**CENTRO UNIVERSITÁRIO SENAC**

**Disciplina: PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A OBJETOS**

**PROCESSO DE MODELAGEM DE PROJETO VOLTADO A GESTÃO DE DADOS DE UM CENTRO UNIVERSITÁRIO**

**São Paulo**

**2023**

**CENTRO UNIVERSITÁRIO SENAC**

**Disciplina: PROJETO INTEGRADOR: DESENVOLVIMENTO DE SISTEMAS ORIENTADO A OBJETOS**

Darkcikelwe Manoel Neves Ribeiro

Júlio Gabriel Moreira Rocha

Laura Cristina Angioletto Martins

Morgana Conceição da Cruz Gomes

Pedro Favoreto Gaya

**PROCESSO DE MODELAGEM DE PROJETO VOLTADO A GESTÃO DE DADOS DE UM CENTRO UNIVERSITÁRIO**

Projeto Integrador Desenvolvimento de Sistemas Orientado a Objetos – Fase 1 apresentado ao Centro Universitário Senac, como exigência parcial para obtenção de aprovação na disciplina Projeto Integrador Desenvolvimento de Sistemas Orientado a Objetos, do curso de Análise e Desenvolvimento de Sistemas.

Orientador (a): Prof. (ª) Enoque Felipe dos Santos Leal.

**São Paulo**

**2023**

**Sumário**

[INTRODUÇÃO E OBJETIVO DO PROJETO INTEGRADOR 4](#_Toc145854554)

[1. MODELAGEM DE PROJETO VOLTADO A GESTÃO DE DADOS DE UM CENTRO UNIVERSITÁRIO 5](#_Toc145854555)

[1.1. Diagrama de Casos de Uso 5](#_Toc145854556)

[1.2 Descrição dos Casos de Uso 5](#_Toc145854557)

[1.3 Diagrama de Classes 10](#_Toc145854558)

[REFERÊNCIAS 14](#_Toc145854559)

# INTRODUÇÃO E OBJETIVO DO PROJETO INTEGRADOR

# MODELAGEM DE PROJETO VOLTADO A GESTÃO DE DADOS DE UM CENTRO UNIVERSITÁRIO

## Diagrama de Casos de Uso

Diagrama

Descrição gerada automaticamente

Figura 1 – Diagrama de Casos de Uso

## 1.2 Descrição dos Casos de Uso

Cenário: Realiza a Prova

Atores: Aluno, Fornecedor, Professor

Objetivo: Garantir que o aluno matriculado seja capaz de realizar as avaliações propostas pela faculdade dentro do ambiente de avaliações.

Pré-Condição: O aluno deve estar matriculado na rede universitária. O aluno deve ter provas agendadas das disciplinas que está matriculado.

Fluxo Principal:

1) O aluno acessa o ambiente de avaliações do portal da faculdade.

2) O aluno realiza a prova.

3) A prova é armazenada por um banco de dados externo através de um fornecedor terceirizado.

4) O fornecedor armazena a prova do aluno.

5) O fornecedor provê uma chave de acesso ao professor.

Fluxos Alternativos:

● Aluno não matriculado

o Um aviso é exibido que o aluno não se encontra matriculado na instituição.

● Aluno sem provas agendadas

o Um aviso é exibido que o aluno não possui provas agendadas para realização.

Pós-Condição: Todos os dados são computados por intermédio do fornecedor, de modo que o aluno e o professor tenham acesso aos resultados das avaliações.

Cenário: Avalia a Prova

Atores: Professor, Fornecedor

Objetivo: Garantir que o professor tenha acesso às provas realizadas pelos alunos, e que possa atribuir uma nota às mesmas.

Pré-condição: O Professor deve ter acesso ao ambiente de avaliações com permissões adicionais. O professor deve estar vinculado a uma disciplinas com provas já realizadas.

Fluxo Principal:

1) O professor acessa o ambiente de avaliações do portal da faculdade.

2) O professor acessa o arquivo da prova realizada pelo aluno.

3) O professor avalia a prova e realiza um novo envio ao banco de dados com a nota do aluno.

4) Os dados enviados pelo professor são armazenados pelo bando de dados.

5) O fornecedor libera acesso ao aluno para fins de consulta.

Fluxos Alternativos:

● Usuário sem permissões de Professor tenta acessar a sessão para professores

o Um aviso é exibido que a sessão é exclusiva para professores.

o O usuário é redirecionado para a tela inicial.

● Professor sem provas a serem avaliadas

o Um aviso é exibido que não existem provas que precisam ser avaliadas.

Pós-condição: As notas das avaliações são salvas no banco de dados e disponibilizadas para a consulta de cada aluno.

Cenário: Consulta a Prova

Atores: Aluno

Objetivo: Permitir que os alunos consultem as notas obtidas nas provas, para que possam decidir se precisam ou não realizar a prova de recuperação.

Pré-condição: O aluno deve estar matriculado na rede universitária. O aluno deve ter realizado uma prova. O professor deve já ter avaliado a prova realizada pelo aluno.

Fluxo Principal:

1) O aluno acessa o ambiente de avaliações do portal da faculdade.

2) O aluno obtém os resultados da avaliação.

Fluxos Alternativos:

1. Aluno sem provas realizadas
2. Um aviso é exibido que não existem provas realizadas pelo aluno.
3. Prova sem avaliação do professor
4. Um aviso é exibido que a prova ainda não foi avaliada pelo professor da disciplina.
5. O aluno é informado sobre a data limite até a qual o professor deve ter avaliado a prova.
6. Pós-condição: O aluno tem acesso a sua nota obtida na prova. A conta do aluno é atualizada para incluir sua nova nota na média da disciplina.

Cenário: Matrícula do Aluno

Ator: Aluno, Fornecedor

Objetivo: Permitir que novos alunos sejam matriculados e tenham acesso às disciplinas e avaliações.

Pré-condição: Pessoa física. Período de matrículas aberto.

Fluxo Principal:

1) A pessoa física acessa o portal da faculdade.

2) Preenche os campos do formulário de acordo com o dado necessário.

3) Assina os termos de compromisso.

4) Realiza o pagamento da matrícula.

5) Os dados são computados pelo sistema da faculdade.

6) Os dados são armazenados do banco de dados do fornecedor.

7) O fornecedor libera o acesso ao aluno.

Fluxos Alternativos:

1. Pessoa jurídica tenta realizar matrícula:
2. Um aviso é exibido que somente pessoas físicas podem realizar a matrícula.
3. Pagamento recusado:
4. Um aviso é exibido que a forma de pagamento escolhida não pôde ser completada.
5. Outras opções de pagamento são exibidas ao usuário.
6. Pós-Condição: O aluno é matriculado e obtém acesso ao ambiente de avaliações.

Cenário: Realizar prova de recuperação

Atores: Aluno, Professor, Fornecedor

Objetivo: Garantir que o aluno tenha a possibilidade de realizar uma nova prova caso sua média seja insuficiente para a aprovação

Pré-Condição: Aluno matriculado na disciplina. Realização das provas regulares. Média do aluno insuficiente para aprovação.

Fluxo Principal:

1) O sistema da faculdade informa ao aluno que a média é insuficiente.

2) O professor envia o arquivo da prova de recuperação para o sistema da faculdade.

3) O sistema armazena a prova no banco de dados do fornecedor.

4) O fornecedor disponibiliza uma chave de acesso ao aluno.

5) O aluno acessa o ambiente de avaliações.

6) O aluno realiza a prova.

Fluxos Alternativos:

1. Aluno não realizou as provas regulares:
2. Um aviso é exibido que somente alunos que realizaram as provas regulares têm direito à prova de recuperação.
3. Aluno já realizou a prova de recuperação para essa disciplina:
4. Um aviso é exibido que somente uma prova de recuperação pode ser realizada por disciplina por semestre.
5. Pós-Condição 1: A pontuação na prova é atingida e o aluno é aprovado.
6. Pós-Condição 2: A pontuação não é atingida e o aluno é reprovado.

Detalhamento do cenário 1 (Realiza a Prova):

O cenário principal tem como objetivo garantir que o aluno matriculado seja capaz de realizar as avaliações propostas pela faculdade dentro do ambiente de avaliações do sistema interno da mesma.

Para ser capaz de atender a demanda de milhares de estudantes matriculados, a faculdade utiliza o sistema de banco de dados de terceiros para armazenar as informações sensíveis e acadêmicas dos estudantes, de modo que haja uma facilidade maior da manutenção dos arquivos e uma segurança reforçada que impeça a quebra da integridade dos dados.

Dessa forma, o sistema da faculdade serve como um intermediador entre os alunos; professores e funcionários de modo que todos tenham acesso aos dados que são necessários e a um ambiente funcional.

Detalhamento cenário 4 (Matrícula do Aluno):

Para que o cenário principal funcione, não deve ser permitido que uma pessoa de qualquer espécie tenha acesso ao sistema da faculdade se não houver cadastro de alguma forma (matrícula, para os alunos; contrato de trabalho, aos funcionários).

Sendo assim, se faz necessário um cenário complementar, onde levamos em conta que uma pessoa não cadastrada no sistema da faculdade possa vir a se registrar e utilizar desse sistema como parte integral dele.

Detalhamento cenário 5 (Realizar prova de recuperação):

Dentro do ambiente de avaliações, o aluno precisa atingir uma média necessária, no caso do Senac, a média é 6.0.

Utilizando um exemplo prático, podemos dizer que a média é a soma da média ponderada entre a Produção Textual Individual (PTI), Quiz e a Prova Trimestral; todas armazenadas no mesmo banco de dados.

Como existe a possibilidade dessa média não ser atingida, é necessário que haja um segundo cenário alternativo que ofereça ao aluno uma nova oportunidade de elevar a sua média. Funciona como o cenário 1, porém com uma nova entrada e uma nova requisição.

Caso o aluno atinja a média, é executada a pós-condição 1 onde o aluno é aprovado. Senão, é executada a pós-condição 2, onde o aluno é reprovado.

## 1.3 Diagrama de Classes

+----------------------------------------------------------------------+

| Pessoa |

+----------------------------------------------------------------------+

| - nome: string |

| - dataNascimento: data |

| - endereco: string |

+----------------------------------------------------------------------+

| + inserirNome(nome: string): void |

| + excluirNome(): void |

| + alterarNome(novoNome: string): void |

| + inserirDataNascimento(data: data): void |

| + excluirDataNascimento(): void |

| + alterarDataNascimento(novaData: data): void |

| + inserirEndereco(endereco: string): void |

| + excluirEndereco(): void |

| + alterarEndereco(novoEndereco: string): void |

+----------------------------------------------------------------------+

+-------------------------------------------------------------+

| PessoaFisica |

+-------------------------------------------------------- ----+

| - cpf: string |

| - tipo: TipoPessoa |

+------------------------------------------------------------+

| + inserirCPF(cpf: string): void |

| + excluirCPF(): void |

| + alterarCPF(novoCPF: string): void |

| + alterarTipo(novoTipo: TipoPessoa): void |

+------------------------------------------------------------+

+-----------------------------------------------------------+

| PessoaJuridica |

+-----------------------------------------------------------+

| - cnpj: string |

| - tipo: TipoPessoa |

+-----------------------------------------------------------+

| + inserirCNPJ(cnpj: string): void |

| + excluirCNPJ(): void |

| + alterarCNPJ(novoCNPJ: string): void |

| + alterarTipo(novoTipo: TipoPessoa): void |

+-----------------------------------------------------------+

+----------------------------------------------------------------------+

| Professor |

+----------------------------------------------------------------------+

| - matricula: string |

| - disciplinas: Lista de Disciplina |

+---------------------------------------------------------------------+

| + inserirMatricula(matricula: string): void |

| + excluirMatricula(): void |

| + alterarMatricula(novaMatricula: string): void |

| + adicionarDisciplina(disciplina: Disciplina): void |

| + removerDisciplina(disciplina: Disciplina): void |

+---------------------------------------------------------------------+

+----------------------------------------------------------------+

| Fornecedor |

+----------------------------------------------------------------+

| - tipo: TipoFornecedor |

+----------------------------------------------------------------+

| + alterarTipo(novoTipo: TipoFornecedor): void |

+----------------------------------------------------------------+

+----------------------------------------------------------------+

| Aluno |

+----------------------------------------------------------------+

| - matricula: string |

| - turma: Turma |

+----------------------------------------------------------------+

| + inserirMatricula(matricula: string): void |

| + excluirMatricula(): void |

| + alterarMatricula(novaMatricula: string): void |

| + alterarTurma(novaTurma: Turma): void |

+----------------------------------------------------------------+

+---------------------------------------------------------------------+

| Turma |

+---------------------------------------------------------------------+

| - numero: string |

| - disciplina: Disciplina |

| - professor: Professor |

+---------------------------------------------------------------------+

| + inserirNumero(numero: string): void |

| + excluirNumero(): void |

| + alterarNumero(novoNumero: string): void |

| + alterarDisciplina(novaDisciplina: Disciplina): void |

| + alterarProfessor(novoProfessor: Professor): void |

+---------------------------------------------------------------------+

+------------------------------------------------------------------+

| Disciplina |

+------------------------------------------------------------------+

| - nome: string |

| - cargaHoraria: int |

+--------------------------------------------------------------------+

| + inserirNome(nome: string): void |

| + excluirNome(): void |

| + alterarNome(novoNome: string): void |

| + inserirCargaHoraria(cargaHoraria: int): void |

| + excluirCargaHoraria(): void |

| + alterarCargaHoraria(novaCargaHoraria: int): void |

+--------------------------------------------------------------------+

+-------------------------------------------------------------------+

| Nota |

+-------------------------------------------------------------------+

| - valor: float |

| - aluno: Aluno |

| - disciplina: Disciplina |

+-------------------------------------------------------------------+

| + inserirValor(valor: float): void |

| + excluirValor(): void |

| + alterarValor(novoValor: float): void |

| + alterarAluno(novoAluno: Aluno): void |

| + alterarDisciplina(novaDisciplina: Disciplina): void |

+--------------------------------------------------------------------+

# REFERÊNCIAS