

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raif Cerveny Passos

**Desenvolvimento de Conteúdo Online para
ensino de Programação**

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raif Cervený Passos

**Desenvolvimento de Conteúdo Online para ensino de
Programação**

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 Sistemas de Informação.

Orientador: André Ricardo Backes

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2018

Agradecimentos

Agradeço primeiramente minha família, que com seu suporte me possibilitou a conclusão do curso. Agradeço também meu orientador André R. Backes por seu apoio, ajuda e compreensão e também a FACOM, que com seus professores me proporcionou conhecimentos que levarei para toda a minha carreira.

Resumo

Esse documento descreve o processo de desenvolvimento de um conteúdo que será disponibilizado por meio de um aplicativo para dispositivos móveis. Esse aplicativo irá auxiliar o aprendizado de programação na Linguagem C. O conteúdo foi organizado utilizando o XML. Além disso, o HTML foi usado como padrão para visualização.

Palavras-chave: Aplicativo. Linguagem C. Programação. Ensino.

Lista de ilustrações

Figura 1 – Estrutura em árvore de um documento XML adaptado de [W3Schools 2018]	10
Figura 2 – Exemplo de código em HTML utilizando algumas tags citadas	12
Figura 3 – Exemplo de tópico em HTML	14
Figura 4 – Página HTML gerada a partir da figura 3	15
Figura 5 – Exemplo de aula com código e comentários	17

Sumário

1	INTRODUÇÃO	6
1.1	Objetivos	6
2	REVISÃO BIBLIOGRÁFICA	8
2.1	A Linguagem C	8
2.2	O modelo HTML	9
2.3	O modelo XML	9
3	ATIVIDADES DESENVOLVIDAS	11
3.1	Elaboração do modelo HTML	11
3.2	Elaboração do modelo XML	12
3.3	Elaboração dos tópicos	13
4	RESULTADOS E DISCUSSÃO	16
5	CONCLUSÃO	18
	REFERÊNCIAS	19

1 Introdução

A compreensão da linguagem de programação C fornece base para o entendimento de várias outras linguagens, sendo que esta linguagem foi a base para desenvolvimento de várias outras amplamente utilizadas. Apesar da linguagem C ter muitos recursos e poder alcançar níveis avançados de programação, ela pode ser utilizada de forma simples em cursos introdutórios e sua dificuldade pode ser aumentada gradativamente, assim aumentando a capacidade dos alunos de criar algoritmos mais avançados.

Mesmo muito conhecida e utilizada, cursos que ofertam as disciplinas que lecionam programação na linguagem C possuem altos índices de reprovação e muitas vezes discussões superficiais, ou seja, não proporcionam um ensino bem estruturado e mais profundo sobre a linguagem, o que acaba por não fornecer ao aluno uma base sólida.

Nesse cenário, um problema surge no mercado de trabalho. Por profissionais não possuírem uma base sólida sobre programação, eles acabam enfrentando dificuldades para propor soluções. Visando aumentar seus conhecimentos, esses profissionais buscam ferramentas para aprendizado mas tem dificuldades para encontrar uma plataforma de ensino acessível, bem fundamentada e que explique o conteúdo de forma prática. Com o acesso à internet em dispositivos móveis, como tablets e celulares, o acesso ao conteúdo pode ser feito de qualquer local, possibilitando assim o enriquecimento do domínio de novos alunos sobre programação, gerando uma base sólida que irá facilitar o aprendizado de outras linguagens, além de melhorar a lógica de programação desses profissionais.

Este trabalho consiste no desenvolvimento do conteúdo teórico que será incluído futuramente em um aplicativo que possui o objetivo de ensinar a linguagem de programação C.

1.1 Objetivos

Atualmente é difícil encontrar uma boa ferramenta que forneça aprendizado sobre a linguagem C. Assim, este trabalho busca fornecer um conteúdo progressivo, de fácil entendimento e acessível para o usuário, sendo o mesmo disponibilizado através de plataformas móveis.

O conteúdo foi desenvolvido como um material teórico bem fundamentado e que fornece em seu início conceitos simples e introduz novos conceitos da linguagem gradativamente. São também disponibilizados exercícios que ajudam na fixação do aprendizado e na melhor compreensão de como as ferramentas da linguagem podem ser utilizadas.

Após a conclusão do aplicativo, espera-se que o mesmo seja utilizado como apoio

para alunos em graduação e também por profissionais que buscam aprender a linguagem C.

2 Revisão Bibliográfica

Este capítulo trata-se de um resumo sobre a linguagem de programação C, o HTML e o XML.

2.1 A Linguagem C

A linguagem C foi originalmente desenvolvida por Dennis Ritchie entre 1969 e 1973. Foi padronizada pelo American National Standards Institute (ANSI) em 1989 e mais tarde pelo International Organization for Standardization (ISO). Por ser uma linguagem de alto nível, com maior reaproveitamento de código, de paradigma procedural e com poder para criação de programas complexos, como sistemas operacionais, a linguagem C se tornou uma das linguagens mais utilizadas.

O uso da linguagem C para estudo de algoritmos fornece ao aluno conceitos fundamentais de programação que são base para várias outras linguagens atuais, por esse motivo é muito usada nos cursos introdutórios de programação. Além disso, é uma linguagem imperativa, ou seja, descreve a computação como ações ou comandos, que alteram o estado de um programa. A linguagem C também fornece construtores que mapeiam eficientemente recursos de baixo nível, como controle de memória e acesso ao microprocessador, sendo assim, é muito usada em aplicações que foram codificadas na linguagem de programação Assembly, o que tornou a mesma muito usada no desenvolvimento de sistemas operacionais. Além disso, a linguagem C possui um propósito geral e fornece suporte para programação estruturada e recursão.

Houve uma revisão da linguagem C no final dos anos 90, o que levou a publicação do ISO/IEC 9899:1999 em 1999, normalmente denominado como C99. O C99 introduziu novas funcionalidades, incluindo novos tipos de dado como `long long int`, um tipo complexo usado para representar números complexos, arrays de tamanho variável, membros de array flexíveis e melhorou o suporte para o ponto flutuante IEEE 754. Várias dessas funcionalidades já haviam sido implementadas anteriormente como extensões em vários compiladores.

Após a C99, em 2007 outra revisão do padrão C foi iniciada, chamada de C11. Sua publicação oficial aconteceu em 2011. O comitê de padrões C adotou guias para limitar a adição de novas ferramentas que ainda não foram testadas pelas implementações existentes. A C11 implementou várias ferramentas novas, que incluíam tipos macro genéricos, melhor suporte para Unicode, operações atômicas, multi-threading e funções de checagem de limites. Além disso, fez com que bibliotecas existentes na C99 se tornassem opcionais

e melhorou a compatibilidade com a linguagem C++.

Pode-se concluir que a linguagem C é complexa porém pode ser lecionada ao aluno em níveis básicos, tornando possível seu aprendizado por alunos sem conhecimentos prévios de programação. Sua criação minimizou os erros cometidos em programas extensos desenvolvidos em linguagens procedurais usadas anteriormente. Pelos recursos oferecidos pela linguagem, a mesma pode ser utilizada para programação de vários tipos de programas voltados para várias plataformas. [C 2018] [BACKES 2012]

2.2 O modelo HTML

O HTML, do inglês *Hypertext Markup Language* (Linguagem de marcação de hipertexto), é a linguagem de marcação padrão para criação de páginas e aplicações web.

Os documentos HTML são interpretados pelos navegadores web, assim, o papel dos navegadores web é receber páginas HTML de um servidor web e as exibirem para os usuários.

Dentro de um documento HTML existem diversos elementos HTML. Eles são os blocos construtores das páginas. Com eles pode-se adicionar imagens ou objetos interativos na página. O HTML também fornece meios de criar documentos estruturados ao denotar estruturas para texto como cabeçalhos, parágrafos, listas, links e outros. Esses elementos são distribuídos por tags.

Códigos desenvolvidos em HTML aceitam programas escritos em linguagens de script, como o JavaScript, que podem afetar o comportamento e o conteúdo da página web. Ao adicionar o CSS, do inglês *Cascading Style Sheets* (Folha de estilo em Cascata), pode-se definir o estilo e layout da página. Assim, temos uma página web completa que irá realizar tarefas definidas e exibir seu conteúdo de forma adequada. O uso do HTML em conjunto com o CSS e a linguagem de programação JavaScript formam o pilar da rede mundial de computadores. [W3Schools 2018]

2.3 O modelo XML

O modelo XML, do inglês *Extensible Markup Language* (Linguagem de marcação extensível), é utilizado para gerar linguagens de marcação para utilidades específicas. Este modelo define um conjunto de regras para codificar documentos em um formato que pode ser interpretado por máquinas ou humanos. Basicamente, é um modelo simples que fornece uma base na qual modelos mais complexos podem ser construídos.

O XML é um formato de dados textual com um suporte forte fornecido pela Unicode para diferentes linguagens humanas. Mesmo que o design do XML foque em

documentos, a linguagem é muito usada para representar estruturas de dados. O XML pode ser visto em uma estrutura de árvore, como pode-se observar na Figura 1. Em cada um dos nós dessa árvore existem várias Strings. A árvore e as Strings formam a informação sobre o conteúdo de um documento XML. [W3Schools 2018]

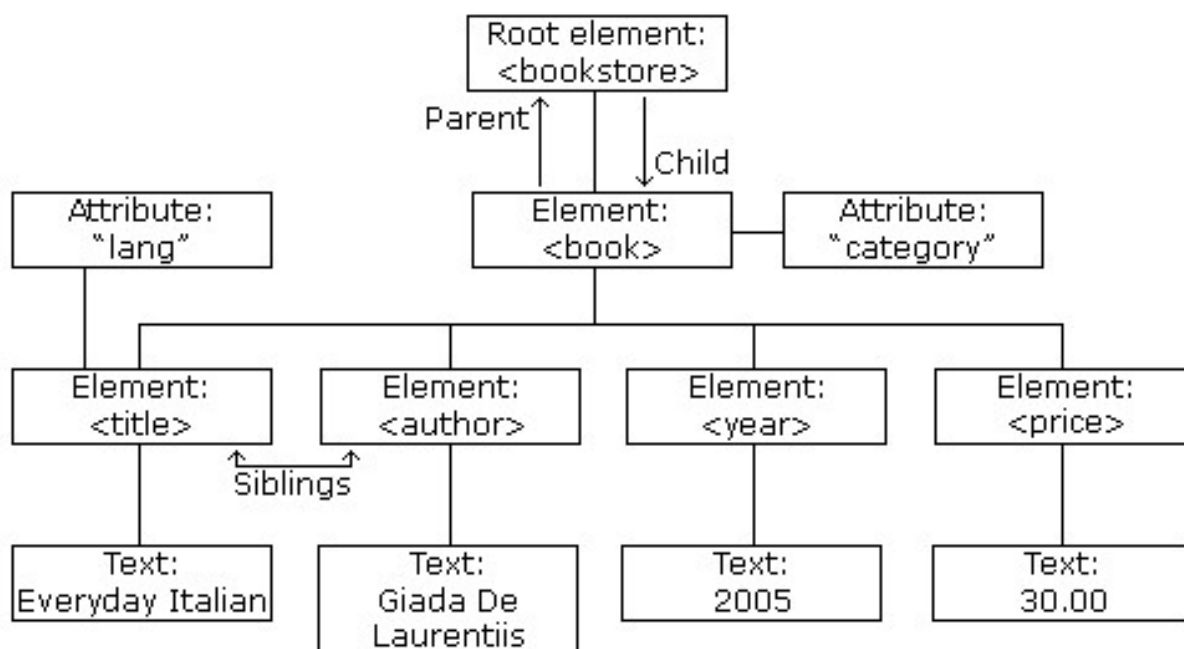


Figura 1 – Estrutura em árvore de um documento XML adaptado de [W3Schools 2018]

3 Atividades Desenvolvidas

3.1 Elaboração do modelo HTML

Nesse trabalho, o HTML foi utilizado como padrão para visualização do conteúdo teórico desenvolvido. Os códigos desenvolvidos na linguagem C foram escritos usando utilizando a IDE CodeBlocks. O site **hilite.me** foi usado com a finalidade de exibir corretamente códigos em linguagem C nas páginas HTML. Nesse site, é possível converter o código em linguagem C e obter o mesmo convertido em HTML como saída, assim o código é exibido na página HTML no mesmo esquema de cores de sintaxe da IDE. Na figura 3 pode-se observar o código que gerou a página HTML exibida na figura 4.

Além do referido site, várias outras tags do HTML também foram usadas. Elas foram usadas para melhor formatar os conteúdos explicativos que acompanhavam cada exemplo de código na linguagem C:

- `<pre>` que define um texto pré formatado;
- `<center>` que centraliza o texto;
- `<h1>` a `<h6>` que definem cabeçalhos;
- `
` que define uma quebra de linha;
- `` para exibição de imagens;
- `` para exibir texto em negrito;
- `<div>` para definir uma divisão ou sessão no documento HTML;
- `<tr>` e `<td>` para elaboração de tabelas;
- `` para agrupar documentos em uma mesma linha;
- `` para definir uma lista não ordenada;
- `` para um item de uma lista;

O uso de algumas dessas tags pode ser observado na figura 2. As tags `<center>` e `<h1>` são usadas no título da página, gerando um título centralizado. Além disso, pode-se observar o uso das tags `
` para quebra de linhas, `` e `` para definir listas e seus itens.

```

<?xml version="1.0" encoding="UTF-8"?>
<aulas>
<topico id="1">
  <texto><![CDATA[<html><center><h1>Comando de atribuição</h1></center>
    <li>O operador "=" (atribuição) armazena o valor ou resultado de uma expressão contida à sua direita na
    variável especificada à sua esquerda!</li><br>
    <li><b>variável = valor ou expressão;</b></li><br>

    <li>Exemplos:
    <font size="2" face="Courier New,Courier, monospace"><b>
    <ul> <font color="blue">int</font> x <font color="red">=</font> <font color="pink">6</font>
    <font color="red">;</font></ul>
    <font color="gray">//O valor 6 é armazenado na variável x!</font></ul>
    <ul><font color="blue">int</font> y <font color="red">=</font> x <font color="red">+</font>
    <font color="pink">2</font><font color="red">;</font> <font color="gray"> //y receberá 6 + 2!</font></ul>
    <ul><font color="blue">float</font> valor <font color="red">=</font> <font color="pink">36.6</font>
    <font color="red">;</font></ul>
    <ul><font color="blue">char</font> letra <font color="red">=</font> <font color="DarkGoldenRod">'a'</font>
    <font color="red">;</font> <font color="gray">//Caracteres são atribuídos entre aspas simples!</font></ul>
    <ul>x <font color="red">=</font> y <font color="red">=</font> <font color="pink">0</font>
    <font color="red">;</font> <font color="gray"> // Lembre-se: A linguagem C suporta múltiplas atribuições!
    </font></ul>
    <li>Podemos atribuir um valor para uma variável no momento ou após sua declaração </li><br>
    <ul><font color="blue">int</font> x <font color="red">=</font><br>
    x<font color="red"> = </font><font color="pink">10</font><font color="red">;</font></ul><br>

    <li> Cuidado com atribuições entre tipos diferentes! </li>
    <li> Exemplo:</li>
    <ul><font color="gray">// Nesse caso, como x é um inteiro, ele não poderá armazenar um valor float.
    </font></ul>
    <ul><font color="gray">// Assim, o que será armazenado em x é apenas o valor inteiro, que nesse caso é 6.
    </font></ul>
    <font size="2" face="Courier New,Courier, monospace"><b>
    <ul> <font color="blue">int</font> x <font color="red">=</font> <font color="pink">6.5</font>
    <font color="red">;</font></ul>

    </ul>
    <li> Erros e alterações no funcionamento do programa poderão decorrer de atribuições entre tipos diferentes!
    </li>
    </font></b>
  </html>]]>
</texto>
</topico>
</aulas>

```

Figura 2 – Exemplo de código em HTML utilizando algumas tags citadas

3.2 Elaboração do modelo XML

O XML foi utilizado em conjunto com o HTML nesse trabalho para organização da estrutura dos documentos. Cada XML armazena várias páginas HTML, através desse recurso, foi possível estruturar os documentos de cada tópico, obtendo assim a organização e estrutura desejadas.

Como pode-se observar na figura 3, a estrutura dos tópicos desenvolvidos foi feita da seguinte forma:

- Aulas: Cobrem cada parte da linguagem, como aula sobre operadores lógicos, aula sobre variáveis e seus tipos;
- Tópico: Cada aula é dividida em tópicos. Por exemplo, a aula sobre variáveis é dividida em tópicos. O primeiro tópico explica o que é uma variável, o segundo explica os tipos que uma variável pode assumir e o terceiro fornece exemplos;
- Texto: É o campo que tem a finalidade de armazenar um único HTML. É dentro dele em que está o HTML referente ao tópico;

Além disso, também é possível observar o uso do comando `<![CDATA[]>`. Esse comando foi usado para indicar que o texto dentro do mesmo não deve ser interpretado como parte da marcação XML. Nesse caso, o texto dentro desse comando é o conteúdo HTML do documento, assim, o mesmo foi utilizado com a finalidade de armazenar o HTML.

3.3 Elaboração dos tópicos

O conteúdo foi desenvolvido com apoio bibliográfico do livro *Linguagem C: Completa e Descomplicada* [BACKES 2012] e o site *Programação descomplicada*¹. O mesmo foi elaborado de forma progressiva, oferecendo inicialmente ao aluno conceitos básicos, como o que é um algoritmo, comandos básicos da linguagem, formato de um programa, dentre outros. Assim, um aluno sem conhecimento prévio de programação pode aprender conceitos básicos, que irão fornecer as noções necessárias a programação na linguagem C. Uma vez que o aluno domine os conceitos iniciais, conteúdos mais avançados são introduzidos.

Cada aula foi desenvolvida de forma a manter o conteúdo de fácil compreensão. O usuário do aplicativo poderá fazer suas aulas de forma rápida e eficiente, uma vez que o conteúdo é oferecido de forma lenta e progressiva, que não demandam grandes fatias de tempo. Além disso, o usuário é levado a entender obstáculos no desenvolvimento de um algoritmo e introduzido às soluções dos mesmos em aulas posteriores.

Alguns tópicos discutem assuntos com maior complexidade que outros. A complexidade do assunto afeta diretamente o aluno, causando dificuldades para a compreensão. Ao dividir tópicos em subtópicos, cada conceito pode ser introduzido de forma lenta e progressiva, facilitando o aprendizado. Além disso, essa divisão ajuda o aluno a melhor compreender como cada tópico posterior complementa o anterior, fazendo com que a fixação do conteúdo se torne mais natural.

¹ <https://programacaodescomplicada.wordpress.com>

Atribuição de Variáveis

Atribuição

- O operador de atribuição '=' é responsável por atribuir um valor para uma variável.
- Forma geral:

```
nome_da_variavel = valor ou expressão;
```

- Exemplo:

```
int main(){  
    int x = 5;  
    int y;  
    y = x + 3;  
    return 0;  
}
```

- A linguagem C também suporta múltiplas atribuições

```
x = y = z = 0;
```

- Dessa forma, as variáveis x, y e z receberão o valor 0

Inicialização de Variáveis

- Podemos inicializar variáveis no momento de sua declaração
- Forma geral:

```
tipo_da_variável nome_da_variável = valor ou expressão;
```

- Exemplo:

```
char ch = 'D';  
int count = 0;  
float pi = 3.141;
```

- Uma variável também pode ser inicializada após a sua declaração
- Exemplo:

```
int num;  
num = 10;
```

- Lembre-se: Até que algum valor seja atribuído para uma nova variável ela terá um valor indefinido

Figura 4 – Página HTML gerada a partir da figura 3

4 Resultados e discussão

Nesse projeto, um conteúdo que será apresentado à um usuário que busca informações sobre a linguagem C foi desenvolvido. Os tópicos foram desenvolvidos de acordo com um modelo. Esse modelo consiste primeiramente na disponibilização do conteúdo seguido de exemplos, como pode ser observado na figura 5. Os exemplos são oferecidos em códigos da linguagem C e esses códigos possuem comentários para melhor compreensão por parte do aluno. Esses comentários explicam o que uma determinada parte do código irá realizar.

Atualmente a teoria conta com os seguintes tópicos:

- Introdução (2 aulas): Fornece conteúdo básico sobre programação, como a estrutura básica de um programa e o uso de comentários no código;
- Tipos e variáveis (7 aulas): Introduz as variáveis da linguagem C, suas declarações e inicializações, o que são palavras-chave e modificadores;
- Leitura e escrita de variáveis (5 aulas): Explica o funcionamento de leitura e escrita de variáveis, sequências de escape, etc;
- Operadores aritméticos (7 aulas): Explica como funciona a atribuição, como usar os operadores aritméticos, operações entre variáveis de diferentes tipos e o cast de variáveis;
- Operadores relacionais (2 aulas): Explica o conceito de verdadeiro e falso em C, operadores lógicos e tabela verdade;
- Operadores lógicos (2 aulas): Explica as limitações dos operadores relacionais (maior que $>$, menor que $<$) e os operadores lógicos;
- Comandos Condicionais (6 aulas): Explica os comandos `if` e `else` e os comandos `switch` e `break`;
- Comandos de Repetição (10 aulas): Explica o conceito de repetições por condições e explica os comandos `while`, `do-while`, `for`, `continue` e `goto`;
- Array (6 aulas): Explica o que é um array e seu uso em geral e como formar matrizes e outros arrays multidimensionais;
- String (5 aulas): Explica o que é uma string e seu uso geral;
- Struct (6 aulas): Explica o conceito de struct, sua declaração e seu uso em geral;

- Funções (9 aulas): Explica o conceito dos vários tipos de funções, seus retornos e seu uso em geral;
- Funções Recursivas (3 aulas): Explica o conceito de recursão matemática e o conceito de recursão em funções;
- Ponteiros (9 aulas): Explica o conceito de ponteiros e suas declarações e seu uso em geral;
- Alocação Dinâmica (7 aulas): Explica o conceito de alocação dinâmica e seu uso em geral;

Tipos Básicos

Tipo

- Todas as variáveis em C possuem um tipo
- O tipo define os valores que a variável pode armazenar
- Cada tipo ocupa uma quantidade diferente de memória
- Existem 4 tipos básicos: **char**, **int**, **float** e **double**

Tipos Básicos

char

- Um byte que armazena o código de um caractere do conjunto de caracteres local
- Sempre ficam entre ‘aspas simples’

```
char UnidadeTemperatura = 'F';  
//pode receber 'C' para Celsius ou 'F' para Fahrenheit  
char opções;  
// pode ser '1', '2', '3'
```

int

- Armazena um número inteiro cujo tamanho depende do processador. Tipicamente de 16 ou 32 bits

```
int NumeroAlunos;  
int Idade;  
int N = 10;
```

Figura 5 – Exemplo de aula com código e comentários

5 Conclusão

Neste projeto foi desenvolvido o conteúdo para um aplicativo para dispositivos móveis que tem a finalidade de ensinar sobre os conceitos da linguagem C. O conteúdo em seu início possui conceitos básicos e novos conteúdos foram introduzidos gradualmente, de acordo com o nível de dificuldade.

Durante a elaboração do trabalho, a maior dificuldade encontrada foi manter o conteúdo simples e objetivo. É importante que o conteúdo seja disponibilizado dessa forma já que o mesmo será estudado via plataformas móveis. Estudantes que buscam aprender com aplicativos esperam aprender o máximo utilizando a menor quantidade de tempo de forma imediata.

As disciplinas que lecionaram a linguagem C durante o curso foram de grande ajuda para a realização do trabalho. Além do conteúdo fornecido por elas, foi possível observar as dificuldades dos alunos e assim desenvolver aulas mais adequadas.

Alguns tópicos abordam assuntos mais complexos e por isso exigem uma discussão mais profunda e cuidadosa. Esses tópicos serão desenvolvidos em trabalhos futuros. Eles são:

- Funções: Explica o conceito de funções, seus componentes e parâmetros, dentre outros conceitos;
- Funções Recursivas: Explica o conceito de recursão matemática e de recursão em funções;
- Ponteiros: Explica o que é um ponteiro, seus usos, limitações e cuidados para evitar erros em seu uso;
- Alocação dinâmica: Explica o que é alocação dinâmica e seus usos;
- Arquivos: Explica o que são arquivos, seus tipos e como os criar, modificar e ler;

Referências

BACKES, A. *Linguagem C: Completa e Descomplicada*. [S.l.]: Elsevier Editora Ltda., 2012. ISBN 9788535269642. Citado 2 vezes nas páginas 9 e 13.

C, L. *Learn C Programming*. 2018. Disponível em: <<https://www.programiz.com/c-programming>>. Citado na página 9.

W3SCHOOLS. *HTML Tutorial*. 2018. Disponível em: <<https://www.w3schools.com/html/default.asp>>. Citado na página 9.

W3SCHOOLS. *XML Tutorial*. 2018. Disponível em: <<https://www.w3schools.com/xml/default.asp>>. Citado na página 10.

W3SCHOOLS. *XML Tutorial*. 2018. Disponível em: <https://www.w3schools.com/xml/xml_tree.asp>. Citado 2 vezes nas páginas 4 e 10.