

04

Rodando o projeto Android e servidor

Neste curso vamos utilizar o mesmo projeto que foi desenvolvido no [segundo curso de testes de unidade](https://cursos.alura.com.br/course/android-boas-praticas-e-cenarios-testes) (<https://cursos.alura.com.br/course/android-boas-praticas-e-cenarios-testes>), porém, com algumas features novas e modificações simulando um projeto legado. Sendo assim, é necessário realizar o download do projeto novamente.

Além do projeto Android, vamos utilizar uma API que vai ficar responsável em manter as informações dos leilões. Isso significa que vamos rodar um servidor similar ao que é visto no primeiro curso de sincronização de Apps [Android com Web Service](https://cursos.alura.com.br/course/android-sincronizacao-com-servidor) (<https://cursos.alura.com.br/course/android-sincronizacao-com-servidor>).

Download dos arquivos iniciais

Os arquivos iniciais podem ser baixados por meio [deste link](https://github.com/alura-cursos/android-testes-mock/archive/arquivos-iniciais.zip) (<https://github.com/alura-cursos/android-testes-mock/archive/arquivos-iniciais.zip>) (Aproximadamente 33 MB). Ao finalizar o download, vai aparecer zip chamado **android-testes-mock-arquivos-iniciais**. Basta apenas realizar a extração. Após a extração vai aparecer os seguintes diretórios:

- **Leilao**: projeto Android com as modificações;
- **servidor**: diretório que mantém o arquivo **jar** do servidor.

Rodando o projeto Android

Para o projeto Android, abra o Android Studio na versão 3.1.3 ou superior e abra-o a partir da opção **Open an existing Android Studio project**, busque o projeto e clique em **OK**. Após finalizar as configurações execute o projeto em um dispositivo que preferir.

Nessa execução, deve aparecer a tela inicial em branco e um Toast com a mensagem: "Não foi possível carregar os leilões". Essa mensagem aparece, pois não foi feita a integração da App com a API.

Rodando a API

Pré-requisitos: é necessário instalar a versão 8 do Java para rodar o executável do servidor sem nenhum problema. Pode ser que ele funcione com outras versões, porém, ele foi desenvolvido nessa versão em específico, por isso ela é exigida.

Para rodar a API, abra o prompt de comando se estiver no Windows ou o terminal caso seja Linux ou Mac OS e vá até o diretório **servidor** que contém o arquivo **server.jar**. Então execute o comando:

```
java -jar server.jar
```

Se tudo der certo, no final vai aparecer uma mensagem similar a esta:

```
2018-07-30 15:12:04.277 INFO 17533 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : 
2018-07-30 15:12:04.281 INFO 17533 --- [           main] b.c.a.leilaoapi.LeilaoApiApplicationKt :
```

Perceba que a mensagem **Tomcat started on port(s) 8080 (http)** indica que a inicialização foi feita e que o servidor está acessível nesta porta via protocolo HTTP. Para testá-lo basta apenas acessar a URL <http://localhost:8080/> (<http://localhost:8080/>).

Possíveis problemas com a execução do servidor

Se tiver algum serviço utilizando a porta 8080, provavelmente o servidor não vai funcionar, dado que a porta padrão é a 8080. Sendo assim, mate qualquer processo que estiver usando essa porta.

Para situações que matar o processo da porta 8080 não for possível, execute o servidor em uma outra porta utilizando o argumento `--port=numero_da_porta`, por exemplo, `java -jar server.jar --port=8081`. **Essa alternativa não permite executar dois servidores que utilizem a mesma referência do mesmo banco de dados (diretório Database).**

Se tudo estiver funcionando como esperado, podemos seguir com a próxima atividade para integrar a App Android com o servidor. Caso contrário, entre em contato com a gente no fórum.