

**Guilherme Carvalho Siqueira Santos**

**Kauã Henrique da Silva Araújo**

**Leandro Bergamini dos Santos Junior**

**Victor Hugo Francisco Nascimento**

**TECH VAGAS**

**Sistema de vagas de estágio**

**Projeto Integrador**

**Guilherme Carvalho Siqueira Santos**  
**Kauã Henrique da Silva Araújo**  
**Leandro Bergamini dos Santos Junior**  
**Victor Hugo Francisco Nascimento**

**TECH VAGAS**  
**Sistema de vagas de estágio**

Projeto Integrador apresentado à Faculdade de Tecnologia Professor José Camargo – Fatec Jales, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistema.

Orientador: Prof. Jefferson Antônio Ribeiro Passarini.

**Guilherme Carvalho Siqueira Santos**  
**Kauã Henrique da Silva Araújo**  
**Leandro Bergamini dos Santos Junior**  
**Victor Hugo Francisco Nascimento**

**TECH VAGAS**  
**Sistema de vagas de estágio**

Projeto Integrador apresentado à Faculdade de Tecnologia Professor José Camargo – Fatec Jales, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistema.

Orientador: Prof. Jefferson Antônio Ribeiro Passarini.

**Banca Examinadora:**

\_\_\_\_\_  
Prof. (Orientador)  
Instituição

\_\_\_\_\_  
Prof.  
Instituição

\_\_\_\_\_  
Prof. (Orientador)  
Instituição

Jales, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_.

## LISTA DE FIGURAS

Figura 1 — Tela do Sistema “Vagas – Sistema de divulgação de vagas” .....	11
Figura 2 — Tela do Sistema “Catho” .....	12
Figura 3 — Tela do Sistema “Indeed” .....	13
Figura 4 — Diagrama de Classe.....	16
Figura 5 — Atores do Sistema .....	20
Figura 6 — Diagrama de caso de uso – Visão Aluno .....	39
Figura 7 — Diagrama de caso de uso – Visão da Empresa.....	40
Figura 8 — Diagrama de caso de uso – Consultar Aluno .....	41
Figura 9 — Diagrama de caso de uso – Alterar Aluno.....	42
Figura 10 — Diagrama de caso de uso – Excluir Aluno .....	44
Figura 11 — Diagrama de Sequência – Cadastrar vaga.....	46
Figura 12 — Diagrama de Sequência – Candidatar a vaga.....	47
Figura 13 — Imagem do primeiro cenário .....	48
Figura 14 — Imagem do segundo cenário .....	49
Figura 15 — Imagem da primeira persona.....	50
Figura 16 — Imagem da Segunda persona .....	51
Figura 17 — Tela de cadastro do usuário.....	52
Figura 18 — Tela de login do usuário.....	53
Figura 19 — Tela de inicial do sistema.....	54
Figura 20 — Tela de empresas cadastradas.....	55
Figura 21 — Protótipo tela de Cadastro.....	57
Figura 22 — Protótipo tela de Login .....	58
Figura 23 — Tela de cadastro.....	59
Figura 24 — Tela de Login .....	60
Figura 25 — Tela Inicial .....	61
Figura 26 — Tela de Vagas Cadastradas.....	62
Figura 27 — Mapeamento do Objeto Relacional.....	65
Figura 28 — Diagrama de Implantação .....	78

## LISTA DE QUADROS

Quadro 1 — Descrição Requisitos Funcionais.....	13
Quadro 2 — Descrição Requisitos Não Funcionais .....	14
Quadro 3 — Descrição Classe Aluno .....	17
Quadro 4 — Descrição Classe Vaga.....	18
Quadro 5 — Descrição Classe Usuario .....	19
Quadro 6 — Descrição Classe Candidato .....	19
Quadro 7 — Descrição Classe Concedente .....	19
Quadro 8 — Lista de Caso de Uso .....	21
Quadro 9 — Documentação do caso de uso Consultar Aluno .....	41
Quadro 10 — Documentação de caso de uso – Alterar Aluno .....	43
Quadro 11 — Documentação de caso de uso – Excluir Aluno .....	44
Quadro 12 — Script SQL – Tabela Aluno .....	66
Quadro 13 — Script SQL – Tabela Cargo .....	66
Quadro 14 — Script SQL – Tabela Concedente.....	66
Quadro 15 — Script SQL – Tabela Coordenador de Estágio .....	67
Quadro 16 — Script SQL – Tabela Curso .....	67
Quadro 17 — Script SQL – Tabela Instituição de Ensino.....	67
Quadro 18 — Script SQL – Tabela Login.....	67
Quadro 19 — Script SQL – Tabela Tipo documento.....	68
Quadro 20 — Script SQL – Tabela Tipo estágio.....	68
Quadro 21 — Script SQL – Tabela Usuário.....	68
Quadro 22 — Script SQL – Tabela Supervisor Estágio.....	68
Quadro 23 — Script SQL – Tabela Vagas.....	69
Quadro 24 — Script SQL – Tabela Apointamento.....	69
Quadro 25 — Script SQL – Tabela matricula .....	69
Quadro 26 — Script SQL – Tabela Documento .....	70
Quadro 27 — Script SQL – Tabela Documento Necessário .....	70
Quadro 28 — Script SQL – Tabela Sessão .....	71
Quadro 29 — Script SQL – Tabela Contrato Estágio .....	71
Quadro 30 — Script SQL – Tabela Documento Versão .....	72

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>7</b>
<b>2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE .....</b>	<b>9</b>
2.1 DESCRIÇÃO DOS OBJETIVOS DO SISTEMA .....	9
2.2 DESCRIÇÃO DO SISTEMA ATUAL .....	10
2.3 ANÁLISE DE SISTEMAS EXISTENTES .....	11
2.4 DESCRIÇÃO DOS PRINCIPAIS PROBLEMAS.....	13
2.5 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS.....	13
2.6 DESCRIÇÃO DOS REQUISITOS NÃO FUNCIONAIS .....	14
<b>3 VISÃO DE CASO DE USO – UML .....</b>	<b>15</b>
3.1 DIAGRAMA DE CLASSES .....	15
3.2 DICIONÁRIO DE CLASSES.....	16
3.3 DEFINIÇÃO DOS ATORES.....	20
3.4 LISTA DE CASOS DE USO .....	21
3.5. DIAGRAMA DE CASOS DE USO .....	38
3.6. DIAGRAMA DE CASOS DE USO INDIVIDUAIS .....	40
3.6.1. DIAGRAMA DE CASO DE USO – CONSULTAR ALUNO.....	41
3.6.2. DIAGRAMA DE CASO DE USO – ALTERAR ALUNO .....	42
3.6.3. DIAGRAMA DE CASO DE USO – EXCLUIR ALUNO .....	43
3.7. DIAGRAMA DE SEQUÊNCIA.....	45
3.7.1. DIAGRAMA DE SEQUENCIA – CADASTRAR VAGA .....	45
3.7.2. DIAGRAMA DE SEQUENCIA – CANDIDATAR A VAGA .....	46
<b>4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX).....</b>	<b>47</b>
4.1 DESCRIÇÃO DE CENÁRIO .....	47
4.2 DESCRIÇÃO DE PERSONAS .....	49
4.3 ESBOÇOS DE TELA (WIREFRAMES) .....	51
4.4 PROTÓTIPOS DE TELA .....	55
4.4.1. DEMONSTRAÇÃO DOS PROTÓTIPOS DAS TELA BASEADASA NOS WIREFRAMES .....	56
4.4.2. DISCUSSÃO DAS TELAS DO TECHVAGAS .....	58
<b>5 BANCO DE DADOS .....</b>	<b>62</b>
5.1 MODELO ENTIDADE RELACIONAMENTO .....	63
5.2 SCRIPT DAS TABELAS .....	65
5.3 MAPEAMENTO OBJETO RELACIONAL – ORM.....	72
<b>6 ARQUITETURA DE SOFTWARE .....</b>	<b>73</b>
6.1 ARQUITETURA DE DESENVOLVIMENTO.....	74
6.1.1 BACK-END .....	75
6.1.2 FRONT-END - WEB .....	75
6.2 SEGURANÇA DA INFORMAÇÃO.....	76
6.3 IMPLANTAÇÃO.....	77
<b>7 CONCLUSÃO .....</b>	<b>79</b>
<b>8 REFERÊNCIAS.....</b>	<b>80</b>

## 1 INTRODUÇÃO

A história do estágio no Brasil remonta a décadas passadas. O primeiro registro legal sobre a prática do estágio no país ocorreu em 1942, com a promulgação do Decreto-Lei nº 4.073 (Brasil, 1942), que instituiu a Lei Orgânica do Ensino Normal. No entanto, foi somente em 1967 que o estágio escolar foi oficialmente estabelecido, por meio da Portaria 1.002 do Ministro do Trabalho (Stremel, 2021), que disciplinava a relação entre as empresas e os estudantes.

Historicamente, o conceito de Estágio Supervisionado foi introduzido no Brasil pela Lei Orgânica do Ensino Normal, promulgada em 1946 (Brasil, 1946). Essa lei definiu o estágio supervisionado como uma etapa fundamental da formação de professores e estabeleceu diretrizes para sua realização. A partir daí o estágio passou a ser reconhecido como uma prática essencial para a preparação profissional dos estudantes.

Ao longo dos anos, a legislação de estágio no Brasil passou por várias alterações. Diversas leis e decretos foram promulgados para regulamentar o estágio, visando garantir direitos e deveres tanto para as empresas como para os estudantes. Essas mudanças refletiram a evolução do mercado de trabalho e das necessidades educacionais.

A Lei nº 11.788, de 25 de setembro de 2008 (Brasil, 2008), conhecida como Lei do Estágio, é a principal legislação que trata do assunto. Essa lei define o estágio como um ato educativo escolar supervisionado, desenvolvido no ambiente de trabalho, que tem como objetivo complementar o processo de ensino e aprendizagem dos estudantes. Ela trouxe importantes mudanças na forma como o estágio é realizado no Brasil, regulamentando a contratação de estagiários e buscando soluções para os desafios enfrentados nessa área.

Com a Lei do Estágio estabelece direitos e responsabilidades tanto para os estagiários quanto para as empresas que os contratam. Além disso, define os papéis e responsabilidades dos agentes envolvidos no estágio, como as instituições de ensino, as empresas e os estagiários.

Atualmente, no Brasil é visto como uma porta de entrada para o mercado de trabalho, sendo bastante valorizado por empresas e instituições. Muitas vezes, empresas utilizam o estágio como uma forma de recrutamento, contratando estagiários que se destacam durante o período de estágio. Além disso, o estágio pode ser uma oportunidade para os estudantes demonstrarem suas habilidades e adquirirem experiência prática na área em que estão estudando.

Em relação às estatísticas, em 2021, as atividades jurídicas foram as que mais empregaram estagiários no Brasil, com um total de 56,7 mil vagas em todo o país. Além disso,

as escolas dos ensinos infantil e fundamental também apresentaram uma demanda significativa por estagiários. No entanto, não há dados estatísticos disponíveis sobre a participação de estudantes e empresas no programa de estágio em geral. A Associação Brasileira de Estágios (Abres) realiza anualmente uma pesquisa para obter informações sobre o número de estagiários no Brasil, utilizando fontes como o Inep, MEC e IBGE para compilar os dados estatísticos sobre a participação de estudantes e empresas no programa de estágio (Jornal do Comercio, 2024; Gazeta Brasil, 2024).

De acordo com a pesquisa divulgada pelo Centro de Integração Empresa-Escola (CIEE), o número de contratos de estágio cresceu 18% em 2022. Nos primeiros seis meses de 2023, o Instituto Euvaldo Lodi (IEL) inseriu mais de 43 mil estudantes em empresas de todo o país, representando um aumento de 30% em comparação com o mesmo período do ano anterior. As oportunidades de estágio são destinadas a estudantes de diversas áreas, como Administração, Biologia, Biomedicina, Comércio Exterior, Economia, Marketing, Jornalismo, Letras, entre outras (Jornal do Comercio, 2024; Gazeta Brasil, 2024).

O desenvolvimento deste projeto visa ampliar e dinamizar o acesso dos alunos da Fatec Jales às oportunidades de estágio e emprego, promovendo uma interação eficaz entre estudantes e empresas. A proposta é criar um sistema digital intuitivo e eficiente, que otimize o processo de cadastramento de vagas pelos alunos e facilite a publicação e gestão dessas oportunidades pelas empresas parceiras.

Para as empresas, o sistema oferecerá uma plataforma centralizada para a divulgação de vagas, gestão de candidatos e acompanhamento do processo seletivo de maneira prática e transparente. Essa integração busca não apenas facilitar a inserção dos alunos no mercado de trabalho, mas também fortalecer o relacionamento entre a Fatec Jales e o setor empresarial, promovendo uma colaboração mútua e vantajosa.



## 2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE

O levantamento de requisitos é uma etapa essencial no ciclo de desenvolvimento de software, sendo fundamental para compreender as necessidades dos usuários e definir de forma precisa o escopo do projeto. De acordo com Roger S. Pressman:

A engenharia de requisitos fornece o mecanismo apropriado para entender o que o cliente deseja, analisando as necessidades, avaliando a exequibilidade, negociando uma condição razoável, especificando e gerindo os requisitos à medida que eles são transformados em um sistema operacional (PRESSMAN, 2015, p.118).

De acordo com Sommerville (2018), o levantamento de requisitos é uma etapa fundamental no ciclo de vida do desenvolvimento de software, pois estabelece a base necessária para criar um sistema que atenda às necessidades dos usuários e agregue valor ao negócio. Esse processo desempenha um papel crucial ao garantir uma compreensão clara e detalhada das expectativas e requisitos do cliente, fornecendo diretrizes essenciais para o desenvolvimento de um software alinhado aos objetivos definidos.

### 2.1 DESCRIÇÃO DOS OBJETIVOS DO SISTEMA

O projeto está estruturado em dois módulos principais, cada um desempenhando funções específicas para atender às demandas relacionadas aos estágios. O primeiro módulo é dedicado à divulgação de oportunidades, permitindo que as empresas publiquem vagas de estágio de maneira prática e acessível. Ele facilita o acesso dos estudantes a informações detalhadas, como requisitos, descrições de atividades e benefícios oferecidos, promovendo uma comunicação clara e eficiente entre as partes. O segundo módulo foca na gestão interna, abrangendo o controle de contratos e o gerenciamento do processo de contratação dos estagiários, garantindo que todas as etapas administrativas sejam realizadas de forma organizada e eficaz.

Essa divisão modular assegura que o sistema seja abrangente e funcional, atendendo tanto à necessidade de conectar estudantes e empresas quanto ao acompanhamento das etapas formais das contratações. Com isso, o projeto não apenas otimiza a divulgação de vagas, mas também oferece suporte completo para a administração dos estágios, contribuindo para uma experiência mais integrada e eficiente para todos os envolvidos.

Este projeto tem como objetivo principal promover um acesso mais amplo e dinâmico aos alunos da Fatec Jales, facilitando a interação entre estudantes e empresas no contexto de

estágios e oportunidades de emprego. Pretende-se desenvolver um sistema intuitivo e eficiente, que simplifique o processo de cadastro de vagas pelos alunos e a disponibilização dessas oportunidades pelas empresas parceiras.

O sistema será projetado para ser limpo, amigável e de fácil navegação, garantindo que tanto os alunos quanto as empresas possam utilizá-lo sem dificuldades. Para os alunos, o sistema oferecerá um portal onde poderão se cadastrar para as vagas de estágio e emprego de maneira rápida e organizada, além de permitir a consulta de informações detalhadas sobre cada oportunidade, incluindo requisitos, descrições de atividades e benefícios oferecidos.

Para as empresas, o sistema proporcionará uma plataforma onde poderão publicar vagas, gerenciar candidatos e acompanhar todo o processo de seleção de maneira prática e transparente. Com isso, espera-se não apenas facilitar a inserção dos alunos no mercado de trabalho, mas também fortalecer a conexão entre a Fatec Jales e o setor empresarial, promovendo uma colaboração mais estreita e produtiva.

Além disso, o projeto visa assegurar que todos os docentes da Faculdade tenham acesso às informações sobre as oportunidades de estágio e emprego, permitindo que possam orientar e apoiar seus alunos de forma mais eficaz. Dessa forma, este sistema contribuirá para uma formação mais completa e alinhada às demandas do mercado de trabalho, reforçando o compromisso da Fatec Jales com a excelência educacional e profissional de seus estudantes.

## 2.2 DESCRIÇÃO DO SISTEMA ATUAL

Atualmente, a Fatec Jales não dispõe de um sistema próprio para a divulgação de vagas de estágio. As oportunidades oferecidas pelas empresas parceiras são compartilhadas por meio de grupos de comunicação dos cursos da instituição, o que limita a centralização e a eficiência no acesso a essas informações.

O sistema de estágio no Brasil é regulamentado pela Lei nº 11.788/2008, popularmente conhecida como Lei do Estágio. Essa legislação define o estágio como um ato educativo escolar supervisionado, desenvolvido no ambiente de trabalho com o objetivo de proporcionar ao estudante uma experiência prática diretamente relacionada à sua área de estudo. A Lei do Estágio abrange estágios de ensino médio, técnico e superior, incluindo estudantes brasileiros e estrangeiros.

Em vigor desde setembro de 2008, a Lei do Estágio trouxe importantes regulamentações para garantir a qualidade e a segurança dos estágios. Entre as principais disposições, destaca-se a carga horária máxima de 6 horas diárias e 30 horas semanais para os estagiários. Além disso, a lei estabelece a necessidade de um termo de compromisso, que deve

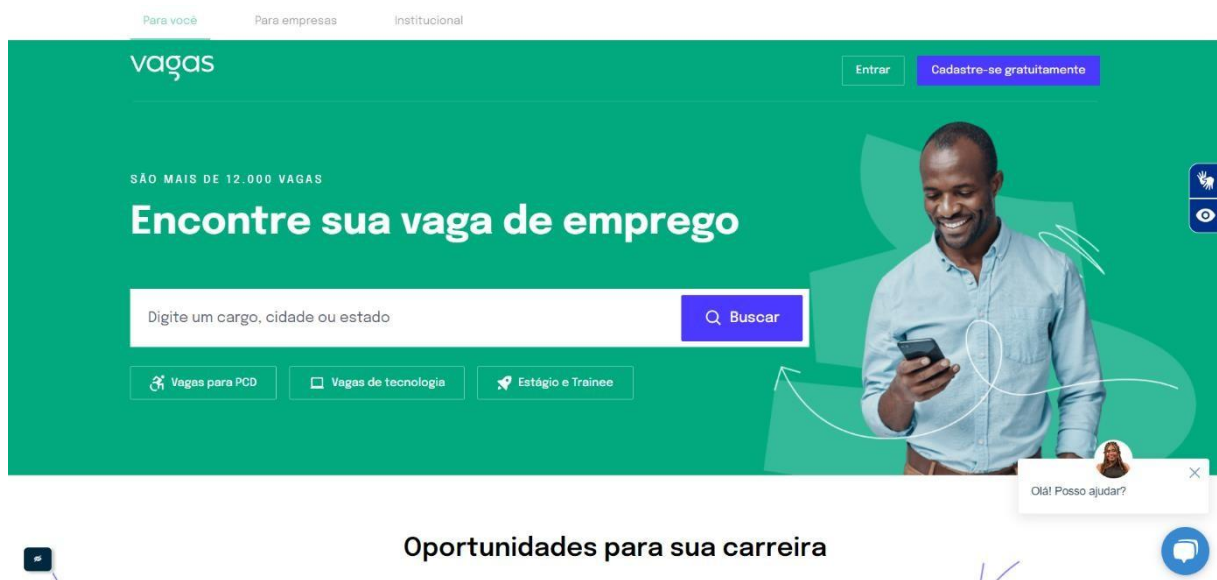
ser firmado entre o estagiário, a instituição de ensino e a empresa, visando assegurar a conformidade e a supervisão adequada das atividades realizadas.

### 2.3 ANÁLISE DE SISTEMAS EXISTENTES

No mundo dinâmico da tecnologia e dos negócios, a análise de sistemas existentes desempenha um papel crucial na identificação e solução de problemas, bem como na melhoria contínua das operações. Este processo envolve o exame detalhado dos sistemas que estão em operação, com o objetivo de entender seu funcionamento, identificar suas forças e fraquezas e descobrir oportunidades para aprimoramento.

O sistema da (Figura 1), é um sistema que tem como objetivo principal conectar candidatos a oportunidades de emprego em diversas áreas e níveis de experiência. Ele oferece uma plataforma gratuita onde empresas podem divulgar vagas e conduzir processos seletivos, enquanto os candidatos podem cadastrar seus currículos, buscar vagas, e se preparar para entrevistas através de conteúdos especializados. O site facilita a interação entre candidatos e empresas, ajudando na recolocação profissional e no desenvolvimento de carreiras.

**Figura 1** — Tela do Sistema “Vagas – Sistema de divulgação de vagas”

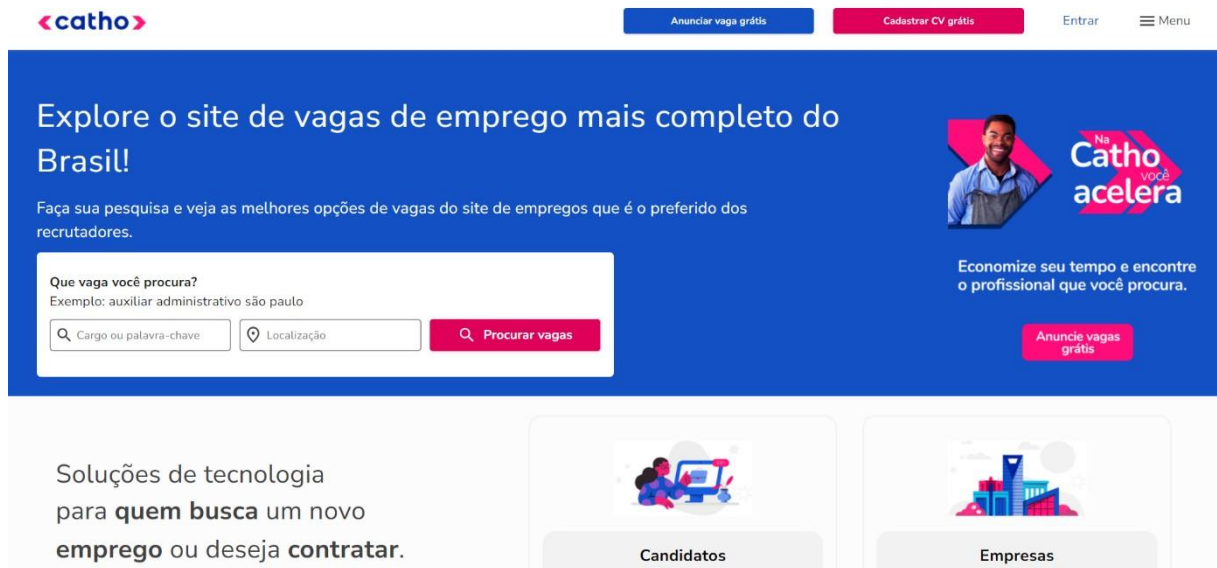


Fonte: Elaborado pelos autores.

O Catho (Figura 2), é um sistema que tem como objetivo de conectar candidatos a oportunidades de emprego em todo o Brasil. Ele oferece uma plataforma onde os usuários podem criar e cadastrar currículos gratuitamente, buscar vagas por área ou cidade, e candidatar-

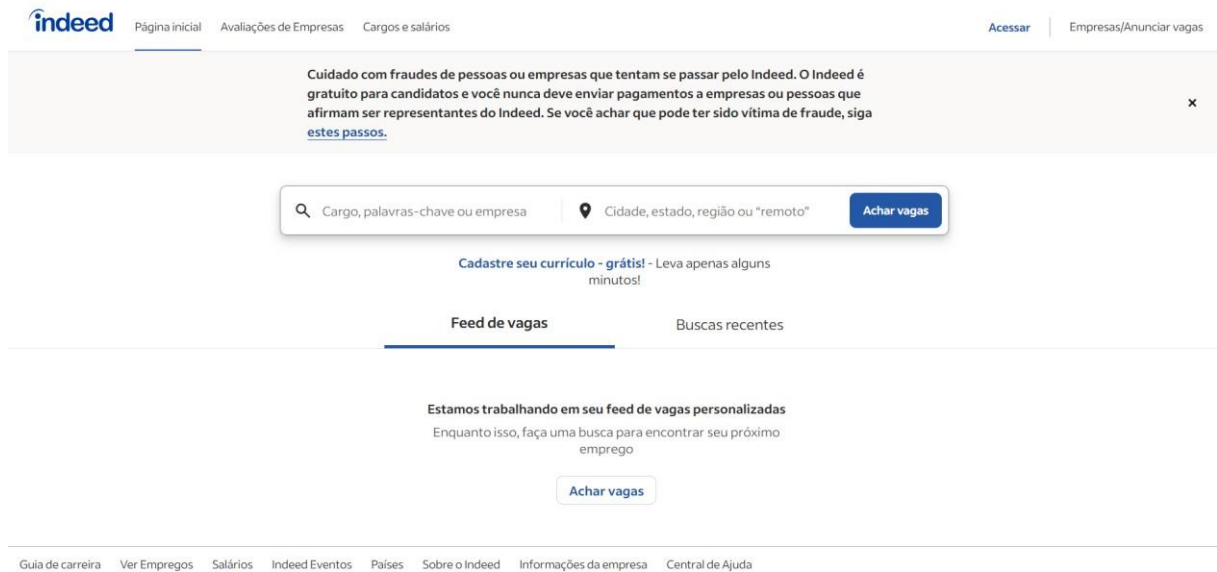
se a posições disponíveis. Para as empresas, o site permite anunciar vagas e buscar currículos de candidatos qualificados. Além disso, a Catho oferece ferramentas e conteúdo para ajudar candidatos na preparação para o mercado de trabalho, como guias de profissões, calculadoras trabalhistas e modelos de currículo.

**Figura 2** — Tela do Sistema “Catho”



Fonte: Elaborado pelos autores.

O Indeed (Figura 3), é sistema que tem como objetivo principal ajudar pessoas a encontrar empregos. Ele permite que os candidatos pesquisem vagas de emprego publicadas em milhares de sites e empresas diretamente em um só lugar. Além disso, os usuários podem cadastrar seus currículos, configurar alertas de emprego, e acessar informações sobre empresas e salários para facilitar a busca de emprego. O Indeed também fornece ferramentas e recursos para ajudar candidatos a se prepararem para o mercado de trabalho, como conselhos de carreira e dicas para entrevistas.

**Figura 3** — Tela do Sistema “Indeed”

Fonte: Elaborado pelos autores.

## 2.4 DESCRIÇÃO DOS PRINCIPAIS PROBLEMAS

Um desafio potencial é o manuseio inicial do software, considerando que a Fatec Jales nunca utilizou um sistema de inscrição para estágios. Esse fator pode tornar o processo um pouco complicado e confuso para os usuários de primeira viagem. Para mitigar esse problema, é essencial implementar um programa de treinamento adequado e fornecer suporte contínuo, garantindo que tanto alunos quanto empresas parceiras se familiarizem rapidamente com a nova plataforma.

## 2.5 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS

Os requisitos funcionais são uma etapa essencial no desenvolvimento de uma aplicação, pois determinam as necessidades dos clientes e especificam as funcionalidades que o software deve oferecer. Eles servem como um guia fundamental para atender às demandas dos usuários e alcançar os objetivos do projeto, orientando tanto a elaboração quanto os testes. Dessa forma, garantem que as equipes compreendam claramente o que deve ser implementado e validado (SOMMERVILLE, 2007).

**Quadro 1** — Descrição Requisitos Funcionais

Requisitos Funcionais	Descrição
-----------------------	-----------

1	Cadastro de usuários	O sistema deve ser capaz de cadastrar usuários para a sua utilização.
2	Cadastro de vagas	O sistema deve ser capaz de cadastrar a vaga que o usuário escolheu.
3	Alteração de dados do usuário	O sistema deve ser capaz de realizar a alteração de dados do usuário.
4	Cadastro de currículo	O sistema deve ser capaz de cadastrar o currículo do usuário caso o mesmo opte por fazer no sistema.
6	Carregar currículo em PDF	O sistema deve ser capaz de conseguir realizar uploads externos caso o usuário deseje enviar seu currículo.
7	Cadastro de vaga	O sistema deve ser capaz de realizar o cadastro de vagas das empresas cadastradas no sistema.
8	Alteração de vaga	O sistema deve ser capaz de realizar alterações na vaga publicada.
9	Visualizar candidatos para vaga	O sistema deve ser capaz de mostrar a visualização dos candidatos pretendentes a vaga.
10	Cancelar vaga	O sistema deve ser capaz de fazer o cancelamento da vaga.

Fonte: Elaborado pelos autores.

## 2.6 DESCRIÇÃO DOS REQUISITOS NÃO FUNCIONAIS

Segundo Paula Filho (2020), os requisitos não funcionais abrangem os requisitos de desempenho e outros atributos de qualidade do produto, incluindo requisitos lógicos relacionados aos dados persistentes e requisitos de natureza técnica, como restrições ao design e implementação. A recomendação é expressar os requisitos não funcionais de maneira precisa e, sempre que possível, quantitativa. Mesmo que seja desafiador formular valores específicos durante o levantamento de requisitos para a primeira versão de um produto, essa precisão é crucial.

### Quadro 2 — Descrição Requisitos Não Funcionais

	Requisitos não funcionais	Descrição
1	Atualização do banco de dados	Aprimorar a agilidade na atualização do banco de dados para que não contenha informações erradas ou desatualizadas.
2	Máquina capaz de utilizar o sistema	É de suma importância que o dispositivo tenha alguns componentes que façam ele conseguir suportar o sistema
3	Auxiliar na utilização do sistema	Oferecemos suporte aos usuários na hora de fazer a atualização explicando as mudanças e como utiliza-lo da melhor maneira.

Fonte: Elaborado pelos autores.

### 3 VISÃO DE CASO DE USO – UML

#### 3.1 DIAGRAMA DE CLASSES

O diagrama de classes destaca-se como um dos elementos mais cruciais da UML, desempenhando um papel fundamental ao oferecer suporte para a maioria dos demais diagramas. Conforme abordado na obra, esse diagrama assume a responsabilidade de definir a estrutura das classes utilizadas pelo sistema, delineando seus atributos, métodos e estabelecendo as inter-relações entre as classes. (GUEDES, 2011).

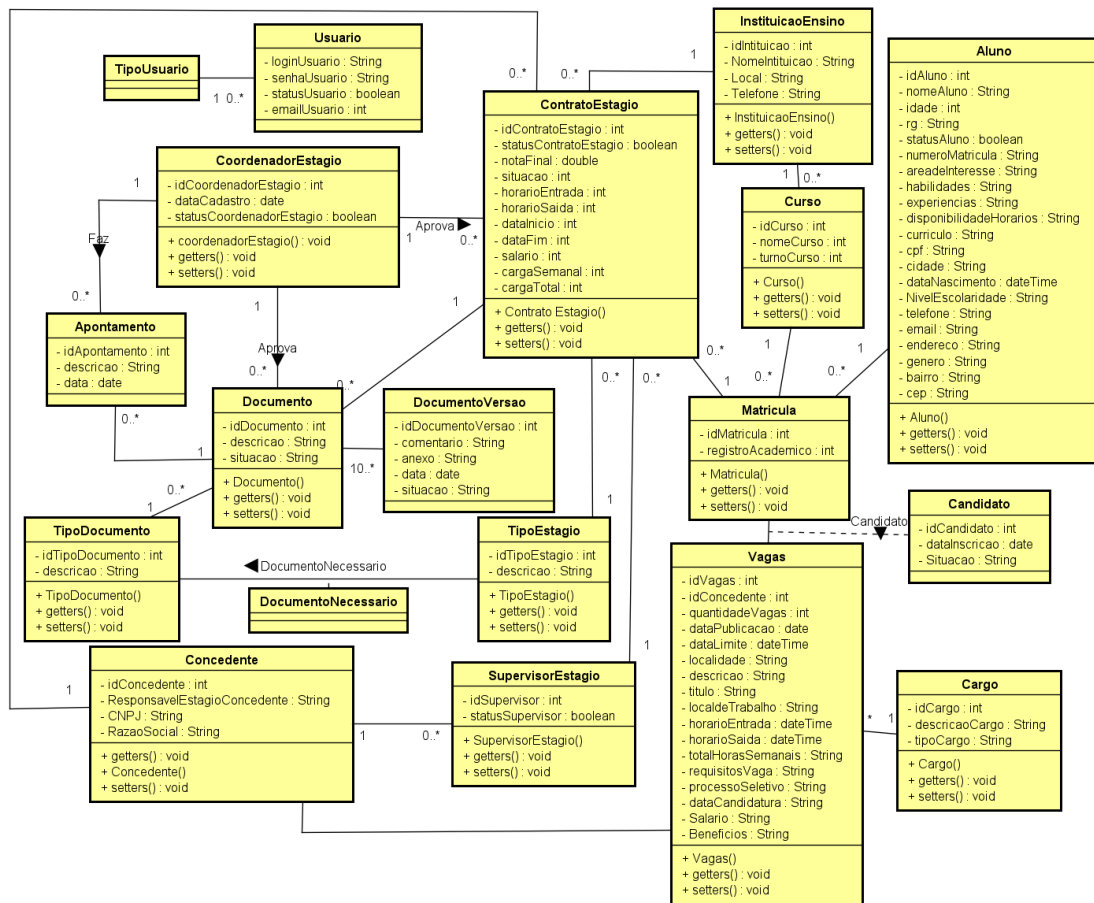
De acordo com o diagrama de classes, foram definidos os atores do sistema, esses atores representam os diferentes papéis e hierarquias que os usuários podem desempenhar. Para cada tipo de ator foram estabelecidos funções e níveis de acesso específicos, de acordo com as responsabilidades no sistema. Os atores — Concedente e Aluno — conforme ilustrado na figura 5, eles herdam esses atributos de suas próprias classes.

O controle de acesso é implementado por validando as credenciais e token do usuário, exigindo que os usuários façam login no sistema utilizando e-mail e senha. Com isso o administrador do sistema ou atores hierarquicamente maiores conseguem limitar as funções disponíveis com o papel do ator no sistema.

O ator Concedente é responsável por disponibilizar vagas de estágio aos alunos. Esse papel pode ser assumido por uma entidade externa, como uma empresa ou organização, um profissional autônomo ou, eventualmente, até pela própria instituição de ensino. Sua principal função é tornar acessíveis as oportunidades de estágio no sistema.

Já o ator Aluno representa o estudante que possui acesso ao portal. Ele pode buscar vagas de estágio, candidatar-se a essas oportunidades e anexar seu currículo diretamente na plataforma. Dessa forma, o sistema oferece aos alunos uma interface para explorar e se conectar com o mercado de trabalho.

Figura 4 — Diagrama de Classe



Fonte: Elaborado pelos autores.

### 3.2 DICIONÁRIO DE CLASSES

Os dicionários de classes são um modelo favorável no processo de análise de requisitos, pois oferecem uma visão abrangente do processamento do sistema de ponta a ponta. Eles contribuem para a identificação dos objetos e das operações realizadas sobre eles, fornecendo informações detalhadas sobre as entidades envolvidas na modelagem e projeção do sistema. Dessa forma, desempenham um papel crucial na compreensão e organização das estruturas e comportamentos necessários para o desenvolvimento da aplicação. (SOMMERVILLE, 2018)

O (Quadro 1) apresenta a estrutura da classe “Aluno”, utilizada para gerenciar as informações dos usuários do sistema. A classe é composta por atributos, cada um com seu tipo de dado e uma descrição detalhada de sua função no contexto do sistema.



**Quadro 3** — Descrição Classe Aluno

Atributo	Tipo	Descrição
idAluno	Integer	Utilizado para localizar o aluno dentro do sistema.
NomeAluno	String	Nomeia os usuarios do sistema.
Idade	Integer	Informa a idade dos usuários .
RG	String	Registro geral dos usuários.
StatusAluno	Boolean	Determina e informa qual o cargo do aluno a qual etapa esta da contratação.
NumeroMatricula	String	Ordena e numera as matrículas.
AreaInteresse	String	Determina qual área o estagiário tem interesse em atuar.
Habilidades	String	É informado quais são as habilidades que o usuários possui.
DisponibilidadesHorários	String	Quais os horários o usuário tem disponivel.
Currículo	String	É apresentado o currículo do usuário.
CPF	String	É apresentado o currículo do usuário .
DataNascimento	DateTime	É informado a data de nascimento do usuário.
NívelEscolaridade	String	Informa qual o nível de escolaridade do usuário.
Telefone	String	Mostra o telefone do usuário.
E-mail	String	E-mail do usuário.
Endereço	String	Informa o endereço do aluno.
Gênero	String	Especifica o gênero do usuário.
Bairro	String	Informa o bairro do usuário.
Cep	String	É informado o código de endereçamento postal.

Fonte: Elaborado pelos autores.

A classe “Vaga” no (Quadro 2) tem como objetivo gerenciar todas as vagas de estágio disponibilizadas pelas empresas no sistema, organizando informações detalhadas sobre cada oportunidade. Para isso, conta com diversos atributos que garantem a identificação e a descrição das vagas.

**Quadro 4** — Descrição Classe Vaga

Atributo	Tipo	Descrição
idVagas	Integer	É utilizado para localizar dentro dos sistemas cada uma das vagas
IdConcedente	Integer	É usado para informar qual o identificador da empresa que está contratando o estagiário
QuantidadeVagas	Integer	Informa exatamente o número de vagas existentes de determinada empresa
DataPublicacao	Date	Informa e oficializa a postagem da vaga
DataLimite	dateTime	É definida uma data limite para a inscrição do estagiário para determinada vaga
Localidade	String	Informa a localidade em que a empresa se encontra rua, bairro e cidade
Descricao	String	Na descrição será descrito todas as informações referentes a vaga de estágio
Titulo	String	Informa como um preview do que o estagiário irá exercer na vaga
LocaldeTrabalho	String	Local de trabalho se refere a empresa ou local físico que o estagiário irá trabalhar
HorarioEntrada	dateTime	Irá definir a entrada em que o estagiário deverá estar para cumprir sua carga horária
HorarioSaida	DateTime	Hora de saída de acordo com a carga horaria presente no contrato e as horas que precisam ser cumpridas na semana
totalHorasSemanais	String	Define a carga horaria total estipulada pelo contrato de acordo com as regras de estágio

Fonte: Elaborado pelos autores.

A classe “Usuário” (Quadro 3) tem como objetivo principal gerenciar o cadastro dos usuários, permitindo que eles acessem o sistema por meio de credenciais válidas.

**Quadro 5** — Descrição Classe Usuario

Atributo	Tipo	Descrição
loginUsuario	String	É o que autentica o usuário no sistema
senhaUsuario	String	É o utilizado juntamente com o login para a autenticação do usuário
statusUsuario	Boolean	Mostra qual o status do usuário dentro dos sistemas
emailUsuario	Integer	É um meio de comunicação com o usuário

Fonte: Elaborado pelos autores.

A classe “candidato” (Quadro 4) tem como objetivo gerenciar e registrar os alunos que se candidataram a vagas de estágio, facilitando o controle e acompanhamento dos processos seletivos.

**Quadro 6** — Descrição Classe Candidato

Atributo	Tipo	Descrição
idCandidato	Integer	Identifica o candidato
dataInscricao	Date	Refere-se ao dia em que o aluno se inscreveu para aquela vaga
Situacao	String	Mostra a situação do candidato perante a vaga

Fonte: Elaborado pelos autores.

A classe “Concedente” (Quadro 5) tem como objetivo gerenciar as empresas responsáveis por disponibilizar vagas de estágio. Ela permite que as empresas adicionem novas vagas ao sistema e realizem a administração completa das oportunidades que oferecem.

**Quadro 7** — Descrição Classe Concedente

Atributo	Tipo	Descrição
idConcedente	Integer	Identifica qual é a empresa que irá conceder o estágio

ResponsavelEstagioConcedente	String	Mostra quem é a pessoa de dentro da empresa que é responsável por conceder o estágio
------------------------------	--------	--

Fonte: Elaborado pelos autores.

### 3.3 DEFINIÇÃO DOS ATORES

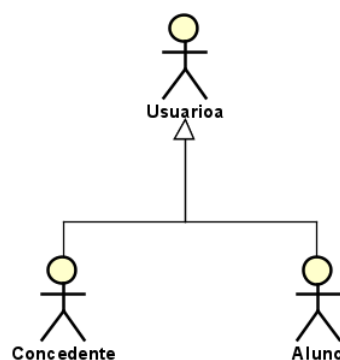
Os atores podem ser representados por usuários, dispositivos de hardware ou sistemas externos que interagem diretamente com as funcionalidades de um sistema. Na notação UML, sua representação gráfica utiliza ícones que simbolizam pessoas, acompanhados de breves descrições que explicam seus papéis. Esse formato visual simplifica a compreensão das interações e funções desempenhadas pelos diferentes elementos envolvidos no sistema (GUEDES, 2011).

O Aluno é o usuário responsável por utilizar a aplicação. Ele deve realizar seu cadastro no sistema e interagir com suas funcionalidades, podendo atuar como estagiário ou se candidatar a uma vaga de estágio.

O Candidato representa o usuário que evolui do status de aluno para um postulante efetivo a uma vaga de estágio. Esse papel é assumido quando o aluno demonstra interesse em uma oportunidade específica, tornando-se um candidato formal no sistema.

O Concedente é o responsável por oferecer as vagas de estágio. Esse usuário pode ser uma empresa, uma pessoa física ou até mesmo a própria instituição de ensino. O concedente é quem disponibiliza as oportunidades para os alunos dentro da aplicação.

**Figura 5** — Atores do Sistema



Fonte: Elaborado pelos autores.

### 3.4 LISTA DE CASOS DE USO

Conforme enfatizado na obra (GUEDES, 2011), o Diagrama de Caso de Uso é o mais abstrato, sendo, portanto, o mais flexível e informal entre as representações, utilizado para proporcionar uma visão global do sistema. Seu propósito central é a identificação dos atores que interagirão com o software, acompanhados de suas respectivas funcionalidades. Este diagrama desfruta de ampla aplicação nas fases iniciais de levantamento e análise de requisitos, sendo, no entanto, igualmente valioso ao longo de todo o processo de modelagem, podendo inclusive servir como alicerce para a elaboração de outros diagramas.

A lista de casos de uso é uma ferramenta fundamental no desenvolvimento de software, pois descreve de forma clara como os diferentes atores interagem com o sistema para atingir objetivos específicos. Cada caso de uso detalha um conjunto de ações realizadas entre o ator e o sistema, contribuindo para a identificação e definição dos requisitos funcionais necessários para o funcionamento do software.

**Quadro 8** — Lista de Caso de Uso

Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
1	Alterar Administrador	Informa dados administrador	Alterar administrador	Retorna mensagem “Alterado com sucesso”
2	Administrador altera matrícula	Informa dados da matrícula	Alterar matrícula	Retorna mensagem “Alterado com sucesso”
3	Administrador altera curso	Informa dados do curso	Alterar curso	Retorna mensagem “Alterado com sucesso”
4	Administrador altera candidato	Informa dados do candidato	Alterar candidato	Retorna mensagem “Alterado com sucesso”
5	Administrador altera instituição de ensino	Informa dados instituição de ensino	Alterar instituição de ensino	Retorna mensagem “Alterado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>6</b>	Administrador altera coordenador de estágio	Informa dados coordenador de estágio	Altera coordenador de estágio	Retorna mensagem “Alterado com sucesso”
<b>7</b>	Administrador altera contrato estágio	Informa dados contrato estágio	Altera contrato estágio	Retorna mensagem “Alterado com sucesso”
<b>8</b>	Administrador altera concedente	Informa dados concedente	Altera concedente	Retorna mensagem “Alterado com sucesso”
<b>9</b>	Administrador altera vagas	Informa dados vagas	Altera vagas	Retorna mensagem “Alterado com sucesso”
<b>10</b>	Administrador altera cargo	Informa dados cargo	Altera cargo	Retorna mensagem “Alterado com sucesso”
<b>11</b>	Administrador altera requisito vagas	Informa dados requisito vaga	Altera requisito vagas	Retorna mensagem “Alterado com sucesso”
<b>12</b>	Administrador altera supervisor estágio	Informa dados supervisor estágio	Altera supervisor estágio	Retorna mensagem “Alterado com sucesso”
<b>13</b>	Administrador altera processo seletivo	Informa dados processo seletivo	Altera processo seletivo	Retorna mensagem “Alterado com sucesso”
<b>14</b>	Administrador altera apontamento	Informa dados apontamento	Altera apontamento	Retorna mensagem” Alterado com sucesso”
<b>15</b>	Administrador altera documento	Informa dados documento	Altera documento	Retorna mensagem” Alterado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>16</b>	Administrador altera tipo documento	Informa dados tipo documento	Altera tipo documento	Retorna mensagem” Alterado com sucesso”
<b>17</b>	Administrador altera documento versão	Informa dados documento versão	Altera documento versão	Retorna mensagem” Alterado com sucesso”
<b>18</b>	Administrador altera documento necessário	Informa documento necessário	Altera documento necessário	Retorna mensagem” Alterado com sucesso”
<b>19</b>	Administrador altera tipo estágio	Informa tipo estágio	Altera tipo estágio	Retorna mensagem” Alterado com sucesso”
<b>20</b>	Cadastrar Administrador	Informa dados administrador	Cadastrar administrador	Retorna mensagem “Cadastrado com sucesso”
<b>21</b>	Administrador cadastra matrícula	Informa dados da matrícula	Cadastrar matrícula	Retorna mensagem “Cadastrado com sucesso”
<b>22</b>	Administrador cadastra curso	Informa dados do curso	Cadastrar curso	Retorna mensagem “Cadastrado com sucesso”
<b>23</b>	Administrador cadastra candidato	Informa dados do candidato	Cadastra candidato	Retorna mensagem “Cadastrado com sucesso”
<b>24</b>	Administrador cadastra instituição de ensino	Informa dados instituição de ensino	Cadastra instituição de ensino	Retorna mensagem “Cadastrado com sucesso”
<b>25</b>	Administrador cadastra coordenador de estágio	Informa dados coordenador de estágio	Cadastra coordenador de estágio	Retorna mensagem “Cadastrado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>26</b>	Administrador cadastra contrato estágio	Informa dados contrato estágio	Cadastra contrato estágio	Retorna mensagem “Cadastrado com sucesso”
<b>27</b>	Administrador cadastra concedente	Informa dados concedente	Cadastra concedente	Retorna mensagem “Cadastrado com sucesso”
<b>28</b>	Administrador cadastra vagas	Informa dados vagas	Cadastra vagas	Retorna mensagem “Cadastrado com sucesso”
<b>29</b>	Administrador cadastra cargo	Informa dados cargo	Cadastra cargo	Retorna mensagem “Cadastrado com sucesso”
<b>30</b>	Administrador cadastra requisito vagas	Informa dados requisito vaga	Cadastra requisito vagas	Retorna mensagem “Cadastrado com sucesso”
<b>31</b>	Administrador cadastra supervisor estágio	Informa dados supervisor estágio	Cadastra supervisor estágio	Retorna mensagem “Cadastrado com sucesso”
<b>32</b>	Administrador cadastra processo seletivo	Informa dados processo seletivo	Cadastra processo seletivo	Retorna mensagem “Cadastrado com sucesso”
<b>33</b>	Administrador cadastra apontamento	Informa dados apontamento	Cadastra apontamento	Retorna mensagem” Cadastrado com sucesso”
<b>34</b>	Administrador cadastra documento	Informa dados documento	Cadastra documento	Retorna mensagem” Cadastrado com sucesso”
<b>35</b>	Administrador cadastra tipo documento	Informa dados tipo documento	Cadastra tipo documento	Retorna mensagem” Cadastrado com sucesso”



<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>36</b>	Administrador cadastra documento versão	Informa dados documento versão	Cadastra documento versão	Retorna mensagem” Cadastrado com sucesso”
<b>37</b>	Administrador cadastra documento necessário	Informa documento necessário	Cadastra documento necessário	Retorna mensagem” Cadastrado com sucesso”
<b>38</b>	Administrador cadastra tipo estágio	Informa tipo estágio	Cadastra tipo estágio	Retorna mensagem” Cadastrado com sucesso”
<b>39</b>	Desativar Administrador	Id Administrador	Desativa Administrador	Retorna mensagem “Desativado com sucesso”
<b>40</b>	Administrador desativa matrícula	Id matrícula	Desativa matrícula	Retorna mensagem “Desativado com sucesso”
<b>41</b>	Administrador desativa curso	Id curso	Desativa curso	Retorna mensagem “Desativado com sucesso”
<b>42</b>	Administrador desativa candidato	Id candidato	Desativa candidato	Retorna mensagem “Desativado com sucesso”
<b>43</b>	Administrador desativa instituição de ensino	Id instituição de ensino	Desativa instituição de ensino	Retorna mensagem “Desativado com sucesso”
<b>44</b>	Administrador desativa coordenador de estágio	Id coordenador de estágio	Desativa coordenador de estágio	Retorna mensagem “Desativado com sucesso”
<b>45</b>	Administrador desativa contrato estágio	Id contrato estágio	Desativa contrato estágio	Retorna mensagem “Desativado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>46</b>	Administrador desativa concedente	Id concedente	Desativa concedente	Retorna mensagem “Desativado com sucesso”
<b>47</b>	Administrador desativa vagas	Id vagas	Desativa vagas	Retorna mensagem “Desativado com sucesso”
<b>48</b>	Administrador desativa cargo	Id cargo	Desativa cargo	Retorna mensagem “Desativado com sucesso”
<b>49</b>	Administrador desativa requisito vagas	Id requisito vaga	Desativa requisito vagas	Retorna mensagem “Desativado com sucesso”
<b>50</b>	Administrador desativa supervisor estágio	Id supervisor estágio	Desativa supervisor estágio	Retorna mensagem “Desativado com sucesso”
<b>51</b>	Administrador desativa processo seletivo	Id processo seletivo	Desativa processo seletivo	Retorna mensagem “Desativado com sucesso”
<b>52</b>	Administrador desativa apontamento	Id apontamento	Desativa apontamento	Retorna mensagem “Desativado com sucesso”
<b>53</b>	Administrador desativa documento	Id documento	Desativa documento	Retorna mensagem “Desativado com sucesso”
<b>54</b>	Administrador desativa tipo documento	Id tipo documento	Desativa tipo documento	Retorna mensagem “Desativado com sucesso”
<b>55</b>	Administrador desativa documento versão	Id documento versão	Desativa documento versão	Retorna mensagem “Desativado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>56</b>	Administrador desativa documento necessário	Id documento necessário	Desativa documento necessário	Retorna mensagem “Desativado com sucesso”
<b>57</b>	Administrador desativa tipo estágio	Id tipo estágio	Desativa tipo estágio	Retorna mensagem “Desativado com sucesso”
<b>58</b>	Exclui Administrador	Id Administrador	Exclui Administrador	Retorna mensagem “Excluído com sucesso”
<b>59</b>	Administrador exclui matrícula	Id matrícula	Exclui matrícula	Retorna mensagem “Desativado com sucesso”
<b>60</b>	Administrador exclui curso	Id curso	Exclui curso	Retorna mensagem “Excluído com sucesso”
<b>61</b>	Administrador exclui candidato	Id candidato	Exclui candidato	Retorna mensagem “Excluído com sucesso”
<b>62</b>	Administrador exclui instituição de ensino	Id instituição de ensino	Exclui instituição de ensino	Retorna mensagem “Excluído com sucesso”
<b>63</b>	Administrador exclui coordenador de estágio	Id coordenador de estágio	Exclui coordenador de estágio	Retorna mensagem “Excluído com sucesso”
<b>64</b>	Administrador exclui contrato estágio	Id contrato estágio	Exclui contrato estágio	Retorna mensagem “Excluído com sucesso”
<b>65</b>	Administrador exclui concedente	Id concedente	Desativa concedente	Retorna mensagem “Excluído com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>66</b>	Administrador exclui vagas	Id vagas	Exclui vagas	Retorna mensagem “Excluído com sucesso”
<b>67</b>	Administrador exclui cargo	Id cargo	Exclui cargo	Retorna mensagem “Excluído com sucesso”
<b>68</b>	Administrador exclui requisito vagas	Id requisito vaga	Exclui requisito vagas	Retorna mensagem “Excluído com sucesso”
<b>69</b>	Administrador desativa supervisor estágio	Id supervisor estágio	Exclui supervisor estágio	Retorna mensagem “Excluído com sucesso”
<b>70</b>	Administrador exclui processo seletivo	Id processo seletivo	Exclui processo seletivo	Retorna mensagem “Excluído com sucesso”
<b>71</b>	Administrador exclui apontamento	Id apontamento	Exclui apontamento	Retorna mensagem” Excluído com sucesso”
<b>72</b>	Administrador exclui documento	Id documento	Exclui documento	Retorna mensagem” Excluído com sucesso”
<b>73</b>	Administrador exclui tipo documento	Id tipo documento	Exclui tipo documento	Retorna mensagem” Excluído com sucesso”
<b>74</b>	Administrador exclui documento versão	Id documento versão	Exclui documento versão	Retorna mensagem” Excluído com sucesso”
<b>75</b>	Administrador exclui documento necessário	Id documento necessário	Exclui documento necessário	Retorna mensagem” Excluído com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>76</b>	Administrador exclui tipo estágio	Id tipo estágio	Exclui tipo estágio	Retorna mensagem” Excluído com sucesso”
<b>77</b>	Consulta Administrador	Id Administrador	Consulta Administrador	Retorna dados administrador
<b>78</b>	Administrador consulta matrícula	Id matrícula	Consulta matrícula	Retorna dados administrador
<b>79</b>	Administrador consulta curso	Id curso	Consulta curso	Retorna dados curso
<b>80</b>	Administrador consulta candidato	Id candidato	Consulta candidato	Retorna dados candidato
<b>81</b>	Administrador consulta instituição de ensino	Id instituição de ensino	Consulta instituição de ensino	Retorna dados instituição de ensino
<b>82</b>	Administrador consulta coordenador de estágio	Id coordenador de estágio	Consulta coordenador de estágio	Retorna dados coordenador de estágio
<b>83</b>	Administrador consulta contrato estágio	Id contrato estágio	Consulta contrato estágio	Retorna dados contrato estágio
<b>84</b>	Administrador consulta concedente	Id concedente	Consulta concedente	Retorna dados concedente
<b>85</b>	Administrador consulta vagas	Id vagas	Consulta vagas	Retorna dados vagas
<b>86</b>	Administrador consulta cargo	Id cargo	Consulta cargo	Retorna dados cargo
<b>87</b>	Administrador consulta requisito vagas	Id requisito vaga	Consulta requisito vagas	Retorna dados requisito vagas
<b>88</b>	Administrador consulta supervisor estágio	Id supervisor estágio	Consulta supervisor estágio	Retorna dados supervisor estágio
<b>89</b>	Administrador consulta processo seletivo	Id processo seletivo	Consulta processo seletivo	Retorna dados processo seletivo
<b>90</b>	Administrador consulta apontamento	Id apontamento	Consulta apontamento	Retorna dados apontamento

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>91</b>	Administrador consulta documento	Id documento	Consulta documento	Retorna dados documento
<b>92</b>	Administrador consulta tipo documento	Id tipo documento	Consulta tipo documento	Retorna dados tipo documento
<b>93</b>	Administrador consulta documento versão	Id documento versão	Consulta documento versão	Retorna dados documento versão
<b>94</b>	Administrador consulta documento necessário	Id documento necessário	Consulta documento necessário	Retorna dados documento necessário
<b>95</b>	Administrador consulta tipo estágio	Id tipo estágio	Consulta tipo estágio	Retorna dados tipo estágio
<b>96</b>	Coordenador Estágio altera apontamento	Informa dados apontamento	Altera apontamento	Retorna mensagem” Alterado com sucesso”
<b>97</b>	Coordenador Estágio altera documento	Informa dados documento	Altera documento	Retorna mensagem” Alterado com sucesso”
<b>98</b>	Coordenador Estágio altera tipo documento	Informa dados tipo documento	Altera tipo documento	Retorna mensagem” Alterado com sucesso”
<b>99</b>	Coordenador Estágio altera documento versão	Informa dados documento versão	Altera documento versão	Retorna mensagem” Alterado com sucesso”
<b>100</b>	Coordenador Estágio altera documento necessário	Informa documento necessário	Altera documento necessário	Retorna mensagem” Alterado com sucesso”
<b>101</b>	Coordenador Estágio altera tipo estágio	Informa tipo estágio	Altera tipo estágio	Retorna mensagem” Alterado com sucesso”
<b>102</b>	Coordenador Estágio cadastra apontamento	Informa dados apontamento	Cadastra apontamento	Retorna mensagem”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
				Cadastrado com sucesso”
<b>103</b>	Coordenador Estágio cadastra documento	Informa dados documento	Cadastra documento	Retorna mensagem” Cadastrado com sucesso”
<b>104</b>	Coordenador Estágio cadastra tipo documento	Informa dados tipo documento	Cadastra tipo documento	Retorna mensagem” Cadastrado com sucesso”
<b>105</b>	Coordenador Estágio cadastra documento versão	Informa dados documento versão	Cadastra documento versão	Retorna mensagem” Cadastrado com sucesso”
<b>106</b>	Coordenador Estágio cadastra documento necessário	Informa documento necessário	Cadastra documento necessário	Retorna mensagem” Cadastrado com sucesso”
<b>107</b>	Coordenador Estágio cadastra tipo estágio	Informa tipo estágio	Cadastra tipo estágio	Retorna mensagem” Cadastrado com sucesso”
<b>108</b>	Coordenador Estágio desativa apontamento	Id apontamento	Desativa apontamento	Retorna mensagem” Desativado com sucesso”
<b>109</b>	Coordenador Estágio desativa documento	Id documento	Desativa documento	Retorna mensagem” Desativado com sucesso”
<b>110</b>	Coordenador Estágio desativa tipo documento	Id tipo documento	Desativa tipo documento	Retorna mensagem” Desativado com sucesso”
<b>111</b>	Coordenador Estágio desativa documento versão	Id documento versão	Desativa documento versão	Retorna mensagem” Desativado com sucesso”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>112</b>	Coordenador Estágio desativa documento necessário	Id documento necessário	Desativa documento necessário	Retorna mensagem” Desativado com sucesso”
<b>113</b>	Coordenador Estágio desativa tipo estágio	Id tipo estágio	Desativa tipo estágio	Retorna mensagem” Desativado com sucesso”
<b>114</b>	Coordenador Estágio exclui apontamento	Id apontamento	Exclui apontamento	Retorna mensagem” Excluído com sucesso”
<b>115</b>	Coordenador Estágio exclui documento	Id documento	Exclui documento	Retorna mensagem” Excluído com sucesso”
<b>116</b>	Coordenador Estágio exclui tipo documento	Id tipo documento	Exclui tipo documento	Retorna mensagem” Excluído com sucesso”
<b>117</b>	Coordenador Estágio exclui documento versão	Id documento versão	Exclui documento versão	Retorna mensagem” Excluído com sucesso”
<b>118</b>	Coordenador Estágio exclui documento necessário	Id documento necessário	Exclui documento necessário	Retorna mensagem” Excluído com sucesso”
<b>119</b>	Coordenador Estágio exclui tipo estágio	Id tipo estágio	Exclui tipo estágio	Retorna mensagem” Excluído com sucesso”
<b>120</b>	Coordenador Estágio consulta apontamento	Id apontamento	Consulta apontamento	Retorna dados apontamento
<b>121</b>	Coordenador Estágio consulta documento	Id documento	Consulta documento	Retorna dados documento
<b>122</b>	Coordenador Estágio consulta tipo documento	Id tipo documento	Consulta tipo documento	Retorna dados tipo documento



<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>123</b>	Coordenador Estágio consulta documento versão	Id documento versão	Consulta documento versão	Retorna dados documento versão
<b>124</b>	Coordenador Estágio consulta documento necessário	Id documento necessário	Consulta documento necessário	Retorna dados documento necessário
<b>125</b>	Coordenador Estágio consulta tipo estágio	Id tipo estágio	Consulta tipo estágio	Retorna dados tipo estágio
<b>126</b>	Supervisor Estágio altera apontamento	Informa dados apontamento	Altera apontamento	Retorna mensagem” Alterado com sucesso”
<b>127</b>	Supervisor Estágio altera documento	Informa dados documento	Altera documento	Retorna mensagem” Alterado com sucesso”
<b>128</b>	Supervisor Estágio altera tipo documento	Informa dados tipo documento	Altera tipo documento	Retorna mensagem” Alterado com sucesso”
<b>129</b>	Supervisor Estágio altera documento versão	Informa dados documento versão	Altera documento versão	Retorna mensagem” Alterado com sucesso”
<b>130</b>	Supervisor Estágio altera documento necessário	Informa documento necessário	Altera documento necessário	Retorna mensagem” Alterado com sucesso”
<b>131</b>	Supervisor Estágio altera tipo estágio	Informa tipo estágio	Altera tipo estágio	Retorna mensagem” Alterado com sucesso”
<b>132</b>	Supervisor Estágio cadastra apontamento	Informa dados apontamento	Cadastra apontamento	Retorna mensagem” Cadastrado com sucesso”
<b>133</b>	Supervisor Estágio cadastra documento	Informa dados documento	Cadastra documento	Retorna mensagem”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
				Cadastrado com sucesso”
<b>134</b>	Supervisor Estágio cadastra tipo documento	Informa dados tipo documento	Cadastra tipo documento	“Cadastrado com sucesso”
<b>135</b>	Supervisor Estágio cadastra documento versão	Informa dados documento versão	Cadastra documento versão	Retorna mensagem” Cadastrado com sucesso”
<b>136</b>	Supervisor Estágio cadastra documento necessário	Informa documento necessário	Cadastra documento necessário	Retorna mensagem” Cadastrado com sucesso”
<b>137</b>	Supervisor Estágio cadastra tipo estágio	Informa tipo estágio	Cadastra tipo estágio	Retorna mensagem” Cadastrado com sucesso”
<b>138</b>	Supervisor Estágio desativa apontamento	Id apontamento	Desativa apontamento	Retorna mensagem” Desativado com sucesso”
<b>139</b>	Supervisor Estágio desativa documento	Id documento	Desativa documento	Retorna mensagem” Desativado com sucesso”
<b>140</b>	Supervisor Estágio desativa tipo documento	Id tipo documento	Desativa tipo documento	Retorna mensagem” Desativado com sucesso”
<b>141</b>	Supervisor Estágio desativa documento versão	Id documento versão	Desativa documento versão	Retorna mensagem” Desativado com sucesso”
<b>142</b>	Supervisor desativa documento necessário	Id documento necessário	Desativa documento necessário	Retorna mensagem” Desativado com sucesso”
<b>143</b>	Supervisor Estágio desativa tipo estágio	Id tipo estágio	Desativa tipo estágio	Retorna mensagem”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
				Desativado com sucesso”
<b>144</b>	Supervisor Estágio exclui apontamento	Id apontamento	Exclui apontamento	Retorna mensagem” Excluído com sucesso”
<b>145</b>	Supervisor Estágio exclui documento	Id documento	Exclui documento	Retorna mensagem” Excluído com sucesso”
<b>146</b>	Supervisor Estágio exclui tipo documento	Id tipo documento	Exclui tipo documento	Retorna mensagem” Excluído com sucesso”
<b>147</b>	Supervisor Estágio exclui documento versão	Id documento versão	Exclui documento versão	Retorna mensagem” Excluído com sucesso”
<b>148</b>	Supervisor Estágio exclui documento necessário	Id documento necessário	Exclui documento necessário	Retorna mensagem” Excluído com sucesso”
<b>149</b>	Supervisor Estágio exclui tipo estágio	Id tipo estágio	Exclui tipo estágio	Retorna mensagem” Excluído com sucesso”
<b>150</b>	Supervisor Estágio consulta apontamento	Id apontamento	Consulta apontamento	Retorna dados apontamento
<b>151</b>	Supervisor Estágio consulta documento	Id documento	Consulta documento	Retorna dados documento
<b>152</b>	Supervisor Estágio consulta tipo documento	Id tipo documento	Consulta tipo documento	Retorna dados tipo documento
<b>153</b>	Supervisor Estágio consulta documento versão	Id documento versão	Consulta documento versão	Retorna dados documento versão
<b>154</b>	Supervisor Estágio consulta documento necessário	Id documento necessário	Consulta documento necessário	Retorna dados documento necessário

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>155</b>	Supervisor Estágio consulta tipo estágio	Id tipo estágio	Consulta tipo estágio	Retorna dados tipo estágio
<b>156</b>	Concedente altera contrato estágio	Informa dados contrato estágio	Cadastra contrato estágio	Retorna mensagem “Cadastrado com sucesso”
<b>157</b>	Concedente altera vagas	Informa dados vagas	Altera vagas	Retorna mensagem “Alterado com sucesso”
<b>158</b>	Concedente altera requisito vagas	Informa dados requisito vaga	Altera requisito vagas	Retorna mensagem “Alterado com sucesso”
<b>159</b>	Concedente cadastra contrato	Informa dados contrato	Cadastra contrato	Retorna mensagem” Cadastrado com sucesso”
<b>160</b>	Concedente cadastra vagas	Informa dados vagas	Cadastra vagas	Retorna mensagem” Cadastrado com sucesso”
<b>161</b>	Concedente cadastra requisito vagas	Informa dados requisito vagas	Cadastra requisito vagas	Retorna mensagem” Cadastrado com sucesso”
<b>162</b>	Concedente desativa contrato	Id contrato	Desativa contrato	Retorna mensagem” Desativado com sucesso”
<b>163</b>	Concedente desativa requisito vagas	Id requisito vagas	Desativa requisito vagas	“Desativado com sucesso”
<b>164</b>	Concedente desativa vagas	Id vagas	Desativa vagas	Retorna mensagem” Desativado com sucesso”
<b>165</b>	Concedente exclui contrato	Id documento contrato	Exclui documento contrato	Retorna mensagem”

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
				Excluído com sucesso”
<b>166</b>	Concedente exclui requisito vagas	Id documento requisito vagas	Exclui documento requisito vagas	Retorna mensagem” Excluído com sucesso”
<b>167</b>	Concedente exclui vagas	Id documento vagas	Exclui documento vagas	Retorna mensagem” Excluído com sucesso”
<b>168</b>	Concedente consulta contrato	Id contrato	Consulta contrato	Retorna dados contrato
<b>169</b>	Concedente consulta requisito vagas	Id requisito vagas	Consulta requisito vagas	Retorna dados requisito vagas
<b>170</b>	Concedente consulta vagas	Id vagas	Consulta vagas	Retorna dados vagas
<b>171</b>	Sistema Lista Administrador		Listar administrador	Retorna lista administrador
<b>172</b>	Sistema Lista matrícula		Listar matrícula	Retorna lista matrícula
<b>173</b>	Sistema Lista curso		Listar curso	Retorna lista curso
<b>174</b>	Sistema Lista candidato		Listar candidato	Retorna lista candidato
<b>175</b>	Sistema Lista instituição de ensino		Listar instituição de ensino	Retorna lista instituição de ensino
<b>176</b>	Sistema Lista coordenador de estágio		Listar coordenador de estágio	Retorna lista coordenador de estágio
<b>177</b>	Sistema Lista contrato estágio		Listar contrato estágio	Retorna lista contrato estágio
<b>178</b>	Sistema Lista concedente		Listar concedente	Retorna lista concedente
<b>179</b>	Sistema Lista vagas		Listar vagas	Retorna lista vagas

<b>Nº</b>	<b>Descrição do Caso de Uso</b>	<b>Entrada</b>	<b>Caso de Uso</b>	<b>Resposta</b>
<b>180</b>	Sistema Lista cargo		Listar cargo	Retorna lista cargo
<b>181</b>	Sistema Lista requisito vagas		Listar requisito vagas	Retorna lista requisito vagas
<b>182</b>	Sistema Lista supervisor estágio		Listar supervisor estágio	Retorna lista supervisor estágio
<b>183</b>	Sistema Lista processo seletivo		Listar processo seletivo	Retorna lista processo seletivo
<b>184</b>	Sistema Lista apontamento		Listar apontamento	Retorna lista apontamento
<b>185</b>	Sistema Lista documento		Listar documento	Retorna lista documento
<b>186</b>	Sistema Lista tipo documento		Listar tipo documento	Retorna lista tipo documento
<b>187</b>	Sistema Lista documento versão		Listar documento versão	Retorna lista documento versão
<b>188</b>	Sistema Lista documento necessário		Listar documento necessário	Retorna lista documento necessário
<b>189</b>	Sistema Lista estágio		Listar tipo estágio	Retorna lista estágio

### 3.5. DIAGRAMA DE CASOS DE USO

Conforme enfatizado na obra (GUEDES, 2011), o Diagrama de Caso de Uso é o mais abstrato, sendo, portanto, o mais flexível e informal entre as representações, utilizado para proporcionar uma visão global do sistema. Seu propósito central é a identificação dos atores que interagirão com o software, acompanhados de suas respectivas funcionalidades. Este diagrama desfruta de ampla aplicação nas fases iniciais de levantamento e análise de requisitos, sendo, no entanto, igualmente valioso ao longo de todo o processo de modelagem, podendo inclusive servir como alicerce para a elaboração de outros diagramas

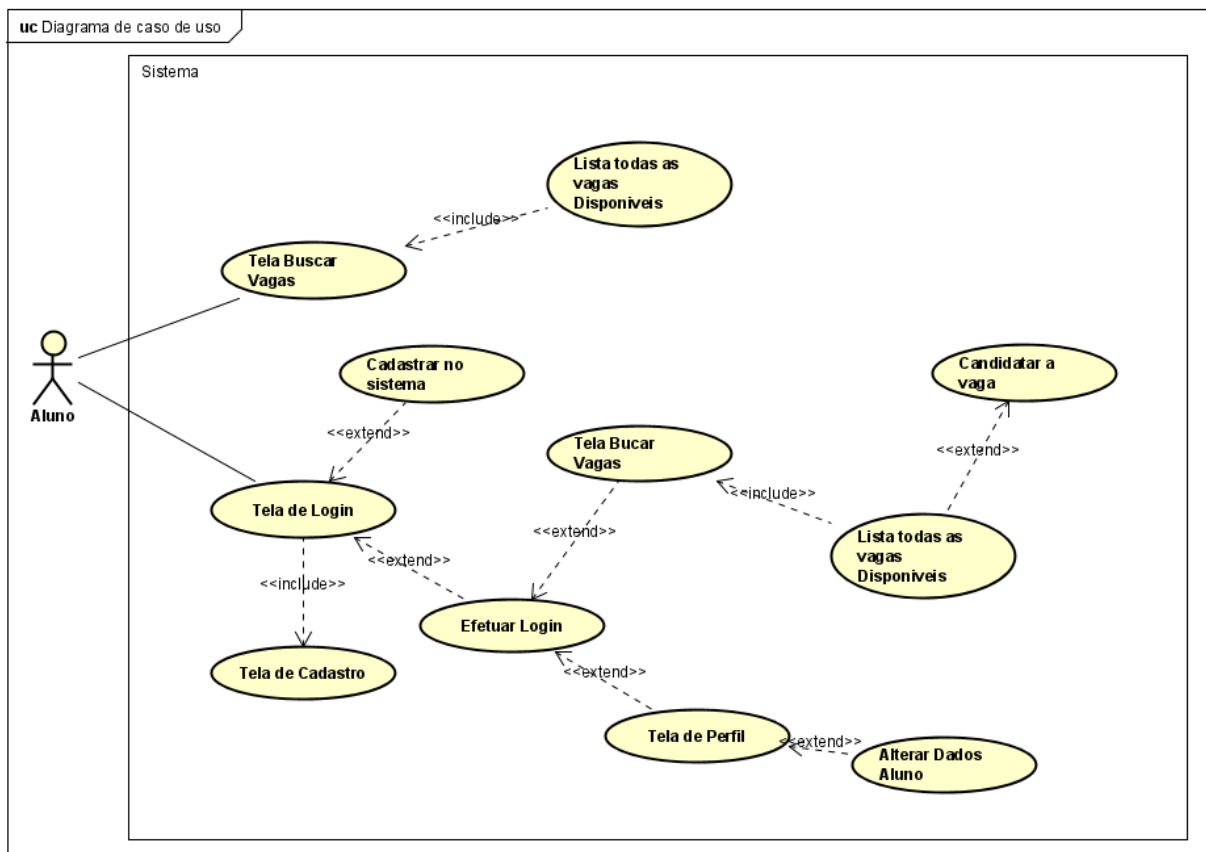
De acordo com a figura 06, o aluno pode interagir com várias funcionalidades do sistema. O sistema oferece diversas funcionalidades para o usuário. Inicialmente, o aluno acessa a tela de cadastro, onde pode criar uma nova conta ao fornecer informações básicas, como nome, e-mail e senha.

Após concluir o cadastro, o aluno utiliza a tela de login para acessar o sistema. Uma vez autenticado, ele tem acesso à tela de busca de vagas, que permite listar todas as oportunidades disponíveis. Essa interface oferece filtros para facilitar a pesquisa por vagas específicas.

O aluno também pode se candidatar a uma vaga diretamente pelo sistema. Além disso, há uma tela de perfil onde ele pode visualizar e editar suas informações pessoais, garantindo que seus dados estejam sempre atualizados.

Para assegurar a consistência e a qualidade dos dados, o sistema realiza validações em tempo real durante todas as etapas de interação.

**Figura 6** — Diagrama de caso de uso – Visão Aluno



Fonte: Elaborado pelos autores.

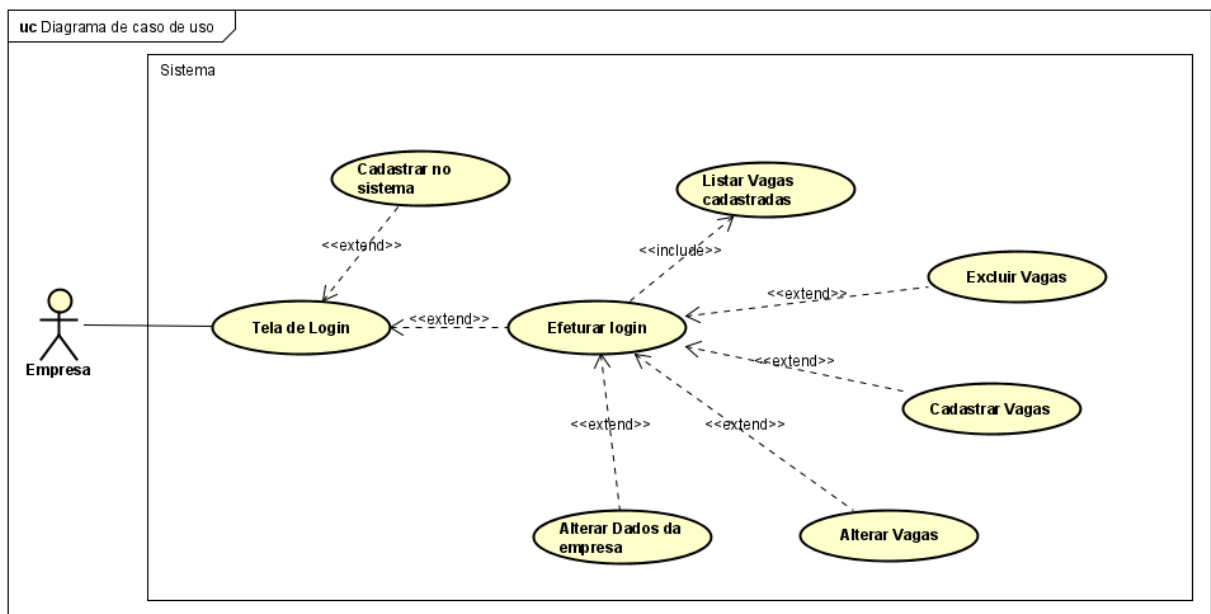
De acordo com a figura 07, A empresa pode realizar diversas ações dentro do sistema. Primeiramente, ela acessa a tela de cadastro, onde pode criar uma nova conta ao fornecer informações essenciais.

Após realizar o cadastro, a empresa utiliza a tela de login para acessar suas funcionalidades. Uma vez autenticada, é possível cadastrar novas vagas por meio de uma interface específica, onde são inseridos os detalhes das oportunidades oferecidas.

Além disso, a empresa pode listar as vagas já cadastradas, obtendo uma visão geral e controle sobre essas oportunidades. Caso necessário, há também a opção de editar ou excluir vagas existentes.

Outra funcionalidade importante é a possibilidade de alterar os dados da empresa, garantindo que as informações estejam sempre atualizadas. Todas essas ações contam com validações em tempo real, assegurando que os dados sejam inseridos corretamente e que as operações sejam realizadas de maneira eficiente.

**Figura 7** — Diagrama de caso de uso – Visão da Empresa



Fonte: Elaborado pelos autores.

### 3.6. DIAGRAMA DE CASOS DE USO INDIVIDUAIS

Conforme destacado na obra (SOMMERVILLE, 2018), os diagramas de casos de uso individuais são fundamentais para detalhar interações específicas entre usuários e o sistema. Eles oferecem uma visão focada em casos de uso particulares, permitindo uma compreensão mais clara de funcionalidades específicas. Esses diagramas são essenciais durante a análise de requisitos, ajudando a esclarecer como cada parte do sistema deve responder a interações

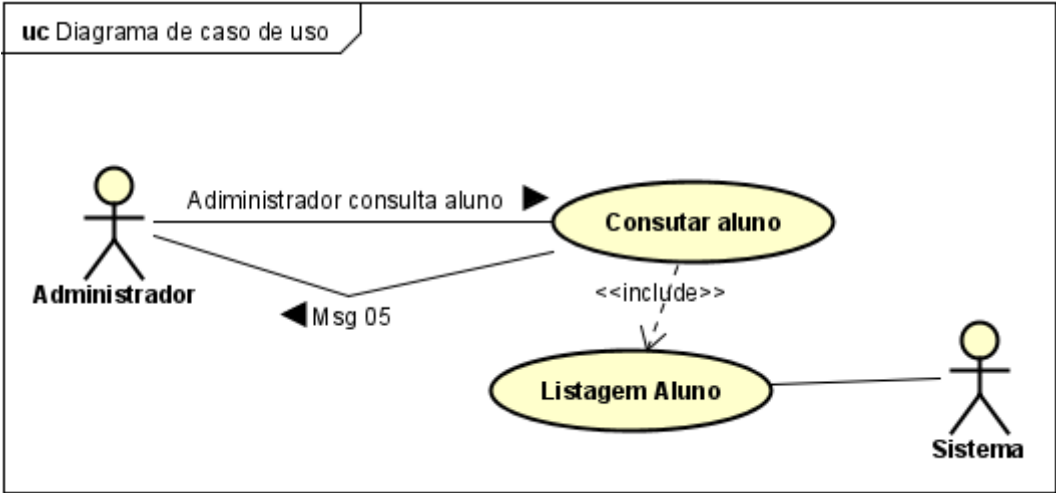


distintas. Além disso, servem como base para o desenvolvimento de outros diagramas de design e implementação.

3.6.1. DIAGRAMA DE CASO DE USO – CONSULTAR ALUNO

O administrador pode consultar dados de alunos no sistema, obtendo informações detalhadas e uma visão geral. A função de consulta facilita o acesso rápido e organizado aos dados estudantis.

Figura 8 — Diagrama de caso de uso – Consultar Aluno



Fonte: Elaborado pelos autores.

Quadro 9 – Documentação do caso de uso Consultar Aluno

Documentação	
Nome do caso de uso	Consultar Aluno
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o Administrador consulte as informações do aluno
Pré-condições	Aluno cadastradas
Pós-condições	O usuário é direcionado a parte desejada do sistema
Fluxo normal	
Ações do ator	Ações do sistema
1. Usuário solicita alunos cadastrados	
	2. Sistema recebe solicitação e mostra tela em lista os alunos cadastrados
3. Usuário seleciona aluno que deseja saber mais detalhes	
	4. Sistema valida solicitação

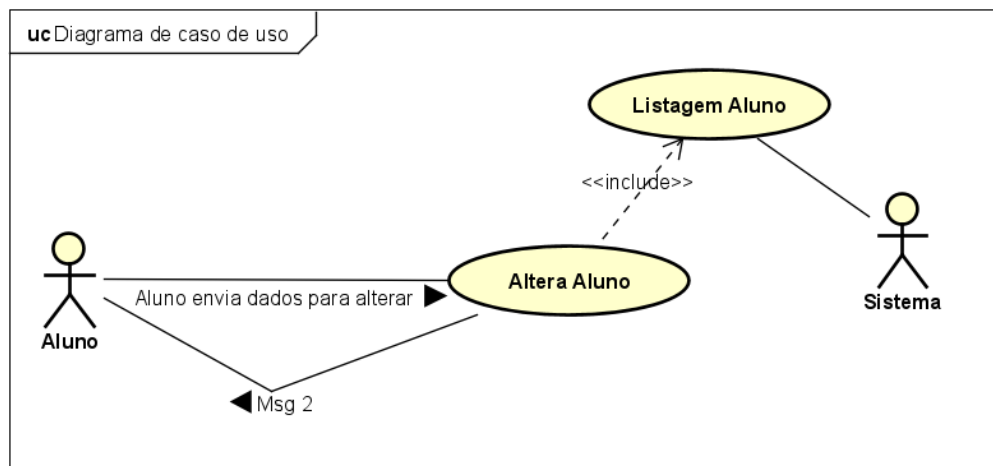
	5. Sistema carrega tela de consulta.
<b>Fluxo alternativo</b>	
	4. Sistema lista solicitação
	4.1 Sistema não consegue conectar com o banco
	4.1.2 Sistema envia mensagem “Erro com o banco de dados”
	4.1.3 Sistema retorna ao fluxo normal

Fonte: Elaborado pelos autores.

### 3.6.2. DIAGRAMA DE CASO DE USO – ALTERAR ALUNO

Relacionado a figura 9, ilustra que o administrador pode alterar dados de alunos no sistema. Ao utilizar a função de alteração, ele atualiza informações específicas, como detalhes do curso. Essa ação está integrada à listagem de alunos, permitindo uma visão geral atualizada. O sistema facilita essas modificações, garantindo que os dados sejam precisos e organizados.

**Figura 9** — Diagrama de caso de uso – Alterar Aluno



Fonte: Elaborado pelos autores.

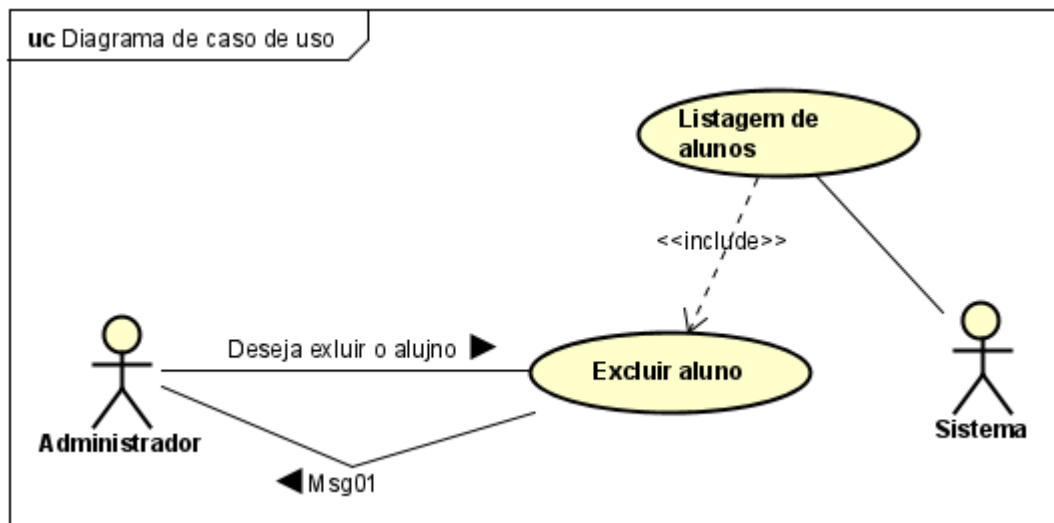
**Quadro 10** — Documentação de caso de uso – Alterar Aluno

<b>Documentação</b>	
<b>Nome do caso de uso</b>	<b>Alterar Aluno</b>
<b>Ator principal</b>	Usuário (Administrador)
<b>Resumo</b>	Esse caso de uso descreve as ações necessárias para que o Administrador altere as informações do aluno
<b>Pré-condições</b>	Aluno já cadastrado
<b>Pós-condições</b>	O usuário é direcionado a parte desejada do sistema
<b>Fluxo normal</b>	
<b>Ações do ator</b>	<b>Ações do sistema</b>
1. Usuário solicita alterar um aluno	
	2. Sistema recebe solicitação e mostra a tela com os dados já cadastrados
3. Após alterar os dados do aluno o usuário aperta no botão “Salvar alterações”	
	4. Sistema valida solicitação e altera dados.
	5. Sistema envia mensagem “Alterado com sucesso”
	6. Sistema retorna para a tela com a tabela de listagem.
<b>Fluxo alternativo</b>	
	4. Sistema valida solicitação e altera dados.
	4.1 Sistema falha ao se comunicar com a controller
	4.1.1 Sistema envia a mensagem de erro
	4.1.2 Sistema retorna ao fluxo normal pelo item 2

Fonte: Elaborado pelos autores.

### 3.6.3. DIAGRAMA DE CASO DE USO – EXCLUIR ALUNO

Com base na figura 11, pode se ver administrador tem a capacidade de excluir alunos do sistema. Ao iniciar a exclusão, ele remove o aluno selecionado da listagem geral. Essa ação está vinculada à listagem de alunos, garantindo que a informação seja atualizada imediatamente. O sistema assegura que a exclusão seja feita de forma eficiente e organizada, mantendo os dados precisos.

**Figura 10** — Diagrama de caso de uso – Excluir Aluno

Fonte: Elaborado pelos autores.

**Quadro 11** — Documentação de caso de uso – Excluir Aluno

Documentação	
Nome do caso de uso	<b>Excluir Aluno</b>
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o Administrador apague as informações do aluno
Pré-condições	Aluno excluído
Pós-condições	O usuário é direcionado a parte desejada do sistema
Fluxo normal	
Ações do ator	Ações do sistema
1.Usuário solicita excluir um aluno	
	2.Sistema abre um dropdown com o botão “Excluir”
3.Usuário aperta o botão “Excluir”	
	4.Sistema valida solicitação
	5.Sistema volta para a tela de listar alunos
Fluxo alternativo	
	4.Sistema lista solicitação
	4.1Sistema não consegue conectar com o banco
	4.1.1Sistema envia mensagem “Erro com o banco de dados”
	4.1.2Sistema retorna ao fluxo normal

Fonte: Elaborado pelos autores.

### 3.7. DIAGRAMA DE SEQUÊNCIA

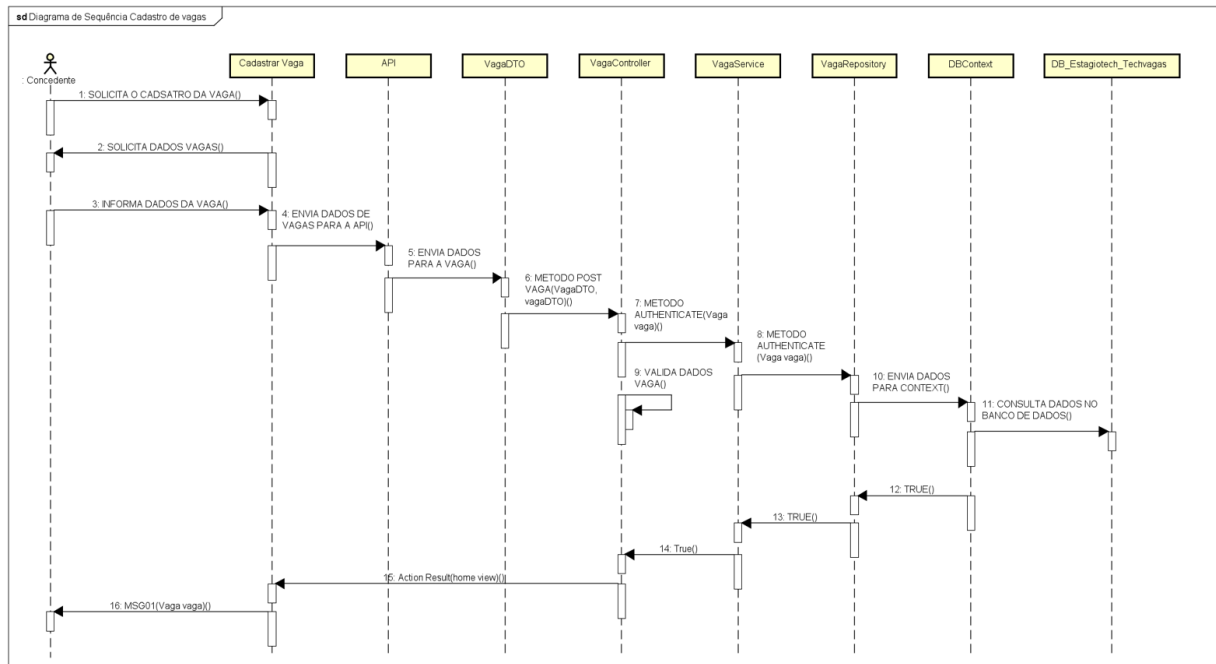
O diagrama de sequência adota um modelo que ilustra o comportamento do usuário ao realizar ações dentro da aplicação, observando a ordem temporal das ações no desenvolvimento do processo. Seu propósito é identificar o evento que inicia o processo, o ator responsável por esse evento e a sequência de desenvolvimento que deve ocorrer. Em resumo, proporciona uma representação visual destacando a interação entre diferentes elementos da aplicação, facilitando a compreensão da lógica temporal e das responsabilidades associadas às ações realizadas (GUEDES, 2011).

#### 3.7.1. DIAGRAMA DE SEQUENCIA – CADASTRAR VAGA

Na figura 11 temos o diagrama de sequência que mostra o processo de cadastro de vagas no sistema TechVagas, evidenciando as interações entre o concedente e os componentes do sistema. O fluxo começa quando o concedente solicita o cadastro e fornece os dados da vaga por meio da interface do sistema.

Esses dados são enviados para a API, que os encapsula em um objeto do tipo VagaDTO e os encaminha ao VagaController. O controlador valida e processa as informações, acionando o VagaService, que realiza verificações adicionais e comunica-se com o VagaRepository para armazenar os dados no banco de dados.

Após o registro bem-sucedido, o sistema retorna uma confirmação ao concedente, finalizando o processo. Esse fluxo garante a validação e integridade dos dados, além de facilitar a organização e manutenção do sistema.

**Figura 11** — Diagrama de Sequência – Cadastrar vaga

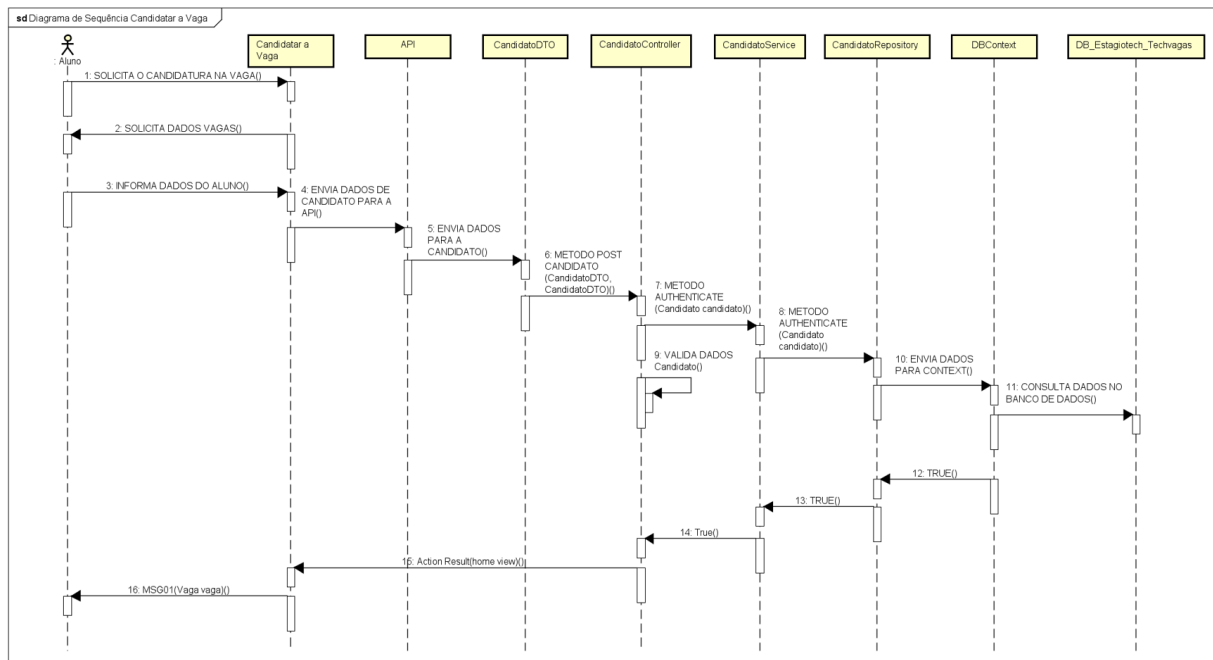
Fonte: Elaborado pelos autores.

### 3.7.2. DIAGRAMA DE SEQUENCIA – CANDIDATAR A VAGA

Na figura 12 esta o diagrama de sequência que representa o processo de candidatura a uma vaga no sistema TechVagas, destacando as interações entre o aluno e os componentes do sistema. O fluxo inicia quando o aluno solicita a candidatura, fornecendo as informações necessárias por meio da interface do sistema.

Esses dados são enviados para a API, que os encapsula em um objeto do tipo CandidatoDTO e os repassa ao CandidateController. O controlador verifica e valida as informações, repassando-as ao CandidateService, que realiza a autenticação e direciona os dados ao CandidateRepository. Este, por sua vez, interage com o DBContext para salvar as informações no banco de dados.

Após a conclusão do processo, o sistema retorna uma confirmação ao aluno, indicando que a candidatura foi realizada com sucesso. Esse fluxo garante a validação e o processamento correto das informações, além de otimizar a interação entre os módulos do sistema.

**Figura 12** — Diagrama de Sequência – Candidatar a vaga

Fonte: Elaborado pelos autores.

#### 4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX)

User Experience (UX), ou Experiência do Usuário, é um conceito central no design de sistemas interativos, focando na qualidade da interação entre o usuário e o produto. A UX envolve todos os aspectos dessa interação, desde a usabilidade e acessibilidade do sistema até a satisfação do usuário ao utilizá-lo. Em outras palavras, UX não é apenas sobre se um sistema é funcional, mas também sobre como ele é percebido e sentido pelo usuário. Segundo Norman (2013), "a experiência do usuário não é apenas sobre a funcionalidade de um produto, mas também sobre o prazer e a facilidade com que ele pode ser utilizado"

De acordo com Sommerville (2018), a experiência do usuário vai além da simples funcionalidade de um sistema, abrangendo fatores como eficácia, eficiência e satisfação durante sua utilização. Ele destaca que a usabilidade — um dos componentes centrais da UX — está diretamente relacionada à facilidade com que os usuários podem aprender a usar um sistema, o quão eficiente ele é em suportar suas tarefas, e a satisfação que proporciona ao longo do uso.

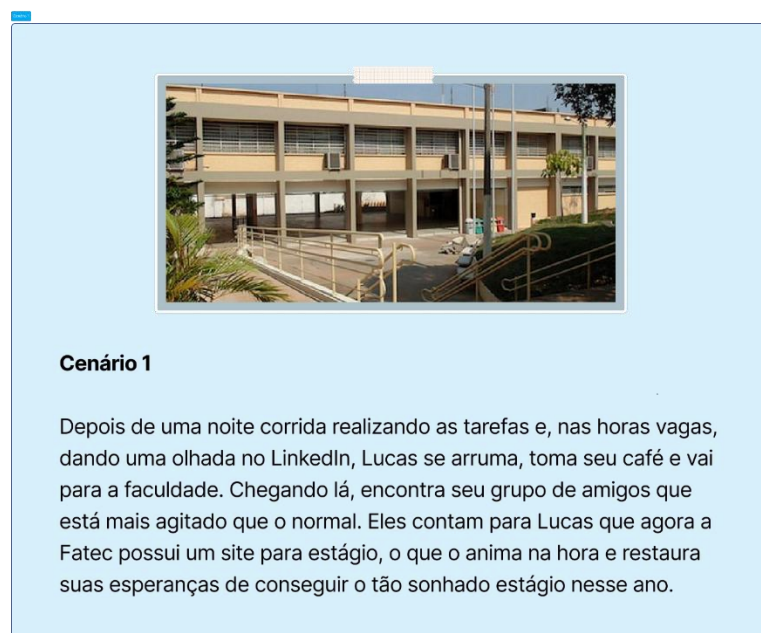
##### 4.1 DESCRIÇÃO DE CENÁRIO

Conforme escrito no livro O Guia para Projetar UX, de Russ Unger e Carolyn Chandler (2013), os cenários representam narrativas detalhadas que ilustram as interações entre um usuário e um sistema. Estas histórias oferecem uma compreensão abrangente ao contextualizar

a persona, seus objetivos, as ações específicas em busca desses objetivos, os desafios enfrentados e os desfechos das interações. Os cenários de persona são ferramentas fundamentais no direcionamento do design de experiência do usuário, permitindo uma avaliação minuciosa da eficácia do sistema em diferentes circunstâncias.

Com base na figura 13 o cenário apresentado ilustra a rotina de Lucas, um estudante que, após realizar suas tarefas e buscar oportunidades no LinkedIn, dirige-se à faculdade. Lá, ele descobre por meio de amigos que a Fatec possui um site dedicado a estágios. Essa notícia renova sua esperança de conquistar um estágio tão almejado neste ano, proporcionando um novo ânimo e motivação para continuar sua busca profissional.

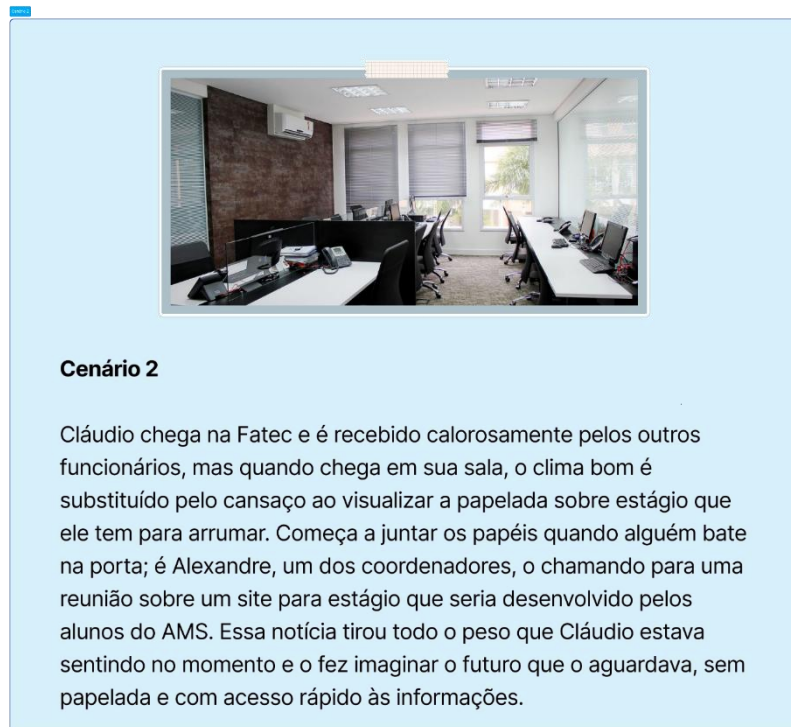
**Figura 13** — Imagem do primeiro cenário



Fonte: Elaborado pelos autores.

Com base na figura 14 mostra-se o cenário onde Cláudio, funcionário da Fatec, que ao chegar ao trabalho é recebido com cordialidade pelos colegas. Porém, ao se deparar com uma pilha de papéis relacionados a estágios, sente-se sobrecarregado. Sua preocupação muda quando Alexandre, um coordenador, o convida para uma reunião sobre o desenvolvimento de um site de estágios pelos alunos do curso AMS. Essa notícia traz alívio e otimismo, permitindo que Cláudio visualize um futuro mais organizado e acessível para essa demanda.



**Figura 14** — Imagem do segundo cenário

Fonte: Elaborado pelos autores.

#### 4.2 DESCRIÇÃO DE PERSONAS

De acordo com Unger e Chandler (2013), personas constituem representações fictícias de usuários envolvidos na interação com um sistema específico. Seu desenvolvimento é direcionado à compreensão aprofundada das necessidades, comportamentos e motivações desses usuários. A criação de personas visa proporcionar aos designers e desenvolvedores uma visão mais humanizada e centrada no usuário, possibilitando a concepção de soluções que correspondam de maneira eficaz às expectativas e requisitos dos usuários finais.

Referente a figura 15 Lucas Henrique é um estudante de 22 anos, cursando o 4º semestre de Análise e Desenvolvimento de Sistemas na Fatec Jales. Apaixonado por tecnologia, ele busca constantemente se atualizar sobre as tendências do mercado e aplicar seus conhecimentos em situações práticas. Seu principal desafio é encontrar um estágio alinhado aos seus interesses e habilidades, evitando processos burocráticos e complexos. Recentemente, Lucas ficou animado ao saber sobre o TechVagas, uma plataforma que promete facilitar o acesso a informações e documentos relacionados a estágios, tornando esse processo mais eficiente e menos estressante.

**Figura 15** — Imagem da primeira persona



## Lucas Henrique

"Estudante"

- Idade: 22 anos.
- Gênero: Masculino.
- Profissão: Estudante.
- Localidade(cidade): Jales.
- Familiaridade com a tecnologia (0 a 10): 10.

---

### Descrição

Lucas Henrique, um aluno dedicado do 4º semestre de Análise e Desenvolvimento de Sistemas na Fatec Jales, é apaixonado por tecnologia. Ele se mantém atualizado com as últimas tendências do mercado de trabalho, sempre em busca de maneiras de aplicar seus conhecimentos em situações práticas.

### Desafios e Objetivos:

Desde o início da faculdade, Lucas tem enfrentado o desafio de encontrar um estágio que esteja alinhado com seus interesses e conhecimentos. Ele busca uma solução que simplifique o caminho para conseguir um estágio, permitindo focar mais em suas habilidades técnicas e menos nas complexidades administrativas. Recentemente, colegas na faculdade mencionaram o EstagioTech, um site que promete centralizar todos os documentos e informações relacionados ao termo de compromisso de estágio, proporcionando uma abordagem mais eficiente e descomplicada para a obtenção de estágios.

Fonte: Elaborado pelos autores.

De acordo com a figura 16, Cláudio Santos, desempenha um papel essencial como supervisor de estágio na Fatec Jales, onde atua há três anos. Aos 40 anos, com formação e experiência na área de Administração, Cláudio é reconhecido por sua habilidade em gerenciar processos burocráticos relacionados a estágios, garantindo conformidade e eficiência na integração dos estudantes ao mercado de trabalho.

**Figura 16** — Imagem da Segunda persona

Fonte: Elaborado pelos autores.

#### 4.3 ESBOÇOS DE TELA (WIREFRAMES)

Como afirma Don Norman, autor de *The Design of Everyday Things*, “protótipos são uma ferramenta fundamental para explorar possíveis soluções de design antes da implementação completa” (Norman, 2013). Os protótipos de tela são representações tangíveis e interativas da proposta de design, permitindo a exploração, comunicação e refinamento de ideias.

Assim, os modelos iniciais de tela não só promovem a comunicação efetiva entre as equipes de design e desenvolvimento, como também têm um papel crucial na detecção antecipada de adaptações necessárias, resultando em economia de tempo e recursos.

De acordo com o apresentado na figura 17, tela de cadastro de usuário, o visitante pode criar uma conta ao fornecer informações essenciais, como nome, e-mail e senha. Essa interface

geralmente inclui validações em tempo real para garantir que os dados sejam inseridos corretamente, além de opções para aceitar os termos de uso e políticas de privacidade.

**Figura 17** — Tela de cadastro do usuário

TECH  
vagas

Fatec Faculdade de Tecnologia CPS CPS São Paulo SÃO PAULO

Login Criar conta

**Criar sua conta**  
Conectando Sonhos a Oportunidades. Crie sua conta e inicie sua jornada para um futuro profissional extraordinário!

Nome

E-mail

Telefone

Senha

Confirmar senha

Criar

Ao clicar em continuar, você concorda com nossos [Termos de serviço](#) e [política de Privacidade](#).

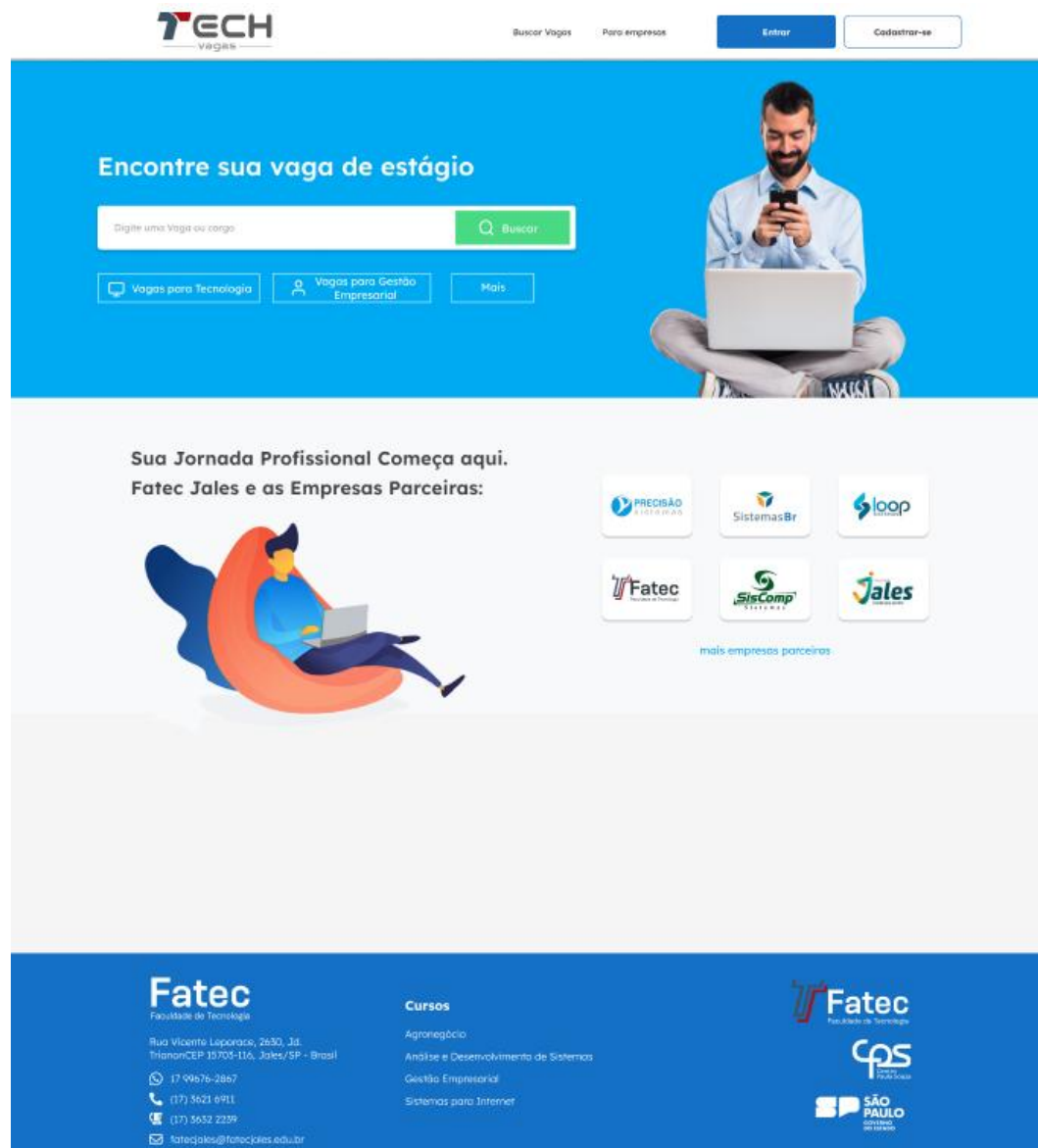
Fonte: Elaborado pelos autores.

De acordo com o apresentado na figura 18, tela de login do usuário possibilita o visitante pode inserir suas credenciais, como nome de usuário e senha, para acessar sua conta de forma segura. Além disso, há opções para recuperar a senha em caso de esquecimento e, muitas vezes, um recurso que permite lembrar as informações de login em dispositivos confiáveis. A interface geralmente inclui feedback visual, informando ao usuário sobre possíveis erros de autenticação.

**Figura 18** — Tela de login do usuário

Fonte: Elaborado pelos autores.

Referente a figura 19, a tela inicial do sistema é projetada para ser intuitiva e acolhedora, incentivando os alunos a explorarem oportunidades de estágio de forma eficiente. No centro, há uma barra de pesquisa que permite aos usuários procurarem vagas específicas, acompanhada de botões de acesso rápido para categorias como “Tecnologia” e “Gestão Empresarial”. Essa organização facilita a navegação e ajuda os alunos a encontrarem rapidamente oportunidades alinhadas aos seus interesses. A seção de parcerias destaca a colaboração com empresas locais, reforçando a conexão entre a academia e o mercado de trabalho.

**Figura 19** — Tela de inicial do sistema

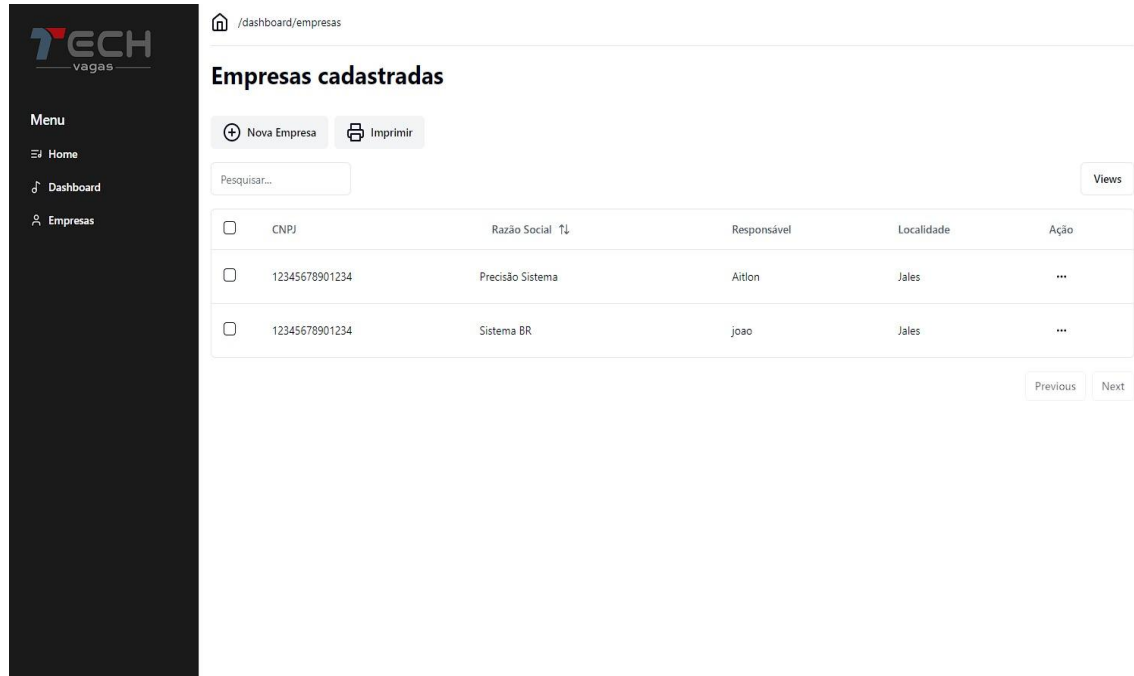
Fonte: Elaborado pelos autores.

Na figura 20, a tela de cadastro de empresas permite que organizações parceiras criem uma conta fornecendo informações importantes, como nome da empresa, CNPJ, e-mail e telefone de contato. A interface é projetada para ser clara e eficiente, incorporando validações em tempo real para garantir que os dados sejam inseridos corretamente. Além disso, há opções para que as empresas aceitem os termos de uso e políticas de privacidade, assegurando conformidade legal e transparência.

Após o cadastro, a empresa pode cadastrar, alterar ou excluir vagas de estágio, inserindo detalhes como descrição da vaga, requisitos, local de trabalho e benefícios oferecidos.

Esse processo não só facilita a entrada de empresas na plataforma, mas também fortalece a rede de parcerias, criando um ecossistema robusto de oportunidades de estágio para os alunos.

**Figura 20** — Tela de empresas cadastradas



Fonte: Elaborado pelos autores.

#### 4.4 PROTÓTIPOS DE TELA

O termo protótipo vem do grego prótos (primeiro) e typos (tipo), e, ao traduzirmos para o sentido literal, significa “primeiro tipo” ou “primeiro modelo”. O conceito de protótipo já está diretamente relacionado à ideia de projetar ou planejar algo, seja uma viagem em família, que necessita de um “plano” para ser organizada, ou até mesmo um projeto acadêmico ou profissional, no qual um protótipo é essencial para que a tarefa seja bem executada e apresente um ótimo desempenho. Nesse contexto, “prototipar é uma maneira de experimentar e aprender. Não é uma simples representação, mas sim um meio de entender melhor as necessidades dos usuários” (BROWN, 2010).

"Prototipar é uma maneira de experimentar e aprender. Não é uma simples representação, mas sim um meio de entender melhor as necessidades dos usuários." (BROWN, 2010).

A prototipagem centrada no usuário é um verdadeiro diálogo entre criadores e usuários, onde a empatia se torna a chave para o sucesso. Imagine um designer não apenas como

alguém que cria, mas como um investigador que busca entender as histórias e sentimentos das pessoas que usarão seu produto. Esse processo envolve um olhar atento às necessidades humanas e um compromisso em traduzir essas percepções em soluções práticas.

Ao criar protótipos, os designers podem testar ideias em um ambiente real, ouvindo atentamente o que os usuários têm a dizer e fazendo ajustes baseados nessas interações. Esse processo vai além da simples funcionalidade; trata-se de construir conexões emocionais e garantir que cada detalhe do design ressoe com as necessidades e desejos de quem o utiliza.

Quando os usuários se tornam co-criadores, o resultado é uma solução que não apenas resolve problemas, mas também faz as pessoas se sentirem valorizadas e compreendidas. Essa abordagem transforma a inovação em uma experiência compartilhada, onde cada feedback é uma oportunidade de crescimento. Assim, a prototipagem deixa de ser apenas um passo no design para se tornar um compromisso com a criação de experiências significativas e humanas.

#### 4.4.1. DEMONSTRAÇÃO DOS PROTÓTIPOS DAS TELA BASEADAS NOS WIREFRAMES

Os wireframes são representações visuais simplificadas que ilustram a estrutura e o layout básico de interfaces de usuário, sendo essenciais no processo de design de experiência do usuário. Segundo Nielsen (2006), "wireframes são esboços visuais que representam a estrutura e o layout básico de uma página da web ou aplicativo" (p. 24). Eles ajudam a visualizar e testar rapidamente o design de uma interface antes de se aprofundar em detalhes mais complexos, como estilos visuais e interações avançadas.

A tela apresentada na figura 21 é o formulário de criação de perfil no sistema Tech Vagas, projetada para ser clara e funcional. O usuário insere dados como e-mail ou nome de usuário, senha (com confirmação) e CPF, com validações em tempo real para evitar erros. O campo de aceite dos termos de uso garante transparência e conformidade legal. O design minimalista e o rodapé com informações institucionais da Fatec Jales agregam credibilidade e simplicidade, enquanto os botões bem posicionados facilitam o cadastro ou redirecionam para o login, proporcionando uma experiência prática e eficiente.



**Figura 21** — Protótipo tela de Cadastro

**TECH**  
vagas

Ajuda

←

Crie seu perfil

Email ou usuário

Senha

De 7 a 20 caracteres (números e letras)

Confirmar senha

CPF

☐ Li e aceito os termos de uso

Entrar

Já possuo cadastro

**Fatec**  
Faculdade de Tecnologia

Rua Vicente Leporace, 2630, Jd.  
TrianonCEP 15703-116, Jales/SP - Brasil

17 99676-2867

(17) 3621 6911

(17) 3632 2259

fatecjales@fatecjales.edu.br

**Cursos**

Agronegócio

Análise e Desenvolvimento de Sistemas

Gestão Empresarial

Sistemas para Internet

**Fatec**  
Faculdade de Tecnologia

**GPS**  
Fatec Jales

**SÃO PAULO**  
GOVERNADOR DO ESTADO

© COPYRIGHT FATEC JALES 2023

Fonte: Elaborado pelos autores.

A tela exibida na figura 22 corresponde ao formulário de login do sistema Tech Vagas, projetada para ser simples e funcional. O usuário pode acessar sua conta inserindo e-mail ou nome de usuário e senha, com a opção "Lembre de mim" para facilitar logins futuros. O design é limpo, priorizando a usabilidade, enquanto o botão "Criar conta" direciona novos usuários para o cadastro. No rodapé, informações institucionais da Fatec Jales reforçam a credibilidade, incluindo endereço, contatos e cursos oferecidos. A interface é direta, garantindo uma experiência rápida e eficiente para o usuário.

**Figura 22** — Protótipo tela de Login

**TECH**  
vagas

Ajuda

←

Entre no seu perfil

Bem-vindo(a) ao Tech Vagas

Email ou usuário

Senha

☐ Lembre de mim

Entrar

Criar conta

**Fatec**  
Faculdade de Tecnologia

Rua Vicente Leporace, 2630, Jd.  
Trionan CEP 15703-116, Jales/SP - Brasil

17 99676-2867  
(17) 3621 6911  
(17) 3632 2239  
fatecjales@fatecjales.edu.br

**Cursos**

Agronegócio  
Análise e Desenvolvimento de Sistemas  
Gestão Empresarial  
Sistemas para Internet

**Fatec**  
Faculdade de Tecnologia

**CPS**  
Centro Paula Souza

**SÃO PAULO**  
GOVERNO DO ESTADO

© COPYRIGHT FATEC\_JALES 2023

Fonte: Elaborado pelos autores.

#### 4.4.2. DISCUSSÃO DAS TELAS DO TECHVAGAS

As telas do sistema TECHVAGAS foram projetadas com foco em proporcionar uma experiência fluida e acessível, tanto para os estudantes quanto para as empresas que utilizam a plataforma. Ao desenvolver a interface, foram considerados aspectos cruciais de usabilidade, acessibilidade, componentização de interface, e padronização visual, garantindo que a navegação seja intuitiva e consistente em todos os dispositivos, desde desktops até smartphones. Cada tela foi pensada para atender aos requisitos funcionais e não funcionais do sistema, assegurando que as interações dos usuários sejam realizadas de forma eficiente e sem frustrações.

Esta tela é destinada à criação de contas de novos usuários. O layout é dividido em duas partes: o lado esquerdo exibe o logotipo do sistema e os parceiros envolvidos (Centro Paula Souza e Governo do Estado de São Paulo), enquanto o lado direito contém o formulário de cadastro. Na aba "Criar conta", o usuário pode preencher os campos obrigatórios, como nome, e-mail, CPF, senha e confirmação de senha, e clicar no botão "Criar Conta" para concluir o registro. Abaixo do formulário, há uma mensagem informando que, ao continuar, o usuário

concorda com os Termos de Uso e a Política de Privacidade, reforçando a segurança e regulamentação do sistema.

**Figura 23** — Tela de cadastro

A interface de cadastro do sistema TECH vagas apresenta uma aba ativa para "Criar conta". O formulário solicita o preenchimento de Nome, E-mail, CPF, Senha e sua confirmação. Um botão "Criar Conta" finaliza o processo. No rodapé, há logos de parceiros e uma declaração de aceitação dos termos de uso e política de privacidade.

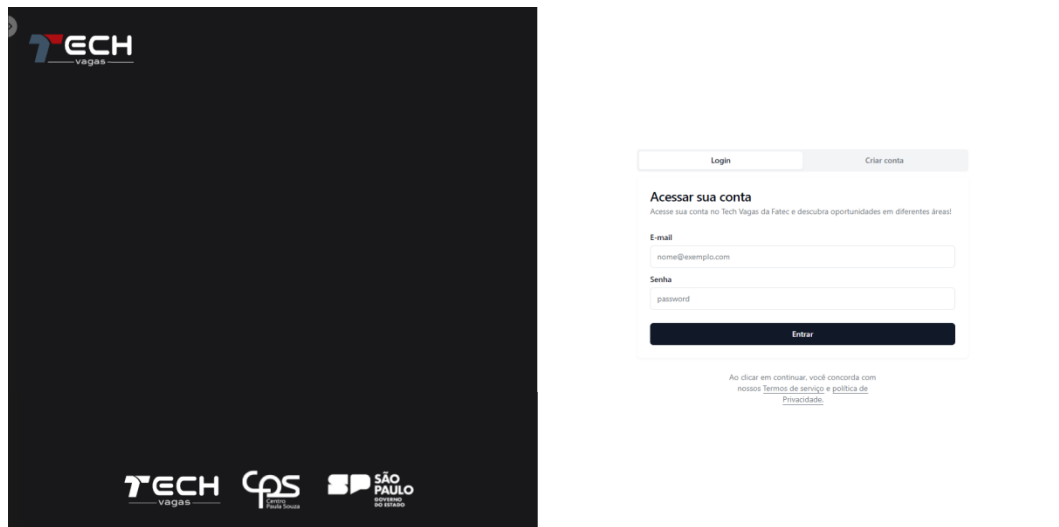
Fonte: Elaborado pelos autores.

validação em tempo real, o que ajuda a reduzir erros durante o cadastro, notificando imediatamente o usuário sobre campos incorretos ou ausentes.

**Feedback Visual:** O sistema utiliza feedback visual claro (em tons de verde para confirmações e vermelho para erros), o que melhora a experiência do usuário, permitindo correções rápidas.

**Segurança:** Durante o processo de cadastro, são utilizadas práticas de segurança como hashing de senhas, garantindo que os dados dos usuários estejam protegidos.

Esta tela é dedicada ao acesso de usuários já registrados. Aqui se apresenta o formulário de login. Na aba "Login", o usuário pode inserir seu e-mail e senha para acessar a conta, utilizando o botão "Entrar" para prosseguir. Abaixo, há um aviso informando que, ao realizar o login, o usuário aceita os Termos de Uso e a Política de Privacidade, reforçando o compromisso com a segurança e a conformidade do sistema.

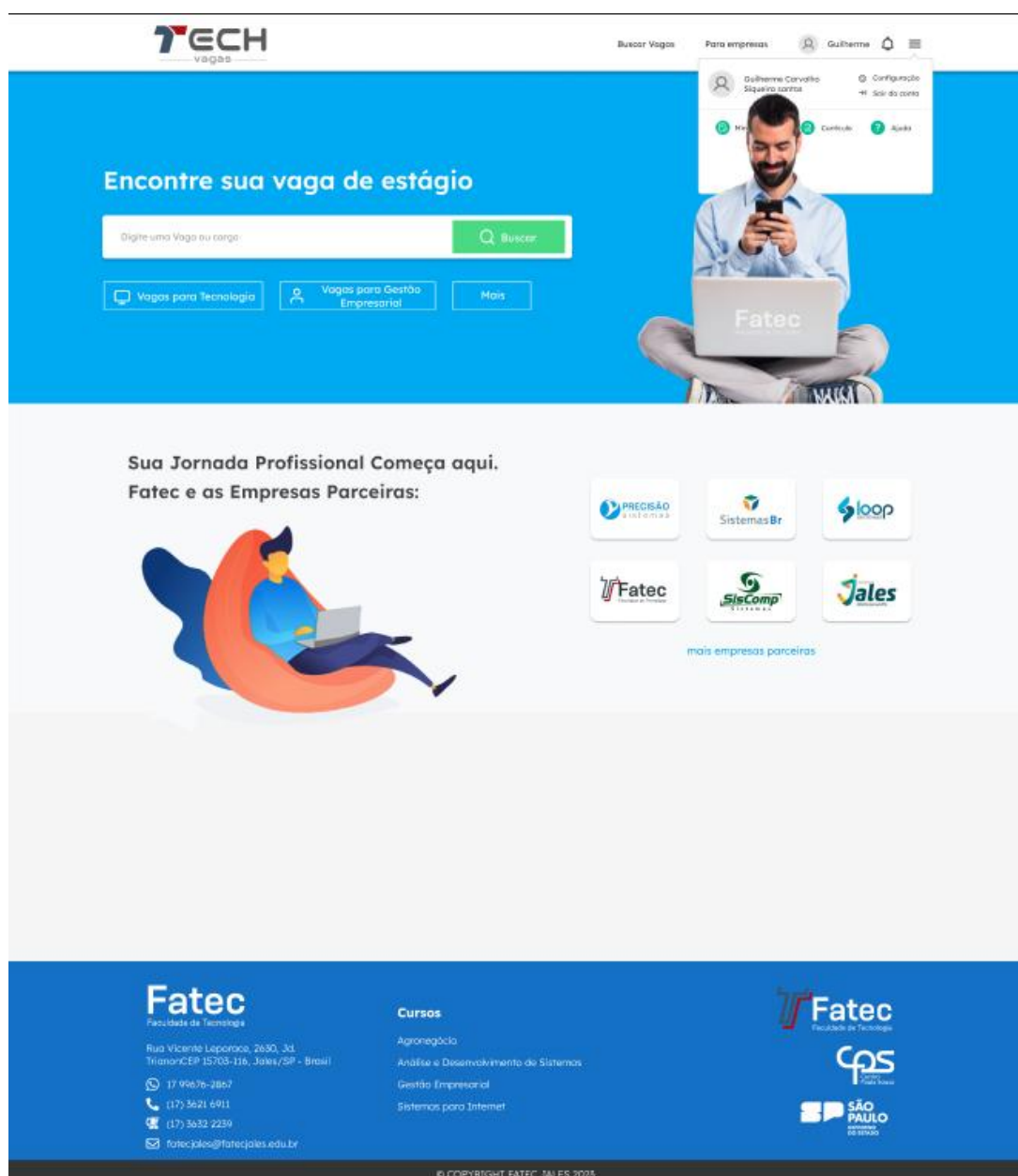
**Figura 24** — Tela de Login

Fonte: Elaborado pelos autores.

**Design Responsivo:** A interface foi desenhada para se adaptar a diferentes tamanhos de tela, o que é essencial para o uso em dispositivos móveis.

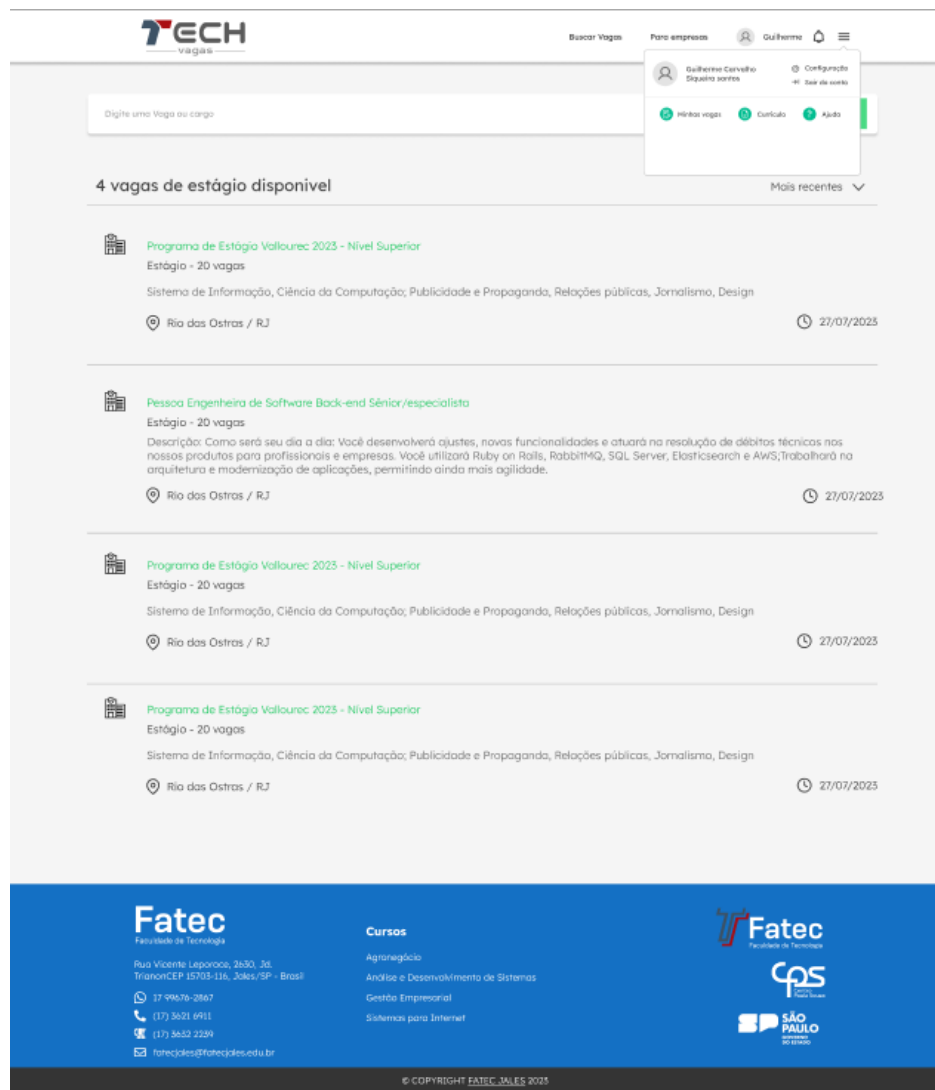
A tela inicial foi projetada com um design minimalista, destacando uma barra de busca centralizada para proporcionar uma navegação intuitiva e ágil pelas oportunidades de estágio. Além disso, inclui atalhos para as seções mais populares, como vagas recomendadas, oportunidades organizadas por categorias (por exemplo, Tecnologia, Gestão Empresarial) e empresas parceiras, facilitando o acesso direto às informações mais relevantes para os usuários.

**Figura 25** — Tela Inicial



Fonte: Elaborado pelos autores.

A tela de Vagas Cadastradas oferece uma interface prática e funcional para que tanto empresas quanto administradores possam visualizar e gerenciar as vagas disponíveis na plataforma. Com um layout eficiente, as oportunidades são apresentadas em formato tabular, destacando informações essenciais como título da vaga, descrição, requisitos e status (ativa, fechada, em processo). A plataforma também conta com filtros de busca que permitem refinar a pesquisa com base em critérios como data de publicação, área de atuação e número de candidatos inscritos.

**Figura 26** — Tela de Vagas Cadastradas

Fonte: Elaborado pelos autores.

**Interatividade:** A interação é dinâmica, com recursos de edição in-line que permitem a atualização rápida de informações, como a alteração da descrição ou requisitos das vagas, sem precisar recarregar a página.

**Gestão de Vagas:** A interface também facilita ações rápidas, como ativar ou desativar vagas, além de oferecer a possibilidade de clonar uma vaga para publicação futura com pequenas modificações.

## 5 BANCO DE DADOS

Um banco de dados é um sistema projetado para armazenar e organizar informações de maneira que facilite o acesso e a manipulação dos dados. Ele permite que os usuários gerenciem grandes volumes de informações de forma estruturada, possibilitando a realização de consultas, relatórios e análises, que são fundamentais em diversos setores. Alguns exemplos

incluem: comércio, saúde e pesquisas. Existem variados tipos de bancos de dados, como os relacionais e não relacionais, sendo que cada um possui características e aplicações distintas.

Segundo o autor C. J. Date e com base no livro *An Introduction to Database Systems* "Um banco de dados é um conjunto de dados que é armazenado de maneira organizada e que pode ser acessado e manipulado com eficiência. A escolha do modelo de dados e das estruturas de armazenamento adequadas é crucial para o desempenho e a integridade do sistema" (DATE, 2004).

Um banco de dados é uma estrutura essencial para a organização e a gestão de informações, permitindo que dados sejam armazenados de forma eficiente e acessados rapidamente. Segundo Elmasri e Navathe, "um sistema de banco de dados é um conjunto de programas que permitem a definição, construção e manipulação de bancos de dados para atender a uma variedade de aplicações" (ELMASRI, NAVATHE, 2016). A capacidade de um banco de dados de lidar com grandes volumes de dados e garantir integridade e segurança é crucial em ambientes corporativos e acadêmicos, onde decisões importantes dependem da precisão e da acessibilidade das informações. Assim, o design e a implementação eficazes de bancos de dados se tornam uma prioridade, impactando diretamente o sucesso de projetos em diversas áreas.

## 5.1 MODELO ENTIDADE RELACIONAMENTO

O Modelo de Entidade-Relacionamento (MER) é uma técnica fundamental no design de bancos de dados que visa representar graficamente as entidades e os relacionamentos entre elas. Ele permite que os analistas identifiquem e organizem os dados de maneira que reflitam as necessidades do negócio, facilitando a comunicação entre as partes interessadas. Como destaca (MARTIN, 1988) "O MER proporciona uma visão clara e concisa da estrutura do banco de dados, ajudando a garantir que todos os aspectos dos requisitos de informação sejam considerados"

Reforçando essa visão, Ramez Elmasri e Shamkant Navathe enfatizam que "O modelo de entidade-relacionamento é essencial para a representação do mundo real em um banco de dados, permitindo uma modelagem lógica que é tanto intuitiva quanto eficiente" (ELMASRI, NAVATHE, 2016). Essa representação visual não apenas simplifica a compreensão dos dados,

mas também orienta o desenvolvimento de um esquema que assegura a integridade e a coerência dos dados ao longo do tempo.

Com base na Figura 27, apresenta-se um diagrama de entidade-relacionamento (DER), que ilustra a estrutura de um banco de dados composto por várias tabelas interligadas. Cada retângulo no diagrama representa uma tabela, e cada linha dentro desses retângulos corresponde a um campo ou coluna da tabela. Esses campos armazenam diferentes tipos de dados, como números inteiros (integer), textos (character varying), valores booleanos (true/false) ou datas e horas (timestamp).

As chaves primárias (PK) são indicadas por ícones de chave e servem para identificar de maneira única cada registro dentro de uma tabela. Já as chaves estrangeiras (FK) são utilizadas para criar o relacionamento entre tabelas, conectando uma chave primária de uma tabela à outra.

As setas que ligam as tabelas indicam esses relacionamentos, mostrando como os dados de uma tabela se relacionam com os dados de outra, geralmente em uma estrutura de um-para-muitos. Isso significa que uma entrada em uma tabela pode estar relacionada a várias entradas em outra tabela, o que é comum em sistemas que lidam com grandes volumes de dados interconectados.

Dessa forma, o diagrama facilita a visualização e o entendimento da organização dos dados e suas interdependências no sistema.





o funcionamento adequado do sistema. Eles facilitam a integração entre a camada de aplicação e o banco de dados, assegurando que as operações realizadas pela aplicação sejam refletidas corretamente no banco de dados.

**Quadro 12** — Script SQL – Tabela Aluno

```
CREATE TABLE aluno (
  alunoid integer GENERATED BY DEFAULT AS IDENTITY,
  nome character varying(150) NOT NULL,
  idade integer NOT NULL,
  rg character varying(12) NOT NULL,
  statusaluno boolean NOT NULL,
  numeromatrícula character varying(50) NOT NULL,
  areainteresse character varying(100) NOT NULL,
  habilidades character varying(100) NOT NULL,
  experiencias character varying(350) NOT NULL,
  disponibilidadehorario character varying(35) NOT NULL,
  curriculo text NOT NULL,
  cpf character varying(14) NOT NULL,
  cidade character varying(50) NOT NULL,
  datanascimento timestamp with time zone NOT NULL,
  nivelescolaridade character varying(80) NOT NULL,
  telefone character varying(14) NOT NULL,
  email character varying(50) NOT NULL,
  endereco character varying(50) NOT NULL,
  genero character varying(30) NOT NULL,
  bairro character varying(30) NOT NULL,
  cep character varying(9) NOT NULL,
  "AlunoModelAlunoId" integer NULL,
  CONSTRAINT "PK_aluno" PRIMARY KEY (alunoid),
  CONSTRAINT "FK_aluno_aluno_AlunoModelAlunoId" FOREIGN KEY
  ("AlunoModelAlunoId") REFERENCES aluno (alunoid)
);
```

Fonte: Elaborado pelos autores.

**Quadro 13** — Script SQL – Tabela Cargo

```
CREATE TABLE cargo (
  cargoid integer GENERATED BY DEFAULT AS IDENTITY,
  descricao character varying(200) NOT NULL,
  tipo character varying(50) NOT NULL,
  CONSTRAINT "PK_cargo" PRIMARY KEY (cargoid)
);
```

Fonte: Elaborado pelos autores.

**Quadro 14** — Script SQL – Tabela Concedente

```
CREATE TABLE concedente (
  concedenteid integer GENERATED BY DEFAULT AS IDENTITY,
```

```

razaosocial character varying(80) NOT NULL,
responsavelestagio character varying(50) NOT NULL,
cnpj character varying(16) NOT NULL,
localidade character varying(50) NOT NULL,
CONSTRAINT "PK_concedente" PRIMARY KEY (concedenteid)
);

```

Fonte: Elaborado pelos autores.

#### **Quadro 15** — Script SQL – Tabela Coordenador de Estágio

```

CREATE TABLE coordenadorestagio (
    coordenadorestagioid integer GENERATED BY DEFAULT AS IDENTITY,
    datacadastro date NOT NULL,
    nomecoordenador text NOT NULL,
    statuscoordenadorestagio boolean NOT NULL,
    CONSTRAINT "PK_coordenadorestagio" PRIMARY KEY (coordenadorestagioid)
);

```

Fonte: Elaborado pelos autores.

#### **Quadro 16** — Script SQL – Tabela Curso

```

CREATE TABLE curso (
    cursoid integer GENERATED BY DEFAULT AS IDENTITY,
    nomecurso character varying(150) NOT NULL,
    turnocurso character varying(100) NOT NULL,
    CONSTRAINT "PK_curso" PRIMARY KEY (cursoid)
);

```

Fonte: Elaborado pelos autores.

#### **Quadro 17** — Script SQL – Tabela Instituição de Ensino

```

CREATE TABLE instituicaoensino (
    id integer GENERATED BY DEFAULT AS IDENTITY,
    nomeinstituicao character varying(200) NOT NULL,
    local character varying(120) NOT NULL,
    telefone character varying(17) NOT NULL,
    CONSTRAINT "PK_instituicaoensino" PRIMARY KEY (id)
);

```

Fonte: Elaborado pelos autores.

#### **Quadro 18** — Script SQL – Tabela Login

```

CREATE TABLE login (
    email text NOT NULL,
    senha text NOT NULL
);

```

Fonte: Elaborado pelos autores.

**Quadro 19** — Script SQL – Tabela Tipo documento

```
CREATE TABLE tipodocumento (
  tipodocumentoid integer GENERATED BY DEFAULT AS IDENTITY,
  descricao character varying(200) NOT NULL,
  statutipodocumento boolean NOT NULL,
  CONSTRAINT "PK_tipodocumento" PRIMARY KEY (tipodocumentoid)
);
```

Fonte: Elaborado pelos autores.

**Quadro 20** — Script SQL – Tabela Tipo estágio

```
CREATE TABLE tipoestagio (
  tipoestagioid integer GENERATED BY DEFAULT AS IDENTITY,
  descricao character varying(200) NOT NULL,
  CONSTRAINT "PK_tipoestagio" PRIMARY KEY (tipoestagioid)
);
```

Fonte: Elaborado pelos autores.

**Quadro 21** — Script SQL – Tabela Usuário

```
CREATE TABLE usuario (
  usuarioid integer GENERATED BY DEFAULT AS IDENTITY,
  nome text NOT NULL,
  cpfcpnpjessoa text NOT NULL,
  email text NOT NULL,
  senha text NOT NULL,
  usertype integer NOT NULL,
  CONSTRAINT "PK_usuario" PRIMARY KEY (usuarioid)
);
```

Fonte: Elaborado pelos autores.

**Quadro 22** — Script SQL – Tabela Supervisor Estágio

```
CREATE TABLE supervisorestagio (
  supervisorid integer GENERATED BY DEFAULT AS IDENTITY,
  nomesupervisor text NOT NULL,
  statussupervisorestagio boolean NOT NULL,
  concedenteid integer NOT NULL,
  "ConcedenteModelconcedenteId" integer NULL,
  CONSTRAINT "PK_supervisorestagio" PRIMARY KEY (supervisorid),
  CONSTRAINT "FK_supervisorestagio_concedente_ConcedenteModelconcedenteId"
FOREIGN KEY ("ConcedenteModelconcedenteId") REFERENCES concedente
(concedenteid),
  CONSTRAINT "FK_supervisorestagio_concedente_concedenteid" FOREIGN KEY
(concedenteid) REFERENCES concedente (concedenteid) ON DELETE CASCADE
);
```

Fonte: Elaborado pelos autores.

**Quadro 23** — Script SQL – Tabela Vagas

```
CREATE TABLE vagas (
  vagasid integer GENERATED BY DEFAULT AS IDENTITY,
  quantidade text NOT NULL,
  datapublicacao date NOT NULL,
  datalimite date NOT NULL,
  localidade character varying(80) NOT NULL,
  descricao character varying(800) NOT NULL,
  titulo character varying(80) NOT NULL,
  localidadetrabalho character varying(20) NOT NULL,
  horarioentrada character varying(20) NOT NULL,
  horariosaida character varying(20) NOT NULL,
  totalhorassemanais character varying(6) NOT NULL,
  "concedenteId" integer NOT NULL,
  "CargoId" integer NOT NULL,
  CONSTRAINT "PK_vagas" PRIMARY KEY (vagasid),
  CONSTRAINT "FK_vagas_cargo_CargoId" FOREIGN KEY ("CargoId")
REFERENCES cargo (cargoid) ON DELETE RESTRICT,
  CONSTRAINT "FK_vagas_concedente_concedenteId" FOREIGN KEY
("concedenteId") REFERENCES concedente (concedenteid) ON DELETE CASCADE
);
```

Fonte: Elaborado pelos autores.

**Quadro 24** — Script SQL – Tabela Apontamento

```
CREATE TABLE apontamento (
  apontamentoid integer GENERATED BY DEFAULT AS IDENTITY,
  "descricaoApontamento" character varying(150) NOT NULL,
  "dataApontamento" date NOT NULL,
  coordenadorestagioid integer NOT NULL,
  "CoordenadorEstagioModelidCoordenadorEstagio" integer NULL,
  CONSTRAINT "PK_apontamento" PRIMARY KEY (apontamentoid),
  CONSTRAINT
"FK_apontamento_coordenadorestagio_CoordenadorEstagioModelidCoo~" FOREIGN
KEY ("CoordenadorEstagioModelidCoordenadorEstagio") REFERENCES
coordenadorestagio (coordenadorestagioid),
  CONSTRAINT "FK_apontamento_coordenadorestagio_coordenadorestagioid"
FOREIGN KEY (coordenadorestagioid) REFERENCES coordenadorestagio
(coordenadorestagioid) ON DELETE CASCADE
);
```

Fonte: Elaborado pelos autores.

**Quadro 25** — Script SQL – Tabela matricula

```
CREATE TABLE matricula (
  matriculaid integer GENERATED BY DEFAULT AS IDENTITY,
  numeromatricula character varying(15) NOT NULL,
  "AlunosAlunoId" integer NULL,
  alunoid integer NOT NULL,
  cursoid integer NOT NULL,
```

```

CONSTRAINT "PK_matricula" PRIMARY KEY (matriculaid),
CONSTRAINT "FK_matricula_aluno_AlunosAlunoId" FOREIGN KEY
("AlunosAlunoId") REFERENCES aluno (alunoid),
CONSTRAINT "FK_matricula_curso_cursoId" FOREIGN KEY (cursoId)
REFERENCES curso (cursoId) ON DELETE CASCADE
);
CONSTRAINT "FK_matricula_curso_cursoId" FOREIGN KEY (cursoId)
REFERENCES curso (cursoId) ON DELETE CASCADE
);

```

Fonte: Elaborado pelos autores.

#### Quadro 26 — Script SQL – Tabela Documento

```

CREATE TABLE documento (
documentoid integer GENERATED BY DEFAULT AS IDENTITY,
descricao character varying(200) NOT NULL,
situacao character varying(200) NOT NULL,
tipodocumentoid integer NOT NULL,
coordenadorestagioId integer NOT NULL,
"CoordenadorEstagioModelidCoordenadorEstagio" integer NULL,
"TipoDocumentoModelidTipoDocumento" integer NULL,
CONSTRAINT "PK_documento" PRIMARY KEY (documentoid),
CONSTRAINT
"FK_documento_coordenadorestagio_CoordenadorEstagioModelidCoord~" FOREIGN
KEY ("CoordenadorEstagioModelidCoordenadorEstagio") REFERENCES
coordenadorestagio (coordenadorestagioId),
CONSTRAINT "FK_documento_coordenadorestagio_coordenadorestagioId" FOREIGN
KEY (coordenadorestagioId) REFERENCES coordenadorestagio (coordenadorestagioId)
ON DELETE CASCADE,
CONSTRAINT
"FK_documento_tipodocumento_TipoDocumentoModelidTipoDocumento" FOREIGN
KEY ("TipoDocumentoModelidTipoDocumento") REFERENCES tipodocumento
(tipodocumentoid),
CONSTRAINT "FK_documento_tipodocumento_tipodocumentoid" FOREIGN KEY
(tipodocumentoid) REFERENCES tipodocumento (tipodocumentoid) ON DELETE
CASCADE
);

```

Fonte: Elaborado pelos autores.

#### Quadro 27 — Script SQL – Tabela Documento Necessário

```

CREATE TABLE documentonecessario (
documentonecessarioId integer GENERATED BY DEFAULT AS IDENTITY,
tipodocumentoid integer NOT NULL,
tipoestagioId integer NOT NULL,
"TipoDocumentoModelidTipoDocumento" integer NULL,
"TipoEstagioModelidTipoEstagio" integer NULL,
CONSTRAINT "PK_documentonecessario" PRIMARY KEY (documentonecessarioId),
CONSTRAINT
"FK_documentonecessario_tipodocumento_TipoDocumentoModelidTipoD~" FOREIGN
KEY ("TipoDocumentoModelidTipoDocumento") REFERENCES tipodocumento
(tipodocumentoid),

```

```

    CONSTRAINT "FK_documentonecessario_tipodocumento_tipodocumentoid"
    FOREIGN KEY (tipodocumentoid) REFERENCES tipodocumento (tipodocumentoid) ON
    DELETE CASCADE,
    CONSTRAINT
    "FK_documentonecessario_tipoestagio_TipoEstagioModelidTipoEstag~" FOREIGN KEY
    ("TipoEstagioModelidTipoEstagio") REFERENCES tipoestagio (tipoestagioid),
    CONSTRAINT "FK_documentonecessario_tipoestagio_tipoestagioid" FOREIGN KEY
    (tipoestagioid) REFERENCES tipoestagio (tipoestagioid) ON DELETE CASCADE
);

```

Fonte: Elaborado pelos autores.

### Quadro 28 — Script SQL – Tabela Sessão

```

CREATE TABLE sessao (
    idsessao integer GENERATED BY DEFAULT AS IDENTITY,
    datahoraabertura text NOT NULL,
    datahorafechamento text NULL,
    tokensessao text NOT NULL,
    statussessao boolean NOT NULL,
    emailpessoa text NOT NULL,
    nivelacesso text NOT NULL,
    "UsuarioId" integer NOT NULL,
    CONSTRAINT "PK_sessao" PRIMARY KEY (idsessao),
    CONSTRAINT "FK_sessao_usuario_UsuarioId" FOREIGN KEY ("UsuarioId")
    REFERENCES usuario (usuarioid) ON DELETE CASCADE
);

```

Fonte: Elaborado pelos autores.

### Quadro 29 — Script SQL – Tabela Contrato Estágio

```

CREATE TABLE contratoestagio (
    "ContratoEstagioid" integer GENERATED BY DEFAULT AS IDENTITY,
    "Status do ContratoEstagio" boolean NOT NULL,
    notafinal character varying(150) NOT NULL,
    situacao character varying(150) NOT NULL,
    "Horario de Entrada" character varying(150) NOT NULL,
    "Horario de Saida" character varying(150) NOT NULL,
    datainicio date NOT NULL,
    datafim date NOT NULL,
    salario character varying(150) NOT NULL,
    cargasemanal character varying(150) NOT NULL,
    cargatotal character varying(150) NOT NULL,
    "CoordenadorEstagioidCoordenadorEstagio" integer NULL,
    coordenadorestagiod integer NOT NULL,
    supervisorestagiod integer NOT NULL,
    "TipoEstagioidTipoEstagio" integer NULL,
    tipoestagioid integer NOT NULL,
    "SupervisorEstagioModelidSupervisor" integer NULL,
    CONSTRAINT "PK_contratoestagio" PRIMARY KEY ("ContratoEstagioid"),

```

```

CONSTRAINT "FK_contratoestagio_coordenadorestagio_CoordenadorEstagioidCoor~"
FOREIGN KEY ("CoordenadorEstagioidCoordenadorEstagio") REFERENCES
coordenadorestagio (coordenadorestagiod),
CONSTRAINT "FK_contratoestagio_supervisorestagio_SupervisorEstagioModelidS~"
FOREIGN KEY ("SupervisorEstagioModelidSupervisor") REFERENCES
supervisorestagio (supervisorid),
CONSTRAINT "FK_contratoestagio_supervisorestagio_supervisorestagiod" FOREIGN
KEY (supervisorestagiod) REFERENCES supervisorestagio (supervisorid) ON DELETE
CASCADE,
CONSTRAINT "FK_contratoestagio_tipoestagio_TipoEstagioidTipoEstagio" FOREIGN
KEY ("TipoEstagioidTipoEstagio") REFERENCES tipoestagio (tipoestagioid)
);

```

Fonte: Elaborado pelos autores.

### Quadro 30 — Script SQL – Tabela Documento Versão

```

CREATE TABLE documentoversao (
    documentoversaoid integer GENERATED BY DEFAULT AS IDENTITY,
    comentario character varying(200) NOT NULL,
    anexo character varying(200) NOT NULL,
    data date NOT NULL,
    situacao character varying(200) NOT NULL,
    "DocumentoidDocumento" integer NULL,
    documentoid integer NOT NULL,
    CONSTRAINT "PK_documentoversao" PRIMARY KEY (documentoversaoid),
    CONSTRAINT "FK_documentoversao_documento_DocumentoidDocumento"
FOREIGN KEY ("DocumentoidDocumento") REFERENCES documento (documentoid)
);

```

Fonte: Elaborado pelos autores.

## 5.3 MAPEAMENTO OBJETO RELACIONAL – ORM

O Mapeamento Objeto-Relacional (ORM) é uma técnica que facilita a interação entre aplicações orientadas a objetos e bancos de dados relacionais. Essa abordagem permite o mapeamento automático de classes e objetos do código para tabelas e registros no banco de dados, dispensando a necessidade de escrever consultas SQL manualmente. Como resultado, o ORM simplifica o desenvolvimento, pois mantém a manipulação de dados dentro do paradigma orientado a objetos, o que torna o código mais fácil de manter e compreender (Macoratti, 2019).

No contexto da linguagem C#, o Entity Framework é uma das ferramentas mais importantes para a implementação de ORM. Ele se comunica diretamente entre o código C# e o banco de dados, manipulando as operações de inserção, leitura, atualização e exclusão de dados, sem que o desenvolvedor precise interagir manualmente com o banco de dados SQL. No Entity Framework, o mapeamento entre classes e tabelas é realizado automaticamente, seja por convenções ou por meio de configurações personalizadas. Isso permite ao programador



manipular dados através de classes C# e utilizar funcionalidades como validação, definição de relações entre entidades e controle de transações, o que simplifica o processo de persistência de dados e reduz a complexidade do código (Macoratti, 2019).

## 6 ARQUITETURA DE SOFTWARE

A arquitetura de software é crucial para o desenvolvimento de sistemas complexos, pois engloba as decisões que determinam a estrutura, o comportamento e a qualidade do sistema ao longo de sua existência. Conforme afirmam Bass, Clements e Kazman (2003), a arquitetura não se limita apenas aos componentes que formam o sistema, mas também define a interação entre esses componentes e os padrões de design que moldam a solução proposta.

Dois elementos essenciais da arquitetura são os componentes e os conectores. Os componentes, como classes, bibliotecas e subsistemas, são responsáveis por realizar funções específicas dentro do sistema. Eles se comunicam através de conectores, que podem ser APIs, chamadas de métodos ou mensagens. Shaw e Garlan (1996) destacam que a escolha dos conectores adequados é fundamental para garantir uma comunicação eficiente entre os componentes, o que influencia diretamente na flexibilidade e escalabilidade do sistema.

Outro aspecto importante da arquitetura são os padrões arquiteturais, que oferecem soluções testadas para problemas recorrentes no design de software. Gamma et al. (1995), em *Design Patterns: Elements of Reusable Object-Oriented Software*, explicam como esses padrões organizam a estrutura interna dos componentes. Um exemplo disso é o padrão Model-View-Controller (MVC), que separa a interface do usuário da lógica de negócios, tornando o sistema mais fácil de manter e atualizar. Outro padrão relevante é o Microservices (Fowler, 2014), que permite dividir o sistema em serviços independentes, otimizando a escalabilidade e a manutenção.

Os estilos arquiteturais fornecem abordagens gerais para o design do sistema. A Arquitetura em Camadas, descrita por Buschmann et al. (1996) em *Pattern-Oriented Software Architecture*, organiza o sistema em diferentes camadas, o que facilita a separação de responsabilidades e melhora a organização do código. A Arquitetura de Microserviços, mais recente, propõe uma estrutura distribuída onde os serviços são independentes, permitindo maior flexibilidade e resiliência do sistema.

A qualidade do software é outro aspecto vital. Segundo Pressman e Maxim (2015), uma arquitetura bem projetada deve atender a critérios de qualidade, como desempenho, segurança e facilidade de manutenção. Esses requisitos são essenciais para garantir que o sistema seja eficaz e sustentável ao longo do tempo, mesmo com mudanças e expansões.

Por fim, a documentação arquitetural é um elemento essencial para garantir que as decisões de design sejam compreendidas e seguidas ao longo do ciclo de vida do sistema. Kruchten (1995) e Clements et al. (2002) afirmam que uma boa documentação serve como um guia para desenvolvedores, equipes de manutenção e stakeholders, ajudando a manter a consistência e a continuidade do projeto à medida que o sistema evolui.

## 6.1 ARQUITETURA DE DESENVOLVIMENTO

A arquitetura de desenvolvimento do TechVagas é responsável por definir a estrutura e a organização do código, além de estabelecer as interações entre os diversos componentes do sistema. Como o TechVagas é uma aplicação baseada em Web API, com separação entre o *back-end* e o *front-end*, essa arquitetura é fundamental para garantir modularidade e facilitar a manutenção do sistema. De acordo com Pressman (2015), a arquitetura de desenvolvimento é crucial para o sucesso de um software, pois “tomar boas decisões durante a definição da arquitetura é essencial para o sucesso de um produto de software. A arquitetura define a estrutura do sistema e determina a sua qualidade”.

Uma Web API (*Application Programming Interface*) é um conjunto de definições e protocolos usados para criar e integrar software de aplicações. Segundo Microsoft (2024):

Web API facilita a criação de serviços que alcancem uma ampla variedade de clientes, incluindo navegadores e dispositivos móveis. [...] Você usa o mesmo framework e os mesmos padrões para criar páginas da web e serviços lado a lado no mesmo projeto.

Cada parte do projeto desempenha um papel específico no fluxo de dados e na regra de negócio. A estrutura do back-end é desenvolvida em C# e é composta por várias camadas, cada uma responsável por uma parte distinta da aplicação. Já no front-end, o TechVagas foi desenvolvido utilizando React, uma biblioteca baseada em JavaScript muito popular para construção de interfaces de usuário dinâmicas e responsivas e utilizamos uma biblioteca de componentes de interface de usuário chamado de ShadCN UI, criada com o Tailwind CSS, voltada para o desenvolvimento de aplicações web. Ela oferece uma coleção de componentes prontos, como botões, tabelas e formulários, que podem ser facilmente copiados e integrados a projetos.

### 6.1.1 BACK-END

A aplicação de backend tem a função de gerenciar as principais funcionalidades do sistema. Quando o usuário faz uma interação com a interface, o servidor processa essa solicitação e envia uma resposta para a aplicação em formato JSON. Esse componente foi criado utilizando a linguagem C# (Microsoft, 2024), enquanto o controle dos dados é feito com o PostgreSQL, um banco de dados relacional de código aberto, muito usado por ser robusto e escalável.

Durante o desenvolvimento do projeto, ficou claro que uma arquitetura em camadas seria importante para organizar e gerenciar as funcionalidades de maneira mais eficiente. A estrutura escolhida tem várias camadas, cada uma com uma responsabilidade específica, o que ajuda a manter o sistema mais organizado e facilita sua manutenção.

A camada DTO (Data Transfer Object), como explica Lima (2023), é um modelo usado para transferir dados entre diferentes partes da aplicação, como o backend e o frontend. Ela também tem a tarefa de validar os dados antes de passá-los para outras partes do sistema.

A camada Controller, segundo Silvestre (2022), é responsável por receber as solicitações dos usuários e encaminhá-las para as camadas seguintes, acionando os métodos necessários conforme o fluxo da aplicação.

A camada Service, como aponta Lima (2023), cuida da lógica de negócios do sistema, incluindo a validação e o controle de acesso aos dados. Ela também se comunica diretamente com a camada Model, que define as classes e estruturas de dados no sistema.

Por fim, a camada Repository, de acordo com Barbosa (2021), gerencia o acesso aos dados, validando as informações necessárias antes de passá-las para a camada Service e garantindo uma comunicação eficiente para processar e consultar os dados.

### 6.1.2 FRONT-END - WEB

O front-end é a parte do desenvolvimento de uma solução, cuja principais responsabilidades são: se comunicar com o servidor e oferecer interações com o usuário através de uma interface. “O front-end é a interface do usuário — a parte de uma aplicação web que o usuário vê e interage. O front-end é essencial porque fornece a apresentação visual dos dados, garantindo que os usuários possam interagir com a aplicação de forma fácil e eficaz” (MELONI, 2018).

O front-end foi desenvolvido utilizando o React, uma biblioteca JavaScript para criação de interfaces de usuário. Esta biblioteca permite a construção de componentes que

gerenciam seus próprios estados e se combinam para formar telas complexas (REACT, 2024). Como interface gráfica da aplicação, o front-end reúne elementos como menus, botões, imagens e layout, com o objetivo de proporcionar uma experiência fluída e intuitiva ao usuário, facilitando sua navegação e interação com a aplicação (TOTVS, 2021).

Além de facilitar a criação de interfaces complexas, o React contribui para uma organização de código mais modular e escalável. Cada componente é independente e atualizado de forma isolada, permitindo uma manutenção mais ágil e reduzindo o risco de falhas ao realizar alterações em partes específicas do código. Essa modularidade facilita o desenvolvimento incremental, em que novos recursos podem ser adicionados de forma contínua, sem impacto negativo no restante do sistema. Ademais, o React é amplamente suportado pela comunidade de desenvolvedores, com uma documentação abrangente de funcionalidades específicas, o que aumenta ainda mais a eficiência no desenvolvimento e na escalabilidade do projeto (REACT, 2024).

Para enriquecer ainda mais a experiência visual e simplificar o design, a biblioteca ShadCN UI foi adotada. Ela oferece componentes estilizados com Tailwind CSS, como botões, tabelas e formulários, que são facilmente personalizáveis e responsivos, facilitando a criação de uma interface consistente e atraente para os usuários. Com essa combinação, o front-end torna-se modular, escalável e esteticamente agradável, proporcionando uma navegação intuitiva e eficiente (REACT, 2024; TOTVS, 2021).

## 6.2 SEGURANÇA DA INFORMAÇÃO

A segurança da informação é essencial para proteger os dados contra acessos não autorizados, vazamentos e alterações. Ela se baseia em três princípios fundamentais: confidencialidade, integridade e disponibilidade (CIA). A confidencialidade garante que apenas usuários autorizados acessem informações sensíveis, a integridade assegura que os dados não sejam alterados sem permissão, e a disponibilidade assegura que as informações estejam acessíveis quando necessário (Anderson, 2021).

O JWT (JSON Web Token) é amplamente utilizado para autenticação e autorização de usuários, permitindo a troca segura de informações entre cliente e servidor. Ele usa criptografia para garantir a autenticidade e a integridade dos dados, dispensando a necessidade de armazenar sessões no servidor e tornando o sistema mais escalável (Bradley, 2020).

A integridade dos dados também é protegida por algoritmos de hash como o SHA-256, que verificam se as informações foram alteradas durante a transmissão. Esse algoritmo é

frequentemente combinado com o JWT para garantir que os tokens não sejam modificados (IBM, 2021).

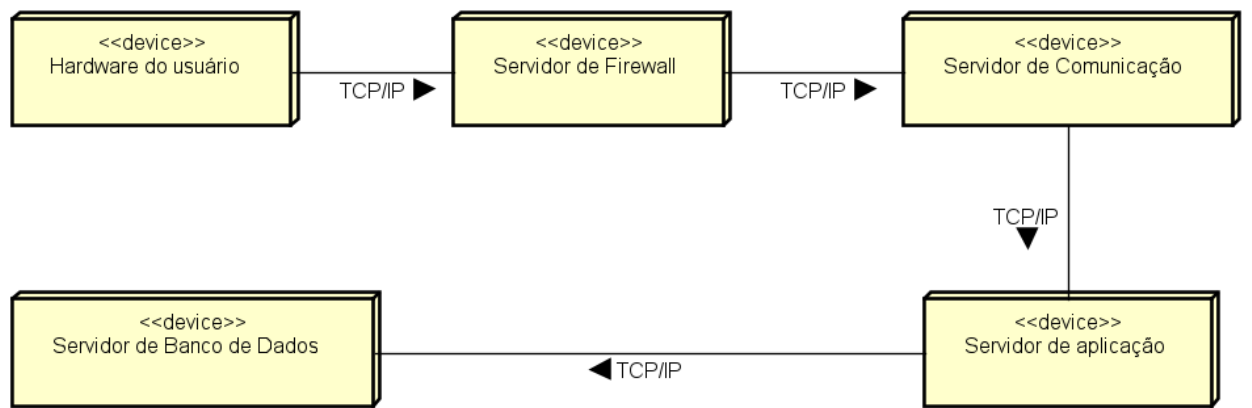
A auditoria das operações do sistema é fundamental para detectar atividades suspeitas, registrando logs detalhados sobre as ações dos usuários, como aponta Schneier (2015). Esses logs são importantes tanto para identificar falhas de segurança quanto para garantir conformidade com regulamentos como o GDPR.

No desenvolvimento de sistemas, a segurança na comunicação entre o frontend e o backend é garantida pelo uso de HTTPS e protocolos como o TLS, que protegem dados sensíveis durante a troca de informações. O uso de JWT sobre uma conexão segura assegura a confidencialidade da comunicação (Williams, 2018).

Em resumo, a segurança da informação é crucial no desenvolvimento de sistemas, envolvendo o uso de tecnologias como JWT, auditorias rigorosas e a proteção das comunicações entre o frontend e o backend, garantindo sempre a proteção dos dados e a conformidade com os requisitos legais.

### 6.3 IMPLANTAÇÃO

O diagrama de implantação é uma representação visual fundamental para o planejamento e a organização da infraestrutura de hardware necessária à implementação do software. Ele detalha os equipamentos que irão suportar o sistema, como servidores e computadores, e define as configurações de conexão e os protocolos necessários para a comunicação e transferência de dados entre esses dispositivos. Além disso, esse diagrama possibilita a visualização da distribuição dos módulos em servidores distintos, o que é particularmente relevante para sistemas distribuídos. Sua utilização é essencial em ambientes que exigem múltiplas máquinas para a execução de módulos ou o armazenamento de dados críticos (GUEDES, 2018). A imagem a seguir apresenta o planejamento da infraestrutura de hardware, destacando a alocação dos módulos do sistema e a comunicação entre os servidores e dispositivos que compõem o ambiente.

**Figura 28** — Diagrama de Implantação

Fonte: Elaborado pelos autores.

O Docker é uma plataforma de virtualização eficiente que possibilita o empacotamento, a distribuição e a execução de aplicações dentro de contêineres, criando ambientes uniformes e padronizados para as fases de desenvolvimento, teste e produção. Cada contêiner encapsula a aplicação e todas as suas dependências, como bibliotecas e configurações, garantindo uma operação estável e consistente em diferentes sistemas operacionais. Esse modelo reduz problemas de compatibilidade e otimiza a utilização dos recursos do sistema, proporcionando maior flexibilidade e adaptabilidade.

A plataforma facilita a instalação e o escalonamento de aplicativos, seja na nuvem ou em servidores locais, sendo essencial para equipes de desenvolvimento que buscam agilidade e segurança na entrega de software (DOCKER, 2024).

A implementação de sistemas distribuídos requer o uso de ferramentas que simplifiquem a configuração, automação e administração de múltiplas aplicações de maneira eficaz e segura. Nesse contexto, o Docker se destaca como uma plataforma robusta para criação e gestão de contêineres — unidades de software leves e autossuficientes que contêm o código e suas dependências. Com o Docker, é possível estabelecer ambientes de desenvolvimento e produção que garantem uniformidade, autonomia de infraestrutura e compatibilidade entre diferentes ambientes, como estações de trabalho locais e servidores na nuvem.

Este estudo descreve a implementação de um sistema integrado utilizando Docker para conectar o front-end, o back-end e o banco de dados, destacando as configurações essenciais para garantir o funcionamento unificado e eficiente do sistema.

Para configurar o ambiente de back-end no Docker, é necessário criar um arquivo denominado Dockerfile, que deve ser colocado no mesmo diretório onde se encontra o arquivo .sln do projeto. Esse Dockerfile conterá as diretrizes essenciais para a construção do contêiner

responsável pela execução da aplicação. Em seguida, a aplicação e suas dependências devem ser compiladas utilizando o comando `dotnet publish -c Release`, que gera o build e prepara o projeto para execução.

No diretório do front-end, é necessário criar um arquivo `.env` com a URL da API, seguindo o formato `VITE_API_URL=https://localhost:7188/api/`, que corresponde à porta em que a API do back-end está sendo executada. Além disso, deve ser adicionado um `Dockerfile` no diretório do front-end, com instruções para compilar e executar a aplicação. Esse `Dockerfile` deve instalar as dependências, copiar os arquivos do projeto e definir as variáveis de ambiente. A construção do front-end é iniciada com a imagem `node:18`, e a aplicação compilada será servida utilizando `nginx:alpine`. Para finalizar, é necessário configurar o Nginx no arquivo `nginx.conf`, que direcionará as requisições ao front-end e, quando necessário, ao contêiner da API.

A integração entre os serviços de back-end, front-end e banco de dados é gerenciada por meio do arquivo `docker-compose.yml`, que define as configurações de cada serviço, incluindo portas, variáveis de ambiente e as dependências entre os contêineres. No `docker-compose.yml`, o serviço PostgreSQL é configurado para ser iniciado primeiro, com variáveis de ambiente para usuário, senha e nome do banco, além de um volume para garantir a persistência dos dados. A API é configurada para depender do PostgreSQL, sendo iniciada apenas quando o banco de dados estiver operacional. O serviço de front-end é iniciado após a configuração dos demais serviços, garantindo que todos os contêineres sejam iniciados na ordem correta e que as dependências estejam devidamente estabelecidas.

Para manter o ambiente Docker organizado e evitar resíduos de configurações anteriores, recomenda-se utilizar o comando `docker-compose down --rm all -v`, que remove todos os contêineres e imagens criados previamente. Em seguida, para construir e inicializar os contêineres, deve-se executar o comando `docker-compose up --build`. Esse processo cria e inicia o ambiente completo, integrando o banco de dados, a API e o front-end, permitindo o acesso a cada serviço conforme as portas configuradas. A estruturação com Docker proporciona um ambiente coeso e modular para desenvolvimento e produção, ideal para aplicações distribuídas.

## 7 CONCLUSÃO

A conclusão deste Projeto Integrador tem como propósito sintetizar os principais aspectos abordados ao longo do desenvolvimento e avaliar a viabilidade técnica e econômica do sistema proposto. O projeto TECHVAGAS foi idealizado com a finalidade de criar uma plataforma eficiente para a gestão de vagas de estágio, visando atender de maneira eficaz as

necessidades de candidatos e empresas. Durante o desenvolvimento, foi possível construir uma solução robusta, alicerçada em uma arquitetura que viabiliza o cadastro, a alteração e o controle de dados de forma organizada e segura.

A implementação das funcionalidades principais, como o cadastro de usuários, empresas e vagas, foi crucial para a criação de uma estrutura sólida e acessível ao público-alvo. Além disso, o sistema incorporou requisitos funcionais e não funcionais essenciais, como a atualização em tempo real do banco de dados e o suporte ao usuário, aspectos que garantem a eficiência da plataforma e a satisfação dos usuários.

Outro ponto importante foi a adaptação da plataforma para dispositivos móveis, o que assegura sua acessibilidade a um público diversificado, incluindo pessoas com limitações. A inclusão de recursos de acessibilidade digital, como validações em tempo real e interfaces adaptativas, demonstrou o compromisso do projeto com a inclusão social, ampliando seu alcance e tornando-a utilizável por um público mais amplo.

Em relação à viabilidade técnica e econômica, a análise realizada aponta para um cenário favorável ao desenvolvimento do sistema. A arquitetura cuidadosamente planejada e a adoção de boas práticas para a manutenção e escalabilidade garantem a sustentabilidade do projeto a longo prazo. Dessa forma, conclui-se que o projeto TECHVAGAS é plenamente viável, oferecendo uma solução inovadora e eficiente para conectar estudantes a oportunidades de estágio. Além disso, essa iniciativa contribui significativamente para o desenvolvimento acadêmico e profissional dos usuários, atendendo à crescente demanda por plataformas eficazes e acessíveis no mercado de estágios.

## 8 REFERÊNCIAS

*Colocar as referências utilizadas no projeto conforme os modelos abaixo:*

ASFHAL, C. R. **Gestão de segurança do trabalho e de saúde ocupacional**. São Paulo: Reichmann & Autores, 2005.

BEAIRD, J. **Princípios do Web Design Maravilhoso**. Rio de Janeiro: Altabooks, 2008.

CHIAVENATO, I. **Gestão de pessoas: o novo papel dos recursos humanos nas organizações**. 3. ed. Rio de Janeiro: Elsevier, 2008.

COSTA, H. J. Acidentes do trabalho: teremos nova lei acidentária?. **Jus Navigandi**, Teresina, v. 9, n. 664, 1 maio 2005. Disponível em: <<http://jus2.uol.com.br/doutrina/texto.asp?id=6662>>. Acesso em: 1 jul. 2010.



DELIBERATO, P. C. P. **Fisioterapia preventiva: fundamentos e aplicações**. São Paulo: Manole, 2002.

DIAS, E. C.; MELO, E. M. de. Políticas públicas em saúde e segurança no trabalho. In: MENDES, R. (Org.). **Patologia do trabalho**. 2. ed. atual. e ampl. São Paulo: Atheneu, 2007. v. 2. p. 1683-1720.

GUEDES, Gilleanes T. A. **UML Uma Abordagem prática**, 3 ed. São Paulo: Novatec, 2008.

GUEDES, Gilleanes T. A. **UML 2 Uma Abordagem prática**, São Paulo: Novatec, 2009.

WATRALL, E; SIARTO, J. Use A Cabeça! **Web Design**. Alta Books, 2009.

BRASIL. Decreto-Lei nº 4.073, de 30 de janeiro de 1942. Dispõe sobre a Lei Orgânica do Ensino Industrial. Diário Oficial [da] República Federativa do Brasil, Poder Executivo, Rio de Janeiro, RJ, 9 fev. 1942. Disponível em: <https://www2.camara.leg.br/legin/fed/declei/1940-1949/decreto-lei-4073-30-janeiro-1942-414503-133697-pe.html>. Acesso em: 05 out. 2024.

STREMEL, Patrícia Mendes e MARTINS, Cyntia Liesenberg. Trabalho docente em contextos de pandemia: ensino remoto e condições de trabalho. *Ensaio: Avaliação e Políticas Públicas em Educação*, Rio de Janeiro, v. 29, n. 112, p. 744-763, jul./set. 2021. Disponível em: <https://www.scielo.br/j/er/a/kKhXcCMp56LZ5R54fsL4PFq/?format=pdf>. Acesso em: 05 out. 2024.

BRASIL. Decreto-Lei nº 8.530, de 2 de janeiro de 1946. Dispõe sobre a Lei Orgânica do Ensino Comercial. Diário Oficial [da] República Federativa do Brasil, Poder Executivo, Rio de Janeiro, RJ, 8 jan. 1946. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/Decreto-Lei/1937-1946/Del8530.htm](https://www.planalto.gov.br/ccivil_03/Decreto-Lei/1937-1946/Del8530.htm). Acesso em: 5 out. 2024.

BRASIL. Lei nº 11.788, de 25 de setembro de 2008. Dispõe sobre o estágio de estudantes; altera a redação do art. 428 da Consolidação das Leis do Trabalho – CLT, aprovada pelo Decreto-Lei nº 5.452, de 1º de maio de 1943, e a Lei nº 9.394, de 20 de dezembro de 1996; revoga as disposições da Medida Provisória nº 2.164-41, de 24 de agosto de 2001, e dá outras providências. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, 26 set. 2008. Disponível em: [https://www.planalto.gov.br/ccivil\\_03/\\_Ato2007-2010/2008/Lei/L11788.htm](https://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Lei/L11788.htm). Acesso em: 05 out. 2024.

JORNAL DO COMMERCCIO. Mercado de estágios cresce no Brasil, liderado pelo setor jurídico. Disponível em: <https://jc.com.br/mercado-estagios-brasil>. Acesso em: 6 nov. 2024.

JORNAL DO COMMERCCIO. Mercado de estágios cresce no Brasil, liderado pelo setor jurídico. Disponível em: <https://jc.com.br/mercado-estagios-brasil>. Acesso em: 6 nov. 2024.

GAZETA BRASIL. Aumento de 18% no número de contratos de estágio em 2022. Disponível em: <https://gazetabrasil.com.br/contratos-estagio-2022>. Acesso em: 6 nov. 2024.

PRESSMAN, R. S. Engenharia de Software: fundamentos, métodos e padrões. 3. ed. Rio de Janeiro: LTC, 2015.

SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

BRASIL. Lei nº 11.788, de 25 de setembro de 2008. Dispõe sobre o estágio de estudantes e dá outras providências. Diário Oficial da União: seção 1, Brasília, DF, p. 1, 26 set. 2008.

SOMMERVILLE, Ian. Engenharia de Software. 8ª ed. São Paulo: Pearson Prentice Hall, 2007.

PAULA FILHO, Wilson de Pádua. Engenharia de Software: Fundamentos, Métodos e Padrões. 3ª ed. Rio de Janeiro: LTC, 2020.

GUEDES, Francisco. UML: Modelagem de Sistemas com UML. 2ª ed. São Paulo: Novatec, 2011.

MICROSOFT. *Web API*. 2024. Disponível em: <https://learn.microsoft.com/en-us/dotnet/web-api/>. Acesso em: 28 nov. 2024.

MICROSOFT. *C#*. 2024. Disponível em: <https://learn.microsoft.com/en-us/dotnet/csharp/>. Acesso em: 28 nov. 2024.

FACEBOOK. *React*. 2024. Disponível em: <https://reactjs.org/>. Acesso em: 28 nov. 2024.

TAILWIND CSS. *ShadCN UI*. 2024. Disponível em: <https://github.com/shadcn/ui>. Acesso em: 28 nov. 2024.

NORMAN, Donald A. The design of everyday things. 2. ed. New York: Basic Books, 2013.

UNGER, Russ; CHANDLER, Carolyn. O Guia para Projetar UX: Princípios de Design de Experiência do Usuário. Rio de Janeiro: Alta Books, 2013.

BROWN, Tim. Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias. Rio de Janeiro: Elsevier, 2010.

NIELSEN, Jakob. Usability engineering. 1. ed. Boston: Academic Press, 2006.

DATE, C. J. An Introduction to Database Systems. 8. ed. Boston: Addison-Wesley, 2004.

ELMASRI, Ramez; NAVATHE, Shamkant B. Fundamentals of Database Systems. 7. ed. Boston: Addison-Wesley, 2016.

MARTIN, James. Principles of software engineering: a unified view. 1. ed. Englewood Cliffs: Prentice-Hall, 1988.

KORTH, Henry F.; SILBERSCHATZ, Abraham. Database system concepts. 5. ed. New York: McGraw-Hill, 2005.

MACORATTI, J. Entity Framework Core: Criando Aplicações com EF Core e C#. 1ª ed. São Paulo: Clube de Autores, 2019.

MELONI, J. C. HTML, CSS & JavaScript All-in-One For Dummies. 3ª ed. Hoboken: Wiley, 2018.

REACT. React: A JavaScript library for building user interfaces. Disponível em: <https://react.dev>. Acesso em: 28 nov. 2024.

TOTVS. O que é front-end e qual a sua importância? Disponível em: <https://www.totvs.com/blog/front-end/>. Acesso em: 28 nov. 2024.

ANDERSON, R. Security Engineering: A Guide to Building Dependable Distributed Systems. 3ª ed. Indianapolis: Wiley, 2021.

BRADLEY, M. JSON Web Token (JWT): An Introduction and Overview. 2020. Disponível em: <https://jwt.io/introduction/>. Acesso em: 28 nov. 2024.

IBM. SHA-256 Hashing Algorithm. Disponível em: <https://www.ibm.com/docs/en/zos/2.4.0?topic=hash-sha256>. Acesso em: 28 nov. 2024.

SCHNEIER, B. Secrets and Lies: Digital Security in a Networked World. 1ª ed. Wiley, 2015.

WILLIAMS, R. TLS and HTTPS: Secure Your Web Applications. 2ª ed. O'Reilly, 2018.

DOCKER. Docker Documentation. 2024. Disponível em: <https://docs.docker.com/>. Acesso em: 28 nov. 2024.