

Documento Oficial do Banco de Dados – Projeto A3

1 - Introdução

O presente documento descreve o **modelo de dados** desenvolvido para o projeto **A3**, elaborado como parte do trabalho acadêmico da disciplina de **SISTEMAS DISTRIBUÍDOS E MOBILE** da Faculdade São Judas Tadeu- Campus Mooca (USJT).

O objetivo do banco de dados é registrar informações sobre **bancos**, **contatos oficiais**, **tipos de golpes**, **denúncias de golpes** e **consultas realizadas**, garantindo integridade referencial, rastreabilidade e facilidade de consulta para análises futuras.

O banco de dados será implementado em **PostgreSQL**, utilizando **chaves primárias**, **estrangeiras** e **tipos adequados de dados**, com **IDs auto-incrementais** para identificação única de registros.

2 - Estrutura Geral do Banco de Dados

O banco de dados é composto pelas seguintes tabelas principais:

1. **bancos** – Registro de diferentes bancos disponíveis no sistema.
2. **contatos_oficiais** – Lista de contatos oficiais de cada banco.
3. **tipos_golpe** – Lista de tipos de golpes que podem ser denunciados.
4. **denuncias** – Registro das denúncias realizadas por usuários.
5. **consultas** – Log das consultas realizadas sobre contatos suspeitos ou oficiais.

A seguir, detalharemos cada tabela, seus campos, tipos e relacionamentos.

3 - Descrição das Tabelas

3.1 - Tabela `bancos`

Campo	Tipo de Dado	Restrição
id_banco	SERIAL	PRIMARY KEY
nome_banco	VARCHAR(100)	NOT NULL
cnpj	VARCHAR(20)	NULL
site_oficial	VARCHAR(200)	NULL
descricao	TEXT	NULL
data_cadastro	TIMESTAMP	DEFAULT NOW()

Observações:

*O ID é auto-incremental para permitir a inclusão de novos bancos sem conflito.
É a tabela central do sistema, referenciada por outras tabelas (`contatos_oficiais` e `denuncias`).*

3.2 - Tabela `contatos_oficiais`

Campo	Tipo de Dado	Restrição
id_contato	SERIAL	PRIMARY KEY
id_banco	INT	FOREIGN KEY → bancos(id_banco)
tipo_contato	VARCHAR(50)	NOT NULL
valor_contato	VARCHAR(150)	NOT NULL
observacao	TEXT	NULL
verificado	BOOLEAN	DEFAULT TRUE
data_validacao	TIMESTAMP	DEFAULT NOW()

Observações:

Mantém referência obrigatória à tabela `bancos`.

Permite rastrear contatos oficiais de cada banco, com validação de autenticidade.

3.3 - Tabela `tipos_golpe`

Campo	Tipo de Dado	Restrição
id_tipo	SERIAL	PRIMARY KEY
nome_tipo	VARCHAR(100)	NOT NULL
descricao	TEXT	NULL

Observações:

Utilizada para classificar e padronizar os golpes denunciados.

Referenciada pela tabela `denuncias`.

3.4 - Tabela `denuncias`

Campo	Tipo de Dado	Restrição
id_denuncia	SERIAL	PRIMARY KEY
id_tipo	INT	FOREIGN KEY → tipos_golpe(id_tipo)
id_banco	INT	FOREIGN KEY → bancos(id_banco)
contato_denunciado	VARCHAR(150)	NOT NULL
descricao	TEXT	NULL
data_denuncia	TIMESTAMP	DEFAULT NOW()

Observações:

Permite registrar todas as denúncias feitas por usuários.

Referência tanto `bancos` quanto `tipos_golpe` para garantir rastreabilidade.

3.5. Tabela `consultas`

Campo	Tipo de Dado	Restrição
id_consulta	SERIAL	PRIMARY KEY
termo_pesquisado	VARCHAR(150)	NOT NULL
resultado_encontrado	VARCHAR(50)	NULL
data_consulta	TIMESTAMP	DEFAULT NOW()

Observações:

*Tabela de log, registra todas as pesquisas realizadas pelo usuário.
Não possui referência direta a `bancos` ou `denuncias`, mas complementa o histórico do sistema.*

4. Descrição dos Campos

A seguir, descrevem-se os campos de cada tabela, detalhando seu propósito e uso dentro do sistema:

4.1 - Tabela bancos

id_banco: Identificador único do banco. É auto-incremental e garante que cada banco tenha um ID exclusivo.

nome_banco: Nome oficial do banco. Campo obrigatório para identificar o banco.

cnpj: Número do CNPJ do banco, utilizado para autenticação formal.

site_oficial: Endereço do site oficial do banco, usado para referência e validação de informações.

descricao: Informações adicionais sobre o banco, como histórico, observações ou características relevantes.

data_cadastro: Data e hora em que o banco foi registrado no sistema, usada para controle temporal.

4.2 - Tabela contatos_oficiais

id_contato: Identificador único do contato oficial, auto-incremental.

id_banco: Referência ao banco ao qual o contato pertence, garantindo integridade entre tabelas.

tipo_contato: Define se o contato é telefone, email, site ou outro tipo.

valor_contato: Valor do contato em si (número de telefone, endereço de email, URL).

observacao: Campo livre para anotações adicionais sobre o contato.

verificado: Indica se o contato foi confirmado como oficial (TRUE) ou não (FALSE).

data_validacao: Data em que a verificação do contato foi realizada.

4.3 - Tabela tipos_golpe

id_tipo: Identificador único do tipo de golpe.

nome_tipo: Nome do golpe, usado para categorização.

descricao: Detalhes explicativos sobre o golpe, como modus operandi ou alerta ao usuário.

4.4 -Tabela denuncias

id_denuncia: Identificador único da denúncia.

id_tipo: Referência ao tipo de golpe denunciado.

id_banco: Banco alvo da denúncia.

contato_denunciado: Número, email ou site do contato denunciado.

descricao: Detalhes da denúncia, relatando como o golpe ocorreu.

data_denuncia: Data e hora em que a denúncia foi registrada.

4.5 - Tabela consultas

id_consulta: Identificador único da consulta.

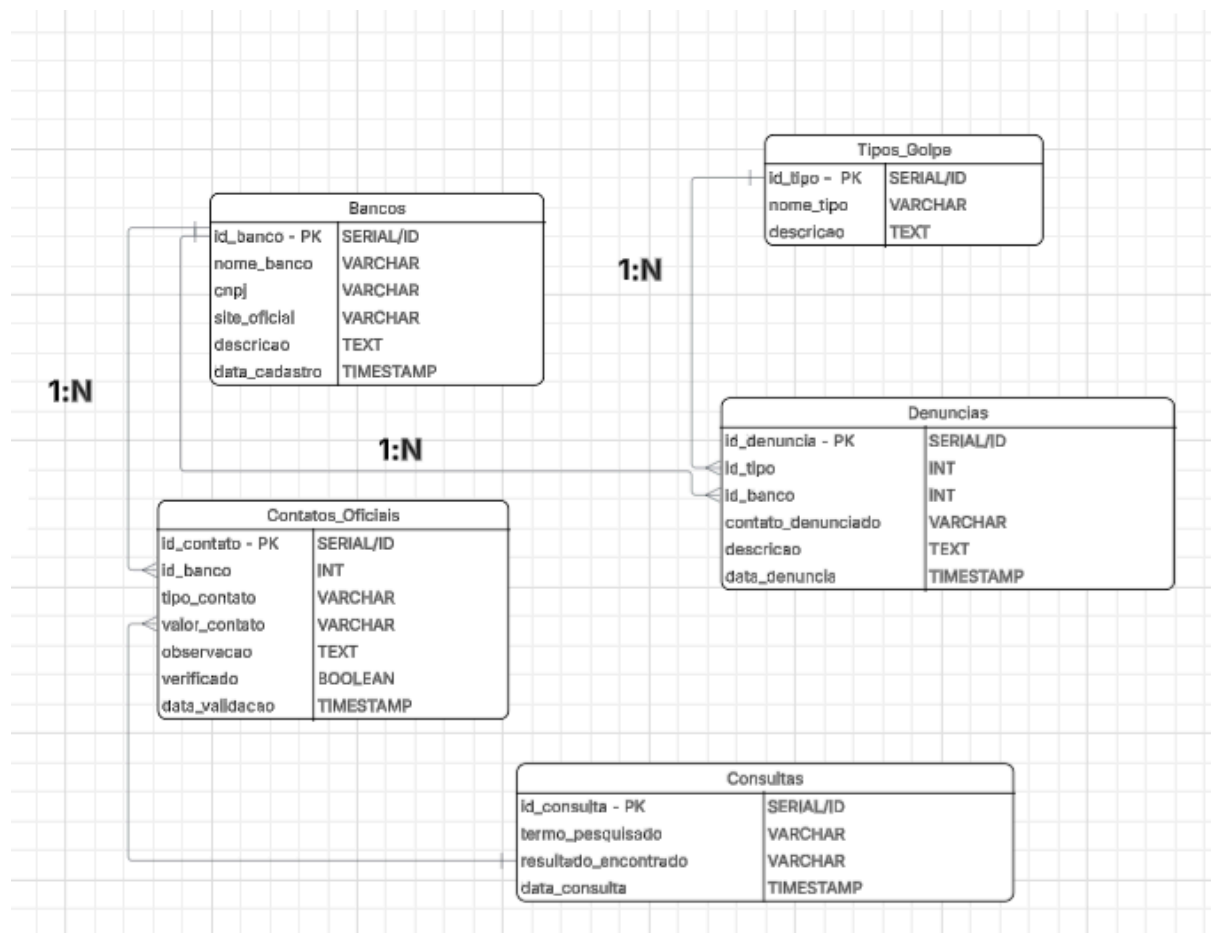
termo_pesquisado: E Mail, número ou site que foi pesquisado no sistema.

resultado_encontrado: Resultado da pesquisa (ex: oficial, suspeito, inexistente).

data_consulta: Data e hora em que a consulta foi realizada.

5 - Relacionamentos entre Tabelas

1. **bancos** → **contatos_oficiais**: um banco pode ter vários contatos oficiais (1:N).
2. **bancos** → **denuncias**: um banco pode ser alvo de várias denúncias (1:N).
3. **tipos_golpe** → **denuncias**: um tipo de golpe pode aparecer em várias denúncias (1:N).



6 - Considerações Finais

O banco de dados foi projetado para **garantir integridade referencial** e **facilidade de manutenção**.

O uso de **IDs auto-incrementais** permite a expansão futura sem conflitos.

Todas as denúncias e contatos oficiais são rastreáveis, assegurando confiabilidade.

O modelo é totalmente compatível com **PostgreSQL**, usando tipos de dados e constraints adequadas.

O projeto conta com dois arquivos principais para implementação:

1. **schema.sql**: contém a definição completa do **esquema do banco de dados**, incluindo a criação das tabelas, tipos de dados, chaves primárias e estrangeiras, e demais constraints.
2. **data.sql**: contém exemplos de **inserts em cada tabela**, permitindo popular o banco de dados com dados iniciais para testes e demonstrações do sistema.