

## Roteiro

InLock .....	2
Problema .....	2
Roteiro BD .....	3
Primeiro arquivo: InLock_BD_DDL_01.sql .....	3
Segundo arquivo: InLock_BD_DML_02.sql .....	3
Terceiro arquivo: InLock_BD_DQL_03.sql .....	3
Roteiro API .....	5
Desafio Extra da API .....	5
Entrega .....	6
BD .....	6
BACKEND .....	6
POSTMAN .....	6
PLANEJAMENTO .....	6

## InLock

Ei, você, atente-se a todos os requisitos que estão sendo solicitados, desde o nome do repositório no GitHub, até o nome dos arquivos.

### Problema

Uma empresa do ramo de tecnologia, a InLock, deseja realizar a criação de sua base de dados para armazenar os jogos que são vendidos em sua loja. Além disso, como eles já possuem um desenvolvedor front-end atuando na empresa, não será necessário construir a parte de interface com o usuário. Para isto, deverá ser desenvolvida uma API (trabalhando com JSON), para que o front-end (ou até mesmo o mobile) realize a construção com base nas informações fornecidas.

Os jogos deverão ter as seguintes características:

Nome, Descrição, Data de Lançamento e Valor. Além disso, o jogo deverá ter um estúdio pelo qual foi desenvolvido.

Por exemplo:

O **Diablo 3** foi lançado em **15 de maio de 2012**, **é um jogo que contém bastante ação e é viciante, seja você um novato ou um fã**. Além disso, seu estúdio é a **Blizzard**. E o jogo custa **R\$ 99,00**.

**Red Dead Redemption II** é um **jogo eletrônico de ação-aventura western** desenvolvido pela **Rockstar Studios**. Lançado mundialmente em **26 de outubro de 2018**. E o jogo custa **R\$ 120**.

Além disso, somente usuários com o perfil de **ADMINISTRADOR** poderão cadastrar um novo jogo e qualquer usuário autenticado, com o perfil de **ADMINISTRADOR** ou **CLIENTE**, poderá realizar a listagem de jogos.

## Roteiro BD

### Primeiro arquivo: InLock\_BD\_DDL\_01.sql

Criar um banco de dados chamado InLock\_Games\_Manha/Tarde;

Criar uma tabela de estúdios com os campos de IdEstudio e NomeEstudio;

Criar uma tabela de jogos com os campos IdJogo, NomeJogo, Descricao, DataLancamento, Valor e IdEstudio;

Criar uma tabela de tipos de usuários contendo os campos IdTipoUsuario e Titulo;

Criar uma tabela de usuários contendo os campos de IdUsuario, Email, Senha e IdTipoUsuario;

*Atente-se na definição dos tipos de dados.*

### Segundo arquivo: InLock\_BD\_DML\_02.sql

Inserir um usuário do tipo ADMINISTRADOR que tenha o email igual a 'admin@admin.com' e senha igual a 'admin'.

Inserir um usuário do tipo CLIENTE que tenha o email igual a 'cliente@cliente.com' e senha igual a 'cliente'.

Inserir três estúdios: um com o nome de **Blizzard**, outro com o nome de **Rockstar Studios** e o último com o nome de **Square Enix**;

Inserir um jogo com o nome de: **Diablo 3**, com data de lançamento de: **15 de maio de 2012**, que contenha a descrição de: **é um jogo que contém bastante ação e é viciante, seja você um novato ou um fã**. Seu estúdio é a **Blizzard**. E o jogo custa **R\$ 99,00**.

Inserir um jogo com o nome de: **Red Dead Redemption II**, com a descrição de: **jogo eletrônico de ação-aventura western**. Seu estúdio será a **Rockstar Studios**. Lançado mundialmente em **26 de outubro de 2018**. E o jogo custa **R\$ 120**.

### Terceiro arquivo: InLock\_BD\_DQL\_03.sql

Listar todos os usuários;

Listar todos os estúdios;

Listar todos os jogos;

Listar todos os jogos e seus respectivos estúdios;

Buscar e trazer na lista todos os estúdios com os respectivos jogos. Obs.: Listar todos os estúdios mesmo que eles não contenham nenhum jogo de referência;

Buscar um usuário por email e senha;

Buscar um jogo por IdJogo;

Buscar um estúdio por IdEstudio;

## Roteiro API

Criar um novo projeto do tipo WebAPI com o seguinte nome: '**Senai.InLock.WebApi**'. Sua API deverá ter as seguintes funcionalidades:

O usuário do tipo **ADMINISTRADOR** poderá realizar o cadastro dos jogos;

O usuário do tipo **CLIENTE** ou **ADMINISTRADOR** poderá visualizar a lista de todos os jogos e seus respectivos estúdios;

Sua autenticação deverá ser feita utilizando JWT (não esquecer de criar um endpoint para gerar o token);

Utilizar o **Swagger** para a documentação de sua API;

Não esquecer de utilizar as abordagens: DataAnnotations e ViewModels;

## Desafio Extra da API

Mostrar a lista de todos os estúdios e incluir na lista, a lista de jogos daquele determinado estúdio;

## Entrega

Criar um **repositório no GitHub** com o nome de senai-backend-inlock-Nome1-Nome2 com a seguinte estrutura de pastas:

### BD

InLock\_BD\_DDL\_01.sql

InLock\_BD\_DML\_02.sql

InLock\_BD\_DQL\_03.sql

Exportar o diagrama do banco de dados e incluir a imagem dentro deste diretório;

### BACKEND

Solução do Projeto (não esqueça que o arquivo .sln **não** armazena os demais arquivos do seu projeto, é preciso entregar toda a estrutura criada)

### POSTMAN

Exportar a coleção de requisições criada com estrutura de pastas

Estudios

Jogos

Login

### PLANEJAMENTO

Convidar os professores para o quadro de organização de tarefas (Trello) e criar um arquivo .txt com o link para fácil acesso