

Fase 1: Modelação e Desenho da Base de Dados

☒ 1.1. ~~Análise de Requisitos e Identificação de Entidades~~

O primeiro passo é compreender o que o sistema precisa de armazenar. Com base nos requisitos funcionais (RF) do documento, identificamos as "coisas" ou "objetos" principais do sistema.

☒ 1.2. ~~Definição de Atributos e Chaves~~

Para cada entidade, definimos que informações (atributos) vamos guardar.

Abaixo temos a estrutura de cada tabela (entidade) da base de dados, separada por módulos:

I. Módulo: **Gestão Administrativa e de Pessoas**

1. Tabela **Departamento**

Armazena os departamentos do instituto.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_departamento	INT	PK	AUTO_INCREMENT
nome	VARCHAR(100)		NOT NULL, UNIQUE
orcamento_anual	DECIMAL(15, 2)		NOT NULL, CHECK (orcamento_anual >= 0)
id_chefe	INT	FK	Refere-se a Colaborador(id_colaborador). Pode ser NULL inicialmente.

2. Tabela **Colaborador**

Regista todos os funcionários: administrativos, professores e coordenadores.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_colaborador	INT	PK	AUTO_INCREMENT
nome_completo	VARCHAR(255)		NOT NULL
email	VARCHAR(100)		NOT NULL, UNIQUE
telefone	VARCHAR(20)		UNIQUE
tipo	ENUM('Administrativo', 'Professor')		NOT NULL. Define a função principal.
titulacao	VARCHAR(100)		Para professores (ex: 'Mestre', 'Doutor'). NULL para administrativos.
id_departamento	INT	FK	NOT NULL. Refere-se a Departamento(id_departamento).

II. Módulo: **Gestão Acadêmica**

3. Tabela **Curso**

Armazena os cursos oferecidos.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_curso	INT	PK	AUTO_INCREMENT
nome	VARCHAR(150)		NOT NULL, UNIQUE
duracao_semestres	INT		NOT NULL, CHECK (duracao_semestres > 0)
id_coordenador	INT	FK	NOT NULL, UNIQUE. Refere-se a Colaborador(id_colaborador). Um coordenador por curso.

4. Tabela **Disciplina**

As disciplinas que pertencem a cada curso.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_disciplina	INT	PK	AUTO_INCREMENT
nome	VARCHAR(150)		NOT NULL
carga_horaria_total	INT		NOT NULL, CHECK (carga_horaria_total > 0)
id_curso	INT	FK	NOT NULL. Refere-se a Curso(id_curso).

5. Tabela Turma

Instância de uma disciplina num ano/semestre específico.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_turma	INT	PK	AUTO_INCREMENT
ano_letivo	INT		NOT NULL (ex: 2024)
semestre	INT		NOT NULL, CHECK (semestre IN (1, 2))
vagas	INT		NOT NULL, CHECK (vagas >= 0)
sala	VARCHAR(20)		
id_disciplina	INT	FK	NOT NULL. Refere-se a Disciplina(id_disciplina).

6. Tabela Horario

Define os dias e horas de aula de uma turma, para evitar sobreposições (RN01).

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_horario	INT	PK	AUTO_INCREMENT
id_turma	INT	FK	NOT NULL. Refere-se a Turma(id_turma).
dia_semana	ENUM('Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta', 'Sábado')		NOT NULL
hora_inicio	TIME		NOT NULL
hora_fim	TIME		NOT NULL

7. Tabela **Professor_Turma**

Tabela associativa para a relação N:M entre Colaborador (Professor) e Turma.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_professor	INT	PK, FK	Refere-se a Colaborador(id_colaborador).
id_turma	INT	PK, FK	Refere-se a Turma(id_turma).

8. Tabela **Aluno**

Registo dos estudantes.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_aluno	INT	PK	AUTO_INCREMENT
nome_completo	VARCHAR(255)		NOT NULL
email	VARCHAR(100)		NOT NULL, UNIQUE
data_nascimento	DATE		NOT NULL
status_propina	ENUM('Paga', 'Pendente')		NOT NULL, DEFAULT 'Pendente'. Para a RN02.

9. Tabela **Matricula**

Tabela associativa para a relação N:M entre Aluno e Turma.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_matricula	INT	PK	AUTO_INCREMENT
id_aluno	INT	FK	NOT NULL. Refere-se a Aluno(id_aluno).
id_turma	INT	FK	NOT NULL. Refere-se a Turma(id_turma).
data_matricula	DATETIME		NOT NULL, DEFAULT CURRENT_TIMESTAMP
<i>Constraint</i>			UNIQUE(id_aluno, id_turma) para evitar matrículas duplicadas.

10. Tabela **Avaliacao**

Registo das notas dos alunos.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_avaliacao	INT	PK	AUTO_INCREMENT
id_matricula	INT	FK	NOT NULL. Refere-se a Matricula(id_matricula).
descricao	VARCHAR(100)		(ex: 'Prova 1', 'Trabalho Prático').
nota	DECIMAL(4, 2)		NOT NULL, CHECK (nota BETWEEN 0 AND 20) (RN03).
peso	DECIMAL(3, 2)		NOT NULL, CHECK (peso > 0 AND peso <= 1). Para cálculo da média.

III. Módulo: **Gestão Operacional**

11. Tabela **Empresa_Terceirizada**

Registo das empresas contratadas.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_empresa	INT	PK	AUTO_INCREMENT
razao_social	VARCHAR(255)		NOT NULL, UNIQUE
nif	VARCHAR(30)		NOT NULL, UNIQUE
tipo_servico	ENUM('Limpeza', 'Segurança', 'Cafetaria')		NOT NULL

12. Tabela **Contrato**

Detalhes dos contratos com as empresas.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_contrato	INT	PK	AUTO_INCREMENT
id_empresa	INT	FK	NOT NULL. Refere-se a Empresa_Terceirizada (id_empresa).
data_inicio	DATE		NOT NULL
data_fim	DATE		NOT NULL
valor_mensal	DECIMAL(15, 2)		NOT NULL, CHECK (valor_mensal > 0)
sla_acordado	DECIMAL(5, 2)		NOT NULL, DEFAULT 90.00. Em percentagem.
data_validade_garantia	DATE		NOT NULL. Para a RN04.

13. Tabela **Pagamento**

Registro dos pagamentos mensais aos fornecedores.

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_pagamento	INT	PK	AUTO_INCREMENT
id_contrato	INT	FK	NOT NULL. Refere-se a Contrato(id_contrato)
data_pagamento	DATE		NOT NULL
mes_referencia	INT		NOT NULL, CHECK (mes_referencia BETWEEN 1 AND 12)
ano_referencia	INT		NOT NULL
percentual_sla_apurado	DECIMAL(5, 2)		NOT NULL. O SLA real medido no mês.
valor_multa	DECIMAL(15, 2)		DEFAULT 0.00. Calculado com base na RNO5.
valor_pago	DECIMAL(15, 2)		NOT NULL. Valor final (valor_mensal - valor_multa).

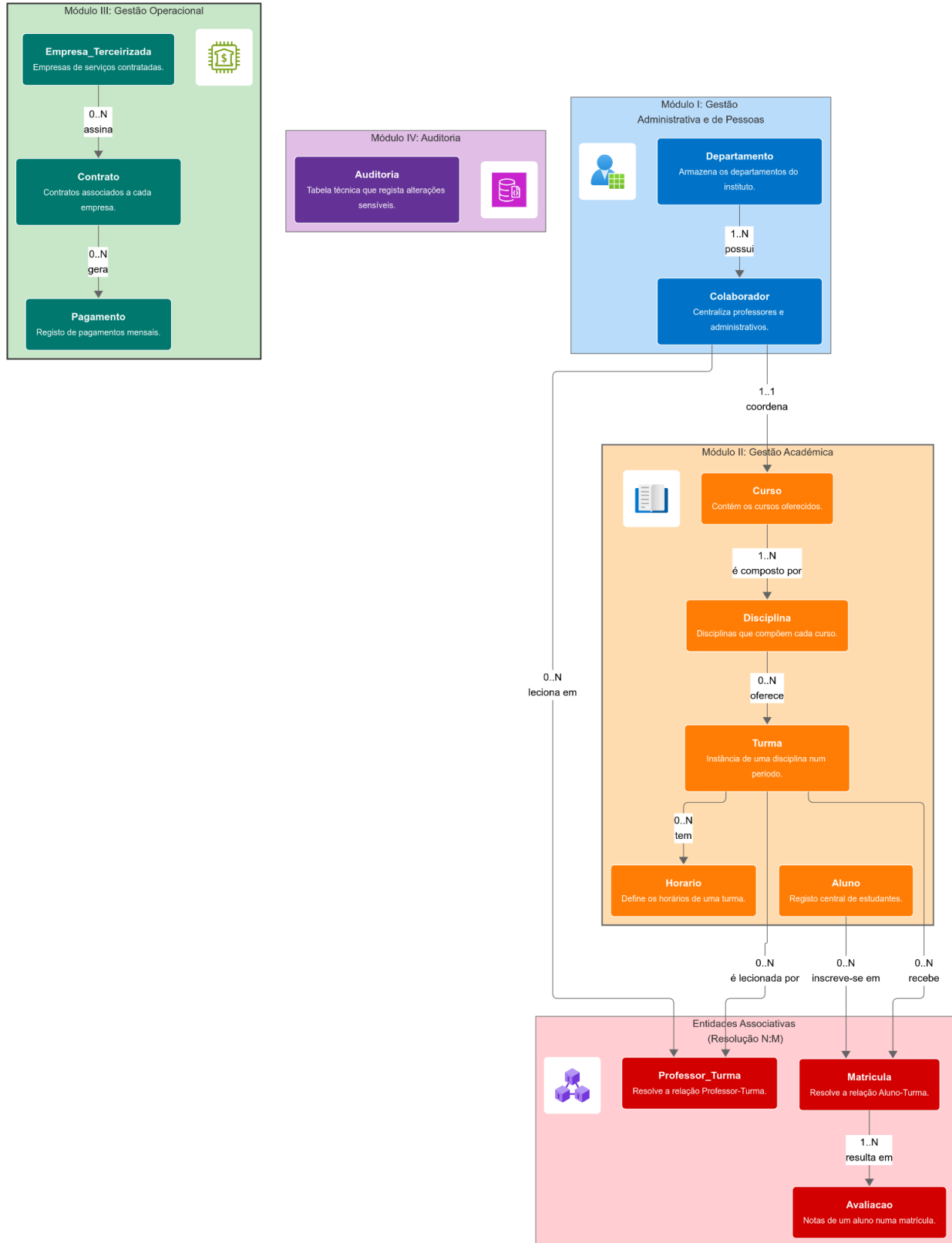
IV. Módulo: Auditoria

14. Tabela Auditoria

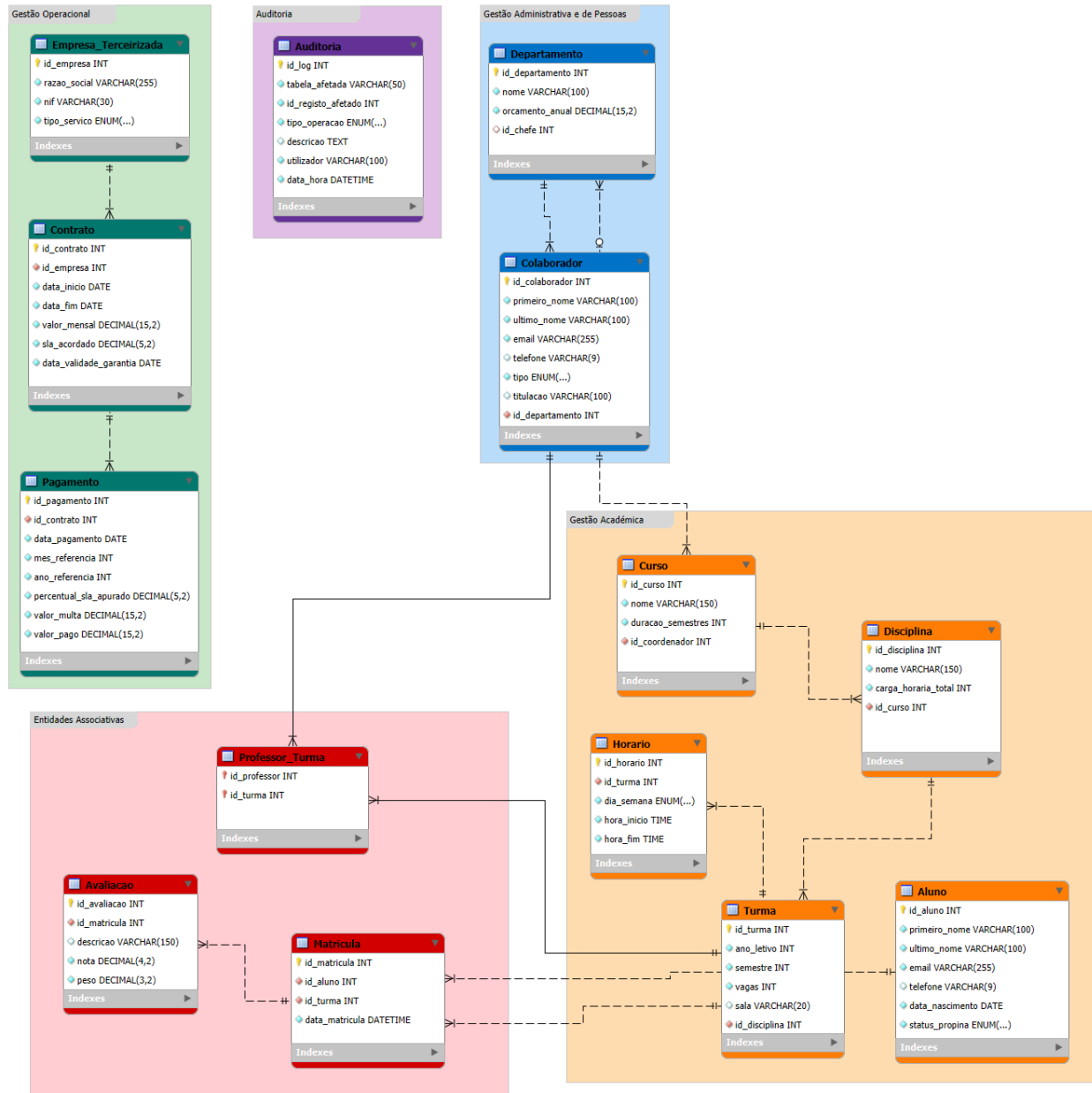
Regista alterações importantes no sistema para rastreabilidade (RF11).

Nome do Atributo	Tipo de Dados	Chave	Restrições/Notas
id_log	INT	PK	AUTO_INCREMENT
tabela_afetada	VARCHAR(50)		NOT NULL (ex: 'Matricula', 'Pagamento').
id_registo_afetado	INT		NOT NULL. ID do registo na tabela afetada.
tipo_operacao	ENUM('INSERT', 'UPDATE', 'DELETE')		NOT NULL
descricao	TEXT		Descrição da alteração (ex: "Aluno X matriculado na turma Y").
utilizador	VARCHAR(100)		NOT NULL. Utilizador do SGBD que fez a alteração.
data_hora	DATETIME		NOT NULL, DEFAULT CURRENT_TIMESTAMP

✓ 1.3. Modelo Conceitual



✓ 1.4. Modelo Lógico



- ☒ ~~Fase 2: Implementação da Base de Dados (DDL)~~
- ☒ ~~Fase 2.5: População de dados antecipada.~~
- ☒ ~~Fase 3: Implementação da Lógica de Negócio (Procedures, Functions, Triggers)~~

Esta é a parte onde implementamos as regras de negócio mais complexas.

☒ ~~3.1. Stored Procedures~~

☒ ~~sp_matricular_aluno(id_aluno, id_turma):~~

1. Verificar se a turma tem vagas disponíveis.
2. Verificar se o aluno tem a propina paga (pode ser uma coluna status_propina na tabela Aluno).
3. Se ambas as condições forem verdadeiras, insere um registo na tabela Matricula. Caso contrário, retorna uma mensagem de erro. (RN02)

☒ ~~sp_alocar_professor(id_professor, id_turma):~~

1. Verificar o horário da turma que se quer alocar.
2. Verificar se o professor já tem outra turma alocada nesse mesmo horário.
3. Se não houver conflito, aloca o professor à turma (insere na tabela Professor_Turma). Caso contrário, retorna erro. (RN01)

☒ ~~sp_processar_pagamento(id_contrato, mes, ano, percentual_sla):~~

1. Verificar se a empresa associada ao contrato tem uma garantia válida (RN04 - pode ser verificado com um Trigger, mas a procedure também pode fazer a verificação).
2. Calcular o valor da multa se o percentual_sla for inferior a 90% (RN05).
3. Inserir o registo na tabela Pagamento com o valor final (valor base - multa).

☒ ~~3.2. Functions~~

☒ ~~fn_calcular_media_aluno(id_aluno, id_disciplina):~~

1. Recebe o ID do aluno e da disciplina.
2. Busca todas as avaliações (nota e peso) desse aluno nessa disciplina.
3. Calcula a média ponderada e retorna o valor.

☒ ~~fn_calcular_sla_mensal(id_contrato, mes, ano):~~

1. Esta função pode ser teórica, mas a ideia seria receber o ID do contrato e o período.

2. Buscaria os registos de performance (que podem vir de uma tabela não especificada de "Ocorrências de SLA") e calcularia o percentual. Para o projeto, pode receber o percentual diretamente na procedure de pagamento.

☒ 3.3. Triggers

☒ ~~trg_auditoria_matriculas:~~

- Disparado AFTER INSERT na tabela Matricula.
- Insere um registo na tabela Auditoria com informações sobre quem matriculou, quem foi matriculado e quando.

☒ ~~trg_auditoria_pagamentos:~~

- Disparado AFTER INSERT na tabela Pagamento.
- Insere um registo na tabela Auditoria informando o valor pago, a empresa e a data.

☒ ~~trg_bloquear_pagamento_sem_garantia:~~

- Disparado BEFORE INSERT na tabela Pagamento.
- Verifica se o contrato associado tem uma garantia válida (ex: uma coluna data_validade_garantia na tabela Contrato).
- Se a garantia não for válida, cancela a operação de inserção e lança um erro (RN04).

☒ Fase 4: Visualização e Testes

Nesta fase, criamos formas de ver os dados e garantimos que tudo funciona como esperado.

☒ 4.1. Views

☒ ~~vw_grade_horaria_curso:~~ Junta as tabelas Curso, Disciplina, Turma e Professor_Turma para mostrar o horário completo de um curso, com disciplina, professor e sala.

☒ ~~vw_carga_horaria_professor:~~ Mostra o total de horas de aula atribuídas a cada professor num determinado período.

☒ ~~vw_resumo_custos_servicos:~~ Agrupa os dados da tabela Pagamento por mês e por tipo de serviço (limpeza, segurança) para mostrar um resumo dos custos operacionais.

☒ 4.2. Consultas de Teste (DML - INSERT e SELECT)

☒ ~~INSERTs:~~ Crie pelo menos 10 INSERTs realistas para cada tabela principal para popular a base de dados com dados de teste.

☒ ~~SELECTs:~~ Escreva pelo menos 15 consultas (SELECT) que demonstrem que as regras de negócio e os requisitos funcionais estão a ser cumpridos.

- Exemplo: Tentar matricular um aluno numa turma cheia (deve falhar).
- Exemplo: Listar todos os alunos com média abaixo de 10 numa disciplina.
- Exemplo: Mostrar os pagamentos com multa de um determinado contrato.

☒ ~~Fase 5: Documentação e Entrega~~

A fase final para organizar e apresentar o seu trabalho.

☒ ~~Checklist para esta fase:~~

☒ ~~[+] Relatório de Regras de Negócio:~~ Escrever 1 página a explicar como cada RN foi implementada (ex: "A RN01 foi implementada na Procedure sp_alocar_professor que verifica conflitos de horário...").

☒ ~~[+] Documentação Técnica:~~ Incluir o diagrama físico final, um guia simples de como executar os seus scripts e justificativas para a criação de índices específicos.

☒ ~~[+] Repositório Git:~~

- Crie um repositório no GitHub.
- Faça commits incrementais (ex: "Fase 1: Modelo ER concluído", "Fase 2: Script DDL para tabelas académicas", etc.).
- Use tags para marcar as versões principais (ex: v1.0-modelo-final, v2.0-implementacao-final).

☒ ~~Documentação Técnica:~~

☒ ~~Relatório de Implementação das Regras de Negócio no SGBD:~~