

**Trabalho Final**  
**SCC0240 – Bases de Dados**  
**Profa. Mirela Teixeira Cazzolato**  
**Grupo 12**

**Links:**

**Github:** <https://github.com/FelipeVolkweis/bd-final>

**Video:** <https://youtu.be/zCdC23oX5Ms>

**Exercício 1:** Descreva, em forma de texto corrido, todas os conjuntos entidades (CE) identificados na problemática e seus respectivos atributos. Em seguida, identifique todas os conjuntos relacionamentos (CRs) entre as entidades, sempre explicando a lógica por trás da escolha da multiplicidade e participação de cada entidade na relação. Identifique também os conjuntos entidades fracos (CEF, se houver), juntamente com CRT que o define como fraco, atributos multivalorados, atributos compostos, atributos de CRs.

No contexto do sistema acadêmico em questão, identificamos diversos Conjuntos de Entidades (CE) que representam os principais objetos manipulados pelo banco de dados, bem como seus atributos e restrições associadas.

O CE Curso reúne todas as informações sobre as formações oferecidas pela instituição. Cada curso é identificado por um Código único e obrigatório, que o distingue de todos os demais. Além do código, registram-se o Nome do curso, o Nível de Ensino (atributo categórico que assume um dos valores Fundamental, Médio, Técnico ou Graduação), a Carga Horária, o Número de Vagas disponíveis e a Ementa, que detalha o conteúdo programático. A unicidade e não nulidade do código garantem a integridade referencial sempre que um curso for citado em outros contextos. Além disso, um curso conta com um conjunto de regras e um conjunto de infraestrutura, ambos como atributos multivalorados.

O CE Disciplina diz respeito às disciplinas que compõem os cursos. Cada disciplina possui uma Sigla única e obrigatória, um número de Aulas Semanais, informações sobre o Material Didático Básico e Capacidade da Turma. A restrição de sigla única e não nula assegura que não haja ambiguidade na identificação de cada disciplina.

O CE genérico Usuário engloba todos os perfis que interagem com o sistema. Seus atributos são Nome, Sobrenome, Telefone, Data de Nascimento, Endereço, Sexo, Email e Senha. O endereço é um atributo composto, detalhado em Cep, Número e Complemento, e o sexo é categórico, aceitando M, F ou O. A combinação de nome, sobrenome e telefone deve ser única e não nula, conforme a especificação, prevenindo duplicidades de cadastro. A partir desse CE genérico, aplicamos a abstração de generalização para criar os CE específicos Professor, Aluno e Administrador, pois cada um desses papéis dispõe de atributos particulares (por exemplo, o professor registra Área e Titulação) e mantém relacionamentos distintos com outras entidades.

O CE Departamento Acadêmico caracteriza as divisões administrativas responsáveis pelos cursos e disciplinas. Cada departamento tem um Código único e obrigatório, além de um Nome descritivo.

O CE Unidade refere-se aos campi ou unidades físicas da instituição, definidos pelos atributos Cidade, Estado, País e Prédio/Bloco. A combinação de cidade, estado e país deve ser única e não nula, evitando a duplicação de registros para localidades diferentes.

O CEA Mensagem armazena comunicados trocados entre usuários ou grupos: registra um Timestamp de envio e o Texto da mensagem, ambos obrigatórios, pois Timestamp é chave adicional e pelo fato de que não existe mensagem sem texto. Para viabilizar comunicações direcionadas, introduzimos o CE Grupo, identificado por um ID, que permite agrupar usuários (por exemplo, todos os alunos de uma turma) e vincular uma única mensagem a todo o grupo de destinatários, reduzindo redundâncias.

O CEA Oferecimento liga cursos, disciplinas, unidades e períodos letivos, e contém apenas o atributo PeríodoLetivo, indicado de forma obrigatória pois é chave adicional, e serve para agendar quais disciplinas são oferecidas em cada período.

Por fim, definimos que o CE fraco Sala depende da existência de uma Unidade, que é a entidade forte dominante. Cada sala, identificada pelo NúmeroSala, armazena também sua Capacidade. A definição de sala como entidade fraca reforça a necessidade de sua associação explícita à unidade que a contém.

Em conjunto, esses CE e seus atributos formam a base da modelagem conceitual do sistema, contemplando tanto a integridade dos dados quanto a flexibilidade necessária para representar diferentes papéis de usuário e estruturas acadêmicas.

## **Conjunto de Relacionamentos e seus atributos:**

### **Relacionamento DáAulaEm entre Professor e Unidade:**

Uma unidade pode ter (0, N) Professores e um professor pode participar de (1, 1) Unidade. Definimos que cada professor só pode ministrar em uma unidade pois mesmo que existissem unidades perto em que ele poderia lecionar, a logística de horários e deslocamento seria mais complexa, pois seria mais difícil alocar aulas em duas unidades específicas, corre risco de conflito de horários, atrasos pelo deslocamento, além de possível custo adicional para a instituição para transporte.

### **Relacionamento EstudaEm entre Aluno e Unidade:**

Uma unidade pode ter (0, N) Alunos e um aluno pode ser inscrito em (1, 1) Unidade. Uma unidade pode possuir vários alunos matriculados, porém cada aluno só pode se matricular em uma unidade. A escolha para um aluno poder se matricular em apenas uma instituição tem a mesma motivação que Professor.

### **Relacionamento Chefiar entre Professor e Departamento:**

Cada Professor pode ser chefe de (0,1) departamento e cada departamento deve ser chefiado por (1, 1) professor apenas. Pela descrição, é definido que cada professor pode ser chefe de apenas um departamento por vez, porém definimos também que apenas um professor pode ser chefe do mesmo departamento, primeiro pela semântica da palavra chefe que sugere apenas um, mas também para evitar conflito de autoridades.

**Relacionamento PertenceA entre Curso e Departamento:**

Um curso pode pertencer a (1, 1) Departamento, e um departamentos pode ter (1, N) Cursos, já que um curso depende a um departamento unicamente, e cada departamento pode ter diversos cursos.

**Relacionamento Compor entre Disciplina e Curso:**

Uma Disciplina pode ser composta por (1, N) Cursos, e um Curso pode compor (1, N) Disciplinas, pois cada curso deve ter no mínimo uma disciplina para existir, podendo ter várias, e uma disciplina deve pertencer a no mínimo um curso, podendo existir em vários.

**Relacionamento ÉCursoPreRequisito entre Curso e Curso:**

Um Curso pode ser pré-requisito de (0,N) Cursos e um Curso pode ter (0,N) Cursos como pré-requisitos, pois um curso pode ter nenhum ou vários cursos como pré requisitos, e cada curso pode ser pré requisito de vários cursos.

**Relacionamento ÉDiscPreRequisito entre Disciplina e Curso:**

Uma Disciplina pode ser pré-requisito de (0,N) Cursos e um Curso pode ter (0,N) Disciplinas como pré-requisitos, bem como Curso, pois cada disciplina pode ser pré-requisito de nenhum ou alguns cursos, e cada curso pode ter nenhuma ou algumas disciplinas como pré-requisito.

**Relacionamento Sala PertenceA Unidade:**

Uma Sala pertence a (1,1) Unidade e uma Unidade possui (1,N) Salas, pois cada sala deve pertencer a uma unidade para existir, e apenas a uma unidade, e cada unidade pode ter várias salas.

**Relacionamento PertenceA entre Departamento e Curso:**

Um Curso pertence a (1,1) Departamento e um Departamento possui (1, N) Cursos, pois cada curso deve pertencer a um departamento apenas obrigatoriamente, e um departamento deve ter no mínimo um curso para existir.

**Relacionamento Oferecer entre Professor e Disciplina:**

Um Professor pode ofertar (0,N) Disciplinas e uma Disciplina é ofertada por (0,N) Professores, pois nem todo professor deve ofertar uma disciplina, e nem toda entidade de Disciplina terá um professor ministrando ela. Como queremos identificar unicamente cada oferecimento de cada disciplina por certo professor, criaremos uma abstração de agregação, que resulta no CEA Oferecimento que discutiremos anteriormente. É fácil compreender que um professor pode lecionar mais de uma disciplina, mas colocamos que também poderiam ser 0 nos seguintes casos: nem todo professor necessariamente estará ministrando alguma disciplina em certo período. Um docente recém-contratado pode ainda não ter sido alocado em nenhuma turma, um professor em licença não entrará na lista de ofertas, e um pesquisador que coordena projetos, mas não leciona naquele semestre. Da mesma forma, uma disciplina pode não ser ofertada em um determinado semestre devido a baixa demanda, ajuste curricular ou falta de corpo docente disponível.

### **Relacionamento ÉResponsávelPor entre Professor e Disciplina:**

Uma Disciplina tem (1,N) Professores responsáveis e cada Professor é responsável por (0,N) Disciplinas. Isso porque cada disciplina precisa ter pelo menos um professor responsável por ela, porém podem existir professores que não são responsáveis por nenhuma disciplina. Permitir que uma disciplina tenha mais de um professor responsável foi uma escolha feita pensando em situações de compartilhamento de tarefas acadêmica, como por exemplo uma disciplina que possui uma parte teórica e uma parte prática, ou uma abordagem interdisciplinar. Auxilia também em divisão de carga, portanto decidimos manter a opção de mais de um professor responsável.

### **Relacionamento Matricular entre Aluno e Oferecimento:**

Aluno pode se relacionar com (1, N) Oferecimentos, e um Oferecimento pode possuir (1,N) Alunos. Definimos que cada aluno deve participar de, no mínimo, uma oferta de disciplina e que cada oferta deve contar com, pelo menos, um aluno matriculado para refletir a seguinte lógica: um aluno sem matrícula não está ativo no curso, e uma turma sem alunos não justifica sua abertura. Esse relacionamento tem os atributos Data de Matrícula, Notas, Bolsas de Estudo e Descontos, Status e Taxas.

Atributo Categórico Status pode receber os valores Confirmada, Pendente, Cancelada, Concluída, Reprovada, Trancada.

Atributos Notas, Bolsas e Descontos são compostos.

### **Relacionamento Avaliar entre Aluno e Oferecimento**

Um aluno pode avaliar (0, N) Oferecimentos e oferecimento pode ser avaliado por (0, N) Oferecimento. Um aluno pode não avaliar nenhum oferecimento que participa, como pode avaliar mais de um. Um oferecimento pode não ter nenhuma avaliação de seus alunos, ou então ter várias avaliações por alunos que realizaram ele.

### **Relacionamento OferecidoEm entre Oferecimento em Sala**

Um oferecimento pode ser realizado em (1,1) sala, e cada sala pode atender (0, N) Oferecimentos. Definimos que cada instância de Oferecimento ocupa exatamente uma sala porque, na prática, alocar uma única sala a cada turma evita qualquer ambiguidade sobre onde as aulas ocorrerão: horários, equipamentos disponíveis e acessos ficam claramente vinculados a um único espaço físico. Já uma sala pode não receber nenhum oferecimento em determinados horários ou semestres e, ao mesmo tempo, acomodar vários oferecimentos em momentos diferentes.

### **Relacionamento DiscPertenceA entre Disciplina e Unidade**

Uma disciplina pode pertencer a (1, 1) Unidade, e cada unidade pode possuir (1, N) Disciplinas.

### **Relacionamento Unidade CursoPertenceA Curso**

Da mesma forma, um curso pode pertencer (1, 1) Unidade, e cada unidade pode possuir (1, N) Disciplinas.

### **Relacionamento ÉCompostoPor entre Grupo e Usuários**

Um grupo pode ser composto por (1, N) Usuários, e cada usuário pode pertencer a (0, N) grupos.

### **Relacionamento EnviarMensagem entre Usuário e Grupo**

O Usuário pode enviar mensagem para (0, N) grupos, um Grupo pode receber mensagens de (0, N) usuários.. Cada mensagem é enviada por um emissor, e que pode enviar mensagens para vários grupos, e cada grupo determinado pode receber diversas mensagens. As duas entidades se relacionam de forma a criar um novo objeto, e usamos a abstração de agregação para definir essa nova entidade Mensagem. Uma abstração de agregação deve ser colocada na modelagem apenas quando é necessário, segundo um dos dois casos: quando é preciso identificar cada instância do conjunto de relacionamentos, ou quando é necessário mais de uma instância do mesmo conjunto de relacionamentos envolvendo as mesmas entidades, o que é o caso nessa situação, pois um mesmo destinatário pode mandar várias mensagens para um mesmo grupo.

---

**Exercício 2:** Faça a Modelagem E-R da base de dados descrita para o sistema de gestão escolar, utilizando o Modelo Entidade-Relacionamento Estendido (ME-RX). Seu diagrama deve representar explicitamente pelo menos uma abstração de Generalização/Especialização e uma de Agregação, caso se apliquem naturalmente ao problema.

***O Diagrama se encontra no final do PDF como anexo.***

---

**Exercício 3:** Faça o mapeamento completo do Modelo Entidade-Relacionamento Estendido (ME-RX), obtido no Exercício 2, para o Modelo Relacional. SEexercício 6: Desenvolva scripts SQL para:ua resposta deve incluir a definição de todas as relações, seus atributos com os respectivos domínios, as chaves primárias, chaves candidatas e chaves estrangeiras. Justifique as decisões de mapeamento para construções mais complexas (ex: generalizações/especializações, atributos multivalorados, diferentes multiplicidades):.

**Passo 1:** Mapear os CE Regulares:

Usuário = { Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha }

Unidade = { Cidade, Estado, País, Bloco, Prédio }

Departamento = { Código, Nome }

Disciplina = { Sigla, CapacidadeTurma, MaterialBasico, NroAulasSemanais, }

Curso = { Codigo, Nome, NivelEnsino, Ementa, CargaHoraria, NumeroVagas }

Grupo = { Id }

**Passo 2:** Mapear os CE Fracos

Sala = { NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade, Capacidade }

**Passo 2.a:** Mapear os CE Agregação

Tanto o relacionamento gerador da agregação de EnviarMensagem como o de Oferecer não possuem atributos próprios, sendo assim, apenas o conjunto entidade agregação é mapeado em ambos os casos (Mensagem e Oferecimento). Em mensagem, usamos Timestamp como chave adicional junto de Id proveniente do CE grupo e Nome, Sobrenome e Telefone provenientes do CEG usuário. Em oferecimento usamos PeríodoLetivo como chave adicional junto de SiglaDisciplina proveniente do CE Disciplina e NomeProf, SobrenomeProf e TelefoneProf provenientes do CEE Professor.

Mensagem = { Timestamp, GrupoId, UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone, Texto }

Oferecimento = { PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMaxMatricula }

**Passo 2.b:** Mapear os CE Generalização

Usaremos o mapeamento 1G, já que possuímos poucas especializações e estas possuem poucos atributos, além disso, como a generalização é total e exclusiva, optamos por colocar o atributo 'Tipo' direto na relação. Desta forma, os atributos 'Area' e 'Titulacao' provenientes da especialização professor podem ser nulos.

Usuário = { Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha, Tipo, Area, Titulacao }

**Passo 3:** Mapear os CR de cardinalidade 1:1

Departamento = { Código, Nome, NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe }

**Passo 4:** Mapear os CR de cardinalidade 1:N

Os relacionamentos EstudaEm e DaAulaEm são mapeados diretamente no usuário devido a decisão do mapeamento 1G tomada anteriormente.

Os relacionamentos DiscPertenceA e CursoPertenceA são mapeados direto na Disciplina e no Curso, respectivamente.

Usuário = { Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha, Tipo, Area, Titulacao, CidadeUnidade, EstadoUnidade, PaisUnidade }

Oferecimento = { PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMaxMatricula, NroSala, CidadeSala, EstadoSala, PaisSala }

Disciplina = { Sigla, CapacidadeTurma, MaterialBasico, NroAulasSemanais, CidadeUnidade, EstadoUnidade, PaísUnidade }

Curso = { Codigo, Nome, NivelEnsino, Ementa, CargaHoraria, NumeroVagas, NroSala, CidadeSala, EstadoSala, PaisSala, CodigoDepartamento, CidadeUnidade, EstadoUnidade, PaísUnidade }

**Passo 5:** Mapear os CR de cardinalidade M:N

AvaliarOferecimento = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Texto, NotaDidatica, NotaMaterial, NotaConteudo, NotaInfra }

MatricularOferecimento = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMatricula, Status, Taxa }

DisciplinaPreRequisito = { CodigoCurso, SiglaDisciplina }

CursoPreRequisito = { CodigoCurso, CodigoCursoPreReq }

ComposicaoCurso = { CodigoCurso, SiglaDisciplina }

DisciplinasEResponsaveis = { SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf }

GrupoUsuarios = { GrupoId, NomeUsuario, SobrenomeUsuario, TelefoneUsuario }

**Passo 6:** Mapear os CR de ordem > 2 – Não tem

**Passo 7:** Mapear os Atributos Multivalorados

Para cada atributo multivalorado foi criada uma nova relação que tem como chave os atributos chave do conjunto entidade/relacionamento ao qual o atributo multivalorado está associado, além disso, o atributo multivalorado também é acoplado a chave, porém na forma de um atributo monovalorado. Uma atenção especial poderia ser dada a relação NotasMatricula caso soubéssemos de antemão quantas notas um aluno poderia ter (caso fossem três notas apenas, por exemplo, poderíamos ter três campos N1, N2, N3 mapeados diretamente na relação de MatriculaOferecimento ao invés de criar a relação NotasMatricula).

RegrasCurso = { CodigoCurso, Regra }

InfraestruturaCurso = { CodigoCurso, Infraestrutura }

NotasMatricula = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Nota }

BolsasMatricula = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Bolsa }

DescontosMatricula = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Desconto }

### **MRel final:**

**Unidade** = { Cidade, Estado, País, Bloco, Prédio }

Dom(Cidade): '[A-Za-zÀ-ÖØ-öø-ÿ]{2,50}', Dom(Estado): '[A-Za-zÀ-ÖØ-öø-ÿ ]{2,50}',

Dom(País): '[A-Za-zÀ-ÖØ-öø-ÿ]{2,50}', Dom(Bloco): '[A-Za-z0-9]{1,10}', Dom(Prédio): '[A-Za-z0-9]{1,10}'.

- **PK:** (Cidade, Estado, País)

**Sala** = { NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade, Capacidade }

Dom(NroSala): '[0-9]{1,4}', Dom(Capacidade): '[0-9]{1,4}',

- **PK:** (NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade)
- **FK:** (CidadeUnidade, EstadoUnidade, PaisUnidade) → Unidade(Cidade, Estado, País)

**Usuário** = { Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha, Tipo, Area, Titulacao, CidadeUnidade, EstadoUnidade, PaisUnidade }

Dom(Nome): '[A-Za-zÀ-ÖØ-öø-ÿ]{2,50}'

Dom(Sobrenome): '[A-Za-zÀ-ÖØ-öø-ÿ]{2,50}'

Dom(Telefone): '\d{8,15}', Dom(DataNasc): formato ISO 'YYYY-MM-DD'

Dom(EnderecoCep): '\d{8}'

Dom(EnderecoNumero): '[A-Za-z0-9]{1,10}'

Dom(EnderecoBairro): '[A-Za-zÀ-ÖØ-öø-ÿ ]{2,50}'

Dom(Sexo): 'M|F|O'

Dom(Email): '[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}'

Dom(Senha): '\{8,255\}'

Dom(Tipo): 'aluno|professor|administrador'

Dom(Area): '[A-Za-zÀ-ÖØ-öø-ÿ ]{2,100}'

Dom(Titulacao): '[A-Za-z0-9. ]{2,20}'

- **PK:** (Nome, Sobrenome, Telefone)
- **FK:** (CidadeUnidade, EstadoUnidade, PaisUnidade) → Unidade(Cidade, Estado, País)
- **Restrição Adicional:**
  - Email deve ser UNIQUE e NOT NULL.
  - Sexo pode assumir apenas valores M, F, O , CHECK constraint



- Tipo pode assumir apenas valores aluno, professor, administrador, CHECK constraint
- DataNasc NOT NULL,
- EnderecoCep NOT NULL
- EnderecoNumero NOT NULL
- EnderecoBairro NOT NULL
- Sexo NOT NULL
- Senha NOT NULL
- Tipo NOT NULL
- CidadeUnidade NOT NULL
- EstadoUnidade NOT NULL
- PaisUnidade NOT NULL

**Departamento** = { Código, Nome, NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe }

Dom(Codigo): '[0-9]{1,6}'

Dom(Nome): '[A-Za-zÀ-ÖØ-öø-ÿ0-9 .,]{2,100}'

- **PK:** (Código)
- **FK:** (NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe) → Usuário (Nome, Sobrenome, Telefone)
- **Restrição Adicional:**
  - Nome NOT NULL
  - NomeProfChefe NOT NULL
  - SobrenomeProfChefe NOT NULL
  - TelefoneProfChefe NOT NULL

**Disciplina** = { Sigla, CapacidadeTurma, MaterialBasico, NroAulasSemanais, CidadeUnidade, EstadoUnidade, PaísUnidade }

Dom(Sigla): '[A-Z0-9]{1,10}'

Dom(CapacidadeTurma): '[0-9]{1,3}'

Dom(NroAulasSemanais): '[0-9]{1,3}'

Dom(MaterialBasico): '.{0,500}'

- **PK:** (Sigla)
- **FK:** (CidadeUnidade, EstadoUnidade, PaísUnidade) → Unidade(Cidade, Estado, País)
- **Restrição Adicional:**
  - CapacidadeTurma NOT NULL
  - NroAulasSemanais NOT NULL

**Curso** = { Codigo, Nome, NivelEnsino, Ementa, CargaHoraria, NumeroVagas, NroSala, CidadeSala, EstadoSala, PaisSala, CodigoDepartamento, CidadeUnidade, EstadoUnidade, PaísUnidade }

Dom(Codigo): '[0-9]{1,6}'

Dom(Nome): '[A-Za-zÀ-ÖØ-öø-ÿ0-9 .,]{2,100}'

Dom(NivelEnsino): 'fundamental | médio | técnico | graduação',

- **PK:** (Codigo)
- **FK:** (NroSala, CidadeSala, EstadoSala, PaisSala) → Sala (NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade)
- **FK:** (CodigoDepartamento) → Departamento(Codigo)
- **FK:** (CidadeUnidade, EstadoUnidade, PaísUnidade) → Unidade(Cidade, Estado, País)
- **Restrição Adicional:**
  - Nome NOT NULL
  - NivelEnsino NOT NULL
  - Ementa NOT NULL
  - CargaHoraria NOT NULL
  - NumeroVagas NOT NULL
  - CodigoDepartamento NOT NULL

**Grupo** = { Id }

Dom(Id): '[0-9]{1,6}'

- **PK:** (Id)

**Mensagem** = { Timestamp, GrupoId, UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone, Texto }

Dom(Timestamp): formato ISO 'YYYY-MM-DD HH:MI:SS',

Dom(Texto): '[A-Za-zÀ-ÖØ-öø-ÿ0-9 ,.] {1,1000}'

- **PK:** (Timestamp, GrupoId, UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone)
- **FK:** (GrupoId) → Grupo(Id)
- **FK:** (UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone) → Usuário(Nome, Sobrenome, Telefone)
- **Restrição Adicional:**
  - Texto NOT NULL

**Oferecimento** = { PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMaxMatricula, NroSala, CidadeSala, EstadoSala, PaisSala }

Dom(DataMaxMatricula)=formato ISO 'YYYY-MM-DD'

Dom(PeriodoLetivo) = '[0-9]{4}\.[1-3]'

- **PK:** (PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **FK:** (SiglaDisciplina) → Disciplina(Sigla)
- **FK:** (NomeProf, SobrenomeProf, TelefoneProf) → Usuario(Nome, Sobrenome, Telefone)
- **FK:** (NroSala, CidadeSala, EstadoSala, PaisSala) → Sala(NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade)
- **Restrição Adicional:**
  - DataMaxMatricula NOT NULL
  - NroSala NOT NULL
  - CidadeSala NOT NULL
  - EstadoSala NOT NULL

- PaisSala NOT NULL

**AvaliarOferecimento** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Texto, NotaDidatica, NotaMaterial, NotaConteudo, NotaInfra }

Dom(Texto) = '[A-Za-zÀ-ÖØ-öø-ÿ0-9 ,.]{1,1000}'

Dom(NotaDidatica) = decimal  $\in$  [0.0, 10.0]

Dom(NotaMaterial) = decimal  $\in$  [0.0, 10.0]

Dom(NotaConteudo) = decimal  $\in$  [0.0, 10.0]

Dom(NotaInfra) = decimal  $\in$  [0.0, 10.0]

- **PK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **FK:** (NomeAluno, SobrenomeAluno, TelefoneAluno)  $\rightarrow$  Usuario(Nome, Sobrenome, Telefone)
- **FK:** (PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)  $\rightarrow$  Oferecimento(PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **Restrição Adicional:**
  - Texto, NotaDidatica, NotaMaterial, NotaConteudo, NotaInfra podem ser NULL cada um individualmente, mas não podem ser todos NULL ao mesmo tempo.

**MatricularOferecimento** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMatricula, Status, Taxa }

Dom(DataMatricula) = formato ISO 'YYYY-MM-DD'

Dom(Status) = 'confirmada | pendente | cancelada | concluida | reprovada | trancada'

Dom(Taxa) = decimal  $\in$  [0.0, infinito)

- **PK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **FK:** (NomeAluno, SobrenomeAluno, TelefoneAluno)  $\rightarrow$  Usuario(Nome, Sobrenome, Telefone)
- **FK:** (PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)  $\rightarrow$  Oferecimento(PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **Restrição Adicional:**
  - DataMatricula NOT NULL
  - Status NOT NULL

**DisciplinaPreRequisito** = { CodigoCurso, SiglaDisciplina }

- **PK:** (CodigoCurso, SiglaDisciplina)
- **FK:** (CodigoCurso)  $\rightarrow$  Curso(Codigo)
- **FK:** (SiglaDisciplina)  $\rightarrow$  Disciplina(Sigla)

**CursoPreRequisito** = { CodigoCurso, CodigoCursoPreReq }

- **PK:** (CodigoCurso, CodigoCursoPreReq)

- **FK:** (CodigoCurso) → Curso(Codigo)
- **FK:** (CodigoCursoPreReq) → Curso(Codigo)

**ComposicaoCurso** = { CodigoCurso, SiglaDisciplina }

- **PK:** (CodigoCurso, SiglaDisciplina)
- **FK:** (CodigoCurso) → Curso(Codigo)
- **FK:** (SiglaDisciplina) → Disciplina(Sigla)

**DisciplinasEResponsaveis** = { SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf }

- **PK:** (SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- **FK:** (SiglaDisciplina) → Disciplina(Sigla)
- **FK:** (NomeProf, SobrenomeProf, TelefoneProf) → Usuario(Nome, Sobrenome, Telefone)

**GrupoUsuarios** = { GrupoId, NomeUsuario, SobrenomeUsuario, TelefoneUsuario }

- **PK:** (GrupoId, NomeUsuario, SobrenomeUsuario, TelefoneUsuario)
- **FK:** (GrupoId) → Grupo(Id)
- **FK:** (NomeUsuario, SobrenomeUsuario, TelefoneUsuario) → Usuario(Nome, Sobrenome, Telefone)

**RegrasCurso** = { CodigoCurso, Regra }

Dom(Regra) = '[A-Za-zÀ-ÖØ-öø-ÿ0-9 ,.]{1,100}'

- **PK:** (CodigoCurso, Regra)
- **FK:** (CodigoCurso) → Curso(Codigo)

**InfraestruturaCurso** = { CodigoCurso, Infraestrutura }

Dom(Infraestrutura) = '[A-Za-zÀ-ÖØ-öø-ÿ0-9 ,.]{1,100}'

- **PK:** (CodigoCurso, Infraestrutura)
- **FK:** (CodigoCurso) → Curso(Codigo)

**NotasMatricula** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Nota }

Dom(Nota) = decimal ∈ [0.0, 10.0]

- **PK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Nota)
- **FK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf) → MatricularOferecimento(NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)

**BolsasMatricula** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Bolsa }

Dom(Bolsa) = '[A-Za-zÀ-ÖØ-öø-ÿ0-9 ,.]{1,100}'

- **PK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Bolsa)

- **FK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf) → MatricularOferecimento(NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)

**DescontosMatricula** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Desconto }

Dom(Desconto) = decimal  $\in$  [0.0, 100.0]

- **PK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Desconto)
  - **FK:** (NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf) → MatricularOferecimento(NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf)
- 

**Exercício 5:** Analise o esquema relacional obtido no Exercício 3 e justifique a forma normal (ex: 1FN, 2FN, 3FN, BCNF) em que cada tabela se encontra, logo após a modelagem. Depois, com base nas dependências funcionais identificadas, normalize toda a base até a BCNF (considerando as que ainda não se encontram nas FNs). Ao aplicar o processo de normalização, detalhe os passos seguidos e apresente o esquema relacional final, normalizado.

### Dependências Funcionais e Normalização:

Cidade, Estado, País → Bloco, Prédio

**Unidade** = {Cidade, Estado, País; Bloco, Prédio}

A tabela unidade já se encontra normalizada para a 1FN, pois todos os atributos são atômicos e monovalorados, e para a 2FN pois a chave é o trio (Cidade, Estado, País) e os atributos Bloco e Prédio dependem desse trio inteiro. Unidade também está normalizada para a 3FN, pois não existem dependências transitivas, ou seja, Bloco e Prédio não dependem de outro atributo não chave, e está na BCNF pois como só existe uma chave candidata reduz-se a 3FN.

---

NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade → Capacidade

**Sala** = { NroSala, CidadeUnidade, EstadoUnidade, PaisUnidade, Capacidade }

Está na 1FN pois os atributos são atômicos e monovalorados. Também está na 2FN porque Capacidade depende do conjunto completo da chave composta (não de parte dela), e na 3FN porque não há nenhuma dependência indireta entre atributos não-chave. Sala está em BCNF porque pois como só existe uma chave candidata reduz-se a 3FN.

---

Nome, Sobrenome, Telefone  $\rightarrow$  DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha, Tipo, Area, Titulacao, CidadeUnidade, EstadoUnidade, PaisUnidade

Email  $\rightarrow$  Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Senha, Tipo, Area, Titulacao, CidadeUnidade, EstadoUnidade, PaisUnidade

**Usuário** = { Nome, Sobrenome, Telefone, DataNasc, EnderecoCep, EnderecoNumero, EnderecoBairro, Sexo, Email, Senha, Tipo, Area, Titulacao, CidadeUnidade, EstadoUnidade, PaisUnidade }

A tabela Usuário está na 1FN, todos atributos monovalorados e atômicos, na 2FN pois todos os atributos dependem da chave. Também está normalizado na 3FN, todos os atributos dependem diretamente da chave primária. Temos que outra chave candidata é Email, pois cada email é único. Definimos a dependência funcional de que todos os atributos não chave dependem do email também, portanto segue a BCNF.

---

Código  $\rightarrow$  Nome, NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe

NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe  $\rightarrow$  Código, Nome

**Departamento** = { Código, Nome, NomeProfChefe, SobrenomeProfChefe, TelefoneProfChefe }

Está na 1FN pois não há atributos multivalores e todos são atômico. Na 2FN e na 3FN, todos os não-chaves (Nome e dados do chefe) dependem diretamente de Código, sem dependências parciais ou transitivas. E, segundo BCNF, pois os atributos não-chave dependem também da chave candidata, que é NomeProfChefe, SobrenomeprofChefe e TelefoneProfChefe, logo Departamento também está em BCNF.

---

Sigla  $\rightarrow$  CapacidadeTurma, MaterialBasico, NroAulasSemanais, CidadeUnidade, EstadoUnidade, PaísUnidade

**Disciplina** = { Sigla, CapacidadeTurma, MaterialBasico, NroAulasSemanais, CidadeUnidade, EstadoUnidade, PaísUnidade }

A tabela está na 1FN, 2FN e 3FN porque Sigla é única chave simples e os demais atributos dependem diretamente dela. Como só existe uma chave candidata, Disciplina está em BCNF.

---

Codigo  $\rightarrow$  Nome, NivelEnsino, Ementa, CargaHoraria, NumeroVagas, NroSala, CidadeSala, EstadoSala, PaisSala, CodigoDepartamento, CidadeUnidade, EstadoUnidade, PaísUnidade

**Curso** = { Codigo, Nome, NivelEnsino, Ementa, CargaHoraria, NumeroVagas, NroSala, CidadeSala, EstadoSala, PaisSala, CodigoDepartamento, CidadeUnidade, EstadoUnidade, PaisUnidade }

A tabela Curso está na 1FN pois todos atributos são monovalorados e atômicos, na 2FN e 3FN pois atributos não-chave dependem diretamente da chave inteira, e a única chave candidata já é a chave primária, então também está normalizado na forma BCNF.

---

Timestamp, GrupoId, UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone → Texto

**Mensagem** = { TimestampData, TimestampHorário, GrupoId, UsuarioEmissorNome, UsuarioEmissorSobrenome, UsuarioEmissorTelefone, Texto }

Não estava na 1FN, pois o atributo Timestamp não era atômico. O domínio de timestamp é 'YYYY-MM-DD HH:MI:SS', porém temos outros atributos na base em que esse domínio ocorre em partes separadas, por exemplo DataNasc, DataMaxMatricula e DataMatricula, com domínio sendo 'YYYY-MM-DD'. Para normalizar, quebramos o atributo não atômico em seus componentes. Escolhemos quebrar em TimestampData, Dom(TimestampData): YYYY-MM-DD e TimestampHorário, Dom(TimestampHorário): HH:MI:SS pois são as partes usadas em nossa base, dessa forma normalizamos para 1FN.

Para as formas normais 2FN, na 3FN e BCNF a relação Mensagem está normalizada, pois o único atributo não-chave depende diretamente da chave inteira, e a única chave candidata é a chave primária.

---

PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf → DataMaxMatricula, NroSala, CidadeSala, EstadoSala, PaisSala

**Oferecimento** = { PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMaxMatricula, NroSala, CidadeSala, EstadoSala, PaisSala }

Está na 1FN pois todos os atributos são monovalorados e atômicos, na 2FN e 3FN pois todos os atributos não-chave dependem da chave inteira sem transitividade, e na BCNF pois há apenas uma chave candidata.

---

NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf → Texto, NotaDidatica, NotaMaterial, NotaConteudo, NotaInfra

**AvaliarOferecimento** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeriodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, Texto, NotaDidatica, NotaMaterial, NotaConteudo, NotaInfra }

A tabela AvaliarOferecimento está na 1FN pois todos atributos são monovalorados e atômicos, na 2FN e 3FN pois atributos não-chave dependem diretamente da chave inteira, e a única chave candidata já é a chave primária, então também está normalizado na forma BCNF.

---

NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf → DataMatricula, Status, Taxa

**MatricularOferecimento** = { NomeAluno, SobrenomeAluno, TelefoneAluno, PeríodoLetivo, SiglaDisciplina, NomeProf, SobrenomeProf, TelefoneProf, DataMatricula, Status, Taxa }

Está na 1FN pois os atributos são atômicos e monovalorados. Também está na 2FN porque DataMatricula, Status e Taxa, os atributos não-chave, depende do conjunto completo da chave composta (não de parte dela), e na 3FN porque não há nenhuma dependência indireta entre atributos não-chave. Está em BCNF porque pois como só existe uma chave candidata reduz-se a 3FN.

---

Todas as demais relações do esquema (DisciplinaPreRequisito, CursoPreRequisito, ComposicaoCurso, DisciplinasEResponsaveis, Grupo, RegrasCurso, InfraestruturaCurso, NotasMatricula, BolsasMatricula e DescontosMatricula) são formadas exclusivamente pelos atributos que compõem suas chaves primárias e não possuem nenhum atributo adicional, o que significa que essas relações literal e formalmente garantem a normalização máxima, pois não carregam nenhum atributo extra que pudesse gerar anomalias ou redundância.

---

#### **Exercício 6:** Explicação das consultas contidas nos scripts de SQL

**Consulta 1 – Distribuição de Matrículas por Status:** É uma consulta que resume a quantidade de alunos por disciplina e status de matrícula, ela é útil para visualizar ocupação de turmas e analisar por exemplo índice de evasão, quantidade de matrículas por turma, etc.

**Consulta 2 – Informações de Alunos com Média Geral e Taxas Pagas:** Essa consulta apresenta os alunos, junto com suas médias e o valor total de taxas que pagaram. Ela é útil para analisar o desempenho acadêmico e o investimento financeiro na educação do aluno.

**Consulta 3 – Ocupação Média das Turmas por Período Letivo:** Essa consulta calcula a taxa média de ocupação das disciplinas ofertadas por período letivo, considerando apenas matrículas confirmadas e concluídas.

**Consulta 4 – Professores com Maior Evasão Relativa:** Apresenta os 15 casos com maior taxa de evasão (cancelamento, reprovação, trancamento) por professor e disciplina. Essa consulta é útil para entender a relação entre

**Consulta 5 – Alunos “mais faladores” versus desempenho:** Relaciona o número de mensagens enviadas por aluno com sua média de notas. A ideia é que a gente consiga medir a relação entre “engajamento” de um aluno com o sistema e suas notas.



**Consulta 6 – Professores com Maior Receita, Evasão e Nota Média:** Consulta os 30 professores com maior receita (receita é a soma de todas as taxas que as matrículas em oferecimentos deste professor), acompanhada de suas taxas de evasão (status de matrícula = 'cancelada','reprovada','trancada') e média de notas dos alunos.

**Consulta 7 – Departamentos com Disciplinas Vinculadas:** Esta consulta reúne todos os departamentos cadastrados na universidade, juntamente com o nome do professor chefe de cada um e as disciplinas ministradas dentro dos cursos que pertencem ao respectivo departamento.

**Exercício 7:** Escolha e crie pelo menos 3 índices em sua base de dados (de diferentes tipos):

```
CREATE INDEX idx_medias_notas_aluno
ON NotasMatricula (
    NomeAluno,
    SobrenomeAluno,
    TelefoneAluno,
    Nota
);
```

```
CREATE INDEX idx_matricula_professor
ON MatricularOferecimento (
    PeríodoLetivo,
    NomeProf,
    SobrenomeProf,
    TelefoneProf,
    Status
);
```

```
CREATE INDEX idx_mensagem_emissor
ON Mensagem (
    GrupoId,
    UsuarioEmissorNome,
    UsuarioEmissorSobrenome,
    UsuarioEmissorTelefone
);
```

Todos índices foram criados sobre queries buscando otimizar a consultas por colunas que não são chaves nas respectivas tabelas, otimizando join e where.

**Exercício 8:** Crie pelo menos 3 visões em sua base de dados. Apresente as consultas utilizadas, que obrigatoriamente devem possuir junções entre (pelo menos) duas outras relações:

**Visão 1: vw\_aluno\_financeiro:**

Para cada usuário que seja do tipo Aluno (CEE Aluno), a view calcula a média das notas (relação NotasMatricula) e o somatório das taxas (relação MatricularOferecimento) que foram cobradas em suas matrículas. Esses dados são, então, agrupados pelo Nome e Sobrenome concatenados, além de ordenar pelos alunos de maior média para o de menor.

**Visão 2: vw\_ofertas\_abertas:**

Para cada registro da tabela Oferecimento (CEA Oferecimento) cuja coluna DataMaxMatricula seja maior ou igual a CURRENT\_DATE, a view conta, a partir da relação MatricularOferecimento, quantos alunos têm Status em ('confirmada','pendente') para gerar alunos\_inscritos, obtém CapacidadeTurma da relação Disciplina, calcula vagas\_disponiveis como CapacidadeTurma – alunos\_inscritos, concatena NomeProf com SobrenomeProf no alias professor e NroSala com CidadeSala no alias sala; as ofertas sem inscrições aparecem com alunos\_inscritos = 0, e o resultado é ordenado por PeriodoLetivo e SiglaDisciplina.

**Visão 3: vw\_vagas\_semestre:**

Para cada combinação de PeriodoLetivo e SiglaDisciplina da relação Oferecimento (CEA Oferecimento), a view soma o atributo CapacidadeTurma da relação Disciplina para gerar vagas\_total, conta na relação MatricularOferecimento todos os registros cujo Status seja diferente de 'cancelada' para obter inscritos, calcula vagas\_disponiveis como vagas\_total – inscritos e determina ocupacao\_pct como  $100.0 * \text{inscritos} / \text{vagas\_total}$ , arredondado para uma casa decimal.

**Exercício 10:** Descreva, de forma sucinta e coesa, quais foram as principais dificuldades encontradas durante a construção deste trabalho e quais observações ou aprendizados você julga importantes ao finalizar o projeto.

De modo geral, como grupo, tivemos bastante dificuldade com os aspectos interpretativos do trabalho. Naturalmente, essa é uma característica bastante marcante em qualquer modelagem feita em domínios diversos de conhecimento, porém em Bases de Dados o nível de abstração é muito maior do que em outras disciplinas que encontramos anteriormente no curso. A partir da descrição textual do problema do trabalho, mais de uma vez, tivemos discordâncias de opiniões sobre como algo deveria ser interpretado e isso, de certa forma, acaba nos aproximando bastante de situações reais que muitas vezes são ambíguas e não possuem um “certo” ou “errado” bem definidos, trazendo a tona a necessidade constante de tomar decisões que sejam intencionais (virtualmente tudo pode estar correto contanto que seja devidamente justificado). Um exemplo palpável que serve de anedota para essa problemática no contexto do nosso grupo foram diversas discordâncias entre utilizar multiplicidade (0,N) ou (1,N) em situações em que não conseguimos concordar se uma determinada entidade é obrigada a aderir a um determinado relacionamento ou não.

Uma outra questão com a qual tivemos bastante dificuldade foi na apresentabilidade do nosso diagrama no draw.io. Conforme fomos modelando o nosso sistema iterativamente, a complexidade foi escalando, e junto disso, o diagrama foi pouco a pouco se tornando mais difícil de ser lido (um membro do grupo afirmou que começou a parecer um “ninho de rato”).

Junto da complexidade do diagrama veio a sobrecarga semântica da nossa modelagem, quanto mais tentávamos aderir aos requisitos textuais da descrição do problema mais a complexidade da modelagem aumentava e mais “pesado” o sistema ficava, fazendo com que precisássemos cortar um pouco o uso de abstrações, em especial as de agregação.

O pensamos que teve bastante peso como aprendizado é a ênfase em justificar as decisões de modelagem que tomamos e que, de fato, as escolhas precisam ser intencionais, não sobrando muito espaço para um trabalho “mecânico”.

---