

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE SISTEMAS DE INFORMAÇÃO

GIOVANI TEDESCO MENEGAT

Code Path:

**Concepção e desenvolvimento de uma plataforma para o auxílio de novos
alunos no aprendizado de programação**

Porto Alegre

2022

GIOVANI TEDESCO MENEGAT

Code Path:

**Concepção e desenvolvimento de uma plataforma para o auxílio de novos
alunos no aprendizado de programação**

Artigo apresentado como requisito parcial
para obtenção do título de Bacharel em
Sistemas de Informação, pelo Curso de
Sistemas de Informação da Universidade
do Vale do Rio dos Sinos (UNISINOS).

Orientador: Prof. Dr. Fernando Marson

Porto Alegre

2022

Code Path:**Concepção e desenvolvimento de uma plataforma para o auxílio de novos alunos no aprendizado de programação**

Giovani Tedesco Menegat

Prof. Dr. Fernando Marson

Resumo: Alunos novos no aprendizado da programação de software apresentam uma grande dificuldade no início, seja pelo excesso de conteúdo para aprender, pela dificuldade natural do aprendizado da programação ou pela falta de conhecimento sobre as tecnologias presentes no mercado e qual caminho ele deve seguir inicialmente. Por isso, o déficit de programadores qualificados no mercado de trabalho não para de crescer. O presente trabalho propõe o desenvolvimento de um software que irá auxiliar o aluno a ingressar no mercado de trabalho de TI, através de uma trilha que ele irá seguir, aprendendo conteúdos mais básicos sobre a área, passo a passo, onde permite o aluno a avaliação de cada etapa, assim no final, será apresentado uma recomendação de qual área seguir, com base nas avaliações. A plataforma tem como objetivo dar ao aluno uma visão geral sobre quais tecnologias estão presentes no mercado e qual o objetivo delas. A plataforma não tem como objetivo a formação em alguma tecnologia específica, mas sim transmitir ao aluno o conhecimento geral de como o mercado funciona atualmente, o que cada profissional de TI faz e pelo que ele é responsável.

Palavras-chave: Desenvolvimento de Software; Introdução à computação; Plataformas de ensino; Sistema de recomendação.

1 INTRODUÇÃO

Cada vez mais podemos perceber o aumento na procura por profissionais na área de Tecnologia da Informação, mas a formação dos mesmos ainda não acompanha esta demanda (BrasscomA, 2021). Apesar da geração atual ser considerada de nativos digitais, muitas escolas ainda pecam no Ensino Básico da informática, como mostra uma pesquisa feita em 2014, pela ONG Todos pela Educação. Esta pesquisa indicou que 48,1% das escolas públicas de Ensino Básico não tem computadores para uso individual dos alunos (ESTADÃO, 2014).

A programação de computadores é importante para o desenvolvimento do Pensamento Computacional, do raciocínio lógico e da criatividade do indivíduo, aumentando sua capacidade de resolução de problemas. Entretanto, a aprendizagem de programação exige alto grau de abstração, além de tempo e esforço de aprendizagem (ARIMOTO e OLIVEIRA, 2019).

Assim, podemos perceber que, além do déficit no ensino da informática nas escolas, a dificuldade no aprendizado da programação também é um empecilho na formação de novos programadores.

De acordo com a Associação das Empresas de Tecnologia da Informação e Tecnologias Digitais (BrasscomB), em 2021, “O relatório estima que as empresas de tecnologia demandem 797 mil talentos de 2021 a 2025. No entanto, com o número de formandos aquém da demanda, a projeção é de um déficit anual de 106 mil talentos – 530 mil em cinco anos”.

Também, segundo a Brasscom, “o Brasil forma 46 mil pessoas com perfil tecnológico por ano”. Isso demonstra a insuficiência na formação de novos profissionais no Brasil.

Novos cursos online e escolas de programação surgem a todo momento, porém parece que ainda não é o suficiente. À medida que a demanda cresce, a formação de novos profissionais não acompanha o ritmo.

Além disto, muitas pessoas que tentam ingressar na área se deparam com um mundo extremamente confuso e complexo à primeira vista, isso muitas vezes provoca a desistência do aprendizado.

Segundo Santos e Costa (2006), existem alguns problemas no processo de ensino de programação, como a falta de motivação dos alunos, devido ao despreparo e ao desânimo, que se cria na crença que a disciplina é um obstáculo extremamente difícil de ser superado.

Por isso a importância de abordar o tema de maneira correta, respeitando as dificuldades de cada pessoa e abordando todos os temas necessários para a evolução no aprendizado.

1.1 Tema

Nesta perspectiva, diante do enorme déficit de profissionais de tecnologia no mercado, percebe-se a necessidade da criação de novos meios para a formação de profissionais capazes de atender a demanda do mercado.

1.2 Delimitação do Tema

Devido à abrangência do assunto, delimitou-se como finalidade a criação de um software para auxiliar os alunos iniciantes na introdução à programação.

1.3 Problema

A dificuldade de desenvolver novos profissionais de TI no mercado brasileiro e o déficit de programadores capazes de preencher as vagas nas empresas.

1.4 Objetivos

1.4.1 Objetivo geral

De forma geral, o objetivo deste projeto é auxiliar os alunos que desejam ingressar na área de programação, através de um software que irá introduzir, passo a passo, o que é necessário para entrar nesta área e irá recomendar ao aluno, ao final do curso, qual área (*front-end* ou *back-end*) o aluno mais se identifica, com base na avaliação dele em cada conteúdo.

Outro foco do projeto é apresentar ao aluno uma visão aprimorada sobre o mundo de programação, onde ele vai ter o poder de decidir qual área ele quer seguir e a partir disto qual linguagem de programação ele quer aprender para se inserir no mercado de trabalho.

Este software irá apresentar uma trilha de conteúdos didáticos, com o foco na introdução do usuário na programação. O aluno irá consumir o conteúdo de cada etapa da trilha e marcar se gostou ou não e assim passar para a próxima etapa. Cada usuário terá seu login e senha e com isso seu progresso na trilha ficará salvo.

1.4.2 Objetivos específicos

Para alcançar o objetivo, foram separadas as seguintes etapas:

- a) Modelagem do sistema;
- b) Criação do *back-end*, utilizando NodeJs;
- c) Criação do *front-end*, utilizando o *framework* VueJS;
- d) Integrar as funcionalidades do *back-end* com as páginas do *front-end*;

e) Validar a proposta e conceito.

Especificando o projeto, será desenvolvido uma plataforma web com uma trilha para o ensino básico sobre programação. Cada etapa da trilha possui um vídeo e um texto explicando o conteúdo e o aluno poderá avaliar se gostou ou não do conteúdo. Com essa avaliação, será apresentado ao usuário uma recomendação de qual área (*front-end* ou *back-end*) ele poderá seguir, com base nas suas avaliações durante o curso.

1.5 Justificativa

O déficit na formação de novos profissionais de TI torna necessário a criação de novas ferramentas para o desenvolvimento de novos programadores. Por isso, este trabalho visa a criação de um software educacional, capaz de mapear os conceitos básicos do mundo da programação, a fim de introduzir o aluno à área de TI da maneira mais amigável possível.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologia da informação

A Tecnologia da Informação (TI) se refere a uma série de soluções e atividades tecnológicas, que envolvem banco de dados, hardwares, softwares e redes, usadas para lidar com as informações (Universidade Católica de Pelotas, 2021).

Além disso, a TI é apontada como um dos mais importantes componentes do ambiente empresarial atualmente, e as empresas têm, cada vez mais, utilizado a tecnologia no nível estratégico como o operacional (Albertin, 2008).

A TI possui um papel fundamental no diferencial das empresas que souberem aproveitá-la adequadamente, sendo assim, um recurso que os negócios possuem para competir e sobreviver no mercado (Rech, 2001).

Assim, pode-se considerar a TI como um elemento essencial numa empresa atualmente, pois a TI constrói toda a estrutura responsável pelo planejamento da

organização, e, além disso, atua no produto interno da empresa, podendo ser o fator decisivo na sobrevivência da empresa.

Poderemos observar no próximo item as características do mercado de TI no Brasil e assim notar como as tecnologias cresceram em importância no país.

2.2 Mercado de TI

O mercado de TI, juntamente com os demais negócios da tecnologia da informação, é um dos ramos que mais cresce e oferece melhores expectativas futuras (Pitol e Zanatta, 2009).

Segundo o estudo divulgado pela consultoria IDC (2022), a expectativa de crescimento do mercado de TI no Brasil é a maior em oito anos, onde o mercado deve crescer 10,6%.

Além disso, para Pastore (2008), o mercado de TI deve crescer mundialmente e os brasileiros precisam estar preparados para ocupar as vagas neste setor. Segundo o autor, existem diversos motivos para aproveitar as oportunidades, como:

- Grande volume de terceirização de serviços, tanto para o Brasil, quanto para outros países. Este setor se beneficia pelo fato de os produtos desenvolvidos não necessitarem o transporte;
- Países que mais precisam de mão de obra estão evitando contratar profissionais de países desenvolvidos, pelo alto custo. Isto pode ser uma grande oportunidade para os profissionais brasileiros.

Porém, Pastore (2008) também cita que a Índia tem vantagem sobre o Brasil atualmente, pois a sua população possui maior fluência na língua inglesa e isso é fundamental para a contratação de empresas de fora.

Para se manter relevante no mercado de TI, é necessário que o aluno permaneça estudando durante toda a sua carreira, pois a todo momento uma tecnologia surge e conquista o mercado. Para isso, é muito importante ao aluno a capacidade de estudar por conta própria, por isso, vamos observar no próximo item o que é a autoaprendizagem.

2.3 Autoaprendizagem

Segundo Fernandes, Scalco e Borsatto (2014), a autoaprendizagem é um método, onde o estudante adquire um conhecimento pertinente a ele, em certas áreas específicas de seu interesse. Este estudo pode ocorrer de forma independente, ou através de uma instituição de ensino. Além disso, os autores também citam os desafios da autoaprendizagem no modelo EAD:

Para ser bem-sucedido no ensino a distância, já que a autonomia é uma das maiores vantagens nessa modalidade, o aprendiz deverá desenvolver uma nova postura, ser disciplinado, organizado, estar constantemente motivado aos estudos, ser persistente, gerenciar seu tempo de estudos e adquirir hábitos de estudo sistemáticos, eficientes, através da utilização de métodos e técnicas adequadas (FERNANDO, SCALCO e BORSATTO, 2014, p. 10).

Assim, segundo os autores, uma das maiores dificuldades do aluno é ter a atitude de querer aprender, ser proativo, desenvolver uma estratégia de estudo para que haja um aprendizado de qualidade.

2.4 Tecnologias

Para se familiarizar com as ferramentas em que o software será construído, serão apresentadas brevemente as tecnologias presentes no software nos próximos itens, como o Javascript, VueJS e NodeJs.

2.4.1 Javacript

Segundo a Mozilla Foundation (2021), “JavaScript (frequentemente abreviado como JS) é uma linguagem de programação leve, interpretada e orientada a objetos com funções de primeira classe, conhecida como a linguagem de scripting para páginas Web, mas também utilizada em muitos ambientes fora dos navegadores [...]”.

Ainda, segundo Flanagan (2013), Javascript foi criada em 1995 pela empresa Netscape, responsável pelo navegador mais utilizado naquela época. A empresa decidiu criar o Javascript pois queria que o conteúdo apresentado na web pudesse ser dinâmico, pois naquela época o conteúdo só poderia ser estático. Com o grande crescimento dos computadores pessoais e assim do acesso aos navegadores, o uso

do Javascript cresceu muito, tornando-se uma das linguagens mais utilizadas no mundo atualmente.

A linguagem cresceu tanto que pode ser utilizada em outros ambientes, sem ser o navegador, como na linguagem de *back-end* NodeJS, um *framework* para criação de aplicativos React Native, nos leitores de PDF da empresa Adobe, entre outros.

2.4.2 VueJS

Vue é um *framework* progressivo do Javascript para a construção de interfaces de usuário (Documentação VueJS, 2021).

O Vue foi criado por Evan You, depois de trabalhar com AngularJS no Google. O seu objetivo era extrair a parte que mais gostava do Angular em um *framework* mais leve. O Vue fez tanto sucesso que foi apontado como uma grande promessa no Github, pois possui a maioria das estrelas entre os projetos de código aberto no site (DEVMEDIA, 2017).

Código 1 – Exemplo de um arquivo .vue

```
1 <template>
2   <div>
3     <!-- HTML -->
4   </div>
5 </template>
6
7 <script>
8   export default {
9     name: 'Nome',
10    data() {
11      return{
12        // variáveis
13      }
14    },
15
16    mounted() {
17      // métodos que são executados quando a página é montada
18    },
19
20    methods: {
21      // métodos
22    }
23  }
24 </script>
25
26 <style>
27   /* CSS */
28 </style>
```

Fonte: elaborado pelo autor (2022)

No Código 1 podemos visualizar um arquivo padrão de Vue. Esse arquivo contém a estrutura básica de funcionamento individual de uma página ou componente no projeto.

Na primeira tag “*template*”, é onde vai o código HTML, criando a estrutura básica da página. Na tag “*script*” vai todo o código Javascript individual da página, com as importações de outros arquivos antes do “*export default*”, logo após vem o nome do arquivo. As variáveis globais no arquivo são criadas dentro do objeto do *return* do método *data()*. Os métodos que o desenvolvedor vai utilizar ficam dentro do objeto “*methods*”.

Na tag “*style*”, vai todo o CSS da página, trazendo assim o estilo dentro da página.

2.4.3 Vuex

Existe algo dentro do Vue chamado de estado. Estado é a “versão” da variável dentro de um único componente do Vue. Isso é bem simples e fácil de implementar, porém a complexidade aumenta muito quando é necessário dividir o mesmo estado de uma variável em muitos componentes (Documentação Vuex, 2021).

Para aplicações simples, o Vuex pode parecer sem sentido, pois não existem muitos componentes e páginas, então o compartilhamento de estados de uma variável é fácil. Porém, em aplicações de média e grande escala o uso de Vuex se torna necessário, pois o compartilhamento de estados das variáveis é complexo e suscetível à erros (Documentação Vuex, 2021).

2.4.4 Node.JS

O Javascript foi criado para ser executado no lado do cliente, ou seja, pelo navegador do usuário, porém, com a evolução da internet, passou a ser utilizada também no *server-side*. (Rockcontent, 2020)

O Node.js foi criado por Ryan Dahl, em 2009, após uma análise em páginas da web e como elas funcionavam. Ryan percebeu que as aplicações não eram

eficientes, utilizando mais recursos que o necessário. Então percebeu a necessidade de criar uma solução *back-end* eficiente. (X-apps, 2019)

Com o Node.js, é possível criar aplicações Javascript para serem executadas em uma máquina, não dependendo de um *browser* para a execução. O Node.js fez muito sucesso pois é uma solução de *back-end* escalável, barata (pois não consome muita memória da máquina) e flexível. Por isso passou a ser amplamente utilizado desde a sua criação, em 2009. (Opus Software, 2018)

2.5 Sistemas de recomendação

A internet nos apresenta uma quantidade muito grande de informações. Um aluno novo na programação se depara com diversos sites que lhe apresentam caminhos diferentes na área, ele geralmente não possui o conhecimento necessário para escolher qual o caminho mais adequado para seguir. Para isso, os sistemas de recomendação atuam como uma ferramenta importante nas plataformas atuais da web.

Reategui e Cazella trazem um conceito muito interessante sobre os sistemas de recomendação na internet atual:

Recomendar produtos, itens ou informações para um usuário da Internet apresenta-se como um dos maiores desafios no mundo virtual. A recomendação adequada de um livro, por exemplo, pode fazer a diferença entre conquistar o usuário ou perdê-lo. Devido a esta necessidade de conquista, a personalização tem se apresentado como um fator facilitador no momento de "cativar" o usuário (REATEGUI; CAZELLA, 2005).

Podemos notar que os sistemas de recomendação já estão presentes em diversos sites, como as lojas virtuais, que fornecem recomendações de produtos baseadas nas buscas ou compras anteriores do usuário. Um sistema de recomendação pode ser de diferentes naturezas, como uma informação, um produto, um curso, ou até um objeto.

3 TAXONOMIA DAS FERRAMENTAS DE ENSINO EXISTENTES

A presente seção trará uma análise e categorização das diversas soluções de ensino de programação presentes no mercado atualmente. A categorização se dará pelas características de cada plataforma de ensino, seus objetivos e seu público-

alvo. O objetivo dessa análise é expor ao leitor o objetivo da criação desta nova plataforma de ensino e onde ela estará presente dentro deste mercado de ensino de programação.

Para construir esta taxonomia, foi necessário classificar os cursos em diversas categorias, se o curso é pago ou não, a forma de ensino da plataforma (interativo, por vídeo e/ou por textos) e se é necessária alguma experiência na programação para utilizar a plataforma. Após isso, foi realizada uma busca nas principais plataformas de ensino de programação online, para separar as principais características de cada uma, para assim ser possível a categorização.

Na pesquisa, foram encontrados onze cursos online, dos quais estavam relacionados com o ensino à distância e contém conteúdo de programação presentes na plataforma. Como poderemos ver abaixo, algumas plataformas são exclusivas para programação, enquanto outras apresentam uma grande gama de conteúdos, porém estão nesta lista pois são constantemente utilizadas para o ensino de programação.

3.1 Cursos encontrados

Como mencionado anteriormente, após a pesquisa, foram encontradas onze plataformas de ensino de programação online, que são:

- Udemy¹
- Udacity²
- Coursera³
- Codecademy⁴
- Skillshare⁵
- Alura⁶
- freeCodeCamp⁷
- W3Schools⁸

¹ <https://www.udemy.com/>

² <https://www.udacity.com/>

³ <https://pt.coursera.org/>

⁴ <https://www.codecademy.com/>

⁵ <https://www.skillshare.com/>

⁶ <https://www.alura.com.br/>

⁷ <https://www.freecodecamp.org/>

⁸ <https://www.w3schools.com/>

- HackerRank⁹
- Code Wars¹⁰
- MDN Web Docs¹¹

3.2 Classificações

Agora, serão apresentadas as classificações propostas para as plataformas de ensino. A classificação foi dividida em três grandes grupos, cada grupo contém seus subgrupos:

1. Modelo de monetização:
 - a. Cursos gratuitos;
 - b. Cursos pagos.
2. Modelo de ensino:
 - a. Cursos interativos;
 - b. Cursos por vídeo;
 - c. Cursos por texto.
3. Modelo de abordagem:
 - a. Cursos que exigem algum conhecimento prévio em programação;
 - b. Cursos introdutórios.

3.2.1 Modelo de monetização

Este modelo de monetização se refere à forma do curso monetizar. Existem dois subgrupos neste modelo, os cursos gratuitos e os cursos pagos, porém nem todos os cursos se encaixam exatamente em cada subgrupo, como poderemos ver a seguir.

Cursos gratuitos são plataformas totalmente grátis para todos os usuários, onde eles podem consumir os conteúdos sem qualquer limite. Já os cursos pagos, são plataformas onde o aluno precisa comprar o curso para ter acesso ao conteúdo. Isso pode ser feito de algumas formas, que são:

⁹ <https://www.hackerrank.com/>

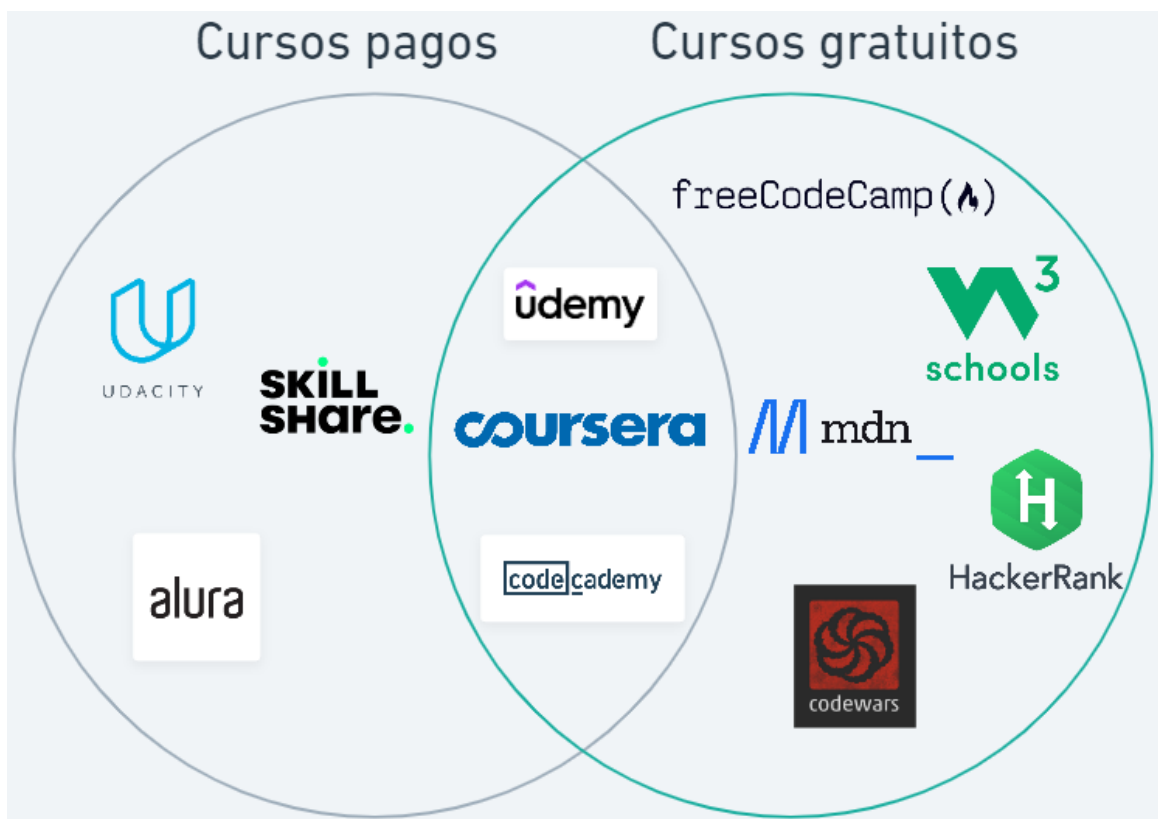
¹⁰ <https://www.codewars.com/>

¹¹ <https://developer.mozilla.org/pt-BR/>

1. Assinatura: o aluno deve pagar um valor mensal para ter acesso a todo o conteúdo da plataforma, onde ele pode consumir tudo, enquanto a assinatura estiver ativa;
2. Compra seletiva: o aluno compra um conteúdo específico dentro da plataforma e ele tem acesso ao conteúdo de forma vitalícia, porém os outros conteúdos continuam bloqueados.

Na Figura 1 podemos ver a classificação dos cursos encontrados no Modelo de monetização.

Figura 1 – Classificação do Modelo de monetização



Fonte: elaborado pelo autor (2022)

3.2.2 Modelo de ensino

O modelo de ensino é referente ao modo da plataforma transmitir o conhecimento para o aluno, e, para isto, existem diferentes formas. Podemos citar três:

1. Cursos interativos: São cursos que apresentam ao aluno uma IDE, onde ele pode escrever o código, ele é compilado assim que o aluno finalizar a edição. Caso o código execute da maneira correta e faça o que o enunciado solicitou dele, o aluno pode passar para a próxima sessão.
2. Cursos por vídeo: São cursos que apresentam ao aluno vídeos com o conteúdo da aula e cabe ao aluno acompanhar o tema. Caso ache necessário, pode reescrever o conteúdo em sua própria IDE.
3. Cursos por texto: São cursos onde os conteúdos são apresentados em forma de texto. O aluno pode ler toda a explicação do tema e observar também pedaços de código presentes na apresentação, para servir de exemplo no conteúdo.

Na Figura 2, podemos observar a classificação dos cursos encontrados no Modelo de ensino.



Figura 2 – Classificação do Modelo de ensino

Fonte: elaborado pelo autor (2022)

3.2.3 Modelo de abordagem

O modelo de abordagem se refere ao tipo de conteúdo presente na plataforma. O conteúdo pode ser focado em programadores iniciantes, que possuem ou não alguma experiência profissional com programação, ou programadores experientes, que já dominam o básico e querem aprender ferramentas ou técnicas novas para se desenvolver profissionalmente.

Os cursos introdutórios apresentam ao aluno um conteúdo básico, em que todo programador deve dominar para entrar no mercado de trabalho. Geralmente estes conteúdos são HTML, CSS e Javascript. Já os cursos que exigem um conhecimento prévio em programação focam no público já experiente no mercado, que deseja evoluir na carreira, ou aprendendo uma ferramenta nova, ou se especializando na ferramenta que trabalha atualmente.

Na Figura 3 podemos observar a classificação dos cursos encontrados no Modelo de abordagem.

Figura 3 – Classificação do Modelo de abordagem



Fonte: elaborado pelo autor (2022)

3.3 Tabela comparativa

Agora, será apresentada uma tabela comparativa entre os modelos com todos os cursos encontrados. Esta tabela tem como finalidade a apresentação mais visual da taxonomia, pois será possível observar os três modelos em apenas uma tabela.

Tabela 1 – Tabela Comparativa

| Cursos | Gratuito | Pago | Interativo | Vídeo | Texto | Introdutório | Especialização |
|---------------|-----------------|-------------|-------------------|--------------|--------------|---------------------|-----------------------|
| Udemy | X | X | | X | | X | X |
| Udacity | | X | | X | | | X |
| Coursera | X | X | | X | | | X |
| Codecademy | X | X | X | | | X | |
| Skill share | | X | | X | | X | |
| Alura | | X | | X | X | X | |
| freeCodeCamp | X | | X | | X | X | |
| W3Schools | X | | X | | X | X | |
| HackerRank | X | | X | | | | X |
| Code Wars | X | | X | | | | X |
| MDN Web Docs | X | | | | X | X | |

Fonte: elaborado pelo autor (2022)

3.4 Considerações sobre a taxonomia

Após a apresentação das diferentes classificações dos cursos, podemos concluir que o mercado atualmente está saturado de opções para o aluno aprender programação, onde ele pode buscar opções gratuitas que ensinam o básico para ele e após uma experiência no mercado, ele pode procurar cursos que o ofereçam uma especialização em determinado assunto.

Porém o objetivo do presente projeto não é criar mais um curso de ensino de programação online, mas sim, um site onde o aluno completamente novo nesta área possa aprender como a área funciona, quais os caminhos ele pode seguir e assim

torná-lo mais confiante para entrar na programação. Muitos alunos iniciantes se desmotivam no início dos estudos e isso se dá por diversos motivos, porém um deles é que eles não possuem muita ideia de como iniciar, por onde iniciar e qual o caminho mais prático para se desenvolver.

Este site irá apresentar ao aluno um mapa, que tem como objetivo organizar os tópicos introdutórios, assim o aluno poderá aprender mais sobre a área antes de iniciar algum ensino técnico em alguma ferramenta. Este mapa será em formato de trilha e esta trilha irá conter um conteúdo por etapa, então o aluno irá assistir vídeos, ou ler textos e quando concluir a etapa ele poderá marcar se gostou ou não do conteúdo e no final ele receberá uma recomendação de qual área ele poderá seguir, com base nas avaliações dele em cada aula. Além da área, serão apresentados alguns cursos específicos, para que o aluno iniciante possa realizar um curso de qualidade.

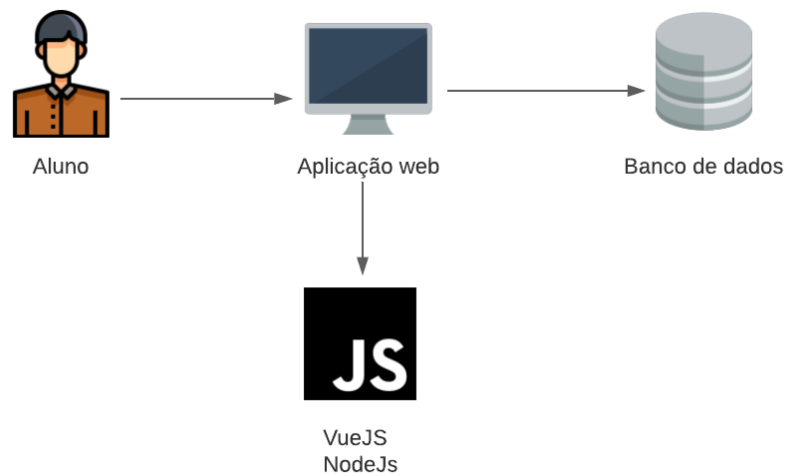
4 METODOLOGIA

Nesta seção será apresentada a implementação da plataforma web. A seção será dividida entre a arquitetura do sistema, apresentando diferentes diagramas para demonstrar como o sistema foi desenvolvido, além dos requisitos funcionais e não funcionais. Também irá descrever a implementação em si, mostrando trechos de código e fotos das telas do sistema.

4.1 Arquitetura do Sistema

A arquitetura do sistema foi desenvolvida de forma simples, onde a aplicação, desenvolvida com *frameworks* Javascript, possui um banco de dados para armazenar os dados dos usuários, aulas e a avaliação dos usuários em cada aula. Podemos ver a Figura 4 demonstrando de forma simples a arquitetura do sistema.

Figura 4 – Arquitetura do sistema



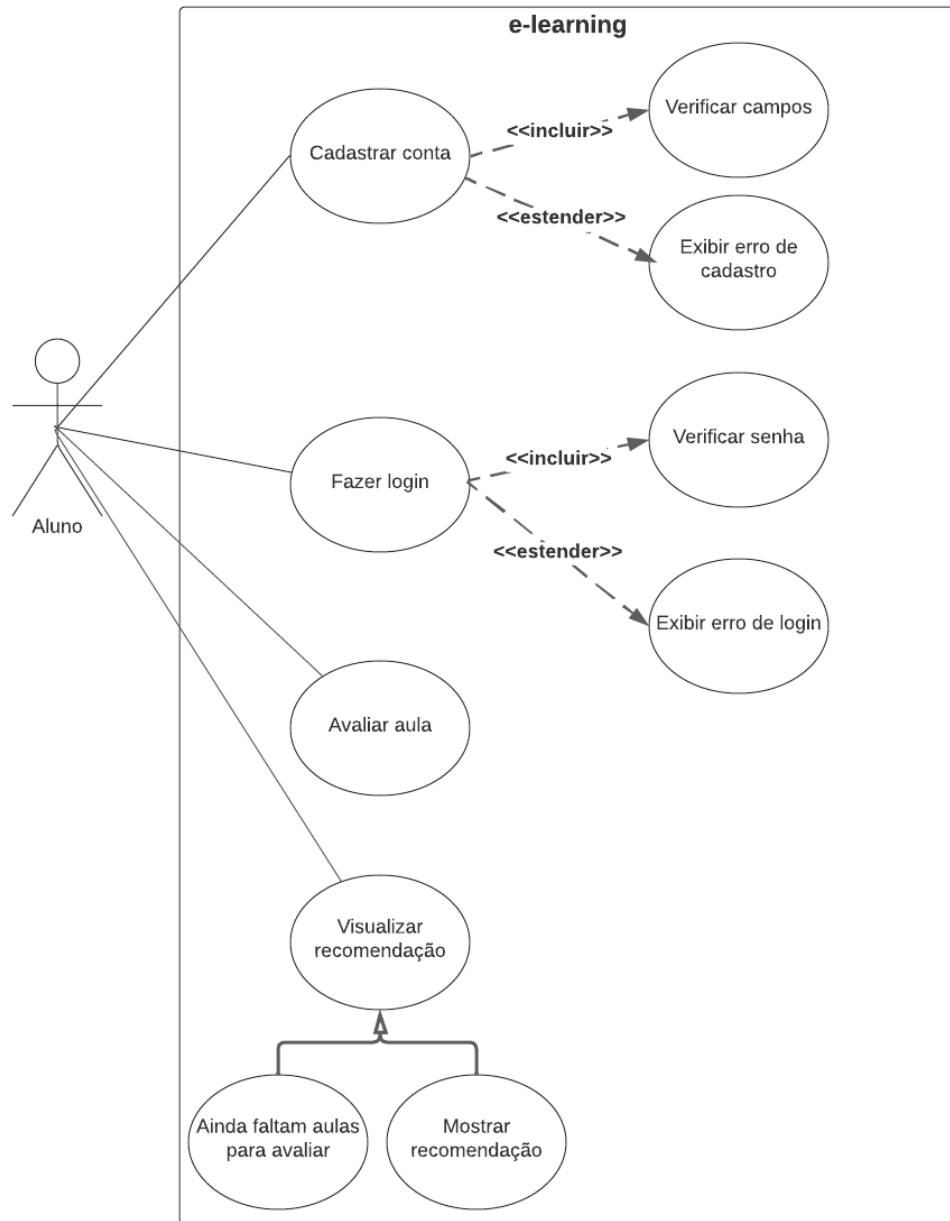
Fonte: elaborado pelo autor (2022)

4.1.1 Diagrama de casos de uso

O diagrama de casos de uso serve para demonstrar as funcionalidades do sistema, onde cada caso de uso do usuário demonstra um requisito do sistema e como ele se comporta em cada caso.

A Figura 5 representa os casos de uso da aplicação, onde o aluno é responsável por toda a interação com o sistema

Figura 5 – Diagrama de casos de uso

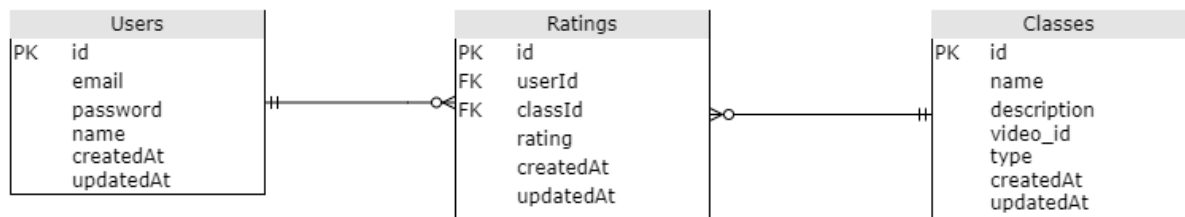


Fonte: elaborado pelo autor (2022)

4.1.2 Diagrama de ER

O diagrama de ER tem como objetivo demonstrar as entidades do banco e seus relacionamentos. Podemos observar na Figura 6 as três tabelas criadas para a implementação do sistema e o relacionamento de cada tabela.

Figura 6 – Diagrama de ER



Fonte: elaborado pelo autor (2022)

A primeira tabela de Users é responsável por armazenar todas as informações dos usuários, como nome, e-mail e senha.

A tabela de Classes é responsável por armazenar as informações das aulas. Nela são armazenados os seguintes campos:

- Nome;
- Descrição do conteúdo;
- Id do vídeo no YouTube;
- Tipo.
 - Básico: É um conteúdo que não se encaixa nem no *front-end*, nem no *back-end*. Porém são conceitos básicos importantes que o aluno precisa conhecer;
 - *Front-end*: Conteúdo voltado para a área de *front-end*;
 - *Back-end*: Conteúdo voltado para a área de *back-end*.

E por último, a tabela Ratings é responsável pelo relacionamento entre a tabela Users e a tabela Classes. Nela constam os seguintes campos:

- *userId*: Id do usuário responsável pela avaliação;
- *classId*: Id da aula avaliada;
- *rating*: Avaliação do usuário para a aula específica.
 - Caso o aluno não goste do conteúdo, o valor armazenado no banco é 0;
 - Caso o aluno goste do conteúdo, o valor armazenado é 1.

4.1.3 Requisitos do sistema

Esta seção apresenta os requisitos funcionais e não funcionais, além das regras de negócio. Requisitos funcionais são todas as funcionalidades que o

software contém, ou seja, todos os problemas ou necessidades devem ser atendidos pelos requisitos funcionais. Já os requisitos não funcionais definem como o sistema funcionará, em questão de desempenho, segurança, usabilidade, tecnologias envolvidas, entre outros. As regras de negócios são declarações sobre como uma empresa conduz seus negócios, descrevendo as premissas e restrições que a operação deve seguir.

Estes requisitos são de importante conhecimento pois eles descrevem de forma simples e direta todas as funcionalidades que o sistema deve possuir e ajudam a entender exatamente o que é esperado do software.

4.1.3.1 Requisitos funcionais

RF01 – O sistema deverá permitir a criação de um usuário;

RF02 – O sistema deverá permitir que o usuário entre na plataforma, através do e-mail e senha cadastrados;

RF03 – O sistema deverá permitir ao usuário a visualização do conteúdo em cada aula da trilha;

RF04 – O sistema deverá permitir a avaliação do conteúdo da aula pelo usuário;

RF05 – O sistema deverá apresentar ao usuário a recomendação de qual caminho seguir na programação, após a avaliação de todas as aulas.

4.1.3.2 Requisitos não funcionais

RNF01 – O sistema deverá funcionar em todos os navegadores atuais;

RNF02 – O sistema deverá ser desenvolvido na linguagem Javascript;

RNF03 – O sistema deverá utilizar o banco de dados MySQL para o armazenamento de dados.

4.1.3.3 Regras de negócio

RN01 – Apenas alunos logados poderão acessar a trilha e receber recomendações;

RN02 – O aluno deve avaliar todas as aulas para receber a recomendação.

4.2 Implementação

Para a escrita do código do sistema foi utilizado o Visual Studio Code, um ambiente de desenvolvimento criado pela Microsoft. Para o desenvolvimento do *back-end* do sistema, foi utilizado o *framework* NodeJS. A escolha pelo NodeJS se deu por conta de vários motivos, como a flexibilidade do Javascript, onde é possível utilizar a ferramenta de gerenciamento de pacotes NPM para instalar bibliotecas no projeto, tornando o desenvolvimento mais ágil e moderno. Também foi levado em consideração a leveza e o suporte do NodeJS, já que para utilizar um ambiente do *framework* não exige muitos recursos computacionais e as principais empresas de armazenamento em nuvem atuais possuem suporte para o *framework*.

Para o armazenamento dos dados do usuário, aulas e avaliações, foi utilizado o banco de dados MySQL. O MySQL é um banco de dados relacional, ou seja, os dados são armazenados em tabelas e os dados são relacionados entre si. Alguns dos motivos para a escolha do MySQL é o alto desempenho, a segurança e o suporte, pois é um dos bancos de dados mais utilizados no mercado.

No *front-end*, foi utilizado o *framework* de Javascript VueJs. Este *framework* foi escolhido por ser leve e moderno e por permitir a utilização de componentes para a montagem das telas, reutilizando código e reduzindo o tamanho final do projeto.

4.2.1 Implementação do Back-end

A implementação do *back-end* foi dividida em três partes, a parte de usuários, a de aulas e a de avaliações. Com o objetivo de facilitar o entendimento, veremos nesta seção cada implementação individualmente.

4.2.1.1 Serviço de usuários

No serviço de usuários, foram criadas todas as funcionalidades necessárias para gerenciar os usuários, como a criação, a autenticação e a remoção.

Para a criação são necessárias três informações: nome, e-mail e senha. No método de criação de usuário, é verificado se já existe um e-mail idêntico armazenado no banco de dados, caso exista, ele retorna um erro, caso não exista, ele cria um registro na tabela de Users com as informações do usuário.

```
const create = async (name, email, password) => {
  const validate = await User.findOne({ where: { email } });

  if (validate) return { message: 'Email já cadastrado' };

  const newUser = await User.create({ name, email, password });

  const token = jwt.sign(
    { userId: newUser.id, email },
    process.env.JWT_SECRET,
    { expiresIn: '7d', algorithm: 'HS256' },
  );

  return { token, email, name: newUser.name, id: newUser.id };
}
```

Fonte: elaborado pelo autor (2022)

Para a exclusão do usuário, é necessário apenas o id dele. Assim o método busca pelo id no banco e exclui ele, caso não encontre o id no banco ele retorna um erro.

Código 3 – Método exclui

```
const exclude = async (id) => {
  const validate = await User.findOne({ where: { id } });

  if (!validate) return { message: 'Usuário não encontrado' };

  await User.destroy({ where: { id } });
}
```

Fonte: elaborado pelo autor (2022)

Para a autenticação, são necessários o e-mail e a senha. O método busca por essas informações no banco e retorna erro caso não encontre, caso encontre, retorna um *token* JWT e o nome de usuário. O *token* é necessário para realizar outras requisições, pois com ele é possível identificar qual usuário está realizando a requisição e é mais seguro que utilizar apenas o id dele.

Código 4 – Método login

```
const login = async (email, password) => {
  const userLogin = await User.findOne({ where: { email, password } });

  if (!userLogin) return { message: 'E-mail e/ou senha incorretos!' };

  const token = jwt.sign(
    { userId: userLogin.id, email },
    process.env.JWT_SECRET,
    { expiresIn: '7d', algorithm: 'HS256' },
  );

  return { token, name: userLogin.name };
}
```

Fonte: elaborado pelo autor (2022)

4.2.1.2 Serviço de aulas

No serviço de aulas, foram criados os dois métodos necessários para o gerenciamento das aulas.

O primeiro método busca todas as aulas cadastradas na tabela Classes e retorna eles juntamente com a avaliação do usuário para cada aula específica, caso exista.

Código 5 – Método getAll

```
const getAll = async (userId) => {
  let classes = await Class.findAll({
    include: { model: Rating, as: 'ratings', where: { userId }, attributes: ['rating'], required: false },
  });

  if (!classes) return { message: 'Algo deu errado' };

  return { classes };
}
```

Fonte: elaborado pelo autor (2022)

O segundo método busca uma aula específica, utilizando o id passado na rota. Nele é retornado todas as informações específicas daquela aula, este método é utilizado nas páginas específicas das aulas, onde o aluno visualiza o conteúdo e avalia a aula.

Código 6 – Método getOne

```
const getOne = async (id) => {
  const classes = await Class.findOne({where: { id }});

  if (!classes) return { message: 'Aula não encontrada' };

  return classes;
};
```

Fonte: elaborado pelo autor (2022)

4.2.1.3 Serviço de avaliações

O serviço de avaliações é o serviço intermediário entre os usuários e suas avaliações, onde contém todos os métodos necessários para esta funcionalidade.

O primeiro método é o de busca de avaliação específica da aula. Este método requer duas informações, o id da aula e o id do usuário. Com estas informações, ele busca na tabela Ratings se existe alguma avaliação na aula específica pelo usuário específico, caso exista, retorne esta avaliação.

Código 7 – Método getRatingByClass

```
const getRatingByClass = async (classId, userId) => {
  const rating = await Rating.findOne({
    where: {
      classId,
      userId
    }
  });

  if (!rating) return { message: 'Rating não encontrado' };

  return rating;
};
```

Fonte: elaborado pelo autor (2022)

O segundo método é o método de salvar avaliações. Este método é responsável por salvar a avaliação específica de cada aula para cada usuário. Ele verifica se já existe uma avaliação nesta aula para este usuário, caso exista retorna erro. Caso o usuário avalie a aula como positiva, é armazenado o valor 1 no campo de rating, caso ele não goste do conteúdo, é armazenado 0.

Código 8 – Método saveRating

```
const saveRating = async (classId, userId, rating) => {  
  const validation = await Rating.findOne({  
    where: {  
      classId,  
      userId  
    }  
  });  
  
  if (validation) return { message: 'Rating já cadastrado' };  
  
  const newRating = await Rating.create({ classId, userId, rating });  
  
  return newRating;  
}
```

Fonte: elaborado pelo autor (2022)

O terceiro método é o responsável por calcular e retornar a recomendação ao usuário. Ele usa como parâmetro apenas o id do usuário e busca na tabela de Ratings todas as avaliações daquele usuário específico e juntamente com isso, busca o valor do campo *type* da tabela de Classes em que exista uma avaliação do usuário.

Este campo *type* contém o tipo de cada aula, que pode ser *front-end* ou *back-end*. Com esse *type*, é realizada a soma de todas as avaliações positivas em cada tipo de conteúdo e o conteúdo que possuir mais avaliações positivas é retornado.

Este método também verifica se o usuário avaliou todas as aulas disponíveis, caso não tenha avaliado todas as aulas, ele retorna a quantidade de aulas que ainda não foram avaliadas.

Código 9 – Método getRecommendation

```

const getRecommendation = async (userId) => {
  const ratingList = await Rating.findAll({
    where: {
      userId
    },
    include: { model: Class, as: 'classes', attributes: ['type'] },
  });

  if (!ratingList) return { message: 'Nenhuma recomendação' };

  if (ratingList.length < 11) {
    if (11-ratingList.length > 1) {
      return { message: `Ainda restam ${11-ratingList.length} aulas para você completar.` };
    } else {
      return { message: `Ainda resta ${11-ratingList.length} aula para você completar.` };
    }
  }

  let frontend = 0;
  let backend = 0;

  ratingList.forEach(({classes, rating}) => {
    if (rating === 1) {
      if (classes.type === "frontend") {
        frontend += 1;
      }
      if (classes.type === "backend") {
        backend += 1;
      }
    }
  });

  if (frontend > backend) {
    return "Front-end";
  } else if (backend > frontend) {
    return "Back-end";
  } else {
    return "Fullstack"
  }
};

```

Fonte: elaborado pelo autor (2022)

4.2.1 Implementação do Front-end

O *front-end* da aplicação é responsável por toda interação do usuário com a aplicação. Esta subseção descreve cada tela e seus componentes, a lógica da implementação e como tudo funciona na parte do cliente.

4.2.1.1 Rotas do Front-end

Existem quatro rotas declaradas no *front-end*, que são:

- Home: a rota da home responde pelo caminho “/” na url. Nela estão presentes todas as aulas.
- Class: a rota da aula específica responde pelo caminho de “/aula/:id”. O **id** presente na rota é o id da aula no banco de dados e o id da rota muda conforme a aula.
- *Dashboard*: a rota de *dashboard* responde pelo caminho “/seu-caminho”. Ela é a responsável por apresentar ao usuário a recomendação, caso ele já tenha avaliado todas as aulas. E um tutorial de como o sistema funciona, caso ele ainda não tenha avaliado todas as aulas.
- Login: a rota de login responde pelo caminho de “/login”. Ela é responsável tanto pelo login, quanto pelo cadastro do usuário.

Código 10 – Rotas do Front-end

```
const routes = [  
  {  
    path: '/',  
    name: 'home',  
    component: () => import('../views/Home/HomeView.vue'),  
    meta: {  
      requiresAuth: true  
    }  
  },  
  {  
    path: '/login',  
    name: 'login',  
    component: () => import('../views/Login/LoginView.vue')  
  },  
  {  
    path: '/aula/:id',  
    name: 'class',  
    component: () => import('../views/Class/ClassView.vue'),  
    meta: {  
      requiresAuth: true  
    }  
  },  
  {  
    path: '/seu-caminho',  
    name: 'dashboard',  
    component: () => import('../views/Dashboard/DashboardView.vue'),  
    meta: {  
      requiresAuth: true  
    }  
  }  
]
```

Fonte: elaborado pelo autor (2022)

4.2.1.2 Tela de Login

A tela de login é responsável pelo cadastro e login do usuário. A tela é a mesma para ambas as funcionalidades, apenas os formulários se alternam conforme a necessidade do usuário. Quando ele clica em *Cadastre-se*, o componente de Cadastro é apresentado e quando ele clica em *Voltar*, o componente de Login é apresentado.

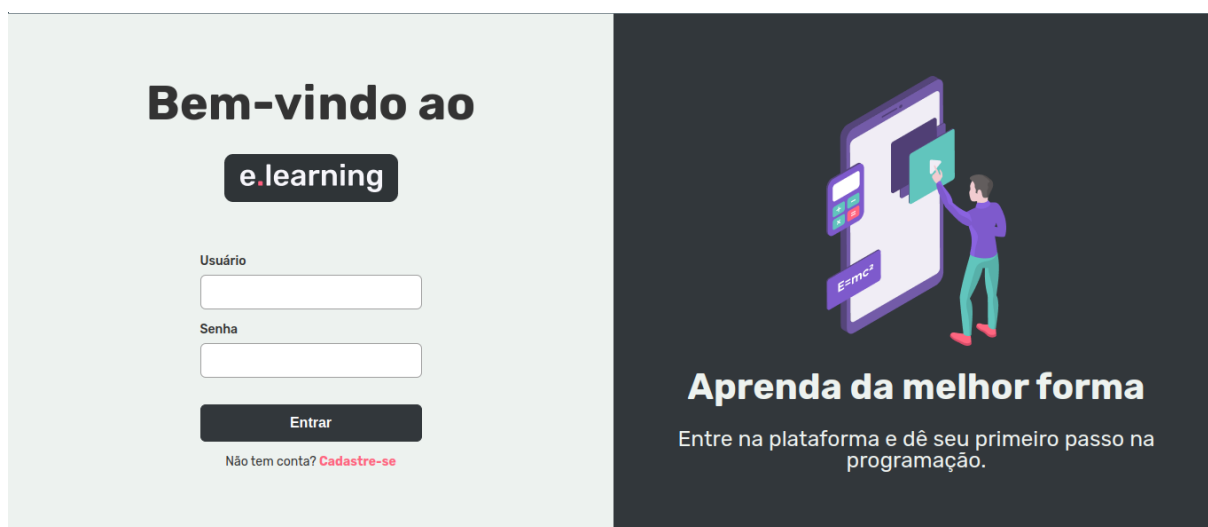
Existem algumas regras de validação nos campos dos formulários. Para estas validações foi utilizado uma biblioteca para VueJS chamada *Vuelidate*, pois é uma biblioteca amplamente utilizada e confiável.

Para o formulário de cadastro, as regras para a validação nos campos são as seguintes:

- Campo de e-mail: O campo é obrigatório e deve ser em formato de e-mail (com o @ e .com presentes).
- Campo de nome: O campo é obrigatório.
- Campo de senha: O campo é obrigatório e deve possuir no mínimo seis caracteres.

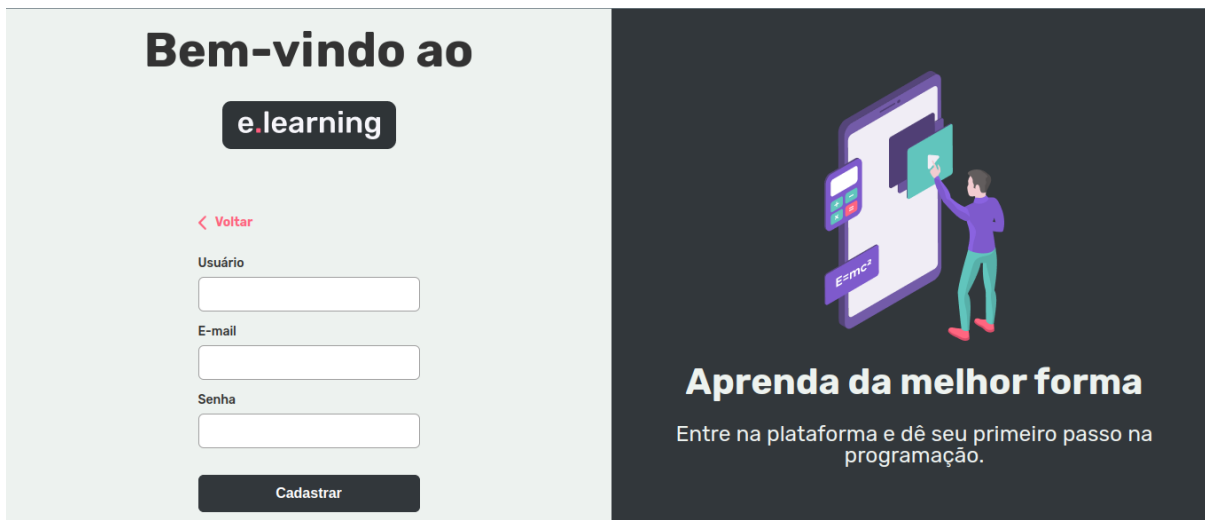
Para o formulário de login, as validações são as mesmas, porém não é necessário informar o nome para logar, apenas o e-mail e a senha.

Figura 7 – Tela de login



Fonte: elaborado pelo autor (2022)

Figura 8 – Tela de cadastro



Bem-vindo ao

e.learning

[< Voltar](#)

Usuário

E-mail

Senha

Cadastrar

Aprenda da melhor forma
Entre na plataforma e dê seu primeiro passo na programação.

Fonte: elaborado pelo autor (2022)

4.2.1.3 Tela Home

A tela *Home* é responsável por listar todas as aulas disponíveis na trilha. Cada aula é renderizada na tela pelo componente *ClassCard*, onde cada *card* é composto por uma imagem, título, descrição e a avaliação do usuário para aquela aula específica, caso possua.

Figura 9 – Tela Home



e.learning [Suas aulas](#) [Seu caminho](#) [Fredri](#)

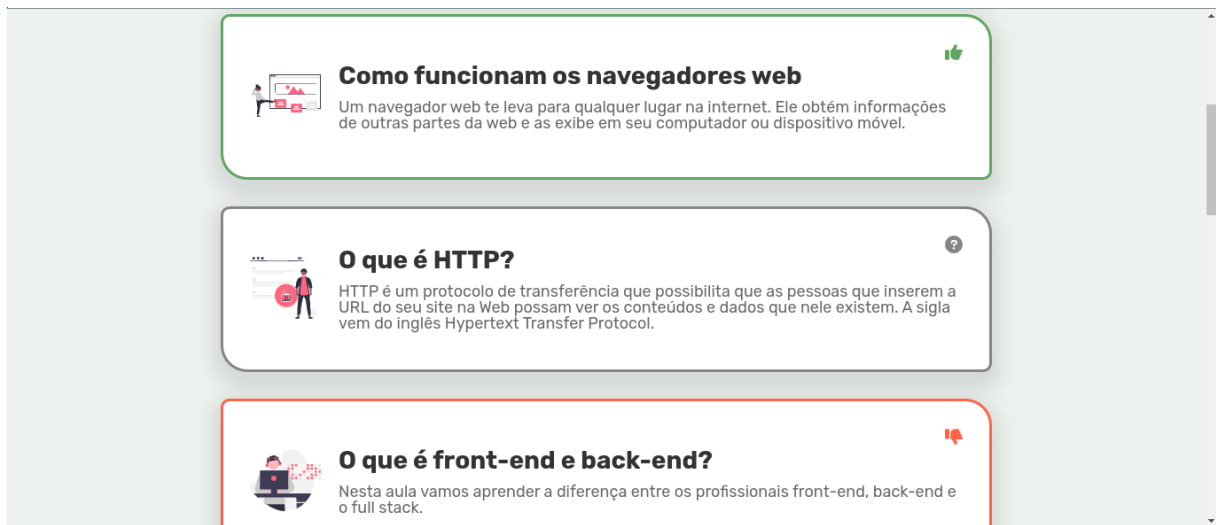
Pesquise pelo título

Como funciona a internet
Nesta aula vamos aprender como funciona a internet. Vamos ver o que acontece por baixo dos panos quando você digita uma URL no navegador.

Como funcionam os navegadores web
Um navegador web te leva para qualquer lugar na internet. Ele obtém informações de outras partes da web e as exibe em seu computador ou dispositivo móvel.

Fonte: elaborado pelo autor (2022)

Figura 10 – Cards na Home com cada um dos feedbacks



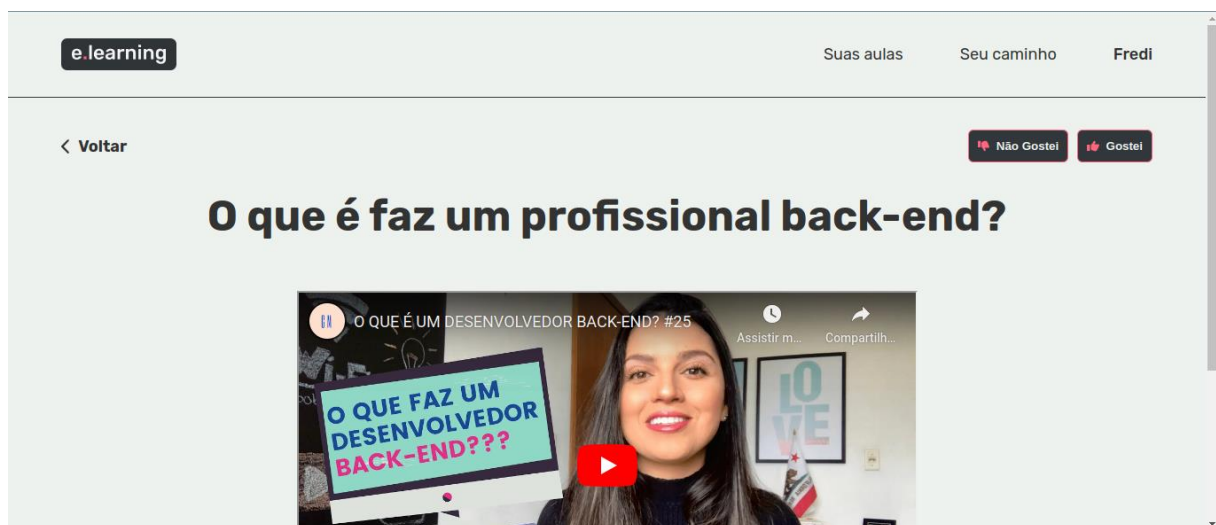
Fonte: elaborado pelo autor (2022)

4.2.1.4 Tela de Aula

A tela da aula é a tela onde o aluno pode visualizar o conteúdo e avaliar a aula. O conteúdo é apresentado por um vídeo do YouTube, que contempla tudo que o aluno precisa conhecer sobre o conteúdo e uma breve descrição do tema.

É nesta tela que o usuário pode avaliar o conteúdo, como *Gostei* ou *Não Gostei*. Quando ele seleciona uma opção, os dois botões são desabilitados, dando destaque somente ao botão que o aluno selecionou. É com essa avaliação que o sistema calcula a recomendação final.

Figura 11 – Tela de Aula



Fonte: elaborado pelo autor (2022)

4.2.1.5 Tela de Seu Caminho

Nesta tela será apresentado ao usuário um tutorial da plataforma, como ela funciona e o que deve ser feito para receber a recomendação final. Como regra de negócio, o sistema deve apenas apresentar ao usuário uma recomendação após ele avaliar todas as aulas disponíveis na trilha. Enquanto o usuário não avaliou todas as aulas, a tela mostra um rápido tutorial da plataforma, como podemos ver na Figura 12.

Figura 12 – Tela de Seu Caminho com o tutorial

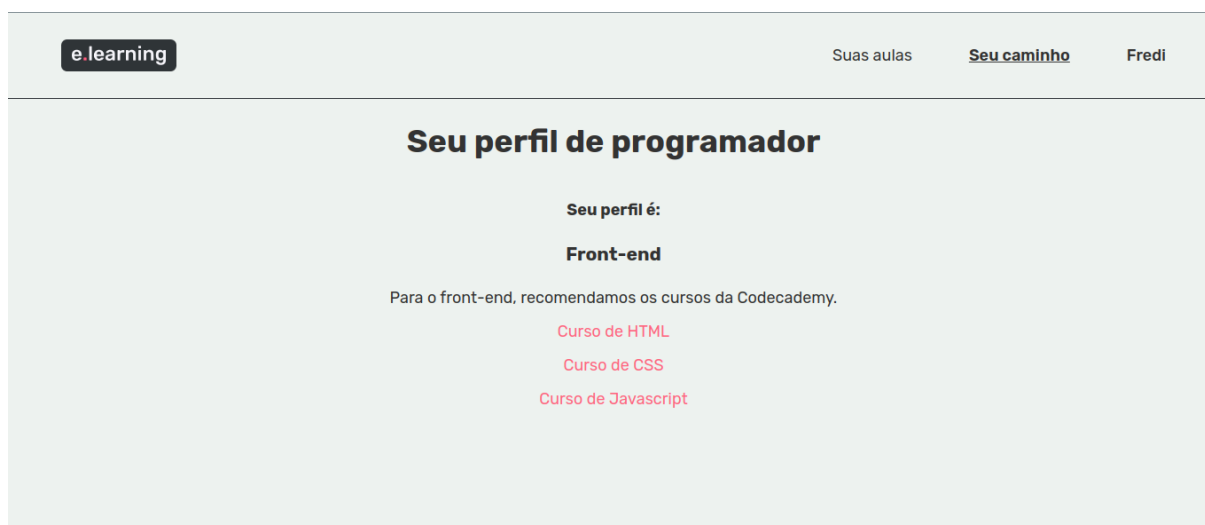


Fonte: elaborado pelo autor (2022)

Após o aluno ter avaliado todas as aulas, o *layout* da tela muda e agora ao invés do tutorial, é apresentada a recomendação ao usuário. O funcionamento da recomendação já foi apresentado na Subseção 4.2.1.3, então será apresentado aqui apenas o *layout* em cada recomendação (*Front-end*, *Back-end* e *Fullstack*).

Quando o usuário gostou mais de conteúdos *front-end*, é apresentado para ele a recomendação de *Front-end* e cursos recomendados para esta área. Foram escolhidos os cursos de HTML, CSS e JavaScript da CodeCademy, pois são cursos extremamente aceitos na comunidade e completos, além disso possuem uma ótima didática.

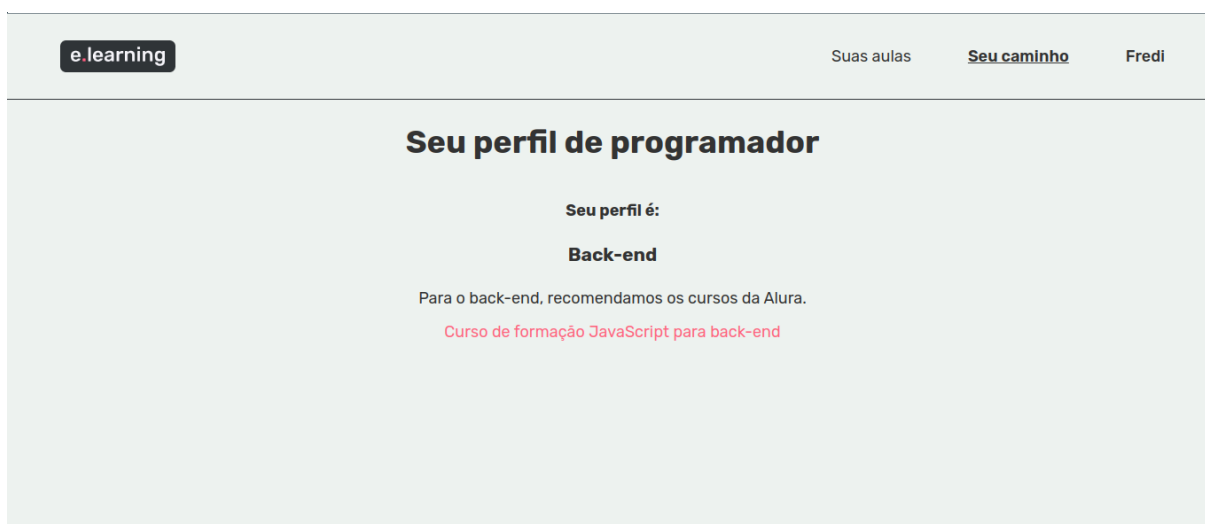
Figura 13 – Tela de recomendação Front-end



Fonte: elaborado pelo autor (2022)

Caso o usuário tenha avaliado como positivo mais conteúdos de *back-end*, será apresentado a recomendação de *Back-end*, assim como os cursos recomendados. Para a área de *back-end*, foi escolhido o curso de JavaScript para *back-end* da Alura, pois este curso contém todo o conteúdo necessário para iniciar na área, além de ser um curso muito conhecido e atual.

Figura 14 – Tela de recomendação Back-end

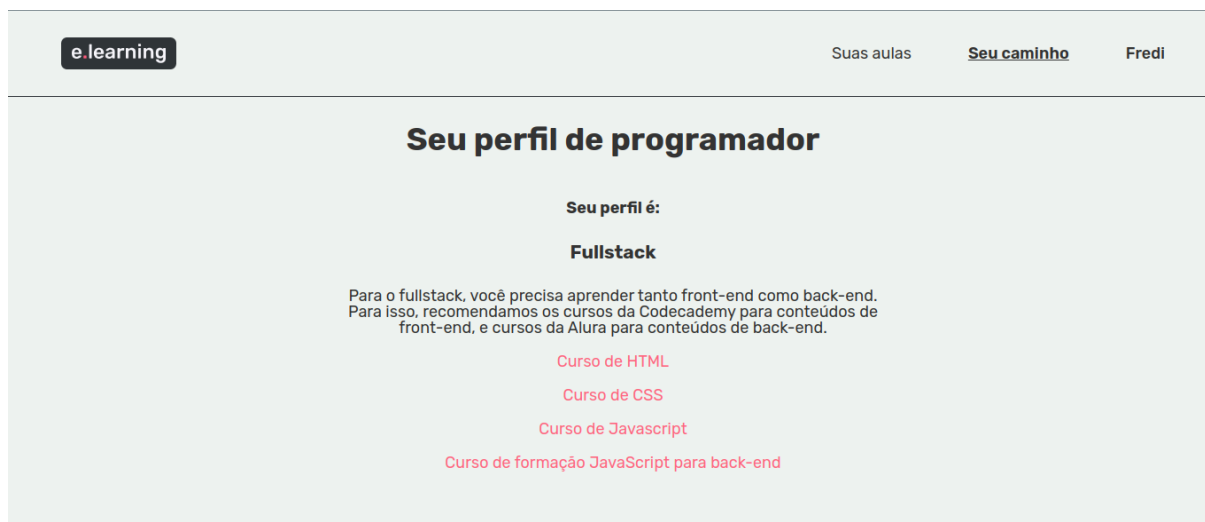


Fonte: elaborado pelo autor (2022)

Porém, existe a possibilidade de a avaliação do aluno ser igual para conteúdo de *front-end* e *back-end*. Para isso, a solução foi apresentar ao usuário a recomendação de *Fullstack*, que é a área que engloba as duas anteriores e quando o aluno se aprofundar em cada área, ele poderá perceber qual gosta mais. Neste

caso, é recomendado ao aluno tanto os cursos da CodeCademy que compõe a recomendação do *front-end*, quanto o curso da Alura, que é a recomendação para alunos do *back-end*.

Figura 15 – Tela de recomendação Fullstack



Fonte: elaborado pelo autor (2022)

5 RESULTADOS

Para validar o projeto, foram planejados dois questionários, além de um teste de usabilidade na aplicação. O primeiro questionário tem como objetivo a identificação do público-alvo na utilização da aplicação, além da validação da ideia. Após isto, é enviado ao usuário o link para a aplicação¹², para que seja possível testar as funcionalidades básicas e por fim, é enviado um questionário para a validar a usabilidade da aplicação.

Para o teste foram convidadas onze pessoas, sendo elas profissionais da área de tecnologia, de diferentes níveis e estágios na carreira, sendo realizado em outubro de 2022. Esta seção tem como objetivo apresentar com detalhes este processo de validação e quais seus resultados em cada passo.

¹² Link para a aplicação: <https://e-learning-frontend.vercel.app/>

5.1 Metodologia Utilizada

Para a avaliação do protótipo¹³, foram utilizados os seguintes passos:

1. Questionário para identificação do público-alvo;
2. Teste de usabilidade da aplicação;
3. Pesquisa quantitativa para validar a satisfação dos usuários após utilizarem a aplicação.

5.1.1 Questionário para Identificação

Para uma melhor análise dos resultados, foi aplicado um questionário para os participantes que teve como objetivo a identificação do público-alvo e a aceitação da ideia entre eles. O questionário com as perguntas pode ser visualizado no Apêndice A.

A primeira questão teve como objetivo identificar a formação dos participantes dentro da área de tecnologia. Como podemos visualizar no Gráfico 1, as respostas foram bem divididas entre os diversos cursos de tecnologia, porém podemos destacar que quatro dos participantes não tem formação na área.

Gráfico 1 – Questão 1 do Questionário 1

Qual a sua formação?
11 respostas



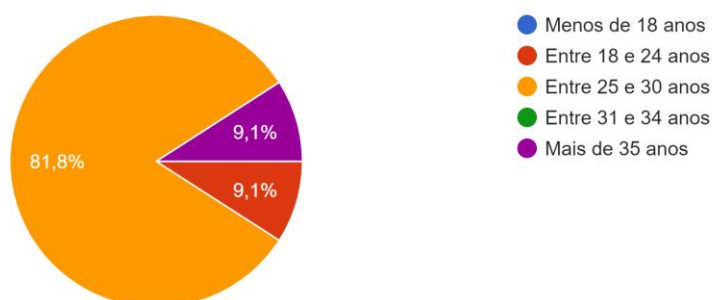
Fonte: elaborado pelo autor (2022)

¹³ Link do *front-end* e *back-end* no github: <https://github.com/GiovaniMenegat/e-learning-frontend> e <https://github.com/GiovaniMenegat/e-learning-backend>

A segunda questão teve o objetivo de identificar a faixa etária dos participantes. Podemos observar no Gráfico 2 que a grande maioria estão entre os 25 e 30 anos.

Gráfico 2 – Questão 2 do Questionário 1

Qual sua idade?
11 respostas

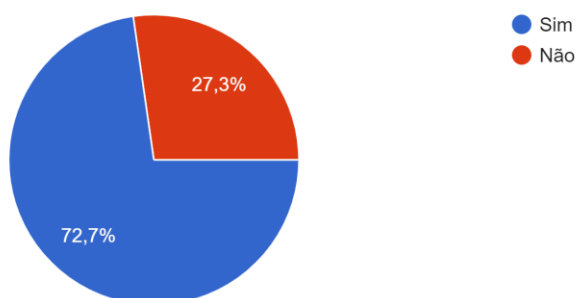


Fonte: elaborado pelo autor (2022)

A terceira questão tinha como objetivo entender se o público-alvo da pesquisa estava atuando na área de programação. Podemos observar que maioria dos participantes atuam na área de programação, havendo oito que disseram que estão atuando e apenas três que responderam que não estão atuando na área de tecnologia atualmente.

Gráfico 3 – Questão 3 do Questionário 1

Você está trabalhando na área de programação?
11 respostas



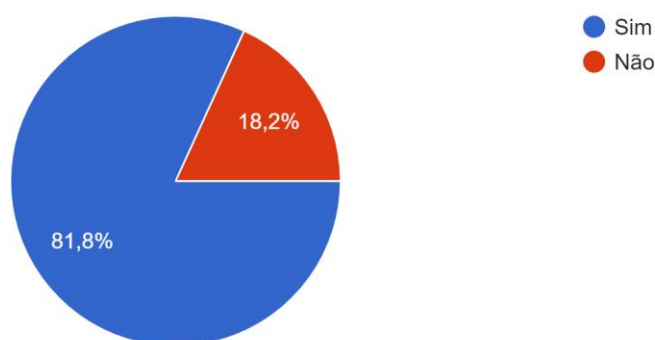
Fonte: elaborado pelo autor (2022)

Na quarta questão, o objetivo era entender se os participantes já haviam realizado algum curso online de programação, caso a resposta fosse “Sim”, era solicitado qual ou quais cursos ele havia feito. Podemos observar, com base na primeira resposta, que a grande maioria dos profissionais da área já realizaram um curso online. Isso prova a adesão dos profissionais neste tipo de curso, pela praticidade e qualidade dos cursos online, muitos optam por realizar eles.

Gráfico 4 – Questão 4 do Questionário 1

Você já realizou algum curso online de programação?

11 respostas



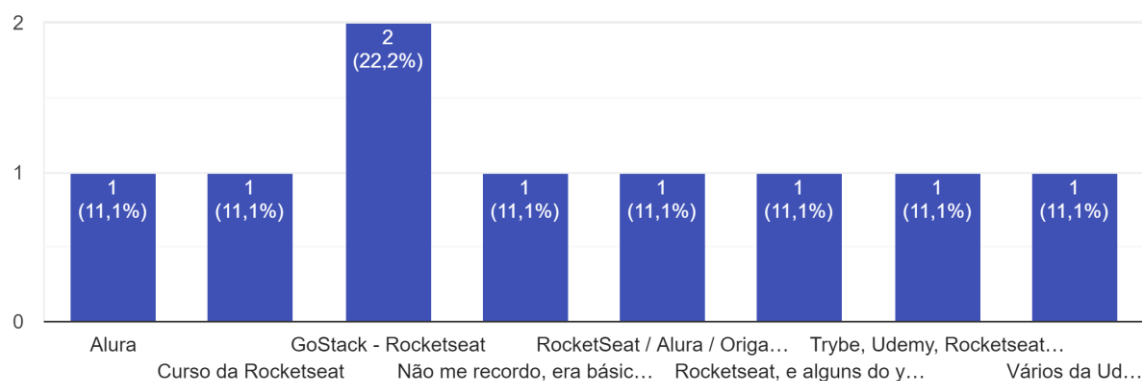
Fonte: elaborado pelo autor (2022)

Para os nove participantes que assinalaram “Sim”, foi perguntado quais cursos foram realizados. As respostas foram variadas, porém podemos destacar a Rocketseat e Udemy.

Gráfico 5 – Questão 5 do Questionário 1

Caso tenha assinalado "Sim", qual?

9 respostas



Fonte: elaborado pelo autor (2022)

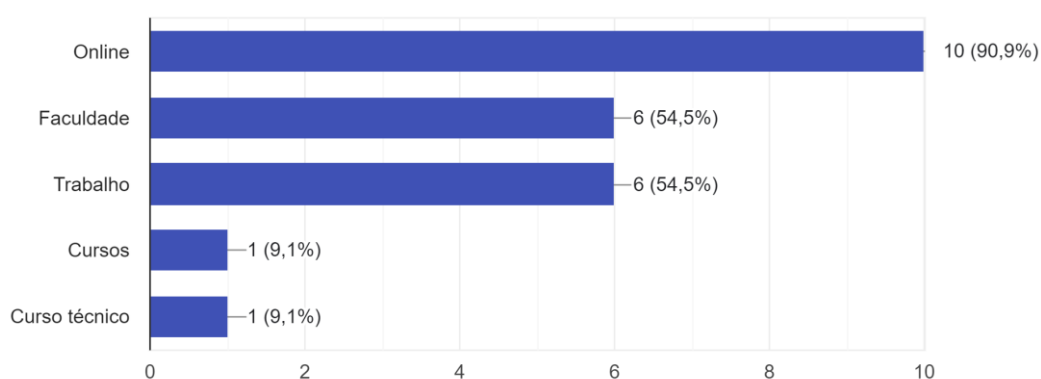
A questão seis foi uma questão de múltipla escolha, onde o participante poderia selecionar mais de uma opção. Nesta questão, foi perguntado como foi o seu aprendizado na programação. O objetivo desta pergunta era entender melhor cada caminho dentro da programação de cada participante.

Podemos observar que grande maioria respondeu que aprendeu a programar na internet e mais da metade respondeu que aprendeu na faculdade e no trabalho.

Gráfico 6 – Questão 6 do Questionário 1

Como foi seu aprendizado na programação?

11 respostas



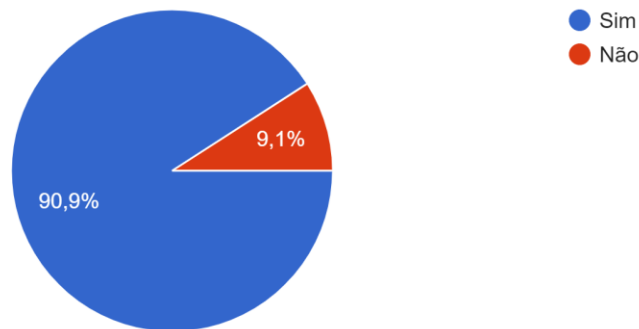
Fonte: elaborado pelo autor (2022)

A questão sete teve como objetivo entender se os participantes sentiram alguma dificuldade em entender conceitos de programação no início do aprendizado. Como podemos observar no Gráfico 7, grande maioria respondeu que sim, sentiu dificuldades em entender conceitos.

Gráfico 7 – Questão 7 do Questionário 1

Você teve dificuldades em entender conceitos de programação no início do aprendizado?

11 respostas



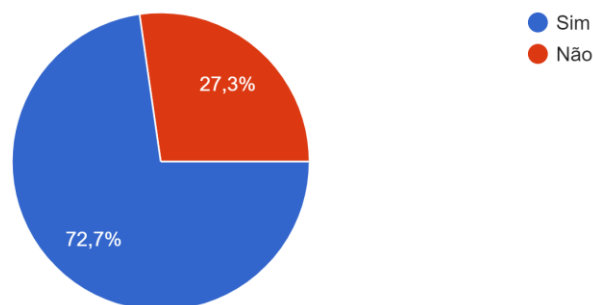
Fonte: elaborado pelo autor (2022)

Na oitava questão, o objetivo era entender a viabilidade do projeto. Para isso, foi questionado se o participante realizaria um curso introdutório na área de programação, onde no final ele iria receber uma recomendação de qual área seguir. Nesta questão, a maioria dos participantes respondeu que sim, realizaria este curso.

Gráfico 8 – Questão 8 do Questionário 1

Você realizaria um curso introdutório na área de programação de software, onde no final iria receber uma recomendação de qual área seguir?

11 respostas



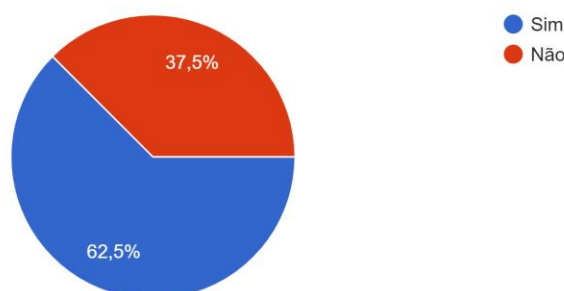
Fonte: elaborado pelo autor (2022)

Quem respondeu que “Sim” foi levado para a nona e última questão. Esta questão tinha como objetivo entender a viabilidade da cobrança na utilização do projeto. Esta questão é importante pois, caso o projeto entre no mercado futuramente, como uma opção de curso, é importante entender se os usuários pagariam para utilizar ele, ou não.

Nesta pergunta, os oito participantes que responderam “Sim” na Questão 8 tiveram respostas distintas, com uma vantagem para “Sim”. Nisso podemos observar que a cobrança é viável, porém é necessário tomar cuidado, pois nem todos os usuários podem aceitar isso.

Gráfico 9 – Questão 9 do Questionário 1

Você pagaria para realizar este curso introdutório?
8 respostas



Fonte: elaborado pelo autor (2022)

5.1.2 Teste de Usabilidade

Após o questionário para identificação do público-alvo, a aplicação foi enviada para todos os onze participantes. Esta etapa foi necessária para entender os possíveis pontos de melhoria na aplicação e para a realização do próximo questionário.

Nesta etapa, os participantes puderam acessar a plataforma e testar todos os requisitos funcionais da aplicação e através desses testes, foram identificados alguns pontos de melhorias. Foi relatado por um usuário uma dificuldade em entender como a plataforma funcionava e com uma melhoria no UX isso poderia ser resolvido. Os demais usuários não relataram dificuldades na aplicação, mas sugeriram diversas melhorias, como um botão de Avançar na tela da aula, para ser

possível acessar o próximo conteúdo da trilha sem a necessidade de voltar para a página inicial, também foi relatada a falta de um botão para a troca da avaliação, pois atualmente não é possível alterar uma avaliação já feita e o usuário pode acabar clicando sem querer e a experiência se tornará negativa.

Apesar destas melhorias citadas, a maioria dos participantes elogiou a plataforma, relatando que gostariam de ter utilizado ela no início de suas carreiras.

5.1.3 Pesquisa Quantitativa

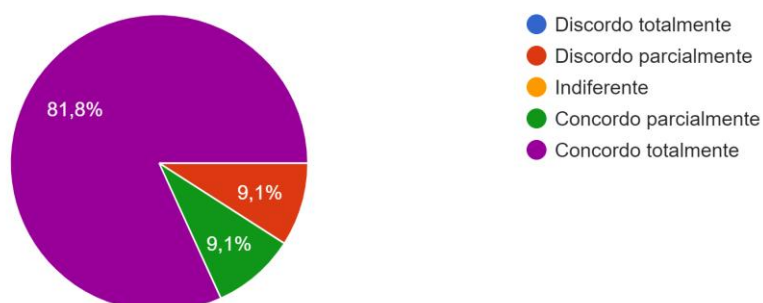
Essa pesquisa foi elaborada com a finalidade de entender a experiência dos usuários no uso da plataforma. Para isso, foram criadas quatro perguntas objetivas sobre a experiência de cada um durante o uso da plataforma. O questionário aplicado pode ser visualizado no Apêndice B.

A primeira questão teve como objetivo verificar se a aplicação é fácil de ser utilizada. Como podemos visualizar no Gráfico 10, nove usuários responderam que concordam totalmente com essa afirmativa, um respondeu que concorda parcialmente e um respondeu que discorda parcialmente.

Gráfico 10 – Questão 1 do Questionário 2

A aplicação é fácil de ser utilizada?

11 respostas



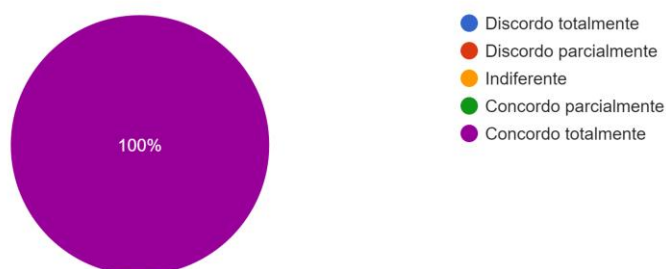
Fonte: elaborado pelo autor (2022)

Na segunda questão, foi perguntado se a aplicação atende aos requisitos funcionais a que se propõe. Para isto, foram listados todos os requisitos funcionais descritos na Subseção 4.1.3.1 no cabeçalho da questão, para que os participantes pudessem avaliar corretamente o atendimento de cada requisito.

Como podemos visualizar no Gráfico 11, todos os usuários responderam que concordam totalmente, o que demonstra que a aplicação atende a todos os requisitos funcionais propostos.

Gráfico 11 – Questão 2 do Questionário 2

A aplicação atende aos requisitos a que se propõe?
11 respostas

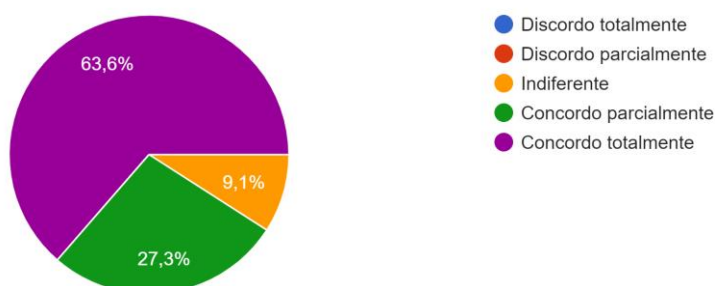


Fonte: elaborado pelo autor (2022)

Na terceira questão, foi perguntado se a aplicação introduz adequadamente o aluno na programação de software. Como podemos visualizar no Gráfico 12, sete participantes responderam que concordam totalmente, três responderam que concordam parcialmente e um respondeu indiferente.

Gráfico 12 – Questão 3 do Questionário 2

A aplicação introduz adequadamente o aluno na programação de software?
11 respostas



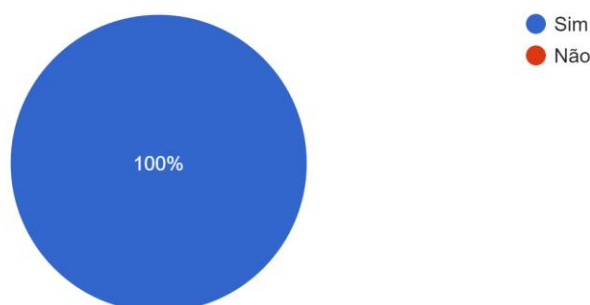
Fonte: elaborado pelo autor (2022)

Na quarta e última questão, foi perguntado se os participantes utilizariam a aplicação e como podemos visualizar no Gráfico 13, todos responderam que sim, utilizariam.

Gráfico 13 – Questão 4 do Questionário 2

Você utilizaria a aplicação?

11 respostas



Fonte: elaborado pelo autor (2022)

6 CONSIDERAÇÕES FINAIS

Este trabalho demonstrou a concepção e implementação de uma plataforma que tem como objetivo auxiliar novos alunos na área da programação, a partir de conteúdos básicos, que todos devem conhecer antes de aprender alguma linguagem de programação.

O tema foi escolhido por interesse pessoal, após perceber que cada vez mais pessoas se interessavam em entrar na área de programação, porém ficavam muito tempo perdidos no início, entre conteúdos técnicos que não continham uma introdução, o que levava muitos a desistirem do aprendizado.

Para o desenvolvimento do software, foi necessário um estudo em diversos cursos de programação presentes no mercado e um levantamento nos conteúdos introdutórios necessários para iniciar na programação de software. Este estudo sobre outros cursos na área mostrou que o mercado está saturado de cursos de programação, que englobam todas as áreas e abrangem todas as tecnologias utilizadas atualmente, porém nenhum deles apresenta ao aluno uma visão geral da área, mostrando conteúdos básicos e como a programação de software funciona como um todo, mas sim, ensinando tecnologias imediatamente, sem a explicação de conceitos, ou passando brevemente sobre eles.

Foram realizados testes para verificar a aceitação da plataforma, através de dois questionários e um teste de usabilidade. Os testes foram relatados e analisados

e, diante destes resultados, foi possível perceber a aceitação dos participantes na plataforma, onde, apesar de serem necessárias melhorias, é uma ideia que pode trazer grandes ganhos para a área no futuro.

O processo de desenvolvimento do presente trabalho foi uma ótima oportunidade de colocar em prática o que foi estudado durante o curso de Sistemas de informação, pois nele foi possível desenvolver um software desde a fase de levantamento de requisitos, até a fase de implantação, descrevendo todo o processo do software neste trabalho. Por isso, entende-se que os objetivos definidos no início do trabalho foram atingidos, independente das melhorias futuras.

5.1 Trabalhos Futuros

Para melhorias futuras, podemos apontar uma melhoria de UX/UI nas telas, promovendo um melhor entendimento de como funciona a plataforma ao usuário final, uma área onde o usuário pode alterar informações do seu perfil ou até remover sua conta, um botão de Avançar dentro da tela de aula, onde permite o aluno avançar o conteúdo sem a necessidade de voltar para a tela inicial, um status de aula vista porém não avaliada no *card* de aulas, filtros na listagem de aulas, que possibilite o usuário encontrar facilmente aulas não avaliadas, não assistidas ou já avaliadas e um aumento/melhora nos conteúdos da trilha, que entregue ao usuário o melhor aprendizado possível, trazendo assim um aumento nas recomendações possíveis ao usuário, abrangendo outras áreas dentro da programação, como Analista de Dados, *Product Owner* (PO), Designer de UX entre outros.

REFERÊNCIAS

ALBERTIN, Alberto Luiz; ALBERTIN, Rosa Maria de Moura. Tecnologia de Informação e Desempenho Empresarial no Gerenciamento de seus Projetos: um Estudo de Caso de uma Indústria. **Revista de Administração Contemporânea**. Curitiba, v. 12, n. 3, p. 599-629, set. 2008. Disponível em: <https://doi.org/10.1590/S1415-65522008000300002>. Acesso em: 19 de mai. de 2022.

ARIMOTO, Maurício; OLIVEIRA, Weldrey. Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27., 2019, Belém. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 244-254. ISSN 2595-6175. Disponível em: <https://doi.org/10.5753/wei.2019.6633>. Acesso em: 16 de mai. de 2022.

BrasscomA. Estudo da Brasscom aponta demanda de 797 mil profissionais de tecnologia até 2025. Brasscom, 2021. Disponível em: <https://brasscom.org.br/estudo-da-brasscom-aponta-demanda-de-797-mil-profissionais-de-tecnologia-ate-2025/>. Acesso em: 12 de abr. de 2022.

BrasscomB. Procura por profissionais de TI será de 420 mil pessoas até 2024 e o Brasil forma apenas 46 mil por ano. Brasscom, 2021. Disponível em: <https://brasscom.org.br/procura-por-profissionais-de-ti-sera-de-420-mil-pessoas-ate-2024-e-o-brasil-forma-apenas-46-mil-por-ano/>. Acesso em: 12 de abr. de 2022.

DE SOUSA, A. I. P. Analfabetismo digital na educação. **EaD & Tecnologias Digitais na Educação**, [S. l.], v. 4, n. 5, p. 52–57, 2017. Disponível em: <https://ojs.ufgd.edu.br/index.php/ead/article/view/5493>. Acesso em: 16 de mai. de 2022.

DE SOUZA, Ivan. **Saiba o que é Node.js, como ele funciona e como usá-lo no seu site.** Rock Content, 2020. Disponível em: <https://rockcontent.com/br/blog/node-js/>. Acesso em: 22 de abr. de 2022.

DOS SANTOS, R. P.; COSTA, H. A. X. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. **INFOCOMP Journal of Computer Science**, [S. l.], v. 5, n. 1, p. 41–50, 2006. Disponível em: <https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/121>. Acesso em: 16 de mai. de 2022.

Fabricio. **Vue.js Tutorial.** DEVMEDIA, 2017. Disponível em: <https://www.devmedia.com.br/vue-js-tutorial/38042>. Acesso em: 15 de abr. de 2022.

FERNANDES, Claudete Teixeira; SCALCO, Sirlésia Vigarani; DA SILVA, Rodrigo Borsatto Sommer. Métodos e Técnicas de Autoaprendizagem. **Maiêutica - Estudos Contemporâneos em Gestão Organizacional**. v. 1, n. 1, 2014. Disponível em:

http://publicacao.uniasselvi.com.br/index.php/GESTAO_EaD/article/view/1257. Acesso em: 23 de mai. de 2022.

FLANAGAN, David. **JavaScript, O guia definitivo**. 6. ed. Porto Alegre: Editora Bookman, 2013.

IDC. Previsões da IDC para 2022 apontam crescimento de 8,2% para o mercado de TIC no Brasil. **IDC**, 08 fev. 2022. Disponível em: <https://www.idc.com/getdoc.jsp?containerId=prLA49041022>. Acesso em: 19 de mai. de 2022.

Lenon. **Node.js – O que é, como funciona e quais as vantagens**. Opus Software, 2018. Disponível em: <https://www.opus-software.com.br/node-js/>. Acesso em: 22 de abr. de 2022.

MDN Web Docs. **JavaScript**. MDN Web Docs, 2021. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Glossary/JavaScript>. Acesso em: 15 de abr. de 2022.

MDN Web Docs. **Sobre JavaScript**. MDN Web Docs, 2021. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript. Acesso em: 15 de abr. de 2022.

PASTORE, José. Os empregos em TI. *In: Artigos*. **José Pastore**. São Paulo, 13 mai. 2008. Disponível em: http://www.josepastore.com.br/artigos/em/em_118.htm. Acesso em: 19 de mai. de 2022.

PITOL, M.; ZANATTA, A. L. **Análise do mercado de trabalho no setor de TI no RS**. 2009. Trabalho de Conclusão de Curso - Curso Ciências da Computação. Universidade de Passo Fundo, Passo Fundo, 2009. Disponível em: <http://www.periodicosibepes.org.br/index.php/reinfo/article/view/1147>. Acesso em: 19 de mai. de 2022.

REATEGUI, Eliseo Berni; CAZELLA, Sílvio César. Sistemas de recomendação. *In: XXV Congresso da Sociedade Brasileira de Computação*. sn, 2005. p. 306-348. Acesso em: 01 de out. de 2022.

RECH, Ionara. **Adoção de novas tecnologias de informação (TI): estudo sobre problemas e ações em grandes empresas da região metropolitana de Porto Alegre/RS**. 2001. Dissertação (Pós-graduação) – Curso de Administração. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001. Disponível em: <https://www.lume.ufrgs.br/bitstream/handle/10183/2330/000318054.pdf>. Acesso em: 19 de mai. de 2022.

Redação. O que é TI? Curso, profissão e mercado de trabalho. *In: Cursos De Graduação EAD*. **UCPel - Universidade Católica de Pelotas**. 9 set. 2021. Disponível em: <https://ead.ucpel.edu.br/blog/o-que-e-ti>. Acesso em: 19 de mai. de 2022.

Redação. Pesquisa: 49,7% das escolas públicas brasileiras ainda não possuem internet. **CanalTech**, 23 jul. 2014. Disponível em:

<https://canaltech.com.br/mercado/Pesquisa-497-das-escolas-publicas-brasileiras-ainda-nao-possuem-internet/>. Acesso em: 16 de mai. de 2022.

SANTOS, Bárbara. 48% das escolas não têm computador para uso do aluno.

Estadão, São Paulo, 21 jul. 2014. Disponível em:

<https://educacao.estadao.com.br/noticias/geral,48-das-escolas-nao-tem-computador-para-uso-do-aluno,1531623>. Acesso em: 16 de mai. de 2022.

VueJS. **Documentação do VueJS**. VueJS, 2022. Disponível em:

<https://br.vuejs.org/v2/guide/index.html>. Acesso em: 15 de abr. de 2022.

Vuex. **Documentação do Vuex**. Vuex, 2022. Disponível em:

<https://v3.vuex.vuejs.org/ptbr/>. Acesso em: 15 de abr. de 2022.

X-Apps. **Entenda o interpretador de códigos Node.js**. X-Apps, 2019. Disponível em: <https://www.x-apps.com.br/node-js>. Acesso em: 22 de abr. de 2022.

APÊNDICE A - QUESTIONÁRIO PARA IDENTIFICAÇÃO DO PÚBLICO-ALVO

1. Qual a sua formação?
 - Sistemas de Informação
 - Ciência da Computação
 - Engenharia de Software
 - Análise e Desenvolvimento de Sistemas
 - Não tenho formação na área de programação de software
 - Outros

2. Qual sua idade?
 - Menos de 18 anos
 - Entre 18 e 24 anos
 - Entre 25 e 30 anos
 - Entre 31 e 34 anos
 - Mais de 35 anos

3. Você está trabalhando na área de programação?
 - Sim
 - Não

4. Você já realizou algum curso online de programação?
 - Sim.
 - Não

5. Caso tenha assinalado "Sim", qual?

6. Como foi seu aprendizado na programação?
 - Online
 - Faculdade
 - Trabalho

7. Você teve dificuldades em entender conceitos de programação no início do aprendizado?

Sim

Não

8. Você realizaria um curso introdutório na área de programação de software, onde no final iria receber uma recomendação de qual área seguir?

Sim

Não

9. Você pagaria para realizar este curso introdutório?

Sim

Não

APÊNDICE B - PESQUISA QUANTITATIVA PARA VALIDAR A SATISFAÇÃO DOS USUÁRIOS APÓS UTILIZAREM A APLICAÇÃO

1. A aplicação é fácil de ser utilizada?

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

Requisitos da plataforma Code Path:

- 1: O sistema deverá permitir a criação de um usuário;
- 2: O sistema deverá permitir que o usuário entre na plataforma, através do e-mail e senha cadastrados;
- 3: O sistema deverá permitir ao usuário a visualização do conteúdo em cada aula da trilha;
- 4: O sistema deverá permitir a avaliação do conteúdo da aula pelo usuário;
- 5: O sistema deverá apresentar ao usuário a recomendação de qual caminho seguir na programação, após a avaliação de todas as aulas.

2. A aplicação atende aos requisitos a que se propõe?

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

3. A aplicação introduz adequadamente o aluno na programação de software?

- Discordo totalmente
- Discordo parcialmente
- Indiferente
- Concordo parcialmente
- Concordo totalmente

4. Você utilizaria a aplicação?

Discordo totalmente

Discordo parcialmente

Indiferente

Concordo parcialmente

Concordo totalmente