



**INSTITUTO
FEDERAL**
Paraná

Projeto de Aplicações Web

Implementação de uma plataforma de emprego destinada a universitários

Bacharelado em Ciência da computação

Heloisa de Sales Mariano

Professor: João Orlando

Pinhais - Paraná
Dezembro de 2025

Contents

1 Planejamento e definição	3
1.1 Introdução	3
1.2 Requisitos do Sistema	3
1.2.1 Requisitos Funcionais (RF)	3
1.2.2 Requisitos não Funcionais (RNF)	4
1.3 User Story	5
2 Modelagem da plataforma	7
2.1 Diagrama de Caso de uso	7
2.2 Diagrama de Entidade-Relacionamento (DER)	9
2.3 Tabelas de Entidades Relacionamentos	10
2.3.1 Relacionamentos Lógicos e Funcionamento do Sistema	10
3 Prototipação	12
3.1 Estrutura	12
3.2 Acesso ao Quant-UX	12
4 Implementação	13
4.1 Tecnologias Utilizadas	13
4.2 Estrutura do Projeto WebWorkifs	14
4.3 Funcionalidades Implementadas	14
4.3.1 Módulo Aluno	15
4.3.2 Módulo Empresa	15
4.3.3 Componentes Bootstrap Utilizados	15
4.3.4 Configuração	16
4.3.5 Segurança	16
4.3.6 Persistência de Dados	16
4.3.7 Responsividade	17
5 Decisões de Implementação e Justificativas	18
5.1 Ausência de Backend	18
5.2 Gestão de Múltiplos Usuários	18
5.3 Organização de Código	18

5.4 Integração OAuth	19
6 Referências	20

1. Planejamento e definição

1.1 Introdução

A aplicação consiste em um sistema web de vagas de emprego voltado para estudantes universitários. A ideia é implementar uma plataforma que facilite a busca e candidatura a oportunidades compatíveis com a rotina acadêmica, suprindo lacunas das plataformas genéricas existentes, que priorizam vagas CLT ou PJ, dificultando o acesso a vagas de estágio.

O problema identificado é que muitos estudantes enfrentam dificuldades em encontrar estágios compatíveis com seus horários e cursos, especialmente aqueles em que o estágio é obrigatório, gerando uma pressão adicional. Além disso, a ausência de experiência prévia dificulta a entrada no mercado de trabalho, mesmo quando os estudantes possuem competências acadêmicas relevantes.

Para contornar essas dificuldades, a plataforma permitirá que os estudantes filtrem vagas de acordo suas preferencias. Adicionalmente, será oferecida uma forma alternativa de currículo, permitindo que os estudantes apresentem trabalhos acadêmicos e projetos desenvolvidos, destacando suas competências e experiências acadêmicas. Dessa forma, a aplicação visa conectar estudantes e empresas de forma mais eficiente, valorizando habilidades e experiências acadêmicas, mesmo na ausência de experiência profissional formal.

1.2 Requisitos do Sistema

1.2.1 Requisitos Funcionais (RF) .

RF01 — O sistema deve permitir o cadastro de usuários.

RF02 — O sistema deve permitir login para aluno e empresa.

RF03 — O aluno deve visualizar vagas disponíveis.

RF04 — A empresa deve criar vagas.

RF05 — O aluno deve se candidatar às vagas.

RF06 — A empresa deve visualizar candidatos.

RF08 — O aluno deve filtrar as vagas por localização e área de interesse desejada.

RF09 — O sistema deve permitir que o aluno modifique suas informações curriculares e adicione projetos pessoais para implementação do seu currículo.

RF10 — O sistema deve permitir que o aluno aceite propostas das vagas.

RF11 — O sistema deve permitir que o aluno veja suas notificações de alerta onde será publicada a proposta das empresas referente a suas candidaturas.

RF12 — O sistema deve permitir que a empresa edite suas vagas.

RF20 — O sistema deve permitir que a empresa delete suas vagas.

RF21 — A empresa deve aceitar o candidato desejado.

RF22 — A empresa deve ter todas as informações do candidato sobre seus horários para não impactar em seus estudos.

RF23 — O sistema deve ter uma verificação de duas etapas para empresas na seleção de criação de vagas, para evitar vagas com informações erradas.

RF24 — A empresa deve visualizar suas vagas criadas.

RF25 — O aluno deve visualizar suas vagas inscritas.

RF26 — O sistema deve impedir que um usuário não logado se inscreva em uma vaga.

RF27 — O sistema deve permitir que os usuários adicionem foto de perfil e banner.

RF28 — O sistema deve permitir que o usuário se deslogue de sua conta.

RF29 — O sistema deve permitir que o aluno visualize seu próprio currículo.

1.2.2 Requisitos não Funcionais (RNF) .

RNF01 — O sistema deve exibir mensagens de sucesso ou erro após ações realizadas, como cadastro de vaga ou candidatura.

RNF02 — O sistema deve garantir que apenas usuários autenticados possam acessar funcionalidades restritas, como se candidatar a uma vaga.

RNF03 — O sistema deve ser compatível com dispositivos móveis e diferentes navegadores.

RNF04 — O sistema deve armazenar os dados dos usuários de forma segura.

RNF05 — O sistema deve oferecer autenticação segura, utilizando senha e verificação de duas etapas.

RNF06 — O sistema deve apresentar interface intuitiva e de fácil navegação, facilitando o uso por estudantes e empresas.

RNF07 — O sistema deve apresentar alertas e notificações de forma clara, informando sobre novas propostas, candidaturas ou atualizações.

RNF08 — O sistema deve apresentar design consistente entre todas as telas, mantendo identidade visual padronizada.

RNF09 — O sistema deve permitir que imagens de perfil e banners sejam carregadas sem distorções, respeitando proporções e limites de tamanho.

RNF10 — O sistema deve permitir que o usuário altere sua senha caso a esqueça no login.

RNF11 — O sistema deve garantir consistência de dados, evitando duplicidade de cadastros ou vagas.

RNF12 — O sistema deve permitir pesquisa e filtragem de vagas de forma clara e rápida.

RNF13 — O sistema deve permitir que os usuários possam alterar suas informações de login.

RNF14 — O sistema deve suportar criação de perfis distintos para aluno e empresa.

1.3 User Story

As User Stories descrevem, de forma simples, o que cada tipo de usuário deseja realizar no sistema e qual é o objetivo dessas ações.

ID	User Story	Critérios de Aceitação
US01	Como aluno, quero criar uma conta para poder acessar a plataforma.	O aluno consegue cadastrar login e senha; recebe confirmação de cadastro.
US02	Como aluno, quero editar meu perfil e fazer upload de projetos para mostrar minhas competências acadêmicas.	O aluno consegue atualizar informações pessoais e adicionar projetos; alterações são salvas com sucesso.
US03	Como aluno, quero visualizar vagas disponíveis e filtrar por área de interesse e localização para encontrar oportunidades compatíveis.	O aluno visualiza a lista de vagas; filtros funcionam corretamente; resultados atualizados.
US04	Como aluno, quero me candidatar a vagas e acompanhar minhas inscrições para gerenciar meus processos seletivos.	O aluno consegue se candidatar; inscrição é registrada; pode visualizar status da candidatura.
US05	Como aluno, quero aceitar propostas de vagas para confirmar minha participação.	O aluno visualiza propostas e consegue aceitá-las; sistema confirma a ação.
US06	Como aluno, quero visualizar meu currículo para acompanhar as informações que as empresas veem.	O aluno consegue acessar seu currículo completo e atualizado.
US07	Como aluno, quero deslogar da plataforma para encerrar minha sessão de forma segura.	O aluno consegue sair do sistema e não tem acesso às funcionalidades sem novo login.

Table 1.1: *User Stories do Aluno*

ID	User Story	Critérios de Aceitação
US08	Como empresa, quero criar uma conta para acessar o sistema.	A empresa consegue cadastrar login e senha; recebe confirmação de cadastro.
US09	Como empresa, quero editar meu perfil para manter minhas informações atualizadas.	A empresa consegue alterar dados de perfil.
US10	Como empresa, quero criar, editar e deletar vagas para gerenciar as oportunidades que oferece.	A empresa consegue criar, alterar e remover vagas; notificações de sucesso/erro aparecem corretamente.
US11	Como empresa, quero visualizar candidatos para analisar os perfis antes de aceitar alguém.	A empresa consegue acessar lista de candidatos.
US12	Como empresa, quero aceitar candidatos para concluir o processo seletivo de cada vaga.	A empresa consegue aceitar candidatos; status atualizado; candidato notificado.

Table 1.2: *User Stories da Empresa*

2. Modelagem da plataforma

2.1 Diagrama de Caso de uso

O Diagrama de Casos de Uso da plataforma apresenta os atores principais Aluno e Empresa e as funcionalidades que cada um pode realizar. Para organizar as relações entre os casos de uso, utilizamos include e extend:

- Include: usado quando um caso de uso depende obrigatoriamente de outro. Ou seja, sempre que a ação principal acontecer, a ação incluída também acontece. Por exemplo: Quando a empresa aceita um candidato, ela precisa visualizar os candidatos antes. Então, "Aceitar candidato" inclui "Visualizar candidatos".
- Extend: usado para ações opcionais ou que só acontecem em situações específicas. Por exemplo: "Aceitar proposta" ocorre somente depois que o aluno se candidatou, sendo uma ação opcional, então é uma extensão de "Candidatar-se a vagas".
- funcionalidades dos Usuários: O aluno pode criar conta, editar seu perfil, fazer upload de projetos, visualizar e filtrar vagas, candidatar-se às vagas, acompanhar suas inscrições, aceitar propostas, deslogar e visualizar seu próprio currículo.
Já a empresa pode criar conta, editar seu perfil, criar, editar e deletar vagas, além de visualizar candidatos e aceitar aqueles desejados.

A Figura 2.2 ilustra o diagrama completo de casos de uso do sistema.

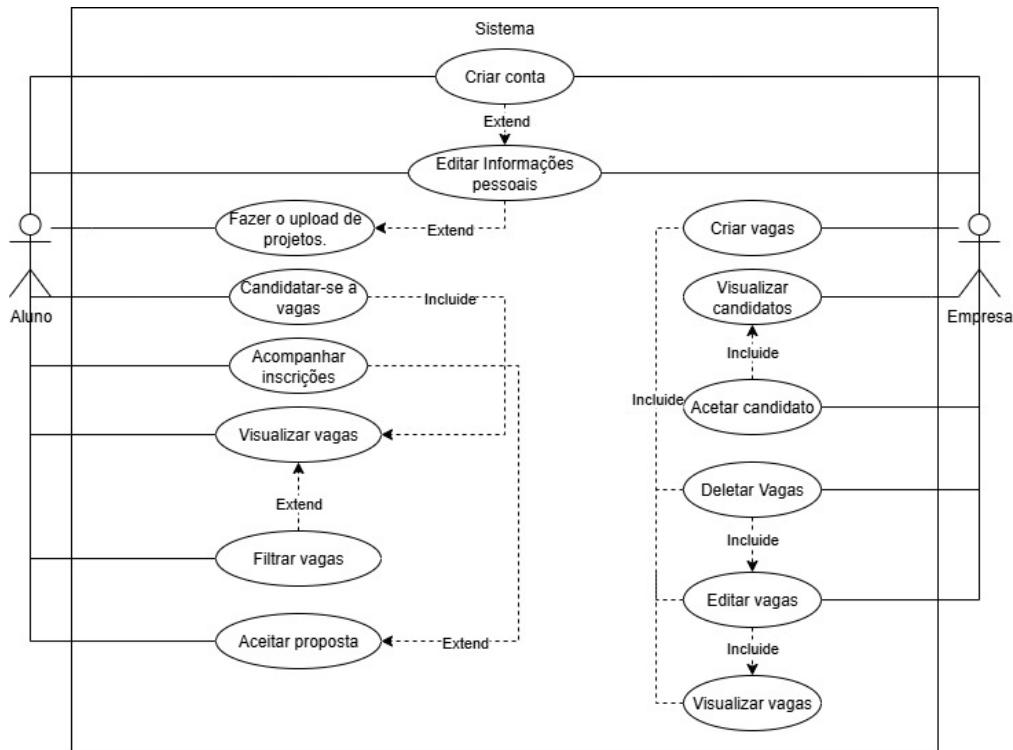


Figure 2.1: O Diagrama de Casos de Uso apresentado ilustra a interação entre os atores do sistema

2.2 Diagrama de Entidade-Relacionamento (DER)

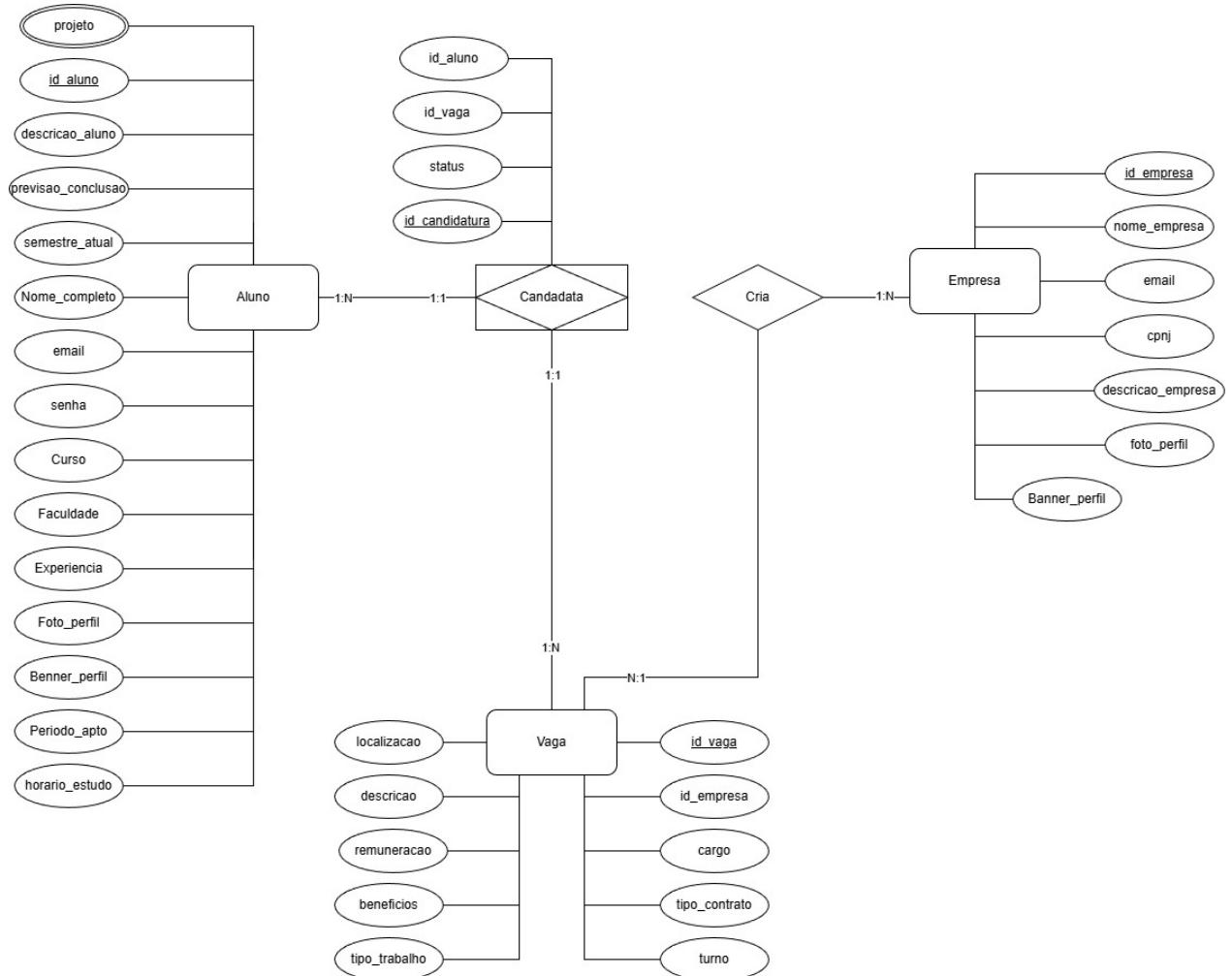


Figure 2.2: Diagrama de Entidade e Relacionamento demonstrando visualmente o funcionamento do sistema

2.3 Tabelas de Entidades Relacionamentos

Table 2.1: Tabela de Entidades e Relacionamentos do Sistema

Entidade 1	Relacionamento	Entidade 2	Cardinal	Explicação
Empresa	cria	Vaga	1:N	Uma empresa pode criar várias vagas, mas cada vaga pertence a apenas uma empresa.
Vaga	recebe	Candidatura	1:N	Cada vaga pode receber várias candidaturas de alunos; cada candidatura está ligada a apenas uma vaga.
Aluno	realiza	Candidatura	1:N	Um aluno pode se candidatar a várias vagas; cada candidatura está associada a um único aluno.
Vaga	criada	Empresa	N:1	Cada vaga é criada por uma única empresa; a empresa pode ter várias vagas publicadas.

2.3.1 Relacionamentos Lógicos e Funcionamento do Sistema .

Algumas ações e relações do sistema não aparecem no diagrama de entidades e relacionamentos porque ele serve apenas para mostrar como os dados estão organizados e conectados. No entanto, para entender como o sistema realmente funciona, é importante explicar essas interações do sistema.

- Aluno/Currículo : Cada aluno possui exatamente um currículo que pode ser editado conforme desejar.
- Aluno/Vaga: Cada aluno pode visualizar uma ou mais vagas, podendo escolher em qual se candidatar. E uma vaga pode ser visualizada por um ou mais alunos.
- Aluno/Empresa recebe Notificação: Cada usuário (aluno ou empresa) pode receber várias notificações sobre candidaturas, propostas ou atualizações. Onde um usuário pode receber nenhuma ou mais notificações.
- Candidatura aprovada/rejeitada por Empresa: Cada candidatura é analisada por uma empresa, que decide o status da candidatura. Ou seja, uma empresa pode gerenciar uma ou mais candidaturas.
- Aluno edita Perfil: Cada aluno possui um perfil que pode ser editado; o perfil armazena informações pessoais e acadêmicas, como nome, curso, semestre, projetos e experiências.

- Empresa edita Perfil: Cada empresa possui um perfil que pode ser editado; o perfil armazena informações da empresa.
- Fotos: Cada usuário pode editar seu perfil pessoal colocando uma foto de perfil e de banner, tanto sendo empresa ou aluno.
- Candidatura/Aluno/Empresa: A empresa visualiza a candidatura vendo o currículo do aluno que se inscreveu na vaga.
- Aluno/Projeto: Um aluno pode ter um ou mais projetos em seu perfil.

3. Prototipação

3.1 Estrutura

O sistema permite que usuários sem cadastro acessem a tela de login e filtrem vagas por cidade e área de interesse. Caso tentem se candidatar sem estar logados, o sistema informa que é necessário realizar o login.

Na tela de login, o usuário escolhe entre “Sou estudante” ou “Sou empresa”. Para estudantes, é possível logar usando email e senha, criar conta caso ainda não possuam cadastro, ou recuperar a senha informando email, nova senha e um PIN de autenticação de dois fatores. Para empresas, o processo é similar, mas também exige o CNPJ.

Após o login, os alunos podem filtrar vagas por cidade, turno e área de interesse, candidatar-se a vagas, gerenciar seu perfil (deslogar, editar informações pessoais, fazer upload de projetos acadêmicos, adicionar foto de perfil e banner), visualizar seu currículo, acessar “Minhas Vagas” para acompanhar candidaturas e receber notificações ou propostas de empresas.

As empresas, por sua vez, podem editar seu perfil (foto, banner, nome da empresa e descrição), criar, editar e visualizar vagas com informações completas sobre cargo, área, tipo de contrato, turno, tipo de trabalho, local, descrição, salário e benefícios, além de visualizar candidatos inscritos em suas vagas e receber notificações sobre candidaturas na seção de alertas.

3.2 Acesso ao Quant-UX

Para a prototipação do sistema, utilizei a plataforma QuantX, que permite criar interfaces de forma visual e interativa, facilitando a definição de telas, fluxos de navegação e experiência do usuário (UX) antes da implementação do sistema.

Clique aqui para acessar o protótipo [Clique aqui](#)

4. Implementação

4.1 Tecnologias Utilizadas

- **Framework Front-end:**

- **Bootstrap 5.3.2** - Framework CSS responsivo para interface moderna e componentes interativos.
- **Bootstrap Icons 1.11.3** - Biblioteca de ícones para elementos visuais.

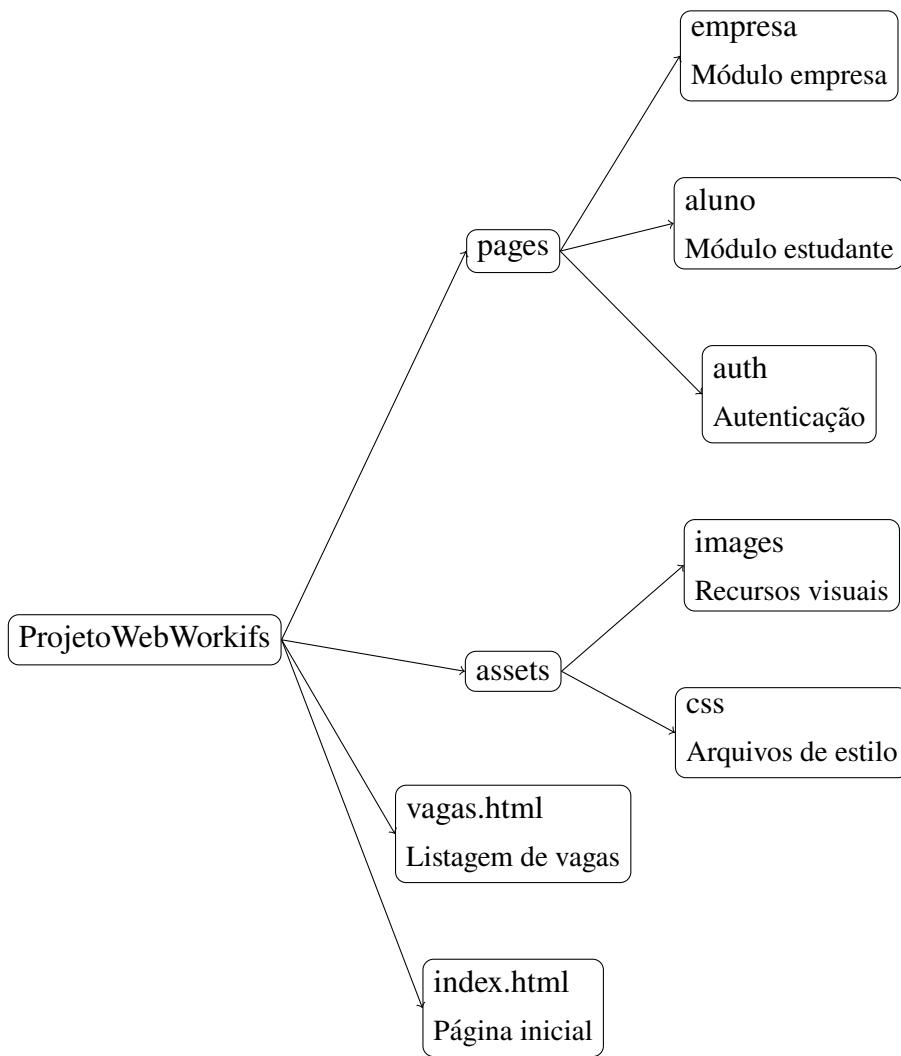
- **Linguagens:**

- **HTML5** - Estruturação semântica das páginas.
- **CSS3** - Estilização customizada com tema verde e branco.
- **JavaScript (Vanilla)** - Lógica de negócio e interatividade.

- **API Pública:**

- **Google OAuth 2.0 / OpenID Connect** - Autenticação social integrada.
- **Client ID:** 431128073932-41s9ru6oaajl8gjdtv dto844dgi7crvn.apps.googleusercontent.com
- **Fluxo:** Implicit Flow com ID Token para obter dados do usuário.

4.2 Estrutura do Projeto WebWorkifs



4.3 Funcionalidades Implementadas

Sistema de Autenticação

- Cadastro e login tradicional com validação
- Integração OAuth 2.0 com Google (login social)
- Recuperação de senha
- Suporte a múltiplos usuários simultâneos
- Persistência de sessão com localStorage

4.3.1 Módulo Aluno

- Perfil editável com informações acadêmicas e experiência
- Busca de vagas com filtros (cidade, área)
- Sistema de candidaturas
- Visualização do próprio currículo
- Acompanhamento de candidaturas com status

4.3.2 Módulo Empresa

- Dashboard com vagas criadas
- CRUD completo de vagas (criar, visualizar, editar)
- Visualização de candidatos por vaga
- Sistema de currículo dos candidatos
- Aceitação de candidaturas
- Bloqueio de edição em vagas preenchidas

4.3.3 Componentes Bootstrap Utilizados

- Navbar responsivo com dropdown
- Cards para exibição de vagas
- Modals para detalhes e formulários
- Forms com validação
- Badges para status
- Toasts para notificações
- Carousel para banners
- Grid system para responsividade
- Integração da API Google OAuth

4.3.4 Configuração

- Criação de projeto no Google Cloud Console
- Configuração de credenciais OAuth 2.0
- Definição de URIs de redirecionamento autorizados
- Implementação do fluxo de autenticação

Fluxo de Autenticação

1. Usuário clica em "Login com Google"
2. Redirecionamento para página de consentimento do Google
3. Google retorna tokens (`access_token` e `id_token`)
4. Callback processa o ID Token (JWT) e extrai dados do usuário
5. Conta é criada automaticamente ou login é realizado
6. Usuário é redirecionado para a aplicação autenticado

4.3.5 Segurança

- Uso de `state` e `nonce` para prevenção de ataques CSRF
- Validação de tokens no callback
- Sessão gerenciada com `sessionStorage` e `localStorage`

4.3.6 Persistência de Dados

- Utilização do `localStorage` do navegador para simular banco de dados:
 - `contasAlunos[]` - Contas de estudantes
 - `contasEmpresas[]` - Contas de empresas
 - `vagasEmpresa[]` - Vagas criadas
 - `candidaturas[]` - Aplicações às vagas
 - `userLogado/empresaLogada` - Sessão ativa

4.3.7 Responsividade

- Design *mobile-first* com breakpoints Bootstrap
- Layout adaptativo para desktop, tablet e mobile
- Navbar colapsável em dispositivos móveis
- Cards e forms responsivos
- Imagens e carrossel otimizados

5. Decisões de Implementação e Justificativas

5.1 Ausência de Backend

Devido à complexidade de configurar um servidor backend e banco de dados SQL (que exigiria hospedagem, configuração de ambiente Node.js/PHP e gerenciamento de servidor), optou-se por uma solução **100% client-side** utilizando localStorage do navegador como camada de persistência. Isso permite que a aplicação funcione de forma totalmente funcional sem necessidade de infraestrutura de servidor.

5.2 Gestão de Múltiplos Usuários

Para suportar múltiplas contas simultâneas, implementou-se o uso de arrays no localStorage:

- `contasAlunos[]` - Contas de estudantes
- `contasEmpresas[]` - Contas de empresas

Essa abordagem supera a limitação inicial de sobreescriver dados de usuário único.

5.3 Organização de Código

Reestruturou-se a arquitetura do projeto em pastas modulares:

- `auth` - Autenticação
- `aluno` - Módulo estudante
- `empresa` - Módulo empresa
- `assets` - Arquivos de estilo e recursos visuais

Essa organização melhora a manutenibilidade e a escalabilidade do projeto, com atualização de todos os caminhos relativos.

5.4 Integração OAuth

Implementou-se o fluxo OAuth 2.0 do Google:

- Gerenciamento de tokens e validação de segurança
- Tratamento de callbacks
- Uso do *Implicit Flow* para autenticação sem necessidade de servidor intermediário

Essa abordagem permite login social totalmente funcional em ambiente client-side.

6. Referências

1. BOOTSTRAP. *Bootstrap 5.3.2 Documentation*. [S.l.]: Bootstrap Team, 2024. Disponível em: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. Acesso em: 04 dez. 2024.
2. GOOGLE INC. *Google Identity: OAuth 2.0 for Client-side Web Applications*. Mountain View: Google Developers, 2024. Disponível em: <https://developers.google.com/identity/protocols/oauth2/javascript-implicit-flow>. Acesso em: 04 dez. 2024.
3. MOZILLA FOUNDATION. *MDN Web Docs: JavaScript*. [S.l.]: Mozilla Developer Network, 2024. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em: 04 dez. 2024.