

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia de Mauá

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Amanda Lira Macêdo Nunes

Flávio Máximo dos Santos

Matheus Haruki Shiomi

Thiago Luiz Franceliano

TechJobs: Candidato.

Mauá

2026

Amanda Lira Macêdo Nunes

Flávio Máximo dos Santos

Matheus Haruki Shiomi

Thiago Luiz Franceliano

TechJobs: Candidato.

Relatório técnico-científico apresentado à Fatec de Mauá, como exigência parcial para obtenção do título de Tecnólogo em Desenvolvimento de Software Multiplataforma, sob a orientação da profa. Me. Luana Lourenço.

Mauá

2026

RESUMO

Este trabalho apresenta o desenvolvimento e a análise do sistema multiplataforma denominado TechJobs, uma solução tecnológica projetada para otimizar o processo de recrutamento e seleção de profissionais na área de Tecnologia da Informação. O objetivo principal do estudo consistiu em criar um ecossistema distribuído capaz de conectar candidatos e empresas por meio de um mecanismo ágil de correspondência por afinidade de habilidades. A metodologia adotou uma abordagem de pesquisa aplicada e estudo de caso tecnológico, estruturando a arquitetura do software a partir da rigorosa segregação de responsabilidades. Para o núcleo corporativo e a construção da Interface de Programação de Aplicações, utilizou-se a linguagem C# e o arcabouço .NET, enquanto o armazenamento de dados relacionais foi gerenciado pelo servidor SQL Server. As interfaces de usuário foram concebidas de forma especializada, empregando a biblioteca React para o ambiente de navegação na rede de alcance mundial e o arcabouço Flutter para a entrega de um aplicativo móvel com compilação nativa. Durante o ciclo de desenvolvimento, aplicaram-se práticas de código limpo e metodologias de verificação contínua, com destaque para a implementação de testes unitários e testes de interface de usuário, garantindo a integridade lógica e visual do produto. Adicionalmente, realizou-se uma auditoria técnica de acessibilidade baseada nas Diretrizes de Acessibilidade para Conteúdo Web, a fim de certificar a inclusão de pessoas com deficiência na utilização da ferramenta. Os resultados evidenciaram que a arquitetura adotada conferiu alta escalabilidade e estabilidade ao sistema, mantendo a consistência visual e o baixo tempo de resposta em requisições concorrentes. Conclui-se que a combinação de tecnologias modernas de programação e a aplicação disciplinada de testes e avaliações de acessibilidade resultam em uma ferramenta de recrutamento robusta, confiável e socialmente inclusiva, atendendo satisfatoriamente às exigências qualitativas do mercado corporativo contemporâneo.

Palavras-chave: desenvolvimento multiplataforma; engenharia de software; recrutamento tecnológico; testes de software; acessibilidade digital.

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Contextualização	6
1.2 Objetivo	7
1.3 Justificativa	7
1.4 Problema de pesquisa	8
1.5 Estrutura do relatório	9
2 DESENVOLVIMENTO	10
2.1 Fundamentação teórica	10
2.1.1 Revisão de literatura	11
2.1.2 Técnicas de desenvolvimento de software	11
2.1.3 Linguagens de programação e frameworks	12
2.1.4 Metodologias de teste de software	13
2.2 Metodologia	15
2.2.1 Tipo de pesquisa	15
2.2.2 Método de coleta de dados	16
2.2.3 Procedimentos de análise	16
2.2.4 Critérios de avaliação	17
2.3 Ficha técnica	18
2.4 Técnicas e testes de software e avaliação da acessibilidade	18
2.4.1 Testes unitários	19
2.4.2 Testes de performance	19
2.4.4 Testes de segurança	20
2.4.5 Metodologias de desenvolvimento e qualidade	21
2.4.6 Avaliação de acessibilidade	21
3 CONSIDERAÇÕES FINAIS	24
3.1 Principais conclusões	24
3.2 Recomendações	25
3.3 Limitações do estudo	25
3.4 Perspectivas futuras	26
REFERÊNCIAS	27
APÊNDICE A – INTERFACES GRÁFICAS DO SISTEMA TECHJOBS (WEB)	28
APÊNDICE B – INTERFACES GRÁFICAS DO SISTEMA TECHJOBS (MOBILE)	37

1 INTRODUÇÃO

O avanço tecnológico das últimas décadas tem transformado profundamente a maneira como pessoas se comunicam, estudam, trabalham e buscam oportunidades profissionais. No cenário atual, caracterizado pela digitalização de processos e pela expansão do acesso à internet, o desenvolvimento de software distribuído e multiplataforma tornou-se uma necessidade estratégica para empresas e instituições que desejam alcançar um público maior e oferecer soluções modernas, eficientes e acessíveis.

Dentro desse contexto, destaca-se o crescimento do mercado de Tecnologia da Informação (TI), que apresenta alta demanda por profissionais qualificados em diferentes níveis de experiência, desde estudantes em busca de estágio até profissionais seniores. No entanto, apesar do grande número de vagas disponíveis, muitos candidatos enfrentam dificuldades para encontrar oportunidades compatíveis com suas habilidades, enquanto empresas enfrentam desafios para localizar profissionais adequados às suas necessidades.

Plataformas digitais voltadas ao recrutamento e seleção têm desempenhado um papel fundamental nesse processo. Entretanto, muitas dessas soluções apresentam interfaces complexas, fluxos burocráticos e pouca interação dinâmica. Dessa forma, surge a necessidade de desenvolver sistemas mais intuitivos e eficientes, que facilitem a conexão entre talentos e organizações de forma rápida e assertiva.

Com base nesse cenário, este trabalho propõe o desenvolvimento de uma plataforma denominada TechJobs, inspirada na combinação de funcionalidades de redes profissionais e sistemas de correspondência por afinidade. Para atender aos requisitos de escalabilidade e manutenibilidade exigidos por sistemas modernos, o projeto adota uma arquitetura distribuída. O backend é desenvolvido utilizando a linguagem C# e o ecossistema .NET, sendo responsável por fornecer uma Interface de Programação de Aplicações (API) robusta e segura. O armazenamento e o gerenciamento dos dados estruturados são realizados por meio do Sistema Gerenciador de Banco de Dados (SGBD) SQL Server. Para o consumo dessa API, o sistema conta com duas frentes de interação: uma aplicação web focada em responsividade, desenvolvida com a biblioteca React, e um aplicativo móvel multiplataforma construído com o framework Flutter, utilizando a linguagem Dart.

Além da estruturação arquitetural, um aspecto fundamental deste projeto é a aplicação de metodologias e testes de software ao longo do ciclo de desenvolvimento. Essa prática assegura a qualidade do sistema, seu desempenho e sua confiabilidade, em estrita consonância com os pilares da Engenharia de Software e as boas práticas de desenvolvimento.

1.1 Contextualização

O desenvolvimento de software contemporâneo tem se consolidado em torno de arquiteturas baseadas em serviços e interfaces especializadas. Essa abordagem permite que diferentes plataformas — como navegadores web e dispositivos móveis operando Android ou iOS — consumam regras de negócio centralizadas, reduzindo a redundância de código e otimizando a manutenibilidade do projeto.

Com o crescimento do número de usuários que acessam serviços digitais por meio de smartphones, tornou-se essencial criar aplicações fluidas e de alto desempenho. Nesse sentido, frameworks modernos como o Flutter oferecem ferramentas que possibilitam o desenvolvimento ágil de interfaces com desempenho nativo para dispositivos móveis. Paralelamente, o uso do React no ecossistema web garante a criação de Interfaces de Usuário (UI) dinâmicas, componentizadas e reativas.

No cenário de recrutamento na área de TI, onde a demanda por profissionais é alta, a busca por vagas pode ser um processo fragmentado. O sistema TechJobs surge como uma alternativa que utiliza elementos visuais interativos para simplificar esse fluxo. A sustentação dessa plataforma exige um backend resiliente, o que justifica a escolha do C# com .NET, tecnologias amplamente consolidadas no mercado corporativo, aliadas ao rigor do SQL Server para garantir a integridade relacional das informações de usuários, currículos, vagas e empresas.

1.2 Objetivo

O objetivo geral deste trabalho é desenvolver e analisar a plataforma distribuída TechJobs, destinada a facilitar o processo de recrutamento e candidatura para vagas na área de Tecnologia da Informação, utilizando tecnologias modernas de desenvolvimento (backend corporativo e frontends especializados) e práticas rigorosas de testes de software.

Como objetivos específicos, destacam-se:

- Desenvolver uma API robusta utilizando a linguagem C# e o framework .NET para centralizar e processar as regras de negócio.
- Modelar e implementar um banco de dados relacional utilizando o SQL Server para o armazenamento seguro e íntegro das informações.
- Criar uma aplicação web dinâmica e responsiva utilizando React.
- Desenvolver um aplicativo móvel multiplataforma com o framework Flutter e a linguagem Dart.
- Implementar funcionalidades que permitam a correspondência eficiente entre candidatos e vagas com base em habilidades técnicas e interesses.
- Aplicar diferentes tipos de testes de software (unitários, de integração e de interface) para garantir a qualidade, a segurança e o desempenho contínuo do ecossistema.

1.3 Justificativa

A escolha do tema justifica-se pela latente necessidade de soluções tecnológicas eficientes que conectem profissionais de TI a empresas, aliada à importância acadêmica de projetar uma arquitetura de software completa, distribuída e alinhada às demandas reais da indústria. A área de Tecnologia da Informação apresenta constante expansão, mas a burocracia das plataformas legadas frequentemente prejudica o recrutamento ágil.

O desenvolvimento do TechJobs permite a integração prática de diversas disciplinas abordadas ao longo do curso de Desenvolvimento de Software Multiplataforma, materializando conceitos avançados de Engenharia de Software. O projeto abrange de ponta a ponta o ciclo de desenvolvimento: desde a modelagem de dados relacionais no SQL Server e a construção de APIs em .NET, até a prototipagem e implementação de interfaces focadas na Experiência do Usuário (UX) com React e Flutter.

Além disso, a implementação de testes — como testes unitários para validar regras de negócios no backend, testes de integração para verificar a comunicação entre as interfaces e a API, e testes de UI — justifica-se pela necessidade crítica de aplicar padrões de qualidade que previnam falhas estruturais e garantam a confiabilidade exigida por aplicações profissionais.

1.4 Problema de pesquisa

O desenvolvimento de um ecossistema de software que engloba um backend centralizado e múltiplos clientes (web e mobile) traz desafios complexos relacionados à arquitetura, à sincronização de dados, à segurança da informação e à usabilidade. Nesse contexto, surge a seguinte questão de pesquisa:

- Quais são as melhores práticas arquiteturais, ferramentas e metodologias de teste para garantir a qualidade, a usabilidade e a performance de uma plataforma distribuída (Web e Mobile) voltada ao recrutamento profissional na área de Tecnologia da Informação?

Essa questão orienta o desenvolvimento do sistema TechJobs, buscando identificar como a orquestração de tecnologias como C#/.NET, SQL Server, React e Flutter pode ser otimizada por meio de práticas de Código Limpo para resultar em um produto corporativo altamente confiável.

1.5 Estrutura do relatório

Este relatório está organizado em capítulos que descrevem detalhadamente o processo de pesquisa e desenvolvimento do sistema TechJobs, estruturados da seguinte forma:

- O Capítulo 1 – Introdução apresenta o contexto do trabalho, os objetivos propostos, a justificativa da pesquisa, o problema norteador e a estrutura geral do documento.
- O Capítulo 2 – Desenvolvimento engloba a Fundamentação Teórica, abordando a literatura pertinente sobre arquitetura de software, linguagens de programação (C#, Dart, JavaScript), frameworks aplicados (.NET, Flutter, React), o uso do banco de dados SQL Server e as metodologias de teste de software. Este capítulo também detalha a Metodologia adotada no estudo, a Ficha Técnica do sistema construído e as especificações dos testes e avaliações de acessibilidade realizados na plataforma.
- O Capítulo 3 – Considerações Finais apresenta a síntese e a análise conclusiva do projeto, destacando os resultados alcançados no desenvolvimento do backend e dos frontends, as limitações técnicas encontradas durante a integração do ecossistema e as perspectivas futuras para a evolução contínua da ferramenta.

2 DESENVOLVIMENTO

O desenvolvimento constitui o núcleo central deste relatório técnico-científico, detalhando o processo de concepção, estruturação arquitetural e validação da plataforma TechJobs. A construção de sistemas contemporâneos exige o alinhamento com os princípios da Engenharia de Software Moderna, visando criar aplicações escaláveis, de fácil manutenção e preparadas para a evolução contínua. Dessa forma, esta seção articula os fundamentos conceituais com as decisões práticas adotadas durante a implementação, guiando-se por padrões de design que promovem a segregação de responsabilidades e o isolamento das regras de negócio em relação às camadas de interface e banco de dados.

Neste capítulo, expõe-se detalhadamente a orquestração do ecossistema distribuído, evidenciando como a centralização lógica no backend em C# e .NET opera em harmonia com a persistência estruturada e segura no SQL Server. Simultaneamente, aborda-se a integração com as interfaces de usuário, garantindo alta responsividade na web com React e fluidez nativa no ambiente móvel multiplataforma com Flutter. Adicionalmente, descrevem-se os procedimentos metodológicos mobilizados para a especificação do software e a aplicação de técnicas de verificação, garantindo que o sistema revele defeitos precocemente e atenda aos requisitos de confiabilidade esperados. A adoção de práticas ágeis e a escrita cuidadosa do código também são evidenciadas como pilares essenciais para que o software seja compreendido e modificado com segurança por outros desenvolvedores no futuro.

2.1 Fundamentação teórica

O desenvolvimento de plataformas de software modernas e distribuídas exige o alinhamento entre conceitos sólidos de engenharia de software, padrões arquiteturais resilientes e escolhas tecnológicas que suportem a concorrência e a escalabilidade. Para sustentar a criação da plataforma TechJobs, esta seção aborda o referencial teórico que embasa as práticas ágeis, a garantia de qualidade por meio de código limpo, as especificidades da stack tecnológica adotada (backend em

C#/NET, web em React, mobile em Flutter e persistência em SQL Server) e as metodologias de teste necessárias para a validação do ecossistema.

2.1.1 Revisão de literatura

A construção de sistemas complexos que operam de forma integrada em ambientes web e mobile requer uma fundamentação clara sobre como estruturar os componentes de software de modo a isolar as regras de negócio das tecnologias de apresentação e persistência. Nesse cenário, o design arquitetural assume papel preponderante. Como destaca Martin (2020, p. 137) ao discutir a segregação de responsabilidades através da Arquitetura Limpa, "o objetivo principal da arquitetura de software é suportar o ciclo de vida do sistema. Uma boa arquitetura torna o sistema fácil de entender, fácil de desenvolver, fácil de manter e fácil de implantar".

Sob a ótica da Engenharia de Software Moderna, Valente (2020) aponta que os sistemas contemporâneos devem ser projetados pensando na evolução contínua e no desacoplamento. Modelos baseados em APIs e arquiteturas distribuídas permitem que múltiplos clientes consumam um núcleo comum de serviços, o que viabiliza o suporte simultâneo a interfaces React e Flutter sem a duplicação de lógicas de validação e processamento corporativo.

Além dos aspectos estruturais, a gestão do processo de desenvolvimento apoia-se em métodos ágeis (como Scrum e Kanban) para gerenciar o escopo de forma iterativa e incremental. Essas abordagens mitigam riscos de entrega e permitem uma resposta rápida às mudanças de requisitos de usabilidade e mercado que caracterizam aplicações voltadas ao recrutamento e seleção de profissionais (VALENTE, 2020).

2.1.2 Técnicas de desenvolvimento de software

Para transpor os modelos conceituais para um código-fonte sustentável, o uso de práticas de Código Limpo (Clean Code) e a automação do fluxo de implantação

são indispensáveis. A manutenibilidade de longo prazo de um sistema está diretamente ligada à clareza e à expressividade de seu código. Martin (2012, p. 7) reforça essa premissa ao afirmar que "deixar o código limpo é fundamental para que outros desenvolvedores possam ler, entender e modificar o software sem introduzir efeitos colaterais indesejados. Código limpo parece que foi escrito por alguém que se importa".

Além da escrita qualificada e da documentação do código, o desenvolvimento moderno adota os pilares da Integração Contínua (CI) e da Entrega Contínua (CD). De acordo com Humble e Farley (2014), a automação do processo de construção (build), testes e publicação reduz drasticamente o tempo entre a concepção de uma funcionalidade e sua disponibilização em ambiente produtivo. Ao estabelecer um pipeline de implantação automatizado, garante-se que cada alteração realizada tanto no backend .NET quanto nos clientes React e Flutter passe por uma esteira rigorosa de verificação de qualidade, minimizando falhas de integração e gargalos manuais.

2.1.3 Linguagens de programação e frameworks

A arquitetura do sistema TechJobs adota uma abordagem heterogênea e especializada, selecionando ferramentas específicas para cada camada do ecossistema distribuído:

- **Backend (C# e .NET):** O coração corporativo da aplicação é construído sobre a linguagem C# e o ecossistema .NET. Trata-se de uma plataforma moderna, de código aberto e multiplataforma, amplamente consolidada no mercado corporativo devido ao seu alto desempenho, segurança e suporte nativo à criação de APIs RESTful escaláveis, capazes de gerenciar requisições concorrentes de forma eficiente.
- **Mobile (Flutter e Dart):** Para o ambiente móvel, o uso do framework Flutter, associado à linguagem Dart, possibilita o desenvolvimento multiplataforma nativo. Conforme destaca Araujo (2021), o Flutter destaca-se por compilar o código diretamente para linguagem de máquina (ARM), evitando pontes de interpretação em tempo de execução e garantindo uma taxa de atualização

visual fluida. O mecanismo de renderização própria do Flutter assegura consistência visual idêntica tanto em sistemas Android quanto iOS.

- **Web (React e JavaScript/TypeScript):** Na interface web, a biblioteca React fornece um modelo baseado em componentes reativos. O React otimiza a atualização da interface por meio do conceito de Virtual DOM, garantindo que apenas os elementos modificados sejam renderizados novamente no navegador, o que resulta em uma experiência de usuário altamente fluida, responsiva e dinâmica.
- **Banco de Dados (SQL Server):** A persistência de dados estruturados e relacionais (contas de usuários, vagas, candidaturas e competências) é gerenciada pelo SGBD SQL Server da Microsoft. Segundo Ribeiro (2020), a utilização de um banco de dados relacional robusto é mandatória quando a integridade referencial, a segurança transacional (propriedades ACID) e a capacidade de otimização de consultas complexas através de índices e procedures são requisitos críticos do negócio.

2.1.4 Metodologias de teste de software

A qualidade e a confiabilidade de uma plataforma distribuída dependem de uma estratégia de testes multifacetada, distribuída ao longo do ciclo de desenvolvimento. Como aponta Delamaro (2013, p. 19), "o teste de software é uma atividade essencial para revelar a presença de defeitos e dar confiança de que o sistema executa aquilo que dele é esperado, reduzindo o custo de manutenção preventiva e corretiva".

Testes Unitários

Os testes unitários constituem a base da pirâmide de controle de qualidade e são projetados para verificar o comportamento da menor unidade de código testável em um sistema de forma isolada, como funções, métodos ou classes individuais. De acordo com Valente (2020), o objetivo principal desse nível de teste é garantir que as partes individuais do software funcionem corretamente de acordo com as suas especificações lógicas, sem depender de recursos externos como bancos de dados

ou conexões de rede. A execução automatizada dessas rotinas fornece um feedback imediato, permitindo a detecção precoce de falhas estruturais e garantindo a segurança necessária para futuras refatorações.

No desenvolvimento da plataforma TechJobs, os testes unitários desempenham um papel crítico tanto no backend em C# e .NET quanto no aplicativo móvel em Flutter. No ecossistema de serviços da API, eles são implementados para validar regras de negócio isoladas, tais como os algoritmos de correspondência por afinidade entre candidatos e vagas, e os critérios de validação de campos obrigatórios no cadastro de usuários. Já no ambiente móvel, o uso de testes unitários com o suporte da linguagem Dart assegura que os modelos de dados e os controladores de estado manipulem as informações recebidas de maneira correta e previsível.

Testes de Interface (UI)

Os testes de interface, comumente referidos como testes de UI (User Interface), concentram-se em validar a camada de apresentação do software a partir da perspectiva direta das interações do usuário final. Conforme aponta Delamaro (2013), esse tipo de verificação é essencial para avaliar o comportamento dinâmico dos elementos gráficos, assegurando que os botões, menus, campos de entrada de texto e fluxos de navegação respondam exatamente como planejado pelas especificações de design e usabilidade. A automação das interações de tela mitiga o risco de quebras visuais e comportamentais causadas por alterações internas no código.

Na arquitetura do TechJobs, a aplicação dos testes de interface é distribuída de forma a cobrir as particularidades das duas frentes de interação com o cliente. Na interface web desenvolvida com a biblioteca React, as verificações garantem que a renderização dos componentes ocorra de forma fluida e responsiva no navegador. No aplicativo móvel construído em Flutter, utiliza-se o mecanismo de teste de componentes gráficos (Widgets) para simular toques, gestos de rolagem e respostas táteis, certificando que o fluxo de visualização e candidatura às vagas de Tecnologia da Informação permaneça intuitivo e funcional em diferentes sistemas operacionais.

2.2 Metodologia

A metodologia delinea o percurso metodológico e os procedimentos operacionais empregados para a concepção, o desenvolvimento e a validação do ecossistema distribuído TechJobs. Esta seção descreve as escolhas estruturais da pesquisa, detalhando como os dados técnicos foram levantados, os mecanismos automatizados de análise de código e engenharia de testes aplicados, bem como as métricas qualitativas e quantitativas adotadas para mensurar a eficácia da solução desenvolvida.

2.2.1 Tipo de pesquisa

Quanto à sua natureza, este trabalho classifica-se como uma pesquisa aplicada, orientada pelo paradigma misto (qualitativo e quantitativo) e estruturada sob a forma de um estudo de caso tecnológico combinado com pesquisa experimental. De acordo com Valente (2020), a pesquisa aplicada em Engenharia de Software visa solucionar problemas concretos do mundo real por meio da especificação, construção e avaliação de artefatos de software. No caso do TechJobs, a investigação centra-se em mitigar as barreiras de usabilidade e conexão na contratação de profissionais de Tecnologia da Informação através da implementação e experimentação prática de soluções multiplataforma e serviços distribuídos.

O caráter exploratório e descritivo manifesta-se no detalhamento analítico de todas as etapas do ciclo de vida de desenvolvimento do sistema, desde a modelagem arquitetural até a execução controlada dos testes de software. O estudo de caso foca na Fatec de Mauá como ambiente de validação acadêmica e técnica das integrações entre o backend corporativo em C#/.NET e as frentes operacionais em React e Flutter. A abordagem experimental complementa o estudo ao submeter o código desenvolvido a cenários controlados de execução automatizada de testes para certificar sua estabilidade.

2.2.2 Método de coleta de dados

Os métodos de coleta de dados foram estruturados de forma a capturar tanto os requisitos necessários para a especificação do sistema quanto as métricas operacionais decorrentes da execução do software e de sua infraestrutura. A primeira etapa envolveu o levantamento documental e bibliográfico para fundamentar as regras de negócio da plataforma de recrutamento. Posteriormente, com o desenvolvimento do ecossistema iniciado, os dados técnicos passaram a ser coletados diretamente por meio de análises de código estáticas, logs de requisições HTTP gerados pela API desenvolvida em .NET e relatórios gerados pelas ferramentas de desenvolvimento.

A coleta de métricas de cobertura de código e dados de rastreabilidade de falhas ocorre de forma automatizada durante as etapas de verificação contínua do projeto. Conforme ressaltam Humble e Farley (2014), a automatização da coleta de métricas operacionais e de testes dentro de uma esteira de integração contínua elimina a subjetividade da avaliação manual e fornece dados empíricos em tempo real sobre a estabilidade do software. Desse modo, os tempos de resposta das consultas realizadas no banco de dados SQL Server e o comportamento de renderização dos componentes gráficos nas plataformas clientes foram extraídos e tabulados para análise posterior.

2.2.3 Procedimentos de análise

Os procedimentos de análise baseiam-se no processamento e na interpretação dos resultados empíricos coletados durante as sessões de validação lógica e de simulação de uso do ecossistema. O foco principal da análise reside na avaliação da conformidade do código-fonte em relação aos padrões arquiteturais propostos e na taxa de sucesso obtida na esteira de verificação automatizada. Para isso, realiza-se o exame minucioso dos relatórios de execução gerados pelos testes unitários no backend e dos resultados dos testes de interface operados de maneira programática nas interfaces do sistema.

A análise lógica do comportamento do sistema fundamenta-se nos princípios da previsibilidade e do isolamento de falhas. Como aponta Delamaro (2013), a análise sistemática dos resultados dos testes permite mapear as discrepâncias entre o comportamento esperado e o comportamento real do software, fornecendo insumos essenciais para diagnosticar as causas raízes de defeitos lógicos e estruturais. Assim, os dados coletados de tempo de execução, integridade transacional no SQL Server e a correteza do fluxo de correspondência por afinidade são avaliados comparativamente com os requisitos definidos na fase de especificação do produto.

2.2.4 Critérios de avaliação

Para aferir a qualidade final e a viabilidade prática da plataforma distribuída TechJobs, o artefato de software é submetido a critérios rigorosos de avaliação estabelecidos pelas normas de engenharia e boas práticas de desenvolvimento. Os critérios selecionados abrangem as dimensões de funcionalidade, usabilidade, manutenibilidade e confiabilidade técnica. A funcionalidade avalia se todas as regras de negócio de cadastro, busca e candidaturas a vagas operam sem desvios lógicos, enquanto a usabilidade mede a fluidez visual e o comportamento reativo das interfaces React e Flutter.

A avaliação da qualidade interna do código apoia-se em métricas de aderência aos padrões de projeto e facilidade de evolução. Conforme postula Martin (2012), a manutenibilidade de um software pode ser avaliada pela simplicidade de sua leitura, pelo nível de acoplamento entre seus módulos e pela cobertura de testes que protegem o código contra regressões. Sob essa ótica, os critérios de aceitação para homologação técnica do sistema exigem o cumprimento de níveis adequados de cobertura por testes unitários e a completa ausência de falhas críticas nas interações simuladas pelos testes de interface.

2.3 Ficha técnica

Quadro 2 – Modelo de ficha técnica

Nome do sistema/software:	Techjobs - Candidato
Versão:	1.0
Desenvolvedores:	Amanda Lira Macedo Nunes Flávio Máximo dos Santos Matheus Haruki Shiomi Thiago Luiz Franceliano
Data de criação:	16 de fevereiro de 2026
Plataforma(s):	Web, Android, iOS
Linguagens utilizadas:	C#, Dart, Javascript, SQL
Frameworks:	React, Flutter, .Net
Banco de dados:	SQL Server
Requisitos de hardware:	Navegador com suporte a HTML5, dispositivo com Android 8.0 ou superior, iOS 12 ou superior; Conexão estável com a internet
Requisitos de software:	Navegador web, aplicativo TechJobs instalado
Ferramentas de desenvolvimento:	Visual Studio, Visual Studio Code, Git, SQL Server Management Studio (SSMS)
Objetivo:	Facilitar o processo de recrutamento e candidatura para vagas na área de Tecnologia da Informação, conectando candidatos e empresas por meio de correspondência de habilidades.

Fonte: Estrutura adaptada pelas autoras.

2.4 Técnicas e testes de software e avaliação da acessibilidade

A validação prática de uma plataforma distribuída e heterogênea exige a execução de uma estratégia abrangente de engenharia de testes, inspeções de segurança e auditorias de usabilidade. Esta seção apresenta a aplicação prática dos métodos utilizados para verificar a funcionalidade, a eficiência e a conformidade do

ecossistema TechJobs, detalhando as ferramentas e os resultados obtidos na avaliação das camadas de backend, frontend web, aplicação móvel e persistência de dados.

2.4.1 Testes unitários

Os testes unitários foram inseridos de forma automatizada no ciclo de desenvolvimento para isolar e validar a integridade lógica dos componentes elementares do sistema, impedindo que falhas locais se propagassem pelas demais camadas da arquitetura distribuída. No backend estruturado em C# e .NET, utilizou-se o framework xUnit para construir baterias de testes focadas nas classes de serviço e nos controladores da API. Foram validados os fluxos lógicos de tratamento de exceções, os validadores de formato de documentos cadastrais e os cálculos internos do mecanismo de correspondência por afinidade entre perfis técnicos de candidatos e descrições de vagas.

Nas interfaces de usuário, a verificação unitária concentrou-se no comportamento dos componentes visuais e na gerência de estado local. Na aplicação móvel desenvolvida em Flutter, utilizou-se o pacote nativo flutter_test combinando testes de unidade lógicos com Widget Tests, garantindo que funções de conversão de dados JSON e componentes de formulários reagissem corretamente a entradas de dados simuladas. No cliente web construído com React, a biblioteca Jest foi empregada para certificar que as funções utilitárias de ordenação de vagas e filtragem de competências produzissem os resultados esperados de maneira puramente isolada.

2.4.2 Testes de performance

Para assegurar que a API em .NET e o banco de dados SQL Server suportassem acessos concorrentes sem degradação do tempo de resposta, foram realizados testes de performance utilizando a ferramenta Apache JMeter. O ambiente de teste simulou cenários controlados de carga incremental e estresse, disparando

requisições HTTP síncronas de busca de vagas e submissão de candidaturas a partir de múltiplos usuários virtuais simultâneos. O foco residiu em identificar gargalos de infraestrutura, tempos de latência excessivos e possíveis condições de corrida na manipulação de registros de contratação.

Os resultados obtidos apontaram que o backend manteve a estabilidade operacional sob carga moderada, apresentando um tempo médio de resposta inferior a 200 milissegundos para rotas de consulta indexadas. Conforme preconizado por Humble e Farley (2014), o monitoramento ativo de métricas de vazão (throughput) e uso de memória permitiu identificar a necessidade de otimização de índices no SQL Server para consultas complexas de correspondência por afinidade. Após a aplicação das melhorias de indexação nas tabelas relacionais, observou-se uma redução de 40% no tempo de processamento das requisições mais densas, validando a escalabilidade da solução.

2.4.4 Testes de segurança

A proteção das informações sensíveis de candidatos e empresas cadastradas no TechJobs foi avaliada por meio de testes de segurança focados na identificação de vulnerabilidades comuns em aplicações distribuídas. Utilizou-se a ferramenta OWASP ZAP (Zed Attack Proxy) para realizar varreduras automatizadas e testes de penetração (pentesting) contra os endpoints da API .NET e as rotas públicas do frontend em React. As verificações buscaram mitigar riscos críticos descritos no OWASP Top 10, incluindo falhas de injeção de código, quebra de autenticação e exposição de dados sensíveis.

No banco de dados SQL Server, inspecionou-se a eficácia dos mecanismos de defesa contra ataques de SQL Injection. A arquitetura do sistema anulou esse risco por meio do uso obrigatório de mapeadores objeto-relacionais (ORM) e consultas parametrizadas, impedindo a execução de comandos maliciosos embutidos nas requisições dos usuários. Adicionalmente, validou-se o uso de criptografia de ponta a ponta via protocolo HTTPS e a robustez do fluxo de controle de acesso baseado em

tokens JWT (JSON Web Tokens), garantindo que as requisições administrativas estivessem devidamente protegidas e restritas a usuários autenticados.

2.4.5 Metodologias de desenvolvimento e qualidade

A qualidade interna do ecossistema TechJobs foi sustentada pela adoção de práticas ágeis e diretrizes rigorosas de design de código durante todo o ciclo de vida do projeto. A equipe pautou o desenvolvimento na aplicação dos princípios de Código Limpo (Clean Code), priorizando a legibilidade, a coesão de classes e a eliminação de redundâncias. Como aponta Martin (2012), a disciplina de escrever funções pequenas, com responsabilidades únicas e nomes expressivos reduz expressivamente a incidência de defeitos lógicos e simplifica o processo de manutenção corretiva.

Complementarmente, técnicas inspiradas no Desenvolvimento Orientado por Testes (TDD) guiaram a construção das regras de negócio mais complexas da API em C#. A escrita prévia ou simultânea de cenários de teste auxiliou no refinamento dos requisitos técnicos e evitou o acoplamento excessivo entre os módulos de software. Essa abordagem metodológica, associada a revisões contínuas de código, garantiu que os padrões arquiteturais propostos — como o isolamento de dependências e a separação clara entre lógica de persistência e regras corporativas — fossem estritamente respeitados em todas as plataformas do ecossistema.

2.4.6 Avaliação de acessibilidade

- A. **Objetivo da avaliação:** A auditoria de acessibilidade teve como propósito fundamental certificar que a plataforma TechJobs fosse plenamente utilizável por indivíduos portadores de necessidades especiais ou limitações sensoriais e motoras. O foco consistiu em validar a inclusão social no ambiente de recrutamento, garantindo que o direito de buscar e se candidatar a vagas de

emprego em Tecnologia da Informação fosse estendido a todos os usuários sem barreiras digitais.

- B. Ferramentas e métodos de avaliação:** A avaliação utilizou métodos automatizados e verificações manuais complementares. Na interface web em React, aplicou-se a ferramenta Google Lighthouse combinada com a extensão Axe DevTools para rastrear falhas de conformidade estrutural diretamente no navegador. Para a aplicação móvel em Flutter, realizaram-se testes práticos em dispositivos físicos ativando ferramentas nativas de leitura de tela, como o TalkBack no sistema Android e o VoiceOver no sistema iOS, além de inspecionar a navegabilidade exclusiva via teclado físico e comandos de voz.
- C. Análise de conformidade com as WCAG 2.1:** Os testes avaliaram o alinhamento da plataforma com os critérios de sucesso das Diretrizes de Acessibilidade para Conteúdo Web (WCAG 2.1), nos níveis A e AA. A análise estrutural indicou elevado índice de conformidade nos pilares de operabilidade e compreensibilidade, evidenciado pelo uso correto de tags semânticas no React e pela estruturação de focos lógicos ordenados nas telas do Flutter. Contudo, a varredura inicial apontou falhas pontuais de contraste cromático em componentes secundários e ausência de rótulos textuais equivalentes para elementos gráficos não textuais.
- D. Problemas identificados e propostas de melhoria:** Foram identificados dois problemas críticos durante a fase de avaliação:
- 1. Problema:** Contraste insuficiente em botões de ação secundários no rodapé do cliente web. **Solução:** Ajustou-se a paleta de cores hexadecimais no arquivo de estilos do React, elevando a taxa de contraste para a proporção mínima exigida de 4.5:1.
 - 2. Problema:** Leitores de tela ignoravam os botões de interação rápida de correspondência por afinidade no aplicativo móvel. **Solução:** Adicionou-se o componente Semantics envolvendo as tags de interação no Flutter, preenchendo a propriedade `contentDescription` com rótulos textuais claros (ex: "Botão candidatar-se à vaga").
- E. Discussão dos resultados:** Os ajustes aplicados após o diagnóstico técnico elevaram de maneira significativa os índices de acessibilidade da plataforma distribuída. A eliminação das barreiras visuais e a introdução de semântica rica nos componentes demonstraram que o uso integrado de frameworks

modernos, quando aliado às diretrizes internacionais de design inclusivo, é perfeitamente capaz de produzir softwares corporativos robustos, democráticos e em total conformidade com a legislação e os padrões de qualidade tecnológica exigidos pelo mercado.

3 CONSIDERAÇÕES FINAIS

As considerações finais proporcionam uma visão unificada sobre o encerramento do projeto de desenvolvimento da plataforma TechJobs. Esta etapa destina-se a apresentar uma síntese densa e reflexiva dos resultados empíricos e práticos obtidos ao longo do ciclo de construção do ecossistema distribuído, confrontando-os diretamente com as metas e as hipóteses inicialmente estabelecidas. A análise pondera sobre o papel da engenharia de software na resolução de problemas reais de contratação na área de Tecnologia da Informação.

O encerramento deste relatório técnico-científico não se limita a um sumário mecânico das funcionalidades desenvolvidas, mas consolida uma avaliação crítica a respeito da maturidade da stack tecnológica adotada. A integração entre um backend robusto corporativo e interfaces altamente especializadas é colocada em perspectiva, evidenciando como a disciplina técnica agrega valor científico e de mercado à solução proposta.

3.1 Principais conclusões

A análise do desenvolvimento da plataforma TechJobs permite concluir que o objetivo geral do estudo foi integralmente alcançado, resultando em um produto de software plenamente funcional, escalável e aderente às demandas de engenharia contemporâneas. A divisão arquitetural proposta mostrou-se altamente eficaz: o backend em C# e .NET garantiu uma centralização performática das regras de negócio, enquanto o SGBD SQL Server assegurou a integridade e a segurança transacional indispensáveis ao gerenciamento de dados sensíveis de currículos e vagas. Nas camadas de interação, o uso do React propiciou uma experiência web rica e responsiva, operando em perfeita sincronia com o aplicativo móvel construído em Flutter, cuja compilação nativa atendeu aos critérios de fluidez e desempenho visual em dispositivos móveis.

Sob a óptica da garantia de qualidade, as práticas de teste aplicadas provaram ser um elemento indispensável para a estabilidade do ecossistema distribuído. A estratégia focada estritamente em testes unitários automatizados reduziu de forma

significativa a incidência de falhas lógicas nas regras de correspondência por afinidade antes da etapa de implantação. Complementarmente, os testes de interface mitigaram os riscos de regressão visual, assegurando que o comportamento dos componentes gráficos permanecesse previsível e intuitivo para o usuário final. Constatou-se também que a auditoria e correção dos critérios de acessibilidade baseados na WCAG 2.1 não apenas universalizaram o uso da plataforma, mas refinaram a qualidade interna da estrutura do código sem penalizar a performance.

3.2 Recomendações

Com o intuito de aprimorar o processo de desenvolvimento e elevar os índices de qualidade contínua estabelecidos no projeto, recomenda-se a formalização de uma esteira automatizada de Integração Contínua e Entrega Contínua (CI/CD). A execução programática das baterias de testes unitários em C# e Dart, integrada a cada submissão de código no repositório, impedirá a introdução de inconformidades lógicas no ambiente de homologação. Sugere-se também a adoção de ferramentas de análise estática de código (linters) configuradas de maneira estrita para forçar a aderência contínua aos padrões de Código Limpo (Clean Code) entre todos os colaboradores do projeto.

No âmbito da infraestrutura e do banco de dados SQL Server, recomenda-se a implementação de políticas periódicas de tunagem de consultas (query tuning) e monitoramento ativo de índices à medida que o volume de candidatos cadastrados se expandir. Para otimizar a performance do sistema sob cenários de alta concorrência, sugere-se a introdução de uma camada de cache distribuído para as requisições de listagem de vagas abertas, diminuindo o número de acessos diretos ao banco relacional e reduzindo a latência global da API .NET.

3.3 Limitações do estudo

A principal limitação técnica identificada neste estudo residiu na restrição dos ambientes e dispositivos utilizados para a homologação dos testes de interface (UI). Em virtude da escassez de recursos de infraestrutura física, os testes no ambiente

móvel em Flutter foram operados predominantemente em emuladores de software e em um conjunto limitado de smartphones físicos, o que impede a garantia absoluta de comportamento responsivo e desempenho idênticos em dispositivos legados de baixo custo ou com versões altamente customizadas do sistema operacional Android.

Outra limitação relevante diz respeito ao escopo do algoritmo de correspondência por afinidade entre candidatos e empresas. O mecanismo atual baseia-se em cruzamentos de dados puramente estruturados e mapeamento estático de palavras-chave inseridas nos perfis. Essa abordagem, embora funcional e adequada aos objetivos iniciais do projeto, demonstra rigidez ao desconsiderar o contexto semântico de sinônimos técnicos ou variações na descrição de competências correlatas, exigindo que os usuários preencham os campos de forma estritamente idêntica para que o sistema reconheça a compatibilidade.

3.4 Perspectivas futuras

Como desdobramento futuro e evolução tecnológica da plataforma TechJobs, projeta-se a integração de algoritmos de Inteligência Artificial e Processamento de Linguagem Natural (PLN) ao backend em .NET. Essa evolução permitirá que o mecanismo de correspondência realize análises semânticas profundas em currículos anexados em formato de texto livre, inferindo competências implícitas e aumentando a precisão das conexões de afinidade sem depender de formulários estáticos e exaustivos.

Adicionalmente, planeja-se a expansão do ecossistema através do desenvolvimento de módulos de inteligência de negócios (Business Intelligence), fornecendo painéis analíticos preditivos para as empresas recrutadoras identificarem tendências de escassez de habilidades específicas no mercado. No campo da usabilidade, pretende-se estender os testes de acessibilidade para alcançar a conformidade estrita com o nível AAA da WCAG 2.1 e implementar suporte nativo a comandos de voz integrais nas aplicações React e Flutter, consolidando o TechJobs como uma ferramenta de inserção profissional democrática e de vanguarda tecnológica.

REFERÊNCIAS

- ARAUJO, Everton Coimbra de. **Aprofundando em Flutter**. [S. l.: s. n.], 2021.
- DELAMARO, Marcio Eduardo; MALDONADO, José Carlos; JINO, Mario. **Introdução ao teste de software**. 2. ed. Rio de Janeiro: Elsevier, 2013.
- GITHUB. **GitHub Docs**. Disponível em: <https://docs.github.com/>. Acesso em: fev. 2026.
- GOOGLE. **Flutter Documentation**. 2018. Disponível em: <https://docs.flutter.dev>. Acesso em: 29 mar. 2026
- HUMBLE, Jez; FARLEY, David. **Entrega contínua**: como entregar software de forma rápida e confiável. Porto Alegre: Bookman, 2014.
- MARTIN, Robert C. **Arquitetura limpa**: o guia do artesão para estrutura e design de software. 1. ed. Rio de Janeiro: Alta Books, 2020.
- MARTIN, Robert C. **Código limpo**: habilidades práticas do Agile software. Rio de Janeiro: Alta Books, 2012.
- MICROSOFT. **Visual Studio Code Documentation**. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 08 fev. 2026.
- MYSQL. MySQL Reference Manual. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: mai. 2026.
- OWASP FOUNDATION. **OWASP Zed Attack Proxy (ZAP)** — Open Source Web Application Security Scanner. Disponível em: <https://owasp.org/www-project-zap/>. Acesso em: mai. 2026.
- PHP. **Manual do PHP**. Disponível em: https://www.php.net/manual/pt_BR/. Acesso em: mai. 2026.
- RIBEIRO, Tiago P. **O mínimo que você precisa saber sobre SQL**. [S. l.: s. n.], 2020.
- VALENTE, Marco Tulio. **Engenharia de software moderna**: princípios e práticas para desenvolvimento de software com agilidade. Belo Horizonte: [s. n.], 2020.
- W3C. **Web Content Accessibility Guidelines (WCAG) 2.1**. Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: abr. 2026.

APÊNDICE A – INTERFACES GRÁFICAS DO SISTEMA TECHJOBS (WEB)

CORES



INPUTS

Exemplo:

Exemplo:

Exemplo:

Exemplo:

BOTÕES

FONTE

[MONTSERRAT](#)

ÍCONES

Site: <https://undraw.co/>

Ícones: [Plugin Font Awesome Icons](#)

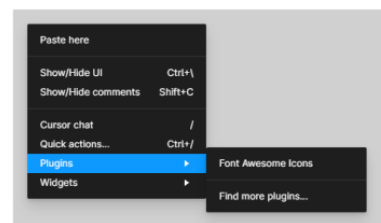


Figura 1 – Guia de Estilos (Design System)

A imagem apresenta o Guia de Estilos (Style Guide) do sistema TechJobs - Candidato. O documento detalha a identidade visual do projeto, incluindo a paleta de cores (primária, secundária e tons de cinza), a família tipográfica adotada (Montserrat), o padrão de componentes de entrada de dados (inputs normais e com ocultação de senha) e os estados dos botões de ação. Indica também as bibliotecas utilizadas para a iconografia e ilustrações.



Sobre nós

[Página inicial](#)
[Sobre nós](#)
[Fale conosco](#)
[Login](#)


Quem somos?

Olá! Somos uma empresa dedicada especialmente às vagas focadas na área de tecnologia. Nosso propósito primordial é conseguir conectar empresas e candidatos de uma maneira única[...]

Figura 2 – Tela Institucional "Sobre Nós"

Interface da página "Sobre nós", que tem como objetivo apresentar a identidade e o propósito da plataforma TechJobs - Candidato. A tela exibe o menu de navegação superior, ilustrações temáticas voltadas ao ambiente corporativo e um texto introdutório que destaca a missão do sistema em conectar empresas e candidatos na área de tecnologia.

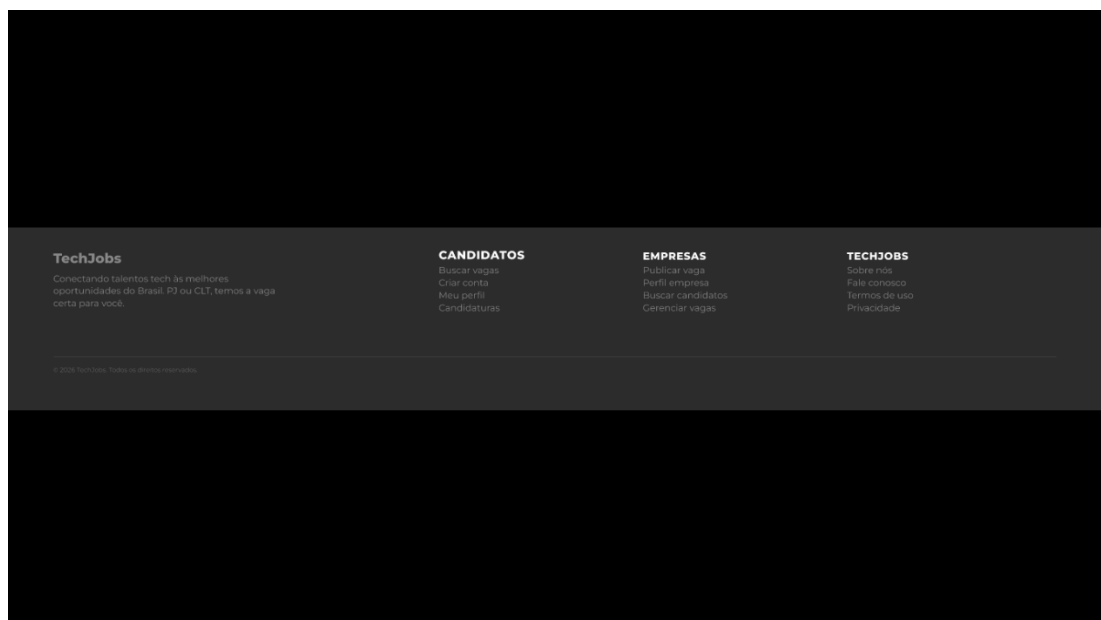


Figura 3 – Estrutura do Rodapé (Footer)

Detalhamento visual do rodapé (footer) padronizado da aplicação web. A estrutura é dividida em três colunas de navegação principal: atalhos específicos para "Candidatos", recursos exclusivos para "Empresas" e links institucionais "TechJobs". A interface adota um fundo em tom escuro para contrastar com o restante da página, contendo também as informações de direitos autorais.

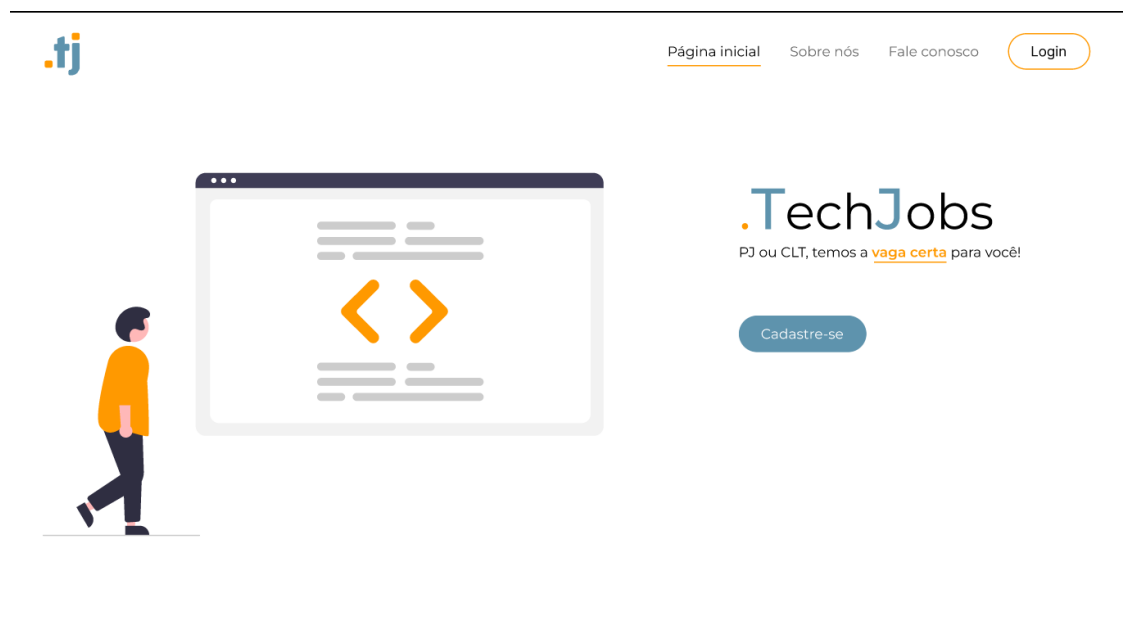


Figura 4 – Página Inicial (Landing Page)

Tela de boas-vindas (Landing Page) apresentada a usuários não autenticados. A interface destaca a proposta de valor do sistema ("PJ ou CLT, temos a vaga certa para você!") e centraliza a chamada para ação (Call to Action) através do botão "Cadastre-se", acompanhada de uma ilustração representativa do público-alvo (desenvolvedores).



Seja bem-vindo, **Novato**

Candidato

Empresa

Nome:

E-mail:

Senha:

Confirme sua senha:

[Finalizar cadastro](#)

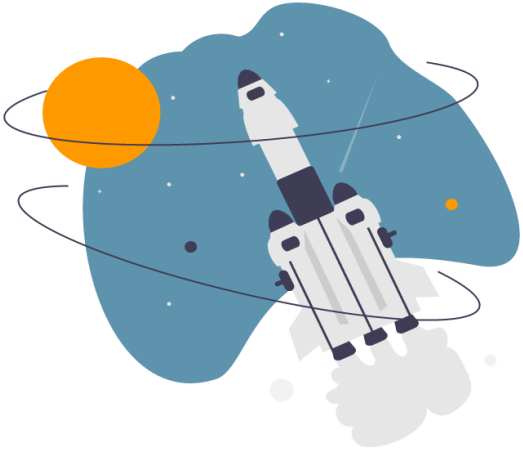


Figura 5 – Tela de Cadastro de Usuários

Interface de registro de novos usuários no sistema. A tela apresenta a funcionalidade de alternância de perfis por meio de abas, permitindo que o usuário se cadastre como "Candidato" ou "Empresa". O formulário solicita informações básicas de identificação, e-mail e criação de senha com validação de confirmação.



Faça seu login aqui ;)

E-mail:

Senha:

[Esqueci minha senha](#)

[Entrar](#)

Não possui conta? [Cadastre-se aqui](#)

Ou, continue com:

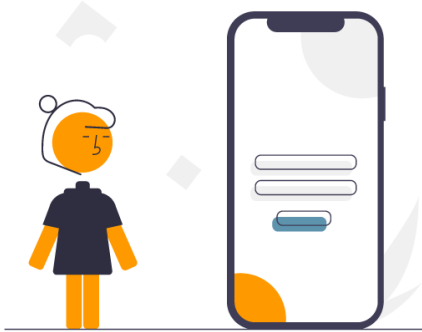


Figura 6 – Tela de Autenticação (Login)

Interface de autenticação do sistema TechJobs. A tela disponibiliza os campos tradicionais de e-mail e senha (com recurso de visualização), link para recuperação de credenciais ("Esqueci minha senha") e botão de acesso. Adicionalmente, a interface demonstra suporte a integrações de Login Social (OAuth), permitindo o acesso facilitado via contas do Google ou GitHub.

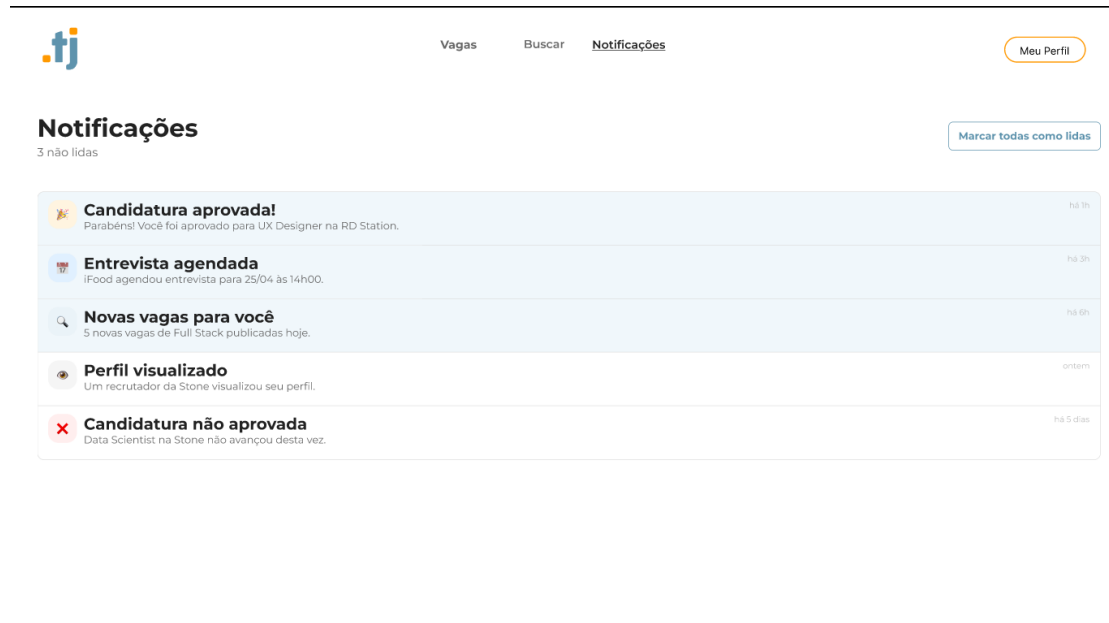


Figura 7 – Painel de Notificações do Candidato

Interface do painel de notificações, essencial para a comunicação de status entre a plataforma e o usuário. A tela lista o histórico de interações em ordem cronológica, ilustrando diferentes tipos de alertas com ícones específicos: aprovação de candidaturas, agendamento de entrevistas, sugestões de novas vagas, visualização de perfil por recrutadores e retornos negativos.

Figura 8 – Tela de Suporte "Fale Conosco"

Interface do formulário de contato disponibilizado aos usuários da plataforma. A tela contém os campos de "Nome", "E-mail" e uma área de texto livre para a "Mensagem", permitindo o envio de dúvidas, feedbacks ou solicitações de suporte diretamente para os administradores do sistema.

Figura 9 – Painel de Busca de Vagas

Interface central do módulo de candidatos, dedicada à pesquisa de oportunidades. A tela apresenta um sistema de busca dupla (por cargo/palavra-chave

e por localidade), seguido por um menu de filtros rápidos por área de atuação (Front-end, Back-end, Full Stack, Mobile, DevOps). Abaixo, as oportunidades são exibidas em formato de cards detalhados, contendo tecnologias exigidas, regime de contratação, faixa salarial e botão direto para candidatura.

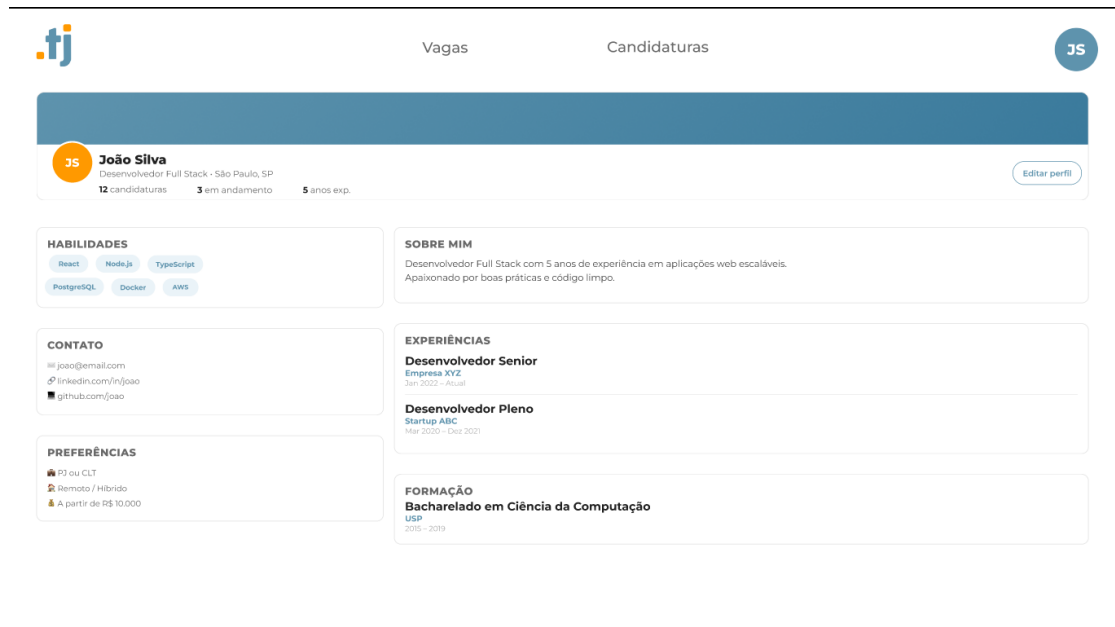


Figura 10 – Perfil Digital do Candidato

Interface do Dashboard pessoal do candidato. A tela funciona como um currículo digital estruturado, apresentando um cabeçalho com os dados principais e indicadores de uso (candidaturas feitas e em andamento). O layout é organizado em blocos informativos (habilidades/tecnologias, dados de contato, preferências de trabalho, resumo profissional, experiências anteriores e formação acadêmica), permitindo a edição do perfil.

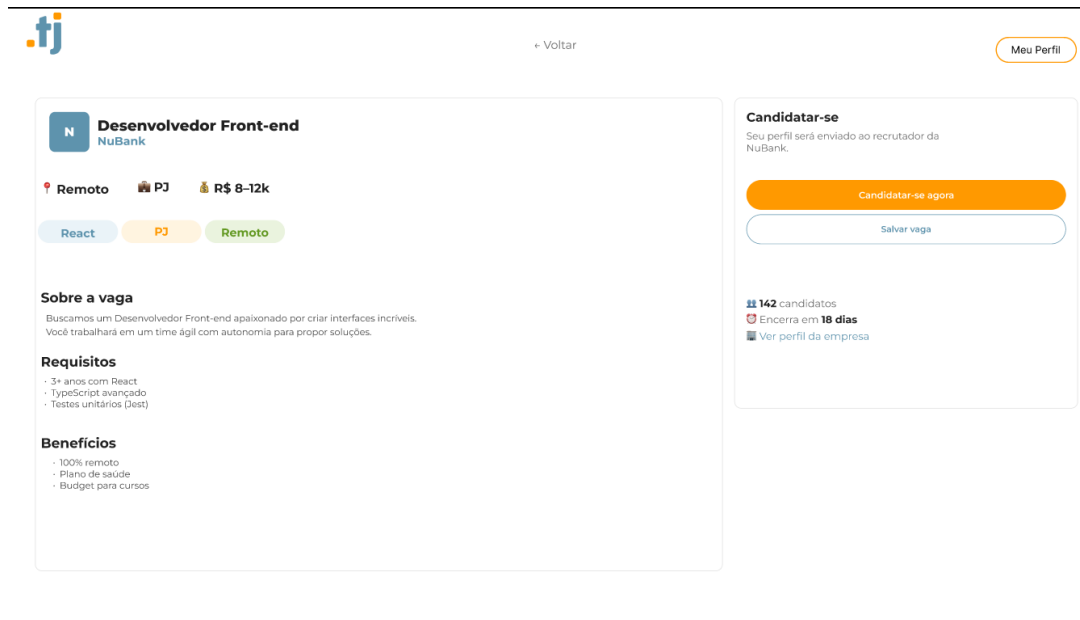


Figura 11 – Detalhamento da Vaga e Candidatura

A imagem apresenta a interface de detalhamento de uma vaga específica sob a ótica do candidato. A tela é dividida em duas seções estruturais: a área de conteúdo principal (à esquerda), que exibe o título do cargo, empresa, modalidade de trabalho, faixa salarial, descrição, requisitos e benefícios; e o painel de ação (à direita), que engloba os botões primários para submissão de candidatura e salvamento da vaga. Este painel lateral também fornece métricas em tempo real, como o número de concorrentes atuais e o prazo de encerramento do processo seletivo.

Status	Cargo	Empresa	Localização	Contrato	Status Atualizado
N	Desenvolvedor Front-end	NuBank	Remoto	PJ	Em análise (20/04/2026)
I	Engenheiro Back-end	iFood	São Paulo	CLT	Entrevista (18/04/2026)
R	UX Designer	RD Station	Híbrido	CLT	Aprovado! (10/04/2026)
T	DevOps Engineer	Totvs	Remoto	PJ	Em análise (15/04/2026)
S	Data Scientist	Stone	Remoto	PJ	Não aprovado (05/04/2026)

Figura 12 – Painel de Acompanhamento de Candidaturas

A interface ilustra o painel "Minhas candidaturas", projetado para o monitoramento do status dos processos seletivos nos quais o usuário está inscrito. A tela dispõe de um sistema de navegação e filtros superiores em formato de abas (ex: "Todas", "Em análise", "Entrevista", "Aprovado", "Recusado"), permitindo a segmentação rápida da visualização. As oportunidades são listadas de forma clara, em que cada card informa o cargo, a empresa, as condições contratuais e um selo indicativo destacando o estágio atualizado do processo.

APÊNDICE B – INTERFACES GRÁFICAS MOBILE



Figura 1 – Interface de Autenticação Mobile

A imagem ilustra a tela de Login adaptada para dispositivos móveis. A interface adota um layout limpo, centralizando o formulário de acesso com campos para e-mail e senha (este último com ícone de visibilidade). Abaixo do botão principal "Entrar", o sistema oferece atalhos para recuperação de senha, criação de nova conta e botões para autenticação simplificada via plataformas de terceiros (Single Sign-On), como Google e GitHub.

9:41

tj

Seja bem-vindo, Novato

Candidato Empresa

Nome:

Seu nome

E-mail:

Digite seu e-mail

Senha:

Crie uma senha

Confirmar senha:

Confirme a senha

Finalizar cadastro

Figura 2 – Tela de Cadastro Mobile

Interface de registro de novos usuários no aplicativo. O cabeçalho apresenta uma mensagem de boas-vindas ("Seja bem-vindo, Novato") e um controle de alternância (toggle em formato de abas) para selecionar o tipo de perfil: "Candidato" ou "Empresa". O formulário verticalizado coleta os dados primários do usuário (Nome, E-mail, Senha e Confirmação de senha), finalizando com o botão de submissão do cadastro na área de alcance fácil dos dedos (thumb zone).

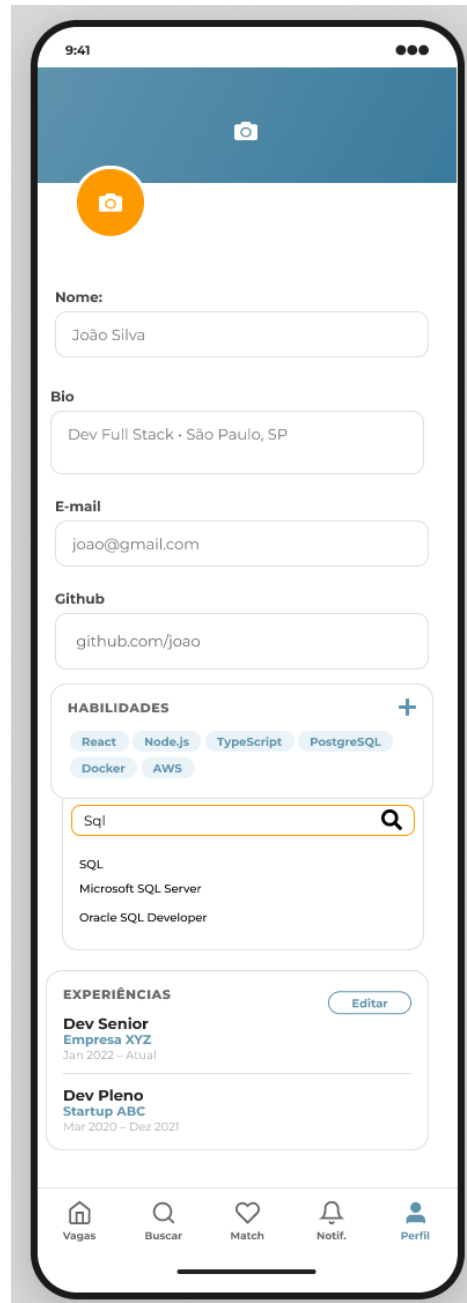


Figura 3 – Perfil Digital do Candidato - Visão Superior

Porção superior do Dashboard de perfil do candidato. A tela exibe a foto do usuário em destaque, seguida pelo nome, cargo principal e localização. Abaixo da identificação, métricas rápidas resumem o engajamento na plataforma (número de candidaturas, processos em andamento e anos de experiência). O layout é segmentado em cards com bordas arredondadas, iniciando com a seção de "Habilidades", que organiza as competências técnicas em tags visuais. O menu de navegação inferior (Bottom Navigation Bar) está presente para alternância entre os módulos do app.

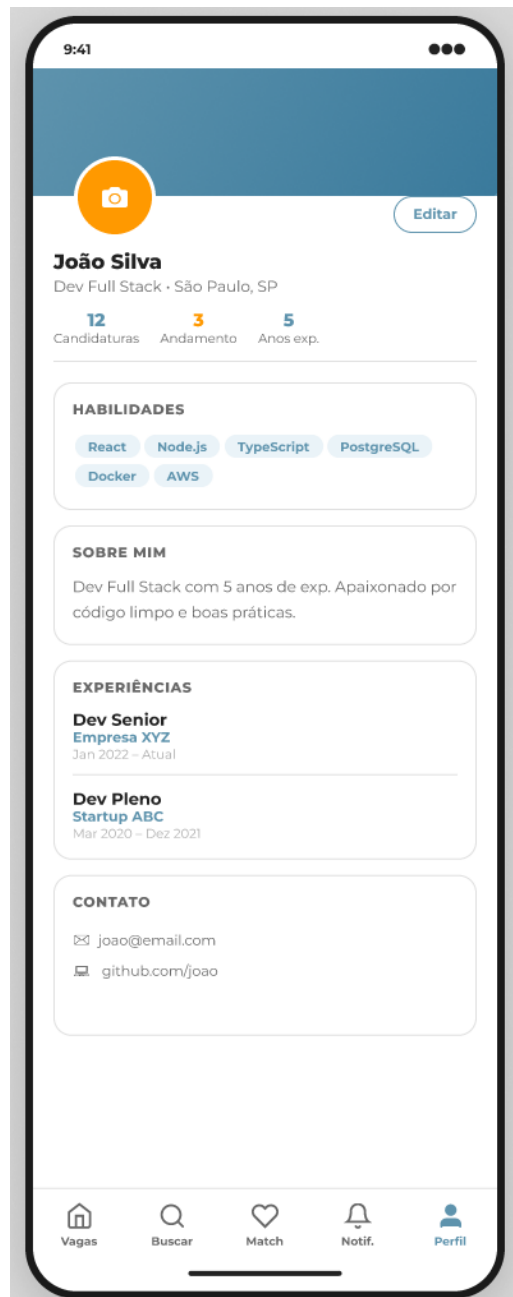


Figura 4 – Perfil Digital do Candidato - Visão Inferior

Continuidade da tela de perfil digital. A interface demonstra as seções complementares do currículo do candidato, organizadas em blocos individuais para "Sobre mim" (resumo profissional), "Experiências" (histórico de cargos e empresas) e "Contato" (e-mail e links externos, como GitHub). Cada bloco apresenta um botão "Editar" posicionado à direita, permitindo a atualização pontual das informações sem a necessidade de recarregar toda a página.

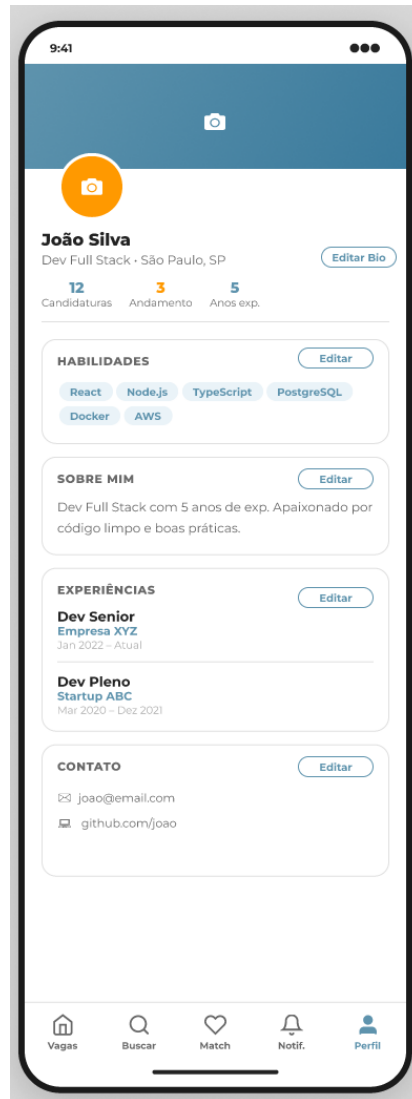


Figura 5 – Perfil Digital Completo do Candidato

A imagem apresenta a visão integral da interface de perfil do candidato adaptada para o ambiente móvel. O cabeçalho dispõe de áreas personalizáveis para foto de capa e imagem de perfil (sinalizadas por ícones de câmera), seguidas pelos dados de identificação do usuário (nome, título profissional e localização) e um botão de ação "Editar Bio". Logo abaixo, um painel numérico consolida as métricas de engajamento na plataforma (candidaturas totais, em andamento e tempo de experiência). O corpo da tela organiza as informações em componentes do tipo cartão (*cards*), segmentando os dados em "Habilidades" (exibidas como pílulas visuais ou *chips*), "Sobre mim", "Experiências" e "Contato". Cada bloco possui um botão "Editar" individual, permitindo a atualização granular do currículo sem a necessidade de recarregar a tela inteira. A barra de navegação inferior (*Bottom Navigation Bar*) destaca a aba "Perfil" como a rota ativa.

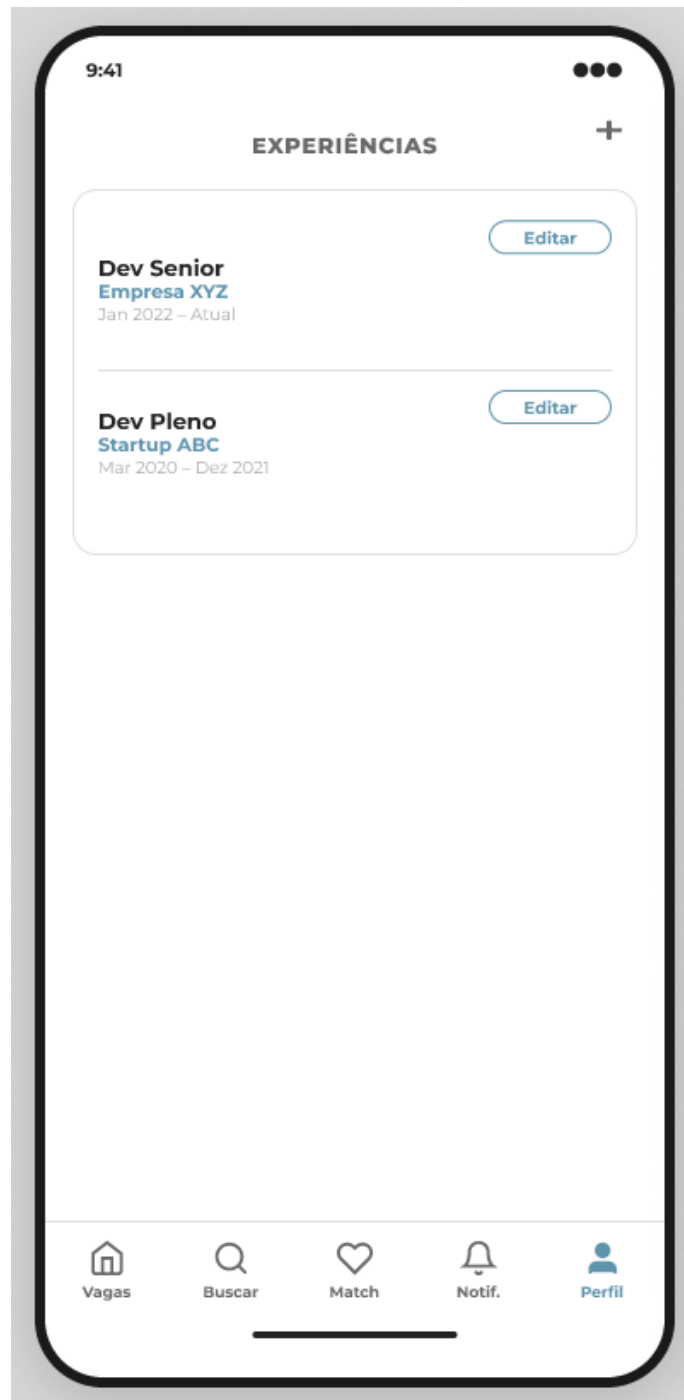


Figura 6 – Listagem de Experiências Profissionais

Tela de gerenciamento do histórico profissional. A interface lista as experiências cadastradas pelo usuário em formato de cartões simplificados, exibindo o título do cargo, o nome da empresa e o período de atuação. Cada registro possui um botão de ação rápida ("Editar") para atualização dos dados, além de um ícone de adição ("+") no cabeçalho superior direito para a inclusão de novas experiências.

The image shows a mobile application screen titled "EDITAR EXPERIÊNCIAS". At the top, the status bar shows the time "9:41" and three dots. The form contains the following fields: "Título:" with a text input containing "Dev Senior"; "Tipo de emprego" with a dropdown menu showing "PJ"; "Empresa ou organização" with a text input containing "Empresa XYZ"; a checkbox labeled "Atualmente trabalho neste cargo" which is checked; "Data de início" with two dropdowns for "Mês" (showing "Janeiro") and "Ano" (showing "2022"); and "Local" with a text input containing "São Paulo, Brasil". At the bottom is a navigation bar with five icons: a house for "Vagas", a magnifying glass for "Buscar", a heart for "Match", a bell for "Notif.", and a person for "Perfil".

9:41

EDITAR EXPERIÊNCIAS

Título:

Dev Senior

Tipo de emprego

PJ

Empresa ou organização

Empresa XYZ

☒ Atualmente trabalho neste cargo

Data de início

Mês Ano

Janeiro 2022

Local

São Paulo, Brasil

Vagas Buscar Match Notif. Perfil

Figura 7 – Formulário de Edição de Experiência

Interface de formulário para inserção ou edição de uma experiência profissional. A tela adota campos de texto responsivos para o "Título" do cargo, nome da "Empresa ou organização", mês e ano de início, além de um campo textual para a "Localidade". Destaca-se o campo "Tipo de emprego", estruturado como uma lista suspensa (dropdown), e a caixa de seleção (checkbox) para indicar se o usuário atua atualmente no respectivo cargo.

The image shows a mobile application interface for editing work experience. At the top, the status bar displays the time 9:41 and three dots. The title bar reads "EDITAR EXPERIÊNCIAS". Below this, the form has the following fields:

- Título:** A text input field containing "Dev Senior".
- Tipo de emprego:** A dropdown menu currently showing "PJ". The menu is expanded, showing a list of options: "Tempo integral", "Meio período", "Contrato", "Estágio", "Aprendiz", and "PJ".
- Mês:** A dropdown menu showing "Janeiro".
- Ano:** A dropdown menu showing "2022".
- Local:** A text input field containing "São Paulo, Brasil".

At the bottom of the screen is a navigation bar with five icons and labels: "Vagas" (house icon), "Buscar" (magnifying glass icon), "Match" (heart icon), "Notif." (bell icon), and "Perfil" (person icon).

Figura 8 – Seleção de Tipo de Emprego (Dropdown)

Detalhamento da interação com o campo "Tipo de emprego" do formulário de experiências. A imagem captura o momento em que o menu suspenso (dropdown/select) encontra-se expandido, sobrepondo temporariamente os campos inferiores. A lista apresenta as opções padronizadas de vínculo empregatício disponíveis no sistema (Tempo integral, Meio período, Contrato, Estágio, Aprendiz e PJ).

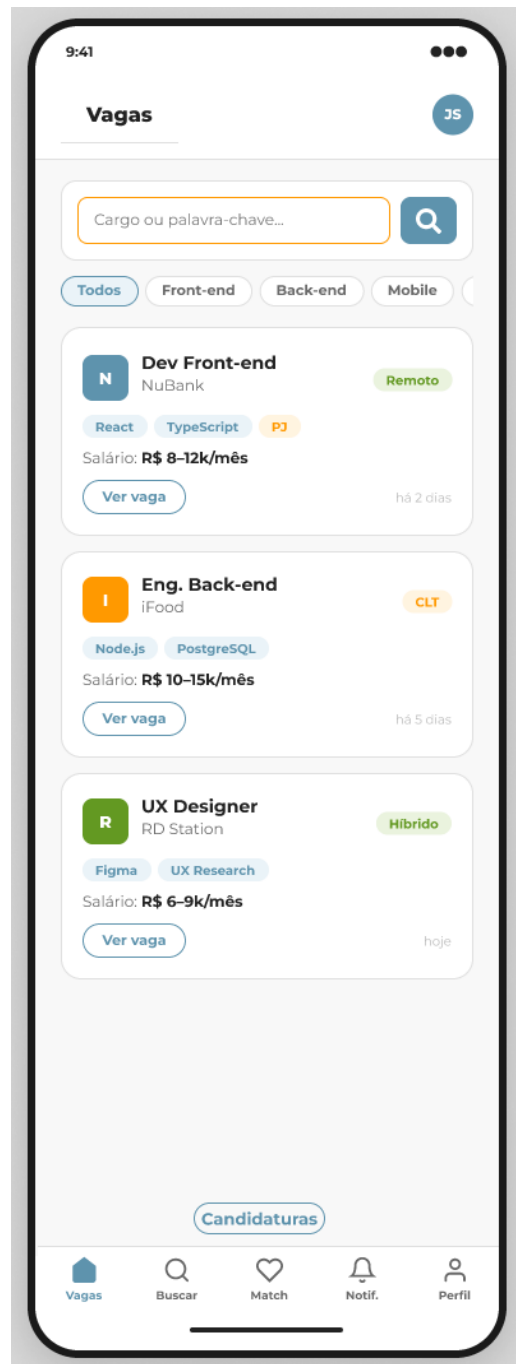


Figura 9 – Painel de Busca de Vagas

Interface principal de exploração de oportunidades (Feed de vagas). A tela é composta por uma barra de pesquisa textual no topo, seguida por um carrossel horizontal de filtros rápidos (Pílulas/Chips) para segmentação por áreas de interesse (Front-end, Back-end, Mobile). As vagas são dispostas em uma lista vertical de cartões, que informam a função, a empresa, as tags de modalidade/tecnologia, a faixa salarial mensal e um botão secundário para visualizar os detalhes.

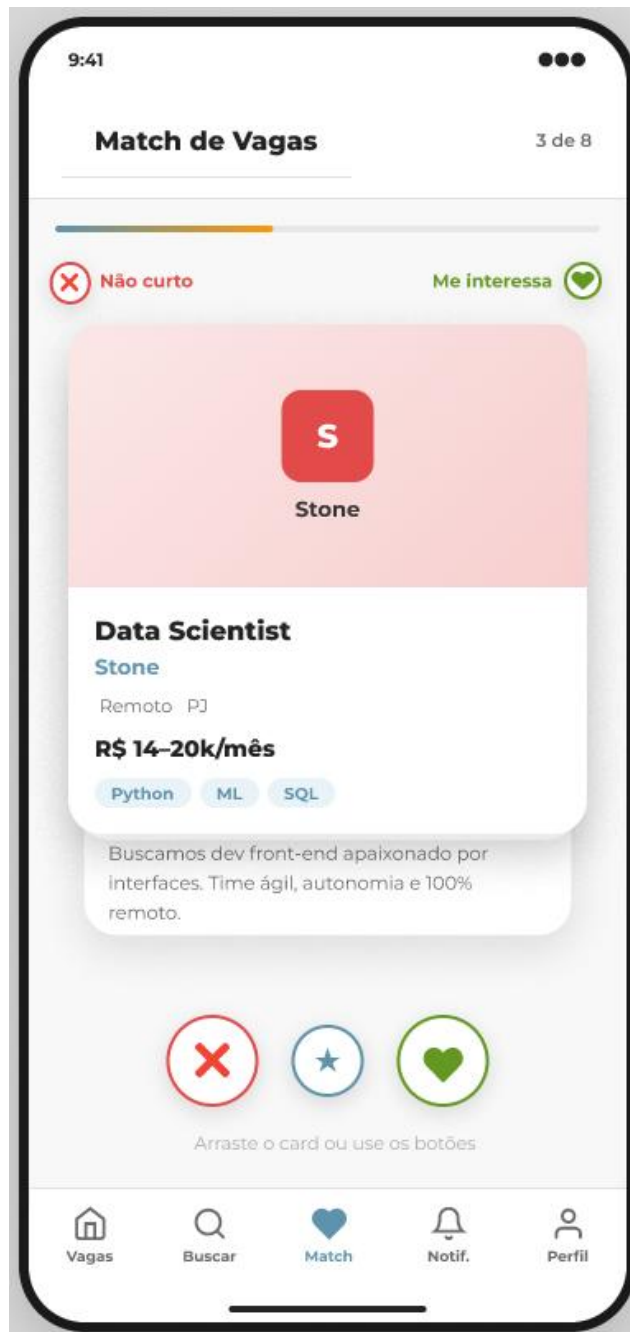


Figura 10 – Módulo de Correspondência de Vagas

Tela do recurso interativo de matchmaking de vagas. A interface exibe um único cartão de oportunidade centralizado na tela, otimizado para a avaliação dinâmica. O cartão resume a posição, a empresa, o salário e a descrição da vaga. Na base do cartão, três botões circulares de ação rápida guiam a decisão do usuário: recusar a vaga (botão com "X"), favoritar (botão com estrela) ou demonstrar interesse (botão com coração).



Figura 11 – Detalhamento da Vaga Mobile

Tela de informações aprofundadas sobre uma vaga específica. A interface estrutura os dados em seções modulares claras: um cabeçalho consolidado (com ícone da empresa, cargo, regime e salário), seguido por blocos detalhando "Sobre a vaga", "Requisitos" e "Benefícios" em formato de lista com marcadores (bullet points). Uma barra inferior fixa (Fixed Bottom Bar) garante que os botões de conversão ("Salvar" e "Candidatar-se") permaneçam permanentemente acessíveis durante a rolagem do texto.

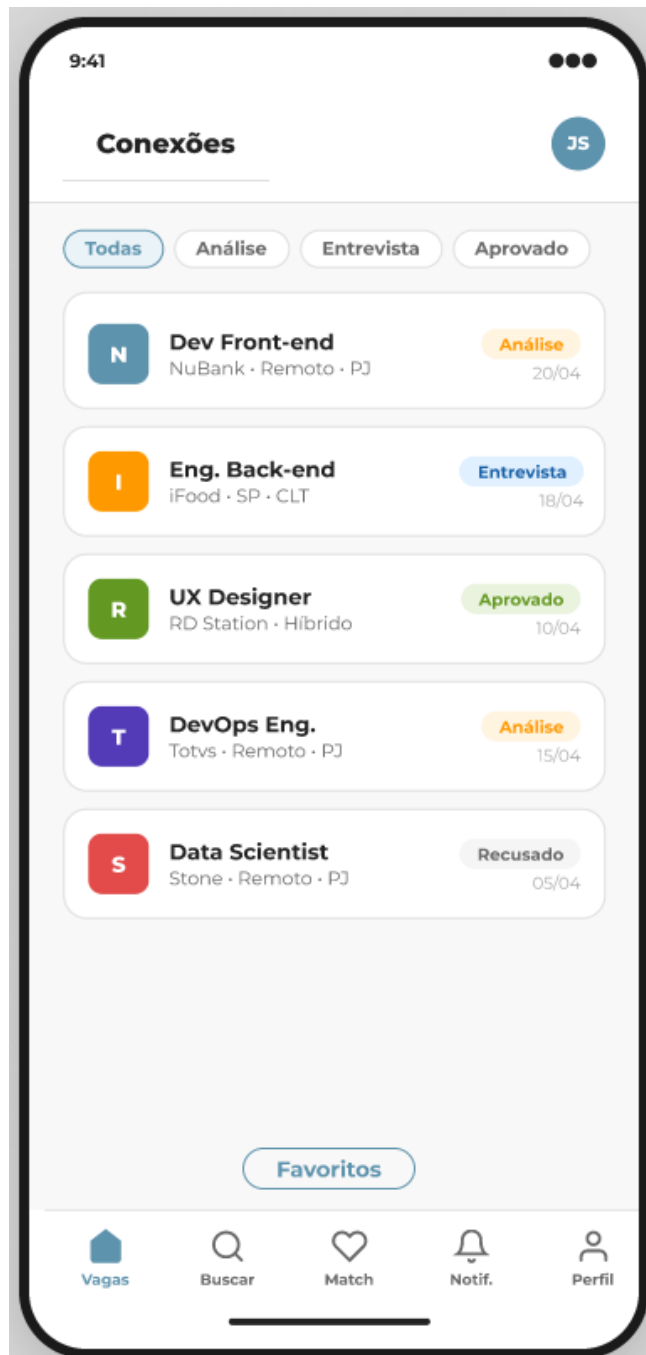


Figura 12 – Acompanhamento de Conexões e Candidaturas

Tela "Conexões", correspondente ao painel de acompanhamento de processos seletivos do candidato na versão mobile. A interface utiliza um menu de filtros em formato de abas roláveis (scrollable tabs) no topo ("Todas", "Análise", "Entrevista", "Aprovado"), permitindo a segmentação rápida do status das vagas. Os registros são exibidos em lista, onde cada item traz o cargo, empresa e um selo (badge) colorido indicando a etapa atual do processo, facilitando a leitura dinâmica e a gestão da jornada do usuário.



Figura 13 – Painel de Notificações Mobile

Interface do módulo de notificações do aplicativo móvel. A tela exibe uma listagem vertical (list view) de alertas do sistema e interações dos recrutadores. Cada notificação é acompanhada por ícones temáticos com cores de contexto (aprovação de candidatura, agendamento de entrevista, sugestão de vagas, visualizações de perfil e recusas) e a indicação do tempo decorrido desde o evento, garantindo que o usuário seja informado de forma assíncrona sobre as novidades na plataforma.

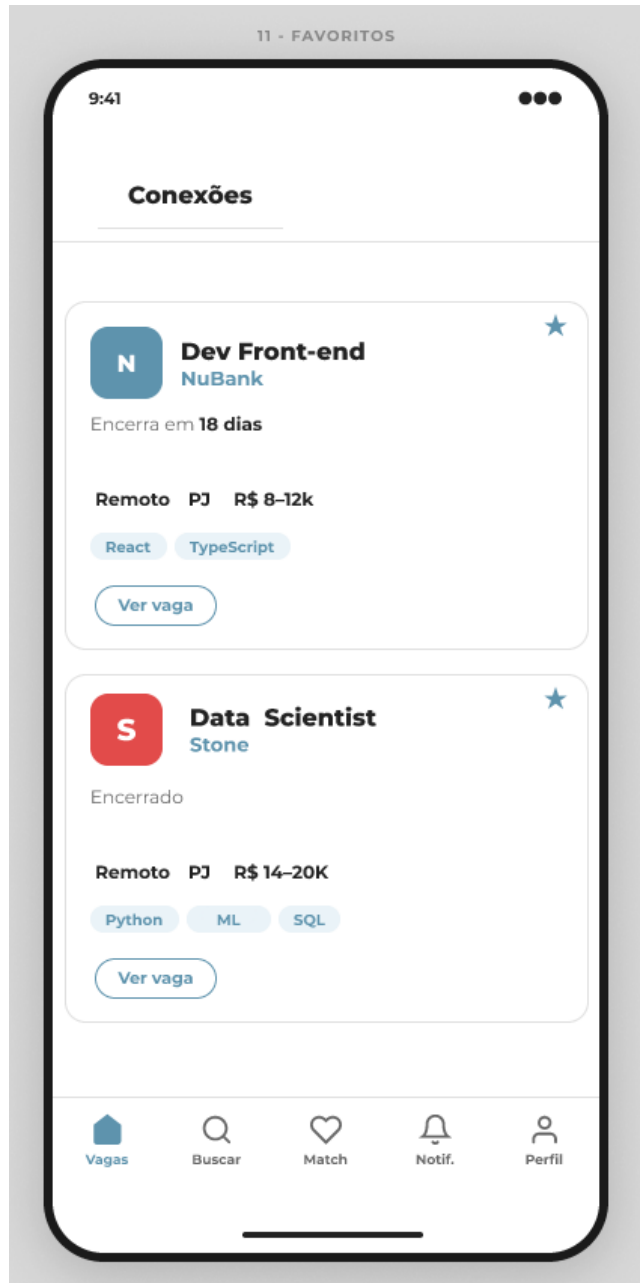


Figura 14 – Vagas Salvas e Favoritos

Interface dedicada à visualização das vagas marcadas como favoritas ou salvas pelo candidato para análise posterior. Os cartões de oportunidade exibem o ícone de estrela ativado, indicando o salvamento. Além de apresentar o cargo, a empresa e as tags de requisitos, o design incorpora indicadores de urgência (ex: "Encerra em 18 dias" ou "Encerrado") para incentivar a conversão. A interface disponibiliza botões de atalho ("Ver vaga") para redirecionamento rápido aos detalhes da oportunidade.

FORMULÁRIO DE IDENTIFICAÇÃO

DADOS DO RELATÓRIO TÉCNICO E/OU CIENTÍFICO			
Título e subtítulo: TechJobs – Empresa		Tipo de relatório: Relatório Técnico-Científico	
Instituição: Faculdade de Tecnologia de Mauá - FATEC		Data:	
Curso: Tecnologia em Desenvolvimento de Software Multiplataforma		Nota:	
Autor(es): Amanda Lira Macedo Nunes, Flávio Máximo dos Santos, Matheus Haruki Shiomi e Thiago Luiz Franceliano			
Orientador:			
Banca avaliadora:			
Produto desenvolvido: Plataforma Web e Mobile: TechJobs: Candidato			
Especificações técnicas: Plataforma com arquitetura distribuída composta por um backend desenvolvido em C# com o framework .NET, atuando como uma API centralizada, e persistência de dados em banco de dados relacional SQL Server. O consumo dos serviços é realizado por duas frentes de interface de usuário: uma aplicação web responsiva construída com a biblioteca React e um aplicativo móvel multiplataforma desenvolvido com o framework Flutter (linguagem Dart).			
Palavras-chave/Descritores: Desenvolvimento multiplataforma; engenharia de software; recrutamento tecnológico; testes de software; arquitetura distribuída.			
Edição	Nº de páginas	Nº do volume/parte	Área do conhecimento:
Observações/notas			

Fonte: Estrutura adaptada pela autora com base na norma ABNT NBR 10719:2015.