

# Projeto CRUD - Clientes

Este repositório contém uma aplicação Spring Boot contemplando toda a parte do backend do que seria uma aplicação CRUD para clientes. Utiliza de serviços de geolocalização de IP e clima para recuperar informações e associar aos clientes cadastrados.

## O projeto

O projeto foi desenvolvido no IntelliJ IDEA. Esta IDE tem uma ótima integração com o Spring e Maven, assim como inteligência de código referência. Para a geração do projeto, foi utilizada a ferramenta Spring Initializr (integrada com a IDE).

## Spring Boot

Para a aplicação em si. Roda o servidor de forma que outras configurações de servidores externas não sejam necessárias, assim temos uma aplicação que rodaria em qualquer máquina.

## H2 database - Banco de dados

Banco de dados "in-memory". Configuração rápida, confiável e suave integração com o Spring Data (JPA).

Para acessar o banco de dados na aplicação:

`{urlBase}/projeto-cliente/banco`

Usuário: adm

Senha: uol

Exemplo (local): <http://localhost:8080/projeto-cliente/banco>

## Maven

Gerenciamento de dependências e build do projeto. O projeto maven foi gerado a partir da ferramenta Spring Initializr, citada anteriormente.

## Swagger

Documentação de toda parte REST da aplicação. Além de documentar, é possível testar os controladores (métodos) de toda a aplicação através de uma interface amigável no contexto: `/swagger-ui.html`.

# Iniciando

Necessário: Maven.

Para executar a aplicação há três maneiras:

## 1. IDE

Para iniciar direto na IDE basta baixar o repositório em sua máquina e importar em seu ambiente. Executar a aplicação como uma aplicação normal Java pela classe principal.

## 2. Maven

Com o repositório baixado, executar o comando: `mvn spring-boot:run`

## 3. Deploy do WAR em servidor

Se preferir rodar em um servidor (para fins de Produção por exemplo), o Maven gera o .war do projeto. Para isso, executar o seguinte comando: `mvn clean install` O .war da aplicação estará disponível em /target (raiz do projeto).

## Acesso

`{urlBase}/projeto-cliente`

Exemplo (local): <http://localhost:8080/projeto-cliente>

# Testes dos métodos REST

Nesta aplicação, foi implementado o framework Swagger para documentação e testes de serviços REST. Utiliza-se de uma interface gráfica "swagger-ui".

Acesso:

`{urlBase}/projeto-cliente/swagger-ui.html`

Exemplo (local): <http://localhost:8080/projeto-cliente/swagger-ui.html>

# Documentação completa

**Documentação da API:** Se referir ao documento API\_Docs\_ProjetoCliente

**Especificação do projeto:** Se referir ao documento TEST\_PLATCORP\_V1

Toda a documentação se encontra na pasta docs do projeto.