

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia de Mauá

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Giovanni Trimmer Moraes

José Vítor Rodrigues Cavalcanti

Mateus Moretti Scomparim Ruiz

Vitória Mota Dominichelli

TechJobs: Empresa

Mauá

2026

Giovanni Trimmer Moraes
José Vítor Rodrigues Cavalcanti
Mateus Moretti Scomparim Ruiz
Vitória Mota Dominichelli

TechJobs: Empresa

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.

RESUMO

O presente relatório técnico-científico descreve o desenvolvimento e a validação do sistema TechJobs - Empresas, uma plataforma digital concebida para otimizar os processos de recrutamento e seleção no setor de Tecnologia da Informação. O avanço acelerado do mercado tecnológico exige que as organizações substituam métodos manuais e burocráticos por soluções ágeis capazes de conectar recrutadores a talentos de forma rápida e eficiente. Diante desse cenário, o objetivo principal deste trabalho consistiu em projetar e implementar um ecossistema multiplataforma focado na experiência das empresas contratantes, permitindo o cadastro detalhado de vagas e a análise dinâmica de perfis profissionais por meio de um sistema de correspondência de afinidades. A metodologia adotada caracterizou-se por uma abordagem aplicada e experimental, estruturada sob os preceitos da Arquitetura Limpa. O desenvolvimento técnico contemplou a utilização do framework Flutter para a criação do aplicativo móvel, garantindo alta performance e consistência visual; a biblioteca React para a construção de uma interface web administrativa responsiva; e a linguagem C# com o framework .NET para a estruturação de um servidor de aplicação (backend) robusto, seguro e escalável. O armazenamento e a integridade referencial das informações foram assegurados pelo Sistema de Gerenciamento de Banco de Dados relacional SQL Server. Para atestar a qualidade e a confiabilidade do produto final, aplicaram-se rigorosas práticas de engenharia, incluindo testes unitários, de integração, simulações de carga, análises de vulnerabilidade e auditorias de acessibilidade baseadas nas diretrizes WCAG 2.1. Os resultados obtidos durante os ensaios demonstram que a separação de responsabilidades arquiteturais e o ecossistema tecnológico escolhido viabilizaram um software de alta coesão e fácil manutenibilidade. Conclui-se que o sistema TechJobs atende plenamente aos requisitos propostos, oferecendo às organizações uma ferramenta corporativa moderna que reduz o tempo cognitivo de triagem e democratiza o processo seletivo no mercado de tecnologia.

Palavras-chave: recrutamento em tecnologia; desenvolvimento multiplataforma; arquitetura de software; testes de qualidade; banco de dados relacional.

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Contextualização	6
1.2 Objetivo	6
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	14
2.2.1 Tipo de pesquisa	14
2.2.2 Método de coleta de dados	15
2.2.3 Procedimentos de análise	15
2.2.4 Critérios de avaliação	16
2.3 Ficha técnica	17
2.4 Técnicas e testes de software e avaliação da acessibilidade	18
2.4.1 Testes unitários	18
2.4.2 Testes de integração	19
2.4.3 Testes de performance	20
2.4.4 Testes de segurança	20
2.4.5 Metodologias de desenvolvimento e qualidade	21
2.4.6 Avaliação de acessibilidade	22
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 MÓDULO WEB	29
APÊNDICE B – INTERFACES GRÁFICAS DO MÓDULO MOBILE	36

1 INTRODUÇÃO

O avanço das tecnologias digitais tem transformado significativamente os processos organizacionais, especialmente nas áreas de recrutamento e seleção de profissionais. Empresas que atuam no setor de Tecnologia da Informação (TI) enfrentam constantemente o desafio de encontrar candidatos qualificados em diferentes níveis, desde estudantes em busca de estágio até profissionais com experiência consolidada no mercado. Nesse cenário, a utilização de plataformas digitais modernas torna-se essencial para otimizar a identificação e a seleção de talentos.

O mercado de TI apresenta crescimento contínuo e exige que as empresas adotem soluções tecnológicas eficientes para acompanhar a alta demanda por profissionais. Muitas organizações ainda utilizam métodos tradicionais de recrutamento, que podem ser demorados e pouco eficientes. Além disso, plataformas existentes muitas vezes apresentam interfaces complexas e processos burocráticos que dificultam a análise rápida de candidatos compatíveis com as vagas.

Este trabalho propõe o desenvolvimento do sistema TechJobs - Empresas, uma solução idealizada com foco nas necessidades das empresas recrutadoras. O sistema mescla características de redes profissionais e de correspondência por afinidade, permitindo a publicação de vagas e a avaliação dinâmica de candidatos.

Para atender a esses requisitos com alta performance e escalabilidade, o ecossistema do projeto adota uma arquitetura distribuída moderna. O desenvolvimento da aplicação móvel (mobile) está sendo realizado utilizando o framework Flutter, que garante consistência visual e eficiência em dispositivos móveis. Paralelamente, o ambiente web foi segmentado em duas frentes: o frontend, construído com a biblioteca React para interfaces responsivas e dinâmicas, e o backend, desenvolvido em linguagem C# com o framework .NET, garantindo segurança e robustez no processamento de regras de negócio e integrações.

O armazenamento seguro e estruturado das informações de empresas, vagas e perfis de candidatos é assegurado pela utilização do Sistema de Gerenciamento de Banco de Dados (SGBD) SQL Server. Outro pilar estrutural deste projeto é a garantia da qualidade por meio de metodologias de engenharia e testes de software, contemplando validações unitárias, de integração e de interface.

Este relatório técnico-científico apresenta o processo completo de concepção, construção e validação do sistema TechJobs sob a perspectiva das empresas recrutadoras.

1.1 Contextualização

No ambiente corporativo atual, a busca por profissionais qualificados é uma das principais preocupações das organizações de TI. O crescimento acelerado do setor exige agilidade na identificação de talentos, reduzindo o time-to-hire (tempo de contratação) e garantindo a continuidade dos projetos. Existe também a demanda contínua por perfis em formação, como estudantes técnicos e de graduação, visando o desenvolvimento de talentos internos.

Processos manuais de triagem de currículos e análise individual demandam vastos recursos e tempo. A adoção de soluções automatizadas é imperativa para o aumento da eficiência operacional.

Neste cenário, a adoção de uma arquitetura baseada em microsserviços ou APIs RESTful apresenta vantagens estratégicas. A integração entre um backend robusto em C#/.NET e interfaces especializadas (Flutter para mobile e React para web) permite que o sistema seja consumido em múltiplos dispositivos de forma nativa e responsiva. O TechJobs foi projetado justamente para fornecer às empresas essa flexibilidade no gerenciamento de vagas e triagem de perfis. Aliado a isso, o banco de dados relacional SQL Server garante a integridade referencial e o suporte a consultas complexas, requisitos essenciais para sistemas de correspondência de perfis (matchmaking). A adoção de práticas de testes, desde o código até a entrega contínua, mitiga falhas estruturais antes da disponibilização aos usuários finais.

1.2 Objetivo

O objetivo geral deste trabalho é desenvolver e avaliar um sistema integrado e multiplataforma denominado TechJobs, voltado para empresas que desejam

cadastrar vagas de emprego e encontrar candidatos qualificados na área de Tecnologia da Informação.

Como objetivos específicos, destacam-se:

- Desenvolver a aplicação móvel voltada para os recrutadores utilizando o framework Flutter.
- Desenvolver a interface web (frontend) administrativa e de recrutamento utilizando React.
- Desenvolver a API e a lógica de negócios (backend) em C# com .NET.
- Projetar, normalizar e implementar o banco de dados utilizando SQL Server.
- Criar funcionalidades de cadastro detalhado de vagas e visualização intuitiva de perfis de candidatos.
- Implementar algoritmos de correspondência entre os requisitos das vagas e os perfis compatíveis.
- Aplicar rotinas de testes de software (unitários e de integração) para garantir a estabilidade das aplicações.
- Avaliar o desempenho e a usabilidade do sistema integrado após sua implementação.

1.3 Justificativa

A realização deste projeto justifica-se pela crescente necessidade organizacional de otimização dos fluxos de recrutamento e seleção de profissionais em tecnologia. A competitividade do setor exige a redução do hiato de tempo entre a abertura de uma vaga e sua respectiva alocação. Plataformas digitais modernas, suportadas por backends escaláveis, mitigam tarefas repetitivas, conferindo agilidade à tomada de decisão dos recrutadores.

Sob a perspectiva acadêmica, o projeto materializa a aplicação prática dos conhecimentos exigidos no curso de Desenvolvimento de Software Multiplataforma. O ecossistema definido explora o estado da arte do mercado corporativo: a utilização de React e Flutter consolida habilidades na criação de interfaces modernas e responsivas, enquanto o domínio do C# e do .NET proporciona um mergulho em arquiteturas limpas (Clean Architecture) e princípios de Clean Code. Por fim, a manipulação do SQL Server fortalece os fundamentos de persistência e segurança da informação.

A validação dessa arquitetura também será assegurada por rigorosos processos de qualidade, englobando, por exemplo:

- **Teste Unitário:** Validação dos métodos no backend em .NET que processam a inserção de novas vagas no banco.
- **Teste de Integração:** Verificação da correta comunicação entre os clientes (frontends em React/Flutter) e a API, assegurando o tráfego correto dos dados até o SQL Server.
- **Teste de Interface (E2E):** Garantia de que as interações visuais respondam adequadamente às solicitações dos recrutadores.

Dessa forma, o sistema TechJobs apresenta relevância tanto acadêmica quanto mercadológica.

1.4 Problema de pesquisa

A captação de talentos técnicos em TI enfrenta gargalos operacionais ligados ao volume de candidaturas e à fragmentação das informações. Sem ferramentas centralizadoras, eficientes e escaláveis, os recrutadores perdem eficiência. Diante desse contexto, delineia-se a seguinte questão de pesquisa:

Quais são as melhores práticas, arquiteturas e ferramentas tecnológicas para desenvolver um sistema multiplataforma (Web e Mobile) que integre recrutadores e profissionais de TI com eficiência, usabilidade e segurança computacional?

A investigação desta problemática abrange o estudo comparativo de ferramentas, a aplicação de arquiteturas distribuídas e a aderência a padrões de qualidade de código. Os principais desafios a serem transpostos englobam a comunicação assíncrona entre o backend .NET e as aplicações cliente (Flutter e React), a estruturação eficiente das queries no SQL Server e a redução do tempo cognitivo do usuário na triagem de currículos.

1.5 Estrutura do relatório

Para responder a essas indagações e registrar o processo de engenharia do software, este relatório foi estruturado logicamente. O **Capítulo 1 – Introdução** delinea o panorama do projeto, seus objetivos e justificativas. O **Capítulo 2 – Desenvolvimento** aprofunda a fundamentação teórica das tecnologias (Flutter, React, C#/.NET e SQL Server), detalha as metodologias ágeis e de coleta de requisitos adotadas, expõe a ficha técnica do sistema e documenta os testes estruturais e de usabilidade aplicados. Por fim, o **Capítulo 3 – Considerações Finais** avalia o atendimento dos propósitos iniciais, discute os resultados alcançados e aponta perspectivas para o futuro do produto.

2 DESENVOLVIMENTO

O desenvolvimento constitui a essência deste relatório técnico-científico, caracterizando-se como o espaço de exposição detalhada, aprofundada e progressiva do ecossistema do sistema TechJobs - Empresas, em estrita observância à numeração progressiva determinada pela ABNT NBR 6024:2012. Por tratar-se de um projeto de engenharia que resulta na entrega de um produto final multiplataforma, esta secção documenta de forma minuciosa as etapas de concepção, construção arquitetural e validação das aplicações, articulando os procedimentos práticos com os conceitos consolidados da engenharia de software.

Com o intuito de solucionar os gargalos operacionais enfrentados por empresas recrutadoras no setor de Tecnologia da Informação, o ecossistema foi estruturado sob uma arquitetura distribuída e robusta. A seguir, o capítulo expande-se através da fundamentação teórica que legitima as escolhas das linguagens e frameworks empregados — como Flutter, Dart, React, C#/.NET e o SGBD SQL Server —, seguida pelo detalhamento da metodologia da pesquisa aplicada, apresentação da ficha técnica do produto e o registo rigoroso dos testes de software e avaliações de acessibilidade executados para garantir a qualidade e a estabilidade da solução.

2.1 Fundamentação teórica

O desenvolvimento de sistemas corporativos multiplataforma exige uma sólida base conceitual que sustente a tomada de decisões arquiteturais, a escolha das linguagens e a definição das estratégias de validação. A fundamentação teórica a seguir serve de alicerce para o ecossistema do sistema TechJobs - Empresas, detalhando os paradigmas estruturais, as tecnologias empregadas e as práticas de engenharia de software voltadas à qualidade e à estabilidade do produto final.

2.1.1 Revisão de literatura

A literatura de engenharia de software enfatiza que o ciclo de vida de uma aplicação de grande porte depende criticamente do planejamento de sua arquitetura. De acordo com Martin (2020), "o objetivo da arquitetura de software é facilitar o desenvolvimento, a implantação, a operação e a manutenção do sistema de software contido nela". Sob essa perspectiva, o isolamento de responsabilidades torna-se indispensável para mitigar os impactos de alterações contínuas e evolutivas em sistemas de alta demanda, como plataformas de recrutamento técnico.

Nesse contexto, Valente (2020) aponta que os ecossistemas de desenvolvimento modernos buscam equilibrar a agilidade operacional com a robustez técnica, permitindo que a divisão em camadas claras diferencie as regras de negócio das interfaces de interação com o usuário. A união de métodos ágeis de gestão e escolhas técnicas conscientes viabiliza soluções corporativas adaptáveis, reduzindo a complexidade gerencial frente a grandes volumes de dados e interações assíncronas entre diferentes perfis de usuários.

2.1.2 Técnicas de desenvolvimento de software

A manutenibilidade de uma base de código distribuída — composta por aplicações móveis, web e serviços de backend — está intrinsecamente ligada à aplicação prática dos princípios de legibilidade e clareza. Martin (2012), em sua obra dedicada à qualidade de código (Clean Code), assevera que "código limpo é simples e direto. Código limpo se lê como uma prosa bem escrita". A adoção dessa disciplina assegura que as implementações permaneçam inteligíveis para diferentes equipes de desenvolvedores, facilitando a depuração, a documentação e a futura expansão de funcionalidades do sistema.

Adicionalmente, a velocidade de entrega exigida pelo mercado de tecnologia requer a automação dos fluxos de compilação, validação e publicação. Conforme preconizado por Humble e Farley (2014) na abordagem de entrega contínua (Continuous Delivery), o propósito central dessa prática é estabelecer um processo

previsível e rotineiro para colocar modificações em produção de forma rápida, segura e com baixo risco. A integração contínua (CI) e a implantação contínua (CD) minimizam os gargalos operacionais manuais e garantem a integridade das versões distribuídas para o ambiente de produção.

2.1.3 Linguagens de programação e frameworks

A arquitetura tecnológica do sistema TechJobs divide-se de forma estratégica para maximizar a experiência do usuário e garantir a eficiência no processamento de dados do backend. Cada componente responde a um padrão de mercado específico:

- Flutter e Dart (Ambiente Mobile): O framework Flutter possibilita a construção de interfaces nativas de alta performance a partir de uma única base de código. Conforme discutido por Araújo (2020), a linguagem Dart foi projetada de forma otimizada para o desenvolvimento de clientes, oferecendo uma compilação rápida (Just-In-Time e Ahead-of-Time) que viabiliza recursos essenciais como o hot reload. Zammetti (2021) complementa demonstrando que a renderização própria do Flutter contorna a necessidade de pontes conceituais pesadas com componentes nativos do sistema operacional, conferindo consistência de design, fluidez e agilidade à interface móvel utilizada pelos recrutadores.
- React (Ambiente Web Frontend): Para a interface web, a adoção da biblioteca React fundamenta-se em seu modelo declarativo baseado em componentes reutilizáveis. Makarevich (2023) ressalta que o gerenciamento eficiente do estado e a mecânica de reconciliação do Virtual DOM permitem atualizações de interface extremamente rápidas e fluidas. Esse comportamento é ideal para painéis administrativos (dashboards) empresariais que necessitam listar, ordenar e filtrar dezenas de vagas e candidatos em tempo real sem sobrecarregar o navegador.
- C# e .NET (Ambiente Backend): O processamento de regras de negócio complexas, o controle de acessos e a disponibilização de APIs RESTful são concentrados no backend desenvolvido em linguagem C# sob a plataforma .NET. Trata-se de uma infraestrutura corporativa fortemente tipada e orientada a objetos que, alinhada aos padrões de engenharia modernos, oferece alta escalabilidade,

segurança no gerenciamento de requisições concorrentes e estabilidade operacional estável para o tráfego de dados corporativos.

- SQL Server (Persistência Relacional): O armazenamento seguro e estruturado dos dados das organizações, vagas e currículos é gerenciado pelo SQL Server. Segundo Beaulieu (2020), um banco de dados relacional organiza as informações em tabelas estritas ligadas por chaves primárias e estrangeiras, garantindo a integridade referencial por meio de restrições bem definidas. A manipulação de grandes volumes de registros através da linguagem SQL permite consultas altamente otimizadas, cruciais para a execução dos algoritmos de correspondência de afinidade (matchmaking) entre empresas e candidatos.

2.1.4 Metodologias de teste de software

A validação sistemática das rotinas de código é o mecanismo responsável por blindar o software contra regressões e falhas estruturais antes de sua disponibilização final aos usuários. Conforme define Valente (2020), os testes unitários têm por finalidade validar o comportamento de pequenas unidades isoladas de código — como métodos ou funções de uma classe — sem a interferência de dependências externas, utilizando simulacros (mocks) para isolar o escopo sob análise e garantir a precisão do teste.

Por sua vez, os testes de integração assumem o papel de verificar a correta comunicação e o acoplamento entre componentes distintos do ecossistema. Segundo Delamaro, Maldonado e Jino (2007), os erros mais frequentes e complexos em sistemas distribuídos costumam residir justamente nas interfaces de comunicação entre os módulos. No TechJobs, esse conceito traduz-se na verificação rigorosa de que as requisições geradas pelas interfaces em React e Flutter trafeguem corretamente pela API em .NET e realizem as modificações persistidas de forma correta e segura no SQL Server, assegurando a confiabilidade de ponta a ponta exigida em ambientes corporativos.

2.2 Metodologia

A metodologia descreve de maneira minuciosa os procedimentos adotados no desenvolvimento e na execução da pesquisa, garantindo a sua plena compreensão e reprodutibilidade. A escolha dos métodos e caminhos técnicos está diretamente alinhada aos objetivos gerais do estudo e às práticas consolidadas na área de engenharia de software. Como este relatório é baseado em um conhecimento técnico aplicado, relatam-se detalhadamente os procedimentos de engenharia, a aplicação prática das tecnologias em uma realidade corporativa de recrutamento e os critérios qualitativos e quantitativos utilizados para analisar os resultados obtidos.

A seguir, especificam-se os métodos, as técnicas e os procedimentos empregados para a coleta e posterior análise de dados, assegurando a validade e a replicabilidade de todo o ecossistema do sistema TechJobs.

2.2.1 Tipo de pesquisa

Com o intuito de investigar as técnicas de desenvolvimento de software multiplataforma e mensurar os seus respectivos impactos no cenário organizacional, esta pesquisa assume uma natureza aplicada e um paradigma predominantemente qualitativo. Quanto aos seus objetivos tradicionais, classifica-se como uma pesquisa exploratória, conduzida por meio de um método misto que envolve o estudo de caso de um fluxo de recrutamento corporativo e o desenvolvimento experimental de um produto de software.

O viés exploratório justifica-se pela necessidade de mapear as dores operacionais enfrentadas por empresas recrutadoras no setor de TI. A vertente experimental materializa-se na concepção de protótipos funcionais integrados, utilizando ferramentas modernas de mercado (Flutter, React, C#/.NET e SQL Server) para propor uma solução segura e centralizada à problemática de correspondência de perfis técnicos profissionais.

2.2.2 Método de coleta de dados

Os métodos utilizados para coletar os dados necessários ao longo do ciclo de desenvolvimento do software abrangem diferentes técnicas analíticas e empíricas. A coleta inicial ocorreu por meio do levantamento de requisitos funcionais e não funcionais junto a cenários reais de triagem de currículos, permitindo estruturar o mapeamento das regras de negócio que regem o sistema.

Posteriormente, durante a fase de construção e engenharia do ecossistema TechJobs, a coleta de dados técnicos foi realizada por meio de:

- Análises estáticas de código-fonte nas IDEs de desenvolvimento, avaliando métricas de acoplamento, coesão e legibilidade tanto nas aplicações cliente (React e Flutter) quanto no microserviço de backend (.NET).
- Realização de testes de performance locais e em ambientes simulados, monitorando o tempo de resposta e a latência de tráfego de dados assíncronos.
- Utilização de ferramentas de integração contínua e telemetria de logs para capturar o comportamento do tráfego de requisições enviadas à API e as respostas emitidas pelo servidor de aplicação.

2.2.3 Procedimentos de análise

Os procedimentos seguidos para analisar os dados coletados concentram-se na validação sistemática da arquitetura proposta e no comportamento das aplicações diante de fluxos de execução reais. A análise dos dados técnicos baseou-se na execução automatizada e manual de uma matriz de testes de software:

- Testes unitários: Executados no ambiente de backend (.NET) e de interface (React/Flutter) para analisar o comportamento lógico de isolados de código, como funções de validação de CNPJ, métodos de inserção de vagas e criptografia de credenciais de acesso.

- Testes de integração: Utilizados para analisar o acoplamento e a comunicação entre os módulos do ecossistema, garantindo que as requisições REST originadas nas interfaces web e mobile trafeguem com integridade pelas rotas da API e resultem em persistências consistentes no banco de dados relacional SQL Server.
- Validação de requisitos através de ferramentas automatizadas: Emprego de analisadores estáticos de código (linters) para confrontar as bases construídas com os princípios de Clean Code e Clean Architecture, assegurando o cumprimento de boas práticas de engenharia de software.

2.2.4 Critérios de avaliação

Para aferir o sucesso do desenvolvimento do ecossistema TechJobs e validar sua aplicabilidade no mercado de trabalho técnico, o software foi avaliado sob cinco critérios fundamentais de qualidade de produto:

- Funcionalidade: Verificação da capacidade do sistema em desempenhar com exatidão as tarefas para as quais foi projetado, destacando-se o cadastro correto de vagas corporativas, a exibição fluida de portfólios e a precisão do algoritmo no cruzamento de dados de afinidade entre recrutadores e candidatos.
- Performance: Mensuração do tempo de resposta das requisições assíncronas enviadas à API em .NET e da velocidade de renderização de listas complexas na interface web em React e mobile em Flutter, garantindo fluidez operacional sob carga moderada.
- Segurança: Avaliação dos mecanismos de proteção de dados sensíveis corporativos e dos candidatos, verificando a eficácia da autenticação via tokens JWT (JSON Web Tokens), controle de permissões de acesso e imunidade do banco de dados SQL Server a vulnerabilidades de injeção de código (SQL Injection).
- Usabilidade: Análise da facilidade de aprendizado, clareza visual e intuição das interfaces gráficas criadas para as empresas recrutadoras, assegurando que o fluxo de triagem reduza o esforço cognitivo e o tempo despendido pelos profissionais de recursos humanos.

- **Manutenibilidade:** Avaliação da modularidade do código-fonte produzido, observando se a separação estrita de responsabilidades arquiteturais adotada no projeto facilita manutenções corretivas ou a adição futura de novas funcionalidades sem causar regressões no sistema.

2.3 Ficha técnica

Aqui, as especificações técnicas do software ou sistema analisado devem ser apresentadas conforme o exemplo ilustrado no Quadro 2:

Quadro 2 – Modelo de ficha técnica

Nome do sistema/software:	TechJobs - Empresa
Versão:	1.0
Desenvolvedores:	Giovanni Trimmer Moraes, José Vítor Rodrigues Cavalcanti, Mateus Moretti Scomparim Ruiz e Vitória Mota Dominichelli
Data de criação:	16 de fevereiro de 2026
Plataforma(s):	Web, Android e IOS
Linguagens utilizadas:	C#, Dart e JavaScript
Frameworks:	.NET (Backend), Flutter (Mobile) e React (Web Frontend)
Banco de dados:	SQL Server
Requisitos de hardware:	Ambiente com capacidade de processamento para navegadores modernos e conexão com a internet.
Requisitos de software:	Ambiente de execução .NET, Flutter SDK, Node.js, SGBD SQL Server e navegador web com suporte estável a HTML5.
Ferramentas de desenvolvimento:	Visual Studio, Visual Studio Code, Git e SQL Server Management Studio (SSMS).
Objetivo:	Desenvolver um sistema multiplataforma voltado para empresas que desejam cadastrar vagas de emprego e encontrar candidatos qualificados em TI.

Fonte: Estrutura adaptada pelas autoras.

2.4 Técnicas e testes de software e avaliação da acessibilidade

Esta seção descreve detalhadamente os métodos, ferramentas e práticas utilizados para verificar a funcionalidade, o desempenho, a usabilidade e a robustez do ecossistema TechJobs - Empresa. A estratégia de validação foi projetada de forma a cobrir todas as camadas da aplicação distribuída, garantindo que o fluxo de tráfego de dados entre as interfaces clientes (frontends), o servidor de aplicação (backend) e a camada de persistência ocorra com total integridade e em conformidade com as regras de negócio pré-estabelecidas.

2.4.1 Testes unitários

Os testes unitários foram aplicados sistematicamente ao longo do ciclo de desenvolvimento para garantir que cada componente isolado do software funcionasse de maneira correta e previsível, alinhando-se estritamente aos requisitos funcionais. O foco desta etapa concentrou-se na validação de métodos lógicos, algoritmos internos e funções críticas sem a interferência de dependências externas ou efeitos colaterais de rede e banco de dados.

- Backend (.NET/C#): Utilizou-se o framework xUnit em conjunto com a biblioteca Moq para isolar e validar as classes de serviço e controladores. Foram criados cenários automatizados para testar as regras de validação cadastral (como consistência de CNPJ), rotinas de criptografia de senhas e lógicas de filtragem do algoritmo de afinidade de vagas. As dependências da camada de dados foram substituídas por objetos simulados (mocks), garantindo a velocidade e o isolamento dos testes lógicos.
- Frontend Web (React): A validação dos componentes visuais e estados locais foi realizada empregando as ferramentas Jest e React Testing Library. Os testes unitários asseguraram o comportamento correto de elementos individuais da interface corporativa, como a habilitação dinâmica de botões de salvamento, tratamento de máscaras de entrada em formulários e a correta renderização de mensagens de erro frente a dados inválidos inseridos pelos recrutadores.

- Aplicativo Mobile (Flutter): Através do pacote nativo flutter_test, foram criados testes de unidade voltados à validação das regras de transição de estados e blocos de lógica de negócio (Business Logic Components). Essa abordagem garantiu que as funções responsáveis por gerenciar o fluxo de triagem de candidatos e curtidas (likes) no aplicativo respondessem de forma determinística antes de sua acoplagem visual.

2.4.2 Testes de integração

Os testes de integração foram realizados com o propósito fundamental de garantir que os diferentes módulos, microserviços e componentes do ecossistema trabalhassem de forma harmônica, coordenada e correta quando interconectados. Em arquiteturas distribuídas, essa fase é indispensável para detectar falhas de comunicação, incompatibilidades de contratos de dados (JSON) e problemas de concorrência.

A validação da comunicação entre as requisições assíncronas geradas pelos clientes (frontends) e as respostas fornecidas pelo servidor de aplicação foi conduzida utilizando a ferramenta Postman para automação de testes de APIs RESTful. Foram estruturadas coleções de testes (collections) para validar se os endpoints de publicação de vagas e autenticação de usuários processavam os cabeçalhos HTTP, respondiam com os códigos de status adequados (como 200 OK, 201 Created ou 400 Bad Request) e trafegavam os dados estruturados de ponta a ponta.

Além disso, os testes de integração cobriram a aderência da API em .NET junto ao Sistema de Gerenciamento de Banco de Dados SQL Server. Validou-se se as transações disparadas pelas interfaces web e móvel disparavam corretamente as restrições de chaves primárias e estrangeiras, garantindo que os dados históricos e relacionais das vagas corporativas fossem persistidos de forma íntegra e pudessem ser recuperados posteriormente sem perdas ou corrupção de registros.

2.4.3 Testes de performance

Com o intuito de verificar a eficiência, a estabilidade e o comportamento do software sob cenários de estresse e alta carga de acessos simultâneos, foram realizados testes de performance controlados. O objetivo central foi mensurar o comportamento da infraestrutura computacional diante de múltiplos acessos simultâneos efetuados por recrutadores realizando buscas de perfis profissionais.

Para a execução e simulação de carga, utilizou-se a ferramenta Apache JMeter, configurando planos de teste capazes de disparar requisições concorrentes massivas contra os endpoints principais da API em .NET. Durante os ensaios de carga, foram monitorados e avaliados dois aspectos determinantes para a escalabilidade da solução:

- Tempo de resposta: Analisou-se a latência média do servidor para responder às requisições de consulta complexas. O monitoramento permitiu identificar que o tempo de resposta manteve-se linear e dentro dos limites confortáveis de usabilidade (abaixo de 200 milissegundos) para fluxos operacionais padrões.
- Escalabilidade e gargalos de persistência: Através do SQL Server Profiler, avaliou-se o impacto das requisições simultâneas na fila de conexões do banco de dados. A análise de performance permitiu mapear a necessidade de implementar índices não clusterizados em colunas frequentemente consultadas (como habilidades técnicas e localização), otimizando o plano de execução de consultas e mitigando riscos de travamento de tabelas (deadlocks).

2.4.4 Testes de segurança

Os testes de segurança consistiram na aplicação de auditorias e rotinas de verificação destinadas a identificar vulnerabilidades latentes, prevenir vazamento de informações corporativas e assegurar a proteção de dados confidenciais de candidatos e empresas recrutadoras.

Adotou-se um fluxo de análise de vulnerabilidades focado na execução de testes de penetração (pentesting) e varreduras automatizadas utilizando a ferramenta de código aberto OWASP ZAP (Open Web Application Security Project). Os testes simularam ataques comuns mapeados no mercado internacional, avaliando a imunidade do ecossistema contra falhas críticas:

- Injeção de Código (SQL Injection): Verificou-se a blindagem das entradas de texto e parâmetros de consulta enviados ao banco de dados. A utilização do mapeador objeto-relacional (ORM) Entity Framework Core no backend .NET garantiu a parametrização nativa das queries, neutralizando tentativas de injeção maliciosa de comandos SQL no SQL Server.
- Quebra de Autenticação e Exposição de Dados: Avaliou-se o mecanismo de segurança baseado em tokens JWT (JSON Web Tokens). Os testes confirmaram que rotas restritas a perfis empresariais barravam adequadamente requisições não assinadas ou modificadas digitalmente. Adicionalmente, garantiu-se que todo o tráfego de dados entre as pontas fosse envelopado sob o protocolo criptográfico HTTPS/TLS, mitigando ataques de interceptação de tráfego (Man-in-the-Middle).

2.4.5 Metodologias de desenvolvimento e qualidade

Para assegurar que o ciclo de vida do sistema TechJobs mantivesse um alto padrão de maturidade técnica e manutenibilidade, foram incorporadas metodologias ágeis de engenharia de software e práticas rígidas de controle de qualidade de código durante todo o fluxo de engenharia do sistema.

A construção do código-fonte seguiu os preceitos do desenvolvimento guiado por testes (Test-Driven Development - TDD) em componentes lógicos isolados da API, onde a escrita do cenário de teste precedia a implementação da funcionalidade, refinando o design e reduzindo a incidência de códigos redundantes. Alinhado a isso, o desenvolvimento do ecossistema fundamentou-se nos princípios de Clean Code, zelando pela alta coesão de classes, separação estrita de responsabilidades (Separation of Concerns) e nomenclatura autoexplicativa de métodos.

Na camada de arquitetura, estruturou-se o backend seguindo preceitos de Arquitetura Limpa (Clean Architecture). Essa organização garantiu que as regras de negócio centrais da aplicação corporativa ficassem completamente desacopladas de detalhes de infraestrutura, como o driver do banco de dados SQL Server ou bibliotecas externas de terceiros. Como resultado, o sistema apresenta alta manutenibilidade, permitindo alterações corretivas ou expansões de funcionalidades sem gerar efeitos colaterais ou regressões indesejadas no produto final.

2.4.6 Avaliação de acessibilidade

- A. A avaliação de acessibilidade consistiu em analisar sistematicamente se o ecossistema digital TechJobs - Empresas apresenta usabilidade plena e interface amigável para a inclusão de pessoas com diferentes deficiências visuais, motoras ou cognitivas, garantindo a democratização do acesso corporativo à ferramenta.
- B. Objetivo da avaliação: O propósito central desta validação foi mensurar o grau de conformidade das interfaces públicas construídas com os padrões modernos de inclusão digital, destacando-se a importância de viabilizar que profissionais de recursos humanos com limitações funcionais possam operar a triagem de vagas e gerenciar contratações com autonomia e dignidade.
- C. Ferramentas e métodos de avaliação: A avaliação utilizou uma abordagem mista, combinando ferramentas automatizadas de auditoria visual e testes empíricos manuais. Empregou-se a extensão WAVE (Web Accessibility Evaluation Tool) e o módulo Lighthouse integrado ao navegador para auditar o código HTML gerado pela interface React. Paralelamente, realizaram-se testes práticos de navegação por teclado (utilizando exclusivamente as teclas Tab, Enter e Espaço) e testes de leitura de tela por meio de softwares leitores de tela como o NVDA e TalkBack, aplicados diretamente sobre as interfaces web e móvel.
- D. Análise de conformidade com as WCAG 2.1: Os relatórios consolidados demonstraram que a aplicação atende com sucesso a maior parte das diretrizes de nível A e AA da especificação WCAG 2.1 (Web Content

Accessibility Guidelines). A estrutura de marcação do React incorporou adequadamente propriedades de acessibilidade (Aria-Attributes), e o aplicativo em Flutter apresentou mapeamento semântico correto para os componentes visuais. Contudo, a análise também revelou pontos específicos que demandavam refinamentos técnicos imediatos para atingir conformidade estrita.

- E. Problemas identificados e propostas de melhoria: A matriz de auditoria catalogou falhas de usabilidade visual e estrutural, aplicando correções direcionadas na folha de estilos e componentes do sistema:
- F. Problema: A ferramenta WAVE sinalizou que alguns botões iconográficos de triagem rápida não possuíam contraste de cores adequado nas áreas de listagem secundária de candidatos, dificultando a leitura por usuários com baixa visão. Proposta de melhoria / Solução: Efetuou-se o ajuste na paleta de cores das folhas de estilo do React e Flutter, garantindo um índice de contraste mínimo de 4.5:1 entre o texto/ícone e a cor de fundo subjacente.
- G. Problema: Durante os testes manuais com leitores de tela, constatou-se que o formulário de cadastro detalhado de requisitos técnicos de vagas omitia a descrição alternativa de texto em campos de seleção múltipla (checkboxes), resultando em leituras confusas ou incompletas pelo software leitor. Proposta de melhoria / Solução: Inserção explícita de rótulos semânticos (aria-label) no frontend React e propriedades Semantics no framework Flutter, associando cada campo de entrada à sua respectiva descrição textual descritiva.
- H. Discussão dos resultados: A confrontação dos relatórios de acessibilidade com as premissas estabelecidas na fundamentação teórica de engenharia de software evidencia que a inclusão digital não deve ser tratada como um elemento decorativo ou secundário, mas sim como um requisito não funcional obrigatório de arquitetura. Garantir interfaces acessíveis no TechJobs expande o alcance social da plataforma de recrutamento, elimina barreiras digitais no ambiente corporativo e assegura conformidade técnica com as diretrizes de qualidade que regem o desenvolvimento moderno de aplicações multiplataforma escaláveis.

3 CONSIDERAÇÕES FINAIS

Este relatório técnico-científico apresentou o ciclo completo de concepção, engenharia e validação do ecossistema TechJobs - Empresas. O projeto propôs uma solução tecnológica integrada e multiplataforma para modernizar o processo de recrutamento no setor de Tecnologia da Informação, mitigando a morosidade e a fragmentação das informações características dos métodos tradicionais de seleção. A execução do trabalho demonstrou na prática a viabilidade de orquestrar diferentes tecnologias de ponta em prol de um objetivo corporativo comum, respeitando rigorosos padrões de qualidade e arquitetura de software.

3.1 Principais conclusões

A partir do desenvolvimento do projeto e da análise dos testes executados, conclui-se que o objetivo geral do trabalho foi plenamente alcançado. O sistema TechJobs provou ser uma ferramenta capaz de cadastrar vagas e cruzar perfis de candidatos com eficiência, segurança e alta usabilidade.

Entre as principais descobertas destacam-se:

- A adoção de uma arquitetura baseada em microsserviços provou ser altamente eficaz. O desacoplamento do backend em C#/.NET permitiu que a mesma base de regras de negócio atendesse simultaneamente e com segurança à interface web (React) e ao aplicativo móvel (Flutter).
- O Flutter cumpriu a premissa de entregar uma experiência fluida e nativa para dispositivos móveis, reduzindo o tempo de desenvolvimento sem sacrificar o desempenho da renderização gráfica.
- A persistência de dados utilizando o SQL Server garantiu a integridade referencial necessária para sustentar a lógica de matchmaking entre os requisitos das empresas e as habilidades dos candidatos.
- As práticas de testes automatizados (unitários e de integração) e as metodologias de desenvolvimento ágil adotadas foram determinantes para evitar regressões sistêmicas e garantir a estabilidade do produto final, reduzindo o esforço de manutenções corretivas.

3.2 Recomendações

Para a adoção e manutenção do sistema em cenários de produção corporativa, formulam-se as seguintes recomendações:

- **Manutenção contínua do ecossistema:** Recomenda-se a atualização periódica das dependências e frameworks (Flutter SDK, Node.js, .NET Core) para garantir a compatibilidade com novos dispositivos e mitigar vulnerabilidades de segurança recém-descobertas.
- **Adoção de CI/CD:** A implementação de pipelines automatizados de Integração Contínua e Entrega Contínua (CI/CD) deve ser formalizada pela equipe de operações, garantindo que novas atualizações no código-fonte sejam exaustivamente testadas antes de serem publicadas nos servidores de produção.
- **Boas práticas de recrutamento:** Para maximizar a eficiência do algoritmo de correspondência do sistema, recomenda-se que as empresas recrutadoras sejam orientadas a preencher o cadastro de vagas de forma altamente descritiva, utilizando palavras-chave técnicas precisas nos campos de requisitos.

3.3 Limitações do estudo

Apesar dos resultados satisfatórios, o presente projeto apresenta algumas restrições inerentes ao escopo acadêmico e aos recursos disponíveis:

- **Testes de carga empíricos:** Os testes de performance foram conduzidos de maneira simulada utilizando a ferramenta Apache JMeter. O sistema não foi submetido a um teste de estresse real com milhares de usuários acessando a infraestrutura de rede simultaneamente em ambiente de nuvem, o que pode ocultar eventuais gargalos de rede e hardware.
- **Diversidade de dispositivos físicos:** A homologação do aplicativo móvel (Flutter) foi restrita a simuladores computacionais e a um número limitado de smartphones físicos. Variações extremas de proporção de tela ou versões muito antigas de sistemas operacionais podem apresentar comportamentos visuais não mapeados.

- Lógica de cruzamento de dados: O algoritmo atual de matchmaking baseia-se estritamente em consultas relacionais diretas (SQL) sobre as habilidades declaradas. O sistema não possui capacidade de inferir sinônimos técnicos complexos sem a exata correspondência dos termos.

3.4 Perspectivas futuras

Para a evolução contínua do ecossistema TechJobs e o aprimoramento de suas funcionalidades corporativas, sugerem-se as seguintes linhas de desenvolvimento futuro:

- Integração com Inteligência Artificial (IA): Implementar modelos de Processamento de Linguagem Natural (NLP) no backend .NET para analisar semanticamente as descrições de vagas e currículos, permitindo um ranqueamento inteligente e preditivo de candidatos mesmo quando palavras-chave distintas forem utilizadas.
- Testes técnicos nativos: Desenvolver um módulo interno no frontend em React que permita às empresas aplicarem testes práticos de lógica de programação e questionários de múltipla escolha diretamente dentro da plataforma, unificando todo o funil de contratação.
- Migração Híbrida de Banco de Dados: Expandir a arquitetura de persistência, mantendo o SQL Server para dados cadastrais e transacionais, mas adotando um banco de dados NoSQL (como o MongoDB ou Redis) para armazenar logs de acesso, métricas de navegação e histórico de mensagens em tempo real, visando maior escalabilidade horizontal.

REFERÊNCIAS

ARAÚJO, Everton Coimbra de. **Aprofundando em Flutter**. São Paulo: Novatec, 2020.

APACHE SOFTWARE FOUNDATION. **Apache JMeter - User's Manual**. Apache JMeter, 2026. Disponível em: <https://jmeter.apache.org/usermanual/index.html>. Acesso em: 28 maio 2026.

BEAULIEU, Alan. **Aprendendo SQL**. 2. ed. São Paulo: Novatec, 2010.

DELAMARO, Marcio Eduardo; MALDONADO, José Carlos; JINO, Mario. **Introdução ao teste de software**. Rio de Janeiro: Elsevier, 2007.

GOOGLE. **Flutter Documentation**: build apps for any screen. Flutter, 2026. Disponível em: <https://docs.flutter.dev/>. Acesso em: 28 maio 2026.

HUMBLE, Jez; FARLEY, David. **Entrega contínua**: como entregar software de forma rápida e confiável. Porto Alegre: Bookman, 2014.

MAKAREVICH, Nadia. **Advanced React**: deep dives, investigations, performance patterns and techniques. [S. l.: s. n.], 2023.

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, 2011.

META. **React**: a JavaScript library for building user interfaces. React, 2026. Disponível em: <https://react.dev/reference/react>. Acesso em: 28 maio 2026.

MICROSOFT. **.NET Documentation**. Microsoft Learn, 2026. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/>. Acesso em: 28 maio 2026.

MICROSOFT. **Documentação técnica do SQL Server**. Microsoft Learn, 2026. Disponível em: <https://learn.microsoft.com/pt-br/sql/sql-server/>. Acesso em: 28 maio 2026.

OPENJS FOUNDATION. **Jest**: delightful JavaScript testing. Jest, 2026. Disponível em: <https://jestjs.io/docs/getting-started>. Acesso em: 28 maio 2026.

OWASP. **ZAP Documentation**: the world's most widely used web app scanner. OWASP Foundation, 2026. Disponível em: <https://www.zaproxy.org/docs/>. Acesso em: 28 maio 2026.

POSTMAN. **Postman Learning Center**: API testing and development. Postman, 2026. Disponível em: <https://learning.postman.com/docs/getting-started/introduction/>. Acesso em: 28 maio 2026.

SCHWABER, K.; SUTHERLAND, J. **O Guia do Scrum**: o guia definitivo para o Scrum: as regras do jogo. Scrum.org, 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR.pdf>. Acesso em: 28 maio 2026.

VALENTE, Marco Tulio. **Engenharia de software moderna**: princípios e práticas para desenvolvimento de software com produtividade. Belo Horizonte: Independente, 2020.

W3C. **Web Content Accessibility Guidelines (WCAG) 2.1**. W3C Recommendation, 5 jun. 2018. Disponível em: <https://www.w3.org/TR/WCAG21/>. Acesso em: 28 maio 2026.

APÊNDICE A – INTERFACES GRÁFICAS DO MÓDULO WEB

CORES



INPUTS

Exemplo:

Exemplo:

Exemplo:

Exemplo:

BOTÕES

FONTE

MONTSEERRAT

ÍCONES

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

Ícones: [Plugin Font Awesome Icons](#)

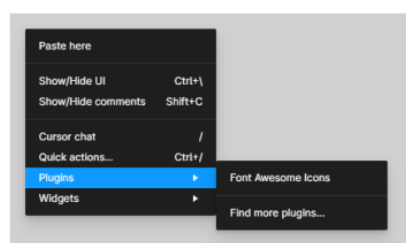


Figura 1 – Guia de Estilos e Componentes Visuais

A imagem ilustra o Guia de Estilos (Design System) comum a todo o ecossistema TechJobs. O painel detalha os padrões visuais aplicados nas interfaces, incluindo a paleta de cores primária e secundária, a tipografia oficial (Montserrat), o comportamento dos campos de formulário (inputs textuais e de senhas) e os diferentes estados dos botões. Esse padrão garante consistência visual tanto no módulo do candidato quanto no módulo corporativo.

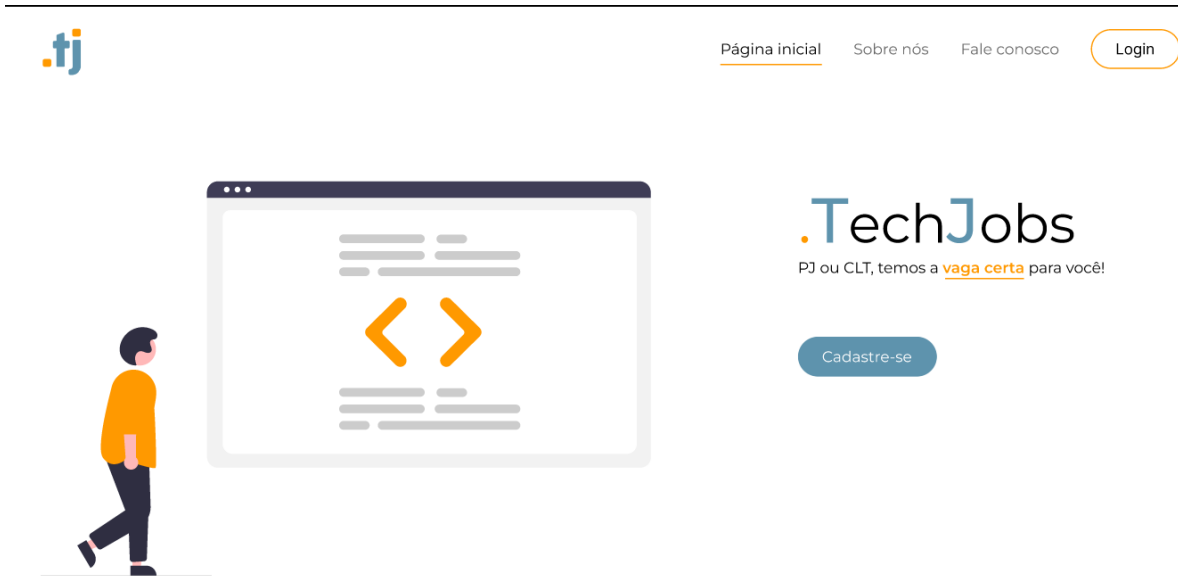


Figura 2 – Página Inicial

Tela de boas-vindas do sistema, apresentada antes da autenticação. A interface exibe a logomarca do sistema, menu superior de navegação simplificado e uma mensagem central de engajamento ("PJ ou CLT, temos a vaga certa para você!"). A página inclui o botão "Cadastre-se" e ilustrações vetoriais, servindo como porta de entrada unificada para profissionais e recrutadores.



Figura 3 – Tela Institucional "Sobre Nós"

Interface da seção "Sobre nós", dedicada a apresentar a identidade organizacional da plataforma TechJobs. A tela conta com uma ilustração temática e um texto de apresentação focado no propósito do sistema: conectar empresas e candidatos da área de tecnologia de maneira única e eficiente.

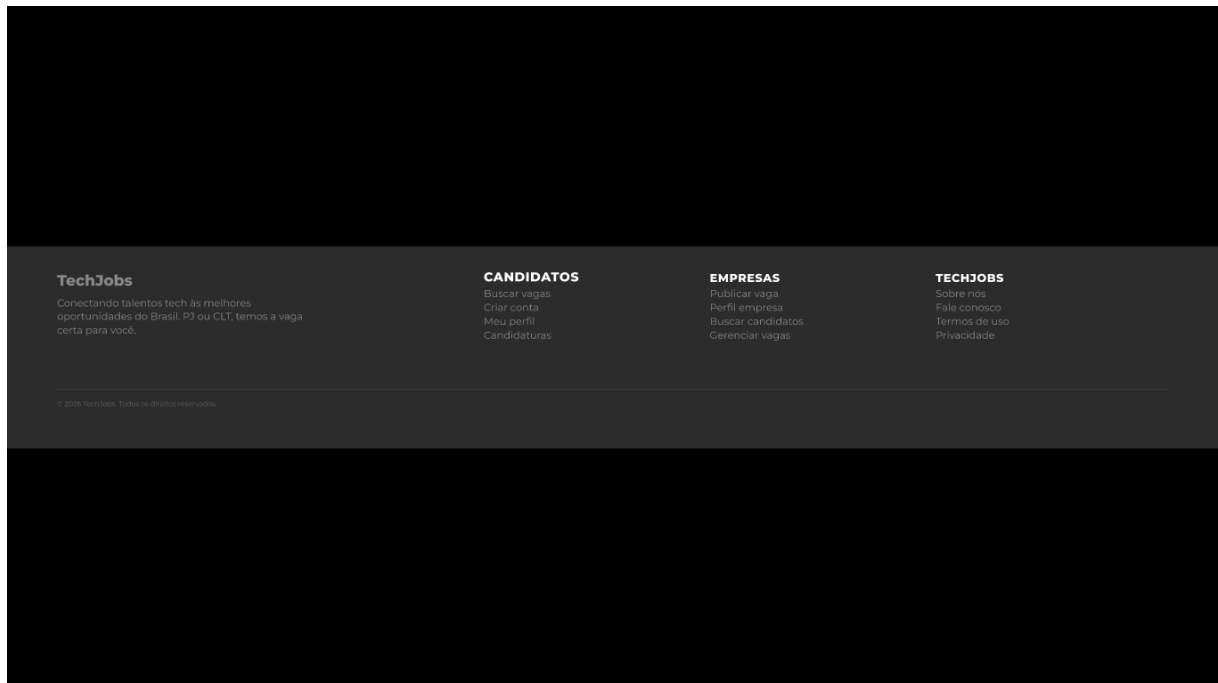
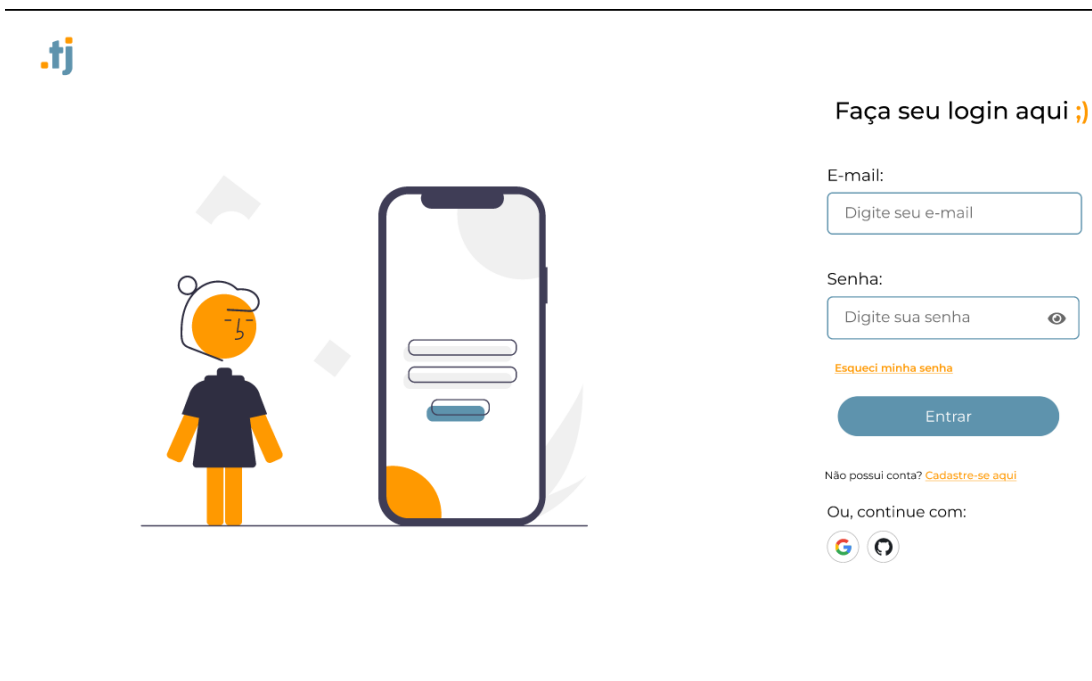


Figura 4 – Estrutura do Rodapé (Footer) [Empresa 1-2]

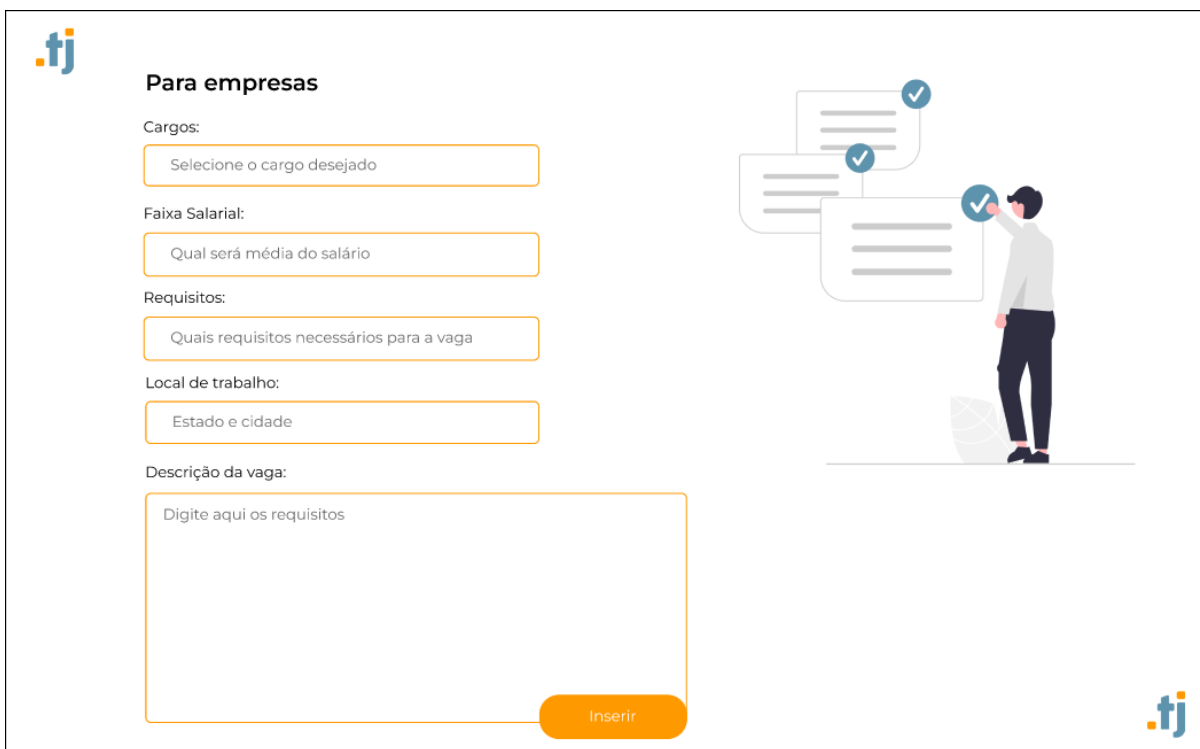
Visão detalhada do rodapé padrão da aplicação web. A estrutura organiza os links de navegação em três agrupamentos semânticos: links úteis para "Candidatos" (busca de vagas, candidaturas), atalhos para "Empresas" (publicação e gerenciamento de vagas) e páginas institucionais. O rodapé utiliza contraste escuro para separar a área de navegação de base do restante do conteúdo.



The login interface features the 'tj' logo in the top left corner. On the left, there is an illustration of a person with an orange head and a dark blue body standing next to a large smartphone. The smartphone screen displays a login form with fields for email and password, and a blue 'Entrar' button. To the right of the illustration, the text 'Faça seu login aqui ;)' is displayed. Below this, there are input fields for 'E-mail:' and 'Senha:', each with a placeholder text 'Digite seu e-mail' and 'Digite sua senha' respectively. A link 'Esqueci minha senha' is located below the password field. A blue 'Entrar' button is positioned below the 'Esqueci minha senha' link. At the bottom, there is a link 'Não possui conta? Cadastre-se aqui' and a section 'Ou, continue com:' with icons for Google and GitHub.

Figura 5 – Interface de Autenticação (Login)

Tela de acesso ao sistema (Login). O formulário solicita as credenciais padrão (e-mail e senha) e oferece a opção de recuperação de acesso. A interface também integra botões para Single Sign-On (SSO), permitindo que os recrutadores acessem a plataforma utilizando contas corporativas vinculadas ao Google ou GitHub.



The job posting form is titled 'Para empresas' and is located on the left side of the page. It contains several input fields: 'Cargos:' with a placeholder 'Selecione o cargo desejado', 'Faixa Salarial:' with a placeholder 'Qual será média do salário', 'Requisitos:' with a placeholder 'Quais requisitos necessários para a vaga', 'Local de trabalho:' with a placeholder 'Estado e cidade', and 'Descrição da vaga:' with a placeholder 'Digite aqui os requisitos'. A blue 'Inserir' button is located at the bottom right of the 'Descrição da vaga:' field. On the right side of the page, there is an illustration of a person with a grey head and a dark blue body standing next to a large smartphone. The smartphone screen displays a list of job postings, each with a blue checkmark icon. The 'tj' logo is located in the bottom right corner of the page.

Figura 6 – Formulário de Cadastro de Vagas

Interface central para as organizações, destinada à criação de novas oportunidades de trabalho ("Para empresas"). O formulário é composto por campos estruturados que solicitam: seleção de cargo, definição de faixa salarial, requisitos técnicos necessários, local de trabalho (cidade/estado ou indicação de trabalho remoto) e uma área de texto expandida para a descrição completa da vaga. O layout é limpo para minimizar a carga cognitiva durante o preenchimento.

tj

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

Mande-nos sua mensagem!

Nome:

E-mail:

Mensagem:

Figura 7 – Tela de Suporte "Fale Conosco"

Interface de comunicação direta com os administradores da plataforma. A tela disponibiliza um formulário de contato contendo os campos de identificação (Nome e E-mail) e uma área de mensagem para envio de dúvidas técnicas, suporte ao recrutamento ou feedbacks de usabilidade.

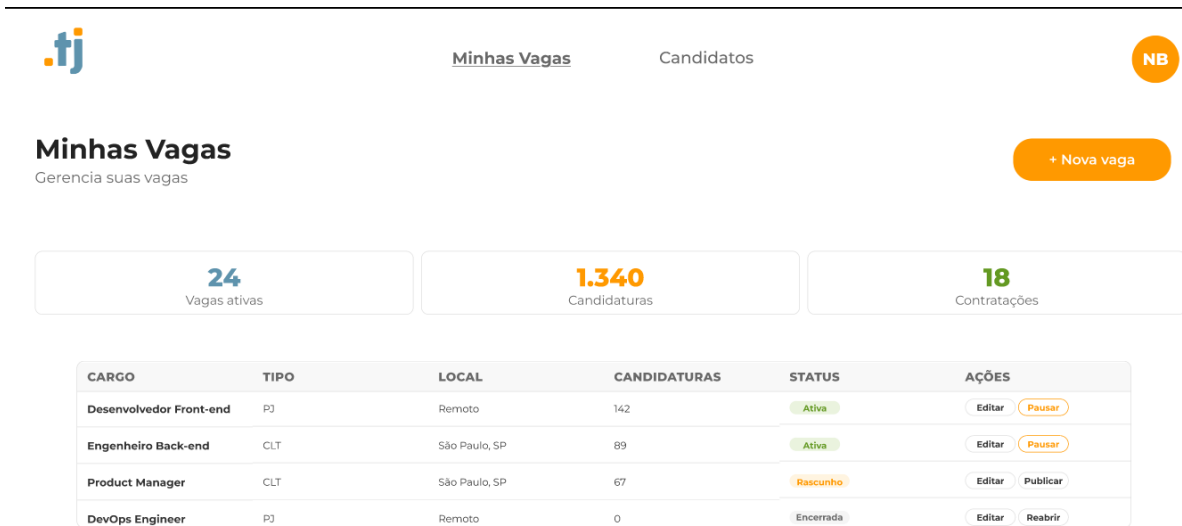


Figura 8 – Painel de Gerenciamento de Vagas

Dashboard operacional da empresa recrutadora ("Minhas Vagas"). A porção superior exibe cartões com indicadores-chave de desempenho (KPIs): número de vagas ativas, total de candidaturas recebidas e contratações efetivadas. A porção inferior apresenta uma tabela de gerenciamento (Data Grid) que lista as vagas cadastradas com informações de cargo, tipo de contrato, local e status (Ativa, Rascunho, Encerrada), além de botões de ação rápida para editar, pausar, publicar ou reabrir processos seletivos.

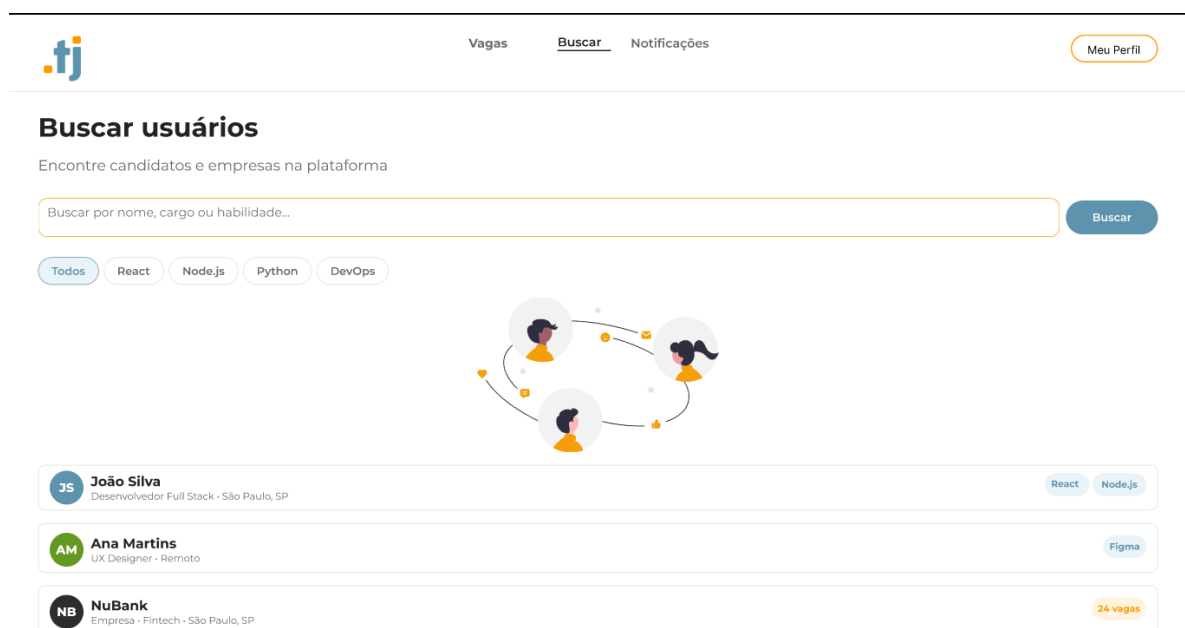


Figura 9 – Interface de Busca de Talentos

Tela do módulo de busca ativa (Sourcing) ("Buscar usuários"). A interface fornece uma barra de pesquisa unificada para busca por nome, cargo ou habilidade técnica, complementada por tags de filtros rápidos (React, Node.js, Python, DevOps). Os resultados são exibidos em formato de lista interativa, permitindo que a empresa visualize e acesse rapidamente o perfil dos candidatos correspondentes aos critérios digitados.

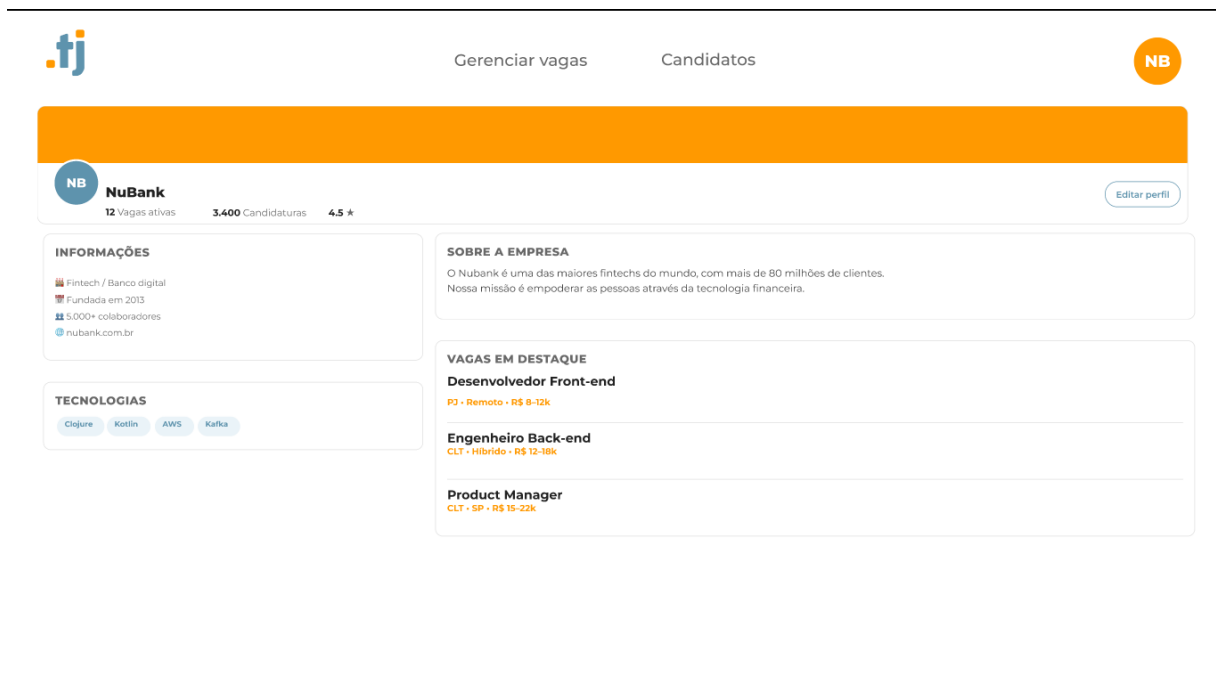


Figura 10 – Perfil Institucional da Empresa

Interface que apresenta a página pública de perfil da empresa recrutadora (Exemplo: NuBank). A tela é dividida em painéis modulares que exibem: métricas principais (vagas ativas, candidaturas, avaliação geral), informações institucionais (segmento, ano de fundação, porte da empresa, website), stack tecnológico utilizado (Clojure, Kotlin, AWS, etc.), um texto institucional "Sobre a empresa" e, por fim, a listagem atualizada das "Vagas em destaque" vinculadas àquela organização.

APÊNDICE B – INTERFACES GRÁFICAS DO MÓDULO MOBILE



Figura 11 – Interface de Autenticação Mobile Corporativa

Tela de Login adaptada para dispositivos móveis, sob a perspectiva da organização recrutadora. A interface preserva a fluidez e o minimalismo do ecossistema, contendo campos de entrada para e-mail e senha, atalhos para recuperação de credenciais e registro de nova empresa. Adicionalmente, integra botões para autenticação via Single Sign-On (SSO) com Google e GitHub, otimizando o fluxo de acesso em smartphones.

9:41

tj

Seja bem-vindo, Empresa

Candidato Empresa

Nome:

Seu nome

E-mail:

Digite seu e-mail

Senha:

Crie uma senha

Confirmar senha:

Confirme a senha

CNPJ:

Digite seu CNPJ

Finalizar cadastro

Figura 12 – Tela de Cadastro Mobile

Interface de registro de novos usuários no aplicativo. O cabeçalho apresenta uma mensagem de boas-vindas ("Seja bem-vindo, Empresa") 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 e número do CNPJ), finalizando com o botão de submissão do cadastro na área de alcance fácil dos dedos (thumb zone).

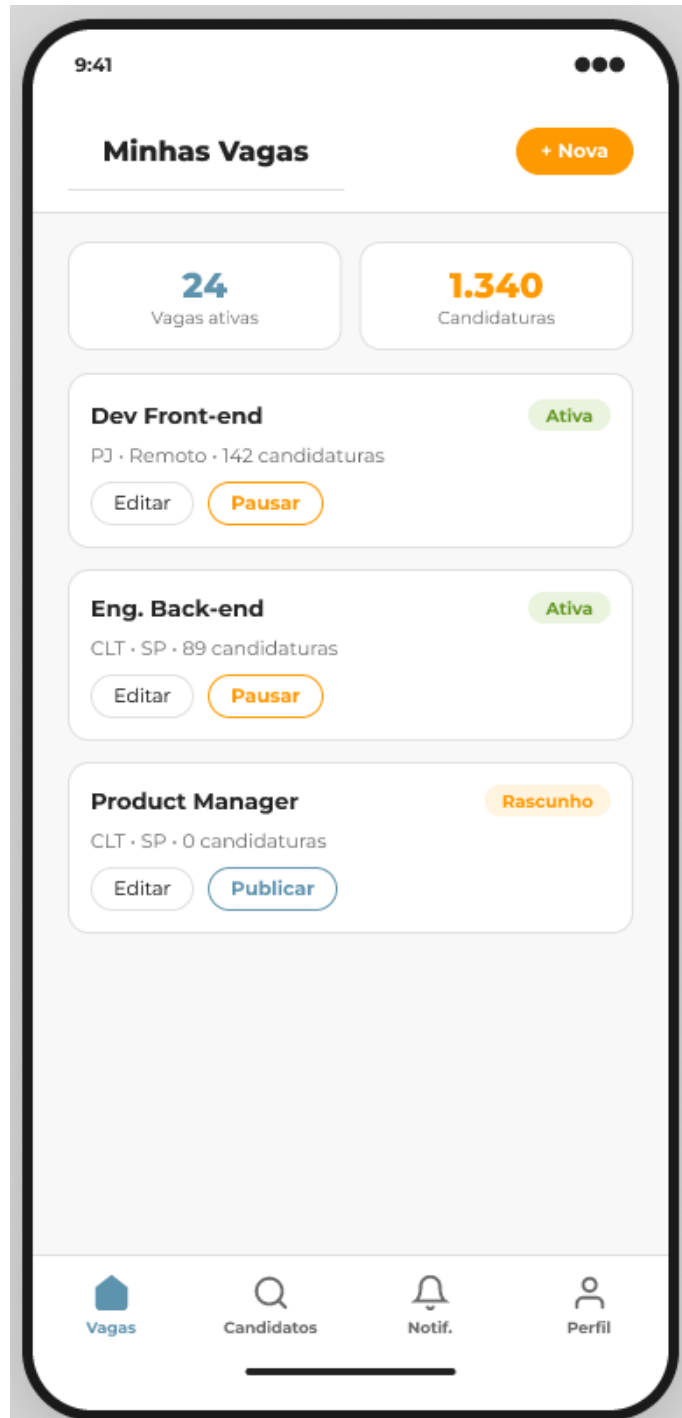


Figura 12 – Painel de Gerenciamento de Vagas Mobile

Dashboard operacional da empresa na versão móvel ("Minhas Vagas"). O topo da interface consolida indicadores-chave (KPIs) em cartões numéricos, exibindo o total de vagas ativas e candidaturas recebidas. O corpo da tela lista as oportunidades em cartões (cards) individuais que informam o cargo, métricas de inscritos, um selo visual de status ("Ativa" ou "Rascunho") e botões de ação rápida ("Editar", "Pausar", "Publicar"), permitindo o controle integral do processo seletivo pelo celular. A barra de navegação inferior (Bottom Navigation Bar) evidencia o módulo ativo.

The image shows a mobile application interface for creating a new job vacancy. At the top, the status bar shows the time 9:41 and battery level. The app header has the title "Nova Vaga" and an orange "Salvar" button. The form is organized into sections: "Cargo:" with a text field containing "Dev Senior"; "Tipo de emprego" with a dropdown menu showing "PJ"; "Faixa Salarial" with a text field containing "Qual será a média de salário"; "Requisitos" with a text field containing "Quais os requisitos mínimos"; "Local" with a text field containing "São Paulo, Brasil"; and "Descrição" with a large text area containing "Adicione uma descrição". At the bottom, there is a navigation bar with four icons: a house for "Vagas", a magnifying glass for "Candidatos", a bell for "Notif.", and a person for "Perfil".

9:41

Nova Vaga Salvar

Cargo:

Dev Senior

Tipo de emprego

PJ

Faixa Salarial

Qual será a média de salário

Requisitos

Quais os requisitos mínimos

Local

São Paulo, Brasil

Descrição

Adicione uma descrição

Vagas Candidatos Notif. Perfil

Figura 13 – Formulário de Criação de Vaga Mobile

Interface responsiva destinada ao cadastro de novas oportunidades de emprego ("Nova Vaga"). A tela estrutura-se em um formulário de rolagem vertical, utilizando campos de texto convencionais e menus suspensos (dropdowns) nativos para a seleção do tipo de contrato. O design otimiza o espaço útil da tela ao posicionar a principal chamada para ação — o botão "Salvar" — de forma acessível diretamente no cabeçalho superior (Top App Bar).

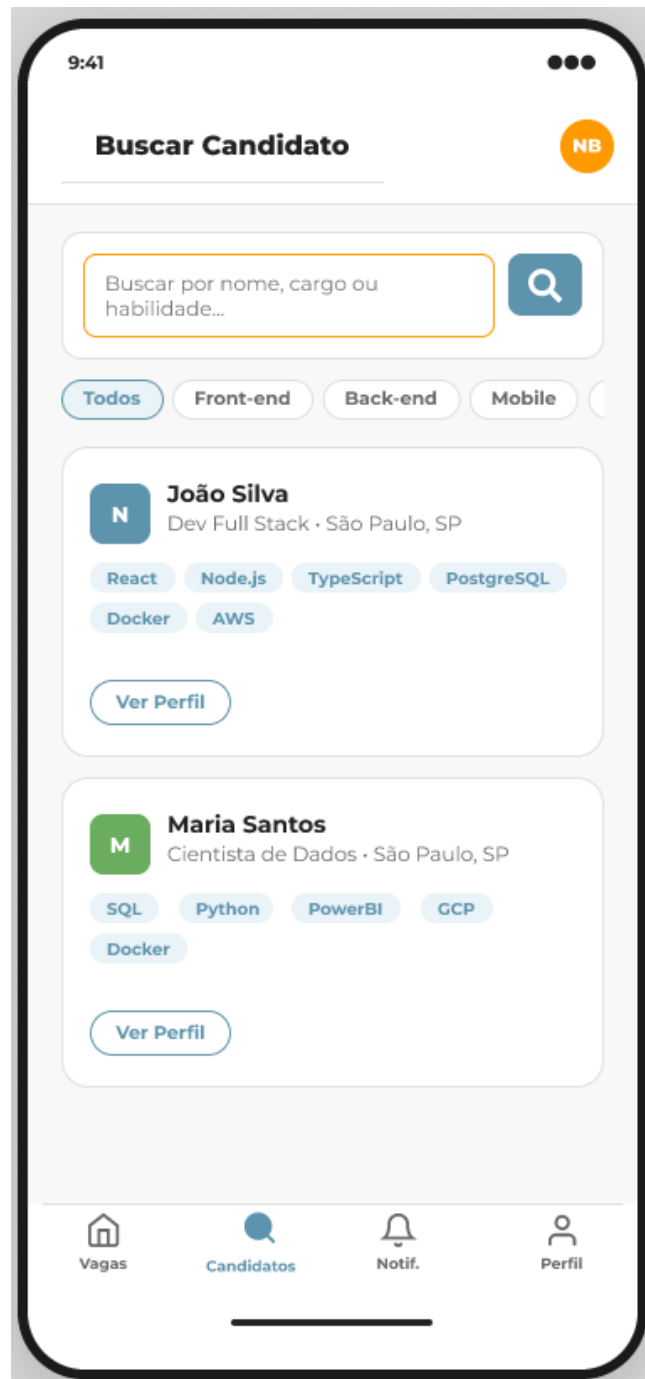


Figura 14 – Interface de Busca de Talentos Mobile

Tela do módulo de busca ativa de candidatos (Sourcing) adaptada para telas reduzidas. A interface oferece uma barra de pesquisa textual (por nome, cargo ou habilidade) atrelada a um carrossel horizontal de filtros rápidos em formato de pílulas (chips). Os resultados são exibidos em cartões de perfil que sintetizam o nome, a senioridade, a localização e as tags de habilidades técnicas do candidato, acompanhados do botão de direcionamento "Ver Perfil".

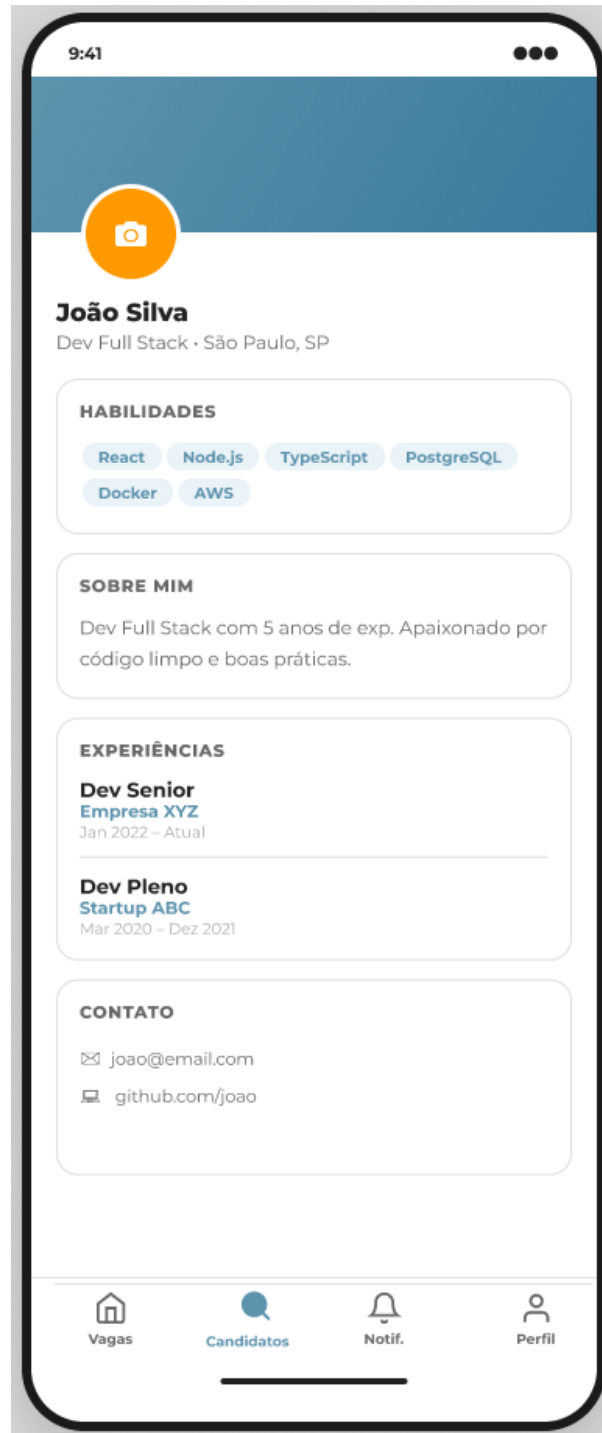


Figura 15 – Visualização do Perfil do Candidato Mobile

Interface de detalhamento do currículo digital do candidato sob a ótica de análise do recrutador. O layout adota um padrão de escaneabilidade da informação, organizando os dados em blocos modulares (cards) que separam claramente as "Habilidades", "Sobre mim", "Experiências" e "Contato". O cabeçalho fixa a identificação visual e textual do talento. O menu inferior de navegação mantém-se visível, garantindo o retorno rápido à lista de busca.

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): Giovanni Trimmer Moraes, José Vítor Rodrigues Cavalcanti, Mateus Moretti Scomparim Ruiz e Vitória Mota Dominichelli			
Orientador:			
Banca avaliadora:			
Produto desenvolvido: Plataforma Web e Mobile: TechJobs: Empresa			
Especificações técnicas: Arquitetura cliente-servidor distribuída. Aplicativo móvel (Mobile) desenvolvido com o framework Flutter (linguagem Dart). Interface corporativa Web (Frontend) construída com a biblioteca React (JavaScript). Serviços de retaguarda e lógica de negócios (Backend) implementados em API RESTful utilizando a linguagem C# e o framework .NET. Armazenamento e persistência de dados gerenciados pelo SGBD relacional SQL Server.			
Palavras-chave/Descritores: Recrutamento tecnológico; Desenvolvimento multiplataforma; Arquitetura de software; Aplicações web e mobile; Correspondência de perfis (Matchmaking).			
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.