**SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL**

**SENAC**

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

**PROJETO INTEGRADOR: MODELAGEM DE SISTEMA ORIENTADO A OBJETOS**

**Integrantes do grupo:**

* Roberto da Rosa Borges Fonseca Lima
* Eduardo Bruno do Nascimento Cruz
* Karoline Marta Marques Moura Marwell
* Laísse de Carvalho Moura
* Vinícius Serafim
* Fernanda Fujishima
* Pedro Henrique Oliveira Fernandes

**EAD - ENSINO À DISTÂNCIA - 2025**

**RESUMO**

A gestão de dados em uma universidade envolve múltiplos tipos de pessoas e relações institucionais. Desde o cadastro de alunos, professores e fornecedores até o controle de responsáveis legais, é fundamental utilizar uma abordagem organizada e sistemática. Neste projeto integrador, propomos a modelagem de um sistema orientado a objetos capaz de contemplar essas necessidades, utilizando a Linguagem Unificada de Modelagem (UML). A aplicação de conceitos como herança, associação e encapsulamento permitirá representar adequadamente os elementos do sistema, suas responsabilidades e interações. O uso de UML facilita o entendimento e a comunicação entre os desenvolvedores e outros stakeholders, além de representar uma prática amplamente adotada na engenharia de software.

Palavras-chave: UML; Sistema orientado a objetos.

Sumário

[1. OBJETIVOS 4](#_Toc200710131)

[2. DIAGRAMA DE CLASSE UML 4](#_Toc200710132)

[1. DESCRIÇÃO DAS CLASSES 5](#_Toc200710133)

[3. DESCRIÇÃO DOS CASOS DE USO 6](#_Toc200710134)

[ **Cadastro de Pessoa Física** 6](#_Toc200710135)

[ **Cadastro de Pessoa Jurídica** 6](#_Toc200710136)

[ **Cadastro de Professores** 7](#_Toc200710137)

[ **Cadastro de Fornecedores** 7](#_Toc200710138)

[ **Cadastro de Alunos** 7](#_Toc200710139)

[4. PLANEJAMENTO DO DESENVOLVIMENTO DA SOLUÇÃO 8](#_Toc200710140)

[1. PROTÓTIPO FUNCIONAL CONCEITUAL 8](#_Toc200710141)

[2. IMPLEMENTAÇÃO ORIENTADA A OBJETOS (ESBOÇO) 8](#_Toc200710142)

[5. REFERÊNCIAS 10](#_Toc200710143)

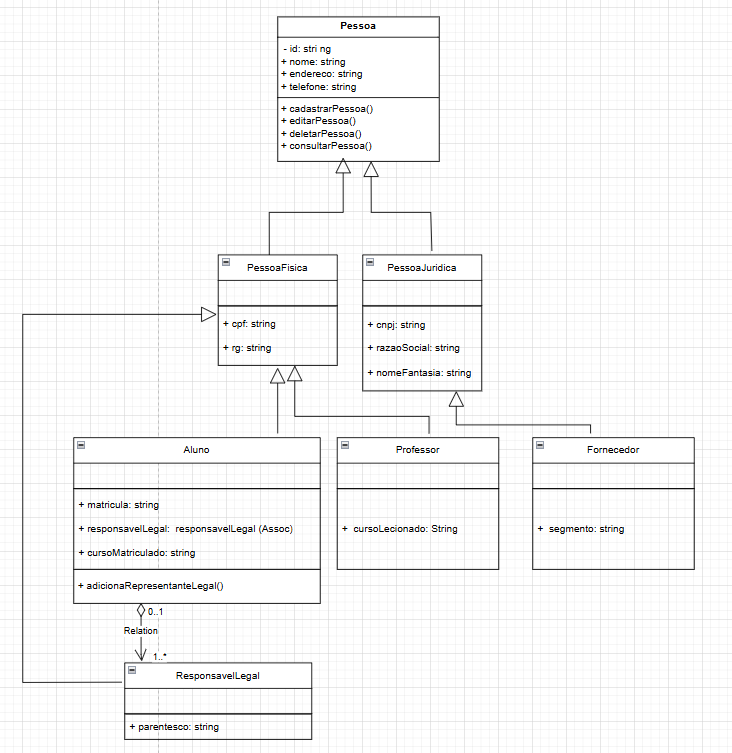
# OBJETIVOS

Este projeto tem os seguintes objetivos:

* Modelar um sistema de gestão de dados para uma universidade utilizando UML.
* Aplicar os conceitos da programação orientada a objetos como herança, associação e encapsulamento.
* Desenvolver diagramas de casos de uso e diagramas de classes.
* Identificar e descrever os principais atores e seus respectivos casos de uso.
* Criar um protótipo conceitual de como seria a interação com o sistema modelado.

# DIAGRAMA DE CLASSE UML

A seguir, apresentamos o diagrama de classes do sistema, contemplando os principais elementos envolvidos nos cadastros e suas relações:

****

# DESCRIÇÃO DAS CLASSES

* **Pessoa**: Classe abstrata que representa uma pessoa genérica. Contém os atributos nome, e-mail e telefone. É base para as demais especializações.
* **PessoaFisica:** Herda de Pessoa. Representa indivíduos com CPF e data de nascimento.
* **PessoaJuridica:** Herda de Pessoa. Representa entidades com CNPJ e razão social.
* **Aluno**: Especialização de PessoaFisica. Inclui matrícula e curso do aluno.
* **Professor**: Especialização de Pessoa Física. Inclui área de atuação e titulação acadêmica.
* **Fornecedor:** Especialização de Pessoa Jurídica. Inclui o segmento de fornecimento.
* **ResponsavelLegal:** Classe associada a Aluno. Contém nome, telefone e grau de parentesco com o aluno.

# DESCRIÇÃO DOS CASOS DE USO

Abaixo estão descritos os principais casos de uso do sistema com seus cenários principais, alternativos, pré e pós-condições.

## **Cadastro de Pessoa Física**

**Cenário Principal**: O usuário acessa o sistema, preenche os campos obrigatórios e confirma o cadastro. O sistema salva os dados no banco e confirma a operação.

**Cenário Alternativo 1**: O usuário tenta cadastrar sem preencher todos os campos obrigatórios. O sistema alerta o erro e impede o salvamento.

**Cenário Alternativo 2:** O sistema está indisponível no momento do envio. Uma mensagem de erro é apresentada, orientando o usuário a tentar novamente mais tarde.

**Pré-condição:** O usuário está autenticado no sistema com permissão para realizar cadastros.

**Pós-condição:** O novo cadastro está salvo e acessível no sistema.

## **Cadastro de Pessoa Jurídica**

**Cenário Principal**: O usuário acessa o sistema, preenche os campos obrigatórios e confirma o cadastro. O sistema salva os dados no banco e confirma a operação.

**Cenário Alternativo 1:** O usuário tenta cadastrar sem preencher todos os campos obrigatórios. O sistema alerta o erro e impede o salvamento.

**Cenário Alternativo 2:** O sistema está indisponível no momento do envio. Uma mensagem de erro é apresentada, orientando o usuário a tentar novamente mais tarde.

**Pré-condição:** O usuário está autenticado no sistema com permissão para realizar cadastros.

**Pós-condição:** O novo cadastro está salvo e acessível no sistema.

## **Cadastro de Professores**

**Cenário Principal:** O usuário acessa o sistema, preenche os campos obrigatórios e confirma o cadastro. O sistema salva os dados no banco e confirma a operação.

**Cenário Alternativo 1:** O usuário tenta cadastrar sem preencher todos os campos obrigatórios. O sistema alerta o erro e impede o salvamento.

**Cenário Alternativo 2**: O sistema está indisponível no momento do envio. Uma mensagem de erro é apresentada, orientando o usuário a tentar novamente mais tarde.

**Pré-condição:** O usuário está autenticado no sistema com permissão para realizar cadastros.

**Pós-condição:** O novo cadastro está salvo e acessível no sistema.

## **Cadastro de Fornecedores**

**Cenário Principal:** O usuário acessa o sistema, preenche os campos obrigatórios e confirma o cadastro. O sistema salva os dados no banco e confirma a operação.

**Cenário Alternativo 1:** O usuário tenta cadastrar sem preencher todos os campos obrigatórios. O sistema alerta o erro e impede o salvamento.

**Cenário Alternativo 2:** O sistema está indisponível no momento do envio. Uma mensagem de erro é apresentada, orientando o usuário a tentar novamente mais tarde.

**Pré-condição:** O usuário está autenticado no sistema com permissão para realizar cadastros.

**Pós-condição:** O novo cadastro está salvo e acessível no sistema.

## **Cadastro de Alunos**

**Cenário Principal:** O usuário acessa o sistema, preenche os campos obrigatórios e confirma o cadastro. O sistema salva os dados no banco e confirma a operação.

**Cenário Alternativo 1:** O usuário tenta cadastrar sem preencher todos os campos obrigatórios. O sistema alerta o erro e impede o salvamento.

**Cenário Alternativo 2:** O sistema está indisponível no momento do envio. Uma mensagem de erro é apresentada, orientando o usuário a tentar novamente mais tarde.

**Pré-condição:** O usuário está autenticado no sistema com permissão para realizar cadastros.

**Pós-condição:** O novo cadastro está salvo e acessível no sistema.

# PLANEJAMENTO DO DESENVOLVIMENTO DA SOLUÇÃO

O projeto foi desenvolvido considerando o modelo incremental de processo de software permitindo entregas parciais e validações frequentes dos componentes. O sistema será evoluído a partir de funcionalidades básicas, como cadastro de pessoas e professores, até o controle completo de alunos, fornecedores e responsáveis legais. A modelagem foi realizada com base nos requisitos levantados a partir de situações comuns em instituições de ensino superior.

As etapas de desenvolvimento incluíram: definição dos requisitos funcionais, modelagem UML, descrição dos casos de uso, elaboração do diagrama de classes, esboço de implementação e montagem do protótipo conceitual.

# PROTÓTIPO FUNCIONAL CONCEITUAL

Como parte do planejamento, foram esboçadas telas simuladas representando a interface do sistema de cadastro de pessoas. Essas telas demonstram a navegação esperada pelo usuário final e a organização dos dados, embora não tenham sido implementadas em uma linguagem específica nesta fase do projeto.

# IMPLEMENTAÇÃO ORIENTADA A OBJETOS (ESBOÇO)

A seguir, apresentamos um exemplo de como a estrutura das classes pode ser implementada em linguagem orientada a objetos (exemplo em C#):

public abstract class Pessoa {

public string Nome { get; set; }

public string Email { get; set; }

public string Telefone { get; set; }

}

public class PessoaFisica : Pessoa {

public string Cpf { get; set; }

public DateTime DataNascimento { get; set; }

}

public class PessoaJuridica : Pessoa {

public string Cnpj { get; set; }

public string RazaoSocial { get; set; }

}

public class Aluno : PessoaFisica {

public string Matricula { get; set; }

public string Curso { get; set; }

}

public class Professor : PessoaFisica {

public string AreaAtuacao { get; set; }

public string Titulacao { get; set; }

}

public class Fornecedor : PessoaJuridica {

public string Segmento { get; set; }

}

public class ResponsavelLegal {

public string Nome { get; set; }

public string Telefone { get; set; }

public string Parentesco { get; set; }

}

# REFERÊNCIAS

* + BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro: Elsevier; Campus, 2007.
  + FOWLER, Martin. UML Essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman – Artmed, 2005.
  + PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. Porto Alegre: McGraw Hill/Artmed, 2011.
  + SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson, 2011.
  + BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. São Paulo: Pearson, 2005.
  + OBJECT MANAGEMENT GROUP. Get Started with UML. Disponível em: <http://www.uml.org/>. Acesso em: 02 jun. 2025.