

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LEONARDO SILVATTI SILVA

**REDE DE APOIO E ENTRADA NO MERCADO DE TRABALHO
PARA ESTUDANTES DE TI**

BAURU

Fevereiro/2022

LEONARDO SILVATTI SILVA

**REDE DE APOIO E ENTRADA NO MERCADO DE TRABALHO
PARA ESTUDANTES DE TI**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista “Júlio de Mesquita Filho”,
Faculdade de Ciências, Campus Bauru.
Orientador: Prof^a. Dr^a. Márcia A. Zanoli Meira
e Silva

BAURU
Fevereiro/2022

Leonardo Silvatti Silva REDE DE APOIO E ENTRADA NO MERCADO DE
TRABALHO PARA ESTUDANTES DE TI/ Leonardo Silvatti Silva. – Bauru,
Fevereiro/2022- 47 p. : il. (algumas color.) ; 30 cm.
Orientador: Prof^a. Dr^a. Márcia A. Zanolli Meira e Silva
Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de
Mesquita Filho”
Faculdade de Ciências
Departamento de Computação, Fevereiro/2022.
1. Ensino 2. Desenvolvimento 3. Mercado de trabalho 4. COVID-19 5. Pandemia

Leonardo Silvatti Silva

REDE DE APOIO E ENTRADA NO MERCADO DE TRABALHO PARA ESTUDANTES DE TI

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof^a. Dr^a. Márcia A. Zanolli Meira e Silva

Orientadora
Universidade Estadual Paulista "Júlio de
Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

**Prof^a. Dr^a. Simone das Graças Domingues
Prado**

Universidade Estadual Paulista "Júlio de
Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Prof. Dr. Kleber Rocha de Oliveira

Universidade Estadual Paulista "Júlio de
Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Bauru, _____ de _____ de _____.

À todos que acreditaram em mim, mesmo quando eu não acreditava.

Agradecimentos

Primeiramente gostaria de agradecer a minha orientadora, que mais que uma professora se tornou uma amiga ao longo dos anos. Me motivou a chegar até aqui e além de tudo sempre esteve lá para ouvir meus problemas, sendo eles relacionados a graduação ou não.

Agradeço também a minha família por ter acreditado em mim em todos os momentos, vocês foram de imensa importância para que eu chegasse até aqui e sempre me apoiaram em todas as diversidades.

Agradeço aos meus amigos que também estavam lá nos meus piores e melhores momentos, nunca irei esquecer de vocês.

E por fim, quero me agradecer por acreditar em mim mesmo, quero me agradecer por todo esse trabalho duro. Quero me agradecer por não tirar folgas. Quero me agradecer por nunca desistir. Quero me agradecer por ser generoso e sempre dar mais do que recebo. Quero me agradecer por tentar sempre fazer mais o certo do que o errado. Quero me agradecer por ser eu mesmo o tempo inteiro.

"O que verdadeiramente somos é aquilo que o impossível cria em nós."

Clarisse Linspector em "A maçã no escuro"

Resumo

A pandemia de COVID-19 trouxe um novo cenário educacional. Com a portaria instaurada pelo Ministério da Educação em março de 2020, instituições de ensino de todo o país adotaram o modelo remoto. Além disso, o país entrou em isolamento social prolongado. Essa situação pode causar severos danos à saúde dos indivíduos. Sendo assim, este projeto propôs a criação de uma rede de apoio para estudantes da área de Tecnologia da Informação visando diminuir os danos decorrentes da quarentena fazendo com que haja um meio de interação assíncrono e direto entre estudantes e professores. Ademais, o projeto conta também com uma área livre onde os usuários podem discutir sobre tecnologias do mercado e assuntos pertinentes. Por fim, a página também reúne profissionais que já estão inseridos no mercado de trabalho para que possam publicar vagas de entrada ou estágio, facilitando o acesso de discentes que desejam adquirir novas experiências. O projeto foi desenvolvido usando ferramentas atuais e de grande aplicação no mundo real e utilizou padrões de construção escaláveis.

Palavras-chave: Ensino. Desenvolvimento. Mercado de Trabalho. COVID-19. Pandemia.

Abstract

The COVID-19 pandemic has brought a new educational scenario. With the ordinance introduced by the Ministry of Education in March 2020, educational institutions across the country adopted the remote model. In addition, the country went into prolonged social isolation. This situation can cause severe damage to the health of individuals. Therefore, this project proposed the creation of a support network for students in the area of Information Technology in order to reduce the damage resulting from the quarantine by providing a means of asynchronous and direct interaction between students and teachers. In addition, the project also has a free area where users can discuss market technologies and relevant issues. Finally, the page also brings together professionals who are already in the job market so that they can publish entry or internship vacancies, facilitating access for students who wish to acquire new experiences. The project was developed using current and widely applicable real-world tools and used scalable construction standards.

Keywords: Teaching. Development. Network. COVID-19. Pandemic.

Lista de figuras

Figura 1 – Lista de recursos disponíveis para o professor na ferramenta Moodle	19
Figura 2 – Visão dos tópicos criados no recurso de Fórum	19
Figura 3 – Tela dentro do tópico selecionado com resposta	20
Figura 4 – Interface do Google Classroom	20
Figura 5 – Área da pergunta feita pelo professor	21
Figura 6 – Seção de busca de empregos no LinkedIn	22
Figura 7 – Detalhes de uma vaga específica	22
Figura 8 – Respostas de uma postagem no <i>feed</i> do usuário	23
Figura 9 – Cor escolhida para o projeto nos botões do sistema	24
Figura 10 – Tipos de fonte categorizadas pelo estudo da Tipografia	26
Figura 11 – Estruturação do diretório de front-end	32
Figura 12 – Diagrama de Entidade Relacionamento do projeto	34
Figura 13 – JWT gerado no momento em que o usuário inicia a sessão	40
Figura 14 – Informações que geraram o token	41
Figura 15 – Visão do painel da ferramenta Amazon S3	43
Figura 16 – Configuração de uma nova instância de banco no RDS	43
Figura 17 – Informações da instância com link de acesso no RDS	44
Figura 18 – Arquivo de configuração da API	44
Figura 19 – Configuração de uma máquina virtual no EC2	45
Figura 20 – Painel EC2 com as máquinas virtuais criadas	45

Lista de códigos

1	Exemplo de entidade “resposta” e seus campos usando TypeORM	35
2	Exemplo da criação de Chaves Estrangeiras	36
3	Criação da classe EntidadeBase	37
4	Classe “Resposta” derivada de EntidadeBase	37
5	Interface do padrão de repositório	38
6	Manipulações básicas do CRUD	38
7	Controlador responsável por retornar os dados via requisição HTTP	39
8	Exemplo de código de um componente de botão.	41

Lista de abreviaturas e siglas

TI	Tecnologia da Informação
JS	Javascript
TS	Typescript
UI	User Interface
UX	User Experience
MUI	Material UI
CSS	Cascading Style Sheets
SASS	Super Awesome Style Sheets
JSON	JavaScript Object Notation
JWT	JSON Web Token
API	Application Programming Interface
REST	Representational State Transfer

Sumário

1	INTRODUÇÃO	14
2	PROBLEMA	15
3	OBJETIVOS	16
3.1	Objetivo Geral	16
3.2	Objetivos Específicos	16
3.3	Justificativa	16
4	FUNDAMENTAÇÃO TEÓRICA	18
4.1	Análise de ferramentas já existentes	18
4.1.1	Moodle	18
4.1.2	Google Classroom	20
4.1.3	LinkedIn	21
4.2	Interface e Experiência de Usuário	23
4.2.1	Cor	24
4.2.2	Tipografia	24
4.2.3	Forma	26
4.3	API Rest	27
5	METODOLOGIA	29
5.1	Ferramentas Utilizadas	29
5.1.1	Linguagem Javascript	29
5.1.2	Superconjunto Typescript	29
5.1.3	React	29
5.1.4	Syntactically Awesome Style Sheets	30
5.1.5	Material UI	30
5.1.6	NodeJs	30
5.1.7	TypeORM	30
5.1.8	PostgreSQL	30
5.1.9	MongoDB	30
5.1.10	Amazon Web Services	31
5.1.11	Amazon Relational Database Service	31
5.1.12	Elastic Computing	31
5.1.13	Simple Cloud Storage	31
5.2	Estrutura do projeto	31

5.2.1	Front-end	31
5.2.2	Back-end	33
6	RESULTADOS	35
6.1	Desenvolvimento do back-end	35
6.2	Desenvolvimento do <i>front-end</i>	41
6.3	Publicação	42
7	CONCLUSÃO	46
	REFERÊNCIAS	47

1 Introdução

Com o rápido avanço da pandemia de COVID-19 no ano de 2020, em 17 de março o Governo Brasileiro emitiu uma portaria que autorizava a substituição do modelo presencial em escolas de ensino fundamental, médio e superior pelo modelo remoto ([BRASIL, 2020](#)).

Por consequência o ensino sofreu com a falta de estrutura para o online e também pelo despreparo de docentes e discentes diante da nova tecnologia. O espaço da faculdade promovia o desenvolvimento das habilidades interpessoais e fomentava o conhecimento através de discussões e um ambiente focado no estudo.

Além do novo panorama educacional, o setor de TI em especial registrou um aumento de contratações no modelo *home-office* desde o início da pandemia causada pelo COVID-19. Em pesquisa realizada pela *GeekHunter*, uma organização focada em recrutamento de profissionais da área de tecnologia da informação, o número de vagas remotas saltou de 25% para 75% no ano de 2020 ([VALOR INTESTE, 2021](#)). Além disso, é possível ver que tanto os profissionais quanto as empresas tendem a adotar o modelo remoto. Segundo os dados do mesmo estudo, 78% dos entrevistados preferem o modelo remoto ou híbrido ao presencial. Esta tendência também está sendo seguida por aproximadamente 60% de empresas que, segundo a pesquisa irão adotar o modelo remoto ou híbrido mesmo após declarado o fim da pandemia ([OLHAR DIGITAL, 2021](#)).

Considerando ambos os aspectos e também um estudo feito usando uma ferramenta de fórum com estudantes de administração ([JACOBSON; FLEURY, 2005](#)), este projeto teve como objetivo analisar as ferramentas de educação utilizadas ao longo da pandemia, observando quais recursos de discussão elas possuem e também verificar como é a interação dos usuários em uma rede social específica para o mercado de trabalho. A partir deste estudo, foi desenvolvida uma nova ferramenta, com uso no ensino na área de Tecnologia da Informação, que visa minimizar a distância causada pelo isolamento.

2 Problema

Com a mudança do panorama de ensino publicada pelo Ministério da Educação, que propôs a mudança do sistema presencial para o sistema remoto, o isolamento social em 2020 foi intensificado. A curto prazo o isolamento eleva o estresse dos indivíduos, fazendo com que o nervosismo aumente e com ele uma carga de ansiedade muito grande recaia sobre as pessoas isoladas. O problema se intensifica quando é estendido ao longo prazo, onde pode ser observado um aumento de distúrbios mentais graves como depressão, pânico entre outros ([REVISTA ARCO, 2020](#)). As medidas paliativas para esta situação envolvem a tentativa de manter uma rotina minimamente organizada e parecida com o que havia antes, intensificar a comunicação com parentes, amigos dentre outros. Pelo fato de aumentarem a interação entre as pessoas, as redes sociais tomaram um papel muito importante durante o contexto da pandemia. Em uma pesquisa realizada, 73% dos entrevistados que usam as plataformas aumentaram seu uso durante o período de isolamento ([DIGIDEIAS, 2021](#)). No âmbito acadêmico, a interação que havia dentro do espaço físico da instituição de ensino foi substituída pelas aulas a distância, que não possuem o mesmo nível de proximidade. Assim sendo, há uma clara falta das discussões que havia em sala de aula, além das reuniões para estudo e trabalho que ocorriam dentro dos laboratórios e espaços comuns.

3 Objetivos

3.1 Objetivo Geral

Definido o problema, este projeto teve como objetivo principal criar um espaço para fomentar o desenvolvimento acadêmico dos alunos, fazendo com que usem o espaço para discussão de tecnologias, dúvidas das disciplinas da graduação e também de desenvolvimento geral. Possibilitar a aproximação dos professores, fazendo com que estejam em contato mais direto com o corpo discente. Além disso, prover aos profissionais de TI em começo de carreira um local de contato com empresas do setor que estejam oferecendo vagas de entrada e estágio nas áreas de desenvolvimento e suporte em tecnologia. Promover a aprimoração de todas as camadas envolvidas com as tendências de código e linguagens em alta no momento.

3.2 Objetivos Específicos

- Estudar como a interface de usuário, cores e tipografia influencia em como os usuários navegam pelos sites de educação e busca de emprego.
- Analisar a demanda do mercado em relação a novas tecnologias de ensino e ambientes online para interação acadêmica e profissional.
- Desenvolver uma nova solução que se adeque à nova realidade imposta pela pandemia do COVID-19, a qual teve um impacto muito grande no setor de tecnologia da informação, fazendo com que o modelo *home office* para trabalho e o ensino à distância ganhassem força.

3.3 Justificativa

Com o avanço rápido da pandemia em 2020, o cenário educacional mudou. Em março do mesmo ano, o governo brasileiro emitiu uma nota obrigando as instituições de ensino a adotarem o modelo remoto de ensino, sem tempo hábil para que as instituições se adaptassem a esse novo modelo. As ferramentas disponibilizadas apesar de darem o suporte necessário para o material educativo, não estimulam a discussão sobre as tecnologias e o ambiente de ajuda mútua que havia no espaço físico. A interação nestes espaços está limitada a disponibilização de material pelo professor para os alunos e a entrega de trabalhos e atividades.

O desenvolvimento de uma nova plataforma que permita a criação de tópicos onde alunos e professores possam interagir entre si traz de volta um pouco da experiência prévia do

ambiente presencial. Além disso, com a inclusão de representantes de empresas, os discentes podem impulsionar suas chances de ingressar no mercado de trabalho, algo que também era facilitado pelo espaço físico, que contava com murais de vagas, feiras de profissões e palestras com representantes do meio corporativo.

4 Fundamentação Teórica

Neste capítulo será apresentado o plano de fundo para o desenvolvimento da rede. Estes tópicos envolvem assuntos pertinentes à formulação teórica que embasou o desenvolvido deste projeto. Os conceitos destacados aqui são: i) a análise de ferramentas que já estão no mercado para entender o que há disponível hoje, tentando ver os pontos fortes de cada *site*; ii) uma breve síntese sobre a interface e experiência do usuário com APIs REST.

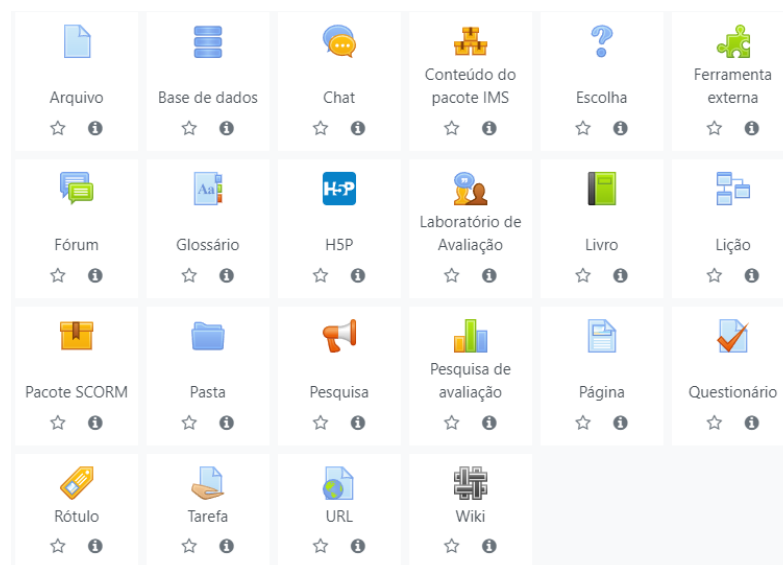
4.1 Análise de ferramentas já existentes

Para o desenvolvimento do projeto foram analisadas previamente três ferramentas juntamente com suas funcionalidades para aproximar o corpo discente do docente. Os sistemas foram escolhidos pela sua popularidade durante o período em que o ensino a distância foi adotado em massa pelas instituições e foram definidos nas seções a seguir. Além disso, também foi analisado o site *LinkedIn* por ser o meio principal de empresas se comunicarem e oferecerem vagas de trabalho para os estudantes à procura de estágio ou de primeiro emprego.

4.1.1 Moodle

No Brasil, 98% das universidades possuem esta ferramenta ([CARNEIRO et al., 2020](#)), além de ser amplamente usada no mundo todo. A plataforma conta com um grande suporte para que o professor publique suas aulas online. Os tópicos são personalizados pelo docente, sendo em seções configuradas de acordo com a demanda. Dentro delas, há a possibilidade de publicar uma atividade a qual pode estar em vários formatos como vídeo, arquivo .pdf ou texto plano no próprio editor embutido e muitas outras opções variadas, como visto na Figura 1.

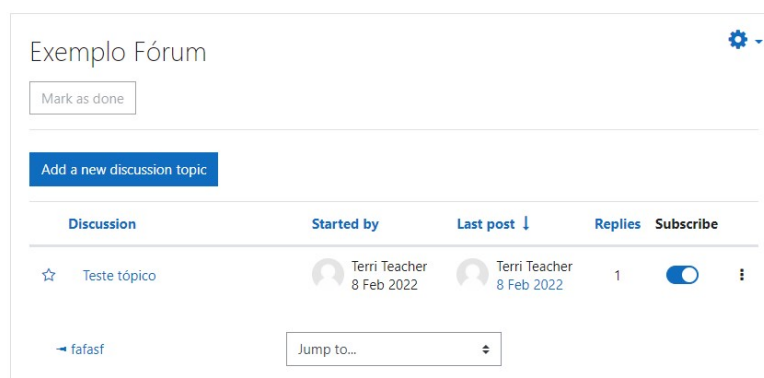
Figura 1 – Lista de recursos disponíveis para o professor na ferramenta Moodle



Fonte: Elaborado pelo autor.

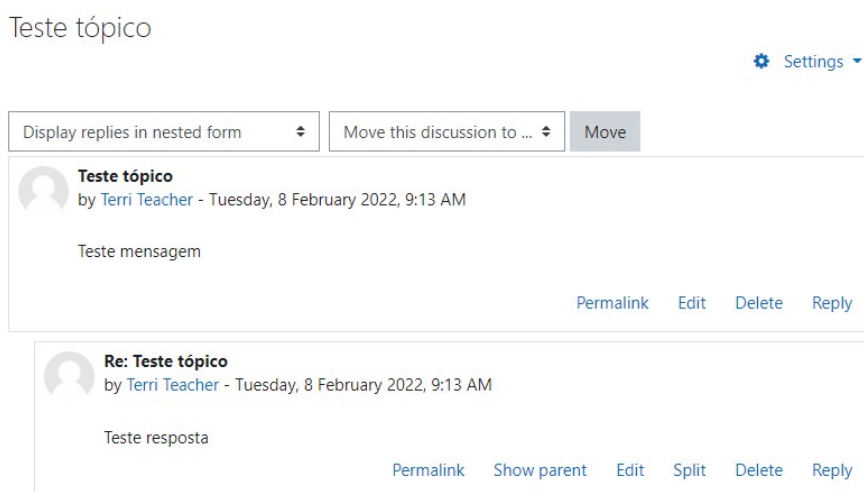
A interação entre os alunos só é permitida caso o professor crie uma atividade do tipo "Discussão". Como se trata somente de uma ferramenta de ensino, é vedada a participação de representantes de organizações corporativas. A visão dessas discussões é apresentada nas Figuras 2 e 3.

Figura 2 – Visão dos tópicos criados no recurso de Fórum



Fonte: Elaborado pelo autor.

Figura 3 – Tela dentro do tópico selecionado com resposta

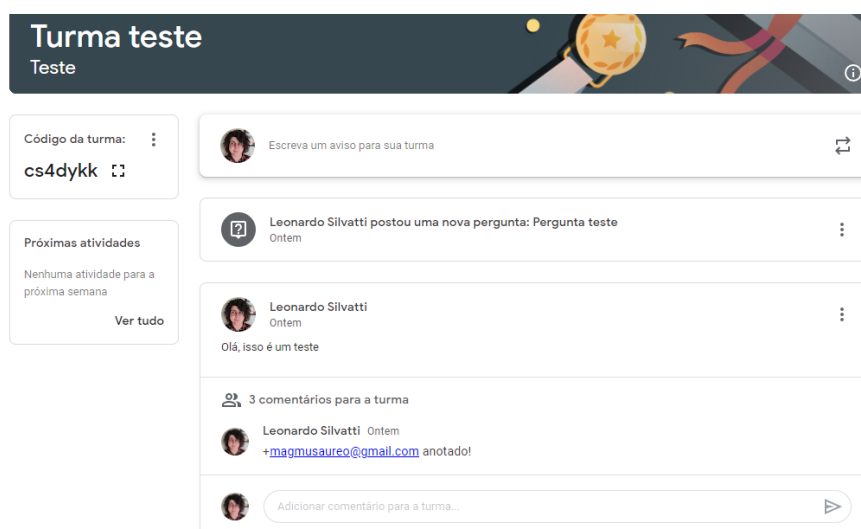


Fonte: Elaborado pelo autor.

4.1.2 Google Classroom

Esta foi a segunda ferramenta analisada que também se encaixa no âmbito educacional. O Google Classroom funciona de forma semelhante ao Moodle, o professor pode criar uma atividade compartilhada com a turma. As atividades ficam em um mural linear e são mostradas em ordem decrescente em relação a mais recente. As opções de tarefas são limitadas a perguntas, exercícios com espaço para envio de arquivos e avisos simples. Dentre as citadas, a que mais se aproxima do modelo de fórum é a "pergunta", onde o professor posta uma dúvida e a turma responde, podendo interagir entre si, como observado na Figura 4.

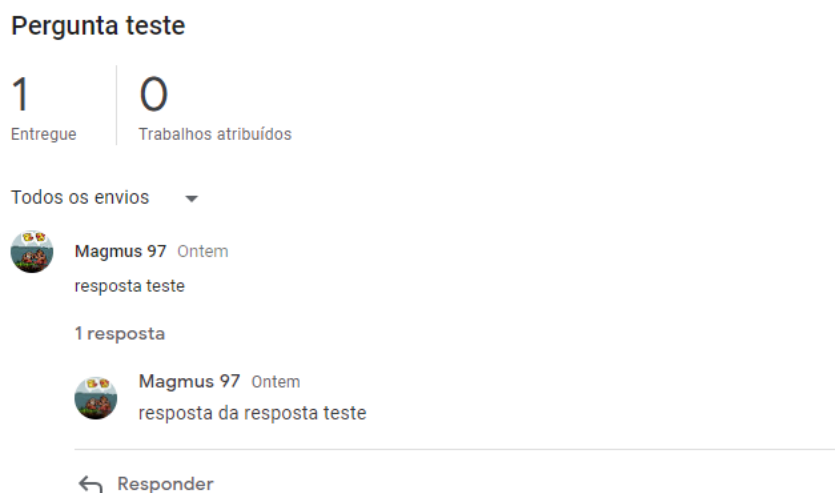
Figura 4 – Interface do Google Classroom



Fonte: Elaborado pelo autor.

Este tipo de atribuição pode ser avaliada pelo professor para compor a nota do aluno, que também é administrada pelo sistema. A interface geral pode ser vista na Figura 5.

Figura 5 – Área da pergunta feita pelo professor



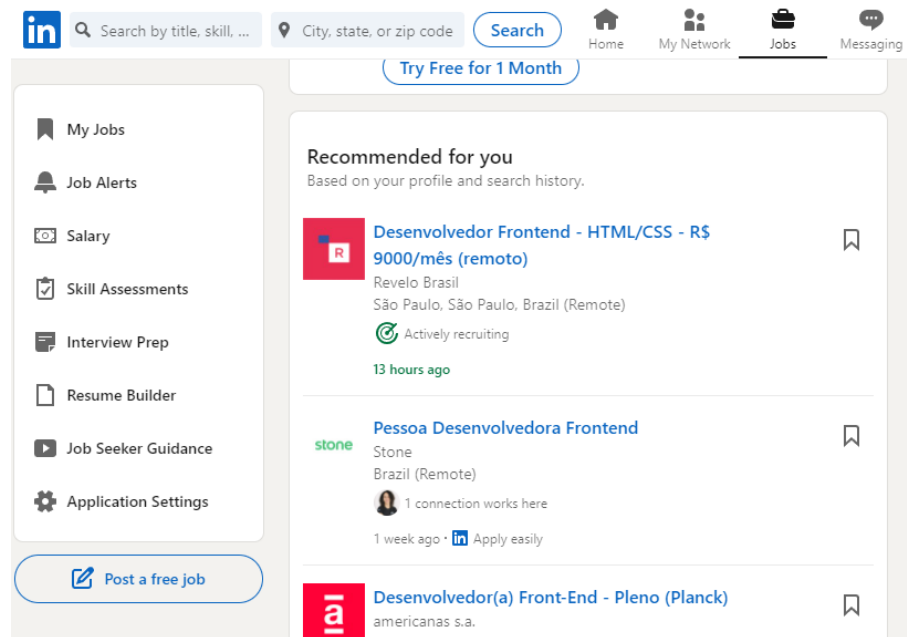
Fonte: Elaborado pelo autor.

O diferencial desta plataforma é ter à disposição outras ferramentas do Google que complementam o ensino a distância, como Meets para reuniões e o Drive para armazenar arquivos da turma.

4.1.3 LinkedIn

A terceira ferramenta observada foi a rede social para mercado de trabalho chamada LinkedIn. Diferentemente das plataformas vistas anteriormente, o LinkedIn tem seu foco no profissional empregado ou em busca de novas oportunidades. Nesta análise foram observadas as áreas de busca de emprego, onde o profissional pode buscar por empregos que se adequem ao seu perfil previamente criado e personalizado com suas experiências educacionais, laborais e gerais, como visto na Figura 6.

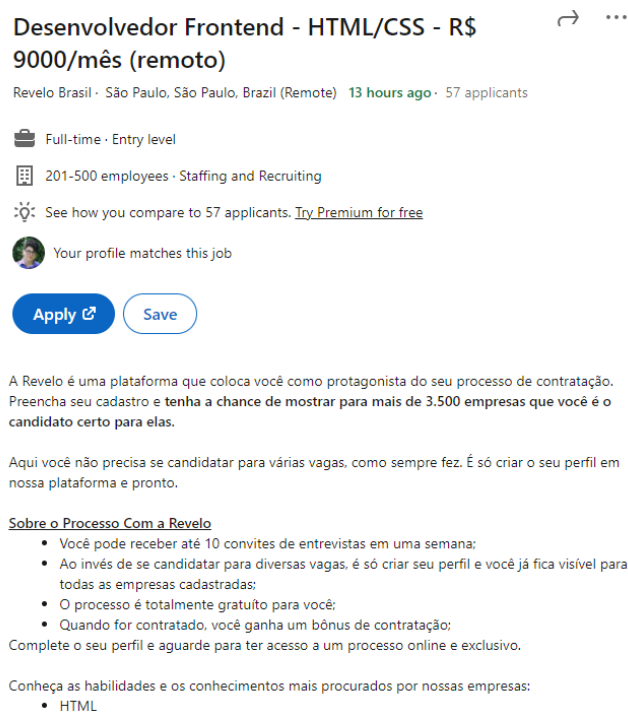
Figura 6 – Seção de busca de empregos no LinkedIn



Fonte: Elaborado pelo autor.

Ao clicar em uma vaga específica, uma área é mostrada com os detalhes do que é ofertado, junto com uma breve descrição e o botão para se inscrever no processo seletivo. A Figura 7 mostra o que o usuário vê ao realizar esta ação.

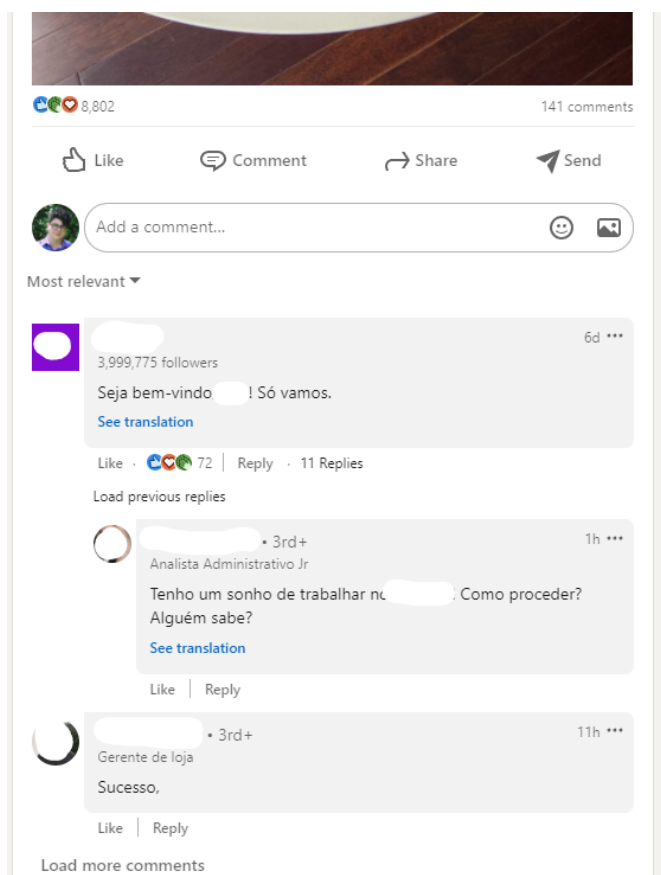
Figura 7 – Detalhes de uma vaga específica



Fonte: Elaborado pelo autor.

Além disso, a rede conta com uma página inicial no estilo *feed*, que foi popularizado pelo *Facebook*. Essas publicações geram discussões entre os membros, que podem comentar diretamente com o criador da postagem e podem responder outros indivíduos que comentaram previamente. A interface deste procedimento é ilustrada na Figura 8.

Figura 8 – Respostas de uma postagem no *feed* do usuário



Fonte: Elaborado pelo autor.

4.2 Interface e Experiência de Usuário

A sigla UI - User Interface ou Interface do Usuário em português - representa tudo que é utilizado na interação com um produto, sendo a intermediária visual (ou não visual) entre o homem e a máquina (celular, computador, TV, dentre outros). Por outro lado, tem-se a Experiência de Usuário, ou simplesmente UX (User eXperience). Embora tenha um escopo mais amplo e não esteja ligado somente ao meio virtual, a UX ganhou grande importância por estar intimamente ligada a UI. O estudo da interface e experiência em relação ao design de páginas de internet pode ser dividido em três áreas maiores, a saber, a cor, a tipografia e a forma. O objetivo de uma boa interface é sempre ser o mais clara possível, comunicando de forma sucinta e precisa tudo o que o usuário precisa saber, além de ser agradável aos olhos, provendo uma boa experiência de utilização.

4.2.1 Cor

A cor influencia não só como as pessoas se sentem, mas o que elas fazem. O estudo dessa área começou com Johann Wolfgang von Goethe, poeta nascido em 1749 na Alemanha, que publicou em 1810 o livro "Teoria das cores". O estudo foi aprimorado posteriormente pela socióloga também alemã, Eva Heller com sua publicação "A psicologia das cores: Como as cores afetam a emoção e a razão". Heller fez diversas pesquisas com grupos variados de indivíduos descobrindo padrões mentais nas pessoas quando se trata de cores. A psicologia da cor pode ajudar a fortalecer uma marca, incentivar as vendas, e até mesmo orientar os visitantes em direção às páginas ou ações específicas em um site. Cores como vermelho podem influenciar sentimentos como raiva, fúria e até mesmo paixão. Por outro lado, o azul desperta um senso de confiança e segurança.

Dito isso, para o projeto seria necessário passar uma cor que instigasse o usuário a ter uma sensação de confiança e também remetesse a um ambiente de conhecimento e compartilhamento de informações, além de não o saturar ao longo do tempo de uso. Por fim, vale lembrar que cores carregam sua devida importância e podem influenciar no comportamento do usuário, porém é importante ressaltar que não se pode basear a criação da interface inteiramente nelas. É importante pensar na acessibilidade ao desenvolver uma interface, pois a interpretação da cor não é absoluta e existem usuários que não verão as cores da mesma maneira que outros, sendo assim é de grande magnitude combinar a cor com a forma para criar uma interação acessível. A cor escolhida foi o código hexadecimal #5934AD, na análise feita por Goethe e Heller, a mistura de azul e roxo passa uma sensação de confiança e criatividade ao visualizador, sem enjoá-lo. A Figura 9 mostra a cor escolhida para o projeto aplicada a um botão nos moldes do Material UI, também utilizado.

Figura 9 – Cor escolhida para o projeto nos botões do sistema



Fonte: Elaborado pelo autor.

4.2.2 Tipografia

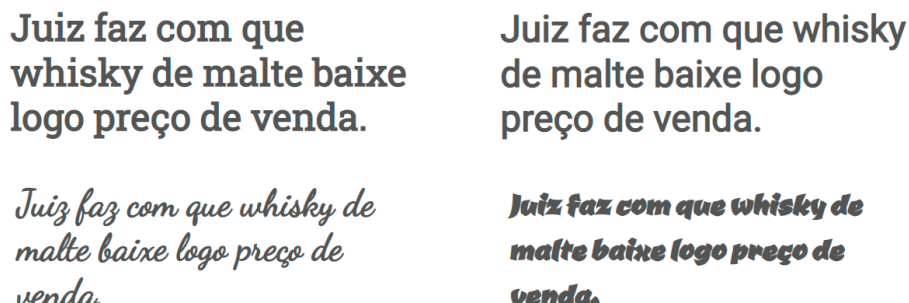
A tipografia é conhecida por ser uma ferramenta essencial do design gráfico, tendo extrema importância na construção de uma marca ou mesmo uma página da internet. Atualmente, o termo Tipografia é usado para representar o estudo, criação e aplicação de caracteres, estilos, formatos e disposição visual de palavras. Um bom estudo de tipografia é capaz de representar graficamente o valor e posicionamento do que será mostrado, além de despertar sentimentos e percepções do público. No âmbito do design de interfaces computacionais existem quatro tipos

de fontes (IFMG, 2021), as quais possuem características únicas, sendo usadas para propósitos distintos.

- **Serif:** São os tipos de fonte mais antigos. Os primeiros exemplares apareceram já no final do século XV. A palavra “serifa” refere-se aos pequenos pés presentes nas partes superior e inferior de cada letra. Esses pequenos floreios originaram-se dos pincéis dos artistas e eram adicionados às letras como elementos decorativos. As fontes serifadas podem ser divididas em várias subcategorias (estilo antigo, clássico, neoclássico, transicional, Clarendon e outros). Hoje as fontes serifadas estão entre os tipos de letras mais populares com estilos como o “Times New Roman” sempre presentes em livros, documentos e até mesmo alguns logotipos. Esse estilo de fonte é caracterizado por um design mais conservador.
- **Sans-serif:** As fontes não serifadas dispensam os floreios de sua predecessora em troca de uma abordagem mais limpa e moderna. Esse contraste faz com que combinem bem com as fontes serifadas. Elas datam do século XIX e tornaram-se imensamente populares nas décadas de 1920 e 30. Durante a metade do século XX, designers alemães expandiram ainda mais o corpo tipográfico com a criação do popular design Helvetica. Essas fontes são definidas pelas suas linhas retas e simples. Elas não apresentam floreios e enfatizam a legibilidade e a simplicidade em prol de um visual mais dimensionável.
- **Handwriting:** Os estilos caligráficos casuais ganharam popularidade no século XX e suavizam consideravelmente os floreios. Esses tipos de fontes também deixam de lado o visual de impressão compacto e favorecem o estilo cursivo, mais natural. As fontes caligráficas são divididas em duas principais subcategorias – formais ou casuais – e são desenhadas para se assemelhar à caligrafia manuscrita. As caligráficas formais são definidas por seus floreios e caracóis, que são chamados de caracteres “caudais”. Geralmente, recomenda-se usar essas fontes com moderação, pois podem afetar a legibilidade e tornar os sinais em palavras ou letras difíceis de entender e dimensionar. As fontes caligráficas casuais são mais discretas que as formais e enfatizam a legibilidade.
- **Display:** As fontes display ou gráficas renunciam às convenções em favor de um corpo tipográfico único e atraente. A maioria dos tipos decorativos é útil para setores e necessidades específicas, pois essas fontes são criadas sob medida para empresas específicas e são raramente usadas para longas sequências de texto. Esses tipos às vezes podem sair de moda se o design for muito específico ou temático.

A Figura 10 mostra os quatro tipos de fonte, da esquerda para a direita e de cima para baixo temos, os tipos *serif*, *sans-serif*, *handwriting* e *display*

Figura 10 – Tipos de fonte categorizadas pelo estudo da Tipografia



Fonte: Elaborado pelo autor.

Como este projeto é uma rede virtual, a escolha foi o uso da fonte não serifada *Montserrat* juntamente com a fonte *Open Sans*.

4.2.3 Forma

A forma complementa o entendimento da interface juntamente com a cor. Ela é importante porque ultrapassa a barreira de usuários com necessidades especiais que podem não enxergar a cor do mesmo jeito que outros. As principais associações de formas são feitas pelos princípios de Gestalt, essa teoria propõe que para entender as partes, é preciso compreender o todo. Em uma UI, a forma como os elementos estão dispostos ao longo da página trará interpretações distintas ao usuário. O estudo de Gestalt conta com 7 princípios básicos (AELA, 2020), sendo eles:

- **Figura-Fundo:** Este princípio afirma que nossa percepção instintivamente percebe objetos como estando ou à frente ou ao fundo. Os seres humanos não são capazes de focar em duas distâncias de profundidade simultaneamente, portanto precisamos escolher apenas uma. Em interfaces este princípio é amplamente aplicado em navegações, modais e caixas de diálogo quando uma ação que necessita de maior foco é trazida à frente.
- **Proximidade:** O princípio da Proximidade afirma que coisas que estão próximas parecem ser mais relacionadas entre si do que se estiverem distantes, mesmo que elas tenham a mesma forma. Esse princípio é tão forte que mesmo que os elementos sejam de cores e formas diferentes, caso estejam agrupados serão tomados como relacionados.
- **Similaridade:** A Similaridade explora o fato de que coisas que são parecidas são percebidas como parte do mesmo grupo e tendem a indicar a mesma função. O cérebro associa formas parecidas a funções parecidas e o mesmo acontece com a cor. A prioridade é dada sempre para a forma.

- **Continuidade:** Este princípio diz que formas em uma linha ou curva são tomados como um grupo mais facilmente do que elementos que não estão dispostos desta forma. Isso acontece porque o olho segue mais naturalmente uma reta ou sinuosidade. Esse princípio supera a similaridade por cores.
- **Fechamento:** O Fechamento diz que formas simples organizadas de maneira conveniente em um espaço sejam percebidas como um único elemento pelo cérebro. Por isso é possível afirmar que, por mais que haja espaço em branco, ele pode servir como um complemento para o entendimento do todo.
- **Região Comum:** Consiste em um grupo de elementos contornados por uma região fechada, seja ela uma borda, um fundo ou um elemento comum que englobe estes subelementos. Esta organização supera as cores, formas e continuidades, pois a percepção de uma região comum é muito mais fácil ao cérebro do que as citadas anteriormente.
- **Ponto Focal:** Por último, o princípio do Ponto Focal afirma que qualquer elemento que se destacar visualmente vai capturar e prender a atenção de quem está visualizando a interface. Dessa forma, este princípio é utilizado quando há uma hierarquia visual ou deseja-se enfatizar as ações principais que um usuário precisa tomar.

4.3 API Rest

A API Rest é um modelo de interface de programação de aplicações que segue as restrições do modelo REST. Essas restrições são traduzidas do inglês *Representational State Transfer* como Transferência de Estado Representacional. Para definir o que é uma API REST é necessário entender o conceito de API, o qual é um conjunto de definições e protocolos para integrar aplicações. Este conjunto funciona basicamente em duas partes, sendo uma chamada e uma resposta. Neste projeto, o exemplo claro de como a API funciona é quando o usuário entra em uma postagem, requisita ao servidor as informações da publicação, como o texto escrito e as respostas por uma chamada, enviando seu token de acesso para que o servidor valide a informação e responda com os dados que foram pedidos pelo visualizador. Assim sendo, uma API funciona como um intermediador de informações, fazendo a ligação entre a base de dados e a interface.

Ao aplicar o REST, um terceiro passo é adicionado. A requisição é feita via HTTP e no *back-end* os dados são transferidos ao que chamamos de *endpoint*, um ponto de acesso exposto publicamente que recebe e retorna requisições. O estado requisitado é transferido pela requisição usando um formato de dado. Por convenção, o padrão mais utilizado é o JSON, que pode ser lido por humanos e máquinas facilmente. Como os pontos de acesso são públicos, as requisições normalmente precisam de fatores de autenticação como parâmetros e cabeçalhos.

Quando o *endpoint* retorna o dado, ele também entrega um código de status que indica se a transação foi bem sucedida ou não.

Por fim, um sistema que utiliza os princípios do REST de maneira aplicada é chamado de RESTful.

5 Metodologia

Para a realização do projeto foram consideradas duas etapas: a) análise das características mais adequadas para o engajamento dos usuários, através de pesquisas online, que resultou na fundamentação teórica apresentada no Capítulo 4; b) fase de desenvolvimento que se divide em infraestrutura e interface de usuário. Neste capítulo esta segunda etapa será descrita com mais detalhes.

5.1 Ferramentas Utilizadas

Nesta seção serão abordadas as ferramentas utilizadas para o desenvolvimento da rede, tanto na parte de interface como na estrutura interna.

5.1.1 Linguagem Javascript

Criada em 1995, é uma linguagem de programação interpretada com orientação a protótipos. É uma linguagem de alto nível multiprogramada e com tipagem fraca. É a base para o Typescript.

5.1.2 Superconjunto Typescript

O desenvolvimento do Typescript (TS) começou em 2010 pela Microsoft, com Anders Hejsberg (que também participou do desenvolvimento do C#, Delphi e Turbo Pascal). A versão inicial foi lançada em outubro de 2012. A ideia por trás do superconjunto era dar ao Javascript uma robustez maior que ajudasse em projetos de larga escala, corrigindo as maiores fraquezas desse tipo de trabalho trazendo uma abordagem de orientação a classes. Como ela se baseia no JS (Javascript), seu aprendizado é de rápida velocidade quando há uma boa base em JS. O código em TS é transpilado para JS e em seguida interpretado pelo interpretador do ambiente, este pode ser um *browser* ou um interpretador dedicado no sistema. Além da tipagem estática, mecanismos de abstração como o operador `typeof`, que atribui o tipo de uma variável a outra, *namespaces* e *decorators*. Atualmente, a versão estável é a 4.5 e foi utilizada neste projeto.

5.1.3 React

React é uma biblioteca de desenvolvimento de interfaces baseada em JS com suporte para TS. Como seu desenvolvimento é aberto, hoje é mantido por diversas empresas como Facebook e Instagram e amplamente utilizado em diversos sites com aplicações variadas. A biblioteca utiliza uma abordagem por componentes que facilita a reutilização de código dentro

do projeto, isso faz com que a performance seja aprimorada e a complexidade de um grande projeto diminua. A versão utilizada foi a 17.0.2 e atualmente é a última versão estável.

5.1.4 Syntactically Awesome Style Sheets

Essa extensão, também chamada de SASS implementa uma linguagem de *script*, a estilização por folhas de estilo. Essa linguagem é interpretada em CSS (*Cascading Style Sheets*) e então lida pelo navegador. A ferramenta consiste em adicionar novos recursos a estilização para que esta se torne mais prática e parecida com uma linguagem de programação. A última versão estável é a 3.7.4 e foi usada no projeto.

5.1.5 Material UI

Também chamado de MUI, é uma biblioteca para React que provê componentes fundamentais para o desenvolvimento de aplicações responsivas e que desejam seguir as normas visuais da interface *material* do Google. Foi utilizada a versão 5.

5.1.6 NodeJs

Esta ferramenta é um ambiente de execução Javascript que permite criar aplicações para serem executadas como *standalones* em uma máquina sem a necessidade de um *browser*. Possui uma alta capacidade de escalabilidade e sua flexibilidade permite a implementação de micros serviços. Possui também integração com o Typescript. No projeto foi usada a versão 16.2.0, última versão estável.

5.1.7 TypeORM

TypeORM é um módulo de gerenciamento de relações de objeto para NodeJs utilizando TS. Esta ferramenta visa criar uma camada de mapeamento entre a camada da aplicação e a camada do banco de dados organizando o acesso. Neste projeto foi utilizada a versão 0.2.

5.1.8 PostgreSQL

O PostgreSQL (ou apenas Postgre) é um banco de dados relacional que estende a linguagem SQL e adiciona novas características que potencializam seu uso.

5.1.9 MongoDB

MongoDB é um programa de servidor orientado a documentos. É classificado como um programa de banco não relacional que utiliza documentos que se assemelham aos *JavaScript Object Notation* (JSON) do Javascript.

5.1.10 Amazon Web Services

A AWS, como é chamada, é uma plataforma na nuvem que possui uma variedade de ferramentas e recursos para a publicação de projetos pessoais e profissionais.

5.1.11 Amazon Relational Database Service

Também chamado de Amazon RDS, é uma ferramenta oferece a configuração e operação de um serviço de banco de dados relacional totalmente na nuvem. Esta ferramenta suporta o PostgreSQL que foi usado neste projeto.

5.1.12 Elastic Computing

O Amazon Elastic Computing, ou EC2, também faz parte do conjunto de ferramentas da AWS. Esta ferramenta é utilizada para prover um servidor para a publicação do projeto.

5.1.13 Simple Cloud Storage

Por último, temos o Amazon Simple Storage Service, também chamado de S3. É um serviço de armazenamento de objetos, como arquivos de mídia e dados de análise, entre outros. Esta ferramenta junto com todas as outras do pacote AWS fornecem a infraestrutura necessária para o projeto.

5.2 Estrutura do projeto

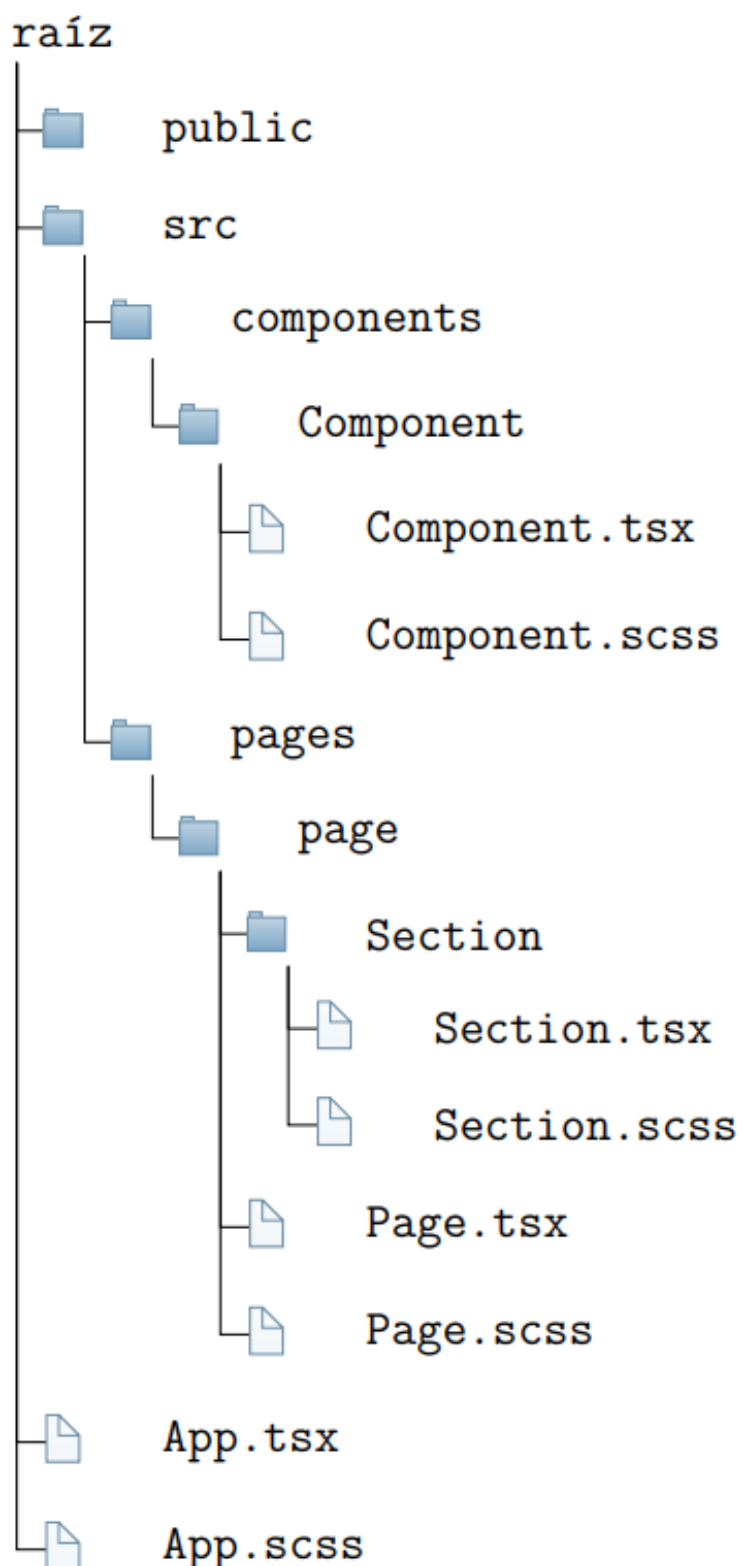
Pelo fato do projeto ser uma aplicação web, com foco em navegadores e dispositivos móveis, possui duas partes distintas, a interface gráfica, também chamada de *front-end* e a infraestrutura onde os dados são armazenados e as rotas para comunicação com a interface são instanciadas, o *back-end*.

5.2.1 Front-end

Com o uso dos componentes, o projeto tem uma abordagem *bottom-up*, ou seja, após criar os elementos menores, estes são agrupados em outros arquivos que formam áreas maiores e por fim, há o agrupamento destas áreas em arquivos que representam as páginas.

A estrutura geral do projeto seguindo o método *bottom-up* é representada de modo simplificado pela Figura 11.

Figura 11 – Estruturação do diretório de front-end



Fonte: Elaborado pelo autor.

Nesta Figura 11 estão representadas as estruturas básicas do projeto. A pasta `public` contém o resultado do processo de compilação, que gera arquivos prontos para serem publicados

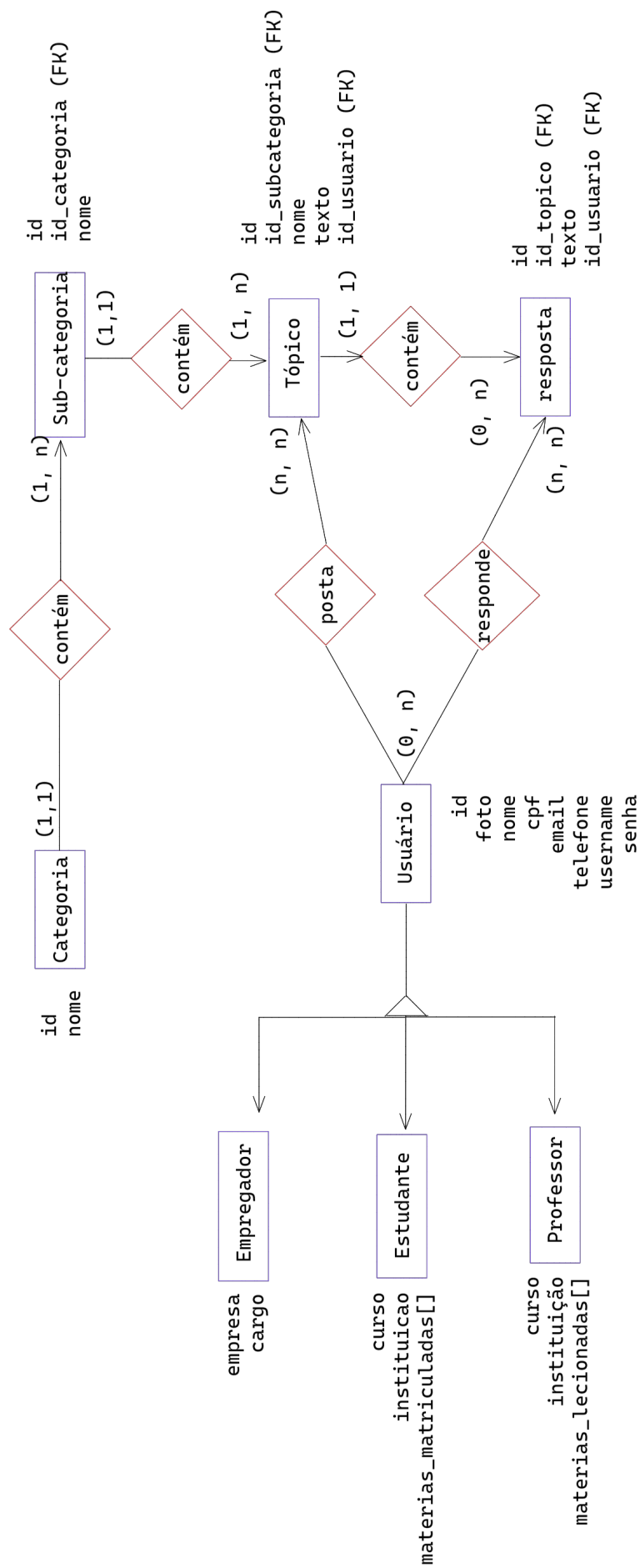
em serviços de infraestrutura e rede. O próximo diretório é o `src`, este contém os arquivos-fonte, e aqui é possível notar a abordagem *bottom-up* mencionada previamente. Cada componente está separado em um diretório próprio, contendo um arquivo `.tsx` que instancia suas próprias funções e também o código *HTML* que será mostrado no navegador. Junto a ele, há um arquivo `.scss`, o qual é responsável pela estilização. O mesmo acontece com as páginas que por sua vez são separadas pelas seções e unidas no arquivo `Page.tsx`. Por fim, na raiz do projeto o item `App.tsx` fica encarregado de unir as páginas por meio da definição das rotas de acesso.

5.2.2 Back-end

O *back-end* usa dois bancos de dados para armazenar toda a estrutura desenvolvida para o projeto, além do TypeORM que aliado ao NodeJs permite a criação de modelos genéricos que podem ser utilizados para todas as entidades e expandidos caso seja necessário expandir o projeto. A Figura 12 mostra um diagrama de entidade relacionamento do projeto com os campos de cada entidade.

Na estruturação interna também estão definidos as limitações de cada tipo de usuário e seus campos específicos, além de fazer a verificação se o e-mail colocado pelo usuário é o institucional.

Figura 12 – Diagrama de Entidade Relacionamento do projeto



Fonte: Elaborado pelo autor

6 Resultados

O principal resultado obtido foi o sucesso no desenvolvimento do projeto. Com o estudo prévio das ferramentas e da interface de usuário, o modelo escolhido foi o de fórum. Neste arquétipo usuários se cadastram para ter acesso a áreas de interesse onde podem se comunicar com os demais bem como iniciar discussões por meio de postagens com respostas.

A plataforma foi dividida em três categorias principais, sendo elas a Graduação, Discussão de novas tecnologias e Mercado de trabalho. Na área do curso, membros da graduação podem interagir em subcategorias destinadas às disciplinas ministradas, tirando dúvidas e discutindo sobre o que está sendo ensinado, fazendo com que haja uma aproximação com o ambiente físico onde espaços comuns de convivência eram usados para desenvolver os trabalhos em grupo, como também realizar atividades acadêmicas. Na área de discussão geral todos podem comentar sobre as tecnologias que estão sendo usadas por profissionais da área de tecnologia em formação ou em atuação no mercado. Por fim, na área destinada ao ambiente corporativo, representantes de empresas podem divulgar vagas de estágio e também de entrada no mercado de trabalho. A ideia é trazer um local semelhante aos murais que eram encontrados nos laboratórios e prédios da instituição, os quais continham folhetos de divulgação das vagas de emprego.

6.1 Desenvolvimento do back-end

A estruturação do *back-end* foi realizada usando o Typescript como linguagem. Juntamente a ele, utilizou-se o NodeJs como *framework* principal aliado ao NestJs e TypeORM para a criação da estrutura de rotas da API REST que se comunicará com a interface. A escolha do modelo REST deve-se ao seu formato de respostas em JSON, que é então interpretado pelo React e mostrado na tela para o usuário.

No Código 1 é possível ver o mapeamento das entidades junto ao banco de dados utilizando o TypeORM.

Código 1 – Exemplo de entidade “resposta” e seus campos usando TypeORM

```
1 public async function up(queryRunner: QueryRunner): Promise<void> {
2     queryRunner.createTable(
3         new Table({
4             name: 'repostas',
5             columns: [
6                 {
7                     name: 'id',
8                     type: 'uuid',
9                     isPrimary: true,
```

```

10         generationStrategy: 'uuid',
11         default: 'uuid_generate_v4()',
12     },
13     {
14         name: 'id_topico',
15         type: 'uuid',
16         isPrimary: true,
17         generationStrategy: 'uuid',
18     },
19     {
20         name: 'id_usuario',
21         type: 'uuid',
22         isPrimary: true,
23         generationStrategy: 'uuid',
24     },
25     {
26         name: 'texto',
27         type: 'varchar',
28     },
29 ]
30 })
31 )
32 }

```

No exemplo apresentado no Código 2 são adicionadas chaves estrangeiras aos campos “id_usuario” e “id_topico”, referenciando respectivamente a resposta ao usuário e ao tópico que ela pertence. Este retorno faz com que a interface consiga mostrar o perfil do usuário atrelado a sua resposta.

Código 2 – Exemplo da criação de Chaves Estrangeiras

```

1  await queryRunner.createForeignKey(
2      'topicos',
3      new TableForeignKey({
4          name: 'topicos',
5          columnNames: ['id_topico'],
6          referencedTableName: 'topicos',
7          referencedColumnNames: ['id'],
8          onDelete: 'SET NULL'
9      }),
10 );
11 await queryRunner.createForeignKey(
12     'topicos',
13     new TableForeignKey({
14         name: 'topicos',
15         columnNames: ['id_usuario'],
16         referencedTableName: 'usuarios',
17         referencedColumnNames: ['id'],

```

```

18         onDelete: 'SET NULL'
19     }},
20 );

```

Para representar a entidade resposta dentro do sistema, foi criada uma classe modelo “EntidadeBase” que contém todos os atributos base, os quais são comuns entre os modelos entidades, como mostrado no Código 3.

Código 3 – Criação da classe EntidadeBase

```

1  class EntidadeBase {
2      @PrimaryGeneratedColumn('uuid')
3      id: string;
4
5      @CreateDateColumn()
6      criado: Date;
7
8      @CreateDateColumn()
9      alterado: Date;
10 }
11
12 export default EntidadeBase;

```

No Código 3, *decorators* foram usados para indicar ao manipulador de dados qual a representação para cada atributo.

- *@PrimaryGeneratedColumn()* - Indica que esse atributo será uma chave primária que será gerada.
- *@CreateDateColumn()* - Indica que esse atributo será uma coluna do tipo “Data”

A representação da classe “Resposta” utiliza uma classe de mesmo nome que herda os atributos bases da classe “EntidadeBase” e a estende, adicionando os campos de interesse que foram definidos anteriormente, o fragmento apresentado no Código 4 mostra sua criação.

Código 4 – Classe “Resposta” derivada de EntidadeBase

```

1  @Entity('respostas')
2  class Resposta extends EntidadeBase {
3      @Column()
4      id_topico: string;
5
6      @Column()
7      id_usuario: string;
8
9      @Column()
10     texto: string;

```

```

11 }
12
13 export default Resposta;

```

O acesso aos dados é realizado por duas camadas responsáveis unicamente por isso e seguem o padrão do Repositório. Esse modelo permite o isolamento das camadas com a camada de controle e manipulação do fluxo, também chamada de camada de negócio. No projeto foi implementada uma interface que define o contrato e modelo que as classes do padrão de repositório devem seguir. Foi considerado dividir o contrato e a camada de repositórios em duas partes, sendo elas uma para leitura e uma para escrita, porém como se trata de um projeto de pequena escala não há a necessidade, sendo possível unir ambos. Em projetos de larga escala, a escolha mais comum é a divisão. O Código 5 mostra a interface do padrão de repositório.

Código 5 – Interface do padrão de repositório

```

1 interface IRepository<T> {
2     getById(id: string): Promise<T | undefined>;
3
4     getAll(): Promise<T[]>;
5
6     create(data: T): Promise<T>;
7
8     update(id: string, data: T): Promise<UpdateResult>;
9
10    delete(id: string, data: T): Promise<boolean>;
11 }
12
13 export default IRepository;

```

Essa interface é implementada pela classe que segue o padrão de repositório genérico, onde qualquer tipo de entidade pode ser instanciada e as manipulações básicas de CRUD são executadas, como demonstrado no Código 6 apresentado a seguir.

Código 6 – Manipulações básicas do CRUD

```

1 @EntityRepository()
2 class Repository<T> implements IRepository<T> {
3     private readonly _entity: ObjectType<T>;
4
5     constructor(entity: ObjectType<T>) {
6         this._entity = entity;
7     }
8
9     async getById(id: string): Promise<T | undefined> {
10         return await getRepository(this._entity).findOne(id);
11     }
12

```

```

13     async getAll(): Promise<T[]> {
14         return await getRepository(this._entity).find(undefined);
15     }
16
17     async create(data: T): Promise<T> {
18         return await getRepository(this._entity).create(data);
19     }
20
21     async update(id: string, data: T): Promise<UpdateResult> {
22         return await getRepository(this._entity).update(id, data);
23     }
24
25     async delete(id: string, data: T): Promise<boolean> {
26         await getRepository(this._entity).update(id, data);
27         return true;
28     }
29 }

```

Assim, para que a interface gráfica do sistema possa acessar a API, utilizou-se a camada de Controladores, os quais expõem os acessos aos dados para o acesso via requisições HTTP. Isso é demonstrado no Código 7.

Código 7 – Controlador responsável por retornar os dados via requisição HTTP

```

1  export default class RespostaController {
2
3      private readonly _repository = new Repository<Resposta>(Resposta
4          );
5
6      async index(response: Response, request: Request):
7          Promise<Response[]> {
8          const list = await this._repository.getAll();
9          return response.status(200).send(list);
10         }
11
12         async store(response: Response, request: Request):
13             Promise<Response> {
14             const data = request.all();
15             const resposta = await this._repository.create(data);
16             return response.status(201).send(resposta);
17         }
18
19         async show(response: Response, request: Request):
20             Promise<Response> {
21             const id = request.params();
22             const list = await this._repository.getById(id);
23             return response.status(200).send(list);
24         }
25
26         async destroy(response: Response, request: Request):
27             Promise<Response> {

```



```

20         const data = request.all()
21         const list = await this._repository.delete(data.id, data);
22         return response.status(204).send();
23     }
24 }

```

Aqui, cada requisição retorna o dado que foi pedido pela interface juntamente com um *status* do protocolo HTTP. Estes valores seguem um padrão universal que diz se a requisição foi finalizada com sucesso ou não, e indicam parcialmente o que foi realizado. No código mostrado temos os seguintes valores:

- **200:** Requisição bem sucedida.
- **201:** A requisição pediu a criação de um dado e o mesmo foi efetuado com sucesso.
- **204:** Neste caso, a requisição foi concluída com sucesso, porém ela não enviou nenhum dado ao servidor.

Por fim, a segurança no login dos usuários é realizada utilizando o padrão *JSON Web Token* (JWT). Este padrão utiliza uma criptografia para proteger os dados enviados e recebidos e também assina digitalmente as requisições, tornando-as únicas. Ao realizar o login enviando as informações para o servidor, um *token* único é retornado confirmando que a sessão foi iniciada com sucesso e envia informações pertinentes para o navegador. Na Figura 13 é mostrado um exemplo do *token*.

Figura 13 – JWT gerado no momento em que o usuário inicia a sessão

```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjAxYzZkZmNhLTcyMmEtNDY2Ny05YzdmLTY2MmUxNDU4ZWMyNyIsIm5hbWUiOiJMZW9uYXJkbyBTaWx2YXR0aSI6InRpcG8iOiJFc3R1ZGFudGUiLCJpYXQiOiJlMTYyMzkwMjJ9.6MfbT2j0mU7uFpNrYEwEwGpKHavjBuwIxCeR9RipeAo

```

Fonte: Elaborado pelo autor.

O dado é composto por três partes, que na Figura 13 estão separadas nas cores vermelho, rosa e azul. A parte vermelha indica o tipo de criptografia usada e define o tipo do *token* com JWT. A rosa é o *payload*, o qual contém um objeto com as informações que foram criptografadas pelo sistema. Por último, a parte azul é a assinatura digital única criada pela união de uma chave com os dados anteriores. A Figura 14 mostra o dado criptografado anteriormente.

Além dos dados do usuário, há um quarto campo chamado de "iat" que representa o tempo de validade do objeto.

Figura 14 – Informações que geraram o token

```
{
  "id": "01c6dfca-722a-4667-9c7f-662e1458ec27",
  "name": "Leonardo Silvatti",
  "tipo": "Estudante",
  "iat": 1516239022
}
```

Fonte: Elaborado pelo autor.

6.2 Desenvolvimento do *front-end*

O *front-end* é programado em Typescript com o React como biblioteca principal. A abordagem que esta estrutura fornece permite que o código seja subdividido em componentes que podem ser reusados em diversas páginas pelo projeto. Isso aumenta a padronização da interface visual e diminui a quantidade de código no projeto, fazendo com que ele se torne mais leve e performático. Assim sendo, podemos definir um componente `Button.tsx` e junto com ele um arquivo `Button.scss` que contém marcações de atributos para definir seu tipo e estilo e importá-lo ao longo do desenvolvimento para evitar que o código seja copiado e colado múltiplas vezes. O Código 8 mostra um exemplo de código de um componente de botão.

Código 8 – Exemplo de código de um componente de botão.

```
1 import React from 'react'
2 import { isMobile } from 'react-device-detect'
3 import Loading from '../Loading/Loading'
4 import './Button.scss'
5
6 interface ButtonProps extends
    React.ButtonHTMLAttributes<HTMLButtonElement> {
7   children: React.ReactNode
8   className?: string
9   loading?: boolean
10  buttonType: 'primary' | 'aux' | 'outline' | 'link'
11  size?: 'sm' | 'md' | 'lg'
12 }
13
14 function Button({
15   children,
16   className,
17   loading,
18   size,
19   buttonType,
20   ...props
21 }: ButtonProps) {
22   return (
```

```

23     <button
24         {...props}
25         className={`button button--${buttonType} button--${size}
26             ${
27                 className ? className : ''
28             }`}
29         style={{ opacity: loading ? 0.7 : 1 }}
30         >
31         {loading ? <Loading /> : children}
32     </button>
33 )
34
35 Button.defaultProps = {
36     size: isMobile ? 'md' : 'sm',
37 }
38
39 export default React.memo(Button)

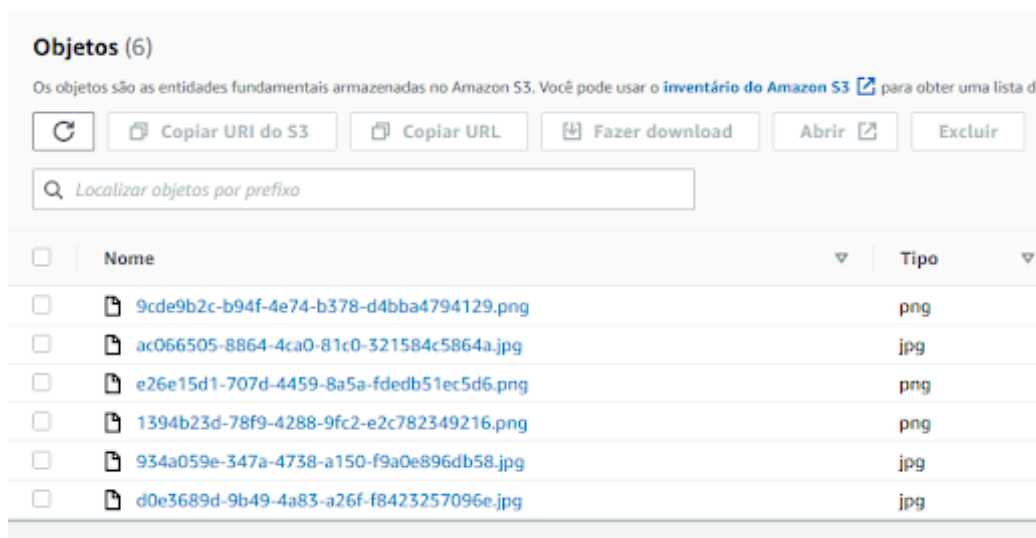
```

No Código 8 a criação de um componente de botão recebe atributos que definem seu tipo, tamanho e também um estado de carregamento quando clicado e a ação que ele realizará está em curso. Além disso, ele também aceita mais opções de customização caso receba classes da seção onde está contido.

6.3 Publicação

A Amazon Web Services é o serviço de nuvem da Amazon que oferece soluções baseadas em nuvem e foi utilizada neste trabalho para publicar a API assim como a interface. Como pontuado na metodologia, foram utilizados os serviços S3, RDS e EC2. O S3 é responsável por armazenar as imagens salvas do projeto, como fotos de perfil de usuário. A Figura 15 mostra a visualização do painel da ferramenta.

Figura 15 – Visão do painel da ferramenta Amazon S3

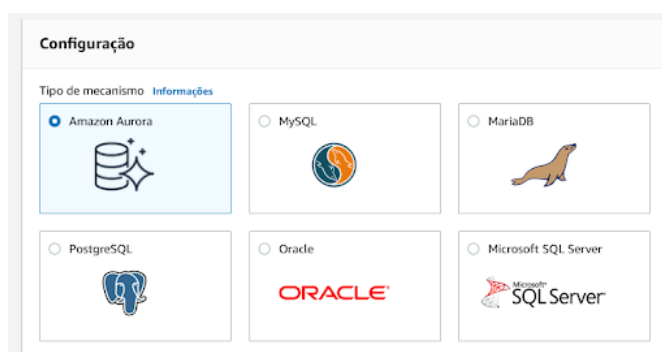


Fonte: Elaborado pelo autor.

A segurança das informações dos usuários é assegurada pois as imagens são salvas com nomes gerados aleatoriamente durante seu *upload* e apenas seu caminho é relacionado ao responsável por colocá-la na plataforma.

Para instanciar os bancos relacionais e não relacionais foi usado o Amazon Relational Database Service (RDS). O serviço oferece diferentes opções de banco de dados. No caso deste projeto foi usado PostgreSQL e MongoDB. A tela de seleção é mostrada na Figura 16.

Figura 16 – Configuração de uma nova instância de banco no RDS



Fonte: Elaborado pelo autor.

Após a configuração o endereço para acessar a base de dados é exposto pelo sistema, isso pode ser observado na Figura 17. Com este caminho em mãos a configuração é realizada para que API desenvolvida na seção de *back-end* aponte para a nova instância criada no RDS, como mostrado na Figura 18.

Figura 17 – Informações da instância com link de acesso no RDS

Segurança e conexão		
Endpoint e porta	Redes	Segurança
Endpoint postgres.cnoh2qavdv8g.us-east-2.rds.amazonaws.com	Zona de disponibilidade us-east-2a	Grupo de segurança da VPC rds-launch-wizard-1 (sg-080be08ecf81c0146)
Porta 5432	VPC vpc-0ef12f331b26fd178	Ativo
	Grupo de sub-redes default	rds-launch-wizard (sg-0cfdc1319647f4ae8)
	Sub-redes subnet-09bd6c8b8ac4bb5c0 subnet-09c4eaf95768b4234 subnet-0a9aae89b8c0ef19	Ativo
		default (sg-05f089f5cfabda7116)
		Ativo
		Publicamente acessível
		Sim
		Autoridade de certificação
		rds-ca-2019
		Data da autoridade de certificado
		August 22, 2024, 02:08 [UTC+2:08]

Fonte: Elaborado pelo autor.

Figura 18 – Arquivo de configuração da API

```

PORT=3333
HOST=0.0.0.0
NODE_ENV=development
APP_KEY=
APP_URL=http://${HOST}:${PORT}
DB_CONNECTION=pg
PG_HOST=posgres.cnoh2qavdv8g.us-east-2.rds.amazonaws.com
PG_PORT=5432
PG_USER=postgres
PG_PASSWORD=12345678
PG_DB_NAME=api_blogs
APP_SECRET_KEY:aaaaaaaaa
AWS_ACCESS_KEY_ID=secret
AWS_SECRET_ACCESS_KEY=secret
AWS_REGION=us-east-2
AWS_S3_BUCKET=s3-api-blogs
SESSION_DRIVER=cookie

```

Fonte: Elaborado pelo autor.

No arquivo de configuração (Figura 18), alguns campos de destaque que são responsáveis por apontar para o banco de dados são:

- **PG_HOST:** Link para o banco de dados.
- **PG_PORT:** Porta usada pelo banco.
- **PG_USER:** Usuário padrão da base.
- **PG_PASSWORD:** Senha do banco.
- **PG_BD_NAME:** Nome da base.

Por fim, o conteúdo é dividido em duas instâncias distintas do Elastic Compute Cloud (EC2) sendo que cada uma destas contém uma aplicação diferente (uma contém a interface e outra o banco com a API), isto se dá por questão de segurança e escalabilidade das ferramentas. A configuração pode ser observada na Figura 19. Ambas utilizam o sistema operacional Linux com a distribuição Ubuntu. As instâncias do EC2 são em sua essência máquinas virtuais, portanto foi necessário instalar as dependências necessárias e executar as aplicações via terminal. Por fim, os IPs de acesso às aplicações de banco e interface são expostos para a Internet. O painel do serviço o qual contém as máquinas é mostrado na Figura 20.

Figura 19 – Configuração de uma máquina virtual no EC2

The screenshot displays the Amazon EC2 console's 'Choose an AMI' page. It lists four operating system images, each with a logo, name, version, and a 'Selecionar' (Select) button. Below each name is a brief description and the root device type, virtualization type, and ENA support status. The AMIs are:

- Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type** (ami-088911543b10876a4 (64 bits x86) / ami-0e5b56e44b3b60aaa (64 bits Arm)). Description: Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard. Tipo de dispositivo raiz: ebs Tipo de virtualização: hvm ENA habilitado: Sim.
- Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** (ami-0c2485d67d416fc4f (64 bits x86) / ami-09ae0a24dd3aca40c (64 bits Arm)). Description: Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type. Tipo de dispositivo raiz: ebs Tipo de virtualização: hvm ENA habilitado: Sim.
- SUSE Linux Enterprise Server 15 SP3 (HVM), SSD Volume Type** (ami-0cb38590c7f60bb35 (64 bits x86) / ami-0a8fa45e5f3aacd1d (64 bits Arm)). Description: SUSE Linux Enterprise Server 15 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled: Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available. Tipo de dispositivo raiz: ebs Tipo de virtualização: hvm ENA habilitado: Sim.
- Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** (ami-090006f29ecb2d79a (64 bits x86) / ami-0e9f8e8acb00ab545 (64 bits Arm)). Description: Ubuntu Server 20.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Fonte: Elaborado pelo autor.

Figura 20 – Painel EC2 com as máquinas virtuais criadas

ID de instância	Estado da inst...	Tipo de inst...	Verificação de status	Status do al...	Zona d...	DNS IPv4 público	Endere...
i-0a9e5ab1b2690e5be	Executando	t2.micro	2/2 verificações aprovac	Sem alar...	sa-east-1a	ec2-54-207-28-125...	54.207.28...
i-012fce933f94a80e5	Executando	t2.micro	2/2 verificações aprovac	Sem alar...	sa-east-1a	ec2-54-207-239-48...	54.207.23...

Fonte: Elaborado pelo autor.

7 Conclusão

Ao fim do trabalho foi obtido uma rede de apoio estudantil focada nos cursos de Tecnologia de Informação. Pelos padrões utilizados a rede é escalável e pode ser incorporada a cursos diferenciados além da área de TI. Os conceitos desenvolvidos englobam todas as partes do processo de criação de um produto real, para isso foram estudados conceitos de Interface de Usuário e Experiência de Usuário, os quais ajudaram na criação de uma interface considerada intuitiva e familiar para que os utilizadores não se sentissem perdidos durante a sessão. Além disso, as tecnologias aplicadas refletem a modernidade das ferramentas dispostas nos dias atuais, desde a programação visual até o desenvolvimento do banco e dos controladores para expor os dados que serão consumidos pela interface.

Acredita-se que num contexto educacional a distância a ferramenta pode atuar como redutora da distância imposta pela situação, fazendo com que haja uma interação maior entre as turmas das matérias lecionadas pelos professores e também entre todos os alunos do curso, estimulando as discussões e interações que haviam no ambiente físico. Isso impacta positivamente no psicológico dos indivíduos, colocando-os próximos uns aos outros mesmo que virtualmente. O processo diminui o dano mental que a situação de isolamento coloca, já que é uma das medidas divulgadas por matérias focadas em saúde ([UNICEF, 2020](#)).

Por fim, a ferramenta se adequa aos novos padrões do mercado de trabalho, que seguirá em sua maioria o modelo remoto de contratação. A área especializada no ambiente corporativo faz com que os estudantes tenham contato com profissionais de recrutamento de forma facilitada e possam ter contato direto com estes contratadores.

Referências

AELA. *Os 7 Princípios de Gestalt e Como Utilizá-los em Projetos de UI Design*. 2020. Disponível em: <<https://medium.com/aela/os-7-princ%C3%AADpios-de-gestalt-e-como-utiliz%C3%A1-los-em-projetos-de-ui-design-46d6d832abf6>>. Acesso em: 09 fev. 2022.

BRASIL. Portaria nº 345, de 19 de março de 2020. *Diário Oficial da União*, Brasília, DF, 2020. ISSN 1677-7042. Disponível em: <<https://www.in.gov.br/en/web/dou/-/portaria-n-345-de-19-de-marco-de-2020-248881422?inheritRedirect=true&redirect=%2Fweb%2Fguest%2Fsearch%3FqSearch%3DPortaria%2520345%2520de%252019%2520de%2520mar%25C3%25A7o%2520de%25202020>>. Acesso em: 20 jan. 2022.

CARNEIRO, L. d. A.; RODRIGUES, W.; FRANÇA, G.; PRATA, D. N. Use of technologies in brazilian public higher education in times of pandemic covid-19. *Research, Society and Development*, v. 9, n. 8, p. e267985485, Jul. 2020. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/5485>>. Acesso em: 20 jan. 2022.

DIGIDEIAS. *Crescimento Das Redes Sociais Durante A Pandemia*. 2021. Disponível em: <https://www.digideias.com.br/crescimento_das_redes_sociais_durante_a_pandemia/>. Acesso em: 28 dez. 2021.

IFMG. *Diferentes tipos de tipografia: GUIA completo para não designers*. 2021. Disponível em: <<https://www.bambui.ifmg.edu.br/portal/uncategorised/diferentes-tipos-de-tipografia-guia-completo-para-nao-designers>>. Acesso em: 09 fev. 2022.

JACOBSON, L. V.; FLEURY, M. T. L. Contribution of an internet discussion forum to a business administration course. *REGE Revista de Gestão*, v. 12, n. 1, p. 12, Mar. 2005. Disponível em: <<https://www.revistas.usp.br/rege/article/view/36511>>. Acesso em: 20 jun. 2021.

OLHAR DIGITAL. *Pós-pandemia: 78% dos profissionais de TI preferem manter o home office*. 2021. Disponível em: <<https://olhardigital.com.br/2021/11/05/pro-pos-pandemia-78-dos-profissionais-de-ti-preferem-manter-o-home-office/>>. Acesso em: 20 dez. 2021.

REVISTA ARCO. *Efeitos colaterais do distanciamento físico na saúde mental*. 2020. Disponível em: <<https://www.ufsm.br/midias/arco/distanciamento-fisico-saude-mental/>>. Acesso em: 28 dez. 2021.

UNICEF. *Efeitos colaterais do distanciamento físico na saúde mental*. 2020. Disponível em: <<https://www.unicef.org/brazil/historias/como-adolescentes-podem-proteger-sua-saude-mental-durante-a-pandemia-de-covid-19>>. Acesso em: 25 jan. 2022.

VALOR INTESTE. *Mercado de tecnologia tem aumento de 310% de vagas em 2020*. 2021. Disponível em: <<https://valorinveste.globo.com/objetivo/empreenda-se/noticia/2021/01/10/mercado-de-tecnologia-tem-aumento-de-310percent-de-vagas-em-2020.ghtml>>. Acesso em: 20 dez. 2021.