

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo França Severino

**Implementação de Sistema Web para
Distribuição didática da FAMAT - UFU
utilizando processo de design, metodologia
SCRUM e aplicação do Framework PACT**

Uberlândia, Brasil

2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Gustavo França Severino

**Implementação de Sistema Web para Distribuição
didática da FAMAT - UFU utilizando processo de design,
metodologia SCRUM e aplicação do Framework PACT**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Ciência da Computação.

Orientador: Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2019

Gustavo França Severino

Implementação de Sistema Web para Distribuição didática da FAMAT - UFU utilizando processo de design, metodologia SCRUM e aplicação do Framework PACT

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 08 de julho de 2019:

Maria Adriana Vidigal de Lima
Orientador

Professor

Professor

Uberlândia, Brasil
2019

“Algo só é impossível até que alguém duvide e resolva provar o contrário.”

Albert Einstein

Resumo

Este trabalho abrange o processo, conceitos, ferramentas e métodos aplicados tanto nos quesitos de desenvolvimento quanto da área de Interação Humano-Computador (IHC), utilizados no desenvolvimento do front-end de um Sistema de Gerenciamento e Distribuição de Disciplinas para a Faculdade de Matemática da Universidade Federal de Uberlândia.

Através da utilização da metodologia ágil Scrum e o framework Angular, foi possível realizar sprints que se adequassem a necessidade do cotidiano dos desenvolvedores envolvidos, de forma que fossem feitas entregas incrementais. Para garantir a usabilidade e aceitação da interface gráfica pelos usuários, foi utilizado um processo de prototipação através da técnica *Wizard of Oz* e foi aplicado testes de usabilidade com futuros usuários fazendo uso da técnica *Think Aloud*.

Palavras-chave: Scrum, Angular, IHC, Teste de Usabilidade, Prototipação.

Listas de ilustrações

Figura 1 – Declaração de Módulo	20
Figura 2 – Declaração de Componentes	21
Figura 3 – Utilização de Componentes	21
Figura 4 – Declaração de Serviços	22
Figura 5 – Utilização de Serviços	22
Figura 6 – Exemplo de <i>event binding</i>	23
Figura 7 – Primeiro exemplo de <i>property binding</i>	23
Figura 8 – Segundo exemplo de <i>property binding</i>	24
Figura 9 – Exemplo de <i>Two-way data binding</i>	24
Figura 10 – Exemplo de utilização de diretivas	24
Figura 11 – Exemplo de roteamento	25
Figura 12 – Processo de design [Fonte: (DINIZ; SANTANA, 2010)]	30
Figura 13 – Diagrama Entidade-Relacionamento	42
Figura 14 – Diagrama dos Objetos com seus atributos	43
Figura 15 – Esboço da tela de login.	44
Figura 16 – Esboço da tela de cadastro.	44
Figura 17 – Esboço da tela de criação de semestre padrão.	45
Figura 18 – Esboço da tela de criação de semestre com clonagem.	46
Figura 19 – Esboço da tela de cadastro de disciplinas, tipos de atividades, turmas e atividades.	47
Figura 20 – Esboço da tela de busca de disciplinas, tipos de atividade, atividades, professores e turma.	48
Figura 21 – Esboço da tela de edição de professores, disciplinas, tipos de atividade, atividades e turma.	49
Figura 22 – Esboço da tela de seleção de disciplinas do usuário.	50
Figura 23 – Esboço da tela padrão de período de seleção de disciplinas do administrador.	50
Figura 24 – Esboço da tela completa de período de seleção de disciplinas do administrador.	51
Figura 25 – Esboço da tela de aprovação de cadastro de professores.	52
Figura 26 – Esboço da área de trabalho.	52
Figura 27 – Esboço da área de trabalho com a barra lateral expandida.	53
Figura 28 – Protótipo da tela de seleção de disciplinas utilizando a técnica <i>Wizard of Oz</i>	54
Figura 29 – Arquivo de orientação para avaliação de protótipo utilizando <i>Think Aloud</i>	55

Figura 30 – Diretório do projeto	56
Figura 31 – Template da tela de <i>login</i>	58
Figura 32 – Template da tela de cadastro	58
Figura 33 – Template da area de trabalho com barra lateral fechada	59
Figura 34 – Template da area de trabalho com barra lateral expandida	59
Figura 35 – Template da tela de cadastro de disciplinas	60
Figura 36 – Template da tela de busca de disciplinas	60
Figura 37 – Template da tela de edição de disciplinas	61
Figura 38 – Template da tela de aprovação de cadastro de professores	61
Figura 39 – Template da tela de alteração de senha e consulta do ranking	62
Figura 40 – Método utilizado pela tela de <i>login</i>	62
Figura 41 – Método utilizado para cadastrar disciplinas	63
Figura 42 – Template da tela de criação de novo semestre simples	65
Figura 43 – Template da tela de criação de novo semestre completa	65
Figura 44 – Template da tela de registro de disciplinas do administrador	66
Figura 45 – Template da tela de seleção de disciplinas do administrador simples	66
Figura 46 – Template da tela de seleção de disciplinas do administrador completa	67
Figura 47 – Template da tela de seleção de disciplinas do professor	67
Figura 48 – Método utilizado pela tela Iniciar novo semestre	68
Figura 49 – Método utilizado para gerar as características das turmas	70
Figura 50 – Método utilizado para atualizar a lista de turmas selecionadas	70
Figura 51 – Método utilizado para inserir cartões de turmas selecionadas no <i>gridster</i>	70
Figura 52 – Método utilizado para atualizar no banco as turmas selecionadas	71

Lista de tabelas

Tabela 1 – Primeira Persona	33
Tabela 2 – Segunda Persona	34
Tabela 3 – Terceira Persona	34
Tabela 4 – Análise do elemento Pessoas	36
Tabela 5 – Análise do elemento Atividades	37
Tabela 6 – Análise do elemento Contexto	38
Tabela 7 – Análise do elemento Tecnologias	39

Lista de abreviaturas e siglas

IHC	<i>Interação Humano-Computador</i>
UI	<i>User Interface</i>
HTML	<i>Hyper Text Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JS	<i>JavaScript</i>
UML	<i>Unified Modeling Language</i>
MVC	<i>Model View Controller</i>
FAMAT	<i>Faculdade de Matemática</i>
IO	<i>Input Output</i>
JSON	<i>JavaScript Object Notation</i>
TI	<i>Tecnologia da Informação</i>
FDD	<i>Feature Driven Development</i>
XP	<i>Extreme Programming</i>
MSF	<i>Microsoft Solutions Framework</i>
POO	<i>Programação Orientada a Objetos</i>
MPHI	<i>Modelo do Processador de Informação Humano</i>
TAM	<i>Technology Acceptance Model</i>
MER	<i>Modelos de entidade-relacionamento</i>
RF	<i>Requisitos Funcionais</i>
RFI	<i>Requisitos Funcionais de Integração</i>
RNF	<i>Requisitos Não Funcionais</i>

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Método	12
1.3	Organização do Trabalho	12
2	REFERENCIAL TEÓRICO	14
2.1	Metodologias de Desenvolvimento	14
2.1.1	Metodologias Ágeis	15
2.1.2	Scrum	16
2.2	Arquitetura	17
2.2.1	<i>Model View Controller (MVC)</i>	17
2.2.2	Frameworks Javascript	18
2.2.3	Angular	19
2.2.3.1	Injeção de Dependência	25
2.3	IHC	25
2.3.1	<i>Framework PACT</i>	26
2.3.1.1	Pessoa	27
2.3.1.1.1	Modelos Mentais	27
2.3.1.2	Atividade	28
2.3.1.3	Contexto	29
2.3.1.4	Tecnologia	29
2.3.2	Processo de design	30
2.3.2.1	<i>Wizard of Oz</i>	31
2.3.3	<i>Think Aloud</i>	31
3	DESENVOLVIMENTO	32
3.1	Planejamento Scrum	32
3.1.1	Entendimento	32
3.1.1.1	O processo de distribuição de disciplinas na FAMAT	32
3.1.1.2	Personas	33
3.1.1.3	Cenários	33
3.1.1.4	PACT	35
3.1.1.5	Análise de Requisitos	35
3.1.1.6	Prioridade dos Requisitos	35
3.1.1.7	Requisitos Funcionais	36
3.1.1.8	Requisitos Não Funcionais	41

3.1.2	Design	41
3.1.3	Antecipação	46
3.1.4	Avaliação	47
3.1.4.1	<i>Think aloud</i>	47
3.1.5	Product Backlog	50
3.2	Desenvolvimento Scrum	53
3.2.1	<i>Sprint 1</i>	53
3.2.1.1	<i>Sprint Planning Meeting</i>	53
3.2.1.2	Execução	54
3.2.1.3	<i>Sprint Review</i>	58
3.2.2	<i>Sprint 2</i>	59
3.2.2.1	<i>Sprint Planning Meeting</i>	59
3.2.2.2	Execução	61
3.2.2.3	<i>Sprint Review</i>	65
3.2.3	<i>Sprint 3</i>	66
3.2.3.1	<i>Sprint Planning Meeting</i>	66
3.2.3.2	Execução	67
3.2.3.3	<i>Sprint Review</i>	70
4	CONCLUSÃO	72
	Conclusão e Trabalhos Futuros	72
	REFERÊNCIAS	73
	ANEXOS	75
	ANEXO A – RESOLUÇÃO 02/2017 - FAMAT	76
	ANEXO B – RELATÓRIO DE ATIVIDADES - MODELO	92

1 Introdução

Desde sua criação, a computação tinha como objetivo a realização de alguma tarefa que era demasiada complexa ou que exigia um esforço muito grande e, por consequência, um espaço de tempo maior para que um ser humano a executasse. Essa tendência se mostrou ainda mais forte após a popularização do computador, no qual os sistemas computacionais se disseminaram e estão cada vez mais presentes no cotidiano das pessoas.

Atualmente com essa prática bastante difundida, uma falha nesse processo passou a ser identificada com frequência. A automatização dos processos era feita de forma que a abstração do problema pudesse ser compreendida pela máquina e, assim, realizada do modo esperado pela mesma. No entanto, nem todo processo consegue ser completamente automatizado, sendo ainda necessária a presença do homem para que esta seja concluída. Assim, notou-se a necessidade de modelar os sistemas computacionais de modo que as interações humano-computador, relação que dá nome à área de pesquisa relacionada (Intereração Humano-Computador - IHC), se tornassem mais simples e precisas.

Dentro das universidades não é diferente, há uma série de processos que muitas vezes podem ser automatizados, como o caso da distribuição didática entre os professores para o semestre seguinte, que será o caso estudado neste trabalho.

Na Faculdade de Matemática (FAMAT) da Universidade Federal de Uberlândia, todo final de semestre inicia-se um processo de seleção de disciplina, em que os professores devem elencar e ordenar por prioridade, quais disciplinas desejam lecionar no próximo semestre.

Tal processo é muito custoso, exige que os professores despendam de um tempo considerável para analisar as opções disponíveis e verificar junto a sua prioridade qual a melhor combinação de horários, exige também que os administradores realizem toda uma documentação das opções que serão disponibilizadas para seleção.

Esta divisão por pontuação dos professores é feita com base nas atividades realizadas durante o ano anterior e registrada por cada um antes do período de seleção, o que leva a um aumento no tempo gasto para a realização da tarefa principal.

Todo este processo é gerenciado por uma comissão de trabalho composta por professores da FAMAT e tem duração de trinta a quarenta e cinco dias. Assim, a automatização deste processo, traria para a FAMAT uma grande economia de tempo, tanto por parte da administração quanto por parte dos docentes, permitindo que o período economizado pela utilização do sistema seja despendido com outras atividades mais relevantes e produtivas.

1.1 Objetivos

Este projeto tem como objetivo geral aplicar os conhecimentos adquiridos no curso de Ciência da Computação para desenvolver um sistema utilizando os *frameworks* PACT e Angular e que possibilite diminuir o esforço e o tempo gasto no processo de seleção e distribuição de disciplinas da FAMAT. Pode-se detalhar este objetivo principal da seguinte forma:

- I. Construir um *front-end* para realizar:
 - a) o registro e gerenciamento das atividades docentes de forma digital;
 - b) o processo de seleção de disciplinas de forma digital;
- II. Diminuir consideravelmente a intervenção humana realizada na distribuição das disciplinas;
- III. Promover interfaces intuitivas e fáceis de serem compreendidas e utilizadas;
- IV. Seguir princípios de IHC para garantir a validade dos itens anteriores e utilizar o processo descrito e especificado na seção Método (1.2) para o desenvolvimento do projeto.

1.2 Método

A concepção do projeto seguiu uma metodologia chamada Processo de design e o desenvolvimento do software seguiu as premissas da metodologia ágil Scrum. Para facilitar e agilizar o andamento do projeto, foi escolhido um framework para desenvolvimento front-end chamado Angular. Para elaborar o design da interface foram levados em consideração princípios de IHC e para validá-lá foram realizados testes de usabilidade, utilizando protótipo produzido com base na técnica *Wizard of Oz* e método de avaliação *Think Aloud*.

1.3 Organização do Trabalho

Este trabalho está estruturado da seguinte forma:

No primeiro capítulo foi descrita a contextualização do problema abordado junto de sua justificativa, as metas a serem alcançadas e uma breve apresentação das ferramentas e conceitos que foram utilizados.

No segundo capítulo são apresentados de forma mais completa os conceitos e ferramentas que compõem a base da implementação do sistema e a justificativa de utilização de cada um deles, formando o referencial teórico construído como base para este trabalho.

No terceiro capítulo é mostrado de forma detalhada todo o processo, incluindo os relatos dos ritos provenientes do Scrum no decorrer do desenvolvimento do projeto, as decisões de arquitetura e alterações realizadas.

No quarto capítulo, são apresentados alguns trabalhos que tiveram como objetivo suprir alguma necessidade parecida com as que este trabalho supre, descrevendo um pouco da abordagem utilizada e suas especificidades.

Finalmente, o capítulo quinto sintetiza os resultados obtidos e o anexo 1 contém os documentos utilizados durante o trabalho.

2 Referencial Teórico

Este capítulo apresenta os conceitos e ferramentas utilizados no desenvolvimento do projeto proposto, as justificativas de escolha e utilização de cada um e, quando conveniente, alternativas que poderiam ser utilizadas e os motivos pelos quais não foram escolhidas.

2.1 Metodologias de Desenvolvimento

Assim como a necessidade do mercado mudou com o tempo, a realidade das equipes que trabalham em um projeto de software também. Programadores que antes trabalhavam sozinhos, agora participam de uma equipe de desenvolvimento em que cada um domina uma tecnologia necessária para a construção do projeto. Além disso, equipes de teste e análise de negócio trabalham muito mais próximos aos desenvolvedores.

Para gerenciar e coordenar tudo isso, entram em cena as metodologias de desenvolvimento de software. Como o próprio nome sugere, se referem aos princípios e práticas a serem adotadas durante o processo de desenvolvimento e andamento do projeto e tem, como descrito em [Leite \(2006\)](#), o objetivo de “*definir de forma clara ‘quem’ faz ‘o que’, ‘quando’, ‘como’ e até mesmo ‘onde’, para todos os que estejam envolvidos diretamente ou não com o desenvolvimento de software*”.

As metodologias para a gestão dos processos de desenvolvimento de software mais conhecidas são as metodologias ágeis e cascata ([PRESSMAN; MAXIM, 2016](#)). A cascata especifica um roteiro que deve ser executado de forma sequencial e rígida, de maneira que as ideias sejam analisadas e validadas sob todos os aspectos. Esse modelo foi criado na década de 70 e ganhou bastante visibilidade a partir dos anos 80. O roteiro define como atividade inicial o levantamento dos requisitos do cliente, e em seguida são estabelecidas as estimativas, o cronograma e a forma de acompanhamento. O projeto é então analisado, construído, testado e finalmente implantado. Uma das vantagens da cascata é que ela não permite pular etapas: a etapa seguinte só pode ser iniciada quando a anterior estiver correta e completamente concluída. Por outro lado, esta sincronização instaura uma situação em que, se um atraso ocorre, todo o processo é afetado. Além disso, o processo faz o software aparecer somente ao final, o que pode ser muito tarde.

Como alternativa à cascata, surgiram diversas metodologias, sendo que dessas, as ágeis tem ganhado maior popularidade devido à tendência atual do mercado de desenvolvimento em realizar entregas cada vez menores, porém mais frequentes e com mais qualidade aos clientes. Estas características reúnem os fundamentos das metodologias

ágeis.

2.1.1 Metodologias Ágeis

Metodologias ágeis são, no contexto da área de Tecnologia da informação (TI), metodologias que buscam ganhar tempo no processo de desenvolvimento de software através da fragmentação de atividades complexas. Assim, espera-se um ganho considerável em termos de qualidade da entrega, que se torna incremental e mais frequente.

Todos os dias surgem novas metodologias com propostas ágeis, as mais conhecidas são: Scrum, Lean, Feature Driven Development (FDD), Extreme Programming (XP) e Microsoft Solutions Framework (MSF). No entanto, apesar de cada uma ter sua particularidade, todas elas tem como base um documento chamado manifesto ágil.

Este documento ([BECK et al., 2001](#)), composto por 4 sentenças e 12 princípios, serve como um norte para a criação de uma metodologia ágil, contendo os princípios e valores que este tipo de metodologia deve seguir ou priorizar.

As 4 sentenças são:

“Indivíduos e interações mais que processos e ferramentas.”

“Software em funcionamento mais que documentação abrangente.”

“Colaboração com o cliente mais que negociação de contratos.”

“Responder a mudanças mais que seguir um plano.”

E seus 12 princípios são:

1. *“Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiada de software com valor agregado.”*
2. *“Mudanças nos requisitos são bem-vindas, mesmo tardivamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.”*
3. *“Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.”*
4. *“Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.”*
5. *“Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.”*
6. *“O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.”*

7. “Software funcionando é a medida primária de progresso.”
8. “Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.”
9. “Contínua atenção à excelência técnica e bom design aumenta a agilidade.”
10. “Simplicidade, a arte de maximizar a quantidade de trabalho não realizado, é essencial.”
11. “As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.”
12. “Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”

2.1.2 Scrum

A metodologia Scrum é bastante conhecida por possuir uma série de cerimônias que ajudam a organizar o processo de desenvolvimento, inclusive, por ser bastante popular, um novo profissional entrou em cena nos últimos anos. Este profissional, chamado de *scrum master*, deve possuir um conhecimento aprofundado sobre esta metodologia, pois fica encarregado, além de outras obrigações, de coordenar o andamento destas cerimônias.

Basicamente a metodologia se divide em três processos macros: planejamento, desenvolvimento e a finalização.

Durante a etapa de planejamento é definido o *product backlog*, que consiste em um conjunto de todas as funcionalidades que são desejadas para o projeto. Tais funcionalidades são definidas pelo *product owner*. É válido lembrar que, inicialmente, nem todas as funcionalidades precisam estar presentes no *product backlog* visto que uma das características da metodologia é a adaptabilidade.

Em seguida, se dá início a etapa de desenvolvimento com o primeiro rito do Scrum, chamado *sprint planning meeting*. Neste rito, o *product owner* define quais funcionalidades tem maior prioridade, ao passo que a equipe de desenvolvimento tenta, através de perguntas, explorar o máximo possível o contexto de cada uma delas, para então quebrá-las em tarefas técnicas. O conjunto dessas tarefas resultantes forma o *sprint backlog*. Essa lista, contém as atividades que após um período definido de tempo, chamado de *sprint*, esperase estarem concluídas, formando assim a primeira entrega incremental da metodologia ágil.

Durante a *sprint* são realizadas diariamente reuniões rápidas, chamadas *daily meetings*, para que toda a equipe possa ser contextualizada sobre os avanços de cada membro

e que possam ser identificados gargalos no processo. Cabe ao *scrum master* correr atrás da resolução destes pontos de gargalo para que o projeto continue fluindo corretamente.

Após a finalização do prazo de uma *sprint* é realizada a terceira cerimônia característica do Scrum, a *sprint review meeting*. Neste rito, é apresentado a evolução do projeto alcançada pela *sprint* anterior, além disso, a equipe tenta fazer uma crítica sobre o processo de desenvolvimento até o momento, elencando os pontos positivos e negativos, para que a próxima *sprint* possa tirar proveito dos erros cometidos na anterior e assim tornar o processo ainda mais ágil e fluido.

Este processo de *sprints* se repete até que o *product backlog* tenha sido esgotado ou, no pior dos casos, o projeto seja cancelado ([ÁGIL, 2013](#)).

2.2 Arquitetura

Basicamente, a arquitetura de um software consiste em um conjunto de especificações, como dito em [Filho \(2008\)](#), que: “*determinam notações para conectar componentes (módulos) e descrever mecanismos de interação, além de técnicas para gerenciar configurações e controlar versões*”.

2.2.1 *Model View Controller* (MVC)

O padrão MVC separa a lógica, os dados e a visualização em três camadas distintas, cada uma exercendo uma tarefa específica para a composição do projeto.

A *View* fica responsável por tornar visual os dados da *Model* e controlar os mecanismos de interação com o usuário. O *Controller* determina como o sistema deve se comportar reagindo às interações do usuário com o sistema. A *Model* se encarrega de manipular os dados que representam o estado do sistema, respondendo às requisições do *Controller*, desde solicitação de informações até alterações no estado da *Model* ([HAVIV, 2016](#)).

Em um projeto web a estrutura em camadas pode ser agrupada em duas categorias macro:

Front-end: camada que comprehende a atuação deste projeto, é composta pela camada *View* do padrão MVC, a qual possui os recursos destinados a interação com o usuário, atuando apenas no lado do cliente. Atualmente o aumento da complexidade do *Front-end* permite aplicar uma arquitetura interna específica para o desenvolvimento desta camada, que será abordado logo abaixo.

Back-end: composto pelas duas camadas restantes do MVC, *Model* e *Controller*. Esta camada atua no lado do servidor e cuida basicamente do armazenamento e proces-

samento de dados para atender às requisições do *Front-end*.

A tecnologia utilizada para desenvolver o *Front-end* deste projeto, abordada na próxima seção, dita uma característica única para a arquitetura. Ela mescla a já conhecida e difundida arquitetura MVC com a separação por *Feature/Components*, ou seja, essa arquitetura gera agrupamentos separados por *features* e dentro desses agrupamentos é utilizado o MVC.

Em outras palavras, em um único lugar encontramos os dados (*Model*), a página HTML (*Hypertext Markup Language*) desse componente (*View*) e a lógica que determina o comportamento que ele deve apresentar (*Controller*).

2.2.2 Frameworks Javascript

Antes de entrar mais a fundo na arquitetura do projeto é necessário compreender melhor o que é, e o porquê da utilização de um *Framework Javascript*.

Um *Framework Javascript* é um *framework* de aplicação escrito em Javascript, utilizado para facilitar o desenvolvimento de aplicações web do lado do cliente. Eles se diferem de bibliotecas Javascript pelo nível de dependência que um projeto sofre. Enquanto as bibliotecas oferecem funções que podem ser utilizadas pela aplicação, os *frameworks* definem toda a estrutura, arquitetura e design da aplicação.

Existem uma série de *frameworks* baseados em javascript, os mais conhecidos atualmente são *Angular*, *React* e *Vue.js* ([BENITTE; GREIF; RAMBEAU, 2017](#)).

Angular é uma plataforma e *framework open source* desenvolvido e mantido pelo Google com o objetivo de facilitar o desenvolvimento de *Single-Page Applications*, Web e mobile, usando HTML, CSS (*Cascading Style Sheets*) e TypeScript. Para utilizá-lo é necessário um ambiente com Node.js (Plataforma assíncrona e orientada a eventos, construída sobre o JavaScript que possibilita a criação de aplicações web eficientes e escaláveis) e um gerenciador de pacotes npm. Este *framework* sofreu várias atualizações até o momento, inclusive foi totalmente reescrito ao ser lançado a sua versão Angular 2. Neste projeto foi utilizado o Angular 6. Dentre suas principais características podemos destacar sua arquitetura que será explicada logo mais ([AFONSO, 2018](#)).

React nada mais é que, como seus próprios criadores disseram: “*Uma biblioteca JavaScript declarativa, eficiente e flexível para a criação de interfaces de usuário*”. Surgiu em 2011 e, assim, como uma grande empresa estava por trás do Angular o React também é fruto de uma grande multinacional, o Facebook.

Como dito anteriormente, diferentemente dos *frameworks* supracitados, o React é uma biblioteca e não um *framework*, ou seja, ele fornece uma coleção de funções que podem ser utilizadas para resolver problemas específicos. Sua principal característica consiste em

ser uma linguagem declarativa, se preocupa com o que o programador quer fazer, em outras palavras, descreve o que uma função faz e não como ela funciona (BARROS, 2016), diferentemente de uma linguagem imperativa que se preocupa em especificar os passos para se atingir um objetivo (BRASIL, 2018).

Vue.js é um *framework* progressivo que tem como objetivo a criação de *User Interfaces* (UI). Foi concebido com o intuito de ser adotável incrementalmente, ou seja, deve ser fácil integrar este *framework* com outras bibliotecas ou projetos existentes, ao passo que da mesma forma, consegue empoderar sofisticadas *Single-Page Applications* (YOU, 2013).

2.2.3 Angular

A primeira grande diferença que pode ser observada ao se comparar Angular com os demais *frameworks* é a utilização do TypeScript. TypeScript é uma linguagem de programação *open-source* desenvolvida e mantida pela Microsoft, que é, antes de mais nada, um superconjunto do JavaScript. A utilização desta derivação se dá devido à disposição de recursos que melhor interagem com o uso da Programação Orientada a Objetos (POO), já que o TypeScript oferece uma forma de corrigir a deficiência que o JavaScript possui ao lidar com POO, como por exemplo sua sintaxe não permitir a declaração de classes e a fraca tipagem de dados (DIONÍSIO, 2016).

Angular é estruturado em módulos. Assim, um conjunto de blocos de código que compartilham um objetivo em comum definem um módulo, e da mesma forma, um conjunto de módulos compõe a aplicação. Tal composição traz grande benefício em termos de agilidade e qualidade na produção de um software, pois fragmenta a aplicação em partes menores e menos complexas, o que as tornam mais fáceis de serem testadas e de receberem manutenção.

Cada módulo possui um arquivo cujo nome remete ao nome do módulo. Neste arquivo, são definidos os componentes que fazem parte deste módulo, são declarados quais destes componentes serão exportados, ou seja, quais componentes outros módulos poderão fazer uso, e também quais componentes externos este módulo fará uso.

Como mostra a Figura 1, foi criado um projeto de nome “exemplo” que possui um módulo chamado “app” que é formado por um único componente chamado “AppComponent”.

A Figura 2 mostra como é feita a declaração de componentes e possui estrutura criada com o auxílio do Angular Cli (ferramenta desenvolvida para automatizar a criação de elementos utilizados em projetos angular). Foi declarado um componente chamado “AppComponent” que como explicado anteriormente possui uma arquitetura interna em MVC. Ao ser gerado, foram criados 4 arquivos:

```

EXPLORER
OPEN EDITORS 1 UNSAVED
• TS app.module.ts exemplo\src\app
EXEMPLO
• exemplo
  • e2e
  • node_modules
  • src
    • app
      • TS app-routing.module.ts
      • app.component.html
      • app.component.scss
      • TS app.component.spec.ts
      • TS app.component.ts
      • TS app.module.ts
    • assets
    • environments
  • browserslist

TS app.module.ts ●
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6
7 @NgModule({
8   declarations: [
9     AppComponent
10   ],
11   imports: [
12     BrowserModule,
13     AppRoutingModule
14   ],
15   providers: [],
16   bootstrap: [AppComponent]
17 })
18 export class AppModule { }

```

Figura 1 – Declaração de Módulo

app.component.html : é referenciado no atributo “templateUrl”, do arquivo “app.component.ts”, dizendo que o template da página desse componente se encontra nesse arquivo.

app.component.scss : é referenciado no atributo “styleUrls”, do arquivo “app.component.ts”, dizendo que os estilos utilizados pelo template desse componente se encontram nesse arquivo.

app.component.spec.ts : é um arquivo criado para auxiliar na execução de testes.

app.component.ts : é o arquivo que possui a declaração do componente propriamente dita. Note que para que o componente possa ser utilizado em algum outro local do código é necessário realizar a importação do componente e, chamar, o seletor deste componente no template, que é referenciado na propriedade “selector” deste mesmo arquivo, neste caso “app-root”. Na Figura 3, é possível ver a utilização de um componente, neste caso o componente “AppComponent” faz uso do componente “AppExemplo”.

A utilização da arquitetura que o Angular dita, tem basicamente como vantagem o reaproveitamento de código. Uma vez criado um componente, para que ele possa ser utilizado em outros locais que não o seu módulo de origem, basta importá-lo no módulo em questão e chamar seu seletor no template do componente que fará uso.

Além de módulos e componentes, existe também o conceito de serviços, que nada mais são do que classes que possuem uma função que pode ser utilizada por outro componente. A Figura 4 mostra como é feita a declaração de um serviço, neste caso o serviço se chama “LogService” e a Figura 5 mostra a utilização desse serviço por um componente, no

```

EXPLORER
OPEN EDITORS
  TS app.component.ts exemplo\src\app
EXEMPLO
  exemplo
    e2e
    node_modules
  src
    app
      TS app-routing.module.ts
      app.component.html
      app.component.scss
      TS app.component.spec.ts
      TS app.component.ts
      TS app.module.ts

TS app.component.ts ×
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'exemplo';
10 }
11

```

Figura 2 – Declaração de Componentes

```

EXPLORER
OPEN EDITORS 1 UNSAVED
  ● app.component.html exemplo\src\app
EXEMPLO
  exemplo
    e2e
    node_modules
  src
    app
      exemplo
        exemplo.component.html
        exemplo.component.scss
        TS exemplo.component.spec.ts
        TS exemplo.component.ts
      TS app-routing.module.ts
      app.component.html
      app.component.scss
      TS app.component.spec.ts
      TS app.component.ts
      TS app.module.ts

app.component.html •
1 <div>
2 | <app-exemplo></app-exemplo>
3 </div>
4
5

```

Figura 3 – Utilização de Componentes

caso o componente “ExemploComponent” faz uso da função “log” do serviço “LogService” ao chamar a sua função “showResult”.

Outro conceito importante é o de *data-binding*. Esta é uma combinação de HTML com a notação angular, uma forma que o angular utiliza para sincronizar os dados presentes nos templates com os dados do *controller*, ou seja, os dados da página HTML com os dados do arquivo .ts.

Existem duas formas de *data-binding*:

Event binding : faz com que sua aplicação responda à entrada do usuário atualizando

```

import { Injectable } from '@angular/core';
@Injectable({
  providedIn: 'root'
})
export class LogService {
  constructor() {}
}

```

Figura 4 – Declaração de Serviços

```

import { Component, OnInit } from '@angular/core';
import { LogService } from '../log.service';

@Component({
  selector: 'app-exemplo',
  templateUrl: './exemplo.component.html',
  styleUrls: ['./exemplo.component.scss']
})
export class ExemploComponent {
  public result: number;
  constructor(
    private logService: LogService
  ) {}
  soma(var1: number, var2: number) {
    this.result = var1 + var2;
  }
  showResult() {
    this.logService.log(this.result);
  }
}

```

Figura 5 – Utilização de Serviços

os dados da sua aplicação. A Figura 6 mostra que o evento de click está preparado para disparar a função “soma” deste componente.

Property binding : faz com que você crie uma conexão entre valores computados na sua aplicação com os dados no HTML. Existem dois modos de se utilizar *property binding*. Na Figura 7 acontece a primeira forma que é chamada de “interpolação”,

```
<div>
  <button (click)="soma(var1,var2)"></button>
</div>
```

Figura 6 – Exemplo de *event binding*

que basicamente mostra o valor da variável contida entre as chaves. Na Figura 8 ocorre a segunda forma do *property binding*.

Two-way Data Binding : fruto da combinação destas duas formas de *data-binding*.

Basicamente é utilizado para que alterações possam ser feitas em ambas as pontas e estas gerem atualizações também em ambas as pontas. A Figura 9 mostra um exemplo dessa forma de conexão.

```
<div>
  <span>{{result}}</span>
</div>
```

Figura 7 – Primeiro exemplo de *property binding*

Outro tópico importante ao se tratar de projetos angular, são as diretivas, instruções que ficam armazenadas em metadados para os templates. Existem diretivas estruturais, *ngFor, que permite iterar sobre um objeto, e o *ngIf, que permite controlar a visualização de algum pedaço de template, diretivas de comportamento, como o ngModel, ngSwitch, ngClass, ngStyle e também possui a notação @Directive que permite criar outras diretivas. Na Figura 10 tem-se um exemplo de utilização de algumas diretivas.

```

1 <div>
2   |   <span [exibir] = "exibir">Teste</span>
3 </div>
4

```

Figura 8 – Segundo exemplo de *property binding*

```

1 <div>
2   |   <input [(ngModel)] = "var1">
3   |   <input [(ngModel)] = "var2">
4 </div>
5

```

Figura 9 – Exemplo de *Two-way data binding*

```

1 <div>
2   |   <input [(ngModel)] = "var1">
3   |   <input [(ngModel)] = "var2">
4   |   <span *ngIf = "mostrarResultado">{{result}}</span>
5   |   <span *ngFor = "let numero of vetorDeNumeros">numero</span>
6 </div>
7

```

Figura 10 – Exemplo de utilização de diretivas

O roteamento de projetos angular é feito através de um módulo chamado “Router” já incluso no angular. Este, permite definir os caminhos de navegação possíveis na aplicação através de uma estrutura hierárquica de caminhos. A Figura 11 mostra a estrutura hierárquica de roteamentos realizada neste projeto.

```

EXPLORER                                     TS app.module.ts ×
OPEN EDITORS
  TS app.module.ts src\app
FRONTEND
  TS class-model.ts
  TS person.view-mo...
  TS subject.view-mo...
  TS user.view-mode...
  services
    TS custom-modal.ser...
    TS custom-modal.ser...
    TS custom-modal.ser...
    TS validate.service.sp...
    TS validate.service.ts
    TS app.component.html
    TS app.component.scss
    TS app.component.spe...
    TS app.component.ts
    TS app.module.ts
  assets
  environments
  favicon.ico
  index.html
  TS main.ts
const appRoutes: Routes = [
  { path: '', component: HomeComponent, children: [
    { path: 'login', component: LoginComponent },
    { path: 'register', component: RegisterComponent }
  ] },
  { path: 'dashboard', component: DashboardComponent, canActivate:[AuthGuard] },
  { path: 'profile', component: ProfileComponent, canActivate:[AuthGuard] },
  { path: 'navbar', component: NavbarComponent },
  { path: 'admin', component: AdminComponent, children: [
    { path: 'subject', component: SubjectComponent },
    { path: 'new-circle', component: NewCircleComponent },
    { path: 'activity', component: ActivityComponent },
    { path: 'activity-record', component: ActivityRecordComponent },
    { path: 'class', component: ClassComponent },
    { path: 'reports', component: ReportsComponent },
    { path: 'subject-selection', component: SubjectSelectionComponent },
    { path: 'teacher', component: TeacherComponent }
  ] },
  { path: 'user', component: UserComponent, children: [
    { path: 'ranking', component: UserRankingComponent },
    { path: 'activity-record', component: UserActivityRecordComponent },
    { path: 'perfil', component: UserPerfilComponent },
    { path: 'reports', component: UserReportsComponent },
    { path: 'subject-selection', component: UserSubjectSelectionComponent }
  ] },
]

```

Figura 11 – Exemplo de roteamento

2.2.3.1 Injeção de Dependência

Por fim, não se pode deixar de falar da Injeção de Dependência ao falar da arquitetura de um projeto angular, pois é esta que torna toda essa estrutura possível de forma eficiente.

Injeção de dependência, nada mais é do que um padrão que visa diminuir o nível de acoplamento do projeto. Angular utiliza este padrão para fornecer instâncias de componentes e serviços para outros componentes e serviços. Basicamente, angular procura no construtor da classe por tipos de componentes ou serviços que serão necessários, para que ele possa criar uma instância deste e disponibilizar no momento em que ele for necessário.

2.3 IHC

IHC se refere ao estudo das relações entre humanos e computadores, que se dá através das interfaces sejam elas gráficas ou físicas. Pode-se utilizar como definição mais formal a dada pela SBC (2019) que diz “A área de Interação Humano-Computador (IHC) se dedica a estudar os fenômenos de comunicação entre pessoas e sistemas computacionais que está na interseção das ciências da computação e informação e ciências sociais e comportamentais e envolve todos os aspectos relacionados com a interação entre usuários e sistemas. A pesquisa em IHC tem por objetivo fornecer explicações e previsões para

fenômenos de interação usuário-sistema e resultados práticos para o projeto da interação.”

Assim, esta área abrange desde a tomada de decisão para concepção de uma interface, através da qual ocorrerá essa comunicação, até sua validação através de testes de usabilidade, que serão abordados neste trabalho. Todo esse processo deve levar em consideração as características, capacidades e limitações de ambas as partes.

No caso do ator humano, é necessário entender como ele capta, processa, interpreta e interage com diferentes estímulos. [Card, Moran e Newell \(1983\)](#) por exemplo, modelaram o processo de receber informação, analisar e agir de acordo com os dados recebidos e chegaram ao Modelo do Processador de Informação Humano (MPIH) que consiste em uma descrição de como o ser humano processa as informações provindas de uma interação com algum sistema computacional a fim de entender e predizer como será o comportamento do usuário. Este modelo, é dividido em três subsistemas, o sistema perceptual, sistema motor e o sistema cognitivo. O primeiro trata dos estímulos sensoriais obtidos da interação com o sistema, o segundo obviamente controla as ações do ator que são tomadas através do processamento dos dados, provindos do sistema perceptual, pelo terceiro sistema.

Já ao se tratar do segundo ator (computador), questões como o tipo de dispositivo e seus recursos disponíveis para serem utilizados, que ficarão encarregados de interagir com o usuário devem ser levados em consideração para projetar uma interação eficiente e dinâmica.

Entender os atores e como se dá essa interação entre eles, permite que engenheiros, desenvolvedores ou designers desenvolvam layouts que possibilitem ampliar qualidades essenciais para uma boa interface, características que giram em torno da usabilidade, acessibilidade e aceitabilidade.

2.3.1 *Framework PACT*

PACT é uma sigla que significa Pessoas, Atividades, Contextos e Tecnologias, um *framework* que auxilia o desenvolvimento de sistemas interativos ([BENYON; TURNER; TURNER, 2005](#)). Em outras palavras, ele garante que o sistema em questão seja fácil de ser utilizado pelas pessoas para o qual foi desenvolvido, para realizar as atividades propostas nos contextos propostos utilizando os recursos que a tecnologia utilizada proporciona.

A interação entre pessoas e sistemas pode variar bastante ao se levar em consideração esses 4 elementos do PACT, por exemplo, as limitações das pessoas que farão uso do sistema podem exigir cuidados especiais na forma do sistema se comunicar com o usuário. Da mesma forma essa comunicação dependerá do tipo de atividade a ser realizada, o contexto da utilização do sistema também deverá ser levado em consideração pois poderá atribuir novas restrições nesse processo e obviamente uma vez que a tecnologia evolui novas formas surgem de estabelecer essa comunicação ou a natureza das atividades

muda ([BENYON; TURNER; TURNER, 2005](#)) e por isso também devem ser levados em consideração.

2.3.1.1 Pessoa

Pessoa é o elemento mais importante do *framework*, já que o sistema está sendo desenvolvido para que ela possa utilizá-lo. Assim, o designer deve compreender que pessoas diferem entre si por inúmeras características, as quais podem ser diferenças físicas, diferenças psicológicas ou sociais.

Ao se tratar de diferenças físicas é intuitivo pensar em características como altura e peso, no entanto, no que tange à utilização de um sistema é necessário levar em consideração a diferença das habilidades cognitivas dos usuários, visto que a deficiência de algum dos sentidos pode determinar uma limitação que o sistema deve cobrir. [Reinius \(2011\)](#) explica que a verdadeira questão a ser levantada nesse quesito é como os aspectos físicos das pessoas influenciarão na utilização do sistema e por isso devem ser levados em consideração.

Quanto às diferenças psicológicas, pode-se considerar a capacidade de memorização, capacidade de se localizar em determinados ambientes, o que pode ser contextualizado em se localizar em um site, capacidade de assimilação de palavras, números e figuras ou até mesmo a questão emocional ao se realizar uma atividade sob pressão.

Já quanto as diferenças sociais o designer deve levar em consideração o nível de experiência do usuário para com o programa ou a tecnologia incorporada, as motivações e objetivos que as levam a utilizar o sistema.

2.3.1.1.1 Modelos Mентais

Basicamente, modelo mental é o entendimento e a compreensão individual que as pessoas tem de alguma coisa, como ela funciona, e no caso de sistemas essa compreensão se abrange à uma visão sobre o que o sistema faz e como deve-se operá-lo para realizar alguma tarefa.

Ter um modelo mental correto de um sistema faz com que um usuário em caso de erro consiga se recuperar e retornar a um estado conhecido, caso contrário, o erro pode fazer com que o usuário não consiga se recuperar e isso comprometa sua interação com o sistema.

Dessa forma o design de uma interface deve apresentar informações suficientes para que um usuário consiga construir um modelo mental útil e correto do sistema e dessa forma a interação será como esperada.

Para avaliar se uma interface está gerando modelos mentais corretos para um usuário utilizam-se modelos de ciclo de execução-avaliação como o Modelo de Donald Norman. Este modelo sugere 7 passos que o usuário toma ao interagir com um sistema:

1. Usuário estabelece a meta.
2. Formula intenção.
3. Especifica ações na interface.
4. Executa ações.
5. Percebe o estado do sistema.
6. Interpreta o estado do sistema.
7. Avalia o estado do sistema com respeito à meta.

Ao aplicar a avaliação é necessário levar em consideração se a meta estabelecida foi atingida, se o estado de meta atingida é perceptível ao usuário, e se o processo tomado até este estado teve desvios ou caminhos fora do contexto. Além disso é necessário filtrar também os tipos de erros cometidos. Erros podem ser classificados em dois grupos: *slip* e *mistake*.

Slips são erros que geralmente acontecem em um contexto em que o usuário comprehende corretamente o sistema e sua meta, formula corretamente a ação a ser tomada porém, toma uma ação incorreta e por isso deve ter um impacto negativo baixo na avaliação da interação. Já *mistake* ocorrem em contextos em que o usuário não comprehende corretamente o sistema e provavelmente nem a meta e por isso cabe um impacto negativo maior na avaliação.

2.3.1.2 Atividade

As características de atividade que o *framework* sugere levar em consideração são os aspectos temporais, cooperação, complexidade, natureza do conteúdo e segurança crítica.

Aspectos temporais se referem à frequência em que a atividade será performada, já que quanto maior a frequência de execução mais fácil a atividade se torna de ser executada, se há limites de tempo para que uma atividade seja executada, e até se há alguma restrição quanto ao tempo de resposta do sistema (REINIUS, 2011).

O aspecto de cooperação tem relação com a necessidade ou não de se realizar a tarefa sozinho. Caso seja necessária a presença de mais de um usuário para realizar a tarefa, comunicação e coordenação se tornam fatores importantes no estudo da interface.

A complexidade se torna relevante ao se definir uma tarefa. Tarefas mal definidas ou vagas tendem a fazer com que o usuário necessite de ajuda para realizá-las.

O fator segurança crítica institui a necessidade do designer de explorar a capacidade de um erro ser cometido no sistema e analisar o quanto impactante tal erro seria. Em seguida, tentar criar *safeguards* para que o sistema possa se recuperar desse erro ou diminuir o impacto causado caso a primeira alternativa não seja possível.

Quanto a natureza do conteúdo, é necessário entender os tipos de interação que a atividade requer, como por exemplo, a entrada de informações como data, telefone ou algo do tipo requer um dispositivo de entrada. Assim, como cita Benyon, Turner e Turner (2005), os meios de comunicação que a atividade requer são tão importantes quanto os dados da interação.

2.3.1.3 Contexto

Segundo o *framework* PACT o contexto também pode ser classificado em três tipos: contexto social, contexto organizacional e circunstâncias físicas em que as atividades acontecem.

Contexto social se refere às adjacências da atividade, por exemplo, se há manuais ou pessoas que possam auxiliar a realizar a atividade, ou ainda se há questões como privacidade. Portanto o ambiente no qual a atividade está inserida deve ser avaliado para estar de acordo com as necessidades da atividade.

Da mesma forma o contexto organizacional considera questões relacionadas ao ambiente corporativo que está inserido, como por exemplo, questões de permissões ou nível de acesso devem ser levados em conta.

As circunstâncias físicas já direcionam a atenção à infraestrutura e descrição do local propriamente dito em que a atividade será executada. Disponibilidade de algum recurso, como por exemplo acesso à internet, se a atividade será realizada ao ar livre ou em ambiente fechado etc.

2.3.1.4 Tecnologia

O último elemento do *framework* se refere à tecnologia. Sistemas interativos comumente consistem de um *hardware* e *software* que fornecem alguma forma de *output* através de algum *input*. A evolução das tecnologias faz com que sua classificação seja dificultada, portanto, analisar as novas possibilidade de ferramentas pode trazer ganhos consideráveis no modo em que a comunicação se dará. Sendo assim, tecnologia considera 4 fatores a serem avaliados: *input*, *output*, comunicação e conteúdo.

Ao se tratar de *inputs* e *outputs*, o *framework* se refere ao modo e aos tipos de dispositivos que serão utilizados para entrada e saída de dados. A natureza dos dados é

determinante para a escolha do dispositivo mais adequado.

Comunicação se refere ao modo como a interação ocorre de uma ponta a outra, por exemplo, o sistema deve deixar claro para o usuário que algo está acontecendo ou aconteceu através de pop-ups, modais de confirmação, modais de notificação, mensagens de erros etc. se preocupando também quanto à forma em que o estado a ser informado é repassado, mensagens precisas e linguagem utilizada adequada de acordo com a expertise do usuário.

Quanto ao conteúdo, os dados apresentados devem ser relevantes, precisos, atualizados e compreensíveis. Caso qualquer uma dessas características deixe de ser cumprida, em certos casos, o sistema deixa de atender ao seu propósito.

2.3.2 Processo de design

O processo de design consiste de 4 etapas: Entendimento, Design, Antecipação e Avaliação.

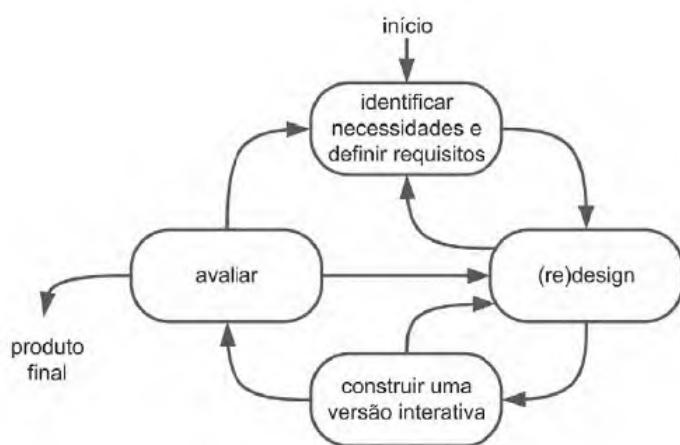


Figura 12 – Processo de design [Fonte: (DINIZ; SANTANA, 2010)]

O entendimento, como o próprio nome diz, consiste em entender o que o sistema deve realizar e como ele deve realizar. Basicamente, se torna um processo de análise de requisitos adicionando um foco especial para a interface. A utilização do framework PACT é de grande ajuda nesta etapa visto que ele cobre todos os elementos envolvidos e como eles interagem entre si, identificando possibilidades e restrições.

A etapa de design pode ser dividida em duas subetapas, o design conceitual e o design físico. No design conceitual são desenvolvidos modelos conceituais cuja função é identificar os principais objetos do sistema e como eles se relacionam com seus atributos, além da criação de cenários de utilização. Ferramentas como Modelos de entidade-relacionamento (MER) e diagramas de casos de uso são bastante utilizadas nesta etapa.

Já no design físico a prioridade é direcionada a como as coisas funcionarão e com a aparência do produto.

A etapa de antecipação tenta esboçar através de alguma mídia as ideias provindas das etapas anteriores para que possam ser avaliadas e validadas na etapa seguinte. Nesta etapa podem ser utilizados esboços, protótipos de alta ou baixa fidelidade, storyboards etc.

Já na etapa de avaliação, o fruto da etapa anterior será avaliado, podendo este processo ser feito através de inspeções realizadas por especialistas ou testes realizados por usuários. Existem vários métodos que podem ser utilizados nesta etapa para cada meio utilizado, por exemplo avaliações realizadas por especialistas podem utilizar avaliações heurísticas ou percurso cognitivo e avaliações realizadas por usuários podem utilizar *think aloud*, *coaching* ou co-descoberta. Cada um destes métodos, possui um processo com particularidades e a escolha de qual será utilizado dependerá de fatores como o tipo de antecipação realizada, quais informações queremos obter com a avaliação e os atores participantes.

2.3.2.1 *Wizard of Oz*

Wizard of Oz é uma das técnicas de prototipação possíveis de serem utilizadas na etapa de antecipação. É uma técnica que produz um protótipo de papel que não necessariamente tem de ser fiel à interface final. Este protótipo tem a função de simular o comportamento do sistema utilizado por uma pessoa. Este processo é feito da seguinte forma: o usuário simula a realização de alguma ação, o “mago” realiza a atualização que o sistema supostamente geraria através de novos protótipos de papel e esse ciclo se repete até o final da avaliação do protótipo (CAHÚ, 2016).

2.3.3 *Think Aloud*

Esta é uma técnica de avaliação que utiliza participação de usuários. Basicamente, o usuário recebe um contexto básico e é colocado para realizar uma série de atividades previamente definidas pelo designer, e o mais importante: é instruído a verbalizar ao máximo suas ações, o raciocínio por trás de cada ação e suas sensações durante o processo. Tal técnica permite identificar se o modelo mental construído pelo usuário coincide com o modelo mental que o designer espera criar com a sua interface além de identificar pontos de frustração e de ocorrência de *mistakes* (DIX et al., 2003).

3 Desenvolvimento

O desenvolvimento deste projeto seguiu a metodologia Scrum em correspondência com as 4 etapas do processo de design, sendo que em cada uma delas constam as sprints que compreenderam cada etapa do processo e seus resultados. Lembrando que o processo de design é iterativo, ou seja, o ciclo do processo de design pode ter ocorrido mais de uma vez.

Antes de iniciar a etapa de entendimento, algumas observações foram feitas. Apesar de se ter dados sobre o corpo docente atual da FAMAT, foi levado em consideração que o projeto desenvolvido deve estar apto a atender necessidades de quaisquer pessoas que possam vir a se tornar professores futuramente, excluindo pessoas com características físicas restritivas como deficiência visual e motora, pois uma adaptação dessa magnitude tomaria muito mais tempo do que o disponível para execução do projeto e sendo tais mudanças adicionadas à seção sugestão de trabalhos futuros.

3.1 Planejamento Scrum

3.1.1 Entendimento

3.1.1.1 O processo de distribuição de disciplinas na FAMAT

A presente seção retrata o processo atual de distribuição de disciplinas da FAMAT para fins de automatização do sistema, objetivo deste trabalho.

Próximo ao final do semestre uma comissão formada por alguns professores do departamento inicia os preparativos para realizar a distribuição de disciplinas que serão lecionadas no próximo semestre. A distribuição das disciplinas ocorre levando-se em consideração um ranking de pontuação calculado com base nas atividades realizadas por cada professor. Assim, antes da distribuição é necessário que cada professor registre as atividades executadas no semestre.

Ao registrar uma atividade o professor deve informar: à qual o anexo, dentro de 4 possíveis, a atividade pertence; os campos a serem preenchidos de acordo com cada anexo e tipo de atividade se encontram no arquivo [B](#) que consta na seção de anexos.

Uma vez que todos os professores registraram suas atividades realizadas, é então feita uma separação do corpo docente em três grupos, G1, G2 e G3, e gerado o ranking de professores. O processo utilizado para realizar tal agrupamento se encontra no arquivo [A](#) que consta na seção de anexos.

Logo em seguida os professores devem escolher pelo menos quinze disciplinas que

Nome:	Helman
Idade:	40 anos
Sexo:	Masculino
Posição:	Coordenador
Intimidade com tecnologia:	Média
Atividades:	Baixa quantidade de atividades.
Descrição:	Confortável com tecnologia, usa um computador desktop localizado em sua sala da faculdade. Ele permanece na faculdade durante 10 horas dentre as quais, 3 horas é voltada para o uso do computador.
Atuação no sistema:	Realizar o processo manual é uma tarefa que demanda muito tempo e esforço como administrador do sistema e além disso, Helman também tem de realizar sua distribuição manualmente como professor, isso faz com que seja totalmente à favor de um sistema que automatize o processo de distribuição.

Tabela 1 – Primeira Persona

desejam lecionar no semestre seguinte e ordená-las conforme sua prioridade. Uma vez finalizada essa etapa acontece então a distribuição das disciplinas entre os professores. As regras que norteiam o processo de distribuição se encontram no arquivo [A](#) que consta na seção de anexos.

3.1.1.2 Personas

Para realizar esta etapa, foram criadas 3 personas, Helman, Yohan e Aline, que abrangem a maior parte dos estereótipos dos potenciais usuários reais, mostrados nas tabelas [1](#), [2](#) e [3](#).

3.1.1.3 Cenários

Para complementar a base utilizada nesta etapa, foram estabelecidos alguns cenários de utilização do sistema. Foram abordados cenários principais de execução do sistema e cenários com mecânica similar a algum já mencionado. Foram definidos:

Cenário 01 - Iniciando novo semestre : Próximo ao final do semestre letivo, o pro-

Nome:	Yohan
Idade:	60 anos
Sexo:	Masculino
Posição:	Professor efetivo
Intimidade com tecnologia:	Baixa
Atividades:	Baixa quantidade de atividades.
Descrição:	Próximo de se aposentar, não possui habilidades com computadores e baixa coordenação motora, em suas aulas usa somente livros e cadernos, somente quando necessário utiliza o computador em sua sala.
Atuação no sistema:	Yohan não é muito receptivo à utilização de novos sistemas para substituir um processo que já funciona a vários anos, principalmente devido à sua inexperiência com informática.

Tabela 2 – Segunda Persona

Nome:	Aline
Idade:	29 anos
Sexo:	Feminino
Posição:	Professora efetiva
Intimidade com tecnologia:	Alta
Atividades:	Grande quantidade de atividades.
Descrição:	Conhece bastante de tecnologia e computadores e está sempre conectada à internet. Utiliza notebook tanto em sua sala quanto em aula.
Atuação no sistema:	Para Aline todo o período de distribuição de disciplina é bastante chato e burocrático, principalmente por ser realizado manualmente. Aline é uma incentivadora da automatização desse processo.

Tabela 3 – Terceira Persona

essor Helman inicia os preparativos para gerar a distribuição de disciplinas para o semestre seguinte. Ele acessa o sistema com seu login de administrador. Verifica que a maioria das turmas disponíveis no semestre anterior também estarão disponíveis no próximo semestre, assim, ele acessa a aba “Novo Semestre” cria um novo período e clona as turmas do período passado para o período recém-criado.

Cenário 02 - Cadastro de turma : Helman verifica que uma nova disciplina foi adicionada ao repertório oferecido, portanto, deseja criar uma turma para que essa disciplina possa ser ofertada. Ele acessa o sistema com seu usuário de administrador, acessa a seção de cadastro de turmas, seleciona a disciplina, preenche as demais informações solicitadas e salva.

Cenário 03 - Cadastro de professores : Aline, recém-contratada, acessa o sistema e

realiza seu cadastro. Uma vez finalizado, Helman verifica em seu login de administrador que há um novo cadastro de professor para ser aprovado. Helman clica no cadastro para ver as informações e valida. Como estão todas as informações corretas, ele aprova o cadastro.

Cenário 04 - Registro de atividades : Yohan após orientado pelo sistema, deseja realizar o cadastro de suas atividades realizadas no semestre. Ele acessa o sistema com seu login e entra na seção “Registro de atividades”. Ele então, para cada atividade que deseja inserir, seleciona o tipo de atividade, preenche os campos requisitados e salva.

Cenário 05 - Abertura do período de seleção de disciplinas : Helman, após finalizar o período de registro de atividades, deseja abrir o período de seleção de disciplinas. Ele abre esta seção entra com as configurações desse período de seleção e salva a alteração.

Cenário 06 - Seleção de disciplinas : Aline, após abertura do período de seleção de disciplina, deseja realizar a sua seleção. Ela acessa o sistema com seu usuário, e se dirige à seção “Seleção de disciplinas”. Aline utiliza os filtros do sistema para visualizar apenas turmas lecionadas nos dias e turnos em que ela possui disponibilidade. Ela adiciona as disciplinas desejadas, ordena as disciplinas de acordo com a prioridade que ela deseja e então salva sua seleção.

3.1.1.4 PACT

Nas tabelas [4](#), [5](#), [6](#) e [7](#) encontra-se a análise PACT realizada, separada por elemento do *framework*. A análise incorpora as considerações necessárias ao entendimento e desenvolvimento do sistema.

Uma vez realizada a análise PACT e levantada a base de conhecimento na fase de entendimento do processo de design, com o auxílio da documentação cedida pelo professor Juliano Gonçalves Oler, que consta no anexo [A](#), pode-se levantar os requisitos que servirão como base para o desenvolvimento do projeto.

3.1.1.5 Análise de Requisitos

3.1.1.6 Prioridade dos Requisitos

Para estabelecer a prioridade dos requisitos foi adotado o seguinte padrão:

Essencial: É o requisito sem o qual o sistema não deixa de executar alguma função vital. Requisitos essenciais são requisitos imprescindíveis e tem prioridade na implementação.

Diferenças Físicas:	Altura, peso, gênero, idade, aparência, habilidade cognitiva e personalidade. Nenhuma medida deve ser tomada especificamente para pessoas com altura, peso, gênero e aparência específicos. No entanto, no que tange habilidades cognitivas e idade, deve-se levar em consideração a utilização de letras maiores e cores de fácil contraste para auxiliar pessoas com visão prejudicada. Quanto à personalidades mais suscetíveis a resistir à utilização do sistema, pode-se focar em tornar a sua utilização mais amigável e agradável.
Diferenças Psicológicas:	Pessoas mais ou menos lógicas. Devem ser levadas em consideração que pessoas menos lógicas podem ter dificuldade de encontrar o caminho correto para executar uma tarefa caso não esteja bem especificado. O ideal é que os caminhos sejam bem definidos e que seja utilizada redundância.
Diferenças Sociais:	Pessoas com mais e menos experiência com informática. Desenvolver telas para que pessoas com pouca experiência com informática consigam ter uma boa desenvoltura sem precisar de auxílio, como telas simples, poucos caminhos possíveis.

Tabela 4 – Análise do elemento Pessoas

Importante: É o requisito sem o qual o sistema ainda funciona, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas se não forem implementados, o sistema poderá ser implantado e usado mesmo assim.

Desejável: É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

3.1.1.7 Requisitos Funcionais

RF-001 Login: o sistema deve permitir que professores e administradores efetuem o login através do número do seu Siape e senha pessoal e conta administradora e senha respectivamente. Prioridade: Essencial.

RF-002 Criar um novo semestre: o sistema deve permitir a criação de um novo período que compreende um semestre letivo, para que seja percorrido todo o ciclo de distribuição de disciplinas. Ao criar o novo semestre, deve apresentar a opção de clonar a estrutura de turmas de semestres anteriores para o novo semestre, sem manter a distribuição das turmas entre o corpo docente. Prioridade: Essencial.

Aspectos temporais:	A maioria das atividades serão realizadas apenas uma vez, são elas: cadastro do usuário, disciplinas, tipos de atividades e de turmas. Outras atividades serão realizadas uma vez por semestre, são elas: criação de um novo semestre, abertura e fechamento dos períodos de seleção de disciplinas e registro de atividades, seleção de disciplina e registro de atividades. A criação de uma opção de clonagem de turmas permite reduzir drasticamente o trabalho de cadastro que seria necessário todo semestre.
Cooperação:	A necessidade de ajuda para utilizar o sistema deve ocorrer apenas em casos onde o usuário possui alguma deficiência motora ou cognitiva.
Complexidade:	A maior complexidade se encontra nas atividades do administrador, no entanto para ambos, a ordem dos itens nos menus será estabelecida de acordo com a ordem necessária de execução, além disso, todas as telas apresentam orientações e acompanhamento visual ou subliminar para auxiliar.
Segurança Crítica:	Nenhum erro crítico pode ser admitido sem que seja possível sua reversão, por isso a realização de backups devem ser implementada e também novos cadastros ou edição de algum objeto já salvo deve ser possível.
Natureza do conteúdo:	As atividades realizadas no sistema não necessitam de nenhum tipo de dispositivo diferente de um mouse, teclado e monitor presentes no computador para utilização do sistema.

Tabela 5 – Análise do elemento Atividades

RF-003 Manter disciplinas: o sistema deve gerenciar (inserir, remover, buscar, editar) as disciplinas ofertadas pela FAMAT, para os diversos cursos de graduação da UFU. Mantendo um link para a ementa de cada uma, seu código, seu nome, sua carga de horas aulas semanais, a qual divisão ela pertence e a qual unidade acadêmica/curso é destinada. Prioridade: Essencial.

RF-004 Manter docentes: o sistema deve gerenciar (inserir, remover, buscar, editar) os docentes da FAMAT. Mantendo nome, siape, data de admissão, tipo de contrato, divisão a qual ele pertence, email, situação do cadastro no SIGAD, se este é um dos administradores, sua pontuação, a quantidade máxima de horas permitidas para lecionar, o grupo ao qual pertence, se está de licença, se realiza alguma atividade administrativa, se é um assessor administrativo, quais as turmas sob sua responsabilidade e senha de cada professor. Prioridade: Essencial

RF-005 Manter tipos de atividades: o sistema deverá gerenciar (inserir, remover,

Contexto Social:	<p>O sistema pode ser utilizado de qualquer local, seja dentro da universidade ou fora, não há restrições quanto a isso. Inicialmente o sistema não emite sons e por isso não há a necessidade de restringir seu uso em locais onde não se pode fazer barulho. No entanto, existem implicações relevantes sobre a necessidade de privacidade ao ser utilizado de acordo com a preferência do usuário, assim, atividades relacionadas ao usuário como registro de atividades e seleção de disciplinas devem ter uma margem de tempo configurável para serem realizadas para que o usuário consiga realizar tal atividade em um momento oportuno.</p>
Contexto Organizacional:	<p>Como dito no tópico acima, o acesso ao sistema não possui restrições de local. Existem restrições quanto a permissões para realizar as atividades. Deve haver um total de cinco usuários com permissão de administrador, os quais serão os únicos habilitados a acessar a área de administrador e realizar as atividades: aprovar, buscar e editar cadastro de novos semestres, clonar semestres, adicionar, buscar e editar turmas, disciplinas e tipos de atividades, editar os períodos de seleção de disciplina e registro de atividade. Quanto as atividades de professor, como selecionar disciplinas, registrar e alterar senha, devem ser realizadas individualmente através dos seus respectivos usuários.</p>
Circunstâncias Físicas:	<p>A única questão a ser levada em consideração nesse tópico é a necessidade de acesso à internet para utilizar o sistema. Para contornar uma restrição dessa magnitude algo como um aplicativo desktop que sincroniza o progresso do registro de atividades e seleção de disciplinas ao se conectar com a internet deve ser pensado, o que foge do escopo deste projeto.</p>

Tabela 6 – Análise do elemento Contexto

buscar, editar) os tipos de atividades acadêmicas que podem ser desenvolvidas por docentes. Mantendo o nome da atividade, o anexo ao qual ela pertence, sua pontuação, se essa pontuação deve ser multiplicada pela quantidade de meses de realização, a validade dos pontos desta atividade, sua descrição. Prioridade: Essencial.

RF-006 Manter atividades realizadas: o sistema deverá gerenciar (inserir, remover, buscar, editar) as atividades acadêmicas realizadas por docentes. Mantendo o nome da atividade, o anexo ao qual ela pertence, sua pontuação e quem a realizou. Além de manter os seguintes campos de acordo com o anexo:

Entrada:	Uma vez que as entradas no sistema consistem apenas de entradas de texto as únicas tecnologias necessárias a serem suportadas são a de teclado e mouse para entradas de seleção e navegação.
Saída:	Dispositivos de saída a serem suportados são telas de computadores que podem variar de tamanho, porém, a resolução mínima suportada sem distorções significativas deve ser de 1280x768.
Comunicação:	Como algumas das pessoas possuem pouca experiência com informática, a linguagem utilizada deve ser cautelosa ao utilizar termos técnicos e deve ser precisa. Além disso, toda e qualquer ação performada ou alteração de estado deve ser notificada ao usuário através de modais e <i>cards</i> de notificações.
Conteúdo:	Todas as informações para serem precisas, devem utilizar filtros para evitar apresentação excessiva de informações irrelevantes quando necessário.

Tabela 7 – Análise do elemento Tecnologias

Anexo 1: Área e ano do Qualis-CAPES, Agência financiadora. Anexo 2: Nenhum campo específico. Anexo 3: Horas aulas semanais, Nº de alunos na turma Anexo 4: Nenhuma campo específico.

Prioridade: Essencial

RF-007 Classificar docentes: o sistema deverá automaticamente classificar cada docente no momento em que um administrador fechar o período de registro de atividades de acordo com suas atividades realizadas, de modo que os docentes pertençam à um grupo. É realizada uma classificação para cada divisão (Matemática e Estatística). Prioridade: Essencial.

Os grupos serão divididos da seguinte forma:

G3: ficam os professores recém contratados e professores que realizaram pelo menos uma atividade de cada um dos anexos 1, 2 e 3, não excedendo 30% do número total do corpo docente, excluindo os docentes com tipo de contrato substituto, temporário e os docentes efetivos em licença. Dentro do grupo a colocação do professor será de acordo com a sua pontuação. G2: ficam os professores que realizaram pelo menos uma atividade de pelo menos dois dos anexos 1, 2, e 3, não excedendo 30% do número total do corpo docente, excluindo os docentes com tipo de contrato substituto, temporário e os docentes efetivos de licença. Dentro do grupo a colocação do professor será de acordo com a sua pontuação. G1: ficam os demais professores que não foram classificados em nenhum dos grupos anteriores.

RF-008 Consultar sua posição no ranking do seu grupo: o sistema deve permitir que o professor consulte sua posição no ranking dentro do seu grupo após o período de registro de atividades ter sido encerrado. A identificação dos professores no ranking deve ser feita através do número do seu Siape. Prioridade: Baixa

RF-009 Listar disciplinas preferenciais dos docentes: o sistema deverá, em um período especificado, permitir que cada docente efetivo da FAMAT, após tomar consciência de sua classificação, e das disciplinas a serem ofertadas, listar e selecionar no mínimo quinze disciplinas que gostaria de ministrar no semestre letivo seguinte e ordená-las por preferência com o auxílio de filtros por turno, curso e dia da semana, e também com um link disponível para conferir a ementa das disciplinas. Prioridade: Essencial

RF-010 Processar atribuição de disciplinas: o sistema deverá, no período especificado, depois de cada docente indicar sua preferência de disciplinas e a partir da indicação de um administrador sobre quais grupos participarão, processar a distribuição efetiva de disciplinas de cada docente, começando pelo docente que obteve maior pontuação no período respectivo. A distribuição se dá em vários turnos, sendo que em cada um apenas uma turma pode ser atribuída a cada docente. Esse ciclo se repete até que todas disciplinas sejam distribuídas ou todos os docentes participantes da distribuição tenham atingido a carga mínima de horas aula semanais. As disciplinas remanescentes serão distribuídas manualmente pelo administrador do sistema. Prioridade: Essencial.

RF-011 Validar atribuições de disciplinas aos docentes: o sistema deverá depois de processar a distribuição de turmas ministradas por cada docente, permitir que um administrador modifique as distribuições, para atender pequenas demandas por parte de docentes, e suas contestações. Prioridade: Importante

RF-012 Esqueceu a senha: o sistema deve permitir que professores solicitem o reset de senha caso a esqueçam utilizando seu siape e email. Prioridade: Essencial

RF-013 Alterar a senha: o sistema deve permitir que professores realizem a alteração de suas senhas pessoais utilizando suas senhas atuais. Prioridade: Baixa

RF-014 Exibir relatório com informações pertinentes: o sistema deve exibir em um dashboard com informações pertinentes ao sistema. Prioridade: Baixa

Administrador: Se o período de registro de atividade está aberto ou fechado; Se o período de seleção de disciplina está aberto ou fechado; Quantos dias faltam para o fechamento do período de registro de atividade; Quantos dias faltam para o fechamento do período de seleção de disciplina; N° de professores; N° de disciplinas;

% de Professores que realizaram ou não seleção de disciplinas; % de Professores que realizaram ou não o registro de atividades;

Professor: Se o período de registro de atividade está aberto ou fechado; Se o período de seleção de disciplina está aberto ou fechado; Quantos dias faltam para o fechamento do período de registro de atividade; Quantos dias faltam para o fechamento do período de seleção de disciplina; Quantidade de turmas sob sua responsabilidade; Disciplinas mais concorridas.

3.1.1.8 Requisitos Não Funcionais

RNF-001 Plataforma web: é desejável que o sistema seja desenvolvido em plataforma web, por independência de sistema operacional. Prioridade: Desejável.

RNF-002 Segurança: o sistema deve manter o sigilo das atividades acadêmicas dos professores, bem como sua categoria e disciplinas ministradas. Prioridade: Essencial.

RNF-003 Servidores de aplicação e banco: o sistema deve ser hospedado entre os servidores da Universidade Federal de Uberlândia, com intuito da própria instituição manter o gerenciamento e a disponibilidade do sistema. Prioridade: Desejável.

RNF-04 Rotinas de backup: o servidor de banco deve ter rotinas administrativas, para que o sistema gere backups dos dados com regularidade mensal. Prioridade: Desejável.

RNF-05 Usabilidade: o sistema deve ser intuitivo e fácil de usar independente do nível de familiaridade do usuário com informática. Prioridade: Essencial.

RNF-06 Navegabilidade: deve ser possível acessar qualquer parte do sistema com no máximo 5 cliques. Prioridade: Desejável.

RNF-07 Aparência: o sistema deve apresentar uma interface amigável e contemporânea. Prioridade: Desejável.

3.1.2 Design

A partir das informações levantadas na etapa de entendimento, foi possível realizar alguns esboços iniciais de como seria o design conceitual. Como as atividades realizadas pelos professores desempenham uma função bastante importante no processo de distribuição de disciplinas, decidiu-se adicionar uma seção no sistema para gerenciar as atividades. Apesar do banco de dados utilizado ser NoSQL é possível identificar certos relacionamentos entre os documentos. Abaixo segue o diagrama entidade-relacionamento que foi fruto da análise da base de conhecimento gerada até o momento:

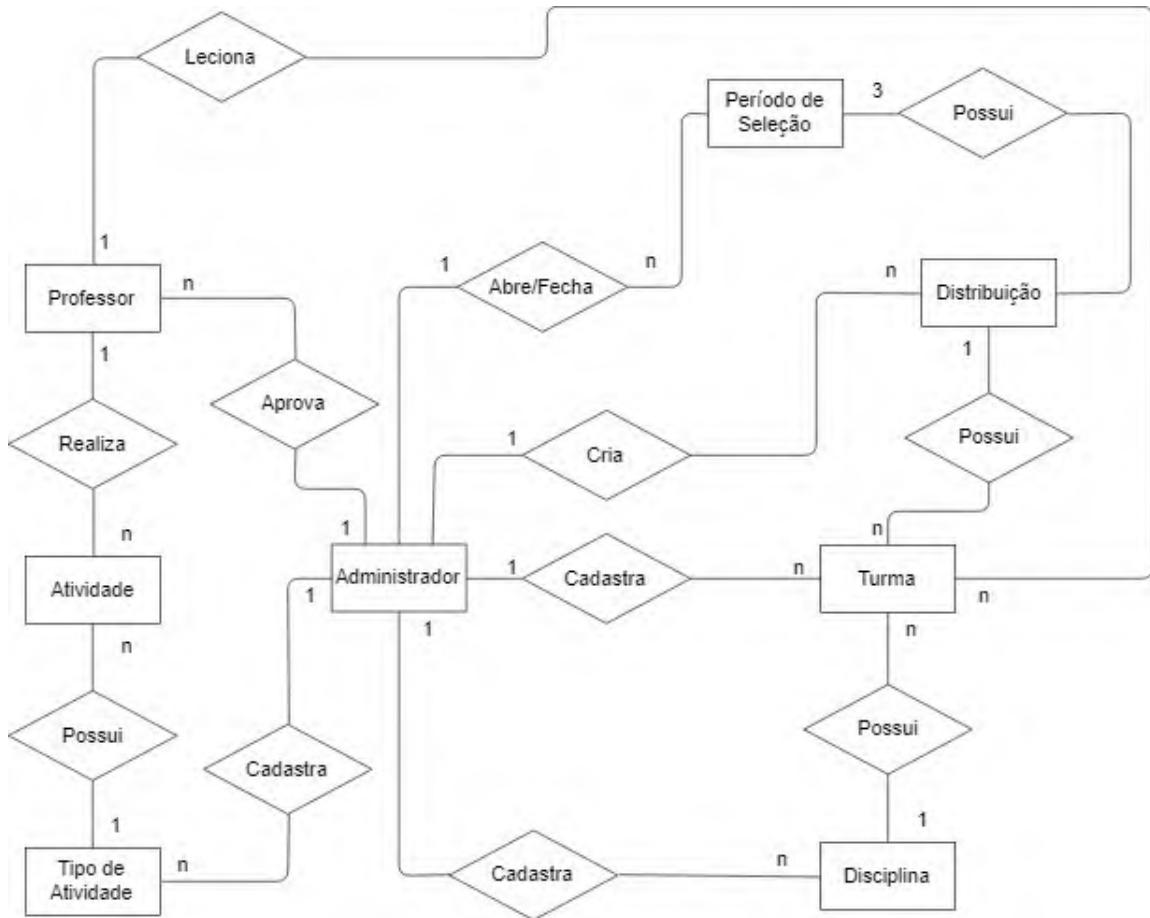


Figura 13 – Diagrama Entidade-Relacionamento

Dessa forma foram identificados os principais objetos do sistema e combinando com os requisitos gerados pode-se obter uma visão mais completa de cada objeto adicionando a eles seus atributos, conforme Modelo do Objetos apresentado na Figura 14. Uma vez definida uma visão conceitual de como o sistema funcionará pode-se começar a esboçar as telas. As figuras da presente seção representam este estudo e demonstram o esboço das ações possíveis no sistema.

Na tela inicial, a intenção utilizada foi a de deixar juntas as duas funções relacionadas ao acesso: login (Figura 15) e cadastro (Figura 16) para simplificar e facilitar ao máximo a entrada no sistema.

Na tela de criação de semestre, optou-se por tornar disponível o modelo padrão de selecionar o ano do novo semestre e o semestre (Figura 17), porém, caso seja de interesse do administrador ao assinalar a caixa de seleção abaixo dos campos do formulário inicial, novos campos aparecerão para possibilitar o processo de clonagem de semestres (Figura 18).

Na tela de cadastro de disciplinas, tipos de atividades, turmas (com uma leve exceção no caso de turmas, pois para seu cadastro há também a presença da matriz

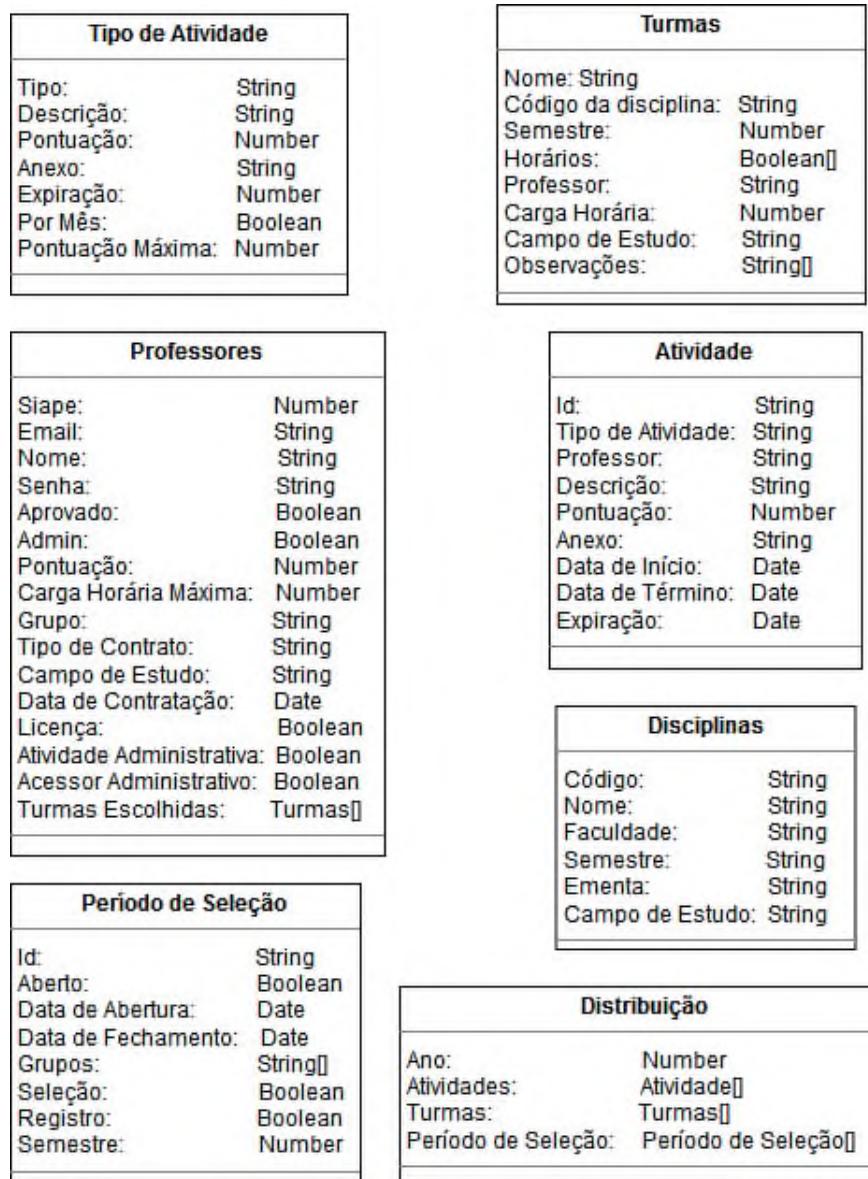


Figura 14 – Diagrama dos Objetos com seus atributos

de horários para ser preenchida) e atividades (Figura 19) foi seguido o seguinte padrão: cada aba direcionará o usuário para uma ação, assim, botões de navegação permanecem agrupados, mantendo e respeitando o conceito de menus de contexto, à esquerda dos campos de formulário há uma caixa de seleção para informar quando o dado de entrada nesse campo está válido, ou em outras palavras, esta etapa está concluída, à direita há um ícone que acompanha o campo do formulário servindo como redundância que remete à ação esperada para esse campo e botões de finalização permanecem agrupados.

Nas telas de busca de disciplinas, tipos de atividade, atividades e turma (Figura 20), o ideal foi utilizar um elemento que suportasse mostrar uma grande quantidade de elementos sem perder a noção da informação que está sendo disposta e ainda assim conseguir filtrar dentre essa gama de informações para obter resultados mais precisos. Por

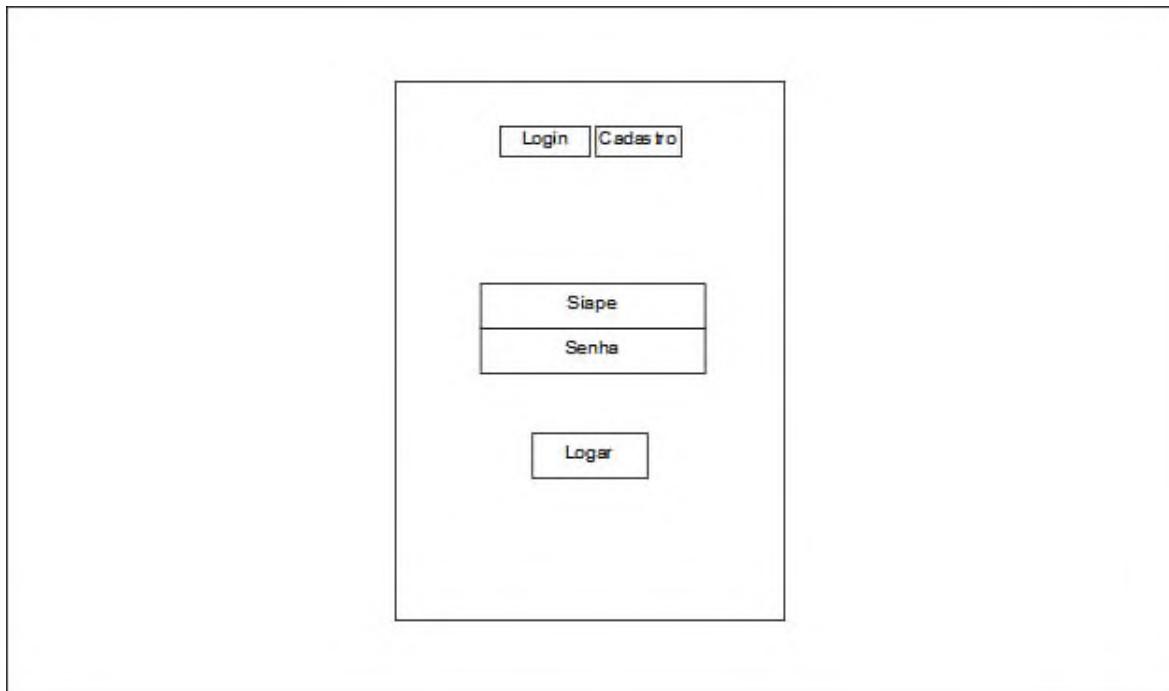


Figura 15 – Esboço da tela de login.

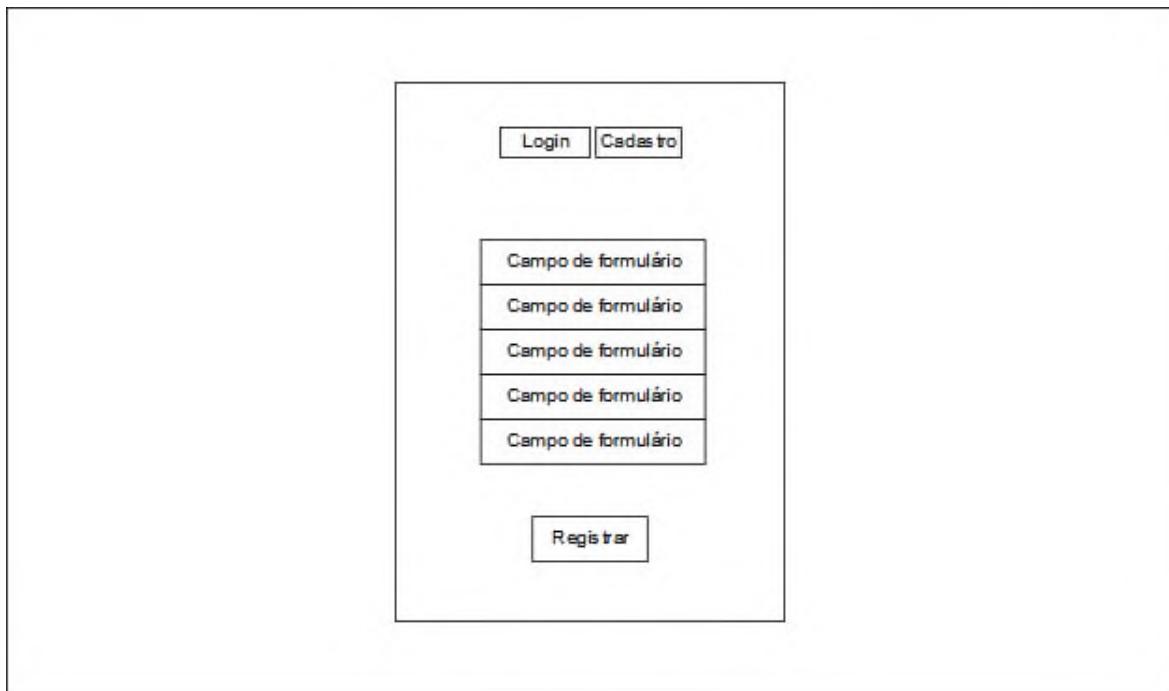
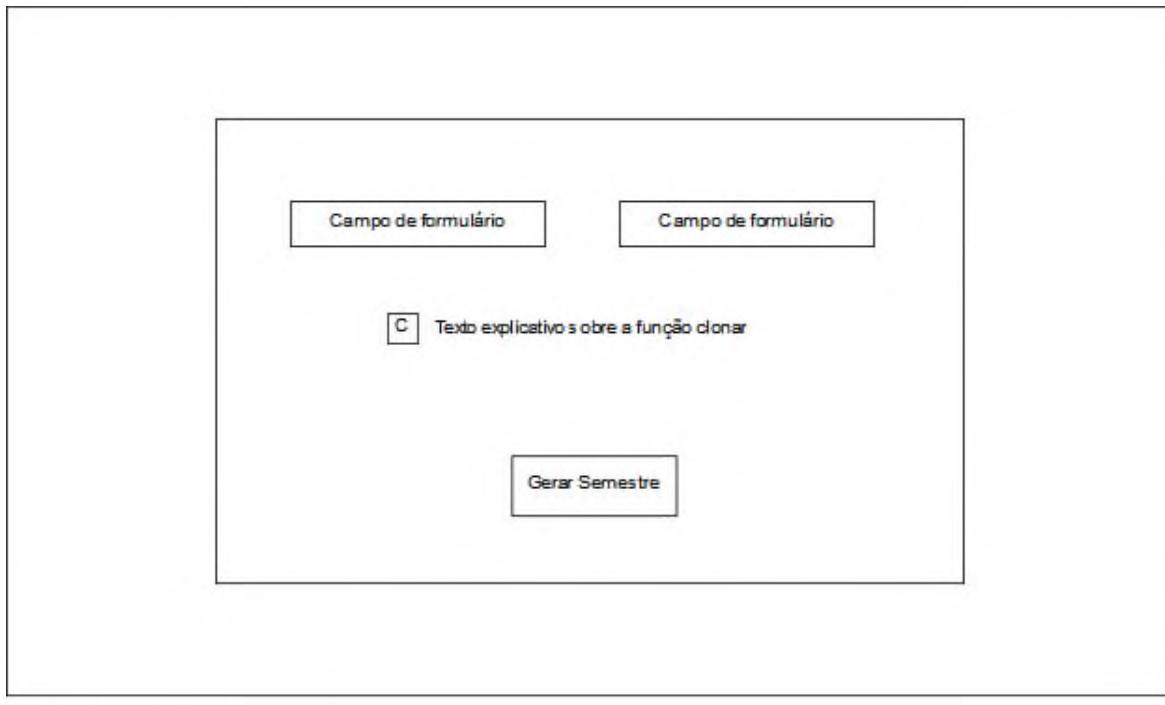


Figura 16 – Esboço da tela de cadastro.

estas razões optou-se pelo uso de uma tabela. Em algumas telas é possível e conveniente filtrar por ano e por isso um campo de seleção de ano foi adicionado ao lado do campo de busca. Ao final da tabela foi adicionado um conjunto de ícones que representam ações possíveis de serem realizadas com aquelas informações, que são editar e excluir.

Na tela de edição de disciplinas, tipos de atividades, atividades, professores e turma



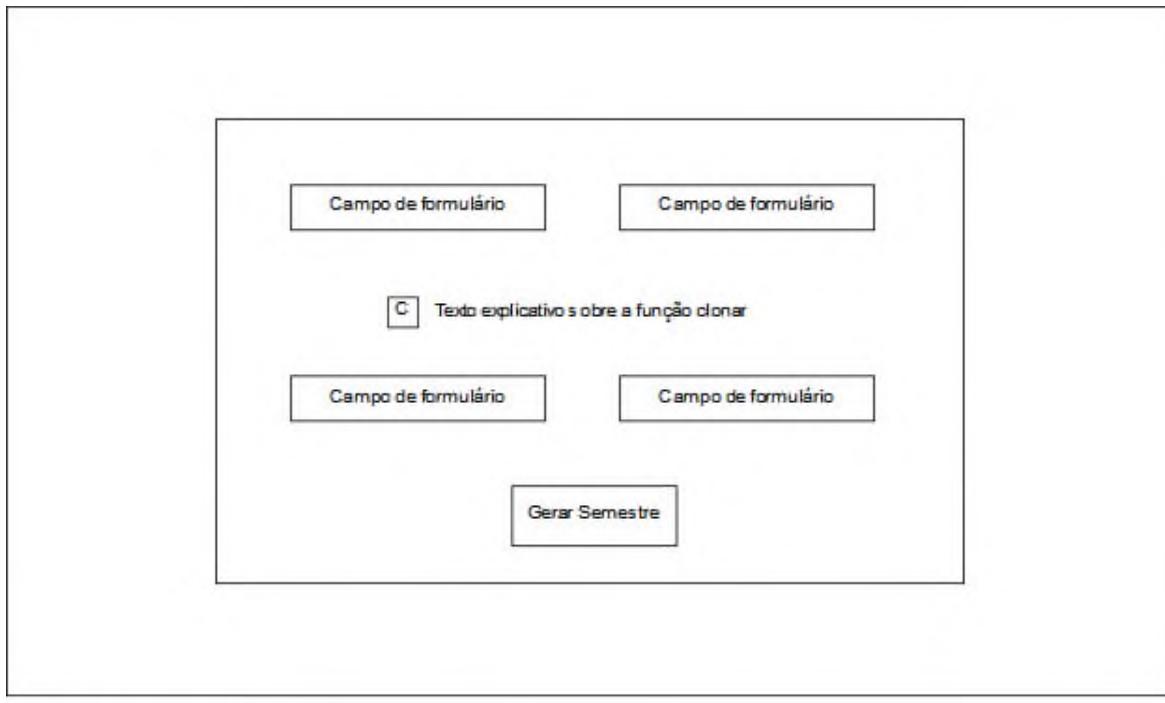
Legenda: C: Checkbox

Figura 17 – Esboço da tela de criação de semestre padrão.

(Figura 21), foi pensado um modo de deixar a edição dinâmica e integrada com a tabela da tela de busca. Assim, ao clicar no ícone que representa a edição uma janela é aberta mostrando todos os campos daquela informação e ao final da janela constam agrupados os botões de finalização.

Na tela de seleção de disciplina (Figura 22), buscou-se seguir algumas premissas. Deixamos agrupados os elementos que possuem o mesmo contexto, assim manter todos os filtros em um único local ou seção ajudaria na compreensão de como as informações estão sendo dispostas já que a intenção é que os filtros sejam aplicados um sobre os elementos filtrados do outro. Outro tópico levado em consideração seriam as informações que um professor precisaria para realizar a seleção de uma disciplina, que seriam os horários que essa disciplina seria lecionada e compará-los com os horários de outras disciplinas já selecionadas para que ele seja informado quando ocorrer algum choque de horário. Outra informação importante seria a visualização da ementa da disciplina, a qual o ideal seria que ao clicar no corpo do cartão de uma disciplina seria aberto uma nova aba no navegador que direcionaria o professor local que contém essa informação. Após selecionar as disciplinas o professor poderia utilizar a função de clicar e arrastar os cartões de disciplina para estabelecer a prioridade desejada.

A tela do período de seleção de disciplina do administrador mostra a primeiro os campos necessário para buscar o estado atual do objeto no banco (Figura 23)e assim não confundir o usuário com os demais campos. Uma vez preenchidos, a tela mostra o período



Legenda: C: Checkbox

Figura 18 – Esboço da tela de criação de semestre com clonagem.

de seleção de disciplina completo com os demais campos de configurações (Figura 24).

A tela de aprovação de cadastro de professores (Figura 25), foi pensada de modo a possibilitar visualização da quantidade de cadastros a serem avaliados e ao mesmo tempo visualizar as informações de um cadastro não aprovado específico sem a necessidade de trocar de tela. Ao final, constam os botões de finalização agrupados (*Aprovar* e *Reprovar*).

Para unir todas essas telas mencionadas até aqui, foi necessário pensar em um modo de navegação que fosse simples e intuitivo para evitar erros dos quais os usuários não consigam se recuperar. De preferência respeitando a regra de 5 cliques para chegar em qualquer lugar do sistema. Assim criou-se uma tela geral (Figura 26), composta por uma área de trabalho, onde aparecerão as telas abertas, um cabeçalho que terá o atalho para sair do sistema e uma barra lateral esquerda que possui ícones que representam cada seção e *links* para cada seção de acordo com a permissão de acesso do usuário. Utilizando o princípio da redundância, ao se passar o mouse sobre a barra lateral esquerda ela expande de tamanho mostrando a descrição de cada ícone (Figura 27).

3.1.3 Antecipação

Como a atividade essencial deste trabalho é a seleção de disciplinas e as demais ações consistem de simples cadastros, as duas etapas seguintes foram realizadas com apenas esse caso de uso.

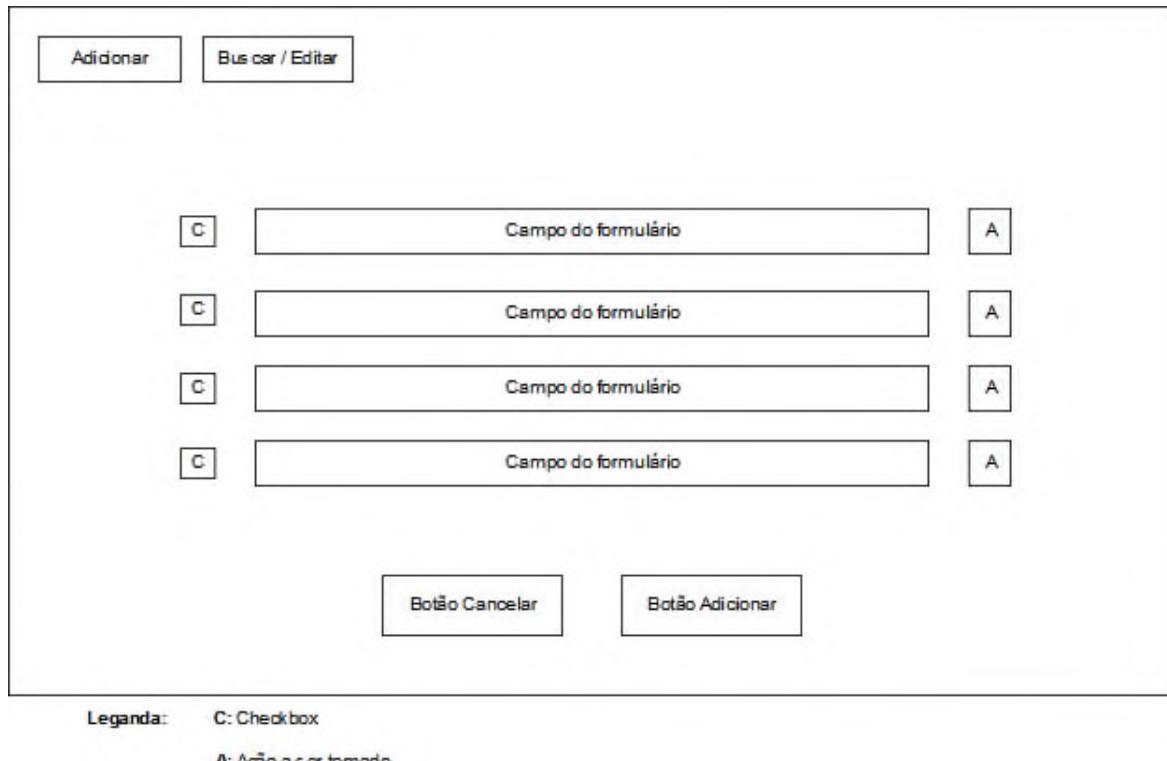


Figura 19 – Esboço da tela de cadastro de disciplinas, tipos de atividades, turmas e atividades.

Uma vez terminada a etapa de design, inicia-se o processo de prototipação, que consiste de um protótipo de papel de alta fidelidade utilizando a técnica *Wizard of Oz*.

Inicialmente, foi feito um desenho da tela inicial de seleção de disciplina dos professores. Em seguida, todas as telas de confirmação, menus de seleção e filtros foram desenhados e recortados. O próximo passo foi criar um conjunto de cartões de disciplinas que estariam teoricamente disponíveis nessa seleção. Ao terminar a criação dos cartões das disciplinas, foram planejados cartões de horários para cada disciplina disponível e cartões especiais em que acontecem sobreposição de horários de duas ou mais disciplinas. Além disso, para finalizar, foram criados os cartões de disciplinas selecionadas, que são os mesmos cartões das disciplinas disponíveis porém, com o ícone de adição alterado para remoção. O resultado da prototipação é mostrado na Figura 28.

3.1.4 Avaliação

3.1.4.1 Think aloud

Uma vez finalizada a prototipação, é possível agora avaliar tal protótipo para identificar possíveis falhas no modelo mental gerado por usuários ao utilizar tal sistema. A técnica utilizada para realizar essa avaliação foi o *Think Aloud*.

Nesse processo, foram selecionadas quatro pessoas aleatórias para utilizar o pro-

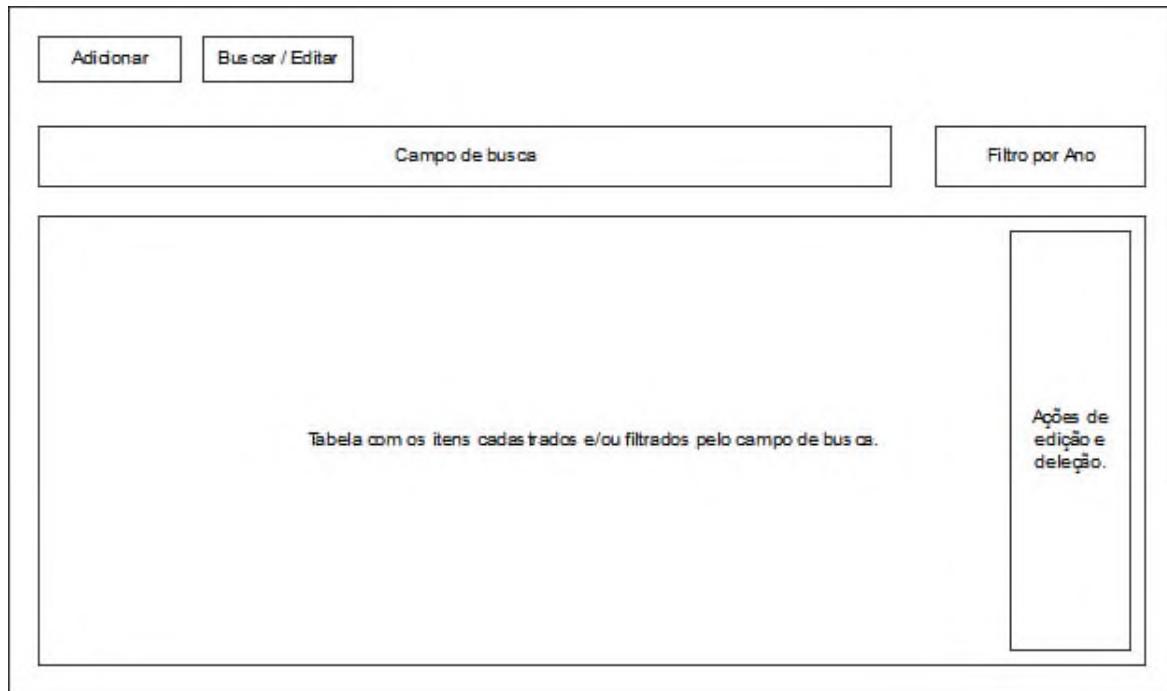


Figura 20 – Esboço da tela de busca de disciplinas, tipos de atividade, atividades, professores e turma.

tótipo e realizar uma sequência de atividades. Assim, foi gerado um documento com uma contextualização para o participante da avaliação junto com a sequência de atividades que ele deve executar. Além disso, os participantes foram instruídos a tentar verbalizar todo e qualquer raciocínio e sensação que eles tiveram durante a avaliação. O documento repassado aos participantes da avaliação está ilustrado na Figura 29.

Após a realização do processo as observações sobre melhorias foram as seguintes:

Caixas de seleção do filtro de dias da semana : A proposta inicial era a tela trazer as caixas de seleção dos dias desmarcados e o usuário iria selecionar os dias que gostaria que as disciplinas disponíveis mostradas tivessem horários nesses dias. No entanto, foi constatado que dentre as quatro avaliações realizadas, todos entenderam que ao assinalar uma caixa de seleção ele estaria excluindo as disciplinas que possuem horários no dia correspondente. Sendo assim, a correção para essa confusão no mapa mental seria deixar todos os dias selecionados e o usuário desmarcaria o dia que não quisesse que aparecessem disciplinas que possuem horários neste dia.

Ementa da disciplina : A proposta inicial da tela consiste de mostrar a ementa da disciplina desejada ao clicar no corpo do cartão da disciplina, porém, tal ação não estava clara para metade dos participantes das avaliações, que questionaram a falta de um modo de ver a ementa da disciplina durante a seleção, mesmo esta não sendo uma das atividades a serem executadas. A correção adotada para suprir essa falha

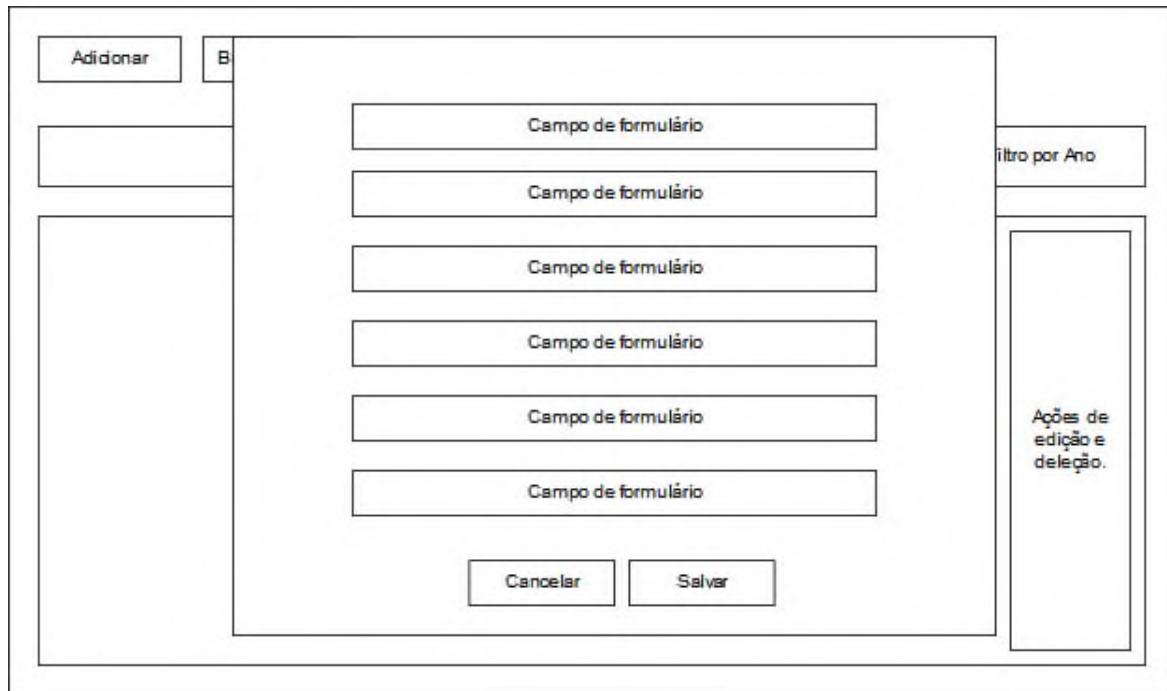


Figura 21 – Esboço da tela de edição de professores, disciplinas, tipos de atividade, atividades e turma.

foi a inclusão de um evento de *hover* (Ao passar o mouse em cima do objeto) no cartão da disciplina, informando a seguinte frase “Clique para ver a ementa”.

Identificador de filtros : Inicialmente, à extrema esquerda na seção onde se encontram os filtros havia um indicador “Filtros:”. A ideia era utilizar do princípio da redundância para ajudar a identificar funções. No entanto, a utilização deste recurso fez com que 50% dos usuários considerassem que apenas o campo “Turno” fosse um filtro propriamente dito. A correção para esse *mistake* foi a retirada desse identificador.

Estipular prioridades : Ao final do procedimento os usuários sentiram dificuldade ao realizar a tarefa número onze. Apenas um usuário intuitivamente realizou esta ordenação de prioridade da forma como foi pensada. Para resolver essa lacuna no modelo mental, utilizamos um outro evento de *hover* que altera o cursor do mouse para o cursor de clicar e arrastar.

As demais partes da tela geravam modelos mentais corretos e por isso as atividades foram concluídas de forma acertiva ou com ocorrência apenas de *slips*. Assim, uma vez que a tela foi validada e feita as devidas correções, podemos definir o *product backlog* para iniciar o desenvolvimento do sistema.



Figura 22 – Esboço da tela de seleção de disciplinas do usuário.

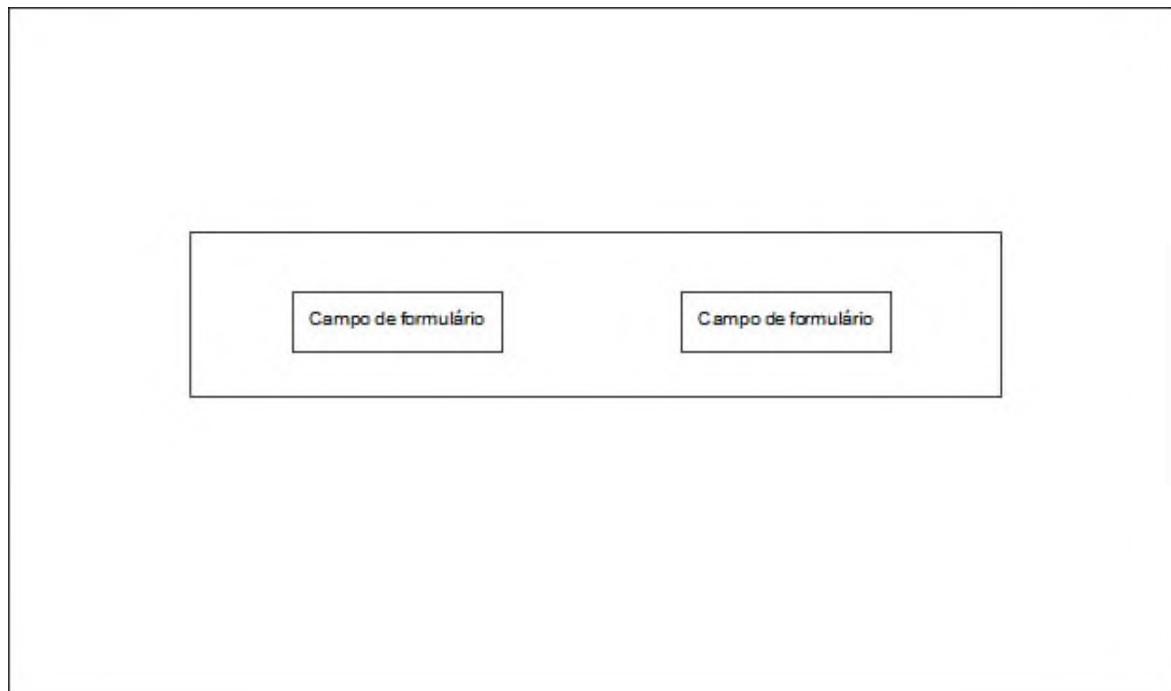


Figura 23 – Esboço da tela padrão de período de seleção de disciplinas do administrador.

3.1.5 Product Backlog

Para auxiliar na sincronização das atividades entre o *front-end* e o *back-end* foi decidido utilizar os requisitos como *product backlog* assim, uma vez que o *sprint backlog* é definido, o *back-end* pode iniciar o desenvolvimento dos serviços que serão utilizados e enquanto isso o *front-end* inicia o desenvolvimento das telas destes requisitos, de forma

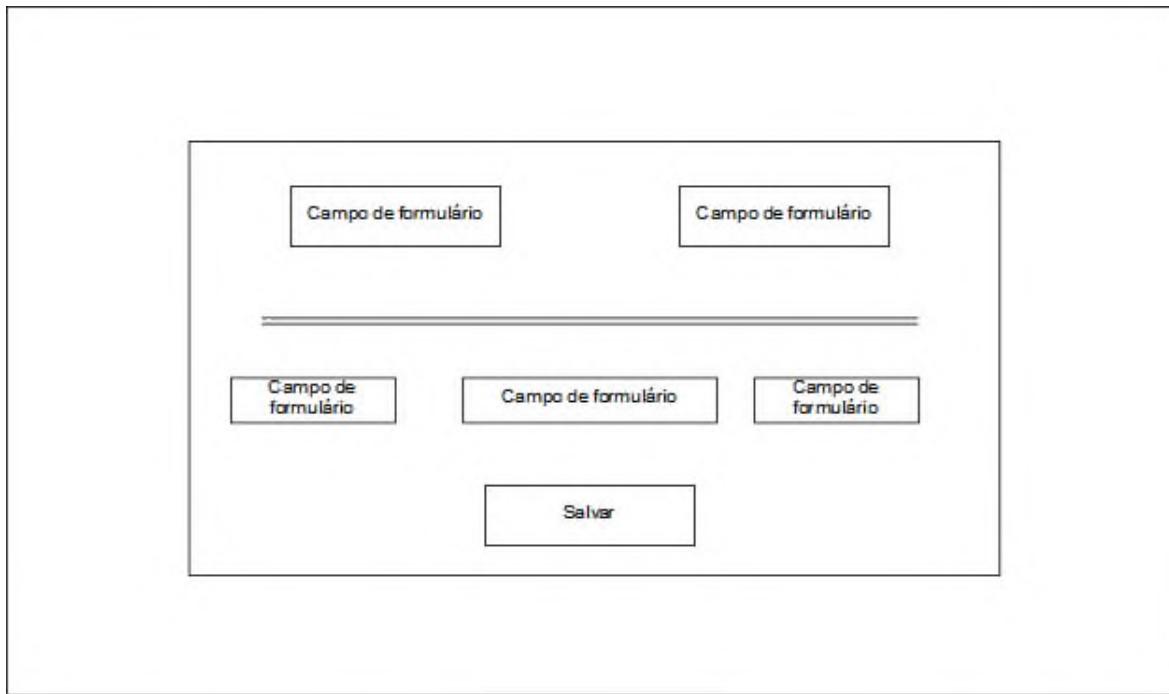


Figura 24 – Esboço da tela completa de período de seleção de disciplinas do administrador.

que na próxima *sprint* o *back-end* cuida da criação dos serviços de outros requisitos e o *front-end* cuida de integração dos serviços com as telas desenvolvidas na *sprint* anterior. Abaixo seguem as atividades que compõem o *product backlog* do *front-end*:

- RF-001 Login;
- RFI-001 Login;
- RF-002 Criar um novo semestre;
- RFI-002 Criar um novo semestre;
- RF-003 Manter Disciplinas;
- RFI-003 Manter Disciplinas;
- RF-004 Manter Docentes;
- RFI-004 Manter Docentes;
- RF-005 Manter Tipos de Atividades;
- RFI-005 Manter Tipos de Atividades;
- RF-006 Manter Atividades Realizadas;
- RFI-006 Manter Atividades Realizadas;

O esboço mostra uma interface web com o seguinte layout:

- Cabeçalho com botões "Adicionar" e "Buscar / Editar".
- Área central intitulada "Lista de cadastros a serem aprovados", que contém uma lista de 6 linhas, cada uma rotulada como "Campo de formulário".
- Área de botões no lado direito com "Reprovar" e "Aprovar".

Figura 25 – Esboço da tela de aprovação de cadastro de professores.

O esboço mostra uma interface web com o seguinte layout:

- Cabeçalho com o texto "Header" e um botão "Sair".
- Área lateral intitulada "Atalhos".
- Área central intitulada "Área de trabalho".

Figura 26 – Esboço da área de trabalho.

- RF-008 Consultar sua posição no ranking do seu grupo;
- RFI-008 Consultar sua posição no ranking do seu grupo;
- RF-009 Listar Disciplinas Preferenciais dos Docentes;
- RFI-009 Listar Disciplinas Preferenciais dos Docentes;

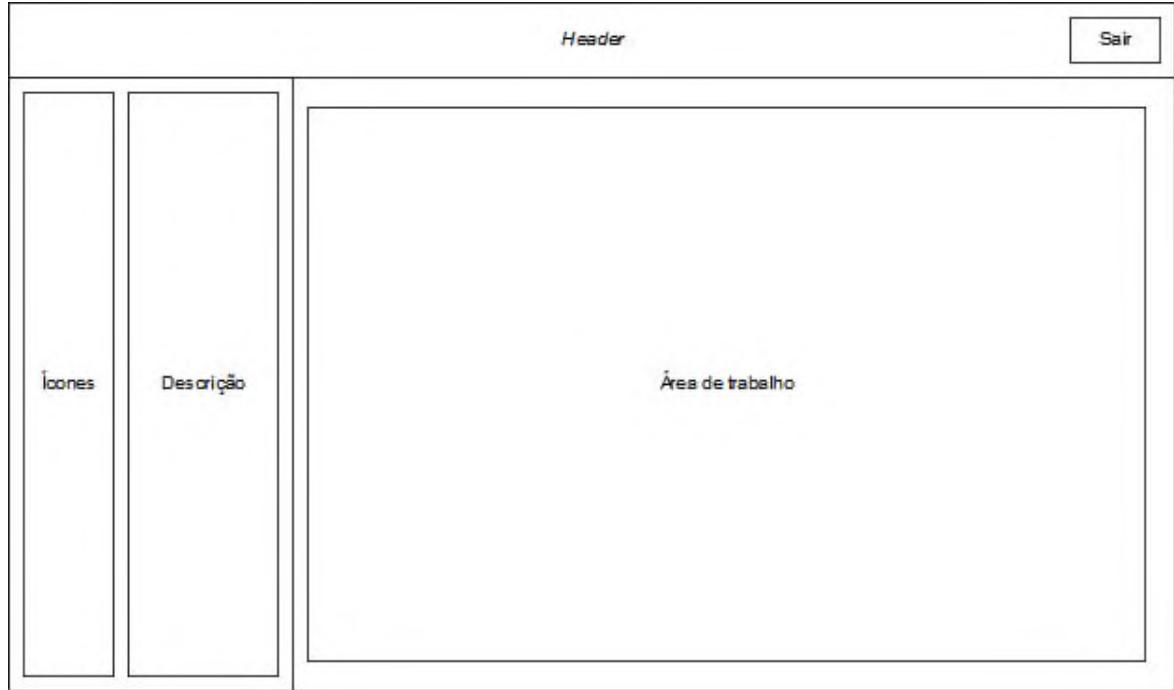


Figura 27 – Esboço da área de trabalho com a barra lateral expandida.

- RF-010 Processar Atribuição de Disciplinas;
- RFI-010 Processar Atribuição de Disciplinas;
- RF-012 Esqueceu a senha;
- RFI-012 Esqueceu a senha;
- RF-013 Alterar a senha;
- RFI-013 Alterar a senha;
- RF-014 Exibir relatório com informações pertinentes;
- RFI-014 Exibir relatório com informações pertinentes.

3.2 Desenvolvimento Scrum

3.2.1 Sprint 1

3.2.1.1 Sprint Planning Meeting

Foi decidido na reunião de *sprint planning* do *sprint 1* que o *sprint backlog* seria composto pelas atividades abaixo. Basicamente as atividades listadas consistem de atividades que tratam do gerenciamento (adicionar, editar, buscar e remover) dos objetos básicos que o sistema possui e que são necessários para o desenvolvimento dos demais requisitos.



Figura 28 – Protótipo da tela de seleção de disciplinas utilizando a técnica *Wizard of Oz*.

- RF-001 *Login*;
- RF-003 Manter Disciplinas;
- RF-004 Manter Docentes;
- RF-005 Manter Tipos de Atividades;
- RF-006 Manter Atividades Realizadas;
- RF-008 Consultar sua posição no ranking do seu grupo;
- RF-012 Esqueceu a senha;
- RF-013 Alterar a senha.

3.2.1.2 Execução

A Figura 30 mostra como foi organizado o diretório do projeto *front-end*. É possível ver que o projeto consiste de um módulo chamado “app”, o qual possui cinco pastas, *components*, que contem todos os componentes utilizados para criação da aplicação, *guards*, que consiste no arquivo de definição do serviço Auth Guard para autenticação, *login*, que consiste do arquivo que contem o model utilizado pelo componente de *login*, *register* que consiste dos arquivos que contem o model utilizado pelos demais componentes para realizar cadastros de novos objetos e a *services* que consiste de todos os serviços e componentes que serão communs a todos os componentes do projeto.

Avaliação de Usabilidade

Você se passará por Yohan Sverikson, professor da Faculdade de Matemática da Universidade Federal de Uberlândia e membro do conselho do curso que realiza reuniões semanais todas as sextas no período integral. Ao final do semestre você precisa executar duas tarefas específicas, cadastrar suas atividades acadêmicas realizadas durante o semestre e elencar por ordem de prioridade as 4 disciplinas/turmas para as quais deseja lecionar no próximo semestre.

Convenientemente, as atividades de Yohan já estão cadastradas, portanto, cabe a você realizar a atividade faltante, elencar as disciplinas que Yohan deseja lecionar no próximo semestre e estabelecer uma ordem de prioridade, já que, pode haver mais de um professor que selecionou uma mesma disciplina para lecionar e também duas disciplinas podem ter seus horários coincidindo. Nesse caso, a disciplina com prioridade maior será a selecionada para Yohan, desde que, não haja um outro professor que a também selecionou e possui um ranking de pontuação maior, nesse caso a segunda disciplina elencada (com prioridade menor) será a escolhida para Yohan.

Durante a atividade de seleção de disciplinas, é necessário que o professor selecione pelo menos 4 disciplinas e um máximo de 5 disciplinas.

Yohan deseja selecionar as seguintes disciplinas:

- Matemática II;
- Geometria Analítica;
- Cálculo I;
- Matemática Discreta;
- Estatística e Probabilidade;

Selecione as disciplinas no sistema de gerenciamento de forma que, a lista de disciplinas selecionadas contenha apenas as matérias listadas acima e cuja ordem de prioridade seja:

- Cálculo I;
- Geometria Analítica;
- Estatística e Probabilidade;
- Matemática Discreta;
- Matemática II;

Para isso siga os passos abaixo:

- 1 – Selecione a opção “Manhã” no filtro Turno;
- 2 – Selecione a opção “Matemática” no filtro Curso;
- 3 – Exclua o dia “Sexta-feira” no filtro de dias da semana já que Yohan não pode lecionar neste dia;
- 4 – Selecione as disciplinas “Matemática II – M02” e “Geometria Analítica – M04”;
- 5 – Selecione a opção “Integral” no filtro Turno;
- 6 – Selecione a disciplina “Matemática Discreta – M10”;
- 7 – Selecione a opção “Engenharia Aeronáutica” no filtro Curso;
- 8 – Selecione as disciplinas “Cálculo I – EA01”;
- 9 – Selecione a opção “Economia” no filtro Curso;
- 10 – Selecione a disciplina “Estatística – ECO2”;
- 11 – Ordene as disciplinas de acordo com a prioridade citada acima;
- 12 – Finalize a seleção.

Figura 29 – Arquivo de orientação para avaliação de protótipo utilizando *Think Aloud*.

No arquivo “app.module.ts” é possível encontrar a declaração de todos os módulos e componentes utilizados no sistema além da hierarquia de rotas do projeto. Já no App Component, de agora em diante ao se referir a um componente subentende-se os quatro arquivos que compõe um componente, em seu template consta apenas o *router-outlet* que é um direcionador para a hierarquia de rotas, ou seja, ao carregar a página o AppComponent é carregado, este por sua vez direciona através do *router-outlet* para a raiz da hierarquia de rotas que é o componente HomeComponent. HomeComponent se encontra dentro da pasta *components* como é mostrado na Figura 30, além de outras 4 pastas, *admin*, que consiste de todos os componentes utilizados para criar a seção acessível pelo

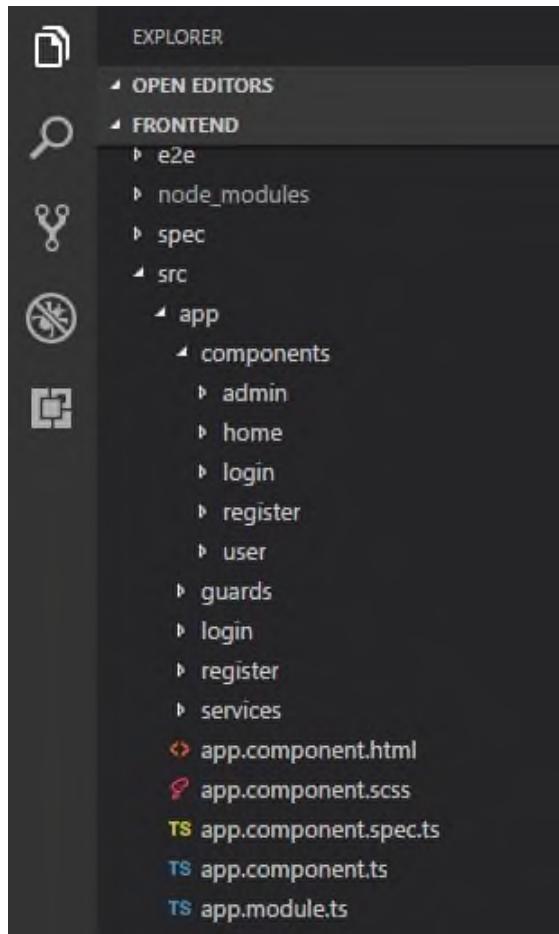


Figura 30 – Diretório do projeto

login de administrador, *login*, que consiste no componente utilizado pelo HomeComponent para realizar o *login*, *register* que consiste no componente utilizado pelo HomeComponent para realizar o cadastro de novos usuários e o *user*, que consiste de todos os componentes utilizados para criar a seção acessível pelo *login* de um professor.

Uma vez explicada a organização dos arquivos será percorrido agora atividade por atividade do backlog realizadas na *sprint 1*.

Como os requisitos dessa *sprint* consistiam apenas da criação das telas as preocupações estavam voltadas para a utilização correta e harmoniosa de cores, linguagem e símbolos, além da preocupação de deixar claro para o usuário a função de cada elemento (*affordance*). Na tela de *login* por exemplo, todos os ícones clicáveis transformavam o cursor do *mouse* para ponteiro e/ou possuía algum outro comportamento que deixava claro que alguma ação poderia ser ali realizada. Como a ideia era deixar o cadastro e o *login* na mesma tela, foi utilizado um componente pronto do Angular Material chamado *Tabs*, que possui uma estrutura que se altera de acordo com a aba ativa, assim foi inserido dentro de cada aba o componente de *login* e o componente de cadastro.

Uma vez passada a tela do HomeComponent foi criada uma forma simples de

navegação entre as várias seções que o projeto necessita, através de um menu lateral. Foram mantidas agrupadas ações referentes a um mesmo objeto ou assunto dentro de uma mesma seção. Isto proporcionou preservar o conceito de menus de contexto, característica que contribui para a intuitividade de um usuário para com um sistema.

A estrutura da seção Disciplinas consiste de uma *Tabs* do Angular Material a qual possui duas abas, uma para adição e outra para busca e edição de disciplinas. Na aba adição existe apenas um formulário com os campos necessários para o cadastro de uma nova disciplina, uma vez que os campos foram preenchidos ao clicar no botão de finalização. Ao final do template uma requisição é enviada ao *endpoint* “subjects/register”.

Na aba de busca foi necessário inserir um componente que permitisse a visualização de uma grande quantidade de dados sem que o usuário se perdesse ou o conteúdo se tornasse incompreensível. Foi utilizado para suprir essa necessidade uma *Table*, um *Paginator* e um *Sort Header* também do Angular Material. A *Table* fica responsável pela disposição dos elementos, o *Paginator* cuida para que caso haja um quantidade excessiva de itens, a tabela não cresça infinitamente e o *Sort Header* faz com que os dados da tabela possam ser filtrados a partir das informações inseridas no campo de entrada. Uma vez aberta a aba de busca, é feita uma requisição ao *endpoint* “/subjects/fetchall” que retorna toda a lista de disciplinas do banco, informação que servirá para popular a tabela.

A tela de edição é acessada através do clique em um ícone que aparece ao final da tabela, o conteúdo da linha que será editado é enviado para a janela que será aberta. Uma vez que qualquer campo tenha seu conteúdo alterado e o botão “Salvar Edição” seja clicado, uma requisição é enviada para o *endpoint* “/subjects/update” para que a informação seja atualizada no banco de dados.

As demais telas das atividades deste backlog funcionam basicamente da mesma maneira como foi explicada a seção de disciplinas, exceto a tela da seção Professores pois nesta consta uma peculiaridade. Como o cadastro de professores não é feito pelo administrador, ele apenas aprova ou reprova os cadastros, ao invés da aba de adição ter um formulário para ser preenchido, nesta aba há um campo no qual após a inicialização do componente é feita uma requisição ao *endpoint* “/users/pendentes” que retorna todos os cadastros pendentes de aprovação. Cada cadastro configura um cartão neste campo que ao ser clicado exibe as informações do cadastro correspondente no formulário contido à direita, para que as informações possam ser visualizadas e assim o cadastro aprovado ou reprovado.

A atividade de alterar senha não teve nenhum conceito ou especificidade aplicado, basta a inserção da senha antiga, da nova senha e clicar no botão “Alterar” para que seja enviada uma requisição para o *endpoint* “/users/changepassword” para que ela seja alterada.

3.2.1.3 Sprint Review

Durante a realização da *sprint* pode ser observado que o refinamento das telas foi gradual e bastante frequente à medida que o modelo mental teve de ser também retrabalhado por causa de interações mal projetadas ou por falhas de modelagem dos objetos. Assim, espera-se que um gasto maior com o projeto das interações e da modelagem possa economizar no tempo de desenvolvimento das telas futuras. Pelo lado positivo, pode-se notar que mantendo um padrão para as atividades semelhantes houve tanto um ganho de agilidade no desenvolvimento quanto facilidade de utilização para ao usuário. A entrega da *sprint* é mostrada nas imagens 31, 32, 33, 34, 35, 36, 37, 38 e 39.

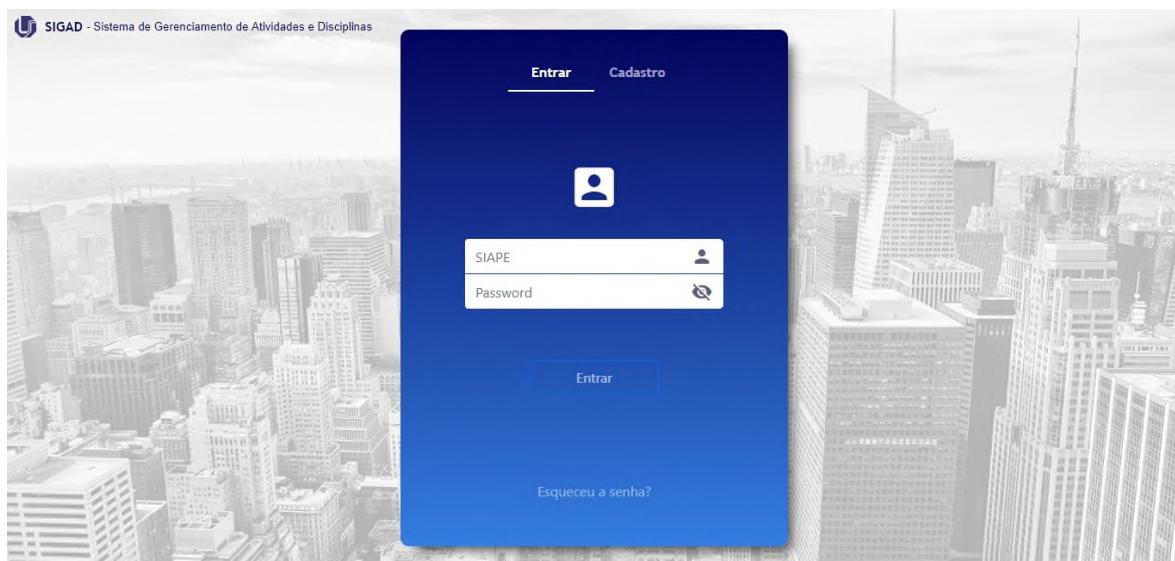


Figura 31 – Template da tela de *login*

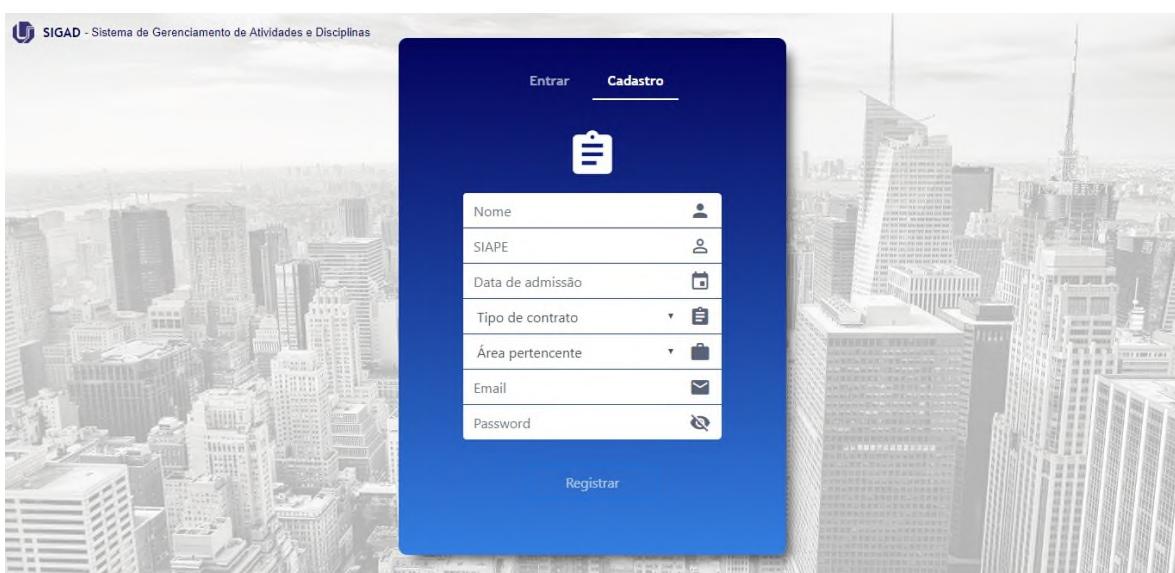


Figura 32 – Template da tela de *cadastro*



Figura 33 – Template da area de trabalho com barra lateral fechada



Figura 34 – Template da area de trabalho com barra lateral expandida

3.2.2 Sprint 2

3.2.2.1 Sprint Planning Meeting

Foi decidido na reunião de *sprint planning* do *sprint 2* que o *sprint backlog* seria composto pelas seguintes atividades:

- RFI-001 Login;
- RFI-003 Manter Disciplinas;
- RFI-004 Manter Docentes;

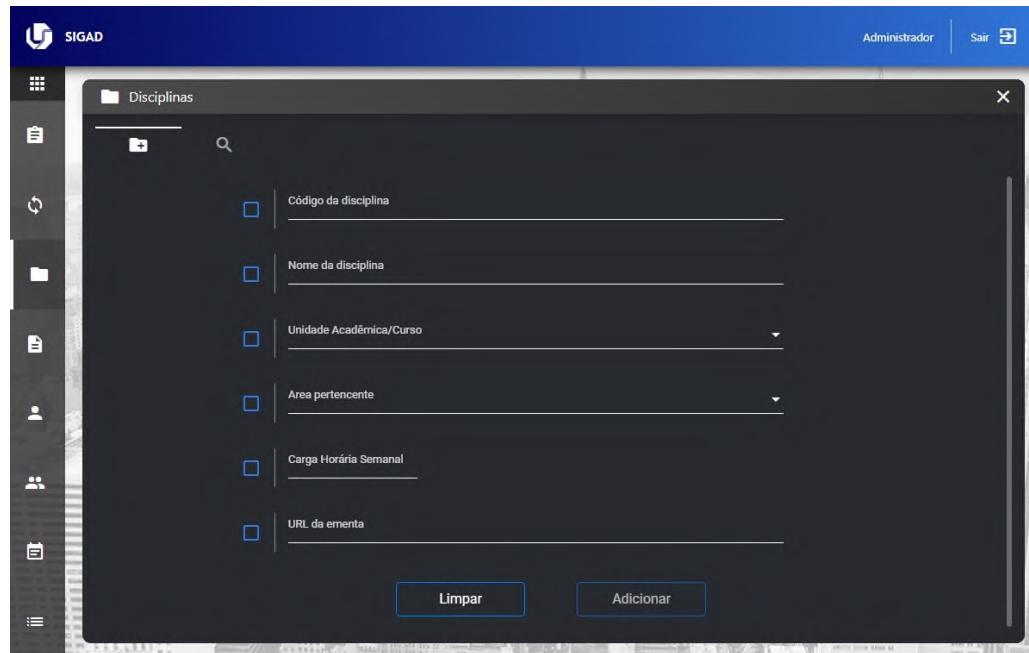


Figura 35 – Template da tela de cadastro de disciplinas

Código	Nome	Área	Unidade acadêmica	Ações
ADM01	Calculo I	Matemática	FAGEN - Administração - Integral	
ADN01	Calculo I	Matemática	FAGEN - Administração - Noturno	
ADM03	Estatística	Estatística	FAGEN - Administração - Integral	
BCC01	Geometria Analítica	Matemática	FACON - Ciência da Computação - Integral	
BCC03	Estatística	Estatística	FACON - Ciência da Computação - Integral	

Figura 36 – Template da tela de busca de disciplinas

- RFI-005 Manter Tipos de Atividades;
- RFI-006 Manter Atividades Realizadas;
- RF-002 Criar um novo semestre;
- RF-009 Listar Disciplinas Preferenciais dos Docentes;
- RF-010 Processar Atribuição de Disciplinas.

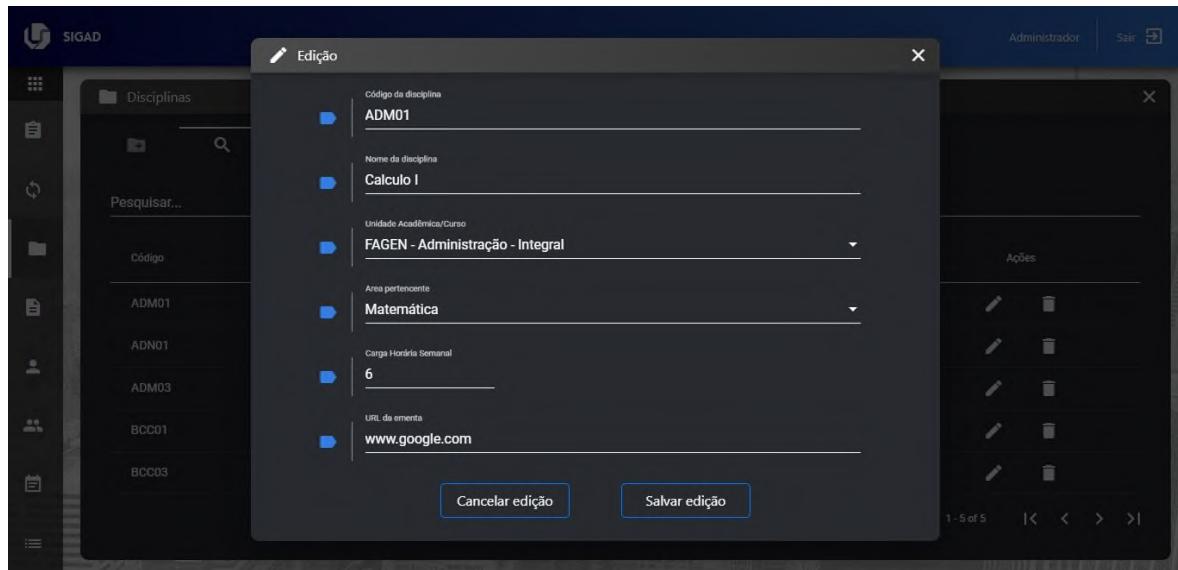


Figura 37 – Template da tela de edição de disciplinas

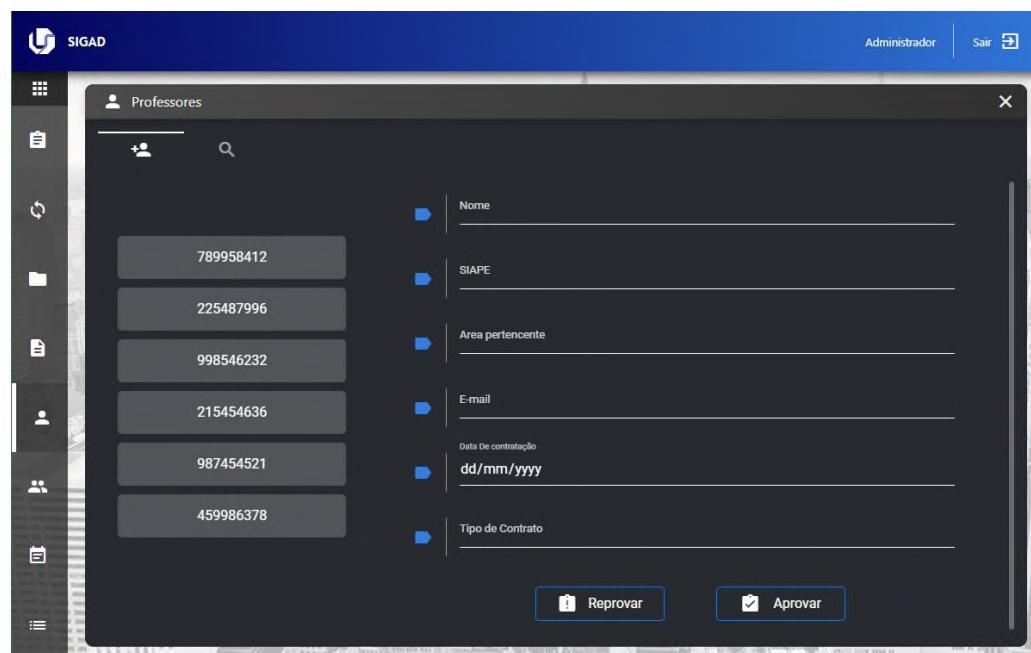


Figura 38 – Template da tela de aprovação de cadastro de professores

Uma vez que os serviços para a maioria dos requisitos acima já foram criados é possível agora realizar a integração do *front-end* com os serviços do *back-end* e assim tornar essas seções funcionais. Além disso, também entra no backlog da *sprint 2* a criação da tela da seção “Novo Semestre”, “Seleção de disciplinas” e “Registro de Atividades”.

3.2.2.2 Execução

Antes de iniciar a integração com os serviços, é necessário ter em mente que toda e qualquer requisição realizada pelo *front-end* que seja feita através de um método *post*,

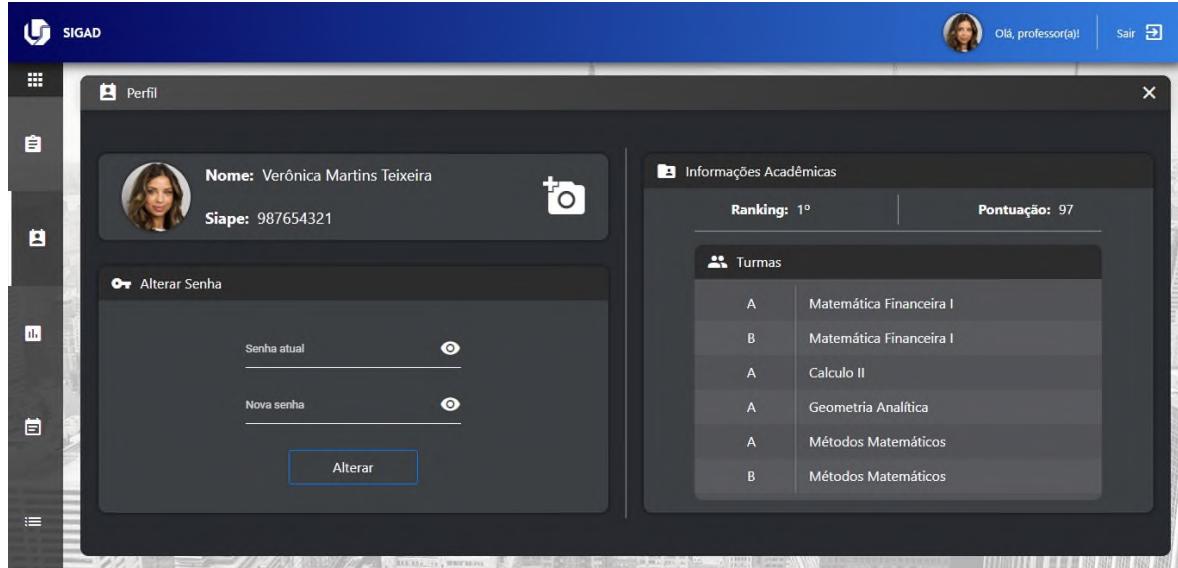


Figura 39 – Template da tela de alteração de senha e consulta do ranking

envia as informações como um objeto Javascript Object Notation (JSON). No requisito de *login* a integração foi bem simples, ao clicar no botão “Entrar” uma requisição é feita através do método “submit” do LoginComponent para o *endpoint* “/users/authenticate”, onde são enviados os campos siape e senha. Caso a requisição retorne sucesso, a função faz uma nova requisição, porém, agora enviando o campo siape para o *endpoint* “/users/bysiape” que retornará o cadastro desse usuário e direciona o usuário para o tipo de visão de acordo com a permissão que a propriedade admin determina, o método submit é mostrado na Figura 40.

```
Submit(){
  this.validateService.authenticateUser(this.credentialsData).subscribe(data => {
    if(data.success){
      this.validateService.storeUserData(data.token,data.user);
      var siape = this.credentialsData.siape;
      this.validateService.getUserBySiape(siape).subscribe(data=>{
        if(data[0].admin){
          this.router.navigate(['admin']);
        }
        else{
          this.router.navigate(['user']);
        }
      });
    }
    else{
      this.customModalService.callModal(data.msg);
      this.router.navigate(['login']);
    }
  });
}
```

Figura 40 – Método utilizado pela tela de *login*

A seguir será explicado como foi feita a integração das funções de adicionar, buscar, editar e remover para o objeto disciplina. Para os demais objetos, atividades, tipos de atividades, professores e turmas o processo é o mesmo, se diferenciando apenas no *payload* das requisições e nos *endpoints*.

A atividade de adição nada mais é do que um simples formulário que recolhe as informações dos campos e atribui para um objeto modelo de disciplina, e que ao clicar no botão “Adicionar” faz uma requisição através do método “submit” do SubjectComponent para o *endpoint* “/subjects/register” que receberá os dados do modelo de disciplina a ser cadastrado. Caso essa requisição retorne sucesso, o modelo de objeto é limpo, a lista de disciplinas é atualizada e um modal aparece informando o sucesso da operação. Em caso de falha um modal surge informando que não possível realizar a operação. Um tratamento de erro mais detalhado deve ser adicionado posteriormente. A Figura 41 mastra o método utilizado nesse cadastro.

```
submit() {
    this.loading = true;
    this.searchBarValue = '';
    this.cleanEdit();
    this.validateService.registerSubjects(this.subjectData).subscribe(data => {
        if (data.success) {
            this.subjectData = new SubjectViewModel({});
            this.updateList();
            this.loading = false;
            this.customModalService.callModal("Cadastro efetuado com sucesso!");
            this.validateService.getSubjects().subscribe(
                data => {
                    this.fullSubjectList = this.searchedSubjectList = data;
                    this.searchBarValue = '';
                }
            );
        } else {
            this.loading = false;
            this.customModalService.callModal("Registro falhou");
        }
    });
}
```

Figura 41 – Método utilizado para cadastrar disciplinas

Na busca de objetos do tipo disciplina, na inicialização do componente é feita uma requisição através do método “updateList” para o *endpoint* “/subjects/fetchall” o qual retornará a lista de disciplinas, lista que populará a tabela. Caso uma pesquisa nessa tabela seja feita, o conteúdo buscado é atribuído a uma propriedade da tabela chamada filter que realizará a filtragem dos dados.

A remoção de disciplinas é feita ao clicar no ícone “lixo” em uma das linhas da tabela. Uma requisição é enviada através do método “deleteSubject” para o *endpoint*

“/subjects/delete” o qual receberá o código da disciplina a ser deletada e logo em seguida é feito o processo de atualização da lista de disciplinas.

A edição de disciplinas é feita ao clicar no ícone “caneta” em uma das linhas da tabela. Uma janela de edição é aberta carregando os dados da linha correspondente, caso o botão “Salvar Edição” seja clicado, uma requisição é enviada através do método “update” do EditSubject para o endpoint “/subjects/update” e a lista de disciplinas é atualizada.

Quanto às novas telas criadas o processo realizado foi o seguinte: para criar um novo semestre foi criada uma nova seção pois o modo inicial como foi concebida a ideia seria deixar esta funcionalidade junto com a criação de turmas, porém, a chance de causar desvios no modelo mental desejado seria muito grande. Assim, foi criada a tela de iniciar novo semestre, a qual possui inicialmente dois campos para inserção dos dados do semestre a ser criado e para evitar novos desvios no modelo mental, caso o administrador queira reaproveitar a criação das turmas de um semestre anterior basta que ele selecione a caixa de seleção com a descrição de clonagem para que novos campos apareçam e ele possa inserir os dados sobre o semestre que terá suas turmas clonadas.

Para ser possível o desenvolvimento dos requisitos RF-009 e RF-010 foi necessário criar as telas “Seleção de Disciplina” e “Registro de Atividade” do administrador e “Seleção de Disciplina” do professor.

Na tela de “Seleção de Disciplina” do administrador, inicialmente são visualizados dois campos, que ao serem preenchidos definem para qual ano e semestre as alterações no período de disciplina valerá. Uma vez preenchidos, a tela mostrará as demais opções de configurações para o período de seleção e logo ao final da tela consta o botão de finalização. Já na tela de “Registro de Atividades” do administrador, só é possível atribuir atualizações ao objeto de seleção para o último ano existente no banco assim, na parte inicial da interface há, em destaque, o ano para o qual as alterações serão atribuídas. Logo abaixo se encontram as demais configurações possíveis para o período.

Na tela de “Seleção de Disciplina” do professor na parte superior constam os campos que permitem filtrar as turmas disponíveis. Logo abaixo há um seção dividida em três partes, a primeira à esquerda consta a lista de turmas disponíveis para seleção, as quais caso ocorra o evento de *hover*, uma dica acima do cartão da turma será exibido, informando que para visualizar a ementa desta disciplina basta clicar no corpo do cartão. Na segunda seção consta uma matriz de horários em que o professor consegue acompanhar tanto as turmas já selecionadas quanto os horários em que elas devem ser lecionadas e na terceira seção há uma lista de turmas já selecionadas pelo professor e ordenadas por sua prioridade.

3.2.2.3 Sprint Review

Durante a realização foi observado um ganho significativo em termos de tempo de desenvolvimento devido a um processo de modelagem e projeto de interação mais detalhado. Ainda assim, ocorreram alguns casos em que tiveram de ser feitas modificações nos layouts para se adaptar à futura integração com os serviços do *back-end*. A entrega da sprint é mostrada nas imagens 42, 43, 44, 45, 46 e 47;

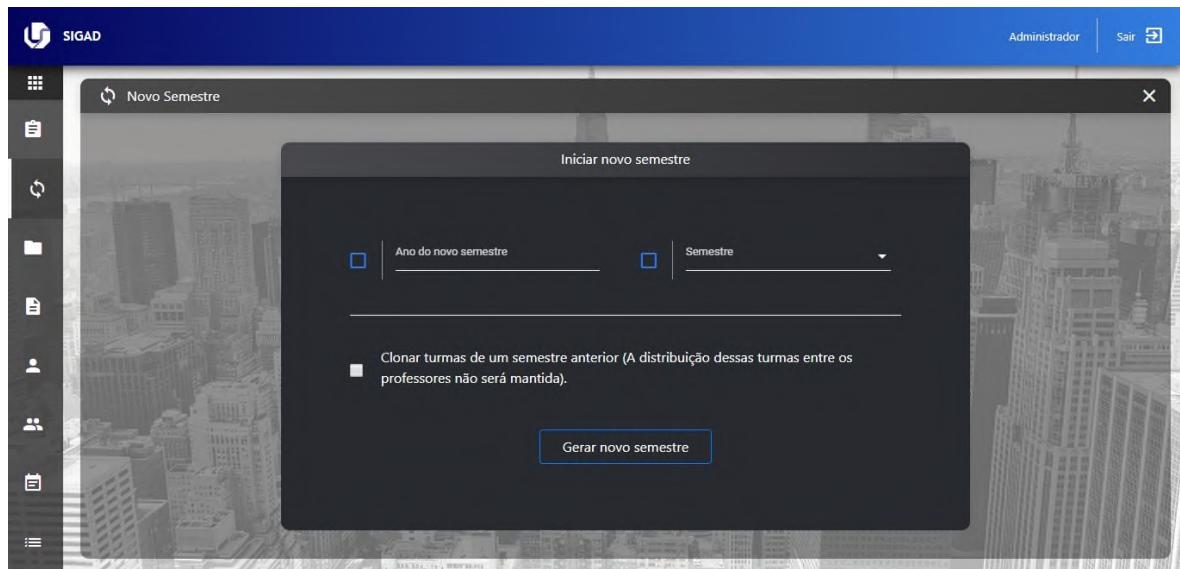


Figura 42 – Template da tela de criação de novo semestre simples

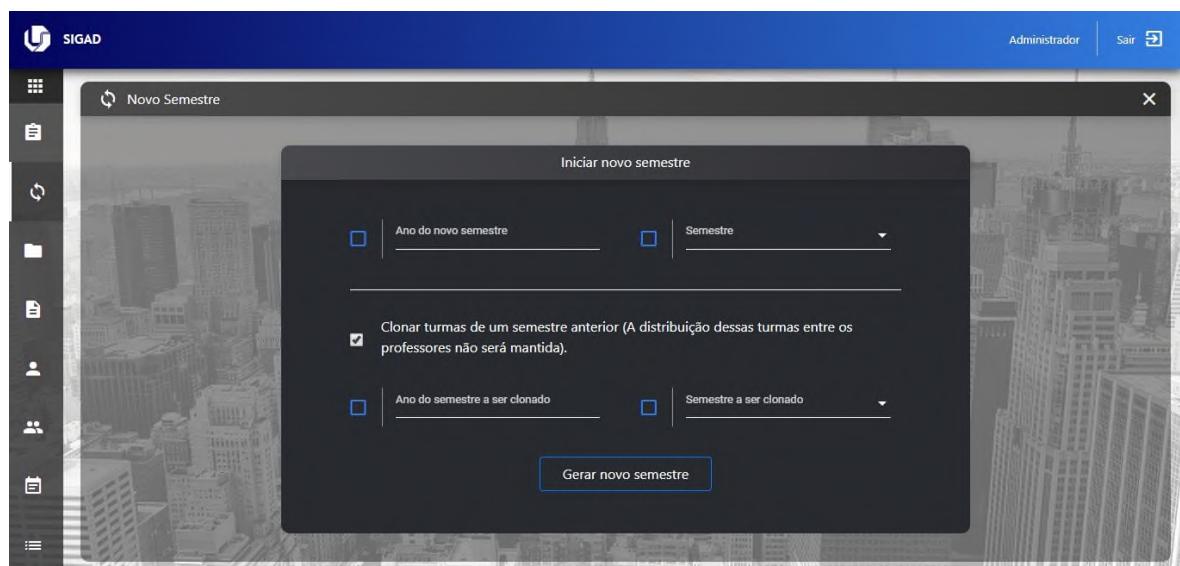


Figura 43 – Template da tela de criação de novo semestre completa

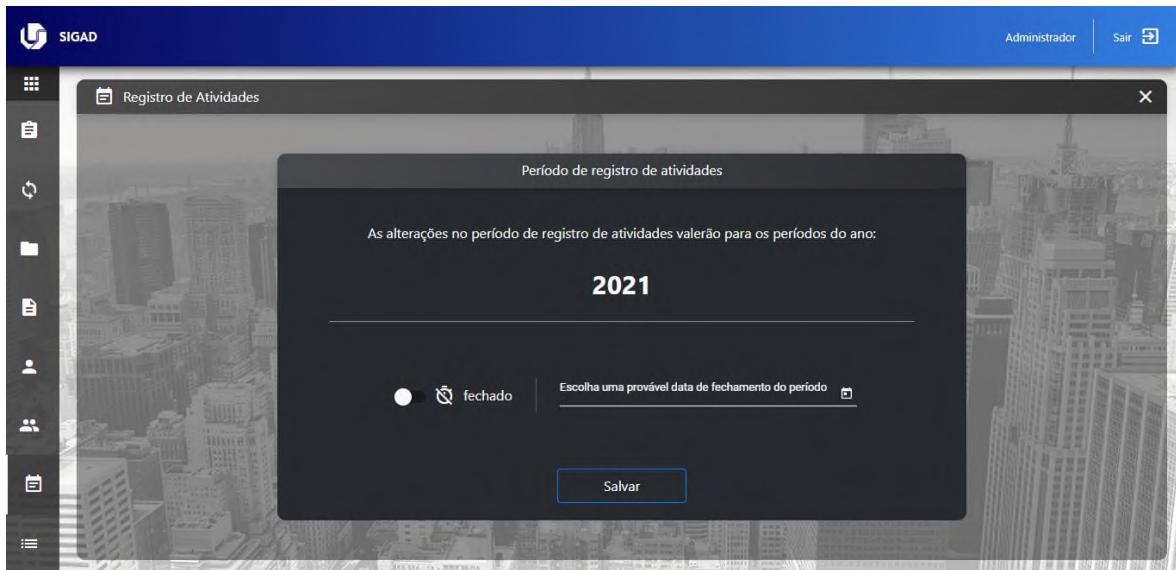


Figura 44 – Template da tela de registro de disciplinas do administrador

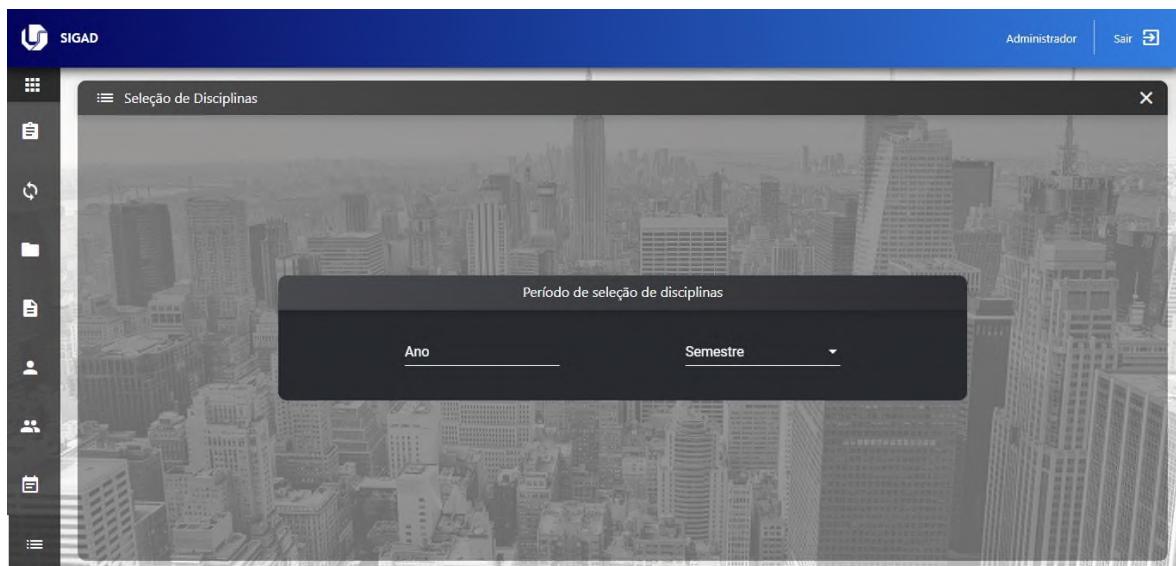


Figura 45 – Template da tela de seleção de disciplinas do administrador simples

3.2.3 Sprint 3

3.2.3.1 Sprint Planning Meeting

Foi decidido na reunião de *sprint planning* do *sprint 3* que o *sprint backlog* seria composto pelas seguintes atividades:

- RFI-002 Criar um novo semestre;
- RFI-009 Listar Disciplinas Preferenciais dos Docentes;
- RFI-010 Processar Atribuição de Disciplinas;

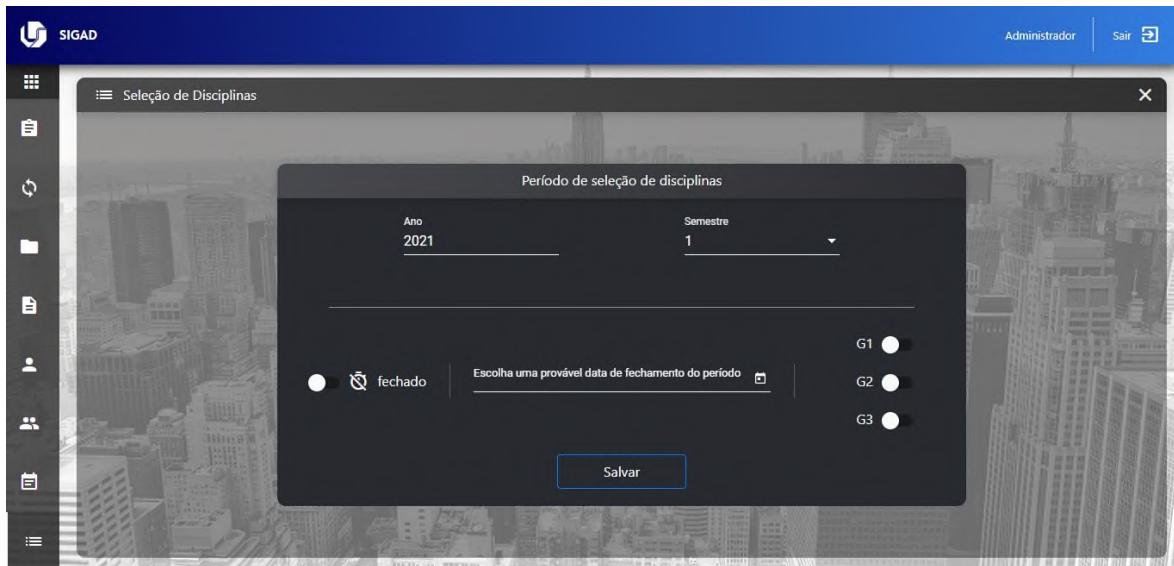


Figura 46 – Template da tela de seleção de disciplinas do administrador completa

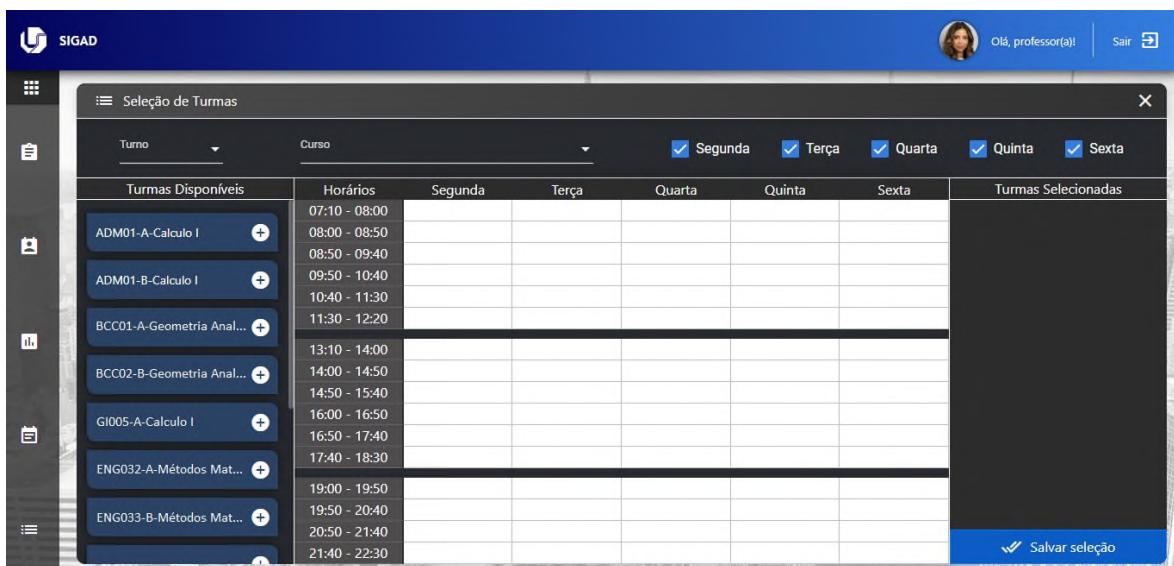


Figura 47 – Template da tela de seleção de disciplinas do professor

3.2.3.2 Execução

Para deixar a seção “Novo Semestre” funcional, ao ser clicado o botão “Gerar novo semestre” é feita uma requisição através do método “generate” do NewCicleComponent para um *endpoint* que varia de acordo com as configurações no novo semestre. Inicialmente é instanciado um objeto modelo de uma distribuição, seus atributos *activity*, *class* e *selectionPeriod* são zerados e a partir daí são algumas verificações para definir qual caminho o algoritmo deverá seguir. Se a variável de clonagem estiver com o conteúdo igual a verdadeiro e os campos que guardam as informações de um possível ano e semestre a ser clonado não estão vazios então é feita uma requisição para o *endpoint* “/distributions/duplicate” o que significa que o administrador quer instanciar um novo semestre a

partir da clonagem de um outro semestre. Caso contrário, será enviada uma requisição para o *endpoint* “/distributions/distinctyears” que retornará os anos de distribuições presentes no banco. Caso o ano do semestre a ser inserido já existe no banco é retornado para o administrador que foi criado um novo semestre porém, nenhuma requisição é feita para o *back-end*. Mas caso essa ano não exista ainda no banco uma requisição é feita para o *endpoint* “/distributions/register” enviando apenas o modelo de objeto de distribuição recem instanciado. A função que coordena esse processo é mostrada na Figura 48;

```
generate(){
    this.distribution.activity = this.distribution.activity ? this.distribution.activity : [];
    this.distribution.class = this.distribution.class ? this.distribution.class : [];
    this.distribution.selectionPeriod = this.distribution.selectionPeriod ? this.distribution.selectionPeriod : [];
    if(this.activateCloning && this.distributionToClone.year && this.distributionToClone.semester){
        this.validateService.duplicateDistribution(this.distribution, this.distributionToClone).subscribe(x => {
            if(x.success == false){
                this.customModalService.callModal('Problema ao clonar semestre');
            } else{
                this.customModalService.callModal('Semestre clonado com sucesso');
            }
        });
    } else{
        this.validateService.getYearRange().subscribe(
            data=>{
                if(data.indexOf(Number(this.distribution.year)) != -1){
                    this.customModalService.callModal('Semestre cadastrado com sucesso');
                } else{
                    this.validateService.newDistribution(this.distribution).subscribe( x => {
                        if( x.success == false){
                            this.customModalService.callModal('Problema ao cadastrar semestre');
                        } else{
                            this.customModalService.callModal('Semestre cadastrado com sucesso');
                        }
                    });
                }
            }
        );
    }
}
```

Figura 48 – Método utilizado pela tela Iniciar novo semestre

Na tela de “Seleção de disciplinas” do administrador o funcionalidade é bem simples, primeiro após ser digitado os campos de ano e semestre, é feita uma busca banco para trazer as informações do período de seleção deste ano e semestre. Assim, é feito uma requisição para o *endpoint* “/selectionperiods/selection”. Caso a requisição retorne um conteúdo os campos de configuração são atualizados de acordo com o estado do objeto retornado, caso contrário é aberta uma janela de dialogo informando que não existe distribuição para tal ano e semestre. Após feita a devida configuração periodo de seleção ao se clicar em “Salvar” é feita uma requisição através do método “save” para o *endpoint* “/selectionperiods/update” o qual realizará a atualização do objeto no banco.

O mesmo ocorre com a tela de “Registro de Atividades” do administrador. No momento de sua inicialização é feita uma requisição para pegar o último ano de distribuição do banco de dados e logo em seguida é feito uma requisição para o *endpoint* “/selectionperiods/register” enviando esse último ano retornado pela requisição anterior,

é retornado portanto o objeto de seleção de disciplina correspondente ao registro de atividade do último ano de distribuição que consta no banco. A partir dessa informações os campos de configurações são atualizados. Após qualquer alteração, caso “Salvar” seja clicado é enviada uma requisição para o *endpoint* “/selectionperiods/update” contendo as alterações no objeto de seleção de disciplina que deverá ser salvo no banco.

Já na tela de “Seleção de Disciplinas”, na inicialização é feita uma verificação para saber se o período de seleção de disciplinas está aberto. Essa verificação é feita através de uma requisição para o *endpoint* “/distributions/distinctyears” que retorna a lista de anos de distribuições do banco, com essa informação é feita uma nova requisição para buscar o objeto de seleção de disciplinas desse ano e por fim é possível verificar se o período está aberto ou não. Caso esteja fechado é aberta uma janela de dialogo informando a impossibilidade de realizar a seleção de disciplinas. Caso contrário é feita duas requisições para obter as turmas disponíveis para seleção. A primeira é uma requisição feita para o *endpoint* “/subjects/fetchall” que retorna todas as disciplinas do banco. A segunda requisição é feita para o *endpoint* “/classes/fetchall” enviando o maior ano de distribuição presente no banco, assim, são retornadas todas as turmas pertencentes à distribuição mais atual do banco.

Uma vez finalizadas as duas requisições, a lista de turmas é passada por uma função chamada “generateCharacteristics”, que atribui características a cada turma que serão utilizadas para realizar a filtragem posteriormente. Basicamente, essa função percorre o atributo *schedule* e verifica os dias da semana e os turnos em que a turma em questão possui horários. Após este processo ter sido finalizado, a coluna de turmas disponíveis será populada e uma vez clicada no corpo do cartão da turma o usuário é direcionado à pagina em que consta a ementa da disciplina. Caso ele clique no ícone “+” do cartão, uma função chamada “selectCourse” é ativada. Esta função atualiza a lista de turmas selecionadas mantidas no *front-end* e também atualiza a matriz de horários preenchendo nos horários correspondentes ao da turma o código da disciplina e o nome da turma além disso, ela chama uma outra função chamada “updateWorkspace” que insere o cartão desta turma no *gridster*.

Uma vez finalizada a seleção das disciplinas o professor pode salvar sua seleção ao clicar no boão “Salvar seleção”, que dispara a execução de uma função chamada “confirmClasses”. Esta função popula uma lista de turmas que são adicionadas de acordo com a ordem de prioridade estabelecida e em seguida envia uma requisição para o *endpoint* “/users/chooseclass” contendo o objeto *user* atualizado com as turmas selecionadas. As funções citadas que são utilizadas no processo de seleção de disciplinas são mostradas nas Figuras 49, 50, 51 e 52.

```

generateCacteristics(aula: ClassModel) {
    var period = [];
    var days = [];
    for (var i = 0; i < aula.schedule.length; i++) {
        for (var j = 0; j < aula.schedule[i].length; j++) {
            if (aula.schedule[i][j]) {
                if (i < 6) {
                    if (period.indexOf('morning') === -1) {
                        period.push('morning');
                    }
                }
                if (i < 12 && i > 5) {
                    if (period.indexOf('afternoon') === -1) {
                        period.push('afternoon');
                    }
                }
                if (i > 12) {
                    if (period.indexOf('night') === -1) {
                        period.push('night');
                    }
                }
            }
        }
    }
}

```

Figura 49 – Método utilizado para gerar as características das turmas

```

selectCourse(course: ListElement){
    for(let i = 0; i < course.label.schedule.length; i++){
        for(let j = 0; j < course.label.schedule[i].length; j++){
            if(course.label.schedule[i][j]){
                this.selectHour(`#${i}-#${j}`, course);
            }
        }
    }
    this.addedList.push(course);
    this.updateWorkspace();
    this.filter();
}

```

Figura 50 – Método utilizado para atualizar a lista de turmas selecionadas

```

updateWorkspace(){
    this.dashboard = [];
    this.addedList.forEach(x => {
        this.dashboard.push({x: 0, y: 0, cols: 0, rows: 0, item: x});
    });
}

```

Figura 51 – Método utilizado para inserir cartões de turmas selecionadas no *gridster*

3.2.3.3 Sprint Review

A finalização deste sprint possibilitou o funcionamento do processo de seleção de disciplinas. Certas adaptações foram necessárias como por exemplo a utilização de um modelo de usuário estático para se fazer requisições que necessitam de um usuário, pois o desenvolvimento de sessões ainda não foi realizado.

```
confirmClasses(){
    this.dashboard;
    let orderedDashboard = [];
    for(let i=0;i< this.dashboard.length;i++){
        if(this.dashboard.find(x => x.y == i)){
            orderedDashboard.push(this.dashboard.find(x => x.y == i))
        }
    }
    orderedDashboard.forEach(x => {
        this.validateService.editUserClass(
            {year: this.biggestYear, siap:'987654321', class:x.ite.label}).subscribe(
            x => {
                console.log(x);
            }
        );
    });
}
```

Figura 52 – Método utilizado para atualizar no banco as turmas selecionadas

4 Conclusão

O objetivo do trabalho foi concluído uma vez que foi construída uma aplicação utilizando as ferramentas definidas no início do projeto. Quanto ao objetivo de utilização do sistema, é possível dizer que foi concluído mesmo que parcialmente, pois houve de fato a digitalização do processo de seleção de disciplinas e gerenciamento de atividades, ainda que algumas partes exijam um refinamento. Processos manuais que antes demandavam uma grande quantidade de tempo como o levantamento das disciplinas disponíveis para a seleção em cada semestre agora podem ser copiadas de um semestre para outro e apenas feitos pequenos ajustes específicos, caso necessário. Todas as estruturas de distribuição podem ser armazenadas em um banco de dados, junto com informações como cadastro de professores, atividade realizadas, disciplinas lecionadas etc.

Além disso, a utilização do framework PACT possibilitou identificar oportunidades de tornar o sistema ainda mais sensível, harmonioso e útil para o problema de distribuição de disciplinas da FAMAT, tornando-o muito mais amigável e fácil de ser utilizado. A utilização do framework Angular possibilitou uma grande economia no que tange tempo de desenvolvimento e qualidade dos layouts. Apesar da construção de todas as funcionalidades essenciais ainda restam algumas atividades a serem desenvolvidas e são indicadas para trabalhos futuros como as que seguem:

- Implementação do requisito: RF-008 Consultar sua posição no ranking do seu grupo;
- Implementação do requisito: RFI-013 Alterar a senha;
- Implementação do requisito: RFI-012 Esqueceu a senha;
- Implementação do requisito: RF-014 Exibir relatório com informações pertinentes.
- Implementação da manutenção: Remover classes CSS não utilizadas para melhorar a velocidade de carregamento;
- Implementação da manutenção: Reorganizar os componentes dentro das pastas *admin* e *user* para módulos *admin* e *user*;
- Implementar um campo para observações nas turmas;
- Adicionar tratamentos de erros específicos;
- Criar o objeto Curso no banco para ser buscado sempre que necessário;
- Adicionar notificações para os professores quando algum dos períodos for aberto ou fechado.
- Realizar a implantação do sistema.

Referências

- AFONSO, A. *O que é Angular?* 2018. <<https://blog.algaworks.com/o-que-e-angular/>>. Citado na página 18.
- BARROS, P. *Imperativo ou Declarativo?* 2016. <<https://medium.com/opensanca/imperativo-ou-declarativo-3e6dffbf301c>>. Citado na página 19.
- BECK, K. et al. *Manifest for Agile Software Development.* 2001. <<http://agilemanifesto.org/>>. Citado na página 15.
- BENITTE, R.; GREIF, S.; RAMBEAU, M. *The state of Javascript 2017.* 2017. <<https://2017.stateofjs.com/2017/front-end/results>>. Citado na página 18.
- BENYON, D.; TURNER, P.; TURNER, S. *Designing Interactive Systems: People, Activities, Contexts, Technologies.* [S.l.]: Essex, England: Pearson Education Limited, 2005. Citado 3 vezes nas páginas 26, 27 e 29.
- BRASIL, U. *React: o que é e como funciona essa ferramenta?* 2018. <<https://tableless.com.br/react-o-que-e-e-como-funciona-essa-ferramenta/>>. Citado na página 19.
- CAHÚ, G. *Técnicas de Prototipação de Software.* 2016. <<https://medium.com/@guilhermecahu/t%C3%A9cnicas-de-prototipa%C3%A7~ao-para-artefatos-digitais-44a5f183f2d6>>. Citado na página 31.
- CARD, S. K.; MORAN, T. P.; NEWELL, A. *The psychology of human-computer interaction.* [S.l.]: Lawrence Erlbaum Associates, 1983. Citado na página 26.
- DINIZ, S.; SANTANA, B. *Interação Humano-Computador.* [S.l.]: Elsevier Editora Ltda., 2010. Citado 2 vezes nas páginas 5 e 30.
- DIONÍSIO, E. J. *Introdução ao TypeScript.* 2016. <<https://www.devmedia.com.br/introducao-ao-typescript/36729>>. Citado na página 19.
- DIX, A. et al. *Human-Computer Interaction.* [S.l.]: Pearson Education Limited, 2003. Citado na página 31.
- FILHO, A. M. d. S. *Introdução ao TypeScript.* 2008. <<https://www.devmedia.com.br/arquitetura-de-software-desenvolvimento-orientado-para-arquitetura/8033>>. Citado na página 17.
- HAVIV, A. Q. *MEAN Web Development.* [S.l.]: Packt Publishing Ltd, 2016. Citado na página 17.
- LEITE, A. F. *Metodologia de desenvolvimento de Software.* 2006. <<https://www.devmedia.com.br/metodologia-de-desenvolvimento-de-software/1903>>. Citado na página 14.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição.* [S.l.]: McGraw Hill Brasil, 2016. Citado na página 14.

- REINIUS, J. *The PACT Analysis Framework*. 2011. Tese (Master of Science), Blekinge Institute of Technology, Karlskrona, Sweden. Citado 2 vezes nas páginas [27](#) e [28](#).
- SBC. *Interação Humano Computador*. 2019. <<http://www.sbc.org.br/14-comissoes/390-interacao-humano-computador>>. Citado na página 25.
- YOU, E. *O Framework Javascript Progressivo*. 2013. <<https://br.vuejs.org/>>. Citado na página [19](#).
- ÁGIL, D. *SCRUM*. 2013. <<https://www.desenvolvimentoagil.com.br/scrum/>>. Citado na página [17](#).

Anexos

ANEXO A – Resolução 02/2017 - FAMAT



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



RESOLUÇÃO Nº 02/2017 DO CONSELHO DA FACULDADE DE MATEMÁTICA

Regulamenta a distribuição da carga horária didática entre os docentes da Faculdade de Matemática

O CONSELHO DA FACULDADE DE MATEMÁTICA DA UNIVERSIDADE FEDERAL DE UBERLÂNDIA, no uso de suas atribuições, em reunião realizada aos 11 dias do mês de maio do ano de 2017;

CONSIDERANDO a necessidade de revisão das normas que regulamentam a distribuição da carga horária didática da Faculdade de Matemática (FAMAT);

R E S O L V E:

Art. 1º A designação da carga horária didática e das disciplinas a serem ministradas por cada docente efetivo(a) da Faculdade de Matemática no segundo semestre de cada ano letivo e no primeiro semestre do ano letivo subsequente levará em conta as atividades desenvolvidas pelo(a) docente no(s) ano(s) anterior(es), conforme consta dos Anexos 1, 2, 3 e 4 desta Resolução. Tais atividades deverão ser descritas nos Relatórios de Atividades Docente, entregues à Secretaria da FAMAT, em data preestabelecida, para que se possa:

- I. Calcular a pontuação de cada docente com base nos Anexos 1 a 4 desta Resolução.
- II. Realizar a divisão do(a)s docentes em grupos, conforme Art. 4º desta Resolução.
- III. Classificar o(a)s docentes pertencentes a cada grupo ou bloco, conforme Arts. 5º e 6º desta Resolução.
- IV. Realizar a distribuição das disciplinas, conforme sistemática descrita nos Arts. 17 e 18 desta Resolução.

Art. 2º O(A)s docentes efetivo(a)s da FAMAT serão considerados em blocos, de acordo com as áreas em que realizaram concurso e o campus universitário em que estão alocado(a)s, conforme segue:

- I. Bloco M: docentes da área da Matemática, Matemática Aplicada ou Educação Matemática alocados nos campi de Uberlândia.
- II. Bloco E: docentes da área de Estatística alocado(a)s nos campi de Uberlândia.
- III. Bloco ME: docentes da área de Matemática, Matemática Aplicada ou Educação Matemática alocados em campi externos a Uberlândia.
- IV. Bloco EE: docentes da área de Estatística alocado(a)s nos campi externos a Uberlândia.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



I – DO RELATÓRIO DE ATIVIDADES DOCENTE E DA CLASSIFICAÇÃO DOS DOCENTES SEGUNDO SUAS ATIVIDADES

Art. 3º Os Relatórios de Atividades Docente deverão ser entregues à Secretaria da FAMAT por todo(a)s o(a)s docentes efetivos da Faculdade, no início de cada ano letivo, em período previamente especificado e em formulário próprio previamente fornecido pela Comissão de Distribuição de Aulas. A composição e as atribuições da Comissão de Distribuição de Aulas estão definidas na Seção IV desta Resolução.

§ 1º Ao ser contratado(a), cada docente será incluído(a) na última posição do grupo G3M ou G3E (dependendo da área de seu concurso). No processo de classificação subsequente, poderá optar por entregar ou não seu relatório de atividades. Se optar por entregar, será classificado(a) normalmente; caso contrário, será classificado(a) na última posição do grupo G3M ou G3E. Para efeito de terminologia, o(a)s docentes nestas condições serão nominado(a)s recém-admitido(a)s. Para efeito de desempate do(a)s recém-admitido(a)s que optarem por não entregar seus relatórios, terá classificação superior aquele(a) que tiver defendido sua tese de doutorado a menos tempo.

§ 2º O(A) docente que estiver afastado(a) para pós-graduação, estágio pós-doutoral, licença maternidade, licença saúde, licença capacitação, ou em gozo de licença prêmio por assiduidade no momento da entrega dos Relatórios de Atividades Docente, deverá apresentar seu relatório e participar de toda a sistemática prevista nesta Resolução. Neste caso, seu relatório deverá conter as atividades realizadas nos dois semestres letivos anteriores a data de início de seu afastamento, excetuando-se as atividades dos Anexos 1 e 4 que serão computadas dentro de seus prazos de validade.

§ 3º O(A) docente que estava afastado(a) em parte do período que deve constar no seu Relatório de Atividades Docente deverá apresentar seu relatório contendo as atividades realizadas nos dois semestres letivos mais recentes em que atuou por completo, excetuando-se as atividades dos Anexos 1 e 4 que serão computadas dentro de seus prazos de validade.

Art. 4º O(A)s docentes pertencentes aos blocos M e E serão preliminarmente classificado(a)s, em cada bloco, de acordo com as pontuações aferidas em seus Relatórios de Atividades Docente. Com base nesta classificação preliminar, o(a)s docentes serão subdividido(a)s em grupos. Considerando N, respectivamente P, como o número total de docentes do bloco M, respectivamente do bloco E, subtraído pelo número de docentes do mesmo bloco que estejam afastado(a)s ou em gozo de licença prêmio, com retorno programado para após o início do semestre letivo subsequente, e para os quais não há professor(a) substituto(a) contratado(a), serão constituídos os seguintes grupos:

- I. Grupo G3M:** Considerando X como sendo a parte inteira do número que representa 30% (trinta por cento) de N e RM o número de docentes recém-admitido(a)s do Bloco M, este grupo será formado pelo(a)s X-RM docentes melhores classificado(a)s do Bloco M, que tenham obrigatoriamente pontuação positiva em todos os Anexos de 1 a 3, e pelos RM docentes recém-admitido(a)s do mesmo bloco. Não havendo X-RM docentes que atendam a esta exigência, o grupo G3M será completado pelo(a)s melhores classificado(a)s do Bloco M que tenham obrigatoriamente pontuação positiva em dois dos Anexos de 1 a 3.



- II. Grupo G3E: Considerando Y como sendo a parte inteira do número que representa 30% (trinta por cento) de P e RE o número de docentes recém-admitido(a)s do Bloco E, este grupo será formado pelo(a)s Y-RE docentes melhores classificado(a)s do Bloco E, que tenham obrigatoriamente pontuação positiva em todos os Anexos de 1 a 3, e pelo(a)s RE docentes recém-admitido(a)s do mesmo bloco. Não havendo Y-RE docentes que atendam a esta exigência, o grupo G3E será completado pelo(a)s melhores classificado(a)s do Bloco E que tenham obrigatoriamente pontuação positiva em dois dos Anexos de 1 a 3.
- III. Grupo G2M: Considerando Z como sendo a parte inteira do número que representa 30% (trinta por cento) de N, este grupo será formado pelo(a)s Z docentes melhores classificado(a)s do Bloco M, excluindo o(a)s já enquadrados no grupo G3M, que tenham obrigatoriamente pontuação positiva em pelo menos dois dos Anexos de 1 a 3.
- IV. Grupo G2E: Considerando W como sendo a parte inteira do número que representa 30% (trinta por cento) de P, este grupo será formado pelo(a)s W docentes melhores classificado(a)s do Bloco E, excluindo o(a)s já enquadrado(a)s no grupo G3E, que tenham obrigatoriamente pontuação positiva em pelo menos dois dos Anexos de 1 a 3.
- V. Grupo G1M: constituído pelo(a)s docentes do Bloco M que não foram enquadrados nos Grupos G3M e G2M.
- VI. Grupo G1E: constituído pelo(a)s docentes do Bloco E que não foram enquadrados nos Grupos G3E e G2E.

Art. 5º Após a subdivisão do(a)s docentes dos Blocos M e E em grupos, conforme disposto no Art. 4º, o(a)s docentes serão classificado(a)s internamente em cada grupo, em função das pontuações aferidas em seus Relatórios de Atividades Docente.

§ 1º Quando o grupo G3M ou G3E for completado por docentes com pontuação positiva em apenas dois dos Anexos de 1 a 3, este(a)s deverão ser classificado(a)s entre si ficando, na classificação final do respectivo grupo, abaixo do(a)s docentes do grupo que tiverem pontuação positiva em todos os Anexos de 1 a 3.

§ 2º O(A)s docentes efetivo(a)s recém-admitido(a)s que optarem por não entregar seus Relatórios de Atividades Docente serão classificado(a)s no grupo G3M ou G3E, nas últimas posições, conforme disposto no § 1º do Art. 3º.

§ 3º O(A)s docentes efetivo(a)s dos blocos M ou E que não entregarem seus Relatórios de Atividades Docente no prazo devido, excetuando-se o(a)s recém-admitidos(a) que optarem por não fazê-lo em face do disposto no § 1º do Art. 3º, serão classificado(a)s no grupo G1M ou G1E, nas últimas posições.

Art. 6º O(A)s docentes dos blocos ME e EE serão classificado(a)s em cada bloco, por campus, de acordo com as pontuações aferidas em seus Relatórios de Atividade Docente.

Parágrafo único. O(A)s docentes dos blocos referidos no *caput* que não entregarem seus Relatórios de Atividades Docente no prazo devido serão classificado(a)s nas últimas posições dos respectivos blocos.



Art. 7º Para fins de desempate na classificação preliminar prevista no *caput* do Art. 4º e nas classificações internas em cada grupo ou bloco previstas nos Arts. 5º e 6º, serão observados, ordenadamente, os seguintes critérios de desempate:

- I. Maior pontuação em atividades do Anexo 1.
- II. Maior pontuação em atividades do Anexo 2.
- III. Maior pontuação em atividades do Anexo 3.
- IV. Maior tempo na carreira do magistério superior na Universidade Federal de Uberlândia.

Art. 8º Em caso de divergência entre a pontuação declarada pelo(a) docente e aquela considerada pela Comissão de Distribuição de Aulas, o(a) docente será informado(a), antes da divulgação da classificação final, e lhe será garantido um prazo de 48 horas para possíveis esclarecimentos. A pontuação e a classificação final, determinadas conforme disposto nesta Resolução, serão informadas a cada docente antes do início da sistemática de distribuição de aulas de que trata a Seção III desta Resolução. Posteriormente, a comissão divulgará no âmbito da FAMAT as pontuações obtidas, por grupo e em ordem decrescente de classificação, sem qualquer referência ou identificação do(a) docente com a referida pontuação.

Art. 9º A divisão do(a)s docentes efetivo(a)s da FAMAT em blocos e grupos e a classificação interna do(a)s docentes em cada grupo ou bloco, conforme disposto nos Arts. 4º, 5º e 6º, serão consideradas para efeito de distribuição de aulas para os dois semestres subsequentes à realização da classificação.

Art. 10. Professore(a)s substituto(a)s ou contratado(a)s em caráter temporário não serão considerado(a)s na composição dos grupos e na classificação do(a)s docentes efetivo(a)s previstas nesta Resolução.

II – DA LISTAGEM DE DISCIPLINAS PREFERENCIAIS

Art. 11. Cada docente efetivo(a) da FAMAT, após tomar ciência de sua classificação final em um dos grupos ou blocos e do conjunto de disciplinas e horários disponíveis para o semestre letivo subsequente, deverá entregar à Secretaria da FAMAT, sempre que solicitado(a) pela Comissão de Distribuição de Aulas, uma lista, denominada Listagem de Disciplinas Preferenciais, de sugestões de disciplinas que gostaria de ministrar no semestre letivo subsequente, em ordem de preferência, na forma e data preestabelecidas pela Comissão.

Parágrafo único. Conforme previsto na sistemática de distribuição de aulas descrita no Art. 17 desta Resolução, cada docente dos blocos M e E será solicitado(a) a entregar sua Listagem de Disciplinas Preferenciais em duas ocasiões.

Art. 12. Para o fim específico da elaboração das Listagens de Disciplinas Preferenciais, as disciplinas oferecidas pela FAMAT serão agrupadas em dois blocos principais de disciplinas: um bloco constando



das disciplinas das áreas de Matemática, Matemática Aplicada e Educação Matemática, que será o bloco de disciplinas sob responsabilidade do(a)s docentes do Bloco M, e outro bloco constando das disciplinas da área de Estatística, que será o bloco de disciplinas sob responsabilidade do(a)s docentes do Bloco E. Ademais, a Comissão de Distribuição de Aulas escolherá dois sub-blocos de disciplinas dentro dos blocos M e E, denominados sub-bloco SM e sub-bloco SE, respectivamente.

§ 1º A carga horária total de cada um dos sub-blocos SM e SE não poderá exceder a 15% (quinze por cento) da carga horária total do bloco principal que o contém.

§ 2º Os sub-blocos SM e SE deverão ser constituídos, entre outras, por disciplinas que apresentam dificuldade de atribuição.

Art. 13. Cada Listagem de Disciplinas Preferenciais de cada docente dos blocos M e E deverá ser constituída por, no mínimo, 15 (quinze) disciplinas relacionadas em ordem decrescente de preferência, com indicação do curso em que serão ministradas, sendo que destas pelo menos 50% devem ser de disciplinas não pertencentes aos cursos de graduação em Matemática ou Estatística da FAMAT e pelo menos duas devem ser disciplinas do turno noturno.

Art. 14. A não entrega pelo(a) docente de sua Listagem de Disciplinas Preferenciais no molde e na data solicitados, assim como a não disponibilidade das disciplinas indicadas na Listagem de Disciplinas Preferenciais do(a) docente, implicará na liberdade para a Comissão de Distribuição de Aulas atribuir para o(a) docente quaisquer disciplinas, respeitando-se os limites de carga horária específicos do grupo em que o(a) docente estiver classificado(a).

Art. 15. O(A) docente que estiver ministrando uma mesma disciplina pela segunda vez consecutiva, não poderá indicar tal disciplina em sua primeira Listagem de Disciplinas Preferenciais.

III – DO PROCESSO DE ATRIBUIÇÃO DE DISCIPLINAS

Art. 16. A atribuição de disciplinas ao(à)s docentes dos Blocos M e E deverá respeitar o limite máximo de 16 horas-aula semanais, para qualquer docente, e os seguintes limites inferiores de carga horária didática, conforme os grupos de docentes:

- I. Para docentes dos grupos G3M e G3E, mínimo de 8 horas-aula semanais;
- II. Para docentes dos grupos G2M e G2E, mínimo de 10 horas-aula semanais;
- III. Para docentes dos grupos G1M e G1E, mínimo de 12 horas-aula semanais.

§ 1º A soma da carga horária didática de cada docente efetivo(a) da FAMAT no segundo semestre de um ano com a carga horária didática do(a) mesmo(a) docente no primeiro semestre do ano letivo subsequente não poderá exceder o montante de 30 horas-aula semanais, exceto se o(a) docente autorizar carga acima desse limite.

§ 2º Ao(À) Diretor(a) da FAMAT, ao(à)s Coordenadore(a)s dos cursos de Pós-Graduação e Graduação regulares da FAMAT e ao(à) coordenador(a) do Mestrado Profissional em Ensino de Ciências e



Matemática em que a FAMAT participa como unidade consorciada (caso este(a) seja docente efetivo(a) da FAMAT), que estarão em exercício da respectiva função no semestre a que se refere a distribuição de aulas, poderá ser atribuída a carga horária mínima de 6 horas-aula semanais.

§ 3º Ao(À) Assessor(a) Acadêmico(a)-Administrativo(a) que estará no exercício da função no semestre a que se refere a distribuição de aulas poderá ser atribuída a carga horária mínima de 8 horas-aulas semanais.

Art. 17. A sistemática de distribuição de aulas para o(as) docentes dos blocos M e E será gerenciada pela Comissão de Distribuição de Aulas, que será responsável por divulgar antecipadamente o cronograma de distribuição e fazer cumprir as seguintes etapas:

- I. Os colegiados dos cursos de graduação regulares da FAMAT poderão indicar até três docentes efetivo(as) da FAMAT, que estejam de comum acordo, para ministrar, cada um(a), uma disciplina do primeiro ou segundo períodos do respectivo curso. O(A) mesmo(a) docente não poderá ser indicado(a) mais do que duas vezes consecutivas, ainda que para disciplinas distintas.
- II. Os colegiados dos programas de pós-graduação regulares da FAMAT e do Mestrado Profissional em Ensino de Ciências e Matemática atribuirão as disciplinas dos programas para o(as) docentes credenciado(a)s.
- III. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a etapa II, e cada docente dos blocos M e E entregará sua primeira Listagem de Disciplinas Preferenciais que será utilizada para a atribuição de uma única disciplina a cada docente (primeira rodada de distribuição). Nesta etapa participam todo(as) o(as) docentes efetivo(as), inclusive o(as) que já participaram das etapas anteriores e não completaram sua carga didática. A listagem de disciplinas, nesta etapa, deve obedecer aos seguintes critérios:
 - (a) Docentes dos grupos G3M ou G3E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 4 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.
 - (b) Docentes dos grupos G2M ou G2E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 5 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.
 - (c) Docentes dos grupos G1M ou G1E devem listar disciplinas com cargas horárias semanais maiores ou iguais a 6 horas, disciplinas dos cursos da própria FAMAT ou disciplinas pertencentes aos sub-blocos SM ou SE.
- Liberam-se das exigências (a)-(c) o(as) docentes referido(as) nos §§ 2º e 3º do Art. 16, e também aquele(as) que já tiveram disciplinas atribuídas nas etapas I ou II da sistemática de distribuição de aulas e que necessitam de carga inferior às estabelecidas nos critérios (a)-(c) para completarem as cargas didáticas mínimas exigidas nos grupos aos quais pertencem.
- V. Com base nas Listagens de Disciplinas Preferenciais do(as) docentes, será atribuída uma disciplina a cada docente, respeitando-se a prioridade dos grupos G3M e G3E sobre os grupos G2M e G2E, e destes sobre os grupos G1M e G1E, além da classificação interna do(as) docentes em cada grupo. O(A)s docentes referido(as) no § 2º do Art. 16 escolherão, nas suas



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



respectivas posições de classificação, suas cargas didáticas do semestre por completo, respeitando-se o limite mínimo estabelecido no mesmo dispositivo.

- VI. A Comissão de Distribuição de Aulas informará a cada docente a(s) disciplina(s) que lhe foram atribuída(s).
- VII. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a realização da etapa VI.
- VIII. Todo(a)s os docentes dos grupos G3M, G3E, G2M e G2E deverão entregar uma nova Listagem de Disciplinas Preferenciais, exceto o(a)s docentes referidos no § 2º do Art. 16.
- IX. Será atribuída uma disciplina a cada docente dos grupos G3M, G3E, G2M e G2E que ainda não tiver atingido a carga horária didática mínima estabelecida no Art. 16, considerando a Listagem de Disciplinas Preferenciais de cada um(a) (entregue na etapa VIII) e respeitando a classificação interna do(a)s docentes no grupo. Ao(À) docente que já tiver atingido a carga horária didática mínima poderá ou não ser atribuída uma nova disciplina, conforme necessidade ou a pedido do(a) docente.
- X. A Comissão de Distribuição de Aulas divulgará a lista de disciplinas e horários disponíveis após a realização da etapa IX.
- XI. Todo(a)s o(a)s docentes dos grupos G1M e G1E deverão entregar uma nova Listagem de Disciplinas Preferenciais, exceto o(a)s docentes listado(a)s no § 2º do Art. 16.
- XII. Nesta etapa a Comissão de Distribuição de Aulas atribuirá as disciplinas restantes. Para tal:
 - (a) Serão atribuídas disciplinas ao(à)s docentes dos grupos G1M e G1E, levando em conta as Listagens de Disciplinas Preferenciais e a classificação nos grupos, respeitando os limites de cargas horárias didáticas estabelecidos no Art. 16.
 - (b) Serão atribuídas disciplinas ao(à)s docentes dos grupos G3M, G3E, G2M e G2E que não tiverem obtido carga didática mínima, de acordo com o Art. 16, nas etapas anteriores.
- XIII. A Comissão de Distribuição de Aulas informará a cada docente as disciplinas que lhe foram atribuídas e, após respeitados os prazos e analisadas as eventuais solicitações de alteração, encaminhará a proposta de atribuição de aulas para deliberação do Conselho da FAMAT.

Parágrafo único. Nas etapas finais da sistemática de distribuição e para fins de completar o processo de distribuição de aulas, a comissão poderá, eventualmente, atribuir para alguns(mas) docentes carga horária didática menor que a mínima prevista para o seu grupo.

Art. 18. A sistemática de distribuição de aulas para o(a)s docentes dos blocos ME e EE seguirá as seguintes fases:

- I. Para cada campus, a Comissão de Distribuição de Aulas divulgará as listas de disciplinas e horários sob responsabilidade do(a)s docentes de cada bloco.
- II. Cada docente entregará sua Listagem de Disciplinas Preferenciais.
- III. A Comissão de Distribuição de Aulas fará a distribuição das disciplinas considerando a classificação do(a)s docentes de cada bloco, conforme Art. 6º, e a Listagem de Disciplinas Preferenciais de cada docente.

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



- IV. A Comissão de Distribuição de Aulas atribuirá as disciplinas restantes ao(à)s professore(a)s substituto(a)s e contratado(a)s em caráter temporário, se houver.
- V. A Comissão de Distribuição de Aulas informará a cada docente as disciplinas que lhe foram atribuídas e, após respeitados os prazos e analisadas as eventuais solicitações de alteração, encaminhará a proposta de atribuição de aulas para deliberação do Conselho da FAMAT.

IV – DA COMISSÃO DE DISTRIBUIÇÃO DE AULAS

Art. 19. Os trabalhos referentes à análise dos Relatórios de Atividade Docente, à divisão do(a)s docentes em grupos (conforme Art. 4º), à classificação do(a)s docentes (conforme Arts. 5º e 6º) e ao gerenciamento da sistemática de distribuição de aulas, serão desenvolvidos por uma comissão, denominada Comissão de Distribuição de Aulas.

Art. 20. A Comissão de Distribuição de Aulas será nomeada pelo(a) Diretor(a) da FAMAT em até 75 (setenta e cinco) dias após o início do primeiro semestre letivo de cada ano e será composta pelos seguintes membros:

- I. O(A) Assessor(a) Acadêmico(a)-Administrativo(a) da FAMAT, como presidente;
- II. O(A)s coordenadore(a)s dos cursos de graduação regulares da FAMAT; e
- III. Dois(uas) representantes do corpo docente efetivo da FAMAT, eleito(a)s pelo Conselho da FAMAT, para a distribuição de aulas do segundo semestre letivo do ano corrente e do primeiro semestre letivo do ano subsequente.

Art. 21. Compete à Comissão de Distribuição de Aulas:

- I. Solicitar ao(à)s docentes efetivo(a)s da FAMAT seus Relatórios de Atividades Docente, nos termos estabelecidos nesta Resolução, e definir a data de entrega dos Relatórios à Secretaria da FAMAT, em tempo de se fazer cumprir esta Resolução.
- II. Solicitar ao(à)s docentes a documentação comprobatória das atividades declaradas em seus Relatórios de Atividades Docente, sempre que julgar necessário fazê-lo.
- III. Efetuar a análise dos Relatórios de Atividades Docente, classificar o(a)s docentes seguindo as pontuações estabelecidas nos Anexos 1, 2, 3 e 4, dividir o(a)s docentes em grupos, de acordo com o Art. 4º, e classificar o(a)s docentes em cada grupo ou bloco, conforme disposto nos Arts. 5º e 6º.
- IV. Dar ciência a cada docente efetivo(a) acerca de seu bloco, grupo e classificação.
- V. Elaborar e divulgar, a todos os docentes efetivos da FAMAT, o cronograma de distribuição de aulas, considerando todas as etapas previstas nos Arts. 17 e 18, além das listagens de disciplinas, com os respectivos horários, em cada etapa dos Arts. 17 e 18 em que tal procedimento está previsto.
- VI. Definir e fazer constar da solicitação da Listagem de Disciplinas Preferenciais, a forma com que cada docente efetivo(a) deverá elaborar sua Listagem, em cada etapa em que isso for necessário.



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



- VII. Definir a cada semestre a composição dos sub-blocos SM e SE, conforme disposto no Art. 12 e em seus parágrafos.
- VIII. Cumprir a sistemática de distribuição de disciplinas estabelecida nos Arts. 17 e 18.
- IX. Dar ciência a cada docente das disciplinas que lhe foram preliminarmente designadas para o semestre subsequente e de seus respectivos horários.
- X. Deliberar, pelo voto da maioria de seus membros, acerca de solicitações encaminhadas pelo(a)s docentes após serem cientificado(a)s das disciplinas que lhes foram preliminarmente designadas e efetuar as alterações que forem consideradas pertinentes e necessárias.
- XI. Dar ciência do resultado de reexame da distribuição preliminar, conforme tal lhe seja solicitado, sempre à luz desta Resolução.
- XII. Encaminhar a proposta de distribuição da carga didática e das disciplinas ao Conselho da FAMAT para deliberação.
- XIII. Resolver os casos omissos.

Art. 22. Das decisões da Comissão de Distribuição de Aulas cabe recurso ao Conselho da FAMAT. O recurso deve ser interposto no prazo de 48 (quarenta e oito) horas após a divulgação da decisão.

V – DISPOSIÇÕES FINAIS E TRANSITÓRIAS

Art. 23. Após a apresentação da proposta de distribuição de aulas por parte da Comissão de Distribuição de Aulas e a aprovação da proposta pelo Conselho da FAMAT, apenas o surgimento de fatos novos justificarão qualquer revisão na distribuição de aulas.

§ 1º A revisão a que se refere o *caput* deste artigo será efetuada conjuntamente pelo(a) Diretor(a) e pelo(a) Assessor(a) Acadêmico(a)-Administrativo(a) da FAMAT.

§ 2º As eventuais revisões efetuadas serão informadas e justificadas, pelo(a) Diretor(a), ao Conselho da FAMAT, na primeira reunião subsequente.

Art. 24. Esta Resolução entra em vigor nesta data.

Uberlândia, 11 de maio de 2017

Marcio Colombo Fenille
Presidente



ANEXOS DA RESOLUÇÃO Nº 02/2017 DO CONSELHO DA FACULDADE DE MATEMÁTICA

Anexo 1 – Atividades de pesquisa e coordenação de projetos.

- Não devem ser considerados neste anexo os artigos publicados em periódicos de divulgação científica ou iniciação científica.
- O(A) docente deverá indicar a área de avaliação de cada artigo (dentre as permitidas em cada item) e o correspondente estrato do Qualis-CAPES do periódico, observando o Qualis mais recente.
- O(A) docente deverá indicar a vigência completa de cada projeto de pesquisa, ensino ou extensão, mas, para o cálculo da pontuação correspondente, deverá considerar apenas os meses do ano anterior contidos no período de vigência, conforme Ref. 8 da tabela a seguir.
- Considerar ano civil (de 1º de janeiro a 31 de dezembro).

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato A1 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	120 pontos por artigo	-
2	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato A2 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	110 pontos por artigo	-
3	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B1 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	100 pontos por artigo	-
4	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B2 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)	75 pontos por artigo	-
5	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B3 dentre as seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar.(Válidos artigos publicados nos quatro anos anteriores)	55 pontos por artigo	-
6	Artigo publicado em periódico científico especializado indexado no Qualis-CAPES no estrato B4 dentre as	35 pontos por artigo	-



**SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



	seguintes áreas de avaliação: Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar. (Válidos artigos publicados nos quatro anos anteriores)		
7	Artigo publicado em periódico científico que não pertence ao Qualis-CAPES das áreas Matemática/Probabilidade e Estatística, Educação, ou Interdisciplinar, em nenhuma das quatro últimas avaliações do Qualis-CAPES, mas que esteja em algum dos estratos A1, A2, B1, B2, B3 ou B4 de outra área do Qualis-CAPES. (Válidos artigos publicados nos quatro anos anteriores)	10 pontos por artigo	-
8	Coordenação de projeto de pesquisa, ensino ou extensão, com financiamento externo, sem remuneração complementar e/ou bolsa para o (a) coordenador(a), comprovada por documento de aprovação do projeto pelo órgão de fomento	5 pontos por mês de coordenação no ano anterior	-

Anexo 2 – Atividades de capacitação, orientação, extensão e administração.

- Devem ser consideradas as atividades realizadas no ano civil (de 1º de janeiro a 31 de dezembro) anterior à entrega do Relatório de Atividades Docente.

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Tese de doutorado defendida pelo(a) docente	120	120
2	Orientação de tese de doutorado	15 pontos por aluno/mês completo	540
3	Orientação de dissertação de mestrado	10 pontos por aluno/mês completo	360
4	Orientação de PIVIC, PIBIC ou projetos de iniciação científica oriundos de projetos de pesquisa com financiamento externo	5 pontos por aluno/mês completo	180
5	Orientação de bolsista do PET (Programa de Educação Tutorial) em pesquisa individual e/ou coletiva	3 pontos por mês completo	72
6	Outras orientações de iniciação científica aprovadas por órgãos oficiais	3 pontos por aluno/mês completo	72

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



7	Orientação de monografia de cursos de especialização não remunerados oferecidos pela FAMAT	4 pontos por aluno/mês completo	96
8	Orientação de Trabalhos de Conclusão de Curso nos cursos de graduação (Não computar orientações já consideradas nos itens 4, 5 e 6)	2 pontos por aluno/mês completo	48
9	Membro titular de banca de defesa de doutorado	30	120
10	Membro titular de banca de defesa de mestrado	15	60
11	Membro titular de banca de defesa de monografia de pós-graduação <i>lato sensu</i>	5	20
12	Membro titular de banca de defesa de monografia de graduação	4	16
13	Membro titular de banca de qualificação de doutorado	10	40
14	Membro titular de banca de qualificação de mestrado	5	20
15	Exercício da Diretoria da FAMAT	30 pontos por mês completo	-
16	Exercício de direção no âmbito da UFU. (Item incluído pela Resolução nº 02/2014 do CONFAMAT, de 04/12/2014)	10 pontos por mês completo	-
17	Exercício de Coordenação de curso de graduação ou programa de pós-graduação regular da FAMAT ou programa de pós-graduação consorciado	15 pontos por mês completo	-
18	Exercício de Assessoria Acadêmica-Administrativa da FAMAT	10 pontos por mês completo	-
19	Coordenação da Semana da Matemática/Estatística ou de eventos científicos promovidos pela FAMAT	60 pontos por evento	60
20	Coordenação de eventos científicos não pontuados no item 19	30 pontos por evento	30
21	Membro da Comissão Organizadora da Semana da Matemática/Estatística da FAMAT ou de eventos científicos promovidos pela FAMAT	15 pontos por evento	30
22	Membro da Comissão Organizadora de eventos científicos não pontuados no item 21	10 pontos por evento	20
23	Membro de Comitê Científico de eventos	5 pontos por evento	10
24	Presidência do Comitê Editorial da Revista Eletrônica Matemática e Estatística em Foco ou de outra revista	40 pontos por edição	-

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



	científica da área de Matemática, Matemática Aplicada, Educação Matemática ou Estatística	publicada	
25	Membro do Comitê Editorial da Revista Eletrônica Matemática e Estatística em Foco ou de outra revista científica	5 pontos por edição publicada	-
26	Coordenação de núcleo, do PICME ou laboratório da FAMAT	2 pontos por mês completo	-
27	Presidência de Núcleo Docente Estruturante de curso de graduação da FAMAT	3 pontos por mês completo	-
28	Presidência da Câmara de Pesquisa, Extensão e Pós-Graduação da FAMAT	3 pontos por mês completo	-
29	Coordenação de estágio supervisionado	5 pontos por mês completo	-
30	Membro do Conselho da FAMAT	5 pontos por mês completo	-
31	Membro de colegiado de curso ou Núcleo Docente Estruturante de curso da UFU (O presidente do Núcleo Docente Estruturante pontua somente na Ref. 27 acima)	2 pontos por mês completo	-
32	Representação da FAMAT junto à Biblioteca	1 ponto por mês completo	-
33	Representação da FAMAT junto ao Conselho de Extensão, Cultura e Assuntos Estudantis	1 ponto por mês completo	-
34	Membro da Comissão Interna de Avaliação de Desempenho Docente (CIADD)	5 pontos por mês completo	-
35	Membro da Comissão de Distribuição de Aulas	6 pontos por distribuição	
36	Membros de comissões internas da Unidade Acadêmica	3 pontos por comissão	-
37	Membro de comissão ou comitê da UFU externo a FAMAT	2 pontos por comissão/ mês completo	48
38	Membro de banca de concurso público para professor(a) efetivo(a)	15	-
39	Membro de banca de processo seletivo para contratação de professore(a)s substituto(a)s	8	-
40	Parecer <i>ad-hoc</i> para publicações em periódicos com corpo editorial	5	20

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120

Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br

Uberlândia-MG - 38408-100



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



41	Parecer <i>ad-hoc</i> para órgãos oficiais de utilidade pública	2	20
42	Coordenação de projeto de extensão sem remuneração complementar e/ou bolsa para o(a) coordenador(a), sem financiamento e aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas)	30	60
43	Coordenação de projeto de extensão sem remuneração complementar e/ou bolsa para o(a) coordenador(a), com financiamento e aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas)	50	100

Anexo 3 – Atividades de ensino (sem remuneração complementar).

- Devem ser consideradas as atividades de ensino realizadas nos dois semestres do ano letivo anterior à entrega do Relatório de Atividades Docente.

Ref.	ATIVIDADE	Pontuação
1	Disciplina ministrada em cursos de graduação, atribuída ao(à) docente pela FAMAT, com aprovação do Conselho da Unidade.	Multiplicar o número de horas-aula semanais da disciplina por 10 (dez). Para disciplina ministrada para turma com mais de 40 estudantes, adicionar 0,25 ponto por estudante acima de 40. (O número de estudantes da turma deve ser entendido como aquele constante do Registro de Resultados da disciplina gerado pelo Portal do Docente).
2	Disciplina ministrada em cursos de pós-graduação <i>stricto sensu</i> cadastradas na FAMAT.	Multiplicar o número de horas-aula semanais da disciplina por 10 (dez).
3	Coordenação de projeto de ensino sem remuneração complementar e/ou bolsa para o(a) coordenador(a), aprovado pelo Conselho da FAMAT. (Válidos somente projetos com pelo menos 30 horas, limitado a dois projetos).	30 pontos por projeto



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



Anexo 4 – Outras atividades.

- Considerar ano civil (de 1º de janeiro a 31 de dezembro).

Ref.	ATIVIDADE	Pontuação	Pontuação máxima no período de avaliação
1	Publicação de livro com selo de editora com corpo editorial. (Válidos livros publicados nos dois anos anteriores)	120	120
2	Publicação de capítulo de livro com selo de editora com corpo editorial. (Válidos livros publicados nos dois anos anteriores)	30	60
3	Palestra/conferência/apresentação oral de trabalho em evento científico internacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	20	60
4	Minicurso/oficina em evento científico internacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	30	60
5	Palestra/conferência/apresentação oral de trabalho em evento científico nacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	10	30
6	Minicurso/oficina ministrado(a) em evento científico nacional. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior).	15	30
7	Apresentação de pôster de trabalho em reunião científica. (Válidas somente apresentações realizadas pelo(a) próprio(a) docente no ano anterior)	5	15
8	Artigo publicado em periódico científico especializado, com corpo editorial. (Válidos artigos publicados no ano anterior, não pontuados no Anexo 1).	10	20

Secretaria da Faculdade Matemática

Av. João Naves de Ávila, 2121 - Campus Santa Mônica - Bloco F – Sala 1F120
Fone (34) 3239-4156 - Fax (34) 3239-4394 – e-mail: famat@ufu.br
Uberlândia-MG - 38408-100

ANEXO B – Relatório de Atividades – Modelo

RELATÓRIO DE ATIVIDADES DOCENTE – FAMAT – 2018

Docente: _____ Campus: _____

Área: Matemática, Matemática Aplicada, Educação Matemática Estatística.

Anexo 1 – Atividades de pesquisa e coordenação de projetos

- Não considerar, neste anexo, artigos publicados em periódicos indexados no Extrato B5 do Qualis-CAPES ou em periódicos de divulgação ou iniciação científica; tais artigos devem ser considerados no Anexo 4.

- Atenção ao Anexo 1 da Resolução Nº 02/2017 do CONFAMAT quanto às áreas do Qualis que permitem pontuação completa (conforme Itens 1 a 6 do Anexo 1); são elas: (1) **Matemática/Probabilidade e Estatística**; (2) **Educação**; (3) **Interdisciplinar**. Artigos em periódicos avaliados de A1 a B4 em outras áreas do Qualis devem ser pontuados exclusivamente conforme Item 7 do Anexo 1 da Resolução.

Artigos científicos publicados em periódicos indexados no Qualis-CAPES nos extratos de A1 a B4, de 1º de janeiro de 2014 a 31 de dezembro de 2017 . (Inserir referência completa de cada artigo)	Área e ano do Qualis-CAPES (considerar a avaliação mais recente em que o periódico conste)	Extrato Qualis-CAPES do periódico	Pontuação conforme Anexo 1 da Resolução Nº 02/2017

Coordenação de projeto de pesquisa, ensino ou extensão, com financiamento externo, sem remuneração complementar e/ou bolsa para o coordenador (indicar o título do projeto)	Agência Financiadora	Período completo da vigência	Nº de meses em que o projeto esteve vigente em 2017	Pontuação conforme Anexo 1 da Resolução Nº 02/2017

Total de pontos do Anexo 1: _____ pontos.

Anexo 2 – Atividades de orientação, extensão e administração.

Considerar atividades realizadas no ano civil de 2017, ou seja, de 1º de janeiro a 31 de dezembro de 2017.

Indicar **Ref.** (na primeira coluna da tabela) de acordo com o Anexo 2 da Resolução Nº 02/2017.

Ref.	ATIVIDADE (Descrever detalhadamente cada atividade. Para atividades de orientação, indicar a natureza e o título do trabalho, o aluno, a vigência e a agência financiadora, se houver.)	Mês ou meses de 2017 em que a atividade foi realizada	Pontuação (conforme Anexo 2 da Resolução Nº 02/2017)

Total de pontos do Anexo 2: **301 pontos.**

Anexo 3 – Atividades de ensino (sem remuneração complementar).

Considerar as atividades de ensino realizadas nos dois semestres do ano letivo de 2017.

Docentes que estiveram afastado(a)s em parte ou em todo o ano letivo de 2017 devem observar o estabelecido nos §§ 2º e 3º do Art. 3º da Resolução Nº 02/2017 do CONFAMAT.

Disciplinas ministradas no 1º semestre letivo de 2017			
Disciplina ministrada em curso de graduação ou pós-graduação stricto sensu (Indicar código/nome da disciplina e a turma)	Horas-aula semanais da disciplina	Número de alunos na turma	Pontuação (conforme Anexo 3 da Resolução Nº 02/2017)

Disciplinas ministradas no 2º semestre letivo de 2017			
Disciplina ministrada em curso de graduação ou pós-graduação stricto sensu (Indicar código/nome da disciplina e a turma)	Horas-aula semanais da disciplina	Número de alunos na turma	Pontuação (conforme Anexo 3 da Resolução Nº 02/2017)

Total de pontos do Anexo 3: _____ **pontos.**

Anexo 4 – Outras atividades

Considerar ano civil (de 1º de janeiro à 31 de dezembro) e seguir os “períodos de validade” de cada atividade conforme indicado no Anexo 4 da Resolução Nº 02/2107 do CONFAMAT.

Indicar **Ref.** (na primeira coluna da tabela) de acordo com o Anexo 4 da Resolução Nº 02/2017.

Ref.	Atividade	Data de ocorrência	Pontuação (conforme Anexo 4 da Resolução Nº 02/2017)

Total de pontos do Anexo 4: _____ **pontos.**

Consolidação da Pontuação

	Pontuação
Anexo 1	
Anexo 2	
Anexo 3	
Anexo 4	
T O T A L	

Assinatura do docente

