

Projeto Lógico de Banco de Dados (2ª VA)

Prof.^a Dra.: Priscilla Kelly Machado Azevedo

Integrantes do Grupo:

1. Pedro Henrique Matos Oliveira
2. João Henrique Araújo de Souza
3. Antônio Carlos da Silva Batista Vaz
4. José Uilton Ferreira de Siqueira

1 Esquema Lógico Relacional

O presente documento tem como objetivo apresentar a transformação do esquema conceitual (MERE), desenvolvido na primeira etapa da avaliação, em um esquema lógico relacional. Esta etapa foca na estruturação das tabelas reais que garantirão o suporte aos dados, definindo os tipos de dados, restrições e relacionamentos necessários para a implementação do banco de dados.

1.1 Descrição e Estrutura do Diagrama

O diagrama apresentado na Figura 1 detalha a estrutura de tabelas, chaves primárias (PK) e chaves estrangeiras (FK) resultantes da normalização. Dentre as principais decisões de projeto visíveis na imagem, destacam-se:

- **Especialização:** A entidade "Usuário" atua como tabela pai para "Estudante" e "Empresa", compartilhando o mesmo identificador para evitar redundância de dados de login.
- **Relacionamentos N:N:** As associações complexas foram transformadas em tabelas associativas (como *Estuda*, *Possui* e *Requer*), garantindo a integridade referencial.
- **Atributos Multivalorados:** O telefone do estudante foi isolado em uma tabela própria (*Telefone_Estudante*) para atender à Primeira Forma Normal.

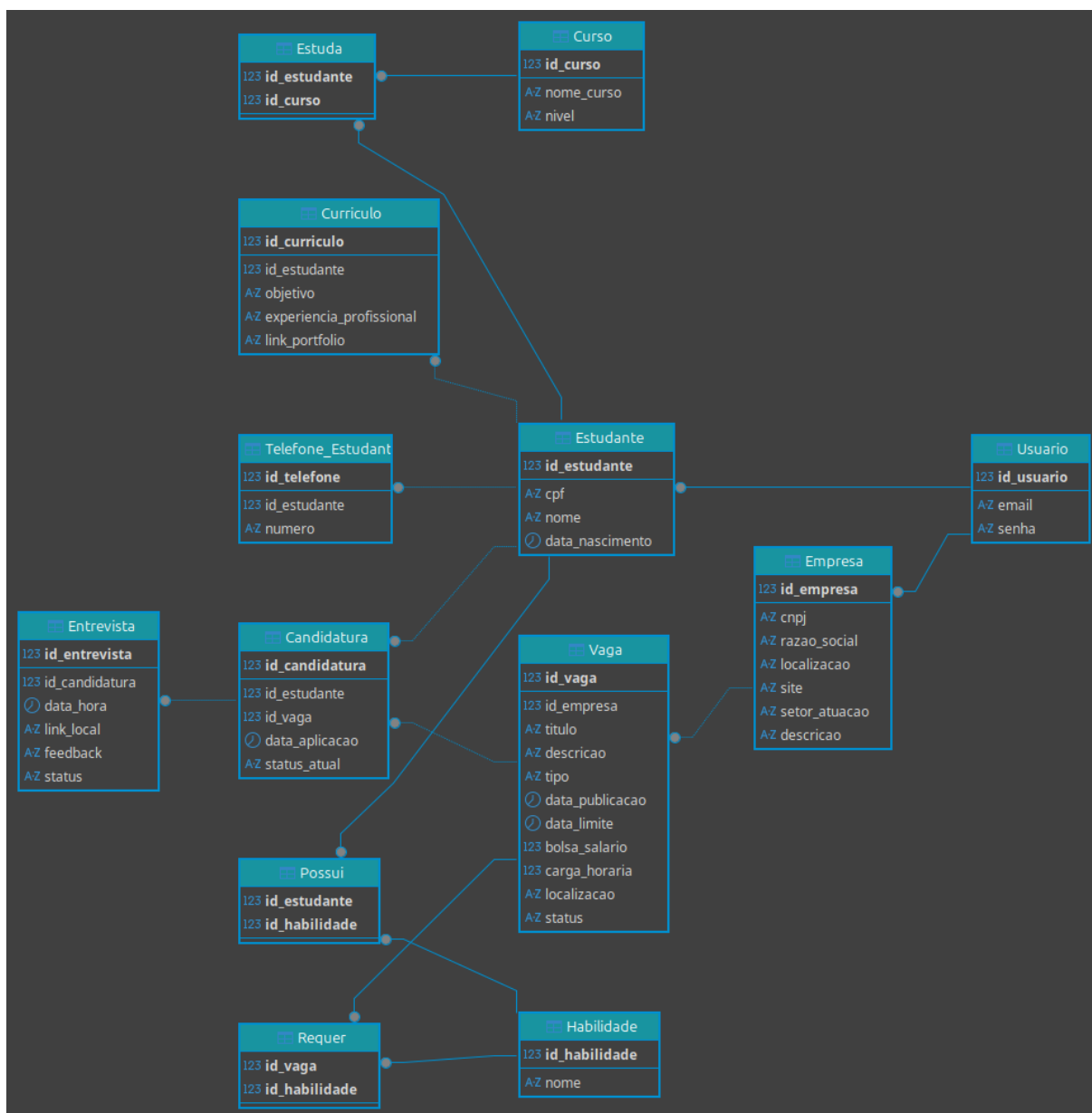


Figura 1: Diagrama Relacional do Sistema de Vagas e Currículos

2 Dicionário de Dados

A seguir, é apresentado o dicionário de dados detalhado. Cada subseção descreve a finalidade da tabela, seguida pela especificação técnica de seus atributos, tipos de dados, restrições de integridade e semântica.

2.1 Tabela: Usuario

Descrição da Tabela: Tabela pai que centraliza o acesso (login) do sistema. Armazena as credenciais de autenticação compartilhadas por Estudantes e Empresas (Especialização/Generalização).

Atributo	Tipo	Restrições	Semântica/Descrição
id_usuario	INT	PK, AI, Not Null	Identificador único e sequencial do usuário no sistema (Chave Surrogada).
e_mail	VARCHAR(100)	Unique, Not Null	Endereço de e-mail utilizado como login. Não pode haver duplicatas.
senha	VARCHAR(100)	Not Null	Hash da senha de autenticação do usuário.

2.2 Tabela: Empresa

Descrição da Tabela: Especialização da entidade Usuário. Armazena os dados jurídicos e institucionais das empresas parceiras que ofertam vagas.

Atributo	Tipo	Restrições	Semântica/Descrição
id_empresa	INT	PK, FK, Not Null	Chave estrangeira referenciando Usuario . Garante a relação 1:1 da herança.
cnpj	VARCHAR(14)	Unique, Not Null	Cadastro Nacional de Pessoa Jurídica (apenas números).
razao_social	VARCHAR(80)	Not Null	Nome oficial/jurídico da empresa.
localizacao	VARCHAR(50)	-	Endereço físico ou cidade sede da empresa.
site	VARCHAR(1000)	-	URL do site oficial ou página da empresa.
setor_atuacao	VARCHAR(30)	-	Segmento de mercado (ex: Tecnologia, Saúde, Varejo).
descricao	LONGTEXT	-	Texto livre para apresentação institucional da empresa.

2.3 Tabela: Estudante

Descrição da Tabela: Especialização da entidade Usuário. Armazena os dados pessoais dos alunos que se candidatam às vagas.

Atributo	Tipo	Restrições	Semântica/Descrição
id_estudante	INT	PK, FK, Not Null	Chave estrangeira referenciando Usuario . Identifica o aluno univocamente.
cpf	VARCHAR(11)	Unique, Not Null	Cadastro de Pessoa Física (apenas números).
nome	VARCHAR(100)	Not Null	Nome completo civil do estudante.
data_nascimento	DATE	-	Data de nascimento para cálculo de idade/elegibilidade.

2.4 Tabela: Telefone_Estudante

Descrição da Tabela: Tabela criada para atender à 1ª Forma Normal (1FN), armazenando os contatos telefônicos dos estudantes (atributo multivalorado).

Atributo	Tipo	Restrições	Semântica/Descrição
id_telefone	INT	PK, AI, Not Null	Identificador único do registro de telefone.
id_estudante	INT	FK, Not Null	Referência ao estudante dono deste número.
numero	VARCHAR(13)	Not Null	Número do telefone com DDD (formato padrão).

2.5 Tabela: Curriculo

Descrição da Tabela: Armazena as informações profissionais e objetivos do estudante. Possui relacionamento 1:1 com a tabela Estudante.

Atributo	Tipo	Restrições	Semântica/Descrição
id_curriculo	INT	PK, AI, Not Null	Identificador interno do currículo.
id_estudante	INT	FK, Unique	Referência ao estudante. A restrição UNIQUE garante que cada aluno tenha apenas um currículo.
objetivo	LONGTEXT	-	Texto descrevendo o objetivo profissional do aluno.
experiencia	MEDIUMTEXT	-	Resumo das experiências profissionais anteriores.
link_portfolio	VARCHAR(1000)	-	URL externa para portfólio (GitHub, Behance, etc.).

2.6 Tabela: Habilidade

Descrição da Tabela: Catálogo de competências (Hard Skills) que podem ser associadas tanto a estudantes quanto a vagas.

Atributo	Tipo	Restrições	Semântica/Descrição
id_habilidade	INT	PK, AI, Not Null	Identificador único da habilidade.
nome	VARCHAR(30)	Not Null	Nome da competência (ex: Java, SQL, Inglês).

2.7 Tabela: Possui

Descrição da Tabela: Tabela associativa que implementa o relacionamento N:N entre Estudante e Habilidade.

Atributo	Tipo	Restrições	Semântica/Descrição
id_estudante	INT	PK, FK	Parte da chave composta. Referencia o estudante.
id_habilidade	INT	PK, FK	Parte da chave composta. Referencia a habilidade que o aluno possui.

2.8 Tabela: Curso

Descrição da Tabela: Catálogo de cursos acadêmicos ou complementares disponíveis no sistema.

Atributo	Tipo	Restrições	Semântica/Descrição
id_curso	INT	PK, AI, Not Null	Identificador único do curso.
nome_curso	VARCHAR(80)	Not Null	Nome oficial do curso (ex: Ciência da Computação).
nivel	VARCHAR(50)	-	Grau acadêmico (ex: Bacharelado, Técnico, Pós-graduação).

2.9 Tabela: Estuda

Descrição da Tabela: Tabela associativa que implementa o relacionamento N:N entre Estudante e Curso.

Atributo	Tipo	Restrições	Semântica/Descrição
id_estudante	INT	PK, FK	Parte da chave composta. Referencia o estudante.
id_curso	INT	PK, FK	Parte da chave composta. Referencia o curso que o aluno está realizando.

2.10 Tabela: Vaga

Descrição da Tabela: Armazena as oportunidades de emprego ou estágio publicadas pelas empresas.

Atributo	Tipo	Restrições	Semântica/Descrição
id_vaga	INT	PK, AI, Not Null	Identificador único da vaga.
id_empresa	INT	FK, Not Null	Referência à empresa ofertante.
titulo	VARCHAR(50)	Not Null	Título da oportunidade (ex: Desenvolvedor Júnior).
descricao	LONGTEXT	-	Descrição detalhada das atividades e responsabilidades.
tipo	VARCHAR(50)	-	Modalidade de contratação (ex: Estágio, CLT, PJ).
bolsa_salario	DECIMAL(10,2)	-	Valor monetário da remuneração (10 dígitos, 2 decimais).
status	VARCHAR(30)	Default 'Aberta'	Estado atual da vaga (ex: 'Aberta', 'Fechada', 'Pausada').

2.11 Tabela: Requer

Descrição da Tabela: Tabela associativa que implementa o relacionamento N:N entre Vaga e Habilidade (requisitos da vaga).

Atributo	Tipo	Restrições	Semântica/Descrição
id_vaga	INT	PK, FK	Parte da chave composta. Referencia a vaga.
id_habilidade	INT	PK, FK	Parte da chave composta. Referencia a habilidade exigida pela vaga.

2.12 Tabela: Candidatura

Descrição da Tabela: Registra o interesse de um estudante em uma vaga específica. Controla o fluxo do processo seletivo.

Atributo	Tipo	Restrições	Semântica/Descrição
id_candidatura	INT	PK, AI, Not Null	Identificador único da aplicação.
id_estudante	INT	FK, Not Null	Estudante que se candidatou.
id_vaga	INT	FK, Not Null	Vaga para a qual o estudante aplicou.
data_aplicacao	TIMESTAMP	Default Current	Data e hora exata do registro da candidatura.
status_atual	VARCHAR(30)	-	Fase atual do candidato (ex: 'Enviada', 'Em Análise', 'Aprovado', 'Rejeitado').

2.13 Tabela: Entrevista

Descrição da Tabela: Gerencia os agendamentos de entrevistas vinculados a uma candidatura existente.

Atributo	Tipo	Restrições	Semântica/Descrição
id_entrevista	INT	PK, AI, Not Null	Identificador único da entrevista.
id_candidatura	INT	FK, Not Null	Referência à candidatura vinculada.
data_hora	TIMESTAMP	Not Null	Data e horário agendado para a entrevista.
link_local	VARCHAR(1000)	-	Link da reunião online ou endereço físico.
feedback	TEXT	-	Parecer do recrutador após a entrevista.
status	VARCHAR(30)	-	Situação do agendamento (ex: 'Agendada', 'Concluída', 'Cancelada').

3 Normalização

O processo de modelagem do banco de dados seguiu rigorosamente as regras de normalização para evitar redundâncias e anomalias. O esquema lógico resultante atende à Segunda Forma Normal (2FN), conforme justificado abaixo:

3.1 Primeira Forma Normal (1FN)

Para atender à 1FN, garantiu-se que todos os atributos fossem atômicos e monovalorados.

- **Decomposição de Multivalorados:** No modelo conceitual, o atributo "telefone" do estudante era multivalorado. Para respeitar a 1FN, este atributo foi removido da tabela *Estudante* e transformado em uma tabela própria denominada *Telefone_Estudante*, onde cada número ocupa uma linha distinta vinculada ao ID do aluno.

3.2 Segunda Forma Normal (2FN)

O esquema está em conformidade com a 2FN, pois não existem dependências parciais nas tabelas que possuem chaves primárias compostas.

- **Tabelas Associativas:** As tabelas oriundas de relacionamentos N:N, como *Estuda* (Estudante-Curso), *Possui* (Estudante-Habilidade) e *Requer* (Vaga-Habilidade), possuem chaves compostas. Nestas tabelas, não existem colunas não-chave que dependam apenas de uma parte da chave.
- **Integridade:** Nas demais tabelas com chaves simples (como *Vaga* ou *Empresa*), todos os atributos dependem totalmente da chave primária inteira.

4 Carga de Dados e Povoamento

O povoamento do banco de dados foi realizado através da execução de scripts de Manipulação de Dados (DML) contendo comandos *INSERT* padrão da linguagem SQL.

Optou-se pela criação manual de um conjunto de dados fictícios para validar as regras de integridade referencial e simular o funcionamento real do sistema. O processo foi estruturado da seguinte forma:

- **Script de Carga:** Foi desenvolvido um arquivo dedicado (*dml.sql*) contendo as inserções ordenadas para respeitar as dependências das chaves estrangeiras (ex: primeiro cria-se o *Usuário*, depois o *Estudante*, depois o *Currículo*).
- **Automação via Docker:** Para garantir a reprodutibilidade do ambiente, o script de carga foi configurado no container Docker para ser executado automaticamente logo após a criação das tabelas (DDL).
- **Dados de Teste:** Foram inseridos registros em todas as tabelas principais e associativas (como *Candidatura* e *Possui*), permitindo testar consultas e verificar o comportamento das restrições de unicidade e não-nulidade.