

Criando um Projeto de testes automatizado

Nesta apostila segue um passo a passo de como você poderá criar um projeto de testes automatizado utilizando as tecnologias Java e Selenium Webdriver.

Configurando o ambiente:

Instalando o JDK

Para que o nosso projeto seja criado e funcione sem problemas, precisamos utilizar uma versão do JDK 1.8 ou superior. Abaixo seguiremos com a instalação e configuração do ambiente utilizando a versão 1.8.

Link download JDK 1.8:

<https://www.oracle.com/br/java/technologies/javase/javase8-archive-downloads.html>

Link download JDK 11:

<https://www.oracle.com/br/java/technologies/javase-jdk11-downloads.html>

Instalando o Maven 3.8.x

O processo de configuração do Maven é similar ao do JAVA, porém o Maven não será instalado, nós apenas iremos baixar e descompactar o seu diretório para aponta-lo no sistema. Siga os passos abaixo para realizar o download e configurar o Maven em sua máquina.

Para realizar o download do maven, acesse o link: <https://maven.apache.org/download.cgi>

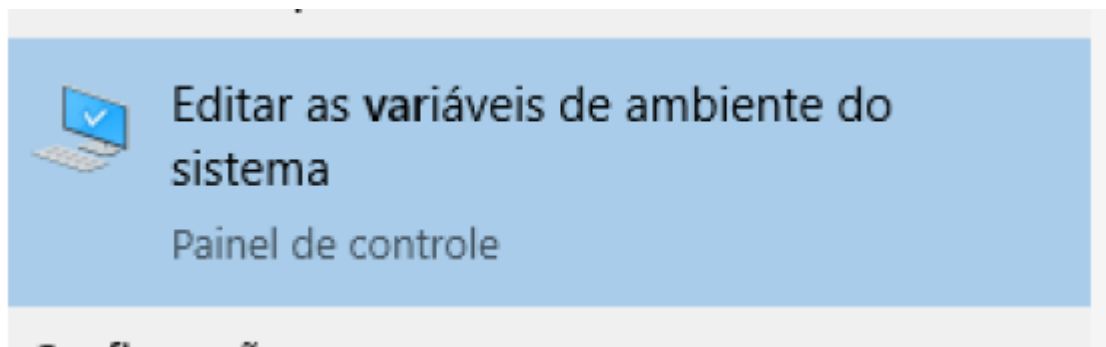
Ao acessar o link acima você será direcionado para o site de download do maven, faça o download do arquivo bin.zip conforme indicado em vermelho na imagem abaixo

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.8.2-bin.tar.gz	apache-maven-3.8.2-bin.tar.gz.sha512	apache-maven-3.8.2-bin.tar.gz.asc
Binary zip archive	apache-maven-3.8.2-bin.zip	apache-maven-3.8.2-bin.zip.sha512	apache-maven-3.8.2-bin.zip.asc
Source tar.gz archive	apache-maven-3.8.2-src.tar.gz	apache-maven-3.8.2-src.tar.gz.sha512	apache-maven-3.8.2-src.tar.gz.asc
Source zip archive	apache-maven-3.8.2-src.zip	apache-maven-3.8.2-src.zip.sha512	apache-maven-3.8.2-src.zip.asc

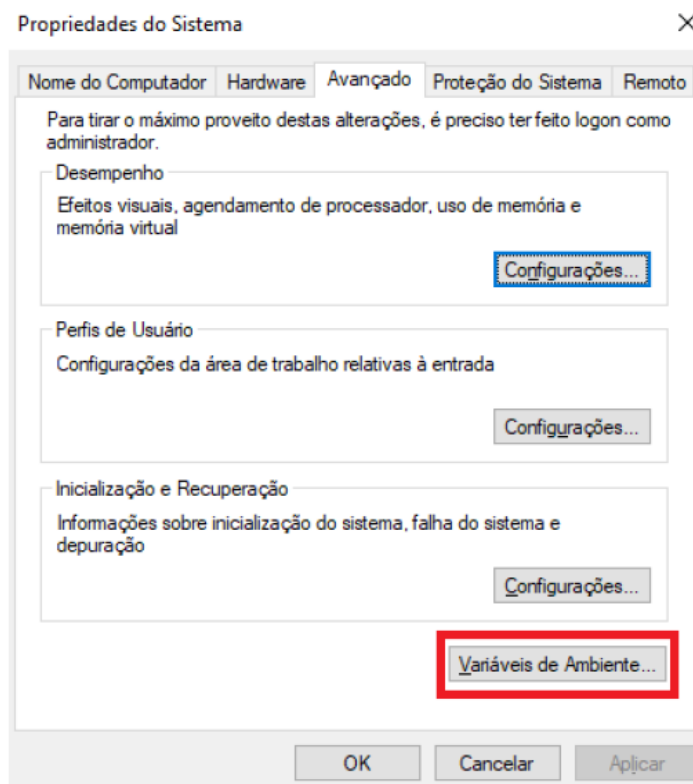
Após realizar o download extraia o arquivo em um local de sua preferência, porém de fácil acesso pois precisaremos utilizar o caminho onde o arquivo foi extraído para realizar as configurações das variáveis de ambiente Maven.

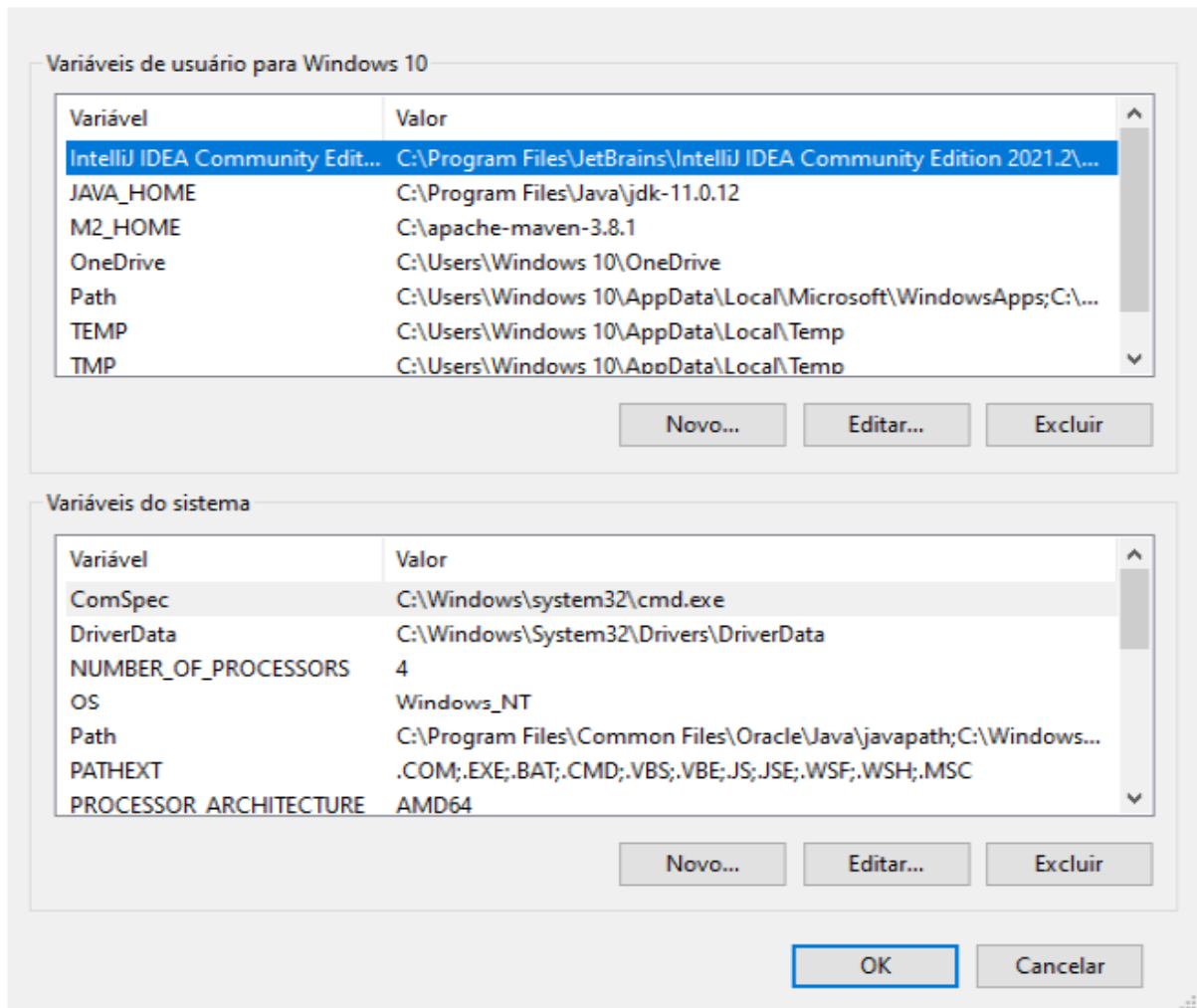
Configurando as Variáveis de ambiente JAVA

No menu iniciar do Windows pesquise por “variáveis” , clique na opção “Editar as variáveis de Ambiente” como no exemplo abaixo:



Em seguida clique em “**Variáveis de Ambiente**” como no exemplo abaixo:





Nessa tela temos 2 opções para configurar nossas variáveis. Configurando as variáveis, em “Variáveis de Usuário ” na parte superior da tela, nossas configurações serão válidas apenas para o usuário logado, ou seja, caso você possua mais usuários configurados em sua máquina, as variáveis não serão visíveis para os demais. Se houver necessidade de configurar as variáveis para todos os usuários existentes na máquina, então faça a configuração em “Variáveis do sistema” na parte inferior da tela.

As configurações e passos realizados serão os mesmos, independentemente de serem configurados apenas para um usuário ou para todo o sistema. Em nosso exemplo seguiremos com a configuração das variáveis para todo o sistema. Em “Variáveis do sistema” clique no botão “Novo”

Nova Variável de Sistema

Nome da variável:

Valor da variável:

Procurar no Diretório... Procurar Arquivo... OK Cancelar

Na opção “**Nome da variável**” coloque “**JAVA_HOME**”, na opção “**Valor da variável**” coloque o caminho do diretório onde o JDK foi instalado, exemplo: “**C:\Program Files\Java\jdk-11.0.12**”

Nova Variável de Sistema

Nome da variável:

Valor da variável:

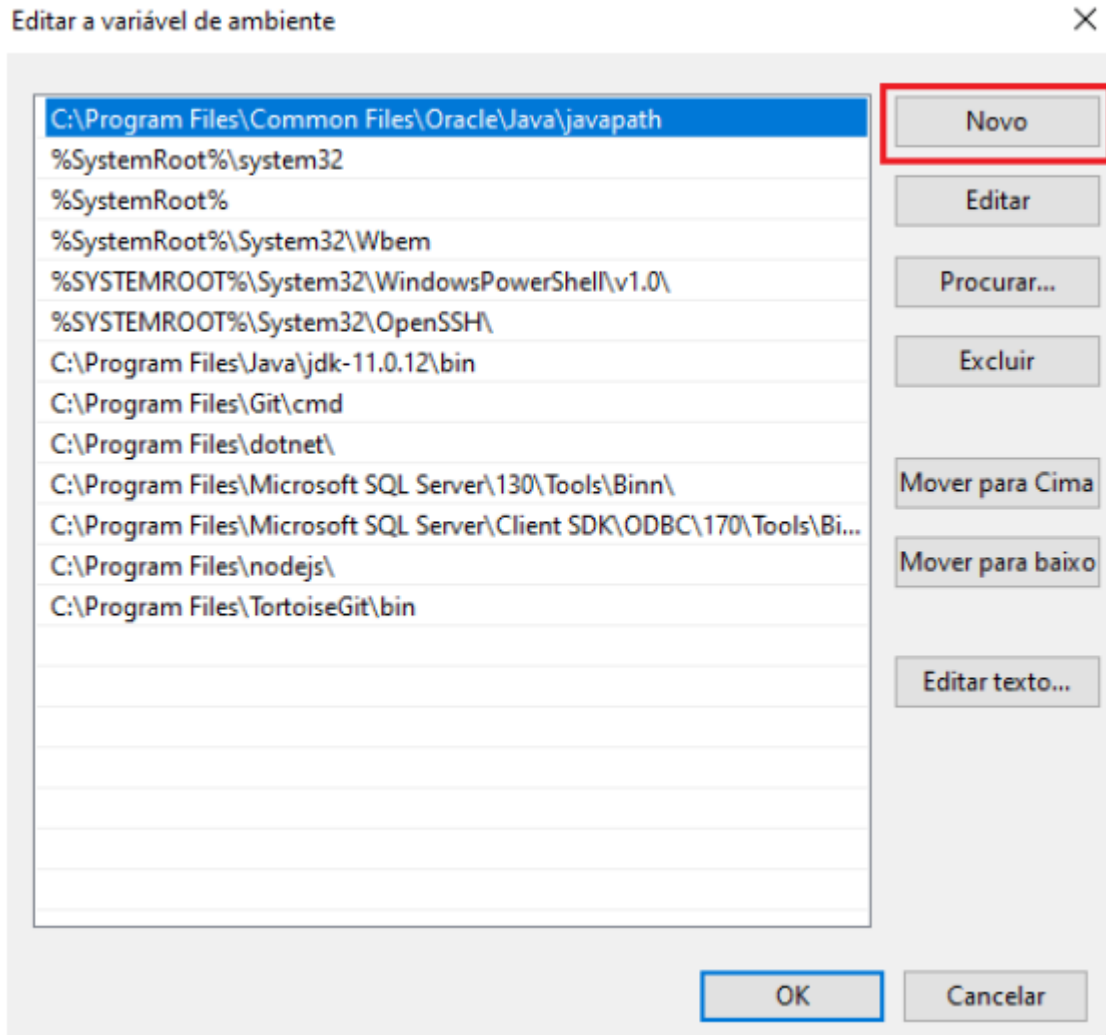
Procurar no Diretório... Procurar Arquivo... OK Cancelar

Após clicar em ok e adicionar as variáveis, volte a tela de variáveis do sistema, localize a variável “**Path**”, há selecione e clique no botão “**Editar**”, como mostra a imagem abaixo:

Variáveis do sistema

Variável	Valor
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Novo... Editar... Excluir



Clique na opção “**Novo**” e será permitido incluir um novo **Path**, inclua o caminho raiz da pasta bin onde o **JDK** foi instalado e clique em “**OK**”.

Realizados os passos a cima, o ambiente Java deverá estar pronto. Para testar iremos abrir o **CMD** e digitar o código “ **java -version** ”, se tudo correu bem a seguinte tela será apresentada:

```
C:\Users\Windows 10>java -version
java version "11.0.12" 2021-07-20 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.12+8-LTS-237)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.12+8-LTS-237, mixed mode)
```

Obs: O CMD deve ser aberto após o término das configurações, caso você o tenha aberto antes de finalizar as configurações, feche e abra-o novamente.

Criando o projeto maven:

Para criar um projeto de testes maven bem estruturado e padronizado, iremos seguir alguns passos que serão apresentados nesse projeto. É importante que os passos sejam seguidos corretamente e que sejam aplicados os padrões e boas práticas aqui apresentados.

Iremos criar o nosso projeto maven por linha de comando, isso pode ser feito através do terminal do próprio **VSCode** (IDE que iremos utilizar) ou através de algum outro terminal de sua preferência como o CMD ou PowerShell.

Com o terminal aberto iremos utilizar o seguinte código: ***mvn archetype:generate -DarchetypeArtifactId=maven-archetype-quickstart***

Ao executar o código a cima, o projeto começará a ser criado, mas antes de finalizar precisamos passar algumas informações para ele:

1º - groupId: exemplo de groupId: " **br.com.exemplo** "

2º - artifactId: exemplo de artifactId: "**automatizado**"

3º - versão: caso não tenha pré-definido, deixe a versão padrão "**1.0-SNAPSHOT**"

4º - Package: exemplo de package: "**automatizado**"

5º - Confirmação das informações passadas, se estiver tudo ok digite " Y " e pressione ENTER

Feito isso nosso projeto será criado com sucesso!