

```
[INFO] -----  
[INFO] Total time: 39.264 s  
[INFO] Finished at: 2021-07-12T00:14:31-05:00  
[INFO] -----  
PS C:\proyectos> tree /f  
Listado de rutas de carpetas  
El número de serie del volumen es 7499-A68B  
C:..  
├── proyecto01  
│   ├── pom.xml  
│   ├── src  
│   │   ├── main  
│   │   │   ├── java  
│   │   │   │   └── App.java  
│   │   │   └── mx  
│   │   │       └── uv  
│   └── test  
│       ├── java  
│       │   └── AppTest.java  
│       └── mx  
│           └── uv  
PS C:\proyectos> |
```

CREACIÓN DE UN PROYECTO JAVA CON MAVEN EN WINDOWS TERMINAL

[Subtítulo del documento]

DESCRIPCIÓN BREVE

En este recurso se describe el procedimiento técnico para crear un proyecto Java en blanco a través de Maven desde la Terminal de Windows.

Rafael Rojano (РдЯ)
Sistemas Web

Contenido

Creación de un proyecto Java con Maven en Windows Terminal	2
Requerimientos:.....	2
La terminal de Windows	2
Prueba del ambiente de ejecución	3
Descripción de los parámetros de Maven.....	4

Creación de un proyecto Java con Maven en Windows Terminal

Requerimientos:

Para seguir este video requieres tener 3 herramientas instaladas, Windows terminal, java al menos la versión 8 y Apache Maven. Se adjuntan las ligas de referencia para cada herramienta, así como la liga para dar lectura a un tutorial de configuración de Java y Maven.

- Instalación de Windows Terminal

<https://www.microsoft.com/es-mx/p/windows-terminal/9n0dx20hk701?activetab=pivot:overviewtab>

- Instalación de Java JDK al menos la versión 8

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

- Instalación de Maven

<http://maven.apache.org/download.cgi#Installation>

- Un tutorial para la configuración puedes leerlo en:

https://dev.to/vanessa_corredor/instalar-manualmente-maven-en-windows-10-50pb

La terminal de Windows

La terminal de windows es una interfaz moderna de múltiples pestañas para aplicaciones de línea de comandos. Puedes apreciar cómo se visualiza en este video.

Para agregar una nueva pestaña basta con presionar el símbolo de más en la barra de título de la terminal.

La terminal también se puede personalizar con temas gráficos, véase Figura 1, y se le puede instalar desde la tienda de Microsoft.

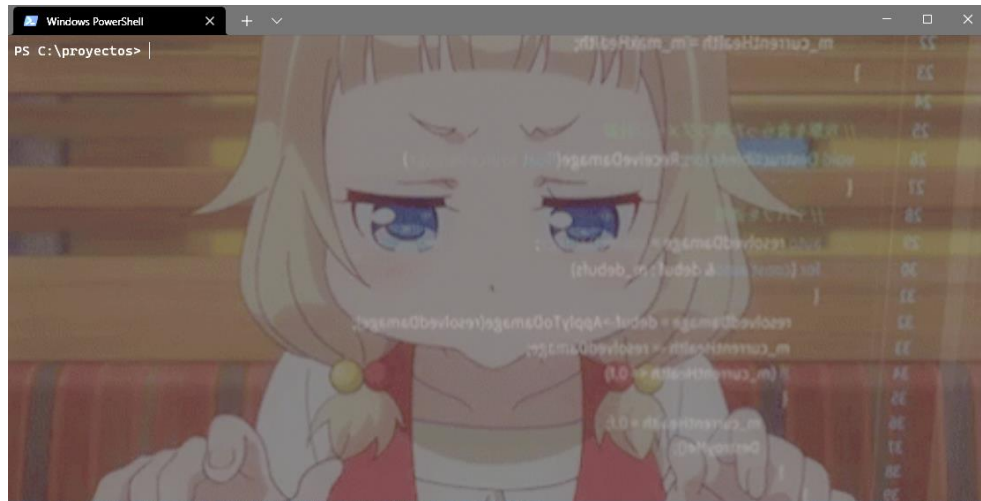


Figura 1. Vista de la terminal de windows.

Prueba del ambiente de ejecución

Una vez que tenemos el ambiente necesario instalado vamos a comprobarlo tecleando los siguientes comandos:

1. Desplegamos la versión de java, para ello teclearemos **java -version**

Como podemos observar en el video la versión instalada es la versión 8

2. Desplegamos la versión de maven, para ello teclearemos **mvn -version**

En la Figura 2, se aprecia que tenemos la versión 3.6.3, así mismo comprobamos que el **JAVA_HOME** que maven reconoce es el mismo que se observó al probar la versión de java.

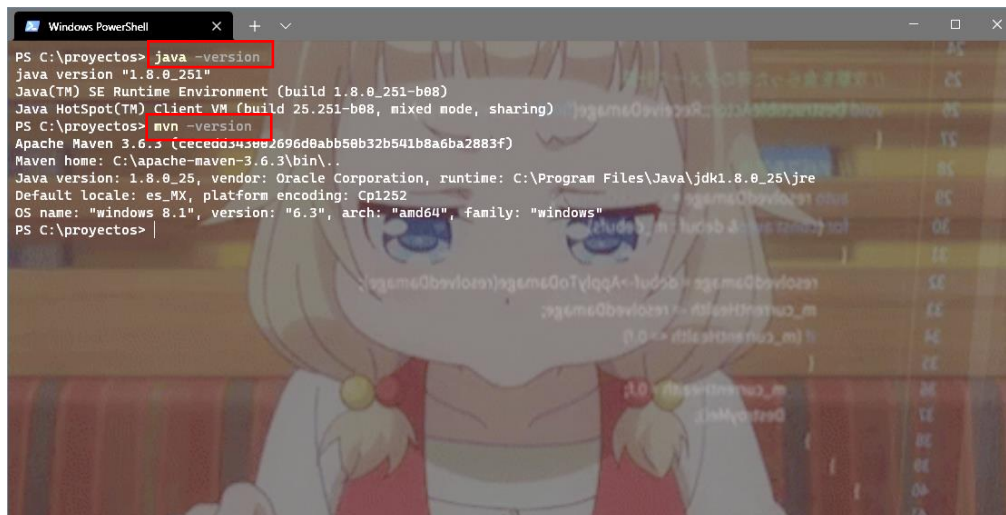


Figura 2. Comprobación del ambiente de ejecución.

Ahora bien, en esta terminal vamos a invocar a Maven, véase Figura 3, con ciertos parámetros para crear nuestro proyecto base, aunque también **tendré abierta la ventana del explorador** para ver gráficamente los resultados de teclear los comandos.

```

PS C:\proyectos> mvn archetype:generate `
>> "-DgroupId=mx.uv"
>> "-DartifactId=proyecto1"
>> "-DarchetypeArtifactId=maven-archetype-quickstart"
>> "-DarchetypeVersion=1.4"
>> "-DinteractiveMode=false"
[INFO] Scanning for projects...
[INFO] -----< org.apache.maven:standalone-pom >-----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----[ pom ]-----
[INFO] >>> maven-archetype-plugin:3.2.0:generate (default-cli) > generate-sources @ standalone-pom >>>
[INFO] <<< maven-archetype-plugin:3.2.0:generate (default-cli) < generate-sources @ standalone-pom <<<
[INFO] --- maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom ---

```

Figura 3. Terminal de Windows ejecutando Maven.

Descripción de los parámetros de Maven

mvn archetype:generate `	El comando a teclear es MVN con el primer parámetro en el formato PLUGIN:GOAL , En este caso archetype:generate . Esto significa que usaremos un patrón o plantilla base, es decir arquetipo y la meta será generarlo. Si bien todos los parámetros se pueden teclear en la misma línea en este ejemplo voy a separar cada parámetro en nuevas líneas. Para ello al final de las instrucciones dejaré un espacio en blanco y teclearé el acento grave .
"-DgroupId=mx.uv" `	El siguiente parámetro que podré entre comillas es "-DgroupId igual uv.mx" Este parámetro indica que vamos a definir una jerarquía de paquetes para las clases. Esto sirve para identificar de manera única el proyecto entre otros. Nota que el nombrado de paquetes se hace como el nombre de dominio en sentido inverso . Es decir, si la universidad tiene el dominio uv.mx el dominio inverso será mx.uv Puedes ver más información en la liga: https://maven.apache.org/guides/mini/guide-naming-conventions.html
"-DartifactId=proyecto1" `	El siguiente parámetro es "-DartifactId igual a proyecto1" este será el nombre del jar y también de la carpeta donde residirá el proyecto. Observa en la ventana del explorador como se ha creado la carpeta proyecto1.
"-DarchetypeArtifactId=maven-archetype-quickstart" `	El siguiente parámetro es "-DarchetypeArtifactId=maven-archetype-quickstart" maven tiene muchos patrones o

	<p>plantillas para crear proyectos, por ejemplo aplicaciones web, portlets, plugins, etc. En este caso usaremos el patrón maven-archetype-quickstart que es un proyecto maven sencillo.</p> <p>Puedes ver más información en la liga: https://maven.apache.org/guides/introduction/introduction-to-archetypes.html</p>
"-DarchetypeVersion=1.4" `	<p>El siguiente parámetro indica la versión del artefacto antes tecleado, te recomiendo ver la liga del repositorio remoto de maven donde se encuentra alojado: https://mvnrepository.com/artifact/org.apache.maven.archetypes/maven-archetype-quickstart</p>
"-DinteractiveMode=false"	<p>El ultimo parámetro es "-DinteractiveMode=false" que le indica a maven que no nos pregunte nada más, es decir que la sesión no sea interactiva.</p>

Ahora al dar enter, la primera vez que efectúes esta acción maven tardará un rato en crear los repositorios locales en tu computadora. En dichos repositorios copiará del repositorio central de maven los artefactos remotos que sean necesarios. Esto tardará dependiendo de tu proveedor de internet.

```
mvn archetype:generate `
"-DgroupId=mx.uv" `
"-DartifactId=proyecto1" `
"-DarchetypeArtifactId=maven-archetype-quickstart" `
"-DarchetypeVersion=1.4" `
"-DinteractiveMode=false"
```

Figura 4. Comandos a ingresar en la terminal.

Por cierto si te preguntas donde se crean estos repositorios locales lo podrás encontrar en la ruta `c:\users\nombre_del_usuario\.m2`, véase Figura 5.

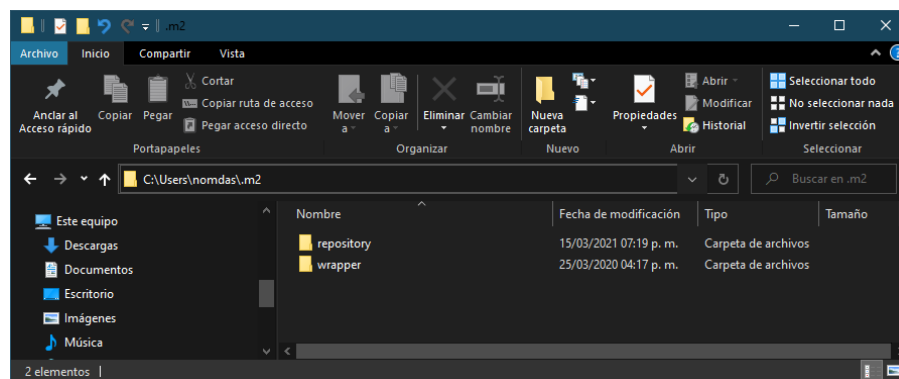


Figura 5. Ruta del repositorio de Maven local.

Es bueno saber que están allí, porque si en algún momento detectas un comportamiento raro al compilar tus programas, como última medida deberás borrar dicha carpeta. Al volver a invocar

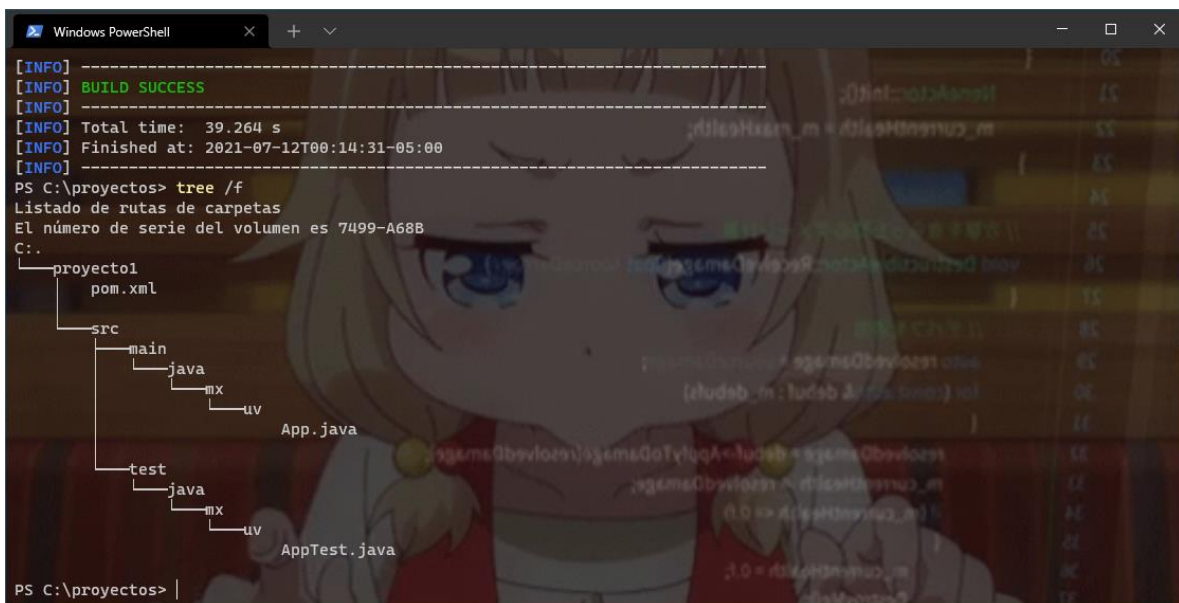
maven el repositorio se volverá a regenerar, aunque tomará algún tiempo como la primera vez que lo tecleaste.

De esta forma se ha creado nuestro proyecto básico maven, mismo que vamos a explorar tecleando, si en tu computadora está el **comando tree /f** que te indicará la jerarquía de directorios que se han creado.

En este caso observamos que el árbol de directorios creados es

- SRC que contiene dos carpetas main y test
- Main y Test a su vez tienen las carpetas de java, así como la estructura de directorios que se indicó en el parámetro groupId, es decir mx, uv
- Ahora explorando los archivos creados veremos tres, el primero es la base de maven, un archivo de configuración XML que se llama POM, por sus siglas de **Project Object Model**. Allí reside todo el chiste de maven.
- Los dos archivos generados que se llamarán App.java y AppTest.java, correspondientes a sus respectivos directorios son, el primero App.java la clase base de nuestro proyecto, el archivo AppTest.java es una clase que puede ser empleada para realizar pruebas unitarias con Junit.

De esta forma hemos concluido la configuración de un proyecto base con Maven, véase Figura 6.



```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 39.264 s
[INFO] Finished at: 2021-07-12T00:14:31-05:00
[INFO] -----
PS C:\proyectos> tree /f
Listado de rutas de carpetas
El número de serie del volumen es 7499-A68B
C:..
├── proyecto1
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   │   ├── mx
│       │   │   └── uv
│       │   └── App.java
│       └── test
│           ├── java
│           │   ├── mx
│           │   └── uv
│           └── AppTest.java
PS C:\proyectos> |
```

Figura 6. Proyecto Maven creado.