

Preparando o ambiente

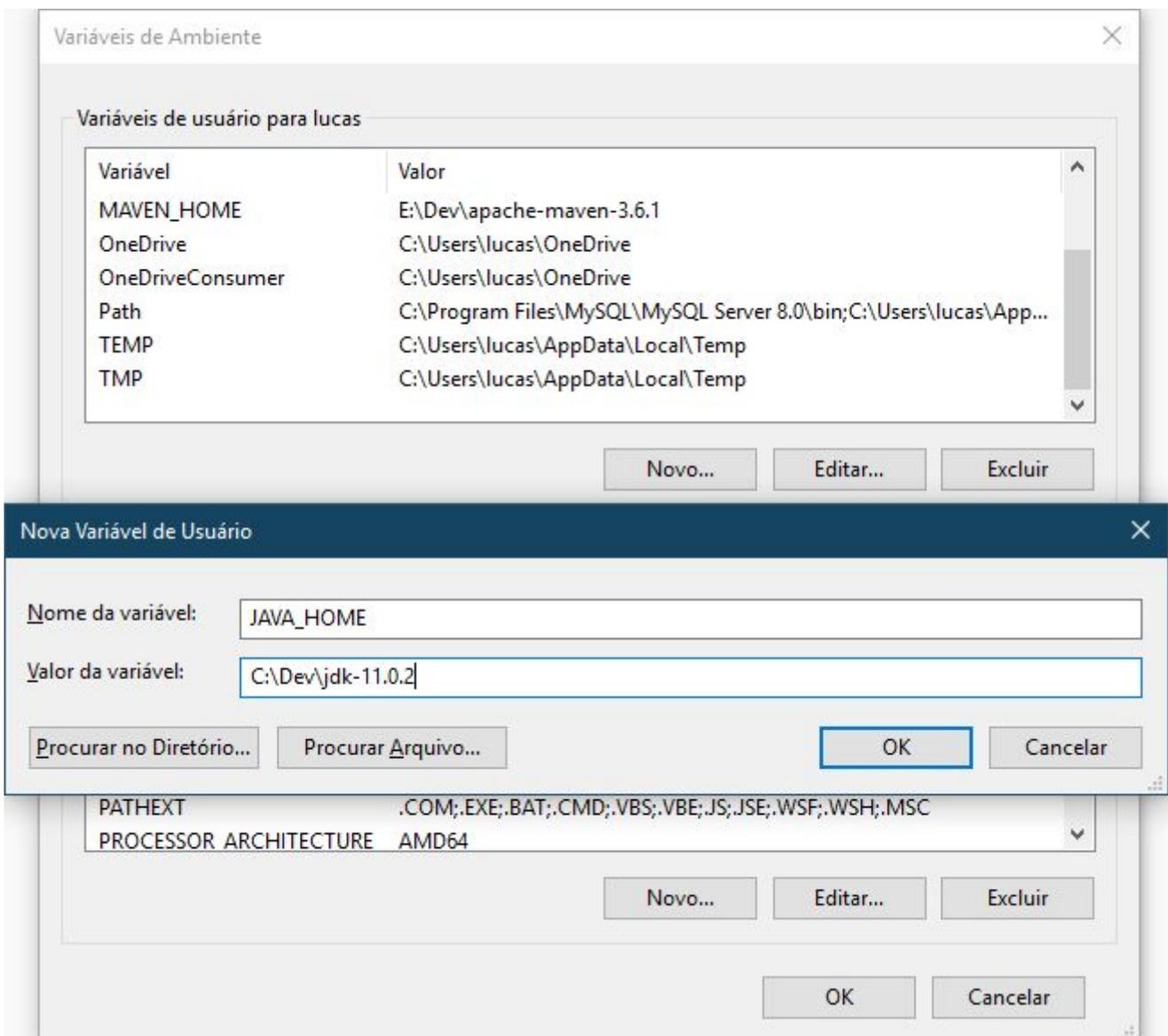
Antes de participar da apresentação, é desejável já estar com algumas das ferramentas que usaremos preparadas para uso, de forma a diminuir o tempo com os preparativos e podermos colocar mais rapidamente a mão na massa.

Java

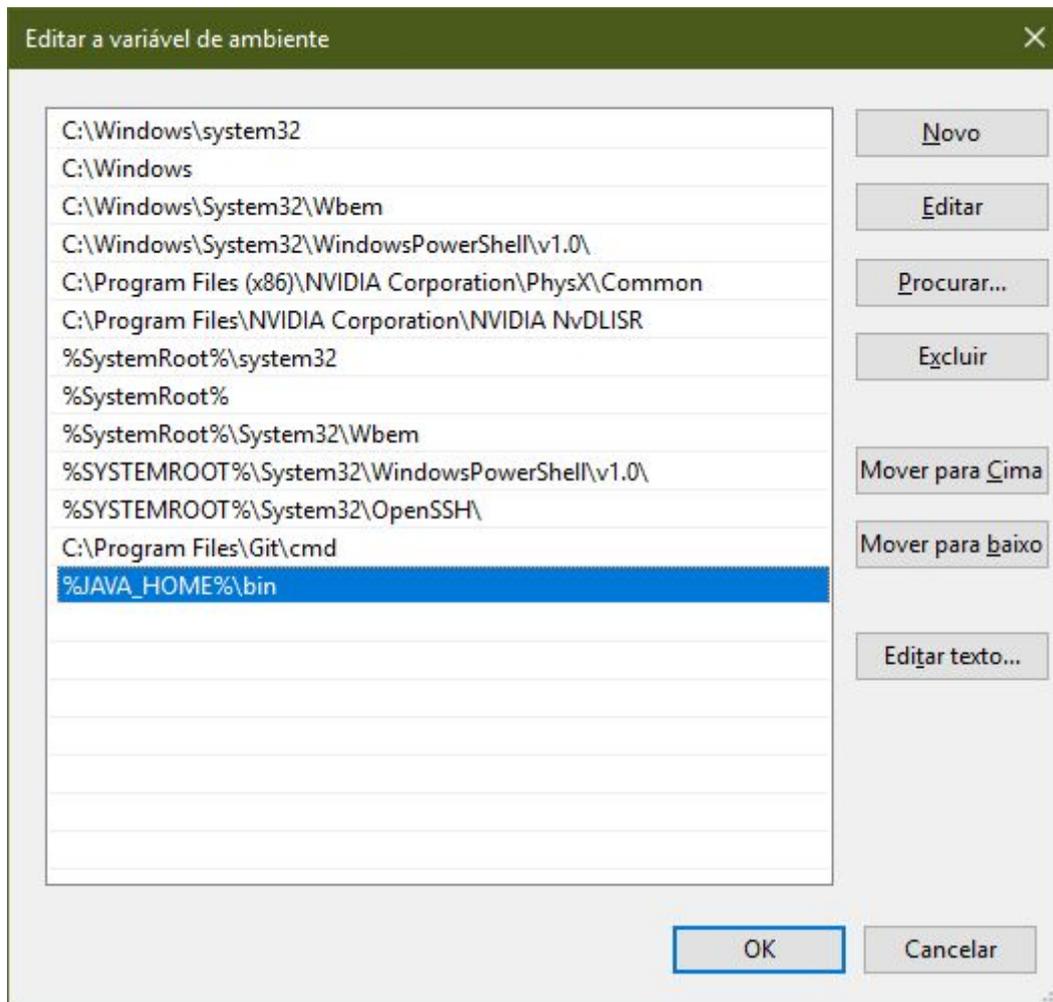
Como vamos desenvolver nosso back-end em Java, devemos baixar uma versão da JDK, o kit de desenvolvimento da linguagem Java. Vamos utilizar o OpenJDK na versão 11.0.2, que pode ser baixado no site do [Java.net](https://java.net).

Baixe a versão 11.0.2 no formato .zip ou .tar.gz de acordo com seu sistema operacional e descompacte-o em alguma pasta (por exemplo, "C:\Dev" ou "/home/user/dev"). Em seguida, é necessário adicionar a pasta da JDK à variável de ambiente JAVA_HOME.

No Windows, se a pasta "jdk-11.0.2" se encontra, por exemplo, no caminho "C:\Dev\jdk-11.0.2", acesse o Painel de Controle > Sistema e Segurança > Sistema > Configurações avançadas do sistema > Variáveis de ambiente. Há uma tabela com as variáveis do usuário e uma com as do sistema; clique no botão "Novo..." de qualquer uma delas, crie uma variável chamada JAVA_HOME e no valor cole o caminho da pasta:



Em seguida, é necessário adicionar outro valor à variável PATH; você pode tanto criá-la ou editá-la na tabela de variáveis de usuário ou editar a já existente nas variáveis de sistema. Adicione o valor “%JAVA_HOME%\bin” nela:



No Linux e no Mac, se a pasta se encontra, por exemplo, no caminho “/home/user/dev/jdk-11.0.2”, para definir a variável de ambiente, edite o arquivo de perfil do seu usuário (normalmente chamado de .bashrc ou .bash_profile) e acrescente as seguintes linhas ao final:

```
export JAVA_HOME=/home/user/dev/jdk-11.0.2
export PATH=$PATH:$JAVA_HOME/bin
```

Para conferir se tudo deu certo, abra o prompt de comando (Windows) ou o terminal (Linux/Mac) e digite o comando “java -version”; se tudo der certo, uma mensagem parecida com a seguinte será exibida:

```
C:\Users\lucas>java -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)
```

Apache Maven

Quando desenvolvemos softwares cada vez mais complexos, é necessário que utilizemos bibliotecas para nos ajudar a executar alguma operação mais avançada sem que precisemos ficar repetindo códigos que outras pessoas já prepararam antes. O problema é que gerenciar essas bibliotecas e suas versões é uma tarefa extremamente demorada e difícil, principalmente quando essas bibliotecas dependem de outras, que dependem de outras...

Para ajudar nisso, a fundação Apache desenvolveu um software chamado Maven que, com a ajuda de um repositório on-line de bibliotecas, facilita demais esse trabalho, buscando e disponibilizando as bibliotecas que precisaremos utilizar, bastando apenas que configurarmos um arquivo XML.

O Maven pode ser baixado no [site oficial](#); basta baixar a versão “bin”, seja no formato .zip ou .tar.gz.

Primeiramente, descompacte o arquivo em alguma pasta (por exemplo: “C:\Dev” ou “/home/user/dev”). Em seguida, edite a variável de ambiente PATH e acrescente o caminho da pasta “bin” do Maven (por exemplo, “C:\Dev\apache-maven-3.6.1\bin”). Para conferir, novamente abra o prompt ou terminal e digite “mvn -version”:

```
C:\Users\lucas>mvn -version
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8fcc8e6af555; 2019-04-04T16:00:29-03:00)
Maven home: E:\Dev\apache-maven-3.6.1\bin\..
Java version: 11.0.2, vendor: Oracle Corporation, runtime: E:\Dev\jdk-11.0.2
Default locale: pt_BR, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Banco de Dados

Os dados gerenciados por nossa aplicação devem ficar armazenados em algum lugar. Normalmente, isso é feito por um servidor de banco de dados, que pode ser relacional (MySQL, Oracle, PostgreSQL etc.) ou não-relacional (Redis, MongoDB, Cassandra etc.). Na nossa aplicação, utilizaremos o MySQL na versão Community.

O [site do MySQL](#) disponibiliza diversos instaladores para o servidor para diversos sistemas operacionais. Normalmente a interface é bem intuitiva e basta seguir o passo-a-passo. No tipo de instalação, é possível selecionar “Custom” para selecionar quais os componentes devem ser instalados; a maioria deles pode ser deixada de lado, pois são apenas utilitários pouco usados. É necessário apenas que se instale o servidor em si e, opcional mas recomendável, o MySQL Workbench, uma interface visual prática para gerenciar nosso banco.

Na instalação, será pedido para definir um nome de usuário (por padrão, “root”) e uma senha. Esse usuário será o administrador do banco e será responsável pela criação e configuração inicial dos bancos que usaremos nas aplicações.

Ao final, o MySQL deve ter sido configurado automaticamente para a variável PATH. Você pode acessá-lo pela linha de comando com o comando “mysql -uroot -psenha” (em que “root” e “senha” são o usuário e a senha que você definiu). Alternativamente, pode-se acessá-lo também pelo MySQL Workbench.

IDE

Juntamente com a linguagem, normalmente é necessário dispor de uma IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) para facilitar o desenvolvimento das aplicações. Para a linguagem Java e para um projeto Spring Boot, as IDEs mais recomendadas são o [IntelliJ IDEA](#), o [NetBeans](#), o [Eclipse](#) ou a [Spring Tool Suite](#).

Para o desenvolvimento com Spring Boot, o IntelliJ possui diversas funcionalidades excelentes na versão, Ultimate, que é paga mas permite uso gratuito por 30 dias, além de possuir parcerias para desenvolvedores de software livre e também com universidades como a USP para uma licença gratuita.

As outras IDEs são gratuitas e podem ser baixadas livremente em seus respectivos sites.

Spring Inicializ

Fazer a configuração inicial de um projeto Spring Boot não é muito simples; o Spring Boot permite que configuremos nossa aplicação com diversas bibliotecas desenvolvidas pela comunidade Spring, além de possuir suporte a outras linguagens (como Kotlin e Groovy) e outras ferramentas de desenvolvimento

(como o Gradle no lugar do Maven). Para facilitar esse primeiro passo, o site [Spring Initializr](#) permite que você selecione quais bibliotecas (chamadas “dependências”) deseja, além da linguagem e versão que deseja usar.

Para nossa aplicação, selecione “Maven Project”, “Java” e, na versão do Spring Boot, selecione a mais recente que não possuir “SNAPSHOT” ou “M” no nome (pois elas normalmente são versões ainda em desenvolvimento e podem ser instáveis). No momento, a versão mais recente é a 2.1.6.

Em seguida, dê um nome de um grupo (normalmente é o domínio web de uma empresa, mas pode ser qualquer nome, com partes preferencialmente separadas por pontos; por exemplo: br.usp.lucas) e de artefato (o nome da aplicação, com partes separadas por hífens; por exemplo, my-application).

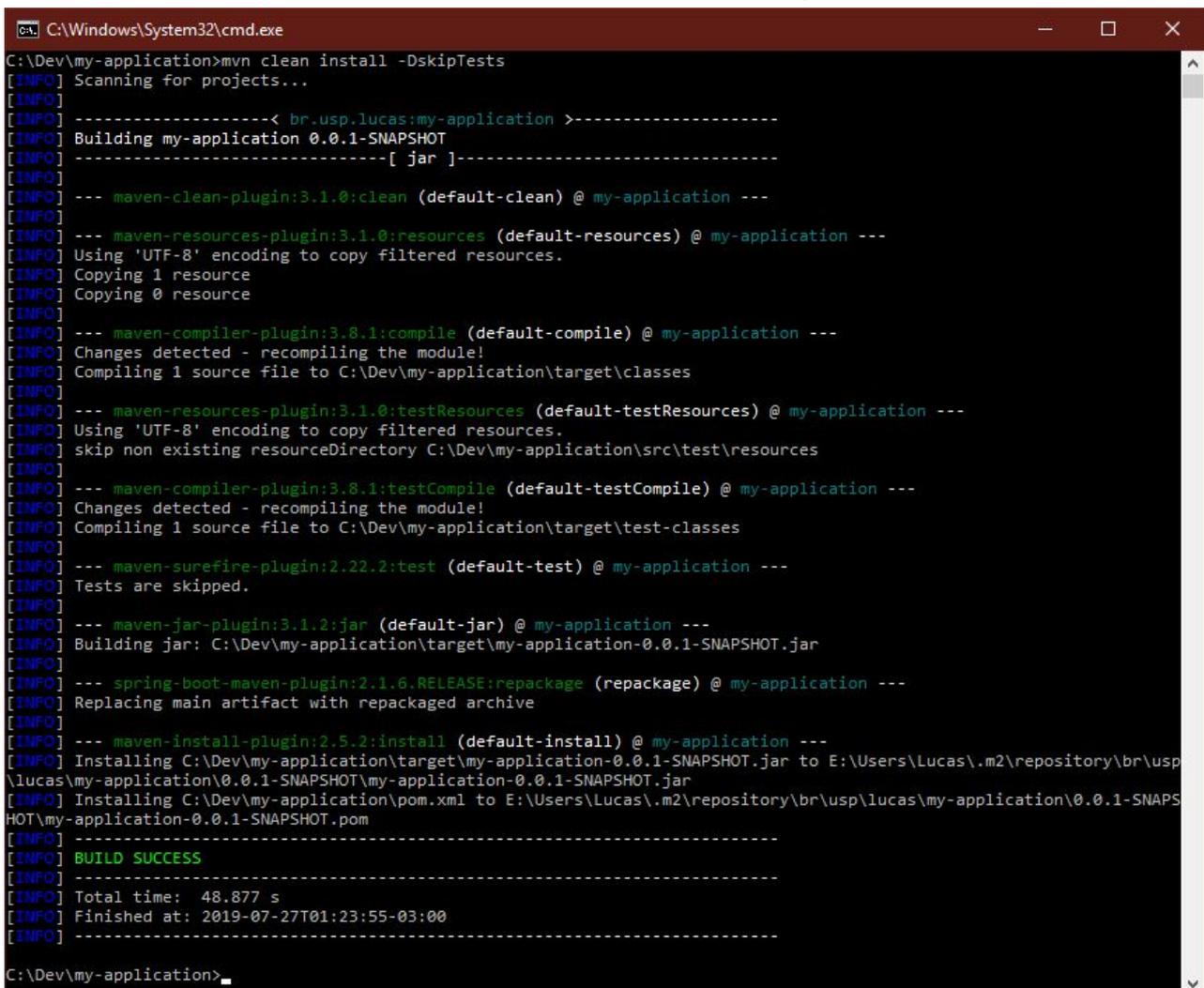
Expandindo o “Options”, você pode definir outro nome para a aplicação (normalmente é o mesmo do artefato), mudar a descrição e o “pacote raiz” da aplicação, onde todas as classes Java deverão ficar. Ainda aqui, selecione o “Packaging” como “Jar” e a versão do Java como a 11.

Na seção seguinte, devemos selecionar as dependências da aplicação. Você pode tanto pesquisá-las ou exibi-las como uma lista, e dá para se ter uma noção da quantidade de opções que temos! Selecione as seguintes:

- Spring Web Starter
- MySQL Driver
- JDBC API

Basta apenas clicar em “Generate the project” e um arquivo .zip será disponibilizado para download com toda a configuração básica do projeto pronta! Descompacte o projeto em alguma pasta (por exemplo, na “C:\Dev” ou “/home/user/dev” utilizada anteriormente).

O próximo passo não precisa ser feito agora, mas se for feito pode economizar um bom tempo durante a apresentação: abra a pasta do projeto no prompt/terminal e execute o comando “mvn clean install -DskipTests” para baixar todas as dependências e fazer uma compilação inicial:



```
C:\Windows\System32\cmd.exe
C:\Dev\my-application>mvn clean install -DskipTests
[INFO] Scanning for projects...
[INFO]
-----< br.usp.lucas:my-application >-----
[INFO] Building my-application 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
--- maven-clean-plugin:3.1.0:clean (default-clean) @ my-application ---
[INFO]
--- maven-resources-plugin:3.1.0:resources (default-resources) @ my-application ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 0 resource
[INFO]
--- maven-compiler-plugin:3.8.1:compile (default-compile) @ my-application ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Dev\my-application\target\classes
[INFO]
--- maven-resources-plugin:3.1.0:testResources (default-testResources) @ my-application ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Dev\my-application\src\test\resources
[INFO]
--- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ my-application ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Dev\my-application\target\test-classes
[INFO]
--- maven-surefire-plugin:2.22.2:test (default-test) @ my-application ---
[INFO] Tests are skipped.
[INFO]
--- maven-jar-plugin:3.1.2:jar (default-jar) @ my-application ---
[INFO] Building jar: C:\Dev\my-application\target\my-application-0.0.1-SNAPSHOT.jar
[INFO]
--- spring-boot-maven-plugin:2.1.6.RELEASE:repackage (repackage) @ my-application ---
[INFO] Replacing main artifact with repackaged archive
[INFO]
--- maven-install-plugin:2.5.2:install (default-install) @ my-application ---
[INFO] Installing C:\Dev\my-application\target\my-application-0.0.1-SNAPSHOT.jar to E:\Users\Lucas\.m2\repository\br\usp\lucas\my-application\0.0.1-SNAPSHOT\my-application-0.0.1-SNAPSHOT.jar
[INFO] Installing C:\Dev\my-application\pom.xml to E:\Users\Lucas\.m2\repository\br\usp\lucas\my-application\0.0.1-SNAPSHOT\my-application-0.0.1-SNAPSHOT.pom
[INFO]
-----
[INFO] BUILD SUCCESS
[INFO]
-----
[INFO] Total time: 48.877 s
[INFO] Finished at: 2019-07-27T01:23:55-03:00
[INFO]
-----
C:\Dev\my-application>
```

Feito isso, o projeto já está preparado para começarmos a desenvolvê-lo :) A pasta do projeto pode ser aberta na IDE que você escolheu, onde você poderá explorar a estrutura que foi gerada automaticamente. Não entraremos em detalhes dela agora; isso será mostrado durante a apresentação.

Até lá!

Qualquer dúvida, entre em contato: lucas_jose_92@hotmail.com ou lucasjss92@gmail.com