

## Variáveis de ambiente e instalação do JDK

### Transcrição

No capítulo anterior nosso script de backup estava funcionando, mas ainda tem um probleminha. Para poder chamar o script, devemos estar com o terminal na mesma pasta onde se encontra o script. Tem que ser possível executar o script a partir de qualquer pasta, não acha?

Vejamos como fazer para isso funcionar. Quando a gente executa algum comando no terminal, o Windows procura esse comando em lugares específicos. Pense que qualquer comando, como `dir` ou `cd`, também são pequenos scripts (essa é uma visão pouco simplificada, mas vale a analogia). Ou seja, o Windows precisa encontrar esses scripts, certo? E, para ajudá-lo, já existem alguns caminhos padrões que o Windows fica varrendo para achar esses pequenos scripts. Esse local padrão, onde se encontram os caminhos para os scripts é o famoso **PATH**.

### Acessando a variável PATH

Esses caminhos ficam salvos dentro de uma variável do sistema com o nome `PATH`. E podemos imprimir esses caminhos no prompt:

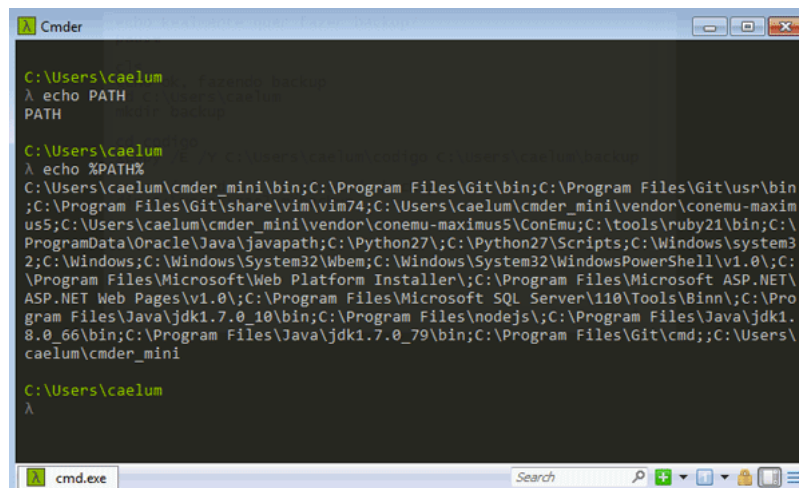
```
echo PATH
```

Opa! Só imprimiu `PATH`. Esqueci uma coisa: se quisermos ver o conteúdo da variável `PATH` devemos colocar um símbolo de `%` antes e depois, ficando assim:

```
echo %PATH%
```

Agora sim! Aparecem um monte de caminhos de diretórios onde o Windows procura executáveis e scripts. Repare que cada caminho está separado por um ponto e vírgula:

```
Path=C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\WINDOWS\sys
```



```
C:\Users\caelum
λ echo PATH
PATH

C:\Users\caelum
λ echo %PATH%
C:\Users\caelum\cmdr_mini\bin;C:\Program Files\Git\bin;C:\Program Files\Git\usr\bin;C:\Program Files\Git\share\vim\vim74;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5\ConEmu;C:\tools\ruby21\bin;C:\ProgramData\Oracle\Java\javapath;C:\Python27;C:\Python27\Scripts;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft\Web Platform Installer\;C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\;C:\Program Files\Java\jdk1.7.0_10\bin;C:\Program Files\nodejs\;C:\Program Files\Java\jdk1.8.0_66\bin;C:\Program Files\Java\jdk1.7.0_79\bin;C:\Program Files\Git\cmd;;C:\Users\caelum\cmdr_mini

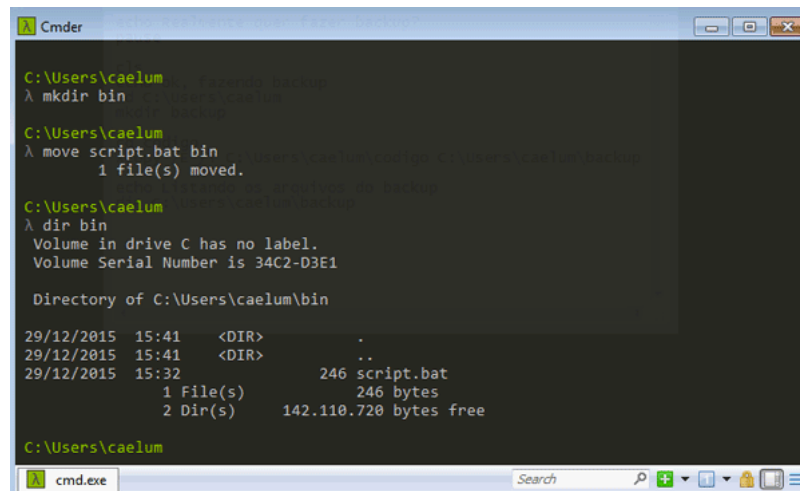
C:\Users\caelum
λ
```

## Alterando o PATH pelo prompt

A nossa tarefa é adicionar a pasta onde se encontra o nosso script de backup no PATH . Assim o Windows também vai poder achar o nosso script. Vou te dar mais uma dica, essa pasta com os executáveis/scripts normalmente é chamada de `bin` . Nem sempre é assim, mas a maioria das ferramentas seguem esse padrão.

Então vamos criar uma pasta `bin` e jogar o nosso script lá dentro. Digite `cd %HOME%` para voltar na pasta raiz do usuário e crie o diretório `bin` com o `mkdir` e em sequência movendo o nosso script para dentro da pasta `bin` com o comando `move` :

```
cd
mkdir bin
move script.bat bin
```



Ótimo!

Agora respire fundo, pois temos que alterar a variável `PATH` para adicionar o nosso novo diretório ( `C:\Users\caelum\bin` ). A ideia é que precisamos redefinir o `PATH` para também incluir o nosso diretório. Baseado no conteúdo do que exibimos da variável `PATH` , podemos ver que temos que adicioná-lo ao final, separando por ponto e vírgula do resto. Em outras palavras:

```
set PATH=%PATH%;C:\Users\caelum\bin
```

Alteramos a variável `PATH` com o comando `set` , baseado no `PATH` existente ( `%PATH%` ) adicionando o novo diretório ( `C:\Users\caelum\bin` ), usando o caractere separador `;` . Será que funcionou? Podemos verificar isso facilmente imprimindo o `PATH` na linha de comando:

```
echo %PATH%
```

```

C:\Users\caelum
λ set PATH=%PATH%;C:\Users\caelum\bin

C:\Users\caelum
λ echo %PATH%
C:\Users\caelum\cmdr_mini\bin;C:\Program Files\Git\bin;C:\Program Files\Git\usr\bin;C:\Program Files\Git\share\vim\vim74;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5\ConEmu;C:\tools\ruby21\bin;C:\ProgramData\Oracle\Java\javapath;C:\Python27\;C:\Python27\Scripts;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft\Web Platform Installer\;C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\;C:\Program Files\Java\jdk1.7.0_10\bin;C:\Program Files\nodejs\;C:\Program Files\Java\jdk1.8.0_66\bin;C:\Program Files\Java\jdk1.7.0_79\bin;C:\Program Files\Git\cmd;C:\Users\caelum\cmdr_mini;C:\Users\caelum\bin

C:\Users\caelum
λ |

```

Lá no final temos o nosso diretório, ou seja, na teoria, o Windows vai encontrar o nosso script. Façamos o teste:

script.bat

Podemos comemorar porque o Windows encontrou o nosso script!!

## Variáveis de ambiente

Vimos rapidamente o comando `set`, que serviu para alterar a variável `PATH`. Existem muitos mais variáveis além da `PATH`, e ainda deram um nome bonito para elas: **Variáveis de ambiente**.

O interessante é que o comando `set` também mostra TODAS essas variáveis! Como? Digite:

`set`

```

C:\Users\caelum
λ set
ALLUSERSPROFILE=C:\ProgramData
ANSICON=107x1000 (107x26)
ANSICON_DEF=7
APPDATA=C:\Users\caelum\AppData\Roaming
architecture=86
CLASSPATH=.;
CMDER_ROOT=C:\Users\caelum\cmdr_mini
CMDER_START=
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=NICO-PC
ComSpec=C:\Windows\system32\cmd.exe
ConEmuANSI=ON
ConEmuAnsiLog=
ConEmuArgs=/Icon "C:\Users\caelum\cmdr_mini\icons\cmdr.ico" /Title Cmder
ConEmuBackHWND=0x000403B4
ConEmuBaseDir=C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5\ConEmu
ConEmuBuild=150913
ConEmuConfig=
ConEmuDir=C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5
ConEmuDrawHWND=0x000403B2
ConEmuDrive=C:
ConEmuHooks=Enabled
ConEmuHWND=0x00090398
ConEmuPID=3964
ConEmuServerPID=5068

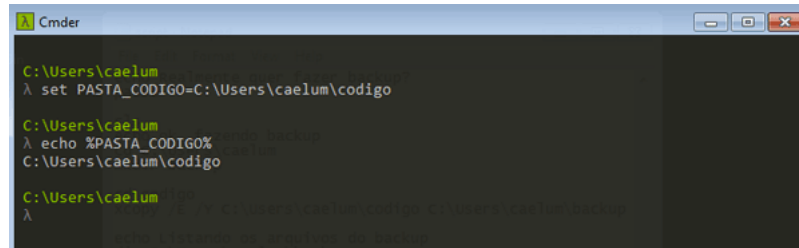
```

Repare que existem variáveis que guardam o nome do computador, do usuário, da arquitetura do computador e muito mais. Até podemos criar novas variáveis e veremos que algumas plataformas de desenvolvimento (por exemplo Java) exigem a criação de variáveis próprias.

Para criar uma variável própria, também utilizamos o comando `set`, fazendo assim:

```
// Criando a variável PASTA_CÓDIGO
set PASTA_CODIGO=C:\Users\caelum\codigo
```

Você pode testar digitando `set` ou `echo %PASTA_CODIGO%`



```
Cmdr
C:\Users\caelum
λ set PASTA_CODIGO=C:\Users\caelum\codigo
C:\Users\caelum
λ echo %PASTA_CODIGO%
C:\Users\caelum\codigo
C:\Users\caelum
λ
```

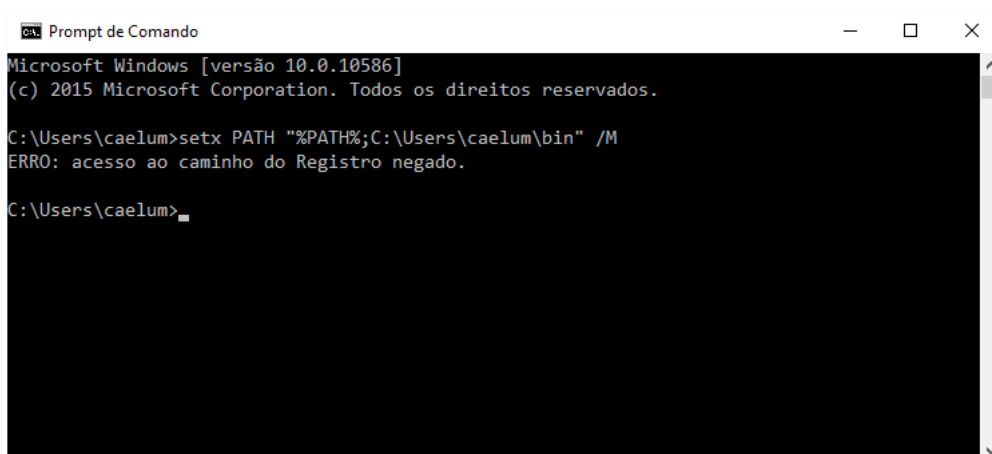
## Variáveis permanentes

Parece tudo ótimo, mas faça um teste: feche e reabra o terminal. Agora digite `echo %PASTA_CODIGO%`. Não funcionou! Para nossa surpresa, a variável que acabamos de criar parou de funcionar! O terminal esqueceu as variáveis :( Esqueceu até da nossa alteração da variável `PATH` !

Infelizmente as nossas variáveis só são lembradas enquanto o terminal está aberto. Mas queremos deixar essas configurações permanentes e existe um modo para isto. A solução é um comando bem parecido com o `set`, o `setx` que realmente salva o `PATH` de maneira permanente, porém como este comando altera as variáveis do sistema permanentemente, ele é especial e **precisa ser executado no prompt de comando tradicional do windows**.

O comando `setx` é praticamente igual ao `set`, só que não tem um `=` nem um `;` e separamos o caminho com aspas. Outro detalhe **muito importante** é adicionar o argumento `/M` no final do `setx`, que indica que queremos alterar uma variável do sistema.

```
setx PATH "%PATH%;C:\Users\caelum\bin" /M
```

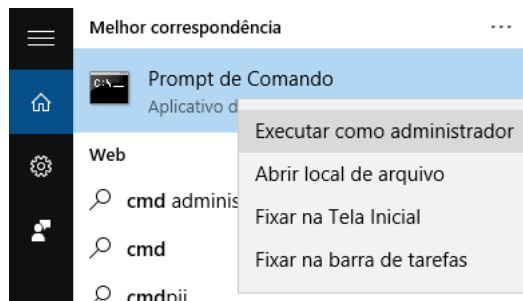


```
Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

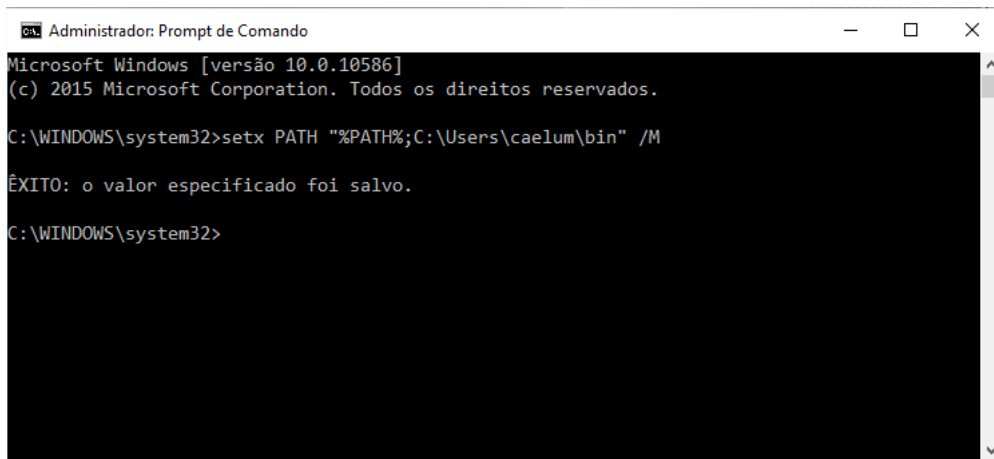
C:\Users\caelum>setx PATH "%PATH%;C:\Users\caelum\bin" /M
ERRO: acesso ao caminho do Registro negado.

C:\Users\caelum>
```

Obtivemos um erro! A alteração das variáveis do sistema é algo tão importante no sistema operacional, que o prompt de comando em modo normal não tem permissão para mexer nelas, nós precisamos do prompt em modo administrador. E para abrir o prompt no modo administrador é algo bem simples, basta clicarmos com o botão direito do mouse e ir em *Executar como administrador*.



Dá até pra ver no topo da janela que o prompt mudou para o modo Administrador, agora é só executar nosso comando setx:



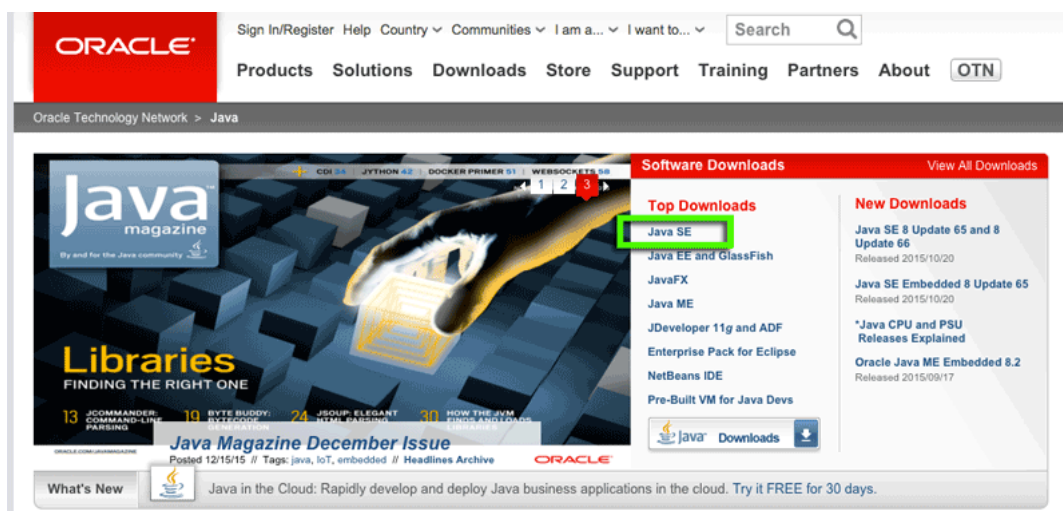
Agora sim nosso PATH foi alterado.

Existe também uma forma de alterar o PATH pela interface gráfica mas isso fica para os exercícios!

## Instalando nossa primeira linguagem

Agora que temos conhecimento de como manipular as variáveis de ambiente do Windows, instalar e configurar uma nova linguagem ou ferramenta é algo bem fácil! Vamos instalar o Java do zero para ver como é exemplo disso:

O primeiro passo é baixar o JDK pelo site, acesse: (<http://www.oracle.com/technetwork/java/>).



Dentro dos *Top Downloads*, escolha o Java SE. Na nova página clique no ícone Java (Java Platform (JDK))

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>  
[.http://www.oracle.com/technetwork/java/javase/downloads/index.html\)](http://www.oracle.com/technetwork/java/javase/downloads/index.html)



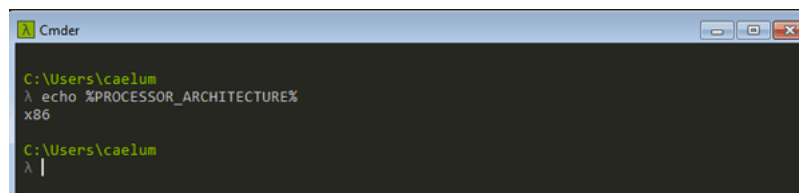
Agora temos que escolher o JDK adequado para a versão do nosso sistema operacional! A versão é a para Windows, claro, mas mesmo assim há duas opções: Um download para x86 e outro para 64bit. E agora?

Java SE Development Kit 8u65		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	77.69 MB	<a href="#">jdk-8u65-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM v8 Hard Float ABI	74.66 MB	<a href="#">jdk-8u65-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	154.67 MB	<a href="#">jdk-8u65-linux-i586.rpm</a>
Linux x86	174.84 MB	<a href="#">jdk-8u65-linux-i586.tar.gz</a>
Linux x64	152.69 MB	<a href="#">jdk-8u65-linux-x64.rpm</a>
Linux x64	172.86 MB	<a href="#">jdk-8u65-linux-x64.tar.gz</a>
Mac OS X x64	227.14 MB	<a href="#">jdk-8u65-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	139.71 MB	<a href="#">jdk-8u65-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	99.01 MB	<a href="#">jdk-8u65-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	140.22 MB	<a href="#">jdk-8u65-solaris-x64.tar.Z</a>
Solaris x64	96.74 MB	<a href="#">jdk-8u65-solaris-x64.tar.gz</a>
Windows x86	181.24 MB	<a href="#">jdk-8u65-windows-i586.exe</a>
Windows x64	186.57 MB	<a href="#">jdk-8u65-windows-x64.exe</a>

## Escolhendo o JDK para sua arquitetura do processador

Essas duas opções são relacionadas com a arquitetura do processador. Para descobrir qual é a opção correta podemos usar o nosso prompt! Abra o terminal e digite:

```
echo %PROCESSOR_ARCHITECTURE%
```

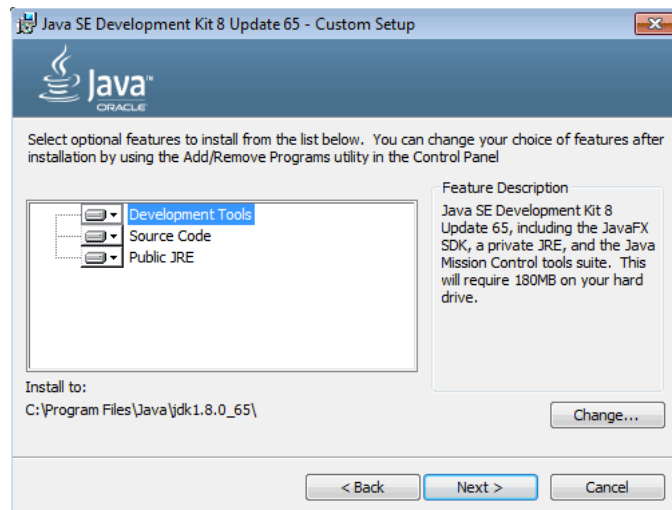


Agora deve aparecer **x86** ou **AMD64** (ou **IA64**). Tudo que tem **64** no final indica que o Windows é de 64 bits, já o **x86** indica que é de 32 bits. Beleza, já sabemos a arquitetura e qual download do Java escolher!

## Instalando o JDK

Escolha então o JDK (Java Development Kit) certo e, depois de ter aceitado a licença, podemos finalmente baixar o JDK. Baixe o arquivo `.exe`, que é algo como `java-8xxx-windows-xxx.exe`.

Execute o arquivo `java-8xxx-windows-xxx.exe` e passe pelo *Wizard* de instalação (pode instalar tudo, incluindo o source, não desmarque nada).



Repare que o diretório de instalação é `C:\Program Files\Java\jdk1.8.0_65`. Vamos lembrar disso!

O registro de sua VM (Virtual Machine) no site da Oracle é opcional.

## Testando Java no prompt e configuração do PATH

Chegou o momento de ver no prompt se o Java realmente foi instalado corretamente. Vamos abrir o `cmd` ou o `cmd` e testar, digite:

```
java -version
```

Deve aparecer:

```
java version "1.8.0_66"  
Java(TM) SE Runtime Environment (build 1.8.0_66-b18)  
Java HotSpot(TM) Client VM (build 25.66-b18, mixed mode, sharing)
```

Funcionou? Se funcionou pode ficar feliz pois o instalador já adicionou a pasta `bin` do Java no PATH!. Mas vamos testar também o compilador do Java:

```
javac -version
```

Não funcionou:

```
'javac' is not recognized as an internal or external command,  
operable program or batch file.
```



```

C:\Users\caelum
λ java -version
java version "1.8.0_66"
Java(TM) SE Runtime Environment (build 1.8.0_66-b18)
Java HotSpot(TM) Client VM (build 25.66-b18, mixed mode, sharing)

C:\Users\caelum
λ javac -version
'javac' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\caelum
λ

```

Hum, vamos verificar o PATH :

```
echo %PATH%
```

```

C:\Users\caelum
λ echo %PATH%
C:\Users\caelum\cmdr_mini\bin;C:\Program Files\Git\bin;C:\Program Files\Git\usr\bin;C:\Program Files\Git\share\vim\vim74;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5;C:\Users\caelum\cmdr_mini\vendor\conemu-maximus5\ConEmu;C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Microsoft\Web Platform Installer\;C:\Program Files\Microsoft ASP.NET\ASP.NET Web Pages\v1.0\;C:\Program Files\Microsoft SQL Server\110\Tools\Binn\;C:\Program Files\Git\cmd;C:\Users\caelum\cmdr_mini;

C:\Users\caelum
λ

```

Não encontramos nada sobre o nosso jdk1.8.0, mas tem um caminho suspeito:

```
C:\ProgramData\Oracle\Java\javapath
```

Vamos verificar esse caminho:

```
dir C:\ProgramData\Oracle\Java\javapath
```

```

16/12/2015  18:46    <DIR>          .
16/12/2015  18:46    <DIR>          ..
16/12/2015  18:46    <SYMLINK>      java.exe [C:\Program Files\Java\jre1.8.0_65\bin\java.exe]
16/12/2015  18:46    <SYMLINK>      javaw.exe [C:\Program Files\Java\jre1.8.0_65\bin\javaw.exe]
16/12/2015  18:46    <SYMLINK>      javaws.exe [C:\Program Files\Java\jre1.8.0_65\bin\javaws.exe]
               3 File(s)                0 bytes
               2 Dir(s)          541.597.696 bytes free

```



```

C:\Users\caelum
λ dir C:\ProgramData\Oracle\Java\javapath
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\ProgramData\Oracle\Java\javapath

17/12/2015  12:56    <DIR>          .
17/12/2015  12:56    <DIR>          ..
17/12/2015  12:56    <SYMLINK>       java.exe [C:\Program Files\Java\jre1.8
.0_66\bin\java.exe]
17/12/2015  12:56    <SYMLINK>       javaw.exe [C:\Program Files\Java\jre1.
8.0_66\bin\javaw.exe]
17/12/2015  12:56    <SYMLINK>       javaws.exe [C:\Program Files\Java\jre1
.8.0_66\bin\javaws.exe]
               3 File(s)              0 bytes
               2 Dir(s)            35.647.488 bytes free

C:\Users\caelum
λ

```

Repare que nessa pasta não tem nada do `javac`. Devemos adicionar ao `PATH` a pasta onde se encontra o `javac`. Nós já sabemos a pasta de instalação do JDK:

```

dir "C:\Program Files\Java\jdk1.8.0_65"
Directory of C:\Program Files\Java\jdk1.8.0_65

```

```

16/12/2015  18:47    <DIR>          .
16/12/2015  18:47    <DIR>          ..
16/12/2015  18:45    <DIR>          bin
06/10/2015  17:46             3.244 COPYRIGHT

```

\\ muito mais infos ...

```

C:\Users\caelum
λ dir "C:\Program Files\Java\jdk1.8.0_66\"
Volume in drive C has no label.
Volume Serial Number is 34C2-D3E1

Directory of C:\Program Files\Java\jdk1.8.0_66

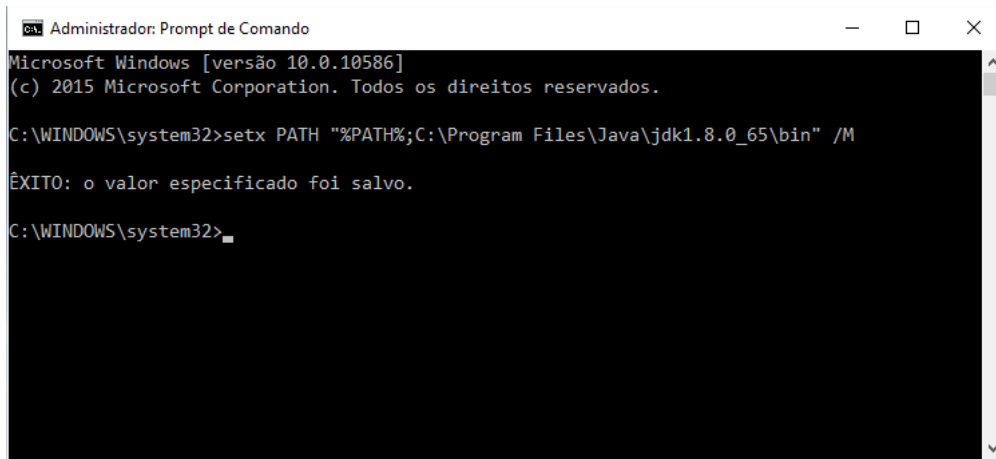
17/12/2015  12:57    <DIR>          .
17/12/2015  12:57    <DIR>          ..
17/12/2015  12:56    <DIR>          bin
09/11/2015  11:40             3.244 COPYRIGHT
17/12/2015  12:56    <DIR>          db
17/12/2015  12:56    <DIR>          include
17/12/2015  12:56             5.104.609 javafx-src.zip
17/12/2015  12:56    <DIR>          jre
17/12/2015  12:56    <DIR>          lib
17/12/2015  12:56             40 LICENSE
17/12/2015  12:56             159 README.html
17/12/2015  12:56             526 release
09/11/2015  11:41             21.243.586 src.zip
17/12/2015  12:56             110.114 THIRDPARTYLICENSEREADME - JAVAFOX.txt
17/12/2015  12:56             177.094 THIRDPARTYLICENSEREADME.txt
               8 File(s)            26.639.372 bytes
               7 Dir(s)            35.647.488 bytes free

```

Obs: Usamos aspas duplas pois tem um espaço no caminho.

Logo no início encontramos a pasta `bin`. Agora só falta adicionar ao `PATH`:

```
setx PATH "%PATH%;C:\Program Files\Java\jdk1.8.0_65\bin" /M
```



```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.10586]
(c) 2015 Microsoft Corporation. Todos os direitos reservados.

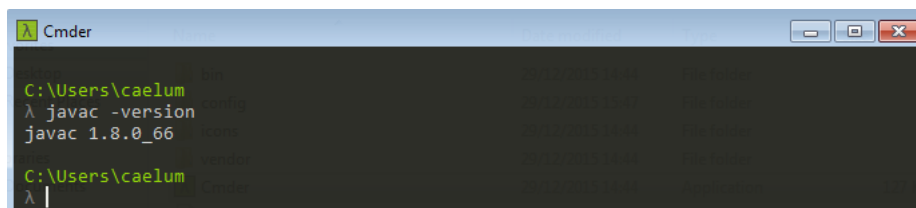
C:\WINDOWS\system32>setx PATH "%PATH%;C:\Program Files\Java\jdk1.8.0_65\bin" /M

ÊXITO: o valor especificado foi salvo.

C:\WINDOWS\system32>
```

Fechando e abrindo o terminal podemos testar novamente o comando:

```
javac -version
javac 1.8.0_65
```



```
Cmder
C:\Users\caelum>javac -version
javac 1.8.0_65
C:\Users\caelum>
```

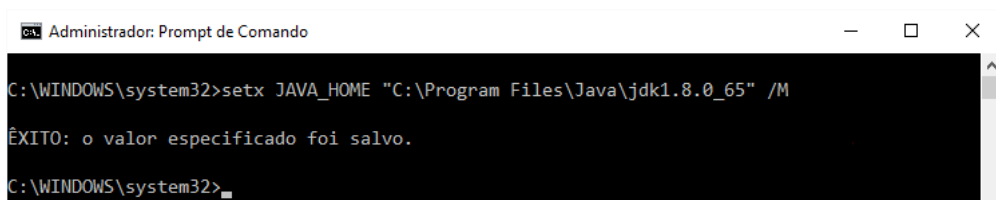
Problema resolvido!

## Criando o JAVA\_HOME

O Java já está funcionando na linha de comando, mas existem outras ferramentas do mundo Java que também querem saber onde o Java foi instalado. E essas ferramentas acreditam tanto que a gente instalou o Java que elas simplesmente assumem a existência de uma variável de ambiente com nome `JAVA_HOME`.

Ok, tudo bem, vamos criar essa variável usando o `setx`, já para deixarmos permanente. Só devemos ter um cuidado: o `JAVA_HOME` aponta para o diretório raiz do Java (sem a pasta `bin`):

```
setx JAVA_HOME "C:\Program Files\Java\jdk1.8.0_65" /M
```



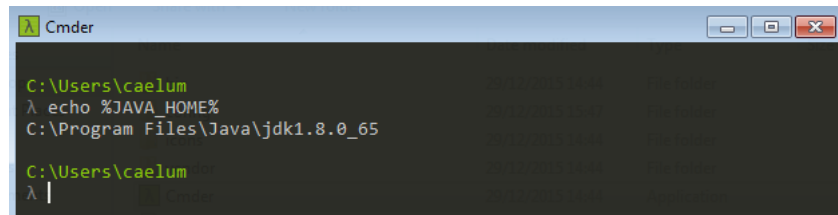
```
Administrador: Prompt de Comando
C:\WINDOWS\system32>setx JAVA_HOME "C:\Program Files\Java\jdk1.8.0_65" /M

ÊXITO: o valor especificado foi salvo.

C:\WINDOWS\system32>
```

Fechando e abrindo o terminal, podemos testar a variável:

```
echo %JAVA_HOME%
C:\Program Files\Java\jdk1.8.0_65
```



```
C:\Users\caelum
λ echo %JAVA_HOME%
C:\Program Files\Java\jdk1.8.0_65

C:\Users\caelum
λ |
```