

# Documentação do Projeto Final - SCC0240 - Base de Dados

## Grupo 10

Alunos:

Breno Gonçalves Rodrigues - 11734142

Laura Neri Thomaz da Silva - 13673221

Mateus Bernal Leffek - 13673318

Nicolas Carreiro Rodrigues - 14600801

## Descrição Dos Conjuntos de Entidades e Relacionamentos(ex1)

### Conjunto de Entidades

A construção do modelo de dados começa com a definição da entidade **Usuário**, que possui o **cpf** como chave primária. Além disso, a entidade contém os atributos **nome**, **sobrenome** e **telefone**, sendo estes três últimos utilizados como chave secundária. Atributos adicionais incluem **endereço**, **data de nascimento**, **sexo**, **email** e **senha**, conforme especificado no enunciado. A entidade **Usuário** é ainda uma generalização, abrangendo três subtipos:

- **Professor**: possui os atributos **Titulação** e **Área de Especialidade**.
- **Aluno**: contém o atributo **Bolsa**, que indica se o aluno é bolsista. Optamos por incluir esse atributo aqui, embora o enunciado sugira associá-lo à matrícula de um aluno em uma disciplina. Essa escolha reflete um modelo mais alinhado a uma unidade escolar do que uma plataforma de cursos.
- **Funcionário**: não possui atributos adicionais destacados no problema. Embora o enunciado menciona uma possível entidade **Operações**, não foram fornecidos detalhes sobre seus atributos. Sabemos apenas que um usuário do tipo **Funcionário** gerencia essas operações, logo decidimos colocar **Operações** como atributo multivalorado de **Funcionário**.

A seguir, temos a entidade **Unidade Escolar**, que representa a localização da escola. Ela inclui os atributos **Nome da Cidade**, **Sigla do Estado**, **Sigla do País** e **Número**

**do prédio.** Embora não seja especificada uma chave primária no enunciado, definimos o atributo **ID Escola** como chave primária para identificar unicamente a entidade.

A entidade **Curso** é definida por **Sigla** (como chave primária), **Nome**, **Classificação**, **Carga Horária** e **Número de Vagas**, refletindo as características de um curso dentro da instituição. A **Disciplina** é identificada pelo atributo **Código**, que funciona como chave primária, e possui ainda os atributos **Nome**, **Quantidade de Aulas Semanais** e **Material Didático Básico**, sendo este último um atributo composto, destinado a indicar os livros ou materiais utilizados como referência na disciplina.

A entidade **Departamento**, representando a divisão interna da instituição, possui **Sigla** como chave primária e **Nome** para identificá-lo. Outra entidade importante é a **Mensagem**, que possui o atributo **ID Mensagem** como chave primária e o atributo **Texto Mensagem**, que contém o corpo da mensagem enviada.

Uma entidade **Aviso**, que possui o atributo **ID Aviso** como chave primária, e **Texto Aviso**. Por fim, a entidade **Regras** é uma **entidade fraca**, com os atributos **Código**, como chave primária, **Frequência Mínima**, **Critério de Aprovação** e **Infraestrutura**, sendo este último um atributo multivalorado, destinado a indicar as diversas necessidades de infraestrutura de um curso.

## Relacionamentos Entre as Entidades

A partir das entidades descritas, os seguintes relacionamentos são estabelecidos:

**Chefia** entre **Professor** e **Departamento**: Este relacionamento descreve os departamentos que um professor pode chefiar. A multiplicidade é de [0,1] entre **Professor** e **Departamento**, indicando que um professor pode ou não chefiar um departamento, e [1:1] entre **Departamento** e **Professor**, ou seja, cada departamento deve ter exatamente um chefe. Essa relação implica que cada departamento terá, obrigatoriamente, um chefe, enquanto o professor poderá chefiar no máximo um departamento.

**Pertence** entre **Curso** e **Departamento**: Este relacionamento descreve quais cursos pertencem a quais departamentos. A multiplicidade é [1,M] entre **Departamento** e **Curso**, significando que um departamento pode ter vários cursos, mas cada curso deve pertencer a um único departamento. A multiplicidade [1,1] entre **Curso** e **Departamento** indica que cada curso está vinculado a exatamente um departamento.

**Ministra** entre **Professor** e **Disciplina**: Este relacionamento indica quais disciplinas são ministradas por quais professores. A multiplicidade [1,M] se aplica a ambos os lados, ou seja, um professor pode ministrar várias disciplinas e cada disciplina deve ter ao menos um professor responsável. Esse relacionamento também é uma agregação chamada **Oferta Disciplina**, que descreve quais disciplinas estão sendo ofertadas em determinado horário e sala. A chave primária dessa agregação é composta por **Dia**, **Hora** e **Sala**, de modo a evitar conflitos de horário e local.

**Auto Relacionamento de Pré-Requisito e Curso**: Indica quais **Cursos** são pré-requisitos para outros. A multiplicidade [0,M] é aplicada em ambos os lados,

indicando que uma disciplina pode ter nenhum ou muitos pré-requisitos. Esse relacionamento **Pré Requisito** também se aplica entre **Curso** e **Disciplina**, indicando quais disciplinas podem ser pré-requisitos para os demais cursos, com multiplicidade [0,M].

**Compor** entre **Disciplina** e **Curso**: Este relacionamento indica quais disciplinas fazem parte de um curso. A multiplicidade [1,M] é aplicada a ambos os lados, ou seja, uma disciplina pode compor um ou vários cursos, e cada curso deve ter ao menos uma disciplina.

**Comunica** entre **Aluno**, **Professor** e **Mensagem**: Este é um relacionamento ternário, que indica a troca de mensagens entre alunos e professores. O relacionamento possui o atributo **Data de Envio** para registrar o momento da comunicação. A cardinalidade é [1:M:N], significando que um professor pode enviar várias mensagens para um aluno ou grupo de alunos.

**Notifica** entre **Funcionário**, **Usuário** e **Avisos**. Este relacionamento indica a troca de avisos entre a **Administração**, que se encaixa no Perfil de Funcionário, e todos os usuários do sistema de forma geral. A cardinalidade é [1:M:N] indicando que a administração pode enviar vários avisos para um ou um grupo de usuários. Esta relação possui um atributo **Data Envio** para indicar a data que o aviso foi enviado.

**Estuda** entre **Aluno** e **Unidade Escolar**: Este relacionamento indica em qual unidade escolar o aluno está matriculado. A multiplicidade é [1,1] entre **Aluno** e **Unidade Escolar**, indicando que cada aluno deve estar matriculado em uma única unidade escolar. Já a multiplicidade [1,M] entre **Unidade Escolar** e **Aluno** indica que uma unidade escolar deve ter, no mínimo, um aluno, mas pode ter vários.

**Lesiona** entre **Professor** e **Unidade Escolar**: Este relacionamento descreve a qual unidade escolar o professor está associado. A multiplicidade [1,1] entre **Professor** e **Unidade Escolar** indica que cada professor pertence a uma única unidade escolar, enquanto a multiplicidade [1,M] entre **Unidade Escolar** e **Professor** indica que uma unidade escolar pode ter vários professores.

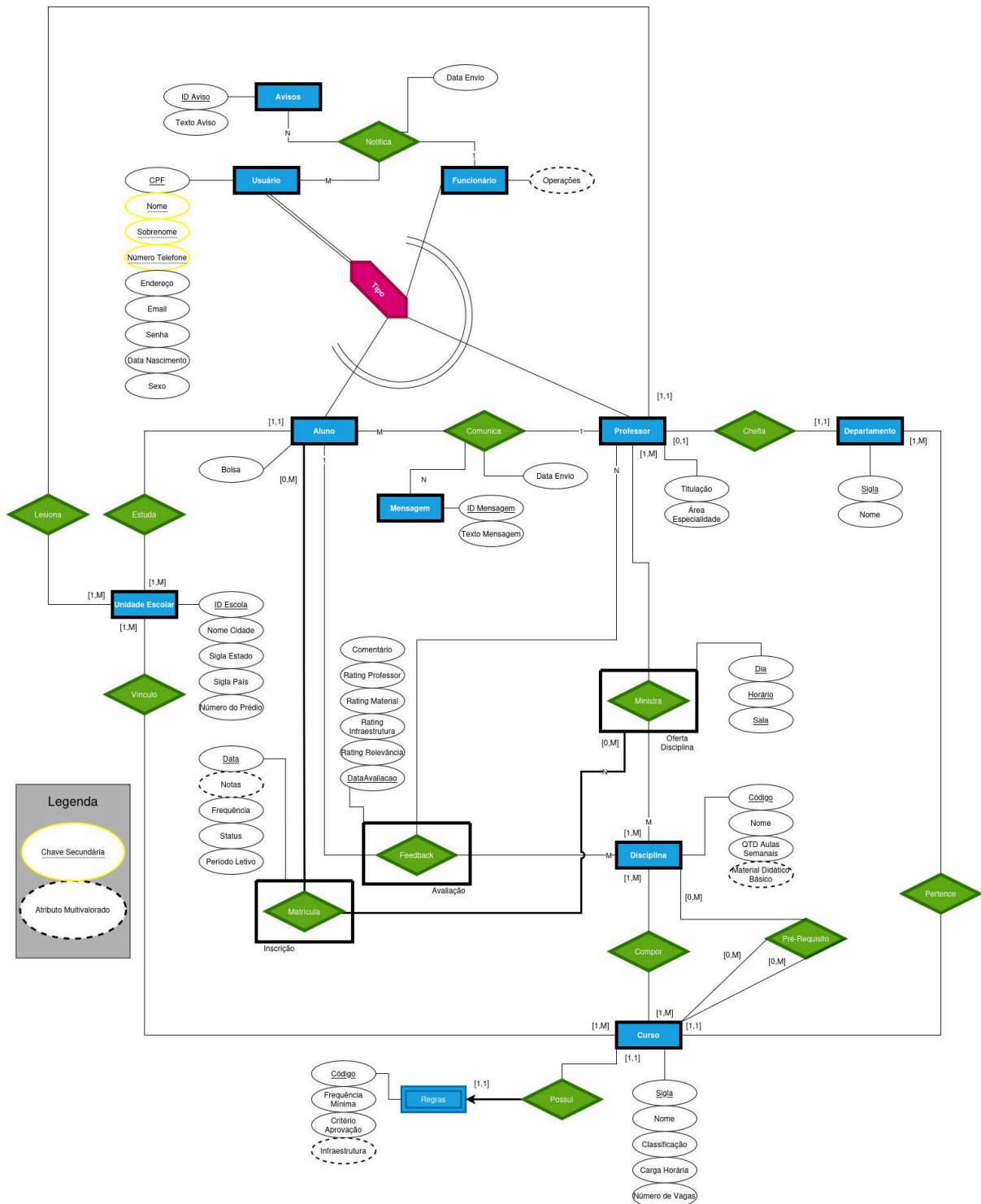
**Vínculo** entre **Curso** e **Unidade Escolar**: Este relacionamento indica quais cursos são oferecidos por cada unidade escolar. A multiplicidade [1,M] em ambos os lados significa que um curso pode ser oferecido por várias unidades escolares e cada unidade escolar oferece, pelo menos, um curso.

**Matrícula** entre **Aluno** e **Oferta Disciplinas**: Este relacionamento descreve a matrícula dos alunos nas disciplinas ofertadas. A multiplicidade [0,M] é aplicada a ambos os lados, significando que um aluno pode se matricular em nenhuma ou várias disciplinas, e uma disciplina pode ter nenhum ou vários alunos matriculados. Essa relação se torna uma agregação chamada **INSCRIÇÃO**, e possui a seguinte chave: **Data** (para garantir que um aluno possa se matricular em uma mesma disciplina em períodos diferentes), **Notas** (como atributo multivalorado), **Frequência**, **Status** e **Período Letivo**. Relacionamento **Feedback** entre **Aluno**, **Professor** e **Disciplina**: Este é um relacionamento ternário com cardinalidade [1:M;N] que permite a avaliação das disciplinas e dos professores por um aluno. Esse relacionamento se torna uma agregação chamada **Avaliação** que tem os seguintes atributos: **Comentário**, **Rating Professor**, **Rating Material**, **Rating Infraestrutura** e **Rating Relevância**. A agregação **Avaliação** possui o atributo **Data**, permitindo que uma mesma avaliação seja feita várias vezes para um mesmo professor e disciplina.

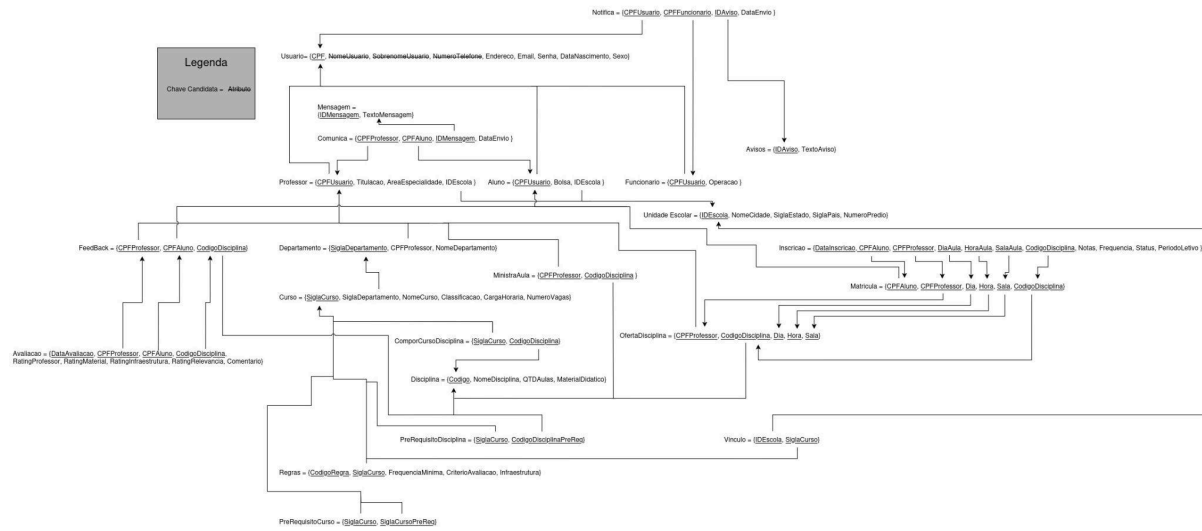
**Possui** entre **Curso** e **Regras**: Este relacionamento define que cada curso

está associado a uma única regra, e cada regra está vinculada a um único curso. A multiplicidade  $[1,1]$  de ambos os lados implica que um curso só pode ter uma regra, e cada regra deve ser associada a um único curso.

## Modelagem Entidade Relacionamento(ex2)



# Modelo Relacional(ex3)



## Justificativa das Formas Normais(ex5)

Nessa questão, vamos analisar as formas normais em que cada tabela do esquema relacional do Exercício 3 que é nosso inicial, assim, justificando suas classificações iniciais com base nas regras da 1FN, 2FN, 3FN, 3FN e BCNF. Após isso, vamos aplicar as transformações necessárias para que tenhamos todas as relações na Forma Normal de Boyce-Codd.

Tabela	Forma Normal Inicial	Justificativa
Unidade Escolas	BCNF	Atributos atômicos e dependem unicamente da chave primária IDEscola
Disciplinas	1FN	“MaterialDidatico” é multivalorado
Departamento	2FN/3FN	NomeDepartamento depende apenas da SiglaDepartamento, não da chave completa
Avisos	BCNF	Tabela simples e sem dependências transitivas ou parciais
Usuário	BCNF	Todos os atributos dependem da chave primária “CPF”
Professor	BCNF	Sem dependências transitivas ou parciais
Funcionario	1FN	“Operacao” é multivalorado
Curso	BCNF	Sem qualquer tipo de dependência
Regras	1FN	“Infraestrutura” é multivalorado
CompCorCursoDisciplina	BCNF	Chave composta e não tem atributos adicionais
PreRequisitoCurso	BCNF	Estrutura de relacionamento praticamente, sem atributos adicionais
PreRequisitoDisciplina	BCNF	Estrutura de relacionamento praticamente, sem atributos adicionais
MinistraAula	BCNF	Várias chaves estrangeiras, sem novos atributos

Aluno	BCNF	Todos os atributos dependem apenas da chave primária "CPFUsuario"
Notifica	BCNF	Todos atributos dependem da chave composta
Mensagem	BCNF	Sem qualquer tipo de dependência
Comunica	BCNF	Todos atributos dependem da chave composta
OfertaDisciplina	BCNF	Todos atributos são parte da chave
Vinculo	BCNF	Relacionamento entre escola e curso
Matricula	BCNF	Todos atributos dependem da chave composta
Inscricao	1FN	"Notas" é multivalorado
FeedBack	BCNF	Chave composta sem dependência parcial ou transitiva
Avaliacao	BCNF	Todos atributos dependem da chave composta

## Normalização até a BCNF

### Disciplina:

- Problema: Atributo multivalorado "MaterialDidatico"
- Solução: Criar uma tabela MaterialDidatico (CodigoDisciplina, Material)

### Departamento:

- Problema: Atributo "NomeDepartamento" depende apenas de "SiglaDepartamento"
- Solução: Separar em duas tabelas, uma sendo: Departamento(SiglaDepartamento, NomeDepartamento); outra sendo: ChefiaDepartamento(SiglaDepartamento, CPFProfessor)

### Funcionario:

- Problema: Atributo multivalorado "Operacao"
- Solução: Criar uma tabela OperacaoFuncionario(CPFUsuario, Operacao).

### Regras:

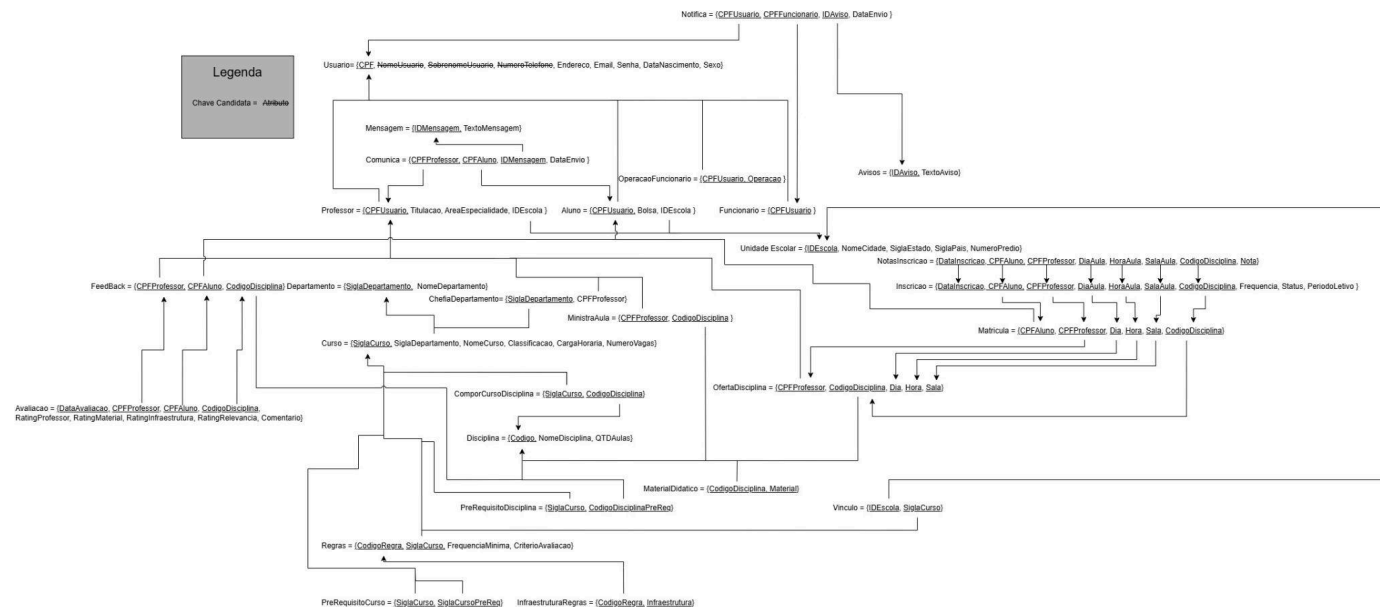
- Problema: Atributo multivalorado "Infraestrutura"
- Solução: Criar uma tabela InfraestruturaRegra(CodigoRegra, Recurso)



## Inscricao:

- Problema: Atributo multivalorado “Notas”
- Solução: Criar uma tabela NotasInscricao(DataInscricao, CPFAluno, CPFProfessor, DiaAula, HoraAula, SalaAula, CodigoDisciplina, Nota).

Com isso em mente, temos o novo esquema relacional com todos na BCNF:



## Justificativa das Escolhas dos Índices(ex7)

### Index 1: B-Tree

Aplicado em: Disciplina(NomeDisciplina)

O índice B-Tree é ideal para colunas com **valores distintos e buscas por igualdade ou prefixos**. A busca por nome de disciplina é comum em sistemas educacionais. Geralmente sem índices, a busca é feita de forma sequencial, com muitos registros, a busca por prefixo melhora significativamente com o índice B-Tree

Resultado Antes do index: 0.162 ms

Resultado Depois do Index: 0.065 ms

### Index 2: Hash

Aplicado em: Aluno(CPFUsuario)

Ideal para **buscas diretas por igualdade**, especialmente em colunas com chaves primárias ou foreign keys. Aqui, vamos usar explicitamente o tipo HASH. Hash indexes não são úteis para ordenações ou ranges, mas otimizam bem a igualdade pura.

Resultado Antes do index: 0.042 ms

Resultado Depois do Index: 0.030 ms

## Index 3: Composto

Aplicado em: OfertaDisciplina(Dia, Hora)

Consulta por dia e hora é frequente em agendas acadêmicas. Um índice composto otimiza **consultas com múltiplas colunas** (especialmente na ordem do índice).

Com esse índice, o banco pode fazer **Bitmap Index Scan** eficiente, em vez de verificar todas as tuplas.

Resultado Antes do index: 0.039 ms

Resultado Depois do Index: 0.029 ms

## Dificuldades e Desafios(ex10)

Durante a criação do diagrama de entidade-relacionamento, encontramos algumas dificuldades em implementar perfeitamente certos aspectos mencionados no enunciado. As três relações — *Comunica*, *Notifica* e *Avaliação* — podem não estar totalmente alinhadas com as especificações exigidas, mas, conforme as justificativas apresentadas no exercício 1, acreditamos que o diagrama é funcional. Ainda se tratando desse caso entre Comunicação de mensagens entre Professores e Alunos, não conseguimos modelar uma “Turma”, para melhor indicar a troca de mensagens entre o mesmo.

Além disso, a falta de experiência em modelagem contribuiu para que o diagrama acabasse ficando um pouco confuso no que diz respeito às conexões entre as entidades. Não conseguimos organizar as relações de forma a garantir uma visualização mais agradável e clara.

Outro desafio que enfrentamos foi a construção do modelo relacional da auto-relação de pré-requisito entre as entidades Curso e Disciplina. Ainda se tratando da modelagem relacional, pelo problema possuir muitas relações, o diagrama relacional ficou uma “bagunça”, quando inserido na documentação, a imagem fica pequena, de forma que não facilite a visualização.

## POSSÍVEIS PROBLEMAS ENCONTRADOS

Com nosso modelo, é possível qualquer aluno fazer uma avaliação de um determinado professor e disciplina, mesmo ele não ter sido inscrito nessa disciplina com esse professor. Outro Problema que podemos citar é que é possível um usuário ser mais de um tipo, ou seja, um mesmo cpf ser aluno, professor e funcionário. Além disso, também é possível um professor oferecer uma disciplina x em uma data, hora e sala, e outra y em uma mesma data, hora e sala que a disciplina x. De acordo com algumas pesquisas, é possível solucionar esses dois últimos problemas criando algumas visões, mas até o momento atual do projeto apenas implementamos visões relacionadas à parte de busca.

Se tratando da parte de índice, observamos que os resultados feitos pelas buscas ainda aparentemente se demonstram por sequential scan, mesmo após a criação de

índices o sgbd opta por escolher sequential scan para algumas buscas, tentamos um código que fornecesse a inserção de dados bem aleatórios para popular as tabelas que efetuamos as buscas, mas mesmo assim na resposta do sgbd ele ainda fica para sequential scan, então apenas observamos os tempos de execução das buscas.

## Links

GitHub:

[https://github.com/rodriguesgbreno/projeto\\_final\\_base\\_de\\_dados\\_scc0240\\_2025](https://github.com/rodriguesgbreno/projeto_final_base_de_dados_scc0240_2025)

Youtube:

<https://www.youtube.com/watch?v=FHgnehBufgg0>