na notícia que vai publicar. Inserir programas no código da página é um recurso que pode agilizar bastante estes processos. O emacs-org-mode permite que se insira código, programação, em várias linguagens, o que se flexibiliza ao usuário na escolha e implementação de várias tarefas com este recurso.

Como exemplo de uso deste recurso, é interessante se fazer um **data mining**, um **text-mining**⁶⁰ na própria página.

Para se fazer o text-mining na página, será aproveitada a tabela **CONVERTE** contendo alguns celulares e respectivos tamanhos. A informação buscada será a de encontrar o tamanho do maior celular da tabela.

Por motivos já citados, será utilizado, nesta dissertação, o paradigma funcional e a linguagem Lisp e dialetos (eLisp - Scheme – Racket).

A tarefa é simples. Basta escolher a linguagem de preferência e inserir o código entre os comandos:

```

#+BEGIN_SRC linguagem parâmetros
código
#+END_SRC
```

A linguagem escolhida será **emacs-lisp** uma versão de **common lisp**.

O bloco de código, de programa, junto com a tabela, fica assim:

⁶⁰ **Text Mining – mineração em texto, ou seja, um data mining (mineração de dados) em texto**
- https://en.wikipedia.org/wiki/Text_mining

```

#+NAME:CONVERTE
| MARCA      | POLEGADAS | CENTÍMETROS |
| iPhone 6   | 5.5       | 13.97       |
| Galaxy S6  | 5.65      | 14.351      |
| lumia 920  | 5.13      | 13.0302     |
| Coolpad Halo | 7.2835   | 18.50009    |
#+TBLFM: $3=(2.54*$2)

#+BEGIN_SRC emacs-lisp :var tabela=CONVERTE[,2] :exports results :results value html
(require 'cl)
(defun maior(s)
  (car (sort (cdr tabela) #'>)))
(format "O maior tamanho entre eles chegou a: %s cm" (maior tabela))
#+END_SRC

```

O resultado fica assim:

MARCA	POLEGADAS	CENTÍMETROS
iPhone 6	5.5	13.97
Galaxy S6	5.65	14.351
lumia 920	5.13	13.0302
Coolpad Halo	7.2835	18.50009

O maior tamanho entre eles chegou a: 18.50009 cm

EXPLICANDO:

O cabeçalho com a linguagem e parâmetros fica:

```

#+BEGIN_SRC emacs-lisp :var tabela=CONVERTE[,2] :exports results :results value html

```

Onde:

- **emacs-lisp** -> é a linguagem
- **:var tabela=CONVERTE[,2]** -> significa que está se criando uma

variável chamada tabela que conterá o conteúdo da terceira coluna de **CONVERTE**.

[,2] significa terceira coluna. No lisp, as colunas começam a ser numeradas a partir do 0. 0 é a primeira coluna, 1 a segunda, e, portanto, 2 é a terceira, a qual possui os tamanhos dos celulares dos quais deseja-se saber qual é o maior.

- **:exports results** -> indica que se deseja exportar os resultados
- **:results value html** -> indica que o resultado vai ser exportado como

HTML

O PROGRAMA, A CODIFICAÇÃO EM ELISP

Inicia importando a linguagem. A sintaxe é: **(require 'cl)**, onde **cl** significa **Common Lisp**.

A FUNÇÃO:

```
(defun maior(s)
  (car (sort (cdr tabela)#'>)))
(format "O maior tamanho entre eles tinha: %s cm" (maior tabela))
```

EXPLICANDO:

- **(defun maior(s) ->** define, cria a função **maior** com o argumento **s**

```
| CENTÍMETROS |
|      13.97  |
|     14.351  |
|    13.0302  |
|    18.50009 |
```

- **tabela = CONVERTE[,2] =** pega a coluna e de **CONVERTE**. As colunas começam pelo índice 0, portanto, o índice 2 equivale à terceira coluna
- **(cdr tabela)->** **cdr** pega a lista **tabela** sem o primeiro elemento. A isto se denomina pegar a cauda da lista.

```
|      13.97  |
|     14.351  |
|    13.0302  |
|    18.50009 |
```

Ou seja:

- **(sort (cdr tabela) #'>)->**ordena (**sort**) a lista em ordem decrescente (**#'>**)

```
| 18.50009 |
| 14.351  |
| 13.97   |
| 13.0302 |
```

Ou seja:

- **(car(sort(cdr tabela)#'>))->** pega o primeiro elemento da lista, ou seja

Ou seja:

```
18.50009
```

- **(format "O maior tamanho entre eles tinha: %s cm (maior tabela))->**

Formata a resposta, ou seja, devolve a string, o texto:

O maior tamanho entre eles tinha:

%s vai ser substituído pelo retorno, o valor inferido pela função maior aplicado no argumento tabela -> (**maior tabela**), seguido da string cm.-> fecha

#+END_SRC -> fecha o bloco de código, programa.

A codificação textual da página no emacs-org-mod fica:

```

emacs@LUCIANO-PC
File Edit Options Buffers Tools Org Tbl Text Help
#++TITLE: BRASIL - 04/01/2016
#++AUTHOR:Correspondente da HominisPress: Tatiana O. Lima
#++OPTIONS: toc:4

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil
*** Atividade industrial mais fraca
*** Implantação do 'circuit breaker'
*** Yuan mais fraco
*** Venda de ações...
** China é o combustível da locomotiva mundial
Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz.

Quando isso acontece, o Brasil também sente essa desaceleração.

```

```

Quando isso acontece, o Brasil também sente essa desaceleração.

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.q1bimg.com/-3yFTcEC5EmwdXCRn1E1CsqdrDw=/s.q1bimg.com/jo/q1/f/original/2015/12/30/dolar-30.12\_v2.jpg

** Natalie Portman, mais uma vez, a atriz do ano!
file:natali.jpg

É interessante ver a atuação da Natalie Portman no filme Ataque dos Clones de 2002.
* OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.
file:debussy.jpg CLAIR THE LUNE

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

** RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!
#++BEGIN_HTML
<iframe width="640" height="360" src="https://www.youtube.com/embed/pc4Z0xpu2rs" frameborder="0" allowfullscreen></iframe>
#++END_HTML
■
** MÚSICA É TUDO DE BOM!
#++BEGIN_HTML
<audio controls><source src="Royals.mp3"></audio>
#++END_HTML

* VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO
#++NAME:CONVERTE
| MARCA | POLEGADAS | CENTÍMETROS |
| iPhone 6 | 5.5 | 13.97 |
| Galaxy S6 | 5.65 | 14.351 |
| lumia 920 | 5.13 | 13.0302 |
| Coolpad Halo | 7.2835 | 18.50009 |
#++TBLFM: $3=(2.54*$2)

#++BEGIN_SRC emacs-lisp :var tabela=CONVERTE[,2] :exports results :results value html
(require 'cl)
(defun maior(s)
  (car (sort (cdr tabela)#'>)))
(format "O maior tamanho entre eles chegou a: %s cm" (maior tabela))
#++END_SRC

* EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.

```


A página em html da notícia fica assim:

← → ↻ | file:///C:/jornal_taty/pa | 📖 ☆ | ☰ ...

BRASIL - 04/01/2016

Table of Contents

- 1 O dólar passa de R\$4 a R\$3,99 em apenas 10 minutos
- 2 VEJA A ESCALADA DO DÓLAR EM 2015
- 3 O UÇA É UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY
- 4 VEJA OS CELULARES QUE FIZERAM A CANÇÃO DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO
- 5 EM 2015 CAIU POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES
- 6 CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 300.000 INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOUR OF CODE" AO MESMO TEMPO

1 O dólar passa de R\$4 a R\$3,99 em apenas 10 minutos

Um dólar chegou a valer mais de sete reais e o momento de crise do mercado financeiro chegou ao Brasil. O dólar fechou em R\$3,99, o menor valor em mais de 10 meses.

1.1 Querê, mas, que você precisa saber sobre o futuro das nossas eleições e sua vida no Brasil

1.2.2 Implantação de 'Voto Inteligente'

1.2.3 Você não deve

1.2.4 Você já sabe

1.2.5 Você já sabe

2 VEJA A ESCALADA DO DÓLAR EM 2015

Disparada do dólar em 2015
O dólar chegou a valer mais de sete reais em 2015, o maior valor em mais de 10 meses.

Mês	Valor (R\$)
Jan	2,20
Fev	3,00
Mar	2,80
Abr	3,00
Mai	3,10
Jun	3,40
Jul	3,80
Ago	5,00
Sep	7,50
Out	6,50
Nov	5,50
Dez	3,99

1.1 Nicolás Perdomo, mais uma vez, o líder do ano!

3 O UÇA É UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.

4 VEJA OS CELULARES QUE FIZERAM A CANÇÃO DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO

Marca	Porcentagem	Comprimento
iPhone 6	1,1	11,17
Galaxy S6	1,1	11,17
Nexus 6	1,1	11,17
Catipal 10	1,1	11,17

5 EM 2015 CAIU POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.

6 CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 300.000 INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOUR OF CODE" AO MESMO TEMPO

Jornal - Computação e Redes - Taty - 04/01/2016
Diretor: Taty - 04/01/2016
Revisor: Taty - 04/01/2016
Editor: Taty

Detalhando melhor:

file:///C:/jornal_taty/paginaDinamica.html

BRASIL - 04/01/2016

Table of Contents

- [1. Dólar passa de R\\$4 e Bovespa cai quase 3%](#)
 - [1.1. Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil](#)
 - [1.1.1. Atividade industrial mais fraca](#)
 - [1.1.2. Implantação do 'circuit breaker'](#)
 - [1.1.3. Yuan mais fraco](#)
 - [1.1.4. Venda de ações](#)
 - [1.2. China é o combustível da locomotiva mundial](#)
- [2. VEJA A ESCALADA DO DÓLAR EM 2015](#)
 - [2.1. Natalie Portman, mais uma vez, a atriz do ano!](#)
- [3. OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.](#)
 - [3.1. RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!](#)
 - [3.2. MÚSICA É TUDO DE BOM!](#)
- [4. VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO](#)
- [5. EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.](#)
- [6. CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 200.000 INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOOR OF CODE" AO MESMO TEMPO](#)

1 Dólar passa de R\$4 e Bovespa cai quase 3%

Mercado reagiu a notícias ruins sobre a economia da China. A moeda norte-americana subiu 2,18%, a R\$ 4,0339 para venda.

1.1 Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil

1.1.1 Atividade industrial mais fraca

1.1.2 Implantação do 'circuit breaker'

1.1.3 Yuan mais fraco

1.1.4 Venda de ações

1.2 China é o combustível da locomotiva mundial

Agora que o mundo consome menos, a China tenta fazer com que os próprios chineses passem a consumir o que ela mesma produz. Quando isso acontece, o Brasil também sente essa desaceleração.

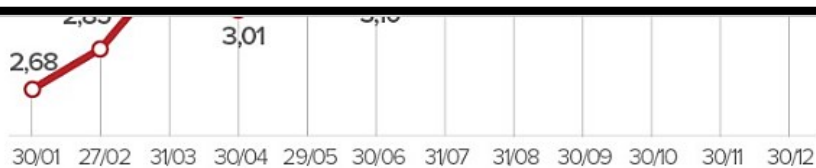
2 VEJA A ESCALADA DO DÓLAR EM 2015

Disparada do dólar em 2015

Veja o valor de fechamento em R\$ da última sessão de cada mês

Mês	Valor de Fechamento (R\$)
1	3,96
2	3,86
3	3,88
4	3,94

continua



Fonte: Reuters

G1.com.br

Infográfico elaborado em: 30/12/2015

2.1 Natalie Portman, mais uma vez, a atriz do ano!



É interessante ver a atuação da Natalie Portman no filme [Ataque dos Clones](#) de 2002.

3 OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.



[CLAIR THE LUNE](#)

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

Continua

[CLAIR THE LUNE](#)

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

3.1 RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!

Queen - Who Wants to Live Forever 

3.2 MÚSICA É TUDO DE BOM!



4 VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO

MARCA	POLEGADAS	CENTÍMETROS
iPhone 6	5.5	13.97
Galaxy S6	5.65	14.351
lumia 920	5.13	13.0302
Coolpad Halo	7.2835	18.50009

O maior tamanho entre eles chegou a: 18.50009 cm

5 EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.



Continua

5 EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.



6 CODE, A LINGUAGEM QUE ATINGIU 60 PAÍSES DO MUNDO E MAIS DE 200.000 INSTITUIÇÕES E ENSINO PARA APRENDEREM A "HOUR OF CODE" AO MESMO TEMPO

Veja um jogo criado em 1 minuto e 30 segundos em CODE



Author: Correspondente da HominisPress: Tatiana O. Lima
 Created: 2016-01-14 qui 01:44
[Emacs](#) 24.5.1 ([Org](#) mode 8.2.10)
[Validate](#)

9.4 PUBLICANDO A PÁGINA NA NUVEM

Não adianta nada fazer uma página e deixá-la no computador. Para tanto, é preciso subi-la para a hospedagem na nuvem.

Existem vários processos e programas para se fazer isto.

O paradigma utilizado nesta dissertação é que se consiga fazer isto sempre usando qualquer dispositivo que tiver em mãos, através de comandos textuais, os quais podem ser lidos por todos.

Os sistemas operacionais **Linux** e **Mac** já possuem recursos para acessar a nuvem textualmente via um terminal especial com protocolo **SSH**⁶¹ (Anexo 2), ou seja: usando um **Terminal SSH**. **Windows** e **Android** necessitam de suporte de aplicativos.

Normalmente se diz que trabalhar com terminais **SSH**, acesso à nuvem e servidores é coisa para **Linux** e **Mac**. Será visto, a seguir, que isto também é fácil, não simples, de fazer também em **Windows**.

9.4.1 Transferindo Arquivos Para A Nuvem, Para A Hospedagem, Utilizando O Windows

Depois que se sabe como, tudo na vida fica mais fácil. Assim é o ato de abrir um terminal SSH no Windows e editar, programar direto na nuvem e transferir arquivos para a mesma. Por mais que os programadores Linux e Mac digam que Windows não serve para isto, ou que é difícil, não é verdade.

A plataforma, o sistema operacional Windows não possui incorporado um terminal **SSH** para comunicação com a nuvem. As plataformas MAC e LINUX, por sua vez, já possuem tal recurso. Mas isto não é impedimento a que se faça isto no Windows.

No Windows pode-se utilizar, entre outras opções mais complexas, um software para fazer esta comunicação: o **PuTTY**. Dois arquivos executáveis, dois programas deverão ser baixados:

- Um para programar direto na nuvem: o **PuTTY.exe**
- Outro para transferir arquivos do desktop para a nuvem (upload) ou da nuvem para o desktop (download): o **pscp.exe**

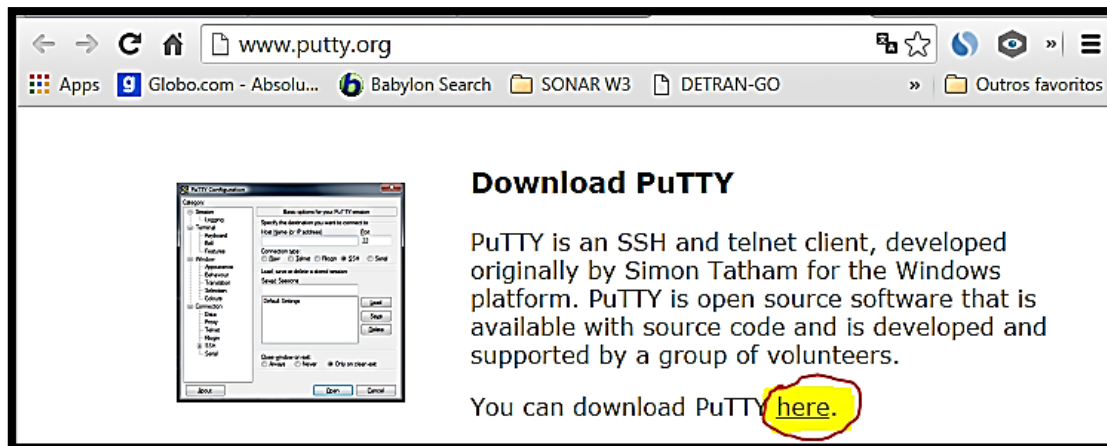
9.4.2 Instalando Os Programas Do Putty

O **PUTTY** não precisa de instalação. Basta baixar os arquivos **.exe** desejados (**PuTTY.exe** e **pscp.exe**) através do endereço www.PuTTY.org e, posteriormente, utilizá-lo, como será visto nos próximos itens.

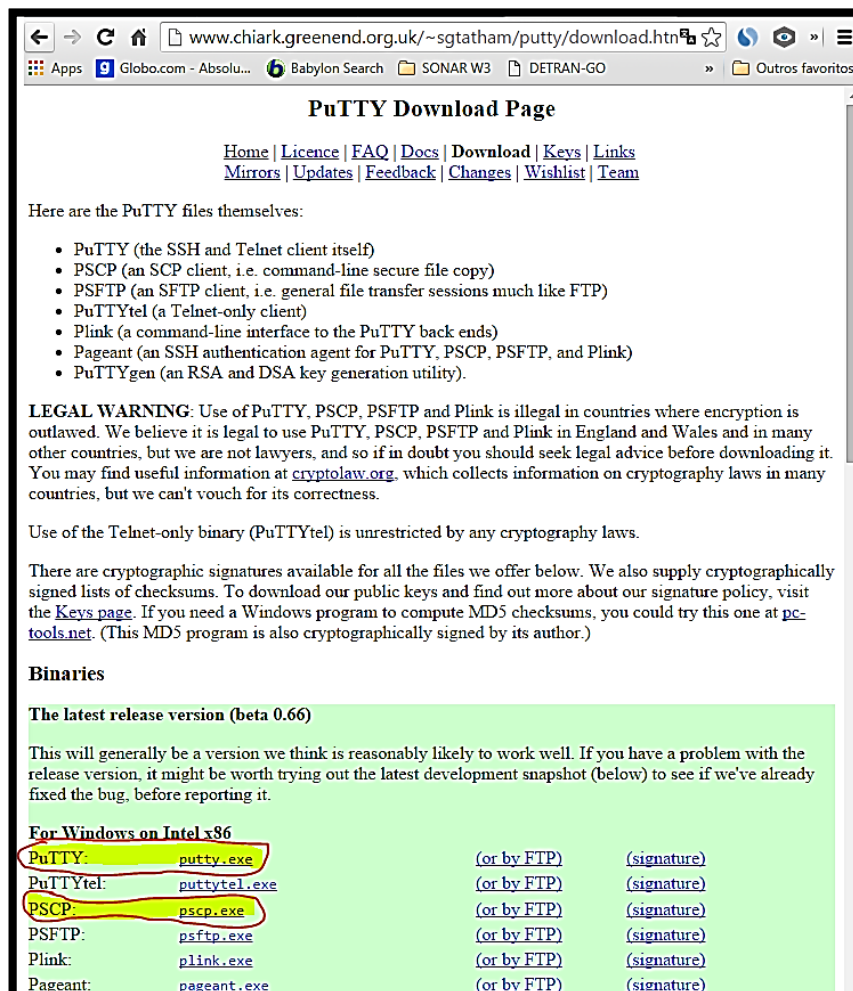
BAIXANDO OS PROGRAMAS DO PuTTY

⁶¹ **SSH – Secure Shell** – É um protocolo de comunicação criptografada (encriptada) que permite comunicação com outras redes e serviços para operar com segurança em redes inseguras e/ ou desconhecidas. https://en.wikipedia.org/wiki/Secure_Shell

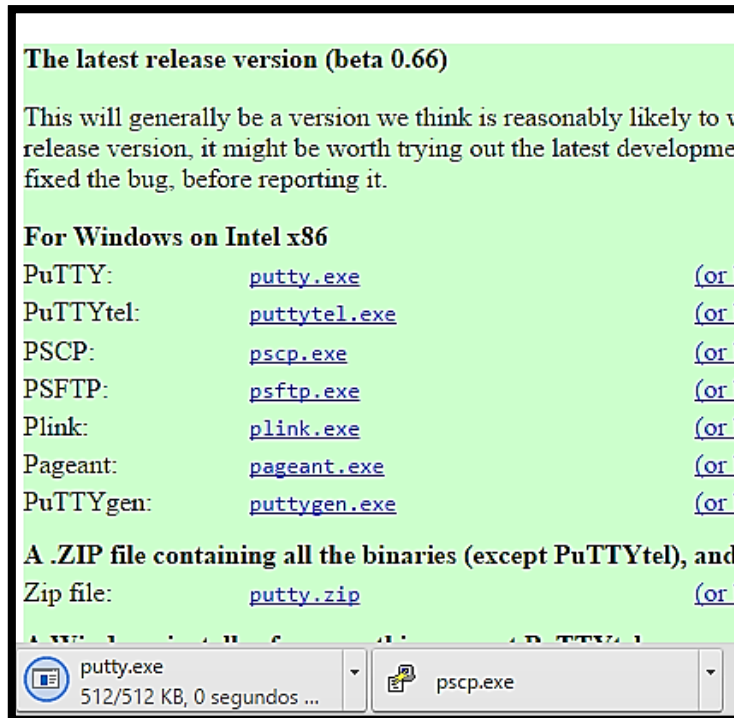
Conforme figura seguinte, clique em **here** para entrar na página de download do **PuTTY**



A página de download do **PuTTY** é aberta. Baixe, faça o download, dos arquivos **putty.exe** (que permite abrir um terminal **SSH** no Windows) e o **pscp.exe** (que permite cópias seguras, criptografadas, do Windows para a nuvem e da nuvem para o Windows).



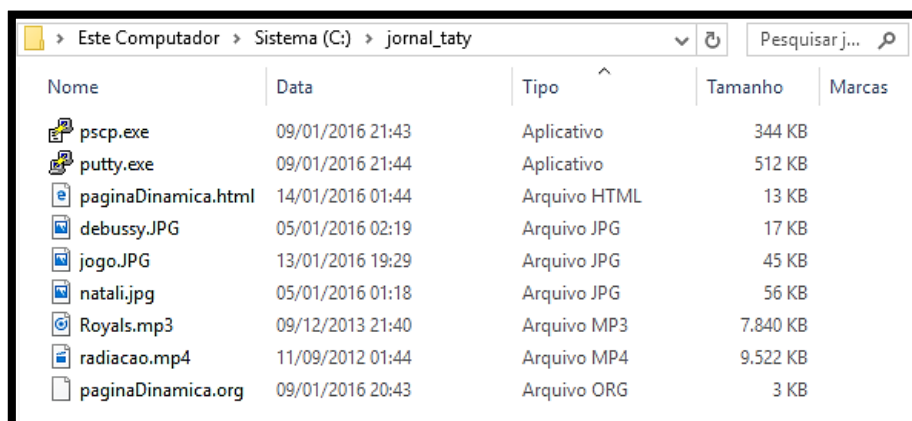
No Windows 10, normalmente os arquivos vão ser baixados para a pasta **Downloads**. No Windows 7, normalmente se pergunta onde se deseja salvar.



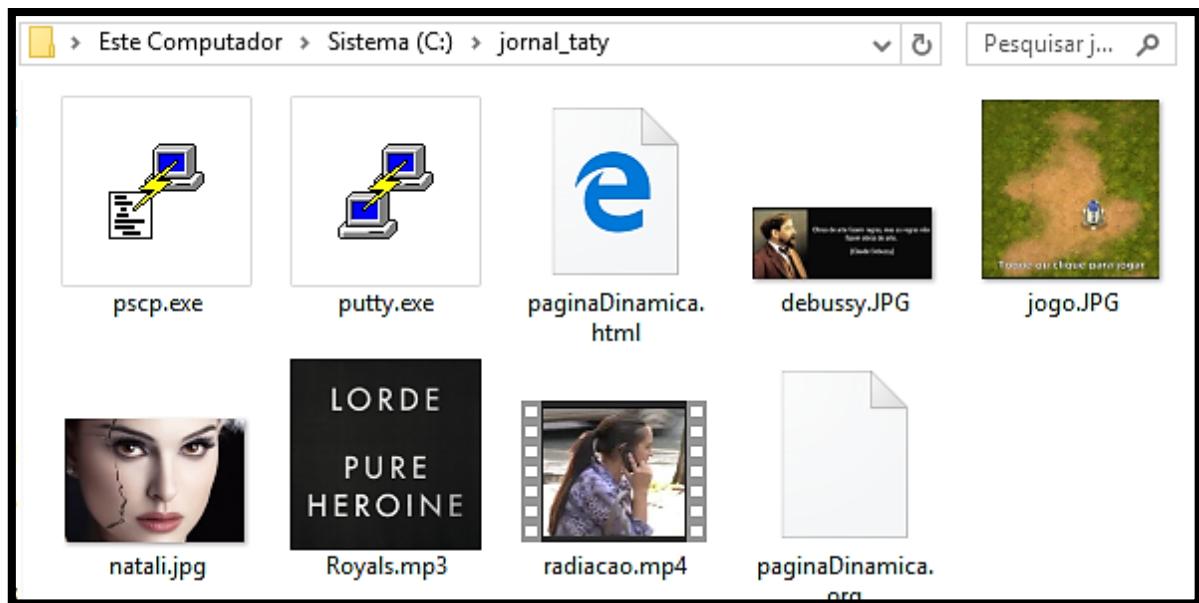
Para organizar a criação das páginas e transferência correta de dados, exigida pelo **PuTTY**, coloque os arquivos baixados na mesma página (em **c:\jornal_taty**) dos arquivos que serão subidos (uploaded) para a nuvem no diretório **public_html** da hospedagem.

No caso do Windows 10 onde os arquivos são baixados na página **Downloads**, depois de baixados, transfira-os ou copie os mesmos para a pasta **c:\jornal_taty**. O **ANEXO EMACS_CONFIGURANDO** explica com detalhes como transferir arquivos de uma pasta para outra.

A pasta **c:\jornal_taty**, depois do download e transferência, fica assim:



ou



Os arquivos estão prontos para subir para a Hospedagem.

Para transferir os arquivos de forma segura para a hospedagem, ou seja, copiar arquivos da máquina pessoal para a máquina remota na hospedagem, na nuvem, utilizar-se á o protocolo **SCP**⁶² que são mais seguros do que o protocolo **FTP**⁶³, onde os dados são criptografados. Esta transferência é facilmente realizável através de um terminal **SSH**, no caso do Windows, disponibilizada pelo programa **PuTTY**, no caso, o **pscp.exe** (**scp** do **PuTTY**).

9.4.3 Transferindo Finalmente Os Arquivos Da Página Dinâmica Criada Para O Diretório Public_Html

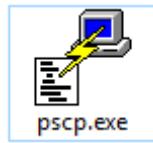
A página que rodará na nuvem precisa ter todos os arquivos utilizados por ela, a saber:

- 1- pagDin3.html
- 2- debussy.jpg
- 3- natali.jpg
- 4- Royals.mp3
- 5- radiacao.mp4
- 6- jogo.jpg

⁶² **SCP** (**S**ecure **C**opy **P**rotocol) - <http://udgwebdev.com/dicas-de-terminal-copiando-arquivos-via-scp/> , https://en.wikipedia.org/wiki/Secure_copy

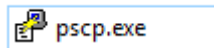
⁶³ **FTP** (**F**ile **T**ransfer **P**rotocol) - https://pt.wikipedia.org/wiki/File_Transfer_Protocol

9.4.3.1 Usando o terminal com o PSCP do Putty para transferir os arquivos



O programa


ou

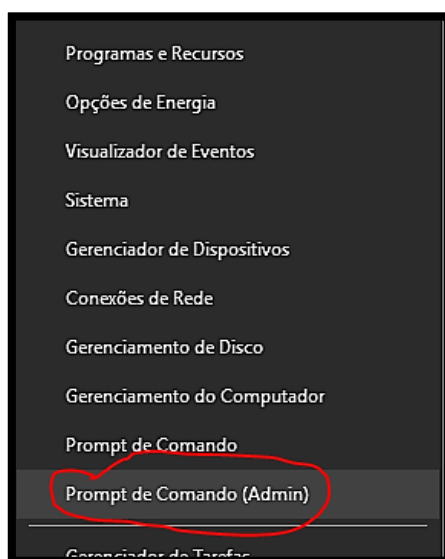


não pode ser executado

diretamente do gerenciador de programas do Windows. Tem-se que executá-lo do terminal do Windows: o **Prompt de Comando** (a janelinha preta com comandos textuais).

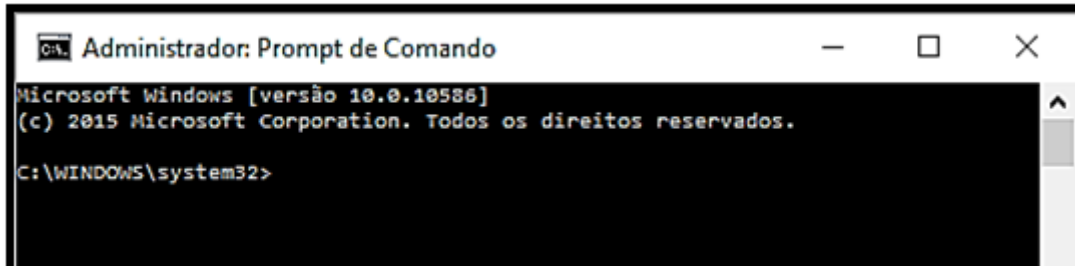
Assim, para abrir o **Prompt de Comando**, no caso do Windows 10, por exemplo, basta colocar o cursor do mouse em cima do botão de iniciar do Windows

10  e clicar com o **botão direito** do mouse. Um menu de opções é aberto com a opção **Prompt de Comando (Admin)**. Clique nesta opção. Você precisará dos privilégios de Administrador (**Admin**)



Clique no executável ou que está na página: **c:\jornal_taty**. A seguinte mensagem é aberta. Clique em **Executar**. Vai ser solicitada a confirmação. Confirme.

Feito isto, o **Prompt de Comando** com privilégios de administrador é aberto:



```

Administrator: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>

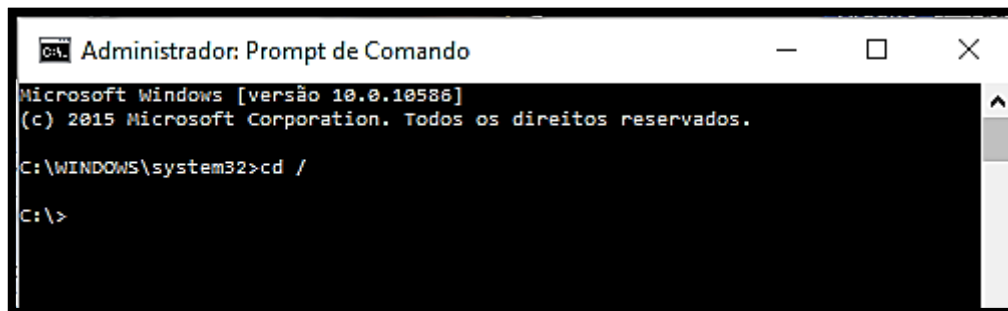
```

Inicialmente o mesmo é aberto no diretório `c:\windows\system32` do Windows.

O programa `pscp.exe` está no diretório `c:\jornal_taty`. Assim, precisa-se ir para este diretório.

Para mudar de diretório, o comando é `cd` (**c**hange **d**irectory). Este comando deve ser seguido de um complemento para decidir o que se quer fazer.

Para mudar para a raiz (`c:`), no Windows, o comando é `cd /` (`cd` espaço `/`).



```

Administrator: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

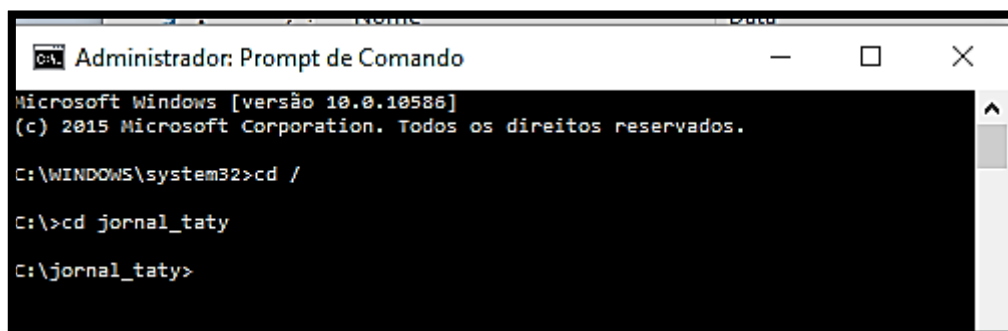
C:\WINDOWS\system32>cd /

C:\>

```

Chegando na raiz, `c:`, basta agora ir para o diretório `jornal_taty`.

Para tanto, digite `cd jornal_taty` e tecla **ENTER**.



```

Administrator: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd /

C:\>cd jornal_taty

C:\jornal_taty>

```

O próximo passo é executar o programa `pscp.exe`. Digite o mesmo e tecla **ENTER**.

```

Administrator: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd /

C:\>cd jornal_taty

C:\jornal_taty>pscp.exe

```

A janela do **Prompt de Comando** permanece, mas, agora, rodando o **pscp.exe** nela. Um menu de opções do **pscp** é aberto.

```

Administrator: Prompt de Comando

Options:
  -V          print version information and exit
  -pgpfp     print PGP key fingerprints and exit
  -p         preserve file attributes
  -q         quiet, don't show statistics
  -r         copy directories recursively
  -v         show verbose messages
  -load sessname Load settings from saved session
  -P port    connect to specified port
  -l user    connect with specified username
  -pw passw  login with specified password
  -1 -2     force use of particular SSH protocol version
  -4 -6     force use of IPv4 or IPv6
  -C         enable compression
  -i key     private key file for user authentication
  -noagent   disable use of Pageant
  -agent     enable use of Pageant
  -hostkey aa:bb:cc:...
             manually specify a host key (may be repeated)
  -batch     disable all interactive prompts
  -unsafe    allow server-side wildcards (DANGEROUS)
  -sftp     force use of SFTP protocol
  -scp      force use of SCP protocol

C:\jornal_taty>

```

Feito isto, tudo está pronto para que se possa fazer a transferência dos arquivos para a nuvem. Mas, antes, precisa-se saber 4(quatro) coisas:

- 1- O **Host Name – domínio** – é o nome do domínio utilizado para contratar sua hospedagem na Web (no caso deste exemplo, é: **coronasdkgames.com.br**)
- 2- Conhecer o **login** da sua hospedagem
- 3- Conhecer o **password** a **senha** de administrador da sua hospedagem.
- 4- A **porta** que a sua hospedagem libera para comunicação **SSH**. No caso conforme já mostrado para a HostGator, a porta é a **2222**.

Estas 4 (quatro) informações podem ser obtidas pela leitura do **ANEXO – DADOS DA HOSPEDAGEM NA NUVEM**, o qual mostra com detalhes como obter estas 4 (quatro) informações fundamentais e essenciais para transmissão de dados

entre o computador e a nuvem, bem como para a programação direta na nuvem que é o próximo item deste capítulo.

No caso deste exemplo, estas informações são:

- 1- HostName – domínio = coronasdkgames.com.br
- 2- Login – coron759
- 3- Password – senha – xxxxxx
- 4- Porta SSH - 2222

9.4.3.2 O comando de transferência de arquivos entre o Windows e a nuvem na hospedagem desejada

Genérica:

```
pscp -P portaSSH -scp arquivo login@domínio:~public_html/
```

No exemplo dado:

```
pscp -P 2222 -scp arquivo coron759@coronasdkgames.com:~public_html/
```

Recordando, os arquivos são:

- 1- paginaDinamica.html
- 2- debussy.jpg
- 3- natali.jpg
- 4- Royals.mp3
- 5- radiacao.mp4

Os comandos para transferência serão

- 1- pscp -P 2222 -scp paginaDinamica.html
coron759@coronasdkgames.com:~/public_html/
- 2- pscp -P 2222 -scp debussy.jpg
coron759@coronasdkgames.com:~/public_html/
- 3- pscp -P 2222 -scp natali.jpg
coron759@coronasdkgames.com:~/public_html/
- 4- pscp -P 2222 -scp Royals.mp3
coron759@coronasdkgames.com:~/public_html/

- 5- pscp -P 2222 -scp radiacao.mp4
coron759@coronasdkgames.com:~/public_html/
- 6- pscp -P 2222 -scp jogo.jpg
coron759@coronasdkgames.com:~/public_html/

Mostrando dois exemplos de caso de transferência:

- 1- paginaDinamica.html

```
C:\jornal_taty>pscp -P 2222 -scp paginaDinamica.html coron759@coronasdkgames.com:~/public_html/
coron759@coronasdkgames.com's password:
```

O sistema para solicitando o password, entre e tecele ENTER. Não aparecerá nada enquanto digita, nem o cursor do mouse se moverá para frente. Se tiver digitado correto, a transmissão começará. Se a transferência aparecer com 100%, sucesso.

```
C:\jornal_taty>pscp -P 2222 -scp paginaDinamica.html coron759@coronasdkgames.com:~/public_html/
coron759@coronasdkgames.com's password:
paginaDinamica.html | 11 kB | 11.6 kB/s | ETA: 00:00:00 | 100%
```

Se digitar a senha incorreta, o sistema devolve a mensagem de Access denied (acesso proibido) e solicita novamente o password.

- 2- Radiacao.mp4

```
C:\jornal_taty>pscp -P 2222 -scp radiacao.mp4 coron759@coronasdkgames.com:~/public_html/
coron759@coronasdkgames.com's password:
radiacao.mp4 | 660 kB | 132.0 kB/s | ETA: 00:01:07 | 6%
```

Como este arquivo é grande, demora a subir, dependendo o tempo de transmissão da velocidade de sua internet.

Completando, sucesso!

```
C:\jornal_taty>pscp -P 2222 -scp radiacao.mp4 coron759@coronasdkgames.com:~/public_html/
coron759@coronasdkgames.com's password:
radiacao.mp4 | 9521 kB | 113.4 kB/s | ETA: 00:00:00 | 100%
C:\jornal_taty>
```

Transmitidos todos os arquivos, basta entrar em um browser, seja do desktop, celular, Smart TV, outros, a página da notícia deverá abrir, como mostra o trecho inicial da mesma, a seguir:



9.5 CRIANDO UMA PÁGINA MAIS DINÂMICA INSERINDO PROGRAMAÇÃO E COMPILAÇÃO JUST IN TIME⁶⁴.

Para exemplificar, será utilizada uma página no **Emacs** com programação em **E-Lisp**.

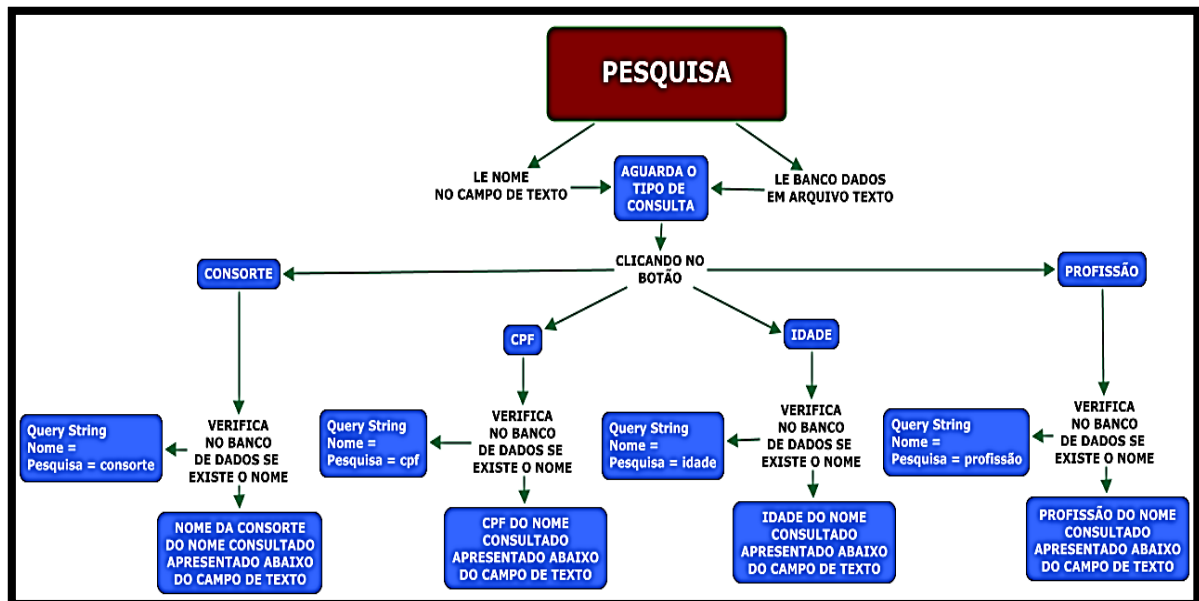
A mesma vai fazer um text mining em um banco de dados textual, a partir de um nome digitado pelo usuário em um campo de texto da interface na página.

As pesquisas em cima do nome serão 4 (quatro):

- 1- Consorte
- 2- CPF
- 3- Idade
- 4- Profissão

A pesquisa é feita conforme o mapa a seguir

⁶⁴ **Just In Tima:** O BROWSE VAI GERANDO A PÁGINA À MEDIDA COM QUE FOR LENDO A MESMA, MODIFICANDO CONFORME INTERAÇÃO COM O USUÁRIO E PROGRAMAÇÃO EXISTENTE



O formato do arquivo texto é o seguinte:

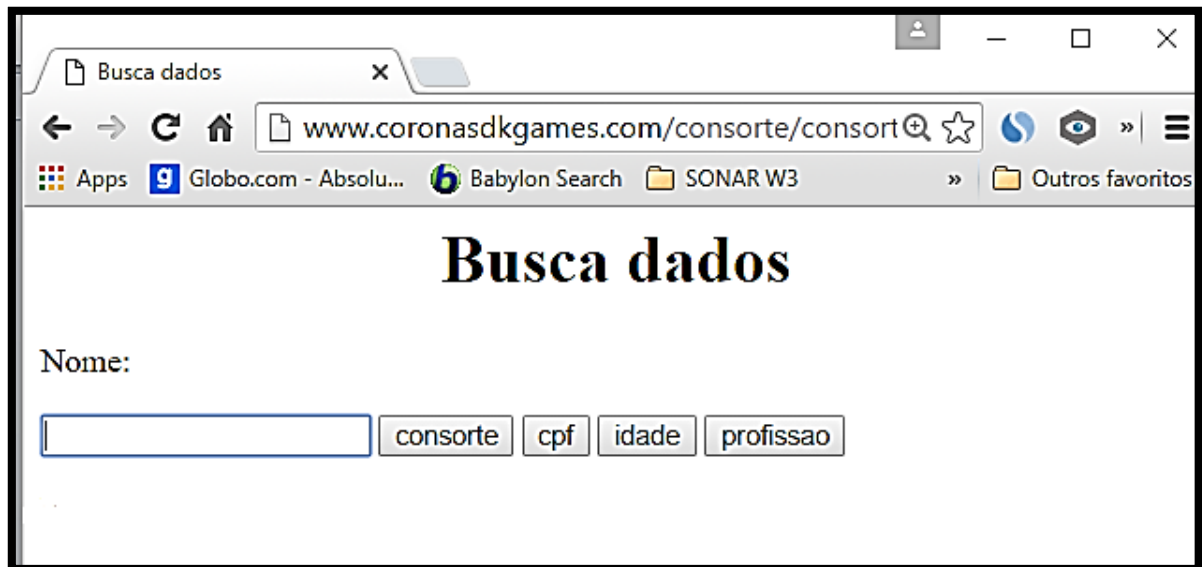
- Um campo por linha, ou seja:

```

cadastros.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
nome1 consorte1 CPF1 idade1 profissão1
nome2 consorte2 CPF2 idade2 profissão2
...
nomeN consorteN CPFN idadeN profissãoN
  
```

Lido o nome no campo de texto e clicado no botão da pesquisa, o resultado é apresentado abaixo do campo de texto.

Assim, a página terá que ter um campo de texto e botões para fazer a pesquisa



Para testes, o banco de dados inicial, que está na nuvem, dentro de: www.conronasdkgames.com/public_html/consorte/cadastro.txt, será:

```

cadastros.txt - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
Vitor Solteiro 09163317680 21 Estudante
Luciano Sandra CPF_Luciano IDADE_Luciano Professor|
Tatiana Solteira CPF_Tatiana Idade_Tatiana Jornalista
  
```

9.6 O PROGRAMA: A PÁGINA DINÂMICA PROGRAMADA

'Como toda a filosofia deste trabalho, deve-se utilizar o recurso já existente na hospedagem, de tal forma que não se tenha barreiras nas implementações e publicações das páginas. Assim como o emacs é uma ferramenta já nativa das hospedagens, a linguagem de programação **ELisp** também é disponibilizada para se programar em alto nível no **Emacs**. O Anexo 3 mostra com detalhes esta linguagem com bastante exemplos. Mesmo assim, o código vai ser comentado com todos os detalhes para que o entendimento não seja comprometido.

9.6.1 Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional **Windows** E Org-Mod.



Para trabalhar na nuvem direto com comandos textuais, o jornalista precisará de um editor na nuvem. Caso a página seja dinâmica, necessite programação, interação com o leitor, banco de dados, outros, o editor deverá possuir recursos para programação multiparadigma (na linguagem à escolha do desenvolvedor). Além disto, este editor e recursos já devem estar presentes na nuvem, na hospedagem do domínio, sem que as **jailShell**⁶⁵ do servidor **Apache**⁶⁶ e da hospedagem bloqueiem as ações desejadas.

Neste caso, a única solução é **emacs (Anexo 10)**.

9.6.1.1 Emacs na nuvem com Windows:

Para trabalhar na nuvem, tem-se que ter um editor que seja reconhecido por todas as plataformas e hospedagens. Para tanto, utiliza-se nas hospedagens da nuvem um emacs mais antigo, padrão: o **emacs-23.1**⁶⁷ (não se pode usar direto outra versão), totalmente estável a mais de 6 anos e sem bugs conhecidos.

Para trabalhar no Windows, será utilizado o programa **PuTTY** mostrado em detalhes nos dois anexos: **TERMINAL SSH NO WINDOWS COM PuTTY** e **DADOS DA HOSPEDAGEM NA NUVEM**.

Estes dois anexos são fundamentais não só para entender a ferramenta **PuTTY**, para abrir um terminal **SSH** no Windows, mas, também, para mostrar como

⁶⁵ **Jailshell** - <https://www.servint.net/university/article/the-tech-bench-jailshell-in-cpanel/> -É um nível do shell (SSH) que limita o acesso do usuário (como uma prisão – **jail cell**) à estrutura de diretório da respectiva hospedagem.

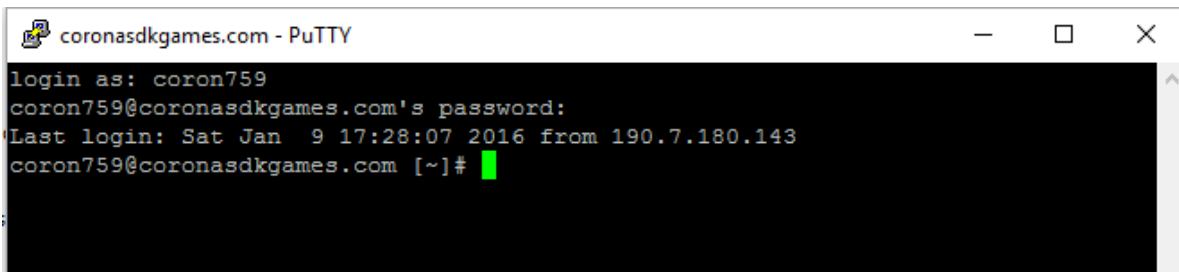
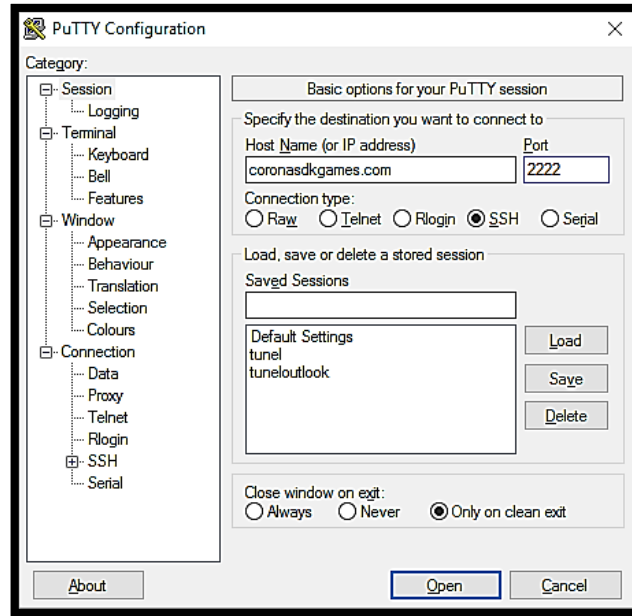
⁶⁶ **APACHE** – Servidor disponibilizado nas hospedagens para uso geral, em hospedagens não proprietárias (onde o usuário pode construir o próprio servidor) https://en.wikipedia.org/wiki/Apache_HTTP_Server

^e **emacs-23.1** - <http://lists.gnu.org/archive/html/info-gnu-emacs/2009-07/msg00000.html>

conhecer os dados de **login**, **password** e **porta SSH** da hospedagem. Sem estes dados não tem como se trabalhar direto na nuvem.

Assim:

- 1- Abra um terminal SSH clicando no executável **putty.exe** (conforme anexo citado) que está na pasta **c:\jornal_taty** do exemplo desta dissertação.



OK! O terminal está pronto para ser utilizado.

- 2- Entre no **emacs** que está na nuvem, versão: **emacs-23.1**
 Digite **emacs-23.1** e aperte a tecla **ENTER**. O emacs será aberto.

```

coronasdkgames.com - PuTTY
File Edit Options Buffers Tools Help
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Get help          C-h (Hold down CTRL and press h)
Emacs manual      C-h r      Browse manuals    C-h i
Emacs tutorial    C-h t      Undo changes      C-x u
Buy manuals       C-h C-m    Exit Emacs        C-x C-c
Activate menubar M-`
(`C-' means use the CTRL key. `M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)
Useful tasks:
Visit New File           Open Home Directory
Customize Startup        Open *scratch* buffer

GNU Emacs 23.1.1 (x86_64-redhat-linux-gnu, GTK+ Version 2.24.23)
of 2015-02-19 on c6b8.bsys.dev.centos.org
Copyright (C) 2009 Free Software Foundation, Inc.

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software--Free as in Freedom--so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-o for information on getting the latest version.
-UUU:%%--F1 *GNU Emacs* Top L1 (Fundamental)-----
For information about GNU Emacs and the GNU system, type C-h C-a.

```

Digite C-x C-f para abrir a linha de comando

```
Find file: ~/|
```

Entre na página **public_html**(toda página tem que estar neste diretório) e crie um arquivo com o nome, por exemplo: **paginaNuvem.org** e aperte a tecla **ENTER**.

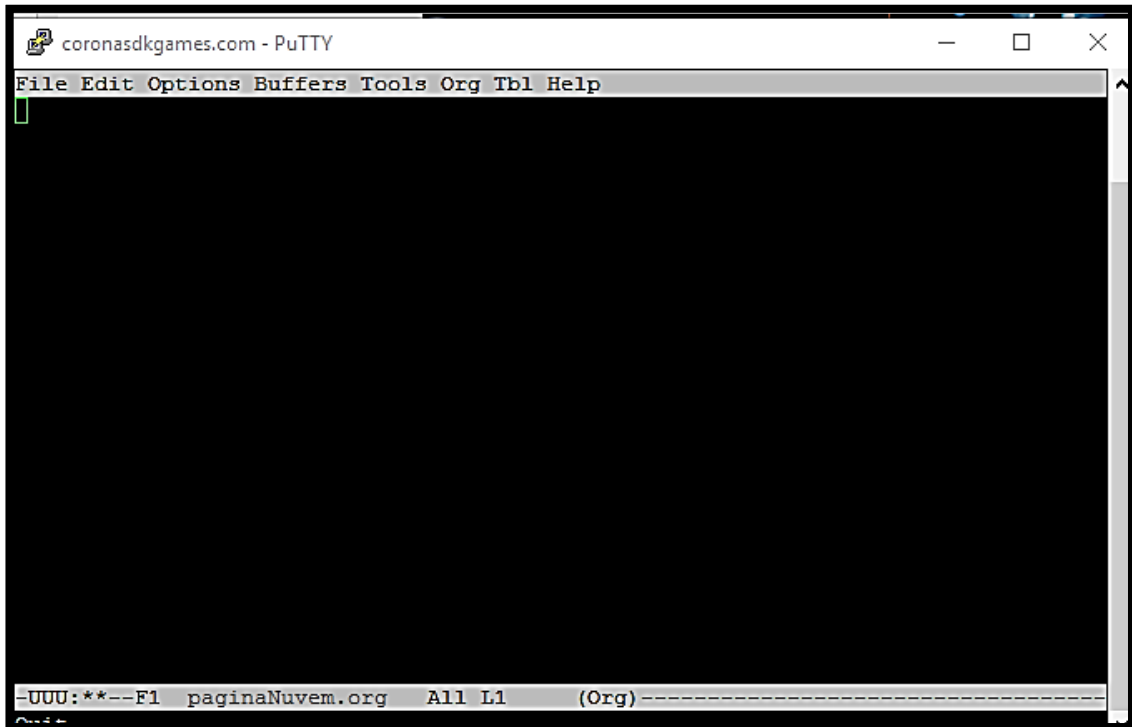
O comando completo é: **public_html/paginaNuvem.org**.

```
Find file: ~/public_html/paginaNuvem.org|
```

Digite **C-x C-s** para salvar a página.

Pronto, a página **.org** está criada!

Como já mencionado, basta salvar um arquivo no **emacs** com a extensão **.org** que o mesmo já reconhece-o como um arquivo em **org-mod**.



- 3- Feito isto, a sequência de passos é a mesma feita na programação **org-mod** realizada anteriormente para desktop.
- 4- Como os arquivos já estão na nuvem, foram transferidos por **scp**, basta digitar o mesmo programa e gerar a página **.html**.
Caso esteja fazendo uma nova página, com novos arquivos, ou acrescentando outros à página existente, basta subir (fazer o upload) dos arquivos novos por **scp**, no caso do Windows, utilizando o programa **pscp.exe** do **PuTTY**, conforme já foi explicado com detalhes neste mesmo capítulo.
- 5- Para obter a mesma página que a do exemplo em desktop, basta digitar na nuvem, no **emacs-23.1**, o mesmo conteúdo, conforme a seguir.

```

coronasdkgames.com - PuTTY
File Edit Options Buffers Tools Org Tbl Help
#+TITLE: BRASIL - 04/01/2016
#+AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:4

* Dólar passa de R$4 e Bovespa cai quase 3%
Mercado reagiu a notícias ruins sobre a economia da China.
A moeda norte-americana subiu 2,18%, a R$ 4,0339 para venda.
** Quatro coisas que você precisa saber sobre o tombo dos mercados chineses e seu efeito no Brasil...
** China é o combustível da locomotiva mundial...

* VEJA A ESCALADA DO DÓLAR EM 2015
http://s2.glbimg.com/~3yfTcECSEmwdXCRn1ElCsqrdw=/s.glbimg.com/jo/g1/f/original/2015/12/30/dolar-30.12\_v2.jpg

** Natalie Portman, mais uma vez, a atriz do ano!...
* OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO JÁ OUVIU, COMPOSTA POR CLAUDE DEBUSSY.
file:debussy.jpg CLAIR THE LUNE

TUDO DE BOM PARA COMEÇAR 2016 COM MAIS TRANQUILIDADE E PAZ NO CORAÇÃO!

** RELEMBRAR QUE OS ANOS PASSAM, MAS AS COISAS BOAS FICAM REGISTRADAS PARA SEMPRE!
#+BEGIN_HTML
<iframe width="640" height="360" src="https://www.youtube.com/embed/pC4ZOxpu2rs" frameborder="0" allowfullscreen></iframe>
#+END_HTML

** MÚSICA É TUDO DE BOM!
#+BEGIN_HTML
<audio controls><source src="Royals.mp3"></audio>
#+END_HTML

* VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS ADOLESCENTES EM 2015 E O RESPECTIVO TAMANHO
#+NAME:CONVERTE
| MARCA | POLEGADAS | CENTÍMETROS |
| iPhone 6 | 5.5 | 13.97 |
| Galaxy S6 | 5.65 | 14.351 |
| Lumia 920 | 5.13 | 13.0302 |
| Coolpad Halo | 7.2835 | 18.50009 |
#+TBLFM: $3=(2.54*$2)

#+BEGIN_SRC emacs-lisp :var tabela=CONVERTE[,2] :exports results :results value html
(require 'cl)
(defun maior(s)
  (car (sort (cdr tabela) #'>)))
(format "O maior tamanho entre eles chegou a: %s cm" (maior tabela))
#+END_SRC

* EM 2015 CAI POR TERRA A CENSURA SOBRE OS PROBLEMAS CAUSADOS PELOS CELULARES.
#+BEGIN_HTML
<video controls><source src="radiacao.mp4"></video>
#+END_HTML

-
UU1 (DOS)**--F1 paginaNuvem.org Top L65 (Org)-----

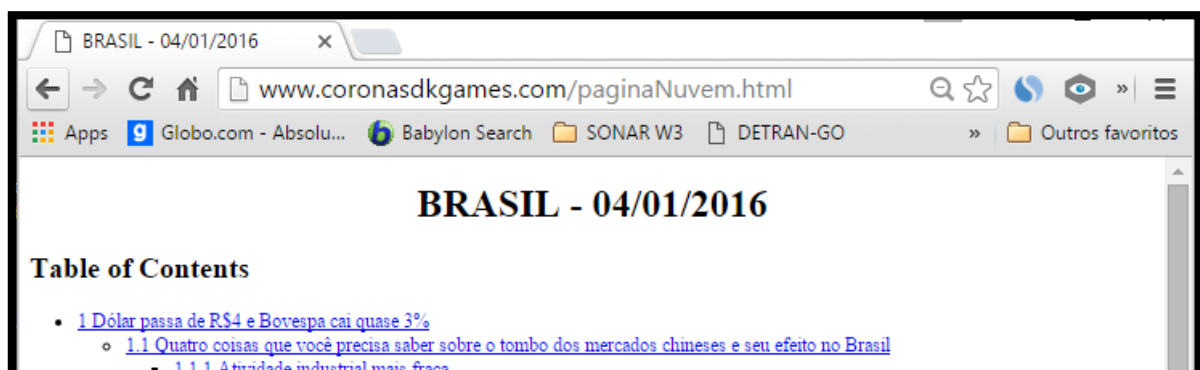
```

6- Salve a página como Html. A diferença é que o comando na nuvem muda.

Para salvar a página, a sequencia de comandos é: **C-c C-e h**

7- Para ver a página, digite no Browser o endereço da mesma, ou seja:

<http://www.coronasdkgames.com/paginaNuvem.html>



9.6.2 Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema **Linux**, **Ubuntu**, E Org-Mod.



O terminal de comandos do Linux (**CLI - Command Line Interface**) é onde se digita comandos para dizer ao computador o que ele deve fazer.

Esse modo é extremamente rápido, versátil e poderoso, tornando-se ideal para tarefas como atualizar uma página na Web.

Para esta dissertação, será utilizado o Linux mais popular atual (01/2016), o **Ubuntu 15.10**.

Ao iniciar o Sistema Operacional e efetuar o login, ter-se-á a seguinte tela:

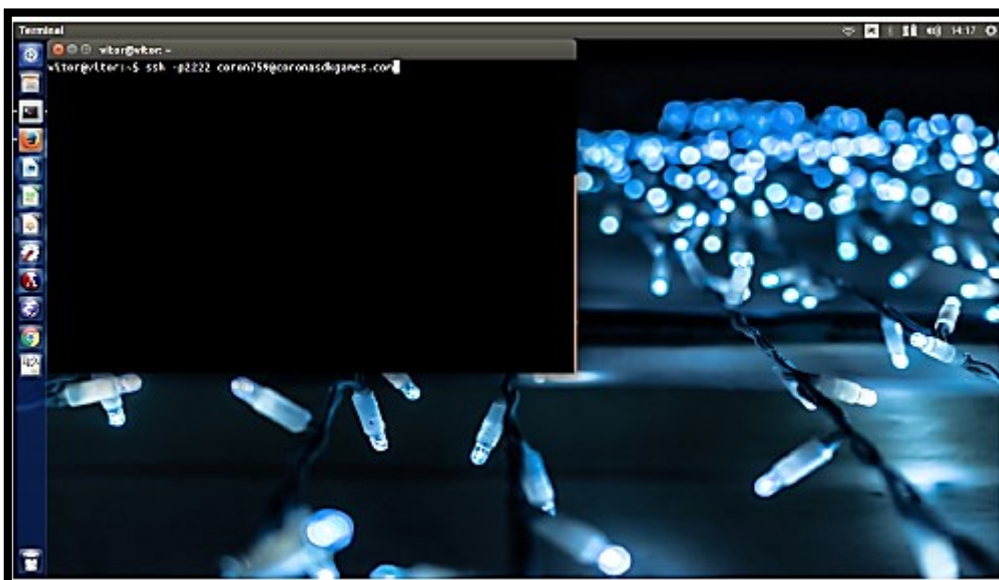


- 1- Para abrir um terminal, utilize o atalho de teclado **C-M-t** (Pressione ao mesmo tempo a tecla **Ctrl**, a tecla **Alt** e a tecla **t**) ou clique no ícone do Terminal, localizado na barra de ferramentas.



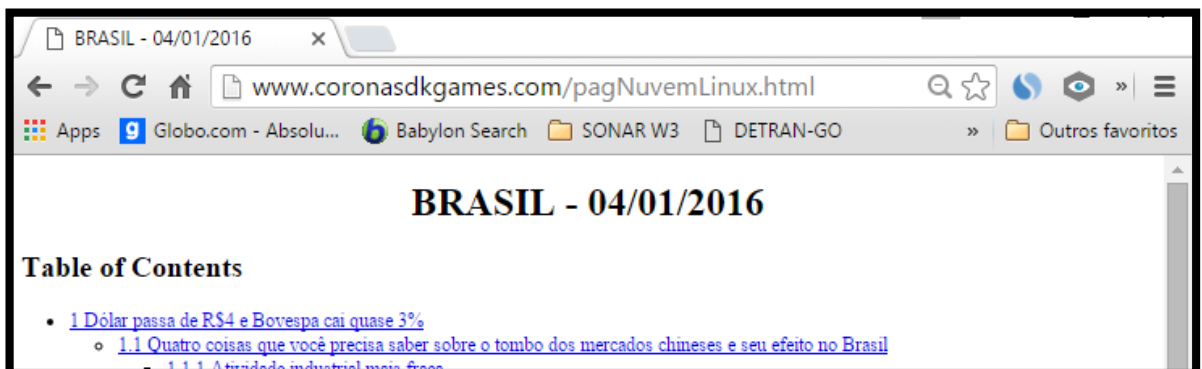
- 2- Para acessar sua máquina na nuvem, utilize o protocolo **SSH**, já nativo no Linux, ao contrário do Windows, já visto.
- 3- Para abrir no **terminal do Linux** um **terminal SSH**, digite no terminal aberto:

ssh -p222 coron759@coronasdkgames.com e aperte a Tecla **ENTER**.

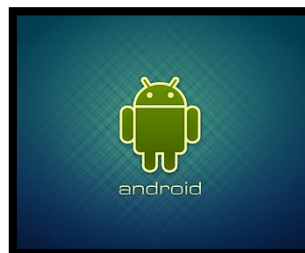


- 4- Feito isto, sua senha da hospedagem é solicitada (a mesma do exemplo do Windows). Digite a mesma e aperte a tecla **ENTER**.
- 5- Entre no diretório **public_html** digitando **cd public_html**, da mesma forma que foi feito no Windows. Lembre-se que toda página html gerada deverá estar na página **public_html**.

- 6- Digite **emacs-23.1 pagNuvemLinux.org** e aperte a tecla **ENTER** e digite **C-x C-s** para salvar.
- 7- Poder-se ia ter feito, também, como no Windows, ou seja, abre o emacs, depois digita-se **C-x C-f**, abre-se alinha de comando, digita-se o nome do arquivo, aperta a tecla **ENTER** e digita-se **C-x C-s** para salvar o arquivo.
- 8- Como pode-se perceber, daqui para frente é tudo igual ao que foi feito no Windows. Digita-se o mesmo arquivo, salva e exporta como **html** com o comando **C-c C-e h**.
- 9- Depois é só abrir a página no browse:



9.6.3 Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional **Android** E Org-Mod.



Como mostrado nos exemplos de programação, edição e criação de páginas em Windows e Linux na nuvem, percebe-se que o único problema, e que diferenciam as ações, na nuvem, é como acessar a hospedagem utilizando um terminal seguro **SSH**. Feito o acesso, entrando no **emacs-23.1** da hospedagem, o resto é tudo igual.

Assim, o único problema é encontrar um sistema para o Android que faz o que o **PuTTY** faz para o Windows.

9.6.3.1 Acessando o servidor via SSH em dispositivos Android

Assim como foi utilizado o **PuTTY** Windows, no Android será utilizado um programa para abrir um terminal SSH entre o computador pessoal e a hospedagem na nuvem.

Nos sistemas Android, tanto para tablets quanto para celulares, com conexão 3G ou Wi-Fi, existem vários programas gratuitos para serem utilizados como clientes em uma conexão SSH.

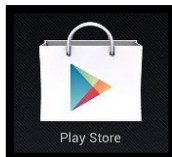
Utilize a frase "**SSH client**" no sistema de pesquisa da Play Store (Central Google de Downloads Android) para escolher qual aplicação desejará instalar. Dentre as várias aplicações disponíveis, destacam-se as duas mais votadas: **Server Auditor** e **JuiceSSH**.

Entre estes dois, o mais amigável, e escolha para este trabalho, é o aplicativo **Server Auditor SSH SFTP cliente**, criado pela **Crystalnix Limited**.

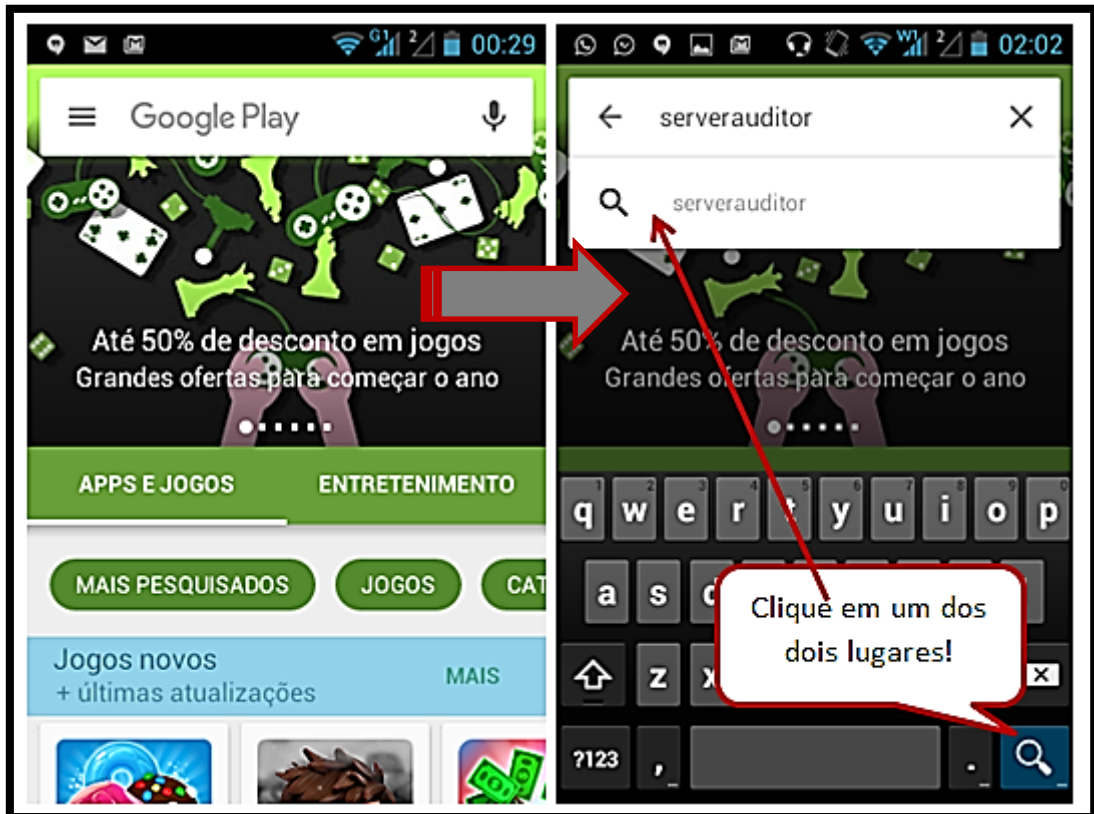
9.6.3.2 Baixando da Play Store o server auditor que transforma seu celular ou tablet em um cliente SSH

Smartphones:

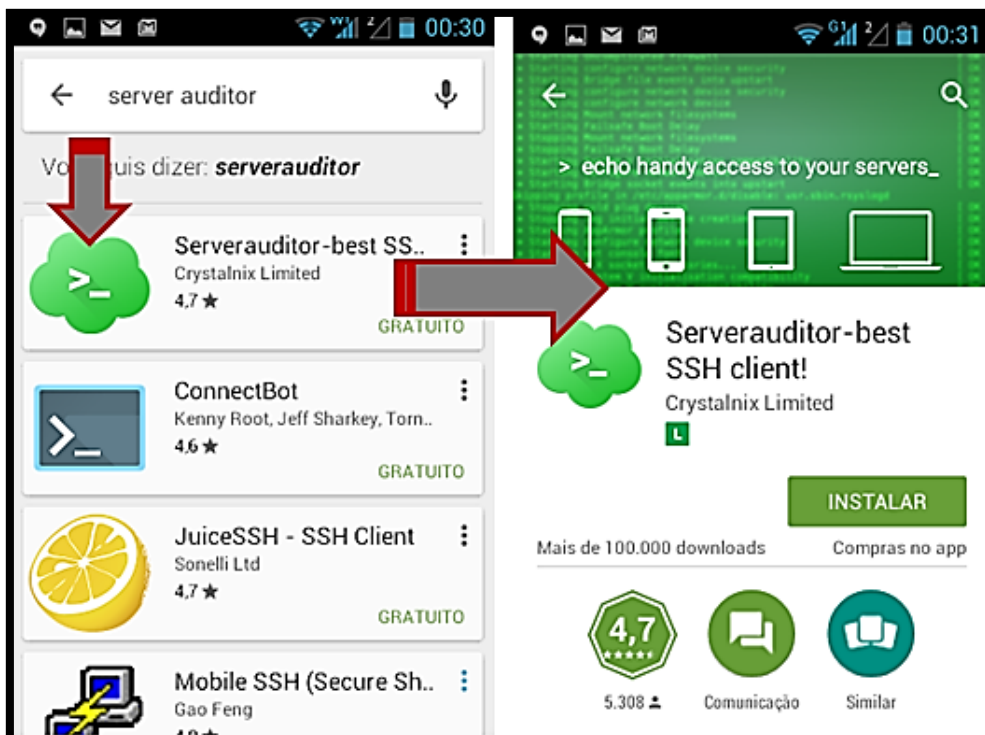
1. Abra o aplicativo Play Store.



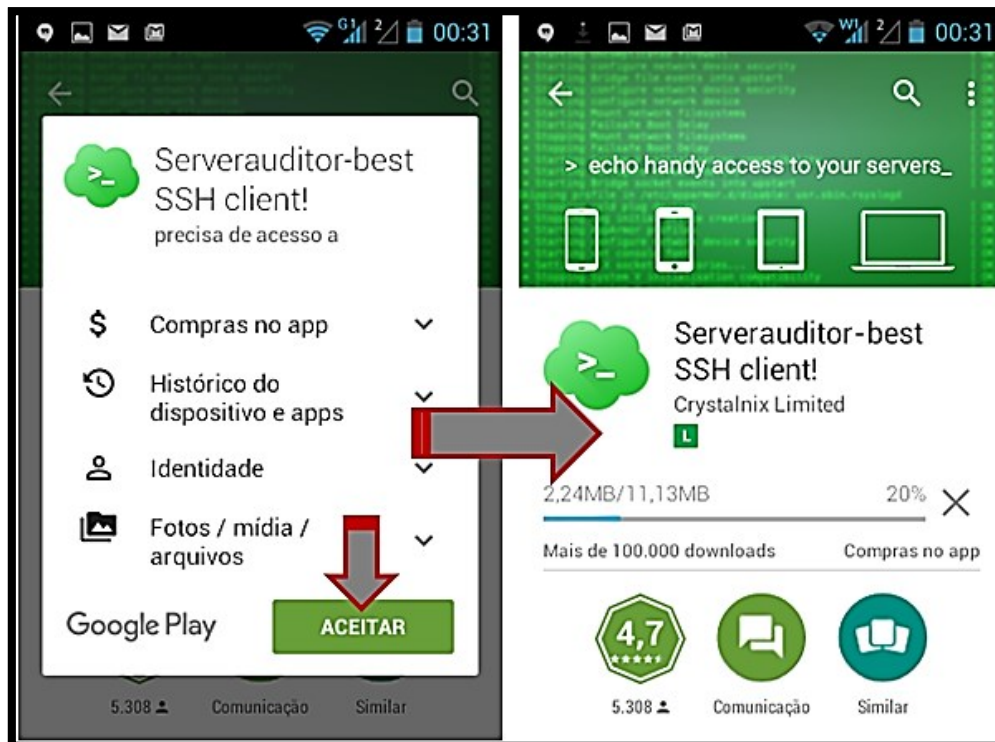
- 2- A janela da Play Store é aberta, clique no campo de texto da parte de cima, onde está escrito **Google Play** e digite **serverauditor** e clique na lupa que aparece enquanto digita.



- 3- O aplicativo é encontrado. Clique no mesmo e a janela de **INSTALAR** é aberta. Clique em **INSTALAR**.



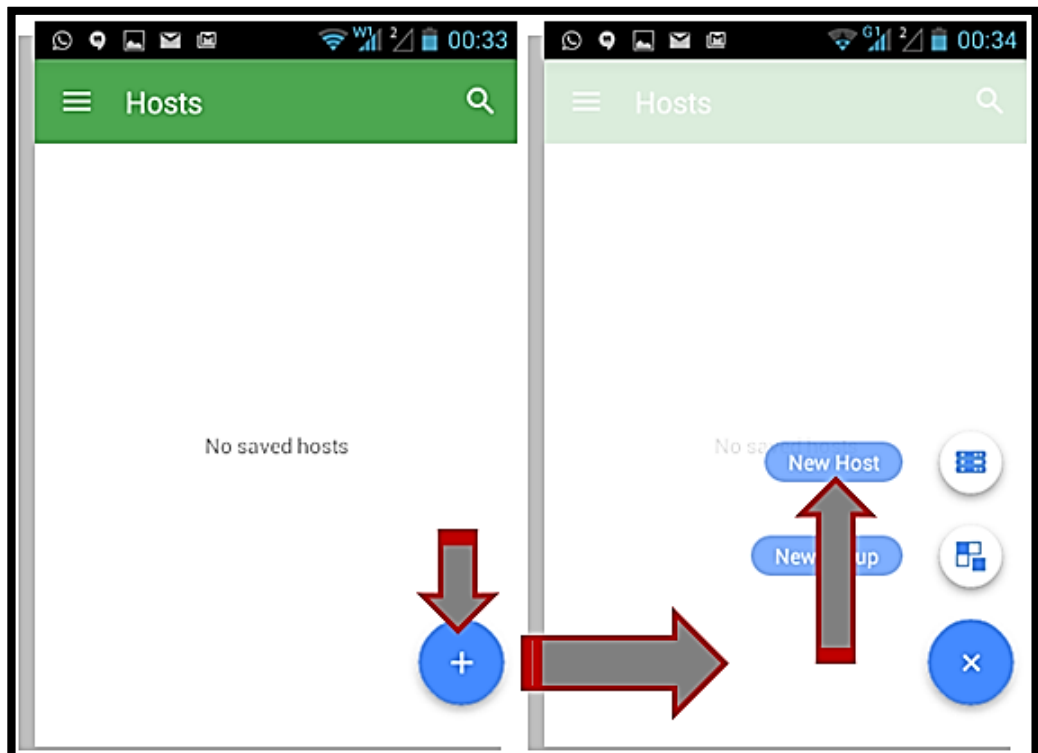
- 4- A janela de permissões é aberta. Bom, não adianta reclamar⁶⁸, tem que aceitar senão o aplicativo não instala.



- 5- Quando acabar de instalar e mandar abrir, se ainda não tiver instalado, antes, este aplicativo, várias informações serão solicitadas, ou seja: as 4 (quatro) solicitadas pelo Windows e Linux, e mais um apelido com que você deseja ser chamado, para aparecer nas telas futuras quando for usar o aplicativo novamente.

Você receberá a informação que não possui **hosts** (hospedagens) cadastrados. Clique na bolinha azul com o sinal de **+**. Uma nova página é aberta para você entrar com uma nova hospedagem (**New Host**). Clique no botão elipsado **New Host**.

⁶⁸ Para quem já tem Whatsapp, não tem mais o que permitir do que já foi permitido.



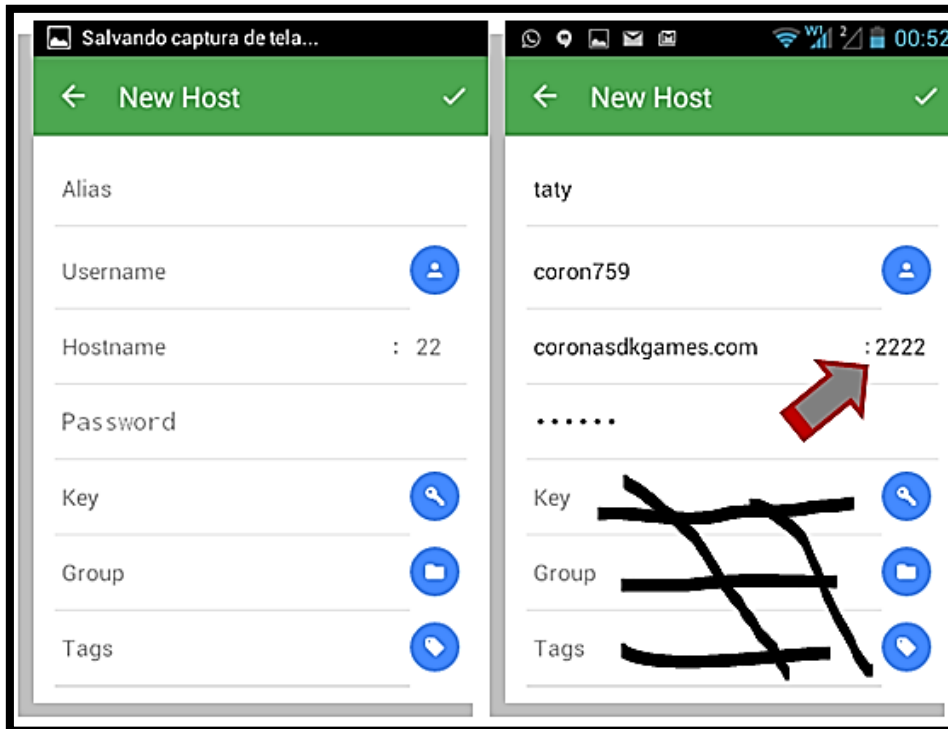
- 6- Ao clicar em **New Host**, a janela para configurar o **host** (seus dados da hospedagem) é aberta. Além das 4(quatro) informações já conhecidas, comoforme dito, pede-se o **Alias**(apelido, nickname)

Neste trabalho, será colocado assim:

- **Alias** – taty
- **Username** – conron759
- **Hostname** – coronasdkgames.com :2222
- **Password** – xxxxxx

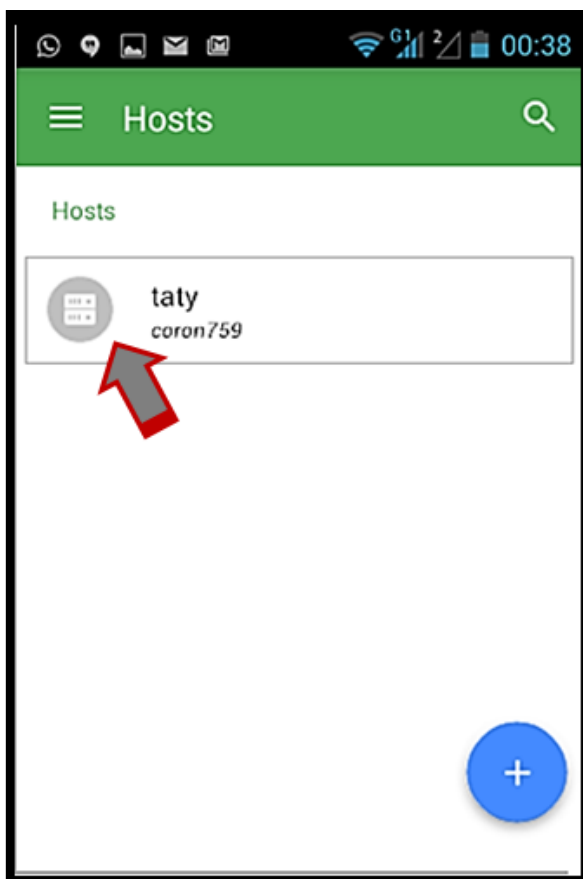
OBS:

- Lembrando que, aqui, Password é a senha da sua hospedagem, não é para criar senha nova.
- **NÃO SE ESQUEÇA DA PORTA SSH, 2222, DEPOIS DO HOSTNAME**

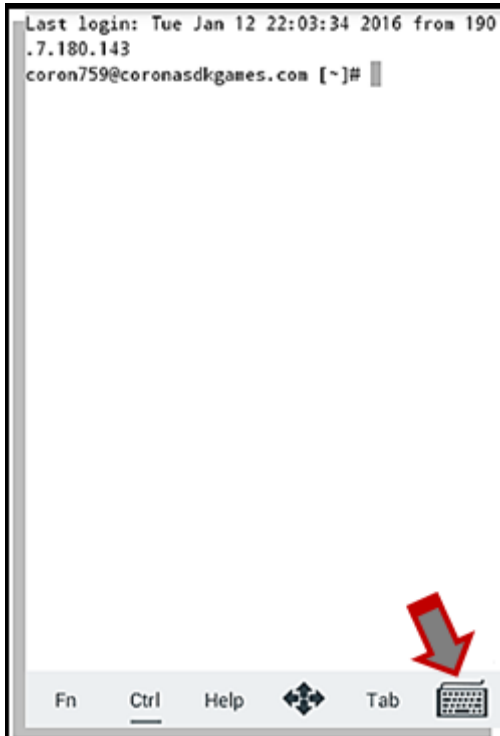


Não é preciso colocar nada em **key**, **Group** e **Tags**.

7- Uma nova janela é aberta com o **host** registrado no seu smartphone Android.



- 8- Clique no seu Host (pode-se criar mais de um host, um para cada hospedagem que quiser acessar). A janela do terminal **SSH é aberta, conforme foi no Windows e Linux**, com o prompt da **coronasdkgames.com**.



- 9- Clique no símbolo de teclado, no canto inferior da janela para iniciar a digitação no terminal. Caso deseje fechar o teclado, clique em **Hide**.
Para entrar no diretório **public_html**, proceda como nos outros terminais, digite **cd public_html** e aperte **ENTER**.



10-A única diferença do que foi mostrado no Linux, Ubuntu, é que, como no teclado dos celulares não tem a tecla **Ctrl** (control), uma tecla **Ctrl** é disponibilizada na barra de menus, na parte inferior da janela



- 11- Agora, abra o editor, o emacs-23.1, como já feito outras vezes, para editar um arquivo já existente ou para iniciar um novo.
- 12-Para tanto, estando em **public_html**, digite **emacs-23.1 android.org**. Desta forma estará abrindo um arquivo com o nome de android.org. Digitando o mesmo arquivo ORG feito nos outros exemplo, tem-se:

```

File Edit Options Buffers Tools Org Tbl Help
p#+TITLE: BRASIL - 04/01/2016
AUTHOR: Tatiana O. Lima
#+OPTIONS: toc:4

* Dólar passa de R$4 e Bovespa cai quase 3%.
* VEJA A ESCALADA DO DÓLAR EM 2015...
* OUÇA UMA DAS MAIS BELAS MÚSICAS QUE O MUNDO
* VEJA OS CELULARES QUE FIZERAM A CABEÇA DOS
* EM 2015 CAI POR TERRA A CENSURA SOBRE OS PS

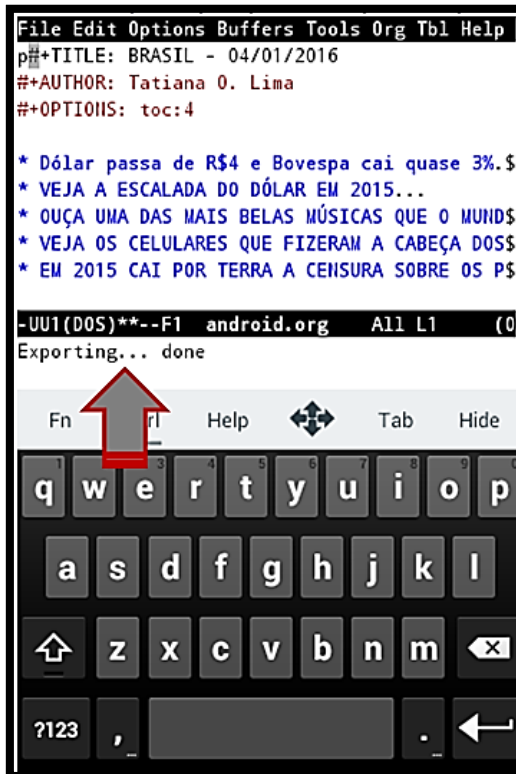
-UU1(DOS)**--F1 android.org All L1 (0)
Exporting... done

  Fn  Ctrl  Help  [Cursor]  Tab  Hide
  q  w  e  r  t  y  u  i  o  p
  a  s  d  f  g  h  j  k  l
  [Home] z  x  c  v  b  n  m  [End]
  ?123 ,  [Space]  .  [Backspace]

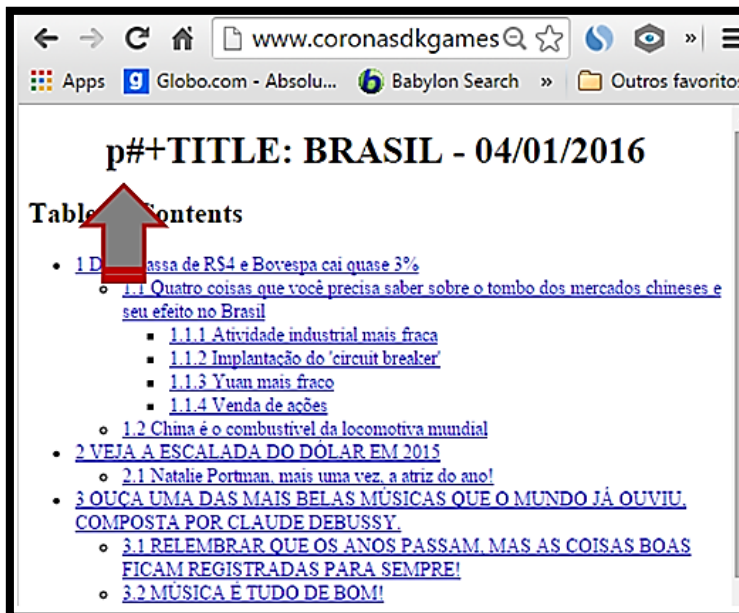
```

Observe que uma letra **p** foi inserida antes de **#+TITLE:**, o que fará com que o título não seja reconhecido e a instrução completa **p#+TITLE** aparecerá na página.

- 13- Para que o menu apareça novamente para poder salvar o arquivo, aperte qualquer parte do texto da janela.
- 14-Assim, salve a página ORG: **C-x C-s**, e depois salve como HTML: **C-c C-e h**. Digite **SSH** auditor no campo de pesquisa e clique na lupa do teclado.



15- Abrindo a página no seu smartphone ou em qualquer outro, ou desktop, ... a página modificada com o **p** no início da mesma é aberta.



9.6.4 Criando Páginas De Notícias Direto Na Nuvem, Com Hipertexto, Multimidia E Programação (Text Mining), Utilizando O Sistema Operacional **Apple** E Org-Mod.



9.6.4.1 Acessando o servidor via SSH em dispositivos Apple

Assim como nos sistemas **Android**, tanto para tablets quanto para celulares, com conexão 3G ou Wi-Fi, nos **Apples** também existem vários programas gratuitos para serem utilizados como clientes em uma conexão **SSH**.

Utilize a frase "**SSH client**" no sistema de pesquisa da Play Store (Central Google de Downloads Android) para escolher qual aplicação desejará instalar. Dentre as várias aplicações disponíveis, destacam-se as duas mais votadas: **Server Auditor** e **vSSH HD Lite**.

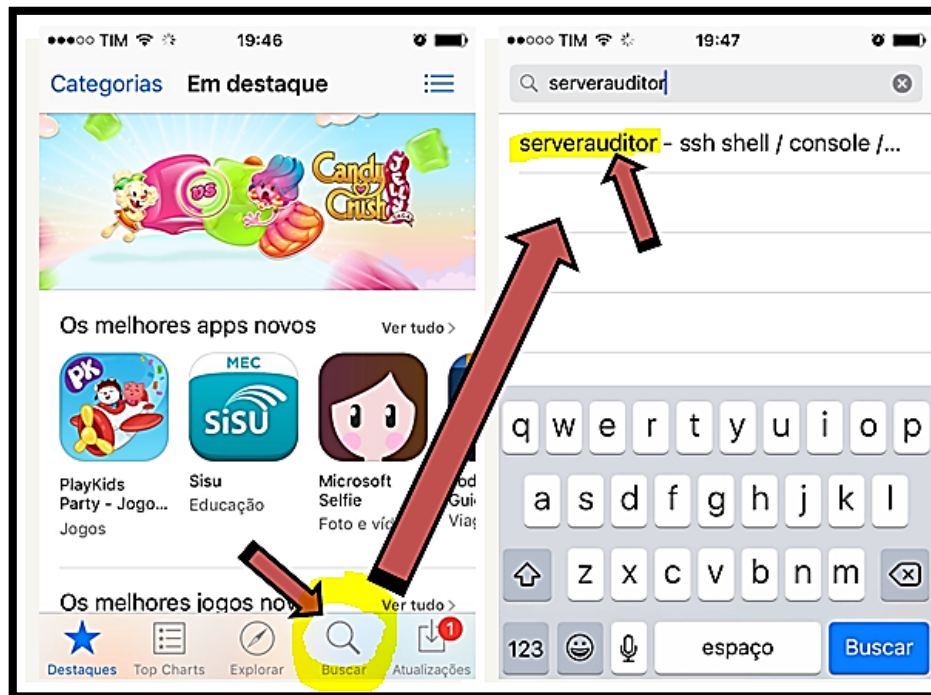
Entre estes dois, o mais amigável, e escolha para este trabalho, é o aplicativo **Server Auditor SSH SFTP cliente**, criado pela Crystalnix Limited, o mesmo já utilizado para Android, o que facilita o entendimento do uso. Será visto que a diferença maior entre o **Server Auditor** do Android e da Apple está no menu da interface, o que não dificulta o uso, pelo contrário, até facilita (no caso da Apple).

9.6.4.2 Baixando da Apple Store o server auditor que transforma seu celular ou tablet em um cliente SSH:

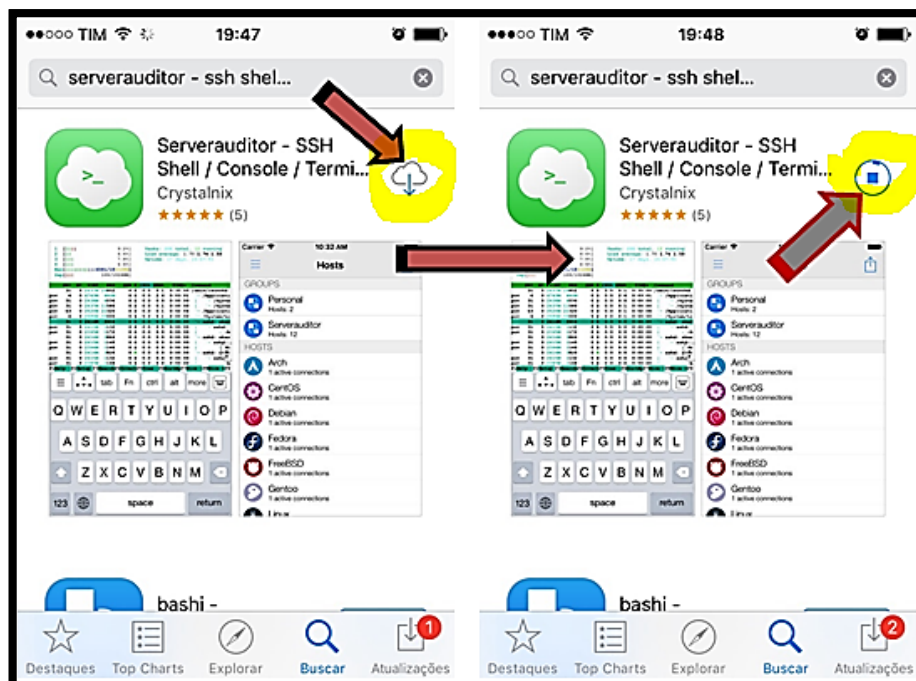
1- Abra o aplicativo Apple Store.



2- A janela da Play Store é aberta. Clique na lupa de buscar, na parte inferior da tela. A janela muda abrindo um campo para pesquisa. Digite **serverauditor**. Feito isto, aparece na linha abaixo o nome do aplicativo. Clique no mesmo.



- 3- O aplicativo é encontrado. Clique n na nuvem no lado esquerdo da tela, abaixo do campo de busca. Ao fazer isto, a nuvem muda para um círculo mostrando o carregamento, o download do aplicativo no celular.

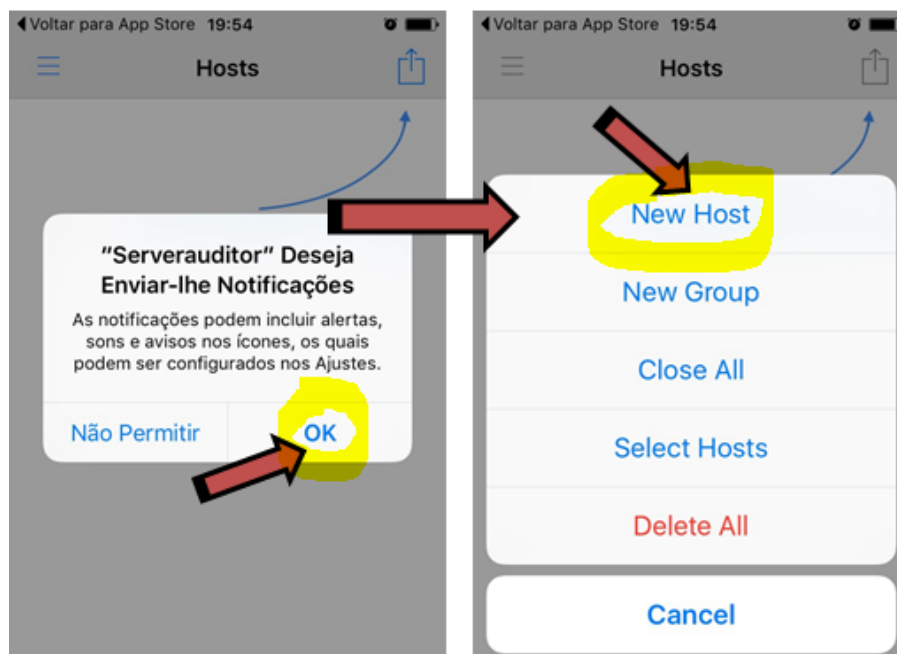


- 4- Quando o aplicativo tiver baixado, a roda de download muda para um botão **ABRIR**. Clicando no botão **ABRIR**, uma nova janela será aberta.

Se for a primeira vez que acessa o **Server Auditor** e não tiver adicionado algum host (hospedagem), A janela que abre solicita que faça isto. Clique no botão indicado na figura.



- 5 Uma janela de solicitação de notificações do server auditor é aberta. Clique em OK concordando. Ao fazer isto, outra janela é aberta. Clique em **New Host** para iniciar o processo de abertura de um terminal **SSH** entre seu **apple** e a hospedagem (host).



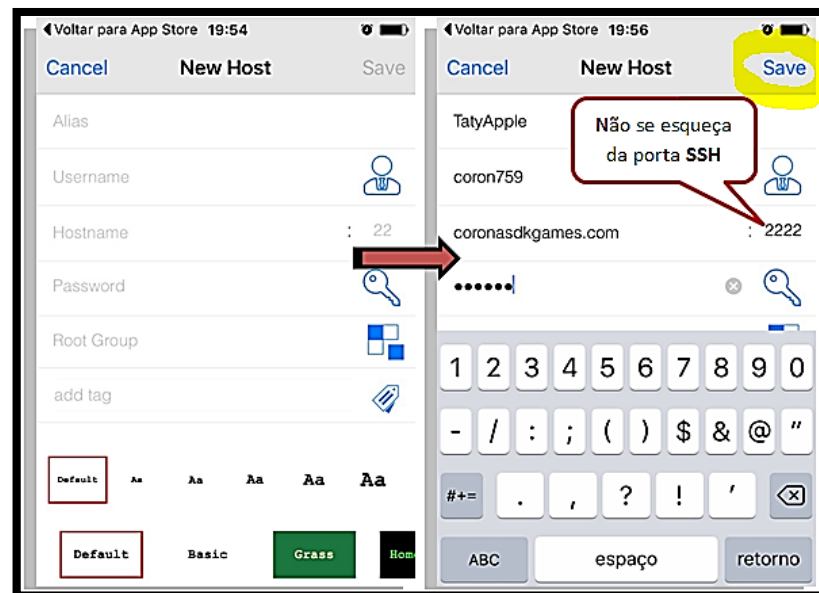
6- A janela de configuração do terminal **SSH** é aberta, semelhante à janela que foi aberta no **Android** (claro, é o mesmo aplicativo)

Segue-se as mesmas instruções:

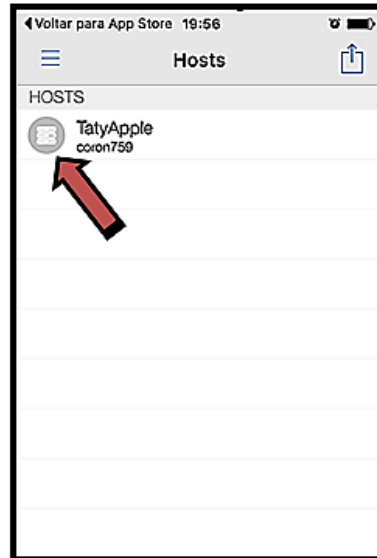
- **Alias** – nickname – Escolha um apelido para sua conexão, como, por exemplo: **tatyApple**
- **Username** - o mesmo: coron759
- **Hostname** - o mesmo: coronasdkgames.com : **2222**
- **Password** – o mesmo: **xxxxxx**

Quando preencher, clique em Save. Não se esqueça de colocar o número da porta **SSH** da sua hospedagem.

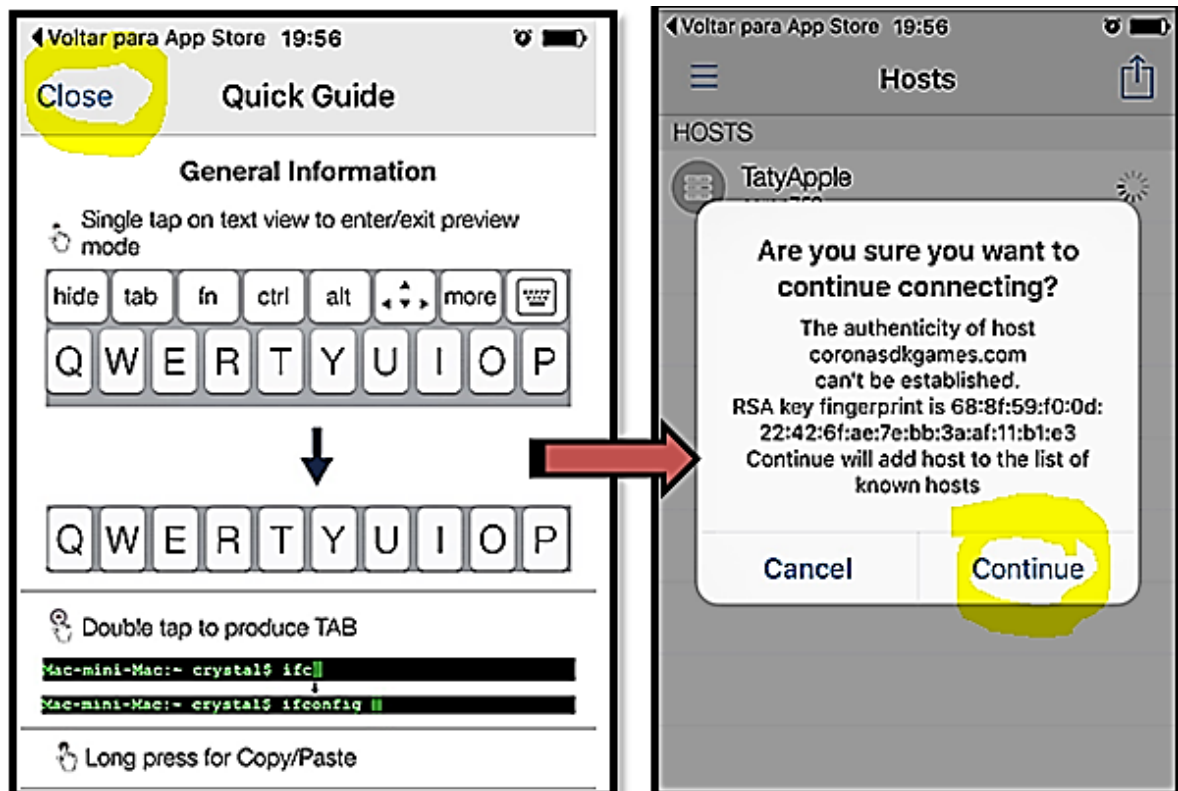
Não é preciso colocar nada em **Root Group** e **add Tags**.



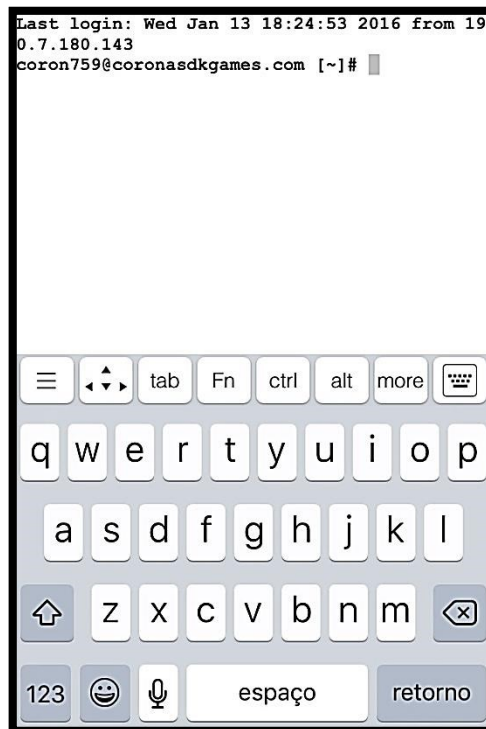
7- Uma nova janela é aberta com o **host** registrado no seu smartphone Apple. Clique no seu Host (pode-se criar mais de um host, um para cada hospedagem que quiser acessar). A janela do terminal **SSH** é **aberta**, conforme foi no **Android**, com o prompt da **coronasdkgames.com**



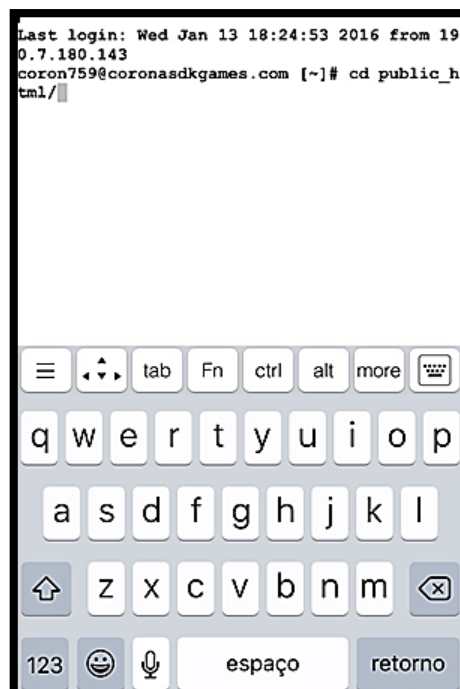
- 8- Um kit de instruções é aberto explicando como usar o **server auditor**. Dê uma olhada, ou clique em **Close**. Uma janela solicitando confirmação da conexão é aberta. Clique em Continue.



- 9- O terminal **SSH** é aberto.

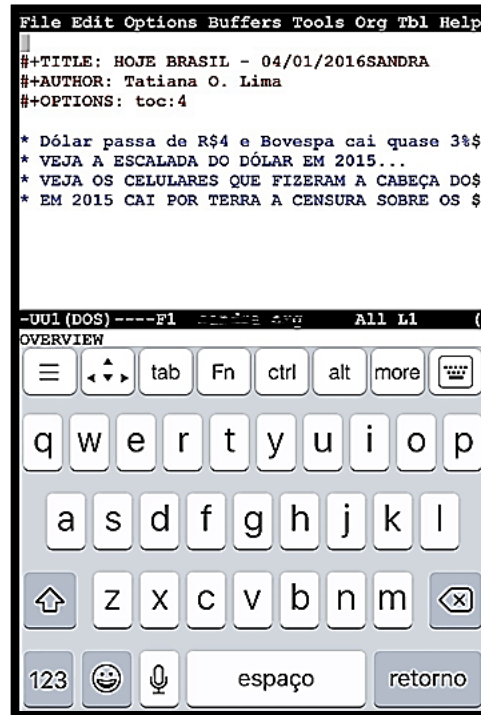


10-Daqui em diante, os procedimentos são os mesmos que foram realizados com o **Android**. Entre no diretório **public_html**, usando o comando **cd public_html** e apertando **ENTER**.



11- Agora, abra o editor, o emacs-23.1, como já feito outras vezes, para editar um arquivo já existente ou para iniciar um novo.

12-Para tanto, estando em **public_html**, digite **emacs-23.1 apple.org**. Desta forma estará abrindo um arquivo com o nome de **apple.org**. Digitando o mesmo arquivo ORG feito nos outros exemplo, obtém-se:



13-Assim, salve a página ORG: **C-x C-s**, e depois salve como HTML: **C-c C-e**
h.

9.7 CRIANDO UMA PÁGINA PROGRAMADA, COM EMACS E LISP, COM UMA PROVA DE CONCEITO DE TEXT MINING, INSERINDO PROGRAMAÇÃO E COMPILAÇÃO JUST IN TIME⁶⁹.

Como toda a filosofia deste trabalho está centrada em se ter o recurso desejado e necessário já existente na hospedagem, de tal forma que não se tenha barreiras nas implementações e publicações das páginas.

Assim como o emacs é uma ferramenta já nativa das hospedagens, a linguagem de programação do **LISP** no **emacs** também é disponibilizada para se programar em alto nível no **Emacs**, sem que se precise acrescentar nada.

O capítulo 7 aborda os conceitos básicos necessários para se compreender o programa que aqui vai ser apresentado.

⁶⁹ **Just in time:** o browse vai gerando a página à medida com que for lendo a mesma, modificando conforme interação com o usuário e programação existente

O Anexo 3 também mostra detalhes desta linguagem com bastante exemplos, e, na internet, existem centenas de sites que, cada um, apresenta uma nova potencialidade, uma nova ferramenta, uma nova forma de fazer o que imaginar com esta fantástica linguagem, que, como já foi dito, impera a mais de 30 anos em grandes corporações de CAD-CAM como o Autocad, por exemplo.

Mesmo assim, o código vai ser comentado com todos os detalhes para que o entendimento não seja comprometido.

Linguagens, principalmente as de extremo alto nível, multiplataformas, multiparadigma, como o LISP, precisam de saber configurar. Assim, neste passo a passo que segue, procura-se mostrar o que ocorre por trás dos bastidores, o antes do acontecer, que não é nem um pouco menos importante do que os resultados que os seguem.

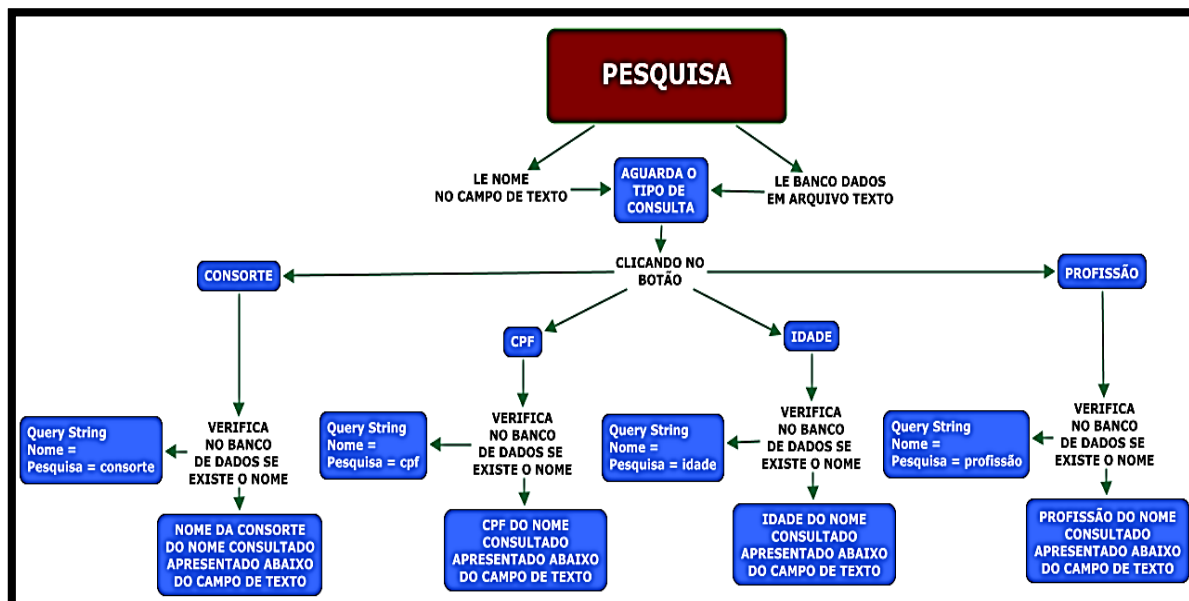
Para exemplificar, será utilizada uma página no **Emacs** com programação em **Lisp**.

A mesma vai fazer um text mining em um banco de dados textual, a partir de um nome digitado pelo usuário em um campo de texto da interface na página.

As pesquisas em cima do nome serão 4 (quatro):

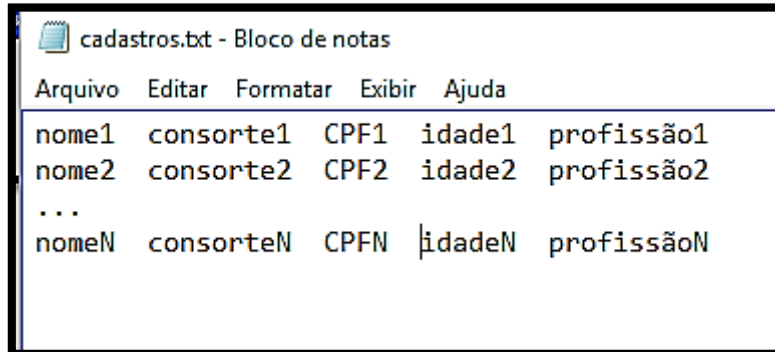
- 5- Consoite
- 6- CPF
- 7- Idade
- 8- Profissão

A pesquisa é feita conforme o mapa a seguir



O formato do arquivo texto é o seguinte:

- Um campo por linha, ou seja:



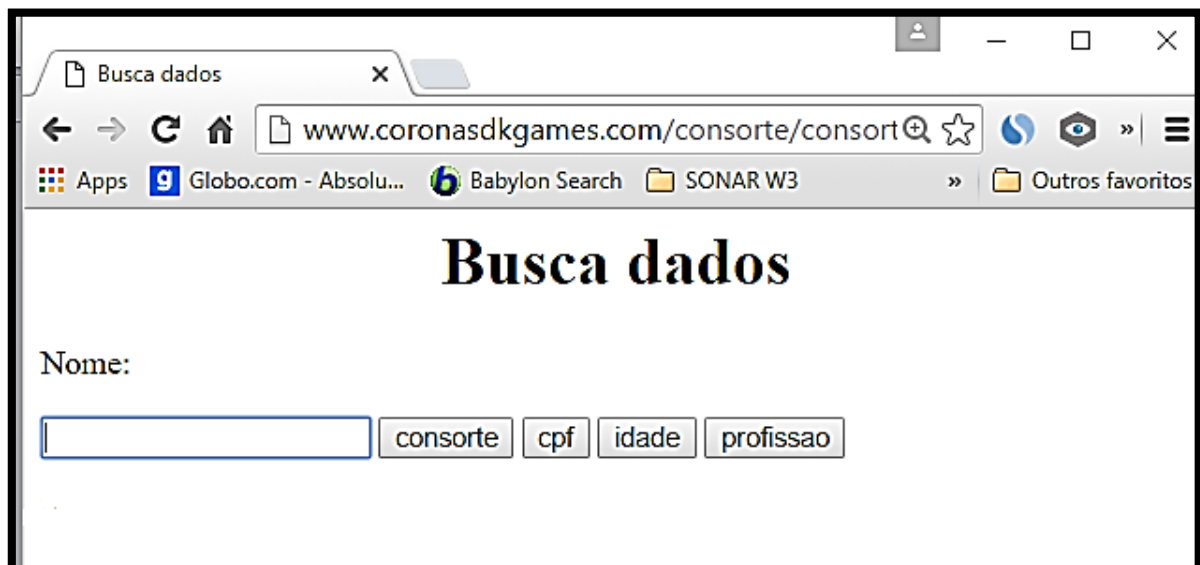
```

nome1 consorte1 CPF1 idade1 profissao1
nome2 consorte2 CPF2 idade2 profissao2
...
nomeN consorteN CPFN idadeN profissaoN
  
```

Lido o nome no campo de texto e clicado no botão da pesquisa, o resultado é apresentado abaixo do campo de texto.

9.8 A PÁGINA PROVA DO CONCEITO

A página terá que ter um campo de texto e botões para fazer cada uma das pesquisas citadas, conforme figura a seguir.



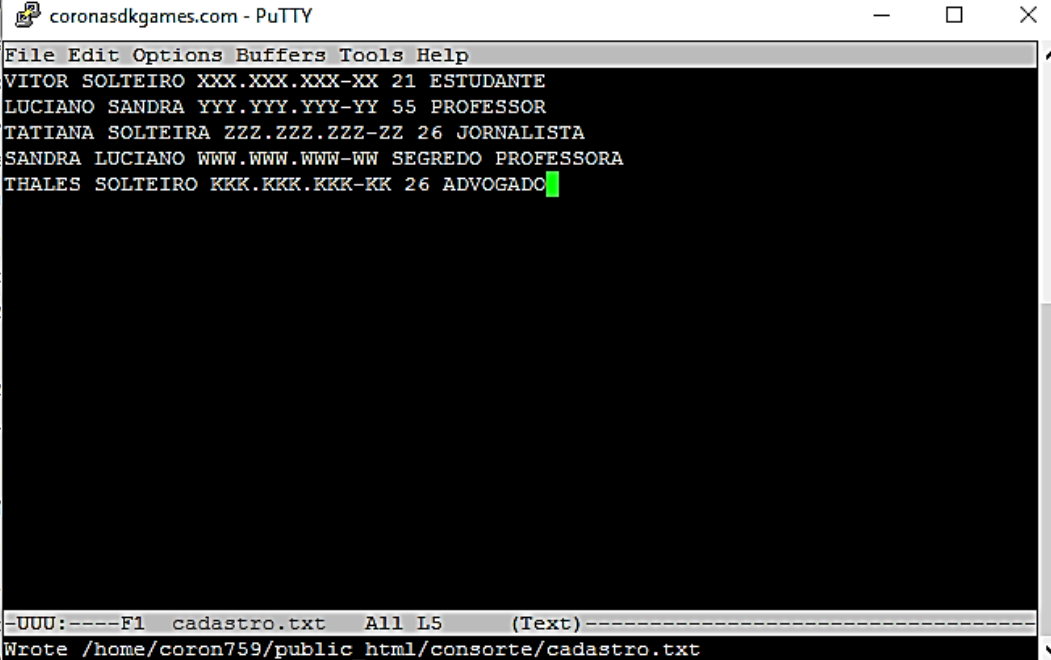
Para testes, o banco de dados inicial, que está na nuvem, dentro de: www.conronasdkgames.com/public_html/consorte/cadastro.txt.

O arquivo cadastro.txt poderá ser alterado direto na nuvem, pelo **emacs**.

Você pode acessar a nuvem pelo Windows, Linux, ...

Usando o **PuTTY**, conforme já detalhado neste capítulo, entra-se na pasta:

```
coron759@coronasdkgames.com [~/public_html]# cd consorte
coron759@coronasdkgames.com [~/public_html/consorte]# emacs-23.1 cadastro.txt
```

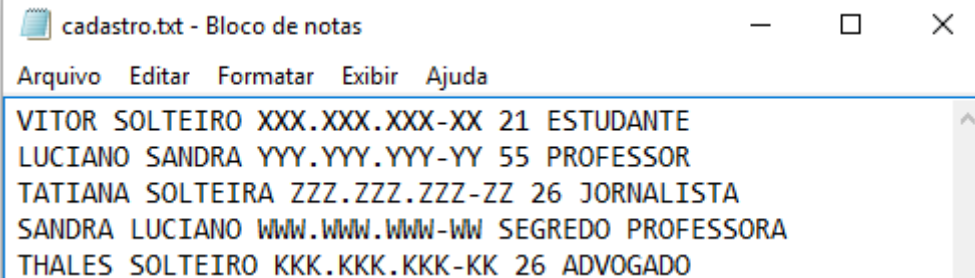


The screenshot shows a PuTTY terminal window titled 'coronasdkgames.com - PuTTY'. The terminal displays the following text:

```
File Edit Options Buffers Tools Help
VITOR SOLTEIRO XXX.XXX.XXX-XX 21 ESTUDANTE
LUCIANO SANDRA YYY.YYY.YYY-YY 55 PROFESSOR
TATIANA SOLTEIRA ZZZ.ZZZ.ZZZ-ZZ 26 JORNALISTA
SANDRA LUCIANO WWW.WWW.WWW-WW SEGREDO PROFESSORA
THALES SOLTEIRO KKK.KKK.KKK-KK 26 ADVOGADO
```

At the bottom of the terminal, a status bar indicates: '-UUU:----F1 cadastro.txt All L5 (Text)-----' and a message: 'Wrote /home/coron759/public_html/consorte/cadastro.txt'.

Você também pode optar por fazer o arquivo utilizando o editor em desktop, como o bloco de notas e, depois, subir o arquivo para a nuvem usando protocolo seguro **scp**.



The screenshot shows a Windows Notepad window titled 'cadastro.txt - Bloco de notas'. The text inside the window is:

```
VITOR SOLTEIRO XXX.XXX.XXX-XX 21 ESTUDANTE
LUCIANO SANDRA YYY.YYY.YYY-YY 55 PROFESSOR
TATIANA SOLTEIRA ZZZ.ZZZ.ZZZ-ZZ 26 JORNALISTA
SANDRA LUCIANO WWW.WWW.WWW-WW SEGREDO PROFESSORA
THALES SOLTEIRO KKK.KKK.KKK-KK 26 ADVOGADO
```

Relembrando, para subir o arquivo para a nuvem, o mesmo deve estar no mesmo diretório do programa **pscp.exe**. No caso dos exemplos, deve estar no diretório **c:/jornal_taty**.

Abre-se um prompt de comando administrador no Windows (continuando no Windows) e entra no diretório **c:/jornal_taty**

```

Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd /

C:\>cd jornal_taty

C:\jornal_taty>pscp.exe

```

Faça a transferência:

```

C:\jornal_taty>pscp -P 2222 -scp cadastro.txt coron759@coronasdkgames.com:~/public_html/consorte/
coron759@coronasdkgames.com's password:
cadastro.txt      | 0 kB | 0.2 kB/s | ETA: 00:00:00 | 100%

C:\jornal_taty>

```

Ainda no PuTTY, crie uma pasta chamada **img** para que se possa armazenar algumas imagens que se possa desejar utilizar e deixa-las organizadas em uma pasta. O comando para se criar uma pasta, um diretório, é **mkdir**. Digintando **ls** seguido do nome da pasta, no caso, **imagem**, observa-se que a mesma está vazia.

```

coron759@coronasdkgames.com [~/public_html/consorte]# mkdir imagens
coron759@coronasdkgames.com [~/public_html/consorte]# ls imagens
./  ../
coron759@coronasdkgames.com [~/public_html/consorte]#

```

9.8.1 Arquivos Executáveis Pelos Browsers

Para rodar arquivos com extensões que não estejam listadas no servidor, é necessário criar um arquivo chamado **.htaccess** e adicionar as extensões desejadas. Isto é interessante para evitar que o sistema operacional na nuvem não fique tutelando muito os arquivos de extensões mais visadas, assim como rackers. Por exemplo: a extensão **.html** é visada nas buscas por páginas HTML. Mas, se você criasse seus arquivos com a extensão **.k**, por exemplo, os mesmos dificilmente seriam alvo de buscas.

Assim, entre no **emacs** e crie o arquivo **.htaccess** no diretório onde a página for rodar. Cada diretório onde for rodar uma pasta, com extensão diferente de **.html**, o mesmo deverá ter um arquivo **.htaccess** de configuração de extensões

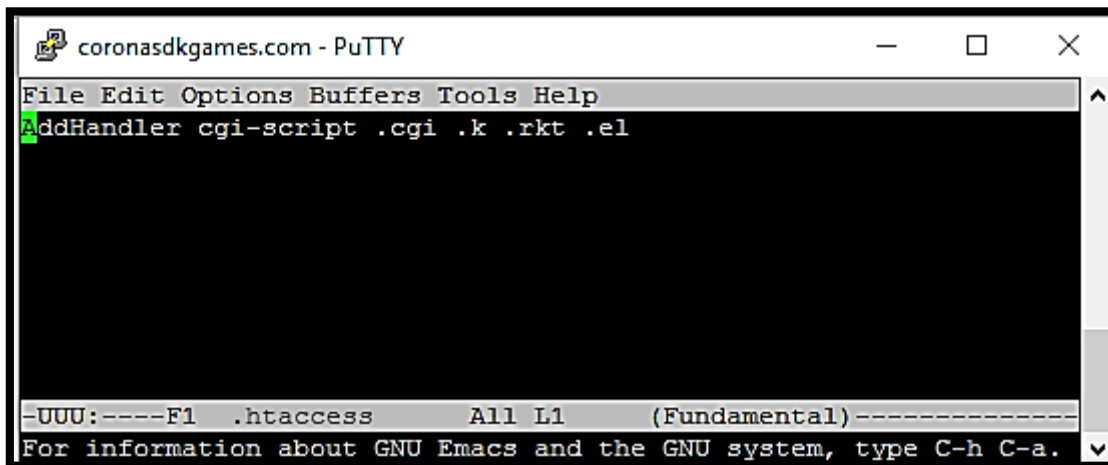
aceitáveis, o que é normal de fazer para páginas programadas, como a que será feita.

A página vai ficar no diretório **consorte**, dentro da pasta **public_html**. Assim, crie a pasta **.htaccess** no diretório **consorte**.

```
coron759@coronasdkgames.com [~/public_html/consorte]# emacs-23.1 .htaccess
```

Digite no mesmo:

AddHandler cgi-script .cgi .el .k .rkt



Com isto, as extensões aceitas serão: **.cgi**, **.k**, **.rkt** e **.el**

Agora:

- Salve com o comando C-x C-s
- Depois saia do Emacs com o comando C-x C-c
- Deixe o terminal aberto. Será necessário utilizá-lo ainda para dar permissão ao Apache executar a página que será criada.

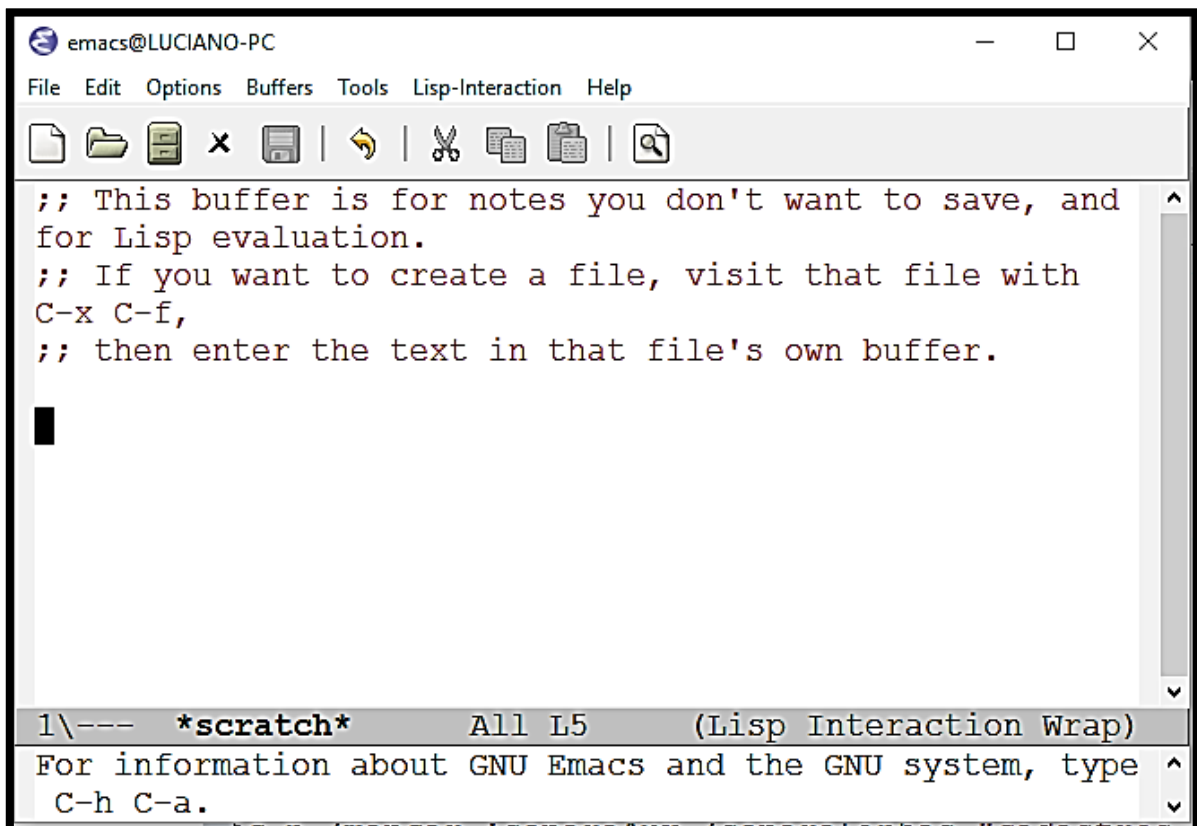
Poder-se-ia criar a página no emacs na nuvem, mas, para exercitar, recordar e ter mais recursos, a página será criada no desktop e depois será subida (uploaded) por protocolo **SSH** para a nuvem.

9.8.2 CRIANDO O PROGRAMA.

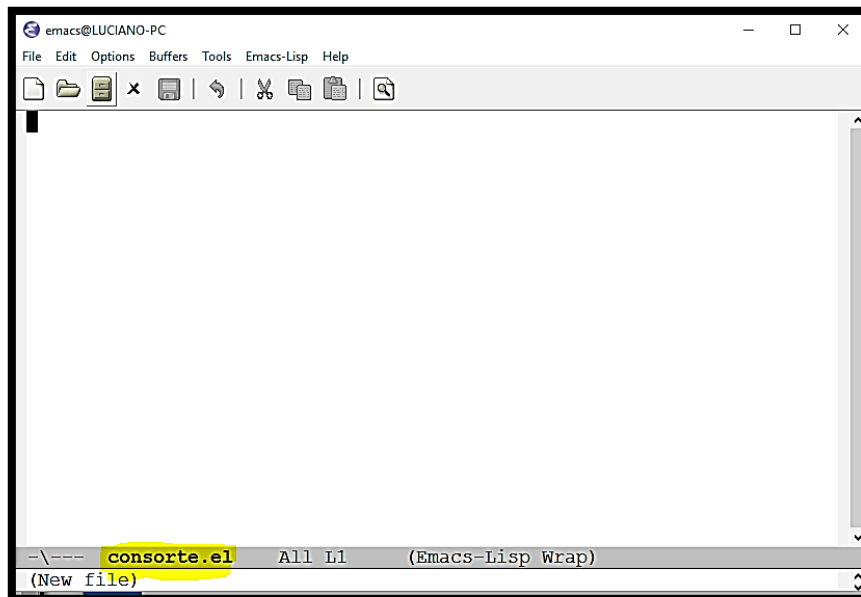
Já foi visto que a interface do programa terá um campo de texto e quatro botões para pesquisas



Abra o **emacs** no seu desktop



Crie o arquivo `consorte.el` com os comandos: **C-x C-f**. A linha de comando é ativada, escreva o nome do arquivo `consorte.el` e aperte **ENTER**.



9.8.2.1 Agora é só iniciar o programa.

Para cada tipo de linguagem que for programar, o cabeçalho do arquivo muda.

Por exemplo:

Se fosse programar em Racket as duas primeiras linhas do arquivo seriam

```
#!/usr/bin/env /home/coron759/public_html/racket/bin/racket
#lang racket
```

Usando LISP, que é o caso do exemplo desta dissertação, as duas primeiras linhas de cabeçalho são:

 A screenshot of the Emacs editor window. The title bar reads 'emacs@LUCIANO-PC'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Emacs-Lisp', and 'Help'. The toolbar contains icons for file operations. The main editing area contains the following text:


```
#!/usr/bin/emacs-23.1 --script
(require 'cl)
```

Onde o comando `#!` é lido como: she(#) bang(!)

(require 'cl) significa que o programa requer o **common lisp**.

Feito o cabeçalho, vem as implementações das funções e, depois, o código HTML da interface.

Funções:

- 1- **Rd_file**: lê o conteúdo de um arquivo, dado o caminho (path) do mesmo, e retorna o conteúdo.

```
(defun rd-file (path)
  "Lê um arquivo e uma monta lista com o conteúdo de cada linha."
  (with-temp-buffer ;;Inicializa um buffer
    (insert-file-contents path) ;;Insere o conteúdo do arquivo no buffer
    inicializado.
    (buffer-string))) ;;Retorna o conteúdo do arquivo.
```

- 2- **separaLinhas**: Separa as linhas do texto pelo caractere de saltar linha (\n), conforme já explicado no capítulo 3.

```
(defun separaLinhas (path)
  "Separa a string com o conteúdo do arquivo em linhas."
  (split-string (rd-file path) "\n"))
```

- 3- **separaAux**: Separa uma string, uma frase, um texto, pelo caractere de espaço, conforme já explicado no capítulo 3.

```
(defun separaAux(str)
  "Divide uma string nos espaços em branco."
  (split-string str " "))
```

- 4- **nome**: função que pega o nome digitado pelo usuário no campo de texto para a pesquisa. Quando o nome é digitado no campo de texto e algum botão é apertado, este nome é encaminhado para a Query String (explicada no capítulo 3), juntamente com a pesquisa definida pelo botão.

```
(defun nome(str)
  "Retira o nome da pessoa da QUERY_STRING."
  (let ((xs (split-string str "&\\|="))) ;;Divide a QUERY_STRING em & e =.
    (loop for x on xs until (equal (car x) "nome") ;;Percorre a lista até
    encontrar a palavra "nome".
      finally (return (cadr x)) ))) ;;Retorna o elemento após "nome".
```

- 5- **pesquisa**: verifica na Query String qual é a pesquisa que deverá ser realizada no nome digitado e retorna o resultado almejado.

```
(defun pesquisa(str)
  "Retira o tipo de pesquisa da QUERY_STRING."
  (let ((xs (split-string str "&\\|=")))) ;;Divide a QUERY_STRING em & e =.
    (loop for x on xs until (equal (car x) "pesquisa") ;;Percorre a lista até
      encontrar a palavra "pesquisa".
      finally (return (cadr x)) ))) ;;Retorna o elemento após "pesquisa".
```

- 6- **consorte**: função associada ao botão **consorte** que retorna o nome do consorte(marido ou mulher) escrito no campo de texto. A função **cadr** foi vista no capítulo 3. A função **mapcar** mapeia a função **separaAux** em todas as linhas do arquivo cadastro. A função **assoc** retorna a primeira associação do primeiro arguento (**n** -> nome) com uma das listas de palavras formadas pela função **separaLinhas** . O nome do consorte é o segundo elemento(**cadr**) da lista encontrada pela função **assoc**.

```
(defun consorte(n)
  "Pesquisa o nome do consorte."
  (cadr (assoc n (mapcar 'separaAux (separaLinhas "cadastros.txt")))))
```

- 7- **cpf**: função associada ao botão **cpf** que retorna o **CPF** escrito no campo de texto. O **cpf** é o terceiro elemento(**caddr**) da lista encontrada pela função **assoc**.

```
(defun cpf(n)
  "Pesquisa o CPF."
  (caddr (assoc n (mapcar 'separaAux (separaLinhas "cadastros.txt")))))
```

- 8- **idade**: função associada ao botão **idade** que retorna a **idade** escrita no campo de texto. A **idade** é o quarto elemento(**caddr**) da lista encontrada pela função **assoc**.

```
(defun idade(n)
  "Pesquisa a idade."
  (caddr (assoc n (mapcar 'separaAux (separaLinhas "cadastros.txt")))))
```

- 9- **profissão**: função associada ao botão **profissão** que retorna a **profissão** escrita no campo de texto. A **profissão** é o quinto elemento(**caddr**) da lista encontrada pela função **assoc**.

```
(defun profissao(n)
  "Pesquisa a profissão."
  (car (caddr (assoc n (mapcar 'separaAux (separaLinhas "cadastros.txt")))))
```

10-busca: função que analisa qual pesquisa foi acionada pelos botões e aplica a função de pesquisa correspondente, devolvendo a resposta desejada, caso tenha no banco de dados textual, ou retornando **nil** se não encontrar.

```
(defun busca(qs)
  "Analisa o tipo de busca e a realiza."
  (if (equal (pesquisa qs) "consorte") (consorte (nome qs))
      (if (equal (pesquisa qs) "cpf") (cpf (nome qs))
          (if (equal (pesquisa qs) "idade") (idade (nome qs))
              (if (equal (pesquisa qs) "profissao") (profissao (nome qs))
                  "Pesquisa não encontrada. Faça uma nova pesquisa!")))) ))
```

9.8.2.2 Código html da interface

9.8.2.2.1 *Cabeçalho do arquivo HTML*

Indicando que o conteúdo é TEXTO HTML(**text/html**) com codificação de caracteres **utf-8** e que o mesmo possuirá um padrão de fundo feito pela imagem **index.jpg**, a qual está no diretório **img**.

```
(princ "Content-type: text/html")
(terpri)
(terpri)

(princ "
<html>
  <head>
    <title>Busca dados</title>
    <meta http-equiv='Content-type' content='text/html;charset=utf-8' />
    <style>
      html{
        background: url('img/index.jpeg');
      }
    </style>
  </head>")
```

Onde:

- **terpri** significa: saltar uma linha
- **princ** - imprime no arquivo HTML que vai ser gerado pelo browser.

9.8.2.2.2 *Corpo do arquivo HTML*

Criando o título: **Busca de dados**, criando um campo de texto e 4 botões.

```
(princ "
  <body>
    <h1 style='text-align:center;'>Busca dados</h1>
    <p>Nome: </p>
    <form method='get'>
      <input type='text' name='nome'>
      <input type='submit' name='pesquisa' value='consorte'>
      <input type='submit' name='pesquisa' value='cpf'>
      <input type='submit' name='pesquisa' value='idade'>
      <input type='submit' name='pesquisa' value='profissao'>
    </form>")

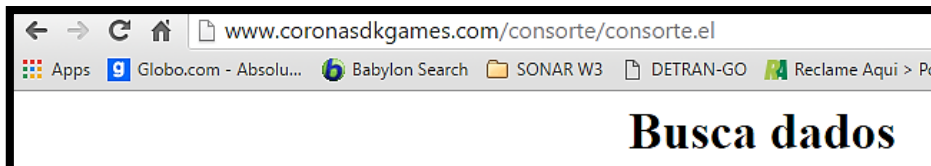
(unless (getenv "QUERY_STRING")
  (setenv "QUERY_STRING" "nome=Vitor&pesquisa=consorte"))
(let* ((txt (getenv "QUERY_STRING")))
  (princ (format "<p>%s</p>" (busca txt))))

(princ "
</body>
</html>")
```

Os elementos do corpo são:

1- TITULO:

```
<h1 style='text-align:center;'>Busca dados</h1>
```



2- CAMPO DE TEXTO

Caption

```
<p>Nome: </p>
```

Nome:

Campo de texto

```
<input type='text' name='nome'>
```

Resultado final.

Nome:

3- BOTÕES

```
<input type='submit' name='pesquisa' value='consorte'>
<input type='submit' name='pesquisa' value='cpf'>
<input type='submit' name='pesquisa' value='idade'>
<input type='submit' name='pesquisa' value='profissao'>
```

Resultado:

Nome:

- 4- FECHAMENTO DO CORPO:** Lê a **Query String** (inicia a mesma com alguma coisa para não dar erro na primeira abertura da página) e chama a função **busca** para ver qual pesquisa deve ser feita.

```
(unless (getenv "QUERY_STRING")
  (setenv "QUERY_STRING" "nome=Vitor&pesquisa=consorte"))
(let* ((txt (getenv "QUERY_STRING")))
  (princ (format "<p>%s</p>" (busca txt))))

(princ "
</body>
</html>")
```

SALVAR:

Salva-se o Arquivo com **C-x C-s**.

SAIR DO EMACS

C-x C-c

9.8.3 Subindo Os Arquivos Para A Nuvem.

Finalmente tudo está quase pronto para rodar na nuvem a página dinâmica programada.

Para isto, tem-se que ainda cumprir as seguintes etapas:

- 1- Entrar no terminal

- 2- Entrar na nuvem no diretório **public-html** da sua hospedagem
- 3- Entrar no diretório dentro de **public_html** onde está a página com respectivo arquivo de extensões **.htaccess**
- 4- Digitar a permissão para o servidor rodar a página quando solicitado por algum browser. Isto é feito digitando o seguinte comando:

chmod a+x nome_do_arquivo

No caso, o **nome_do_arquivo** é **consorte.el**:

```
coron759@coronasdkgames.com [~/public_html/consorte]# chmod a+x consorte.el
coron759@coronasdkgames.com [~/public_html/consorte]#
```

Agora é rodar a página.

Se ao rodar a página, como segue, vir uma mensagem de erro, do tipo:



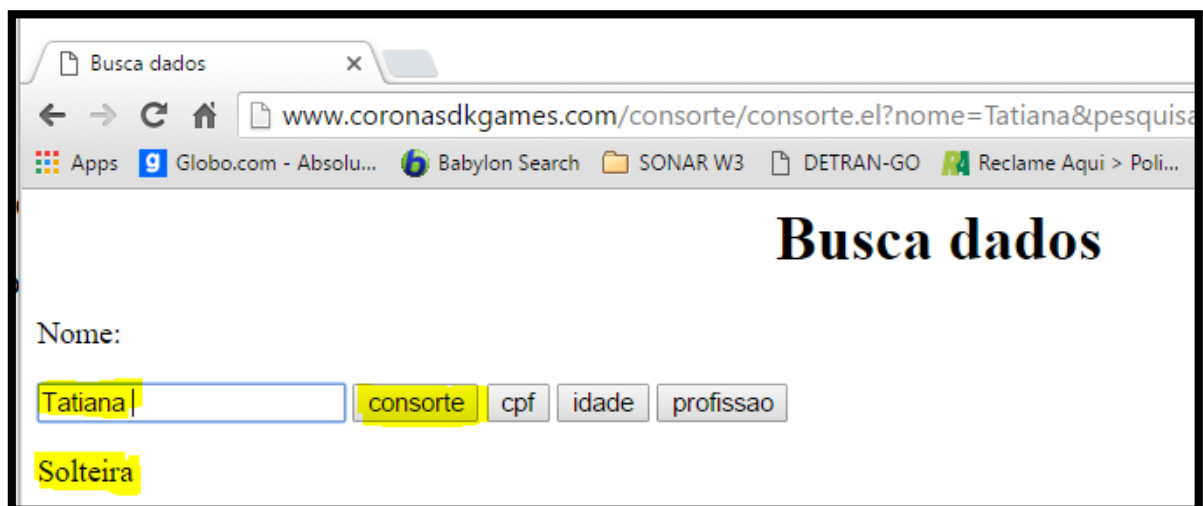
Isto significa que:

- ou tem um erro na sua página,
- ou você não deu o acesso **chmod a+x nome_do_arquivo** dentro do diretório onde está a página
- você digiou a extensão errada ou não colocou a mesma no arquivo **.htaccess**

Se tudo der certo, você verá a página **consorte.el** corretamente



e poderá realizar as pesquisas desejadas.



9.9 CONCLUSÃO

Conhecendo os parâmetros, os dados de acesso da hospedagem, fica fácil, simples, transferir com segurança dados entre cliente (seu computador) e servidor (a nuvem), Anexo 1, bem como construir páginas dinâmicas e estáticas diretamente na hospedagem, em tempo real, independentemente da plataforma na qual se está desenvolvendo (Windows, Linux, Android ou Apple), independentemente do dispositivo (mobile ou desktop) e independentemente do browser que fará a leitura.

CONSIDERAÇÕES FINAIS

Com a internet, a forma de produzir notícia mudou, e isso significa alterar a maneira que os jornalistas encaram a tecnologia, para adaptar os jornais da melhor forma possível aos nativos digitais. Vê-se que muitos jornais ainda não conseguiram habituar-se ao meio online, onde velocidade e qualidade é requisito obrigatório. A pesquisa documental fez com que fosse visto ainda mais a necessidade de encontrar uma solução para otimizar o processo de produção e publicação da notícia.

A democratização da informação, simplicidade de acesso e facilitação técnica na formatação e mineração de dados em informações emergentes e existentes em grandes big datas (text mine) podem trazer tanto benefícios para os jornais quanto para os leitores.

Capacitar os jornalistas de maneira que se tornem aptos a programarem suas páginas de jornal na própria nuvem e utilizando quaisquer dispositivo com acesso à internet pode ser o diferencial que falta para agilizar o processo de produção da notícia e fazer com que os jornais se minem com mais notícias inéditas e menos reprodução de informação transmitida por internautas que poderiam ser leitores.

Vários paradigmas e respectivas linguagens foram testados no laboratório de Inteligência Artificial da Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia, buscando atender aos requisitos exigidos pelos jornais, como multimídia, hipertextos e publicação em tempo real. A escolha da linguagem Emacs-orgmode-emas-lisp para prova dos conceitos derivados dos objetivos dessa dissertação é a mais aderente e vai de encontro a esta pesquisa. A prova de conceito mostra que é possível utilizar Paradigmas e linguagens para implementação de páginas jornalísticas hipermídia interativas, para edição on-line, direta na nuvem, gerando código executável ou compilação *just in time* na nuvem, independente de sistema operacional, browsers e dispositivos.

Estruturar uma página em HTML é algo simples, mas não é fácil. Demanda muita atenção. Qualquer espaço, aspas, enfim, qualquer detalhe que for esquecido ou acrescentado compromete a página toda. Pela simplicidade, agilidade na implementação das páginas, por independer da censura da jailshell, por ser

padrão em todos os sistemas operacionais da atualidade, por ser nativo nas hospedagens da nuvem e por ser uma linguagem multiparadigma, além do melhor editor para tais propósitos, definiu-se pelo emacs Org-Mode.

O profissional jornalista possui pouca fluência ou nenhuma fluência em programação. Ele está acostumado a edição de texto em editores com linguagem em mark-up, mas anseiam por recursos mais complexos de páginas dinâmicas Web multimídia, com hipertextos e que rodem em tecnologia mobile e desktop com igual qualidade e precisão. O produto gerado por essa tese provou que utilizando a tecnologia que já é de conhecimento do jornalista, a construção textual, é possível atender sua demanda com maior eficácia.

TRABALHOS FUTUROS

Aplicar o conhecimento levantado nessa tese em cursos de Jornalismo e em grandes redações jornalísticas do país. Isso a fim de validar a prova de conceito e contribuir efetivamente com a formação do jornalista no Brasil.

Fazer um levantamento das críticas declaradas na mídia, bem como induzidas através de entrevistas e formulários, do grau de satisfação e das críticas que os leitores, principalmente pagantes, têm como conceito dos jornais que escolhidos para lerem e satisfazerem seus perfis de interesse. De posse de tais informações, elaborar um sistema humano/informatizado que viabilize uma análise comparativa entre os elementos gráficos dos jornais impressos e digitais, tendo a base de dados levantada como pilar desta etapa.

Levantar as sugestões do que tecnicamente seria interessante e relevante na montagem dos jornais, principalmente digitais, para que o trabalho pudesse ser melhor concretizado na forma com que foi idealizado pelo autor e talvez cortado ou editado pela redação por motivos, limitações técnicas.

Após ter conhecimento de formas eficientes de programação de páginas na nuvem, a intenção é pensar em grande escala: montar um jornal de grande circulação. No caso, necessita de um servidor, banco de dados e recursos que os servidores padrões, genéricos das hospedagens não atendem em velocidade, recursos e liberdade de acessos e buscas de informações. Neste caso, DrRacket é a escolha de linguagem.

Para trabalho futuro, a intenção é descobrir novas estratégias de continuar atraindo o público leitor para os jornais na web. Para isso, tem-se como objetivo posterior criar um montador de páginas personalizadas para web, de acordo com a preferência do leitor.

REFERÊNCIAS

AGNER, Luiz; et. al. **Jornalismo para plataformas móveis: avaliação interdisciplinar de comunicabilidade de interfaces gestuais**. In SOUTO, Virginia Tiradentes; PORTUGAL, Cristina (ORGs). Anais do 7º CIDI e 7º CONGIC. Sociedade Brasileira de Design da Informação. Brasília: 2015. https://doi.org/10.5151/designpro-CIDI2015-cidi_81

ALCADIPANI, Rafael. **O declínio dos jornais**. GV Executivo. Vol.6 Nº2. Mar./Abr. 2007. <https://doi.org/10.12660/gvexec.v6n2.2007.34558>

ARNT, Hérís. **Do jornal impresso ao digital: novas funções comunicacionais**. INTERCOM, Salvador, 2002. Disponível em <<http://www.portcom.intercom.org.br/pdfs/1afd5a712c7cd14a1199a27ab1defe9c.pdf>>. Acesso em 07 Janeiro 2016.

BARBER, Phil. **A Brief History of Newspapers, Historic Newspapers and Early Imprints**. Cambrige, Reino Unido, 2002. Disponível em <www.historicpages.com/>. BUENO, Thaisa. **Resumo atividades e funções do jornalista**. Thaisa Bueno, 2011. Disponível em <<http://jornalismo.ufma.br/thaisa/2011/08/30/resumo-atividades-e-funcoes-do-jornalista/>>. Acesso em 07 Janeiro 2016.

BRASIL; Presidência da República; Secretaria de Comunicação Social. **Pesquisa brasileira de mídia 2015 : hábitos de consumo de mídia pela população brasileira**. – Brasília : Secom, 2014. Disponível em <<http://www.secom.gov.br/atuacao/pesquisa/lista-de-pesquisas-quantitativas-e-qualitativas-de-contratos-atuais/pesquisa-brasileira-de-midia-pbm-2015.pdf>> Acesso em 07 Janeiro 2016.

CANAVILHAS, João. **Webjornalismo: Da pirâmide invertida à pirâmide deitada**. BOCC - Biblioteca On-line de Ciências da Comunicação, 2006. Disponível em <www.bocc.ubi.pt>. Acesso em 07 Janeiro 2016.

FERRARI, Pollyana. **Jornalismo digital**. São Paulo: Contexto, 2004.

FERREIRA, Fernanda Vasques; SOUZA, Flávia Simone Pereira de. **Pioneirismo, inovações, crises financeiras e a transição de plataforma na história do Jornal do Brasil**. POLIANTEA: 2014. Volumen X, número 18. p. 145-176. <https://doi.org/10.15765/plnt.v10i18.526>

FILHO, Gelson Amaro de Souza. **Jornalismo on-line: guia teórico e prático**. Faculdade de Comunicação Social “Jornalista Roberto Marinho”, Presidente Prudente, 2006. Disponível em <<http://webjornalismo.gelsonamaro.com/>>. Acesso em 07 Janeiro 2016.

FRANÇA, Livia Valença da Silva. **Marketing X e Consumidor Y: Os Desafios de Propor Novas Experiências de Marca e Comunicação para um Novo Público Brasileiro**. Intercom, Rio de Janeiro, 2015. Disponível em <<http://portalintercom.org.br/anais/nacional2015/resumos/R10-3884-1.pdf>>. Acesso em 07 Janeiro 2016.

IBOBE. **Brasileiros com Internet no smartphone já são mais de 70 milhões**. Associação Brasileira de Comunicação Empresarial, 2015. Disponível em <http://aberje.com.br/acervo_not_ver.asp?ID_NOTICIA=13565&EDITORIA=Geral>. Acesso em 07 Janeiro 2016.

Jornais: Breve História, Associação Nacional de Jornais. Disponível em <http://www.anj.org.br/jornais-breve-historia-2/>. Acesso em 07 Janeiro 2016.

LANDOW, George P. **Hipertexto: La convergência de la teoria crítica contemporânea y la tecnología**. Tradução de Patrick Ducher. Barcelona: Paidós, 1992.

LARANJEIRA, Álvaro. **A imersão da humanidade no universo on-line**. Revista Famecos, Porto Alegre, n. 9, p. 167-172, dez. 1998. <https://doi.org/10.15448/1980-3729.1998.9.3022>

LINDOO, Edward C. **The Future of Newspapers: A study of the World Wide Web and its relationship to the electronic publishing of newspapers**. Local Free, 1998. Disponível em www.localfreepress.com/admin/research. Acesso em 16 Dezembro 2015.

MARSHALL, Leandro. **O Jornalismo na Era da Publicidade**. Summus, São Paulo, 2003.

MELECH, Edgard Cesar. **Morte do jornal de papel e novas tecnologias digitais: desafios e perspectivas**. Comunicação & Mercado/UNIGRAN - Dourados - MS, vol. 01, n. 02 – edição especial, p. 56-64, nov 2012.

MOHERDAUI, Luciana. **Em busca de um modelo de composição para os jornais digitais**. Contemporânea: vol. 6, nº 2. Dez.2008. Disponível em <http://www.portalseer.ufba.br/index.php/contemporaneaposcom/article/view/3530/2583>>. Acesso em: 20 jul. 2015.

REDAÇÃO PORTAL IMPRENSA. **"New York Times" anuncia fim de sua versão impressa na Tailândia**. Redação Portal IMPRENSA: 2015. Disponível em <http://portalimprensa.com.br/noticias/internacional/75321/new+york+times+anuncia+fim+de+sua+versao+impressa+na+tailandia>>. Acesso em 07 de Janeiro de 2016.

SÃO PAULO. **Jornal 'New York Times' impresso deixa de ser distribuído no Brasil**. Folha de São Paulo: 2013. Disponível em <http://www1.folha.uol.com.br/mercado/2013/05/1280906-jornal-new-york-times-impreso-deixa-de-ser-distribuido-no-brasil.shtml>>. Acesso em 07 de Janeiro de 2016.

STEPHANOWITZ, Helena. **Crise econômica na velha mídia é mais grave que a do país que ela pinta**. Rede Brasil Atual, 2015. Disponível em <http://www.redebrasilatual.com.br/blogs/helena/2015/08/crise-economica-na-velha-midia-e-mais-grave-que-a-do-pais-que-ela-pinta-8714.html>>. Acesso em 07 Janeiro 2016.

TERCEIRO, José R. **Sociedade Digital**. Relógio D'água, Lisboa, 1997.

VICTOR, Paulo. **Internauta passa quase quatro horas por dia na web**. Agência Brasil: 2014. Disponível em <<http://agenciabrasil.ebc.com.br/pesquisa-e-inovacao/noticia/2014-03/internauta-passa-quase-quatro-horas-por-dia-na-web>>.

Acesso em 07 Janeiro 2016.

WOLF, Mauro. **Teorias da comunicação de massa**. São Paulo: Martins Fontes: 2003.

ZANELLA, S. Entrevista concedida à Claudia Quadros no mês de dezembro de 2008, em Curitiba.

ANEXO 1 INSTALANDO O EMACS NO WINDOWS

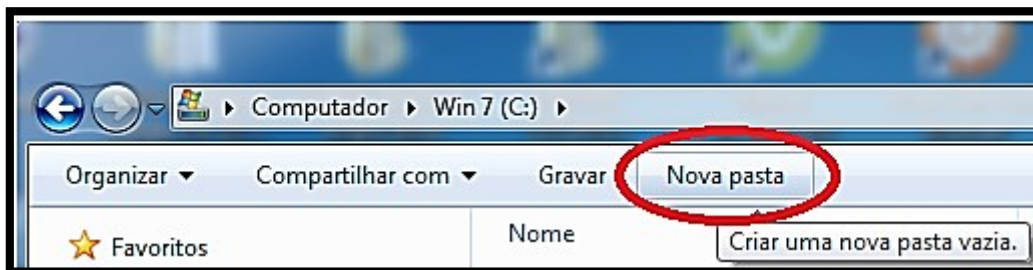
É importante padronizar onde instalar o programa. Saber onde está instalado. Não deixar à cargo do sistema operacional, no caso, o Windows, escolher instalar onde bem desejar. Saber onde tudo está instalado é fundamental para um profissional poder fazer alterações, configurações e outras ações que se façam necessárias, como é este, o caso. Usar o Emacs no Linux é simples. Tudo foi criado para estar no local certo. Tudo funciona em uma pasta chamada **home**, pasta esta que, no Windows, ninguém sabe dizer onde está, mas, seguindo os passos que serão mostrados, este problema deixa de existir e pode-se trabalhar no Windows com o Emacs sem nenhum problema, como um bom profissional trabalharia.

- **INSTALANDO O EMACS NA RAIZ DO DISCO C DO WINDOWS, DENTRO DE UMA PASTA CHAMADA **emacs**, OU SEJA:**

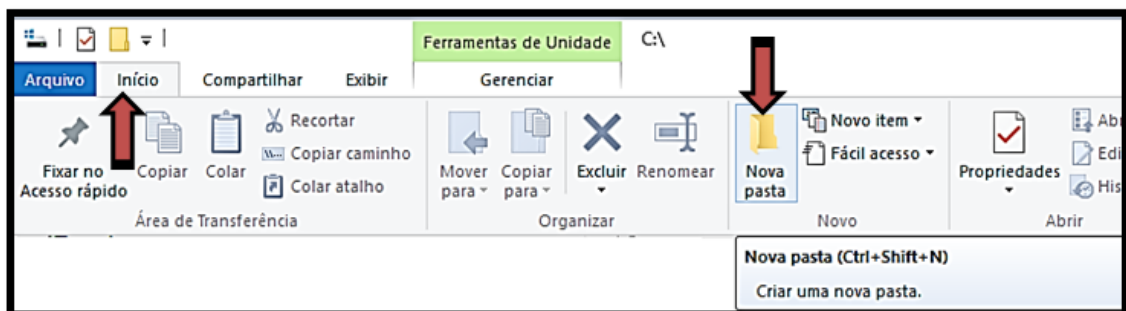
- Instalar o **emacs** em **C:\lemacs**.

Para tanto, crie um diretório na pasta C: chamada emacs. Em cada versão do Windows isto é feito de uma forma diferente.

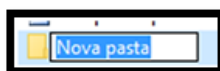
No Windows 7, estando no gerenciador de arquivos, entra-se no disco C: e clica-se, na barra de menus, em **Nova pasta**



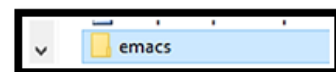
No Windows 10, estando no gerenciador de arquivos, entra-se no disco C:, clique no menu **Início** e em **Nova Pasta**



Feito isto, em qualquer um das versões citadas, ou mesmo mais antigas, uma pasta é aberta:



Nomeie a mesma com o nome **emacs**.



- **BAIXANDO O EMACS PARA WINDOWS**

Existem vários repositórios de onde você pode baixar o emacs.

Dois deles são:

- http://sourceforge.net/projects/emacsbinw64/?source=typ_redirect
- <http://mirror.nbtelecom.com.br/gnu/emacs/windows/>

O **Windows 7** geralmente abre uma janela para que o usuário escolha onde quer baixar o arquivo que vai fazer o download (baixar), já no **Windows 10** o mesmo normalmente baixa para o diretório Downloads (e aí começam os problemas, a falta de padronização).

Se o Windows perguntar onde deseja baixar (Abrir), já se sabe: No **disco C**, dentro do diretório **emacs**.

Se o mesmo for baixado direto para o diretório **Downloads**, sem problemas, é só depois transferir o arquivo baixado para a pasta **emacs** no **disco C**.

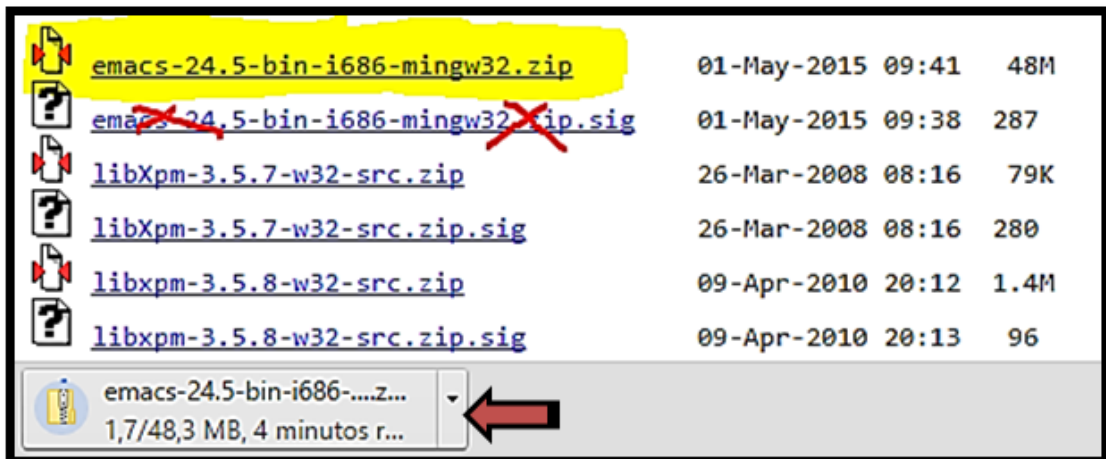
IMPORTANTE: ESCOLHA A ÚLTIMA VERSÃO COM EXTENSÃO .zip E NÃO A .sig⁷⁰.

DOWNLOAD AUTOMÁTICO PARA A PASTA DOWNLOAD, OU OUTRA PASTA PRÉ PROGRAMADA (é possível escolher na configuração)

Suponha a escolha para baixar ter sido no endereço:

- <http://mirror.nbtelecom.com.br/gnu/emacs/windows/>

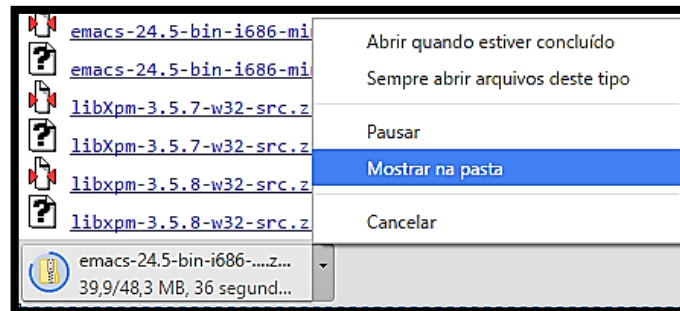
No caso de baixar automático, na parte inferior da página onde estiver dando o download, aparece a mensagem do tipo:



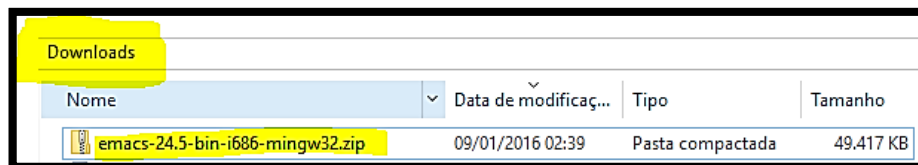
⁷⁰ **ARQUIVOS .sig – sig vem de signature: assinatura.** Os mesmo são arquivos textuais que são lidos de forma diferentes por softwares diferentes. O mesmo não vem, neste caso, com o emacs. O certo é o arquivo **.zip**. Observe os tamanhos de cada um:

emacs-24.5-bin-i686-mingw32.zip	09/01/2016 02:39	Pasta compactada	49.417 KB
emacs-24.5-bin-i686-mingw32.zip.sig	09/01/2016 01:53	Arquivo SIG	1 KB

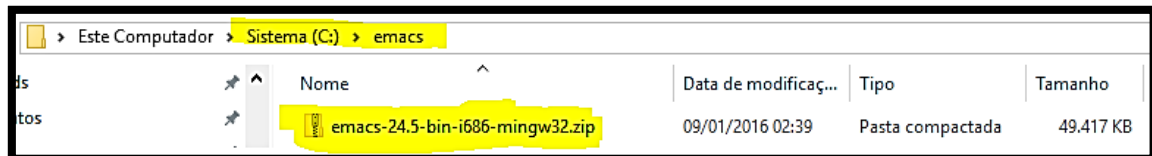
Após o download terminar, clique na setinha ao lado e clique em **Mostrar na pasta**. O sistema operacional abrirá a pasta onde baixou, mesmo que não saiba onde é:



No qual está o arquivo zipado do **emacs**:

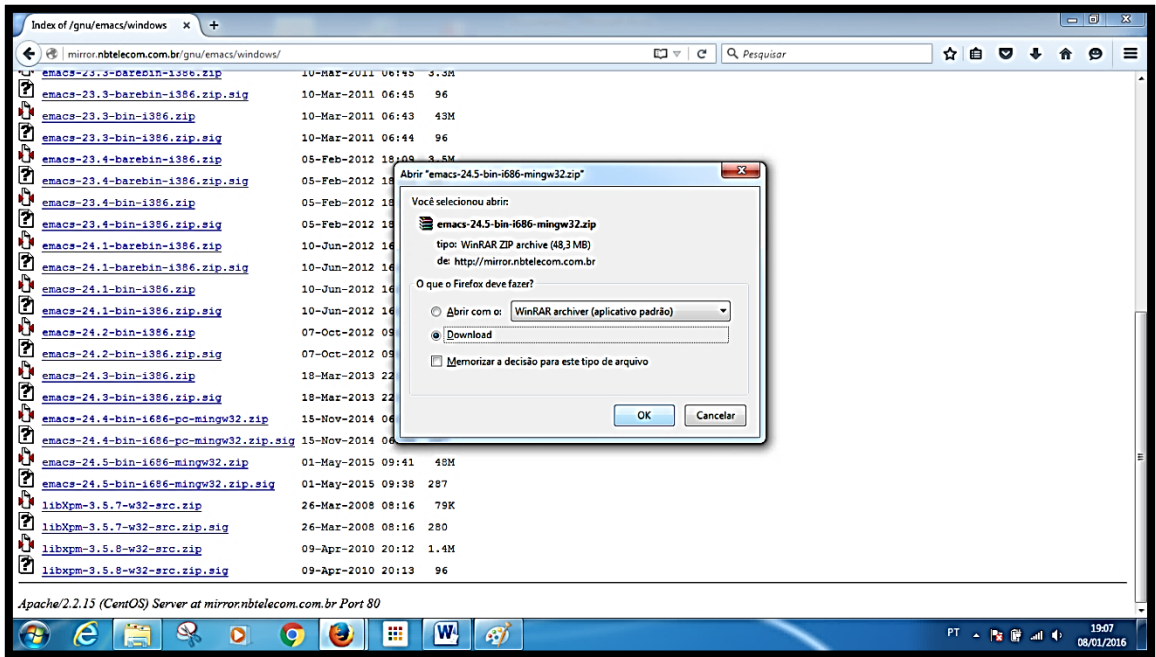


O mesmo deverá ser transferido ou copiado para a pasta **C:\emacs**

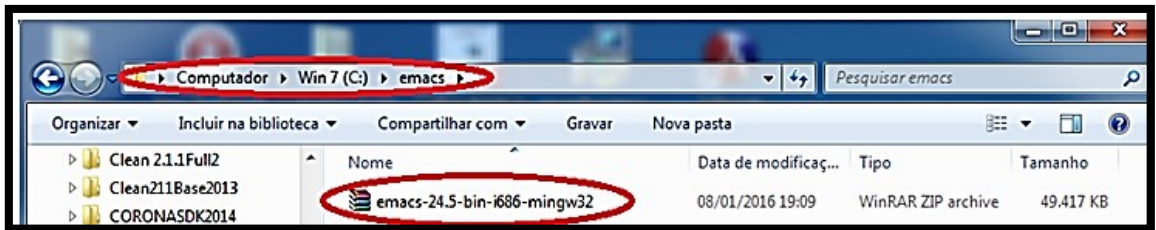


DOWNLOAD COM SOLICITAÇÃO DE EM QUE PASTA BAIXAR.

Nestes casos, como é o padrão do Windows 7 (caso você não tenha alterado) normalmente a seguinte tela aparecerá:

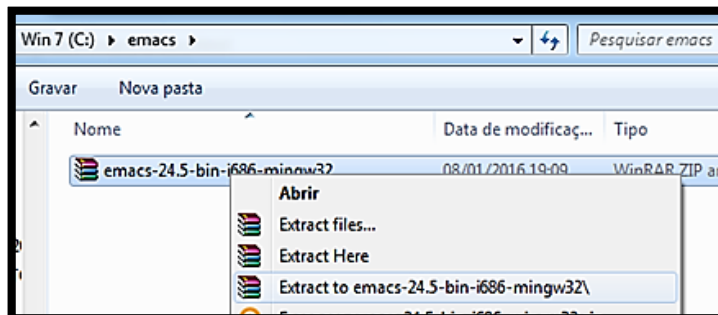


Escolha abrir na pasta C:\emacs

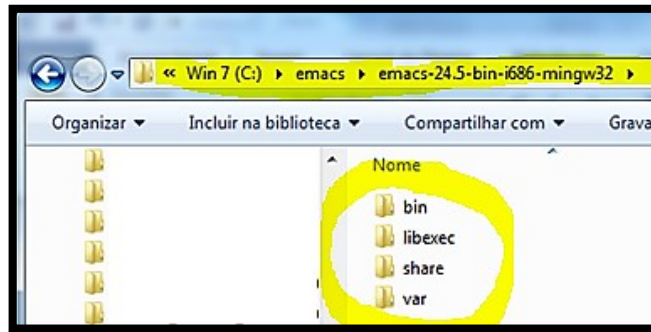


DESCOMPACTANDO O ARQUIVO NA PASTA emacs

Descompacte os arquivos na pasta C:\emacs



Feito isto, tem-se uma pasta chamada emacs-24.5-bin-i686-mingw32. Dentro dela está a pasta bin, libexec, share e var.



VARIÁVEL DE AMBIENTE⁷¹ **HOME**

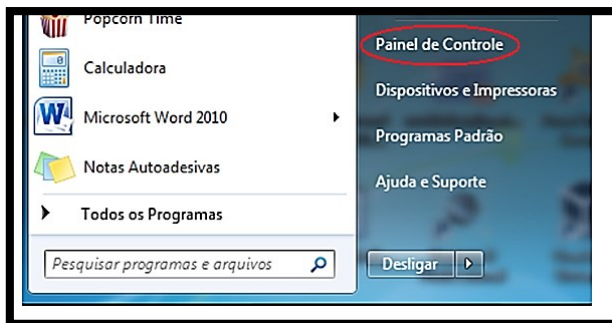
Este é o grande "segredo", por assim dizer, para que se possa trabalhar com o emacs como um profissional do Linux trabalha.

Criando a variável HOME:

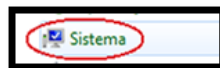
- **NO WINDOWS 7 -> Clique com o botão esquerdo do mouse no ícone do menu **Iniciar****



Clique em **Painel de Controle**



CLIQUE no item **Sistema**

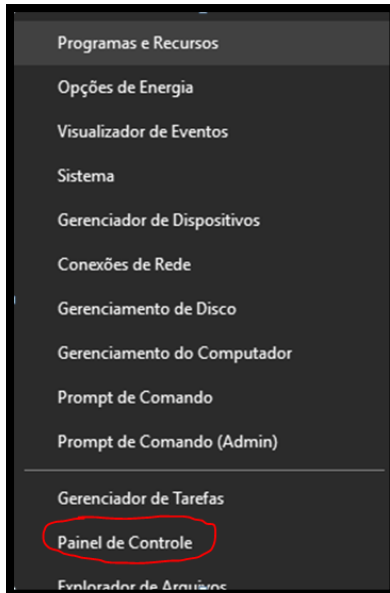


NO WINDOWS 10 -> Clique com o botão direito do mouse no ícone do menu **Iniciar**

⁷¹ **Variável de ambiente:** Uma variável de ambiente é uma variável, um local no computador, que é criada no sistema operacional que poderá ser utilizada por qualquer programa instalado no sistema operacional, uma variável que não pertence apenas ao programa instalado.



Clique em **Painel de Controle**

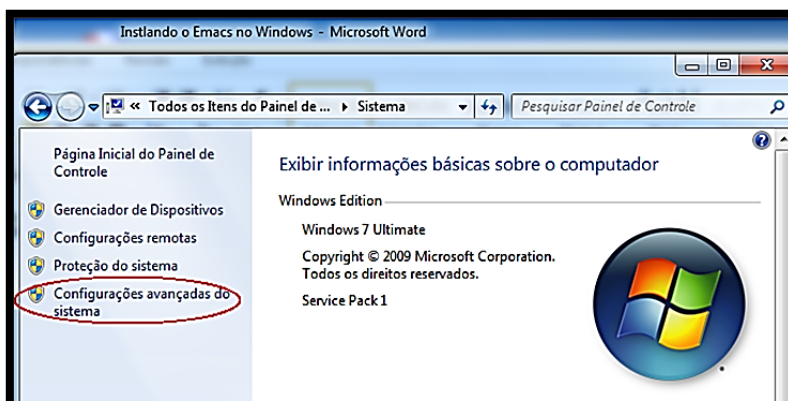


CLIQUE no item **Sistema**

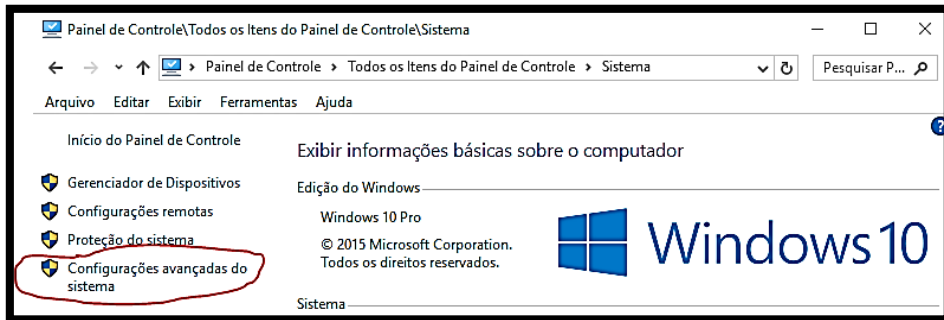


Ao entrar em Sistema, Clique em **Configurações Avançadas do Sistema**

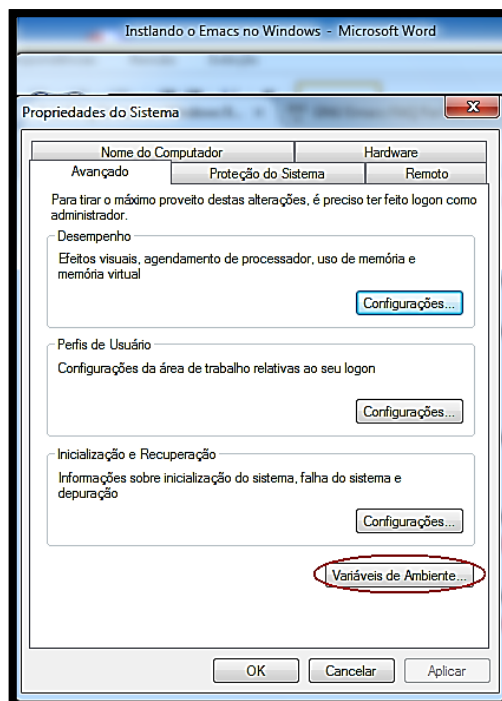
No Windows 7



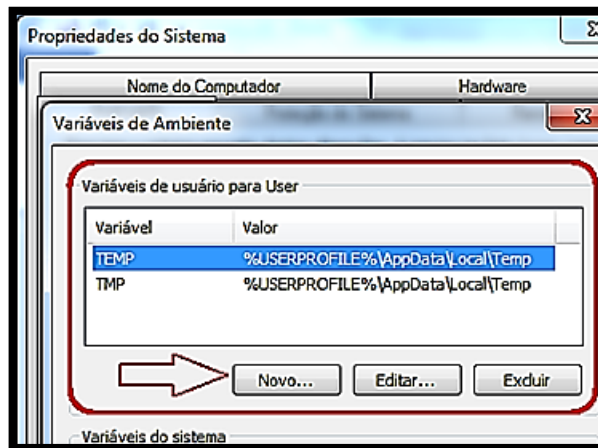
No Windows 10



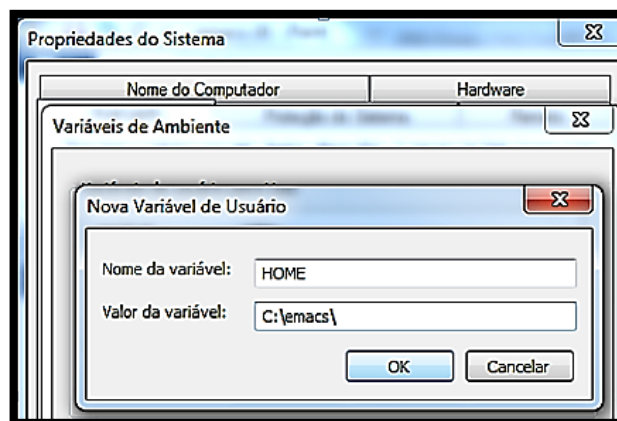
Clique em **Variáveis de Ambiente...** onde você vai criar a variável **HOME**, necessária para um trabalho profissional no **emacs**, quando for necessário personaliza-lo, configurá-lo de forma diferente da padrão com que ele vem (o que geralmente é necessário).



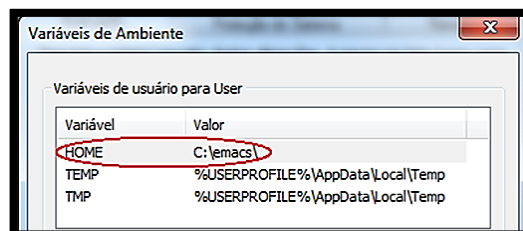
Em **Variáveis de Ambiente** crie uma nova variável chamada **HOME** com o valor **C:\emacs**. Clique em **Novo...** e preencha os campos como as imagens a seguir



Entre com a variável e o valor:



Clique em OK e veja que a variável foi criada.



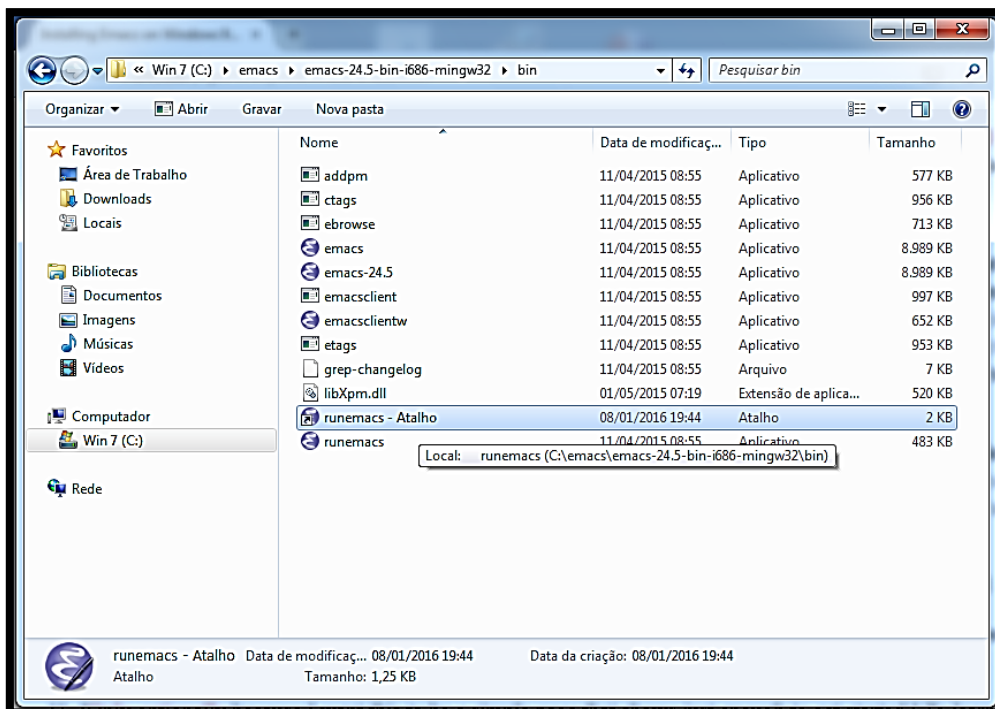
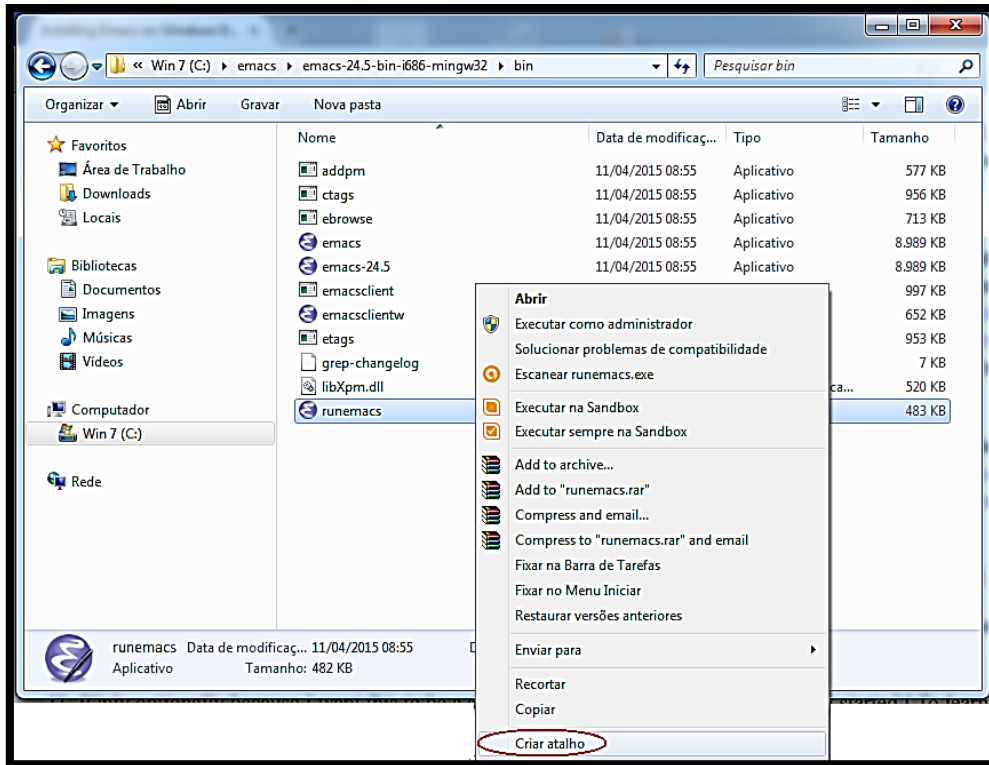
Clique em OK para fechar a janela.

IMPORTANTE!

PARA QUE A INSTALAÇÃO TENHA EFEITO E PARA QUE O SISTEMA OPERACIONAL ASSOCIE A VARIÁVEL HOME AO EMACS, DEVE-SE REINICIAR O COMPUTADOR ANTES DE USAR O EMACS.

APÓS REINICIAR O WINDOWS:

Quando ligar o computador novamente, vá até a pasta `c:\emacs\ emacs-24.5-bin-i686-mingw32\bin`, procure pelo arquivo `runemacs`. Crie um atalho e mova-o para a área de trabalho para acessar facilmente.



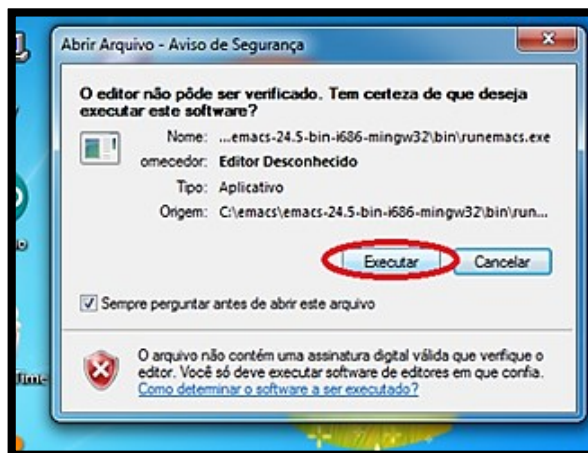
ABRINDO, EXECUTANDO FINALMENTE O EMACS

Seguidas as instruções, o **Emacs** está pronto para ser corretamente executado.

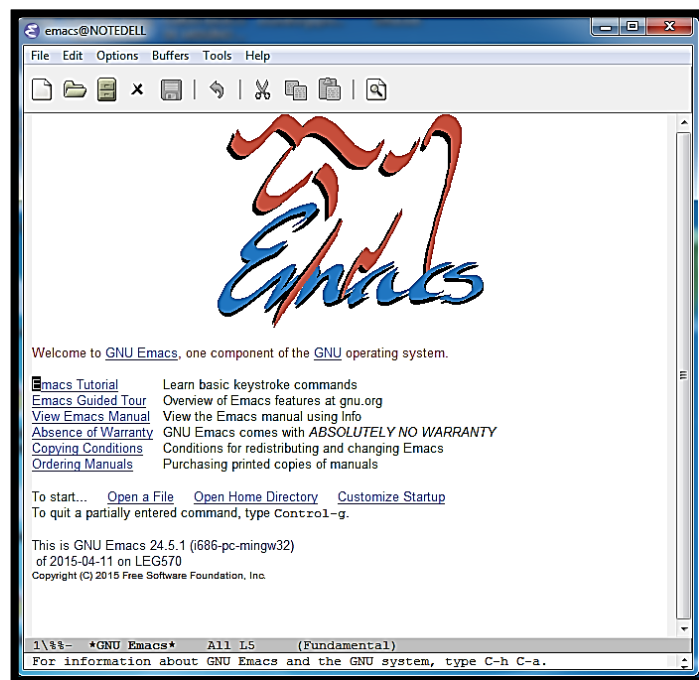
Vá até a **Área de Trabalho(desktop)** e execute o **Emacs**



No **Windows 7** a seguinte informação aparece. Clique em **Executar**.



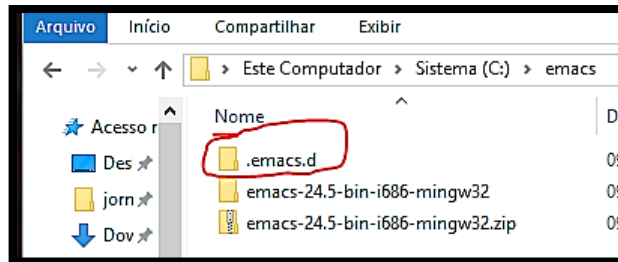
O Emacs é aberto, comprovando que a instalação teve sucesso.



IMPORTANTE: Todos os passos até aqui foram relevantes para que a pasta de configuração do EMACS fosse criada, o que no Linux é algo convencional e normal. Sem isto, a pasta **.emacs.d** não é criada e não se consegue personalizar o emacs para algumas inserções necessárias na criação de páginas multimídia interativas.

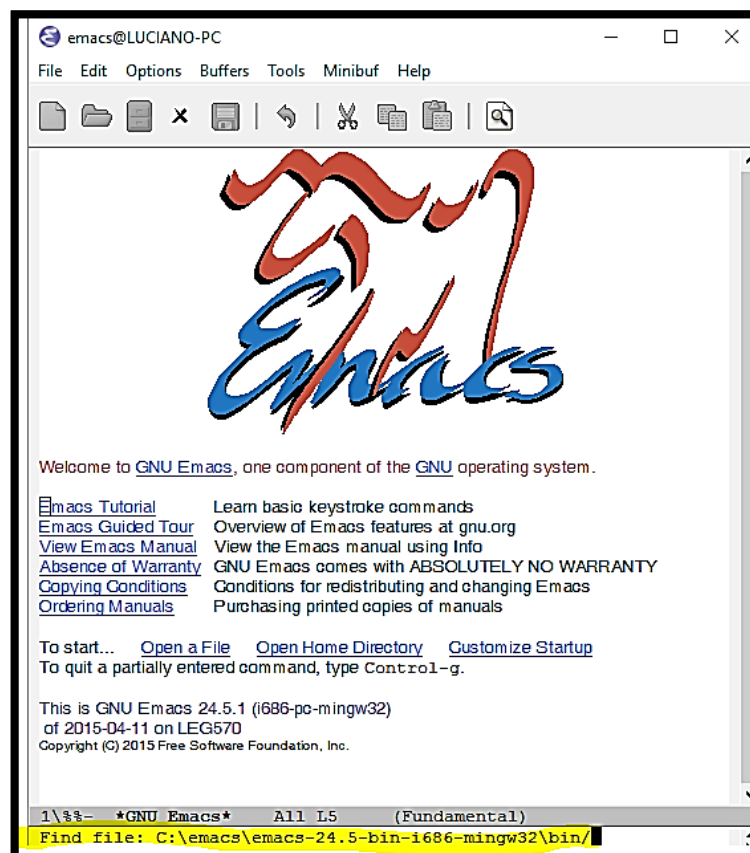
VERIFICANDO QUE A PASTA **.emacs.d** FOI CRIADA

3. Feche o **emacs** e volte na pasta **c:\lemacs**.
4. Verifique que a pasta **.emacs.d** foi criada.



Dentro dela será criado o arquivo de configuração do **emacs**.

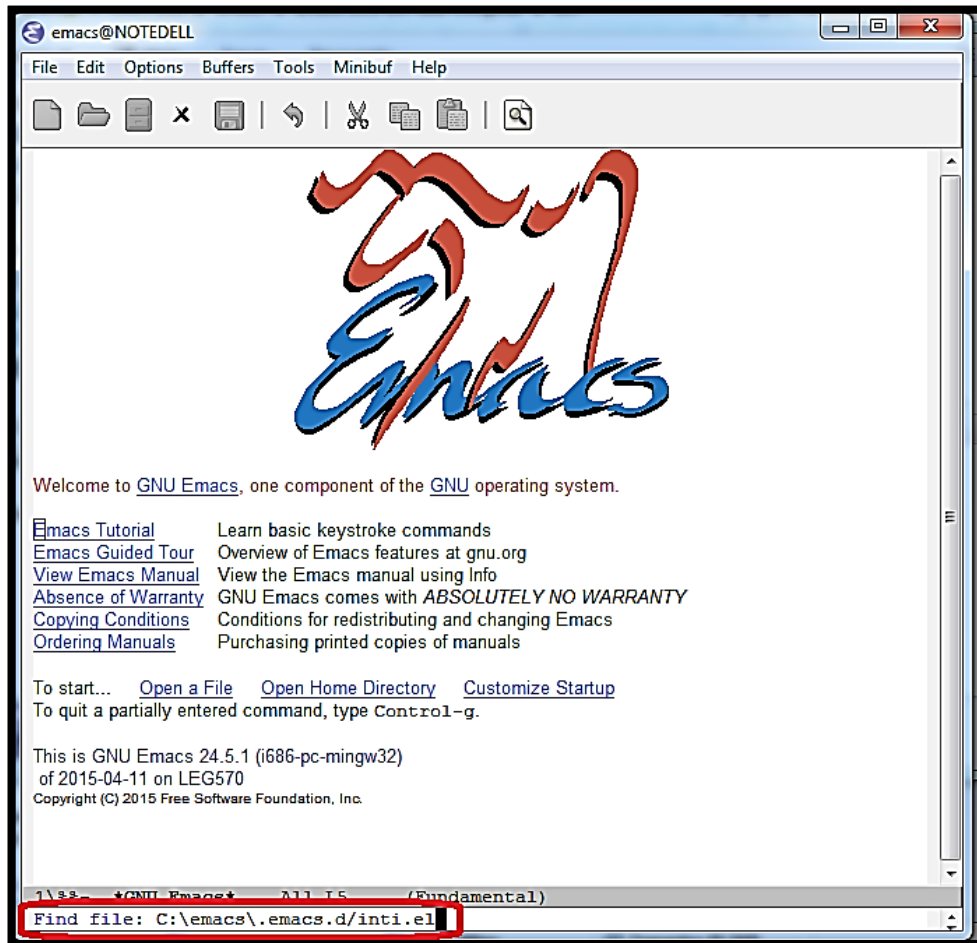
3- Abra o emacs, digite **C-x**⁷². **C-f**⁷³ (onde **C-** é a tecla **Ctrl – control**) A linha de comando é ativada na parte inferior da janela do emacs.



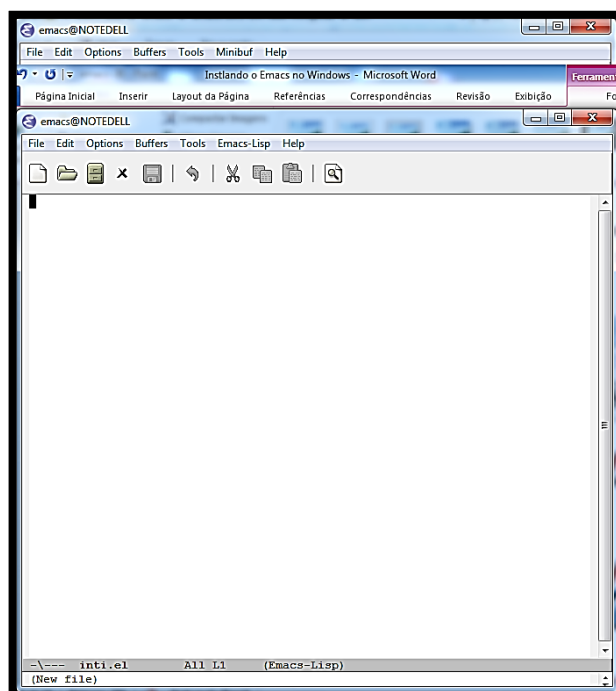
⁷² **C-x** Aperte a tecla **Ctrl** e depois s tecla **X**. São só duas teclas, não tem o traço (-).

⁷³ **C-f** Aperte a tecla **Ctrl** e depois s tecla **f**. São só duas teclas, não tem o traço (-).

4- Apague a linha de comando e digite nela: `c:/emacs\.emacs.d/init.el` e aperte **ENTER**.



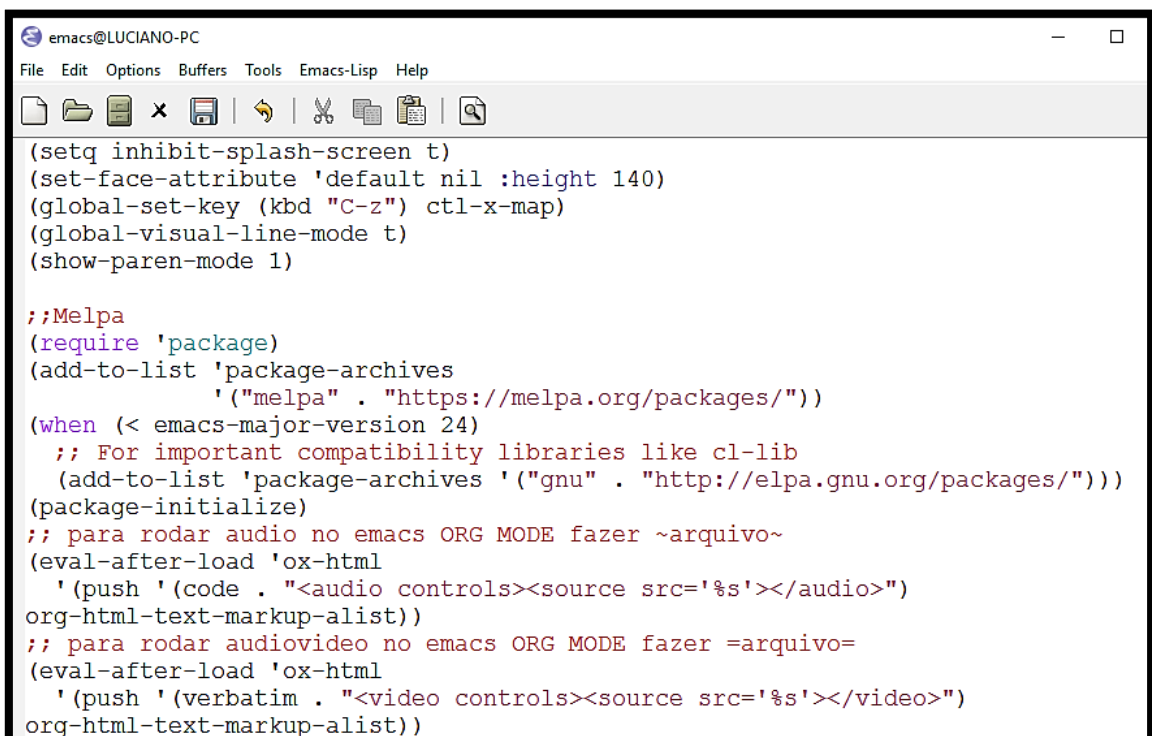
Uma página em branco no **emacs** é iniciada.



Nesta linha deve ser colocada uma configuração padrão, que pode ser sempre esta, sem modificações, portanto, não se precisa entender os comandos inseridos nela. Na mesma está especificando o tamanho da janela, que é para aceitar os caracteres especiais, checagem automática de parênteses e importação de arquivos.

Existem configurações diferentes e que serem para qualquer coisa. Algumas pessoas até postam orgulhosos que configuraram o emacs completamente em apenas 20 minutos. Como já foi dito, o **emacs** não é apenas um editor de texto, é um ambiente de programação multiparadigma completo,

Digite o texto que segue para colocar as configurações básicas e suficientes para os trabalhos a que esta dissertação se propõe.



```

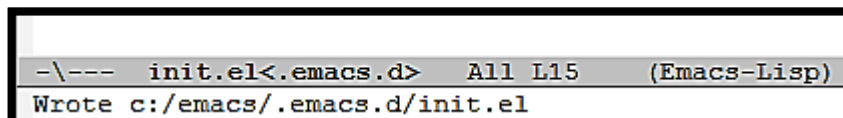
emacs@LUCIANO-PC
File Edit Options Buffers Tools Emacs-Lisp Help
[Icons]
(setq inhibit-splash-screen t)
(set-face-attribute 'default nil :height 140)
(global-set-key (kbd "C-z") ctl-x-map)
(global-visual-line-mode t)
(show-paren-mode 1)

;;Melpa
(require 'package)
(add-to-list 'package-archives
  ("melpa" . "https://melpa.org/packages/"))
(when (< emacs-major-version 24)
  ;; For important compatibility libraries like cl-lib
  (add-to-list 'package-archives ("gnu" . "http://elpa.gnu.org/packages/")))
(package-initialize)
;; para rodar audio no emacs ORG MODE fazer ~arquivo~
(eval-after-load 'ox-html
  '(push '(code . "<audio controls><source src='%s'></audio>")
    org-html-text-markup-alist))
;; para rodar audiovideo no emacs ORG MODE fazer =arquivo=
(eval-after-load 'ox-html
  '(push '(verbatim . "<video controls><source src='%s'></video>")
    org-html-text-markup-alist))

```

Feito isto, salve o arquivo digitando **C-x C-s**,

Vai aparecer a seguinte mensagem na barra de comando:



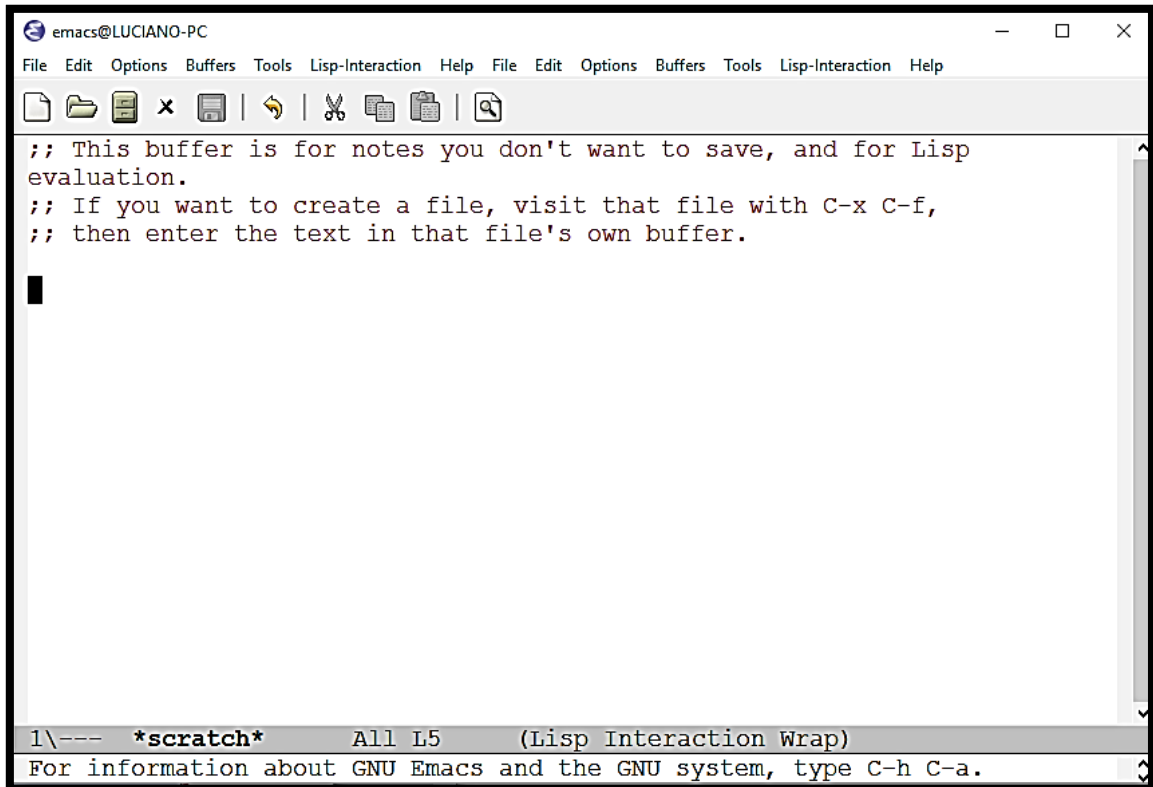
```

-\\--- init.el<.emacs.d> All L15 (Emacs-Lisp)
Wrote c:/emacs/.emacs.d/init.el

```

Feche o **emacs** e abra-o novamente.

Observe que a tela de entrada do emacs mudou.

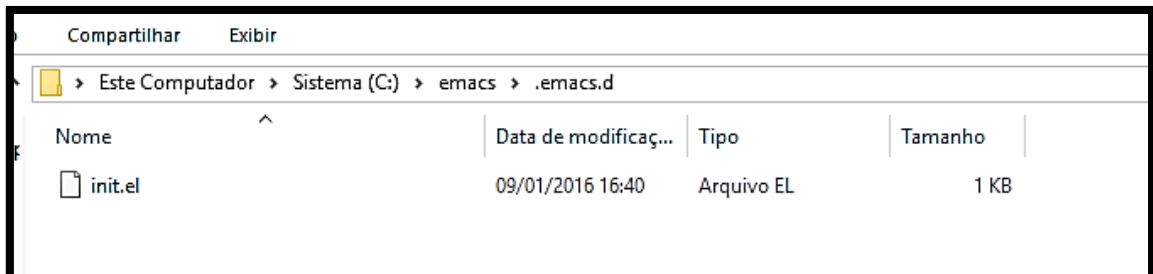


The screenshot shows the Emacs editor window titled "emacs@LUCIANO-PC". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Lisp-Interaction", and "Help". The toolbar contains icons for file operations. The main text area contains the following instructions:

```
;; This buffer is for notes you don't want to save, and for Lisp
evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.
```

The status bar at the bottom shows the current buffer is "1\---- *scratch*" with "All L5" lines and "(Lisp Interaction Wrap)". A footer message reads: "For information about GNU Emacs and the GNU system, type C-h C-a."

O arquivo **init.el** foi criado



The screenshot shows a Windows File Explorer window with the address bar set to "Este Computador > Sistema (C:) > emacs > .emacs.d". The main area displays a table of files:

Nome	Data de modificaç...	Tipo	Tamanho
init.el	09/01/2016 16:40	Arquivo EL	1 KB

Assim, o emacs está configurado e pronto para uso.

ANEXO 2 – DADOS DA HOSPEDAGEM NA NUVEM

QUATRO PARÂMETROS, INFORMAÇÕES NECESSÁRIAS PARA ABRIR UM TERMINAL PUTTY NO WINDOWS (BEM COMO EM QUALQUER OUTRO SISTEMA OPERACIONAL), FAZER TRANSMISSÃO DE DADOS SEGURA ENTRE DESKTOP E NUVEM USANDO SCP.

Antes de executar o **PUTTY** (assim como no Mac e no Linux) é necessário se conhecer 4 (quatro) coisas.

- 1- O **Host Name – domínio** – é o nome do domínio utilizado para contratar sua hospedagem na Web (no caso deste exemplo, é: **coronasdkgames.com.br**)
- 2- Conhecer o **login** da sua hospedagem
- 3- Conhecer o **password** a **senha** de administrador da sua hospedagem.
- 4- A **porta** que a sua hospedagem libera para comunicação **SSH**. No caso conforme já mostrado para a HostGator, a porta é a **2222**.

1- HOST NAME – o host name é o domínio da hospedagem (no caso do exemplo é **coronasdkgames.com.br**).

2 e 3- LOGIN (nome do usuário) e PASSWORD (senha)

Primeiro, quando você contrata uma hospedagem, como na HostGator, por exemplo, você recebe um e-mail, após o pagamento ser aprovado, recebendo duas informações importantes imprescindíveis para acessar a nuvem, na sua hospedagem:

- O **login** – usuário de acesso ao **cPanel** (painel de controle) e **FTP**⁷⁴
- O **password** - a senha de acesso ao **cPanel** e **FTP**

O E-mail: Assunto ->

Bem vindo à HostGator Brasil - Dados do seu Plano

⁷⁴ **FTP** ou **File Transfer Protocol** (em português, Protocolo de Transferência de Arquivos) é uma forma bastante rápida e versátil de transferir arquivos, sendo uma das mais usadas na Internet.

CONTEÚDO DO EMAIL :

Bem vindo à Hostgator [REDACTED]

Você adquiriu um plano M3.

É muito importante que antes de utilizar o plano de hospedagem, você leia os **Termos de Serviço** e **Política de E-mails**. O não cumprimento dos termos, pode acarretar na suspensão temporária ou definitiva do seu plano de hospedagem.

Termos de Serviço: <http://www.hostgator.com.br/tos.shtml>

Políticas de Emails: <http://www.hostgator.com.br/emails.shtml>

Lembrando que para planos compartilhados (P, M, G & Revendas) existe um limite no envio de e-mails, onde você poderá realizar no máximo 500 disparos por hora, com delay de no mínimo 6 segundos, totalizando no máximo o envio de 5.000 mensagens por dia. O não cumprimento poderá acarretar em suspensão temporária ou definitiva da conta.

Prezando pelo seu conforto, a HostGator disponibilizou algumas informações importantes para ajudá-lo na configuração e utilização do seu plano de hospedagem:

<http://hostgator.com.br/primeirospassos.shtml>

- Informações básicas sobre seu plano de hospedagem:

O domínio principal é: coronasdkgames.com.br

Usuário de acesso ao cPanel e FTP: [REDACTED] **login**

Senha de acesso ao cPanel e FTP: [REDACTED] **password**

- Informações sobre as configurações de DNS:

DNS Master: ns64.hostgator.com.br IP Master: 50.97.97.24
 DNS Slave: ns65.hostgator.com.br IP Slave: 50.97.97.25

- O registro de domínio e propagação de DNS pode levar até 24 horas. Neste caso, enquanto o domínio não estiver acessível, navegue pelo painel administrativo por um link provisório que pode ser encontrado abaixo:

Painel Administrativo (cPanel): <http://50.97.97.24/cpanel>
 Visualização do Website:: <http://50.97.97.24/~coron759/>

- Após o período de propagação, utilize o acesso direto:

Painel Administrativo (cPanel): <http://coronasdkgames.com.br/cpanel>
 Visualização do Website: <http://coronasdkgames.com.br>

4- PORTA PARA UTILIZAR A COMUNICAÇÃO SEGURA NA NUVEM UTILIZANDO SSH

Novamente, utilizando o Hostgator como exemplo, e o domínio coronasdkgames.com, o procedimento é o seguinte.

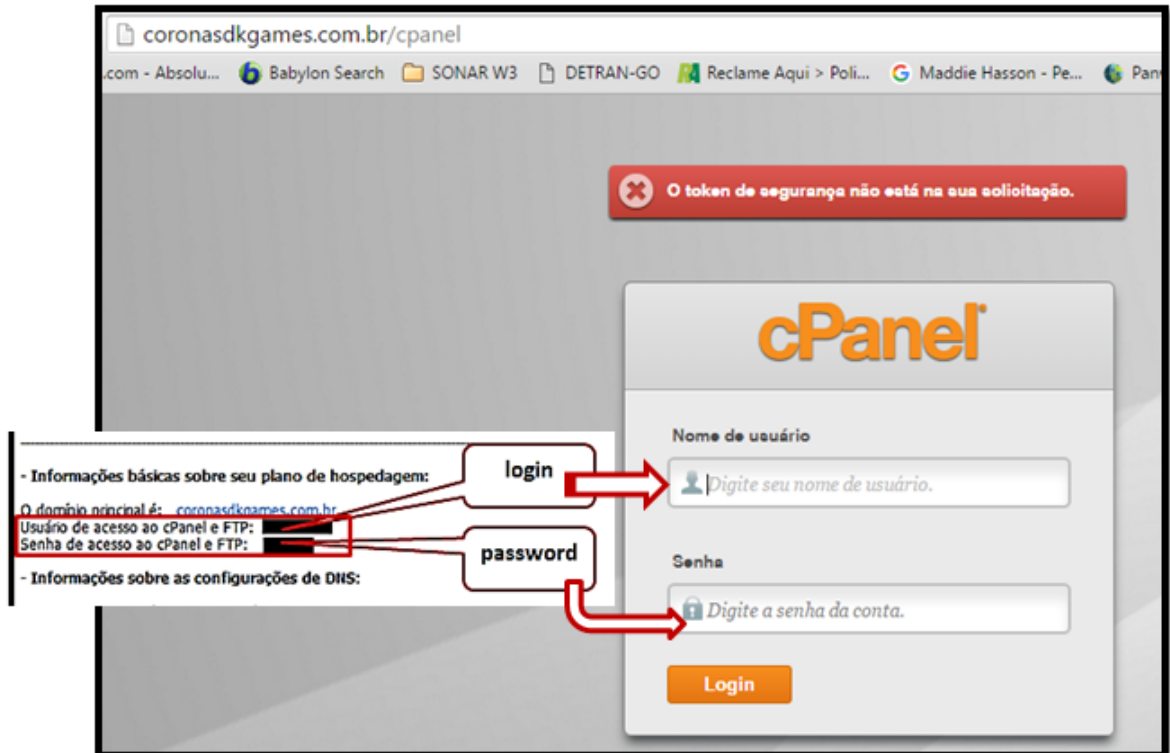
- 5- No email da figura anterior, tem a informação do endereço do Painel Administrativo (cPanel), no caso:

Painel Administrativo (cPanel): <http://coronasdkgames.com.br/cpanel>

- 6- Entre com este endereço no browser de sua escolha

<http://coronasdkgames.com.br/cpanel>

- 7- A janela de login do cPanel é aberta. Entre com Nome do usuário que é o login, o usuário de acesso ao Cpanel mostrado anteriormente e com a senha de acesso ao Cpanel



- 8- O Painel de Controle é aberto. Clique em Ativar/Desativar SSH. Se na sua hospedagem não tiver algo parecido, no campo de busca, digite SSH para localizar onde entrar. Os dados que você precisa estão, no HostGator, em HostGator Links, conforme figura a seguir:



9- Ao clicar em Ativar/Desativar SSH, a janela de ativação do SSH é aberta contendo a informação do número da porta disponibilizada pela Hostgator, ou sua hospedagem, para acessar a porta de comunicação segura SSH, no caso, a 2222.



ANEXO 3 - SSH

O QUE É SSH?

- **SSH** significa: **SECURE SHELL**⁷⁵ -> execução de comandos à distância com segurança.

O **SSH** é um protocolo⁷⁶ de comunicação muito seguro (mesmo em redes inseguras) para se conectar remotamente a um servidor Linux (sim, nativamente Linux), através da internet ou rede local (intranet). Isto significa que você poderá se comunicar com outros computadores, com a nuvem, fazer downloads e uploads com segurança, já que todos os dados trafegados neste protocolo são criptografados com uma criptografia segura.

WINDOWS - No caso do **Putty (Anexo 4)**, permite-se realizar esta comunicação **SSH** na plataforma Windows, ainda bastante utilizada no Brasil em todos os domínios da informação, seja em trabalhos acadêmicos, profissionais e principalmente em atividades informais.

VANTAGENS DO USO DO PROTOCOLO SSH

SSH é sinônimo de SEGURANÇA.

Com o protocolo **SSH** você poderá:

- acessar **remotamente, com segurança**, o servidor através de computadores, notebooks, tablets e celulares;
- acessar o servidor, **remotamente, com segurança**, utilizando a internet ou intranet (rede doméstica);
- acessar o servidor, **remotamente, com segurança**, utilizando modems com fio, wireless, 3G e 4G;
- transferir arquivos, tanto download como upload, **com segurança**, entre servidor e cliente (**Anexo 3**);
- acessar **remotamente, com segurança**, o seu provedor de hospedagem de sites;
- executar **remotamente, com segurança**, aplicações no servidor;
- executar **remotamente, com segurança**, comandos bash⁷⁷ no servidor.

⁷⁵ **Shell** – Um programa, um interpretador, responsável pela comunicação entre o usuário e o computador. O mesmo geralmente, e de preferência, aceita comandos vindos de uma interface de linha de comando (CLI), como no DOS (sistema operacional Windows antes das interfaces visuais, hoje chamado de **prompt de comando – uma janelinha preta** - somente com comandos textuais).

⁷⁶ **PROTOCOLO**: Um conjunto organizado, estruturado de informações, instruções, que garantem que as mesmas serão executadas, interpretadas corretamente, em qualquer lugar onde forem utilizadas.

⁷⁷ **Comandos bash** - O **bash** é um interpretador de **comandos**, uma espécie de tradutor entre o sistema operacional e o usuário, normalmente conhecido como shell, no Linux.

O SSH garante a segurança dos dados transmitidos entre cliente e servidor. Todos os dados são criptografados, inclusive senhas, e, desta forma, mesmo que alguém consiga interceptá-los, a criptografia protege o acesso ao conteúdo da informação.

Cita-se a criptografia do protocolo **SSH** como vantagem, devido aos outros serviços de troca de informação, download e upload, tais como: FTP, Telnet, rLogin, outros, não criptografarem as informações entre cliente e servidor. A criptografia do SSH é segura o bastante para se confiar nas transferências de informação/dados.

ANEXO 4 – TERMINAL SSH NO WINDOWS COM O PuTTY

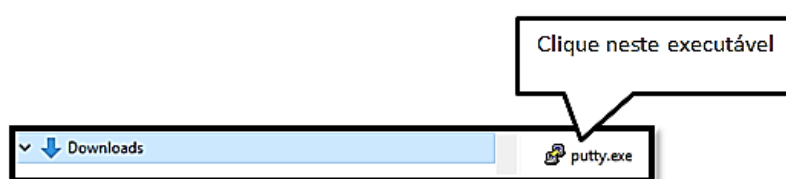
ABRINDO UM TERMINAL SSH COM O PUTTY

O **PUTTY** não precisa de instalação, como já foi dito anteriormente neste capítulo, assim, após clicar no executável baixado, um terminal **SSH** já será aberto para você trabalhar.

De posse das 4 (quatro) informações descritas anteriormente (se não tiver, nem continue, procure-as com quem contratou pra você a hospedagem), você está pronto para iniciar, abrir um terminal **SSH** no Windows, utilizando o **PUTTY**.

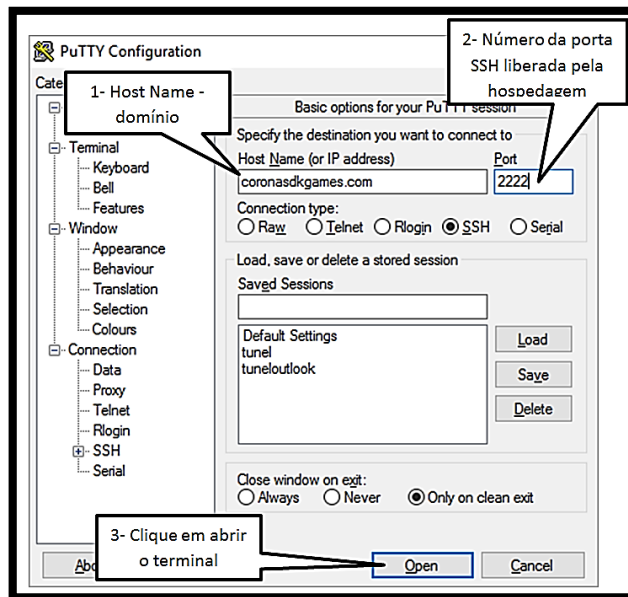
A única coisa difícil, por assim dizer, para utilizar o **PUTTY** e abrir um terminal **SSH** com ele, é conhecer estas 4(quatro) informações: o **hostname (domínio)**, a **porta SSH**, o **login** e o **password**.

Ao clicar no executável do **PUTTY**, que normalmente estará na pasta Download (ou onde você tiver escolhido salvar),



a página de configuração do **PUTTY** é aberta.

Entre com o **Host Name** (o **nome do domínio** de registro na hospedagem) e a porta para comunicação **SSH** liberada pelo servidor.



Ao fazer isto, o terminal é aberto, onde serão solicitadas as outras duas informações:

- primeiro, o **login**,
- depois, o **password**.

Dando tudo certo, o terminal é aberto com o prompt formado da seguinte forma:

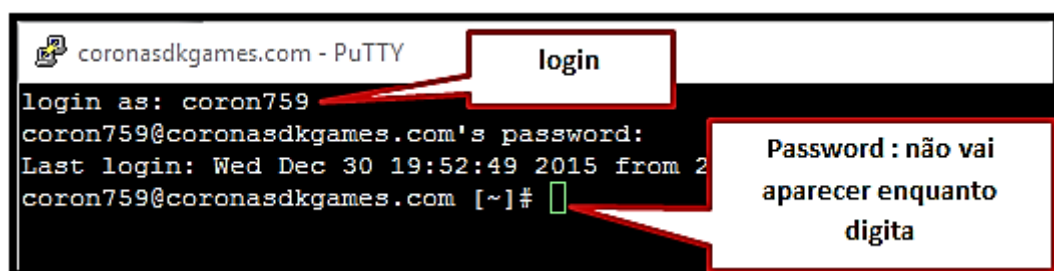
login@domínio [~] #

No caso do exemplo, o login é **coron759** e o domínio **coronasdkgames.com** (já informado ao **PUTTY** na janela de configuração).

Assim, o prompt de comando deverá ficar:

corona759@coronasdkgames.com.br [~]#

Se algo der errado, vier uma mensagem de erro e não esse prompt, feche a janela do terminal e execute o **PUTTY** quantas vezes forem necessárias até que você entre corretamente o login e o password (dê **ENTER** após entrar).



Digite **ls** (**vem de lista, listar**) e clique no botão **ENTER**. Este comando lista todo o conteúdo do que está na **coronasdkgames.com.br** (no caso do exemplo).



```

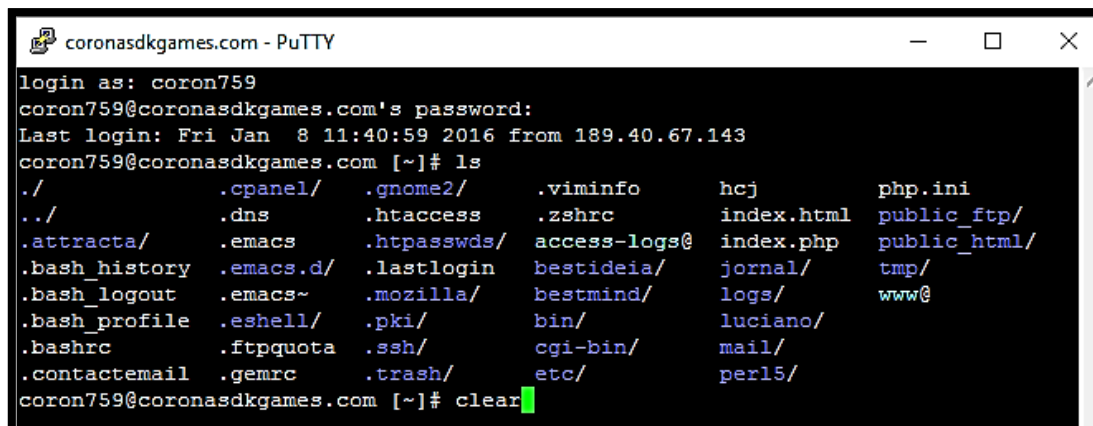
coronasdkgames.com - PuTTY
login as: coron759
coron759@coronasdkgames.com's password:
Last login: Fri Jan 1 19:10:19 2016 from 190.7.180.143
coron759@coronasdkgames.com [~]# ls
./                .cpanel/        .gnome2/        .viminfo        hcj             public_html/
../              .dns            .htaccess       .zshrc          index.html     tmp/
.attracta/      .emacs         .htpasswd/     access-logs@   index.php      www@
.bash_history   .emacs.d/      .lastlogin     bestideia/     logs/
.bash_logout    .emacs~       .mozilla/      bestmind/      mail/
.bash_profile   .eshell/      .pki/          bin/           perl5/
.bashrc         .ftpquota     .ssh/          cgi-bin/       php.ini
.contactemail  .gemrc        .trash/        etc/           public_ftp/
coron759@coronasdkgames.com [~]#

```

Isto ocorrendo, é mais uma confirmação que o terminal está funcionando e que você já está conectado à nuvem, no local de sua hospedagem. Neste momento, não faça nada que possa destruir o conteúdo da mesma, caso saiba executar os comandos do **bach** (**shell**). Nesta linha, você tem direito de fazer tudo. Apagar, renomear, etc... etc...

USANDO O TERMINAL SSH DO PuTTY: COMANDOS BÁSICOS NECESSÁRIOS À ESTA PESQUISA.

Limpando a janela do terminal: basta digitar o comando **clear** e apertar a tecla **ENTER**

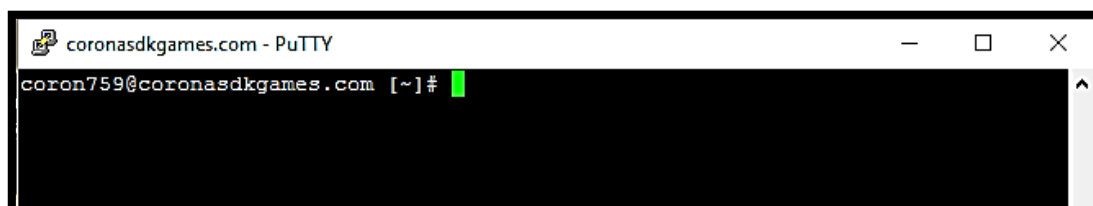


```

coronasdkgames.com - PuTTY
login as: coron759
coron759@coronasdkgames.com's password:
Last login: Fri Jan 8 11:40:59 2016 from 189.40.67.143
coron759@coronasdkgames.com [~]# ls
./                .cpanel/        .gnome2/        .viminfo        hcj             php.ini
../              .dns            .htaccess       .zshrc          index.html     public_ftp/
.attracta/      .emacs         .htpasswd/     access-logs@   index.php      public_html/
.bash_history   .emacs.d/      .lastlogin     bestideia/     jornal/        tmp/
.bash_logout    .emacs~       .mozilla/      bestmind/      logs/          www@
.bash_profile   .eshell/      .pki/          bin/           luciano/
.bashrc         .ftpquota     .ssh/          cgi-bin/       mail/
.contactemail  .gemrc        .trash/        etc/           perl5/
coron759@coronasdkgames.com [~]# clear

```

ENTER



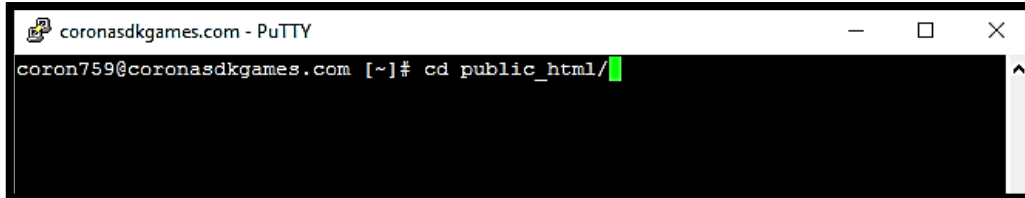
```

coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]#

```

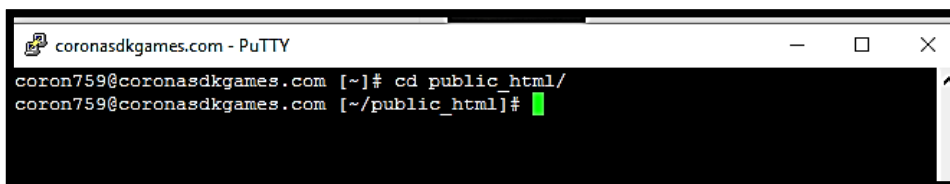
Entrando no diretório **public_html**. Para isto, utilize o comando **cd** de mudar (change) de diretório(directoy).

Digite: **cd public_html/** e aperte a tecla **ENTER**.



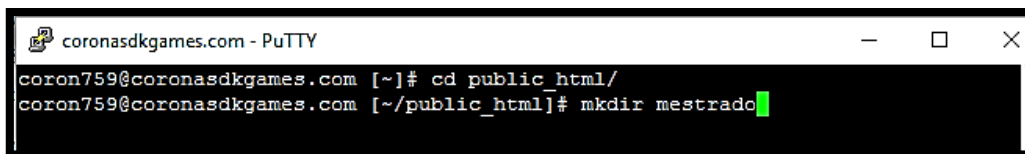
```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
```

ENTER



```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]#
```

Para testar, crie(make) um diretório (directory) de nome **mestrado** dentro de **public_html**. O comando é **mkdir** **mestrado** e aperte a tecla **ENTER**.



```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]# mkdir mestrado
```

ENTER



```
coronasdkgames.com - PuTTY
coron759@coronasdkgames.com [~]# cd public_html/
coron759@coronasdkgames.com [~/public_html]# mkdir mestrado
coron759@coronasdkgames.com [~/public_html]#
```

Observe que não retorna nenhuma mensagem de sucesso ou de erro, apenas o prompt de comando **coron759@coronasdkgames.com [~/public_html]#**, o que significa que a operação foi um sucesso.

Para comprovar que tudo deu certo, digite o comando **ls** e aperte a tecla **ENTER**. Verá que o diretório **mestrado** foi criado.

```

coronasdkgames.com - PuTTY
accountinggateway.php  helcio.html          productimage.php
addons/                hominispress/        products.php
admin/                 includes/            productupdates.php
ajax/                  index.php            programasc.rar
alltut.pdf             init.php             racket/
b.jpg                  internet.html        remote.php
bestideia/            javascript/          rkt/
bestmind/              jornal/              rss.php
biblioteca/           language/            scriptc.html
bottom.php            lib/                 scriptc2.html
brands.php             login.php            search.php
business/             loja2/               sitemap.php
cache/                 loja3/               subscribe.php
camargo.html           mapas2n/             tags.php
captcha.php           mega/                templates/
cart.php               mestrado/            top.php

```

Entre na pasta **mestrado**, digite o comando **ls** e verá que a pasta está criada, mas vazia.

```

compare.php            mypage.k*            wishlist.php
config/                natali.jpg           xml.php
convertMIDI.rar       news.php             xmlsitemap.php
debussy.jpg           noradi/
coron759@coronasdkgames.com [~/public_html]# cd me
mega/
coron759@coronasdkgames.com [~/public_html]# cd mestrado/
coron759@coronasdkgames.com [~/public_html/mestrado]# ls
./ ../
coron759@coronasdkgames.com [~/public_html/mestrado]#

```

Para transitar, mover, entre diretórios, o comando básico, como já visto, é o **cd**.

O comando **cd** pode ser acrescido de outros comandos para realizar algumas tarefas básicas. Como exemplo, parta do diretório do exemplo anterior: o diretório **mestrado**. A partir dele, os comandos para mudar de diretório são:

- Digite **cd /** e aperte a tecla **ENTER** para ir ao diretório raiz.

```

coron759@coronasdkgames.com [~/public_html/mestrado]# cd /
coron759@coronasdkgames.com [/#]#

```

- Digite **cd** e aperte a tecla **ENTER** para ir ao seu diretório pessoal.

```

coron759@coronasdkgames.com [~/public_html/mestrado]# cd
coron759@coronasdkgames.com [~]#

```

- Digite **cd ..** e aperte a tecla **ENTER** para acessar um diretório de nível acima do atual.

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd ..
coron759@coronasdkgames.com [~/public_html]#
```

- Digite **cd** - e aperte a tecla **ENTER** para voltar ao diretório anterior

```
coron759@coronasdkgames.com [~/public_html/mestrado]# cd -
/home/coron759/public_html
coron759@coronasdkgames.com [~/public_html]#
```

OBS. NÃO SE PODE ESQUECER DE DAR UM ESPAÇO ENTRE O COMANDO **cd e o respectivo complemento (/ ou .. ou - ou).**

ANEXO 5 – SERVIDOR E CLIENTE

DESMISTIFICANDO CLIENTE E SERVIDOR.

Antes de qualquer coisa, deve-se entender claramente o que se chama de cliente e servidor. É algo bem mais simples do que parece à primeira vista.

Muitas vezes, quando se refere a **servidor**, imagina-se um grande sistema de computação que realiza, centraliza, centenas de operações e usuários. No caso de **cliente**, sabe-se que somos nós, um dos usuários dos servidores.

Este conceito geralmente causa um certo desconforto para usuários não tão especializados em computação.

Apesar deste conceito estar correto, é importante desmistificá-lo, conceituando de uma forma mais simples:

- **SERVIDOR:**
Máquina⁷⁸ real ou virtual que vai ser acessada pelo usuário remotamente.
- **CLIENTE:**
A sua máquina, ou seja, a máquina que vai acessar outro computador remotamente (o servidor).

Exemplificando:

Imagine a seguinte situação onde você tem uma máquina em sua casa e outra máquina no escritório.

Situação 1- Acessando a máquina do escritório através da máquina de sua casa.

⁷⁸ Computador, celular, tablet ...

- **Cliente:** a máquina de sua casa, a que está acessando a outra máquina remotamente.
- **Servidor:** a máquina do escritório, ou seja, a que está sendo acessada remotamente.

Situação 2- Acessando a máquina de sua casa através da máquina do escritório.

- **Cliente:** a máquina do escritório, a que está acessando a outra máquina remotamente.
- **Servidor:** a máquina de sua casa, ou seja, a que está sendo acessada remotamente.

Observação: As duas máquinas poderiam estar na mesma casa ou no mesmo local. O que diferencia cliente de servidor é quem está sendo acessado (servidor) e quem está acessando (cliente).

ANEXO 6 TRANSFERIR ARQUIVOS ATRAVÉS DE UM TERMINAL PELO COMANDO "SCP" NO LINUX.

Para copiar arquivos de ou para, um servidor SSH, utilizaremos nestes exemplos o *scp* (Secure CoPy) em linha de comando ou seja, através de um terminal tanto em sistemas GNU/Linux, Windows, Android ou IOS. O comando *scp* tem a vantagem de criptografar as informações que vão trafegar na rede, tornando o comando seguro contra invasão de privacidade.

Para isso é preciso entender o conceito de que em sistemas GNU/Linux, todo usuário tem sua pasta particular, localizada em `/home/`. Por exemplo, o usuário "gerente", tem sua pasta particular localizada em `/home/gerente/`. Nenhum outro usuário consegue acesso a esta pasta, exceto o usuário "root", que tem amplo acesso a todo o sistema.

Assim, como você estará logado remotamente como um usuário comum (sem ser o "root"), você não poderá acessar os arquivos contidos na pasta `/home/` dos outros usuários. Portanto se você estiver logado, por exemplo, como "gerente", e quiser privacidade em seus arquivos, salve-os na pasta `/home/gerente/`.

Transferir arquivos do servidor para a máquina local (o Cliente).

Neste exemplo o arquivo a ser copiado, "slack.pdf", está localizado no servidor remoto na pasta privada do usuário `/home/gerente`. Este arquivo será copiado para a máquina local (cliente), na pasta `/home/kncenter`.

Observe que neste comando, o parâmetro **-P1990** (HOST – METAWEB – no HOSTGATOR é **-P2222**) refere-se a porta que foi configurada no servidor e deve ser indicado com a letra "P" maiúscula.

Observe também que logo após o nome do host, é utilizado ":" seguido do endereço do arquivo no servidor.

1. Na máquina local que neste exemplo tem como usuário "kncenter", abra um terminal (como mostrado no [item 6](#)) e digite o comando:
`scp -P1990 gerente@kncenter.ddns.com.br:/home/gerente/slack.pdf /home/kncenter/slack.pdf`

e tecle <Enter>.

```
kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 gerente@kncenter.ddns.com.br:/home/gerente/slack.pdf /home/kncenter/slack.pdf
```

2. Digite a senha e tecle <Enter>.

```
kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 gerente@kncenter.ddns.com.br:/home/gerente/slack.pdf /home/kncenter/slack.pdf
gerente@kncenter.ddns.com.br's password: █
```

3. Observe o indicativo de que o arquivo foi copiado do servidor remoto para a máquina local.

```
kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 gerente@kncenter.ddns.com.br:/home/gerente/slack.pdf /home/kncenter/slack.pdf
gerente@kncenter.ddns.com.br's password:
slack.pdf 100% 0 0.0KB/s 00:00
kncenter@kncenter:~$ █
```

4. Você pode substituir a pasta privada do usuário remoto "/home/gerente" e do usuário local "/home/kncenter", pelo caractere til "~" colocado após os dois pontos. Em sistemas GNU/Linux, o til (~) representa a pasta do usuário.

```

kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 gerente@kncenter.ddns.com.br:~/slack.pdf ~/slack.pdf

```

5. Utilize o parâmetro "-r" para copiar uma pasta inteira. Neste exemplo, o comando copia uma pasta chamada "mid". Observe o til (~) substituindo a pasta home do usuário.

```

kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 -r gerente@kncenter.ddns.com.br:~/mid ~/mid

```

6. Digite a senha, e tecle <Enter>

```

kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 -r gerente@kncenter.ddns.com.br:~/mid ~/mid
gerente@kncenter.ddns.com.br's password:

```

7. Observe que todos os arquivos da pasta "mid" foram copiados.

```

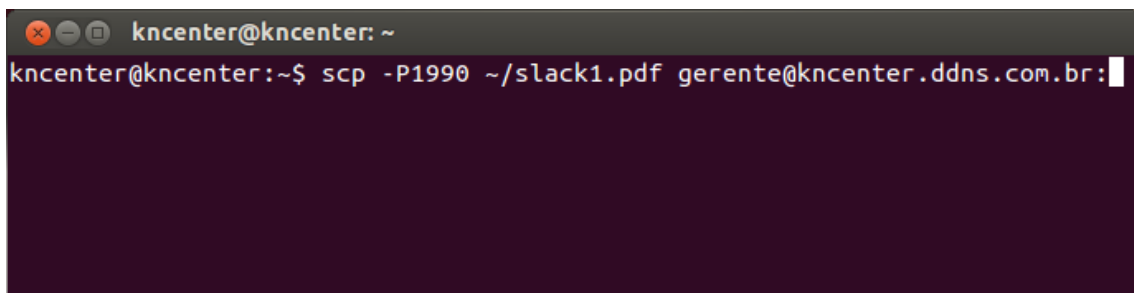
kncenter@kncenter: ~
kncenter@kncenter:~$ scp -P1990 -r gerente@kncenter.ddns.com.br:~/mid ~/mid
gerente@kncenter.ddns.com.br's password:
mid02.mid          100%   0   0.0KB/s   00:00
mid04.mid          100%   0   0.0KB/s   00:00
mid01.mid          100%   0   0.0KB/s   00:00
mid03.mid          100%   0   0.0KB/s   00:00
kncenter@kncenter:~$

```

Transferir arquivos da máquina local (o Cliente) para o servidor.

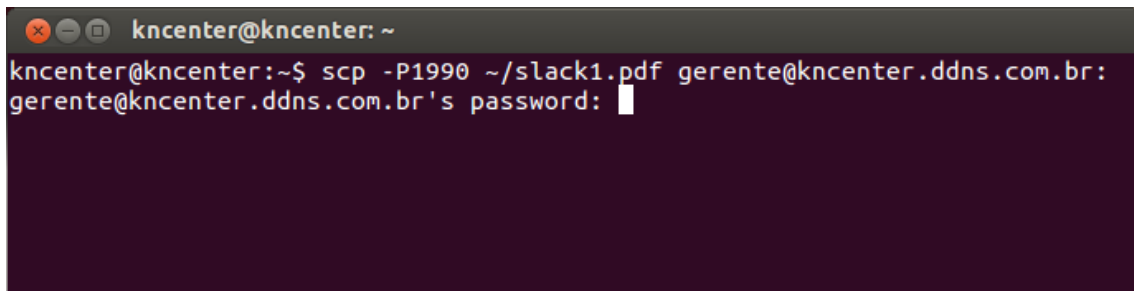
Neste exemplo, o arquivo slack1.pdf, será copiado da máquina local para o servidor remoto. Como o usuário comum, no caso "gerente", não tem os privilégios do "root", o único local que poderá salvar arquivos no servidor, é na pasta privada do usuário, no caso "/home/gerente/", portanto esse caminho não precisa ser digitado, podendo ser substituído pelo caractere ":" após o nome do host.

1. Digite o comando e tecle <Enter>. Observe o caractere ":" colocado no final.



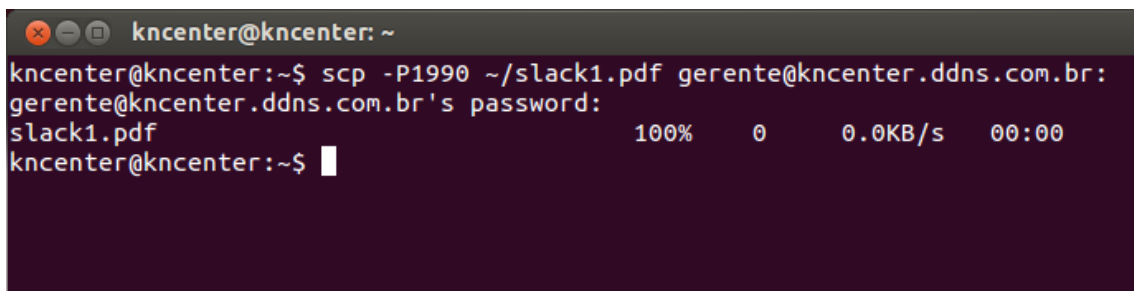
```
kncenter@kncenter: ~  
kncenter@kncenter:~$ scp -P1990 ~/slack1.pdf gerente@kncenter.ddns.com.br:
```

2. Digite a senha, e tecle <Enter>.



```
kncenter@kncenter: ~  
kncenter@kncenter:~$ scp -P1990 ~/slack1.pdf gerente@kncenter.ddns.com.br:  
gerente@kncenter.ddns.com.br's password:
```

3. Observe que o arquivo foi 100% copiado para o servidor remoto.



```
kncenter@kncenter: ~  
kncenter@kncenter:~$ scp -P1990 ~/slack1.pdf gerente@kncenter.ddns.com.br:  
gerente@kncenter.ddns.com.br's password:  
slack1.pdf          100%   0   0.0KB/s   00:00  
kncenter@kncenter:~$
```

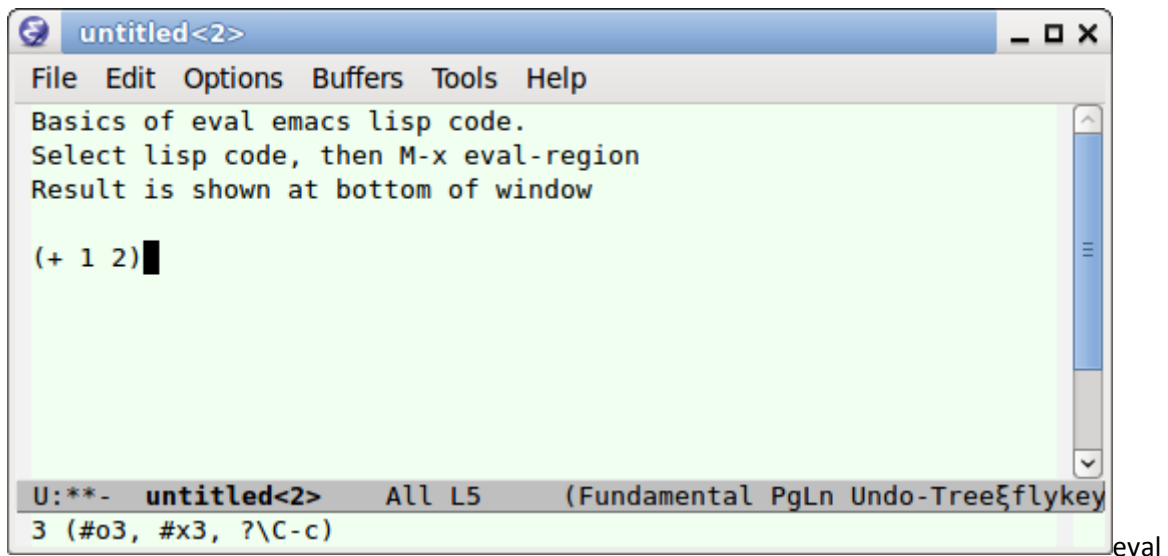
ANEXO 7 EMACS LISP

This is a short, practical, tutorial of Emacs Lisp the language.

To evaluate elisp code, for example, type `(+ 3 4)`, then move your cursor to after the closing parenthesis, then call `eval-last-sexp` `【Ctrl+x Ctrl+e】`. Emacs will evaluate the lisp expression to the left of the cursor. Alternatively, you can select the lisp code, then call `eval-region`.

Alternatively, you can call `ielm`. It will start a interactive elisp command line interface.

To find the inline documentation of a function, call `describe-function` `【Ctrl+h f】`.



emacs lisp basics. [> [Emacs: How to Evaluate Emacs Lisp Code](#)]

PRINTING

```
; printing
(message "hi")

; printing variable values
(message "Her age is: %d" 16) ; %d is for number
(message "Her name is: %s" "Vicky") ; %s is for string
```

```
(message "My list is: %S" (list 8 2 3)) ; %S is for
any lisp expression
```

You can see the output in the buffer named “*Messages*”. You can switch to it by calling `view-echo-area-messages` **【Ctrl+h e】** .

More detail: [Emacs Lisp's print, princ, prin1, format, message.](#)

ARITHMETIC FUNCTIONS

```
(+ 4 5 1) ; ⇒ 10
(- 9 2) ; ⇒ 7
(- 9 2 3) ; ⇒ 4
(* 2 3) ; ⇒ 6
(* 2 3 2) ; ⇒ 12
(/ 7 2) ; ⇒ 3 (integer part of quotient)
(/ 7 2.0) ; ⇒ 3.5
(% 7 4) ; ⇒ 3 (mod, remainder)
(expt 2 3) ; ⇒ 8 (power; exponential)
```

WARNING: single digit decimal number such as `2.` needs a zero after the dot, like this: `2.0`. For example, `(/ 7 2.)` returns 3, not 3.5.

```
;; 3. is a integer, 3.0 is a float
(integerp 3.) ; returns t
(floatp 3.) ; returns nil
(floatp 3.0) ; returns t
```

Function names that end with a “p” often means it return either true or false. (The “p” stands for “predicate”) `t` means true; `nil` means false.

```
(string-to-number "3")
(number-to-string 3)
\(info "\(elisp\) Numbers"\)
```

TRUE & FALSE

In elisp, the symbol `nil` is false, anything else is considered true. Also, `nil` is equivalent to the empty list `()`, so `()` is also false.

```
;; all the following are false. They all evaluate to
"nil"
(if nil "yes" "no") ; => "no"
(if () "yes" "no") ; => "no"
(if '() "yes" "no") ; => "no"
(if (list) "yes" "no") ; => "no", because (list) eval to
a empty list, same as ()
```

By convention, the symbol `t` is used for true.

```
(if t "yes" "no") ; => "yes"
(if 0 "yes" "no") ; => "yes"
(if "" "yes" "no") ; => "yes"
(if [] "yes" "no") ; => "yes". The [] is vector of 0
elements
```

There is no “boolean datatype” in elisp. Just remember that `nil` and empty list `()` are false, anything else is true.

Here's `and` and `or`.

```
(and t nil) ; => nil
(or t nil) ; => t
```

Comparing numbers:

```
(< 3 4) ; less than
(> 3 4) ; greater than
(<= 3 4) ; less or equal to
(>= 3 4) ; greater or equal to

(= 3 3) ; => t
(= 3 3.0) ; => t

(/= 3 4) ; not equal. => t
```

Comparing strings:

```
(string-equal "this" "this") ; => t. Case matters.
```

For generic equality test, use `equal`. It tests if two values have the same datatype and value.

```
;; testing if two values have the same datatype and
value.
(equal "abc" "abc") ; => t
(equal 3 3) ; => t
(equal 3.0 3.0) ; => t
(equal 3 3.0) ; => nil. Because datatype doesn't match.

;; testing equality of lists
(equal '(3 4 5) '(3 4 5)) ; => t
```

```
(equal '(3 4 5) '(3 4 "5")) ; => nil
```

```
;; testing equality of symbols
```

```
(equal 'abc 'abc) ; => t
```

There's also the function `eq`, but don't use it unless you know what you are doing.

To test for inequality, the `/=` is for numbers only, and doesn't work for strings and other lisp data. Use `not` to negate your equality test, like this:

```
(not (= 3 4)) ; => t
```

```
(/= 3 4) ; => t. "/=" is for comparing numbers only
```

```
(not (equal 3 4)) ; => t. General way to test
inequality.
```

[\(info "\(elisp\) Comparison of Numbers"\)](#) ◇ [\(info "\(elisp\) Equality Predicates"\)](#)

EVEN & ODD

```
(= (% n 2) 0) ; test even
```

```
(= (% n 2) 1) ; test odd
```

VARIABLES

GLOBAL VARIABLES

`setq` is used to set variables. Variables need not be declared, and is global.

```
(setq x 1) ; assign 1 to x
```

```
(setq a 3 b 2 c 7) ; assign 3 to a, 2 to b, 7 to c
```


LOCAL VARIABLES

To define local variables, use `let`. The form is: `(let (var1 var2 ...) body)` where *body* is (one or more) lisp expressions. The body's last expression's value is returned.

```
(let (a b)
  (setq a 3)
  (setq b 4)
  (+ a b)
) ; returns 7
```

Another form of `let` is this: `(let ((var1 val1) (var2 val2) ...) body)`. Example:

```
(let ((a 3) (b 4))
  (+ a b)
) ; returns 7
```

This form lets you set values to variable without using many `setq` in the body. This form is convenient if you just have a few simple local vars with known values.

[\(info "\(elisp\) Variables"\)](#)

IF THEN ELSE

The form for “if” expression is: `(if test body)`.

If you want a “else” part, the form is `(if test true body false body)`.

Examples:

```
(if (< 3 2) (message "yes") )
(if (< 3 2) (message "yes") (message "no") )

(if nil (message "yes") (message "no") ) ; prints no
```

[\(info "\(elisp\) Control Structures"\)](#)

If you do not need a “else” part, you should use the function `when` instead, because it is more clear. The form is this: `(when test expr1 expr2 ...)`. Its meaning is the same as `(if test (progn expr1 expr2 ...))`.

BLOCK OF EXPRESSIONS

Sometimes you need to group several expressions together as one single expression. This can be done with `progn`.

```
(progn (message "a") (message "b"))
;; is equivalent to
(message "a") (message "b")
```

The purpose of `(progn ...)` is similar to a block of code `{...}` in C-like languages. It is used to group together a bunch of expressions into one single parenthesized expression. Most of the time it's used inside “if”. For example:

```
(if something
  (progn ; true
    ...
  )
  (progn ; else
    ...
  )
)
```

`progn` returns the last expression in its body.

```
(progn 3 4 ) ; ⇒ 4
\(info "\(elisp\) Sequencing"\)
```

ITERATION

The following code shows a loop using the `while` function.

The form is: `(while test body)`, where *body* is one or more lisp expressions.

```
(setq x 0)

(while (< x 4)
  (print (format "yay %d" x))
  (setq x (1+ x)))
```

[\(info "\(elisp\) Iteration"\)](#)

In the following sample code, it inserts Unicode chars 32 to 126. First, it sets a local variable *x* to 32. Then it starts a `while` loop, insert the corresponding Unicode char (in current buffer), then increase *x* by 1.

```
(let ((x 32))
  (while (< x 127)
    (ucs-insert x)
    (setq x (+ x 1))))
```

Note: There is no “for” loop construct.

BREAK/EXIT A LOOP

See: [Emacs Lisp: Throw & Catch, Exit a Loop.](#)

SEQUENCE, LIST, VECTOR, ARRAY

[Emacs Lisp: Vector](#)

[Emacs Lisp: List](#)

DEFINING A FUNCTION

Basic function definition is of the form: `(defun function name (param1 param2 ...) "`

`doc string" body)` . Example:

```
(defun myFunction () "testing" (message "Yay!") )
```

When a function is called, the last expression in the function's definition body is returned. (there's no "return statement" as do in other languages.)

DEFINING COMMANDS

To make a function available for interactive use, add `(interactive)` right after the doc string.

Evaluate the following code. Then, you can call it by `execute-extended-command` **【Alt+x】**

```
(defun yay ()
  "Insert "Yay!" at cursor position."
  (interactive)
  (insert "Yay!"))
```

The following is a basic function definition, taking one argument from `universal-argument` **【Ctrl+u】**. You can call it by typing **【Ctrl+u 7 Alt+x myFunction】**.

```
(defun myFunction (myArg)
  "Prints the argument"
  (interactive "p")
  (message "Your argument is: %d" myArg)
)
```

The following is a basic function definition taking region as arg. Note the `(interactive "r")`. The "r" is a code that tells emacs that the function will receive the buffer's begin/end text selection positions as its argument.

```
(defun myFunction (myStart myEnd)
  "Prints region start and end positions"
  (interactive "r")
  (message "Region begin at: %d, end at: %d" myStart
myEnd)
```

```
)
```

In summary:

- The `(interactive ...)` clause is a way to make your function interactively callable, and a way to fill out your function's parameters when used interactively.
- A function with the `(interactive ...)` clause is called a **command**, and can be called by `execute-extended-command` (that is, pressing **【Alt+x】**).

The `(interactive "x...")` form takes a single-letter code to indicate how the function gets its arguments from user. There are about 30 codes for `interactive`, but the most useful are the following:

- `(interactive)`, for commands that takes no argument.
- `(interactive "n")`, prompt user for a number as argument. (prompt string can follow right after "n" as part of the string, like this: `(interactive "nWhat is your age?")`.)
- `(interactive "s")`, prompt user for a string as argument.
- `(interactive "r")`, for commands that takes 2 arguments, the beginning and ending positions of the current region. This form is typically used for commands that act on a text selection.

Here is a function definition template that majority of elisp commands follow:

```
(defun myCommand ()
  "One sentence summary of what this command do.
More detailed documentation here."
  (interactive)
  (let (localVar1 localVar2 ...)
    ; do something here ...
    ; ...
    ; last expression is returned
  )
)
```

See also:

- [Emacs Lisp: Function Optional Parameters](#)
- [Emacs Lisp: Inline Docstring Markup](#)

[\(info "\(elisp\) Defining Functions"\)](#) ◇ [\(info "\(elisp\) Defining Commands"\)](#)

THE CONCEPT OF SYMBOLS IN LISP

LISP differs from most programming languages such as C, [Java](#), [Perl](#), [Python](#), in that it deals with symbols, as opposed to just variables and values.

In practice, this means that in lisp, variables can be manipulated in its un-evaluated state. The situation is like the need for the “evaluate” command in many languages, where the programmer can build code as strings and do `evaluate(string)` to achieve meta-programming. In lisp, variable's unevaluated form are always available. You just put a apostrophe in front of it. This is why lisp refers to variable or function names as “symbols”. This makes meta-programming more powerful.

For example, in most languages, once you defined `x=3`, you cannot manipulate the variable “x” because it gets evaluated to 3 right away. If you want, you have to build a string `"x"` and manipulate this string, then finally use something like `evaluate(string)` to achieve the effect. In most languages, the use of `evaluate()` breaks down quickly because the language is not designed for doing it. It's slow, difficult to debug, and there lacks many facilities for such meta programming.

The ability to meta-program has many applications. For example, when you need to take user input as code (such as math formulas), or need to manipulate math expressions, or writing programs that modify itself at run-time. Lisp macros is a direct result of the ability to manipulate symbols. [\(info "\(elisp\) Macros"\)](#)

Lisp symbol is a advanced topic. But if you are curious, you can read this short intro: [Emacs Lisp Symbol](#).

Others:

[Overview of Text-Processing in Emacs Lisp](#)

[Emacs Lisp Examples](#)

[Tips for Editing Lisp Code with Emacs](#)

ANEXO 8- ORG-MODE – MANUAL

Manual ofical [Org-mode](#), com sumário completo e coletânea apenas das principais funções e recurso do mesmo.

Table of Contents: <http://orgmode.org/manual/index.html#Top>

The Org Manual

Short Contents

[Org Mode Manual](#)

[1 Introduction](#)

[2 Document structure](#)

[3 Tables](#)

[4 Hyperlinks](#)

[5 TODO items](#)

[6 Tags](#)

[7 Properties and columns](#)

[8 Dates and times](#)

[9 Capture – Refile – Archive](#)

[10 Agenda views](#)

[11 Markup for rich export](#)

[12 Exporting](#)

[13 Publishing](#)

[14 Working with source code](#)

[15 Miscellaneous](#)

[Appendix A Hacking](#)

[Appendix B MobileOrg](#)

[Appendix C History and acknowledgments](#)

[Appendix D GNU Free Documentation License](#)

[Concept index](#)

[Key index](#)

[Command and function index](#)

[Variable index](#)

Table of Contents

[Org Mode Manual](#)

[1 Introduction](#)

[1.1 Summary](#)

- [1.2 Installation](#)
- [1.3 Activation](#)
- [1.4 Feedback](#)
- [1.5 Typesetting conventions used in this manual](#)
- [2 Document structure](#)
 - [2.1 Outlines](#)
 - [2.2 Headlines](#)
 - [2.3 Visibility cycling](#)
 - [2.3.1 Global and local cycling](#)
 - [2.3.2 Initial visibility](#)
 - [2.3.3 Catching invisible edits](#)
 - [2.4 Motion](#)
 - [2.5 Structure editing](#)
 - [2.6 Sparse trees](#)
 - [2.7 Plain lists](#)
 - [2.8 Drawers](#)
 - [2.9 Blocks](#)
 - [2.10 Footnotes](#)
 - [2.11 The Orgstruct minor mode](#)
 - [2.12 Org syntax](#)
- [3 Tables](#)
 - [3.1 The built-in table editor](#)
 - [3.2 Column width and alignment](#)
 - [3.3 Column groups](#)
 - [3.4 The Orgtbl minor mode](#)
 - [3.5 The spreadsheet](#)
 - [3.5.1 References](#)
 - [3.5.2 Formula syntax for Calc](#)
 - [3.5.3 Emacs Lisp forms as formulas](#)
 - [3.5.4 Durations and time values](#)
 - [3.5.5 Field and range formulas](#)
 - [3.5.6 Column formulas](#)
 - [3.5.7 Lookup functions](#)
 - [3.5.8 Editing and debugging formulas](#)
 - [3.5.9 Updating the table](#)
 - [3.5.10 Advanced features](#)
 - [3.6 Org-Plot](#)
- [4 Hyperlinks](#)
 - [4.1 Link format](#)
 - [4.2 Internal links](#)
 - [4.2.1 Radio targets](#)
 - [4.3 External links](#)
 - [4.4 Handling links](#)
 - [4.5 Using links outside Org](#)
 - [4.6 Link abbreviations](#)

- [4.7 Search options in file links](#)
- [4.8 Custom Searches](#)
- [5 TODO items](#)
 - [5.1 Basic TODO functionality](#)
 - [5.2 Extended use of TODO keywords](#)
 - [5.2.1 TODO keywords as workflow states](#)
 - [5.2.2 TODO keywords as types](#)
 - [5.2.3 Multiple keyword sets in one file](#)
 - [5.2.4 Fast access to TODO states](#)
 - [5.2.5 Setting up keywords for individual files](#)
 - [5.2.6 Faces for TODO keywords](#)
 - [5.2.7 TODO dependencies](#)
 - [5.3 Progress logging](#)
 - [5.3.1 Closing items](#)
 - [5.3.2 Tracking TODO state changes](#)
 - [5.3.3 Tracking your habits](#)
 - [5.4 Priorities](#)
 - [5.5 Breaking tasks down into subtasks](#)
 - [5.6 Checkboxes](#)
- [6 Tags](#)
 - [6.1 Tag inheritance](#)
 - [6.2 Setting tags](#)
 - [6.3 Tag hierarchy](#)
 - [6.4 Tag searches](#)
- [7 Properties and columns](#)
 - [7.1 Property syntax](#)
 - [7.2 Special properties](#)
 - [7.3 Property searches](#)
 - [7.4 Property Inheritance](#)
 - [7.5 Column view](#)
 - [7.5.1 Defining columns](#)
 - [7.5.1.1 Scope of column definitions](#)
 - [7.5.1.2 Column attributes](#)
 - [7.5.2 Using column view](#)
 - [7.5.3 Capturing column view](#)
 - [7.6 The Property API](#)
- [8 Dates and times](#)
 - [8.1 Timestamps, deadlines, and scheduling](#)
 - [8.2 Creating timestamps](#)
 - [8.2.1 The date/time prompt](#)
 - [8.2.2 Custom time format](#)
 - [8.3 Deadlines and scheduling](#)
 - [8.3.1 Inserting deadlines or schedules](#)
 - [8.3.2 Repeated tasks](#)
 - [8.4 Clocking work time](#)

- [8.4.1 Clocking commands](#)
- [8.4.2 The clock table](#)
- [8.4.3 Resolving idle time and continuous clocking](#)
- [8.5 Effort estimates](#)
- [8.6 Taking notes with a timer](#)
- [9 Capture – Refile – Archive](#)
- [9.1 Capture](#)
 - [9.1.1 Setting up capture](#)
 - [9.1.2 Using capture](#)
 - [9.1.3 Capture templates](#)
 - [9.1.3.1 Template elements](#)
 - [9.1.3.2 Template expansion](#)
 - [9.1.3.3 Templates in contexts](#)
- [9.2 Attachments](#)
- [9.3 RSS feeds](#)
- [9.4 Protocols for external access](#)
- [9.5 Refile and copy](#)
- [9.6 Archiving](#)
 - [9.6.1 Moving a tree to the archive file](#)
 - [9.6.2 Internal archiving](#)
- [10 Agenda views](#)
- [10.1 Agenda files](#)
- [10.2 The agenda dispatcher](#)
- [10.3 The built-in agenda views](#)
 - [10.3.1 The weekly/daily agenda](#)
 - [10.3.2 The global TODO list](#)
 - [10.3.3 Matching tags and properties](#)
 - [10.3.4 Timeline for a single file](#)
 - [10.3.5 Search view](#)
 - [10.3.6 Stuck projects](#)
- [10.4 Presentation and sorting](#)
 - [10.4.1 Categories](#)
 - [10.4.2 Time-of-day specifications](#)
 - [10.4.3 Sorting agenda items](#)
 - [10.4.4 Filtering/limiting agenda items](#)
- [10.5 Commands in the agenda buffer](#)
- [10.6 Custom agenda views](#)
 - [10.6.1 Storing searches](#)
 - [10.6.2 Block agenda](#)
 - [10.6.3 Setting options for custom commands](#)
- [10.7 Exporting agenda views](#)
- [10.8 Using column view in the agenda](#)
- [11 Markup for rich export](#)
 - [11.1 Structural markup elements](#)
 - [11.2 Images and Tables](#)

- [11.3 Literal examples](#)
- [11.4 Include files](#)
- [11.5 Index entries](#)
- [11.6 Macro replacement](#)
- [11.7 Embedded LaTeX](#)
 - [11.7.1 Special symbols](#)
 - [11.7.2 Subscripts and superscripts](#)
 - [11.7.3 LaTeX fragments](#)
 - [11.7.4 Previewing LaTeX fragments](#)
 - [11.7.5 Using C_DLaTeX to enter math](#)
- [11.8 Special blocks](#)
- [12 Exporting](#)
 - [12.1 The export dispatcher](#)
 - [12.2 Export back-ends](#)
 - [12.3 Export settings](#)
 - [12.4 ASCII/Latin-1/UTF-8 export](#)
 - [12.5 Beamer export](#)
 - [12.5.1 Beamer export commands](#)
 - [12.5.2 Beamer specific export settings](#)
 - [12.5.3 Sectioning, Frames and Blocks in Beamer](#)
 - [12.5.4 Beamer specific syntax](#)
 - [12.5.5 Editing support](#)
 - [12.5.6 A Beamer example](#)
 - [12.6 HTML export](#)
 - [12.6.1 HTML export commands](#)
 - [12.6.2 HTML Specific export settings](#)
 - [12.6.3 HTML doctypes](#)
 - [12.6.4 HTML preamble and postamble](#)
 - [12.6.5 Quoting HTML tags](#)
 - [12.6.6 Links in HTML export](#)
 - [12.6.7 Tables in HTML export](#)
 - [12.6.8 Images in HTML export](#)
 - [12.6.9 Math formatting in HTML export](#)
 - [12.6.10 Text areas in HTML export](#)
 - [12.6.11 CSS support](#)
 - [12.6.12 JavaScript supported display of web pages](#)
 - [12.7 LaTeX and PDF export](#)
 - [12.7.1 LaTeX export commands](#)
 - [12.7.2 LaTeX specific export settings](#)
 - [12.7.3 Header and sectioning structure](#)
 - [12.7.4 Quoting LaTeX code](#)
 - [12.7.5 LaTeX specific attributes](#)
 - [12.8 Markdown export](#)
 - [12.9 OpenDocument Text export](#)
 - [12.9.1 Pre-requisites for ODT export](#)

- [12.9.2 ODT export commands](#)
- [12.9.3 ODT specific export settings](#)
- [12.9.4 Extending ODT export](#)
- [12.9.5 Applying custom styles](#)
- [12.9.6 Links in ODT export](#)
- [12.9.7 Tables in ODT export](#)
- [12.9.8 Images in ODT export](#)
- [12.9.9 Math formatting in ODT export](#)
- [12.9.10 Labels and captions in ODT export](#)
- [12.9.11 Literal examples in ODT export](#)
- [12.9.12 Advanced topics in ODT export](#)
- [12.10 Org export](#)
- [12.11 Texinfo export](#)
 - [12.11.1 Texinfo export commands](#)
 - [12.11.2 Texinfo specific export settings](#)
 - [12.11.3 Document preamble](#)
 - [12.11.4 Headings and sectioning structure](#)
 - [12.11.5 Indices](#)
 - [12.11.6 Quoting Texinfo code](#)
 - [12.11.7 Texinfo specific attributes](#)
 - [12.11.8 An example](#)
- [12.12 iCalendar export](#)
- [12.13 Other built-in back-ends](#)
- [12.14 Export in foreign buffers](#)
- [12.15 Advanced configuration](#)
- [13 Publishing](#)
 - [13.1 Configuration](#)
 - [13.1.1 The variable `org-publish-project-alist`](#)
 - [13.1.2 Sources and destinations for files](#)
 - [13.1.3 Selecting files](#)
 - [13.1.4 Publishing action](#)
 - [13.1.5 Options for the exporters](#)
 - [13.1.6 Links between published files](#)
 - [13.1.7 Generating a sitemap](#)
 - [13.1.8 Generating an index](#)
 - [13.2 Uploading files](#)
 - [13.3 Sample configuration](#)
 - [13.3.1 Example: simple publishing configuration](#)
 - [13.3.2 Example: complex publishing configuration](#)
 - [13.4 Triggering publication](#)
- [14 Working with source code](#)
 - [14.1 Structure of code blocks](#)
 - [14.2 Editing source code](#)
 - [14.3 Exporting code blocks](#)
 - [14.4 Extracting source code](#)

- [14.5 Evaluating code blocks](#)
- [14.6 Library of Babel](#)
- [14.7 Languages](#)
- [14.8 Header arguments](#)
 - [14.8.1 Using header arguments](#)
 - [14.8.2 Specific header arguments](#)
 - [14.8.2.1 :var](#)
 - [14.8.2.2 :results](#)
 - [14.8.2.3 :file](#)
 - [14.8.2.4 :file-desc](#)
 - [14.8.2.5 :file-ext](#)
 - [14.8.2.6 :output-dir](#)
 - [14.8.2.7 :dir and remote execution](#)
 - [14.8.2.8 :exports](#)
 - [14.8.2.9 :tangle](#)
 - [14.8.2.10 :mkdirp](#)
 - [14.8.2.11 :comments](#)
 - [14.8.2.12 :padline](#)
 - [14.8.2.13 :no-expand](#)
 - [14.8.2.14 :session](#)
 - [14.8.2.15 :noweb](#)
 - [14.8.2.16 :noweb-ref](#)
 - [14.8.2.17 :noweb-sep](#)
 - [14.8.2.18 :cache](#)
 - [14.8.2.19 :sep](#)
 - [14.8.2.20 :hlines](#)
 - [14.8.2.21 :colnames](#)
 - [14.8.2.22 :rownames](#)
 - [14.8.2.23 :shebang](#)
 - [14.8.2.24 :tangle-mode](#)
 - [14.8.2.25 :eval](#)
 - [14.8.2.26 :wrap](#)
 - [14.8.2.27 :post](#)
 - [14.8.2.28 :prologue](#)
 - [14.8.2.29 :epilogue](#)
- [14.9 Results of evaluation](#)
 - [14.9.1 Non-session](#)
 - [14.9.1.1 :results value](#)
 - [14.9.1.2 :results output](#)
 - [14.9.2 Session](#)
 - [14.9.2.1 :results value](#)
 - [14.9.2.2 :results output](#)
- [14.10 Noweb reference syntax](#)
- [14.11 Key bindings and useful functions](#)
- [14.12 Batch execution](#)

- [15 Miscellaneous](#)
- [15.1 Completion](#)
- [15.2 Easy templates](#)
- [15.3 Speed keys](#)
- [15.4 Code evaluation and security issues](#)
- [15.5 Customization](#)
- [15.6 Summary of in-buffer settings](#)
- [15.7 The very busy C-c C-c key](#)
- [15.8 A cleaner outline view](#)
- [15.9 Using Org on a tty](#)
- [15.10 Interaction with other packages](#)
- [15.10.1 Packages that Org cooperates with](#)
- [15.10.2 Packages that lead to conflicts with Org mode](#)
- [15.11 org-crypt.el](#)
- [Appendix A Hacking](#)
- [A.1 Hooks](#)
- [A.2 Add-on packages](#)
- [A.3 Adding hyperlink types](#)
- [A.4 Adding export back-ends](#)
- [A.5 Context-sensitive commands](#)
- [A.6 Tables and lists in arbitrary syntax](#)
- [A.6.1 Radio tables](#)
- [A.6.2 A LaTeX example of radio tables](#)
- [A.6.3 Translator functions](#)
- [A.6.4 Radio lists](#)
- [A.7 Dynamic blocks](#)
- [A.8 Special agenda views](#)
- [A.9 Speeding up your agendas](#)
- [A.10 Extracting agenda information](#)
- [A.11 Using the property API](#)
- [A.12 Using the mapping API](#)
- [Appendix B MobileOrg](#)
- [B.1 Setting up the staging area](#)
- [B.2 Pushing to MobileOrg](#)
- [B.3 Pulling from MobileOrg](#)
- [Appendix C History and acknowledgments](#)
- [C.1 From Carsten](#)
- [C.2 From Bastien](#)
- [C.3 List of contributions](#)
- [Appendix D GNU Free Documentation License](#)
- [Concept index](#)
- [Key index](#)
- [Command and function index](#)
- [Variable index](#)

9.10 ORG MODE MANUAL

This manual is for Org version 8.3.2 (release_8.3.2).

Copyright © 2004–2015 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover Texts being “A GNU Manual,” and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled “GNU Free Documentation License.”

(a) The FSF’s Back-Cover Text is: “You have the freedom to copy and modify this GNU manual.”

- [Introduction](#): Getting started
- [Document structure](#): A tree works like your brain
- [Tables](#): Pure magic for quick formatting
- [Hyperlinks](#): Notes in context
- [TODO items](#): Every tree branch can be a TODO item
- [Tags](#): Tagging headlines and matching sets of tags
- [Properties and columns](#): Storing information about an entry
- [Dates and times](#): Making items useful for planning
- [Capture – Refile – Archive](#): The ins and outs for projects
- [Agenda views](#): Collecting information into views
- [Markup](#): Prepare text for rich export
- [Exporting](#): Sharing and publishing notes
- [Publishing](#): Create a web site of linked Org files
- [Working with source code](#): Export, evaluate, and tangle code blocks
- [Miscellaneous](#): All the rest which did not fit elsewhere
- [Hacking](#): How to hack your way around
- [MobileOrg](#): Viewing and capture on a mobile device
- [History and acknowledgments](#): How Org came into being
- [GNU Free Documentation License](#): The license for this documentation.

- [Main Index](#): An index of Org's concepts and features
- [Key Index](#): Key bindings and where they are described
- [Command and Function Index](#): Command names and some internal functions
- [Variable Index](#): Variables mentioned in the manual

--- The Detailed Node Listing ---

Introduction

- [Summary](#): Brief summary of what Org does
- [Installation](#): Installing Org
- [Activation](#): How to activate Org for certain buffers
- [Feedback](#): Bug reports, ideas, patches etc.
- [Conventions](#): Typesetting conventions in the manual

Document structure

- [Outlines](#): Org is based on Outline mode
- [Headlines](#): How to typeset Org tree headlines
- [Visibility cycling](#): Show and hide, much simplified
- [Motion](#): Jumping to other headlines
- [Structure editing](#): Changing sequence and level of headlines
- [Sparse trees](#): Matches embedded in context
- [Plain lists](#): Additional structure within an entry
- [Drawers](#): Tucking stuff away
- [Blocks](#): Folding blocks
- [Footnotes](#): How footnotes are defined in Org's syntax
- [Orgstruct mode](#): Structure editing outside Org
- [Org syntax](#): Formal description of Org's syntax

Visibility cycling

- [Global and local cycling](#): Cycling through various visibility states
- [Initial visibility](#): Setting the initial visibility state
- [Catching invisible edits](#): Preventing mistakes when editing invisible parts

Tables

- [Built-in table editor](#): Simple tables
- [Column width and alignment](#): Overrule the automatic settings
- [Column groups](#): Grouping to trigger vertical lines
- [Orgtbl mode](#): The table editor as minor mode
- [The spreadsheet](#): The table editor has spreadsheet capabilities
- [Org-Plot](#): Plotting from org tables

The spreadsheet

- [References](#): How to refer to another field or range
- [Formula syntax for Calc](#): Using Calc to compute stuff
- [Formula syntax for Lisp](#): Writing formulas in Emacs Lisp
- [Durations and time values](#): How to compute durations and time values
- [Field and range formulas](#): Formula for specific (ranges of) fields
- [Column formulas](#): Formulas valid for an entire column
- [Lookup functions](#): Lookup functions for searching tables
- [Editing and debugging formulas](#): Fixing formulas
- [Updating the table](#): Recomputing all dependent fields
- [Advanced features](#): Field and column names, parameters and automatic recalc

Hyperlinks

- [Link format](#): How links in Org are formatted
- [Internal links](#): Links to other places in the current file
- [External links](#): URL-like links to the world
- [Handling links](#): Creating, inserting and following
- [Using links outside Org](#): Linking from my C source code?
- [Link abbreviations](#): Shortcuts for writing complex links
- [Search options](#): Linking to a specific location
- [Custom searches](#): When the default search is not enough

Internal links

- [Radio targets](#): Make targets trigger links in plain text

TODO items

- [TODO basics](#): Marking and displaying TODO entries
- [TODO extensions](#): Workflow and assignments
- [Progress logging](#): Dates and notes for progress
- [Priorities](#): Some things are more important than others
- [Breaking down tasks](#): Splitting a task into manageable pieces
- [Checkboxes](#): Tick-off lists

Extended use of TODO keywords

- [Workflow states](#): From TODO to DONE in steps
- [TODO types](#): I do this, Fred does the rest
- [Multiple sets in one file](#): Mixing it all, and still finding your way
- [Fast access to TODO states](#): Single letter selection of a state
- [Per-file keywords](#): Different files, different requirements
- [Faces for TODO keywords](#): Highlighting states
- [TODO dependencies](#): When one task needs to wait for others

Progress logging

- [Closing items](#): When was this entry marked DONE?
- [Tracking TODO state changes](#): When did the status change?
- [Tracking your habits](#): How consistent have you been?

Tags

- [Tag inheritance](#): Tags use the tree structure of the outline
- [Setting tags](#): How to assign tags to a headline
- [Tag hierarchy](#): Create a hierarchy of tags
- [Tag searches](#): Searching for combinations of tags

Properties and columns

- [Property syntax](#): How properties are spelled out
- [Special properties](#): Access to other Org mode features
- [Property searches](#): Matching property values

- [Property inheritance](#): Passing values down the tree
- [Column view](#): Tabular viewing and editing
- [Property API](#): Properties for Lisp programmers

Column view

- [Defining columns](#): The COLUMNS format property
- [Using column view](#): How to create and use column view
- [Capturing column view](#): A dynamic block for column view

Defining columns

- [Scope of column definitions](#): Where defined, where valid?
- [Column attributes](#): Appearance and content of a column

Dates and times

- [Timestamps](#): Assigning a time to a tree entry
- [Creating timestamps](#): Commands which insert timestamps
- [Deadlines and scheduling](#): Planning your work
- [Clocking work time](#): Tracking how long you spend on a task
- [Effort estimates](#): Planning work effort in advance
- [Timers](#): Notes with a running timer

Creating timestamps

- [The date/time prompt](#): How Org mode helps you entering date and time
- [Custom time format](#): Making dates look different

Deadlines and scheduling

- [Inserting deadline/schedule](#): Planning items
- [Repeated tasks](#): Items that show up again and again

Clocking work time

- [Clocking commands](#): Starting and stopping a clock
- [The clock table](#): Detailed reports
- [Resolving idle time](#): Resolving time when you've been idle

Capture – Refile – Archive

- [Capture](#): Capturing new stuff

- [Attachments](#): Add files to tasks
- [RSS feeds](#): Getting input from RSS feeds
- [Protocols](#): External (e.g., Browser) access to Emacs and Org
- [Refile and copy](#): Moving/copying a tree from one place to another
- [Archiving](#): What to do with finished projects

Capture

- [Setting up capture](#): Where notes will be stored
- [Using capture](#): Commands to invoke and terminate capture
- [Capture templates](#): Define the outline of different note types

Capture templates

- [Template elements](#): What is needed for a complete template entry
- [Template expansion](#): Filling in information about time and context
- [Templates in contexts](#): Only show a template in a specific context

Archiving

- [Moving subtrees](#): Moving a tree to an archive file
- [Internal archiving](#): Switch off a tree but keep it in the file

Agenda views

- [Agenda files](#): Files being searched for agenda information
- [Agenda dispatcher](#): Keyboard access to agenda views
- [Built-in agenda views](#): What is available out of the box?
- [Presentation and sorting](#): How agenda items are prepared for display
- [Agenda commands](#): Remote editing of Org trees
- [Custom agenda views](#): Defining special searches and views
- [Exporting agenda views](#): Writing a view to a file
- [Agenda column view](#): Using column view for collected entries

The built-in agenda views

- [Weekly/daily agenda](#): The calendar page with current tasks
- [Global TODO list](#): All unfinished action items

- [Matching tags and properties](#): Structured information with fine-tuned search
- [Timeline](#): Time-sorted view for single file
- [Search view](#): Find entries by searching for text
- [Stuck projects](#): Find projects you need to review

Presentation and sorting

- [Categories](#): Not all tasks are equal
- [Time-of-day specifications](#): How the agenda knows the time
- [Sorting agenda items](#): The order of things
- [Filtering/limiting agenda items](#): Dynamically narrow the agenda

Custom agenda views

- [Storing searches](#): Type once, use often
- [Block agenda](#): All the stuff you need in a single buffer
- [Setting options](#): Changing the rules

Markup for rich export

- [Structural markup elements](#): The basic structure as seen by the exporter
- [Images and tables](#): Images, tables and caption mechanism
- [Literal examples](#): Source code examples with special formatting
- [Include files](#): Include additional files into a document
- [Index entries](#): Making an index
- [Macro replacement](#): Use macros to create templates
- [Embedded LaTeX](#): LaTeX can be freely used inside Org documents
- [Special blocks](#): Containers targeted at export back-ends

Structural markup elements

- [Document title](#): Where the title is taken from
- [Headings and sections](#): The document structure as seen by the exporter
- [Table of contents](#): The if and where of the table of contents
- [Lists](#): Lists
- [Paragraphs](#): Paragraphs

- [Footnote markup](#): Footnotes
- [Emphasis and monospace](#): Bold, italic, etc.
- [Horizontal rules](#): Make a line
- [Comment lines](#): What will *not* be exported

Embedded LaTeX

- [Special symbols](#): Greek letters and other symbols
- [Subscripts and superscripts](#): Simple syntax for raising/lowering text
- [LaTeX fragments](#): Complex formulas made easy
- [Previewing LaTeX fragments](#): What will this snippet look like?
- [CDLaTeX mode](#): Speed up entering of formulas

Exporting

- [The export dispatcher](#): The main exporter interface
- [Export back-ends](#): Built-in export formats
- [Export settings](#): Generic export settings
- [ASCII/Latin-1/UTF-8 export](#): Exporting to flat files with encoding
- [Beamer export](#): Exporting as a Beamer presentation
- [HTML export](#): Exporting to HTML
- [LaTeX and PDF export](#): Exporting to LaTeX, and processing to PDF
- [Markdown export](#): Exporting to Markdown
- [OpenDocument Text export](#): Exporting to OpenDocument Text
- [Org export](#): Exporting to Org
- [Texinfo export](#): Exporting to Texinfo
- [iCalendar export](#): Exporting to iCalendar
- [Other built-in back-ends](#): Exporting to a man page
- [Export in foreign buffers](#): Author tables and lists in Org syntax
- [Advanced configuration](#): Fine-tuning the export output

HTML export

- [HTML Export commands](#): How to invoke HTML export
- [HTML doctypes](#): Org can export to various (X)HTML flavors

- [HTML preamble and postamble](#): How to insert a preamble and a postamble
- [Quoting HTML tags](#): Using direct HTML in Org mode
- [Links in HTML export](#): How links will be interpreted and formatted
- [Tables in HTML export](#): How to modify the formatting of tables
- [Images in HTML export](#): How to insert figures into HTML output
- [Math formatting in HTML export](#): Beautiful math also on the web
- [Text areas in HTML export](#): An alternative way to show an example
- [CSS support](#): Changing the appearance of the output
- [JavaScript support](#): Info and Folding in a web browser

LaTeX and PDF export

- [LaTeX export commands](#): How to export to LaTeX and PDF
- [Header and sectioning](#): Setting up the export file structure
- [Quoting LaTeX code](#): Incorporating literal LaTeX code
- [LaTeX specific attributes](#): Controlling LaTeX output

OpenDocument text export

- [Pre-requisites for ODT export](#): What packages ODT exporter relies on
- [ODT export commands](#): How to invoke ODT export
- [Extending ODT export](#): How to produce 'doc', 'pdf' files
- [Applying custom styles](#): How to apply custom styles to the output
- [Links in ODT export](#): How links will be interpreted and formatted
- [Tables in ODT export](#): How Tables are exported
- [Images in ODT export](#): How to insert images
- [Math formatting in ODT export](#): How LaTeX fragments are formatted
- [Labels and captions in ODT export](#): How captions are rendered
- [Literal examples in ODT export](#): How source and example blocks are formatted
- [Advanced topics in ODT export](#): Read this if you are a power user

Math formatting in ODT export

- [Working with LaTeX math snippets](#): How to embed LaTeX math fragments

- [Working with MathML or OpenDocument formula files](#): How to embed equations in native format

Advanced topics in ODT export

- [Configuring a document converter](#): How to register a document converter
- [Working with OpenDocument style files](#): Explore the internals
- [Creating one-off styles](#): How to produce custom highlighting etc
- [Customizing tables in ODT export](#): How to define and use Table templates
- [Validating OpenDocument XML](#): How to debug corrupt OpenDocument files

Texinfo export

- [Texinfo export commands](#): How to invoke Texinfo export
- [Document preamble](#): File header, title and copyright page
- [Headings and sectioning structure](#): Building document structure
- [Indices](#): Creating indices
- [Quoting Texinfo code](#): Incorporating literal Texinfo code
- [Texinfo specific attributes](#): Controlling Texinfo output
- [An example](#)

Publishing

- [Configuration](#): Defining projects
- [Uploading files](#): How to get files up on the server
- [Sample configuration](#): Example projects
- [Triggering publication](#): Publication commands

Configuration

- [Project alist](#): The central configuration variable
- [Sources and destinations](#): From here to there
- [Selecting files](#): What files are part of the project?
- [Publishing action](#): Setting the function doing the publishing
- [Publishing options](#): Tweaking HTML/LaTeX export
- [Publishing links](#): Which links keep working after publishing?
- [Sitemap](#): Generating a list of all pages

- [Generating an index](#): An index that reaches across pages

Sample configuration

- [Simple example](#): One–component publishing
- [Complex example](#): A multi–component publishing example

Working with source code

- [Structure of code blocks](#): Code block syntax described
- [Editing source code](#): Language major–mode editing
- [Exporting code blocks](#): Export contents and/or results
- [Extracting source code](#): Create pure source code files
- [Evaluating code blocks](#): Place results of evaluation in the Org mode buffer
- [Library of Babel](#): Use and contribute to a library of useful code blocks
- [Languages](#): List of supported code block languages
- [Header arguments](#): Configure code block functionality
- [Results of evaluation](#): How evaluation results are handled
- [Noweb reference syntax](#): Literate programming in Org mode
- [Key bindings and useful functions](#): Work quickly with code blocks
- [Batch execution](#): Call functions from the command line

Header arguments

- [Using header arguments](#): Different ways to set header arguments
- [Specific header arguments](#): List of header arguments

Using header arguments

- [System–wide header arguments](#): Set global default values
- [Language–specific header arguments](#): Set default values by language
- [Header arguments in Org mode properties](#): Set default values for a buffer or heading
- [Language–specific header arguments in Org mode properties](#): Set language–specific default values for a buffer or heading
- [Code block specific header arguments](#): The most common way to set values
- [Header arguments in function calls](#): The most specific level

Specific header arguments

- [var](#): Pass arguments to code blocks
- [results](#): Specify the type of results and how they will be collected and handled
- [file](#): Specify a path for file output
- [file-desc](#): Specify a description for file results
- [dir](#): Specify the default (possibly remote) directory for code block execution
- [exports](#): Export code and/or results
- [tangle](#): Toggle tangling and specify file name
- [mkdirp](#): Toggle creation of parent directories of target files during tangling
- [comments](#): Toggle insertion of comments in tangled code files
- [padline](#): Control insertion of padding lines in tangled code files
- [no-expand](#): Turn off variable assignment and noweb expansion during tangling
- [session](#): Preserve the state of code evaluation
- [noweb](#): Toggle expansion of noweb references
- [noweb-ref](#): Specify block's noweb reference resolution target
- [noweb-sep](#): String used to separate noweb references
- [cache](#): Avoid re-evaluating unchanged code blocks
- [sep](#): Delimiter for writing tabular results outside Org
- [hlines](#): Handle horizontal lines in tables
- [colnames](#): Handle column names in tables
- [rownames](#): Handle row names in tables
- [shebang](#): Make tangled files executable
- [tangle-mode](#): Set permission of tangled files
- [eval](#): Limit evaluation of specific code blocks
- [wrap](#): Mark source block evaluation results
- [post](#): Post processing of code block results
- [prologue](#): Text to prepend to code block body

- [epilogue](#): Text to append to code block body

Miscellaneous

- [Completion](#): M-TAB knows what you need
- [Easy templates](#): Quick insertion of structural elements
- [Speed keys](#): Electric commands at the beginning of a headline
- [Code evaluation security](#): Org mode files evaluate inline code
- [Customization](#): Adapting Org to your taste
- [In-buffer settings](#): Overview of the #+KEYWORDS
- [The very busy C-c C-c key](#): When in doubt, press C-c C-c
- [Clean view](#): Getting rid of leading stars in the outline
- [TTY keys](#): Using Org on a tty
- [Interaction](#): Other Emacs packages
- [org-crypt](#): Encrypting Org files

Interaction with other packages

- [Cooperation](#): Packages Org cooperates with
- [Conflicts](#): Packages that lead to conflicts

Hacking

- [Hooks](#): How to reach into Org's internals
- [Add-on packages](#): Available extensions
- [Adding hyperlink types](#): New custom link types
- [Adding export back-ends](#): How to write new export back-ends
- [Context-sensitive commands](#): How to add functionality to such commands
- [Tables in arbitrary syntax](#): Orgtbl for LaTeX and other programs
- [Dynamic blocks](#): Automatically filled blocks
- [Special agenda views](#): Customized views
- [Speeding up your agendas](#): Tips on how to speed up your agendas
- [Extracting agenda information](#): Post-processing of agenda information
- [Using the property API](#): Writing programs that use entry properties
- [Using the mapping API](#): Mapping over all or selected entries

Tables and lists in arbitrary syntax

- [Radio tables](#): Sending and receiving radio tables
- [A LaTeX example](#): Step by step, almost a tutorial
- [Translator functions](#): Copy and modify
- [Radio lists](#): Sending and receiving lists

MobileOrg

- [Setting up the staging area](#): Where to interact with the mobile device
- [Pushing to MobileOrg](#): Uploading Org files and agendas
- [Pulling from MobileOrg](#): Integrating captured and flagged items

9.10.1 Summary

Org is a mode for keeping notes, maintaining TODO lists, and project planning with a fast and effective plain-text system. It also is an authoring system with unique support for literate programming and reproducible research.

Org is implemented on top of Outline mode, which makes it possible to keep the content of large files well structured. Visibility cycling and structure editing help to work with the tree. Tables are easily created with a built-in table editor. Plain text URL-like links connect to websites, emails, Usenet messages, BBDB entries, and any files related to the projects.

Org develops organizational tasks around notes files that contain lists or information about projects as plain text. Project planning and task management makes use of metadata which is part of an outline node. Based on this data, specific entries can be extracted in queries and create dynamic *agenda views* that also integrate the Emacs calendar and diary. Org can be used to implement many different project planning schemes, such as David Allen's GTD system.

Org files can serve as a single source authoring system with export to many different formats such as HTML, LaTeX, Open Document, and Markdown. New export backends can be derived from existing ones, or defined from scratch.

Org files can include source code blocks, which makes Org uniquely suited for authoring technical documents with code examples. Org source code blocks are fully functional; they can be evaluated in place and their results can be captured in the file. This makes it possible to create a single file reproducible research compendium.

Org keeps simple things simple. When first fired up, it should feel like a straightforward, easy to use outliner. Complexity is not imposed, but a large amount of functionality is available when needed. Org is a toolbox. Many users actually run only a (very personal) fraction of Org's capabilities, and know that there is more whenever they need it.

All of this is achieved with strictly plain text files, the most portable and future-proof file format. Org runs in Emacs. Emacs is one of the most widely ported programs, so that Org mode is available on every major platform.

There is a website for Org which provides links to the newest version of Org, as well as additional information, frequently asked questions (FAQ), links to tutorials, etc. This page is located at <http://orgmode.org>. An earlier version (7.3) of this manual is available as a [paperback book from Network Theory Ltd.](#)

9.11

9.12 DOCUMENT STRUCTURE

Org is based on Outline mode and provides flexible commands to edit the structure of the document.

1. [Outlines](#): Org is based on Outline mode
2. [Headlines](#): How to typeset Org tree headlines
3. [Visibility cycling](#): Show and hide, much simplified
4. [Motion](#): Jumping to other headlines
5. [Structure editing](#): Changing sequence and level of headlines
6. [Sparse trees](#): Matches embedded in context
7. [Plain lists](#): Additional structure within an entry
8. [Drawers](#): Tucking stuff away
9. [Blocks](#): Folding blocks
10. [Footnotes](#): How footnotes are defined in Org's syntax
11. [Orgstruct mode](#): Structure editing outside Org
12. [Org syntax](#): Formal description of Org's syntax

9.12.1 1 Outlines

Org is implemented on top of Outline mode. Outlines allow a document to be organized in a hierarchical structure, which (at least for me) is the best representation of notes and thoughts. An overview of this structure is achieved by folding (hiding) large parts of the document to show only the general document structure and the parts currently being worked on. Org greatly simplifies the use of outlines by compressing the entire show/hide functionality into a single command, `org-cycle`, which is bound to the `<TAB>` key.

9.12.2 2 Headlines

Headlines define the structure of an outline tree. The headlines in Org start with one or more stars, on the left margin [1](#) [2](#). For example:

```
* Top level headline

** Second level

*** 3rd level
    some text

*** 3rd level
    more text

* Another top level headline
```

Note that a headline named after `org-footnote-section`, which defaults to 'Footnotes', is considered as special. A subtree with this headline will be silently ignored by exporting functions.

Some people find the many stars too noisy and would prefer an outline that has whitespace followed by a single star as headline starters. [Clean view](#), describes a setup to realize this.

An empty line after the end of a subtree is considered part of it and will be hidden when the subtree is folded. However, if you leave at least two empty lines, one empty line will remain visible after folding the subtree, in order to structure the collapsed view. See the variable `org-cycle-separator-lines` to modify this behavior.

9.12.2.1 Footnotes

[1] See the variables `org-special-ctrl-a/e`, `org-special-ctrl-k`, and `org-ctrl-k-protect-subtree` to configure special behavior of C-a, C-e, and C-k in headlines.

[2] Clocking only works with headings indented less than 30 stars.

9.12.3 3 Visibili ty cycling

- [Global and local cycling](#): Cycling through various visibility states

9.12.3.1 2.3.1 Global and local cycling

Outlines make it possible to hide parts of the text in the buffer. Org uses just two commands, bound to `<TAB>` and `S-<TAB>` to change the visibility in the buffer.

`<TAB>` (org-cycle)

Subtree cycling: Rotate current subtree among the states

```
, -> FOLDED -> CHILDREN -> SUBTREE -- .
'-----'
```

The cursor must be on a headline for this to work¹. When the cursor is at the beginning of the buffer and the first line is not a headline, then `<TAB>` actually runs global cycling (see below)². Also when called with a prefix argument (`C-u <TAB>`), global cycling is invoked.

`S-<TAB>` (org-global-cycle)

`C-u <TAB>`

Global cycling: Rotate the entire buffer among the states

```
, -> OVERVIEW -> CONTENTS -> SHOW ALL --.
'-----'
```

When `S-<TAB>` is called with a numeric prefix argument `N`, the `CONTENTS` view up to headlines of level `N` will be shown. Note that inside tables, `S-<TAB>` jumps to the previous field.

`C-u C-u <TAB>` (`org-set-startup-visibility`)

Switch back to the startup visibility of the buffer (see [Initial visibility](#)).

`C-u C-u C-u <TAB>` (`show-all`)

Show all, including drawers.

`C-c C-r` (`org-reveal`)

Reveal context around point, showing the current entry, the following heading and the hierarchy above. Useful for working near a location that has been exposed by a sparse tree command (see [Sparse trees](#)) or an agenda command (see [Agenda commands](#)). With a prefix argument `show`, on each level, all sibling headings. With a double prefix argument, also show the entire subtree of the parent.

`C-c C-k` (`show-branches`)

Expose all the headings of the subtree, `CONTENT` view for just one subtree.

`C-c <TAB>` (`show-children`)

Expose all direct children of the subtree. With a numeric prefix argument `N`, expose all children down to level `N`.

`C-c C-x b` (`org-tree-to-indirect-buffer`)

Show the current subtree in an indirect buffer³. With a numeric prefix argument `N`, go up to level `N` and then take that tree. If `N` is negative then go up that many levels. With a `C-u` prefix, do not remove the previously used indirect buffer.

`C-c C-x v` (`org-copy-visible`)

Copy the *visible* text in the region into the kill ring.

9.12.3.2 Footnotes

[1] see, however, the option `org-cycle-emulate-tab`.

[2] see the option `org-cycle-global-at-bob`.

[3] The indirect buffer (see [Indirect Buffers](#)) will contain the entire buffer, but will be narrowed to the current tree. Editing the indirect buffer will also change the original buffer, but without affecting visibility in that buffer.

- [Initial visibility](#): Setting the initial visibility state

9.12.3.3 2.3.2 Initial visibility

When Emacs first visits an Org file, the global state is set to OVERVIEW, i.e., only the top level headlines are visible¹. This can be configured through the variable `org-startup-folded`, or on a per-file basis by adding one of the following lines anywhere in the buffer:

```
#+STARTUP: overview

#+STARTUP: content

#+STARTUP: showall

#+STARTUP: showeverything
```

The startup visibility options are ignored when the file is open for the first time during the agenda generation: if you want the agenda to honor the startup visibility, set `org-agenda-inhibit-startup` to `nil`.

Furthermore, any entries with a ‘VISIBILITY’ property (see [Properties and columns](#)) will get their visibility adapted accordingly. Allowed values for this property are `folded`, `children`, `content`, and `all`.

C-u C-u <TAB> (org-set-startup-visibility)

Switch back to the startup visibility of the buffer, i.e., whatever is requested by startup options and ‘VISIBILITY’ properties in individual entries.

9.12.3.4 Footnotes

[1] When `org-agenda-inhibit-startup` is non-`nil`, Org will not honor the default visibility state when first opening a file for the agenda (see [Speeding up your agendas](#)).

- [Catching invisible edits](#): Preventing mistakes when editing invisible parts

9.12.3.5 2.3.3 Catching invisible edits

Sometimes you may inadvertently edit an invisible part of the buffer and be confused on what has been edited and how to undo the mistake. Setting `org-catch-invisible-edits` to `non-nil` will help prevent this. See the docstring of this option on how Org should catch invisible edits and process them.

9.12.4 4 Motion

The following commands jump to other headlines in the buffer.

`C-c C-n` (`outline-next-visible-heading`)

Next heading.

`C-c C-p` (`outline-previous-visible-heading`)

Previous heading.

`C-c C-f` (`org-forward-same-level`)

Next heading same level.

`C-c C-b` (`org-backward-same-level`)

Previous heading same level.

`C-c C-u` (`outline-up-heading`)

Backward to higher level heading.

`C-c C-j` (`org-goto`)

Jump to a different place without changing the current outline visibility. Shows the document structure in a temporary buffer, where you can use the following keys to find your destination:

<code><TAB></code>	Cycle visibility.
<code><down></code> / <code><up></code>	Next/previous visible headline.
<code><RET></code>	Select this location.
<code>/</code>	Do a Sparse-tree search

The following keys work if you turn off `org-goto-auto-isearch`

n / p	Next/previous visible headline.
f / b	Next/previous headline same level.
u	One level up.
0-9	Digit argument.
q	Quit

See also the option `org-goto-interface`.

9.12.5 5 Structure re editing

M-<RET> (org-insert-heading)

Insert a new heading/item with the same level as the one at point.

If the cursor is in a plain list item, a new item is created (see [Plain lists](#)). To prevent this behavior in lists, call the command with one prefix argument. When this command is used in the middle of a line, the line is split and the rest of the line becomes the new item or headline. If you do not want the line to be split, customize `org-M-RET-may-split-line`.

If the command is used at the *beginning* of a line, and if there is a heading or an item at point, the new heading/item is created *before* the current line. If the command is used at the *end* of a folded subtree (i.e., behind the ellipses at the end of a headline), then a headline will be inserted after the end of the subtree.

Calling this command with C-u C-u will unconditionally respect the headline's content and create a new item at the end of the parent subtree.

If point is at the beginning of a normal line, turn this line into a heading.

C-<RET> (org-insert-heading-respect-content)

Just like M-<RET>, except when adding a new heading below the current heading, the new heading is placed after the body instead of before it. This command works from anywhere in the entry.

M-S-<RET> (org-insert-todo-heading)

Insert new TODO entry with same level as current heading. See also the variable `org-treat-insert-todo-heading-as-state-change`.

`C-S-<RET>` (`org-insert-todo-heading-respect-content`)

Insert new TODO entry with same level as current heading. Like `C-<RET>`, the new headline will be inserted after the current subtree.

`<TAB>` (`org-cycle`)

In a new entry with no text yet, the first `<TAB>` demotes the entry to become a child of the previous one. The next `<TAB>` makes it a parent, and so on, all the way to top level. Yet another `<TAB>`, and you are back to the initial level.

`M-<left>` (`org-do-promote`)

Promote current heading by one level.

`M-<right>` (`org-do-demote`)

Demote current heading by one level.

`M-S-<left>` (`org-promote-subtree`)

Promote the current subtree by one level.

`M-S-<right>` (`org-demote-subtree`)

Demote the current subtree by one level.

`M-S-<up>` (`org-move-subtree-up`)

Move subtree up (swap with previous subtree of same level).

`M-S-<down>` (`org-move-subtree-down`)

Move subtree down (swap with next subtree of same level).

`M-h` (`org-mark-element`)

Mark the element at point. Hitting repeatedly will mark subsequent elements of the one just marked. E.g., hitting `<M-h>` on a paragraph will mark it, hitting `<M-h>` immediately again will mark the next one.

`C-c @` (`org-mark-subtree`)

Mark the subtree at point. Hitting repeatedly will mark subsequent subtrees of the same level than the marked subtree.

`C-c C-x C-w` (`org-cut-subtree`)

Kill subtree, i.e., remove it from buffer but save in kill ring. With a numeric prefix argument `N`, kill `N` sequential subtrees.

C-c C-x M-w (org-copy-subtree)

Copy subtree to kill ring. With a numeric prefix argument N, copy the N sequential subtrees.

C-c C-x C-y (org-paste-subtree)

Yank subtree from kill ring. This does modify the level of the subtree to make sure the tree fits in nicely at the yank position. The yank level can also be specified with a numeric prefix argument, or by yanking after a headline marker like '****'.

C-y (org-yank)

Depending on the options `org-yank-adjusted-subtrees` and `org-yank-folded-subtrees`, Org's internal `yank` command will paste subtrees folded and in a clever way, using the same command as C-c C-x C-y. With the default settings, no level adjustment will take place, but the yanked tree will be folded unless doing so would swallow text previously visible. Any prefix argument to this command will force a normal `yank` to be executed, with the prefix passed along. A good way to force a normal yank is C-u C-y. If you use `yank-pop` after a yank, it will yank previous kill items plainly, without adjustment and folding.

C-c C-x c (org-clone-subtree-with-time-shift)

Clone a subtree by making a number of sibling copies of it. You will be prompted for the number of copies to make, and you can also specify if any timestamps in the entry should be shifted. This can be useful, for example, to create a number of tasks related to a series of lectures to prepare. For more details, see the docstring of the command `org-clone-subtree-with-time-shift`.

C-c C-w (org-refile)

Refile entry or region to a different location. See [Refile and copy](#).

C-c ^ (org-sort)

Sort same-level entries. When there is an active region, all entries in the region will be sorted. Otherwise the children of the current headline are sorted. The command prompts for the sorting method, which can be alphabetically, numerically, by time (first timestamp with active preferred, creation time, scheduled time, deadline time), by priority, by TODO keyword (in the sequence the keywords have been defined in the setup) or by the value of a property. Reverse sorting is possible as well. You can also supply your own function to extract the sorting key. With a C-u prefix, sorting will be case-sensitive.

C-x n s (org-narrow-to-subtree)

Narrow buffer to current subtree.

C-x n b (org-narrow-to-block)

Narrow buffer to current block.

C-x n w (widen)

Widen buffer to remove narrowing.

C-c * (org-toggle-heading)

Turn a normal line or plain list item into a headline (so that it becomes a subheading at its location). Also turn a headline into a normal line by removing the stars. If there is an active region, turn all lines in the region into headlines. If the first line in the region was an item, turn only the item lines into headlines. Finally, if the first line is a headline, remove the stars from all headlines in the region.

When there is an active region (Transient Mark mode), promotion and demotion work on all headlines in the region. To select a region of headlines, it is best to place both point and mark at the beginning of a line, mark at the beginning of the first headline, and point at the line just after the last headline to change. Note that when the cursor is inside a table (see [Tables](#)), the Meta-Cursor keys have different functionality.

9.12.6 6

Sparse trees

An important feature of Org mode is the ability to construct *sparse trees* for selected information in an outline tree, so that the entire document is folded as much as possible, but the selected information is made visible along with the headline structure above it¹. Just try it out and you will see immediately how it works.

Org mode contains several commands for creating such trees, all these commands can be accessed through a dispatcher:

C-c / (org-sparse-tree)

This prompts for an extra key to select a sparse-tree creating command.

C-c / r (org-occur)

Prompts for a regexp and shows a sparse tree with all matches. If the match is in a headline, the headline is made visible. If the match is in the body of an

entry, headline and body are made visible. In order to provide minimal context, also the full hierarchy of headlines above the match is shown, as well as the headline following the match. Each match is also highlighted; the highlights disappear when the buffer is changed by an editing command², or by pressing `C-c C-c`. When called with a `C-u` prefix argument, previous highlights are kept, so several calls to this command can be stacked.

`M-g n` or `M-g M-n` (next-error)

Jump to the next sparse tree match in this buffer.

`M-g p` or `M-g M-p` (previous-error)

Jump to the previous sparse tree match in this buffer.

For frequently used sparse trees of specific search strings, you can use the option `org-agenda-custom-commands` to define fast keyboard access to specific sparse trees. These commands will then be accessible through the agenda dispatcher (see [Agenda dispatcher](#)). For example:

```
(setq org-agenda-custom-commands
      '(("f" occur-tree "FIXME")))
```

will define the key `C-c a f` as a shortcut for creating a sparse tree matching the string 'FIXME'.

The other sparse tree commands select headings based on TODO keywords, tags, or properties and will be discussed later in this manual.

To print a sparse tree, you can use the Emacs command `ps-print-buffer-with-faces` which does not print invisible parts of the document³. Or you can use `C-c C-e C-v` to export only the visible part of the document and print the resulting file.

9.12.6.1 Footnotes

^[1] See also the variable `org-show-context-detail` to decide how much context is shown around each match.

^[2] This depends on the option `org-remove-highlights-with-change`

^[3] This does not work under XEmacs, because XEmacs uses selective display for outlining, not text properties.

9.12.7 7 Plain lists

Within an entry of the outline tree, hand-formatted lists can provide additional structure. They also provide a way to create lists of checkboxes (see [Checkboxes](#)). Org supports editing such lists, and every exporter (see [Exporting](#)) can parse and format them.

Org knows ordered lists, unordered lists, and description lists.

- *Unordered* list items start with ‘-’, ‘+’, or ‘*’¹ as bullets.
- *Ordered* list items start with a numeral followed by either a period or a right parenthesis², such as ‘1.’ or ‘1)’³. If you want a list to start with a different value (e.g., 20), start the text of the item with [20]⁴. Those constructs can be used in any item of the list in order to enforce a particular numbering.
- *Description* list items are unordered list items, and contain the separator ‘ :: ’ to distinguish the description *term* from the description.

Items belonging to the same list must have the same indentation on the first line. In particular, if an ordered list reaches number ‘10.’, then the 2-digit numbers must be written left-aligned with the other numbers in the list. An item ends before the next line that is less or equally indented than its bullet/number.

A list ends whenever every item has ended, which means before any line less or equally indented than items at top level. It also ends before two blank lines⁵. In that case, all items are closed. Here is an example:

```
** Lord of the Rings
```

```
My favorite scenes are (in this order)
```

```
1. The attack of the Rohirrim
```

```
2. Eowyn's fight with the witch king
```

```
+ this was already my favorite scene in the book
```

```
+ I really like Miranda Otto.
```

```
3. Peter Jackson being shot by Legolas
```

```
- on DVD only
```

```
He makes a really funny face when it happens.
```



```

    But in the end, no individual scenes matter but the film as a
whole.

```

```

    Important actors in this film are:

```

```

    - Elijah Wood :: He plays Frodo

```

```

    - Sean Austin :: He plays Sam, Frodo's friend. I still
remember

```

```

        him very well from his role as Mikey Walsh in The Goonies.

```

Org supports these lists by tuning filling and wrapping commands to deal with them correctly⁶, and by exporting them properly (see [Exporting](#)). Since indentation is what governs the structure of these lists, many structural constructs like `#+BEGIN_...` blocks can be indented to signal that they belong to a particular item.

If you find that using a different bullet for a sub-list (than that used for the current list-level) improves readability, customize the variable `org-list-demote-modify-bullet`. To get a greater difference of indentation between items and their sub-items, customize `org-list-indent-offset`.

The following commands act on items when the cursor is in the first line of an item (the line with the bullet or number). Some of them imply the application of automatic rules to keep list structure intact. If some of these actions get in your way, configure `org-list-automatic-rules` to disable them individually.

```
<TAB>      (org-cycle)
```

Items can be folded just like headline levels. Normally this works only if the cursor is on a plain list item. For more details, see the variable `org-cycle-include-plain-lists`. If this variable is set to `integrate`, plain list items will be treated like low-level headlines. The level of an item is then given by the indentation of the bullet/number. Items are always subordinate to real headlines, however; the hierarchies remain completely separated. In a new item with no text yet, the first `<TAB>` demotes the item to become a child of the previous one. Subsequent `<TAB>`s move the item to meaningful levels in the list and eventually get it back to its initial position.

```
M-<RET>    (org-insert-heading)
```

Insert new item at current level. With a prefix argument, force a new heading (see [Structure editing](#)). If this command is used in the middle of an item, that item is *split* in two, and the second part becomes the new item⁷. If this command is executed *before item's body*, the new item is created *before* the current one.

M-S-<RET>

Insert a new item with a checkbox (see [Checkboxes](#)).

S-up

S-down

Jump to the previous/next item in the current list⁸, but only if `org-support-shift-select` is off. If not, you can still use paragraph jumping commands like C-<up> and C-<down> to quite similar effect.

M-up

M-down

Move the item including subitems up/down⁹ (swap with previous/next item of same indentation). If the list is ordered, renumbering is automatic.

M-left

M-right

Decrease/increase the indentation of an item, leaving children alone.

M-S-<left>

M-S-<right>

Decrease/increase the indentation of the item, including subitems. Initially, the item tree is selected based on current indentation. When these commands are executed several times in direct succession, the initially selected region is used, even if the new indentation would imply a different hierarchy. To use the new hierarchy, break the command chain with a cursor motion or so.

As a special case, using this command on the very first item of a list will move the whole list. This behavior can be disabled by configuring `org-list-automatic-rules`. The global indentation of a list has no influence on the text *after* the list.

C-c C-c

If there is a checkbox (see [Checkboxes](#)) in the item line, toggle the state of the checkbox. In any case, verify bullets and indentation consistency in the whole list.

C-c -

Cycle the entire list level through the different itemize/enumerate bullets ('-', '+', '*', '1.', '1') or a subset of them, depending on `org-plain-list-ordered-item-terminator`, the type of list, and its indentation. With a numeric prefix argument N, select the Nth bullet from this list. If there is an active region when calling this, selected text will be changed into an item. With a prefix argument, all lines will be converted to list items. If the first line already was

a list item, any item marker will be removed from the list. Finally, even without an active region, a normal line will be converted into a list item.

C-c *

Turn a plain list item into a headline (so that it becomes a subheading at its location). See [Structure editing](#), for a detailed explanation.

C-c C-*

Turn the whole plain list into a subtree of the current heading. Checkboxes (see [Checkboxes](#)) will become TODO (resp. DONE) keywords when unchecked (resp. checked).

S-left/right

This command also cycles bullet styles when the cursor is on the bullet or anywhere in an item line, details depending on `org-support-shift-select`.

C-c ^

Sort the plain list. You will be prompted for the sorting method: numerically, alphabetically, by time, by checked status for check lists, or by a custom function.

9.12.7.1 Footnotes

[1] When using ‘*’ as a bullet, lines must be indented or they will be seen as top-level headlines. Also, when you are hiding leading stars to get a clean outline view, plain list items starting with a star may be hard to distinguish from true headlines. In short: even though ‘*’ is supported, it may be better to not use it for plain list items.

[2] You can filter out any of them by configuring `org-plain-list-ordered-item-terminator`.

[3] You can also get ‘a.’, ‘A.’, ‘a)’ and ‘A)’ by configuring `org-list-allow-alphabetical`. To minimize confusion with normal text, those are limited to one character only. Beyond that limit, bullets will automatically fallback to numbers.

[4] If there's a checkbox in the item, the cookie must be put *before* the checkbox. If you have activated alphabetical lists, you can also use counters like `[@b]`.

[5] See also `org-list-empty-line-terminates-plain-lists`.

[6] Org only changes the filling settings for Emacs. For XEmacs, you should use Kyle E. Jones' `filladapt.el`. To turn this on, put into `.emacs`: `(require 'filladapt)`

[7] If you do not want the item to be split, customize the variable `org-M-RET-may-split-line`.

[8] If you want to cycle around items that way, you may customize `org-list-use-circular-motion`.

[9] See `org-list-use-circular-motion` for a cyclic behavior.

9.12.8 8 Drawer s

Sometimes you want to keep information associated with an entry, but you normally don't want to see it. For this, Org mode has *drawers*. They can contain anything but a headline and another drawer. Drawers look like this:

```
** This is a headline

Still outside the drawer

:DRAWERNAME:

This is inside the drawer.

:END:

After the drawer.
```

You can interactively insert drawers at point by calling `org-insert-drawer`, which is bound to `<C-c C-x d>`. With an active region, this command will put the region inside the drawer. With a prefix argument, this command calls `org-insert-property-drawer` and add a property drawer right below the current headline. Completion over drawer keywords is also possible using `<M-TAB>`.

Visibility cycling (see [Visibility cycling](#)) on the headline will hide and show the entry, but keep the drawer collapsed to a single line. In order to look inside the drawer, you need to move the cursor to the drawer line and press `<TAB>` there. Org mode uses the `PROPERTIES` drawer for storing properties (see [Properties and columns](#)), and you can also arrange for state change notes (see [Tracking TODO state changes](#)) and

clock times (see [Clocking work time](#)) to be stored in a drawer LOGBOOK. If you want to store a quick note in the LOGBOOK drawer, in a similar way to state changes, use

C-c C-z

Add a time-stamped note to the LOGBOOK drawer.

You can select the name of the drawers which should be exported with `org-export-with-drawers`. In that case, drawer contents will appear in export output. Property drawers are not affected by this variable: configure `org-export-with-properties` instead.

9.12.9 9

Blocks

Org mode uses begin...end blocks for various purposes from including source code examples (see [Literal examples](#)) to capturing time logging information (see [Clocking work time](#)). These blocks can be folded and unfolded by pressing TAB in the begin line. You can also get all blocks folded at startup by configuring the option `org-hide-block-startup` or on a per-file basis by using

```
#+STARTUP: hideblocks
```

```
#+STARTUP: nohideblocks
```

9.12.10 10

Footnotes

Org mode supports the creation of footnotes. In contrast to the `footnote.el` package, Org mode's footnotes are designed for work on a larger document, not only for one-off documents like emails.

A footnote is started by a footnote marker in square brackets in column 0, no indentation allowed. It ends at the next footnote definition, headline, or after two consecutive empty lines. The footnote reference is simply the marker in square brackets, inside text. For example:

```
The Org homepage[fn:1] now looks a lot better than it used to.
```

```
...
```

```
[fn:1] The link is: http://orgmode.org
```

Org mode extends the number-based syntax to *named* footnotes and optional inline definition. Using plain numbers as markers (as `footnote.el` does) is supported for backward compatibility, but not encouraged because of possible conflicts with LaTeX snippets (see [Embedded LaTeX](#)). Here are the valid references:

`[1]`

A plain numeric footnote marker. Compatible with `footnote.el`, but not recommended because something like ‘`[1]`’ could easily be part of a code snippet.

`[fn:name]`

A named footnote reference, where `name` is a unique label word, or, for simplicity of automatic creation, a number.

`[fn:: This is the inline definition of this footnote]`

A LaTeX-like anonymous footnote where the definition is given directly at the reference point.

`[fn:name: a definition]`

An inline definition of a footnote, which also specifies a name for the note. Since Org allows multiple references to the same note, you can then use `[fn:name]` to create additional references.

Footnote labels can be created automatically, or you can create names yourself. This is handled by the variable `org-footnote-auto-label` and its corresponding `#+STARTUP` keywords. See the docstring of that variable for details.

The following command handles footnotes:

`C-c C-x f`

The footnote action command.

When the cursor is on a footnote reference, jump to the definition. When it is at a definition, jump to the (first) reference.

Otherwise, create a new footnote. Depending on the option `org-footnote-define-inline1`, the definition will be placed right into the text as part of the reference, or separately into the location determined by the option `org-footnote-section`.

When this command is called with a prefix argument, a menu of additional options is offered:

- `s` Sort the footnote definitions by reference sequence. During editing, Org makes no effort to sort footnote definitions into a particular sequence. If you want them sorted, use this command, which will

also move entries according to `org-footnote-section`. Automatic sorting after each insertion/deletion can be configured using the option `org-footnote-auto-adjust`.

- `r` Renumber the simple $f_n:N$ footnotes. Automatic renumbering after each insertion/deletion can be configured using the option `org-footnote-auto-adjust`.
- `S` Short for first `r`, then `s` action.
- `n` Normalize the footnotes by collecting all definitions (including inline definitions) into a special section, and then numbering them in sequence. The references will then also be numbers. This is meant to be the final step before finishing a document (e.g., sending off an email).
- `d` Delete the footnote at point, and all definitions of and references to it.

Depending on the variable `org-footnote-auto-adjust`², renumbering and sorting footnotes can be automatic after each insertion or deletion.

`C-c C-c`

If the cursor is on a footnote reference, jump to the definition. If it is a the definition, jump back to the reference. When called at a footnote location with a prefix argument, offer the same menu as `C-c C-x f`.

`C-c C-o` or `mouse-1/2`

Footnote labels are also links to the corresponding definition/reference, and you can use the usual commands to follow these links.

`C-c '`

`C-c '`

Edit the footnote definition corresponding to the reference at point in a separate window. This may be useful if editing footnotes in a narrowed buffer. The window can be closed by pressing `C-c '`.

9.12.10.1 Footnotes

[1] The corresponding in-buffer setting is: `#+STARTUP: fninline` or `#+STARTUP: nofninline`

[2] the corresponding in-buffer options are `fnadjust` and `nofnadjust`.

9.12.11 11

The Orgstr uct minor mode

If you like the intuitive way the Org mode structure editing and list formatting works, you might want to use these commands in other modes like Text mode or Mail mode as well. The minor mode `orgstruct-mode` makes this possible. Toggle the mode with `M-x orgstruct-mode RET`, or turn it on by default, for example in Message mode, with one of:

```
(add-hook 'message-mode-hook 'turn-on-orgstruct)
```

```
(add-hook 'message-mode-hook 'turn-on-orgstruct++)
```

When this mode is active and the cursor is on a line that looks to Org like a headline or the first line of a list item, most structure editing commands will work, even if the same keys normally have different functionality in the major mode you are using. If the cursor is not in one of those special lines, Orgstruct mode lurks silently in the shadows.

When you use `orgstruct++-mode`, Org will also export indentation and autofill settings into that mode, and detect item context after the first line of an item.

You can also use Org structure editing to fold and unfold headlines in *any* file, provided you defined `orgstruct-heading-prefix-regexp`: the regular expression must match the local prefix to use before Org's headlines. For example, if you set this variable to `";; "` in Emacs Lisp files, you will be able to fold and unfold headlines in Emacs Lisp commented lines. Some commands like `org-demote` are disabled when the prefix is set, but folding/unfolding will work correctly.

9.12.12 12 Org syntax

A reference document providing a formal description of Org's syntax is available as [a draft on Worg](#), written and maintained by Nicolas Goaziou. It defines Org's core internal concepts such as headlines, sections, affiliated keywords, (greater) elements and objects. Each part of an Org file falls into one of the categories above. To explore the abstract structure of an Org buffer, run this in a buffer:

```
M-: (org-element-parse-buffer) RET
```

It will output a list containing the buffer's content represented as an abstract structure. The export engine relies on the information stored in this list. Most interactive commands (e.g., for structure editing) also rely on the syntactic meaning of the surrounding context.

9.12.13 Stru ctural marku p eleme nts

1. [Document title](#): Where the title is taken from

9.12.13.1 Document title

The title of the exported document is taken from the special line

```
#+TITLE: This is the title of the document
```

If you are exporting only a subtree, its heading will become the title of the document. If the subtree has a property `EXPORT_TITLE`, that will take precedence.

2. [Headings and sections](#): The document structure as seen by the exporter

9.12.13.2 Headings and sections

The outline structure of the document as described in [Document structure](#), forms the basis for defining sections of the exported document. However, since the outline

structure is also used for (for example) lists of tasks, only the first three outline levels will be used as headings. Deeper levels will become itemized lists. You can change the location of this switch globally by setting the variable `org-export-headline-levels`, or on a per-file basis with a line

```
#+OPTIONS: H:4
```

3. [Table of contents](#): The if and where of the table of contents

9.12.13.3 Table of contents

The table of contents is normally inserted directly before the first headline of the file. The depth of the table is by default the same as the number of headline levels, but you can choose a smaller number, or turn off the table of contents entirely, by configuring the variable `org-export-with-toc`, or on a per-file basis with a line like

```
#+OPTIONS: toc:2          only include two levels in TOC
```

```
#+OPTIONS: toc:nil       no default TOC at all
```

If you would like to move the table of contents to a different location, you should turn off the default table using `org-export-with-toc` or `#+OPTIONS` and insert `#+TOC: headlines N` at the desired location(s).

```
#+OPTIONS: toc:nil       no default TOC
```

```
...
```

```
#+TOC: headlines 2      insert TOC here, with two headline levels
```

Moreover, if you append 'local' parameter, the table contains only entries for the children of the current section¹. In this case, any depth parameter becomes relative to the current level.

```
* Section
```

```
#+TOC: headlines 1 local insert local TOC, with direct children only
```

The same `TOC` keyword can also generate a list of all tables (resp. all listings) with a caption in the document.

```
#+TOC: listings        build a list of listings
```

```
#+TOC: tables          build a list of tables
```

The headline's title usually determines its corresponding entry in a table of contents. However, it is possible to specify an alternative title by setting `ALT_TITLE` property accordingly. It will then be used when building the table.

9.12.13.4 Footnotes

[1] For LaTeX export, this feature requires the `titletoc` package. Note that `titletoc` must be loaded *before* `hyperref`. Thus, you may have to customize `org-latex-default-packages-alist`.

4. [Lists](#): Lists

9.12.13.5 Lists

Plain lists as described in [Plain lists](#), are translated to the back-end's syntax for such lists. Most back-ends support unordered, ordered, and description lists.

5. [Paragraphs](#): Paragraphs

9.12.13.6 Paragraphs, line breaks, and quoting

Paragraphs are separated by at least one empty line. If you need to enforce a line break within a paragraph, use `'\'` at the end of a line.

To keep the line breaks in a region, but otherwise use normal formatting, you can use this construct, which can also be used to format poetry.

```
#+BEGIN_VERSE

Great clouds overhead

Tiny black birds rise and fall

Snow covers Emacs

-- AlexSchroeder

#+END_VERSE
```

When quoting a passage from another document, it is customary to format this as a paragraph that is indented on both the left and the right margin. You can include quotations in Org mode documents like this:

```
#+BEGIN_QUOTE

Everything should be made as simple as possible,

but not any simpler -- Albert Einstein

#+END_QUOTE
```

If you would like to center some text, do it like this:

```
#+BEGIN_CENTER

Everything should be made as simple as possible, \\

but not any simpler

#+END_CENTER
```

6. [Footnote markup](#): Footnotes

9.12.13.7 Footnote markup

Footnotes defined in the way described in [Footnotes](#), will be exported by all backends. Org allows multiple references to the same note, and multiple footnotes side by side.

7. [Emphasis and monospace](#): Bold, italic, etc.

9.12.13.8 Emphasis and monospace

You can make words ***bold***, */italic/*, underlined, =verbatim= and ~code~, and, if you must, '+strike-through+'. Text in the code and verbatim string is not processed for Org mode specific syntax, it is exported verbatim.

To turn off fontification for marked up text, you can set `org-fontify-emphasized-text` to `nil`. To narrow down the list of available markup syntax, you can customize `org-emphasis-alist`. To fine tune what characters are allowed before and after the markup characters, you can tweak `org-emphasis-regexp-components`. Beware that changing one of the above variables will no take effect until you reload Org, for which you may need to restart Emacs.

8. [Horizontal rules](#): Make a line

9.12.13.9 Horizontal rules

A line consisting of only dashes, and at least 5 of them, will be exported as a horizontal line.

9. [Comment lines](#): What will *not* be exported

9.12.13.10 Comment lines

Lines starting with zero or more whitespace characters followed by one '#' and a whitespace are treated as comments and, as such, are not exported.

Likewise, regions surrounded by '#+BEGIN_COMMENT' ... '#+END_COMMENT' are not exported.

Finally, a 'COMMENT' keyword at the beginning of an entry, but after any other keyword or priority cookie, comments out the entire subtree. In this case, the subtree is not exported and no code block within it is executed either¹. The command below helps changing the comment status of a headline.

```
C-c ;
```

Toggle the 'COMMENT' keyword at the beginning of an entry.

9.12.13.11 Footnotes

^[1] For a less drastic behavior, consider using a select tag (see [Export settings](#)) instead.

9.12.14 Export settings

Export options can be set: globally with variables; for an individual file by making variables buffer-local with in-buffer settings (see [In-buffer settings](#)), by setting individual keywords, or by specifying them in a compact form with the #+OPTIONS keyword; or for a tree by setting properties (see [Properties and columns](#)). Options set at a specific level override options set at a more general level.

In-buffer settings may appear anywhere in the file, either directly or indirectly through a file included using ‘#+SETUPFILE: filename’ syntax. Option keyword sets tailored to a particular back-end can be inserted from the export dispatcher (see [The export dispatcher](#)) using the `Insert template` command by pressing <#>. To insert keywords individually, a good way to make sure the keyword is correct is to type #+ and then to use M-<TAB> for completion.

The export keywords available for every back-end, and their equivalent global variables, include:

‘AUTHOR’

The document author (`user-full-name`).

‘CREATOR’

Entity responsible for output generation (`org-export-creator-string`).

‘DATE’

A date or a time-stamp¹.

‘EMAIL’

The email address (`user-mail-address`).

‘LANGUAGE’

The language used for translating some strings (`org-export-default-language`). E.g., ‘#+LANGUAGE: fr’ will tell Org to translate *File* (english) into *Fichier* (french) in the clocktable.

‘SELECT_TAGS’

The tags that select a tree for export (`org-export-select-tags`). The default value is `:export:`. Within a subtree tagged with `:export:`, you can still exclude entries with `:noexport:` (see below). When headlines are selectively exported with `:export:` anywhere in a file, text before the first headline is ignored.

‘EXCLUDE_TAGS’

The tags that exclude a tree from export (`org-export-exclude-tags`). The default value is `:noexport:`. Entries with the `:noexport:` tag will be unconditionally excluded from the export, even if they have an `:export:` tag. Code blocks contained in excluded subtrees will still be executed during export even though the subtree is not exported.

‘TITLE’

The title to be shown. You can use several such keywords for long titles.

The `#+OPTIONS` keyword is a compact² form that recognizes the following arguments:

' :

Toggle smart quotes (`org-export-with-smart-quotes`).

* :

Toggle emphasized text (`org-export-with-emphasize`).

- :

Toggle conversion of special strings (`org-export-with-special-strings`).

::

Toggle fixed-width sections (`org-export-with-fixed-width`).

< :

Toggle inclusion of any time/date active/inactive stamps (`org-export-with-timestamps`).

\n :

Toggle line-break-preservation (`org-export-preserve-breaks`).

^ :

Toggle TeX-like syntax for sub- and superscripts. If you write "`^:{"`", '`a_{b}`' will be interpreted, but the simple '`a_b`' will be left as it is (`org-export-with-sub-superscripts`).

arch :

Configure export of archived trees. Can be set to `headline` to only process the headline, skipping its contents (`org-export-with-archived-trees`).

author :

Toggle inclusion of author name into exported file (`org-export-with-author`).

c :

Toggle inclusion of CLOCK keywords (`org-export-with-clocks`).

creator :

Toggle inclusion of creator info into exported file (`org-export-with-creator`).

d :

Toggle inclusion of drawers, or list drawers to include (`org-export-with-drawers`).

date:

Toggle inclusion of a date into exported file (`org-export-with-date`).

e:

Toggle inclusion of entities (`org-export-with-entities`).

email:

Toggle inclusion of the author's e-mail into exported file (`org-export-with-email`).

f:

Toggle the inclusion of footnotes (`org-export-with-footnotes`).

H:

Set the number of headline levels for export (`org-export-headline-levels`). Below that level, headlines are treated differently. In most back-ends, they become list items.

inline:

Toggle inclusion of inlinetasks (`org-export-with-inlinetasks`).

num:

Toggle section-numbers (`org-export-with-section-numbers`). It can also be set to a number 'n', so only headlines at that level or above will be numbered. Finally, irrespective of the level of a specific headline, the numbering of it can be disabled by setting the `UNNUMBERED` property to `non-nil`. This also affects subheadings.

p:

Toggle export of planning information (`org-export-with-planning`). "Planning information" is the line containing the `SCHEDULED:`, the `DEADLINE:` or the `CLOSED:` cookies or a combination of them.

pri:

Toggle inclusion of priority cookies (`org-export-with-priority`).

prop:

Toggle inclusion of property drawers, or list properties to include (`org-export-with-properties`).

stat:

Toggle inclusion of statistics cookies (`org-export-with-statistics-cookies`).

tags:

Toggle inclusion of tags, may also be `not-in-toc` (`org-export-with-tags`).

tasks:

Toggle inclusion of tasks (TODO items), can be `nil` to remove all tasks, `todo` to remove DONE tasks, or a list of keywords to keep (`org-export-with-tasks`).

tex:

Configure export of LaTeX fragments and environments. It may be set to `verbatim` (`org-export-with-latex`).

timestamp:

Toggle inclusion of the creation time into exported file (`org-export-timestamp-file`).

title:

Toggle inclusion of title (`org-export-with-title`).

toc:

Toggle inclusion of the table of contents, or set the level limit (`org-export-with-toc`).

todo:

Toggle inclusion of TODO keywords into exported text (`org-export-with-todo-keywords`).

l:

Toggle inclusion of tables (`org-export-with-tables`).

When exporting only a subtree, each of the previous keywords³ can be overridden locally by special node properties. These begin with `'EXPORT_'`, followed by the name of the keyword they supplant. For example, `'DATE'` and `'OPTIONS'` keywords become, respectively, `'EXPORT_DATE'` and `'EXPORT_OPTIONS'` properties.

If `org-export-allow-bind-keywords` is `non-nil`, Emacs variables can become buffer-local during export by using the `BIND` keyword. Its syntax is `'#+BIND: variable value'`. This is particularly useful for in-buffer settings that cannot be changed using specific keywords.

The name of the output file to be generated is taken from the file associated to the buffer, when possible, or asked to you otherwise. For subtree export, you can also set `EXPORT_FILE_NAME` property. In all cases, only the base name of the file is retained, and a back-end specific extension is added.

9.12.14.1 Footnotes

[1] The variable `org-export-date-timestamp-format` defines how this time-stamp will be exported.

[2] If you want to configure many options this way, you can use several `#+OPTIONS` lines.

[3] With the exception of ‘SETUPFILE’.

9.13 TABLES

Org comes with a fast and intuitive table editor. Spreadsheet-like calculations are supported using the Emacs`calc` package (see [Calc](#)).

1. [Built-in table editor](#): Simple tables

9.13.1 The built-in table editor

Org makes it easy to format tables in plain ASCII. Any line with ‘|’ as the first non-whitespace character is considered part of a table. ‘|’ is also the column separator¹.

A table might look like this:

```
| Name | Phone | Age |
|-----+-----+-----|
| Peter | 1234 | 17 |
| Anna | 4321 | 25 |
```

A table is re-aligned automatically each time you press `<TAB>` or `<RET>` or `C-c C-c` inside the table. `<TAB>` also moves to the next field (`<RET>` to the next row) and creates new table rows at the end of the table or before horizontal lines. The indentation of the table is set by the first line. Any line starting with ‘|-’ is considered as a horizontal separator line and will be expanded on the next re-align to span the whole table width. So, to create the above table, you would only type

```
| Name | Phone | Age |
```

```
|-
```

and then press `<TAB>` to align the table and start filling in fields. Even faster would be to type `|Name|Phone|Age` followed by `C-c <RET>`.

When typing text into a field, Org treats ``, `<Backspace>`, and all character keys in a special way, so that inserting and deleting avoids shifting other fields. Also, when typing *immediately after the cursor was moved into a new field with* `<TAB>`, `S-<TAB>` or `<RET>`, the field is automatically made blank. If this behavior is too unpredictable for you, configure the options `org-enable-table-editor` and `org-table-auto-blank-field`.

Creation and conversion

```
C-c | (org-table-create-or-convert-from-region)
```

Convert the active region to a table. If every line contains at least one TAB character, the function assumes that the material is tab separated. If every line contains a comma, comma-separated values (CSV) are assumed. If not, lines are split at whitespace into fields. You can use a prefix argument to force a specific separator: `C-u` forces CSV, `C-u C-u` forces TAB, `C-u C-u C-u` will prompt for a regular expression to match the separator, and a numeric argument N indicates that at least N consecutive spaces, or alternatively a TAB will be the separator. If there is no active region, this command creates an empty Org table. But it is easier just to start typing, like `|Name|Phone|Age <RET> |- <TAB>`.

Re-aligning and field motion

```
C-c C-c (org-table-align)
```

Re-align the table and don't move to another field.

```
C-c SPC (org-table-blank-field)
```

Blank the field at point.

```
<TAB> (org-table-next-field)
```

Re-align the table, move to the next field. Creates a new row if necessary.

```
S-<TAB> (org-table-previous-field)
```

Re-align, move to previous field.

```
<RET> (org-table-next-row)
```

Re-align the table and move down to next row. Creates a new row if necessary. At the beginning or end of a line, <RET> still does NEWLINE, so it can be used to split a table.

M-a (org-table-beginning-of-field)

Move to beginning of the current table field, or on to the previous field.

M-e (org-table-end-of-field)

Move to end of the current table field, or on to the next field.

Column and row editing

M-<left> (org-table-move-column-left)

M-<right> (org-table-move-column-right)

Move the current column left/right.

M-S-<left> (org-table-delete-column)

Kill the current column.

M-S-<right> (org-table-insert-column)

Insert a new column to the left of the cursor position.

M-<up> (org-table-move-row-up)

M-<down> (org-table-move-row-down)

Move the current row up/down.

M-S-<up> (org-table-kill-row)

Kill the current row or horizontal line.

M-S-<down> (org-table-insert-row)

Insert a new row above the current row. With a prefix argument, the line is created below the current one.

C-c - (org-table-insert-hline)

Insert a horizontal line below current row. With a prefix argument, the line is created above the current line.

C-c <RET> (org-table-hline-and-move)

Insert a horizontal line below current row, and move the cursor into the row below that line.

C-c ^ (org-table-sort-lines)

Sort the table lines in the region. The position of point indicates the column to be used for sorting, and the range of lines is the range between the nearest horizontal separator lines, or the entire table. If point is before the first column, you will be prompted for the sorting column. If there is an active region, the mark specifies the first line and the sorting column, while point should be in the last line to be included into the sorting. The command prompts for the sorting type (alphabetically, numerically, or by time). You can sort in normal or reverse order. You can also supply your own key extraction and comparison functions. When called with a prefix argument, alphabetic sorting will be case-sensitive.

Regions

`C-c C-x M-w` (org-table-copy-region)

Copy a rectangular region from a table to a special clipboard. Point and mark determine edge fields of the rectangle. If there is no active region, copy just the current field. The process ignores horizontal separator lines.

`C-c C-x C-w` (org-table-cut-region)

Copy a rectangular region from a table to a special clipboard, and blank all fields in the rectangle. So this is the “cut” operation.

`C-c C-x C-y` (org-table-paste-rectangle)

Paste a rectangular region into a table. The upper left corner ends up in the current field. All involved fields will be overwritten. If the rectangle does not fit into the present table, the table is enlarged as needed. The process ignores horizontal separator lines.

`M-<RET>` (org-table-wrap-region)

Split the current field at the cursor position and move the rest to the line below. If there is an active region, and both point and mark are in the same column, the text in the column is wrapped to minimum width for the given number of lines. A numeric prefix argument may be used to change the number of desired lines. If there is no region, but you specify a prefix argument, the current field is made blank, and the content is appended to the field above.

Calculations

`C-c +` (org-table-sum)

Sum the numbers in the current column, or in the rectangle defined by the active region. The result is shown in the echo area and can be inserted with `C-y`.

`S-<RET>` (org-table-copy-down)

When current field is empty, copy from first non-empty field above. When not empty, copy current field down to next row and move cursor along with it. Depending on the option `org-table-copy-increment`, integer field values will be incremented during copy. Integers that are too large will not be incremented. Also, a `0` prefix argument temporarily disables the increment. This key is also used by shift-selection and related modes (see [Conflicts](#)).

Miscellaneous

`C-c `` (org-table-edit-field)

Edit the current field in a separate window. This is useful for fields that are not fully visible (see [Column width and alignment](#)). When called with a `C-u` prefix, just make the full field visible, so that it can be edited in place. When called with two `C-u` prefixes, make the editor window follow the cursor through the table and always show the current field. The follow mode exits automatically when the cursor leaves the table, or when you repeat this command with `C-u C-u C-c ``.

`M-x org-table-import RET`

Import a file as a table. The table should be TAB or whitespace separated. Use, for example, to import a spreadsheet table or data from a database, because these programs generally can write TAB-separated text files. This command works by inserting the file into the buffer and then converting the region to a table. Any prefix argument is passed on to the converter, which uses it to determine the separator.

`C-c |` (org-table-create-or-convert-from-region)

Tables can also be imported by pasting tabular text into the Org buffer, selecting the pasted text with `C-x C-x` and then using the `C-c |` command (see above under *Creation and conversion*).

`M-x org-table-export RET`

Export the table, by default as a TAB-separated file. Use for data exchange with, for example, spreadsheet or database programs. The format used to export the file can be configured in the option `org-table-export-default-format`. You may also use


```

| 1 | one | | | one | |
| 2 | two | | ----\ | 2 | two |
| 3 | This is a long chunk of text | | ----/ | 3 | This=> |
| 4 | four | | | four |
|----+-----| |----+-----|

```

Fields that are wider become clipped and end in the string ‘=>’. Note that the full text is still in the buffer but is hidden. To see the full text, hold the mouse over the field—a tool-tip window will show the full content. To edit such a field, use the command `C-c `` (that is `C-c` followed by the grave accent). This will open a new window with the full field. Edit it and finish with `C-c C-c`.

When visiting a file containing a table with narrowed columns, the necessary character hiding has not yet happened, and the table needs to be aligned before it looks nice. Setting the option `org-startup-align-all-tables` will realign all tables in a file upon visiting, but also slow down startup. You can also set this option on a per-file basis with:

```

#+STARTUP: align

#+STARTUP: noalign

```

If you would like to overrule the automatic alignment of number-rich columns to the right and of string-rich column to the left, you can use ‘<r>’, ‘<c>’² or ‘<l>’ in a similar fashion. You may also combine alignment and field width like this: ‘<r10>’. Lines which only contain these formatting cookies will be removed automatically when exporting the document.

9.13.2.1 Footnotes

[1] This feature does not work on XEmacs.

[2] Centering does not work inside Emacs, but it does have an effect when exporting to HTML.

3. [Column groups](#): Grouping to trigger vertical lines

9.13.3 Column groups

When Org exports tables, it does so by default without vertical lines because that is visually more satisfying in general. Occasionally however, vertical lines can be useful to structure a table into groups of columns, much like horizontal lines can do for groups of rows. In order to specify column groups, you can use a special row where the first field contains only '/'. The further fields can either contain '<' to indicate that this column should start a group, '>' to indicate the end of a column, or '<>' (no space between '<' and '>') to make a column a group of its own. Boundaries between column groups will upon export be marked with vertical lines. Here is an example:

```
| N | N^2 | N^3 | N^4 | ~sqrt(n)~ | ~sqrt[4](N)~ |
|---+-----+-----+-----+-----+-----+
| / | < | | > | < | > |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 4 | 8 | 16 | 1.4142 | 1.1892 |
| 3 | 9 | 27 | 81 | 1.7321 | 1.3161 |
|---+-----+-----+-----+-----+-----+

#+TBLFM:
$2=$1^2::$3=$1^3::$4=$1^4::$5=sqrt($1)::6=sqrt(sqrt(($1)))
```

It is also sufficient to just insert the column group starters after every vertical line you would like to have:

```
| N | N^2 | N^3 | N^4 | sqrt(n) | sqrt[4](N) |
|---+-----+-----+-----+-----+-----+
| / | < | | < | | |
```

4. [Orgtbl mode](#): The table editor as minor mode

9.13.4 The Orgtbl

minor mode

If you like the intuitive way the Org table editor works, you might also want to use it in other modes like Text mode or Mail mode. The minor mode Orgtbl mode makes this possible. You can always toggle the mode with `M-x orgtbl-mode RET`. To turn it on by default, for example in Message mode, use

```
(add-hook 'message-mode-hook 'turn-on-orgtbl)
```

Furthermore, with some special setup, it is possible to maintain tables in arbitrary syntax with Orgtbl mode. For example, it is possible to construct LaTeX tables with the underlying ease and power of Orgtbl mode, including spreadsheet capabilities. For details, see [Tables in arbitrary syntax](#).

5. [The spreadsheet](#): The table editor has spreadsheet capabilities

9.13.5 The spreadsheet

The table editor makes use of the Emacs `calc` package to implement spreadsheet-like capabilities. It can also evaluate Emacs Lisp forms to derive fields from other fields. While fully featured, Org's implementation is not identical to other spreadsheets. For example, Org knows the concept of a *column formula* that will be applied to all non-header fields in a column without having to copy the formula to each relevant field. There is also a formula debugger, and a formula editor with features for highlighting fields in the table corresponding to the references at the point in the formula, moving these references by arrow keys

1. [References](#): How to refer to another field or range

9.13.5.1 References

To compute fields in the table from other fields, formulas must reference other fields or ranges. In Org, fields can be referenced by name, by absolute coordinates, and by relative coordinates. To find out what the coordinates of a field are, press `C-c ?` in that field, or press `C-c }` to toggle the display of a grid.

9.13.5.1.1 Field references

Formulas can reference the value of another field in two ways. Like in any other spreadsheet, you may reference fields with a letter/number combination like B3, meaning the 2nd field in the 3rd row. However, Org prefers¹ to use another, more general representation that looks like this:

```
@row$column
```

Column specifications can be absolute like \$1, \$2,...\$N, or relative to the current column (i.e., the column of the field which is being computed) like \$+1 or \$-2. \$< and \$> are immutable references to the first and last column, respectively, and you can use \$>>> to indicate the third column from the right.

The row specification only counts data lines and ignores horizontal separator lines (hlines). Like with columns, you can use absolute row numbers @1, @2,...@N, and row numbers relative to the current row like @+3 or @-1. @< and @> are immutable references the first and last² row in the table, respectively. You may also specify the row relative to one of the hlines: @I refers to the first hline, @II to the second, etc. @-I refers to the first such line above the current line, @+I to the first such line below the current line. You can also write @III+2 which is the second data line after the third hline in the table.

@0 and \$0 refer to the current row and column, respectively, i.e., to the row/column for the field being computed. Also, if you omit either the column or the row part of the reference, the current row/column is implied.

Org's references with *unsigned* numbers are fixed references in the sense that if you use the same reference in the formula for two different fields, the same field will be referenced each time. Org's references with *signed* numbers are floating references because the same reference operator can reference different fields depending on the field being calculated by the formula.

Here are a few examples:

@2\$3	2nd row, 3rd column (same as C2)
\$5	column 5 in the current row (same as E&)
@2	current column, row 2
@-1\$-3	the field one row up, three columns to the left
@-I\$2	field just under hline above current row, column 2
@>\$5	field in the last row, in column 5

9.13.5.1.2 Range references

You may reference a rectangular range of fields by specifying two field references connected by two dots '..'. If both fields are in the current row, you may simply use '\$2..\$7', but if at least one field is in a different row, you need to use the general @row\$column format at least for the first field (i.e the reference must start with '@' in order to be interpreted correctly). Examples:

\$1..\$3	first three fields in the current row
\$P..\$Q	range, using column names (see under Advanced)
\$<<<..\$>>	start in third column, continue to the last but one
@2\$1..@4\$3	6 fields between these two fields (same as A2..C4)
@-1\$-2..@-1	3 fields in the row above, starting from 2 columns on the left
@I..II	between first and second hline, short for @I..@II

Range references return a vector of values that can be fed into Calc vector functions. Empty fields in ranges are normally suppressed, so that the vector contains only the non-empty fields. For other options with the mode switches 'E', 'N' and examples see [Formula syntax for Calc](#).

9.13.5.1.3 Field coordinates in formulas

One of the very first actions during evaluation of Calc formulas and Lisp formulas is to substitute @# and \$# in the formula with the row or column number of the field where the current result will go to. The traditional Lisp formula equivalents are org-table-current-dline and org-table-current-column. Examples:

```
if(@# % 2, $#, string(""))
```

Insert column number on odd rows, set field to empty on even rows.

```
$2 = '(identity remote(FOO, @@#$1))
```

Copy text or values of each row of column 1 of the table named FOO into column 2 of the current table.

```
@3 = 2 * remote(FOO, @1$$#)
```

Insert the doubled value of each column of row 1 of the table named FOO into row 3 of the current table.

For the second/third example, the table named `FOO` must have at least as many rows/columns as the current table. Note that this is inefficient³ for large number of rows/columns.

9.13.5.1.4 Named references

'\$name' is interpreted as the name of a column, parameter or constant. Constants are defined globally through the option `org-table-formula-constants`, and locally (for the file) through a line like

```
#+CONSTANTS: c=299792458. pi=3.14 eps=2.4e-6
```

Also properties (see [Properties and columns](#)) can be used as constants in table formulas: for a property ':Xyz:' use the name '\$PROP_Xyz', and the property will be searched in the current outline entry and in the hierarchy above it. If you have the `constants.el` package, it will also be used to resolve constants, including natural constants like '\$h' for Planck's constant, and units like '\$km' for kilometers⁴. Column names and parameters can be specified in special table lines. These are described below, see [Advanced features](#). All names must start with a letter, and further consist of letters and numbers.

9.13.5.1.5 Remote references

You may also reference constants, fields and ranges from a different table, either in the current file or even in a different file. The syntax is

```
remote (NAME-OR-ID, REF)
```

where `NAME` can be the name of a table in the current file as set by a `#+NAME: Name` line before the table. It can also be the ID of an entry, even in a different file, and the reference then refers to the first table in that entry. `REF` is an absolute field or range reference as described above for example `@3$3` or `$somename`, valid in the referenced table.

Indirection of `NAME-OR-ID`: When `NAME-OR-ID` has the format `@ROW$COLUMN` it will be substituted with the name or ID found in this field of the current table. For example `remote($1, @>$2) => remote(year_2013, @>$1)`. The format `B3` is not supported because it can not be distinguished from a plain table name or ID.

9.13.5.2 Footnotes

[1] Org will understand references typed by the user as ‘B4’, but it will not use this syntax when offering a formula for editing. You can customize this behavior using the option `org-table-use-standard-references`.

[2] For backward compatibility you can also use special names like `$LR5` and `$LR12` to refer in a stable way to the 5th and 12th field in the last row of the table. However, this syntax is deprecated, it should not be used for new documents. Use `@>$` instead.

[3] The computation time scales as $O(N^2)$ because the table named `FOO` is parsed for each field to be read.

[4] `constants.el` can supply the values of constants in two different unit systems, `SI` and `cgs`. Which one is used depends on the value of the variable `constants-unit-system`. You can use the `#+STARTUP` options `constSI` and `constcgs` to set this value for the current buffer.

2. [Formula syntax for Calc](#): Using Calc to compute stuff

9.13.5.3 Formula syntax for Calc

A formula can be any algebraic expression understood by the Emacs `Calc` package. Note that `calc` has the non-standard convention that ‘/’ has lower precedence than ‘*’, so that ‘a/b*c’ is interpreted as ‘a/(b*c)’. Before evaluation by `calc-eval` (see [calc-eval](#)), variable substitution takes place according to the rules described above. The range vectors can be directly fed into the Calc vector functions like ‘`vmean`’ and ‘`vsum`’.

A formula can contain an optional mode string after a semicolon. This string consists of flags to influence Calc and other modes during execution. By default, Org uses the standard Calc modes (precision 12, angular units degrees, fraction and symbolic modes off). The display format, however, has been changed to `(float 8)` to keep tables compact. The default settings can be configured using the option `org-calc-default-modes`.

List of modes:

p20

Set the internal Calc calculation precision to 20 digits.

n3, s3, e2, f4

Normal, scientific, engineering or fixed format of the result of Calc passed back to Org. Calc formatting is unlimited in precision as long as the Calc calculation precision is greater.

D, R

Degree and radian angle modes of Calc.

F, S

Fraction and symbolic modes of Calc.

T, t

Duration computations in Calc or Lisp, see [Durations and time values](#).

E

If and how to consider empty fields. Without ‘E’ empty fields in range references are suppressed so that the Calc vector or Lisp list contains only the non-empty fields. With ‘E’ the empty fields are kept. For empty fields in ranges or empty field references the value ‘nan’ (not a number) is used in Calc formulas and the empty string is used for Lisp formulas. Add ‘N’ to use 0 instead for both formula types. For the value of a field the mode ‘N’ has higher precedence than ‘E’.

N

Interpret all fields as numbers, use 0 for non-numbers. See the next section to see how this is essential for computations with Lisp formulas. In Calc formulas it is used only occasionally because there number strings are already interpreted as numbers without ‘N’.

L

Literal, for Lisp formulas only. See the next section.

Unless you use large integer numbers or high-precision-calculation and `-display` for floating point numbers you may alternatively provide a ‘printf’ format specifier to reformat the Calc result after it has been passed back to Org instead of letting Calc already do the formatting¹. A few examples:

<code>\$1+\$2</code>	Sum of first and second field
<code>\$1+\$2;%.2f</code>	Same, format result to two decimals
<code>exp(\$2)+exp(\$1)</code>	Math functions can be used
<code>\$0;%.1f</code>	Reformat current cell to 1 decimal

<code>(\$3-32)*5/9</code>	Degrees F -> C conversion
<code>\$c/\$1/\$cm</code>	Hz -> cm conversion, using <code>constants.el</code>
<code>tan(\$1);Dp3s1</code>	Compute in degrees, precision 3, display SCI 1
<code>sin(\$1);Dp3%.1e</code>	Same, but use printf specifier for display
<code>taylor(\$3,x=7,2)</code>	Taylor series of \$3, at x=7, second degree

Calc also contains a complete set of logical operations, (see [Logical Operations](#)). For example

```
if($1 < 20, teen, string(""))
```

"teen" if age \$1 is less than 20, else the Org table result field is set to empty with the empty string.

```
if("$1" == "nan" || "$2" == "nan", string(""), $1 + $2); E f-1
```

Sum of the first two columns. When at least one of the input fields is empty the Org table result field is set to empty. 'E' is required to not convert empty fields to 0. 'f-1' is an optional Calc format string similar to '%.1f' but leaves empty results empty.

```
if(typeof(vmean($1..$7)) == 12, string(""), vmean($1..$7); E
```

Mean value of a range unless there is any empty field. Every field in the range that is empty is replaced by 'nan' which lets 'vmean' result in 'nan'. Then 'typeof == 12' detects the 'nan' from 'vmean' and the Org table result field is set to empty. Use this when the sample set is expected to never have missing values.

```
if("$1..$7" == "[]", string(""), vmean($1..$7))
```

Mean value of a range with empty fields skipped. Every field in the range that is empty is skipped. When all fields in the range are empty the mean value is not defined and the Org table result field is set to empty. Use this when the sample set can have a variable size.

```
vmean($1..$7); EN
```

To complete the example before: Mean value of a range with empty fields counting as samples with value 0. Use this only when incomplete sample sets should be padded with 0 to the full size.

You can add your own Calc functions defined in Emacs Lisp with `defmath` and use them in formula syntax for Calc.

9.13.5.4 Footnotes

[1] The ‘printf’ reformatting is limited in precision because the value passed to it is converted into an ‘integer’ or ‘double’. The ‘integer’ is limited in size by truncating the signed value to 32 bits. The ‘double’ is limited in precision to 64 bits overall which leaves approximately 16 significant decimal digits.

3. [Formula syntax for Lisp](#): Writing formulas in Emacs Lisp

9.13.5.5 Emacs Lisp forms as formulas

It is also possible to write a formula in Emacs Lisp. This can be useful for string manipulation and control structures, if Calc's functionality is not enough.

If a formula starts with an apostrophe followed by an opening parenthesis, then it is evaluated as a Lisp form. The evaluation should return either a string or a number. Just as with `calc` formulas, you can specify modes and a `printf` format after a semicolon.

With Emacs Lisp forms, you need to be conscious about the way field references are interpolated into the form. By default, a reference will be interpolated as a Lisp string (in double-quotes) containing the field. If you provide the ‘N’ mode switch, all referenced elements will be numbers (non-number fields will be zero) and interpolated as Lisp numbers, without quotes. If you provide the ‘L’ flag, all fields will be interpolated literally, without quotes. I.e., if you want a reference to be interpreted as a string by the Lisp form, enclose the reference operator itself in double-quotes, like “\$3”. Ranges are inserted as space-separated fields, so you can embed them in list or vector syntax.

Here are a few examples—note how the ‘N’ mode is used when we do computations in Lisp:

```
'(concat (substring $1 1 2) (substring $1 0 1) (substring $1 2))
```

Swap the first two characters of the content of column 1.

```
'(+ $1 $2);N
```

Add columns 1 and 2, equivalent to Calc's `$1+$2`.

```
'(apply '+ '($1..$4));N
```

Compute the sum of columns 1 to 4, like Calc's `vsum($1..$4)`.

4. [Durations and time values](#): How to compute durations and time values

9.13.5.6 Durations and time values

If you want to compute time values use the `T` flag, either in Calc formulas or Emacs formulas:

Task 1	Task 2	Total
2:12	1:47	03:59:00
3:02:20	-2:07:00	0.92

```
#+TBLFM: @2$3=$1+$2;T::@3$3=$1+$2;t
```

Input duration values must be of the form `HH:MM[:SS]`, where seconds are optional. With the `T` flag, computed durations will be displayed as `HH:MM:SS` (see the first formula above). With the `t` flag, computed durations will be displayed according to the value of the option `org-table-duration-custom-format`, which defaults to `'hours'` and will display the result as a fraction of hours (see the second formula in the example above).

Negative duration values can be manipulated as well, and integers will be considered as seconds in addition and subtraction.

5. [Field and range formulas](#): Formula for specific (ranges of) fields

9.13.5.7 Field and range formulas

To assign a formula to a particular field, type it directly into the field, preceded by `':='`, for example `':=vsum(@II..III)'`. When you press `<TAB>` or `<RET>` or `C-c C-c` with the cursor still in the field, the formula will be stored as the formula for this field, evaluated, and the current field will be replaced with the result.

Formulas are stored in a special line starting with `'#+TBLFM:'` directly below the table. If you type the equation in the 4th field of the 3rd data line in the table, the formula will look like `'@3$4=$1+$2'`. When inserting/deleting/swapping columns and rows with the appropriate commands, *absolute references* (but not relative ones) in stored formulas are modified in order to still reference the same field. To avoid this, in particular in range references, anchor ranges at the table borders (using `@<`, `@>`, `$<`, `$>`), or at hlines using the `@I` notation. Automatic adaptation of field references does of course not happen if you edit the table structure with normal editing commands—then you must fix the equations yourself.

Instead of typing an equation into the field, you may also use the following command

`C-u C-c = (org-table-eval-formula)`

Install a new formula for the current field. The command prompts for a formula with default taken from the ‘#+TBLFM:’ line, applies it to the current field, and stores it.

The left-hand side of a formula can also be a special expression in order to assign the formula to a number of different fields. There is no keyboard shortcut to enter such range formulas. To add them, use the formula editor (see [Editing and debugging formulas](#)) or edit the #+TBLFM: line directly.

`$2=`

Column formula, valid for the entire column. This is so common that Org treats these formulas in a special way, see [Column formulas](#).

`@3=`

Row formula, applies to all fields in the specified row. `@>=` means the last row.

`@1$2..@4$3=`

Range formula, applies to all fields in the given rectangular range. This can also be used to assign a formula to some but not all fields in a row.

`$name=`

Named field, see [Advanced features](#).

6. [Column formulas](#): Formulas valid for an entire column

9.13.5.8 Column formulas

When you assign a formula to a simple column reference like `$3=`, the same formula will be used in all fields of that column, with the following very convenient exceptions: (i) If the table contains horizontal separator hlines with rows above and below, everything before the first such hline is considered part of the table *header* and will not be modified by column formulas. Therefore a header is mandatory when you use column formulas and want to add hlines to group rows, like for example to separate a total row at the bottom from the summand rows above. (ii) Fields that already get a value from a field/range formula will be left alone by column formulas. These conditions make column formulas very easy to use.

To assign a formula to a column, type it directly into any field in the column, preceded by an equal sign, like ‘`=$1+$2`’. When you press `<TAB>` or `<RET>` or `C-c`

`C-c` with the cursor still in the field, the formula will be stored as the formula for the current column, evaluated and the current field replaced with the result. If the field contains only '=', the previously stored formula for this column is used. For each column, Org will only remember the most recently used formula. In the '#+TBLFM:' line, column formulas will look like '\$4=\$1+\$2'. The left-hand side of a column formula cannot be the name of column, it must be the numeric column reference or \$>.

Instead of typing an equation into the field, you may also use the following command:

```
C-c = (org-table-eval-formula)
```

Install a new formula for the current column and replace current field with the result of the formula. The command prompts for a formula, with default taken from the '#+TBLFM' line, applies it to the current field and stores it. With a numeric prefix argument (e.g., `C-5 C-c =`) the command will apply it to that many consecutive fields in the current column.

7. [Lookup functions](#): Lookup functions for searching tables

9.13.5.9 Lookup functions

Org has three predefined Emacs Lisp functions for lookups in tables.

```
(org-lookup-first VAL S-LIST R-LIST &optional PREDICATE)
```

Searches for the first element `S` in list `S-LIST` for which

```
(PREDICATE VAL S)
```

is `t`; returns the value from the corresponding position in list `R-LIST`. The default `PREDICATE` is `equal`. Note that the parameters `VAL` and `S` are passed to `PREDICATE` in the same order as the corresponding parameters are in the call to `org-lookup-first`, where `VAL` precedes `S-LIST`. If `R-LIST` is `nil`, the matching element `S` of `S-LIST` is returned.

```
(org-lookup-last VAL S-LIST R-LIST &optional PREDICATE)
```

Similar to `org-lookup-first` above, but searches for the *last* element for which `PREDICATE` is `t`.

```
(org-lookup-all VAL S-LIST R-LIST &optional PREDICATE)
```

Similar to `org-lookup-first`, but searches for *all* elements for which `PREDICATE` is `t`, and returns *all* corresponding values. This function can not be used by itself in a formula, because it returns a list of values. However, powerful lookups can be built when this function is combined with other Emacs Lisp functions.

If the ranges used in these functions contain empty fields, the `E` mode for the formula should usually be specified: otherwise empty fields will not be included in `S-LIST` and/or `R-LIST` which can, for example, result in an incorrect mapping from an element of `S-LIST` to the corresponding element of `R-LIST`.

These three functions can be used to implement associative arrays, count matching cells, rank results, group data etc. For practical examples see [this tutorial on Worg](#).

8. [Editing and debugging formulas](#): Fixing formulas

9.13.5.10 Editing and debugging formulas

You can edit individual formulas in the minibuffer or directly in the field. Org can also prepare a special buffer with all active formulas of a table. When offering a formula for editing, Org converts references to the standard format (like `B3` or `D&`) if possible. If you prefer to only work with the internal format (like `@3$2` or `$4`), configure the option `org-table-use-standard-references`.

`C-c =` or `C-u C-c =` (org-table-eval-formula)

Edit the formula associated with the current column/field in the minibuffer. See [Column formulas](#), and [Field and range formulas](#).

`C-u C-u C-c =` (org-table-eval-formula)

Re-insert the active formula (either a field formula, or a column formula) into the current field, so that you can edit it directly in the field. The advantage over editing in the minibuffer is that you can use the command `C-c ?`.

`C-c ?` (org-table-field-info)

While editing a formula in a table field, highlight the field(s) referenced by the reference at the cursor position in the formula.

`C-c }`

Toggle the display of row and column numbers for a table, using overlays (`org-table-toggle-coordinate-overlays`). These are updated each time the table is aligned; you can force it with `C-c C-c`.

`C-c {`

Toggle the formula debugger on and off (`org-table-toggle-formula-debugger`). See below.

`C-c '` (org-table-edit-formulas)

Edit all formulas for the current table in a special buffer, where the formulas will be displayed one per line. If the current field has an active formula, the cursor in the formula editor will mark it. While inside the special buffer, Org

will automatically highlight any field or range reference at the cursor position. You may edit, remove and add formulas, and use the following commands:

`C-c C-c` or `C-x C-s` (org-table-fedit-finish)

Exit the formula editor and store the modified formulas. With `C-u` prefix, also apply the new formulas to the entire table.

`C-c C-q` (org-table-fedit-abort)

Exit the formula editor without installing changes.

`C-c C-r` (org-table-fedit-toggle-ref-type)

Toggle all references in the formula editor between standard (like `B3`) and internal (like `@3$2`).

`<TAB>` (org-table-fedit-lisp-indent)

Pretty-print or indent Lisp formula at point. When in a line containing a Lisp formula, format the formula according to Emacs Lisp rules. Another `<TAB>` collapses the formula back again. In the open formula, `<TAB>` re-indents just like in Emacs Lisp mode.

`M-<TAB>` (lisp-complete-symbol)

Complete Lisp symbols, just like in Emacs Lisp mode.

`S-<up>/<down>/<left>/<right>`

Shift the reference at point. For example, if the reference is `B3` and you press `S-<right>`, it will become `C3`. This also works for relative references and for hline references.

`M-S-<up>` (org-table-fedit-line-up)

`M-S-<down>` (org-table-fedit-line-down)

Move the test line for column formulas in the Org buffer up and down.

`M-<up>` (org-table-fedit-scroll-down)

`M-<down>` (org-table-fedit-scroll-up)

Scroll the window displaying the table.

`C-c }`

Turn the coordinate grid in the table on and off.

Making a table field blank does not remove the formula associated with the field, because that is stored in a different line (the `'#+TBLFM'` line)—during the next recalculation the field will be filled again. To remove a formula from a field, you

have to give an empty reply when prompted for the formula, or to edit the '#+TBLFM' line.

You may edit the '#+TBLFM' directly and re-apply the changed equations with `C-c C-c` in that line or with the normal recalculation commands in the table.

9.13.5.10.1 Using multiple '#+TBLFM' lines

You may apply the formula temporarily. This is useful when you switch the formula. Place multiple '#+TBLFM' lines right after the table, and then press `C-c C-c` on the formula to apply. Here is an example:

```
| x | y |
|---+---|
| 1 |   |
| 2 |   |
#+TBLFM: $2=$1*1
#+TBLFM: $2=$1*2
```

Pressing `C-c C-c` in the line of '#+TBLFM: \$2=\$1*2' yields:

```
| x | y |
|---+---|
| 1 | 2 |
| 2 | 4 |
#+TBLFM: $2=$1*1
#+TBLFM: $2=$1*2
```

Note: If you recalculate this table (with `C-u C-c *`, for example), you will get the following result of applying only the first '#+TBLFM' line.

```
| x | y |
|---+---|
```

```
| 1 | 1 |
| 2 | 2 |

#+TBLFM: $2=$1*1

#+TBLFM: $2=$1*2
```

9.13.5.10.2 Debugging formulas

When the evaluation of a formula leads to an error, the field content becomes the string ‘#ERROR’. If you would like see what is going on during variable substitution and calculation in order to find a bug, turn on formula debugging in the `Tbl` menu and repeat the calculation, for example by pressing `C-u C-u C-c = <RET>` in a field. Detailed information will be displayed.

9. [Updating the table](#): Recomputing all dependent fields

9.13.5.11 Updating the table

Recalculation of a table is normally not automatic, but needs to be triggered by a command. See [Advanced features](#), for a way to make recalculation at least semi-automatic.

In order to recalculate a line of a table or the entire table, use the following commands:

```
C-c * (org-table-recalculate)
```

Recalculate the current row by first applying the stored column formulas from left to right, and all field/range formulas in the current row.

```
C-u C-c *
```

```
C-u C-c C-c
```

Recompute the entire table, line by line. Any lines before the first hline are left alone, assuming that these are part of the table header.

```
C-u C-u C-c * or C-u C-u C-c C-c (org-table-iterate)
```

Iterate the table by recomputing it until no further changes occur. This may be necessary if some computed fields use the value of other fields that are computed *later* in the calculation sequence.

```
M-x org-table-recalculate-buffer-tables RET
```

Recompute all tables in the current buffer.

M-x org-table-iterate-buffer-tables RET

Iterate all tables in the current buffer, in order to converge table-to-table dependencies.

10. [Advanced features](#): Field and column names, parameters and automatic recalc

9.13.5.12 Advanced features

If you want the recalculation of fields to happen automatically, or if you want to be able to assign *names*¹ to fields and columns, you need to reserve the first column of the table for special marking characters.

C-# (org-table-rotate-recalc-marks)

Rotate the calculation mark in first column through the states ‘ ’, ‘#’, ‘*’, ‘!’, ‘\$’. When there is an active region, change all marks in the region.

Here is an example of a table that collects exam results of students and makes use of these features:

```
|---+-----+-----+-----+-----+-----+-----|
|  | Student | Prob 1 | Prob 2 | Prob 3 | Total | Note |
|---+-----+-----+-----+-----+-----+-----|
| ! |          |      P1 |      P2 |      P3 |  Tot  |      |
| # | Maximum |      10 |      15 |      25 |   50  | 10.0 |
| ^ |          |      m1 |      m2 |      m3 |   mt  |      |
|---+-----+-----+-----+-----+-----+-----|
| # | Peter   |      10 |      8  |      23 |   41  | 8.2  |
| # | Sam     |      2  |      4  |      3  |    9  | 1.8  |
|---+-----+-----+-----+-----+-----+-----|
|  | Average |          |          |          | 25.0  |      |
| ^ |          |          |          |          |   at  |      |
```

```

| $ | max=50 | | | | | |
|---+-----+-----+-----+-----+-----+-----|

#+TBLFM:      $6=vsum($P1..$P3)::$7=10*$Tot/$max;%.1f::$at=vmean(@-
II..@-I);%.1f

```

Important: please note that for these special tables, recalculating the table with `C-u C-c *` will only affect rows that are marked '#' or '*', and fields that have a formula assigned to the field itself. The column formulas are not applied in rows with empty first field.

The marking characters have the following meaning:

'\!'

The fields in this line define names for the columns, so that you may refer to a column as '\$Tot' instead of '\$6'.

'\^'

This row defines names for the fields *above* the row. With such a definition, any formula in the table may use '\$m1' to refer to the value '10'. Also, if you assign a formula to a names field, it will be stored as '\$name=...'.

'_'

Similar to '^', but defines names for the fields in the row *below*.

'\\$',

Fields in this row can define *parameters* for formulas. For example, if a field in a '\$' row contains 'max=50', then formulas in this table can refer to the value 50 using '\$max'. Parameters work exactly like constants, only that they can be defined on a per-table basis.

'\#'

Fields in this row are automatically recalculated when pressing <TAB> or <RET> or `S-<TAB>` in this row. Also, this row is selected for a global recalculation with `C-u C-c *`. Unmarked lines will be left alone by this command.

'*'

Selects this line for global recalculation with `C-u C-c *`, but not for automatic recalculation. Use this when automatic recalculation slows down editing too much.

'\ ,'

9.13.5.13 Footnotes

[1] Such names must start by an alphabetic character and use only alphanumeric/underscore characters.

6. [Org-Plot](#): Plotting from org tables

9.13.6 Org-Plot

Org-Plot can produce graphs of information stored in org tables, either graphically or in ASCII-art.

9.13.6.1 Graphical plots using Gnuplot

Org-Plot produces 2D and 3D graphs using Gnuplot <http://www.gnuplot.info/> and gnuplot-mode <http://xafs.org/BruceRavel/GnuplotMode>. To see this in action, ensure that you have both Gnuplot and Gnuplot mode installed on your system, then call `C-c "g` or `M-x org-plot/gnuplot <RET>` on the following table.

```
#+PLOT: title:"Citas" ind:1 deps:(3) type:2d with:histograms
set:"yrange [0:]"
```

Sede	Max cites	H-index
Chile	257.72	21.39
Leeds	165.77	19.68
Sao Paolo	71.00	11.50
Stockholm	134.19	14.33
Morelia	257.56	17.67

Notice that Org Plot is smart enough to apply the table's headers as labels. Further control over the labels, type, content, and appearance of plots can be exercised through the `#+PLOT:` lines preceding a table. See below for a complete list of Org-plot options. The `#+PLOT:` lines are optional. For more information and examples see the Org-plot tutorial at <http://orgmode.org/worg/org-tutorials/org-plot.html>.

9.13.6.1.1 Plot Options

`set`

Specify any `gnuplot` option to be set when graphing.

`title`

Specify the title of the plot.

`ind`

Specify which column of the table to use as the `x` axis.

`deps`

Specify the columns to graph as a Lisp style list, surrounded by parentheses and separated by spaces for example `dep: (3 4)` to graph the third and fourth columns (defaults to graphing all other columns aside from the `ind` column).

`type`

Specify whether the plot will be `2d`, `3d`, or `grid`.

`with`

Specify a `with` option to be inserted for every `col` being plotted (e.g., `lines`, `points`, `boxes`, `impulses`, etc...). Defaults to `lines`.

`file`

If you want to plot to a file, specify "*path/to/desired/output-file*".

`labels`

List of labels to be used for the `deps` (defaults to the column headers if they exist).

`line`

Specify an entire line to be inserted in the Gnuplot script.

`map`

When plotting `3d` or `grid` types, set this to `t` to graph a flat mapping rather than a `3d` slope.

timefmt

Specify format of Org mode timestamps as they will be parsed by Gnuplot. Defaults to '%Y-%m-%d-%H:%M:%S'.

script

If you want total control, you can specify a script file (place the file name between double-quotes) which will be used to plot. Before plotting, every instance of \$datafile in the specified script will be replaced with the path to the generated data file. Note: even if you set this option, you may still want to specify the plot type, as that can impact the content of the data file.

9.13.6.2 ASCII bar plots

While the cursor is on a column, typing `C-c " a` or `M-x orgtbl-ascii-plot <RET>` create a new column containing an ASCII-art bars plot. The plot is implemented through a regular column formula. When the source column changes, the bar plot may be updated by refreshing the table, for example typing `C-u C-c *`.

```
| Sede           | Max cites |           |
|-----+-----+-----|
| Chile          |    257.72 | WWWWWWWWWWWW |
| Leeds          |    165.77 | WWWWWWWWh   |
| Sao Paolo     |     71.00 | WWW;        |
| Stockholm     |    134.19 | WWWWWW:     |
| Morelia       |    257.56 | WWWWWWWWWWWWH |
| Rochefourchat |     0.00  |           |

#+TBLFM: $3='(orgtbl-ascii-draw $2 0.0 257.72 12)
```

The formula is an elisp call:

```
(orgtbl-ascii-draw COLUMN MIN MAX WIDTH)
```

COLUMN

is a reference to the source column.

MIN MAX

are the minimal and maximal values displayed. Sources values outside this range are displayed as ‘too small’ or ‘too large’.

WIDTH

is the width in characters of the bar-plot. It defaults to ‘12’.

9.14 HYPERLINKS

Like HTML, Org provides links inside a file, external links to other files, Usenet articles, emails, and much more.

1. [Link format](#): How links in Org are formatted

9.14.1 Link format

Org will recognize plain URL-like links and activate them as clickable links. The general link format, however, looks like this:

```
[[link][description]]           or alternatively           [[link]]
```

Once a link in the buffer is complete (all brackets present), Org will change the display so that ‘description’ is displayed instead of ‘[[link][description]]’ and ‘link’ is displayed instead of ‘[[link]]’. Links will be highlighted in the face `org-link`, which by default is an underlined face. You can directly edit the visible part of a link. Note that this can be either the ‘link’ part (if there is no description) or the ‘description’ part. To edit also the invisible ‘link’ part, use `C-c C-1` with the cursor on the link.

If you place the cursor at the beginning or just behind the end of the displayed text and press `<BACKSPACE>`, you will remove the (invisible) bracket at that location. This makes the link incomplete and the internals are again displayed as plain text. Inserting the missing bracket hides the link internals again. To show the internal structure of all links, use the menu entry `Org->Hyperlinks->Literal links`.

2. [Internal links](#): Links to other places in the current file

9.14.2 Internal links

If the link does not look like a URL, it is considered to be internal in the current file. The most important case is a link like ‘[[#my-custom-id]]’ which will link to the entry with the `CUSTOM_ID` property ‘my-custom-id’. You are responsible yourself to make sure these custom IDs are unique in a file.

Links such as ‘[[My Target]]’ or ‘[[My Target][Find my target]]’ lead to a text search in the current file.

The link can be followed with `C-c C-o` when the cursor is on the link, or with a mouse click (see [Handling links](#)). Links to custom IDs will point to the corresponding headline. The preferred match for a text link is a *dedicated target*: the same string in double angular brackets, like ‘<<My Target>>’.

If no dedicated target exists, the link will then try to match the exact name of an element within the buffer. Naming is done with the `#+NAME` keyword, which has to be put in the line before the element it refers to, as in the following example

```
#+NAME: My Target

| a | table      |
|---+-----|
| of | four cells |
```

If none of the above succeeds, Org will search for a headline that is exactly the link text but may also include a TODO keyword and tags¹.

During export, internal links will be used to mark objects and assign them a number. Marked objects will then be referenced by links pointing to them. In particular, links without a description will appear as the number assigned to the marked object². In the following excerpt from an Org buffer

```
- one item

- <<target>>another item

Here we refer to item [[target]].
```

The last sentence will appear as ‘Here we refer to item 2’ when exported.

In non-Org files, the search will look for the words in the link text. In the above example the search would be for ‘my target’.

Following a link pushes a mark onto Org's own mark ring. You can return to the previous position with `C-c &`. Using this command several times in direct succession goes back to positions recorded earlier.

- [Radio targets](#): Make targets trigger links in plain text

9.14.2.1 Footnotes

[1] To insert a link targeting a headline, in-buffer completion can be used. Just type a star followed by a few optional letters into the buffer and press `M-<TAB>`. All headlines in the current buffer will be offered as completions.

[2] When targeting a `#+NAME` keyword, `#+CAPTION` keyword is mandatory in order to get proper numbering (see [Images and tables](#)).

3. [External links](#): URL-like links to the world

9.14.3 External links

Org supports links to files, websites, Usenet and email messages, BBDB database entries and links to both IRC conversations and their logs. External links are URL-like locators. They start with a short identifying string followed by a colon. There can be no space after the colon. The following list shows examples for each link type.

<code>http://www.astro.uva.nl/~dominik</code>	on the web
<code>doi:10.1000/182</code>	DOI for an electronic resource
<code>file:/home/dominik/images/jupiter.jpg</code>	file, absolute path
<code>/home/dominik/images/jupiter.jpg</code>	same as above
<code>file:papers/last.pdf</code>	file, relative path
<code>./papers/last.pdf</code>	same as above
<code>file:/myself@some.where:papers/last.pdf</code>	file, path on remote machine

<code>/myself@some.where:papers/last.pdf</code>	same as above
<code>file:sometextfile::NNN</code>	file, jump to line number
<code>file:projects.org</code>	another Org file
<code>file:projects.org::some words</code>	text search in Org file ¹
<code>file:projects.org::*task title heading search in Org</code>	

file²

<code>file+sys:/path/to/file</code>	open via OS, like double-click
<code>file+emacs:/path/to/file</code>	force opening by Emacs
<code>docview:papers/last.pdf::NNN</code>	open in doc-view mode at page
<code>id:B7423F4D-2E8A-471B-8810-C40F074717E9</code>	Link to heading by ID
<code>news:comp.emacs</code>	Usenet link
<code>mailto:adent@galaxy.net</code>	Mail link
<code>mhe:folder</code>	MH-E folder link
<code>mhe:folder#id</code>	MH-E message link
<code>rmail:folder</code>	RMAIL folder link
<code>rmail:folder#id</code>	RMAIL message link
<code>gnus:group</code>	Gnus group link
<code>gnus:group#id</code>	Gnus article link
<code>bbdb:R.*Stallman</code>	BBDB link (with regexp)
<code>irc:/irc.com/#emacs/bob</code>	IRC link
<code>info:org#External links</code>	Info node or index link

<code>shell:ls *.org</code>	A shell command
<code>elisp:org-agenda</code>	Interactive Emacs command
<code>elisp:(find-file-other-frame "Elisp.org")</code>	Emacs form to evaluate

On top of these built-in link types, some are available through the `contrib/` directory (see [Installation](#)). For example, these links to VM or Wanderlust messages are available when you load the corresponding libraries from the `contrib/` directory:

<code>vm:folder</code>	VM folder link
<code>vm:folder#id</code>	VM message link
<code>vm://myself@some.where.org/folder#id</code>	VM on remote machine
<code>vm-imap:account:folder</code>	VM IMAP folder link
<code>vm-imap:account:folder#id</code>	VM IMAP message link
<code>wl:folder</code>	WANDERLUST folder link
<code>wl:folder#id</code>	WANDERLUST message link

For customizing Org to add new link types [Adding hyperlink types](#).

A link should be enclosed in double brackets and may contain a descriptive text to be displayed instead of the URL (see [Link format](#)), for example:

```
[[http://www.gnu.org/software/emacs/][GNU Emacs]]
```

If the description is a file name or URL that points to an image, HTML export (see [HTML export](#)) will inline the image as a clickable button. If there is no description at all and the link points to an image, that image will be inlined into the exported HTML file.

Org also finds external links in the normal text and activates them as links. If spaces must be part of the link (for example in ‘`bbdb:Richard Stallman`’), or if you need to remove ambiguities about the end of the link, enclose them in square brackets.

9.14.3.1 Footnotes

[1] The actual behavior of the search will depend on the value of the option `org-link-search-must-match-exact-headline`. If its value is `nil`, then a fuzzy text search will be done. If it is `t`, then only the exact headline will be matched, ignoring spaces and cookies. If the value is `query-to-create`, then an exact headline will be searched; if it is not found, then the user will be queried to create it.

[2] Headline searches always match the exact headline, ignoring spaces and cookies. If the headline is not found and the value of the option `org-link-search-must-match-exact-headline` is `query-to-create`, then the user will be queried to create it.

4. [Handling links](#): Creating, inserting and following

9.14.4 Handling links

Org provides methods to create a link in the correct syntax, to insert it into an Org file, and to follow the link.

C-c l (org-store-link)

Store a link to the current location. This is a *global* command (you must create the key binding yourself) which can be used in any buffer to create a link. The link will be stored for later insertion into an Org buffer (see below). What kind of link will be created depends on the current buffer:

Org	mode	buffers
-----	------	---------

For Org files, if there is a '<<target>>' at the cursor, the link points to the target. Otherwise it points to the current headline, which will also be the description¹.

If the headline has a `CUSTOM_ID` property, a link to this custom ID will be stored. In addition or alternatively (depending on the value of `org-id-link-to-org-use-id`), a globally unique `ID` property will be created and/or used to construct a link². So using this command in Org buffers will potentially create two links: a human-readable from the custom ID, and one that is globally unique and works even if the entry is moved from file to file. Later, when inserting the link, you need to decide which one to use.

Email/News clients: VM, Rmail, Wanderlust, MH-E, Gnus
 Pretty much all Emacs mail clients are supported. The link will point to the current article, or, in some GNUS buffers, to the group. The description is constructed from the author and the subject.

Web browsers: W3 and W3M

Here the link will be the current URL, with the page title as description.

Contacts: BBDB

Links created in a BBDB buffer will point to the current entry.

Chat: IRC

For IRC links, if you set the option `org-irc-link-to-logs` to `t`, a `'file:/'` style link to the relevant point in the logs for the current conversation is created. Otherwise an `'irc:/'` style link to the user/channel/server under the point will be stored.

Other files

For any other files, the link will point to the file, with a search string (see [Search options](#)) pointing to the contents of the current line. If there is an active region, the selected words will form the basis of the search string. If the automatically created link is not working correctly or accurately enough, you can write custom functions to select the search string and to do the search for particular file types—see [Custom searches](#). The key binding `C-c 1` is only a suggestion—see [Installation](#).

Agenda view

When the cursor is in an agenda view, the created link points to the entry referenced by the current line.

5. [Using links outside Org](#): Linking from my C source code?

9.14.5 Using links outside Org

You can insert and follow links that have Org syntax not only in Org, but in any Emacs buffer. For this, you should create two global commands, like this (please select suitable global keys yourself):

```
(global-set-key "\C-c L" 'org-insert-link-global)
```

```
(global-set-key "\C-c o" 'org-open-at-point-global)
```

6. [Link abbreviations](#): Shortcuts for writing complex links

9.14.6 Link abbreviations

Long URLs can be cumbersome to type, and often many similar links are needed in a document. For this you can use link abbreviations. An abbreviated link looks like this

```
[[linkword:tag][description]]
```

where the `tag` is optional. The *linkword* must be a word, starting with a letter, followed by letters, numbers, '-', and '_'. Abbreviations are resolved according to the information in the variable `org-link-abbrev-alist` that relates the linkwords to replacement text. Here is an example:

```
(setq org-link-abbrev-alist
      '(("bugzilla" . "http://10.1.2.9/bugzilla/show_bug.cgi?id=")
        ("url-to-ja" . "http://translate.google.fr/translate?sl=en&tl=ja&u=%h")
        ("google" . "http://www.google.com/search?q=")
        ("gmap" . "http://maps.google.com/maps?q=%s")
        ("omap" . "http://nominatim.openstreetmap.org/search?q=%s&polygon=1")
        ("ads" . "http://adsabs.harvard.edu/cgi-bin/nph-abs_connect?author=%s&db_key=AST"))) )
```

If the replacement text contains the string '%s', it will be replaced with the tag. Using '%h' instead of '%s' will url-encode the tag (see the example above, where we need to encode the URL parameter.) Using '%(my-function)' will pass the tag to a custom function, and replace it by the resulting string.

If the replacement text doesn't contain any specifier, it will simply be appended to the string in order to create the link.

Instead of a string, you may also specify a function that will be called with the tag as the only argument to create the link.

With the above setting, you could link to a specific bug with `[[bugzilla:129]]`, search the web for 'OrgMode' with `[[google:OrgMode]]`, show the map location of the Free Software Foundation `[[gmap:51 Franklin Street, Boston]]` or of Carsten

office [[omap:Science Park 904, Amsterdam, The Netherlands]] and find out what the Org author is doing besides Emacs hacking with [[ads:Dominik,C]].

If you need special abbreviations just for a single Org buffer, you can define them in the file with

```
#+LINK: bugzilla http://10.1.2.9/bugzilla/show_bug.cgi?id=
#+LINK: google http://www.google.com/search?q=%s
```

In-buffer completion (see [Completion](#)) can be used after '[' to complete link abbreviations. You may also define a function `org-PREFIX-complete-link` that implements special (e.g., completion) support for inserting such a link with `C-c C-l`. Such a function should not accept any arguments, and return the full link with prefix.

7. [Search options](#): Linking to a specific location

9.14.7 Search options in file links

File links can contain additional information to make Emacs jump to a particular location in the file when following a link. This can be a line number or a search option after a double l colon. For example, when the command `C-c l` creates a link (see [Handling links](#)) to a file, it encodes the words in the current line as a search string that can be used to find this line back later when following the link with `C-c C-o`.

Here is the syntax of the different ways to attach a search to a file link, together with an explanation:

```
[[file:~/code/main.c::255]]

[[file:~/xx.org::My Target]]

[[file:~/xx.org::*My Target]]

[[file:~/xx.org::#my-custom-id]]

[[file:~/xx.org::/regexp/]]
```

255

Jump to line 255.

My Target

Search for a link target '<<My Target>>', or do a text search for 'my target', similar to the search in internal links, see [Internal links](#). In HTML export (see [HTML export](#)), such a file link will become an HTML reference to the corresponding named anchor in the linked file.

*My Target

In an Org file, restrict search to headlines.

#my-custom-id

Link to a heading with a CUSTOM_ID property

/regexp/

Do a regular expression search for `regexp`. This uses the Emacs command `occur` to list all matches in a separate window. If the target file is in Org mode, `org-occur` is used to create a sparse tree with the matches.

As a degenerate case, a file link with an empty file name can be used to search the current file. For example, `[[file::find me]]` does a search for 'find me' in the current file, just as `'[[find me]]'` would.

9.14.7.1 Footnotes

[1] For backward compatibility, line numbers can also follow a single colon.

8. [Custom searches](#): When the default search is not enough

9.14.8 Custom Searches

The default mechanism for creating search strings and for doing the actual search related to a file link may not work correctly in all cases. For example, BibTeX database files have many entries like 'year="1993"' which would not result in good search strings, because the only unique identification for a BibTeX entry is the citation key.

If you come across such a problem, you can write custom functions to set the right search string for a particular file type, and to do the search for the string in the file. Using `add-hook`, these functions need to be added to the hook variables `org-create-file-search-functions` and `org-execute-file-search-functions`. See the docstring for these variables for more information. Org actually uses this mechanism for BibTeX database files, and you can use the corresponding code as an implementation example. See the file `org-bibtex.el`.

ANEXO 9 – ATALHOS DO EMACS

Teclas de atalhos para o EMACS/XEMACS

Tabela: Alguns comandos básicos do EMACS. C=Ctrl; S=Shift; M=Alt ou Esc	
C-x C-f	Abrir arquivo
C-x C-s	Salvar arquivo
C-x C-w	Salvar como
C-x s	Salvar todos arquivos abertos
C-x i	Inserir arquivo no documento corrente
C-x C-b	Alternar para um arquivo já carregado
C-x C-c	Sair do Emacs
C-x v-v	Faz checkin/checkout de arquivo de repositório CVS/RCS
C-x C-m c utf-8	Prefixo para operações com UTF-8
C-g	Interromper comando em execução
C-f	Ir ao próximo caractere (igual a seta para direita)
C-b	Ir ao caractere anterior (igual a seta para esquerda)
M-v	Voltar uma página (igual a PgUp)
C-v	Avançar uma página (igual a PgDown)
C-a	Ir ao início da linha (igual a Home)
C-e	Ir ao fim da linha (igual a End)
M- <	Ir início do documento (igual a C-Home)
M- >	Ir ao fim do documento (igual a C-End)
C-s	Busca incremental
C-r	Busca incremental reversa

C-M-n	Avança ao próximo parêntese que casa com aquele sob o cursor
C-M-p	Avança ao último parêntese que casa com aquele sob o cursor
M-%	Encontrar e trocar
C-M-%	Encontrar e trocar (versão expressões regulares)
C-espaco	Marcar início de bloco
C-w	Cortar bloco
M-w	Copiar bloco
C-y	Colar bloco
S-Del	Apagar o bloco selecionado
C-_	Undo (desfazer)
C-k	Apagar do cursor até o fim da linha
M-c	Capitalizar palavra (i.e., deixar inicial maiúscula)
M-l	Deixar palavra em minúsculas
M-u	Deixar palavra em maiúsculas
C-k	Apagar do cursor até o fim da linha
C-t	Troca caractere sob cursor com caractere anterior

COMANDOS NO EMACS

Para executar um comando no emacs pode:

- selecionar o comando (com o botão esquerdo do rato) nos menus;
- digitar a sequência de teclas correspondente;
- digitar o comando a partir do minibuffer (depois de Esc-X).

9.14.9 Sair do emacs

Comando	Descrição
Menu Files/"Exit emacs"	para sair do emacs

Esc-x kill-emacs	
^x ^c	

Exercício: Entrar e sair do emacs usando cada uma das formas de execução de comandos.

MANIPULAÇÃO DE FICHEIROS

Comando	Descrição
^x ^f	para abrir um ficheiro cujo nome deve ser introduzido no minibuffer
^x ^s	para guardar as alterações feitas (ao ficheiro que está no buffer em utilização)
^x ^w	para guardar as alterações feitas para um (outro) ficheiro cujo nome deve ser introduzido

Exercício:

Escrever, compilar e executar o seguinte programa em C no ficheiro `prog.c`

```
#include <stdio.h>
main()
printf("ola mundo\n");
}
```

MANIPULAÇÃO DE BUFFERS E DE JANELAS

Comando	Descrição
^x b	para trocar de buffer
^x ^b	para listar todos os buffers
^x k	para matar/fechar um buffer

Comando	Descrição	Em alternativa pode...
---------	-----------	------------------------

^x 2	para dividir a janela na horizontal	
^x 3	para dividir a janela na vertical	
^x 1	para ficar só com uma janela	clicar com o rato na janela que pretende
^x o	para mudar de janela	