

**INSTITUTO
FEDERAL**
Brasília

Instituto Federal de Brasília

Campus Brasília

Curso Superior de Tecnologia em Sistemas para Internet

**TROCAVAGA: APLICAÇÃO WEB PARA ENCONTRAR VAGAS NAS ESCOLAS
PÚBLICAS DO DISTRITO FEDERAL**

Por

ÁDISON SOUZA GUALBERTO e JÚLIO CEZAR GONCALVES DIAS

Tecnólogo

BRASÍLIA

2023

Ádison Souza Gualberto e Júlio Cezar Goncalves Dias

**TROCAVAGA: APLICAÇÃO WEB PARA ENCONTRAR VAGAS NAS
ESCOLAS PÚBLICAS DO DISTRITO FEDERAL**

Trabalho de conclusão de curso apresentado à Banca de qualificação do Curso Superior de Tecnologia em Sistemas de Internet do Instituto Federal de Brasília – *Campus* Brasília, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas de Internet.

Orientador: Prof. Me. Fábio Ferraz Fernandez

Brasília

2023

D541 Dias, Júlio Cezar Gonçalves .

Trocavaga: aplicação web para encontrar vagas nas escolas públicas do Distrito Federal / Júlio Cezar Gonçalves Dias ; orientação Fábio Ferraz Fernandez. — Brasília, DF: 2023.
55 f. : il. ; 30 cm.

Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet)
— Instituto Federal de Brasília, Campus Brasília, Brasília, DF, 2023.
Orientador(a): Fábio Ferraz Fernandez.

1. Transfêrência escolar. 2. Permuta de vagas escolares. 3. Aplicação web. I. Fernandez, Fábio Ferraz, orient. II. Instituto Federal de Brasília. III. Título.

CDU: 004.738.5

RESUMO

DIAS, Júlio Cezar Gonçalves. GUALBERTO, Ádison Souza. TROCAVAGA: aplicação web para encontrar vagas nas escolas públicas. 2023. 56f. Trabalho de Conclusão de Curso (Graduação) - Tecnologia em Sistemas para Internet. Instituto Federal de Brasília - Campus Brasília. Brasília/DF, 2023.

A transferência escolar, procedimento que permite o deslocamento do estudante de uma unidade escolar para outra, garante aos estudantes e familiares a escolha de uma unidade que melhor atenda às necessidades pessoais, podendo impactar diretamente na continuidade dos estudos e na qualidade de vida, sobretudo quando o aluno pode estudar próximo a sua residência. No entanto, no âmbito do Distrito Federal, constatou-se grande insatisfação por parte dos usuários da rede pública de ensino, quanto à disponibilidade de vagas para transferência, visto que só é possível realizar o procedimento se houver a existência prévia de vaga para reserva na escola desejada. Diante disso, o objetivo do trabalho foi construir uma aplicação web - Trocavaga - que viabilizasse a transferência escolar independentemente da reserva de vagas e do calendário escolar. No desenvolvimento utilizamos, entre outras, a linguagem de programação JavaScript por meio do Node.js e Handlebars.js para a aplicação web com as funcionalidades de cadastro, login, busca e exibição dos dados provenientes do sistema de gerenciamento de dados relacional - SQLite e autenticação por meio do Passport.js, garantindo a segurança dos dados e o controle de acesso. Como resultado, implementamos uma aplicação que permite ao responsável cadastrar dados de contato, informações da vaga que ele ocupa e da vaga desejada e publicá-los para que os interessados em permuta possam entrar em contato, além de permitir que o responsável faça busca por outras vagas..

Palavras-chave: Transferência escolar pública. Permuta de Vagas escolares, Aplicação web.

ABSTRACT

DIAS, Júlio Cezar Gonçalves. GUALBERTO, Ádison Souza. TROCAVAGA: aplicação web para encontrar vagas nas escolas públicas. 2023. 56f. Trabalho de Conclusão de Curso (Graduação) - Tecnologia em Sistemas para Internet. Instituto Federal de Brasília - Campus Brasília. Brasília/DF, 2023.

The school transfer, a process that enables students to move from one school unit to another, ensures that students and their families can choose a unit that best meets their personal needs. This choice can directly impact the continuity of studies and quality of life, especially when the student can study near their residence. However, within the Federal District, significant dissatisfaction has been observed among users of the public school system regarding the availability of transfer slots. This is because the transfer process is only possible if there is a prior vacancy reservation in the desired school. In light of this, the objective of the project was to develop a web application - Trocavaga - that would facilitate school transfers independent of vacancy reservations and the school calendar. In the development process, we utilized programming languages such as JavaScript through Node.js and Handlebars.js for the web application's features, including registration, login, search, and display of data from the relational data management system - SQLite. Authentication was implemented through Passport.js to ensure data security and access control. As a result, we implemented an application that allows the user to register contact information, details about the current position, and the desired position. Users can then publish this information so that those interested in transferring can get in touch. Additionally, the application enables users to search for other available transfer opportunities.

Keywords: School transfer. Exchange. Public school. Web application.

LISTA DE FIGURAS

Figura 2.1 <i>Censo Escolar 2021</i>	10
Figura 2.2 Número de Escolas	11
Figura 2.3 Evolução de transferências	12
Figura 2.4 Tela de busca de vagas GIER	15
Figura 2.5 Tela de busca de vagas Anajusfe	16
Figura 2.6 Tela de Transferências SED	17
Figura 3.1 Casos de uso	31
Figura 3.2 Modelo Conceitual.....	31
Figura 3.3 Modelo Lógico	32
Figura 4.1 Arquitetura implementada	34
Figura 4.2 Página inicial da Aplicação.....	36
Figura 4.3 Cadastrar Vaga	37
Figura 4.4 Tela de Login.....	38
Figura 4.5 Vagas Disponíveis.....	38
Figura 4.6 Ver vaga.....	39
Figura 4.7 Perfil	40
Figura 4.8 Estrutura do projeto.....	41
Figura 4.9 Package.json.....	41
Figura 4.10 App.js	42
Figura 4.11 Código Barra de pesquisa	43
Figura 4.12 Trocavagas.js	44
Figura 4.13 trocavaga.js	45
Figura 4.14 Connection.js.....	46
Figura 4.15 app.db	46
Figura 4.16 index.handlebars	47
Figura 4.17 add.handlebars.....	48
Figura 4.18 view.handlebars	49

LISTA DE TABELAS

Tabela 2.1 Comparação de Sistemas Semelhantes	18
Tabela 3.1 Requisitos Funcionais: RF01 - Manter usuário.....	27
Tabela 3.2 Requisitos Funcionais: RF02 - Manter vagas.....	27
Tabela 3.3 Requisitos Funcionais: RF03 - Exibir resultado da busca	28
Tabela 3.4 Requisitos Funcionais: RF04 - Enviar notificação	28
Tabela 3.5 Requisitos Funcionais: RF05 - Categorizar os dados.....	28
Tabela 3.6 Requisitos Funcionais: RF06 - Exibir resultados	29
Tabela 3.7 Requisitos Não-Funcionais: RNF01 - Usabilidade.....	29
Tabela 3.8 Requisitos Não-Funcionais: RNF02 - Disponibilidade	29
Tabela 3.9 Requisitos Não-Funcionais: RNF03 - Portabilidade	30
Tabela 3.10 Requisitos Não-Funcionais: RNF04 - Segurança	30

SUMÁRIO

1.	INTRODUÇÃO.....	8
1.1.	Objetivo geral	8
1.2.	Objetivos específicos	9
2.	CONCEITOS E REVISÃO DE LITERATURA.....	10
2.1.	O cenário da educação no Distrito Federal.....	10
2.2.	Demanda de transferências realizadas durante o ano letivo.....	11
2.3.	Procedimento para transferência entre escolas do Distrito Federal	13
2.4.	Sistemas semelhantes	14
2.4.1.	Gier	14
2.4.2.	Anajusfe	15
2.4.3.	SED (Secretaria Escolar Digital)	16
2.4.4.	Comparação entre sistemas semelhantes	17
2.5.	Tecnologias utilizadas.....	18
2.5.1.	Visual Studio Code.....	18
2.5.2.	HTML (HyperText Markup Language).....	18
2.5.3.	CSS (Cascading Style Sheets).....	19
2.5.4.	Javascript	19
2.5.5.	Vercel	20
2.5.6.	Firebase	20
2.5.7.	Node.js	20
2.5.8.	Express	21
2.5.9.	SQLite	21
2.5.10.	Handlebars	22
2.5.11.	Passport.....	22
3.	METODOLOGIA.....	23
3.1.	Idealização	23
3.2.	Caracterização do projeto	23

3.3.	Requisitos de usuário.....	25
3.4.	Requisitos funcionais	27
3.5.	Requisitos não funcionais	29
3.6.	Diagrama de Casos de Uso Macro	30
3.7.	Modelo do banco de dados	31
4.	ANÁLISE E DISCUSSÃO DOS RESULTADOS	33
4.1.	Relatos sobre o desenvolvimento e análise.....	33
4.2.	Apresentação do <i>frontend</i> do sistema.....	35
4.2.1.	Página inicial	35
4.2.2.	Cadastrar Vaga	36
4.2.3.	Página de login.....	37
4.2.4.	Vagas Disponíveis.....	38
4.2.5.	Detalhamento da vaga	38
4.2.6.	Perfil	39
4.3.	Apresentação do <i>frontend</i> do sistema.....	40
4.3.1.	Estrutura do Projeto	40
4.3.2.	Package.json.....	41
4.3.3.	App.js	42
4.3.4.	Trocavagas.js	44
4.3.5.	Trocavaga.js.....	44
4.3.6.	Connection.js.....	45
4.3.7.	app.db	46
4.3.8.	Handlebars.....	47
5.	CONCLUSÃO E TRABALHOS FUTUROS.....	50
	REFERÊNCIAS BIBLIOGRÁFICAS	52

1. INTRODUÇÃO

Segundo o artigo 205 da Constituição Federal brasileira de 1988 (Brasil, 1988), a educação é direito de todos e dever do Estado e da família, devendo ser promovida e incentivada com a colaboração da sociedade. No entanto, no âmbito do Distrito Federal, observa-se que pais e alunos têm encontrado um obstáculo no acesso à educação, no que tange à escolha da escola mais próxima de suas residências, sobretudo quando precisam solicitar a transferência escolar.

A Carta de Serviços da Secretaria de Estado de Educação do Distrito Federal, publicada em 2022, prevê duas formas de solicitar a mudança de Unidade Escolar na rede pública. A primeira é o remanejamento escolar, que deve ser feito para o ano letivo subsequente à solicitação. A segunda é a transferência escolar, que pode ser solicitada no decurso do ano letivo, porém está condicionada à reserva prévia da vaga na escola de interesse. Enquanto aquela exige dos pais e alunos o planejamento com antecedência - para o ano seguinte, esta os condiciona à reserva de uma vaga que nem sempre está disponível após iniciado o ano letivo. Consequentemente, quando os usuários do ensino público têm a necessidade iminente de mudança de residência, local de trabalho ou outra justificativa para a troca não programada de escola, eles encontram, frequentemente, dificuldade para obterem a vaga desejada.

Diante disso, a proposta deste trabalho é desenvolver uma solução que maximize o acesso às vagas para transferência, conectando usuários da rede pública de ensino do Distrito Federal que desejem mudar de escola e, desta forma, viabilizar a permuta entre eles.

1.1. Objetivo geral

Desenvolver uma aplicação web que ofereça cadastro e publicação da vaga ocupada, bem como da vaga desejada pelos usuários da rede pública de ensino do Distrito Federal, além de permitir a busca detalhada dessas vagas por outros usuários cadastrados e viabilizar o contato entre eles.

1.2. Objetivos específicos

- Desenvolver uma aplicação web responsiva que permita o cadastro, armazenamento e gerenciamento das informações dos usuários e das vagas;
- Implementar funcionalidade que permita o cruzamento dos dados armazenados para encontrar convergência de interesses e notificar automaticamente os usuários quando a vaga para permuta for encontrada;
- Construir um ambiente de acesso restrito, condicionado à autenticação de usuário, para busca categorizada pelas vagas e para exibição dos dados de contato do responsável pela publicação.

2. CONCEITOS E REVISÃO DE LITERATURA

Neste capítulo serão apresentados conceitos referentes à transferência escolar, o cenário da educação do Distrito Federal e aspectos legais, além de sistemas semelhantes ao proposto neste trabalho.

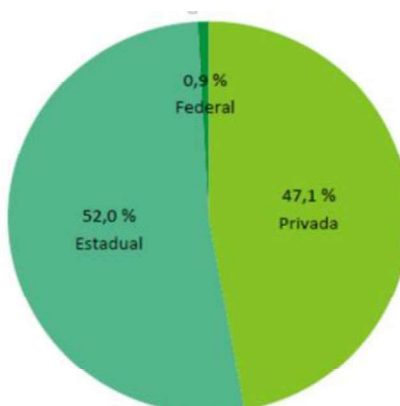
2.1. O cenário da educação no Distrito Federal

O Brasil oferece garantias de diversos direitos aos seus cidadãos, dentre eles está o direito social à educação, através de sua Carta Magna, a Constituição Federal de 1988 (CF/88). Esta prevê em seu artigo 205 que a Educação é um direito reservado a todos, cabendo ao Estado e à Família o garantir, visando o pleno desenvolvimento do cidadão, qualificando-o para o trabalho.

A Constituição Federal ainda, em seu artigo 23, define que é de competência comum entre a União, os Estados, o Distrito Federal e os Municípios o dever de proporcionar os meios necessários para o acesso à educação.

Segundo o Censo Escolar da Educação Básica de 2021, figura 2.1, no que se refere à realidade do Distrito Federal, este possui um total de 1.252 escolas de educação básica, onde que desse total, 52,0% são da rede distrital, 47,1% da rede privada e 0,9% da rede federal, conforme Figura 2.1.

Figura 2.1 Censo Escolar 2021

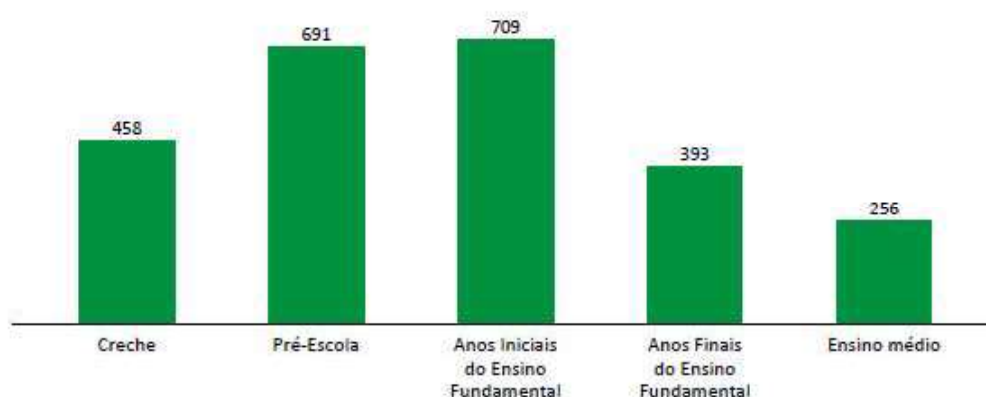


Fonte: Deed/Inep – Censo Escolar

É possível verificar também que, dentre todas as etapas de ensino, as que contemplam maior número de escolas ofertantes são a etapa da pré-escola com 691

e os anos iniciais do ensino com 709 escolas (Figura 2.2). Ou seja, há uma crescente de escolas ofertantes desde a etapa da Creche, passando para a Pré-escola e finalizando nos anos iniciais do Ensino fundamental, e quando verificado o número de oferta para os Anos finais do Ensino Fundamental (393) e do Ensino Médio (256) há acentuado declínio na quantidade de escolas ofertantes.

Figura 2.2 Número de Escolas



Fonte: Deed/Inep – Censo Escolar

Ainda conforme o Censo Escolar da Educação Básica de 2021, no Distrito Federal, foi registrado um total de 644.293 matrículas na educação básica, distribuídas entre as seguintes etapas ou modalidades: Educação Infantil (97.817), Ensino Fundamental (369.128), Ensino Médio (116.843), Educação de Jovens e Adultos – EJA (36.946), Educação Profissional (28.380) e Educação Especial (17.518). A soma de cada uma das etapas ou modalidades não representa necessariamente o total de matrículas apresentado inicialmente, uma vez que algumas etapas ou modalidades inserem-se umas dentro das outras.

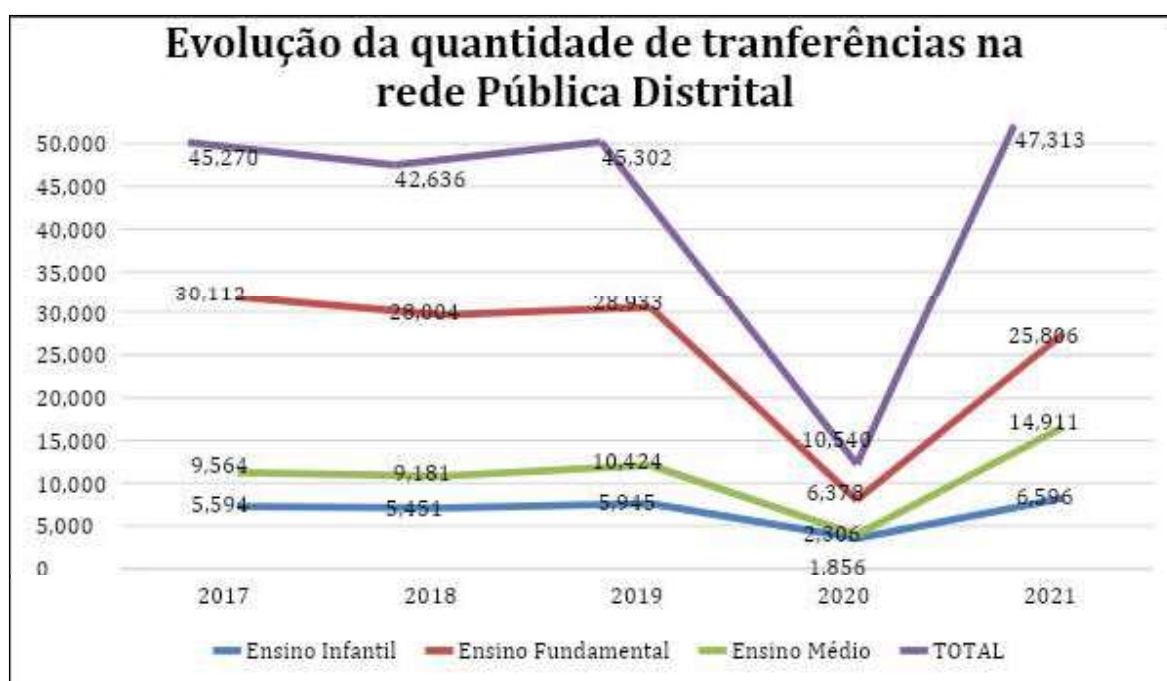
2.2. Demanda de transferências realizadas durante o ano letivo

Considerando o total de inscritos na educação Básica do Distrito Federal, também verificou-se a necessidade de levantar o quantitativo de transferências realizadas durante o ano de 2021. Esta informação não estava disponível publicamente, desta forma, houve a necessidade da solicitação por meio do portal Participa DF, plataforma de participação social do Governo do Distrito Federal, que

oferece o sistema e-Sic (Acesso à Informação), onde é possível solicitar dados de serviços públicos de interesse social, por meio da Lei nº 12.527, de 18 de novembro de 2011 (Lei de Acesso à Informação – LAI).

A solicitação foi prontamente atendida pela Secretaria de Estado da Educação do Distrito Federal, protocolado por meio do processo SEI-GDF (Sistema Eletrônico de Informações do Governo do Distrito Federal) de número: 00080-00107612/2023-39, fornecendo a quantidade de transferências realizadas dos anos de 2017 a 2021, conforme a Figura 2.3.

Figura 2.3 Evolução de transferências



Fonte: Elaborado pelos autores

É possível depreender dos dados que, no decorrer dos anos, existem variações discretas na quantidade de transferências realizadas em cada uma das modalidades de ensino e também na quantidade total. Essa tendência só foi excetuada em 2020, ano do ápice da pandemia de COVID-19, doença infecciosa causada pelo Coronavírus (Sars-CoV-2), onde que a rede de educação distrital passou por várias dificuldades: atrasos no calendário escolar e ensino à distância, alterando drasticamente as transferências realizadas naquele ano.

Considerando o ano de 2021, com a sinalização da disponibilização das vacinas e retração dos casos de contaminação da COVID-19, os dados retornaram a apresentar um padrão de quantidade de transferências e até mesmo mostrando um moderado aumento na quantidade total, evidenciando ainda mais a uma real necessidade e demanda por ferramentas que auxiliem os pais e alunos na efetivação da transferência de acordo com suas necessidades.

2.3. Procedimento para transferência entre escolas do Distrito Federal

Conforme define a Carta de Serviços da Secretaria de Estado de Educação publicada em 2022, a transferência escolar é o procedimento que permite o deslocamento do estudante de uma Unidade Escolar para outra, e pode ser realizada em todas as etapas da Educação, desde o ensino pré-escolar até o ensino médio. O Regimento da Rede Pública de Ensino do Distrito Federal (2019) no artigo Art. 265, parágrafo único, classifica a transferência escolar em duas divisões:

“I- Transferência Interna: refere-se à movimentação de estudantes dentro da Rede Pública de Ensino do Distrito Federal;

II- Transferência Externa: refere-se à movimentação de estudantes da Rede Pública de Ensino do Distrito Federal para instituições educacionais da rede privada do Distrito Federal ou para outros Estados da Federação e/ou países.”

Vale destacar que o escopo deste projeto limita-se às transferências internas, ou seja, transferências com origem e destino pertencentes à Secretaria de Educação do Distrito Federal. Nessa modalidade, há um dispositivo que busca viabilizar a movimentação e alocação dos estudantes na rede de ensino no âmbito do Distrito Federal. Trata-se do Regimento da Rede Pública de Ensino do Distrito Federal, no Art. 259 que prevê: *“A transferência do estudante de uma para outra instituição educacional ou unidade escolar far-se-á, preferencialmente, nos períodos de férias e de recessos escolares”*.

Infere-se do supracitado artigo que a priorização por transferências nos períodos de férias e recesso busca facilitar à gestão das Unidades Escolares quanto ao encaminhamento dos estudantes a outras escolas, e recebimento por transferência de novos alunos, uma vez que, de acordo com a Carta de Serviços ao Cidadão (2022), o estudante (ou responsável) deverá previamente reservar a vaga na escola de interesse para depois solicitar a transferência e preencher um instrumento próprio

dirigido ao Diretor da unidade escolar por membro da família e/ou responsável legal ou pelo estudante, se maior de idade. Ou seja, em caso de inexistência de vaga na escola desejada, a Secretaria de Educação não prevê uma alternativa para estudantes que tenham interesse em trocar de vaga entre si. Desta forma, a transferência escolar só poderia ser efetiva se eles buscarem as vagas em períodos pré-determinados. Na prática, diante da necessidade de mudança antes do período das férias ou recesso, o aluno terá menor probabilidade de obtê-la.

Nesse contexto, a proposta deste trabalho é desenvolver uma solução que atenda aos estudantes e responsáveis que desejem trocar de instituição de ensino, independente do calendário letivo proposto ou de existência prévia de vaga. Para isso, viabilizaremos a conexão dos interessados na permuta, ficando a encargo destes a formalização das transferências em instrumento próprio conforme previsto na norma.

2.4. Sistemas semelhantes

Nesta seção são apresentadas aplicações web com características similares à da proposta elaborada neste trabalho. Dois deles foram desenvolvidos para realização de permuta entre servidores e o outro para transferência escolar de alunos.

2.4.1. *Gier*

Desenvolvido pela Prefeitura de Guarulhos - São Paulo em 2023, esse sistema automatizou o processo de permuta entre profissionais da Secretaria de Educação que desejavam atuar em uma nova escola. A plataforma possui um ambiente restrito e busca automaticamente os dados do servidor após o fornecimento das respectivas credenciais. Apesar de ser acessado por meio do navegador, o sistema não é responsivo e restringe o acesso de dispositivos mobile.

Para participar do programa de permutas, é necessário que o servidor confirme sua inscrição. Desta forma, a vaga atualmente ocupada por ele ficará disponível para busca por outros servidores. A partir daí ele poderá utilizar a ferramenta de busca, localizar escolas que deseja e manifestar interesse, de acordo com a tela de busca rápida do GIER (Figura 2.4).

Figura 2.4 Tela de busca de vagas GIER

Escola	Tipo da Vaga	Período	Classe	+ Informações	Interessados	Status	Ações
EPG ZELIA GATTAI	SEDE	MANHÃ	-	1	1		Ações
EPG DORIVAL CAYMMI	SEDE	MANHÃ	EII D	1	4		Ações
EPG CAROLINA MARIA DE JESUS	SEDE	TARDE	EII E	1	2		Ações

Fonte: Portal Educação de Guarulhos (2023)

2.4.2. Anajusfe

A Associação Nacional dos Servidores do Judiciário Federal (Anajusfe) desenvolveu uma plataforma muito prática e útil a mais 100 mil servidores que compõem o Poder Judiciário Federal (ANAJUSFE, 2023). Uma de suas principais funcionalidades é o sistema de permuta e redistribuição que permite a troca de vagas entre ocupantes dos mais diversos cargos, especialidades e carreiras em todo território nacional.

De acordo com a tela de busca de permuta e redistribuição da Anajusfe (Figura 2.5), para encontrar vagas nesse sistema, não é necessário criar cadastro ou fazer login, basta marcar os campos necessários à busca como “Onde estou” e “Para onde quero ir”. No entanto, para divulgar uma vaga é obrigatório o preenchimento dos dados pessoais e meios para contato (ANAJUSFE, 2023).

Figura 2.5 Tela de busca de vagas Anajusfe

Quero resultados para
☐ Permuta ☐ Redistribuição

Onde estou

Cargo ▾

Especialidade ▾

Matrícula

Justiça ▾

Lotação

Selecione um Estado ▾

Cidade

Para onde quero ir
Estado

Marcar todos

Desmarcar todos

☐ AC ☐ AL ☐ AM ☐ AP ☐ BA ☐ CE ☐ DF ☐ ES ☐ GO
☐ MA ☐ MG ☐ MS ☐ MT ☐ PA ☐ PB ☐ PE ☐ PI ☐ PR
☐ RJ ☐ RN ☐ RR ☐ RO ☐ RS ☐ SC ☐ SE ☐ SP ☐ TO
☐ -

Orgão

Marcar todos

Desmarcar todos

☐ CETS ☐ JE ☐ JF ☐ JM ☐ JT ☐ TJ-DF ☐ OUTROS

BUSCAR

Fonte: Anajusfe (2023)

2.4.3. SED (Secretaria Escolar Digital)

Desenvolvido pela Secretaria de Educação do Estado de São Paulo, esse é um sistema criado para centralizar diversas operações inerentes à gestão da administração escolar, possuindo funcionalidades para professores, alunos e responsáveis (SED, 2023). Dentre as diversas ferramentas que esse robusto sistema disponibiliza (Figura 2.6), destaca-se a Inscrição de Deslocamento/Transferência online, cuja finalidade é transferir alunos com matrícula ativa na rede pública estadual que queiram mudar de escola por iniciativa própria, a despeito de alteração de endereço (SED, 2023).

Essa funcionalidade é restrita ao responsável pelo aluno, que deverá indicar o tipo de transferência e o motivo. Após isso, é possível realizar a busca em uma lista

de escolas em todos os municípios do estado de São Paulo, ou em um mapa com as escolas mais próximas de sua localização (SED, 2023).

Figura 2.6 Tela de Transferências SED

Dados da Nova Inscrição:

Tipo Movimentação: INTENÇÃO DE TRANSFERÊNCIA ▾

Tipo Transferencia: INSCRIÇÃO SEM ALTERAÇÃO DE ENDEREÇO DO

Motivo: INTERESSE DO ALUNO ▾

Pesquisa Por: Listagem de Escolas ▾

Município: Listagem de Escolas
Escolas mais Próximas ▾

Escola: SELECIONE... ▾

Unidade: SELECIONE... ▾

Interesse pelo turno integral? ☐

Necessidade pelo período Noturno? ☐

Fonte: SED (2023)

Esse sistema diferencia-se do aplicativo proposto neste projeto, pois só informatiza a reserva de vagas já existentes, porém não possibilita ao responsável ou estudante a busca por vagas de alunos que tenham interesse na troca.

2.4.4. Comparação entre sistemas semelhantes

Por meio da tabela comparativa 2.1 podemos evidenciar as principais diferenças e similaridades entre os sistemas elencados nesta seção. Destaca-se que não foi encontrado, após ampla pesquisa, nenhum sistema com a mesma finalidade proposta neste trabalho. Por conseguinte, nenhum dos sistemas apresentou as funcionalidades necessárias à permuta entre estudantes de escola pública, e nem sequer são direcionados ao Distrito Federal.

Tabela 2.1 Comparação de Sistemas Semelhantes

Características	GIER	Anajusfe	SED	TrocaVaga
Transferência de estudantes	✗	✗	✓	✓
Permuta entre estudantes	✗	✗	✗	✓
Autenticação para acesso às vagas	✓	✗	✓	✓
Aplicável à educação do DF	✗	✗	✗	✓

Fonte: Elaborado pelos autores

2.5. Tecnologias utilizadas

Nesta seção são apresentadas as ferramentas e tecnologias utilizadas para o desenvolvimento do projeto.

2.5.1. Visual Studio Code

O Visual Studio Code é um editor de texto multiplataforma, gratuito que oferece um conjunto de funcionalidades e é direcionado a programadores que buscam um IDE (ambiente de desenvolvimento integrado) simples e completo, disponível para Windows, macOS e Linux.

O editor possui entre suas principais funcionalidades: Git Integrado, Meet IntelliSense (ajuda de preenchimento de código), aplicação de Debug, suporte para várias linguagens de programação e extensões. Fornece suporte interno para JavaScript, TypeScript e Node.js e com um ecossistema avançado de extensões para outras linguagens e runtimes (como C++, C#, Java, Python, PHP, Go, .NET). (MICROSOFT, 2023).

2.5.2. HTML (HyperText Markup Language)

Por se tratar de um elemento de construção básico para navegadores web, o HTML (Linguagem de Marcação de Hipertexto) será a linguagem de marcação que usaremos para estruturar o conteúdo de nossas páginas web. Ele é composto por diversos de elementos que podem delimitar ou agrupar diferentes partes do conteúdo, conforme a especificidade da proposta (MOZILLA, 2023).

2.5.3. CSS (*Cascading Style Sheets*)

Para tornar o conteúdo de nossas páginas mais atraente e facilitar a definição dos grupos de estilos, será utilizado o CSS (Cascading Style Sheets - Folhas de Estilo em Cascata) nos nossos códigos. O CSS permite estilizar um documento de texto simples, possibilitando a alteração das cores e estilo dos links. Além disso, ele também é utilizado para criar layouts, adicionar efeitos e animações (MOZILLA, 2023).

2.5.4. Javascript

Será empregada a linguagem de programação Javascript como elemento fundamental no desenvolvimento do front-end e back-end do projeto.

Esta linguagem foi criada pela Netscape em parceria à Sun Microsystems em 1995, o JavaScript é uma linguagem multiparadigma de tipagem dinâmica, fraca e implícita. É considerada a linguagem de programação da Web, sendo moderna, robusta e eficiente (SILVA, 2010).

O Javascript é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem browser, tais como node.js, Apache CouchDB e Adobe Acrobat (MOZILLA, 2023).

2.5.4.1. ReactJS

A interface gráfica será concebida através da utilização do ReactJS, biblioteca JavaScript que permite a criação de interfaces de usuário a partir de peças individuais chamadas componentes (REACT, 2023).

Foi criada por desenvolvedores do Facebook (HÁMORI, 2021), e é uma biblioteca que tem como atributos ser rápida, escalável e flexível, além de possuir uma comunidade de desenvolvedores robusta, e encontra-se ainda em acelerada expansão (TEAM, 2021).

2.5.4.2. NextJS

O NextJS é um framework para o ReactJS. Ele permite que sejam criados aplicativos da Web de pilha completa, estendendo os recursos mais recentes

do React e integrando poderosas ferramentas JavaScript baseadas em Rust para compilações mais rápidas (NEXTJS, 2023).

Ainda pode-se usar o React para criar a interface do usuário e adotar gradualmente os recursos do Next.js para resolver requisitos comuns de aplicativos, como roteamento, busca de dados, integrações, entre outros (NEXTJS, 2023).

2.5.5. Vercel

Plataforma inicialmente criada sob a denominação Zeit, em 2015, e renomeada para Vercel em 2020, esta ferramenta atua como uma nuvem front-end, oferecendo estruturas, fluxos de trabalho e infraestrutura para criar aplicações Web, de alto desempenho, com tempo e custos reduzidos, além de trabalhar juntamente com o Next.JS para maximizar ainda mais sua performance (VERCEL, 2023).

Assim, ela permite uma integração facilitada com o Back-end, conectando a página a qualquer fonte de dados, com uma API (Interface de Programação de Aplicação), funcionando em ambiente de desenvolvimento comum (VERCEL, 2023).

2.5.6. Firebase

É uma plataforma, desenvolvida pelo Google, que visa disponibilizar ferramentas para as funcionalidades mais comuns de uma aplicação, como: autenticação de usuário, armazenamento, escalabilidade, serviço de notificações, entre outros (FIREBASE, 2023).

O Firebase é uma plataforma de desenvolvimento que executa por dentro da Google Cloud e permite que os desenvolvedores construam aplicações para dispositivos móveis e web sem precisarem de seus próprios servidores, visando a criação de apps de alta qualidade, de maneira ágil e simples. A plataforma inclui vários recursos altamente integrados que podem ser combinados, incluindo a análise e um back-end para dispositivos móveis, além de ferramentas de monetização e aprimoramento de aplicativos (FIREBASE, 2023).

2.5.7. Node.js

Uma popular ferramenta de código aberto, o Node.js foi especialmente projetado para ser eficiente e escalável em ambientes de servidor, permitindo que os

desenvolvedores usem JavaScript para escrever aplicativos do lado do servidor. Ele é um ambiente de tempo de execução JavaScript e plataforma cruzada. (Foundation, 2023).

Vale destacar a vantagem de utilizar essa plataforma, visto que ela adota uma arquitetura não bloqueante, resultando em menor uso de memória e uma utilização mais eficiente do poder de processamento dos servidores quando comparada a outras plataformas que utilizam linguagens distintas (PEREIRA, 2014, p.1)

2.5.8. Express

Lançado em 2010, o Express é o framework para Node.js mais popular da atualidade. Ele oferece soluções para gerenciar requisições HTTP, integração para inserir dados em templates, definir configurações comuns da aplicação web e adicionar novos processos de requisição. Apesar de sua abordagem minimalista, os desenvolvedores têm a liberdade de criar pacotes de *middleware* específicos para resolver problemas no desenvolvimento de aplicações, como cookies, sessões, login de usuários e outros (MOZILLA, 2023).

2.5.9. SQLite

O sistema de gerenciamento de banco de dados SQLite é uma biblioteca em linguagem C, que implementa um mecanismo de banco de dados relacional, pequeno, veloz, independente, altamente confiável e abrangente, sendo reconhecido como o mecanismo de banco de dados mais amplamente utilizado globalmente (SQLite, 2023).

Esse SGBD (Sistema de Gerenciamento de Banco de Dados) possui código-fonte de domínio público, podendo ser utilizado gratuitamente e para qualquer finalidade. O formato de arquivo SQLite é estável, funciona em diversas plataformas e é compatível com versões anteriores. Os arquivos de banco de dados SQLite são frequentemente utilizados como contêineres para transferir conteúdo entre sistemas e como formato de arquivo de arquivamento de dados a longo prazo (SQLite, 2023).

2.5.10. Handlebars

Este é um mecanismo de template para JavaScript que simplifica a geração de HTML dinâmico. O Handlebars permite a criação de modelos que são uma combinação de HTML, texto e expressões incorporadas (Handlebars.js, 2023).

Uma expressão no Handlebars consiste em um conteúdo delimitado por chaves duplas. Ao executar o modelo, essas expressões são trocadas pelos valores provenientes de um objeto de entrada (Handlebars.js, 2023).

2.5.11. Passport

Visando a garantir acesso restrito às principais funcionalidades da aplicação e com isso dar maior proteção aos dados pessoais dos usuários será utilizada a tecnologia Passport (também conhecido como Passport.js) que é um *middleware* para autenticação em aplicativos Node.js (Passport, 2023).

Este recurso simplifica o processo de autenticação, fornecendo uma estrutura flexível e modular que pode ser facilmente integrada em diferentes aplicativos web. O Passport suporta vários métodos de autenticação, como autenticação local (usando nome de usuário e senha), autenticação via mídias sociais (por exemplo, usando contas do Google, Facebook, Twitter, etc.) entre outros (Passport, 2023).

3. METODOLOGIA

Neste capítulo elucidaremos como foi concebida a ideia para a execução do projeto, a caracterização do objeto de estudo e quais tecnologias utilizadas para desenvolver a aplicação.

3.1. Idealização

A idealização do objeto do trabalho iniciou-se a partir da observação da demanda por transferências entre escolas públicas do Distrito Federal, verificadas por meio de um grupo do Facebook, onde pessoas da Região administrativa de São Sebastião anunciavam, reiteradamente, as informações de vagas no intuito de encontrar uma permuta. A partir daí, realizamos a busca na mesma rede social pelo termo “troco vaga escola”, verificamos a existência de publicações semelhantes em diversos outros grupos, com os mais diferentes temas em muitas regiões administrativas. No entanto, os dados dessas publicações não eram estruturados, as publicações não seguiam um padrão e estavam dispersas em várias páginas com temas aleatórios.

Em seguida, fomos em busca de plataformas que forneciam o mesmo tipo de serviço ou algum serviço similar e, após essa pesquisa, não encontramos nenhuma plataforma que disponibilizasse esse serviço de maneira padronizada, organizada e centralizada, dentro do Distrito Federal.

Devido à falta de ferramentas que disponibilizassem essa solução aos usuários do serviço público de ensino e por ser uma questão de relevância social, que envolvem questões de: permanência do aluno na escola, econômicas, benefícios na qualidade de vida advindas de estudar perto de sua moradia, entre outras, decidimos então construir uma aplicação web que atendesse a esse tipo de demanda para responsáveis e alunos no Distrito Federal.

3.2. Caracterização do projeto

A proposta baseia-se na criação de uma aplicação web que entregue usabilidade e simplicidade. Segundo Nielsen (1993), usabilidade é um requisito de qualidade de software que envolve a facilidade de uso e aprendizado de um sistema.

Essas características são fundamentais para o alcance de qualidade. Quanto mais usabilidade um sistema apresenta, mais simples e satisfatória é sua utilização.

Esta aplicação será baseada na arquitetura de 3 camadas de computação física e lógica, formada pelas camadas de apresentação (interface com o usuário), de aplicativo (processamento de dados) e de dados (armazenamento e gerenciamento de dados).

Assim, utilizaremos o editor de código-fonte Visual Studio Code de modo a facilitar integração do conjunto de linguagens Html/CSS/JavaScript e também a plataforma Vercel para a sustentação da camada de apresentação com o emprego de estruturas pré-moldadas e fluxos de trabalho facilitados auxiliado pelo NextJS, permitindo, através da interface com o usuário, a pesquisa de vagas para alunos do Ensino Infantil, Fundamental e Médio.

O Firebase será utilizado no processo de autenticação. Para que um novo usuário faça uma pesquisa no portal de vagas é preciso primeiramente que ele realize seu cadastro. Para isso se faz necessária a validação dos dados cadastrados para visualização da interface de acesso restrito.

A camada de dados será sustentada pelo Cloud Firestore, que é SGBD do Firebase. Portanto a plataforma será utilizada para a autenticação do usuário, para o armazenamento das informações cadastrais e para os serviços de notificações, quando das combinações automáticas entre destino requerido e vaga à disposição entre dois usuários.

Desta forma, o Vercel fará a integração entre front-end e back-end, com a utilização da biblioteca ReactJS para JavaScript pois será necessária a utilização de componentes tanto para possibilitar o cadastro do usuário, como para a pesquisa de vagas e, em complemento, utilizaremos o NextJS para a extensão dos recursos do ReactJS.

A realização do cadastro dar-se-á por meio do registro das seguintes informações: nome do responsável, CPF, e-mail, telefone (opcional), vaga de que dispõe e vaga que tem interesse. Após o cadastro o sistema fará uma varredura buscando a combinação dos seguintes requisitos: vaga à disposição e vaga pretendida com os usuários anteriormente cadastrados. Caso a busca encontre

correspondência, o sistema exibirá imediatamente um alerta com os dados da vaga correspondente. Também será enviado um e-mail para os respectivos usuários com os dados de contato referentes à combinação pertinente.

Caso não seja encontrada imediatamente uma correspondência pelo sistema, ou o usuário não tenha obtido sucesso ao entrar em contato com o ofertante da vaga, poderá ser realizada a busca por filtros, podendo ser feita tanto por localização (Região Administrativa) quanto por instituição de ensino (lista predefinida na aplicação). Também será possível pesquisar por duas formas: pela vaga de destino (local ou escola almejada) ou pela vaga de origem (vaga ocupada atualmente). Uma vez que for encontrada a vaga desejada, o sistema deve permitir que o usuário entre em contato com o portador da vaga de interesse, por meio do e-mail ou telefone (se existir) cadastrados para contato.

Após a comunicação e acordo entre as partes, fica a critério dos usuários combinarem o encontro dentro da secretaria da instituição de ensino para a formalização da troca de vagas conforme padrão e normatização estabelecida pela Secretaria de Estado de Educação do Distrito Federal.

Desta forma, o principal objetivo do sistema é facilitar e viabilizar a busca das vagas pretendidas pelo usuário do sistema público de ensino, por meio do anúncio de vagas e da disponibilização de canais para a comunicação entre usuários, possibilitando, desta forma, a transferência escolar com auxílio da permuta de vagas.

3.3. Requisitos de usuário

Os requisitos de usuário são inicialmente desenvolvidos em alto nível, de forma mais ampla. Sommerville (2011) afirma que, em geral, os requisitos de usuário são definidos em uma introdução à especificação de requisitos de sistema. A seguir elencamos esses requisitos (RU), de acordo com os objetivos específicos do projeto.

RU01 - O sistema deve disponibilizar uma tela para cadastro de usuários.

RU02 - O sistema deve verificar a data de nascimento do usuário, permitindo apenas o cadastro de maiores de idade.

RU03 - O sistema deve disponibilizar uma tela para realizar login.

RU04 - O sistema deve disponibilizar uma funcionalidade que permita a recuperação da senha de acesso do usuário.

RU05 - O sistema deve permitir que o usuário selecione em uma lista a região administrativa, nome da escola, ano e turno que ocupa atualmente.

RU06 - O sistema deve permitir que o usuário selecione em uma lista a região administrativa, nome da escola, ano e turno da vaga desejada.

RU07 - O sistema deve exibir o resultado da busca pela vaga na base de dados após envio dos dados, apresentando os dados para contato apenas da vaga com data de cadastro mais antiga.

RU08 - O sistema deve enviar um e-mail automático de confirmação, após localizar a vaga, para o usuário que buscou a vaga, apresentando os dados para contato do usuário ofertante da vaga.

RU09 - O sistema deve enviar um e-mail automático para o usuário ofertante da vaga, com os dados de contato do novo usuário que buscou pela respectiva vaga.

RU10 - O sistema deve permitir que o usuário altere os dados pessoais de sua conta.

RU11 - O sistema deve permitir que o usuário altere os dados das vagas de origem e destino.

RU12 - O sistema deve permitir que o usuário exclua a oferta de vaga.

RU13 - O sistema deve conter funcionalidade que permita aplicar filtros pela vaga de origem, vaga de destino, região administrativa, nome da instituição de ensino, ano e turno.

RU14 - O sistema deve permitir, após login, que os usuários que não obtiveram retorno na busca automática acessem a funcionalidade de busca por vagas.

RU15 - O sistema deve exibir até cinquenta resultados por página.

RU16 - O sistema deve exibir, após a respectiva seleção, os dados para contato do ofertante da vaga.

3.4. Requisitos funcionais

Segundo Sommerville (2011), os requisitos funcionais são declarações que descrevem os serviços que o sistema deve oferecer, como o sistema deve responder a certas entradas e como ele deve se comportar em determinadas situações. Em casos específicos, os requisitos funcionais também podem incluir o sistema não deve fazer. A seguir são listados os requisitos funcionais (RF) deste projeto:

Tabela 3.1 Requisitos Funcionais: RF01 - Manter usuário

RF01 - Manter usuário	
Descrição	Requisito do usuário atendido
O sistema deve permitir o cadastro de usuário, armazenar os dados pessoais, de acesso e de recuperação de senha.	RU01, RU02, RU03, RU04 e RU10

Tabela 3.2 Requisitos Funcionais: RF02 - Manter vagas

RF02 - Manter vagas	
Descrição	Requisito do usuário atendido
O sistema deve armazenar uma lista de instituições de ensino e regiões do DF e permitir que os usuários cadastrem os dados referentes às vagas de origem e de destino.	RU05, RU06, RU11e RU12

Tabela 3.3 Requisitos Funcionais: RF03 - Exibir resultado da busca

RF03 - Exibir resultado da busca automática	
Descrição	Requisito do usuário atendido
O sistema deve realizar o cruzamento dos dados e apresentar uma vaga que corresponda à busca.	RU07

Tabela 3.4 Requisitos Funcionais: RF04 - Enviar notificação

RF04 - Enviar notificação	
Descrição	Requisito do usuário atendido
O sistema deve enviar um e-mail notificando aos pares correspondes à troca de vaga	RU08, RU09

Tabela 3.5 Requisitos Funcionais: RF05 - Categorizar os dados

RF05 - Categorizar os dados	
Descrição	Requisito do usuário atendido
O sistema deve estruturar os dados de forma que possam ser recuperados pela busca categorizada	RU13

Tabela 3.6 Requisitos Funcionais: RF06 - Exibir resultados

RF06 - Exibir resultados	
Descrição	Requisito do usuário atendido
O sistema deve apresentar lista de vagas correspondentes à busca categorizada	RU14, RU15 e RU16

3.5. Requisitos não funcionais

Os requisitos não funcionais, para Sommerville (2011), são limitações impostas aos serviços ou funcionalidades oferecidas pelo sistema. Eles abrangem tempo, processo de desenvolvimento e definições normativas. Ao contrário das características ou serviços específicos do sistema, os requisitos não funcionais geralmente se aplicam ao sistema como um todo. Abaixo serão definidos requisitos não funcionais (RNF) relacionados a usabilidade, disponibilidade, portabilidade e segurança.

Tabela 3.7 Requisitos Não-Funcionais: RNF01 - Usabilidade

RNF01 - Usabilidade
Descrição de requisito não funcional
O sistema deverá conter uma interface simples e apresentar uma linguagem concisa e de fácil compreensão baseando-se nas heurísticas de Nielsen.

Tabela 3.8 Requisitos Não-Funcionais: RNF02 - Disponibilidade

RNF02 - Disponibilidade
Descrição de requisito não funcional

O sistema deverá ter recursos para priorizar o melhor desempenho e disponibilidade com escalabilidade garantida no plano Blaze- Firebase (200 mil conexões simultâneas/banco de dados).

Tabela 3.9 Requisitos Não-Funcionais: RNF03 - Portabilidade

RNF03 - Portabilidade
Descrição de requisito não funcional
O sistema deverá ser compatível com os principais navegadores disponíveis no mercado: Mozilla Firefox, Google Chrome, Opera, Safari e Microsoft Edge.

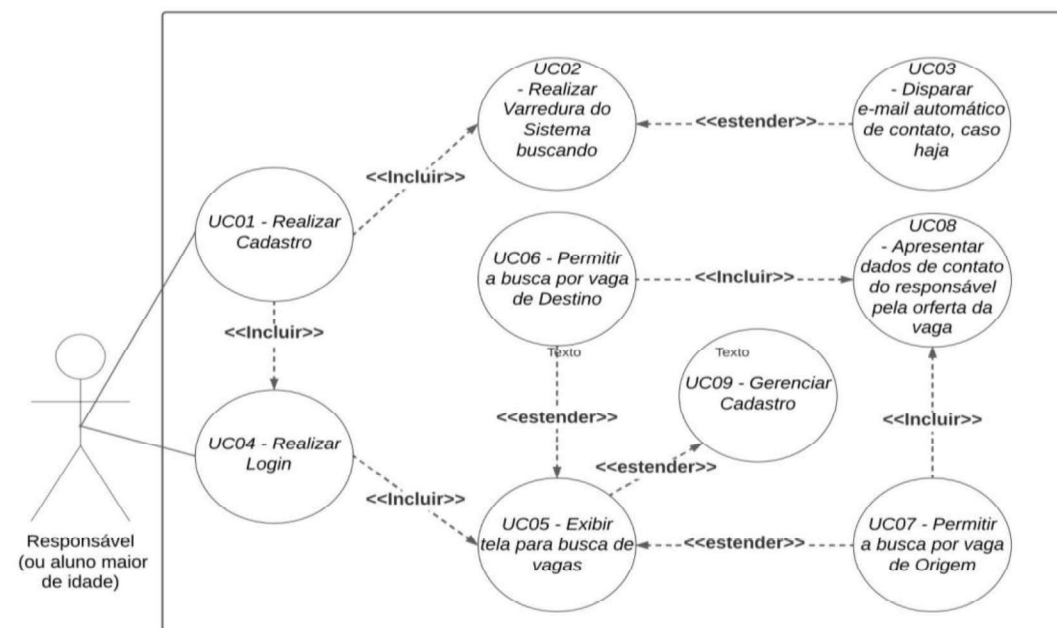
Tabela 3.10 Requisitos Não-Funcionais: RNF04 - Segurança

RNF04 - Segurança
Descrição de requisito não funcional
O sistema deve garantir a proteção dos dados dos usuários e estar em consonância com a Lei Geral de Proteção de Dados (LGPD).

3.6. Diagrama de Casos de Uso Macro

Nesta seção, através do Diagrama de Casos de Uso Macro (Figura 3.1), demonstramos as interações entre o usuário e o portal de vagas e ainda as funcionalidades apresentadas dentro deste ambiente a partir dos requisitos funcionais do sistema.

Figura 3.1 Casos de uso

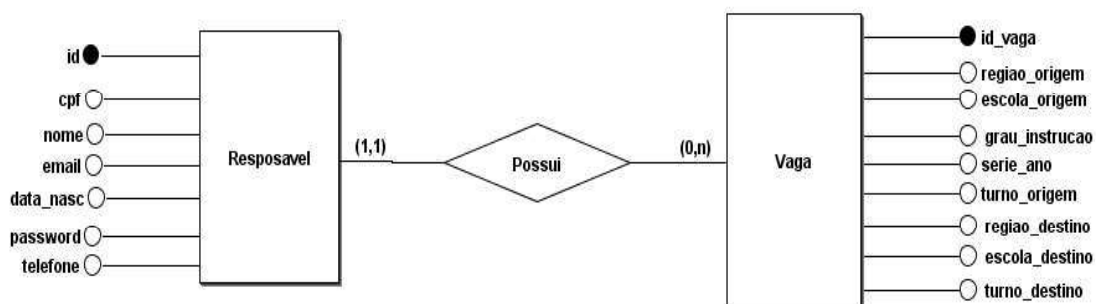


Fonte: Elaborado pelos autores

3.7. Modelo do banco de dados

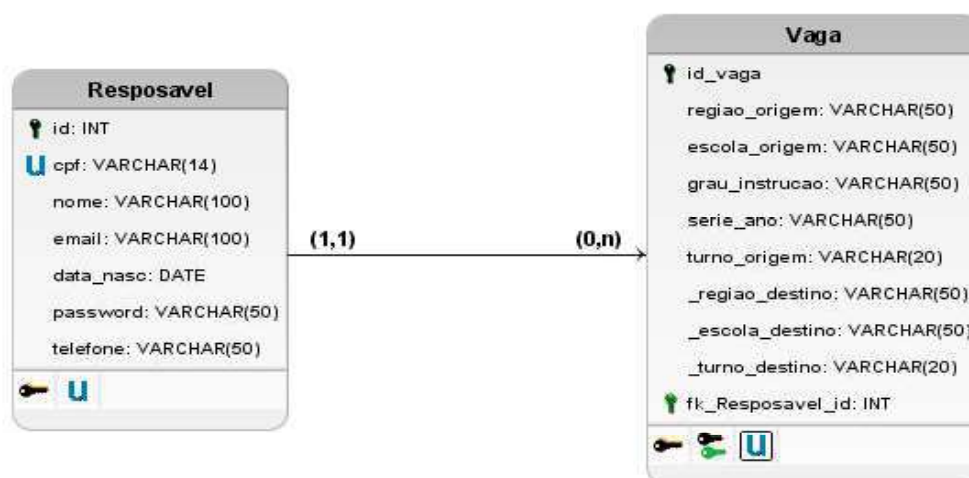
A fim de facilitar a estruturação e organização dos dados, além de demonstrar de maneira didática o banco de dados relacional a ser implementado, elaboramos o modelo conceitual e lógico de dados. A Figura 3.1 representa o modelo conceitual e a Figura 3.2 representa o modelo lógico.

Figura 3.2 Modelo Conceitual



Fonte: Elaborado pelos autores

Figura 3.3 Modelo Lógico



Fonte: Elaborado pelos autores

A partir dos requisitos e modelagem realizada foi possível realizar o desenvolvimento do sistema como apresentado no próximo capítulo.

4. ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção, apresentaremos relatos referentes ao desenvolvimento da aplicação web, análise do sistema e os resultados obtidos durante o período de desenvolvimento.

4.1. Relatos sobre o desenvolvimento e análise.

Para desenvolvimento desse projeto, foi feita uma análise preliminar de sistemas afins para tentarmos compreender quais funcionalidades nossa aplicação poderia apresentar e quais lacunas deveria suprir, visto que não encontramos um sistema que viabilizasse permutas no segmento proposto. A partir daí, iniciamos o levantamento de requisitos e a definição das tecnologias para o projeto. Deu-se início à exploração e estudo das linguagens de programação, frameworks e demais recursos previstos para a implementação. No entanto, no processo de desenvolvimento percebemos que algumas tecnologias definidas inicialmente não seriam mais necessárias ou deveriam ser substituídas por outras mais adequadas.

Na fase de estruturação do banco de dados, foi verificado que o banco de dados fornecido pelo Firebase (Cloud Firestore) não seria o mais adequado para o nosso projeto, pois se tratava de um banco orientado a documentos, ou seja, NoSQL (Not Only SQL). Logo, optamos por um Sistema de Gerenciamento de Banco de Dados em que fosse possível utilizar a linguagem SQL (Structured Query Language) para definir e manipular os dados. Nesse caso, o SQLite mostrou-se o mais adequado, pois além de ser relacional, é um SGBD pequeno, leve e rápido (SQLite, 2023).

Ao renunciar o armazenamento proveniente do Firebase, também dispensamos seu recurso de autenticação (Firebase Authentication). Com isso, identificamos, após pesquisas, uma nova tecnologia para autenticação mais congruente com as alterações evidenciadas no projeto, o Passport.js. Esse recurso ofereceu uma estrutura flexível e modular que foi inserido na aplicação e permitiu uma integração com o banco de dados relacional do SQLite, simplificando o desenvolvimento da função de autenticação e provendo a segurança necessária. Por meio do SQLite foram criadas duas tabelas. A primeira recebe os dados referentes à autenticação do usuário (responsável pela vaga) em uma integração com o

Passport.js. A segunda recebe os dados referentes à vaga desejada e a ofertada pelo responsável. Destarte foi estabelecido o relacionamento entre as tabelas por meio do atributo de chave primária e chave estrangeira, conforme a modelagem apresentada inicialmente.

Ademais, durante o processo de desenvolvimento foi identificado que prescindia-se do uso de ReactJs, pois as interfaces da aplicação não apresentaram um nível de interatividade e reatividade que demandassem o recurso. Consequentemente, o NextJS que é um framework para o ReactJS também foi dispensado. No entanto, foi necessária a tecnologia de *backend* Node.JS. que é um ambiente de tempo de execução JavaScript, multiplataforma e de código aberto (Foundation, 2023). E para simplificar sua utilização, foi adotado o Express.js, que é um *framework* flexível do Node.js que fornece um conjunto de recursos para web (Foundation, 2023). Finalmente, para criar templates reutilizáveis e dinâmicos, gerando conteúdo HTML de forma eficiente foi adotado o Handlebars.js

Desse modo, a arquitetura do sistema implementado pode ser representada, de maneira resumida, pela figura 4.1 a seguir:

Figura 4.1 Arquitetura implementada



Fonte: Elaborado pelos autores

Visando a assegurar que as funcionalidades do sistema estivessem em conformidade com os requisitos estabelecidos, foram conduzidos testes, correções, ajustes de interface e outras medidas cruciais, ao longo do processo de desenvolvimento da aplicação. No que diz respeito aos requisitos não funcionais, com

relação à usabilidade, buscou-se maior ênfase na criação de uma interface simples, com uma linguagem de fácil compreensão, além de um menu com apenas quatro opções e um botão pesquisar centralizado na homepage, para facilitar a interação e o aprendizado, conforme preconiza a literatura de heurística supracitada. Quanto à portabilidade, as tecnologias adotadas permitiram a compatibilidade com diversos navegadores, além de um layout responsivo, adaptando a interface de usuário a diferentes tamanhos de tela e dispositivos. No tocante à segurança, a adoção do Passport.JS ofereceu recursos avançados de segurança e autenticação para validar e proteger os dados dos usuários.

Por fim, destaca-se que a aplicação atendeu satisfatoriamente aos requisitos funcionais necessários à sua implementação e posterior utilização pelos usuários da rede pública de ensino, pois ele oferece: cadastro de usuário; armazenamento de dados pessoais; acesso, recuperação de senha e autenticação; listagem de instituições de ensino e regiões do DF; cadastro de dados referentes às vaga de origem e de destino; estruturação de dados para busca categorizada e exibição de vagas correspondentes à busca categorizada. Contudo, devido a limitações técnicas, imperícia e prazo restrito, a funcionalidade de notificação automática de usuários está elencada como trabalhos futuros, não afetando o propósito deste projeto.

4.2. Apresentação da interface do sistema

Esta seção apresenta a interface da aplicação desenvolvida, a organização dos elementos gráficos e visuais como a logomarca, a ordenação dos textos, disposição de menus e botões, bem como as funcionalidades que os elementos desempenham nas páginas. O sistema, testado pelos autores, apresentou os resultados esperados e definidos na análise de requisitos e posteriormente será disponibilizado aos estudantes da rede pública de ensino do Distrito Federal.

4.2.1. Página inicial

A figura 4.2 apresenta a página inicial com a logomarca no canto superior esquerdo e o menu de navegação no canto superior direito. As funcionalidades das páginas acessadas por meio dos botões: “Perfil”, “Vagas Disponíveis” e “Procurar”,

vão direcionar o usuário para a página de login, pois são restritas aos usuários cadastrados.

Figura 4.2 Página inicial da Aplicação



fonte: Elaborado pelos autores

4.2.2. Cadastrar Vaga

A figura 4.3 mostra a página “Cadastrar Vaga” que contém o formulário que deverá ser preenchido pelos novos usuários. Nele serão preenchidos os dados pessoais do responsável legal bem como os dados referentes a vaga ocupada atualmente na rede pública e a vaga desejada. Ao clicar no botão “Enviar” esses dados serão armazenados no banco de dados.

Figura 4.3 Cadastrar Vaga

Divulgue a vaga preenchendo o formulário

Preencha os dados do Responsável:

Nome do Responsável:

E-mail para contato:

Telefone de contato:

Preencha os dados da vaga de origem (atual):

Região Administrativa:

Escola:

Grau de Instrução:

Série/ Ano:

Turno:

Preencha os dados da vaga de destino (desejada):

Região Administrativa:

Escola:

Turno:

Fonte: Elaborado pelos autores

4.2.3. Página de login

A figura 4.4 mostra a página de login, cuja autenticação é realizada por meio do Passport.js, sendo requisito necessário para acesso às funcionalidades de busca por vagas e administração do perfil de usuário (responsável).

Figura 4.4 Tela de Login

CPF:

Senha:

[Esqueceu a senha?](#)

[Não tem uma conta? Criar conta](#)

Fonte: Elaborado pelos autores

4.2.4. Vagas Disponíveis

A figura 4.5 mostra a página “vagas disponíveis”, cujo acesso é restrito e está condicionado ao login. Essa medida visa a dar mais segurança aos dados pessoais dos usuários, limitando seu acesso a usuários cadastrados. Após digitar um termo, ou parte de uma palavra referente à vaga desejada, o sistema exibe uma lista correspondente.

Figura 4.5 Vagas Disponíveis

TrocaVaga
O seu portal de vagas do ensino básico da Rede Pública do Distrito Federal

Encontre a melhor vaga!

Digite a vaga que está buscando

Veja as nossas vagas em destaque

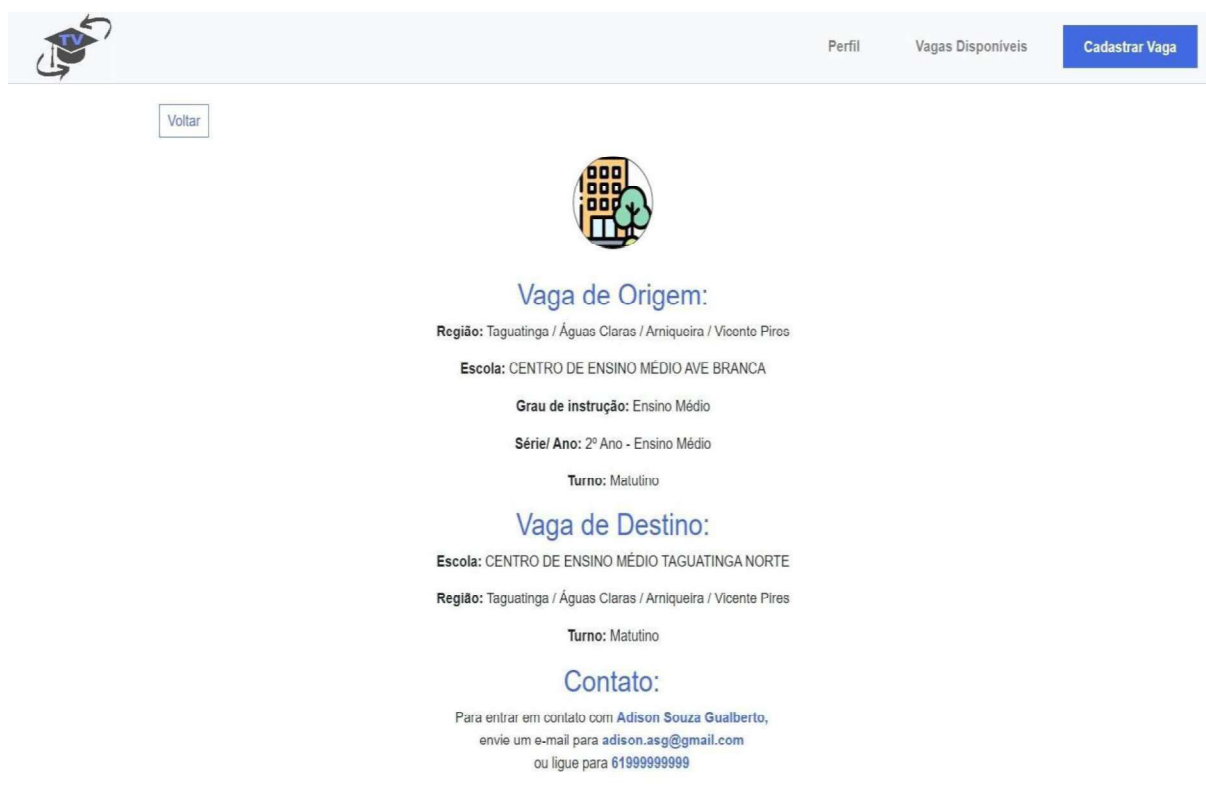
	Série: 2º Ano - Ensino Médio Origem: CENTRO DE ENSINO MÉDIO AVE BRANCA - Taguatinga / Águas Claras / Amiqueira / Vicente Pires Destino: CENTRO DE ENSINO MÉDIO TAGUATINGA NORTE - Taguatinga / Águas Claras / Amiqueira / Vicente Pires	Ver vaga NOVA
	Série: 1º Período - Educação Infantil Origem: ASSOC DE MAES PAIS AMIGOS REABILITADORES DE EXCESCOLA PARQUE - AMP - Plano Piloto / Cruzeiro / Jardim Botânico / Lago Norte / Varjão Destino: CENTRO DE EDUCAÇÃO INFANTIL NO DIÁLOGO E INFINITO II - Núcleo Bandeirante / Cantagrande / Buarque de Figueiredo	Ver vaga NOVA
	Série: 4º Série - Ensino Fundamental Origem: CENTRO DE EDUCAÇÃO DA PRIMEIRA INFÂNCIA CUTIA - Samambala Destino: CENTRO DE ENSINO FUNDAMENTAL 03 DE BRAZILÂNDIA - Brasília	Ver vaga NOVA

Fonte: Elaborado pelos autores

4.2.5. Detalhamento da vaga

A figura 4.6 mostra a tela que é exibida após se clicar no botão “Ver vaga”, presente em cada item correspondente à busca realizada. Nela estão presentes as informações da vaga ofertada, e da vaga buscada pelo responsável, além dos dados de contato como e-mail (obrigatório) e telefone que é fornecido opcionalmente.

Figura 4.6 Ver vaga



Vaga de Origem:

Região: Taguatinga / Águas Claras / Arniqueira / Vicente Pires

Escola: CENTRO DE ENSINO MÉDIO AVE BRANCA

Grau de instrução: Ensino Médio

Série/ Ano: 2º Ano - Ensino Médio

Turno: Matutino

Vaga de Destino:

Escola: CENTRO DE ENSINO MÉDIO TAGUATINGA NORTE

Região: Taguatinga / Águas Claras / Arniqueira / Vicente Pires

Turno: Matutino

Contato:

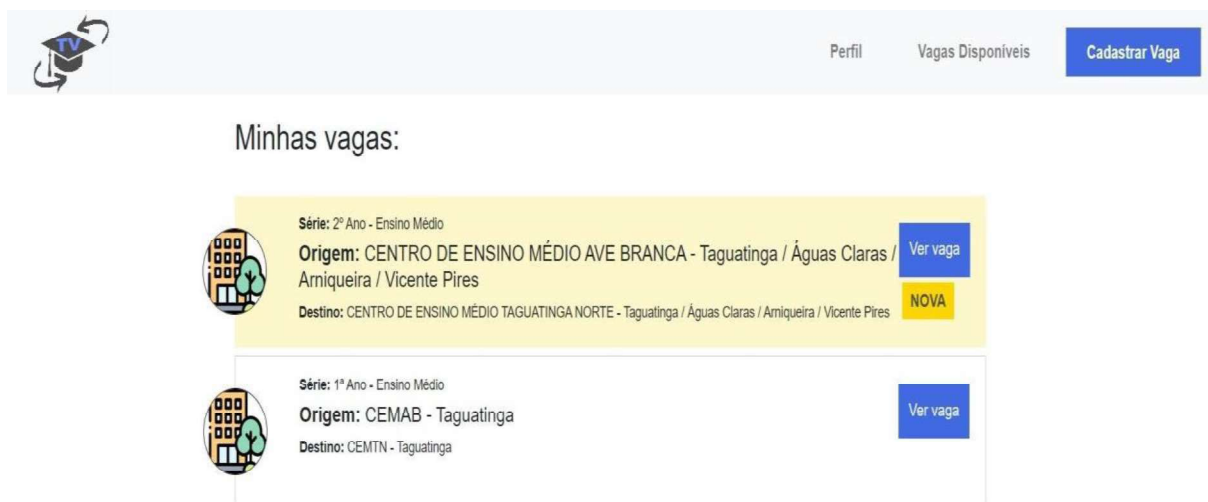
Para entrar em contato com **Adison Souza Gualberto**,
 envie um e-mail para adison.asg@gmail.com
 ou ligue para **61999999999**

Fonte: Elaborado pelos autores

4.2.6. Perfil

A figura 4.7 mostra a página “Perfil” que é acessada por meio do menu superior. Após a autenticação, o usuário pode gerenciar as vagas publicadas em seu perfil.

Figura 4.7 Perfil



Fonte: Elaborado pelos autores

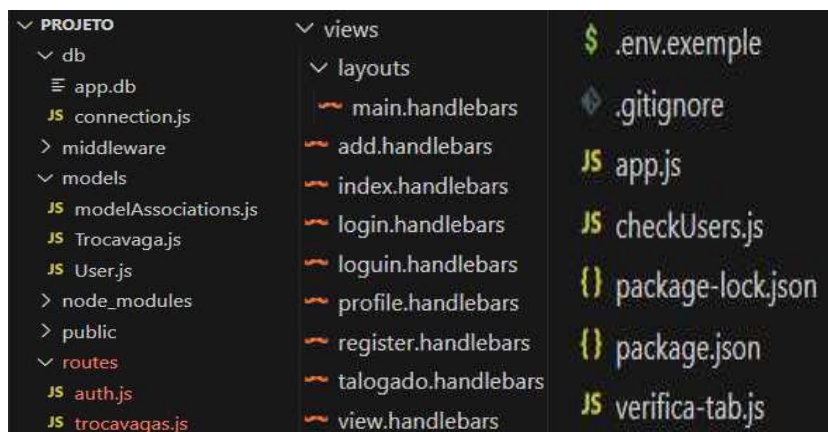
4.3. Apresentação da implementação backend do sistema

Esta seção apresenta a lógica utilizada para a execução da aplicação e a manipulação dos dados. Consiste na implementação e manutenção, realizada com o nodejs, banco de dados e a lógica de aplicação trabalhando em segundo plano, dando suporte às atividades executadas no frontend.

4.3.1. Estrutura do Projeto

A figura 4.8 exibe a estruturação do projeto amplamente, com os principais arquivos e pastas. Os arquivos essenciais para a execução do projeto são: package.json, onde se encontram informações de autoria e de gerenciamento das dependências do projeto; app.js, que exibe a configuração e inicialização básica do servidor web, das rotas e do banco de dados; trocavagas.js, localizado dentro da pasta routes, onde são realizadas as configurações das rotas de acesso para cada página .handlebars; Trocavaga.js, localizado na pasta models, realizando a criação do banco de dados e configuração dos dados a serem captados; connection.js, localizado na pasta db, responsável pela conexão com o banco de dados; app.db, também localizado na pasta db, que contém a visualização da estrutura do banco de dados criado; e por fim os arquivos .handlebars, contidos dentro da pasta views, que representam cada uma das páginas do frontend.

Figura 4.8 Estrutura do projeto



Fonte: Elaborado pelos autores

4.3.2. *Package.json*

A figura 4.9 exibe o arquivo package.json onde são apresentados metadados sobre o projeto, que no caso mostra dados sobre os autores, tipo de licença e as dependências: Body-parser, express-handlebars, express-session, passport, sequelize, sqlite3, dependências de desenvolvimento baseada no nodemon, entre outros.

Figura 4.9 Package.json

```
"author": "\"Adison <adison.asg@gmail.com>, Julio <juliocezargd@gmail.com>",
"license": "ISC",
"dependencies": {
  "bcryptjs": "^2.4.3",
  "body-parser": "^1.20.2",
  "dotenv": "^16.3.1",
  "express": "^4.18.2",
  "express-handlebars": "^7.1.2",
  "express-session": "^1.17.3",
  "mime": "^3.0.0",
  "nodemailer": "^6.4.17",
  "passport": "^0.4.1",
  "passport-local": "^1.0.0",
  "pg": "^8.11.3",
  "sequelize": "^6.34.0",
  "sqlite3": "^5.1.6"
},
"devDependencies": {
  "nodemon": "^3.0.1"
}
}
```

Fonte: Elaborado pelos autores

4.3.3. App.js

A figura 4.10 exibe o arquivo app.js, utilizado para a configuração e inicialização básica e ponto de entrada principal neste aplicativo Node.js. Aqui foi realizada a configuração do servidor Express além da configuração das rotas de acesso, configurações de sessão, conexão com o banco de dados, configurações globais e inicialização da aplicação.

Figura 4.10 App.js

```
const express = require('express');
const exphbs = require('express-handlebars');
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;
const bcrypt = require('bcryptjs');
const session = require('express-session');
const bodyParser = require('body-parser');
const Sequelize = require('sequelize');
const db = require('./db/connection');
const User = require('./models/User');
const path = require('path');
const mime = require('mime-types');
```

Fonte: Elaborado pelos autores

A figura 4.11 exibe parte do código utilizado na barra de pesquisa da página principal no frontend. Este código tem como finalidade a busca de vagas, por uma palavra ou fragmento de palavra, contendo o texto cadastrado na vaga, seja pela origem ou destino da vaga anunciada.

Figura 4.11 Código Barra de pesquisa

```

app.get('/', (req, res) => {
  let search = req.query.trocavaga;
  let query = '%' + search + '%';

  if (!search) {
    Trocavaga.findAll({
      order: [
        ['createdAt', 'DESC']
      ]
    })
    .then(trocavagas => {
      res.render('index', {
        trocavagas
      });
    })
    .catch(err => console.log(err));
  } else {
    Trocavaga.findAll({
      where: {
        [Op.or]: [
          {
            escola_origem: {
              [Op.like]: query
            }
          },
          {
            escola_destino: {
              [Op.like]: query
            }
          },
          {
            regioao_origem: {
              [Op.like]: query
            }
          },
          {
            regioao_destino: {
              [Op.like]: query
            }
          }
        ]
      },
      order: [
        ['createdAt', 'DESC']
      ]
    })
    .then(trocavagas => {
      res.render('index', {
        trocavagas
      });
    })
    .catch(err => console.log(err));
  }
});

```

Fonte: elaborado pelos autores

4.3.4. Trocavagas.js

A figura 4.12 exibe o arquivo trocavagas.js, utilizado com a finalidade de compreender e organizar as rotas de acesso da aplicação, além de conduzir a inserção de dados no banco.

Figura 4.12 Trocavagas.js

```
const express = require('express');
const router = express.Router();
const Trocavaga = require('../models/Trocavaga');

// rota de teste
router.get('/test', (req, res) => {
  res.send('deu certo');
});

// detalhe da vaga -> view/1, view/2
router.get('/view/:id', (req, res) => Trocavaga.findOne({
  where: {id: req.params.id}
}).then(Trocavaga => {

  res.render('view', {
    Trocavaga
  });

}).catch(err => console.log(err)));

// form da rota de envio
router.get('/add', (req, res) => {
  res.render('add');
});

// form da rota de envio
router.get('/profile', (req, res) => {
  res.render('profile');
});

// form da rota de envio
router.get('/loguin', (req, res) => {
  res.render('loguin');
});

// add vaga via post
router.post('/add', (req, res) => {
```

Fonte: elaborado pelos autores

4.3.5. Trocavaga.js

A figura 4.13 exibe o arquivo Trocavaga.js, onde é realizada a importação do Sequelize, biblioteca para trabalhar com bancos de dados relacionais no Node.js, caracterizando os dados a serem inseridos no banco de dados.

Figura 4.13 trocavaga.js

```
const Sequelize = require('sequelize');
const db = require('../db/connection');
const Trocavaga = db.sequelize.define('trocavaga', {
  responsavel: {
    type: Sequelize.STRING,
  },
  email: {
    type: Sequelize.STRING,
  },
  telefone: {
    type: Sequelize.STRING,
  },
  regioao_origem: {
    type: Sequelize.STRING,
  },
  escola_origem: {
    type: Sequelize.STRING,
  },
  grau_instrucao: {
    type: Sequelize.STRING,
  },
  serie_ano: {
    type: Sequelize.STRING,
  },
  turno_origem: {
    type: Sequelize.STRING,
  },
  regioao_destino: {
    type: Sequelize.STRING,
  },
  escola_destino: {
    type: Sequelize.STRING,
  },
  turno_destino: {
    type: Sequelize.STRING,
  },
  new_trocavaga: {
    type: Sequelize.INTEGER,
  }
});
module.exports = Trocavaga
```

Fonte: elaborado pelos autores

4.3.6. Connection.js

A figura 4.14 exibe o arquivo connection.js, que realiza a conexão com o banco de dados sqlite, através do Sequelize.

Figura 4.14 Connection.js

```
const Sequelize = require('sequelize');

const sequelize = new Sequelize({
  dialect: 'sqlite',
  storage: './db/app.db'
});

module.exports = sequelize
```

Fonte: elaborado pelos autores

4.3.7. app.db

A figura 4.15 exibe o arquivo app.db, local onde os dados do banco de dados são armazenados, onde temos as seguintes entidades: id, responsavel, email, telefone, regioao_origem, escola_origem, grau_instrucao, serie_ano, turno_origem, regioao_destino, escola_destino, turno_destino, new_trocavaga, createdAt e updatedAt.

A entidade id é auto incrementável para cada vaga cadastrada, a entidade new_trocavaga é cadastrada automaticamente com “0” para paga antiga e “1” para vagas novas, definidas através das variáveis: createAt e createAt, que captam a data e hora do cadastro, e os dados das demais entidades são captadas através da página cadastrar vagas no frontend.

Figura 4.15 app.db

db > app.db					
SELECT * FROM trocavagas					
	id	responsavel	email	telefone	regiao_origem
1	6	fabio	fabio.123@gmail.com	61944444444	Taguatinga
2	21	Adison Souza Gualb...	adison.asg@gmail.c...	61999999999	Taguatinga / Águas...
	escola_origem	grau_instrucao	serie_ano	turno_origem	regiao_destino
	CEMAB	Ensino Médio	1ª Ano - Ensino Mé...	Vespertino	Taguatinga
	CENTRO DE ENSINO M...	Ensino Médio	2ª Ano - Ensino Mé...	Matutino	Taguatinga / Águas...
	escola_destino	turno_destino	new_trocavaga	createdAt	updatedAt
	CEMTN	Vespertino	0	2023-10-31 19:23:5...	2023-10-31 19:23:5...
	CENTRO DE ENSINO M...	Matutino	1	2023-11-05 00:15:3...	2023-11-05 00:15:3...

Fonte: elaborado pelos autores

4.3.8. Handlebars

A figura 4.16 exibe parte do arquivo index.handlebars, trecho que mostra como foi realizada a busca de dados para serem apresentadas nas views da página principal do frontend, traz as informações de cadastro da vaga de forma resumida, com as informações da série, Origem e destino da vaga e de a vaga é recente.

Figura 4.16 index.handlebars

```

    {{#each trocavagas}}
      <li class="list-group-item {{#if
dataValues.new_trocavaga }}new-trocavaga{{/if}}">
        
        <p><b>Série: </b>{{dataValues.serie_ano}}</p>
        <h5><b>Origem: </b>{{dataValues.escola_origem}} -
{{dataValues.regiao_origem}}</h5>
        <p><b>Destino: </b>{{dataValues.escola_destino}} -
{{dataValues.regiao_destino}}</p>
        <span class="new-trocavaga-label">Nova</span>
        <a class="btn btn-primary"
href="/trocavagas/view/{{dataValues.id}}">Ver vaga</a>
      </li>
    {{/each}}

```

Fonte: elaborado pelos autores

O arquivo add.handlebars é responsável pela lógica de captação dos dados inseridos na página Cadastrar Vaga no frontend. Um dos pontos de maior dificuldade nesta seção foi como mostrar as possibilidades de seleção da escola, visto que a lista total de escolas cadastradas é de mais de 800 escolas. Optamos pela utilização de um select dinâmico (Figura 4.17), onde que após escolher qual região administrativa (primeiro select), seria realizado um filtro para a seleção das escolhas (segundo select), mostrando apenas as escolas pertencentes àquela região.

Figura 4.17 add.handlebars

```

<div class="form-group">
  <label for="primeiroSelect">Região
  Administrativa:</label>
  <select class="form-control" id="primeiroSelect"
  name="regiao_origem" onchange="carregarEscolas()">
    <option value="selecione">Selecione a Região
    Administrativa</option>
    <option value="Brazlândia">Brazlândia
    <option value="Ceilândia / Sol Nascente / Pôr do
    Sol">Ceilândia / Sol Nascente / Pôr do Sol
  </select>
  <div class="form-group">
    <label for="segundoSelect">Escola:</label>
    <select class="form-control" id="segundoSelect"
    name="escola_origem">
      <option value="selecione">Selecione uma Região
      Primeiro</option>
    </select>

    <script>
      function carregarEscolas() {
        var primeiroSelect =
        document.getElementById("primeiroSelect");
        var segundoSelect =
        document.getElementById("segundoSelect");

        var regiaoSelecionada = primeiroSelect.value;

        // Limpar o segundo select
        segundoSelect.innerHTML = "";
        if (regiaoSelecionada === "Brazlândia") {
          // Se o Brazlandia for selecionada, preencher com
          escolas de Brazlandia
          var escolasBrazlandia = ["Selecione uma Escola",
          "CENTRO DE EDUCAÇÃO DA PRIMEIRA INFÂNCIA FLAMBOYANT", "CENTRO DE
          EDUCAÇÃO DA PRIMEIRA INFÂNCIA SAGUI", "CENTRO DE EDUCAÇÃO INFANTIL
          01 de BRAZLÂNDIA", "CENTRO DE EDUCAÇÃO INFANTIL 02 DE BRAZLÂNDIA",
          "CENTRO DE EDUCAÇÃO INFANTIL 03 de BRAZLÂNDIA", "CENTRO DE EDUCAÇÃO
          INFANTIL TIA NAIR", "CENTRO DE EDUCACIONAL INCRA 08", "CENTRO DE
          EDUCACIONAL IRMÃ MARIA REGINA VELANES REGIS", "CENTRO DE ENSINO
          ESPECIAL 01 DE BRAZLANDIA", "CENTRO DE ENSINO FUNDAMENTAL 01 de
          BRAZLÂNDIA", "CENTRO DE ENSINO FUNDAMENTAL 02 de BRAZLÂNDIA",
          "CENTRO DE ENSINO FUNDAMENTAL 03 de BRAZLÂNDIA", "CENTRO DE ENSINO
          FUNDAMENTAL INCRA 07 de BRAZLÂNDIA", "CENTRO DE ENSINO FUNDAMENTAL
          VENDINHA", "CENTRO DE ENSINO MÉDIO 01 de BRAZLÂNDIA", "CENTRO DE
          ENSINO MÉDIO 02 de BRAZLÂNDIA", "CENTRO EDUCACIONAL 02 de
          BRAZLÂNDIA", "CENTRO EDUCACIONAL 04 de BRAZLÂNDIA", "CENTRO
          INTERESCOLAR DE LINGUAS DE BRAZLANDIA", "ESCOLA CLASSE 01 DE
          BRAZLANDIA", "ESCOLA CLASSE 01 INCRA 08", "ESCOLA CLASSE 03 DE
          BRAZLÂNDIA", "ESCOLA CLASSE 05 DE BRAZLÂNDIA", "ESCOLA CLASSE 06 DE
          BRAZLANDIA", "ESCOLA CLASSE 07 DE BRAZLÂNDIA", "ESCOLA CLASSE 08 DE
          BRAZLÂNDIA", "ESCOLA CLASSE 09 de BRAZLÂNDIA", "ESCOLA CLASSE
          ALMESCOLA CLASSEEGAS", "ESCOLA CLASSE BUCANHÃO", "ESCOLA CLASSE
          CHAPADINHA", "ESCOLA CLASSE INCRA 06", "ESCOLA CLASSE PÓLO AGRÍCOLA
          DA TORRE", "ESCOLA PARQUE DA NATUREZA DE BRAZLANDIA", "JARDIM DE
          INFÂNCIA MENINO JESUS", "CAIC PROFESSOR BENEDITO CARLOS DE
          OLIVEIRA", "CENTRO DE EDUCAÇÃO DA PRIMEIRA INFÂNCIA AROEIRA"];

          escolasBrazlandia .forEach(function(escola) {

```

Fonte: elaborado pelos autores

A figura 4.18 exibe parte do arquivo view.handlebars, que é o detalhamento das informações cadastradas de uma determinada vaga, trazendo para o usuário, no frontend, todas as informações cadastradas para uma vaga na tela cadastrar vaga.

Figura 4.18 view.handlebars

```

</div>
<div class="col-md-12">
  <h2 class="main-color">Vaga de Origem:</h2>
  <p><b>Região:</b> {{Trocavaga.dataValues.regiao_origem}}</p>
  <p><b>Escola:</b> {{Trocavaga.dataValues.escola_origem}}</p>
  <p><b>Grau de instrução:</b>
  {{Trocavaga.dataValues.grau_instrucao}}</p>
  <p><b>Série/ Ano:</b> {{Trocavaga.dataValues.serie_ano}}</p>
  <p><b>Turno:</b> {{Trocavaga.dataValues.turno_origem}}</p>

  <h2 class="main-color">Vaga de Destino:</h2>
  <p><b>Escola:</b> {{Trocavaga.dataValues.escola_destino}}</p>
  <p><b>Região:</b> {{Trocavaga.dataValues.regiao_destino}}</p>
  <p><b>Turno:</b> {{Trocavaga.dataValues.turno_destino}}</p>

  <h2 class="main-color">Contato:</h2>
  <p>
    Para entrar em contato com <span class="bold main-
color">{{Trocavaga.dataValues.responsavel}},</span>
    <br>envie um e-mail para <span class="bold main-
color">{{Trocavaga.dataValues.email}}</span><br>
    ou ligue para <span class="bold main-
color">{{Trocavaga.dataValues.telefone}}</span>
    <br></br>
  </p>
</div>

```

Fonte: elaborado pelos autores

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como propósito principal o desenvolvimento de uma solução que desse uma maior efetividade ao processo de transferência escolar, visto que havia uma queixa recorrente a esse respeito por parte da sociedade. O primeiro desafio seria construí-lo de maneira independente, sem recursos, cooperação ou integração com sistemas da Secretaria de Estado de Educação do Distrito Federal. Dessa maneira, surgiu a ideia de um sistema que viabilizasse a troca de vagas entre os estudantes, permitindo a transferência sem a necessidade da reserva prévia de uma vaga na escola desejada.

As pesquisas e o aprendizado sobre as linguagens de programação e tecnologias definidas no projeto foram fundamentais para a construção dessa aplicação. As tecnologias empregadas no desenvolvimento do software possibilitaram a criação de um sistema que atende satisfatoriamente sua finalidade, permitindo o cadastro de informações dos responsáveis e das vagas, o acesso dos dados em ambiente restrito, a gestão de dados categorizados, a publicação bem como a exibição das listas de vagas para permuta. Além disso, a segurança dos dados dos usuários e do sistema é fortalecida pela autenticação do Passport.js. Como resultado final, os objetivos e os principais requisitos do projeto, de maneira geral, foram alcançados.

Vale ressaltar que a aplicação, nessa versão inicial, não atendeu ao requisito de notificação automática dos responsáveis por e-mail. Essa funcionalidade fica definida como trabalho futuro, podendo vir a figurar como um recurso pago, como medida para custear sua manutenção e divulgação. Outrossim, deverá ser criado os termos de uso para utilização da plataforma, sobretudo em consonância com a Lei Geral de Proteção de Dados (LGPD). Outra proposta de melhoria e atualização futura é a criação de uma ferramenta que permita ao responsável cadastrar e manter a vaga de vários alunos (filhos por exemplo) na mesma conta do responsável, simultaneamente.

Por fim, conclui-se que o aprimoramento, implementação, publicação e difusão dessa aplicação web poderá torná-la uma valiosa ferramenta para os estudantes da rede pública de ensino, tendo em vista que a possibilidade de escolher a escola para transferência oferece benefícios que vão desde a comodidade, economia de tempo e

a flexibilidade para atender às necessidades educacionais individuais e circunstâncias de cada aluno e família.

REFERÊNCIAS BIBLIOGRÁFICAS

Acesso à Informação. **PARTICIPA DF**, 2023. Disponível em: <<https://www.participa.df.gov.br/>>. Acesso em: 28 de junho de 2023.

BRASIL. Constituição (1988). **Constituição da República Federativa do Brasil**. Brasília, DF: CAPÍTULO III DA EDUCAÇÃO, DA CULTURA E DO DESPORTO Seção I DA EDUCAÇÃO Senado Federal: Centro Gráfico, 1988.

BRASIL. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). **Resumo Técnico do Distrito Federal, Censo Escolar da Educação Básica**, 2021. Brasília: Inep, 2021a. Disponível em: <<https://www.gov.br/inep/pt-br/centrais-de-conteudo/acervo-linha-editorial/publicacoes-institucionais/estatisticas-e-indicadores-educacionais/resumo-tecnico-do-distrito-federal-censo-da-educacao-basica-2021/>>. Acesso em: 08 de maio de 2023.

BRASIL. Lei nº 12.527, de 18 de novembro de 2011. **Lei de Acesso à Informação – LAI**. Brasília, DF: Diário Oficial da União, 1990.

Carta de Serviços ao Cidadão, **Educação-DF**, 2022. Disponível em: <https://www.educacao.df.gov.br/wp-content/uploads/2019/06/Carta_Servicos_SEEDF_2022.pdf/>. Acesso em: 08 de maio de 2023.

COVID 19, **Portal SES-DF**, 2023. Disponível em: <<https://www.saude.df.gov.br/coronavirus/>>. Acesso em: 28 de junho de 2023.

FOUNDATION, O. **Aplicativos da Web**. OpenJS Foundation, 2023. Disponível em: <<https://expressjs.com/pt-br/>>. Acesso em: 25 de outubro de 2023.

FOUNDATION, O. **Introdução ao Node.js**. OpenJS Foundation, 2023. Disponível em: <<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs/>>. Acesso em: 26 Out. 2023.

GOOGLE. **Documentation for Firebase**. Google, 2023. Disponível em: <<https://firebase.google.com/?hl=pt-br/>>. Acesso em: 28 de maio de 2023.

HÁMORI, F. **3 reasons to learn React (and how to get started)**, 2018. Disponível em: <<https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>>. Acesso em: 28 de maio de 2023.

Handlebars.J.S. **What is Handlebars**. Handlebars.J.S, 2023. Disponível em: <<https://handlebarsjs.com/guide/#what-is-handlebars/>>. Acesso em: 25 de outubro de 2023.

MICROSOFT. **Documentation for Visual Studio Code**. Microsoft, 2023. Disponível em: <<https://code.visualstudio.com/docs/>>. Acesso em: 30 de maio de 2023.

MOZILLA. **Introdução Express/Node**. Mozilla, 2023. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction/>. Acesso em: 27 de outubro de 2023

MOZILLA. **Documentation for JavaScript**. Mozilla, 2023. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/>>. Acesso em: 30 de maio de 2023

MOZILLA. **HTML basics**. Mozilla, 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics/> Acesso em: 29 de maio de 2023.

MOZILLA. **What is CSS**. Mozilla, 2023. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS/> Acesso em: 29 de maio de 2023

NEXTJS. **Documentation for NextJS**. NextJS, 2023. Disponível em: <<https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs/>>. Acesso em: 30 de maio de 2023.

NIELSEN, Jakob. **Usability Engineering**. [s.l.]: Morgan Kaufmann, 1993.

PASSPORT. **Overview**. Passport.JS, 2023. Disponível em: <<https://www.passportjs.org/concepts/authentication/>>. Acesso em: 06 de novembro de 2023.

PEREIRA, C. R. **Aplicações web real-time com Node.js**. [S.l.]: Editora Casa do Código, 2014.

Permuta e Distribuição, **Anajusfe**, 2023. Disponível em: <<https://www.anajusfe.org.br/permuta-redistribuicao/>>. Acesso em: 08 de maio de 2023.

Processo de Escolha e Permuta, **Portal Educação Guarulhos**, 2023. Disponível em: <<https://portaleducacao.guarulhos.sp.gov.br/siseduc/portal/site/listar/categoria/71/>>. Acesso em: 08 de maio de 2023.

Regimento da Rede Pública de Ensino do Distrito Federal, **Educação-DF**, 2019. Disponível em: <<https://www.educacao.df.gov.br/wp-content/uploads/2017/08/Regimento-SEEDF-COMPLETO-FINAL.pdf>> Acesso em: 08 de maio de 2023.

SOMMERVILLE, I. **Engenharia de software**, 9a. São Paulo, SP, Brasil, 2011.

SQLite. **What Is SQLite.** SQLite, 2023. Disponível em:<
<https://www.sqlite.org/index.html>>. Acesso em: 25 de outubro de 2023.

VERCEL. **Documentation for Vercel.** Vercel, 2023. Disponível em:
<<https://vercel.com/>> Acesso em: 29 de junho de 2023.