



UD 02.INSTALACIÓN Y USO DE ENTORNOS IDE

Entornos de desarrollo (ED)

1.1 ¿Qué es un Entorno de Desarrollo Integrado (IDE)?

• Un ...

... Entorno de Desarrollo Integrado o Integrated Development Environment (IDE)

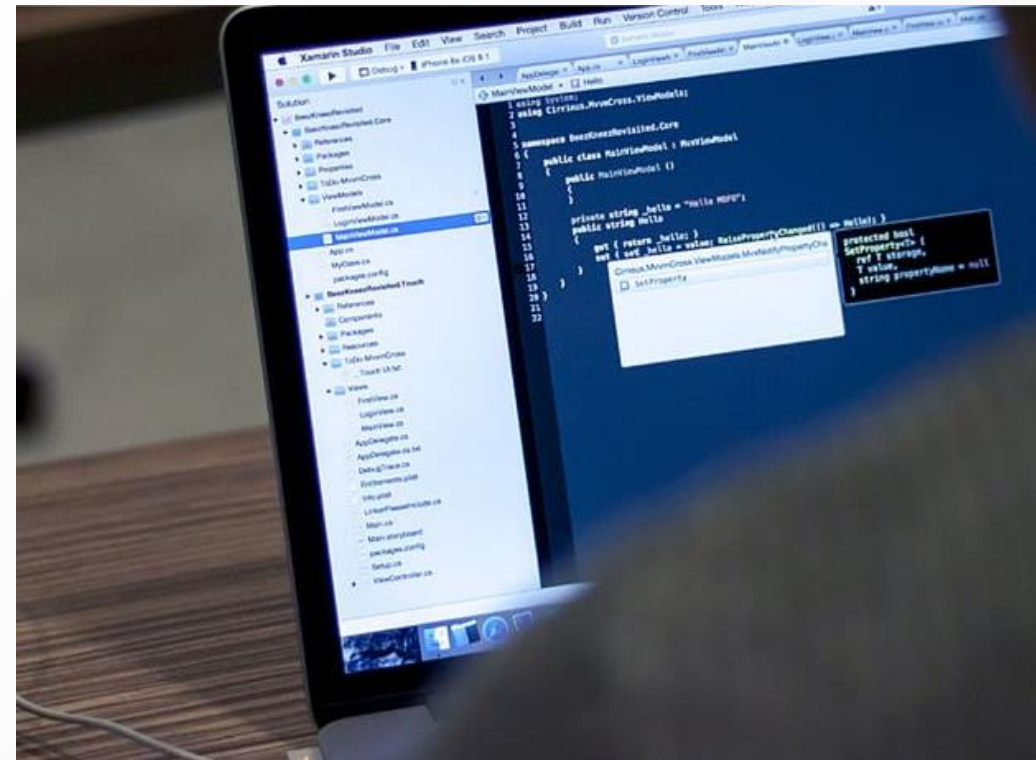
... es un programa informático

... que tiene el objetivo de asistir al programador en la tarea de

... diseñar y codificar software mediante la inclusión de múltiples herramientas destinadas para dicha tarea.

• En definitiva,

... un IDE es un programa que ayuda a programar.



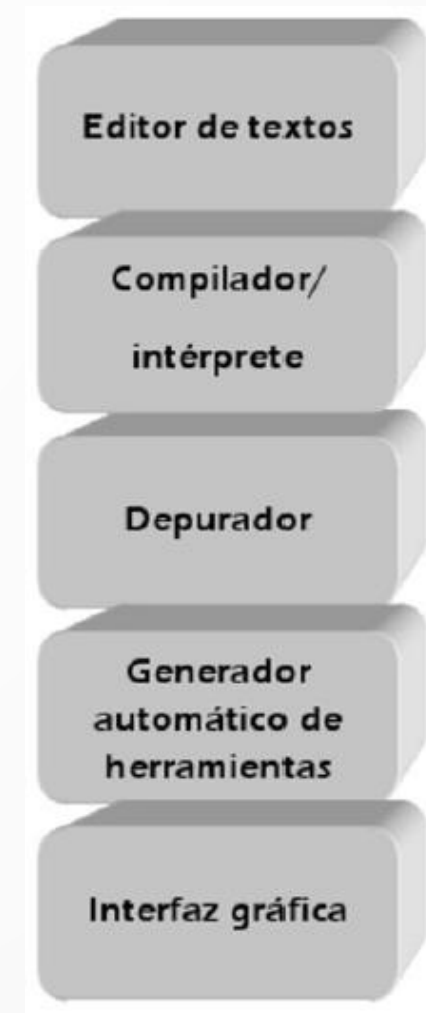
1.1 ¿Qué es un Entorno de Desarrollo Integrado (IDE)?

- Los elementos básicos de un IDE son:

- **Un editor de código:** para escribir el código fuente
- **Un compilador:** para generar el código objeto y el ejecutable de lenguajes no interpretados
- **Un intérprete:** para traducir el código
- **Un depurador:** que ayuda a corregir errores
- **Generador automático de herramientas:** documentación, control de versiones
- **Interfaz gráfica:** nos permite hacer uso de los diferentes plugins de una forma fácil y sencilla

- Sus características principales son:

- Pueden ser para uno o más lenguajes de programación
- Ayudan en la visualización del código fuente
- Permite moverse rápidamente entre los ficheros de la aplicación



1.1 ¿Qué es un Entorno de Desarrollo Integrado (IDE)?

- Usar un IDE supone una GRAN ayuda a la programación.

- La ayuda principal es el autocompletado de código que, cada vez que comenzamos a escribir una palabra reservada, aparece una ayuda contextual indicando las opciones con las que seguir.

- Pero...

- Pueden hacer que el programador se acomode

- Pueden generar dependencia

- Consumen más recursos

- Algunos son de pago

- Hoy en día existen editores de texto que realizan las funciones básicas de un IDE:

- Notepad++, Texpad, Klite...

- Todos ellos identifican las palabras reservadas y los elementos clave coloreándolos.

- Entonces... **¿vale la pena usar un IDE?**

1.2 ¿Por qué usar un IDE?

- Un IDE ofrece mucho más que simplemente autocompletar.
- Herramientas para **refactorizar (optimizar el código)*** y depurar que hacen, atención, **más corto el ciclo de vida del software**.
- **Personalización de la interfaz**: con más o menos barras de herramientas, con atajos de teclado, con comandos personalizados, con distintas ubicaciones física de los distintos paneles, etc.
- **Virtualización de la ejecución del código**, de modo que se puede probar la funcionalidad del programa sin tener que desplegarla realmente.
- ***refactorización** lo veremos este curso

Depurador	• Analiza elementos del programa y revisa errores
GUI	• (Graphical User Interface) Crea ventanas, botones...
Control de versiones	• Guarda el código fuente sin perder versiones anteriores
Plantillas	• Crea programas en base a uno creado como modelo
Prueba	• Prueba programas
Multiplataforma	• Uso en distintos sistemas operativos
Multilenguaje	• Uso con distintos lenguajes de programación
Libre/pago	• Permite la opción de uso pagando/sin pagar
Plugins	• Instala nuevas funciones
Bases de datos	• Interactúa con bases de datos
Tomcat	• Puede incluir el servidor de aplicaciones web interactivas Tomcat
UML	• Realizar gráficos con el lenguaje de modelado UML
Documentación	• Documentar el código
Refactorización	• Optimiza el código
Formateo de código	• Organiza el código fuente para que sea más legible

1.2 ¿Por qué usar un IDE?

•Un IDE permite, también, el desarrollo colaborativo.

–Desarrollar el software de manera **descentralizada y distribuida**, a través de la integración con el **control de versiones***

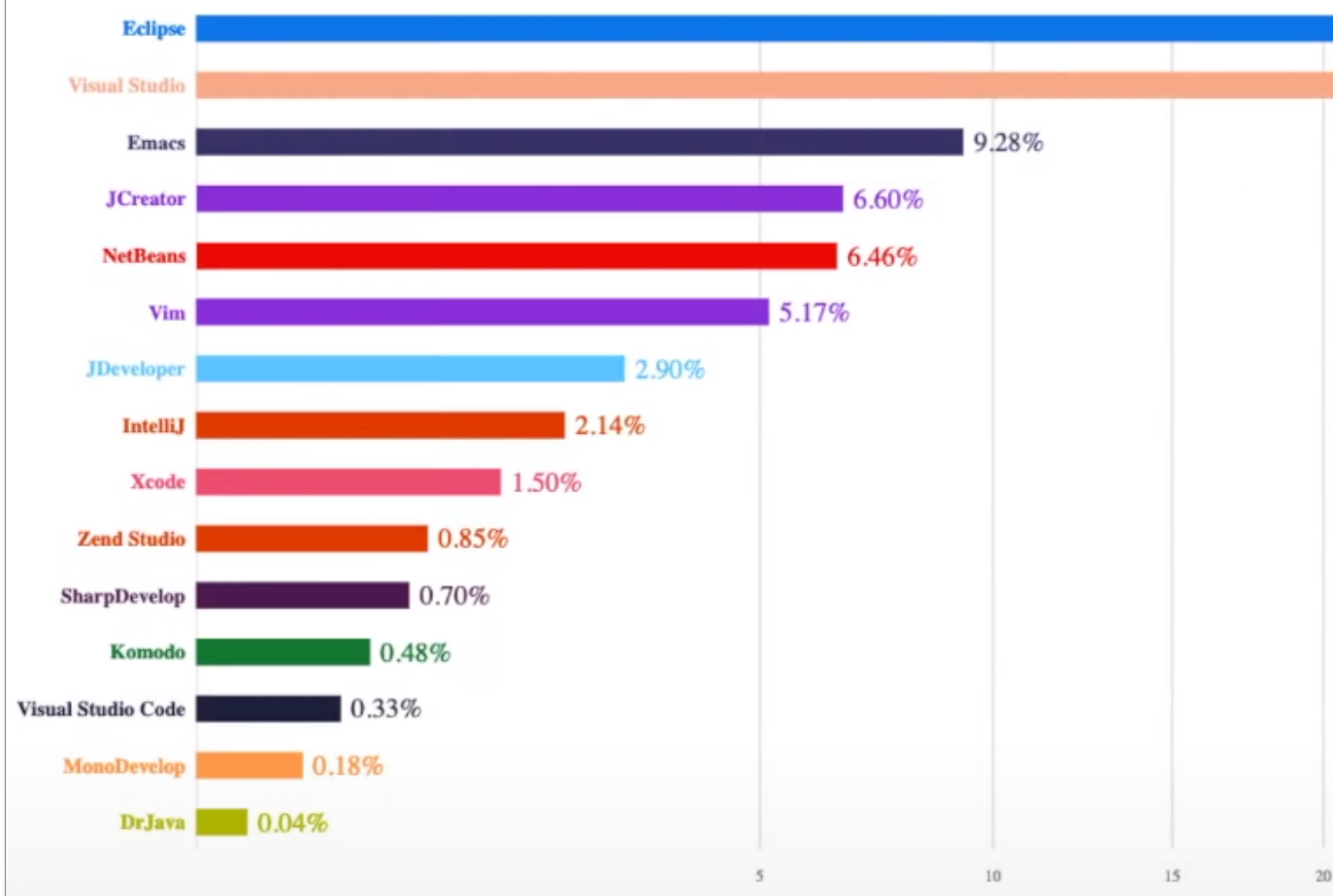
–Así, varios programadores pueden trabajar sobre el mismo código mejorándolo, tal y como ocurre en el software libre.

–***control de versiones** lo veremos en noviembre

Depurador	•Analiza elementos del programa y revisa errores
GUI	•(Graphical User Interface) Crea ventanas, botones...
Control de versiones	•Guarda el código fuente sin perder versiones anteriores
Plantillas	•Crea programas en base a uno creado como modelo
Prueba	•Prueba programas
Multiplataforma	•Uso en distintos sistemas operativos
Multilenguaje	•Uso con distintos lenguajes de programación
Libre/pago	•Permite la opción de uso pagando/sin pagar
Plugins	•Instala nuevas funciones
Bases de datos	•Interactúa con bases de datos
Tomcat	•Puede incluir el servidor de aplicaciones web interactivas Tomcat
UML	•Realizar gráficos con el lenguaje de modelado UML
Documentación	•Documentar el código
Refactorización	•Optimiza el código
Formateo de código	•Organiza el código fuente para que sea más legible

1.3 Evolución de los IDE

The Most Popular IDE (Top IDE index) 2004 - 2019



Worldwide, Oct 2022 compared to a year ago:

Rank	Change	IDE	Share	Trend
1		Visual Studio	28.14 %	-0.4 %
2	↑	Visual Studio Code	13.34 %	+1.7 %
3	↓	Eclipse	12.63 %	-1.0 %
4		Android Studio	8.82 %	-0.5 %
5		pyCharm	8.46 %	+0.5 %
6		IntelliJ	7.29 %	+0.9 %
7		NetBeans	4.57 %	-0.1 %
8	↑	Sublime Text	3.74 %	+0.4 %
9	↓	Xcode	3.04 %	-0.4 %
10		Atom	2.68 %	-0.4 %
11		RStudio	2.49 %	-0.2 %
12		Code::Blocks	1.55 %	-0.3 %
13		Vim	0.99 %	+0.2 %
14		PhpStorm	0.51 %	+0.0 %
15		Xamarin	0.31 %	-0.1 %

1.3 Evolución de los IDE

El uso de los entornos integrados de desarrollo se ratifica y afianza en los 90 y hoy en día contamos con infinidad de IDE, tanto de licencia libre como propietaria

Tabla de los IDE más relevantes hoy en día:

Entorno de desarrollo	Lenguajes que soporta	Tipo de licencia
NetBeans	C/C++, Java, JavaScript, PHP, Python.	De uso público.
Eclipse	Ada, C/C++, Java, JavaScript, PHP.	De uso público.
Microsoft Visual Studio.	Basic, C/C++, C#.	Propietario.
C++ Builder.	C/C++.	Propietario.
JBuilder.	Java.	Propietario.

1.4 IDEs para JAVA

- **Eclipse**
Proyecto multiplataforma (Windows, Linux, Mac) de código abierto, fundado por IBM en noviembre de 2001, desarrollado en Java
- **Netbeans**
Proyecto multiplataforma (Windows, Linux, Mac, Solaris) de código abierto, fundado por Sun Microsystems en junio de 2000, desarrollado en Java
- **JBuilder**
Herramienta multiplataforma (Windows, Linux, Mac), propiedad de la empresa Borland, aparecida en 1995. Un clásico. Última versión en 2008
- **JDeveloper**
Herramienta multiplataforma (Windows, Linux, Mac) gratuita, propiedad de la empresa Oracle, aparecida en 1998, inicialmente basada en JBuilder pero desarrollada desde 2001 en Java
- **Jcreator**
Es un producto comercial de la compañía Xinox Software. Utiliza el J2SDK de Sun Microsystems para compilar y ejecutar los programas. A diferencia de otros IDEs para Java este programa no es una aplicación nativa Java.
- **BlueJ**
Es un IDE para Java desarrollado principalmente con propósitos educativos, pero también es adecuado para el desarrollo de software a pequeña escala.
- **Visual Studio Code**
Desarrollado por Microsoft para Windows , Linux y macOS.
Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario requiriendo tus datos de uso del programa legalmente.

1.5 Editores de código

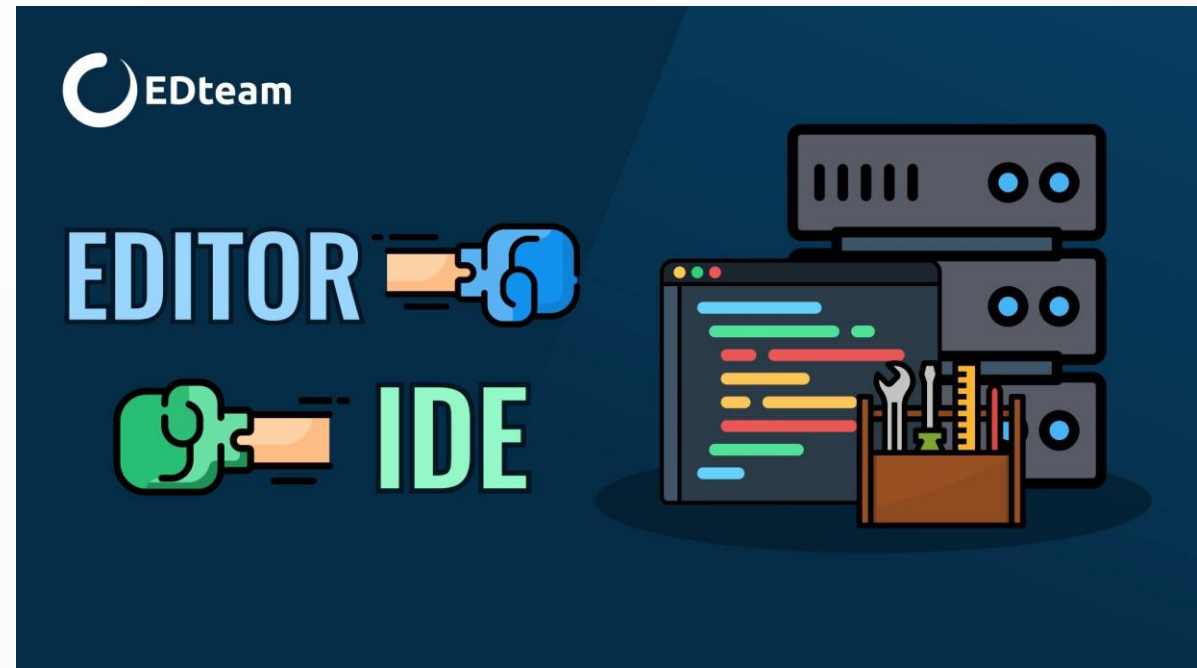
Los editores de código no tienen tantas características como los IDE, pero presenta una serie de ventajas:

- Son más rápidos y simples.

Aunque tienen varias desventajas:

- Solo podemos trabajar con ficheros concretos, en lugar de proyectos enteros.
- Disponen de menos cantidad de plugins.

El editor de texto más conocido en Windows es el **bloc de notas**, mientras que en Ubuntu es **Gedit**.



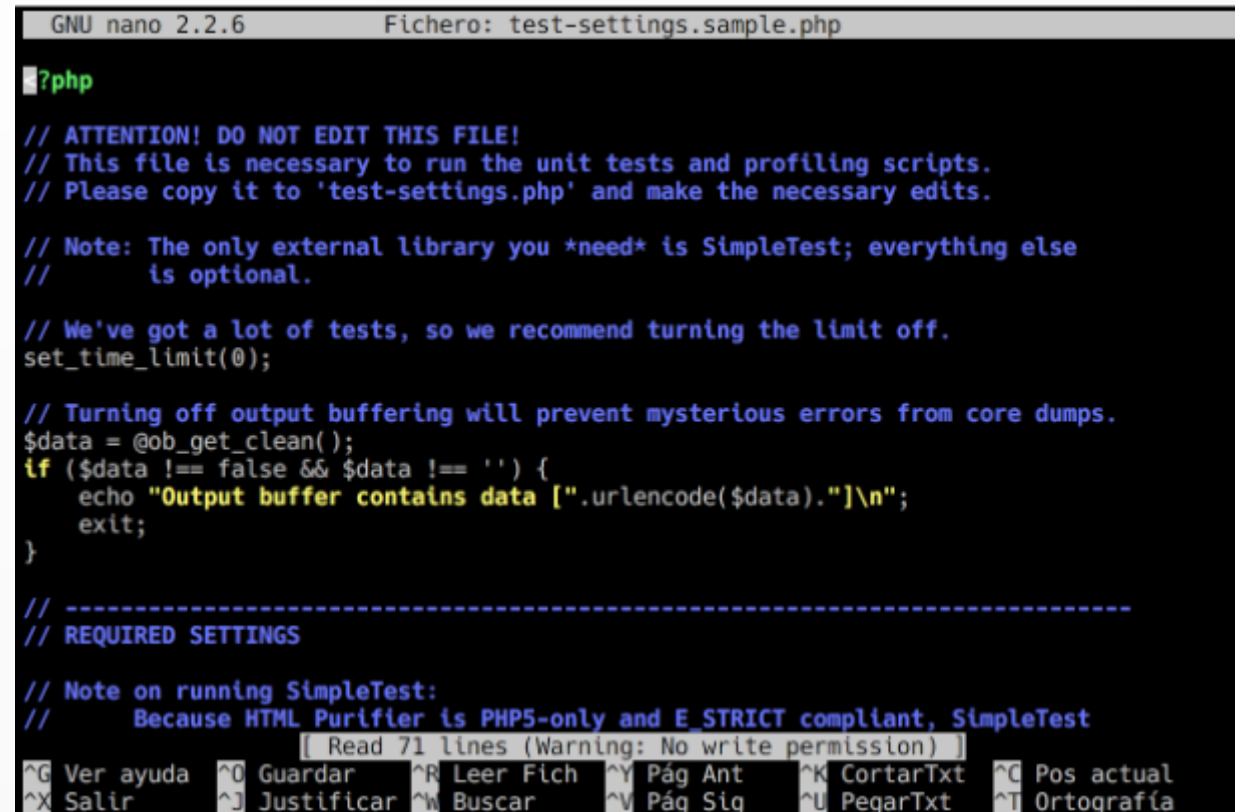
1.5 Editores de código

En las distribuciones Linux es muy común el uso de editores de código disponibles desde la terminal.

Los más conocidos son:

- vi
- vim
- nano

No disponen de interfaz gráfica y tenemos que ayudarnos de combinaciones de teclas para su uso:



```
GNU nano 2.2.6 Fichero: test-settings.sample.php

?php

// ATTENTION! DO NOT EDIT THIS FILE!
// This file is necessary to run the unit tests and profiling scripts.
// Please copy it to 'test-settings.php' and make the necessary edits.

// Note: The only external library you *need* is SimpleTest; everything else
//       is optional.

// We've got a lot of tests, so we recommend turning the limit off.
set_time_limit(0);

// Turning off output buffering will prevent mysterious errors from core dumps.
$data = @ob_get_clean();
if ($data !== false && $data !== '') {
    echo "Output buffer contains data [".urlencode($data)."]\n";
    exit;
}

// -----
// REQUIRED SETTINGS

// Note on running SimpleTest:
//   Because HTML Purifier is PHP5-only and E_STRICT compliant, SimpleTest
//   [ Read 71 lines (Warning: No write permission) ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

JDK en modo consola

1.6 Entorno JDK

Podemos diferenciar entre:

- **Java SE (Java Standard Edition)**

- ▣ es la versión estándar de la plataforma, siendo esta plataforma base para todo entorno de desarrollo en Java en cuanto a aplicaciones cliente, de escritorio o web

- **Java EE (Java Enterprise Edition)**

- ▣ es la versión más grande de Java y se utiliza en general para crear aplicaciones grandes de cliente / servidor y para desarrollo de WebServices

- En este curso se utilizarán las funcionalidades de Java SE

1.6 Entorno JDK

- La compilación de Java no genera ejecutables sino código byte-code que se almacena

- ▣ en el directorio bin
- ▣ en forma de ficheros .class

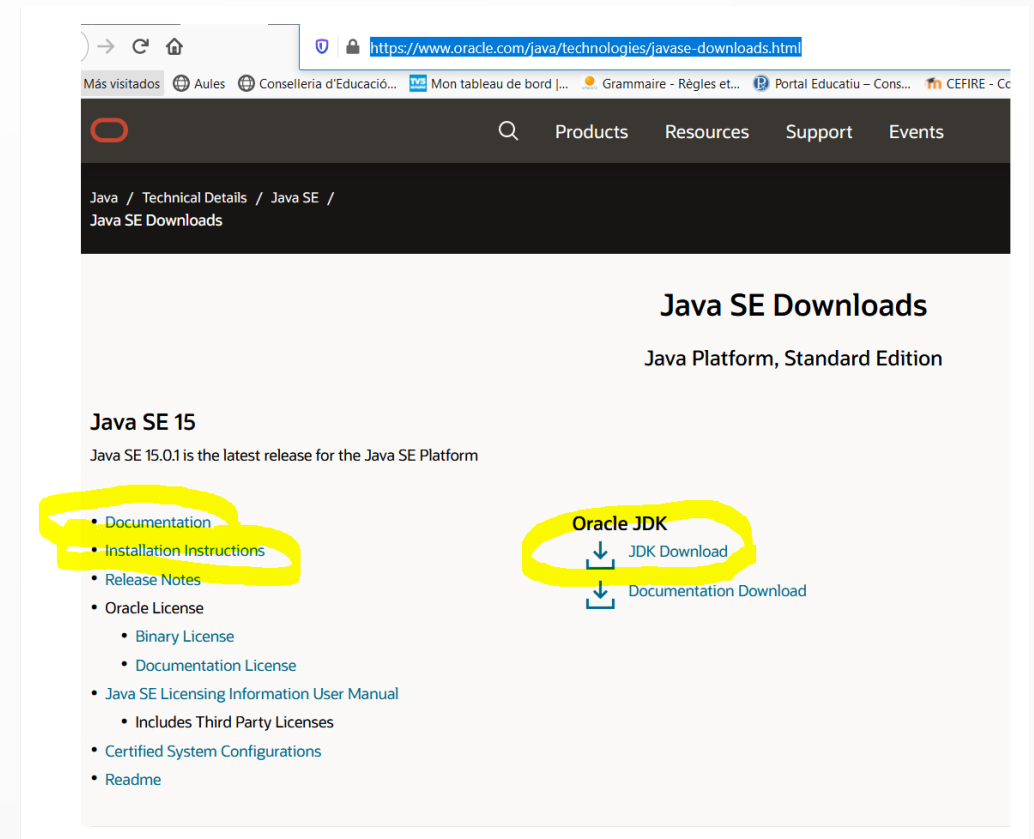


- La distribución de las aplicaciones y posterior desarrollo se puede llevar a cabo creando alguno de los siguientes tipos de encapsulamientos:
 - ▣ **JAR** (Java Archive): es un formato de archivo independiente de la plataforma que permite que varios archivos puedan ser encapsulados dentro de uno solo, de modo que éste pueda ser una aplicación completa de fácil movilidad y ejecución
 - ▣ **WAR** (Web Application archive): es un archivo JAR (con la extensión WAR) usado para distribuir una colección de archivos JSP, servlets, clases Java, archivos XML y contenido web estático (HTML). En conjunto constituyen una aplicación Web
 - ▣ **EAR** (Enterprise Archive File): es un formato para empaquetar varios módulos en un solo archivo. Permite desplegar varios módulos de éstos en un servidor de aplicaciones. Contiene archivos XML, llamados descriptores de despliegue, que describen cómo efectuar esta operación (EAR = JAR + WAR)

1.6 Entorno JDK

La página oficial:

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



1.6 Entorno JDK

- En mi caso tengo instalado el JDK en
C:\Program Files\Java\jdk-XX.X.X

Nota: Compruébalo con `javac --version`

Podemos
comprobarlo:

- ▣ Cmd
- ▣ (ó Símbolo del Sistema)

```
c:\>javac -version
javac 15.0.1

c:\>cd C:\Program Files\Java\jdk-15.0.1\bin

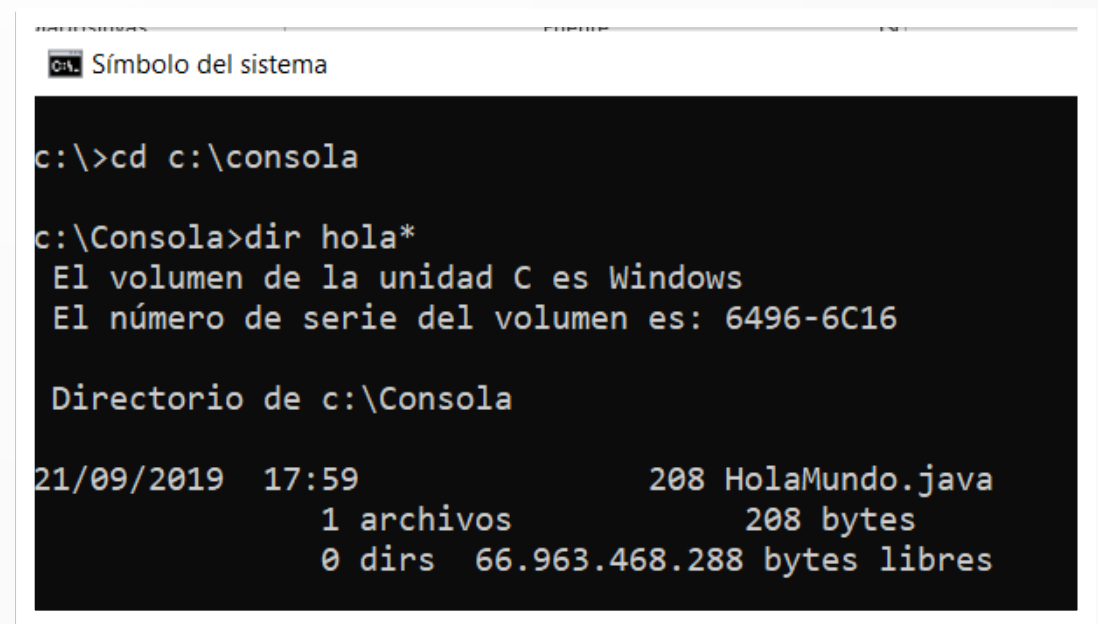
C:\Program Files\Java\jdk-15.0.1\bin>dir java*.exe
El volumen de la unidad C es Windows
El número de serie del volumen es: 6496-6C16

Directorio de C:\Program Files\Java\jdk-15.0.1\bin

17/11/2020  21:16                50.832 java.exe
17/11/2020  21:16                20.624 javac.exe
17/11/2020  21:16                20.624 javadoc.exe
17/11/2020  21:16                20.624 javap.exe
17/11/2020  21:16                50.832 javaw.exe
                    5 archivos          163.536 bytes
                    0 dirs 66.957.983.744 bytes libres
```

1.6 Compilar y ejecutar desde consola

- En primer lugar, debo escribir mi programa con el bloc de notas o algún editor similar
 - ▣ Supongamos que tenemos el fichero HolaMundo.java en la carpeta c:\Consola>
 - ▣ Vemos que está el fichero:



```
Símbolo del sistema

c:\>cd c:\consola

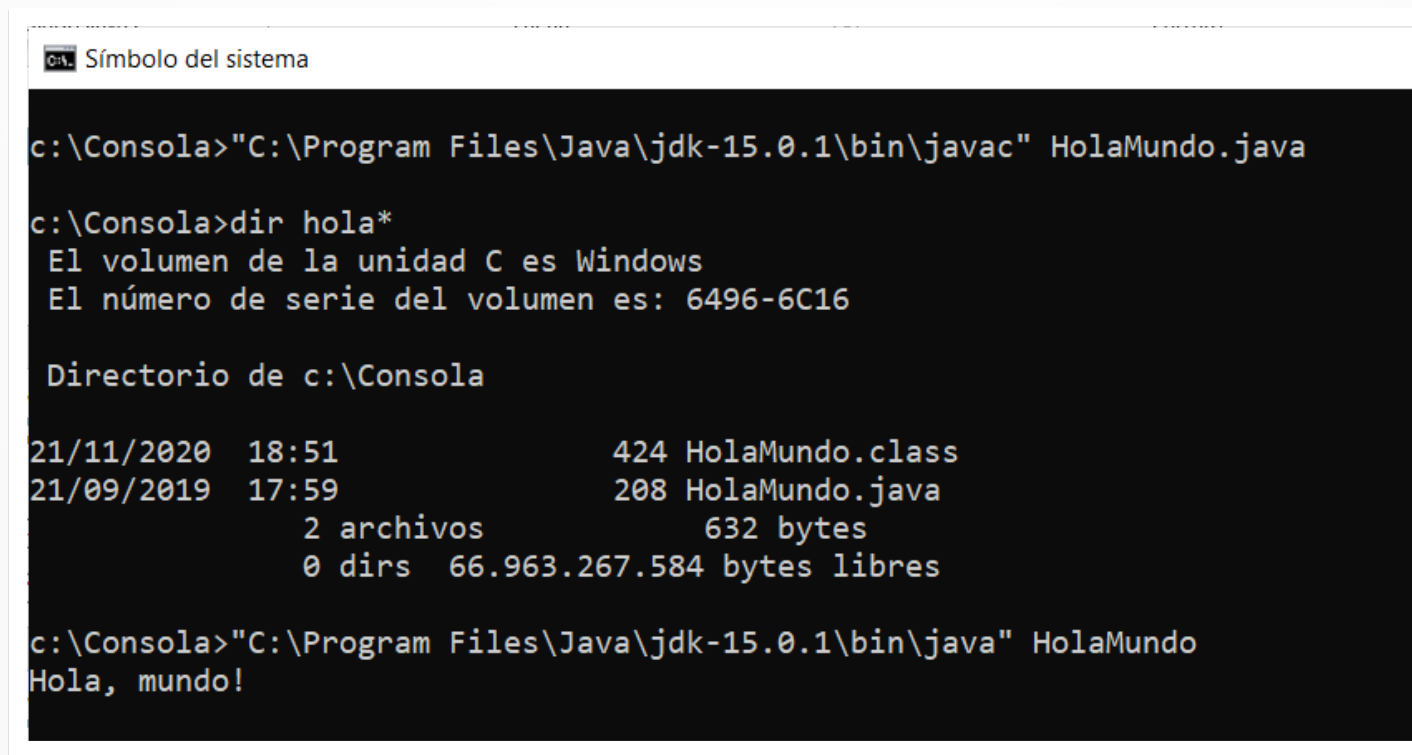
c:\Consola>dir hola*
El volumen de la unidad C es Windows
El número de serie del volumen es: 6496-6C16

Directorio de c:\Consola

21/09/2019  17:59                208 HolaMundo.java
                1 archivos                208 bytes
                0 dirs 66.963.468.288 bytes libres
```

1.6 Compilar y ejecutar desde consola

- Compilamos y ejecutamos
- Si no configuramos la variable PATH, debemos especificar la ruta completa al archivo ejecutable cada vez que lo ejecutemos. Por ejemplo:



```
c:\Consola>"C:\Program Files\Java\jdk-15.0.1\bin\javac" HolaMundo.java

c:\Consola>dir hola*
El volumen de la unidad C es Windows
El número de serie del volumen es: 6496-6C16

Directorio de c:\Consola

21/11/2020  18:51                424 HolaMundo.class
21/09/2019  17:59                208 HolaMundo.java
                2 archivos                632 bytes
                0 dirs 66.963.267.584 bytes libres

c:\Consola>"C:\Program Files\Java\jdk-15.0.1\bin\java" HolaMundo
Hola, mundo!
```


1.6 Compilar y ejecutar desde consola

Establecer PATH

- Es útil establecer la variable PATH de forma permanente para JDK.
- Para ello:
 - ▣ Selecciona Panel de control y luego Sistema
 - ▣ Haz clic en “Configuración Avanzada del Sistema” y luego en “Variables de Entorno”
 - ▣ Agrega la ubicación de la carpeta bin de la instalación de JDK a la variable PATH en Variables del sistema
 - En nuestro caso `C:\Program Files\Java\jdk-XX.X.X\bin`
 - ▣ Recuerda:
 - La variable de entorno PATH es una serie de directorios separados por punto y coma (;) y no distingue entre mayúsculas y minúsculas
 - Microsoft Windows busca programas en los directorios PATH en orden, de izquierda a derecha
 - Solo debe tener un JDK (con un directorio bin) en la ruta. Si hay más de uno, los siguientes serán ignorados
 - La nueva ruta entra en vigencia en cada nueva ventana de comando que abras después de configurar la variable PATH

1.6 Compilar y ejecutar desde consola

Establecer PATH

The image shows the Windows 'Sistema' (System) control panel window and the 'Propiedades del sistema' (System Properties) dialog box. In the control panel, the 'Configuración avanzada del sistema' (Advanced system settings) link is circled in red. The 'Propiedades del sistema' dialog box is open, showing the 'Opciones avanzadas' (Advanced) tab. The 'Rendimiento' (Performance) section has the 'Configuración...' button circled in red. The 'Variables de entorno...' (Environment variables) button at the bottom right is also circled in red.

Sistema

Panel de control > Sistema y seguridad > Sistema

Ventana principal del Panel de control

- Administrador de dispositivos
- Configuración de Acceso remoto
- Protección del sistema
- Configuración avanzada del sistema**

Ver información básica acerca del equipo

Edición de Windows

Windows 10 Pro

© 2019 Microsoft Corporation. Todos los derechos reservados.

Sistema

Procesador: AMD A10-6800B APU with Radeon(tm) HD Graphics 4.10 GHz

Memoria instalada (RAM): 8,00 GB (7,23 GB utilizable)

Tipo de sistema: Sistema operativo de 64 bits, procesador x64

Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo: DSP-PC41

Nombre completo de equipo: DSP-PC41

Descripción del equipo:

Grupo de trabajo: WORKGROUP

Activación de Windows

Windows está activado [Lea los Términos de licencia del software de Microsoft](#)

Id. del producto: 00331-20463-52506-AA328

Propiedades del sistema

Nombre de equipo | Hardware

Opciones avanzadas | Protección del sistema | Acceso remoto

Para realizar la mayoría de estos cambios, inicie sesión como administrador.

Rendimiento

Efectos visuales, programación del procesador, uso de memoria y memoria virtual

Configuración...

Perfiles de usuario

Configuración del escritorio correspondiente al inicio de sesión

Configuración...

Inicio y recuperación

Inicio del sistema, errores del sistema e información de depuración

Configuración...

Variables de entorno...

Aceptar Cancelar Aplicar

1.6 Compilar y ejecutar desde consola

Establecer PATH

■ Editar la variable Path:

■ Solicitar Nuevo valor

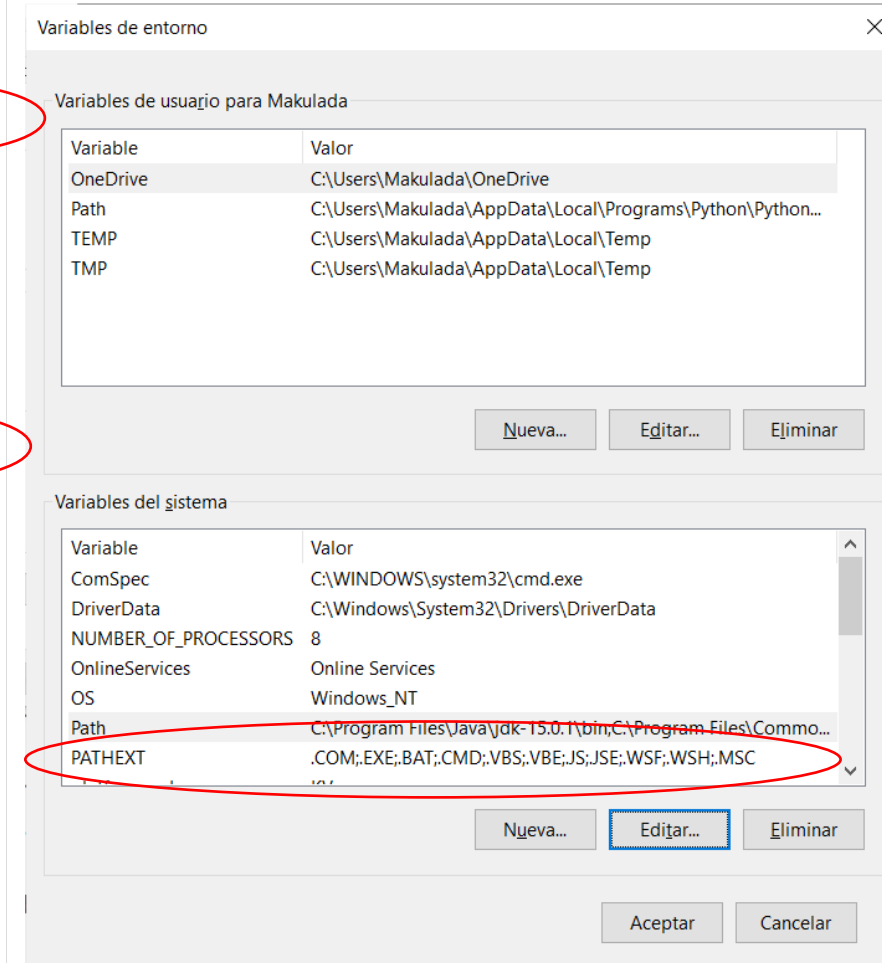
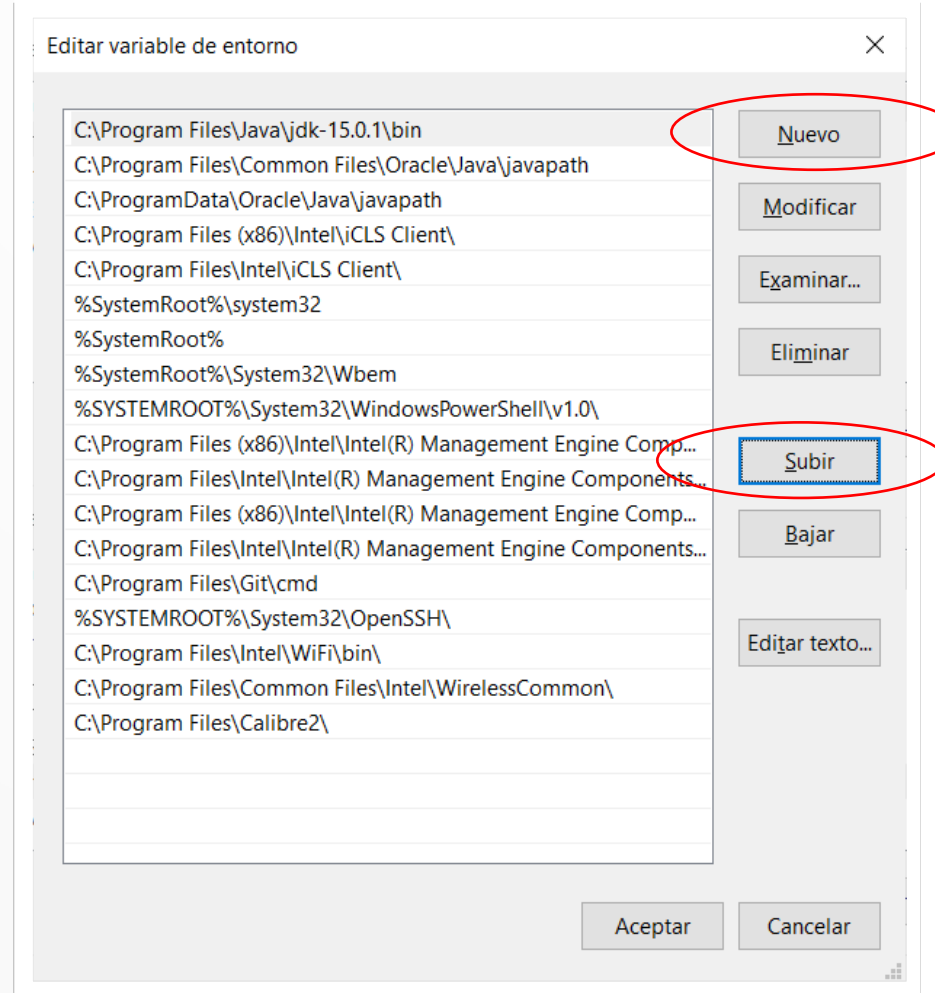
■ Añadir

C:\Program Files\Java\jdk-XX.X.X\bin

■ Subir al primer puesto

■ Aceptar

XX.X.X es el número de tu versión de JAVA.



1.6 Compilar y ejecutar desde consola

- Con este cambio y desde el directorio donde esta HolaMundo.java ya podrás compilar y ejecutar el programa:

- Compilar:

`javac HolaMundo.java`

- Ejecutar:

`java HolaMundo (¡Ojo! Sin extensión)`

```
E:\consola>dir *.java
El volumen de la unidad E es Imma
El número de serie del volumen es: 280F-DBA8

Directorio de E:\consola

29/11/2018  11:58                204 HolaMundo.java
01/12/2016  10:22                517 PideNumero.java
02/12/2018  21:06                314 HolaNom.java
                3 archivos              1.035 bytes
                0 dirs  40.596.406.272 bytes libres

E:\consola>javac HolaMundo.java

E:\consola>java HolaMundo
Hola, mundo!
E:\consola>
```

1.7 Ficheros .JAR

- Los ficheros JAR (Java ARchives) permiten recopilar en un solo fichero varios ficheros diferentes almacenándolos en un formato comprimido para que ocupen menos espacio. Es por tanto algo similar a un fichero .zip. La particularidad de los ficheros .jar es que no necesitan ser descomprimidos para ser usados.
- Una aplicación Java habitualmente se compone de muchos archivos java. Hay que compilarlos todos para crear los archivos class, y para que un usuario pueda usar el programa es necesario que tenga todos esos archivos class. Esto no es práctico ni una forma sencilla para que los usuarios utilicen nuestras aplicaciones. Por ello, lo habitual es distribuir las aplicaciones Java como ficheros JAR que contienen todo lo necesario para ejecutar el programa por línea de comandos o incluso con doble click.

1.7 Ficheros .JAR

Crear

- Para crear un JAR primero es necesario crear el Manifest, un archivo de texto que contiene información sobre la versión del programa, el creador y la clase principal a ejecutar.
 - Con el bloc de notas creamos un nuevo archivo de texto llamado Manifest.txt con el texto de abajo, y lo guardamos en la misma carpeta del archivo HolaMundo.java
 - Manifest-version: 1.0
Created-By: 1.0 (DAW)
Main-Class: HolaMundo
- Ahora para crear el archivo JAR ejecutaremos el comando:

```
C:\Users\raulp>jar cfm HolaMundo.jar Manifest.txt HolaMundo.class
```

1.7 Ficheros .JAR

Ejecutar

- El intérprete de Java es capaz de ejecutar los archivos comprimidos en un archivo jar directamente
 - ▣ Por ejemplo, si hemos recopilado todos los ficheros necesarios para ejecutar una aplicación en un fichero “HolaMundo.jar” podemos lanzar la aplicación desde una terminal de texto con:
 - > `java -jar HolaMundo.jar`