

INSTALACIÓN DE JDK JAVA

Es recomendable tener actualizado la Máquina Virtual Java de nuestro equipo para poder disfrutar de las mejoras que trae así como por cuestiones de seguridad, pero si eres desarrollador, también debes actualizar regularmente el JDK.

- El JDK

[Java Downloads | Oracle](#)

Java SE Development Kit 16.0.2

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.



The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

JDK 16 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

JDK 16.0.2 [checksum](#)

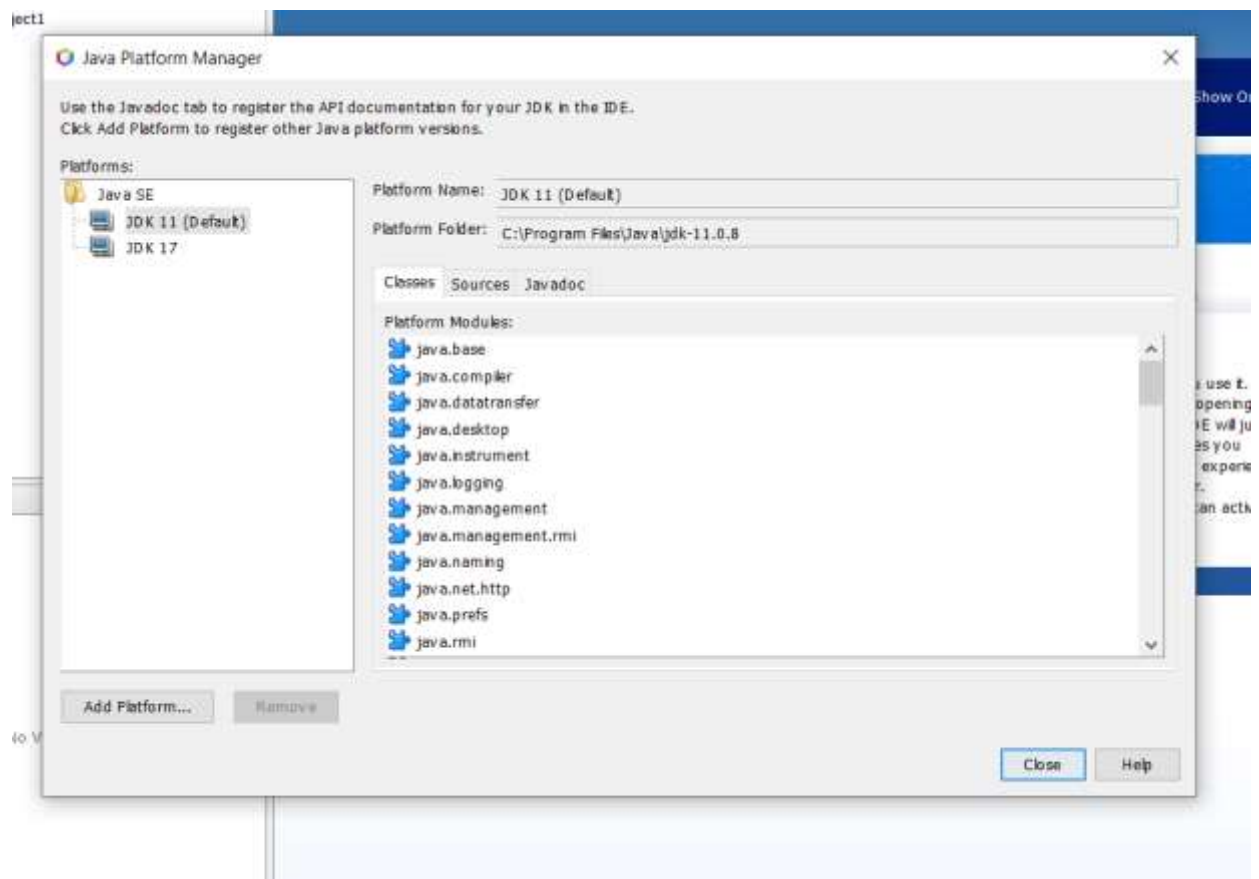
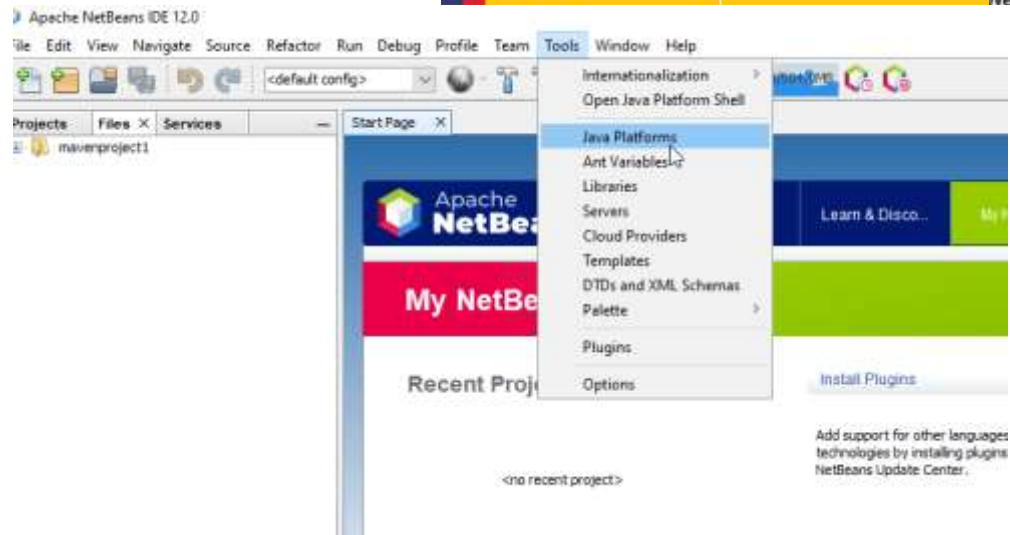
Documentation Download

Linux macOS **Windows**

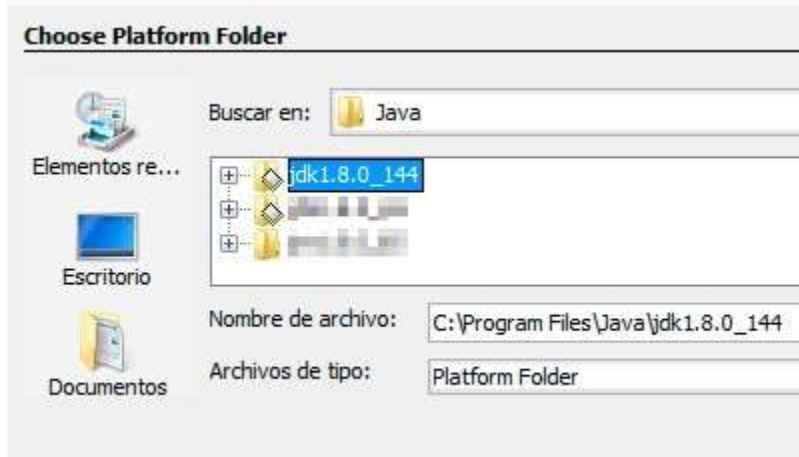
Product/file description	File size	Download
x64 Installer	150.58 MB	 jdk-16.0.2_windows-x64_bin.exe
x64 Compressed Archive	168.8 MB	 jdk-16.0.2_windows-x64_bin.zip

- Instalación en Netbeans

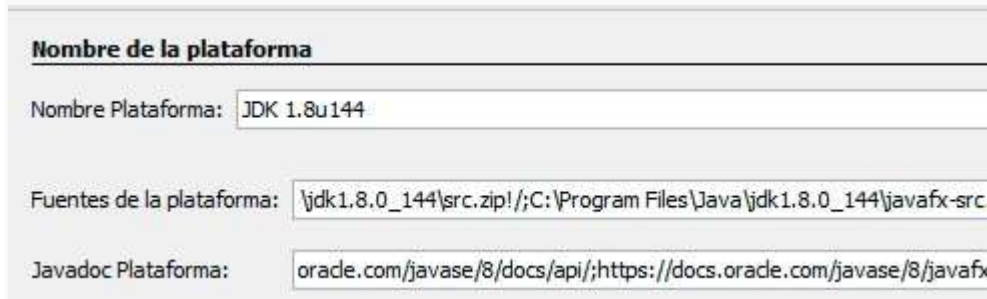
[Apache NetBeans Releases](#)



En esta ventana podrás ver las plataformas que tienes actualmente, presiona el botón **Add Platform...**, en tipo de plataforma deja marcada la opción «**Java Standard Edition**», presiona *siguiente*, selecciona la versión del JDK que deseas agregar y presiona *siguiente*



A continuación dala un nombre y presiona *Terminar*



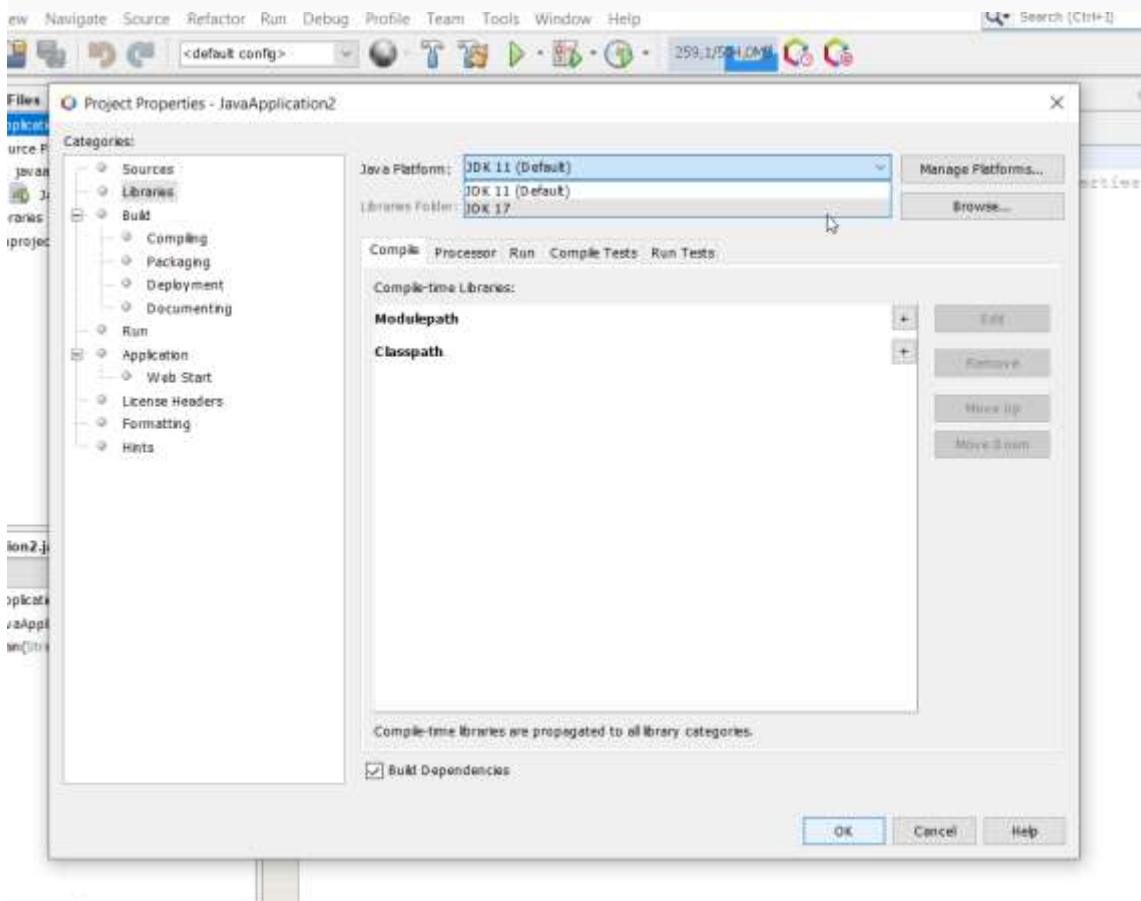
La nueva plataforma aparecerá en el Java Platform Manager



Cambiar JDK de un proyecto

Para cambiar el JDK de un proyecto, clic derecho sobre el proyecto y selecciona propiedades

(*properties*), a continuación en «**Categoría**» seleccionar «**Bibliotecas**», desplegando el combobox podremos seleccionar la plataforma que queremos usar y para realizar los cambios, presionamos *aceptar*.



Para usar el nuevo JDK tendríamos que realizar el paso anterior para cada proyecto con el que trabajemos, algo sencillo pero por ese contratiempo, podemos cambiar el JDK predeterminado de forma permanente.

Cambiar JDK predeterminado

Abre la ubicación donde tienes instalado Netbeans y busca la carpeta **etc** dentro de esa carpeta encontraras un archivo con el nombre de **netbeans.conf**, ábrelo con un editor de texto y busca **netbeans_jdkhome**, modifica el valor por la ruta del nuevo JDK, es decir **netbeans_jdkhome=»C:\Program Files\Java\jdk16"** y guarda los cambios. Si no te permite guardar los cambios, debes abrir el archivo como *administrador*. Reinicia Netbeans y todos los proyectos utilizaran ahora el JDK actualizado.

Configurar la variable de entorno **JAVA_HOME** para Java

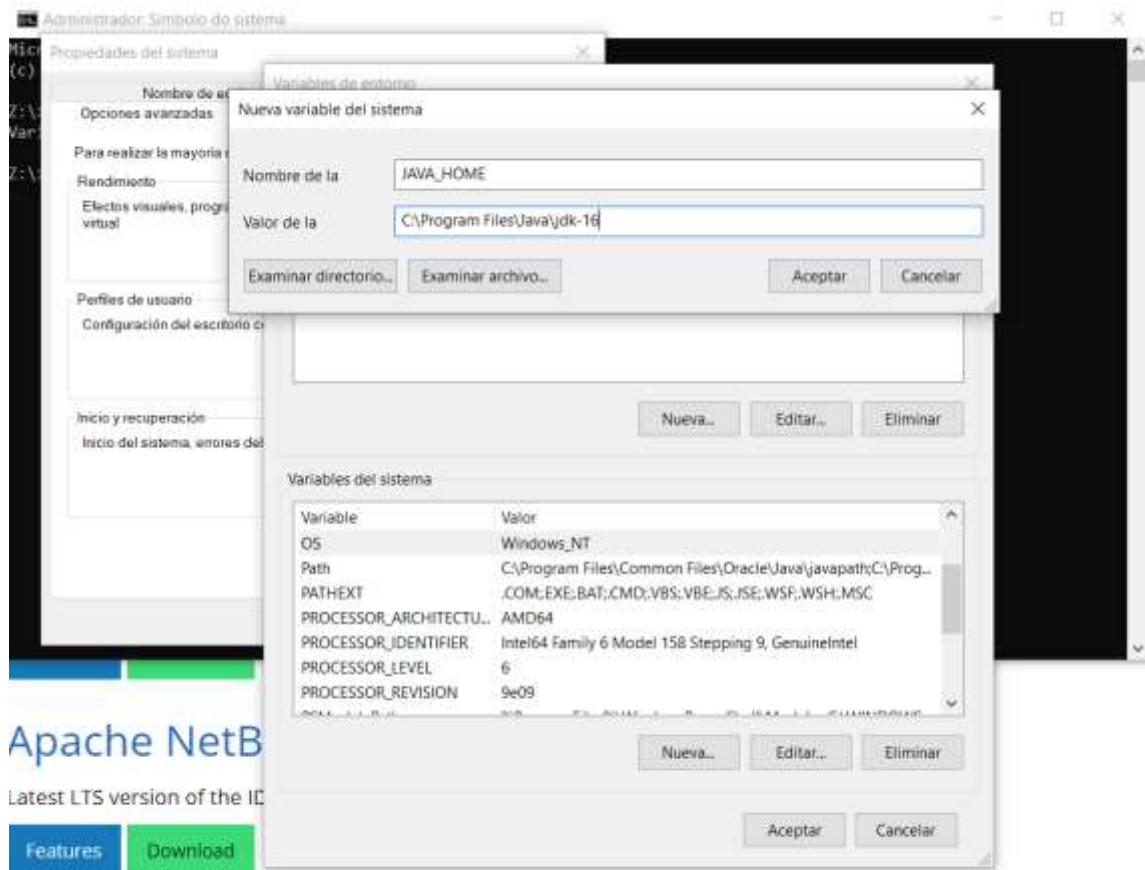
JAVA_HOME es una variable que se emplea para indicar la ruta donde se encuentra instalado el JDK (*Java Development Kit*).

Cómo configurar la variable **JAVA_HOME**

En Windows, para crear la variable **JAVA_HOME** se pueden seguir los siguientes pasos:

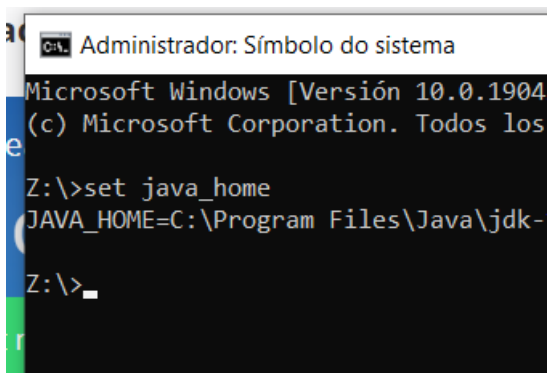
1. Hacer clic en el botón derecho de **"Equipo"**.
2. Seleccionar **"Propiedades"**.
3. Pinchar en **"Configuración avanzada del sistema"** > **"Variables de entorno"**.
4. Pulsar en el botón **"Nueva..."** (variable del sistema).
5. En el **"Nombre de la variable"** escribir **JAVA_HOME** y en el **"Valor de la variable"** teclear, por ejemplo (para la versión JDK 8u151):

C:\Program Files\Java\jdk1.8.0_151



Comprobación

Después de crear la variable **JAVA_HOME**, si se abre una nueva consola y se teclea **SET JAVA_HOME**, en la pantalla se podrá visualizar:



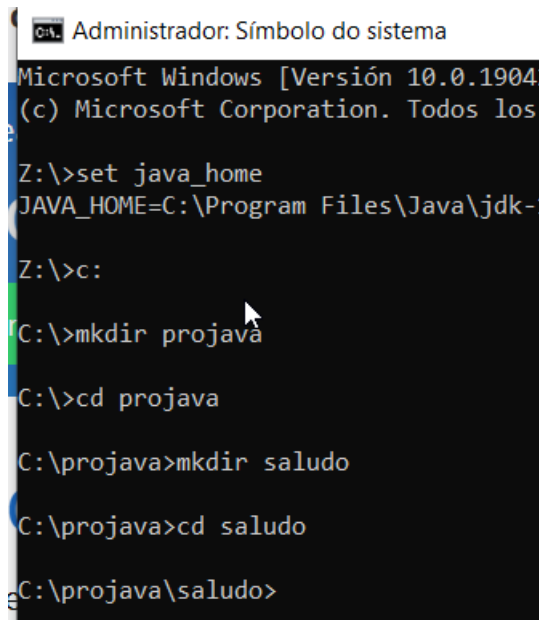
HolaMundo.java desde la consola (paso a paso)

EJEMPLO { HolaMundo }

Después de realizar la instalación del JDK en la computadora, para ejecutar desde la consola un primer programa que muestre por pantalla el mensaje "¡Hola mundo!", se pueden seguir los siguientes pasos:

Paso 1: Crear una carpeta para el programa

Por ejemplo en Windows, crear una carpeta llamada "**saludo**" dentro de otra llamada "**projava**" en la unidad **C**, tal y como se muestra en la imagen siguiente:



```
Administrador: Símbolo de sistema
Microsoft Windows [Versión 10.0.19041.1]
(c) Microsoft Corporation. Todos los derechos reservados.

Z:\>set java_home
JAVA_HOME=C:\Program Files\Java\jdk-11.0.10

Z:\>c:

C:\>mkdir projava

C:\>cd projava

C:\projava>mkdir saludo

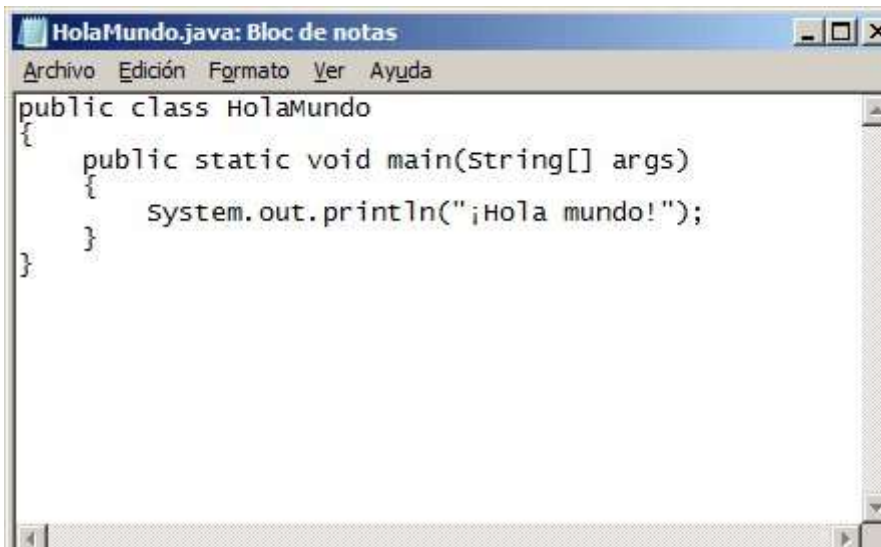
C:\projava>cd saludo

C:\projava\saludo>
```

Nota: si desconoce los comandos mkdir y cd del intérprete de comandos de Windows, en el Tutorial cmd se explica su uso.

Paso 2: Crear el archivo `HolaMundo.java` en la carpeta "saludo"

En Windows, el archivo **"HolaMundo.java"** se puede crear, por ejemplo, con el *Bloc de notas*:



```
public class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.println("¡Hola mundo!");
    }
}
```

A continuación, en la consola –con el comando `DIR`– se podrá ver el archivo **"HolaMundo.java"** creado.



```
C:\projava\saludo>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: B86F-72ED

Directorio de C:\projava\saludo

17/12/2017  19:54    <DIR>          .
17/12/2017  19:54    <DIR>          ..
17/12/2017  19:56             131 HolaMundo.java
                   1 archivos             131 bytes
                   2 dirs 382.011.977.728 bytes libres

C:\projava\saludo>
```

El archivo **"HolaMundo.java"** contiene el código fuente del programa.


Nota: en Java, el archivo (**.java**) que contiene a una clase, tiene que nombrarse igual –incluyendo minúsculas y mayúsculas– que dicha clase, en este caso **HolaMundo**. De no hacerse así, cuando se intente compilar el archivo, se producirá un error.

Paso 3: Compilar el archivo **HolaMundo.java**

Para ello, se puede teclear:

```
javac HolaMundo.java
```

En la pantalla se podrá visualizar:



```
C:\projava\saludo>javac holaMundo.java
C:\projava\saludo>DIR
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: B86F-72ED

Directorio de C:\projava\saludo
17/12/2017  19:57    <DIR>          .
17/12/2017  19:57    <DIR>          ..
17/12/2017  19:57                425 HolaMundo.class
17/12/2017  19:56                131 HolaMundo.java
                2 archivos             556 bytes
                2 dirs 382.010.441.728 bytes libres

C:\projava\saludo>
```

Obsérvese que, se ha creado el archivo **"HolaMundo.class"**. De no ser así –y suponiendo que el código fuente esté bien escrito, es decir, no contenga errores de sintaxis– una posible razón es que la [configuración de la variable de entorno PATH](#) sea incorrecta.

Por otra parte, hay que tener en cuenta que, dado que Java diferencia –respeta– las letras minúsculas de las mayúsculas, si por ejemplo se hubiese escrito **Class** en lugar de **class**, esto hubiese sido considerado como un error de sintaxis y, al intentar realizar la compilación, en la pantalla se hubiese visto lo siguiente:

```
C:\projava\saludo>javac HolaMundo.java
HolaMundo.java:1: error: class, interface, or enum expected
public Class HolaMundo
      ^
HolaMundo.java:3: error: class, interface, or enum expected
    public static void main<String[] args>
                  ^
HolaMundo.java:6: error: class, interface, or enum expected
    }
    ^
3 errors
C:\projava\saludo>
```

Nota: `javac` es el compilador de Java. De modo que, `javac` sirve para compilar el código fuente (`.java`) de los programas escritos en Java y generar los archivos (`.class`) correspondientes. Dichos archivos, con extensión (`.class`) se podrán ejecutar en la máquina virtual de Java (*Java Virtual Machine*, JVM).

Paso 4: Ejecutar el programa `HolaMundo` Para ello, se puede escribir:

```
java HolaMundo
```

El resultado será:

```
C:\projava\saludo>java HolaMundo
¡Hola mundo!
C:\projava\saludo>
```

Fíjese que, en la consola se ha mostrado el mensaje "**¡Hola mundo!**", sin comillas, que era el propósito de este primer programa escrito en Java.

Nota: `java` es el intérprete de Java; es un componente de la JVM capaz de interpretar los archivos (`.class`) en la computadora.

Explicación del código fuente del archivo `HolaMundo.java`

Respecto al código fuente del programa "`HolaMundo.java`", en la primera línea se ha definido la clase (`class`) llamada en este caso `HolaMundo`, indicando que es pública (`public`), es decir, visible –accesible– desde el exterior de dicha clase.

```
public class HolaMundo
{
    public static void main(String[] args)
    {
        System.out.println("¡Hola mundo!");
    }
}
```

Nota: todos los programas escritos en Java están compuestos por una o más clases.

Después –entre llaves `{}`– se ha escrito el contenido de la clase `HolaMundo` que, en este caso, contiene al método `main`, cuya signatura es:

```
public static void main(String[] args)
```

Todos los programas escritos en Java tienen que tener un método `main` declarado en alguna de sus clases, ya que, la ejecución de un programa desarrollado en Java siempre se iniciará en dicho método. El método `main` siempre se declara:

- Público (`public`): para que pueda ser llamado desde el exterior de la clase que lo contiene, es decir, desde fuera de la clase `HolaMundo`.
- Estático (`static`): indicando –de este modo– que se trata de un método de clase (en este caso asociado a la clase `HolaMundo`) y, por tanto, puede ser invocado aunque no exista ningún objeto

(instancia) de dicha clase. Más adelante estudiaremos los métodos de clase con más profundidad.

- De tipo vacío (**void**): ya que, este método nunca devolverá ningún valor. Más adelante veremos que los métodos pueden devolver distintos tipos de valores: números, caracteres, etc.

Nota: **class**, **public**, **static** y **void** son palabras clave reservadas.

Por otra parte, para que el programa pueda ejecutarse pasándole argumentos –aunque este no sea el caso– en la signatura de **main** se tiene que incluir como parámetro un array (arreglo) de cadenas (**String**) entre los paréntesis **()**. El nombre del array lo puede asignar el programador. Ahora bien, por convenio, suele llamarse **args**. Para indicar que se trata de un array se utilizan los corchetes **[]**.

```
(String[] args)
```

A continuación, también entre llaves **{}**, se ha escrito el contenido del método **main** con una única sentencia (instrucción) para mostrar en la consola la cadena de caracteres "**¡Hola mundo!**":

```
System.out.println("¡Hola mundo!");
```

En esta sentencia, se está invocando –llamando– al método **println** del objeto **System.out**, pasándole la cadena "**¡Hola mundo!**", para que la muestre en la pantalla. **System.out** se refiere a la salida (*output*) estándar, es decir, la pantalla del ordenador.

Véase que, al final de la sentencia se ha escrito un punto y coma (**;**). Esto es obligatorio, ya que, en Java todas las sentencias tienen que terminar con un punto y coma.

NETBEANS Y ALTERNATIVAS

NetBeans

Description

Un entorno de desarrollo integrado de código abierto y gratuito para desarrolladores de software. Obtiene todas las herramientas que necesita para crear aplicaciones profesionales de escritorio, empresariales, web y móviles con el lenguaje Java, C / C ++ e incluso lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby. NetBeans IDE es fácil de instalar y usar de inmediato y se ejecuta en muchas plataformas, como Windows, Linux, Mac OS X y Solaris.

La sección de descarga ofrece paquetes / instaladores preempaquetados para cada lenguaje de programación compatible que contiene solo los complementos necesarios

Alternativas



El código combina la interfaz de usuario optimizada de un editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo.

Posibilidad de añadir las extensiones necesarias para poder utilizarlo con diversos lenguajes de programación. Además, es multiplataforma.



Eclipse es una plataforma de desarrollo extensible con tiempos de ejecución y marcos de aplicaciones para crear, implementar y administrar software en todo el ciclo de vida del software. Mucha gente conoce Eclipse como un IDE de Java, pero es mucho más que eso. Eclipse consiste en más de 60 proyectos de código abierto diferentes, organizados en 6 categorías diferentes:

- *. Desarrollo Empresarial
- *. Desarrollo de dispositivos y dispositivos integrados
- *. Rich Client Platform (Aplicaciones de Internet enriquecidas)
- *. Marcos de aplicaciones
- *. Application Lifecycle Management (ALM)
- *. Arquitectura Orientada a Servicios (SOA)



Description

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) de Microsoft. Se utiliza para desarrollar programas informáticos para Microsoft Windows, así como sitios web, aplicaciones web y servicios web. Visual Studio utiliza plataformas de desarrollo de software de Microsoft, como Windows API, Windows Forms, Windows Presentation Foundation, Windows Store y Microsoft Silverlight. Puede producir tanto código nativo como código administrado.

Visual Studio incluye un editor de código que admite IntelliSense así como también refactorización de código. El depurador integrado funciona tanto como un depurador de nivel de origen como un depurador de nivel

de máquina. Otras herramientas integradas incluyen un diseñador de formularios para crear aplicaciones GUI, diseñador web, diseñador de clases y diseñador de esquemas de bases de datos. Acepta complementos que mejoran la funcionalidad en casi todos los niveles, incluida la compatibilidad con sistemas de control de código fuente (como Subversion) y el agregado de nuevos conjuntos de herramientas como editores y diseñadores visuales para lenguajes específicos del dominio o conjuntos de herramientas para otros aspectos del ciclo de vida del desarrollo de software (como el cliente de Team Foundation Server: Team Explorer)

Visual Studio admite diferentes lenguajes de programación y permite que el editor de códigos y el depurador admitan (en diversos grados) casi cualquier programación Idioma, siempre que exista un servicio específico de idioma. Los lenguajes incorporados incluyen C, C ++ y C ++ / CLI (a través de Visual C ++), VB.NET (a través de Visual Basic .NET), C # (a través de Visual C #) y F # (a partir de Visual Studio 2010). El soporte para otros idiomas, como M, Python y Ruby, entre otros, está disponible a través de los servicios de idiomas que se instalan por separado. También es compatible con XML / XSLT, HTML / XHTML, JavaScript y CSS.

VS está orientado a C, C ++ y C # no a Java



IntelliJ IDEA es un IDE de Java gratuito / comercial de JetBrains. Su diseño se centra en la productividad del programador. Algunas de sus características están destinadas a acelerar el desarrollo y permitir a los programadores concentrarse en la funcionalidad, mientras que IntelliJ IDEA maneja las tareas de codificación de rutina.

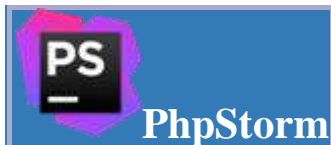
IntelliJ IDEA está disponible en dos ediciones: Ultimate (pagado) y Community (gratis y de código abierto).

IDEA es compatible con varios lenguajes de programación con Funciones avanzadas de inspección y refactorización de código, como Groovy, Javascript, Actionscript, Ruby / JRuby, PHP y otros. La Edición de la Comunidad e

s de Código Abierto, pero pierde la mayoría de las funciones para el servidor y el desarrollo web.



Python y Django IDE con refactorizaciones, finalización de código, análisis de código sobre la marcha y otras características, incluida la integración de control de versiones, implementación, integración con rastreadores de problemas y depuración remota. También hay una versión...



Smart PHP Code Editor El editor que en realidad "obtiene" su código y entiende profundamente su estructura, es compatible con todas las funciones del lenguaje PHP para proyectos modernos y heredados. Proporciona la mejor terminación de código, refactorizaciones..



Code :: Blocks es un IDE de C ++ gratuito creado para satisfacer las necesidades más exigentes de sus usuarios. Está diseñado para ser muy extensible y totalmente configurable. Un IDE con todas las funciones que necesita un desarrollador, que tiene una apariencia simple.



Komodo Edit es un editor de código rápido, inteligente, gratuito y de código abierto. Cambiar tu confiable editor de código es difícil, pero prueba Komodo Edit (o su hermano mayor Komodo IDE): valdrá la pena. ¿Windows, Mac, Linux? Sí. PHP, Python, Ruby, JavaScript, Perl,...



Qt Creator es un entorno de desarrollo integrado C ++, JavaScript y QML multiplataforma que forma parte del SDK para Qt Framework. Incluye un depurador visual y un diseño y formularios de GUI integrados diseñador. Las características del editor incluyen resaltado de sintaxis...



Aptana Studio es un entorno de desarrollo web completo que combina poderosas herramientas de creación con una colección de servicios de alojamiento y colaboración en línea que lo ayudan a usted y a su equipo a hacer más. Incluye soporte para PHP, CSS, FTP y más ...



JetBrains WebStorm es un IDE comercial para JavaScript, CSS & HTML creado en la plataforma IntelliJ IDEA de JetBrains. WebStorm proporciona la finalización automática de código, el análisis de código sobre la marcha, el soporte de refactorización y la integración de VCS.



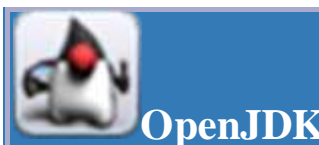
UltraEdit es el editor ideal de texto, HTML y HEX, y un editor avanzado de PHP, Perl, Java y JavaScript para programadores. Un ganador del premio de la industria, UltraEdit admite el manejo de archivos de 64 bits basado en disco (estándar) en plataformas Windows de 32 bits (Wind...



Xcode es el principal entorno de desarrollo de Apple para Mac OS X. Además de estar incluido en el DVD con cada copia de Mac OS X, la última versión para desarrolladores de Xcode siempre está disponible de forma gratuita para los miembros de ADC e incluye todas las herramientas ...



KDevelop es un IDE (Entorno de Desarrollo Integrado) de código abierto y gratuito para Linux, Windows (beta) y macOS (experimental). Se trata de un IDE de complemento completo para C / C ++, Python, JavaScript. y PHP. Se basa en KDevPlatform y en las bibliotecas KDE y Qt.



OpenJDK es la versión gratuita de la plataforma de desarrollo Java. El proyecto OpenJDK consta de varios componentes. Principalmente estos son HotSpot (la máquina virtual), la biblioteca de clases de Java y

el compilador javac de Java. El complemento del navegador web y Web

...



El IDE C / C ++ multiplataforma inteligente. CLion es un nuevo IDE que lo ayuda a ser más productivo mientras desarrolla aplicaciones con C y C ++ para Windows, Linux y OS X. Aproveche la experiencia comprobada de JetBrains en calidad de código, navegación de código eficiente, f...



Alternativas a Bloodshed Dev-C ++ para Windows, Linux, Mac, Android, BSD y más. Filtrar por licencia para descubrir solo alternativas gratuitas o de código abierto. Esta lista contiene un total de más de 25 aplicaciones similares a Bloodshed Dev-C ++. Bloodshed Dev-C ++ es un ...



Codifique de forma más inteligente y rápida con Komodo IDE, el entorno de desarrollo profesional multiplataforma para lenguajes dinámicos y tecnologías abiertas. El galardonado conjunto de características de Komodo incluye una funcionalidad de editor estándar como formato de cód...



Herramientas de desarrollo rápido de aplicaciones para aplicaciones nativas de Android, iOS y escritorio. La forma sencilla de programar



rápidamente aplicaciones del mundo real. Lenguaje de programación similar a Visual Basic.



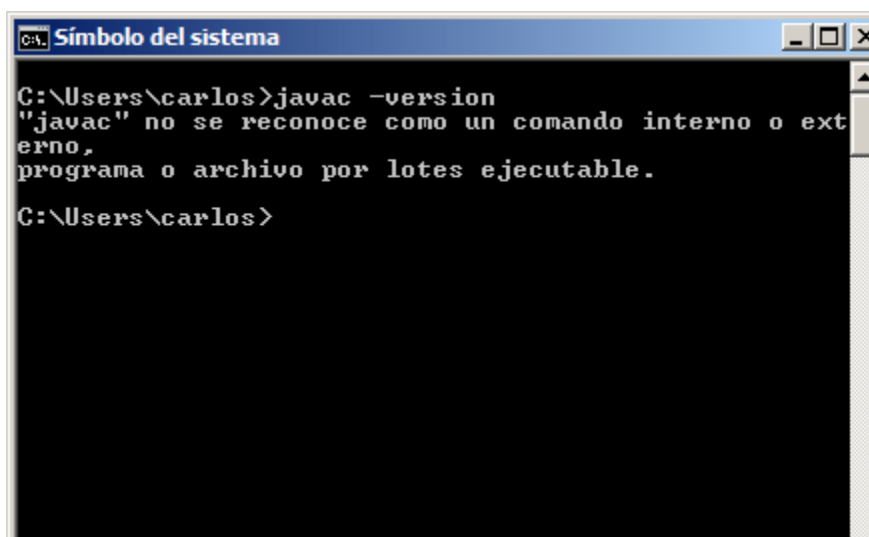
Codiad es un código abierto, basado en web, IDE en la nube y editor de código con una huella y requisitos mínimos

Discontinuado "Dada su antigüedad y la cantidad de alternativas viables ahora disponibles, Codiad ya no se encuentra bajo mantenimiento activo por parte de los colaboradores principales".

ANEXOS (NO ES MATERIAL EVALUABLE DEL MÓDULO)

Variable Entorno PATH

En la variable **PATH** se pueden indicar las rutas (carpetas o directorios) de búsqueda de archivos ejecutables en el intérprete de comandos. Por tanto, si todavía no se ha configurado la variable **PATH** y se intenta ejecutar el compilador de Java (**javac**) desde una carpeta distinta a donde está ubicado –y sin especificar dicha ubicación– en la pantalla se mostrará un mensaje haciendo saber que **javac** no se reconoce. Por ejemplo en Windows se podrá ver:

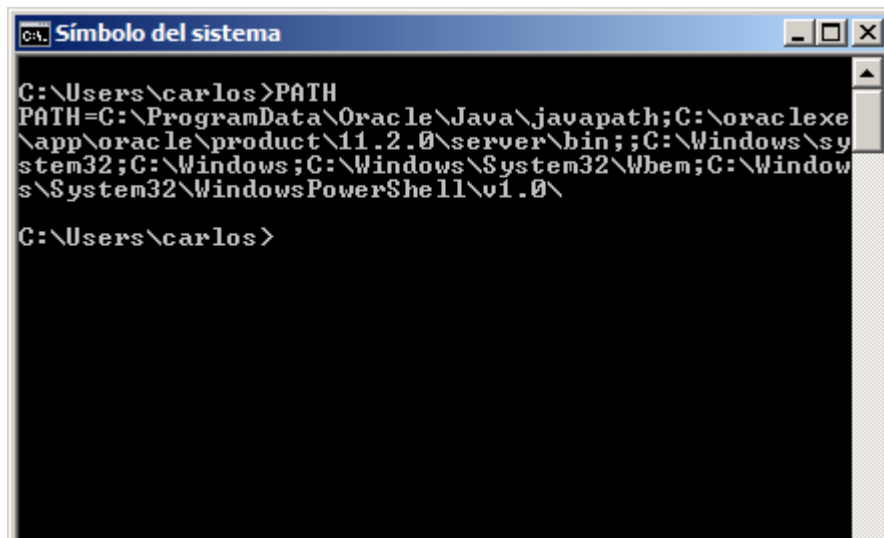


```
C:\Users\carlos>javac -version
"javac" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.
C:\Users\carlos>
```

Según la versión que se haya instalado, la ruta de la carpeta donde está ubicado el compilador de Java será diferente. Por ejemplo, al instalar la versión JDK 8u151 en Windows, la ruta sería:

```
C:\Program Files\Java\jdk1.8.0_151\bin
```

Para ver el contenido de la variable **PATH**, se puede ejecutar el comando también llamado **PATH**:



```
C:\Users\carlos>PATH
PATH=C:\ProgramData\Oracle\Java\javapath;C:\oraclexe\app\oracle\product\11.2.0\server\bin;;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Users\carlos>
```

En este caso, como era de esperar, se puede ver que la variable **PATH** no contiene la ruta donde está el compilador de Java.

Cómo configurar la variable **PATH**

En Windows, para que **javac.exe** sea reconocido, se pueden seguir los siguientes pasos:

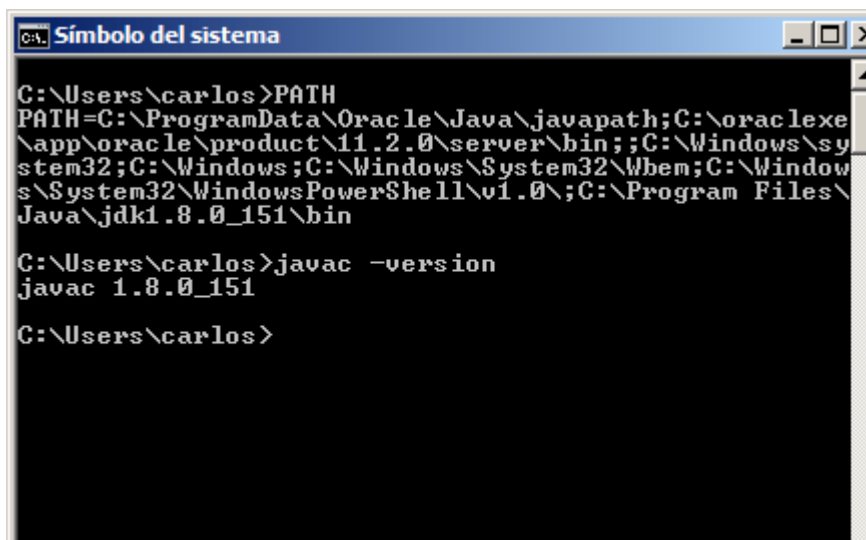
1. Hacer clic en el botón derecho de **"Equipo"**.
2. Seleccionar **"Propiedades"**.
3. Pinchar en **"Configuración avanzada del sistema"** > **"Variables de entorno"**.
4. Seleccionar la variable **PATH**.
5. Pulsar en el botón **"Editar"**.

6. En el "**Valor de la variable**" –sin borrar lo que ya hay– añadir al final un punto y coma ";" seguido de la ruta donde se encuentre el archivo **javac.exe**. Por ejemplo:

```
;C:\Program Files\Java\jdk1.8.0_151\bin
```

Comprobación

Después de realizar los pasos anteriores, al abrir una nueva consola y teclear otra vez **PATH** y **javac -version**, se podrán ver los cambios realizados:



```
C:\Users\carlos>PATH
PATH=C:\ProgramData\Oracle\Java\javapath;C:\oraclexe\app\oracle\product\11.2.0\server\bin;;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\Java\jdk1.8.0_151\bin

C:\Users\carlos>javac -version
javac 1.8.0_151

C:\Users\carlos>
```

Ahora, el programa **javac.exe** sí es reconocido, ya que, se encuentra en la ruta añadida a la variable **PATH**. Por consiguiente, el compilador de Java se podrá utilizar desde cualquier carpeta –sin necesidad de especificar la ruta de su ubicación– para compilar archivos que contengan código fuente (**.java**) y generar archivos ejecutables (**.class**).

102

En este **tutorial de introducción a CMD** se explica cómo utilizar el intérprete de comandos de *Windows*. Se incluyen múltiples ejemplos y ejercicios resueltos.

Comandos CMD (MS-DOS) en Windows

629

Los comandos mostrados en la tabla siguiente pueden ser ejecutados en el **intérprete de comandos de Windows**. De cada uno de ellos se indica su *funcionalidad*. Ahora bien, hay que tener en cuenta que no todos los comandos se pueden ejecutar en todas las versiones de *Microsoft Windows* (*Windows XP*, *Windows 7*, *Windows 10*, *Windows Server 2008*, *Windows Server 2012*...).

Lista de comandos CMD en Windows (MS-DOS)

Comando	Funcionalidad
ASSOC *	Muestra o modifica las asociaciones de las extensiones de archivos.
AT	Planifica comandos y programas para ejecutarse en un equipo. Está en desuso, en su lugar se recomienda utilizar schtask.exe .
ATTRIB	Muestra o cambia los atributos del archivo.
BREAK *	Establece o elimina la comprobación extendida de <i>Ctrl+C</i> en los sistemas <i>MS-DOS</i> . Este comando ya no está en uso. Se incluye para preservar la compatibilidad con archivos de <i>MS-DOS</i> ya existentes, pero no tiene ningún efecto en la línea de comandos porque la funcionalidad es automática.

Lista de comandos CMD en Windows (MS-DOS)

Comando	Funcionalidad
CALCS	Muestra o modifica las listas de control de acceso (ACLs) de archivos.
CALL *	Llama a un programa por lotes desde otro.
CD *	Muestra el nombre del directorio actual o cambia a otro directorio.
CHCP	Muestra o establece el número de página de códigos activa.
CHDIR *	Muestra el nombre del directorio actual o cambia a otro directorio.
CHKDSK	Comprueba un disco y muestra un informe de su estado.
CHKNTFS	Muestra o modifica la comprobación de disco al iniciar (arrancar).
CLS *	Borra la pantalla.
CMD	Inicia una nueva instancia del intérprete de comandos de Windows.
COLOR *	Establece los colores del primer plano y fondo predeterminados de la consola.
COMP	Compara el contenido de dos archivos o un conjunto de archivos.

Lista de comandos CMD en Windows (MS-DOS)

<i>Comando</i>	<i>Funcionalidad</i>
COMPACT	Muestra o cambia el estado de compresión de archivos en particiones NTFS.
CONVERT	Convierte volúmenes FAT a volúmenes NTFS. No puede convertir la unidad actual.
COPY *	Copia uno o más archivos a otro lugar (en otra ubicación).
DATE *	Muestra o establece la fecha.
DEL *	Elimina uno o más archivos.
DIR *	Muestra una lista de archivos y subdirectorios en un directorio.
DISKPART	Muestra o configura las propiedades de partición de disco.
DISKCOMP	Compara el contenido de dos disquetes.
DOSKEY	Edita líneas de comando, recupera comandos de Windows y crea macros.
DRIVERQUERY	Muestra el estado y las propiedades actuales del controlador de dispositivo.
ECHO *	Muestra mensajes, o activa y desactiva el eco.
ENDLOCAL *	Termina la búsqueda de variables de entorno del archivo por lotes.

Lista de comandos CMD en Windows (MS-DOS)

<i>Comando</i>	<i>Funcionalidad</i>
ERASE *	Elimina uno o más archivos.
EXIT *	Sale del programa cmd.exe (intérprete de comandos).
FC	Compara dos archivos o conjunto de archivos y muestra las diferencias entre ellos.
FIND	Busca una cadena de texto en uno o más archivos.
FINDSTR	Busca cadenas de texto en archivos.
FOR *	Ejecuta un comando para cada archivo en un conjunto de archivos.
FORMAT	Formatea un disco para usarse con Windows.
FSUTIL	Muestra o configura las propiedades del sistema de archivos.
FTYPE *	Muestra o modifica los tipos de archivo usados en asociaciones de extensión de archivo.
GOTO *	Direcciona el intérprete de comandos de Windows a una línea con etiqueta.
GPRESULT	Muestra información de directiva de grupo por equipo o usuario.
GRAFTABL	Permite a Windows mostrar un conjunto de caracteres extendidos en modo gráfico.

Lista de comandos CMD en Windows (MS-DOS)

Comando	Funcionalidad
HELP	Proporciona información de ayuda para los comandos Windows.
ICACLS	Muestra, modifica, hace copias de seguridad o restaura listas de control de acceso (ACL) para archivos y directorios.
IF *	Ejecuta procesos condicionales en programas por lotes.
LABEL	Crea, cambia o elimina la etiqueta del volumen de un disco.
MD *	Crea un directorio.
MKDIR *	Crea un directorio.
MKLINK *	Crea vínculos simbólicos y vínculos físicos.
MODE	Configura un dispositivo de sistema.
MORE	Muestra la información pantalla por pantalla.
MOVE *	Mueve uno o más archivos de un directorio a otro.
OPENFILES	Muestra archivos compartidos abiertos por usuarios remotos como recurso compartido de archivos.
PATH *	Muestra o establece una ruta de búsqueda para archivos ejecutables.

Lista de comandos CMD en Windows (MS-DOS)

Comando	Funcionalidad
PAUSE *	Suspende el proceso de un archivo por lotes y muestra un mensaje.
POPD *	Restaura el valor anterior del directorio actual guardado en PUSHD.
PRINT	Imprime un archivo de texto.
PROMPT *	Cambia el símbolo de comandos de Windows.
PUSHD *	Guarda el directorio actual y después lo cambia.
RD *	Elimina un directorio.
RECOVER	Recupera la información legible de un disco dañado o defectuoso.
REM *	Registra comentarios (notas) en archivos por lotes o CONFIG.SYS.
REN *	Cambia el nombre de uno o más archivos.
RENAME *	Cambia el nombre de uno o más archivos.
REPLACE	Reemplaza archivos.
RMDIR *	Elimina (quita) un directorio.
ROBOCOPY	Utilidad avanzada para copiar archivos y árboles de directorios.

Lista de comandos CMD en Windows (MS-DOS)

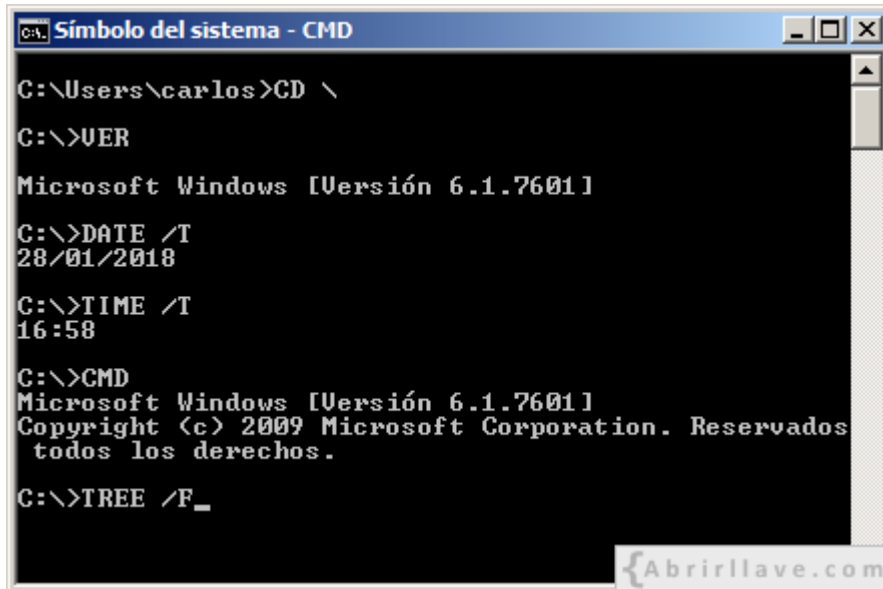
<i>Comando</i>	<i>Funcionalidad</i>
SC	Muestra o configura servicios (procesos en segundo plano).
SCHTASKS	Programa comandos y programa para ejecutarse en un equipo
SET *	Muestra, establece o elimina (quita) variables de entorno en Windows.
SETLOCAL *	Inicia la localización de los cambios de entorno en un archivo por lotes.
SHIFT *	Cambia la posición de parámetros reemplazables en archivos por lotes.
SHUTDOWN	Permite el apagado local o remoto de un equipo.
SORT	Ordena la salida.
START *	Inicia otra ventana para ejecutar un programa o comando especificado.
SUBST	Asocia una ruta de acceso con una letra de unidad.
SYSTEMINFO	Muestra las propiedades y la configuración específicas del equipo.
TASKKILL	Termina o interrumpe un proceso o aplicación que se está ejecutando.

Lista de comandos CMD en Windows (MS-DOS)

Comando	Funcionalidad
TASKLIST	Muestra todas las tareas en ejecución, incluidos los servicios.
TIME *	Muestra o establece la hora del sistema.
TITLE *	Establece el título de la ventana de una sesión de cmd.exe .
TREE	Muestra gráficamente la estructura de directorios de una unidad o ruta de acceso.
TYPE *	Muestra el contenido de uno o más archivos de texto (<i>.bat</i> , <i>.txt</i> ,...).
VER *	Muestra la versión de Windows.
VERIFY *	Comunica a Windows si debe comprobar que los archivos se escriben de forma correcta en un disco.
VOL *	Muestra la etiqueta del volumen y el número de serie del disco.
WMIC	Muestra información de WMI en el shell de comandos interactivo.
XCOPY	Copia archivos y árboles de directorios.

Ejemplos de uso de comandos CMD

Véase en la siguiente imagen la ejecución de algunos comandos:



```
C:\Users\carlos>CD \  
C:\>VER  
Microsoft Windows [Versión 6.1.7601]  
C:\>DATE /T  
28/01/2018  
C:\>TIME /T  
16:58  
C:\>CMD  
Microsoft Windows [Versión 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. Reservados  
todos los derechos.  
C:\>TREE /F_  
  
{Abrirllave.com}
```

(*) Con un asterisco se han marcado los comandos internos de **cmd.exe**. En cuanto a los comandos externos, lo normal es que estén ubicados en el directorio *C:\WINDOWS\System32*. Por ejemplo: *help.exe*, *tree.exe*, *xcopy.exe*...