

## Mãos à obra: Criando arquivo Dockerfile

Uma vez que temos o ambiente de desenvolvimento e o ambiente de produção configurados de diferentes formas, tivemos o problema de não conseguir acessar a aplicação da Casa do Código no ambiente de produção que configuramos através do *Elastic Beanstalk*, vamos resolver esse problema com o Docker.

O primeiro passo é criar justamente uma imagem customizada com a aplicação da Casa do Código com o Tomcat na versão 9 através do arquivo **Dockerfile** para que façamos primeiramente o teste local, se funcionar o teste local, levaremos para o ambiente de produção. Abra o editor de texto de sua preferência e monte o **Dockerfile** abaixo:

```
FROM tomcat:9-jre8
RUN rm -r /usr/local/tomcat/webapps/
COPY ROOT.war /usr/local/tomcat/webapps/
```

Salve esse arquivo na pasta **Beanstalk\_dev** que criamos nas etapas anteriores e irá conter todo o conteúdo de desenvolvimento. Feito isso, volte até o eclipse e altere o projeto da Casa do Código para o perfil de desenvolvimento indo até a classe **ServletSpringMVC** presente no pacote *br.com.casadocodigo.loja.configuration*

```
@Override
public void onStartUp(ServletContext servletContext) throws ServletException {
    super.onStartUp(servletContext);
    servletContext.addListener(RequestContextListener.class);
    servletContext.setInitParameter("spring.profiles.active", "dev");
}
```

Feito isso, a nossa aplicação da Casa do Código para funcionar, precisa realizar a comunicação com o banco de dados. Vamos nesse teste local configurar um container para trabalhar com o banco de dados, para isso, devemos alterar o método **dataSourceDev** presente na classe **JPAConfiguration** com o perfil de desenvolvimento. Vamos passar tais informações de conexão como sendo uma variável de ambiente no container que iremos configurar:

```
@Bean
@Profile("dev")
public DataSource dataSourceDev() {
    DriverManagerDataSource dataSource = new DriverManagerDataSource();
    dataSource.setUsername(System.getenv("USUARIO"));
    dataSource.setPassword(System.getenv("SENHA"));
    dataSource.setUrl(System.getenv("JDBC_CONNECTION_STRING"));
    dataSource.setDriverClassName("com.mysql.jdbc.Driver");
    return dataSource;
}
```

Feito isso, exporte o projeto para o arquivo *ROOT.war*, salvando esse arquivo na pasta **Beanstalk\_dev**. Nesse momento, a pasta **Beanstalk\_dev** deverá conter tanto o arquivo *Dockerfile* como o *ROOT.war*

Agora nós devemos montar essa imagem customizada, com a aplicação da Casa do Código e o Tomcat na versão 9. Para isso, caso ainda não tenha o Docker instalado, você poderá acompanhar os passos de instalação nos links abaixo de acordo com seu sistema operacional:

- [Windows \(https://cursos.alura.com.br/course/docker-e-docker-compose/task/28504\)](https://cursos.alura.com.br/course/docker-e-docker-compose/task/28504)

- [MacOS \(https://cursos.alura.com.br/course/docker-e-docker-compose/task/28592\)](https://cursos.alura.com.br/course/docker-e-docker-compose/task/28592)
- [Linux \(https://cursos.alura.com.br/course/docker-e-docker-compose/task/28593\)](https://cursos.alura.com.br/course/docker-e-docker-compose/task/28593)

Abra o *Docker Quick Start Terminal* se estiver no Windows ou o terminal se estiver no Linux ou no MacOS. No terminal, mova os diretórios com o comando `cd [nome do diretório]` até chegar a pasta **Beanstalk\_dev**<sup>1</sup> onde temos os arquivos salvos e digite:

```
docker build -t "casadocodigo:v1-dev" .
```

Esse processo deverá demorar alguns minutos, no final devemos ter a imagem construída.

<sup>1</sup>. No Windows, para que você consiga acessar a pasta **Beanstalk\_dev** e **Beanstalk\_prod** através do *Docker Quickstart Terminal* elas deverão estar dentro de uma pasta a qual o programa conseguirá acessar que por padrão deverá estar dentro da pasta referente ao seu usuário. Por exemplo: C:\Users\Rafael