**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL**

**SENAC**

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADOR III:** [**DESENVOLVIMENTO DE SISTEMAS DINÂMICOS BASEADOS NA WEB**](https://senacsp.blackboard.com/webapps/blackboard/execute/courseMain?course_id=_251173_1) **TEMA: Modelagem de um Sistema Orientado a Objetos**

Integrantes do grupo:

Arthur Barbosa Lopes

Cauã Vieira Almeida

Leonardo Henrique Maciel Ferreira

Rebeca de Vasconcelos Carício

Sofia Freire de Andrade Clark

**EAD - ENSINO À DISTÂNCIA - 2024**

Integrantes do grupo:

Arthur Barbosa Lopes

Cauã Vieira Almeida

Leonardo Henrique Maciel Ferreira

Rebeca de Vasconcelos Carício

Sofia Freire de Andrade Clark

**PROJETO INTEGRADOR III:** [**DESENVOLVIMENTO DE SISTEMAS DINÂMICOS BASEADOS NA WEB**](https://senacsp.blackboard.com/webapps/blackboard/execute/courseMain?course_id=_251173_1)

Enoque Felipe dos Santos Leal

**Modelagem de um Sistema Orientado a Objetos**

**EAD - ENSINO À DISTÂNCIA - 2024**

**RESUMO**

Este trabalho tem como objetivo modelar um sistema orientado a objetos para a gestão de dados em uma grande universidade, utilizando a Linguagem Unificada de Modelagem (UML). O foco da modelagem está no cadastro de diferentes tipos de pessoas que interagem com o sistema, como alunos, professores, fornecedores, pessoas físicas e jurídicas. Para isso, foram elaborados diagramas de casos de uso, que representam visualmente os processos de interação dos atores com o sistema, além de cenários que descrevem fluxos principais e alternativos. Também foi desenvolvido um diagrama de classes, que detalha as principais entidades do sistema e seus relacionamentos.

Palavras-chave: UML, modelagem orientada a objetos, diagrama de classe, sistema de gestão universitária

**SUMÁRIO**

[1. INTRODUÇÃO - Diagrama de caso de uso 5](#_heading=h.gjdgxs)

[2. DESCRIÇÃO DOS CENÁRIOS DOS CASOS DE USO 6](#_heading=h.2et92p0)

[2.1. Cenário Principal: 6](#_heading=h.tyjcwt)

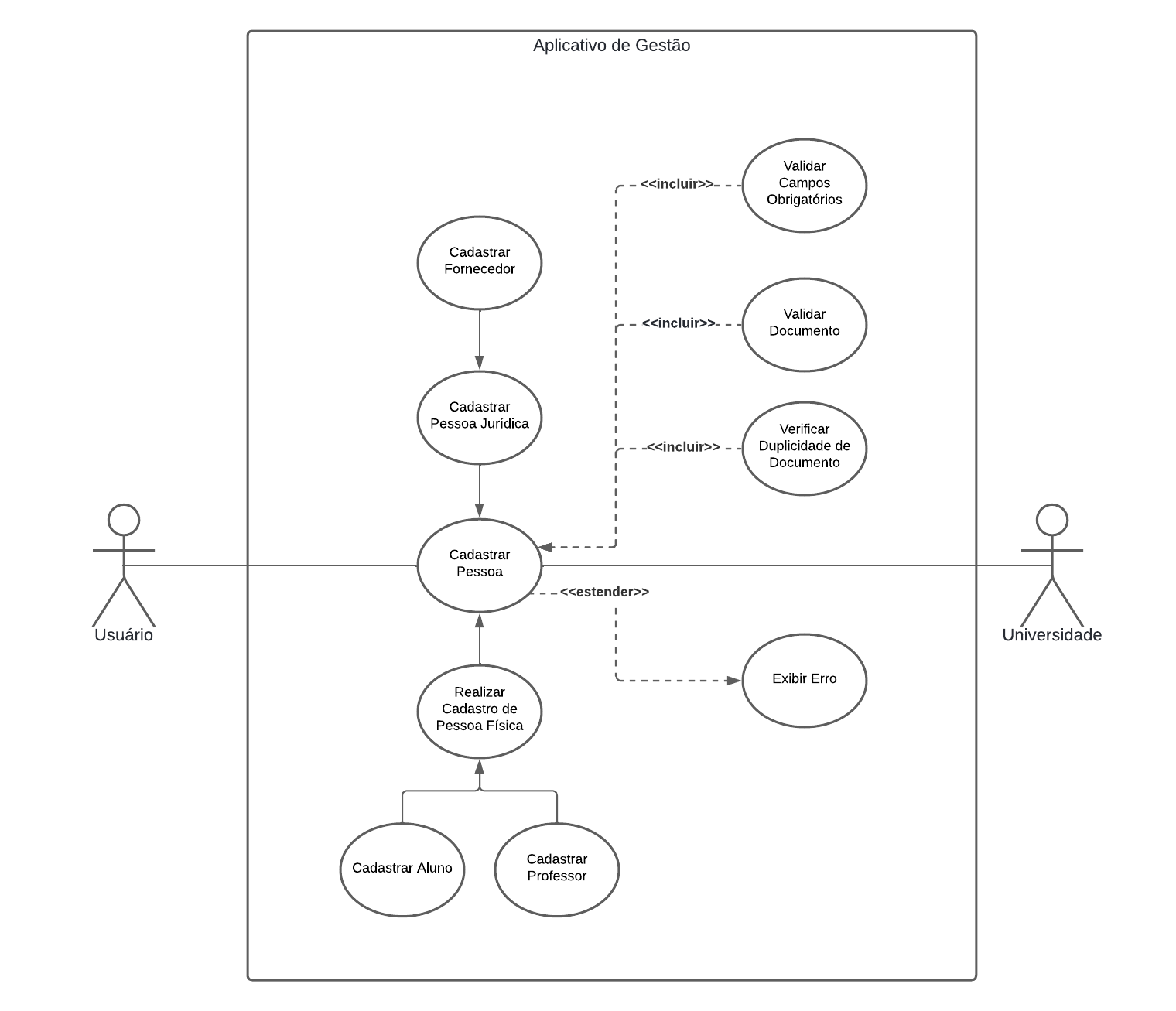
[2.2. Cenário alternativo 1: 6](#_heading=h.3dy6vkm)

[2.3. Cenário alternativo 2: 7](#_heading=h.y41e6zpj2xa5)

[3. DIAGRAMA DE CLASSE DO PROJETO PROPOSTO 8](#_heading=h.4d34og8)

[REFERÊNCIAS](#_heading=h.7tky8k5uo148)10

# **INTRODUÇÃO - Diagrama de caso de uso**

Neste trabalho, serão apresentados os diagramas de casos de uso para o sistema de gestão de dados de uma grande universidade. O objetivo dos diagramas é ilustrar as interações de diferentes tipos de usuários, como alunos, professores, pessoas físicas, pessoas jurídicas e fornecedores, com as funções de cadastro do sistema. O uso da UML permite que essas interações sejam descritas de maneira visual e padronizada, facilitando o entendimento por parte dos desenvolvedores e demais envolvidos no projeto.

### 1.1. Descrição Detalhada do Diagrama

O diagrama apresenta um Caso de Uso para um Aplicativo de Gestão no contexto universitário. Ele está estruturado com os seguintes elementos:

1. **Ator "Usuário"** (localizado à esquerda do diagrama):
   1. Realiza ações relacionadas ao cadastro de diferentes tipos de entidades.
2. **Ator "Universidade"** (localizado à direita do diagrama):
   1. Está envolvido com a exibição de erros e outras validações.
3. **Casos de Uso Principais:**
   1. **Cadastrar Pessoa**
      1. Inclui três subcasos:
         1. **Cadastrar Fornecedor**:
            1. Ligado a "Cadastrar Pessoa Jurídica".
         2. **Cadastrar Pessoa Jurídica**:
            1. Inclui a validação de campos obrigatórios, documentos e duplicidade de documentos.
         3. **Realizar Cadastro de Pessoa Física**:
            1. Estende o caso "Cadastrar Pessoa" e é subdividido em:

**Cadastrar Aluno**

**Cadastrar Professor**

* 1. **Validar Campos Obrigatórios**:
     1. É um caso de uso incluído por vários fluxos principais.
  2. **Validar Documento**:
     1. Verifica se o documento informado segue os requisitos necessários.
  3. **Verificar Duplicidade de Documento**:
     1. Verifica se o documento já foi registrado anteriormente.
  4. **Exibir Erro**:
     1. Estendido pelo caso "Cadastrar Pessoa" caso ocorra algum problema.

1. As relações incluem:
   1. **<<include>>** para casos que são obrigatórios em diferentes fluxos, como validação de campos.
   2. **<<extend>>** para a exibição de erros quando necessário.

# **DESCRIÇÃO DOS CENÁRIOS DOS CASOS DE USO**

## Cenário Principal:

O cenário principal, ou seja, como o sistema deve operar para realizar o cadastro. Será utilizado o Cadastro de Aluno, sendo o ator o aluno.

1. Pré-condições:

· Número de matrícula emitido, CPF válido no banco de dados.

1. Fluxo normal:

1. Clicar em realizar cadastro

2. Inserir informações pessoais

3. Confirmar e-mail válido

1. Fluxo alternativo: Aluno menor de 18 anos

· Apresentar mensagem de que é necessário cadastrar um responsável.

1. Pós condição:

· Aluno acessa sistema com usuário e senha informados no cadastro.

## Cenário alternativo 1:

O primeiro cenário alternativo será com o Cadastro Pessoa Jurídica. Nesses cenários nos deparamos com fluxos alternativos que podem ocorrer durante o processo de cadastro. Um bom ponto de partida é pensar em erros ou eventos excepcionais que podem acontecer como no exemplo abaixo de certidão negativa não consta para pessoa jurídica.

1. Ator: Pessoa Jurídica
2. Pré-condições:

· Envio de documentos validados pela Receita Federal

1. Fluxo normal:

1. Clicar em realizar cadastro

2. Inserir informações da empresa

3. Confirmar e-mail válido

1. Fluxo alternativo: Certidão Negativa não consta

· Apresentar mensagem de que é necessário o envio da Certidão Negativa para finalizar o cadastro.

1. Pós condição:

· Acessa sistema com usuário e senha informados no cadastro e vincula pessoas físicas à empresa

## Cenário alternativo 2:

Último cenário considerado será o de Cadastro de Fornecedor:

1. Ator: Fornecedor
2. Pré-condições:

· CPF vinculado à uma empresa no banco de dados

1. Fluxo normal:

1. Clicar em realizar cadastro

2. Inserir informações pessoais

3. Confirmar vínculo empresarial

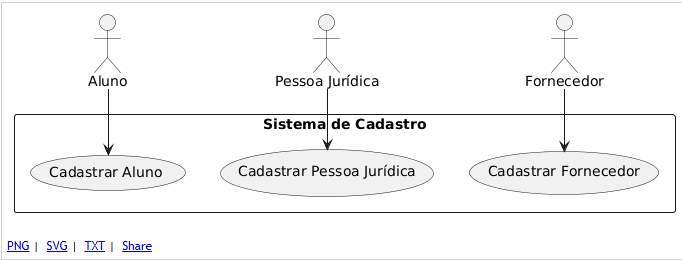
4. Confirmar e-mail válido

1. Fluxo alternativo: CPF não consta junto à empresa informada

· Apresentar mensagem de que é necessário conferir vínculo junto à empresa para concluir cadastro.

1. Pós condição:

· Acessar sistema com usuário e senha informados.



# **DIAGRAMA DE CLASSE DO PROJETO PROPOSTO**

Diagrama de Classe (descrição textual):

1. Classe Pessoa

● Atributos:

○ nome: String

○ cpf: String

○ email: String

○ dataNascimento: Date

● Métodos:

○ validarDocumento(): Boolean

○ verificarDuplicidadeDocumento(): Boolean

2. Classe Aluno (herda de Pessoa)

● Atributos:

○ numeroMatricula: String

○ responsavel: String (para casos de menores de 18 anos)

● Métodos:

○ realizarCadastro(): void

○ verificarMenorDeIdade(): Boolean

3. Classe Professor (herda de Pessoa)

● Atributos:

○ documentoDocencia: String

○ rg: String

● Métodos:

○ validarDocumentoDocencia(): Boolean

○ realizarCadastro(): void

4. Classe PessoaJuridica

● Atributos:

○ cnpj: String

○ razaoSocial: String

○ certidaoNegativa: Boolean

● Métodos:

○ validarDocumentosReceitaFederal(): Boolean

○ vincularPessoaFisica(pessoa: Pessoa): void

5. Classe Fornecedor (herda de Pessoa)

● Atributos:

○ vinculoEmpresa: PessoaJuridica

● Métodos:

○ validarVinculoEmpresa(): Boolean

○ realizarCadastro(): void

Relações:

● Herança: Aluno, Professor e Fornecedor herdam de Pessoa.

● Associação: Fornecedor tem um vínculo com PessoaJuridica.

● Composição: PessoaJuridica pode vincular várias Pessoa (representando os

funcionários, por exemplo).

Notas sobre o diagrama:

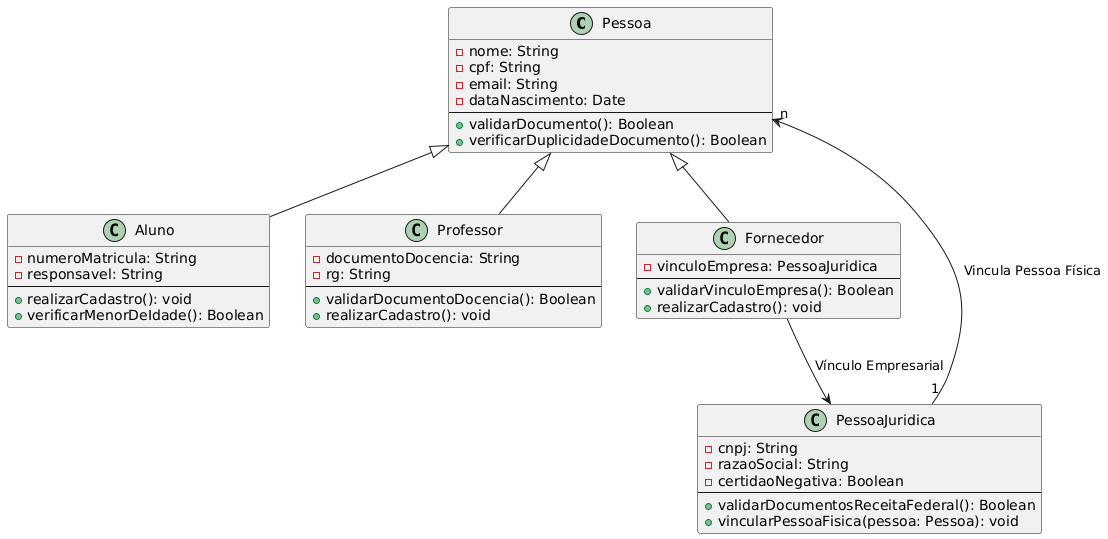
● Cada entidade reflete as funções de cadastro e validação mostradas nos fluxos do

caso de uso.

● Métodos de validação e regras de negócio, como verificar duplicidade de

documentos e validar campos obrigatórios, estão representados dentro das classes,

reforçando os fluxos do sistema.



**4. DIAGRAMA EM FORMATO MARKDOWN**

**## Diagrama de Classe**

**@startuml**

**actor "Usuário" as Usuario**

**actor "Universidade" as Universidade**

**rectangle "Aplicativo de Gestão" {**

**usecase "Cadastrar Pessoa" as UC\_CadastrarPessoa**

**usecase "Cadastrar Fornecedor" as UC\_CadastrarFornecedor**

**usecase "Cadastrar Pessoa Jurídica" as UC\_CadastrarPessoaJuridica**

**usecase "Realizar Cadastro de Pessoa Física" as UC\_CadastrarPessoaFisica**

**usecase "Cadastrar Aluno" as UC\_CadastrarAluno**

**usecase "Cadastrar Professor" as UC\_CadastrarProfessor**

**usecase "Validar Campos Obrigatórios" as UC\_ValidarCampos**

**usecase "Validar Documento" as UC\_ValidarDocumento**

**usecase "Verificar Duplicidade de Documento" as UC\_VerificarDuplicidade**

**usecase "Exibir Erro" as UC\_ExibirErro**

**UC\_CadastrarPessoa <|-- UC\_CadastrarPessoaJuridica**

**UC\_CadastrarPessoa <|-- UC\_CadastrarPessoaFisica**

**UC\_CadastrarPessoaFisica <|-- UC\_CadastrarAluno**

**UC\_CadastrarPessoaFisica <|-- UC\_CadastrarProfessor**

**UC\_CadastrarPessoaJuridica <|-- UC\_CadastrarFornecedor**

**UC\_CadastrarPessoa -- UC\_ValidarCampos : <<include>>**

**UC\_CadastrarPessoa -- UC\_ValidarDocumento : <<include>>**

**UC\_CadastrarPessoa -- UC\_VerificarDuplicidade : <<include>>**

**UC\_CadastrarPessoa ..> UC\_ExibirErro : <<extend>>**

**}**

**Usuario --> UC\_CadastrarPessoa**

**Universidade --> UC\_ExibirErro**

**@enduml**

**### 1. Classe Pessoa**

**- \*\*Atributos:\*\***

**- `nome`: String**

**- `cpf`: String**

**- `email`: String**

**- `dataNascimento`: Date**

**- \*\*Métodos:\*\***

**- `validarDocumento()`: Boolean**

**- `verificarDuplicidadeDocumento()`: Boolean**

**### 2. Classe Aluno (herda de Pessoa)**

**- \*\*Atributos:\*\***

**- `numeroMatricula`: String**

**- `responsavel`: String (para casos de menores de 18 anos)**

**- \*\*Métodos:\*\***

**- `realizarCadastro()`: void**

**- `verificarMenorDeIdade()`: Boolean**

**### 3. Classe Professor (herda de Pessoa)**

**- \*\*Atributos:\*\***

**- `documentoDocencia`: String**

**- `rg`: String**

**- \*\*Métodos:\*\***

**- `validarDocumentoDocencia()`: Boolean**

**- `realizarCadastro()`: void**

**### 4. Classe PessoaJuridica**

**- \*\*Atributos:\*\***

**- `cnpj`: String**

**- `razaoSocial`: String**

**- `certidaoNegativa`: Boolean**

**- \*\*Métodos:\*\***

**- `validarDocumentosReceitaFederal()`: Boolean**

**- `vincularPessoaFisica(pessoa: Pessoa)`: void**

**### 5. Classe Fornecedor (herda de Pessoa)**

**- \*\*Atributos:\*\***

**- `vinculoEmpresa`: PessoaJuridica**

**- \*\*Métodos:\*\***

**- `validarVinculoEmpresa()`: Boolean**

**- `realizarCadastro()`: void**

**## Relações**

**- \*\*Herança:\*\* Aluno, Professor e Fornecedor herdam de Pessoa.**

**- \*\*Associação:\*\* Fornecedor tem um vínculo com PessoaJuridica.**

**- \*\*Composição:\*\* PessoaJuridica pode vincular várias Pessoa (representando os funcionários, por exemplo).**

**### Notas sobre o diagrama:**

**- Cada entidade reflete as funções de cadastro e validação mostradas nos fluxos do caso de uso.**

**- Métodos de validação e regras de negócio, como verificar duplicidade de documentos e validar campos obrigatórios, estão representados dentro das classes, reforçando os fluxos do sistema.**

**---**

**# DESCRIÇÃO DOS CENÁRIOS DOS CASOS DE USO**

**## Cenário Principal**

**### Cadastro de Aluno**

**\*\*Ator:\*\* Aluno**

**\*\*Pré-condições:\*\***

**- Número de matrícula emitido, CPF válido no banco de dados.**

**\*\*Fluxo normal:\*\***

**1. Clicar em realizar cadastro.**

**2. Inserir informações pessoais.**

**3. Confirmar e-mail válido.**

**\*\*Fluxo alternativo:\*\* Aluno menor de 18 anos**

**- Apresentar mensagem de que é necessário cadastrar um responsável.**

**\*\*Pós-condição:\*\***

**- Aluno acessa o sistema com usuário e senha informados no cadastro.**

**---**

**## Cenário Alternativo 1**

**### Cadastro Pessoa Jurídica**

**\*\*Ator:\*\* Pessoa Jurídica**

**\*\*Pré-condições:\*\***

**- Envio de documentos validados pela Receita Federal.**

**\*\*Fluxo normal:\*\***

**1. Clicar em realizar cadastro.**

**2. Inserir informações da empresa.**

**3. Confirmar e-mail válido.**

**\*\*Fluxo alternativo:\*\* Certidão Negativa não consta**

**- Apresentar mensagem de que é necessário o envio da Certidão Negativa para finalizar o cadastro.**

**\*\*Pós-condição:\*\***

**- Acessa o sistema com usuário e senha informados no cadastro e vincula pessoas físicas à empresa.**

**---**

**## Cenário Alternativo 2**

**### Cadastro de Fornecedor**

**\*\*Ator:\*\* Fornecedor**

**\*\*Pré-condições:\*\***

**- CPF vinculado à uma empresa no banco de dados.**

**\*\*Fluxo normal:\*\***

**1. Clicar em realizar cadastro.**

**2. Inserir informações pessoais.**

**3. Confirmar vínculo empresarial.**

**4. Confirmar e-mail válido.**

**\*\*Fluxo alternativo:\*\* CPF não consta junto à empresa informada**

**- Apresentar mensagem de que é necessário conferir vínculo junto à empresa para concluir o cadastro.**

**\*\*Pós-condição:\*\***

**- Acessa o sistema com usuário e senha informados.**

**Prototipação**

**Tela de celular com foto de homem

Descrição gerada automaticamentePara um melhor entendimento da prototipação há uma descrição junto as imagens.**

**Começando pela tela de login, podemos ver a homeopage do site da universidade. Ao clicar em “Sou novo aqui! Realizar cadastro.” podemos ver a seleção do tipo de cadastro sendo estes aluno, professor ou colaborador. Começando pelo perfil do tipo aluno podemos ver o formulário a ser preenchido. Preenchendo o CPF irá verificar no banco de dados se há uma matrícula emitida para este mesmo CPF, então se houver a caixa do texto irá ficar verde e irá aparecer um balão com uma mensagem informando que há uma matrícula vinculada.**

**Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente**

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

**Outro campo responsivo para o tipo aluno é a data de nascimento, pois se for menor de 18 anos exige um cadastro de pessoa física responsável por este aluno. Como mostra na imagem a caixa de texto fica laranja e um novo formulário é aberto para preenchimento.**

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

**Na tela seguinte é preenchido o endereço, e após há uma tela para cadastro de e-mail, usuário e senha.**

**Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente**

# 

# **Interface gráfica do usuário, Aplicativo Descrição gerada automaticamente**

**O próximo perfil de cadastro a ser feito é o de professor, ele segue o mesmo fluxo, mudando apenas os campos da imagem a seguir. Há uma verificação do CPF para conferir os documentos que a universidade validou no banco de dados estão corretos e vinculados ao CPF inserido. Com o registro validado segue para as telas de endereço e posteriormente de cadastro de login.**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

# Interface gráfica do usuário, Site Descrição gerada automaticamente

# **Interface gráfica do usuário, Aplicativo, Site Descrição gerada automaticamente**

# **Interface gráfica do usuário, Aplicativo, Site Descrição gerada automaticamente**

**Por último o perfil de Colaborador (Pessoa Jurídica) aonde a verificação junto ao banco de dados da universidade é feita pelo CNPJ abrindo então campos para adição de um contato principal (pessoa física) seguido das telas para cadastro de endereço e login.**

**Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

# **Interface gráfica do usuário, Aplicativo Descrição gerada automaticamente**

# **REFERÊNCIAS:**

OLIVEIRA, Henrique Pontes Gonçalves de. *Análise de sistemas*. São Paulo: Senac

MARIN, Matt; LINDLEY, David. Markdown Guide. Disponível em: https://www.markdownguide.org/basic-syntax/. Acesso em: 22 nov. 2024.

**PLANTTEXT.** *PlantText - UML Editor*. Disponível em:<https://www.planttext.com/>. Acesso em: 22 nov. 2024.