

## CLICK VAGA: UMA APLICAÇÃO PARA CONECTAR EMPRESAS E CANDIDATOS A ESTÁGIO E EMPREGO

Gabriel Henrique Leal Garcia  
Graduando em Engenharia de Software – Uni-FACEF  
gabrielgarciasp@gmail.com

Willian Henrique Neves Filho  
Graduando em Engenharia de Software – Uni-FACEF  
willianhenrique\_neves@outlook.com

Carlos Eduardo de França Roland  
Mestre em Desenvolvimento Regional – Uni-FACEF  
roland@facef.br

### Resumo

Não é de hoje que se ouve falar sobre o desemprego e que ele vem aumentando gradativamente com crises e afins. Com o avanço pandêmico do Coronavírus e suas variações, vê-se um grande aumento de pessoas desempregadas. Por outro lado, também há empresas com dificuldade para encontrar profissionais capacitados para preencher suas vagas. Sendo assim, este artigo tem o objetivo de apresentar a solução para esta questão de pesquisa, conectando candidatos a estágio e emprego e empresas de forma simples e com poucos cliques. Na proposta, o empregador poderá fazer uma prospecção ativa, consultando candidatos através de *tags* relevantes para a vaga de emprego aberta. Quanto ao candidato, ele deverá poder ter o seu currículo avaliado por um profissional de Recursos Humanos ou de Psicologia para melhor apresentação ao mercado. O resultado obtido com o projeto foi a implementação de um *Minimum Viable Product* (MVP). No futuro este MVP poderá ser continuado e melhorado, o que permitirá sua comercialização.

**Palavras-chave:** Aplicação web. Desemprego. Vagas de emprego. Vagas de estágio.

### Abstract

*It is not new that one hears about unemployment and that it has been gradually increasing with crises and the like. As the pandemic advances, there is a huge increase in unemployed people. On the other hand, there are also companies struggling to find qualified professionals to fill their vacancies. Therefore, this article aims to present the solution to this research question, connecting internship and job candidates and companies in a simple way and with a few clicks. In the proposal, the employer will be able to do an active prospecting, consulting candidates through tags relevant to the open job vacancy. As for the candidate, he should be able to have his resume evaluated by an HR or Psychology professional for better presentation to the market. The result obtained with the project was the implementation of an Minimum Viable Product (MVP). In the future, this prototype can be continued and improved, which will allow its commercialization.*

**Keywords:** Internships. Jobs. Unemployment. Web application.

## 1 Introdução

Com base no cenário pandêmico de 2020 e 2021, presenciamos muitas pessoas perdendo seus empregos e enfrentando dificuldade para se recolocar. Por outro lado, com o avanço da vacinação, dentre outras medidas sanitárias, vimos que as empresas estão retomando as contratações, mas enfrentam um impasse para encontrar profissionais capacitados com as características necessárias para determinada função.

Segundo o portal G1 (2020, *online*), o desemprego decorrente da Covid-19 atingiu 1,5 milhão de pessoas no mês de junho de 2020. Em outra notícia do site em setembro de 2020, o desemprego chegou a uma alta de 27,6%. Por outro lado, em uma publicação do portal Uol (2020, *online*), várias empresas estão enfrentando dificuldades para contratar profissionais, em especial na área de TI.

Com base nessas evidências, o objetivo deste artigo é apresentar uma solução para facilitar a integração entre candidatos e vagas, tornando esse processo rápido e satisfatório para os dois lados.

Os artefatos da Engenharia de Software foram utilizados para facilitar a visão do projeto e servir como documentação de referência para a implementação do protótipo. Documentos como BPMN, Diagrama de Caso de Uso e Diagrama de Entidade-Relacionamento, são apresentados neste trabalho.

Para arquitetar uma aplicação que atendesse às necessidades do público-alvo, através de pesquisa de campo *online*, coletamos dados de pessoas que estão procurando emprego ou que já estão empregadas e realizamos o envio de um segundo formulário para recrutadores de empresas, para entendermos melhor as dificuldades enfrentadas num processo de *UX Research*. Em paralelo, realizamos o estudo das soluções existentes no mercado, para levantarmos funcionalidades essenciais e pontos de melhoria (*Benchmark*).

No aspecto técnico do projeto, foram definidas tecnologias para o desenvolvimento rápido, mas que ao mesmo tempo não renunciem à segurança e à performance do produto. Para isso, o protótipo foi desenvolvido utilizando a linguagem de programação JavaScript, juntamente com suas tecnologias integradas, como TypeScript e Node.js. Pensando em persistência de dados, foi utilizado o Sistema de Gerenciamento de Banco de Dados (SGBD) *MySQL*, em função do seu custo-benefício e desempenho com os recursos de Mapeamento Objeto Relacional (*Object Relational Mapper - ORM* do termo em inglês) disponíveis para JavaScript.

Ao longo deste artigo, o leitor encontra, na seção 2, a apresentação das tecnologias que compõem este trabalho. Na seção 3 é contextualizada a Qualidade de Software que norteou o projeto; na seção 4 são apresentados detalhadamente o problema e a proposta de solução. Já na seção 5, é apresentado o modelo de negócios do projeto através do Canvas; na seção 6, são mostrados os artefatos da Engenharia de Software que foram criados, como o BPMN por exemplo. Na seção 7, é apresentada a prototipação da interface do produto, para em seguida, na seção 8, ser descrita a implementação do software. Por fim seguem as considerações sobre a execução do projeto.

## 2. Tecnologias Envolvidas no Projeto

A seguir, é apresentada a contextualização das tecnologias utilizadas para implementação do software.

## 2.1 JavaScript

De acordo com FLANAGAN (2013), JavaScript é uma linguagem de programação, voltada a especificar comportamentos de componentes de uma página web.

[...] JavaScript é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais. A sintaxe de JavaScript é derivada da linguagem Java, das funções de primeira classe de Scheme e da herança baseada em protótipos de Self (FLANAGAN, 2013, p. 1).

Ainda segundo o autor, JavaScript se tornou a linguagem mais onipresente da história, pois a maioria dos sites a utilizam e também diversos dispositivos conseguem interpretá-la, como computadores, videogames, *smartphones* e *tablets*.

Diferente do que muitos pensam, JavaScript não é o mesmo que “Java Interpretado”, mas uma tecnologia totalmente diferente e única. Considerando um nível mais profundo da linguagem, uma característica que chama a atenção é a possibilidade de utilização do paradigma funcional e da orientação a objetos, sem a necessidade de qualquer configuração ou adaptação para tal utilização MOZILLA (2021, *online*).

## 2.2 Node.JS

Segundo PEREIRA (2014), com o lançamento do Node.js em 2009, o mundo das linguagens *server-side* conheceu um novo modelo, pois então seria possível a utilização de JavaScript do lado servidor. Uma característica que difere o Node.js dos concorrentes é que ele trabalha de forma *single-thread*, com uma solução chamada de *Event-loop*, que trata cada requisição como um evento e os armazena em uma pilha. Outra característica é a utilização de chamadas não-bloqueantes, ou seja, o Node.js pode realizar várias chamadas de forma assíncrona, sem a necessidade de esperar uma chamada retornar para chamar outra.

O legal é que o mecanismo JavaScript é independente do navegador no qual está hospedado. Esse recurso-chave possibilitou o surgimento do Node.js. O V8 foi escolhido para ser o mecanismo que impulsionou o Node.js em 2009 e, conforme a popularidade do Node.js explodiu, o V8 se tornou o mecanismo que agora fornece uma quantidade incrível de código do lado do servidor escrito em JavaScript (NODE FOUNDATION, 2021, *online*, tradução nossa).

## 2.3 TypeScript

Segundo TYPESCRIPT (2021, *online*), TypeScript tem o objetivo de auxiliar desenvolvedores JavaScript, adicionando verificação de tipo de dados (tipagem) no código. Todo o código escrito em TypeScript é posteriormente transformado em JavaScript, para que então os navegadores ou o Node.js consigam interpretar o código.

De acordo com ADRIANO (2021, p. 9), “Ele foi projetado para auxiliar no desenvolvimento de códigos simples até os mais complexos, utilizando os princípios de Orientação a Objetos, como classes, tipagens, interfaces, *generics* etc.”. Ainda

segundo o autor, ao adicionar tipagem no código, é possível obter um melhor desempenho no momento de depuração do projeto.

## 2.4 MySQL

De acordo com MYSQL (2021, *online*), MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), de código aberto, distribuído e mantido pela *Oracle Corporation*.

O MySQL Server foi originalmente desenvolvido para lidar com grandes bancos de dados com muito mais rapidez do que as soluções existentes e tem sido usado com sucesso em ambientes de produção altamente exigentes por vários anos. Embora em constante desenvolvimento, o MySQL Server hoje oferece um conjunto rico e útil de funções. Sua conectividade, velocidade e segurança tornam o MySQL Server altamente adequado para acessar bancos de dados na Internet (MYSQL, 2021, *online*).

Segundo CARVALHO (2015, p. 15), quando vamos iniciar um projeto, é sempre uma grande incógnita qual tecnologia utilizar. É aconselhável sempre levar em conta o preço do licenciamento para utilização de uma determinada tecnologia e a qualidade oferecida.

Ao escolher o MySQL como opção de Sistema Gerenciador de Banco de Dados (SGBD), além de uma ferramenta gratuita criada na base da licença de software livre, você também está optando por qualidade, robustez e segurança. Estes são adjetivos que um gerenciador deve ter, pois guardar seus dados ou de seus clientes com segurança é o mais importante (CARVALHO, 2015, p. 15).

## 3. Qualidade de Software

PRESSMAN E MAXIM (2016, p.414) afirmam que “qualidade de software pode ser definida como: uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que o produzem e para aqueles que o utilizam”. Podemos tratá-la como uma função que recebe diversas variáveis que são preenchidas com dados que se diferem de acordo com o projeto e os fatores de satisfação dos *stakeholders*.

SOMMERVILLE (2013, p.457) afirma que a qualidade de software pode ser subjetiva, tendo que ser analisada a partir de parâmetros de avaliação do software pré-definidos. Esses parâmetros podem envolver documentação de requisitos, regras de negócio, processos e código, nível dos testes executados, nível de confiabilidade, nível de desempenho, viabilidade e construção de um produto útil, cada um tendo o seu devido peso de avaliação pois não é possível criar um sistema onde todos os parâmetros de avaliação atinjam o mais alto nível.

É impossível que algum sistema seja otimizado em todos esses atributos — por exemplo, melhorar a robustez pode levar à perda de desempenho. O plano de qualidade, portanto, deve definir os atributos de qualidade mais importantes para o software que está sendo desenvolvido. Pode ser que a eficiência seja crítica e outros fatores tenham de ser sacrificados para obtenção disso. Se no plano de qualidade você estabeleceu que um atributo é crítico, os engenheiros que trabalham no desenvolvimento podem cooperar para esse atributo. O plano deve incluir também uma definição de processo de avaliação de qualidade. Isso deve ser uma maneira de avaliar se alguma

qualidade, como manutenibilidade ou robustez, está presente no produto (SOMMERVILLE, 2013, p.457).

Tanto o fornecedor da solução, quanto o usuário que irá consumir o software ganham com a sua qualidade. Os benefícios vão além do financeiro e atingem barreiras de experiência do usuário e facilidade para manutenção do produto.

Ao agregar valor tanto para o fabricante quanto para o usuário de um produto de software, um software de alta qualidade gera benefícios para a empresa de software e para a comunidade de usuários. A empresa fabricante do software ganha valor agregado pelo fato de um software de alta qualidade exigir menos manutenção, menos correções de erros e menos suporte ao cliente. Isso permite que os engenheiros de software despendam mais tempo criando aplicações novas e menos tempo em manutenções. A comunidade de usuários ganha um valor agregado, pois a aplicação fornece a capacidade de agilizar algum processo de negócio. O resultado é: (1) maior receita gerada pelo produto de software, (2) maior rentabilidade quando uma aplicação suporta um processo de negócio e/ou (3) maior disponibilidade de informações cruciais para o negócio (MAXIM e PRESSMAN, 2016, p.415).

#### 4. Contexto do Problema e Solução

Segundo dados do IBGE, no trimestre de maio a julho de 2021, a taxa de desemprego chegou a 13,7% da população em idade para trabalhar, aproximadamente 14,1 milhões de desempregados. O desemprego afeta diretamente a economia e a situação familiar, visto que sem a renda da prestação de serviço, com o tempo, não existirá dinheiro suficiente para a compra de alimentos para refeições diárias.

Por sua vez, a taxa de desemprego impacta proporcionalmente na procura por emprego, onde a busca é superior às vagas de empregos populares existentes. Esse problema somado com a falta de esmero no processo seletivo gera novas dificuldades. Foi constatado através de pesquisa bibliográfica (*Desk Research*) que os empregadores falham na divulgação de suas vagas de emprego, redigindo a descrição da vaga com falta de clareza no objetivo da função que deverá ser exercida, portanto, alimentando a possibilidade de más contratações. Após uma prospecção ativa das soluções existentes no mercado, foi constatado que nenhuma oferece soluções para todas as necessidades levantadas.

Considerando a incidência dos problemas listados, tem-se aumento na taxa de desemprego, impacto negativo na produção pelas empresas em razão da má contratação, alta rotatividade de funcionários, redução de faturamento mensal e fechamento de empresas.

Face tal cenário, o Click Vaga foi proposto como uma plataforma de software que oferece uma solução prática, intuitiva e funcional, necessária tanto para o candidato quanto para o empregador. Junto ao estudo das soluções existentes no mercado, conseguimos lapidar uma proposta que atendesse da melhor forma os dois lados (Candidatos e Empresas). Analisamos que para o cadastro dos Candidatos, deveria ser solicitado o menor número de dados pessoais, possibilitando se ingressar somente com os dados absolutamente necessários para uso da plataforma. Após se cadastrar, os Candidatos poderão inserir seus currículos, preenchendo campos pré-definidos pela plataforma, mas que podem ser informados de maneira diferente para cada currículo. Com isso o Candidato poderá criar vários modelos de currículo, o que lhe permitirá escolher qual modelo enviar para determinada vaga.

Do outro lado temos as empresas, ou melhor, os Recrutadores das empresas. Eles iniciam seu cadastro na plataforma como qualquer outro usuário, mas marcam seu perfil como um de Recrutador. Esta funcionalidade permite que este grupo de usuários consiga cadastrar vagas e controlar processos seletivos. Para o cadastro de vagas, alguns campos são de preenchimento obrigatório e foram levantados em uma pesquisa, que mostrou que candidatos sentiam falta de algumas informações em vagas de outras soluções existentes, como por exemplo o salário ou carga horária semanal.

Para realizar a conexão candidato e vaga, a plataforma disponibiliza dois processos. O primeiro permite que os usuários Candidatos possam visualizar as vagas que estão abertas, possibilitando se candidatar para cada uma delas. Já a segunda forma de realizar esta conexão, é do lado da empresa. Recrutadores após realizar o cadastro de uma vaga, poderão pesquisar possíveis candidatos que tenham perfil adequado aos requisitos necessários para aquela vaga. A plataforma realizará um processo de busca, onde serão verificados os requisitos da vaga e as habilidades dos candidatos. Então um relatório será gerado com esses possíveis candidatos, onde a empresa poderá entrar em contato e convidar o candidato para o processo seletivo.

Todos os currículos cadastrados poderão receber um selo de verificação, sendo esta funcionalidade opcional para o candidato. A revisão será feita por um profissional da área de Recursos Humanos (RH) ou da Psicologia, que irá avaliar se o currículo está adequado às boas práticas de apresentação profissional, sendo aprovado ou não. Para ambas as situações, o profissional avaliador poderá enviar um *feedback* para o candidato. Esta funcionalidade é um recurso *premium* e deve ser pago pelo candidato. Já os profissionais da área de RH e de Psicologia, são parceiros da plataforma e recebem uma porcentagem do valor para cada currículo analisado.

## 5. Modelo de Negócios

O Canvas é uma ferramenta utilizada na análise da viabilidade de um projeto, sendo possível ter um esboço de toda a estrutura do negócio. Esse *framework* estratégico foi criado para auxiliar a elaboração de um modelo de negócios e, portanto, é uma “linguagem comum para descrever, visualizar, avaliar e alterar modelos de negócio” (OSTERWALDER e PIGNEUR).

O modelo utilizado é composto por 9 grupos (Figura 1) que esclarecem o que se entrega de viável para a futura aquisição do cliente, para quem estamos entregando o produto, como planejamos entregá-lo e quais serão os custos operacionais e as receitas comerciais. A seguir são descritos cada um dos grupos.

**Figura 1 — Modelo Canvas do projeto**

Parceiros Chave	Atividades Chave	Proposta de Valor	Relação com o Cliente	Segmentos de Cliente
<ul style="list-style-type: none"> <li>- Profissionais de RH e Psicologia;</li> <li>- Faculdades do município da cidade de Franca-SP;</li> </ul>	<ul style="list-style-type: none"> <li>- Segurança;</li> <li>- Usabilidade;</li> <li>- Anuncio de vagas no LinkedIn;</li> <li>- Interação com usuários do LinkedIn;</li> </ul>	<ul style="list-style-type: none"> <li>- Plataforma web, com adaptação para visualização mobile;</li> <li>- Estabilidade e interoperabilidade;</li> <li>- Boa experiência do usuário;</li> <li>- Vagas de diversos tipos e para diversos segmentos diferentes;</li> <li>- Transparência;</li> <li>- Confiabilidade;</li> <li>- Alcance de anuncio de vaga;</li> <li>- Comodidade;</li> <li>- Visualizar currículos de candidatos;</li> </ul>	<ul style="list-style-type: none"> <li>- LinkedIn;</li> <li>- Email;</li> <li>- Plataforma online;</li> </ul>	<ul style="list-style-type: none"> <li>- Desempregados;</li> <li>- Pessoas já empregadas que procuram um novo desafio;</li> <li>- Empresários de empresas de tecnologia;</li> <li>- Empresários procurando a contratação de um bom funcionário;</li> </ul>
		<p><b>Recursos Chave</b></p> <ul style="list-style-type: none"> <li>- Plataforma web;</li> </ul>	<p><b>Canais</b></p> <ul style="list-style-type: none"> <li>- Pacotes de diferentes valores que dão direitos a difentes funcionalidades do sistema;</li> <li>- Impulsionar currículo;</li> <li>- Anuncios no Google Ads;</li> </ul>	
<p><b>Estrutura de Investimento</b></p> <ul style="list-style-type: none"> <li>- Hospedagem do site;</li> <li>- Profissional para montagem e publicação de anuncios de vagas;</li> <li>- Profissional para interação com usuários no LinkedIn;</li> <li>- Profissional para área de negócios;</li> <li>- Profissional para área de desenvolvimento;</li> </ul>			<p><b>Fontes de Receita</b></p> <ul style="list-style-type: none"> <li>- Assinatura de pacotes para anuncio de vaga e processo seletivo;</li> <li>- Impulsioneamento de currículo;</li> </ul>	

**Fonte:** os autores

1. Segmentos de Clientes: desempregados, pessoas já empregadas que procuram um novo desafio, empresários de empresas de tecnologia e empresários procurando a contratação de um bom funcionário.
2. Proposta de Valor: plataforma web, com adaptação para visualização *mobile*, interoperabilidade, foco em UX Design, vagas de diversas classificações e segmentos, transparência, confiabilidade, alcance de anúncio de vaga, comodidade e visualização de currículos de candidatos.
3. Canais: pacotes de diferentes valores que dão direitos a diferentes funcionalidades do sistema, impulsionar currículo e anúncios no Google Ads.
4. Relação com o Cliente: LinkedIn, e-mail e a nossa plataforma *online*.
5. Fontes de Renda: assinatura de pacotes para anúncio de vaga e processo seletivo e impulsionamento de currículo.
6. Recursos-chave: plataforma web.
7. Atividades-chave: segurança, usabilidade, anúncio de vagas e interação com usuários do LinkedIn.
8. Parceiros-chave: profissionais de RH e de Psicologia e faculdades de Franca-SP.
9. Estrutura de Custos: hospedagem do site, profissional para montagem e publicação de anúncios de vagas, profissional para interação com usuários no LinkedIn, profissional para área de negócios e profissional para área de desenvolvimento.

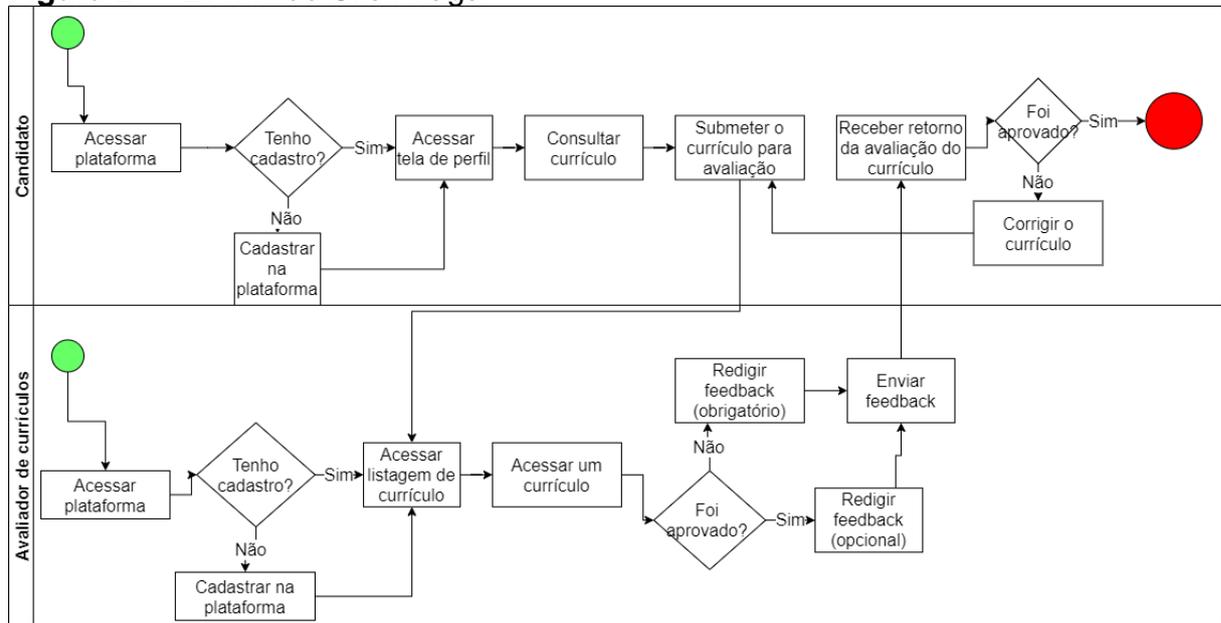
## 6. Artefatos da Engenharia de Software

A seguir, são apresentados os diagramas e artefatos da Engenharia de Software usados para o projeto da plataforma.

## 6.1. Business Process Model and Notation (BPMN)

BPMN é uma representação gráfica com o objetivo de deixar claro o fluxo dos processos a serem automatizados. Segundo SOMMERVILLE (2013), o BPMN é composto por atividades que são representadas em forma de retângulos, eventos que são representados por círculos, losangos que representam decisões e setas que representam a sequência de atividades do processo. A Figura 2 mostra a representação gráfica do processo principal.

**Figura 2 — BPMN do Click Vaga**

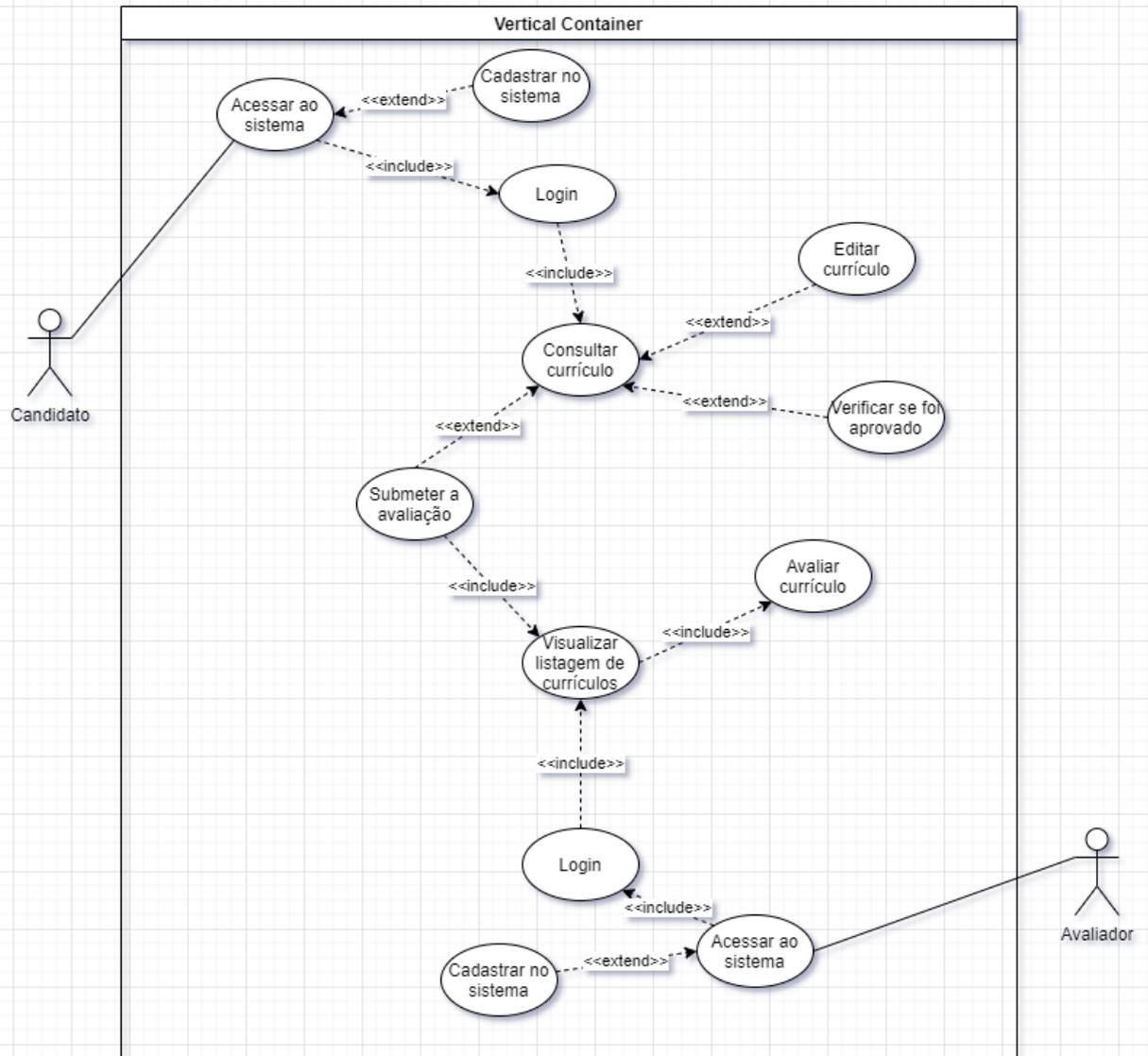


Fonte: os autores

## 6.2. Diagrama de Casos de Uso

Segundo FREITAS (2015), um sistema é representado em UML por 5 diferentes visões, cada uma definida por um conjunto de diagramas. Seguindo pela visão de modelo de usuário, temos o Caso de Uso, um diagrama importante para descrever os cenários de uso do sistema, diante da visão do usuário final.

**Figura 3 — Diagrama de Casos de Uso**



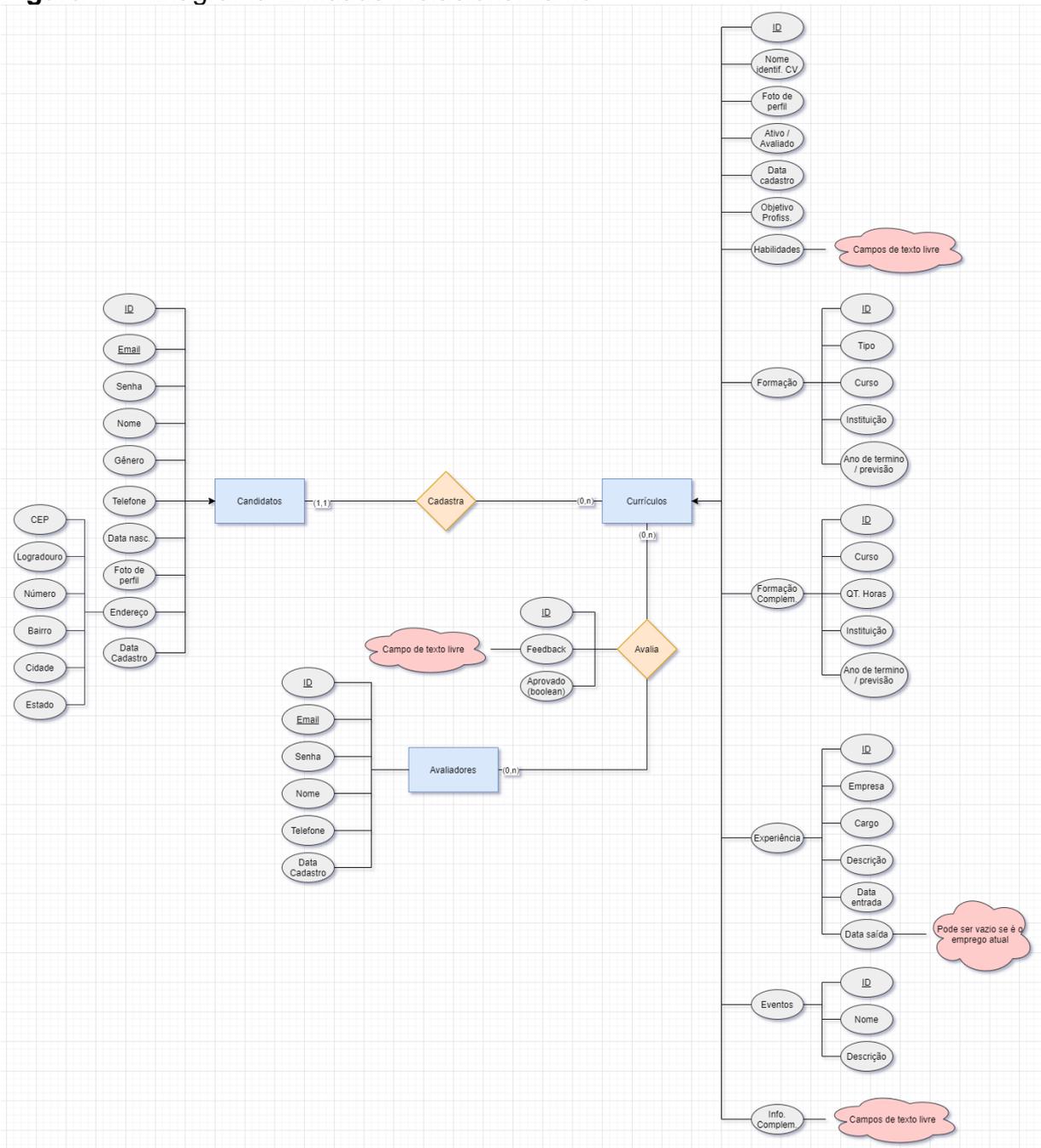
**Fonte:** os autores

### 6.3. Diagrama Entidade-Relacionamento (DER)

De acordo com JOEL (2014), o Diagrama Entidade-Relacionamento é a representação gráfica de Modelo Conceitual (MER). Este diagrama é muito importante para a engenharia do modelo de dados, uma vez que permite a visualização das informações com menos abstrações.

Um outro ponto muito importante citado por JOEL (2014), é que com o DER, o time fica mais integrado, uma vez que é uma linguagem mais alto nível e comum, facilitando o entendimento e o alinhamento de todos. A Figura 4 ilustra o diagrama elaborado para a plataforma.

**Figura 4 — Diagrama Entidade-Relacionamento**

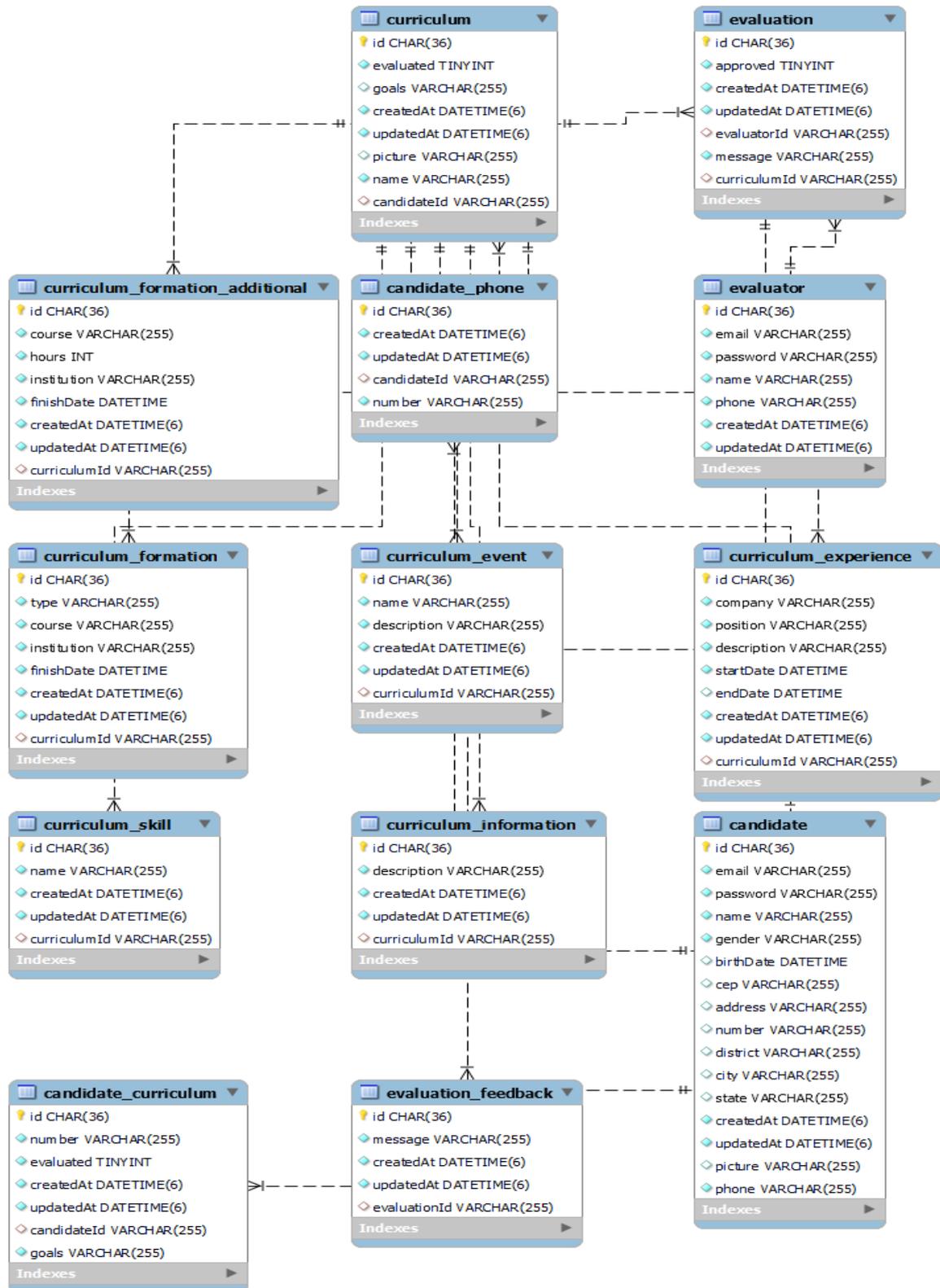


**Fonte:** os autores

## 6.4. Modelagem do Banco de Dados

Segundo o modelo apresentado no Diagrama Entidade-Relacionamento da Figura 5, a Modelagem do Banco de Dados é um artefato de mais baixo nível e que também auxilia o entendimento da persistência e do fluxo dos dados no sistema. Com ele, todos os membros da equipe, principalmente *Database Administrator* (DBA) e desenvolvedores conseguem projetar melhor o sistema, extrair relatórios e muito mais.

Figura 5 — Modelagem de Banco de Dados



Fonte: os autores

## 6.5. Demais Artefatos

Para melhor visualização e organização dos artefatos usados no projeto, como Mitigação de Riscos, Análise SWOT, Personas e muito mais, é disponibilizado o *link* do repositório no GITHUB em Garcia (2021).

## 7. Prototipação da Interface do Software

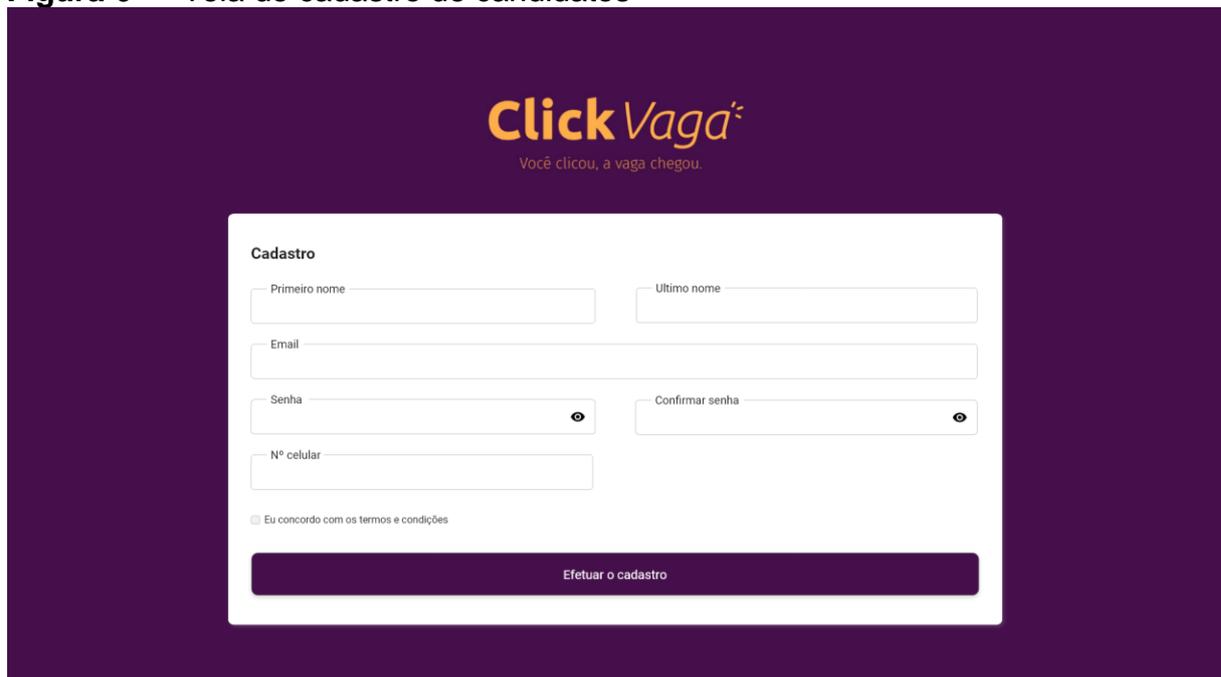
A seguir serão apresentadas as telas que compõem o fluxo operacional do protótipo. Todas as telas foram desenhadas utilizando o Adobe XD, uma ferramenta criada pela Adobe, com foco na prototipação de telas *web* e *mobile*.

Em decorrência do tempo para desenvolvimento do projeto, os artefatos apresentados a seguir estarão limitados ao escopo do protótipo (MVP), uma versão reduzida da solução projetada, tendo sido prototipadas as telas dos processos de Candidatos.

### 7.1. Cadastro do Candidato

Para que o usuário possa acessar a plataforma, ele necessita ter previamente uma conta cadastrada na base de dados. Para realizar este cadastro, ele pode acessar a tela da Figura 6 e preencher informações básicas e obrigatórias, que serão utilizadas posteriormente na plataforma. Essas informações são: primeiro nome, último nome, e-mail, senha, confirmação de senha e número de telefone. Após preencher todas as informações, o usuário precisa marcar a caixa de seleção que confirma que ele está de acordo com os termos e condições de uso definidos pela plataforma.

**Figura 6** — Tela de cadastro de candidatos



**Click Vaga**  
Você clicou, a vaga chegou.

**Cadastro**

Primeiro nome

Último nome

Email

Senha

Confirmar senha

Nº celular

Eu concordo com os termos e condições

**Efetuar o cadastro**

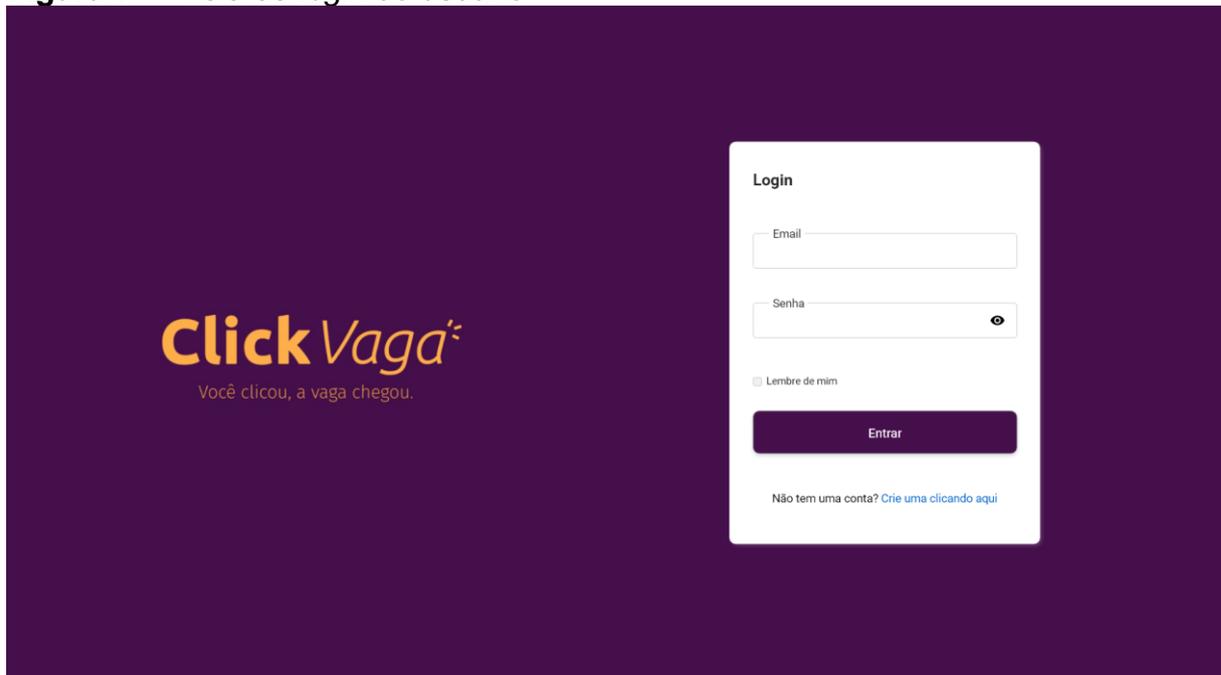
**Fonte:** os autores

## 7.2. Login do Usuário

Após o usuário realizar seu cadastro na plataforma, ele estará apto a começar a utilizar as funcionalidades. Para isso, o usuário deverá realizar seu *login* (Figura 7). Os campos Email e Senha são obrigatórios e deverão ser preenchidos conforme o informado na tela de cadastro.

Caso o usuário ainda não possua uma conta, nesta tela ele também encontra a opção para realizar seu cadastro. Ao clicar no *link*, ele será redirecionado para a tela de cadastro de usuários.

**Figura 7** — Tela de *login* do usuário



**Fonte:** os autores

## 7.3. Perfil do Candidato

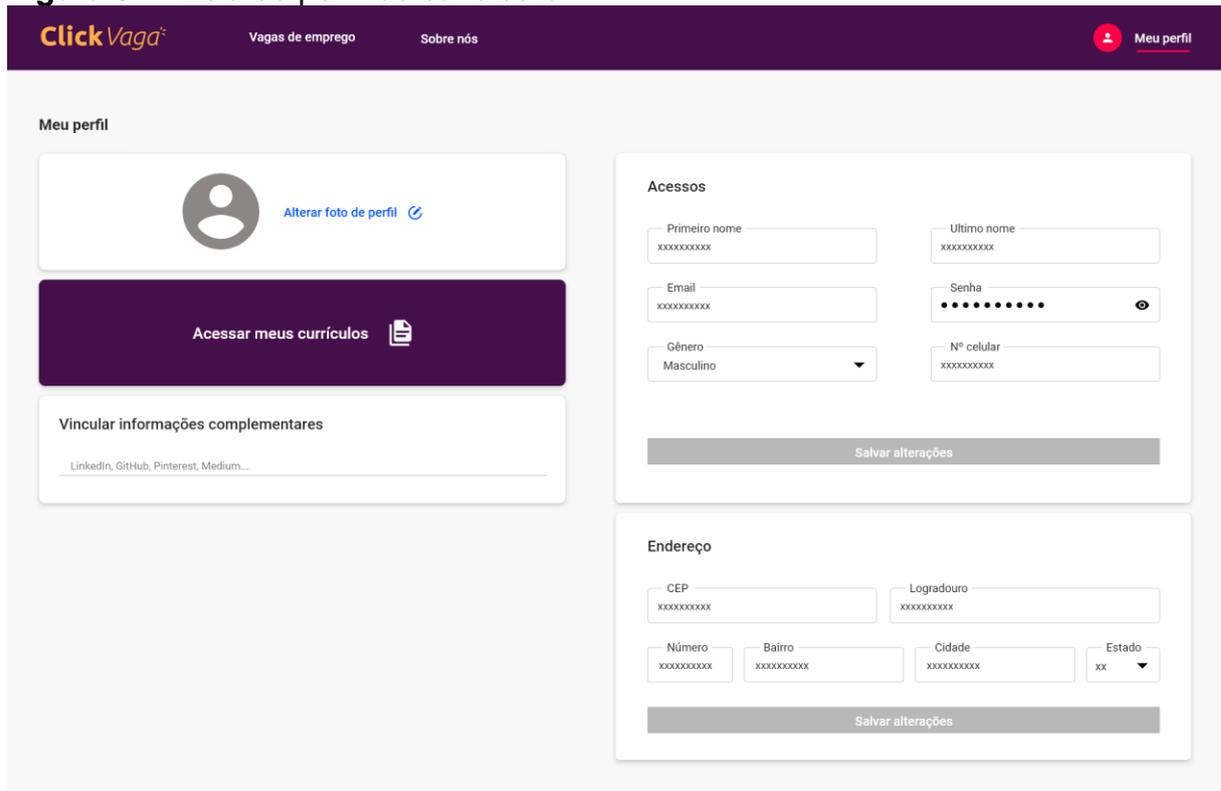
Esta é a primeira tela que o Candidato acessa, após sua autenticação. Ela possui quatro seções, sendo elas: Meu perfil, Vincular informações complementares, Acessos e Endereços (Figura 8).

Na seção Meu perfil, é possível enviar uma imagem como foto de perfil, conforme ilustração da Figura 9. O usuário também pode selecionar a opção para visualizar ou editar seus dados.

Já na seção Vincular informações complementares, o usuário poderá incluir os *links* para seu perfil em outras plataformas, como LinkedIn, Behance, Medium e outros. Após incluir esses dados, o usuário poderá visualizar as alterações conforme a ilustração da Figura 9.

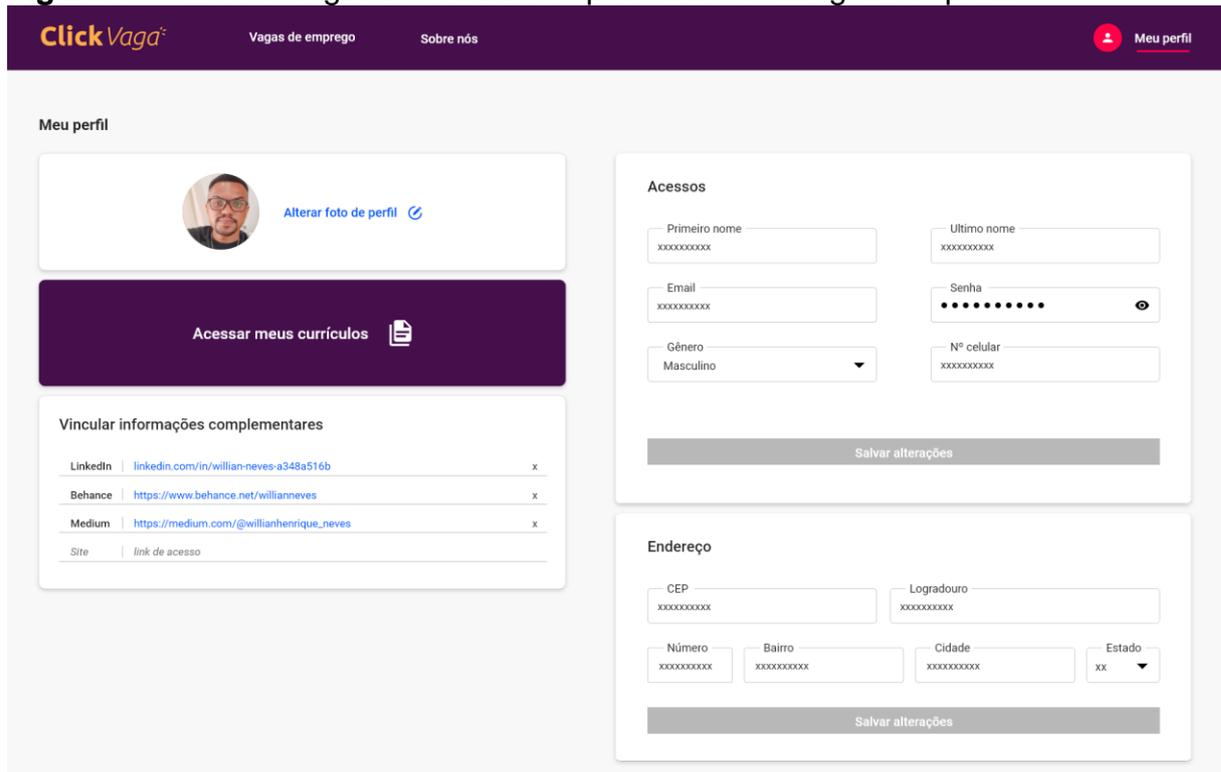
Por último, na seção Acessos e Endereço, o usuário poderá alterar os dados cadastrados na plataforma, para mantê-los o mais atualizados possível e não perder eventuais contatos de recrutadores.

**Figura 8** — Tela de perfil do candidato



Fonte: os autores

**Figura 9** — Tela de login do candidato após envio de imagem de perfil



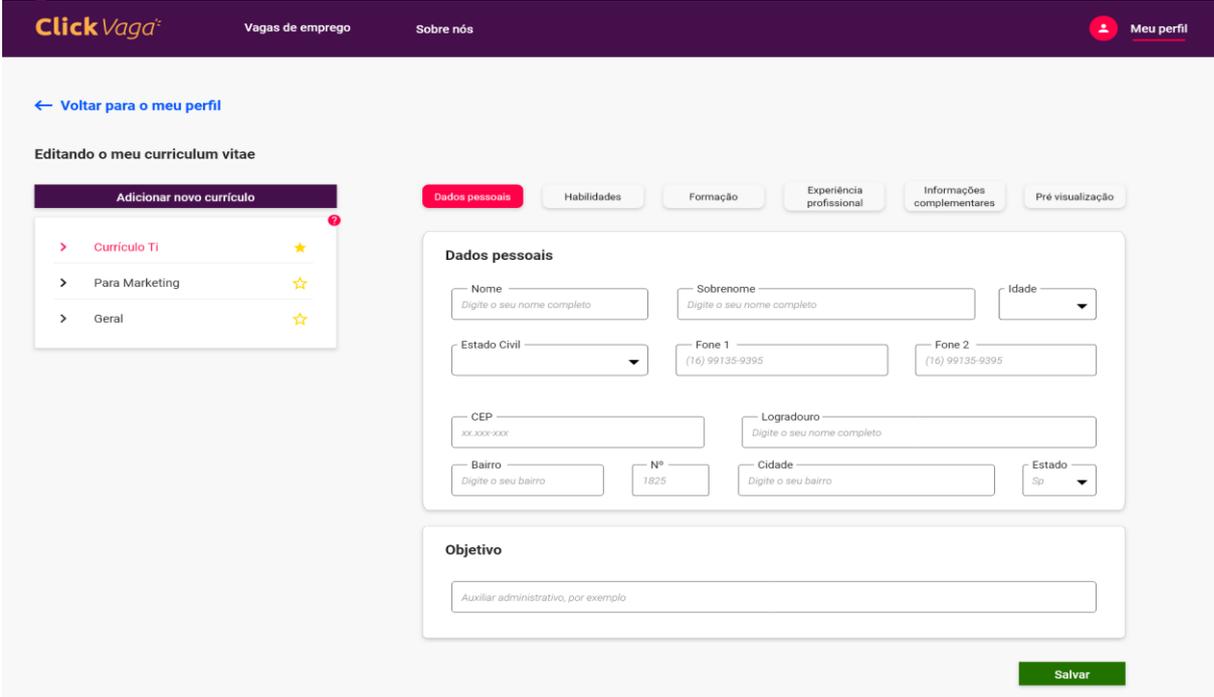
Fonte: os autores

## 7.4. Currículo do Candidato

Nesta tela (Figura 11) é onde o candidato poderá cadastrar ou editar seus currículos na plataforma. Estes currículos poderão ser posteriormente visualizados por recrutadores.

O cadastro ou edição do currículo é feito de forma segmentada (em forma de abas), para facilitar o preenchimento das informações. Ao clicar no botão Pré visualização, o usuário será levado para uma página de visualização deste currículo. Por fim, o botão Salvar irá persistir os dados na base de dados e levará o candidato para tela inicial do perfil (Figura 8).

**Figura 11** — Tela de currículo do candidato



**Fonte:** os autores

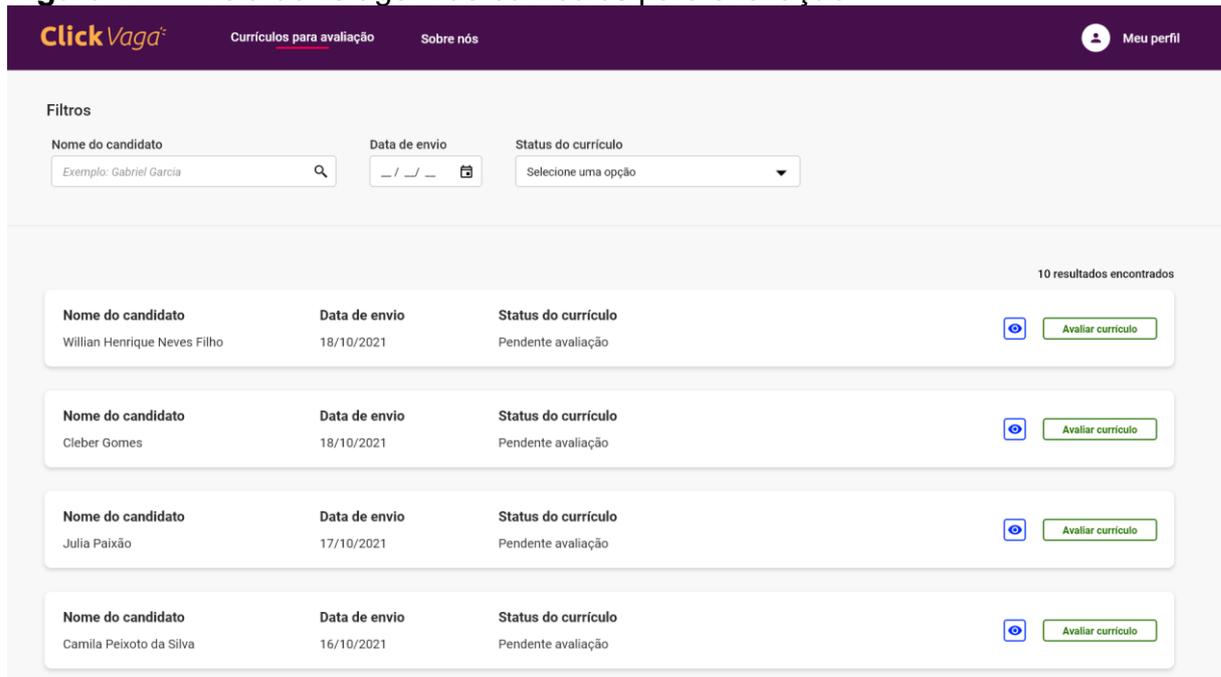
## 7.5. Listagem de currículos a serem avaliados

Após o candidato cadastrar o seu currículo, é possível enviá-lo para avaliação. Caso o candidato opte por essa opção, então o seu currículo será listado na tela de Listagem de Currículos (Figura 12) acessada pelos avaliadores.

Na tela de Listagem de Currículos, é possível visualizar inicialmente informações de maior relevância para o avaliador, como o nome do candidato, data de envio para a avaliação do currículo e o seu *status*. Em cada tupla de informações desta tela, é possível visualizar o currículo e fazer a sua aprovação (Figura 13).

Para finalizar a avaliação do currículo o avaliador deve inserir um *feedback*, com o objetivo de contribuir para a melhora do documento (Figura 14). Em casos de reprovação, o envio do *feedback* é obrigatório.

Figura 12 — Tela de listagem de currículos para avaliação



**Click Vaga** Currículos para avaliação Sobre nós Meu perfil

**Filtros**

Nome do candidato:

Data de envio:

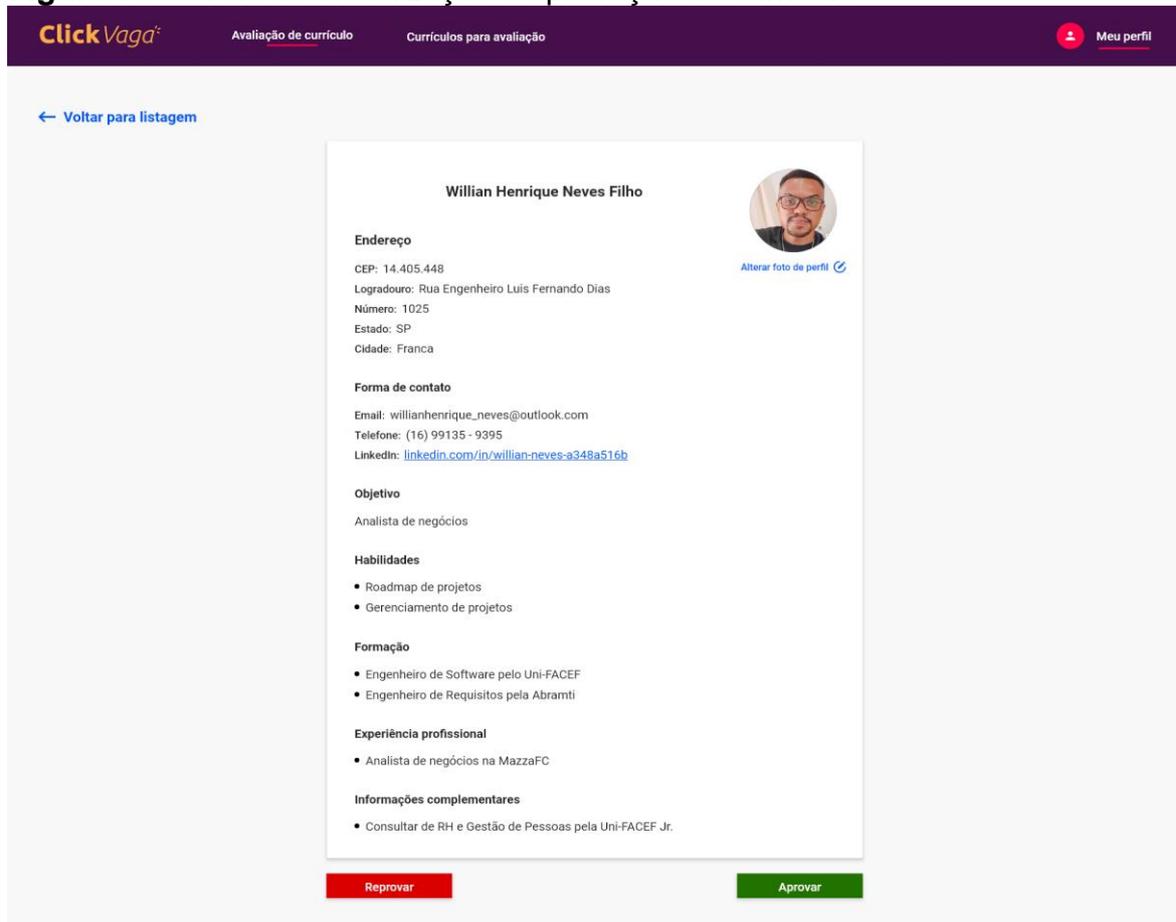
Status do currículo:

10 resultados encontrados

Nome do candidato	Data de envio	Status do currículo	
Willian Henrique Neves Filho	18/10/2021	Pendente avaliação	<input type="button" value="👁️"/> <input type="button" value="Avaliar currículo"/>
Cleber Gomes	18/10/2021	Pendente avaliação	<input type="button" value="👁️"/> <input type="button" value="Avaliar currículo"/>
Julia Paixão	17/10/2021	Pendente avaliação	<input type="button" value="👁️"/> <input type="button" value="Avaliar currículo"/>
Camila Peixoto da Silva	16/10/2021	Pendente avaliação	<input type="button" value="👁️"/> <input type="button" value="Avaliar currículo"/>

Fonte: os autores

Figura 13 — Tela de visualização e aprovação de currículo



**Click Vaga** Avaliação de currículo Currículos para avaliação Meu perfil

[← Voltar para listagem](#)

**Willian Henrique Neves Filho** 

**Endereço**  
CEP: 14.405.448  
Logradouro: Rua Engenheiro Luis Fernando Dias  
Número: 1025  
Estado: SP  
Cidade: Franca

**Forma de contato**  
Email: willianhenrique\_neves@outlook.com  
Telefone: (16) 99135 - 9395  
Linkedin: [linkedin.com/in/willian-neves-a348a516b](https://www.linkedin.com/in/willian-neves-a348a516b)

**Objetivo**  
Analista de negócios

**Habilidades**

- Roadmap de projetos
- Gerenciamento de projetos

**Formação**

- Engenheiro de Software pelo Uni-FACEF
- Engenheiro de Requisitos pela Abramti

**Experiência profissional**

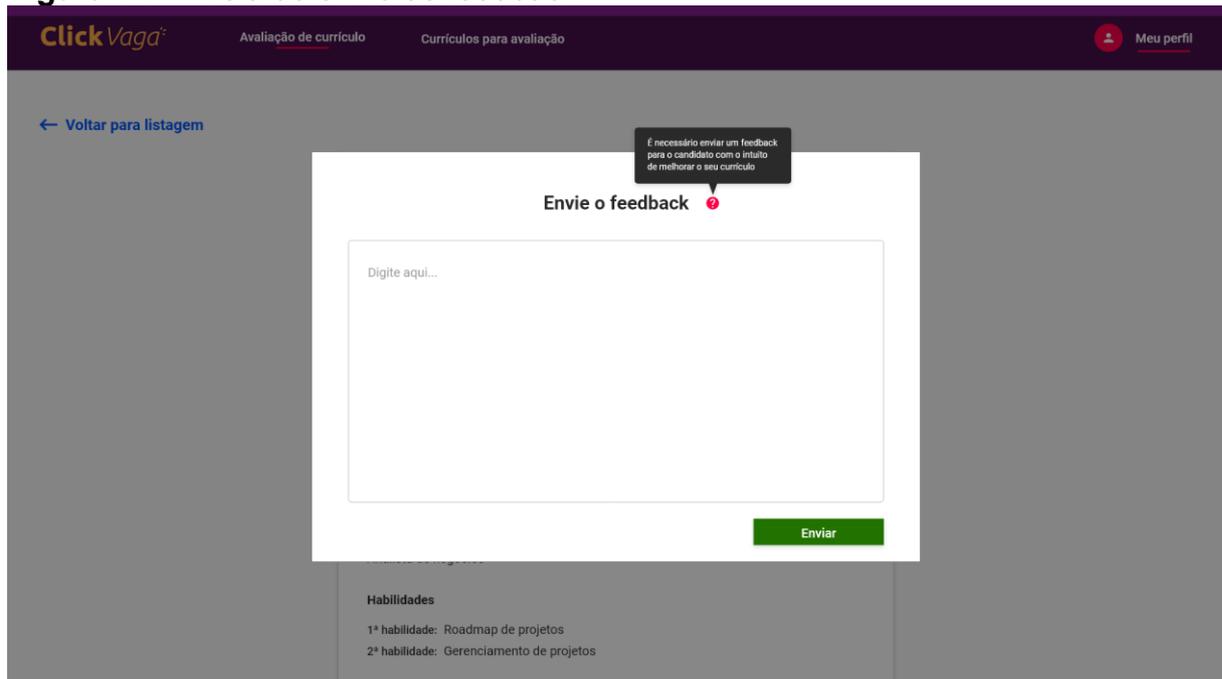
- Analista de negócios na MazzaFC

**Informações complementares**

- Consultar de RH e Gestão de Pessoas pela Uni-FACEF Jr.

Fonte: os autores

Figura 14 — Tela de envio de feedback



Fonte: os autores

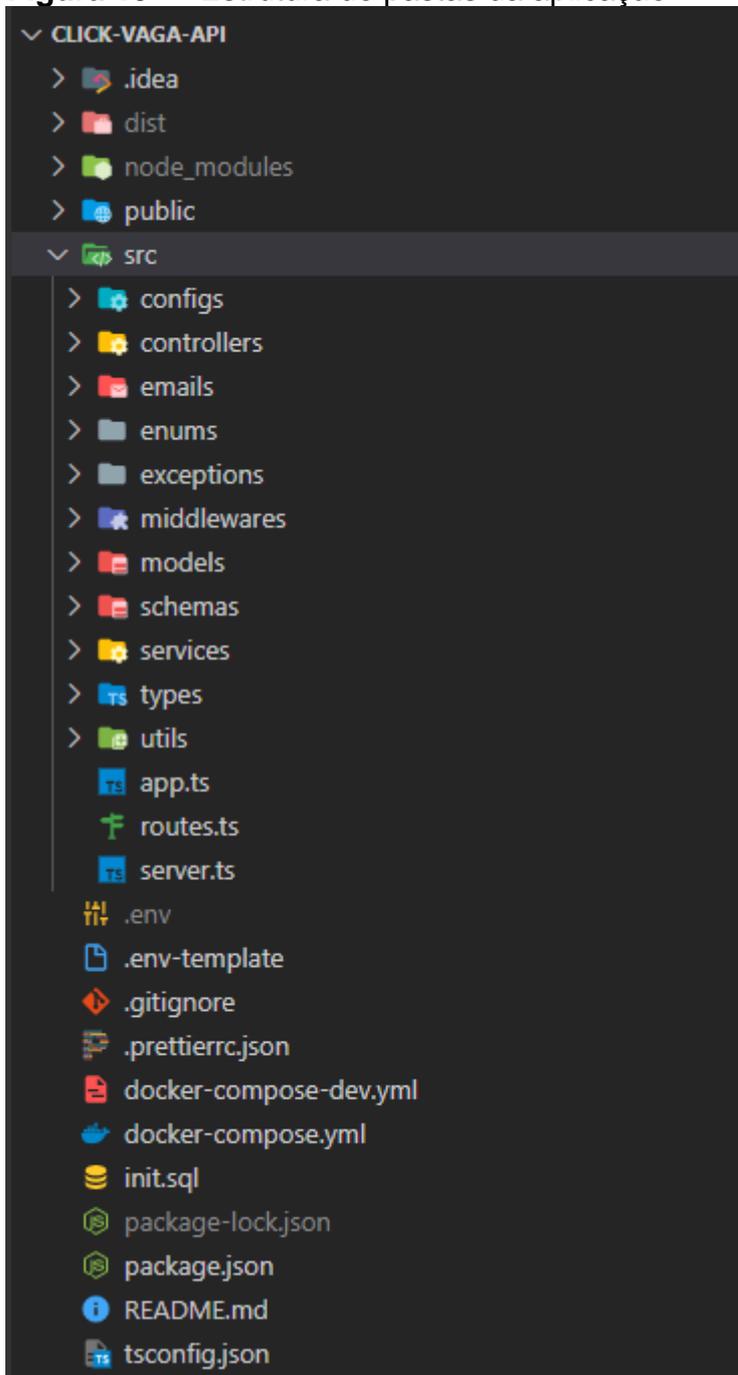
## 8. Implementação do Software

Pensando na arquitetura da aplicação, foi adotado o uso de modelos de *Application Programming Interface* (API), para o desacoplamento das aplicações *backend* e *frontend*. Ao utilizar este modelo, é possível realizar um desenvolvimento em paralelo, o que reduz grande parte do tempo de escrita de código. Por questões relacionadas a tempo e visando uma melhor qualidade de código, o MVP possui apenas a implementação do lado *backend*, conforme apresentado a seguir.

Na Figura 15 é apresentada a estrutura de pastas da aplicação que é descrita a seguir.

1. dist: Arquivos JavaScript gerados após o processo de transpilação.
2. public: Arquivos digitais, como imagens, vídeos, músicas, etc;
3. src: Todos os arquivos TypeScript fonte (*source*) da aplicação:
  - a. configs: Configuração da aplicação, como por exemplo definições do banco de dados;
  - b. controllers: Controladores responsáveis por receber requisições HTTP e repassar para o serviço final;
  - c. emails: Templates HTML para o envio de e-mails;
  - d. exceptions: Exceções personalizadas, como “*NotFound*”, “*BadRequest*” e outros;
  - e. middlewares: Middlewares HTTP, como por exemplo autenticadores;
  - f. models: Modelos ou entidades;
  - g. schemas: Validadores que verificam se todos os campos de uma requisição estão válidos;
  - h. services: Serviços que são chamados pelos controladores, responsáveis por comunicar com o banco de dados, processar dados e retornar ao controlador;

- i. types: Tipagens utilizadas nos serviços;
- j. utils: Funções utilitárias;
- k. app e server: Arquivos inicializadores da aplicação;

**Figura 15** — Estrutura de pastas da aplicação

**Fonte:** os autores

Para realizar a comunicação com o banco de dados, foi utilizado a tecnologia *Object-Relational Mapping* (ORM). Esta tecnologia facilita o desenvolvimento do aplicativo, uma vez que não é necessária a escrita de código SQL, podendo assim escrever *querys* em linguagem TypeScript. Na Figura 16 é apresentado o método responsável por cadastrar novos candidatos no banco de dados. Pode-se notar que em nenhum momento foi necessário utilizar código SQL.

Visando a melhor qualidade de código, foram escritos testes unitários para os principais métodos da aplicação (Figura 17). Todos os testes são executados com as bibliotecas Jest e Supertest, que são bem conhecidas na comunidade JavaScript. Com isso, é possível ter maior segurança que o sistema continua com todas suas características após uma alteração, uma vez que caso alterado incorretamente, os testes unitários não vão ser aprovados e indicarão quais pontos do código estão incorretos, conforme a Figura 18.

Todo o código fonte desenvolvido para o protótipo está disponível no *link* do repositório do GITHUB em Garcia (2021).

**Figura 16** — Função para criação de novos candidatos

```
78  async function __createCandidate(entity: CandidateCreateRequest) {
79    if (await __checkExistsCandidateByEmail(entity.email)) {
80      throw new ConflictError('This email is already in use')
81    }
82
83    const candidate = new Candidate()
84    candidate.email = entity.email
85    candidate.password = await encrypt(entity.password)
86    candidate.name = entity.name
87    candidate.gender = entity.gender
88    candidate.phone = entity.phone
89
90    await getRepository(Candidate).save(candidate)
91  }
```

Fonte: os autores

**Figura 17** — Exemplo de testes unitários

```
describe('Should encrypt', () => {
  it('Should hash', async () => {
    const hash = await encrypt(STRING)
    expect(hash).not.toBe(STRING)
  })
})

describe('Should compare', () => {
  it('Should return true', async () => {
    const equals = await compare(STRING, HASH)
    expect(equals).toBe(true)
  })

  it('Should return false', async () => {
    const equals = await compare(STRING + 'ABC', HASH)
    expect(equals).toBe(false)
  })
})
```

Fonte: os autores

Figura 18 — Exemplo de teste unitário com falha

```
FAIL tests/utils/bcrypt.test.ts
  Should encrypt
    [x] Should hash (61 ms)
  Should compare
    [x] Should return true (58 ms)
    [x] Shold return false (53 ms)

  • Should compare › Shold return false

  expect(received).toBe(expected) // Object.is equality

  Expected: true
  Received: false

   19 |         it('Shold return false', async () => {
   20 |             const equals = await compare(STRING + 'ABC', HASH)
>  21 |             expect(equals).toBe(true)
      |                             ^
   22 |         })
   23 |     })
   24 |

at tests/utils/bcrypt.test.ts:21:24
at step (tests/utils/bcrypt.test.ts:33:23)
at Object.next (tests/utils/bcrypt.test.ts:14:53)
at fulfilled (tests/utils/bcrypt.test.ts:5:58)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 2 passed, 3 total
Snapshots:  0 total
Time:        4.203 s, estimated 5 s
Ran all test suites.
# _
```

Fonte: os autores

### Considerações finais

Este trabalho teve como objetivo apresentar uma solução para o problema de facilitar o relacionamento de Candidatos a vagas de estágios e empregos com Empresas que buscam profissionais qualificados para suas vagas, pelo desenvolvimento de um MVP para mostrar o funcionamento de algumas funcionalidades da solução. Com a implementação, o objetivo principal do projeto foi alcançado.

Durante o desenvolvimento deste trabalho, foi possível aplicar grande parte do conhecimento adquirido durante o curso. Práticas e conceitos de projeto como diagramas, modelagem de dados, bancos de dados, linguagens de programação, metodologia de pesquisa e vários outros, foram essenciais para este trabalho.

Com o desenvolvimento do MVP, foi possível colocar em prática os principais pontos da solução, como o cadastro de usuários, acesso à plataforma, manipulação de currículos e avaliações realizadas por profissionais.

No futuro, este trabalho deverá ser melhorado, visando receber a implementação de novas funcionalidades no *backend* e a implementação por completo do *frontend*. Outra tecnologia aplicada ao projeto foi a utilização de API que facilitou o desacoplamento de código, permitindo e facilitando melhorias futuras, inclusive a criação de um aplicativo *mobile*.

## Referências

ADRIANO, Thiago da Silva Adriano. Guia prático de Typescript: Melhore suas aplicações JavaScript. São Paulo: Casa do Código, edição. 2021. Disponível em: <<https://bit.ly/3cZPJjn>>. Acesso em: 28 nov. 2021.

CARVALHO, Vinícius Carvalho. MySQL: Comece com o principal banco de dados open source do mercado. São Paulo: Casa do Código, edição. 2015. Acesso em: 28 nov. 2021.

FREITAS, Romualdo. Análise e Projeto de Software. UFTM, 2015. Disponível em: <<https://bit.ly/3nbYmh1>>. Acesso em: 09 ago. 2021.

FLANAGAN, David. JavaScript. O Guia Definitivo. Porto Alegre: Bookman Editora, Edição. 2013. Disponível em: <<https://bit.ly/3vYzAD3>>. Acesso em: 20 mar. 2021.

GARCIA, Gabriel Henrique Leal. Repositório contendo artefatos gerados durante o trabalho. Disponível em: <<https://github.com/gabrielgarciasp/click-vaga-artefatos>>. Acesso em: 06 out. 2021.

JOEL. Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER), 2014. Disponível em: <<https://bit.ly/3jTHRE4>>. Acesso em: 09 ago. 2021.

MAXIM, Bruce R.; PRESSMAN, Roger S. Engenharia de Software, uma abordagem profissional. 8. ed. São Paulo: AMGH EDITORA LTDA. 2016. Acesso em: 25 mar. 2021.

MOZILLA, Mozilla: Sobre JavaScript. Disponível em: <[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript)>. Acesso em 28 nov. 2021.

GLOBOPLAY, Globoplay: Sobre o desemprego no Brasil. Disponível em: <<https://globoplay.globo.com/v/8604069/>>. Acesso em 28 nov. 2021.

SILVEIRA, Daniel: Desemprego diante a pandemia no Brasil. Disponível em: <<https://g1.globo.com/economia/noticia/2020/09/23/diz-ibge-276percent.ghtml>>. Acesso em 28 nov. 2021.

LAZARETTI, Bruno: Aumento de vagas de emprego no mercado de TI. Disponível em: <<https://economia.uol.com.br/noticias/redacao/2020/09/30/na-contramao-do-mercado-empresas-de-ti-lutam-para-preencher-vagas-abertas.htm>>. Acesso em 28 nov. 2021.

MYSQL, MySQL: Documentação MySQL. Disponível em: <<https://dev.mysql.com/doc>>. Acesso em 14 abr. 2021.

NODE FOUNDATION: The V8 JavaScript Engine. Disponível em <<https://nodejs.dev>>. Acesso em 28 de nov. de 2021.

OSTERWALDER, Alexander; PIGNEUR, Yves. Business Model Generation - Inovação em Modelos de Negócios: Um Manual para Visionários, Inovadores e Revolucionários. 1. ed. Rio de Janeiro: Alta Books, agosto 2011.

PEREIRA, Caio. Aplicações web real-time com Node.js. São Paulo: Casa do Código, edição. 2014. Disponível em: <<https://bit.ly/3tRh9OK>>. Acesso em 20 mar. 2021.

SOMMERVILLE, Ian. Engenharia de Software. Tradução de Ivan Bosnic, Kalinka Oliveira. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TYPESCRIPT. TypeScript: Typed JavaScript at Any Scale. Disponível em: <<https://www.typescriptlang.org>>. Acesso em 25 mar. 2021.

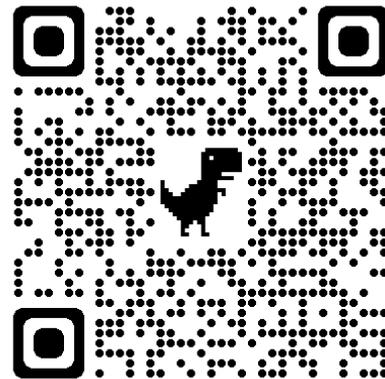
Desemprego. IBGE, 2021. Disponível em: <<https://www.ibge.gov.br/explica/desemprego.php>>. Acesso em: 08/11/2021

## Acessos rápidos a referências principais

Desemprego no Brasil em 2021



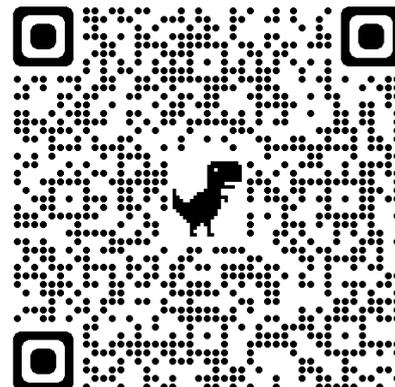
Personas



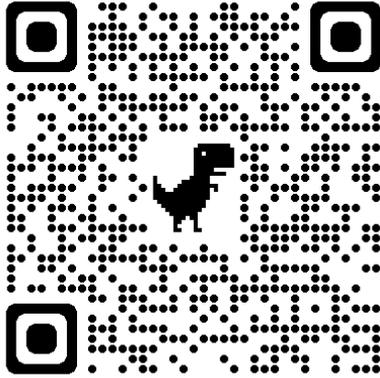
Canvas



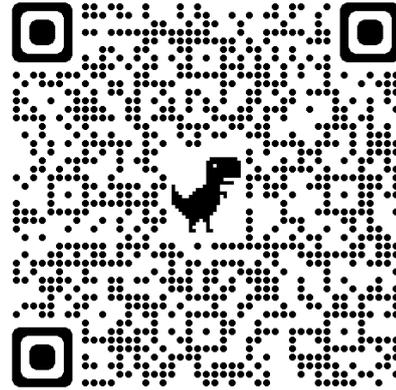
Matriz SWOT



BPMN



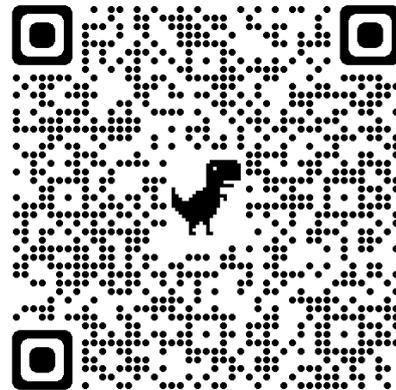
Exemplo de cenários de teste



EAP



Mitigação de Riscos



Qualidade de itens de acordo com as ISOs

