Roteiro

[InLock 2](#_Toc18246791)

[Problema 2](#_Toc18246792)

[Roteiro BD 3](#_Toc18246793)

[Primeiro arquivo: 01\_CRIACAO\_BD.sql 3](#_Toc18246794)

[Segundo arquivo: 02\_INSERCAO\_BD.sql 3](#_Toc18246795)

[Terceiro arquivo: 03\_SELECAO\_BD.sql 3](#_Toc18246796)

[Roteiro API 5](#_Toc18246797)

[Desafios Extras da API 5](#_Toc18246798)

[Entrega 6](#_Toc18246799)

[BD 6](#_Toc18246800)

[BACKEND 6](#_Toc18246801)

[Postman 6](#_Toc18246802)

# 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 e 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**, poderão realizar a listagem de jogos.

# Roteiro BD

## Primeiro arquivo: 01\_CRIACAO\_BD.sql

Criar um banco de dados chamado M\_InLock/T\_InLock;

Criar uma tabela de estúdios com os campos de EstudioId, NomeEstudio e País de Origem (PaisOrigem), Data de Criação (DataCriacao); Além disso, ao cadastrar o estúdio, deverá ser incluído na tabela um campo para mostrar qual foi o usuário que realizou o cadastro do jogo chamado UsuarioId;

Criar uma tabela de jogos com os campos JogoId, NomeJogo, Descricao, DataLancamento, Valor e EstudioId;

Criar uma tabela de usuários contendo os campos de UsuarioId, Email, Senha e Permissão do Usuário;

*Atente-se na definição dos tipos de dados e nome dos campos.*

## Segundo arquivo: 02\_INSERCAO\_BD.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**; Todos estes foram o usuário [admin@admin.com](mailto:admin@admin.com) que realizou o cadastro.

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**. E 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: 03\_SELECAO\_BD.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, mesmo que eles não contenham nenhum jogo de referência;

Buscar um usuário por email e senha;

Buscar um jogo por JogoId;

Buscar um estúdio por EstudioId;

# 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 crud dos jogos;

O usuário do tipo **ADMINISTRADOR** poderá realizar o crud dos estúdios;

Qualquer usuário autenticadopoderá 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 uma das abordagens: DataAnnotations ou ViewModels, para validação;

## Desafios Extras da API

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

Mostrar uma lista com os 5 jogos mais caros;

Mostrar os jogos por data de lançamento (os mais recentes primeiro);

Mostrar na lista de lançamentos, quantos dias faltam para o lançamento. Caso o lançamento já tenha sido feito, apresentar como 0 (zero);

Realizar uma busca por nome do jogo;

Realizar uma busca por nome do estúdio, mas que apresente todos os jogos;

Realizar uma busca por país e mostrar todos os estúdios daquele determinado país;

Realizar uma busca por país e além de mostrar os estúdios, apresentar os jogos;

Criar um endpoint para os administradores visualizarem somente os estúdios que foram inseridos por ele;

Criar um endpoint para visualizar somente os estúdios com data de criação inferior a 10 dias passados (dado a data atual);

# Entrega

Criar um **repositório no GitHub (ou utilizar o seu de backend existente)** com o nome de 2s2019-t2-sprint-2-inlock com a seguinte estrutura de pastas:

## BD

01\_CRIACAO\_BD.sql

02\_INSERCAO\_BD.sql

03\_SELECAO\_BD.sql

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

## BACKEND

Solução do Projeto;

## Postman

Exportar a coleção;