

# HISTÓRICO DE REVISÃO

Versão	Data	Nome	Atividade
	19/09/2023	Piterson Alves	Desenvolvimento Backend
	23/09/2023	Júlia Alcantara	Revisão Backend
	24/09/2023	Piterson Alves	Desenvolvimento Frontend
	27/09/2023	Leonardo Zimmermann	Revisão Frontend
	28/09/2023	Robson Apolinário	Documentação Técnica/Detalhada
	29/09/2023	Gabriel Lerri	Revisão de Documentação
	29/09/2023	Gabriel Lerri	Revisão de Final

## Integrantes:

RM 95449 – Júlia Alcantara Felix

RM 93183 – Gabriel Lerri de Almeida Rodrigues

RM 95192 – Leonardo Zimmermann

RM 94949 – Piterson Roberto Alves Antonio

RM 95107 – Robson Apolinário Teixeira

# Atividade - Challenge Idwall 2º Semestre

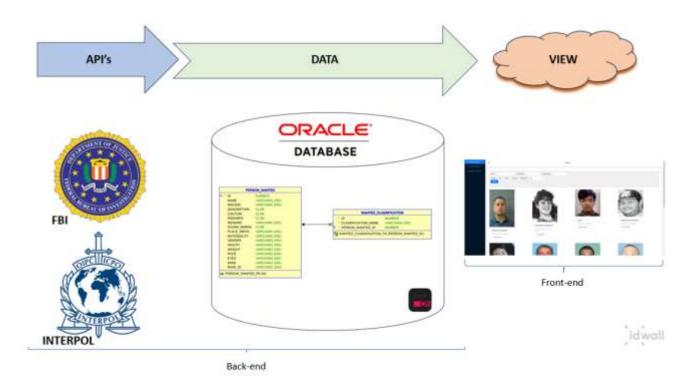
#### 1. Introdução

Nesta documentação, apresentaremos as técnicas avançadas de API que planejamos utilizar para coletar dados relevantes das páginas oficiais do FBI e da Interpol.

O sistema é composto por um front-end React com JavaScript e um back-end Java com JPA para persistência de dados no banco Oracle-FIAP.

#### 2. Documentação

### 2.1 Desenho da Solução



#### 2.2 Documentação Técnica

**IMPORTANTE:** Para acessar a documentação técnica, se faz necessário rodar a aplicação e depois acessar o link abaixo:

Link: http://localhost:8080/swagger-ui/index.html

#### 2.3 Documentação Detalhada

### 2.3.1 Coleta de Dados do FBI e Interpol (Back-end)

**Descrição:** O FBI e a Interpol detêm de uma API, onde conseguimos realizar a busca e coleta das informações dos procurados.

Passos: Consumo da API em nossa aplicação com Webflux, mapeamento dos dados para o Banco

Linguagem: Java

**Bibliotecas Utilizadas:** Java 17, Spring Boot 3, Webflux (lib reativa), Spring Boot Starter Data JDBC e Oracle Data Base JDBC

**Desafios Enfrentados:** Percorrer a paginação da API do FBI, mapear o retorno da API para a tabela do banco de dados, manter a conexão com o banco enquanto carregamos uma grande massa de dados.

**Atualização:** Consideramos uma periodicidade de atualização de 3 minutos nas bases das agências do FBI e Interpol, afim de coletar novos dados.

#### 2.3.2 Visualizações dos Procurados (Front-end)

**Descrição:** Criação de View para listagem dos procurados das agências FBI e Interpol, com filtros e visualizações detalhadas do procurado.

Passos: Consumo da nossa API para buscar os dados no banco com Axios e exibição na tela.

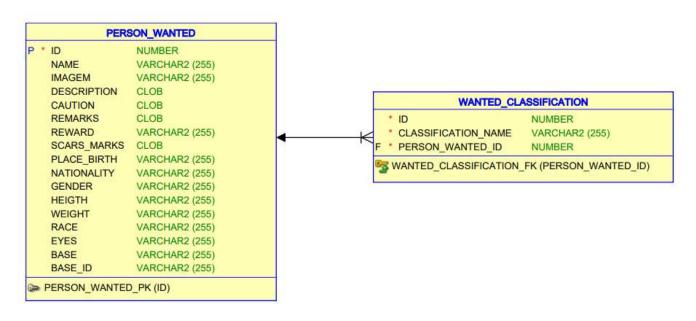
Linguagem: React, TypeScript e Nodejs.

Bibliotecas Utilizadas: Ant Design UI e Axios.

Desafios Enfrentados: Exibir os dados na tela, realizar a paginação, realizar a edição.

#### 2.4 Banco de Dados

#### 2.4.1 Modelagem



#### 2.4.2 Scripts SQL

CREATE TABLE person\_wanted ( id NUMBER PRIMARY KEY,

```
name VARCHAR2(255),
 image VARCHAR2(255),
 description CLOB,
 caution CLOB,
 remarks CLOB,
 reward VARCHAR2(255),
 scars_marks CLOB,
 place_birth VARCHAR2(255),
 nationality VARCHAR2(255),
 gender VARCHAR2(255),
 height VARCHAR2(255),
 weight VARCHAR2(255),
 race VARCHAR2(255),
 eyes VARCHAR2(255),
 base VARCHAR2(255)
);
-- criando uma sequence para a tabela person_wanted
CREATE SEQUENCE person_wanted_seq
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOCACHE
NOCYCLE;
```

```
-- alterando a tabela para person_wanted para usar a sequência na coluna ID
ALTER TABLE person_wanted
MODIFY (
  id NUMBER DEFAULT person_wanted_seq.NEXTVAL
);
CREATE TABLE wanted_classification (
  id NUMBER PRIMARY KEY,
  classification name VARCHAR(255) NOT NULL,
  person_wanted_id INT
);
-- criando uma sequence para a tabela wanted_classification
CREATE SEQUENCE wanted_classification_seq
START WITH 1
INCREMENT BY 1
MINVALUE 1
NOCACHE
NOCYCLE;
-- alterando a tabela para wanted_classification para usar a sequência na coluna ID
ALTER TABLE wanted_classification
MODIFY (
  id NUMBER DEFAULT wanted_classification_seq.NEXTVAL
);
```