



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS - CMPF
BACHARELADO EM CIÊNCIA E TECNOLOGIA

GABRIEL CALDAS BARROS E SÁ

**DESENVOLVIMENTO *FRONT-END*: MÓDULO ALUNO DO *SOFTWARE*
*EDUCACIONAL LOOP ACADEMIC***

Pau dos Ferros - RN

2019

GABRIEL CALDAS BARROS E SÁ

**DESENVOLVIMENTO *FRONT-END*: MÓDULO ALUNO DO SOFTWARE
EDUCACIONAL *LOOP ACADEMIC***

Monografia apresentada como requisito parcial para obtenção do título de Bacharel em Ciência e Tecnologia pela Universidade Federal Rural do Semi-Árido, UFERSA.

Orientadora: Profa. Dra. Náthalee Cavalcanti de Almeida Lima

Pau dos Ferros - RN

2019

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S111d Sá, Gabriel Caldas Barros e.
DESENVOLVIMENTO FRONT-END: MÓDULO ALUNO DO
SOFTWARE EDUCACIONAL LOOP ACADEMIC / Gabriel
Caldas Barros e Sá. - 2019.
59 f. : il.

Orientadora: Náthalee Cavalcanti de Almeida
Lima.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2019.

1. Desenvolvimento web. 2. Software
Educativo. 3. Programação. 4. Front-End. 5.
Informática na Educação. I. Cavalcanti de Almeida
Lima, Náthalee, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

GABRIEL CALDAS BARROS E SÁ

**DESENVOLVIMENTO FRONT-END: MÓDULO ALUNO DO SOFTWARE
EDUCACIONAL LOOP ACADEMIC**

Monografia apresentada à Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciência e Tecnologia.

Defendida em: 31 / 07 / 2019.

BANCA EXAMINADORA

Náthalee Cavalcanti de Almeida Lima

Náthalee Cavalcanti de Almeida Lima, Profa. Dra. (UFERSA)
Presidente

Jarbele Cássia da Silva Coutinho

Jarbele Cássia da Silva Coutinho, Profa. Ma. (UFERSA)
Membro Examinador

Thiago Pereira Rique

Thiago Pereira Rique, Prof. Me. (UFERSA)
Membro Examinador

Dedico este trabalho a meus amados e queridos pais, em especial à minha mãe, que sempre me deu forças e me apoiou mesmo à distância, sendo a pessoa que me inspira a seguir em frente e nunca desistir dos meus sonhos.

“O trabalho duro vence o dom natural.”
(Rock Lee)

Agradecimentos

Acima de tudo, agradeço a Deus por me dar forças para enfrentar as dificuldades e continuar dia após dia nessa dura batalha que é a vida, por permitir que fosse capaz de conquistar coisas que eu jamais imaginei alcançar. Obrigado, Pai, por estar sempre ao meu lado.

Agradeço, em especial, a minha querida professora orientadora, profa. Dra. Náthalee Cavalcanti, que foi muito importante na minha decisão de seguir na área da Engenharia da Computação. Por todas as lições, pela paciência e grande ajuda no desenvolvimento deste trabalho, que é algo totalmente novo para mim. Obrigado por ser além de uma grande professora, uma grande pessoa!

Não posso deixar de agradecer a toda a equipe do LABIE, discentes e docentes, que me acolheu e participou ativamente na elaboração desta monografia, principalmente às professoras Dra. Laysa Mabel e Me. Jarbele Coutinho, assim como a profa. Náthalee, que também faz parte do grupo, que sempre estiveram dispostas a me ajudar quando eu precisei.

Deixo meu agradecimento ao aluno Dyego Magno por ter começado este belo projeto, ao qual eu estou dando continuidade, e por ter se colocado à disposição para assistir-me, bem como os demais discentes do LABIE.

Com grande carinho, agradeço ao professor Dr. Hidalyn Theodory, que me ajudou incontáveis vezes desde que ingressei na universidade e que demonstra enorme afeição pela profissão. Me sinto honrado de ter sido seu aluno!

Agradeço, também, aos amigos que fiz ao ingressar na UFERSA, que fizeram esse duro caminho se tornar um pouco mais leve com todos os momentos de diversão e também de estudo, em especial, a Thiago Aquino, José Lira (o saudoso Ben 10), Felipe William, Fagner Rezende, Yam Souto e José Luan.

A minha esposa e ao meu pai, que mesmo com todas as dificuldades da vida, sempre esteve disposto a me ajudar da forma que ele pôde.

Por fim, agradeço à pessoa mais especial em minha vida: minha mãe, Cida, a razão da minha existência, a mulher mais forte e perfeita que eu conheço, a mulher que daria a vida por mim. Sem ela eu não teria chegado tão longe e não seria capaz de trilhar meus sonhos. Foi ela quem sempre me incentivou a estudar, a seguir em

frente, me guiou nos caminhos fatigantes e me apoiou em minhas decisões. Foi ela quem me fez ser o homem que sou hoje, e espero me tornar pelo menos 10% da pessoa que ela é. Obrigado, mainha, eu te amo!

RESUMO

Uma grande parcela dos estudantes da área de computação e tecnologias enfrenta dificuldades nas disciplinas de programação introdutória, e isso reflete na falta de preparo para o mercado de trabalho. Para amenizar esse problema, que também ocorre nas mais diversas áreas do ensino, *Softwares* Educacionais têm sido utilizados como meio de auxiliar tanto o aluno quanto o professor no processo de ensino-aprendizagem. Diante disso, este trabalho busca desenvolver o *Client Side* das principais funcionalidades do módulo Aluno do Software Educacional *web* nomeado *Loop Academic*, desenvolvido em HTML5, CSS, *JavaScript* e outras ferramentas, cujo alvo são os estudantes dos componentes curriculares que envolvem programação introdutória. Para tanto, a metodologia utilizada consiste na escolha das ferramentas de desenvolvimento *web*, estudo do protótipo já existente do módulo Aluno, seleção das principais funcionalidades e, por fim, a codificação do *front-end*, resultando em um sistema estático que servirá como base para trabalhos complementares futuros que visem desenvolver os demais componentes do *Software*.

Palavras-chave: Desenvolvimento *web*. *Software* Educacional. Programação. *Front-End*. Informática na Educação.

ABSTRACT

A large amount of students in computer and technology areas face many difficulties in introductory programming subjects, and this reflects on the lack of preparation to enter in the job market. To reduce this problem, which also occurs in many others studying areas, Educational Softwares have been used to assist both the students and the teachers in the teaching-learning process. With this in mind, this research focuses on the development of the Client Side and the main functionalities of the Student module of the educational web software named Loop Academic, built using HTML5, CSS, JavaScript and other tools, whose targets are the students of the subjects that involve introductory computer programming. Therefore, the methodology applied consists in the choice of the web development tools, study of the already existing prototype of the Student module, the selection of the main functionalities and, at last, coding the front-end, resulting in a static system that will serve as a base for future works that aim to develop the other components of the system.

Keywords: Web development. Educational Software. Programming. Front-End. Information technology on Education.

LISTA DE FIGURAS

Figura 1. Exemplo de documento HTML.....	24
Figura 2. Resultado do trecho HTML da Figura 1.....	24
Figura 3. Exemplo folha de estilo externa de CSS	26
Figura 4. Representação gráfica do sistema de grades do <i>Bootstrap</i> 4 ...	29
Figura 5. Exemplo de código <i>JavaScript</i> escrito através da biblioteca <i>jQuery</i>	30
Figura 6. Metodologia do desenvolvimento do SE	33
Figura 7. Tela inicial do <i>Loop Academic</i>	40
Figura 8. Trecho do HTML da página inicial	41
Figura 9. Trecho do CSS referente a página inicial	41
Figura 10. Perfis de Usuário	42
Figura 11. Cadastro do Aluno - <i>Loop Academic</i>	43
Figura 12. Tela de boas-vindas	43
Figura 13. HTML - <i>Modal</i> do <i>Bootstrap</i>	44
Figura 14. Função do <i>JavaScript</i>	45
Figura 15. Menu Principal do <i>Loop Academic</i>	45
Figura 16. Lista de Exercícios	46
Figura 17. Exercício da Lista de Exercícios	47
Figura 18. Material de Apoio	47
Figura 19. Eventos controlados via <i>JavaScript</i>	48
Figura 20. Exemplo de Material de Apoio	48
Figura 21. HTML referente ao Material de Apoio	49
Figura 22. Emblemas	49
Figura 23. Tela de Desempenho	50

LISTA DE GRÁFICOS

Gráfico 1: Relação das linguagens mais populares dentre os usuários profissionais do <i>StackOverflow</i>	27
Gráfico 2: Tipos de desenvolvedores usuários do <i>StackOverflow</i>	27

LISTA DE TABELAS

Tabela 1. Critérios de avaliação de linguagens e as características que as afetam.....	34
Tabela 2. Avaliação das linguagens HTML, CSS e <i>JavaScript</i> : critérios técnicos.....	35
Tabela 3. Avaliação das linguagens HTML, CSS e <i>JavaScript</i> : critérios gerais.....	35
Tabela 4. Principais características de algumas IDEs para desenvolvimento web.....	36

LISTA DE ABREVIATURAS E SIGLAS

AJAX	<i>Asynchronous JavaScript And XML</i>
API	<i>Application Programming Interface</i>
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
HTML	<i>Hyper Text Markup Language</i>
IDE	<i>Integrated Development Environment</i>
JIT	<i>Just-in-Time</i>
LABIE	Laboratório de Pesquisa em Informática na Educação
MOJO	Modo de Integração com os Juízes <i>Online</i>
POO	Programação Orientada a Objetos
PPV	Plataforma de Programação Visual
REDU	Rede Social Educativa
SE	<i>Software</i> Educacional
SO	Sistema Operacional
SVG	<i>Scalable Vector Graphics</i>
UFERSA	Universidade Federal Rural do Semi-Árido
UFScar	Universidade Federal de São Carlos
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1. INTRODUÇÃO	15
1.1. OBJETIVOS	17
1.1.1. OBJETIVO GERAL	17
1.1.2. OBJETIVOS ESPECÍFICOS	17
1.2. ORGANIZAÇÃO DO TRABALHO	18
2. REFERENCIAL TEÓRICO	19
2.1. INFORMÁTICA NA EDUCAÇÃO	19
2.2. SOFTWARE EDUCACIONAL	20
2.3. LOOP ACADEMIC	22
2.4. DESENVOLVIMENTO WEB	22
2.4.1. FRONT-END	23
2.4.1.1. HTML	23
2.4.1.2. CSS	25
2.4.1.3. JAVASCRIPT	26
2.4.1.4. FRAMEWORK BOOTSTRAP	28
2.4.1.5. JQUERY	29
3. TRABALHOS RELACIONADOS	31
4. METODOLOGIA	33
4.1. ESCOLHA DAS FERRAMENTAS	33
4.2. ESTUDO DO PROTÓTIPO DO MÓDULO ALUNO	37
4.3. SELEÇÃO DAS PRINCIPAIS FUNCIONALIDADES	38
4.4. CODIFICAÇÃO	38
5. RESULTADOS E DISCUSSÕES	40
5.1. DESENVOLVIMENTO DO FRONT-END	40
6. CONSIDERAÇÕES FINAIS	51
REFERÊNCIAS	52
ANEXO A - Código HTML	57

1. INTRODUÇÃO

Uma parte significativa dos estudantes entra para o mercado de trabalho ainda, de certa forma, despreparada, e isso é refletido na crescente insatisfação da indústria perante este fato, o que instiga as instituições de ensino, especialmente superior, a se reinventarem e procurarem novos métodos para tornar o processo de ensino-aprendizagem mais eficaz (CALLAHAN; PEDIGO, 2002).

Em especial no ensino de programação de computadores em cursos de tecnologia, como Engenharia de *Software*, Engenharia da Computação, Ciência da Computação, entre outros, os chamados *Softwares* Educacionais (SEs) têm sido usados como forma de auxílio no processo de aprendizado do aluno. De acordo com Jucá (2006), os SEs são quaisquer programas inseridos e utilizados de forma adequada na educação, tanto no ensino quanto na aprendizagem. Oliveira et al. (2001) classificam, ainda, os SEs de duas formas: aplicativos, que não possuem propósito educativo mas podem ser utilizados para este fim, como editores de texto; e os educativos, que são desenvolvidos especificamente para auxiliar o ensino-aprendizagem.

O uso de SEs no Brasil foi inicialmente instigado pelo que ocorria em países como França e Estados Unidos da América, onde o uso de ferramentas computacionais no meio acadêmico já estava sendo aplicado. Então, na década de 70, a Universidade de São Carlos (UFScar) foi pioneira na inserção de *softwares* em seus mecanismos de ensino e estudo (PENHA, 2015). Tendo em vista o cenário tecnológico de rápida evolução, que exige cada vez mais eficácia e eficiência no aprendizado, a utilização de SEs de modo a complementar o ensino convencional tem se mostrado como uma saída prática e útil. De acordo com Jucá (2006), essas tecnologias servem como amparo no processo de ensino-aprendizagem e os professores universitários devem ser capazes de avaliar em que momento devem utilizá-las e quais os benefícios que elas podem trazer para a construção do conhecimento.

O conhecimento básico em informática tem se tornado extremamente necessário na sociedade atual. Por isso, é importante que o ensino seja capaz de fornecer as competências específicas para que indivíduo possa ter a percepção e

interação com a evolução tecnológica do cotidiano (JUCÁ, 2006). Nas disciplinas de programação introdutória isso se mostra ainda mais relevante, tendo em vista que, de acordo com Gomes (2010), há um elevado número de casos de insucesso nas mesmas em todo o mundo, independente da linguagem de programação utilizada. Isso é perceptível também a partir da análise dos dados obtidos na pesquisa realizada por Holanda (2018), onde quase metade dos alunos das disciplinas de Algoritmos, Laboratório de Algoritmos e Informática Aplicada declararam possuir dificuldades em lógica de programação e entendimento da sintaxe.

A principal dificuldade que os estudantes enfrentam nas disciplinas de programação está no raciocínio lógico, que também se torna um grande desafio para os professores, que precisam lidar com turmas de alunos com diferentes níveis de conhecimento, além da defasagem de conteúdos básicos, como a matemática, que está intimamente ligada à lógica (SOARES; CARVALHO, 2017).

O ensino-aprendizagem de programação de computadores é uma das maiores dificuldades enfrentadas dentro dos cursos de tecnologia (CASPERSEN e KOLLING, 2009). Devido a isso, a utilização de SEs se mostra benéfica e útil, pois, de acordo com Mercado (2002), um SE é um recurso didático-pedagógico que serve de amparo ao professor na missão de ensinar e no processo de aprendizagem do aluno. Lima et al. (2012) diz que o número de SEs nacionais tem crescido devido às constantes inovações tecnológicas e a percepção de que o uso do computador influencia positivamente a educação. Além disso, os SEs podem ser desenvolvidos para inúmeras plataformas e sistemas operacionais de acordo com o foco e necessidade, o que os torna bastante flexíveis.

Segundo Caritá, Sanches e Padovan (2011), o novo paradigma da sociedade, chamado de Sociedade da Informação, se baseia no uso da tecnologia como mediadora da obtenção e difusão do conhecimento. Portanto, saber usufruir dessas tecnologias voltadas a educação é uma necessidade real das instituições de ensino atualmente, pois cada geração se torna mais conectada (PENHA, 2015).

Tendo em vista os fatos evidenciados anteriormente, pode-se constatar a importância do desenvolvimento de SEs voltados à programação introdutória de computadores. Desse modo, ao perceber a dificuldade enfrentada por muitos estudantes nas disciplinas de programação introdutória e os benefícios que um SE

pode trazer tanto aos estudantes quanto aos professores, este trabalho visa minimizar essa situação-problema ao desenvolver um meio, neste caso, o *front-end* de uma plataforma *online* caracterizada como um SE do tipo educativo. O sistema deve servir de auxílio tanto para alunos quanto para professores dos componentes curriculares que envolvem programação introdutória de computadores ao facilitar e dinamizar o ensino-aprendizagem dos mesmos.

1.1. OBJETIVOS

Nesta seção, são apresentados o objetivo geral e os objetivos específicos desta pesquisa.

1.1.1. OBJETIVO GERAL

Este trabalho tem o propósito de realizar o desenvolvimento *web* no eixo *client side*, também chamado de *front-end*, do módulo Aluno presente na documentação e protótipo do SE denominado *Loop Academic*, apresentado no trabalho realizado pelo discente Dyego Magno Oliveira Souza¹, cujo público-alvo são os estudantes de disciplinas voltadas à programação introdutória de computadores.

1.1.2. OBJETIVOS ESPECÍFICOS

A fim de atingir o objetivo geral, os objetivos específicos foram definidos e estão listados a seguir:

- I. Promover a análise da importância de um SE e sua influência no processo de ensino-aprendizagem tanto de modo geral quanto aplicado ao ensino de programação introdutória;
- II. Realizar um estudo das principais linguagens e ferramentas voltadas ao desenvolvimento *web*, sendo algumas delas o HTML, CSS e o *JavaScript*;
- III. Definir tanto as linguagens (de marcação, *script* e afins) quanto o Ambiente Integrado de Desenvolvimento (IDE) e as ferramentas externas (como *Frameworks* e *Libraries*), a fim de utilizá-las para desenvolver o *front-end* do

¹ Souza, Dyego (2019)

SE de acordo com o protótipo já existente e, também, de acordo com o objetivo específico II;

- IV. Realizar uma análise minuciosa das funcionalidades presentes no módulo Aluno a partir do estudo da documentação e do protótipo do mesmo, com o propósito de selecionar as principais para que estas tenham seu *front-end* desenvolvido.

1.2. ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado da seguinte forma: No capítulo 2, o referencial teórico é disposto e dividido em três partes de maior relevância: a seção 2.1 analisa a informática na educação; a 2.2 explana sobre os SEs em geral e a 2.3 faz uma breve explanação sobre o desenvolvimento *web*, sendo este último ainda dividido em seis subseções: a 2.3.1 tem foco no *front-end*, a 2.3.1.1 apresenta o HTML, já as subseções 2.3.1.2 e 2.3.1.3 tratam do CSS e do *JavaScript*, respectivamente, enquanto a 2.3.1.4 explana sobre o *framework Bootstrap* e a 2.3.1.5, sobre a *library jQuery*. O capítulo 3 trata sobre os trabalhos relacionados à este, enquanto o 4 apresenta a metodologia e está dividido em quatro seções: a 4.1 justifica a escolha das ferramentas, a 4.2 expõe o estudo do protótipo do módulo Aluno, a 4.3 apresenta as principais funcionalidades selecionadas e a seção 4.4 trata da do processo de codificação do sistema. O capítulo 5 expõe os resultados e discussões obtidos durante este trabalho onde, na seção 5.1, são exibidos os resultados do desenvolvimento do *front-end*. Por fim, o capítulo 6 dispõe sobre as considerações finais e propostas para trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo dispõe sobre a revisão e análise do que a literatura apresenta a respeito dos três principais temas abordados nesta pesquisa, sendo eles Informática na Educação, *Softwares* Educacionais e Desenvolvimento *Web*.

2.1. INFORMÁTICA NA EDUCAÇÃO

De forma geral, a educação deve ser constituída da busca do aluno pela informação, na sua utilização para resolução de problemas e da transformação dessa informação em conhecimento, e não mais apenas uma instrução que o professor transmite ao aluno (VALENTE, 2014). Silva e Serafim (2016, p. 72) consideram que “os meios digitais têm um enorme potencial para o ensino, mas é difícil realizar esse potencial se eles são considerados apenas tecnologias e não formas de cultura e comunicação”, e, ainda, que “[...] as tecnologias não substituem o professor, porém podem possibilitar mudanças em sua metodologia”.

É importante frisar que a tecnologia sempre foi importante para a sociedade, especialmente como forma de divulgação de conhecimento. Koile e Singer (2006) destacam que o uso de tecnologia na educação melhora significativamente o aprendizado do aluno. Já Brito Junior (2016) realça que “a combinação de Educação e Tecnologia resulta em oportunidades de crescimento no processo de ensino-aprendizagem”, e, também, que as tecnologias de informação e comunicação são agregadas à educação com o objetivo de melhorar significativamente o processo de ensino-aprendizagem.

De fato, o desenvolvimento da ciência e, conseqüentemente, da tecnologia, tem gerado mudanças na interação entre a escola e o aluno, apresentando a necessidade da implantação de novas mídias na educação, onde essas ferramentas devem servir como facilitadoras do aprendizado (DIOGINIS et al., 2015), ou seja, o uso desses instrumentos, especialmente aqueles voltados à informática, tem se tornado cada vez mais útil e necessário no meio educacional.

Apesar disso, é imprescindível a assistência do professor no sentido de criar ambientes de aprendizado ideais para o aluno, utilizando, além dos métodos

convencionais, a informática, pois a tecnologia por si só não gera conhecimento, mas sim o modo como esses recursos são utilizados (KOCH, 2013). No entanto, o processo de ensino-aprendizagem depende de forma majoritária do aluno, pois, de acordo com Moran (2000, p-17-18):

As mudanças na educação dependem também dos alunos. Alunos curiosos e motivados facilitam enormemente o processo, estimulam as melhores qualidades do professor, tornam-se interlocutores lúcidos e parceiros de caminhada do professor-educador. Alunos motivados aprendem e ensinam, avançam mais, ajudam o professor a ajudá-los melhor.

Com relação ao ensino de programação introdutória, que é o foco do desenvolvimento do SE deste trabalho, e que geralmente é ofertada nos períodos iniciais dos cursos de tecnologia, computação e engenharias, boa parte dos alunos possui maior dificuldade de aprendizado nos conceitos iniciais, como lógica, o que gera, então, um alto grau de reprovações e desistências (PRIETCH e PAZETO, 2010). Além disso, pelo fato de que essas disciplinas possuem conteúdos acumulativos, se os alunos não se dedicarem desde o início, isso aumenta a possibilidade de que os mesmos não consigam acompanhar o curso e, por conseguinte, reprovem ou desistam (RAPOSO; DANTAS, 2016).

Ainda de acordo com Raposo e Dantas (2016), os professores precisam buscar formas de despertar o interesse dos alunos pelas disciplinas de programação, fazendo com que os mesmos possam manter um ritmo constante de estudos. Esse objetivo pode ser atingido justamente a partir do uso da informática na educação. No entanto, Brito Junior (2016) ressalta que os profissionais da educação precisam superar grandes desafios referentes ao uso desses recursos, como, por exemplo, os SEs, no âmbito da educação tradicional brasileira.

2.2. SOFTWARE EDUCACIONAL

Um SE é um recurso didático-pedagógico com a finalidade de auxiliar o ensino-aprendizagem e que é usado da melhor forma quando há a mediação do professor, sendo, então, um meio de ensino (BRITO JUNIOR, 2016). Esse recurso serve de apoio não só para o aluno como também para o professor. O processo de

adaptação de muitas pessoas para tais recursos é um ponto negativo, porém a diminuição da curva de aprendizado e a maior abrangência de informações são aspectos positivos do SE, e, portanto, contribuem positivamente com o processo de aprendizagem (PENHA, 2015).

Um fato que atesta a vantagem de se utilizar o SE na educação é que os jovens aprendem melhor quando o ambiente é condizente com o seu cotidiano (FIALHO; MATOS, 2010, p.123). No entanto, para que o SE seja, de fato, eficaz, ele deve ser de fácil utilização, ser capaz de sustentar a atenção do usuário e ser de simples compreensão, dentre outros fatores (TAVARES, 2017).

Para que um *software* seja considerado educacional, basta que seja inserido no meio em questão - desde que atenda às necessidades do contexto - mesmo que não tenha sido projetado com tal objetivo (TEIXEIRA; BRANDÃO, 2003). Em complemento, Oliveira, Menezes e Moreira (2001) consideram que aqueles SE feitos com o objetivo específico de auxiliar no processo de aprendizagem são chamados de *softwares* educativos, que são um subgrupo dos SEs.

Os SEs podem, ainda, ser classificados conforme seu foco e usabilidade, podendo ser *softwares* de exercício, *softwares* de simulação, aplicativos, jogos educativos, tutoriais, *softwares* de linguagem de programação e *softwares* de investigação, como dicionários (TAVARES, 2017).

Abaixo, são descritas as características de cada um:

- A. **Softwares de exercício:** são aqueles que apresentam questões e recebem respostas do usuários, avaliando seu desempenho;
- B. **Softwares de simulação:** permitem que o aluno possa participar de atividades por meio de simulações do real;
- C. **Aplicativos:** não são desenvolvidos com o objetivo educacional, mas podem ser usados para tal;
- D. **Jogos educativos:** permitem que o usuário aprenda de forma lúdica e dinâmica;

- E. **Tutoriais:** são *softwares* que ensinam através da apresentação de informações e instruções ao aluno;
- F. **Softwares de linguagem de programação:** são aqueles que possibilitam a criação de novos *softwares* sem grandes conhecimentos de programação;
- G. **Softwares de investigação:** *Softwares* que permitem ao usuário encontrar informações, como dicionários.

De forma geral, os SEs podem ser desenvolvidos para inúmeras plataformas de acordo com as necessidades do meio: *mobile*, *desktop* e *web*. O sistema deste trabalho trata de uma plataforma *web*.

2.3. LOOP ACADEMIC

O SE denominado *Loop Academic*, de acordo com Souza, Dyego (2019), é uma plataforma *web* ainda em desenvolvimento cujo foco é auxiliar o processo de ensino-aprendizagem das disciplinas de programação introdutória em cursos do nível superior, tendo em vista que a maioria dos estudantes dos cursos de computação e afins apresentam dificuldade nessas disciplinas. O sistema, que está em processo de desenvolvimento, é composto por três módulos: Módulo Aluno, destinado aos discentes de disciplinas de programação introdutória; módulo Professor, voltado aos docentes das mesmas disciplinas; e o módulo Monitor, designado para os monitores de programação.

No trabalho realizado por Souza, Dyego (2019), foi executado o processo de elicitação de requisitos a partir da aplicação de *surveys* com alunos principiantes em programação, a prototipação e, por fim, a avaliação da usabilidade do módulo Aluno; enquanto este trabalho trata do desenvolvimento do *front-end* do mesmo módulo.

2.4. DESENVOLVIMENTO WEB

A internet tem se expandido de forma exponencial e isso causa um impacto significativo em diversos setores, inclusive na educação, e essa popularização se deve a facilidade que o usuário tem de obter acesso à informação e a criação de conteúdo (GINIGE; MURUGESAN, 2001). Um sistema *web* se caracteriza por parte

do processamento da aplicação ocorrer em um computador remoto, denominado Servidor (MILLETO; BERTAGNOLLI, 2014, p. 4).

Entretanto, existem duas etapas do desenvolvimento *web*: o *front-end*, voltado a interação do usuário, onde seus processos são realizados pelo próprio navegador; e o *back-end*, no qual os processos são realizados pelo servidor. Cada etapa exige diferentes modos de construção e linguagens.

A partir da análise consoante os critérios listados no capítulo 4, foram escolhidas a linguagem de marcação HTML5, a linguagem de estilo CSS e a linguagem de *script* JavaScript, bem como a IDE (Integrated Development Environment) *Visual Studio Code* como ambiente de desenvolvimento, o *framework* *Bootstrap* e a *library* *jQuery*, que serão detalhadas nas próximas subseções.

2.4.1. FRONT-END

Como já dito anteriormente, o processo de desenvolvimento de páginas ou sistemas *web* geralmente envolve duas fases: o *front-end* e o *back-end*. De acordo com Fernandes (2017), o *front-end* é voltado principalmente a parte da interface da página, responsável por exibir e receber informações (que podem ser tratadas e armazenadas através do *back-end*), levando em consideração conceitos da Engenharia de *Software* como a usabilidade. Além disso, esse eixo, também chamado de *Client-Side*, é interpretado diretamente no navegador do usuário, e isto é um fator técnico que o diferencia do *Server-Side*, que é executado em um servidor.

Como o *front-end* é voltado principalmente aos aspectos gráficos, as principais linguagens utilizadas em seu desenvolvimento são aquelas capazes de manipular interfaces, componentes, interações e eventos de uma página *web*, sendo elas o HTML, CSS e o *JavaScript*.

2.4.1.1. HTML

A linguagem de marcação *HyperText Markup Language* (HTML), que atualmente está na versão 5, é a linguagem padrão para criação de páginas *web* e descreve a estrutura das mesmas através do uso de *tags*, que são elementos que “informam” ao navegador como cada informação deve ser mostrada, por exemplo, como parágrafos, links e cabeçalhos (W3SCHOOLS, 2019a). A Figura 1 representa

um exemplo simples da utilização do HTML, onde a instrução `<!DOCTYPE>` especifica ao navegador a versão HTML do documento, enquanto as tags `<head>` e `<body>` definem a “cabeça” e “corpo” da página, respectivamente. O `<title>` é usado para determinar o título que é mostrado na aba de um navegador *web*, enquanto as tags `<h2>` e `<p>` são elementos de texto (cabeçalho e parágrafo), e devem sempre ser posicionadas dentro do `<body>`.

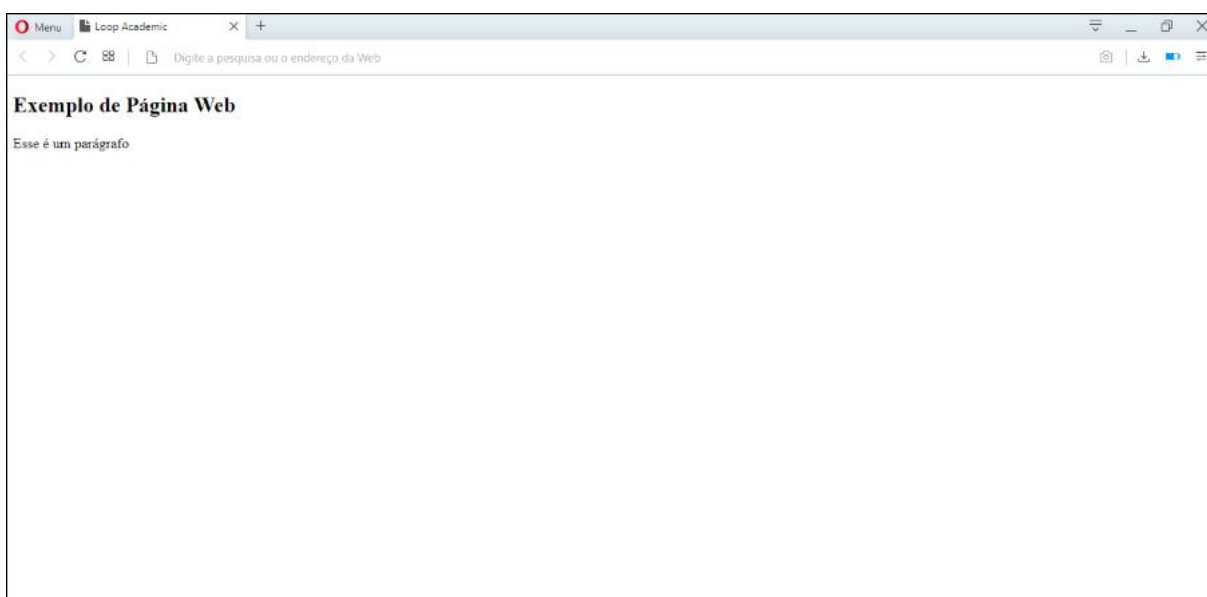
Figura 1. Exemplo de documento HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Loop Academic</title>
5   </head>
6   <body>
7     <h2>Exemplo de Página Web</h2>
8     <p>Esse é um parágrafo</p>
9   </body>
10 </html>
```

Fonte: Próprio autor

Um navegador é capaz de interpretar um código HTML. A Figura 2 apresenta a tela que é exibida pelo *browser* a partir do código da Figura 1.

Figura 2. Resultado do trecho HTML da Figura 1



Fonte: Próprio autor

De acordo com Silva (2016), o HTML surgiu na década de 90 como um documento utilizado para dispor o conteúdo na tela do navegador, e que, na versão 5, além de ter boa compatibilidade com *browsers* antigos, apresentou uma série de recursos antes indisponíveis e que tornam a ferramenta mais poderosa para a construção de páginas *web*. Abaixo, alguns desses recursos estão listados:

- Elementos gráficos: `<canvas>`, `<svg>`;
- Atributos de formulários, como *date*, *number* e *range*;
- *Tags* de elementos semânticos, como `<header>` e `<footer>`.

2.4.1.2. CSS

A partir da versão 3.2 do HTML, alguns elementos de estilização do conteúdo foram adicionados à linguagem, e isso gerou uma certa dificuldade para os desenvolvedores devido à quantidade de informação que deveria ser adicionada à cada página, o que tornava o desenvolvimento lento e caro. Para sanar este problema, a *World Wide Web Consortium* (W3C) desenvolveu uma linguagem de estilos de cascata, ou, do inglês, *Cascading Style Sheets* (CSS), cuja função é descrever a forma e estilo que os elementos e conteúdos do HTML devem ser exibidos na tela do navegador (W3SCHOOLS, 2019b).

No geral, o CSS, que atualmente está na versão 3, é capaz de modificar as cores e os efeitos visuais das páginas *web* (SILVA, 2016), tais como cor do texto, tamanho da fonte, aplicar efeitos de gradiente, definir margens e até mesmo criar animações. O CSS pode ser escrito de três formas:

- **Estilo *Inline*:** isto é, dentro de um único elemento do documento HTML através do atributo *style*, modificando os atributos apenas da *tag* a qual está inserido;
- **Folha de Estilo Interna:** também sendo escrito no documento HTML, mas dentro da *tag style*, sendo capaz de modificar os atributos de toda a página;
- **Folha de Estilo Externa:** escrito em um arquivo externo que contém apenas o CSS, dessa forma é possível alterar a aparência de todas as páginas que fizerem referência à folha de estilo.

A Figura 3 exemplifica um trecho de uma folha de estilos, onde a propriedade *background-color* indica a cor do seletor “*body*”, que representa a *tag* de mesmo nome de um documento HTML. Já para o seletor “*p*”, também referente ao HTML, a propriedade *font-family* determina o tipo de fonte, enquanto *font-size* define o tamanho da fonte e *color*, sua cor.

Figura 3. Exemplo folha de estilo externa de CSS

```
1 body {  
2   background-color: lightgray;  
3 }  
4  
5 p {  
6   font-family: 'Times New Roman', Times, serif;  
7   font-size: 16px;  
8   color: brown;  
9 }
```

Fonte: Próprio autor

2.4.1.3. JAVASCRIPT

O *JavaScript* é uma linguagem de *scripts* multiparadigma que pode ser interpretada ou compilada *Just-in-Time* (JIT). Oficialmente, seu nome é *ECMAScript*, porém a marca registrada pela *Oracle*² leva o nome de *JavaScript* (MDN, 2019). É interessante destacar que, embora tenham nomes semelhantes, *Java* e *JavaScript* são linguagens totalmente diferentes, tanto na sintaxe quanto na aplicação.

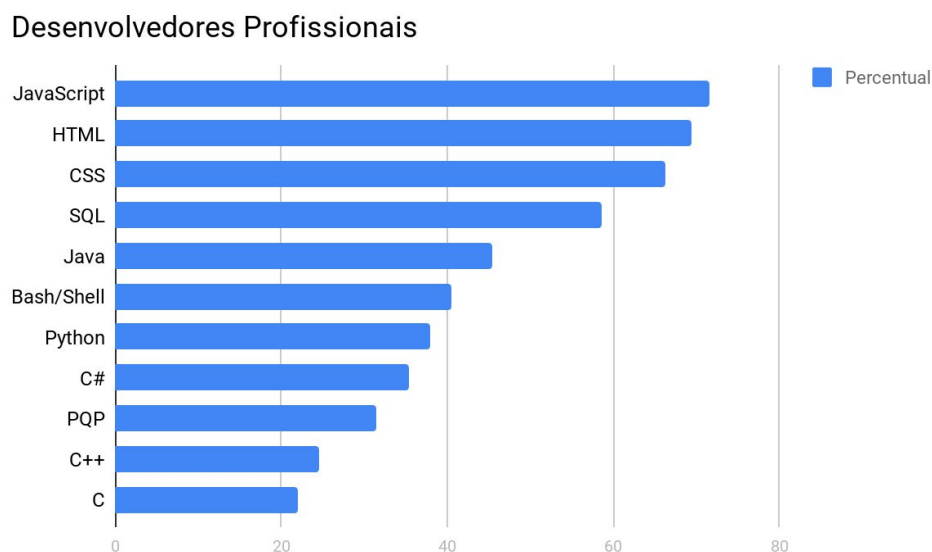
Essa linguagem é de grande importância para o processo de desenvolvimento *web* pois permite criar páginas interativas que não são possíveis apenas com HTML e CSS ao adicionar respostas às ações e eventos do DOM³, entre outros. Além disso, de acordo com uma pesquisa realizada em 2018 pelo *StackOverflow*, um dos maiores sites de perguntas e respostas voltadas à programação e desenvolvimento, o *JavaScript* é a linguagem mais popular dentre os desenvolvedores profissionais, conforme o Gráfico 1. O Gráfico 2 mostra, ainda, que aproximadamente 40 % dos desenvolvedores participantes da pesquisa trabalham com *front-end* e 50 % se

² Link da página oficial da Oracle: <https://www.oracle.com/index.html>

³ *Document Object Model*, é um documento criado pelo navegador que permite que o *JavaScript* acesse e manipule cada elemento do HTML.

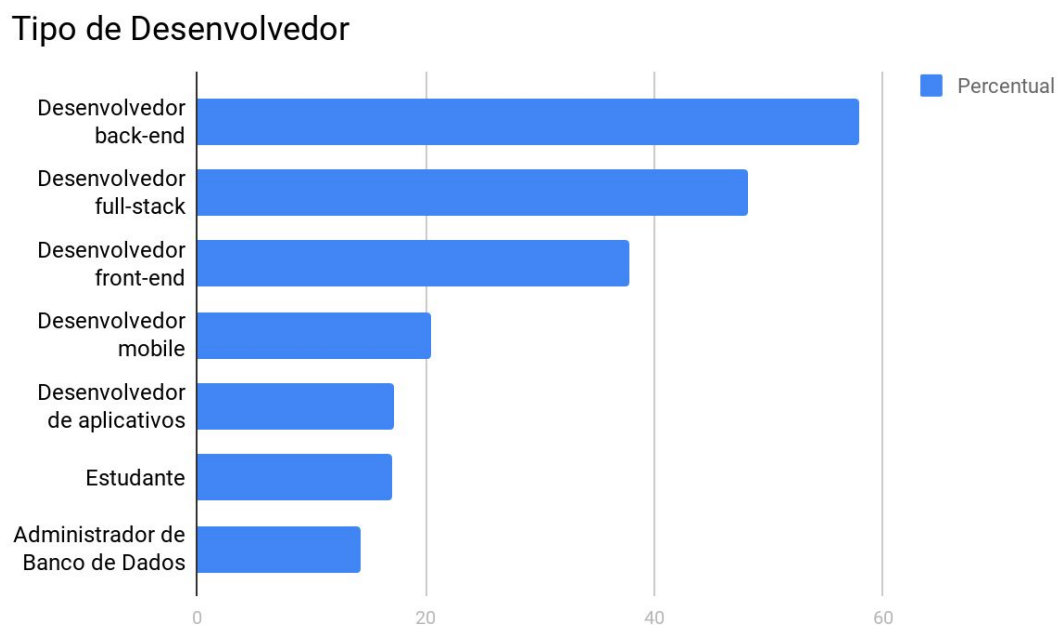
declaram *full-stack*, ou seja, trabalham tanto com o *front-end* quanto com o *back-end*, números significativos que podem ter influenciado na alta popularidade do *JavaScript*.

Gráfico 1: Relação das linguagens mais populares dentre os usuários profissionais do *StackOverflow*



Fonte: Adaptado de *StackOverflow* (2018)

Gráfico 2: Tipos de desenvolvedores usuários do *StackOverflow*



Fonte: Adaptado de *StackOverflow* (2018)

Além disso, o *JavaScript* possui uma grande quantidade de ferramentas externas que podem ser utilizadas para aumentar o poder da linguagem, tal como o *jQuery*.

2.4.1.4. **FRAMEWORK BOOTSTRAP**

O *framework Twitter Bootstrap*, cuja versão utilizada nesta pesquisa foi a 4, é, de acordo com os desenvolvedores, um kit de ferramentas escrito em CSS e *JavaScript* e que possui *plugins* construídos em *jQuery*, e permite criar sites responsivos (não se limitando a isso) através de seu sistema de grades.⁴ De acordo com Balasubramanee et al. (2013, tradução nossa):

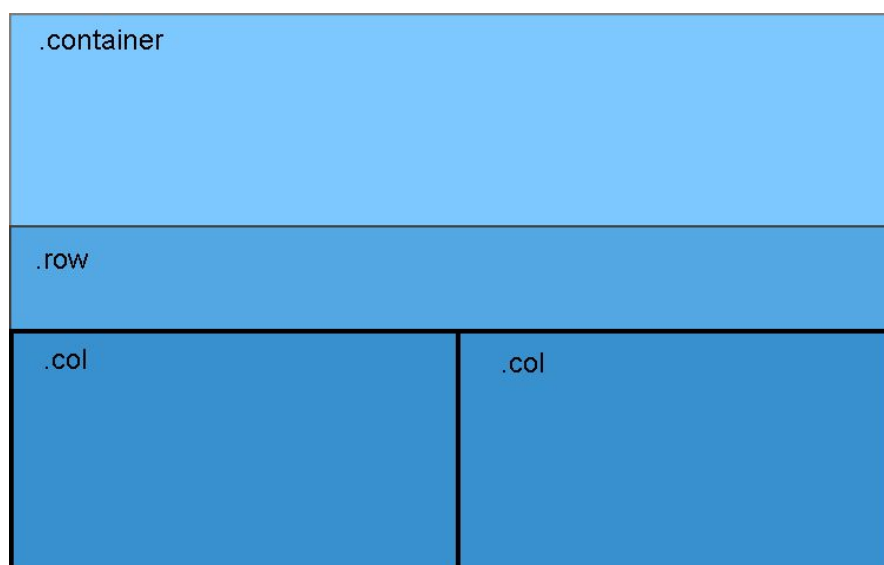
O *Twitter Bootstrap* é um *framework* poderoso que fornece um conjunto de classes de CSS e funções do *JavaScript* para facilitar o processo do desenvolvimento do *front-end*. Sua característica de *design* responsivo permite suporte tanto para telas de dispositivos *mobile* quanto computadores *desktop* [...].

É importante salientar que o presente trabalho não tem foco na responsividade do sistema, tendo em vista que foi desenvolvido com base no protótipo do módulo Aluno que, por sua vez, apresenta apenas a versão *web* para *desktop*.

De acordo com as informações do desenvolvedor, a formatação padrão do *Bootstrap* para páginas *web* é composta por contêineres em sua parte mais externa e que, por sua vez, agrupam em seu interior linhas e colunas, que podem ser especificados através do uso das classes *.container* ou *.container-fluid*, *.row*, *.col* e suas variações, que estão predefinidas na documentação do *Bootstrap* e podem ser habilitadas em elementos HTML como, por exemplo, *div* e *nav* (BOOTSTRAP, 2019). A estrutura se comporta conforme a Figura 4, onde o elemento com a classe *.container* atua como uma espécie de recipiente para o elemento *.row*, que funciona como uma linha de uma matriz matemática, onde elementos com a classe *.col* se acomodam e agem como colunas.

⁴ Retirado da página oficial do *Bootstrap*: <https://getbootstrap.com>

Figura 4. Representação gráfica do sistema de grades do *Bootstrap 4*



Fonte: Próprio autor

2.4.1.5. *jQUERY*

Sendo classificado como uma *library* de *JavaScript*, o *jQuery* é versátil e leve. De acordo com Silva (2016) e com os próprios desenvolvedores⁵, esta biblioteca permite que a manipulação do DOM do HTML, animações, eventos, CSS e AJAX⁶ sejam simplificadas em relação ao código em *JavaScript* puro, além de ter compatibilidade com a maioria dos navegadores. A versão utilizada neste trabalho foi a 3.3.1.

A Figura 5 mostra um exemplo de aplicação do *jQuery* escrita *inline*, ou seja, dentro do próprio arquivo HTML, onde a primeira função ativa o atributo “*show*” da *modal* (elemento do *Bootstrap*) assim que a página é carregada, ou seja, mostra esse elemento na tela; a segunda função ativa o atributo “*hide*”, que esconde a mesma *modal* ao receber um evento de clique em um elemento especificado no HTML.

⁵ Link da página oficial do *jQuery*: <https://jquery.com>

⁶ É uma técnica que faz uso do *JavaScript* e do DOM para permitir que páginas *web* sejam atualizadas de forma assíncrona.

Figura 5. Exemplo de código *JavaScript* escrito através da biblioteca *jQuery*

```
5 <script>
6   $(window).on('load', function()
7   {
8     $('#tela_validacao1').modal('show');
9   })
10  $(document).ready(function()
11  {
12    $('#validade_code1').click(function()
13    {
14      $('#tela_validacao1').modal('hide');
15    })
16  })
17 </script>
```

Fonte: Próprio autor

3. TRABALHOS RELACIONADOS

Este capítulo dispõe sobre resumos de alguns trabalhos científicos que possuem relação com esta pesquisa.

No trabalho apresentado por Souza, Cláudio (2016), o autor afirma que há uma grande dificuldade por parte dos alunos no entendimento das disciplinas de programação introdutória, seja por falta de preparo dos estudantes quanto por falta de recursos didáticos que auxiliem os alunos e professores. O autor, então, apresenta um estudo sobre a implementação do VisuAlg, um programa que simula o tradicional pseudocódigo através do chamado Portugol, porém de uma forma mais próxima das linguagens de programação tradicionais e que visa auxiliar tanto os estudantes quanto os professores no ensino-aprendizagem das disciplinas de programação, ao passo que faz uma comparação entre o método tradicional de ensino e as novas abordagens. Por fim, o artigo constata que a utilização de ferramentas de auxílio ao ensino devem ser utilizadas mais amplamente e destaca que o uso do VisuAlg em estágios iniciais de programação tem sido produtivo e eficiente em relação a melhora do aprendizado dos alunos, pois permite que os mesmos possam ter um contato mais próximo com os ambientes reais de programação.

Santos Sobrinha et al. (2016) trata do desenvolvimento de uma *Application Programming Interface* (API) desenvolvida em C++ e *Python*, voltada a comunicação e controle de um robô pedagógico, além do desenvolvimento de uma Plataforma de Programação Visual (PPV), cujo foco é tornar mais atrativo o processo de aprendizagem de programação tanto para alunos iniciantes quanto para alunos que possuem algum tipo de deficiência auditiva. Em outras palavras, os autores almejam utilizar a robótica e a PPV para auxiliar os alunos em disciplinas como programação e matemática. No decorrer do trabalho, os autores apresentam diversos exemplos de aplicações práticas tanto da API quando da PPV e mostram como esse tipo de ferramenta pode ser útil para despertar mais interesse e facilitar o aprendizado dos alunos nas disciplinas citadas.

Os autores Junior, Sandro e Souza (2014) procuram analisar o quão efetivo o ensino de Programação Orientada a Objetos (POO) utilizando a Rede Social Educativa - REDU - pode ser. Para tal, foram realizadas amostragens através dos alunos do componente curricular POO do curso de Licenciatura em Computação. Como resultados, os autores constataram que o uso dessa plataforma como opção de metodologia de ensino se mostrou benéfica e foi capaz de auxiliar o professor no processo de planejamento, ensino e avaliação do ensino de POO. Portanto, de acordo com o trabalho, a utilização da rede social em questão no meio acadêmico se apresenta como uma nova opção de método de ensino onde o professor deixa de ser apenas um transmissor de informação e passa a ser um mediador.

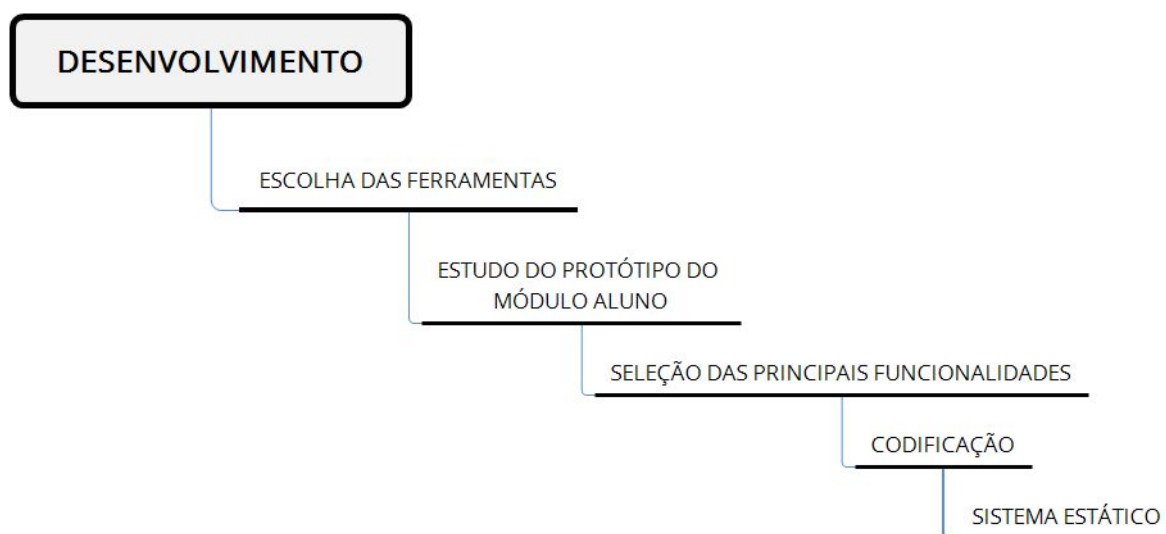
O artigo apresentado por Camargo e Fortunato (2018) trata de um estudo sobre o uso do programa Scratch aplicado ao ensino-aprendizagem de programação entre o período de 2010 à 2016. Para tanto, foram realizadas pesquisas no repositório da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e os dados obtidos demonstraram que o uso do *software* aumentou bastante com o passar dos anos, passando de poucas pesquisas na área no ano de 2010 a muitas no período de 2016. Os autores afirmam, então, que esse aumento do interesse sobre o Scratch demonstra que a ferramenta está cumprindo seu propósito de auxiliar no ensino-aprendizagem de programação.

Chaves et al. (2014) trata sobre o MOJO, uma ferramenta que integra tanto o sistema de juízes *online* quanto o *Moodle*, que é um ambiente virtual de aprendizagem; a ideia de envolver ambos os sistemas partiu da necessidade de um maior envolvimento do professor de programação e da disponibilização de mais questões de cunho prático. Essa plataforma tem como foco reduzir a sobrecarga de tarefas sobre o professor a partir da automatização dos processos, onde o docente pode gerenciar os recursos didáticos e agilizar os processos de elaboração, submissão e avaliação de atividades. Os resultados apresentados nesta pesquisa mostram que a grande parte dos professores concordam que uma plataforma como o MOJO seria útil na redução do tempo gasto com correções, submissões e elaborações de atividades e que também seria mais fácil acompanhar os alunos devido a redução da sobrecarga sobre os professores, além de poder dar um *feedback* mais rápido ao discente.

4. METODOLOGIA

Para a realização do trabalho proposto, uma série de procedimentos foram utilizados. Todos esses processos estão descritos neste capítulo e seguem a estrutura apresentada na Figura 6.

Figura 6. Metodologia do desenvolvimento do *front-end* do SE



Fonte: Próprio autor

4.1. ESCOLHA DAS FERRAMENTAS

Para se desenvolver um *software*, seja ele qual for, se faz necessária a codificação através de linguagens que podem ser interpretadas ou compiladas, podendo ser de programação, *script* ou ainda, marcação e estilo - essas últimas voltadas ao universo *web*. Para tal, um simples editor de texto pode ser utilizado para escrever tais códigos, no entanto, há ferramentas chamadas de IDEs que agregam outras funcionalidades e recursos capazes de aumentar a produtividade do desenvolvedor, entre outros fatores.

Para selecionar as linguagens mais adequadas para realização deste trabalho, foram levados em consideração critérios técnicos, de acordo com o modelo apresentado em Sebesta (2015), resumido na Tabela 1, que, apesar de possuírem foco em linguagens de programação, foram aplicados neste trabalho também às linguagens de marcação e estilo pelo fato de não existir uma métrica ou critérios

consolidados para a avaliação de linguagens *front-end*. Além disso, também foram utilizados os critérios gerais exibidos na Tabela 3, sendo eles: Curva de aprendizado, Relevância no mercado atual e Portabilidade. Já para as IDEs, os recursos disponíveis entre alguns modelos, sendo eles ATOM, *Visual Studio Code* e *NetBeans*, foram comparados a fim de encontrar a mais adequada às necessidades desta pesquisa.

Tabela 1. Critérios técnicos de avaliação de linguagens e as características que as afetam

Característica	CRITÉRIO		
	LEGIBILIDADE	FACILIDADE DE ESCRITA	CONFIABILIDADE
Simplicidade	✓	✓	✓
Ortogonalidade	✓	✓	✓
Tipos de Dados	✓	✓	✓
<i>Design</i> da Sintaxe	✓	✓	✓
Suporte a abstração		✓	✓
Expressividade		✓	✓
Verificação de Tipo			✓
Manipulação de Exceções			✓
<i>Aliasing</i> Restrito			✓

Fonte: Adaptado de Sebesta (2015)

De acordo com o Sebesta (2015), o critério de legibilidade se caracteriza, dentre outros fatores, pela facilidade de entendimento do código, quantidade de recursos de escrita disponíveis e a ortogonalidade. Já a facilidade de escrita é um critério que varia muito de acordo com o *software*, pois, por exemplo, sistemas que necessitem de um acesso mais profundo ao gerenciamento de memória irão precisar utilizar linguagens de nível mais baixo; no entanto, este critério também está relacionado a legibilidade e a quantidade de recursos de escrita. Por fim, a confiabilidade está intimamente relacionada à segurança e verificação de erros que a linguagem oferece.

A partir da análise dos critérios técnicos dispostos na Tabela 1, as linguagens HTML, CSS e *JavaScript* foram avaliadas, conforme a Tabela 2.

Tabela 2. Avaliação das linguagens HTML, CSS e *JavaScript*: critérios técnicos

Critérios→	LEGIBILIDADE			FACILIDADE DE ESCRITA			CONFIABILIDADE		
	Baixa	Média	Alta	Baixa	Média	Alta	Baixa	Média	Alta
Linguagens ↓									
HTML			✓			✓	Não se aplica		
CSS			✓			✓	Não se aplica		
JavaScript			✓			✓		✓	

Fonte: Próprio autor

Como HTML e CSS não são linguagens de programação e tratam apenas da parte visual dos elementos de uma página *web*, o critério de confiabilidade não se aplica a elas.

A tabela 3 apresenta a avaliação das linguagens com base nos três critérios gerais citados, que remetem a necessidades do próprio trabalho e autor, como tempo disponível para desenvolvimento desta pesquisa.

Tabela 3. Avaliação das linguagens HTML, CSS e *JavaScript*: critérios gerais

Critérios→	Curva de Aprendizagem			Relevância			Portabilidade		
	Baixa	Média	Alta	Baixa	Média	Alta	Baixa	Média	Alta
Linguagens ↓									
HTML	✓					✓			✓
CSS	✓					✓			✓
JavaScript		✓				✓			✓

Fonte: Próprio autor

As linguagens HTML e CSS são relativamente fáceis de se aprender quando comparadas à outros tipos de linguagens, como de programação. Por esse motivo sua curva de aprendizado é baixa, já o *JavaScript* demanda um pouco mais de estudo da sintaxe e de seu funcionamento, que resulta em uma curva média. Em se

tratando de relevância, ou seja, o quão importantes essas linguagens são no mercado atual, todas mostraram possuir uma alta relevância, conforme o Gráfico 1. Já a portabilidade, que está relacionada a quão bem uma linguagem pode ser aplicada em diferentes arquiteturas, apresenta um valor alto para todas as linguagens avaliadas, tendo em vista que, por serem focadas na *web*, apresentam alta compatibilidade com os principais navegadores do mercado e podem ser utilizadas independente da arquitetura da máquina e de seu Sistema Operacional (SO).

Em seguida, foi realizada uma revisão dos recursos e funcionalidades de três das principais IDEs disponíveis e utilizadas para o desenvolvimento *web*, sendo elas o *Visual Studio Code*, *ATOM* e *Netbeans*, a fim de se escolher a que melhor se adapta às necessidades deste trabalho. Na Tabela 4, os principais recursos dessas ferramentas estão dispostos.

Tabela 4. Principais características de algumas IDEs para desenvolvimento *web*

ATOM	Visual Studio Code	Netbeans
<i>OpenSource</i>	<i>OpenSource</i>	<i>OpenSource</i>
Multiplataforma	Multiplataforma	Multiplataforma
Altamente customizável	Leve	Possui <i>templates</i> de códigos
Capaz de trabalhar com inúmeros projetos de uma vez	Extensões para diversas linguagens	Capaz de trabalhar com inúmeros projetos de uma vez
Auxilia processos de edição de texto	Diversas opções de <i>debugging</i>	Suporte de <i>plug-ins</i> para diversas linguagens
	<i>IntelliSense</i> : sistema de auxílio para autocompletar palavras	Auxilia processos de edição de texto

Fonte: Próprio autor

Nota-se que todos os ambientes de desenvolvimento citados na Tabela 4 possuem recursos similares, no entanto, o *Visual Studio Code* se destaca pelo alto desempenho e capacidade de trabalhar facilmente com inúmeras linguagens e

projetos de uma única vez, proporcionando um ambiente completo para o desenvolvimento *web*, que requer a codificação paralela do HTML, CSS e *JavaScript* para gerar bons resultados.

Quando se trata de *Frameworks* e *Libraries*, o *Bootstrap* se mostrou muito útil devido à sua alta capacidade de proporcionar a customização das páginas *web* ao se utilizar de containers, colunas e linhas para dividir a tela do *browser*, ao mesmo tempo que facilita a responsividade, conforme exposto na seção 2.3.5. Por esses motivos, o *Bootstrap* foi escolhido como uma das ferramentas para auxiliar o desenvolvimento. Além disso, o *jQuery*, que já foi discutido na seção 2.3.6, também foi adotado por simplificar o próprio *JavaScript*, com a fácil manipulação de eventos e do DOM.

4.2. ESTUDO DO PROTÓTIPO DO MÓDULO ALUNO

De acordo com Souza, Dyego (2019), o sistema *Loop Academic* deverá possuir três módulos quando estiver finalizado, sendo eles: Aluno, Professor e Monitor. Esta etapa consiste na análise dos requisitos e funcionalidades presentes no protótipo do módulo Aluno desenvolvido no trabalho anterior, conforme já citado, a fim de listá-los. Devido a isso, foi realizado o estudo do protótipo e de sua documentação com a finalidade de se fazer um levantamento de todos os requisitos presentes no módulo em questão, baseado no diagrama de Casos de Uso apresentado por Souza, Dyego (2019).

Abaixo, estão listados todos os requisitos funcionais do módulo Aluno.

- Realizar Cadastro como Aluno;
- Realizar Login;
- Resolver Exercício;
- Consultar Código de Apoio;
- Consultar Material de Apoio;
- Consultar Dicas;
- Consultar Desempenho;
- Enviar Dúvidas;
- Criar Tópico no Fórum;

- Responder Tópico no Fórum;
- Consultar Emblemas.

4.3. SELEÇÃO DAS PRINCIPAIS FUNCIONALIDADES

Com base no levantamento dos requisitos e no tempo disponível para realização do trabalho em questão, apenas as funcionalidades julgadas como principais foram escolhidas para serem desenvolvidas, onde as listadas com prioridade Essencial no documento de requisitos, feito por Souza, Dyego (2019), foram selecionadas, bem como algumas marcadas com prioridade Importante, que, por sua vez, não necessitam diretamente do *back-end* e são básicas para o funcionamento inicial do sistema. As principais funcionalidades selecionadas estão listadas a seguir:

- Realizar Cadastro como Aluno;
- Realizar Login;
- Resolver Exercício;
- Consultar Código de Apoio;
- Consultar Material de Apoio;
- Consultar Dicas;
- Consultar Desempenho;
- Consultar Emblemas.

Os requisitos relacionados ao fórum e as dúvidas não foram priorizados pelo fato de requererem, de forma mais intensa, o *back-end*, portanto, foi entendido pelos autores que é mais interessante que estas funcionalidades sejam feitas futuramente com o *front-end* em conjunto com o *back-end*.

Cada funcionalidade listada possui ações internas que podem ser classificadas como fluxo principal, fluxos alternativos e de exceção, como as funções de código inválido, retornar, cancelar e salvar, de acordo com o protótipo do módulo Aluno.

4.4. CODIFICAÇÃO

Após a seleção das linguagens, das ferramentas e das funcionalidades, foi iniciado o processo de codificação do *front-end* dos requisitos funcionais, bem como seus fluxos principais, alternativos e de exceção do *front-end* do sistema, caracterizado pelo uso do HTML, CSS, *JavaScript* e ferramentas externas para moldar a plataforma baseado no protótipo de interface já existente. Para tal, o HTML foi utilizado para definir a estrutura de cada página, com suas seções, cabeçalhos, etc, além do CSS juntamente com o *Bootstrap*, para estilizar e personalizar cada documento HTML. Além disso, o *JavaScript* e *jQuery* foram responsáveis pela manipulação das ações do sistema, bem como o tratamento dos eventos, tais como clique, carregamento da página, entre outros.

Ao final da codificação do *front-end* um sistema estático é desenvolvido, ou seja, o usuário não é capaz de armazenar dados e informações no mesmo, processo este que será possível após o desenvolvimento e implementação do *back-end* em trabalhos futuros.

5. RESULTADOS E DISCUSSÕES

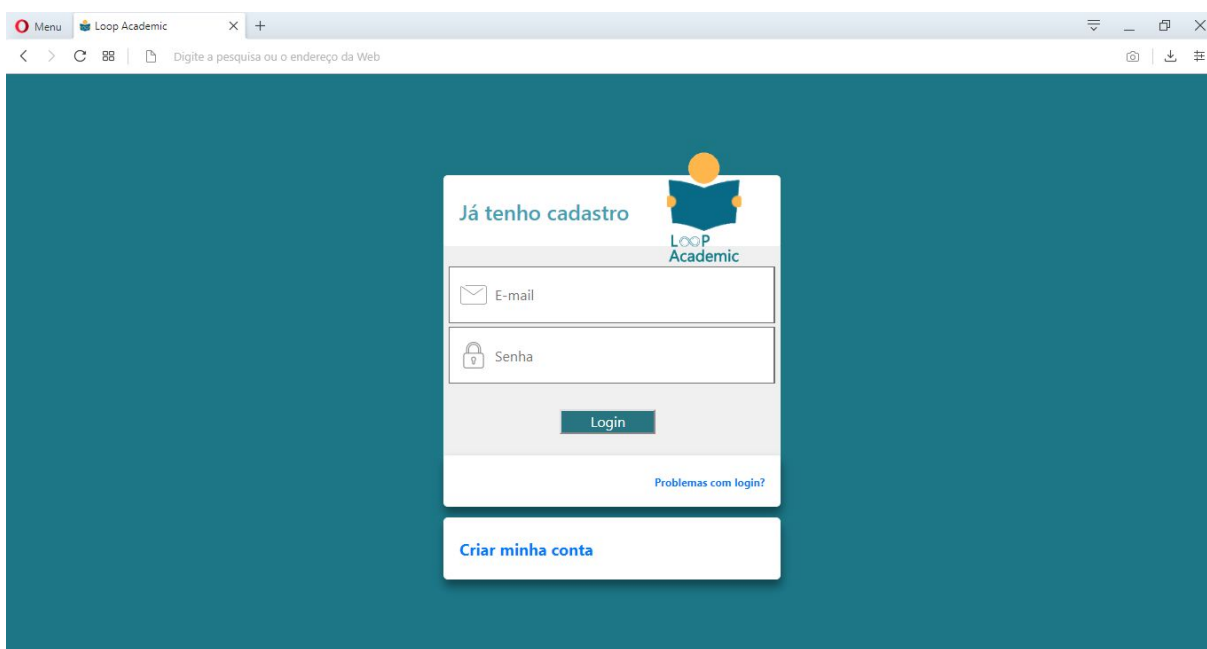
Este capítulo apresenta os resultados obtidos a partir do desenvolvimento do *front-end* do módulo Aluno do *Loop Academic*, bem como uma breve discussão sobre os mesmos.

5.1. DESENVOLVIMENTO DO *FRONT-END*

Conforme destacado nos capítulos anteriores, foi utilizado o HTML para estruturação das páginas estáticas, o CSS para estilizá-las e, por fim, o *JavaScript* para auxiliar na interação entre o usuário e o sistema e manipular o comportamento das páginas.

A primeira parte do sistema consiste na implementação dos requisitos de cadastro e *login*, conforme a Figura 7. É importante ressaltar que, pelo fato deste trabalho focar na implementação do *front-end*, portanto não há validação dos dados inseridos em nenhum dos campos de cadastro ou *login*, pois para que isso ocorra, é necessário que o *back-end* esteja implementado, permitindo *linkar* o sistema a um banco de dados.

Figura 7. Tela inicial do *Loop Academic*



Fonte: Próprio autor

Para tal, a estrutura presente na Figura 8 foi construída utilizando os elementos do *Bootstrap* para organizar o documento em linhas e colunas. Os trechos internos contendo textos e formulário foram omitidos com a intenção de simplificar a leitura do código. Por não possuir *back-end*, a ação do formulário é apenas redirecionar a tela de boas vindas. Os ids e classes são utilizados para permitir que o CSS modifique seletores específicos do HTML, conforme a Figura 9.

Figura 8. Trecho do HTML da página inicial

```

1 <body style="background-color: rgb(28, 118, 134)">
2   <div class="container">
3     <div class="row justify-content-center">
4       <div class="col-10 col-md-4" id="login_screen">
5         ...
6       </div>
7     </div>
8     <div class="row justify-content-center">
9       <div class="col-10 col-md-4" id="login_box">
10        <form action="boas_vindas.html">
11          ...
12        </form>
13      </div>
14    </div>
15    <div class="row justify-content-center">
16      <div class="col-10 col-md-4" id="bottom_box">
17        ...
18      </div>
19    </div>
20    <div class="row justify-content-center">
21      <div class="col-10 col-md-4" id="sign_up_box">
22        ...
23      </div>
24    </div>
25  </div>
26 </body>

```

Fonte: Próprio autor

Figura 9. Trecho do CSS referente a página inicial

```

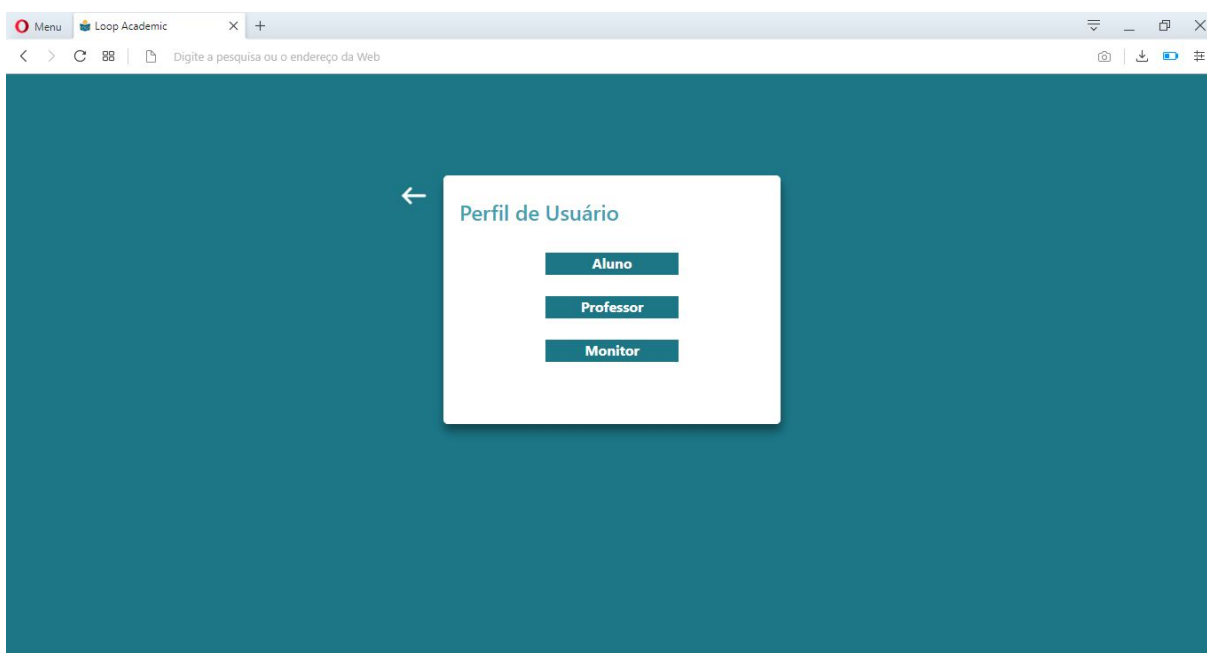
1 #login_screen {
2   margin-top: 10%;
3   background-color: white;
4   min-height: 80px;
5   border-style: double;
6   border-color: white;
7   border-top-left-radius: 5px;
8   border-top-right-radius: 5px;
9 }
10 #login_box {
11   background-color: rgb(240, 240, 240);
12   padding-left: 0.5%;
13   padding-top: 2%;

```

Fonte: Próprio autor

Ao clicar em “Criar minha conta”, o usuário é redirecionado a um menu de escolha de perfil de usuário, que permite que ele informe se deseja realizar o cadastro como aluno, professor ou monitor, como mostrado na Figura 10. As opções “Professor” e “Monitor” são referentes aos módulos que serão desenvolvidos futuramente, isto é, ainda não foi realizada a documentação completa tampouco o protótipo desses módulos, conforme Souza, Dyego (2019).

Figura 10. Perfis de Usuário



Fonte: Próprio autor

Ao selecionar a opção “Aluno”, um formulário de cadastro é exibido, conforme a Figura 11, que permite que o aluno crie sua conta no sistema. Após possuir uma conta cadastrada, o discente poderá realizar o *login* através da tela inicial do sistema de acordo com a Figura 7 e, em seguida, se deparará com uma tela de boas-vindas, (Figura 12), onde será necessário que o aluno insira seu código de Turma Virtual, código esse que deverá ser divulgado pelo próprio professor após o mesmo cadastrar uma Turma Virtual no sistema (funcionalidade ainda não disponível neste trabalho).

Figura 11. Cadastro do Aluno - Loop Academic

The screenshot shows a web browser window with the title 'Loop Academic'. The main content area has a teal background. A white modal form titled 'Dados Cadastrais - Aluno' is centered. It contains the following fields: 'Nome completo', 'Instituição de Ensino', 'Matrícula', 'E-mail', 'Confirme o E-mail', 'Senha', and 'Confirme a Senha'. A teal 'Cadastrar' button is at the bottom right of the form.

Fonte: Próprio autor

Figura 12. Tela de boas-vindas

The screenshot shows a web browser window with the title 'Loop Academic'. The main content area has a dark teal background. A white modal form titled 'Boas Vindas!' is centered. It contains the following text: 'Olá, futuro(a) programador(a)', 'É com enorme satisfação que desejamos-lhes nossas boas vindas ao Loop Academic! o/', 'Neste Software Educacional, você terá acesso à uma serie de funcionalidades que lhe auxiliarão no aprendizado de Algoritmos, por exemplo: listas de exercicios de programação, material de apoio ao estudo de Algoritmos, envio de dúvidas ao professor ou monitor, fórum de interação com os colegas do curso e muito mais! o/', 'Para ter acesso à todas as funcionalidades, é necessário que você informe abaixo o Código da Turma Virtual criada pelo seu professor. É com ele que você terá acesso aos materiais exclusivos da sua turma!', 'Por favor, digite o seu Código da Turma Virtual', and a text input field for 'Código da Turma Virtual'. At the bottom right of the form are two buttons: 'Não possuo' (red) and 'Enviar' (green).

Fonte: Próprio autor

A tela de boas vindas foi concebida através do uso do *modal*, que é um *plugin* do *Bootstrap* construído em CSS e *JavaScript (jQuery)* aplicado ao HTML. A figura 13 mostra a estrutura utilizada pelo autor, com CSS *inline*, para desenvolver a tela

da Figura 12. Além disso, o código *jQuery* apresentado na Figura 5 também foi aplicado ao documento HTML para controlar quando a *modal* deveria ser exibida ou escondida.

Figura 13. HTML - *Modal* do *Bootstrap*

```

1 <div class="row">
2 <div class="col-12">
3   <!-- Modal 1 -->
4   <div class="modal fade" id="tela_validacao1" tabindex="-1" data-backdrop="static" data-
5     keyboard="false" role="dialog" aria-labelledby="myModalTitle" aria-hidden="true">
6     <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
7       <div class="modal-content">
8         <div class="modal-header">
9           <div style="width: 100%; text-align: center;">
10             ...
11           </div>
12         </div>
13         <div class="modal-body" style="font-size: 14px;">
14           <div class="row">
15             <div class="col-12" style="text-align: justify;">
16               ...
17             </div>
18             <input type="text" name="codigo_virtual" id="virtual_code" placeholder="Código da
19               Turma Virtual">
20             <br>
21             <div class="row justify-content-end">
22               <div class="col-4">
23                 <br>
24                 <button type="button" data-toggle="modal" data-target="#tela_validacao2"
25                   id="validate_code1">Não possui</button>
26                 <input type="submit" value="Enviar" onclick="validate_input('virtual_code')"
27                   id="validate_code2">
28               </div>
29             </div>
30           </div>
31         </div>
32       </div>

```

Fonte: Próprio autor

Mais uma vez, por ser apenas o *front-end*, a validação do código é apenas simbólica e realizada pela função *validate_input()*, importada do *script* do *JavaScript*, (Figura 14), que realiza o teste condicional para a entrada do usuário e o redireciona para uma tela de código inválido caso nada seja inserido ou para uma tela de código válido caso algum dado seja inserido.

Figura 14. Função do *JavaScript*

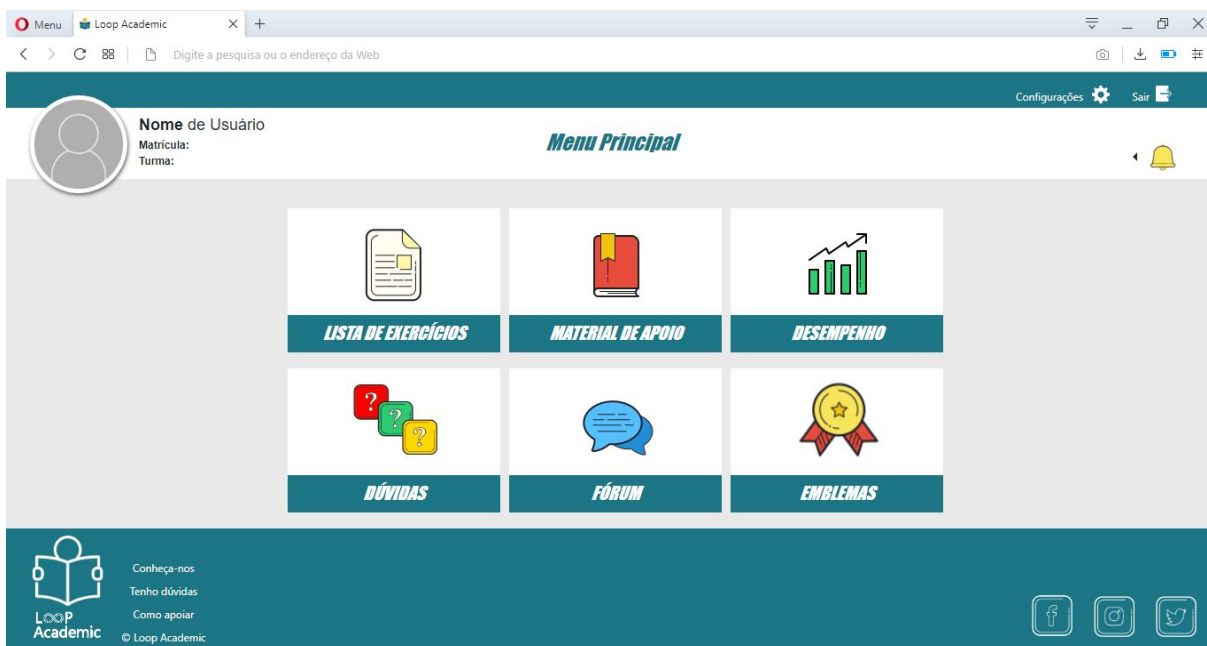
```

1 function validate_input(id) {
2   var value_input = document.getElementById(id).value;
3   if (value_input == "")
4   {
5     window.location.href = "codigo_invalido.html";
6   }
7   else
8   {
9     window.location.href = "codigo_valido.html";
10  }
11 }
12

```

Fonte: Próprio autor

Ao ter o código validado, em caso positivo, o discente é redirecionado ao menu principal, exibido na Figura 15, caso contrário, o sistema ativa o fluxo de exceção, notifica o usuário que o código é inválido e oferece a opção de inseri-lo novamente ou voltar à tela inicial. O menu principal, assim como todas as demais funcionalidades implementadas, foram desenvolvidas utilizando a estrutura de grades, já mencionada e mostrada, do *Bootstrap*.

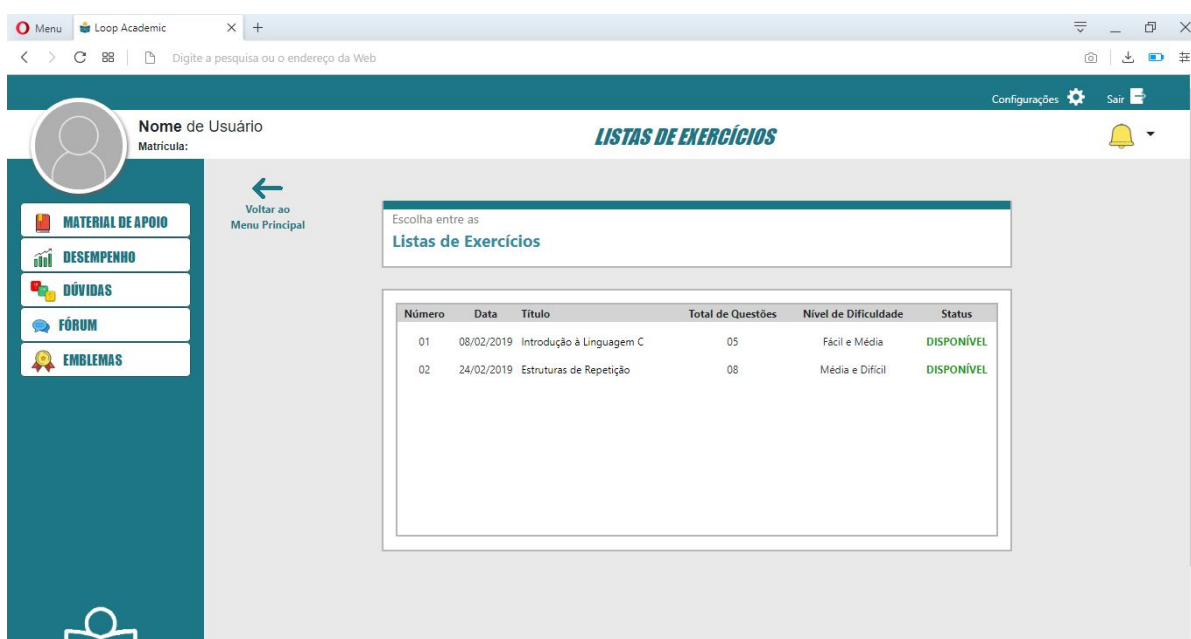
Figura 15. Menu Principal do *Loop Academic*

Fonte: Próprio autor

Ao acessar o menu principal, o discente pode escolher entre uma das seis opções, neste caso, os menus Lista de Exercícios, Materiais de Apoio, Desempenho e Emblemas estão disponíveis.

Ao escolher a opção Lista de Exercícios, o discente tem acesso a todos os exercícios disponíveis e pode selecionar qual deles deseja responder, conforme a Figura 16. Ao selecionar um dos exercícios, é apresentada a Figura 17, na qual existe a possibilidade do usuário escrever e executar o código no próprio sistema através do compilador da Repl.it⁷ embutido na página.

Figura 16. Lista de Exercícios

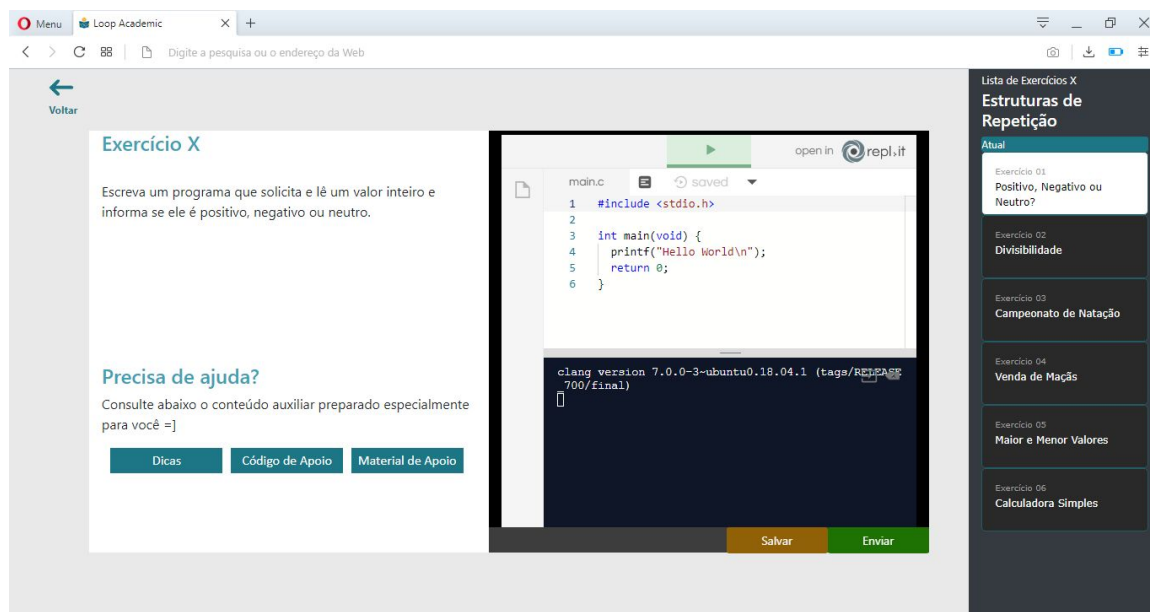


Fonte: Próprio autor

O usuário pode, então, escrever seu código e executá-lo diretamente do navegador através do compilador *online* citado, além de ver seu *output*. Há, ainda, as opções de salvar o código para responder mais tarde ou enviar para o professor corrigir, no entanto, só estarão efetivamente implementadas após o desenvolvimento do *back-end*. As guias “Dicas” e “Código de Apoio” são controladas via *modal* do *Bootstrap* e exibem conteúdo de ajuda referente a cada questão. Já a guia “Material de Apoio” redireciona o usuário para o menu que leva o mesmo nome, de acordo com a Figura 18.

⁷ Página inicial da Repl.it: <https://repl.it>

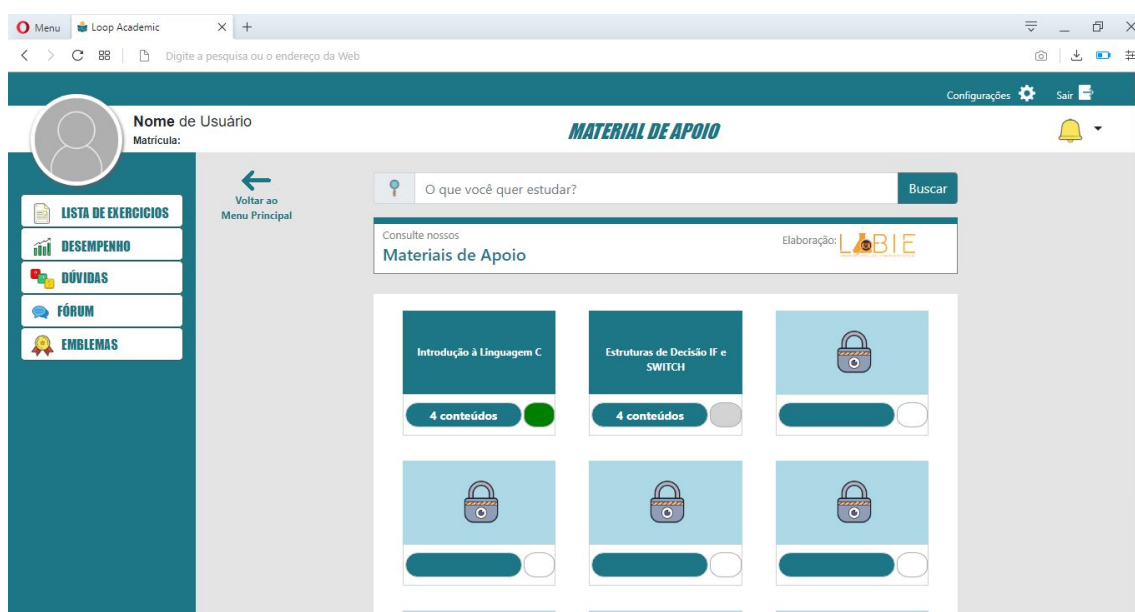
Figura 17. Exercício da Lista de Exercícios



Fonte: Próprio autor

No menu Material de Apoio, (Figura 18), são exibidos os materiais disponíveis, que são desbloqueados à medida que são visualizados. Os materiais acessíveis dentro do sistema são compostos por produções da equipe do Laboratório de Pesquisa em Informática na Educação - LABIE, e também por videoaulas produzidas pelo *Programming Loop*, canal do Youtube composto por estudantes/monitores da própria UFERSA - Campus Pau dos Ferros.

Figura 18. Material de Apoio



Fonte: Próprio autor

Através de eventos *onclick*, o *JavaScript*, escrito *inline*, é capaz de fazer com que a página exiba uma mensagem de conteúdo bloqueado quando o usuário clica em um dos materiais ainda não acessíveis, consoante o código da Figura 19.

Figura 19. Eventos controlados via *JavaScript*

```

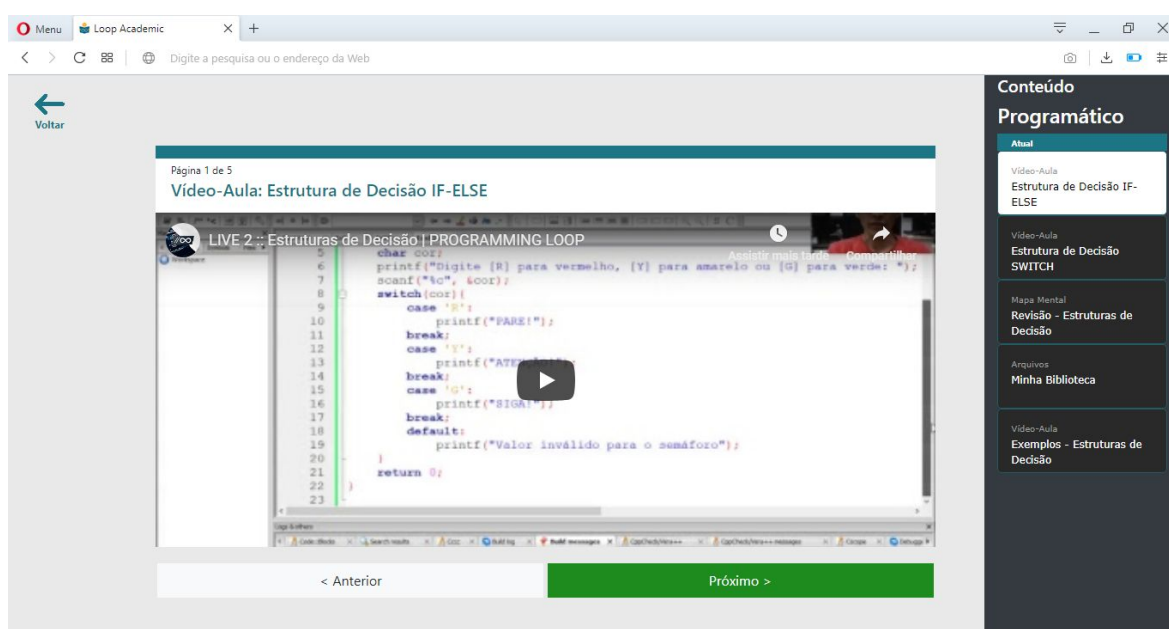
1 <div class="col col_content">
2 <div class="content_section" style="background-color: lightblue;">
3 <p><b><!--CONTEÚDO BLOQUEADO--></b></p>
4 
5 <div class="button_box" style="background-color: white;">
6 <div>
7 <button class="conteudos" onclick="window.alert('Conteúdo bloqueado! :()');"
  onmouseover="new_color(this)" onmouseout="old_color(this)"></button>
8 </div>
9 <div class="circle" style="background-color: white;"></div>
10 </div>
11 </div>
12 </div>

```

Fonte: Próprio autor

A Figura 20 mostra um dos materiais de apoio, composto pelas videoaulas do canal *Programming Loop* embutidas na página. O menu lateral especifica qual material está sendo exibido no momento além de permitir que o usuário selecione-os livremente, enquanto os botões “Anterior” e “Próximo” permitem a navegação entre os conteúdos. Ambas as funções citadas são controladas via *JavaScript*, enquanto parte do estilo fica por conta das classes padrão do *Bootstrap*, segundo a Figura 21.

Figura 20. Exemplo de Material de Apoio



Fonte: Próprio autor

Figura 21. HTML referente ao Material de Apoio

```

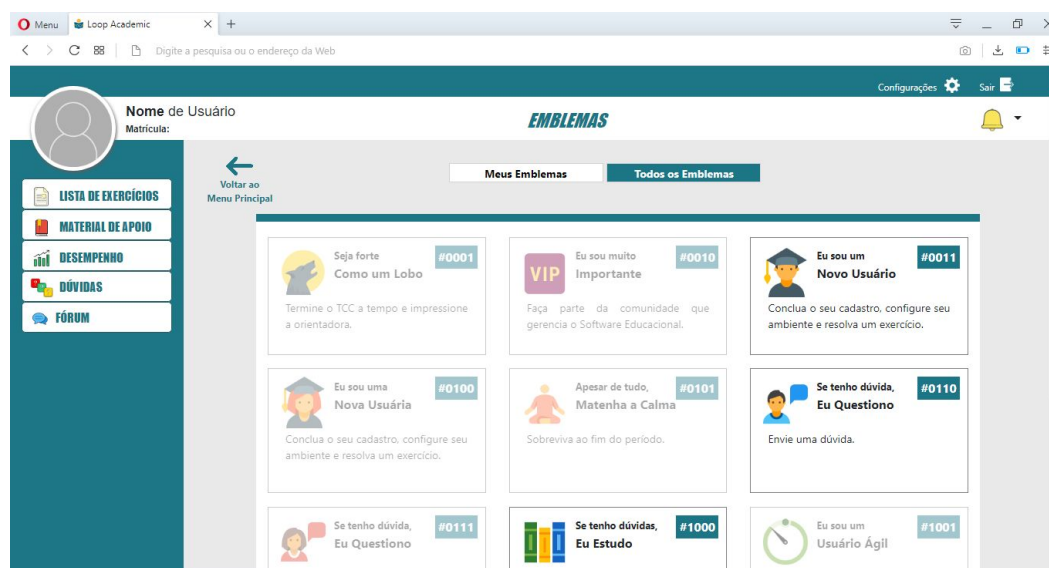
1 <div class="row justify-content-center">
2 <section class="col-11" style="margin-top: 8%;">
3 <div id="top_info">
4 <span>Página 1 de 5</span>
5 <h5>Vídeo-Aula: Estrutura de Decisão IF-ELSE</h5>
6 </div>
7 <div class="embed-responsive embed-responsive-21by9">
8 <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/U8ixPxL54Sk"
  frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
  allowfullscreen></iframe>
9 </div>
10 <button class="back btn-light" type="button" disabled>< Anterior</button>
11 <button class="next" onclick="window.location.href='material_2.html';" type="button">Próximo >
12 </section>
13 </div>

```

Fonte: Próprio autor

O código HTML utilizado para estruturar a página da Figura 20 pode ser encontrado no Anexo A.

Para representar como os emblemas obtidos pelo discente serão exibidos no *Loop Academic*, optou-se por representar alguns deles como se já tivessem sido conquistados (aqueles que não estão opacos) apenas de forma ilustrativa na Figura 22.

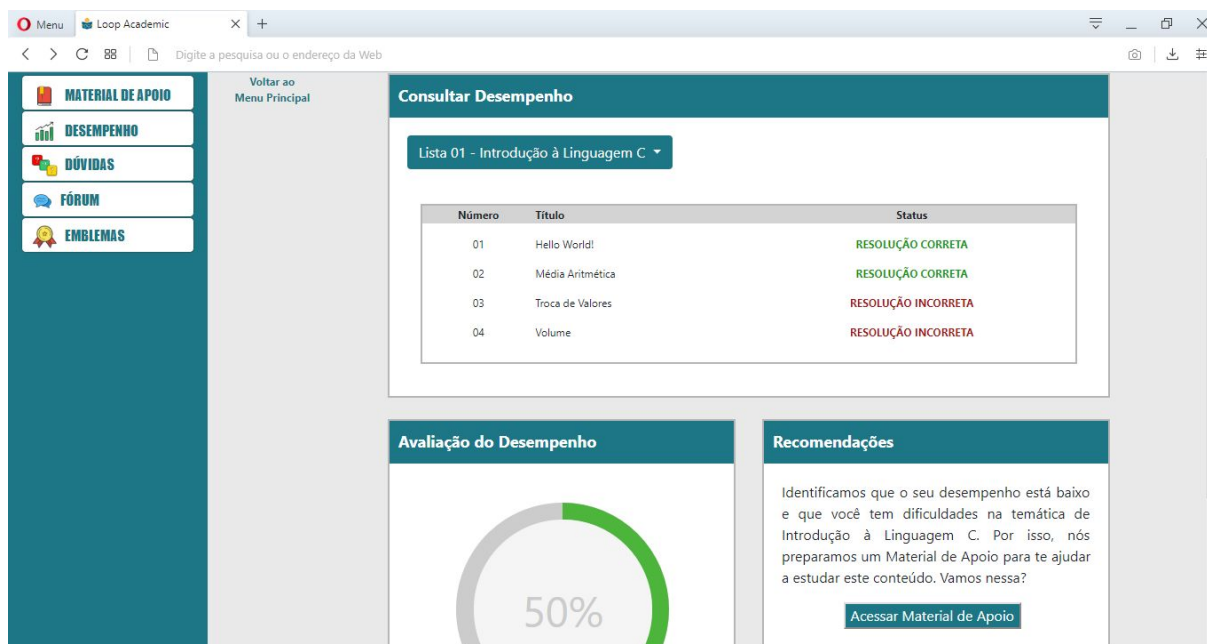
Figura 22. Tela de Emblemas

Fonte: Próprio autor

Por fim, a Figura 23 exibe a tela de Desempenho do usuário, onde informações referentes a cada lista de exercícios são mostradas, bem como a

porcentagem de acertos e recomendações de estudo caso o desempenho do estudante não tenha sido satisfatório.

Figura 23. Tela de Desempenho



Fonte: Próprio autor

Em todas as páginas do *Loop Academic*, o *Bootstrap* foi utilizado para facilitar a configuração dos elementos do HTML e sua disposição na tela, com o intuito de deixar o sistema o mais próximo possível do protótipo de interface existente. O *jQuery* foi requisitado apenas em alguns momentos quando os eventos referentes a alguns elementos do *Bootstrap* necessitaram de ações complementares, como no caso da exibição da *modal* ao abrir a página de boas vindas do sistema.

6. CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento do *front-end* do SE *Loop Academic*, que visa auxiliar o processo de ensino-aprendizagem e reduzir as dificuldades que a grande parte dos estudantes das disciplinas de programação introdutória possuem. Com isso, o *front-end* referente às principais funcionalidades do módulo Aluno foi desenvolvido utilizando várias ferramentas do desenvolvimento *web* e seguindo os parâmetros definidos através do estudo realizado anteriormente, conforme já citado no texto, no qual o protótipo da plataforma *web* em questão foi construído com base nas premissas da Engenharia de Software.

A partir dos resultados obtidos, nota-se que as principais funcionalidades do *client-side* do módulo Aluno foram implementadas com sucesso, do ponto de vista da codificação das telas conforme apresentadas no protótipo; e que as ferramentas escolhidas beneficiaram o desenvolvimento do SE, pois as versões atuais do HTML e CSS são poderosas para criação de páginas *web*, além do *JavaScript*, que é capaz de manipular elementos do HTML de forma a garantir uma maior dinâmica ao sistema. Além disso, o *Visual Studio Code* se mostrou bastante eficaz para o desenvolvimento *web* e o *framework* e a *library* escolhidos se mostraram extremamente efetivos, melhorando tanto a produtividade quanto a capacidade de personalização das páginas, especialmente o *Bootstrap*, que foi utilizado durante todo o projeto.

No entanto, é necessário que os trabalhos futuros contemplem não só os dois requisitos ainda não produzidos (Fórum e Dúvidas), mas também o *server-side* do sistema. Além disso, percebe-se também que os dois módulos ainda não documentados, Professor e Monitor, necessitam de estudos completos que partam desde a elicitação dos requisitos funcionais e não funcionais até a prototipagem e, por fim, o desenvolvimento tanto do *front-end* quanto do *back-end* para que o *Loop Academic* seja completamente desenvolvido.

REFERÊNCIAS

BALASUBRAMANEE, Viknes et al. Twitter bootstrap and AngularJS: Frontend frameworks to expedite science gateway development. **2013 IEEE International Conference On Cluster Computing (cluster)**, [s.l.], set. 2013. IEEE. <http://dx.doi.org/10.1109/cluster.2013.6702640>.

BOOTSTRAP. **Introduction**. Disponível em: <https://getbootstrap.com/docs/4.3/getting-started/introduction/>>. Acesso em 29 de jun. 2019.

BRITO JUNIOR, Ozonias de Oliveira. **Abordagens para avaliação de software educativo e sua coerência com os modelos de qualidade de software**. Dissertação de Mestrado. Universidade Federal da Paraíba - UFPB/PPGI, João Pessoa - PB, 2016.

CALLAHAN, D., PEDIGO, B. **Educating Experienced IT Professionals by Addressing Industry's Needs**. IEEE Software, 2002, v. 19(5), p. 57-62.

CAMARGO, Ivan Rodrigues de; FORTUNATO, Ivan. O Scratch como auxiliar no processo de ensino-aprendizagem de linguagem de programação: um balanço da pós-graduação nacional entre 2010 e 2016. **Revista on line de Política e Gestão Educacional**, [S.l.], p. 608-626, may 2018. ISSN 1519-9029. Disponível em: <https://periodicos.fclar.unesp.br/rpge/article/view/10754/7621>>. Acesso em: 30 june 2019. doi:https://doi.org/10.22633/rpge.v22.n2.mai/ago.2018.10754.

CARITÁ, E.; SANCHES, L. M. P.; PADOVAN, V. Uso de redes sociais no processo ensino-aprendizagem. In: **17º CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA**, 2011, Manaus.

CASPERSEN, Michael; KOLLING, Michael. Stream: A first programming process. **ACM Transactions on Computing Education**, Nova York, EUA, v. 9, n. 1, Article 4, 2009.

CHAVES, José Osvaldo Mesquita et al. MOJO: UMA FERRAMENTA PARA INTEGRAR JUÍZES ONLINE AO MOODLE NO APOIO AO ENSINO E APRENDIZAGEM DE PROGRAMAÇÃO. **Holos**, [s.l.], v. 5, n. , p.246-251, 30 set. 2014. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN). <http://dx.doi.org/10.15628/holos.2014.1904>.

DIOGINIS, Maria Lucineide et al. As novas tecnologias no processo de ensino aprendizagem. **Colloquium Humanarum**, Presidente Prudente, v. 12, n. Especial, p.1155-1162, out. 2015. Trimestral. Disponível em: <http://www.unoeste.br/site/enepe/2015/suplementos/area/Humanarum/Educacao/AS%20NOVAS%20TECNOLOGIAS%20NO%20PROCESSO%20DE%20ENSINO%20APRENDIZAGEM.pdf>>. Acesso em: 29 jun. 2019.

FERNANDES, Tales Pinheiro. **Proposta inicial de um processo de desenvolvimento web front-end utilizando web components**. 2017. 64 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Ciências

Tecnológicas, Universidade Estadual do Norte do Paraná, Bandeirantes, 2017.

FIALHO, Neusa; MATOS, Elizete. **A arte de envolver o aluno na aprendizagem de ciências utilizando softwares educacionais**. Educar em Revista, Curitiba, n. especial 2, p. 121-136, 2010. Editora UFPR. Disponível em: <<http://www.scielo.br/pdf/er/nspe2/07.pdf>> Acesso em: 03 mai. 2019.

GOMES, Anabela de Jesus Gomes. **Dificuldades de aprendizagem de programação de computadores**: contributos para a sua compreensão e resolução. Tese de Doutorado. Universidade de Coimbra - UC/Departamento de Engenharia Informática, Coimbra - Portugal, 2010.

HOLANDA, Wallace, et al. **Desafios na aprendizagem de programação introdutória em cursos de TI da UFERSA, campus Pau dos Ferros**: um estudo exploratório. ECOP 2018, 19 a 21 de Março de 2018, Pau dos Ferros, RN – ISSN 2526-7574 – v. 2, p. 90-96. Disponível em: <<https://periodicos.ufersa.edu.br/index.php/ecop>>. Acesso em: 04 mai. 2019

GINIGE, A., MURUGESAN, S. **Web Engineering: an Introduction**, IEEE Multimedia, v. 8, n. 1, 2001, p. 14-18.

JUCÁ, Sandro C. Silveira. A relevância dos softwares educativos na educação profissional. **Revista Ciências e Cognição**, Rio de Janeiro, v. 8, 2006. Disponível em: <<http://www.cienciasecognicao.org/revista/index.php/cec/article/view/571/359>>. Acesso em: 03 mai. 2019.

JUNIOR, Elias Vidal Bezerra; SANDRO, Gomes Alex; SOUZA, Flavia Veloso Costa. **Análise da prática docente no processo de ensino de programação orientada a objeto mediado por meio de rede social educativa**. EaD & Tecnologias Digitais na Educação, Dourados, v. 2, n. 3, p. 111-115, nov. 2014. ISSN 2318-4051. Disponível em: <<http://ojs.ufgd.edu.br/index.php/ead/article/view/3414/2105>>. Acesso em: 30 jun. 2019.

KOCH, Marlene Zimmermann. **As tecnologias no cotidiano escolar: uma ferramenta facilitadora no processo ensino-aprendizagem**. 2013. 36 f. Monografia (Especialização) - Curso de Gestão Educacional, Centro de Pós-graduação A Distância, Universidade Federal de Santa Maria, Sarandi, 2013. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/498/Koch_Marlene_Zimmermann.pdf?sequence=1>. Acesso em: 29 jun. 2019.

Koile, K., & Singer, D. (2006). Improving learning in CS1 via tablet-PC-based-in-class assessment. **ICER 06: Proceedings of the Second International Workshop on Computing Education Research**. p. 119-126. ACM; Canterbury, UK.

LIMA, M. M. de; LIMA, A. R. de; MONTEIRO, A. C. C.; JÚNIOR, E. H. C.; GOMES, L. d. Q. L. Uma Revisão Sistemática da Literatura dos Processos de Desenvolvimento de Software Educativo. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2012. v. 23, n. 1.

MDN, Web Docs - Mozilla. **JAVASCRIPT BASICS**. 2019. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics>. Acesso em 30 de jun. 2019.

MERCADO, L. P. L. **Novas tecnologias na educação**: Reflexões sobre a prática. Ed. UFAL, Maceió, 2002.

MILLETO, Evandro Manara; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de Software II**: Introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP. Rio Grande do Sul: bookman, 2014.

MORAN, José Manuel et al. **Novas Tecnologias e mediação pedagógica**. 6. Ed. Campinas; Papirus, 2000.

OLIVEIRA, C. C., Menezes, E. I., Moreira, M. **Ambientes Informativos de Aprendizagem**: produção e avaliação de software educativo. Campinas: Editora Papirus. 2001.

PENHA, Daniel. **Software Educacional para o Ensino-Aprendizagem de Equação do 2º Grau**: Desenvolvimento e Avaliação. Lavras - MG, 2015. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/5249>>. Acesso em: 03 mai. 2019.

Prietch, S. S., & Pazeto, T. A. **Estudo sobre a Evasão em um Curso de Licenciatura em Informática e Considerações para Melhorias**. WEIBASE, Maceió/AL. 2010.

RAPOSO, E. H. S.; DANTAS, V. O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**, [s.l.], v. 5, n. 27, p.577-586, 7 nov. 2016. Sociedade Brasileira de Computação - SBC.

SANTOS, Maria Adélia Icó dos. **Utilização de Realidade Aumentada no Desenvolvimento de Software Educacional: um exemplo em alguns conceitos na Astronomia**. 2015. 106 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Computação Aplicada, Universidade Estadual de Feira de Santana, Feira de Santana - Ba, 2015. Disponível em: <<http://tede2.uefs.br:8080/handle/tede/366>>. Acesso em: 30 jun. 2019.

SANTOS SOBRINHA, Vitória Heliane Pereira et al. Plataforma para Auxílio ao Ensino de Programação e Robótica Pedagógica. **Revista Principia - Divulgação Científica e Tecnológica do IFPB**, [S.l.], n. 31, p. 104-112, dez. 2016. ISSN 2447-9187. Disponível em: <<http://periodicos.ifpb.edu.br/index.php/principia/article/view/727>>. Acesso

em: 29 Jun. 2019.

doi:<http://dx.doi.org/10.18265/1517-03062015v1n31p104-112>.

SEBESTA, Robert W. **Concepts of Programming Languages**, 10^a ed. Pearson, 2015.

SILVA, FS., SERAFIM, ML. Redes sociais no processo de ensino e aprendizagem: com a palavra o adolescente. In: **SOUSA, RP., et al., orgs. Teorias e práticas em tecnologias educacionais [online]**. Campina Grande: EDUEPB, 2016, p. 67-98. ISBN 978-85-7879-326-5. Disponível em: <<http://books.scielo.org/id/fp86k/pdf/sousa-9788578793265-04.pdf>>. Acesso em: 05 mai. 2019.

SILVA, Gabriel Dumont de Lima e. **Desenvolvimento de um sistema web para gerenciamento de pedidos de delivery de comida em Diamantina/MG**. 2016. 76 f. TCC (Graduação) - Curso de Sistemas de Informação, Faculdade de Ciências Exatas, Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina - Mg, 2016.

SOARES, Felipe A. L.; CARVALHO, Rodrigo B. de. **Proposta de um Portal Educacional para Estudantes de Programação de Computadores**. ABAKÓS, Instituto de Ciências Exatas e Informática. PUC-Minas, 2017. Disponível em: <<http://periodicos.pucminas.br/index.php/abakos/article/view/P.2316-9451.2017v5n2p36/11700>>. Acesso em: 04 mai. 2019.

SOUZA, Cláudio Morgado de. VisuAlg - Ferramenta de Apoio ao Ensino de Programação. **Revista Eletrônica Teccen**, [s.l.], v. 2, n. 2, p.01-2, 3 out. 2016. Semestral. Universidade Severino Sombra. <http://dx.doi.org/10.21727/teccen.v2i2.234>. Disponível em: <<http://editora.universidadedevassouras.edu.br/index.php/TECCEN/article/view/234>>. Acesso em: 04 jul. 2019.

SOUZA, Dyego Magno Oliveira. **Desenvolvimento e avaliação do protótipo do Loop Academic: Um Software Educacional para o auxílio no processo de ensino-aprendizagem de programação introdutória**. 2019. 98 f. TCC (Graduação) - Curso de Tecnologia da Informação, Universidade Federal Rural do Semi-Árido, Pau dos Ferros, 2019.

STACKOVERFLOW. **Developer Survey Results 2018**. Disponível em: <<https://insights.stackoverflow.com/survey/2018/>>. Acesso em 30 de jun. 2019.

TAVARES, Jéssika Lima. **Modelos, Técnicas e Instrumentos de Análise de Softwares Educacionais**. UFPB. Trabalho de Conclusão de Curso, 2017. Disponível em: <<https://repositorio.ufpb.br/jspui/bitstream/123456789/2563/1/JLT19062017.pdf>>. Acesso em: 04 mai. 2019.

TEIXEIRA, Adriano; BRANDÃO, Edemilson. Software educacional: difícil começo. **Revista Novas Tecnologias na Educação – RENOTE**. v. 1, n. 1.

CINTED/UFRGS, fev. 2003. Disponível em:

<<https://seer.ufrgs.br/renote/article/view/13629/7699>>. Acesso em: 04 mai. 2019.

VALENTE, J. Armando. A Comunicação e a Educação baseada no uso das Tecnologias Digitais de Informação e Comunicação. **Revista UNIFESO**. v. 1, n. 1, 2014, p. 141-166.

W3Schools. **HTML INTRODUCTION**. Disponível em:

<https://www.w3schools.com/html/html_intro.asp>. Acesso em: 30 de jun. 2019.

W3Schools. **CSS INTRODUCTION**. Disponível em:

<https://www.w3schools.com/css/css_intro.asp>. Acesso em: 30 de jun. 2019.

ANEXO A - Código HTML

Dispõe do código HTML utilizado para codificação da estrutura da página *web* exibida na Figura 20. Nele, diversos elementos da linguagem são utilizados, como as *tags* `<div>`, `<button>`, `<nav>` e ``, bem como trechos de CSS *inline* e classes padrão do *Bootstrap*, como a *sidebar*, que foi incorporada a uma tag `<nav>` para criar um menu lateral.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <title>Loop Academic</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap-theme.min.css"
integrity="sha384-6pzBo3FDv/PJ8r2KRkGHifhEocL+1X2rVCTTkUfGk7/0pbek5mMa1upzvWbrUbOZ"
crossorigin="anonymous">
    <link rel="stylesheet" href="system_style.css">
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="shortcut icon" href="Favicon.png" sizes="32x32" type="image/ico">

    <script src="script_1.js"></script>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1cLHTMga3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYYolly6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
```

```

</head>

<body style="background-color: rgb(233, 233, 233)">
  <div class="container-fluid">
    <div class="row">
      <div class="col-1">
        <a id="voltar_seta" href="material.html"><p>Voltar</p></a>
      </div>
      <div class="col-9">
        <div class="row justify-content-center">
          <section class="col-11" style="margin-top: 8%;">
            <div id="top_info">
              <span>Página 1 de 5</span>
              <h5>Vídeo-Aula: Estrutura de Decisão IF-ELSE</h5>
            </div>
            <div class="embed-responsive embed-responsive-21by9">
              <iframe class="embed-responsive-item"
src="https://www.youtube.com/embed/U8ixPxL54Sk" frameborder="0" allow="accelerometer; autoplay;
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
            </div>
            <button class="back btn-light" type="button" disabled>< Anterior</button>
            <button class="next" onclick="window.location.href='material_2.html';"
type="button">Próximo ></button>
          </section>
        </div>
      </div>

      <nav class="col-md-2 d-none d-md-block bg-dark sidebar">
        <div class="sidebar-sticky">
          <ul class="nav flex-column">
            <li class="nav-item">
              <h5 style="color: white;">Conteúdo</h5><h4 style="color: white;">Programático</h4>
            </li>
            <li class="nav-item right_bar">
              <div id="current_material">Atual</div>
              <a class="nav-link active" style="background-color: white; color: black;"
href="material_video.html">

```

```

        <small>Vídeo-Aula</small><p>Estrutura de Decisão IF-ELSE</p> <span
class="sr-only">(current)</span>
    </a>
</li>
<li class="nav-item right_bar">
    <a class="nav-link" href="material_2.html">
        <small>Vídeo-Aula</small><p>Estrutura de Decisão SWITCH</p>
    </a>
</li>
<li class="nav-item right_bar">
    <a class="nav-link" href="material_3.html">
        <small>Mapa Mental</small><p>Revisão - Estruturas de Decisão</p>
    </a>
</li>
<li class="nav-item right_bar">
    <a class="nav-link" href="material_4.html">
        <small>Arquivos</small><p>Minha Biblioteca</p>
    </a>
</li>
<li class="nav-item right_bar">
    <a class="nav-link" href="material_5.html">
        <small>Vídeo-Aula</small><p>Exemplos - Estruturas de Decisão</p>
    </a>
</li>
</ul>
</div>
</nav>
</div>
</body>
</html>

```