

**Guilherme Carvalho**

**Victor Hugo**

**Kauã Henrique**

Leandro Bergamini

**TECHVAGAS**

**Projeto Integrador**

Disciplinas Envolvidas: Engenharia de Software, Técnicas Avançados de Banco de Dados Relacional e não Relacional, Técnicas Avançadas de Programação, Técnicas Avançadas de Programação Web e Mobile, Gestão Ágil de Projetos de Software.

Jales

2023

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc144809989)

[2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE 4](#_Toc144809990)

[2.1 Descrição dos objetivos do sistema 4](#_Toc144809991)

[2.2 Descrição do sistema atual 4](#_Toc144809992)

[2.3 Descrição dos principais problemas 4](#_Toc144809993)

[2.4 Descrição dos requisitos funcionais 4](#_Toc144809994)

[2.5 Descrição dos requisitos não funcionais 4](#_Toc144809995)

[3 VISÃO DE CASO DE USO – UML 5](#_Toc144809996)

[3.1 Diagrama de Classes 5](#_Toc144809997)

[3.2 Dicionário de classes 5](#_Toc144809998)

[3.3 Definição dos Atores 5](#_Toc144809999)

[3.4 Lista de Casos de Uso 5](#_Toc144810000)

[3.4. Diagrama de Casos de Uso 5](#_Toc144810001)

[3.5. Diagrama de Casos de uso individuais 6](#_Toc144810002)

[3.6. Diagrama de Sequência 7](#_Toc144810003)

[3.7. Diagrama de Comunicação 7](#_Toc144810004)

[3.8. Diagrama de Atividade 7](#_Toc144810005)

[4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º semstre) 8](#_Toc144810006)

[4.1 Descrição de cenário 8](#_Toc144810007)

[4.2 Descrição de personas 8](#_Toc144810008)

[4.3 Esboços de tela (wireframes) 8](#_Toc144810009)

[4.4 Protótipos de tela 8](#_Toc144810010)

[5 BANCO DE DADOS 9](#_Toc144810011)

[5.1 Modelo Entidade Relacionamento 9](#_Toc144810012)

[5.2 Script das tabelas 9](#_Toc144810013)

[6 ARQUITETURA DE SOFTWARE 10](#_Toc144810014)

[6.1 Arquitetura de desenvolvimento 10](#_Toc144810015)

[6.2 Telas do sistema 10](#_Toc144810016)

[7 CONCLUSÃO 11](#_Toc144810017)

[8 REFERÊNCIAS 12](#_Toc144810018)

# 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, 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, 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. 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, 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. 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.

# LEVANTAMENTO DE REQUISITOS DE SOFTWARE

O levantamento de requisitos é um estágio crucial no ciclo de desenvolvimento de software, desempenhando um papel fundamental na compreensão das necessidades dos usuários e na definição clara do escopo do projeto. Para 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, 2011, p.118).

Conforme observado por Sommerville (2018), o levantamento de requisitos não apenas estabelece a base fundamental, mas também se configura como um estágio essencial no ciclo de vida do desenvolvimento de software. Sua importância reside na capacidade de construir um software que atenda às demandas dos usuários e agregue valor ao negócio. Esse processo crítico assegura o sucesso do projeto ao garantir uma compreensão clara e abrangente das expectativas e requisitos do cliente, delineando diretrizes cruciais para a construção de um sistema que verdadeiramente satisfaça as necessidades e objetivos estabelecidos.

## Descrição dos objetivos do sistema

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.

## Descrição do sistema atual

Atualmente, 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.

## 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 fornece suporte contínuo, garantindo que tanto alunos quanto empresas parceiras se familiarizem rapidamente com a nova plataforma.

## Descrição dos requisitos funcionais

### Login Administrador

O sistema deve permitir que o usuário consiga efetuar login como administrador, validando se os dados informados estão corretos.

### Alterar Administrador

O sistema deve permitir o administrador alterar seus dados, caso precise.

### Logout Administrador

O sistema deve permitir que o administrador efetue logout de sua sessão, retornando a página inicial do sistema.

### Cadastrar Aluno

O aluno deve realizar o registro no sistema

### Listar Alunos

O sistema deve permitir que o administrador liste todos os alunos cadastrados no sistema.

### Excluir Aluno

O sistema deve permitir que o administrador exclua um aluno cadastrado no sistema.

### Login Aluno

O sistema deve permitir que o usuário consiga efetuar login como aluno, validando se os dados informados estão corretos.

### Alterar Aluno

O sistema deve permitir que o administrador e o aluno alterem seus dados, caso precise.

### Logout Aluno

O sistema deve permitir que o aluno efetue logout de sua sessão, retornando a página inicial do sistema.

### Cadastrar Matrícula

O sistema deve permitir que o aluno cadastre sua matrícula.

### Alterar Matrícula

O sistema deve permitir que o administrador e o aluno alterem sua matrícula.

### Listar Matrícula

O sistema deve permitir que o administrador liste todas as matrículas cadastradas no sistema.

### Excluir Matrícula

O sistema deve permitir que o administrador exclua uma matrícula caso necessário

### Validação Empresa

O sistema deve notificar o administrador quando uma empresa enviar uma solicitação de cadastro, permitindo que o administrador avalie a legitimidade da empresa.

### Cadastro Empresa

O sistema deve possibilitar a registro da empresa após a confirmação do administrador

### Alterar Empresa

O sistema deve permitir que o administrador e a empresa alterem os seus dados caso necessário.

### Listar Empresa

O sistema deve permitir que o administrador liste todas as empresas cadastradas no sistema.

### Excluir Empresa

O sistema deve permitir que o administrador exclua uma empresa caso necessário.

### Cadastrar Vagas

O sistema deve permitir que uma empresa possa cadastrar suas vagas no sistema

### Alterar Vaga

O sistema deve permitir que a empresa possa alterar os dados da vaga caso necessário.

### Listar Vaga

O sistema deve permitir que o aluno e o administrador listem todas as vagas do sistema

### Candidatar Vaga

O sistema deve permitir que o aluno se candidate para uma vaga

### Processo Seletivo

O sistema deve viabilizar que o estudante acesse seu processo na vaga, enquanto a empresa tem a capacidade de modificar o status do processo seletivo, retornando essas atualizações ao aluno.

### Finalizar Vaga

O sistema deve permitir que a empresa possa finalizar a vaga caso já tenha obtido o número de candidatos.

### Cadastrar Cargo

O sistema deve permitir que a empresa possa cadastrar o cargo desejado para a vaga

### Alterar Cargo

O sistema deve permitir que a empresa possa alterar o cargo da vaga caso necessário

## Descrição dos requisitos não funcionais

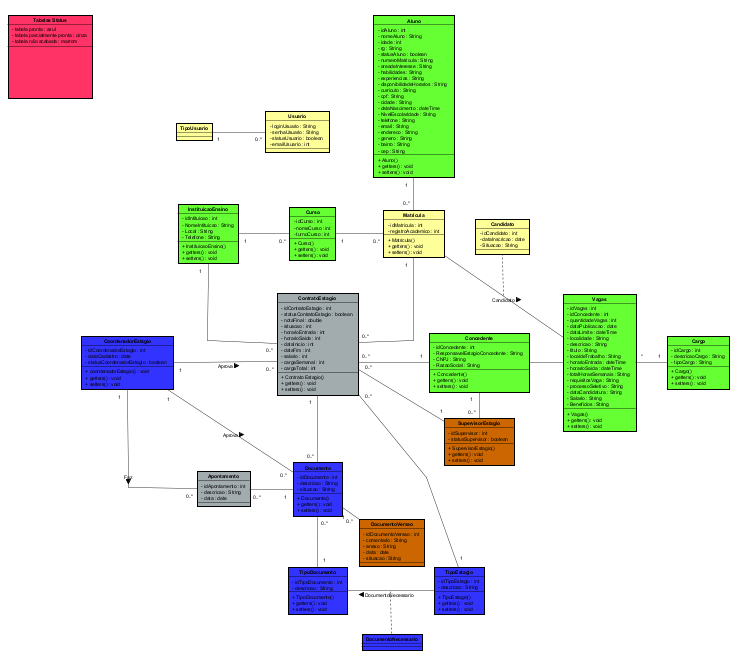
Os requisitos não funcionais presentes no sistema será o de: segurança, desempenho, segurança autentificarão e autorização, usabilidade, interface agradável, desempenho de banco de dados e backup realizado semanalmente.

# VISÃO DE CASO DE USO – UML

## 3.1 Diagrama de Classes

No contexto do livro "UML2: uma abordagem prática" de Guedes, 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, 2018).

Nesta subseção, será exibido o diagrama de classes do sistema, que engloba a configuração da camada de negócios correspondente.

Figura 1 - Diagrama de Classes.

Fonte: Elaborado pelos autores.

## 3.2 Dicionário de classes

Nesta subseção, serão apresentados os dicionários de classe, que desempenham um papel fundamental na identificação dos atributos e na definição de suas funções na regra de negócio do sistema desenvolvido.

Aluno: tem como objetivo definir quem é um possível candidato a vagas e novo estagiário da instituição.

**Quadro 1** — Descrição Classe Aluno

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idAluno | Integer | Utilizado para localizar o aluno dentro do sistema |
| NomeAluno | String | Nomeia os usuários 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 usuário pode oferecer |
| Experiencias | String | Informa quais foram as experiencia que o usuário possui |
| DisponibilidadesHorários | String | Quais os horários o usuário tem disponível |
| Currículo | String | É apresentado o currículo do usuário |
| CPF | String | É o cadastro de pessoa física |
| DataNacimento | dateTime | É informado a data de nascimento do usuário |
| NivelEscolaridade | 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

Vagas: tem como objetivo definir qual a vaga está disponível para o candidato e todas as suas exigências.

**Quadro 2** — Descrição Classe Vagas

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idVagas | Integer | É utilizado para localizar dentro dos sistemas cada uma das vagas |
| IdConcendente | 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 |
| DataPublicaçao | Date | Informa e oficializa a postagem da vaga |
| DataLimite | dateTime | É defina 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 |
| Descriçao | String | Na descrição será descrito todas as informações referentes a vaga de estágio |
| Titulo | String | Informa como um previa 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 | Ira 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

Contrato Estágio: tem como objetivo definir quis os termos e reponsabilidades irão conter no contrato do aluno.

**Quadro 3** — Descrição Classe Contrato Estágio

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idContratoEstagio | Integer | Utilizado para identificar os contratos presentes dentro do sistema |
| StatusContratoEstagio | Boolean | Mostra em qual etapa está a contratação do usuário |
| NotaFinal | Double | Nota que avalia o desempenho do estagiário |
| Situação | Integer | Mostra qual a atual situação do estagiário |
| HorárioEntrada | Integer | De acordo com o período e a carga horaria é definido o horário de entrada |
| HorarioSaida | Integer | De acordo com o horário de entrada é definido o horário de saída |
| DataInicio | Integer | É definido no contrato o prazo de início contrato como estagiário |
| DataFim | Integer | De acordo com as regras de estágio é definido um prazo limite de dois anos ou menos para o estagiário |
| Salario | Integer | Salário pode variar de acordo com a empresa e de acordo a área em que ele irá atuar |
| CargaSemanal | Integer | É definido quanto de horas o estagiário deverá cumprir na semana |
| CargaTotal | Integer | É somado a carga horaria total que o estagiário deverá entregar |

Fonte: Elaborado pelos autores

Usuário: tem como objetivo definir as informações de cadastro e login do usuário

**Quadro 4** — Descrição Classe Usuário

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| loginUsuario | String | É o que autêntica 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

Matrícula: tem como objetivo definir as informações da matrícula realizada pelo aluno

**Quadro 6** — Descrição Classe Matrícula

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idMatricula | Integer | Identifica as matrículas que foram feitas |
| registroAcademico | Integer | Mostra o registro acadêmico do aluno |

Fonte: Elaborado pelos autores

Instituição Ensino: tem como objetivo definir as informações da instituição de ensino que pode ou não conceder uma vaga ou estágio

**Quadro 7** — Descrição Classe Instituição Ensino

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |

Fonte: Elaborado pelos autores

Candidato: tem como objetivo definir as informações do candidato, nessa classe ele passa de um aluno para postulante a vaga de estágio

**Quadro 8** — 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

Processo: tem como objetivo de mostrar ao candidato como está ocorrendo o processo seletivo referente a sua vaga

**Quadro 9** — Descrição Classe Processo Seletivo

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idProcessoSeletivo | Integer | Identifica qual o processo seletivo |
| dataCandidatura | Date | Refere-se ao dia em que o processo seletivo foi publicado |

Fonte: Elaborado pelos autores

Coordenador Estágio: tem como objetivo identificar e registrar um dos atores do sistema

**Quadro 10** — Descrição Classe Coordenador Estágio

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idCoordenadorEstagio | Integer | Identifica quem é o coordenador de estágio |
| dataCadastro | Date | Refere-se ao dia em que o coordenador de estágio foi cadastrado no sistema |
| StatusCoordenadorEstagio | Boolean | Define o status do coordenador, se ele está inativo ou ativo no banco |

Fonte: Elaborado pelos autores

Concedente: tem como objetivo identificar qual a empresa que vai conceder a vaga e o estágio para um candidato

**Quadro 11** — Descrição Classe Concedente

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idConcedente | Integer | Identifica qual é a empresa que irá conceder o estágio |
| ResposnsavelEstagioConcedente | String | Mostra quem é a pessoa de dentro da empresa que é responsável por conceder o estágio |

Fonte: Elaborado pelos autores

Cargo: tem como objetivo mostrar qual o cargo os novos estagiários irão trabalhar

**Quadro 12** — Descrição Classe Cargo

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idCargo | Integer | Identifica qual é o cargo |
| descricaoCargo | String | Descreve como será o cargo exercido com detalhes |
| tipoCargo | String | Define quais funções o estagiário irá exercer na instituição |

Fonte: Elaborado pelos autores

Requisito Vagas: mostra quais são os requisitos para que um aluno entre naquela vaga específica

**Quadro 13** — Descrição Classe Requisito Vagas

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idRequisitoVagas | Integer | Identifica qual é o requisito daquela vaga |
| requisitosVagas | String | Detalha quais os requisitos específicos para aquela vaga |

Fonte: Elaborado pelos autores

Supervisor Estágio: tem como objetivo identificar e registrar um dos atores do sistema

**Quadro 14** — Descrição Classe Requisito Vagas

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idSupervisor | Integer | Identifica qual é o identificador do supervisor de estágio |
| statusSupervisor | Boolean | Define o status do supervisor, se ele está inativo ou ativo no banco |

Fonte: Elaborado pelos autores

Documento: tem como objetivo mostrar qual o documento que está sendo cadastrado

**Quadro 15** — Descrição Classe Requisito Vagas

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idDocumeto | Integer | Identifica qual é o identificador do documento |
| descricaoDocumento | String | Descreve como está aquele documento no momento |
| situacaoDocumento | String | Define qual a situação que o documento se encontra |

Fonte: Elaborado pelos autores

Apontamento: tem como objetivo mostrar qualquer observação referente a um documento específico

**Quadro 16** — Descrição Classe Apontamento

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idApontamento | Integer | Identifica qual é o apontamento |
| descricaoApontamento | String | Descreve o motivo daquele apontamento estar acontecendo |
| dataApontamento | Date | Define qual a data que aquele apontamento foi feito |

Fonte: Elaborado pelos autores

Tipo Documento: tem como objetivo mostrar qual o tipo do documento

**Quadro 17** — Descrição Classe Tipo Documento

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idTipoDocumento | Integer | Identifica qual é o tipo do documento |
| descricaoTipoDocumento | String | Mostra qual a descrição do tipo documento |

Fonte: Elaborado pelos autores

Documento Versão: tem como objetivo mostrar qual é a versão de um determinado documento

**Quadro 18** — Descrição Classe Documento Versão

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idDocumentoVersao | Integer | Identifica qual é a versão do documento |
| comentarioDocumentoVersao | String | Mostra os comentários feitos na versão atual do documento |
| anexoDocumentoVersao | String | Identifica qual o anexo da versão do documento |
| dataDocumentoVersao | Date | Mostra a data da versão do documento |
| situacaoDocumentoVersao | String | Identifica a situação da versão do documento |

Fonte: Elaborado pelos autores

Tipo Estágio: tem como objetivo mostrar qual é o tipo do estágio

**Quadro 19** — Descrição Classe Tipo Estágio

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idTipoEstagio | Integer | Identifica qual o tipo do estágio |
| descricaoTipoEstagio | String | Descreve como será aquele estágio |

Fonte: Elaborado pelos autores

Documento Necessário: É uma classe gerada partindo de tipo estágio e tipo documento, ela identifica qual o documento o aluno terá que apresentar para provar que já faz algum estágio ou não

**Quadro 20** — Descrição Classe Documento Necessário

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |

Fonte: Elaborado pelos autores

## 3.3 Definição dos Atores

O diagrama de caso de uso concentra-se predominantemente em dois elementos fundamentais: atores e casos de uso. Os atores representam os usuários do sistema, assim como qualquer hardware especial ou sistema integrado que possa utilizar as funcionalidades e serviços oferecidos pelo sistema. Na representação gráfica, símbolos como bonecos são empregados para identificar os atores, acompanhados de breves descrições que detalham suas funções específicas no sistema. Essa representação visual simplifica a compreensão dos papéis desempenhados por diferentes componentes. Cada descrição associada a um ator destaca sua função específica no contexto do sistema, facilitando a análise e compreensão durante o processo de modelagem, conforme preconizado no livro "UML 2 - Uma Abordagem Prática".

**Aluno:** responsável pelo uso da aplicação. Esse usuário deve realizar o seu cadastro, e interagir com o sistema sendo o estagiário ou um possível candidato à vaga.

**Candidato:** responsável pelo uso da aplicação. É o usuário que passa do “status” de aluno para entrar de fato como um postulante a vaga de estágio.

**Concedente:** responsável por conceder o estágio. Esse usuário pode ser uma empresa, uma pessoa ou a própria instituição, é responsável por conceder as vagas de estágio para os alunos.

**Supervisor Estágio:** responsável por supervisionar o estagiário. Esse usuário deve realizar o supervisionamento do estagiário, atribuir tarefas e assim contratos por exemplo, pode ser da própria instituição de ensino ou da concedente de estágio.

**Coordenador Estágio:** responsável por coordenar as vagas de estágio. Esse usuário na maioria das vezes é da própria instituição, é o responsável da instituição por distribuir e criar vagas de estágio.

## 3.4 Lista de Casos de Uso

Conforme destacado no livro "UML 2 - Uma Abordagem Prática", o Diagrama de Caso de Uso é considerado o mais abstrato, conferindo-lhe maior flexibilidade e informalidade. Ele é empregado para fornecer uma visão abrangente do sistema, com o propósito de identificar os atores que interagirão com o software, juntamente com as funcionalidades correspondentes. Embora seja comumente utilizado nas fases de levantamento e análise de requisitos, sua aplicação estende-se ao longo de todo o processo de modelagem, podendo inclusive servir como base para a elaboração de outros diagramas.

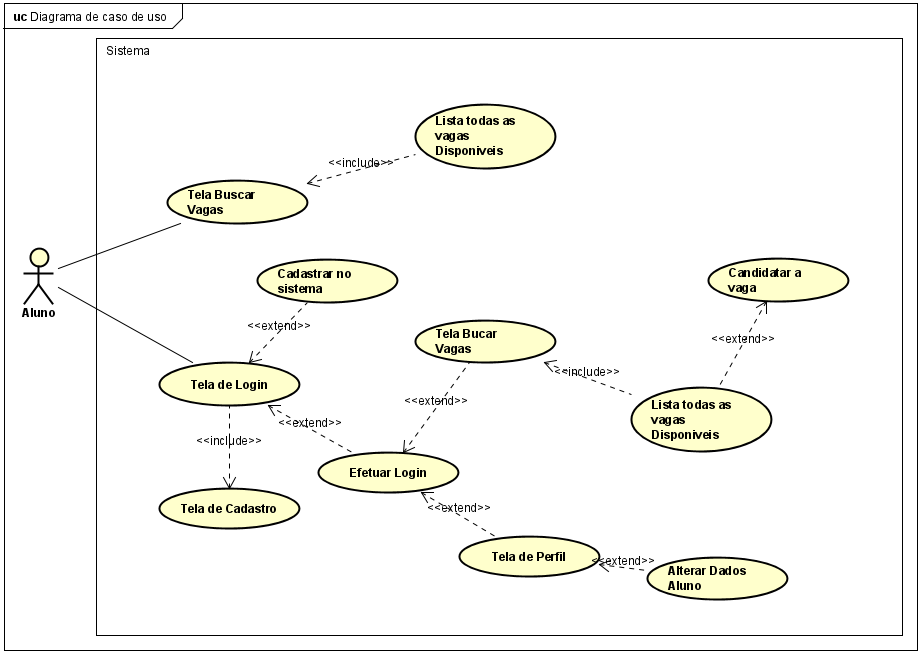
Quadro 21 – Lista de Casos de Uso

| **Nº** | **Descrição do Caso de Uso** | **Entrada** | **Caso de Uso** | **Resposta** |
| --- | --- | --- | --- | --- |
|  | Alterar Administrador | Informa dados administrador | Alterar administrador | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera matrícula | Informa dados da matrícula | Alterar matrícula | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera curso | Informa dados do curso | Alterar curso | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera candidato | Informa dados do candidato | Alterar candidato | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera instituição de ensino | Informa dados instituição de ensino | Altera instituição de ensino | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera coordenador de estágio | Informa dados coordenador de estágio | Altera coordenador de estágio | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera contrato estágio | Informa dados contrato estágio | Altera contrato estágio | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera concedente | Informa dados concedente | Altera concedente | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera vagas | Informa dados vagas | Altera vagas | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera cargo | Informa dados cargo | Altera cargo | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera requisito vagas | Informa dados requisito vaga | Altera requisito vagas | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera supervisor estágio | Informa dados supervisor estágio | Altera supervisor estágio | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera processo seletivo | Informa dados processo seletivo | Altera processo seletivo | Retorna mensagem “Alterado com sucesso” |
|  | Administrador altera apontamento | Informa dados apontamento | Altera apontamento | Retorna mensagem” Alterado com sucesso” |
|  | Administrador altera documento | Informa dados documento | Altera documento | Retorna mensagem” Alterado com sucesso” |
|  | Administrador altera tipo documento | Informa dados tipo documento | Altera tipo documento | Retorna mensagem” Alterado com sucesso” |
|  | Administrador altera documento versão | Informa dados documento versão | Altera documento versão | Retorna mensagem” Alterado com sucesso” |
|  | Administrador altera documento necessário | Informa documento necessário | Altera documento necessário | Retorna mensagem” Alterado com sucesso” |
|  | Administrador altera tipo estágio | Informa tipo estágio | Altera tipo estágio | Retorna mensagem” Alterado com sucesso” |
|  | Cadastra Administrador | Informa dados administrador | Cadastrar administrador | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra matrícula | Informa dados da matrícula | Cadastrar matrícula | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra curso | Informa dados do curso | Cadastrar curso | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra candidato | Informa dados do candidato | Cadastra candidato | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra instituição de ensino | Informa dados instituição de ensino | Cadastra instituição de ensino | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra coordenador de estágio | Informa dados coordenador de estágio | Cadastra coordenador de estágio | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra contrato estágio | Informa dados contrato estágio | Cadastra contrato estágio | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra concedente | Informa dados concedente | Cadastra concedente | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra vagas | Informa dados vagas | Cadastra vagas | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra cargo | Informa dados cargo | Cadastra cargo | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra requisito vagas | Informa dados requisito vaga | Cadastra requisito vagas | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra supervisor estágio | Informa dados supervisor estágio | Cadastra supervisor estágio | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra processo seletivo | Informa dados processo seletivo | Cadastra processo seletivo | Retorna mensagem “Cadastrado com sucesso” |
|  | Administrador cadastra apontamento | Informa dados apontamento | Cadastra apontamento | Retorna mensagem” Cadastrado com sucesso” |
|  | Administrador cadastra documento | Informa dados documento | Cadastra documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Administrador cadastra tipo documento | Informa dados tipo documento | Cadastra tipo documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Administrador cadastra documento versão | Informa dados documento versão | Cadastra documento versão | Retorna mensagem” Cadastrado com sucesso” |
|  | Administrador cadastra documento necessário | Informa documento necessário | Cadastra documento necessário | Retorna mensagem” Cadastrado com sucesso” |
|  | Administrador cadastra tipo estágio | Informa tipo estágio | Cadastra tipo estágio | Retorna mensagem” Cadastrado com sucesso” |
|  | Desativa Administrador | Id Administrador | Desativa Administrador | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa matrícula | Id matrícula | Desativa matrícula | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa curso | Id curso | Desativa curso | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa candidato | Id candidato | Desativa candidato | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa instituição de ensino | Id instituição de ensino | Desativa instituição de ensino | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa coordenador de estágio | Id coordenador de estágio | Desativa coordenador de estágio | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa contrato estágio | Id contrato estágio | Desativa contrato estágio | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa concedente | Id concedente | Desativa concedente | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa vagas | Id vagas | Desativa vagas | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa cargo | Id cargo | Desativa cargo | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa requisito vagas | Id requisito vaga | Desativa requisito vagas | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa supervisor estágio | Id supervisor estágio | Desativa supervisor estágio | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa processo seletivo | Id processo seletivo | Desativa processo seletivo | Retorna mensagem “Desativado com sucesso” |
|  | Administrador desativa apontamento | Id apontamento | Desativa apontamento | Retorna mensagem” Desativado com sucesso” |
|  | Administrador desativa documento | Id documento | Desativa documento | Retorna mensagem” Desativado com sucesso” |
|  | Administrador desativa tipo documento | Id tipo documento | Desativa tipo documento | Retorna mensagem” Desativado com sucesso” |
|  | Administrador desativa documento versão | Id documento versão | Desativa documento versão | Retorna mensagem” Desativado com sucesso” |
|  | Administrador desativa documento necessário | Id documento necessário | Desativa documento necessário | Retorna mensagem” Desativado com sucesso” |
|  | Administrador desativa tipo estágio | Id tipo estágio | Desativa tipo estágio | Retorna mensagem” Desativado com sucesso” |
|  | Exclui Administrador | Id Administrador | Exclui Administrador | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui matrícula | Id matrícula | Exclui matrícula | Retorna mensagem “Desativado com sucesso” |
|  | Administrador exclui curso | Id curso | Exclui curso | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui candidato | Id candidato | Exclui candidato | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui instituição de ensino | Id instituição de ensino | Exclui instituição de ensino | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui coordenador de estágio | Id coordenador de estágio | Exclui coordenador de estágio | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui contrato estágio | Id contrato estágio | Exclui contrato estágio | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui concedente | Id concedente | Desativa concedente | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui vagas | Id vagas | Exclui vagas | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui cargo | Id cargo | Exclui cargo | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui requisito vagas | Id requisito vaga | Exclui requisito vagas | Retorna mensagem “Excluído com sucesso” |
|  | Administrador desativa supervisor estágio | Id supervisor estágio | Exclui supervisor estágio | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui processo seletivo | Id processo seletivo | Exclui processo seletivo | Retorna mensagem “Excluído com sucesso” |
|  | Administrador exclui apontamento | Id apontamento | Exclui apontamento | Retorna mensagem” Excluído com sucesso” |
|  | Administrador exclui documento | Id documento | Exclui documento | Retorna mensagem” Excluído com sucesso” |
|  | Administrador exclui tipo documento | Id tipo documento | Exclui tipo documento | Retorna mensagem” Excluído com sucesso” |
|  | Administrador exclui documento versão | Id documento versão | Exclui documento versão | Retorna mensagem” Excluído com sucesso” |
|  | Administrador exclui documento necessário | Id documento necessário | Exclui documento necessário | Retorna mensagem” Excluído com sucesso” |
|  | Administrador exclui tipo estágio | Id tipo estágio | Exclui tipo estágio | Retorna mensagem” Excluído com sucesso” |
|  | Consulta Administrador | Id Administrador | Consulta Administrador | Retorna dados administrador |
|  | Administrador consulta matrícula | Id matrícula | Consulta matrícula | Retorna dados administrador |
|  | Administrador consulta curso | Id curso | Consulta curso | Retorna dados curso |
|  | Administrador consulta candidato | Id candidato | Consulta candidato | Retorna dados candidato |
|  | Administrador consulta instituição de ensino | Id instituição de ensino | Consulta instituição de ensino | Retorna dados instituição de ensino |
|  | Administrador consulta coordenador de estágio | Id coordenador de estágio | Consulta coordenador de estágio | Retorna dados coordenador de estágio |
|  | Administrador consulta contrato estágio | Id contrato estágio | Consulta contrato estágio | Retorna dados contrato estágio |
|  | Administrador consulta concedente | Id concedente | Consulta concedente | Retorna dados concedente |
|  | Administrador consulta vagas | Id vagas | Consulta vagas | Retorna dados vagas |
|  | Administrador consulta cargo | Id cargo | Consulta cargo | Retorna dados cargo |
|  | Administrador consulta requisito vagas | Id requisito vaga | Consulta requisito vagas | Retorna dados requisito vagas |
|  | Administrador consulta supervisor estágio | Id supervisor estágio | Consulta supervisor estágio | Retorna dados supervisor estágio |
|  | Administrador consulta processo seletivo | Id processo seletivo | Consulta processo seletivo | Retorna dados processo seletivo |
|  | Administrador consulta apontamento | Id apontamento | Consulta apontamento | Retorna dados apontamento |
|  | Administrador consulta documento | Id documento | Consulta documento | Retorna dados documento |
|  | Administrador consulta tipo documento | Id tipo documento | Consulta tipo documento | Retorna dados tipo documento |
|  | Administrador consulta documento versão | Id documento versão | Consulta documento versão | Retorna dados documento versão |
|  | Administrador consulta documento necessário | Id documento necessário | Consulta documento necessário | Retorna dados documento necessário |
|  | Administrador consulta tipo estágio | Id tipo estágio | Consulta tipo estágio | Retorna dados tipo estágio |
|  | Coordenador Estágio altera apontamento | Informa dados apontamento | Altera apontamento | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio altera documento | Informa dados documento | Altera documento | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio altera tipo documento | Informa dados tipo documento | Altera tipo documento | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio altera documento versão | Informa dados documento versão | Altera documento versão | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio altera documento necessário | Informa documento necessário | Altera documento necessário | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio altera tipo estágio | Informa tipo estágio | Altera tipo estágio | Retorna mensagem” Alterado com sucesso” |
|  | Coordenador Estágio cadastra apontamento | Informa dados apontamento | Cadastra apontamento | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio cadastra documento | Informa dados documento | Cadastra documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio cadastra tipo documento | Informa dados tipo documento | Cadastra tipo documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio cadastra documento versão | Informa dados documento versão | Cadastra documento versão | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio cadastra documento necessário | Informa documento necessário | Cadastra documento necessário | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio cadastra tipo estágio | Informa tipo estágio | Cadastra tipo estágio | Retorna mensagem” Cadastrado com sucesso” |
|  | Coordenador Estágio desativa apontamento | Id apontamento | Desativa apontamento | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio desativa documento | Id documento | Desativa documento | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio desativa tipo documento | Id tipo documento | Desativa tipo documento | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio desativa documento versão | Id documento versão | Desativa documento versão | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio desativa documento necessário | Id documento necessário | Desativa documento necessário | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio desativa tipo estágio | Id tipo estágio | Desativa tipo estágio | Retorna mensagem” Desativado com sucesso” |
|  | Coordenador Estágio exclui apontamento | Id apontamento | Exclui apontamento | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio exclui documento | Id documento | Exclui documento | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio exclui tipo documento | Id tipo documento | Exclui tipo documento | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio exclui documento versão | Id documento versão | Exclui documento versão | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio exclui documento necessário | Id documento necessário | Exclui documento necessário | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio exclui tipo estágio | Id tipo estágio | Exclui tipo estágio | Retorna mensagem” Excluído com sucesso” |
|  | Coordenador Estágio consulta apontamento | Id apontamento | Consulta apontamento | Retorna dados apontamento |
|  | Coordenador Estágio consulta documento | Id documento | Consulta documento | Retorna dados documento |
|  | Coordenador Estágio consulta tipo documento | Id tipo documento | Consulta tipo documento | Retorna dados tipo documento |
|  | Coordenador Estágio consulta documento versão | Id documento versão | Consulta documento versão | Retorna dados documento versão |
|  | Coordenador Estágio consulta documento necessário | Id documento necessário | Consulta documento necessário | Retorna dados documento necessário |
|  | Coordenador Estágio consulta tipo estágio | Id tipo estágio | Consulta tipo estágio | Retorna dados tipo estágio |
|  | Supervisor Estágio altera apontamento | Informa dados apontamento | Altera apontamento | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio altera documento | Informa dados documento | Altera documento | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio altera tipo documento | Informa dados tipo documento | Altera tipo documento | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio altera documento versão | Informa dados documento versão | Altera documento versão | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio altera documento necessário | Informa documento necessário | Altera documento necessário | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio altera tipo estágio | Informa tipo estágio | Altera tipo estágio | Retorna mensagem” Alterado com sucesso” |
|  | Supervisor Estágio cadastra apontamento | Informa dados apontamento | Cadastra apontamento | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio cadastra documento | Informa dados documento | Cadastra documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio cadastra tipo documento | Informa dados tipo documento | Cadastra tipo documento | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio cadastra documento versão | Informa dados documento versão | Cadastra documento versão | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio cadastra documento necessário | Informa documento necessário | Cadastra documento necessário | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio cadastra tipo estágio | Informa tipo estágio | Cadastra tipo estágio | Retorna mensagem” Cadastrado com sucesso” |
|  | Supervisor Estágio desativa apontamento | Id apontamento | Desativa apontamento | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor Estágio desativa documento | Id documento | Desativa documento | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor Estágio desativa tipo documento | Id tipo documento | Desativa tipo documento | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor Estágio desativa documento versão | Id documento versão | Desativa documento versão | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor desativa documento necessário | Id documento necessário | Desativa documento necessário | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor Estágio desativa tipo estágio | Id tipo estágio | Desativa tipo estágio | Retorna mensagem” Desativado com sucesso” |
|  | Supervisor Estágio exclui apontamento | Id apontamento | Exclui apontamento | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio exclui documento | Id documento | Exclui documento | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio exclui tipo documento | Id tipo documento | Exclui tipo documento | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio exclui documento versão | Id documento versão | Exclui documento versão | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio exclui documento necessário | Id documento necessário | Exclui documento necessário | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio exclui tipo estágio | Id tipo estágio | Exclui tipo estágio | Retorna mensagem” Excluído com sucesso” |
|  | Supervisor Estágio consulta apontamento | Id apontamento | Consulta apontamento | Retorna dados apontamento |
|  | Supervisor Estágio consulta documento | Id documento | Consulta documento | Retorna dados documento |
|  | Supervisor Estágio consulta tipo documento | Id tipo documento | Consulta tipo documento | Retorna dados tipo documento |
|  | Supervisor Estágio consulta documento versão | Id documento versão | Consulta documento versão | Retorna dados documento versão |
|  | Supervisor Estágio consulta documento necessário | Id documento necessário | Consulta documento necessário | Retorna dados documento necessário |
|  | Supervisor Estágio consulta tipo estágio | Id tipo estágio | Consulta tipo estágio | Retorna dados tipo estágio |
|  | Concedente altera contrato estágio | Informa dados contrato estágio | Cadastra contrato estágio | Retorna mensagem “Cadastrado com sucesso” |
|  | Concedente altera vagas | Informa dados vagas | Altera vagas | Retorna mensagem “Alterado com sucesso” |
|  | Concedente altera requisito vagas | Informa dados requisito vaga | Altera requisito vagas | Retorna mensagem “Alterado com sucesso” |
|  | Concedente cadastra contrato | Informa dados contrato | Cadastra contrato | Retorna mensagem” Cadastrado com sucesso” |
|  | Concedente cadastra vagas | Informa dados vagas | Cadastra vagas | Retorna mensagem” Cadastrado com sucesso” |
|  | Concedente cadastra requisito vagas | Informa dados requisito vagas | Cadastra requisito vagas | Retorna mensagem” Cadastrado com sucesso” |
|  | Concedente desativa contrato | Id contrato | Desativa contrato | Retorna mensagem” Desativado com sucesso” |
|  | Concedente desativa requisito vagas | Id requisito vagas | Desativa requisito vagas | Retorna mensagem” Desativado com sucesso” |
|  | Concedente desativa vagas | Id vagas | Desativa vagas | Retorna mensagem” Desativado com sucesso” |
|  | Concedente exclui contrato | Id documento contrato | Exclui documento contrato | Retorna mensagem” Excluído com sucesso” |
|  | Concedente exclui requisito vagas | Id documento requisito vagas | Exclui documento requisito vagas | Retorna mensagem” Excluído com sucesso” |
|  | Concedente exclui vagas | Id documento vagas | Exclui documento vagas | Retorna mensagem” Excluído com sucesso” |
|  | Concedente consulta contrato | Id contrato | Consulta contrato | Retorna dados contrato |
|  | Concedente consulta requisito vagas | Id requisito vagas | Consulta requisito vagas | Retorna dados requisito vagas |
|  | Concedente consulta vagas | Id vagas | Consulta vagas | Retorna dados vagas |
|  | Sistema Lista Administrador |  | Listar administrador | Retorna lista administrador |
|  | Sistema Lista matrícula |  | Listar matrícula | Retorna lista matrícula |
|  | Sistema Lista curso |  | Listar curso | Retorna lista curso |
|  | Sistema Lista candidato |  | Listar candidato | Retorna lista candidato |
|  | Sistema Lista instituição de ensino |  | Listar instituição de ensino | Retorna lista instituição de ensino |
|  | Sistema Lista coordenador de estágio |  | Listar coordenador de estágio | Retorna lista coordenador de estágio |
|  | Sistema Lista contrato estágio |  | Listar contrato estágio | Retorna lista contrato estágio |
|  | Sistema Lista concedente |  | Listar concedente | Retorna lista concedente |
|  | Sistema Lista vagas |  | Listar vagas | Retorna lista vagas |
|  | Sistema Lista cargo |  | Listar cargo | Retorna lista cargo |
|  | Sistema Lista requisito vagas |  | Listar requisito vagas | Retorna lista requisito vagas |
|  | Sistema Lista supervisor estágio |  | Listar supervisor estágio | Retorna lista supervisor estágio |
|  | Sistema Lista processo seletivo |  | Listar processo seletivo | Retorna lista processo seletivo |
|  | Sistema Lista apontamento |  | Listar apontamento | Retorna lista apontamento |
|  | Sistema Lista documento |  | Listar documento | Retorna lista documento |
|  | Sistema Lista tipo documento |  | Listar tipo documento | Retorna lista tipo documento |
|  | Sistema Lista documento versão |  | Listar documento versão | Retorna lista documento versão |
|  | Sistema Lista documento necessário |  | Listar documento necessário | Retorna lista documento necessário |
|  | Sistema Lista estágio |  | Listar tipo estágio | Retorna lista estágio |

## 3.4. Diagrama de Casos de Uso

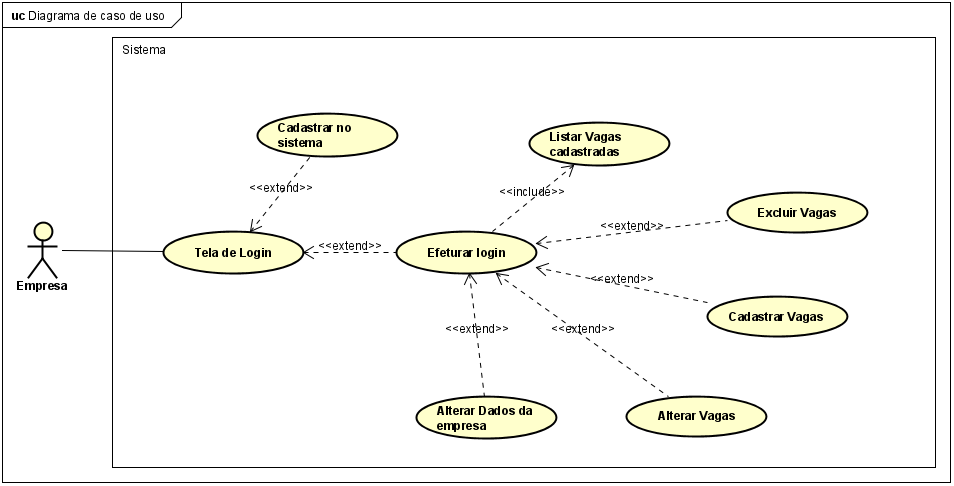
Conforme enfatizado na obra "UML 2 - Uma Abordagem Prática", o Diagrama de Caso de Uso é o mais abstrato, sendo, portanto, o mais flexível e informal entre as representações, utilizado para proporcionar uma visão global do sistema. Seu propósito central é a identificação dos atores que interagirão com 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.

**Figura 02** — Diagrama de Contexto – Visão do Aluno



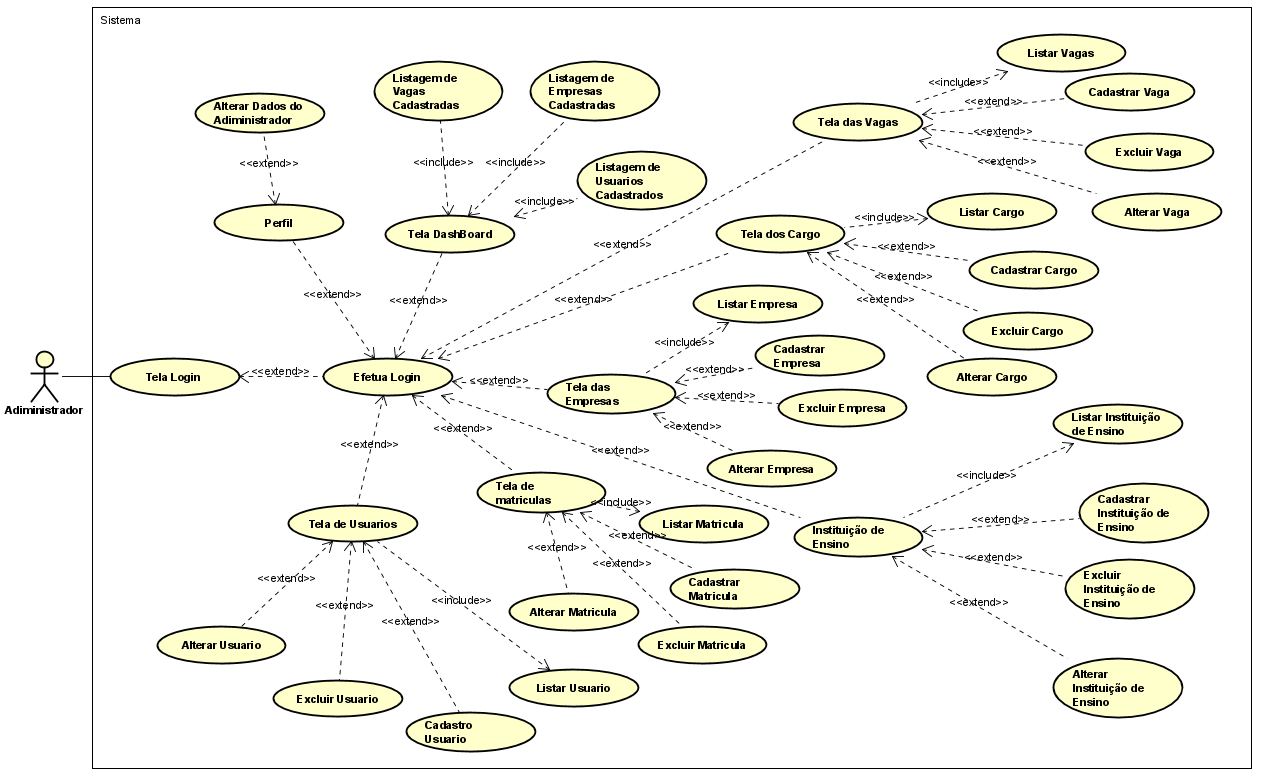
Fonte: Elaborado pelos autores.

**Figura 03** — Diagrama de Contexto – Visão do Empresa



Fonte: Elaborado pelos autores.

**Figura 04** — Diagrama de Contexto – Visão do Administrador



**Figura 05** — Diagrama de Contexto – Visão geral do sistema

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

## 3.5. Diagrama de Casos de uso individuais

Os diagramas apresentados nessa subseção visam as ações específicas do usuário Administrador na.

**Figura 06** — Diagrama de Contexto – Cadastro Aluno

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro 1 – Documentação do caso de uso Cadastrar Aluno

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Cadastrar Aluno** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que (Administrador) adicione novos alunos |
| **Pré-condições** | Preencher todos os campos para concluir o cadastro do aluno |
| **Pós-condições** | O usuário é direcionado a parte desejada do sistema |
| **Fluxo normal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. Usuário solicita o cadastro de um novo aluno |  |
|  | 1. Sistema recebe solicitação e mostra tela com formulário para cadastro |
| 1. Usuário informa dados do aluno e clica no botão “Cadastrar” |  |
|  | 1. Sistema valida os campos adicionados |
|  | 1. Sistema retorna mensagem “Aluno cadastrado com sucesso” |
| **Fluxo alternativo** | |
|  | 1. Sistema valida campos e dados |
|  | * 1. Sistema identifica dados diferente do esperado ou em branco |
|  | * + 1. Sistema envia mensagem “Dados inválidos” |
|  | * + 1. Sistema retorna ao fluxo normal pelo item 2 |

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Consultar Aluno

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro 2 — 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 |  |
|  | 1. Sistema recebe solicitação e mostra tela em lista os alunos cadastrados |
| 1. Usuário seleciona aluno que deseja saber mais detalhes |  |
|  | 1. Sistema valida solicitação |
|  | 1. Sistema carrega tela de consulta. |
| **Fluxo alternativo** | |
|  | 1. Sistema lista solicitação |
|  | * 1. Sistema não consegue conectar com o banco |
|  | * + 1. Sistema envia mensagem “Erro com o banco de dados” |
|  | * + 1. Sistema retorna ao fluxo normal |

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Alterar Aluno

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro 3 – Documentação do caso de uso Alterar Aluno

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

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Excluir Aluno

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro 4 — Documentação do caso de uso Aluno.

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Excluir Aluno** |
| **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 já cadastrado |
| **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 |  |
|  | 1. Sistema abre um dropdown com o botão “Excluir” |
| 1. Usuário aperta o botão “Excluir” |  |
|  | 1. Sistema valida solicitação |
|  | 1. Sistema volta para a tela de listar alunos |
| **Fluxo alternativo** | |
|  | 1. Sistema lista solicitação |
|  | * 1. Sistema não consegue conectar com o banco |
|  | * + 1. Sistema envia mensagem “Erro com o banco de dados” |
|  | * + 1. Sistema retorna ao fluxo normal |

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Cadastrar Curso

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro ?? — Documentação do caso de uso Cadastrar Curso.

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Cadastrar Curso** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que o Administrador cadastre as informações do aluno |

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Cadastrar Curso** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que (Administrador) adicione novos cursos |
| **Pré-condições** | Preencher todos os campos para concluir o cadastro do curso |
| **Pós-condições** | O usuário é direcionado a parte desejada do sistema |
| **Fluxo normal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. Usuário solicita o cadastro de um novo curso |  |
|  | 1. Sistema recebe solicitação e mostra tela com formulário para cadastro |
| 1. Usuário informa dados do curso e clica no botão “Cadastrar” |  |
|  | 1. Sistema valida os campos adicionados |
|  | 1. Sistema retorna mensagem “Curso cadastrado com sucesso” |
| **Fluxo alternativo** | |
|  | 1. Sistema valida campos e dados |
|  | * + 1. Sistema identifica dados diferente do esperado ou em branco |
|  | * + 1. Sistema envia mensagem “Dados inválidos” |
|  | * + 1. Sistema retorna ao fluxo normal pelo item 7 |

Fonte: Elaborado pelos autores

**Figura 05** — Diagrama de Contexto – Consultar Curso

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro ?? — Documentação do caso de uso Consultar Curso.

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Consultar Curso** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que o Administrador consulte as informações do curso |
| **Pré-condições** | Curso cadastrados |
| **Pós-condições** | O usuário é direcionado a parte desejada do sistema |
| **Fluxo normal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. Usuário solicita cursos cadastrados |  |
|  | 1. Sistema recebe solicitação e mostra tela em lista os cursos cadastrados |
| 1. Usuário seleciona curso que deseja saber mais detalhes |  |
|  | 1. Sistema valida solicitação |
|  | 1. Sistema carrega tela de consulta. |
| **Fluxo alternativo** | |
|  | 1. Sistema lista solicitação |
|  | * + 1. Sistema não consegue conectar com o banco |
|  | * + 1. Sistema envia mensagem “Erro com o banco de dados” |
|  | * + 1. Sistema retorna ao fluxo normal |

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Alterar Curso

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro ?? – Documentação do caso de uso Alterar Curso

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Alterar Curso** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que o Administrador altere as informações do curso |
| **Pré-condições** | Curso já cadastrado |
| **Pós-condições** | O usuário é direcionado a parte desejada do sistema |
| **Fluxo normal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. Usuário solicita alterar um curso |  |
|  | 1. Sistema recebe solicitação e mostra a tela com os dados já cadastrados |
| 1. Após alterar os dados do curso o usuário aperta no botão “Salvar alterações” |  |
|  | 1. Sistema valida solicitação e altera dados. |
|  | 1. Sistema envia mensagem “Alterado com sucesso” |
|  | 1. Sistema retorna para a tela com a tabela de listagem. |
| **Fluxo alternativo** | |
|  | 1. Sistema valida solicitação e altera dados. |
|  | * 1. Sistema falha ao se comunicar com a controller |
|  | * 1. Sistema envia a mensagem de erro |
|  | * 1. Sistema retorna ao fluxo normal pelo item 10 |

Fonte: Elaborado pelos autores.

**Figura 05** — Diagrama de Contexto – Excluir Curso

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Quadro ?? — Documentação do caso de uso Curso.

|  |  |
| --- | --- |
| **Documentação** | |
| **Nome do caso de uso** | **Excluir Curso** |
| **Ator principal** | Usuário (Administrador) |
| **Resumo** | Esse caso de uso descreve as ações necessárias para que o Administrador apague as informações do curso |
| **Pré-condições** | Curso já cadastrado |
| **Pós-condições** | O usuário é direcionado a parte desejada do sistema |
| **Fluxo normal** | |
| **Ações do ator** | **Ações do sistema** |
| 1. Usuário solicita excluir um curso |  |
|  | 1. Sistema abre um dropdown com o botão “Excluir” |
| 1. Usuário aperta o botão “Excluir” |  |
|  | 1. Sistema valida solicitação |
|  | 1. Sistema volta para a tela de listar cursos |
| **Fluxo alternativo** | |
|  | 1. Sistema lista solicitação |
|  | * 1. Sistema não consegue conectar com o banco |
|  | * + 1. Sistema envia mensagem “Erro com o banco de dados” |
|  | * + 1. Sistema retorna ao fluxo normal |

Fonte: Elaborado pelos autores.

## 3.6. 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, 2018).

Figura 5 — Diagrama de Sequência – Cadastro Aluno

Diagrama

Descrição gerada automaticamente

Fonte: Desenvolvidos pelos autores.

Gráfico de caixa estreita

Descrição gerada automaticamenteFigura 6 — Diagrama de Sequência – Altera Aluno

Fonte: Desenvolvidos pelos autores.

Diagrama

Descrição gerada automaticamenteFigura 7 — Diagrama de Sequência – Exclui Aluno

Fonte: Desenvolvidos pelos autores.

Figura 8 — Diagrama de Sequência – Lista Aluno

Diagrama, Gráfico de caixa estreita

Descrição gerada automaticamente

Fonte: Desenvolvidos pelos autores.

Figura 9 — Diagrama de Sequência – Cadastro Curso

Diagrama

Descrição gerada automaticamente

Fonte: Desenvolvidos pelos autores.

Figura 10 — Diagrama de Sequência – Exclui Curso

Diagrama, Desenho técnico

Descrição gerada automaticamente

Fonte: Desenvolvidos pelos autores.

Figura 11 — Diagrama de Sequência – Altera Curso

Diagrama

Descrição gerada automaticamente com confiança baixa

Fonte: Desenvolvidos pelos autores.

Figura 12 — Diagrama de Sequência – Lista Curso

Diagrama

Descrição gerada automaticamente

Fonte: Desenvolvidos pelos autores.

# DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º semestre)

## Descrição de cenário

Conforme escrito no livro "O Guia para Projetar UX" de Russ Unger e Carolyn Chandler, 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.

Figura 13 – Imagem do primeiro cenário



Fonte: Elaborado pelos autores.

Figura 14 – Imagem do segundo cenário

Fonte: Elaborado pelos autores.

## Descrição de personas

De acordo com "O Guia para Projetar UX" de Russ Unger e Carolyn Chandler, 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.

 Figura 15 – Imagem da segunda persona

Fonte: Elaborado pelos autores.

 Figura 16 – Imagem da segunda persona

Fonte: Elaborado pelos autores.

## Protótipos de tela

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". Os protótipos de tela, são representações tangíveis e interativas da proposta de design, permitindo a exploração, comunicação e refinamento de ideias".

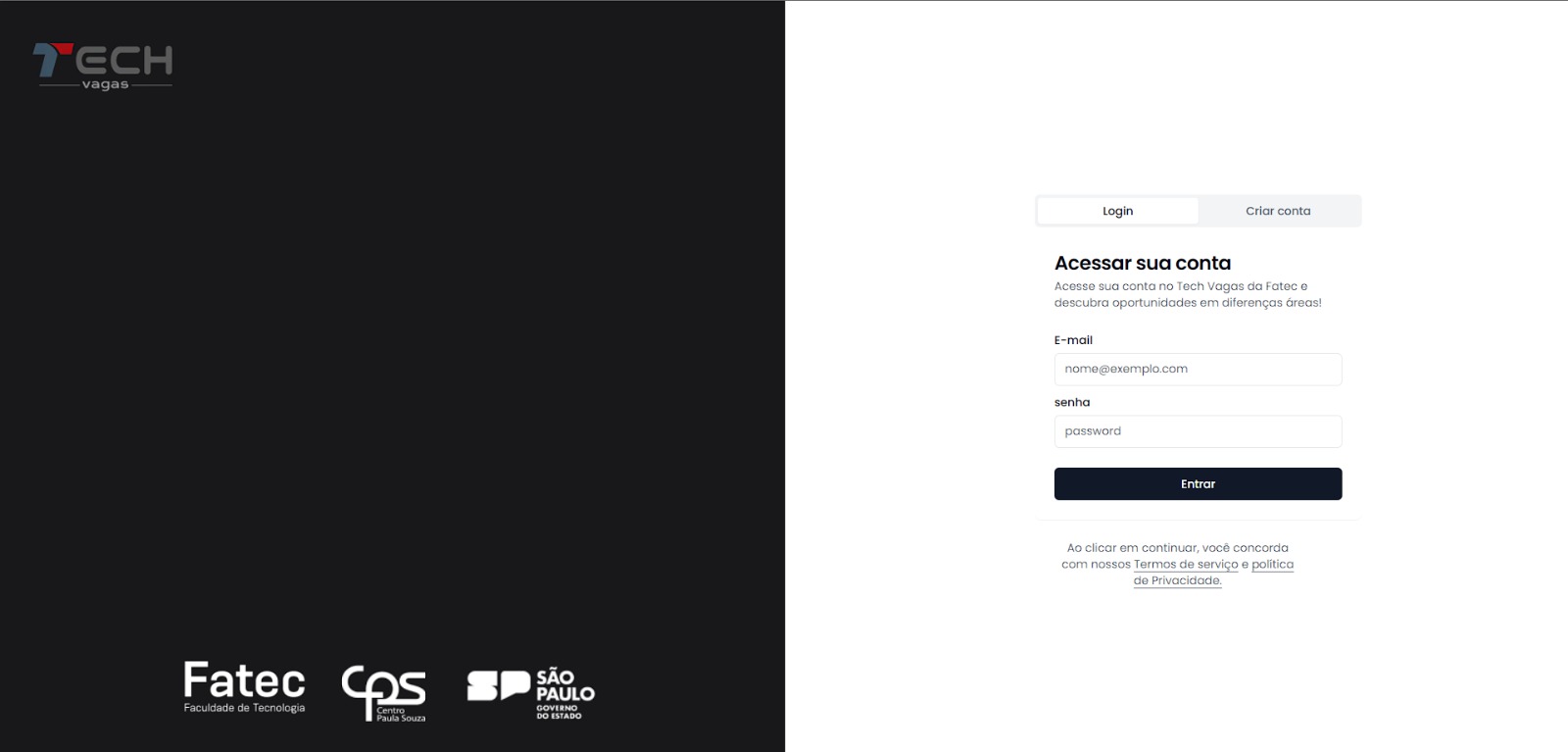
Assim, os modelos iniciais de tela não só promovem a comunicação efetiva entre as equipes de design e desenvolvimento, como também têm um papel crucial na detecção antecipada de adaptações necessárias, resultando em economia de tempo e recursos.

* Tela de cadastro ao usuário



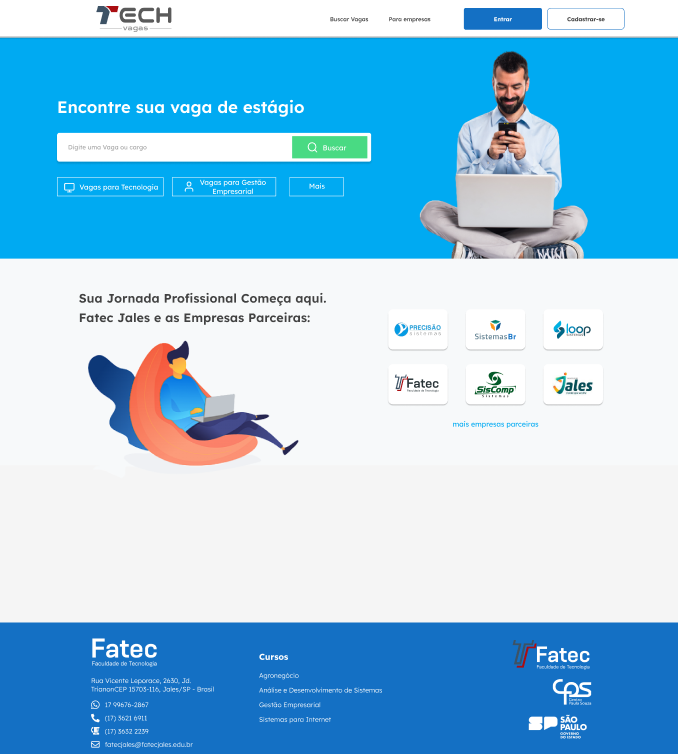
Fonte: Elaborado pelos autores.

* Está tela tem como finalidade realizar o cadastro do usuário, utilizando suas informações.
* Tela de login do usuário



Fonte: Elaborado pelos autores.

* Tela Inicial



Fonte: Elaborado pelos autores.

* Irá mostrar algumas vagas em destaque, as parcerias com outras empresas e uma barra de pesquisa caso queira pesquisar alguma vaga em especifico.

Interface gráfica do usuário, Aplicativo, Email

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

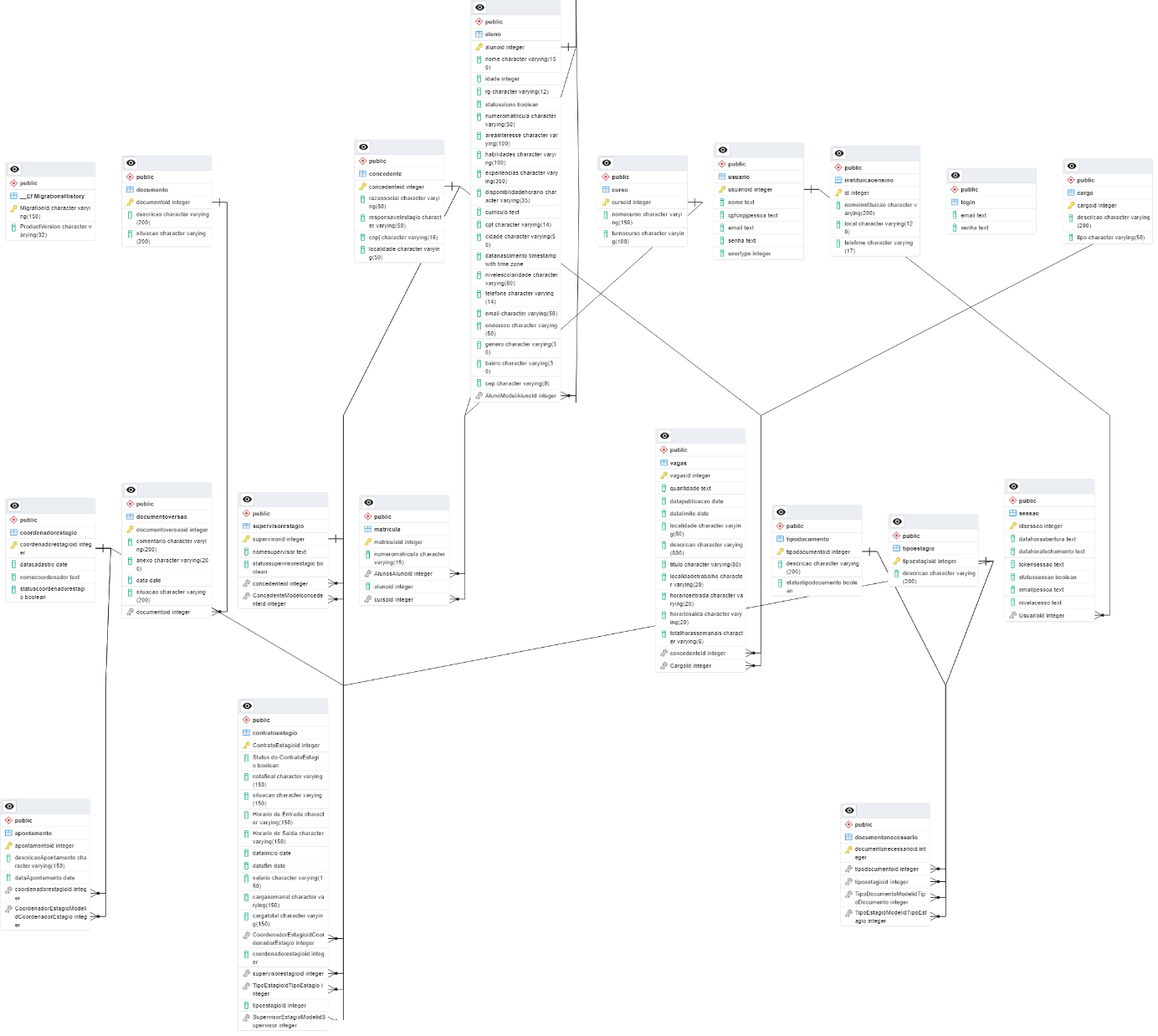
* Tela de Cadastro de empresas do próprio sistema TECHVAGAS

# BANCO DE DADOS

## Modelo Entidade Relacionamento

O Modelo Entidade-Relacionamento (MER), conforme abordado no livro "Banco de Dados: Projeto e Implementação", proporciona uma representação visual das entidades (representações de objetos do mundo real), atributos (características de uma entidade) e relacionamentos (conexões entre diferentes entidades) que compõem o sistema de armazenamento de dados. Trata-se de uma ferramenta que facilita a compreensão ao capturar relações complexas e estruturar os dados de forma acessível e organizada.

Figura 9 – Mapeamento do Objeto Relacional



Fonte: Elaborado pelos autores

## Script das tabelas

Script das tabelas, conforme abordado no livro “Banco de dados: Projeto e Implementação”, consistem em diretrizes ou conjuntos de comandos redigidos numa linguagem de consulta estruturada (SQL) com o propósito de criar, modificar ou remover tabelas em um banco de dados. Esses conjuntos de instruções são empregados para estabelecer a estrutura de um banco de dados relacional, abrangendo definições de tabelas, campos e restrições.

Quadro 32 – 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,  numeromatricula 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 33 – 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 34 – 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 35 – 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,  CONSTRAINT "PK\_documento" PRIMARY KEY (documentoid)  ); |

Fonte: Elaborado pelos autores.

Quadro 36 – Script SQL – Tabela TipoDocumento

|  |
| --- |
| CREATE TABLE tipodocumento (  tipodocumentoid integer GENERATED BY DEFAULT AS IDENTITY,  descricao character varying(200) NOT NULL,  statustipodocumento boolean NOT NULL,  CONSTRAINT "PK\_tipodocumento" PRIMARY KEY (tipodocumentoid)  ); |

Fonte: Elaborado pelos autores.

Quadro 37 – Script SQL – Tabela TipoEstagio

|  |
| --- |
| 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 38 – 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 39 – Script SQL – Tabela DocumentoVersao

|  |
| --- |
| CREATE TABLE documentoversao (  documentoversaoid integer GENERATED BY DEFAULT AS IDENTITY,  comentario character varying(200) NOT NULL,  anexo character varying(200) NOT NULL,  date date NOT NULL,  situacao character varying(200) NOT NULL,  documentoid integer NOT NULL,  CONSTRAINT "PK\_documentoversao" PRIMARY KEY (documentoversaoid),  CONSTRAINT "FK\_documentoversao\_documento\_documentoid" FOREIGN KEY (documentoid) REFERENCES documento (documentoid) ON DELETE CASCADE  ); |

Fonte: Elaborado pelos autores.

Quadro 40 – 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,  vagasid integer NOT NULL,  CONSTRAINT "PK\_cargo" PRIMARY KEY (cargoid),  ); |

Fonte: Elaborado pelos autores.

Quadro 41 – Script SQL – Tabela CoordenadorEstagio

|  |
| --- |
| 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 42 – Script SQL – Tabela Login

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

Fonte: Elaborado pelos autores.

Quadro 43 – Script SQL – Tabela InstituicaoEnsino

|  |
| --- |
| CREATE TABLE instituicaoensino (  instituicaoensinoid integer GENERATED BY DEFAULT AS IDENTITY,  nomeinstituicao character varying(200) NOT NULL,  localinstituicao character varying(120) NOT NULL,  telefoneinstituicao character varying(17) NOT NULL,  CONSTRAINT "PK\_instituicaoensino" PRIMARY KEY (instituicaoensinoid),  ); |

Fonte: Elaborado pelos autores.

Quadro 44 – Script SQL – Tabela Usuario

|  |
| --- |
| CREATE TABLE usuario (  usuarioid integer GENERATED BY DEFAULT AS IDENTITY,  nome text NOT NULL,  cpfcnpjpessoa 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 45– Script SQL – Tabela SupervisorEstagio

|  |
| --- |
| CREATE TABLE supervisorestagio (  supervisorid integer GENERATED BY DEFAULT AS IDENTITY,  nomesupervisor boolean NOT NULL,  concedenteid integer NOT NULL,  “ConcedenteModelconcedenteId” integer NOT NULL,  senha text NOT NULL,  usertype integer NOT 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 46– 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 47– Script SQL – Tabela Matricula

|  |
| --- |
| CREATE TABLE matricula (  matriculaid integer GENERATED BY DEFAULT AS IDENTITY,  numeromatricula character varying(15) NOT NULL,  "AlunosAlunoId" integer NOT 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  ); |

Fonte: Elaborado pelos autores.

Quadro 48– Script SQL – Tabela Matricula

|  |
| --- |
| CREATE TABLE matricula (  matriculaid integer GENERATED BY DEFAULT AS IDENTITY,  numeromatricula character varying(15) NOT NULL,  "AlunosAlunoId" integer NOT 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  ); |

Fonte: Elaborado pelos autores.

Quadro 49– Script SQL – Tabela DocumentoNecessario

|  |
| --- |
| CREATE TABLE documentonecessario (  documentonecessarioid integer GENERATED BY DEFAULT AS IDENTITY,  tipodocumentoid integer NOT NULL,  tipoestagioid integer NOT NULL,  "TipoDocumentoModelidTipoDocumento" integer NOT NULL,  "TipoEstagioModelidTipoEstagio" integer NOT 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 50– 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 51– Script SQL – Tabela ContratoEstagio

|  |
| --- |
| 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,  coordenadorestagioid integer NOT NULL,  supervisorestagioid 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 (coordenadorestagioid),  CONSTRAINT "FK\_contratoestagio\_supervisorestagio\_SupervisorEstagioModelidS~" FOREIGN KEY ("SupervisorEstagioModelidSupervisor") REFERENCES supervisorestagio (supervisorid),  CONSTRAINT "FK\_contratoestagio\_supervisorestagio\_supervisorestagioid" FOREIGN KEY (supervisorestagioid) REFERENCES supervisorestagio (supervisorid) ON DELETE CASCADE,  CONSTRAINT "FK\_contratoestagio\_tipoestagio\_TipoEstagioidTipoEstagio" FOREIGN KEY ("TipoEstagioidTipoEstagio") REFERENCES tipoestagio (tipoestagioid)  ); |

Fonte: Elaborado pelos autores.

Quadro 52 – Script SQL – Tabela Candidato

|  |
| --- |
| CREATE TABLE Candidato (  idCandidato integer GENERATED BY DEFAULT AS IDENTITY,  dataInscricao date NOT NULL,  situacao text NOT NULL,  ); |

Fonte: Elaborado pelos autores.

# ARQUITETURA DE SOFTWARE

## 6.1 Arquitetura de desenvolvimento

A arquitetura de software representa a estrutura essencial que orienta o desenvolvimento de uma aplicação, traduzindo os requisitos do sistema em uma representação tangível. No contexto da TechVagas, a estrutura de software constitui o alicerce essencial do sistema, proporcionando uma perspectiva estruturada e unificada da interação entre os elementos. Conforme destacado por Ralph Johnson, coautor do livro "Design Patterns: Elements of Reusable Object-Oriented Software", a arquitetura de software desempenha o papel de "espinha dorsal" que sustenta a aplicação, delineando as conexões entre os distintos módulos e garantindo uma estrutura sólida e resistente.

A necessidade de uma arquitetura sólida na TechVagas vai além da simples organização do código. De acordo com Martin Fowler, especialista em arquitetura de software, "a relevância da arquitetura é evidente ao lidar com a complexidade do desenvolvimento de software". No âmbito da TechVagas, a complexidade do sistema exige uma arquitetura meticulosamente estruturada, não apenas para simplificar a implementação atual, mas também para viabilizar a manutenção subsequente e a expansão contínua do projeto.

Portanto, a arquitetura de software na TechVagas transcende a formalidade técnica, sendo uma estratégia essencial para garantir clareza na concepção, flexibilidade na evolução e capacidade de resposta às necessidades em constante mudança.

## 6.1.1 Linguagem de Programação C#

Figura 2 – Imagem da Logo do C#



Fonte: Cwork Sistemas (2018)

Acima podemos ver a logo da linguagem implementada no TechVagas, o C# desempenha um papel crucial na construção de uma API robusta e eficiente. Como destacado por Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft, "C# é projetado para ser uma linguagem de programação simples, moderna e orientada a objetos". A escolha dessa linguagem casa perfeitamente com a necessidade do projeto e com o que foi aprendido no curso, com o auxílio do docente capacitado, fica melhor o suporte para o grupo com alguma dúvida, tendo também como uma das principais características a fácil manutenção.

Martin Fowler, especialista em padrões de arquitetura de software, ressalta que "C# é particularmente adequado para a implementação de arquiteturas em camadas". Ao utilizar essas camadas dentro do TechVagas temos como vantagem coesão e flexibilidade.

Além disso, Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI, destaca a integração do C# com o ecossistema .NET, por oferecer acesso a uma extensa variedade de bibliotecas e ferramentas, essa integração eficiente se mostra crucial para assegurar uma comunicação fluida entre as diversas camadas da aplicação e o banco de dados do TechVagas. Esse processo cria uma base sólida para o desenvolvimento contínuo e sustentável da API.

## 6.1.2 Visual Studio

Figura 3 – Imagem da Logo do Visual Studio



Fonte: 1000 logos (2023)

Acima vemos a logo do Visual Studio que foi escolhido como ambiente de desenvolvimento para a implementação da API em C# deste, respaldada por sua reputação consolidada e recursos abrangentes. Como enfatiza Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft, o Visual Studio é reconhecido por sua robustez e conjunto de ferramentas integradas, proporcionando um ambiente de desenvolvimento completo para a criação de aplicativos empresariais de alta qualidade, oferendo funcionalidades avançadas para a implementação da arquitetura em camadas, deixando melhor a implementação e entendimento do código fonte, e isso é fundamental para as necessidades do ESTAGIOTECH.

A interligação entre o Visual Studio e o ecossistema .NET é ressaltada por especialistas como Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI. Essa integração simplifica significativamente o processo de desenvolvimento em C#, facilitando uma comunicação contínua entre as variadas camadas da aplicação e o banco de dados do projeto.

Assim, ao escolher o Visual Studio como o ambiente principal para o desenvolvimento da API em C#, a equipe do ESTAGIOTECH almeja não apenas uma ferramenta robusta e estabelecida, mas também um ambiente que simplifique a adoção de boas práticas de arquitetura e promova um fluxo de trabalho eficiente para a concepção de uma aplicação escalável e de excelência em qualidade.

## 6.1.3 Visual Studio Code

Figura 4 – Imagem da Logo do Visual Studio



Fonte: Medium (2019)

A escolha do Visual Studio Code como ambiente de desenvolvimento para o TechVagas reflete a preferência por uma ferramenta versátil e amplamente adotada na comunidade de desenvolvimento. Martin Fowler, um autor de renome, ressalta que o Visual Studio Code é reconhecido por sua leveza, simplicidade e capacidade de expansão, fornecendo um ambiente de codificação eficaz e focado no desenvolvedor. Com uma interface intuitiva e suporte sólido para múltiplas linguagens de programação, o Visual Studio Code se destaca como uma escolha versátil.

## 6.1.4 PostgreSQL

Figura 5 – Logo do PostgreSQL



Fonte: Medium (2022)

A escolha do PGAdmin como ferramenta de administração do PostgreSQL se baseia em sua interface intuitiva e na amplitude de recursos oferecidos. Segundo Joe Conway, membro sênior da comunidade PostgreSQL, "o PGAdmin é uma ferramenta essencial para administrar o PostgreSQL, permitindo visualizar e manipular dados, criar consultas complexas e gerenciar esquemas de banco de dados de maneira eficaz."

A opção pelo PostgreSQL e pelo PGAdmin no TechVagas está fundamentada na necessidade de um sólido sistema de gerenciamento de banco de dados e uma ferramenta intuitiva para uma administração eficiente de dados. Esse conjunto, endossado por especialistas reconhecidos, garante a confiabilidade e a eficácia na administração do banco de dados, contribuindo para a estabilidade e integridade do sistema como um todo.

A administração eficaz do banco de dados, liderada pelo PostgreSQL e facilitada pelo PGAdmin, ressalta a importância de escolhas estratégicas em todas as etapas do desenvolvimento. Ao seguir os princípios de arquitetura delineados por especialistas renomados, o TechVagas assegura uma base sólida para a expansão futura, a manutenção eficaz e a sustentabilidade do sistema. Nesse contexto, a arquitetura de software não é apenas uma camada técnica, mas sim um alicerce vital que orienta o projeto rumo à eficiência, escalabilidade e sucesso duradouro.

## 6.1.5 Linguagem de programação

Figura 5 – Logo do React

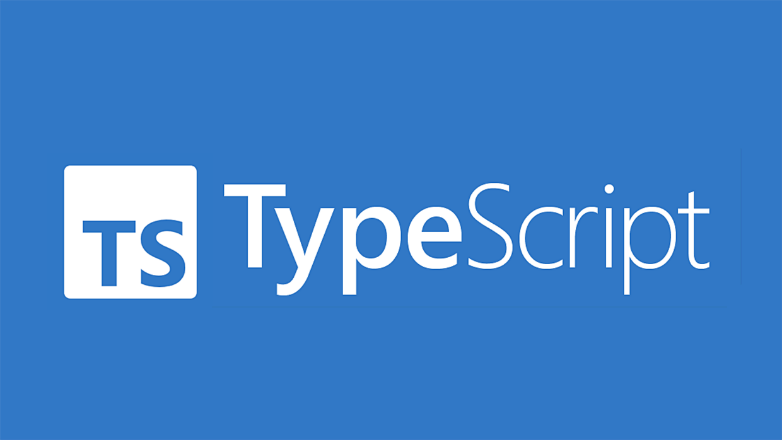
Logotipo, nome da empresa

Descrição gerada automaticamente

Fonte: Medium (2022)

Na imagem acima, observamos o logotipo do React, um framework de JavaScript desenvolvida pelo Facebook com o propósito de simplificar a criação de interfaces de usuário dinâmicas e eficientes. Baseando-se no conceito de componentes reutilizáveis, o React facilita a construção de interfaces interativas, tornando-as mais escaláveis e simples de manter. No âmbito do projeto TechVagas, o React foi escolhido para a implementação da interface do usuário, proporcionando uma experiência de usuário responsiva e atraente. Ao permitir a criação de componentes independentes, o React promove a modularidade do código, facilitando a manutenção e a adição de novos recursos ao longo do tempo.

Figura 6 – Logo do typescript



Fonte: Medium (2022)

O TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. Ela é um superset do JavaScript, o que significa que todo código JavaScript válido também é código TypeScript, mas o TypeScript oferece recursos adicionais e uma camada adicional de tipagem estática que o JavaScript puro não possui.

**Tipagem Estática:**

* O TypeScript introduz a capacidade de declarar tipos de variáveis, parâmetros de função, e retorno de função, entre outros.
* Essa tipagem estática ajuda a identificar erros durante o desenvolvimento, antes mesmo de o código ser executado.

**Superset do JavaScript:**

* O TypeScript é construído sobre o JavaScript, o que significa que qualquer código JavaScript válido pode ser gradualmente migrado para TypeScript.
* Isso facilita a adoção do TypeScript em projetos existentes.

**Compilação para JavaScript:**

* O código TypeScript é compilado para JavaScript, permitindo que seja executado em qualquer ambiente que suporte JavaScript.

**Orientação a Objetos:**

* TypeScript suporta programação orientada a objetos, incluindo classes, interfaces, herança e polimorfismo.
* Esses recursos tornam o desenvolvimento mais estruturado e modular.

**Módulos e Namespaces:**

* TypeScript oferece suporte a módulos e namespaces para organizar e estruturar o código de maneira mais eficiente.

**Integração com Ferramentas de Desenvolvimento:**

* O TypeScript é frequentemente utilizado em conjunto com ferramentas modernas de desenvolvimento, como o Visual Studio Code, proporcionando uma experiência de desenvolvimento mais robusta.

**Melhoria na Manutenção de Código:**

* A tipagem estática e a estruturação de código facilitam a manutenção e evolução do software ao longo do tempo.

**Comunidade Ativa:**

* O TypeScript tem uma comunidade ativa e suporte contínuo da Microsoft, o que significa que há uma grande quantidade de recursos, documentação e bibliotecas disponíveis.

Figura 7 – Logo do shadcn



Fonte: Medium (2023)

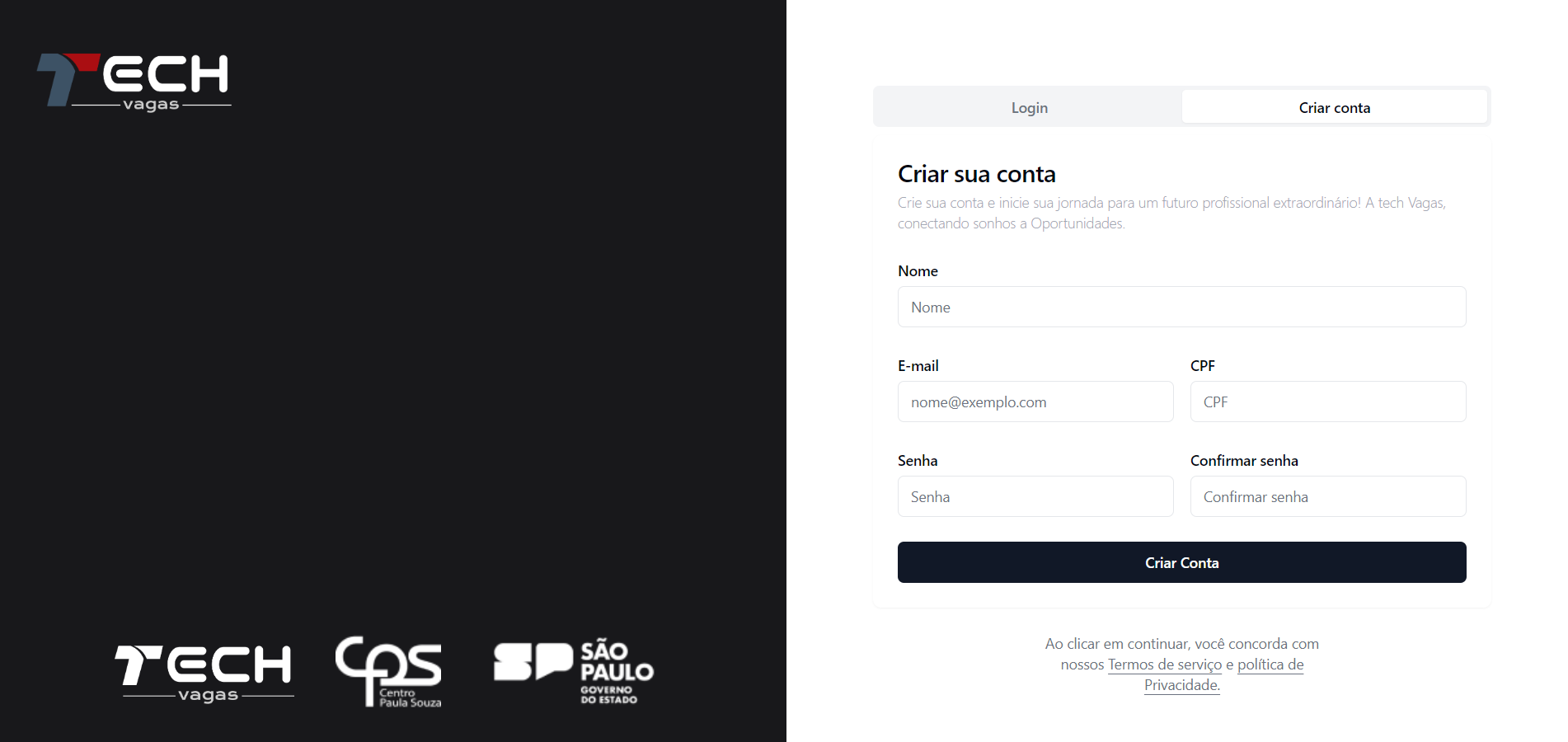
O Shadcn é uma robusta coleção de componentes reutilizáveis prontos para serem integrados aos seus aplicativos. Esses componentes foram desenvolvidos utilizando as tecnologias Radix UI e Tailwind CSS, garantindo um alto padrão de qualidade e flexibilidade.

Simples de implementar, basta escolher os componentes necessários, copiar o código correspondente e incorporá-lo ao seu projeto. A partir daí, você pode personalizar conforme suas especificações, adaptando os elementos de acordo com as necessidades específicas da sua aplicação. Essa abordagem simplificada agiliza o desenvolvimento, permitindo que você concentre seus esforços na criação de uma experiência única e eficiente para os usuários.

## 6.2 Telas do sistema

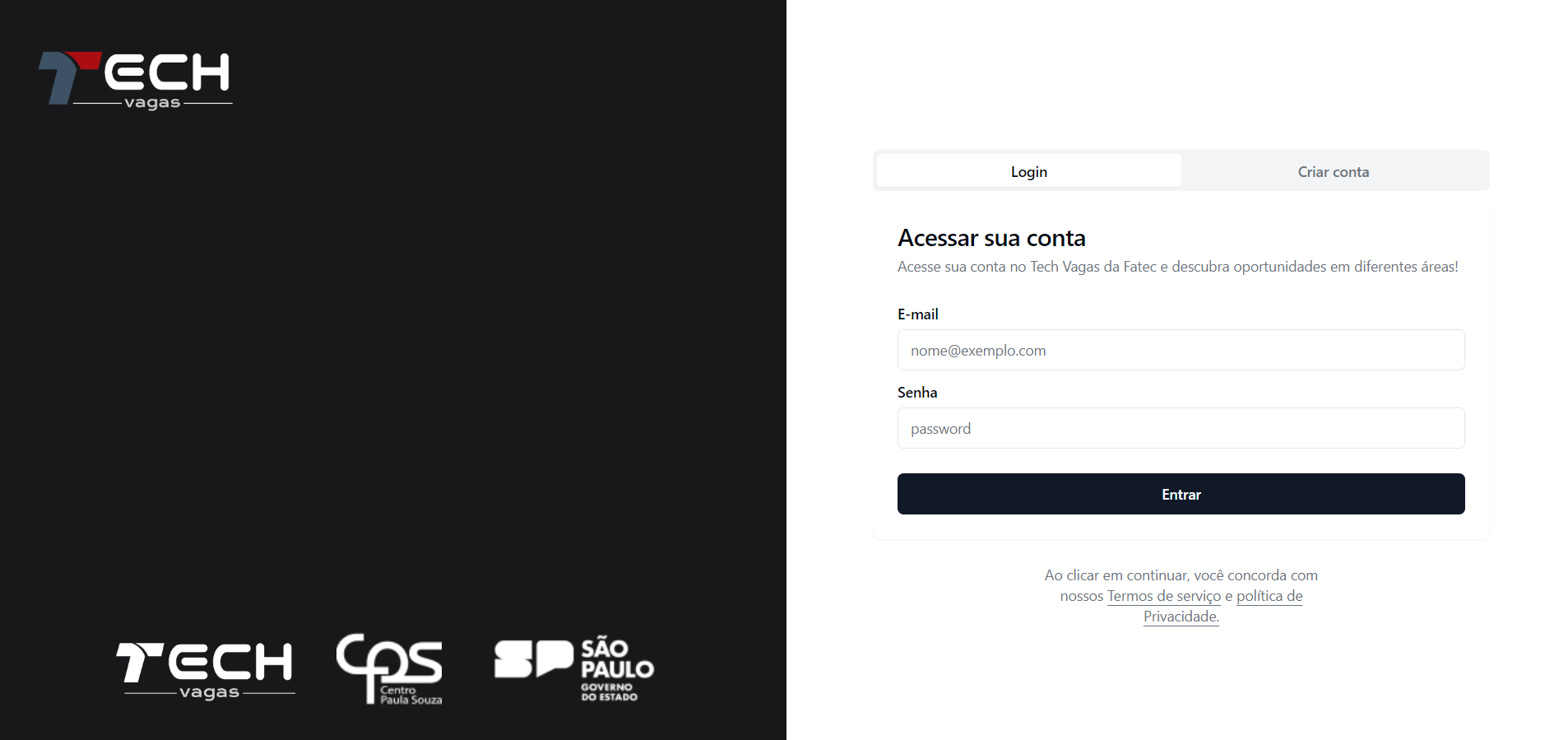
As telas de um sistema representam as interfaces visuais que possibilitam a interação dos usuários com as funcionalidades e informações disponíveis. Don Norman, um pioneiro em design centrado no usuário, destaca que "a interface é o produto". Nessa perspectiva, as telas desempenham um papel fundamental no projeto, sendo o ponto de contato para que usuários e administradores interajam com as diversas funcionalidades do sistema.

**Figura** – Imagem da Tela de cadastro



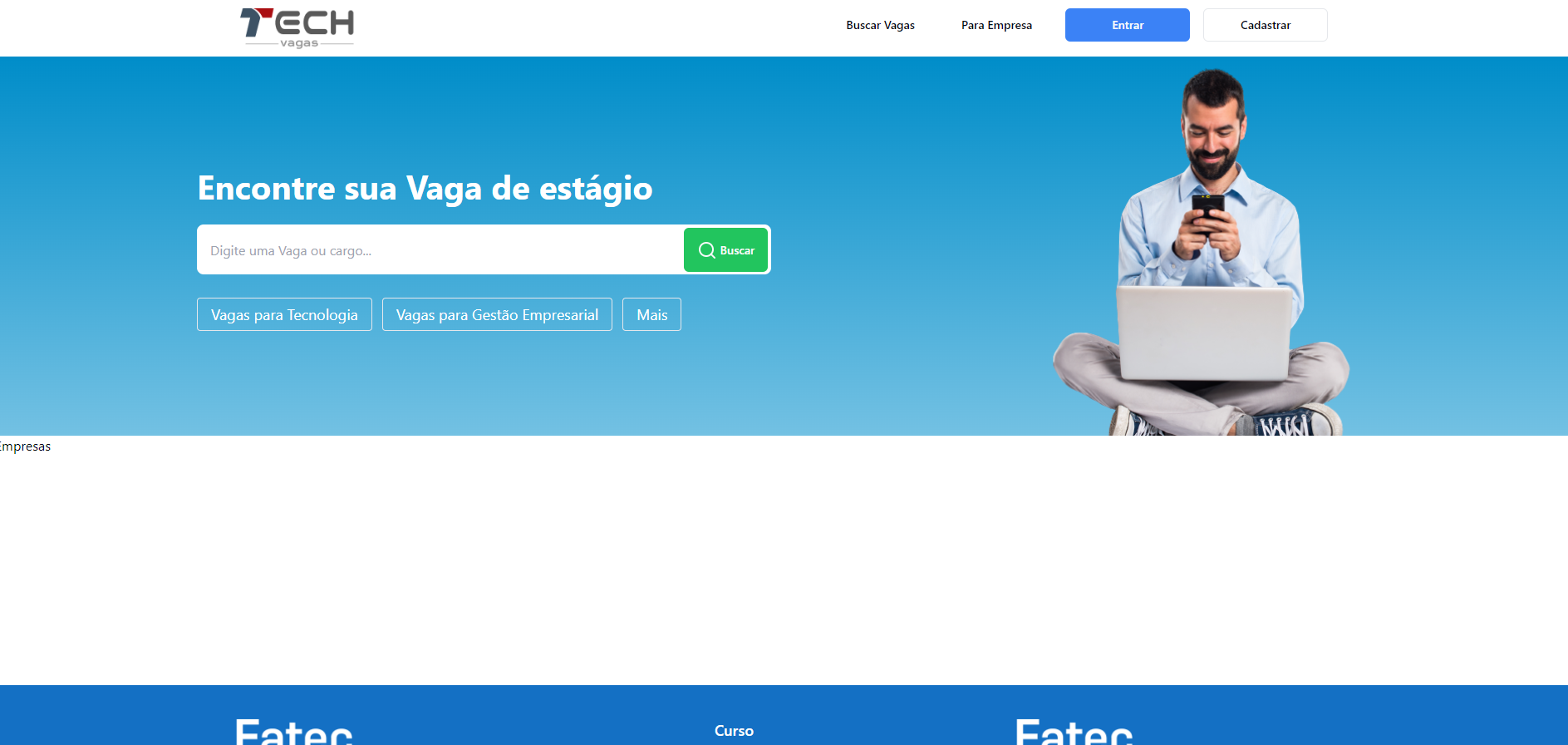
Fonte: Elaborado pelos autores

**Figura** – Imagem da Tela de login



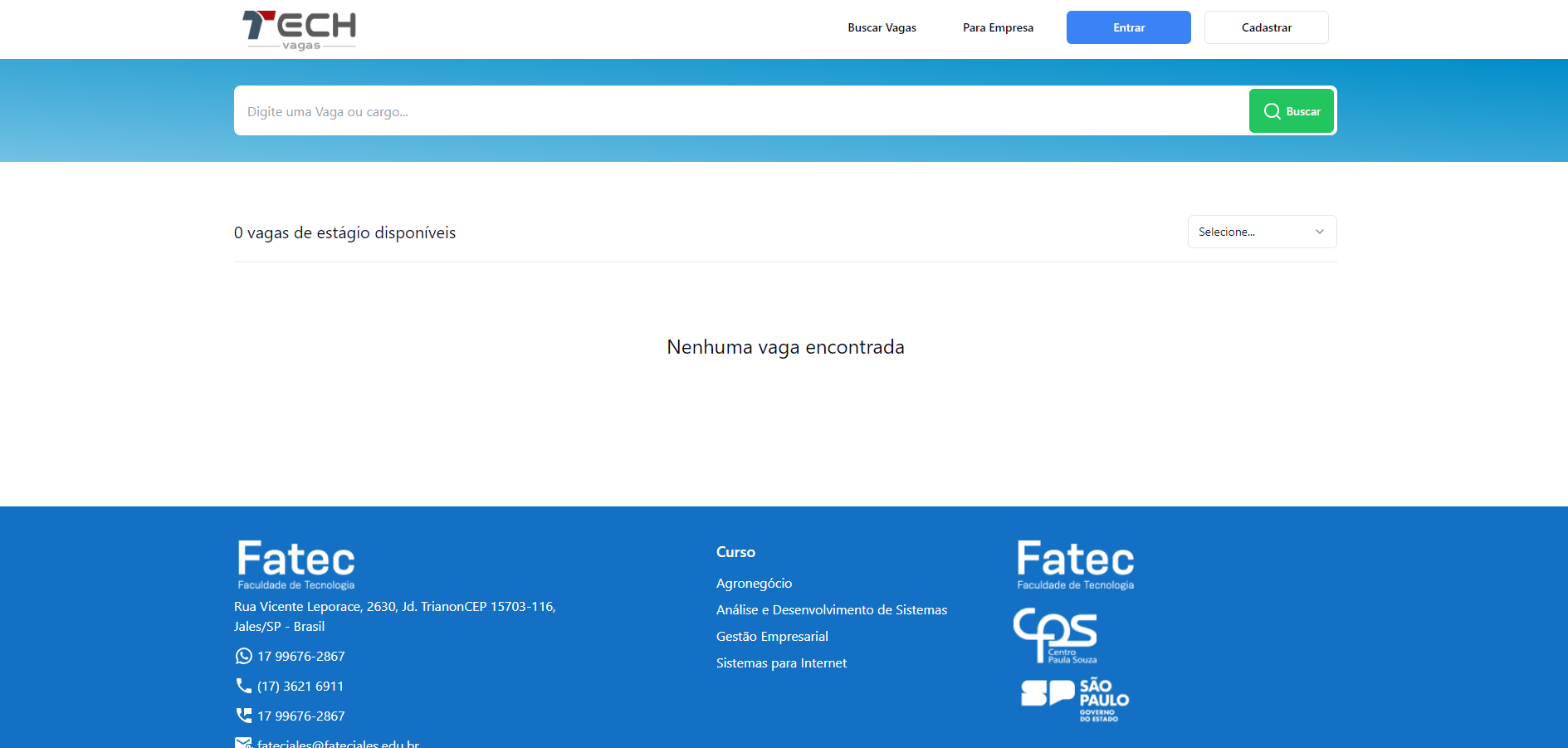
Fonte: Elaborado pelos autores

**Figura** – Imagem da Tela de inicio



Fonte: Elaborado pelos autores

**Figura** – Imagem da Tela de listagem de vagas



Fonte: Elaborado pelos autores

# CONCLUSÃO

Em síntese, o desenvolvimento e implementação do sistema de gestão de vagas para os alunos da Fatec Jales representam um avanço significativo na otimização dos recursos educacionais e na melhoria da experiência acadêmica. Através deste projeto, foi possível constatar não apenas a necessidade premente de uma solução eficaz para a distribuição de vagas, mas também a viabilidade técnica e operacional para a sua implementação.

Ao longo deste estudo, identificamos desafios enfrentados pelos estudantes na obtenção de disciplinas desejadas, evidenciando a importância de um sistema que promova a equidade e a eficiência no acesso aos cursos. A análise das tecnologias utilizadas e a revisão da literatura destacam que as soluções propostas estão alinhadas com as melhores práticas e tendências contemporâneas em gestão acadêmica.

Os resultados obtidos nas fases de teste e implementação reforçam a eficácia do sistema, demonstrando que não apenas atende às necessidades específicas da Fatec Jales, mas também cria um modelo replicável para outras instituições de ensino superior. A aceitação positiva por parte dos alunos e a redução perceptível de problemas relacionados à alocação de vagas corroboram a relevância prática deste projeto.

Portanto, com base nos resultados apresentados, concluímos que o sistema de gestão de vagas proposto não apenas atende às necessidades imediatas da Fatec Jales, mas também representa uma solução viável e eficaz para melhorar a qualidade do processo de alocação de disciplinas. Recomenda-se a implementação plena do sistema, aproveitando as lições aprendidas durante o desenvolvimento, e sugere-se que outras instituições de ensino considerem a adaptação desse modelo para otimizar seus próprios processos acadêmicos. Este trabalho contribui não apenas para o avanço tecnológico na gestão educacional, mas também para o aprimoramento contínuo da experiência do aluno no ambiente acadêmico.

# 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.