



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

GABRIEL DE CARVALHO FONSECA

**HIVE:
UMA REDE INTERLIGADA DE CONTRATAÇÃO**

CAMPINA GRANDE - PB

2023

GABRIEL DE CARVALHO FONSECA

HIVE:

UMA REDE INTERLIGADA DE CONTRATAÇÃO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Professor Dr. João Arthur Brunet Monteiro

CAMPINA GRANDE - PB

2023

GABRIEL DE CARVALHO FONSECA

HIVE:

UMA REDE INTERLIGADA DE CONTRATAÇÃO

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

**Professor Dr. João Arthur Brunet Monteiro
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Carlos Wilson Dantas de Almeida
Examinador – UASC/CEEI/UFCG**

**Melina Mongiovi
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 17 de Novembro de 2023.

CAMPINA GRANDE - PB

RESUMO

Com a pandemia houve um crescimento de 671% na procura por profissionais na área de TI, criando-se um grande número de oportunidades para candidatos. Porém é possível no entanto perceber que há também a dificuldade de encontrar a vaga adequada para o profissional em questão, podendo muitas vezes não se identificar com as oportunidades que lhe são oferecidas em diversas plataformas de divulgação de vagas. Este trabalho propõe a criação de uma plataforma web, que irá se utilizar de conhecimentos de algoritmos de relevância por documento para que haja um ranqueamento das oportunidades de emprego, e assim sejam sugeridas as melhores oportunidades para o usuário. Ao final deste projeto foi desenvolvido como resultado uma plataforma web para busca e cadastro de oportunidades com a ênfase na transparência de informações para seus usuários, com a sua validação sendo realizada através de um formulário disponibilizado para os usuários, coletando informações a respeito da usabilidade do sistema e recebendo um feedback por parte deles de pontos a serem melhorados futuramente. Em geral os resultados apontam para boas avaliações na usabilidade e facilidade do sistema, com algumas sugestões de melhorias nos pontos de responsividade da plataforma e integração de algumas funcionalidades no sistema.

Hive: Uma Rede interligada de contratação

Gabriel de Carvalho Fonseca

Gabriel.fonseca@ccc.ufcg.edu.br

Universidade Federal de Campina Grande

Campina Grande, Paraíba, Brasil

João Arthur Brunet

joao.arthur@computacao.ufcg.edu.br

Universidade Federal de Campina Grande

Campina Grande, Paraíba, Brasil

Resumo

Com a pandemia houve um crescimento de 671% na procura por profissionais na área de TI, criando-se um grande número de oportunidades para candidatos. Porém é possível no entanto perceber que há também a dificuldade de encontrar a vaga adequada para o profissional em questão, podendo muitas vezes não se identificar com as oportunidades que lhe são oferecidas em diversas plataformas de divulgação de vagas. Este trabalho propõe a criação de uma plataforma web, que irá se utilizar de conhecimentos de algoritmos de relevância por documento para que haja um ranqueamento das oportunidades de emprego, e assim sejam sugeridas as melhores oportunidades para o usuário. Ao final deste projeto foi desenvolvido como resultado uma plataforma web para busca e cadastro de oportunidades com a ênfase na transparência de informações para seus usuários, com a sua validação sendo realizada através de um formulário disponibilizado para os usuários, coletando informações a respeito da usabilidade do sistema e recebendo um feedback por parte deles de pontos a serem melhorados futuramente. Em geral os resultados apontam para boas avaliações na usabilidade e facilidade do sistema, com algumas sugestões de melhorias nos pontos de responsividade da plataforma e integração de algumas funcionalidades no sistema.

1. INTRODUÇÃO

Com a pandemia houve um crescimento na demanda por profissionais na área de TI, de acordo com a CNN Brasil houve um crescimento de 671% na procura por profissionais, criando-se um grande número de oportunidades[1]. Com uma quantidade tão crescente de vagas é natural que as empresas utilizem plataformas web para realizar as suas divulgações e atingir o maior número possível de candidatos disponíveis no mercado.

No entanto, há um grande fator negativo que é a falta de transparência nas oportunidades, exigindo muitas qualificações dos possíveis candidatos, mas não disponibilizando em troca informações detalhadas a respeito da vaga, como por exemplo o modelo de contratação, projeto que irá participar ou remuneração que estariam dispostos a negociar. Muitas vezes é preciso recorrer a sites de terceiros para encontrar essas informações, que podem estar desatualizadas ou incorretas. De acordo com um levantamento feito pela consultoria de RH Luandre, 94% dos entrevistados consideram importante saber com antecedência a quantidade de etapas na seleção[2].

Essa falta de informação não somente é prejudicial, mas pode ser frustrante para diversos profissionais que gostariam de poder alinhar suas expectativas com as vagas de trabalho que procuram.

Neste trabalho proponho a resolução do problema com o desenvolvimento de uma plataforma web de busca de vagas com

ênfase na transparência de informações para os candidatos, onde os mesmos terão informações prévias a respeito das oportunidades. Um dos pilares da plataforma é a utilização de um algoritmo de busca que irá fazer um ranqueamento dos resultados levando em consideração a descrição da vaga com as palavras chaves utilizadas na busca.

Como resultado deste projeto foi desenvolvido o sistema web para cadastro e busca de oportunidades, utilizando o padrão arquitetural REST, composto por um frontend em React e um backend em Spring Boot, onde é possível realizar o cadastro de usuários e oportunidades de emprego. Para a avaliação foi disposto um formulário conferindo a usabilidade do sistema e recebendo também um feedback geral da experiência do usuário a partir de voluntários para testar a plataforma web. No final foram obtidos resultados positivos quanto a sua utilização, em destaque para sua facilidade de uso, porém foram destacados alguns pontos de melhoria, como a integração das funcionalidades e a falta de responsividade para algumas resoluções de monitores.

Este documento está organizado da seguinte maneira. Na segunda seção irei detalhar como foram levantados os requisitos e descrever a metodologia adotada. Na terceira seção, a solução web será descrita, sendo essa dividida em três subseções, uma descrevendo as tecnologias do BackEnd, uma descrevendo o FrontEnd e finalmente descrevendo a Arquitetura adotada. Na quarta seção contará com a demonstração de resultados obtidos através de avaliações de usuários. Na quinta seção será descrita a experiência de aprendizado e desenvolvimento da ferramenta. E na sexta e última seção, como esse trabalho poderá ser evoluído futuramente.

2. METODOLOGIA

Para esse projeto foi feito um levantamento de serviços com referência nas necessidades de possíveis usuários reais e levando em consideração também o que é oferecido em questão de funcionalidades em diversas plataformas concorrentes, entre elas o LinkedIn, ProgramaThor e GlassDoor. Considerando o problema inicial que seria falta de transparência, houve também um levantamento de quais seriam os aspectos que geralmente são descritivos de uma oportunidade mas que não estão sempre presentes em sua descrição nas plataformas anteriormente citadas. Cheguei a conclusão que seriam: o salário proposto, possibilidade de negociação salarial, o nível necessário de experiência para a vaga, se a vaga seria CLT ou PJ e caso ela fosse remota, híbrida ou presencial.

Uma vez reunidos os requisitos necessários, eles foram listados e então selecionados a partir dos quais seriam essenciais para uma primeira versão do MVP. Foi feito então um backlog do produto utilizando o quadro kanban do trello, onde foram categorizados e ordenados por prioridade, auxiliando assim a visualização do

andamento de forma organizada do projeto. Utilizando a framework Scrum, foram então adotadas sprints de duas semanas para o desenvolvimento das stories criadas. Para versionamento do projeto foi utilizado o Github, sendo criado um repositório tanto para o backend quanto para o frontend sendo ambos open source, onde são realizados os commits por feature cadastrada no backlog.

Uma vez concluído, o projeto foi validado realizando testes com usuários reais, sendo eles também profissionais da área de computação, onde seguiram os fluxos disponíveis do sistema para suas diversas funcionalidades. Ao final eles responderam um formulário criado utilizando o SUS, System Usability Scale, uma escala numérica que tenta medir a usabilidade do sistema, fornecendo também um feedback no final a respeito dos pontos que achavam negativos. Com isso foi possível traçar conclusões e novos objetivos para o futuro da aplicação.

3. SOLUÇÃO

Foi desenvolvida uma plataforma web, denominada Hive, onde será feita a divulgação do processo de contratação, com ênfase em transparência. Essa plataforma foi criada utilizando um BackEnd com o framework Spring Boot e um FrontEnd com o framework React. Spring foi selecionado devido aos fator de familiaridade com a ferramenta agilizando o seu desenvolvimento, enquanto que React foi escolhida devido ao fato de ser uma framework amplamente utilizada comercialmente, contando com diversos recursos que auxiliam no seu uso, como por exemplo o Material-UI que conta com uma vasta lista de elementos prontos para serem utilizados na criação dos componentes do site.

Um dos pontos de importância do sistema é a utilização do algoritmo de ranqueamento TF-IDF, Term Frequency-Inverse Document Frequency. Ao realizar uma pesquisa na barra de busca do site é então enviado para o servidor as palavras-chave que vão compor a query no banco de dados. É feita então uma busca por todas as vagas de emprego que contêm, seja em seu título ou descrição, pelo menos um dos termos utilizado na busca pelo usuário. Uma vez recuperados do banco de dados, eles são ordenados de acordo com a pontuação do TF-IDF.

A pontuação de relevância de cada documento é obtido através do produto da frequência com qual o termo aparece em um certo documento com a sua importância do termo no conjunto de documentos em questão, sendo calculado através da raridade de sua presença na coleção de documentos. Quanto mais raro o termo, maior é o peso da sua pontuação, isso permite que palavras geralmente utilizadas, mas que não carregam significado, sejam desconsideradas.

Irei detalhar e listar a seguir os requisitos funcionais e não funcionais do sistema.

3.1 Requisitos Funcionais

As principais funcionalidades foram estruturadas em stories com seus respectivos clientes e níveis de risco, que estão classificados em 3 níveis, podendo ser de risco baixo, risco moderado e risco alto. A seguir estão listados os principais serviços:

- Criação de página do candidato: Como um novo usuário da plataforma, serei capaz de realizar o cadastro e criação de uma página com minhas informações profissionais. Será possível também a atualização com

novas informações. Esta funcionalidade possui um risco baixo.

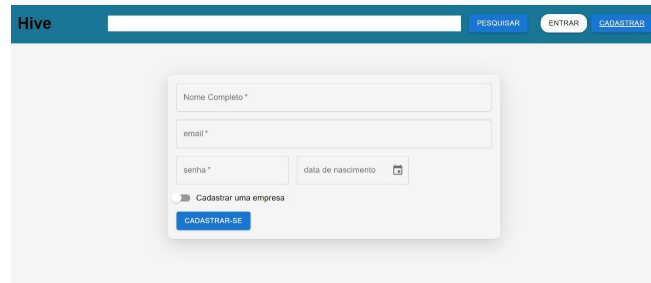


Figura 1: Página de cadastro de usuário

- Página da Empresa: Como um usuário dono de uma empresa ou negócio será possível realizar a criação de uma página para seus respectivos empreendimentos, podendo dispor de suas informações e cadastrar processos seletivos através do perfil do dono da página. Será possível realizar a deleção ou atualização da mesma. Esta funcionalidade possui um risco baixo.

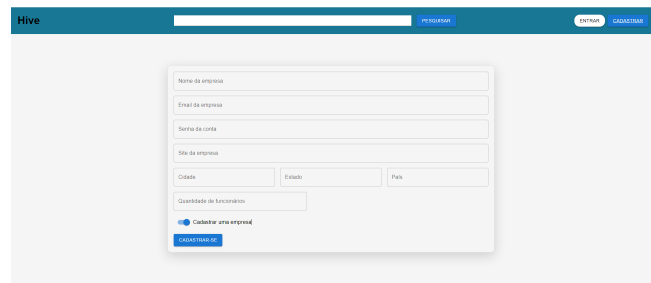


Figura 2: Página de cadastro da empresa

- Criação de um processo seletivo: Como um usuário cadastrado e dono de uma página de empresa, devo ser capaz de realizar o cadastro de um novo processo seletivo através de um formulário disponível na página de oportunidades correntes da minha empresa. Esse formulário contará com diversos campos que deverão ser preenchidos para garantir uma transparência na descrição da oportunidade. Risco baixo

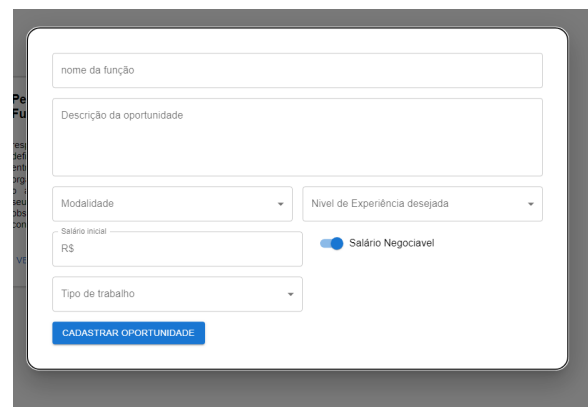


Figura 3: Cadastro de uma vaga de emprego

- Encerrar processo seletivo: Como um usuário cadastrado e dono de uma página de empresa, devo poder a qualquer momento encerrar um processo seletivo em andamento que seja pertencente a minha empresa. Risco baixo.



Figura 4: Tela de encerramento da vaga

- Participação em um processo seletivo: Como um usuário já cadastrado deve ser possível realizar o cadastro em um processo seletivo selecionando o currículo que irei enviar para a determinada vaga. Risco baixo

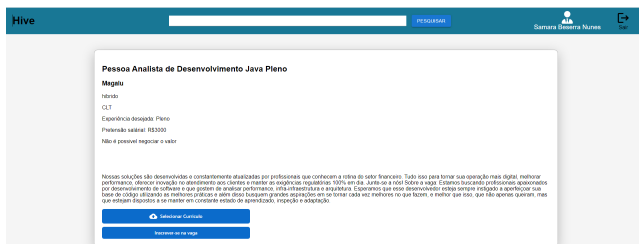


Figura 5: Tela que demonstra detalhes da vaga.

- Ferramenta de busca: Como usuário ou utilizando a página da empresa, deve ser possível realizar uma busca que retorna itens correspondentes às palavras chaves, ordenados por relevância. Risco alto.



Figura 6: Tela inicial de Busca

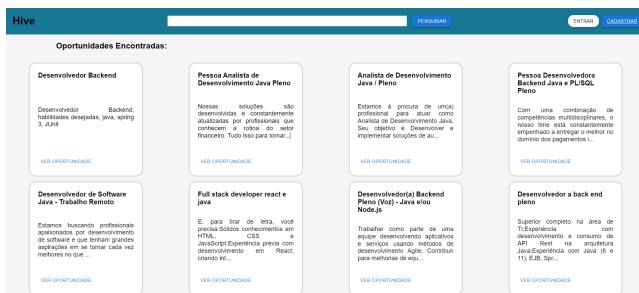


Figura 7: Tela de resultado da Busca

- Listagem de candidatos: Como um usuário dono de uma empresa, serei capaz de acessar a página da oportunidade que cadastrei, conferindo os candidatos que escolheram se cadastrar na vaga. Poderei ver então informações de email para contato do usuário, seu nome e o currículo enviado na sua inscrição. Risco moderado.

Nome do participante	email	currículo
Jonseca	fonseca@email.com	VER CURRÍCULO
matheus lisboa oliveira dos santos	matheusmat102@gmail.com	VER CURRÍCULO
rafaela	rafa.silva8022@gmail.com	VER CURRÍCULO
Sammara Nunes	samaranunes@gmail.com	VER CURRÍCULO

Figura 8: Tela de listagem dos candidatos

Os serviços de criação de usuário e da página da empresa são de baixo risco por se tratarem de uma simples inserção, exclusão e atualização de informações no banco de dados, além da disponibilização de informações para o cliente web.

O serviço de cadastrar o processo de vaga é de baixo risco já que envolve apenas realizar a criação do processo. O encerramento de um processo seletivo também possui uma dificuldade baixa uma vez que envolve apenas atualizar na base de dados o status de uma vaga, impedindo que novos usuários se cadastrem nela..

Na funcionalidade de Ferramenta de busca será implementado o algoritmo de frequência de documentos inversa, tf-idf. Através dele será feito um ranqueamento dos documentos recuperados do banco de dados utilizando uma query que será recuperada como entrada pelo usuário. O ranqueamento é feito com base na relevância de acordo com a frequência dos termos da query na descrição dos documentos, que será a descrição de cada vaga. Por envolver essa complexidade na implementação do algoritmo, é considerado de alto risco.

Além dos requisitos funcionais, será detalhado a seguir também os requisitos não-funcionais necessários para o sistema.

3.2 Requisitos não-funcionais

Os seguintes requisitos não-funcionais foram levantados e dados como necessário para que o sistema consiga atingir um bom nível de desempenho e qualidade para o usuário:

- disponibilidade: A implantação foi realizada em um primeiro momento em uma máquina virtual na AWS. O servidor está rodando 24 horas garantindo a disponibilidade do sistema sempre que que um usuário for utiliza-la.
- segurança: Garantia de que ao utilizarem o serviço da plataforma os dados e seu acesso estarão seguros. As informações sensíveis do usuário são codificadas através de um processo de Hash antes de serem salvas no banco de dados através de funções prontas da biblioteca Bcrypt. A segurança durante as requisições da aplicação se dá através da utilização de um JWT token durante a sessão do usuário.
- Usabilidade: Interface de usuário que gere uma boa experiência e que seja de fácil utilização. Utilizando-se do Material-UI é possível garantir diversos elementos que garantem uma boa experiência para o usuário. Ao desenvolver-se também foi focado em manter o design minimalista e intuitivo.

- **Desempenho:** A aplicação foi planejada para que as requisições sejam do menor tamanho possível, sendo sempre quebradas em requisições menores, como por exemplo na paginação da aplicação onde a busca é realizada a cada página, evitando assim uma busca por dados desnecessária, dando um melhor desempenho.

Na seguinte seção irei detalhar a arquitetura adotada e seus componentes.

3.3 Arquitetura

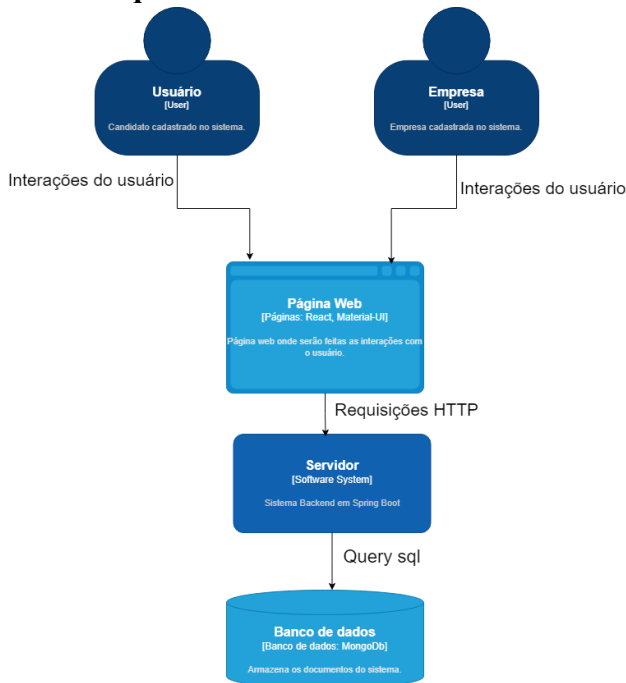


figura 9: Arquitetura geral do serviço

Para este projeto foi adotado uma arquitetura utilizando o padrão arquitetural REST onde o sistema foi dividido nos seguintes componentes: servidor Backend, desenvolvido com o Framework Spring Boot, banco de dados não-relacional, adotando o banco MongoDB que se encarregará da persistência dos dados em documentos e um Frontend, desenvolvido em React, no qual se encarregará de atuar como uma plataforma de utilização para o cliente.

Ambas as partes do sistema são open-source em um repositório público no github, sendo utilizado também para cuidar do versionamento do projeto..

nas próximas seções irei detalhar cada um dos componentes do sistema.

3.3.1 Backend

Para a tecnologia do Backend foi adotada a linguagem de programação Java em conjunto com o framework Spring Boot, facilitando assim a implementação da solução uma vez que a framework não somente é rica em recursos, mas conta com suporte a diversas tecnologias, permitindo autenticação com filtros de segurança e até mesmo uma interface específica que dá suporte ao banco de dados escolhido

Foi escolhido o modo de autenticação por token, onde no Backend é gerado um token JWT, onde são armazenadas informações relevantes para autenticação do usuário, quando o usuário efetua um login no sistema é retornado esse token para o frontend onde será guardado e enviado em futuras requests. Se utilizando dos recursos do Spring Security foi implementado um filtro customizado onde são abordadas todas as requests feitas ao servidor, então confere-se se ela possui um bearer no header da requisição contendo o token.

O sistema foi dividido em camadas seguindo o padrão mais tradicional para projetos Backend com as seguintes divisões: Dados, onde são modelados os documentos que serão salvos na base do banco de dados, Services, onde está definida propriamente a lógica de negócios e os Controllers, responsáveis pela disponibilização dos serviços para consumo externo através de rotas HTTP. Já na estrutura de pastas foi adotado a recomendação feita pela própria documentação do Spring, onde os componentes são agrupados pelas pastas das suas respectivas entidades do sistema, como mostrado na figura 7.

No banco de dados foi escolhido o MongoDB, um banco de dados não-relacional, que conta com um bom suporte para utilização na framework mencionada, onde estão salvos os documentos referentes às estruturas de empresa, usuário e vagas de processos seletivos.

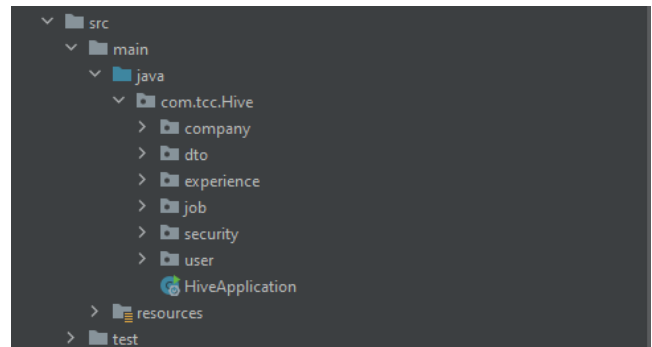


Figura 10: Estrutura das pastas do projeto

```

_id: ObjectId('652ee971b2b636ddcd3c5664')
jobTitle: "Pessoa Desenvolvedora Fullstack III NodeJs/React"
companyID: "ti@email.com"
income: 4000
negotiable: true
typeOfJob: "Presencial"
modality: "PJ"
seniority: "Pleno"
active: true
jobDesc: "responsabilidades: Participar da definição dos objetivos de suas entre..."

```

Figura 11: Estrutura de um documento do tipo Job no MongoDB

3.3.2 Frontend

Para o Frontend foi decidido a adoção de JavaScript como linguagem de programação e o Framework React. Nela é possível dividir os elementos da página em componentes, que são renderizados na tela permitindo uma decomposição tanto da lógica quanto da renderização em pequenas partes, possibilitando também a reutilização dos mesmos, facilitando o desenvolvimento. Na Figura 7 é possível conferir o exemplo de como são agrupados esses elementos no projeto, para cada pasta é separado o arquivo JS e CSS respectivos daquele componente.

A estrutura de pastas adotado pelo projeto no frontend conta com uma separação dos componentes em pastas, contando com seus arquivos JavaScript e CSS. Para realizar as requisições para o backend foi adotado o Axios, um cliente HTTP baseado em promises, sendo declarado em um arquivo separado ao iniciar o projeto contendo a url base, e sendo referenciado em cada componente ao realizar uma requisição.

Para agilidade no desenvolvimento e garantir também uma ótima experiência para o usuário, foi utilizado o Material-UI, uma biblioteca de componentes para React que fornece uma vasta variedade de elementos prontos para utilização imediata. Foi adotada a abordagem de criar componentes grandes que representam as páginas e então quebrá-los em unidades cada vez menores. Está sendo utilizado a biblioteca React Router para realizar um roteamento das páginas no lado do cliente, nela foi possível implementar os fluxos entre os componentes e estabelecer as urls para cada página criada.

Através do Figma, uma ferramenta para desenvolvimento de interfaces e protótipos, foi planejado o layout das telas e identidade visual do site, além da estruturação dos fluxos de navegação.

A plataforma está integrada com o servidor Backend, permitindo assim a utilização de suas funcionalidades através do browser como cliente. A autenticação como foi mencionada na seção anterior é feita através de um token, no qual o front recebe após realizar o login do usuário. Esse token então é guardado no local storage, permanecendo salvo entre as sessões ou até seu tempo expirar. Todas as requisições que necessitam de autenticação contém o Header com o bearer token.

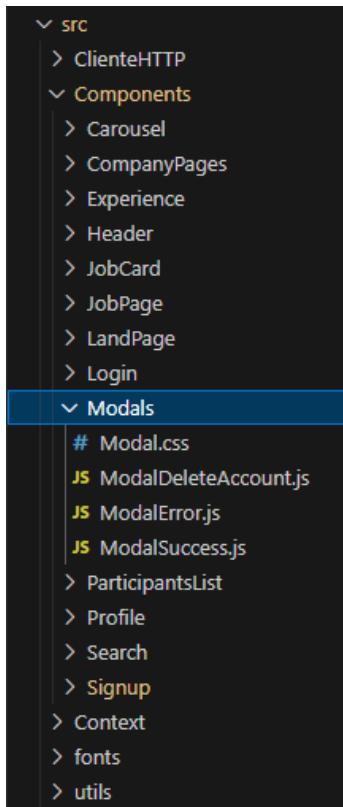


Figura 12: Estrutura de pastas do FrontEnd

4. AVALIAÇÃO

Para realizar a avaliação do projeto foi elaborado um questionário com base no System Usability Scale(SUS), uma ferramenta utilizada para medir a usabilidade do sistema, composta de 10 afirmações, utilizando a escala Likert de avaliação que vai do valor de 1 a 5, com 1 significando que “discorda completamente” e 5 significando que “concordo plenamente”. Para realizar o teste foram convidados graduandos do curso de computação e desenvolvedores já graduados na área, no qual 6 pessoas se voluntariaram para testar o sistema. Além disso, foi feita uma décima primeira pergunta, focada no contexto da aplicação, no qual questiona a importância para os candidatos a respeito da transparência de informações de uma vaga de emprego, além de um feedback sobre pontos negativos durante sua utilização.

No teste elas foram conduzidas por um roteiro a realizarem os fluxos de cadastro do usuário, de busca por uma vaga, candidatar-se a uma vaga, alterar seus dados cadastrais, cadastrar experiências novas no perfil e deletar suas respectivas contas. Duas pessoas das seis se disponibilizaram a realizar o fluxo de criação de uma empresa, cadastro de uma vaga, encerrar uma vaga, visualizar os candidatos a uma vaga criada e deletar a empresa.

Ao concluírem os fluxos determinados para teste da aplicação, eles foram submetidos a um formulário com 10 afirmações para darem a nota de acordo com sua experiência. Em geral os resultados apontam para uma facilidade de uso na ferramenta, atingindo assim o objetivo de simplicidade para o usuário. É possível conferir também que os principais pontos a serem melhorados na aplicação seriam a correção de inconsistências do sistema e a integração das funcionalidades.

Também houve uma coleta de feedback em termos gerais da aplicação e o que os usuários mais notaram durante sua utilização. Os pontos negativos foram focados na apresentação do frontend, destacando-se uma falta de detalhes e instruções em alguns pontos, como por exemplo qual o tipo de arquivo esperado e o tamanho máximo ao se cadastrar em uma vaga, e a falta de uma boa responsividade pelos componentes em diferentes resoluções. A partir desses resultados é possível então traçar os próximos passos para a evolução da aplicação, com um foco na plataforma web, refatorando os componentes e enriquecendo a plataforma com mais detalhes de utilização para o usuário nas páginas.

Eu acho que gostaria de usar esse sistema com frequência.

6 respostas

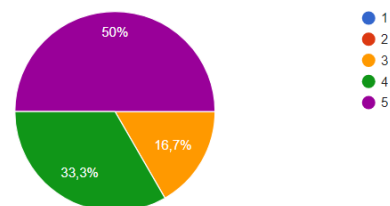


Figura 13: Resultados da primeira afirmação.

Eu acho o sistema desnecessariamente complexo.

6 respostas

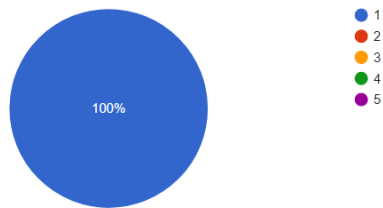


Figura 14: Resultados da segunda afirmação.

Eu achei o sistema fácil de usar.

6 respostas

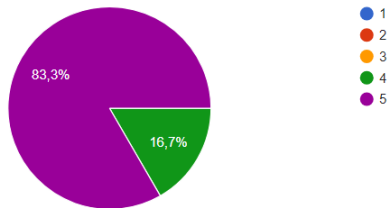


Figura 15: Resultados da terceira afirmação.

Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.

6 respostas

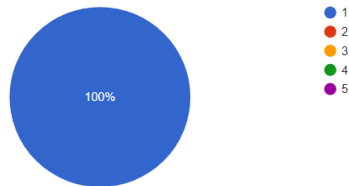


Figura 16: Resultados da quarta afirmação.

Eu acho que as várias funções do sistema estão muito bem integradas.

6 respostas

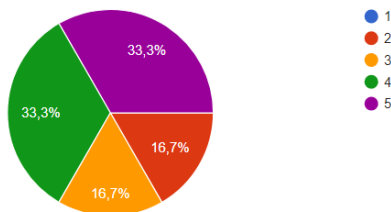


Figura 17: Resultados da quinta afirmação.

Eu acho que o sistema apresenta muita inconsistência.

6 respostas

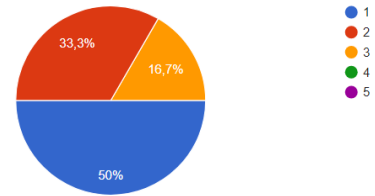


Figura 18: Resultados da sexta afirmação.

Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.

6 respostas

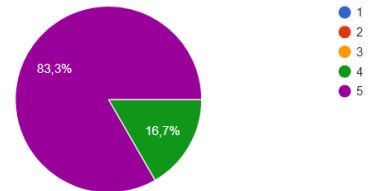


Figura 19: Resultados da sétima afirmação.

Eu achei o sistema atrapalhado de usar.

6 respostas

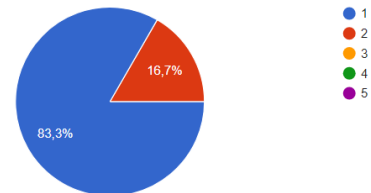


Figura 20: Resultados da oitava afirmação.

Eu me senti confiante ao usar o sistema.

6 respostas

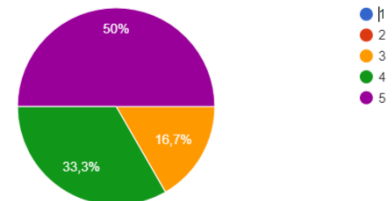


Figura 21: Resultados da nona afirmação.

Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

6 respostas

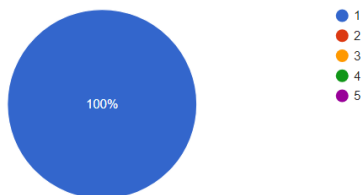


Figura 22: Resultados da décima afirmação.

Eu considero importante saber o máximo de informações possíveis a respeito da vaga

6 respostas

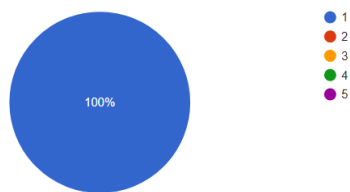


Figura 23: Resultados da décima primeira afirmação.

5. EXPERIÊNCIA

O problema foi escolhido depois de passar por uma busca por oportunidades no mercado privado como desenvolvedor de software, porém percebendo como era frustrante a falta de transparência que começava desde a descrição do anúncio até mesmo durante o processo de seleção em si, tive a ideia da plataforma com ênfase na transparência do processo.

Durante o processo de desenvolvimento pude aplicar o conhecimento e técnica que foi adquirido ao longo da graduação, além de desenvolver ainda mais as habilidades em programação resultando em um agregamento profissional, principalmente no processo de criação do backlog e refinamento das stories propostas.

Foram escolhidas as linguagens com o motivo de continuar a me aprofundar tanto em Spring quanto em React, linguagens modernas e que contam com uma ampla rede de recursos e referências.

5.1 Desafios e limitações

Foram encontrados desafios durante todo o projeto que são padrões no ambiente de desenvolvimento de software. Desde a decisão de como modelar a arquitetura, a implementação do código em si, principalmente no contato com React no qual nunca havia me aprofundado antes, e a escolha de onde realizar o deploy da aplicação.

Dentre os principais destaca-se o desenvolvimento de uma plataforma web, onde não somente foi necessário pensar nos componentes lógicos da aplicação, mas também em como proporcionar uma experiência fluida e agradável para o usuário. Também foi difícil escolher uma plataforma que desse suporte às aplicações desenvolvidas. No final foi optado por realizar

manualmente em uma VM na AWS, foram criadas duas instâncias e então em cada uma está um container rodando a aplicação, em um container está o frontend e no outro o backend juntamente com o banco de dados. Foi um limitador também a necessidade de serviços pagos para algumas funcionalidades, como o envio de email, que por causa disso não puderam ser implementadas nessa primeira versão do aplicativo.

6. TRABALHOS FUTUROS

Muitos pontos ainda podem ser melhorados na aplicação, como a interface que atualmente não está contando com a melhor experiência para o usuário possível. Será refatorada futuramente para deixá-la mais intuitiva e com uma responsividade para maior variedade de resoluções possíveis, além da adição de mais informações para melhorar a experiência do usuário nas páginas de candidatura para vagas.

Em termos funcionais a aplicação ainda pode ser expandida, contando com: a introdução de cadastro de recrutadores das empresas, onde poderão gerenciar o processo seletivos, melhorias no algoritmo de ranqueamento para obter melhores resultados e um histórico onde as empresas poderão manter os funcionários que gostariam de convidar para futuros processos. Será levado em conta também o feedback adquirido pelos usuários que viram inconsistências no sistema, que serão enumeradas e então entram no backlog para correções e alterações. Um serviço de envio de email também será implementado para recuperação de email e notificações para os usuários.

7. REFERÊNCIAS

- [1] <https://www.cnnbrasil.com.br/economia/procura-por-profissionais-de-tecnologia-cresce-671-durante-a-pandemia/>
- [2] <https://g1.globo.com/trabalho-e-carreira/noticia/2022/10/06/94percent-dos-candidatos-a-vagas-de-trabalho-querem-transparencia-nas-selecoes-mostra-levantamento.ghtml>
- [3] <https://g1.globo.com/jornal-hoje/noticia/2023/03/13/area-de-ti-deve-gerar-quase-420-mil-vagas-ate-2025-mas-faltam-profissionais.ghtml>
- [4] <https://octoverse.github.com/>
- [5] <https://docs.spring.io/spring-framework/reference/index.html>
- [6] <https://react.dev/learn>
- [7] <https://www.mongodb.com/docs/>
- [8] <https://reactrouter.com/en/main/start/overview>
- [9] <https://docs.docker.com/>
- [10] <https://docs.aws.amazon.com/ec2/>
- [11] <https://mui.com/material-ui/getting-started/>
- [12] <https://kubernetes.io/pt-br/>
- [13] <https://docs.docker.com/get-started/>
- [14] <https://medium.com/@adityamdk/tf-idf-implementation-in-java-f6c4d1d97e3b>
- [15] <https://learn.microsoft.com/pt-br/azure/architecture/best-practices/api-design>
- [16] <https://en.wikipedia.org/wiki/Tf-idf>