

# IGESTA



## Modelagem de Dados - Banco PostgreSQL



## INTEGRANTES



Fellipe Meira Brandão



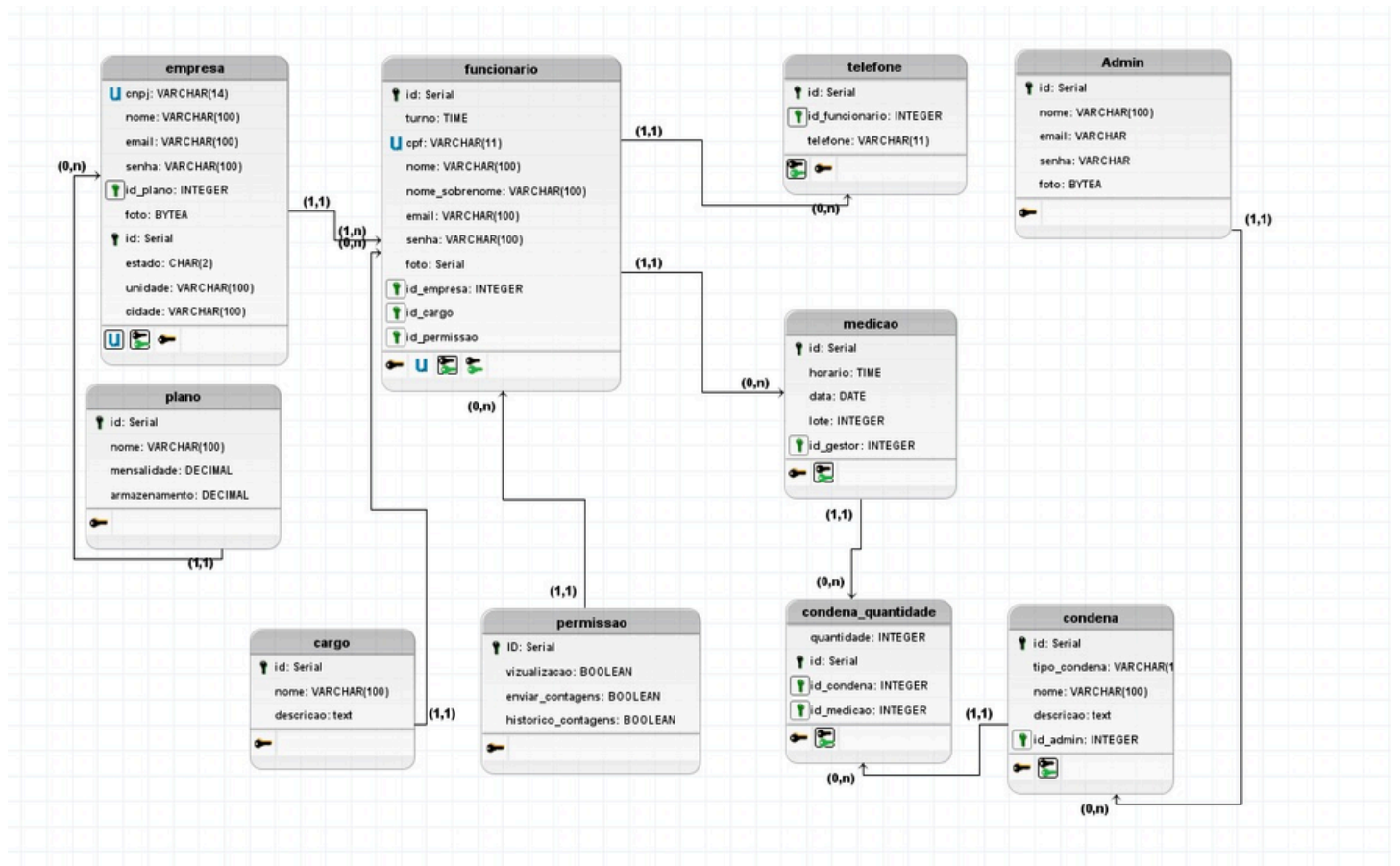
Gabriel Martins



Rafael Cruz

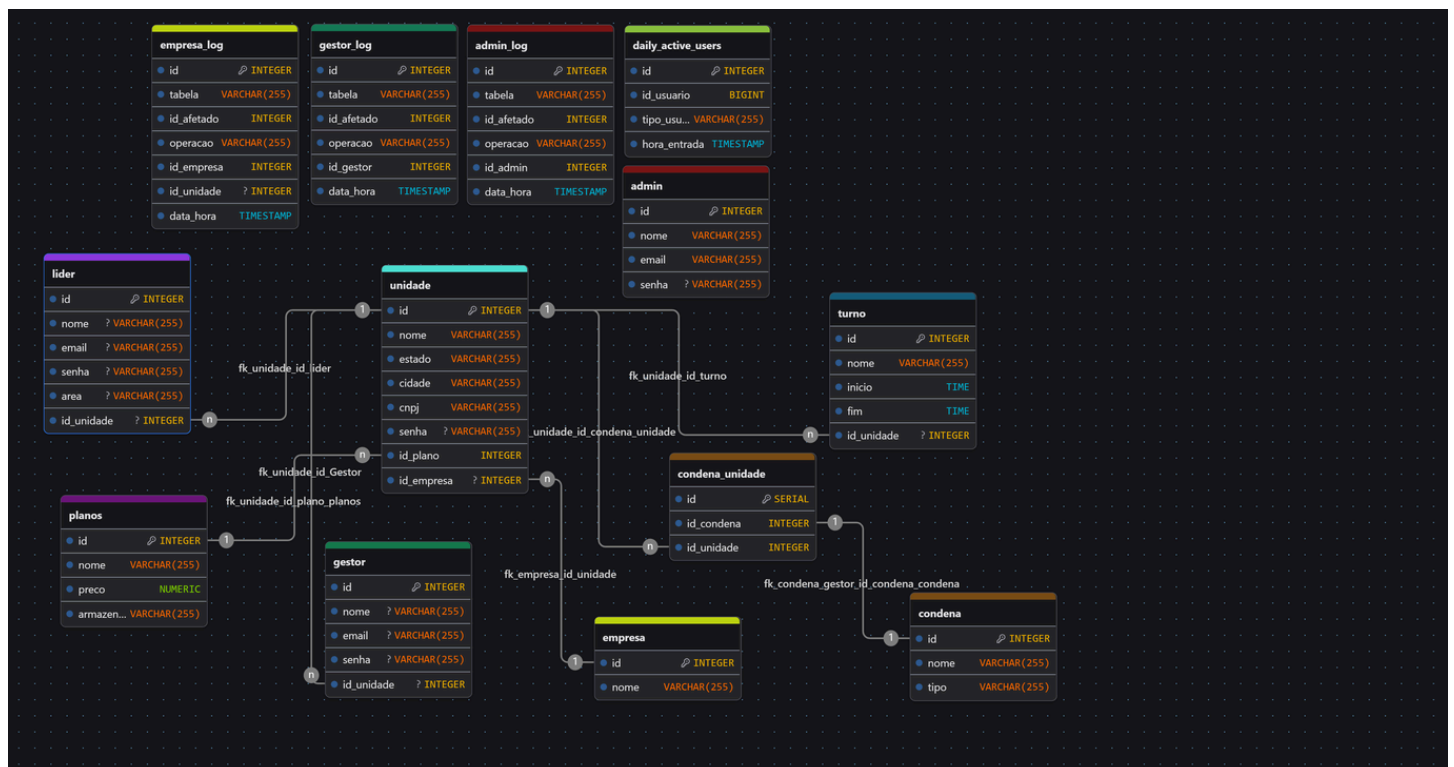
# Modelagem do Banco de Dados SQL - Sem normalização

Nosso modelo não normalizado contém 10 tabelas: Empresa, Funcionário, Plano, Cargo, Permissão, Medição, condena\_quantidade, Condena, Telefone e Admin



# Modelagem do Banco de Dados SQL - Normalizado

No modelo normalizado transformamos as tabelas Funcionário, Cargo e Permissão nas tabelas Gestor e Líder, para separar as funções e permissões dentro do app, além disso adaptamos a tabela Medição e condena\_quantidade para o MongoDB. Também dividimos a tabela Empresa em Unidade e Empresa, visto que os planos variam de unidade para unidade. Também criamos a uma tabela intermediária entre Condena e Unidade, que armazena as condenas utilizadas nas contagens, que varia por unidade.



## Tabelas do modelo normalizado e descrição

### planos

- id - ID do plano de armazenamento
- nome - Nome do plano de armazenamento
- preco - Preço do plano
- armazenamento - Espaço de armazenamento do plano (Gigabytes)

## líder

- id - ID do líder
- nome - Nome inteiro do líder
- email - Email associado ao líder
- senha - Senha da conta do líder
- area - Área que o líder presta serviço (Halal | Pendura | Sangria | Escaldagem | Evisceração)
- id\_unidade - ID da unidade que o líder trabalha

## turno

- id - ID do turno
- nome - Nome para referenciar o horário do turno (Manhã | Tarde | Noite)
- inicio - Horário do início do turno (00:00:00)
- fim - Horário de término do turno (00:00:00)
- id\_unidade - ID da unidade que irá obedecer tais turnos (Pode ser que os turnos de uma empresa seja diferente da outra)

## unidade

- id - ID da unidade
- nome - Nome da unidade de tal empresa
- estado - Estado que essa unidade pertence
- cidade - Cidade que essa unidade pertence
- cnpj - CNPJ correspondente a unidade
- senha - Senha de acesso para a conta da Unidade (CNPJ e senha)
- id\_plano - ID do plano que a unidade está usando
- id\_empresa - ID da empresa que essa unidade faz parte

## empresa

- id - ID da empresa
- nome - Nome da empresa

## gestor

- id - ID do gestor
- nome - Nome do gestor
- email - email do gestor
- senha - senha da conta do gestor
- id\_unidade - ID da unidade que o gestor trabalha

## condena

- id - ID da condena
- nome - Nome da condena
- tipo - Tipo da condena se é **parcial** ou **total**

## condena\_unidade

- id\_condena - ID da condena que o gestor vai aderir
- id\_unidade - ID da unidade que corresponde as condenas escolhidas para uso

## admin

- id - ID do administrador do App
- nome - Nome do administrador do App
- email - email do administrador do App

# Tabelas de Log e DAU

## empresa\_log

- id - ID do log
- tabela - Tabela que sofreu alteração da conta de empresa
- id\_afetado - ID que foi modificado de alguma maneira
- operacao - Consta se foi um INSERT, UPDATE ou DELETE
- id\_empresa - ID da empresa que fez a operação
- id\_unidade - ID da unidade que fez a operação
- data\_hora - Data e horário que foi executado a ação da empresa

## gestor\_log

- id - ID do log
- tabela - Tabela que sofreu alteração da conta de gestor
- id\_afetado - ID que foi modificado de alguma maneira
- operacao - Consta se foi um INSERT, UPDATE ou DELETE
- id\_gestor - ID do gestor que fez a operação
- data\_hora - Data e horário que foi executado a ação do gestor

## admin\_log

- id - ID do log
- tabela - Tabela que sofreu alteração de uma conta do admin
- id\_afetado - ID que foi modificado de alguma maneira

- operacao - Consta se foi um INSERT, UPDATE ou DELETE
- id\_admin - ID do admin que fez a operação
- data\_hora - Data e horário que foi executado a ação do admin

## daily\_active\_users

- id - ID que representa a entrada de um usuário
- id\_usuario - ID do usuário que entrou no app.
- tipo\_usuario - Representa o tipo do usuário (Gestor | Líder | Unidade)
- hora\_entrada - Data e horário que o usuário entrou no app.

## Functions

### tipo\_usuario

Essa função identifica o tipo de usuário (**Gestor**, **Líder** ou **Unidade**) no momento do login no aplicativo, com base nas credenciais informadas: **e-mail** (para **gestores** e **líderes**) ou **CNPJ** (para **unidades**) e a senha.

### verificar\_empresa\_existente

Verifica se uma empresa já existe no banco de dados com base nos oito primeiros dígitos do CNPJ da unidade, que representam a raiz do CNPJ utilizada para identificar empresas e suas respectivas unidades. Facilita o reconhecimento do relacionamento quando uma nova unidade é inserida.

### log\_admin

Função utilizada por uma trigger para registrar na tabela “**admin\_log**” todos os dados inseridos ou alterados por administradores.

### log\_gestor

Função utilizada por uma trigger para registrar na tabela “**gestor\_log**” todos os dados inseridos ou alterados por gestores.

## log\_empresa

Função utilizada por uma trigger para registrar na tabela “**empresa\_log**” todos os dados inseridos ou alterados por empresas/unidades.

## Procedures

### inserir\_unidade\_por\_nome

Insere uma nova unidade no banco de dados e verifica se a empresa correspondente já existe, utilizando a função “**verificar\_empresa\_existente**” antes de realizar a inserção.

### atualizar\_condenas\_unidade

Atualiza a tabela “**condena\_unidade**” quando uma unidade passa a utilizar um novo tipo de condena ou deixa de utilizá-la, garantindo que as informações de vínculo entre unidades e tipos de condena permaneçam atualizadas no banco de dados.

### atualizar\_plano\_unidade

Atualiza o plano selecionado pela unidade, registrando a nova escolha no banco de dados e substituindo o plano anteriormente associado

## Triggers

### Admin

- **trg\_log\_admin** - chamada após Insert/Update/Delete feito por um admin na tabela “**admin**”.
- **trg\_log\_admin\_planos** - chamada após Insert/Update/Delete feito por um admin na tabela “**planos**”.
- **trg\_log\_admin\_condena** - chamada após Insert/Update/Delete feito por um admin na tabela “**condena**”.

### Gestor

- **trg\_log\_gestor** - chamada após Insert/Update/Delete feito por um gestor na tabela “**gestor**”.

- **trg\_log\_gestor\_lider** - chamada após Insert/Update/Delete feito por um gestor na tabela "lider".
- **trg\_log\_gestor\_condena** - chamada após Insert/Update/Delete feito por um admin na tabela "condena\_unidade".

## Empresa/Unidade

- **trg\_log\_empresa**- chamada após Insert/Update/Delete feito por uma unidade na tabela "empresa".
- **trg\_log\_empresa\_unidade** - chamada após Insert/Update/Delete feito por uma unidade na tabela "unidade".
- **trg\_log\_unidade\_turno** - chamada após Insert/Update/Delete feito por uma unidade na tabela "turno".

## Índices

MongoDB

### db.registros.createIndex({unidade:1})

O index acima cria um índice simples no campo "unidade" da coleção "registros".

O motivo para criar esse índice é melhorar o desempenho das consultas que filtram ou ordenam os documentos com base no campo "unidade", já que ele é muito utilizado nas APIs de consulta.

### db.registros.createIndex({unidade: 1, data:-1})

O índice composto acima cria um índice nos campos "unidade" (ordem crescente) e "data" (ordem decrescente) da coleção "registros".

O motivo para criar esse índice composto é otimizar consultas que filtram por "unidade" e ordenam os resultados por "data" em ordem decrescente. Esses campos são muito utilizados nas APIs que consultam os registros e nas consultas que o chatbot realiza para fornecer respostas rápidas e precisas aos usuários.