

Hola Mundo en Java, con el IDE Eclipse

Hacemos un Hola Mundo en Java usando el IDE Eclipse para crear un proyecto, compilar y poner en ejecución el código fácilmente.

Ya sabemos lo [que es Java y hemos configurado nuestro entorno de trabajo](#), descargando tanto el JDK como el IDE Eclipse. Ahora podemos poner manos a la obra con un primer ejemplo, realizando el típico "Hola Mundo".

Pero antes de comenzar debemos advertir que Java no es tan sencillo como otros lenguajes por varios factores. Para comenzar, en cantidad de lenguajes de programación que hayamos podido experimentar, puedes hacer un "Hola Mundo" con una única línea de código, dos a lo sumo, mientras que en Java tienes que realizar varias líneas, que a primera instancia pueden parecer algo complejas. Luego que se trata de un lenguaje que requiere compilación y que normalmente se programa en un entorno de desarrollo, algo complejo de administrar para los novatos, como es el caso de Eclipse. No obstante, no cabe desanimarse, porque lo vamos a explicar todo lo mejor posible.



Crear un proyecto Java en Eclipse

Abrimos nuestra instalación de Eclipse y lo primero que nos pregunta es la localización de nuestro área de trabajo en una ventana titulada "Select workspace". Esto es porque Eclipse no se instala como los programas habituales, sino que se copian los archivos en una carpeta del ordenador y listo. [Esto se explicó en el artículo anterior](#).

Nuestro área de trabajo, "workspace", es donde guardamos todos nuestros proyectos. Podemos dejar aquella ruta que Eclipse indica de manera predeterminada. Si la cambias, simplemente acuérdate de la carpeta física que estás configurando para poder indicarla las siguientes veces que abras el IDE.

Una vez abierto Eclipse, encontramos un entorno complejo con varios paneles. ¡No te asustes!, aunque parezca que no se sabe para qué es cada cosa, con un poco de experiencia podrás ubicarlas y entender la utilidad de cada panel.

Lo que vamos a hacer para comenzar es crear nuestro primer proyecto. Hacemos "File / New / Project...". Nos sale una ventana donde seleccionamos "Java project". Ahora encontramos una ventana de configuración del proyecto que tenemos que rellenar convenientemente.

1. Asignamos un nombre al proyecto.
2. Dejamos marcada "Use default location", para que use la ruta de nuestro *workspace* configurado y Eclipse se encargará de hacer un subdirectorio para este nuevo proyecto.
3. JRE puedes dejar la que tienes por defecto en Eclipse.
4. En project layout puedes dejar también la configuración que te sale "create separate folders..."
5. Pulsamos "Finish"



Cuando terminas el proceso en el lado izquierdo, panel de "projects", nos sale el proyecto que acabamos de crear. Es un árbol desplegable en el que si examinas dentro encontrarás: **Carpeta src**: es donde está el código fuente de tus clases, todos los .class que irás creando con tu código Java.

Carpeta JRE: encuentras toda una serie de .jar, que es una forma de englobar varios archivos de clases. En definitiva, aquí encontrarás todas las clases que dispones en Java para realizar cualquier tipo de tarea.

Nota: Nota: Todas las clases del JRE son con las que partes de salida en Java, por lo que para desarrollar cualquier tipo de programa tendrás alguna clase específica para poder usar.

Como desarrolladores también podremos crear un .jar y pasarlo a otros compañeros para que dispongan de un conjunto de clases creadas por nosotros.

Crear un nuevo código para implementar el "Hola Mundo"

Ahora vamos a crear nuestro primer archivo dentro del proyecto para hacer nuestro primer ejemplo en Java. Es bien simple realmente, pero recuerda que Java solo trabaja con [programación orientada a objetos \(POO\)](#), por lo que alguna cosa puede ser poco familiar para ti, si no conoces este paradigma.

1. Botón derecho en el proyecto que acabas de crear
2. Seleccionas "New / Class". (Un class es una clase de POO)
3. En el campo "Name" indicas el nombre de tu clase
4. Todo lo demás lo puedes dejar como está, más adelante explicaremos qué otras cosas te aparecen en esta pantalla y el por qué de su existencia
5. Pulsas el Botón "Finish" para acabar



Con esto hemos creado ya nuestro primer archivo y Eclipse nos mostrará el código con el que partimos. Observarás que el archivo ya tiene un código escrito, es la declaración de una clase de Programación Orientada a Objetos (class en inglés).

Todo en Java va dentro de las clases, exceptuando los "import" que van fuera de las llaves de tu "class". De momento no nos tenemos que preocupar más que con el código para crear una salida con el mensaje "Hola Mundo", pero aquí es donde decíamos que Java es un poco doloroso para hacer nuestro primer programa, porque necesitamos varias cosas para que todo esto funcione.

Creo que lo mejor sería ver el código fuente del "Hola Mundo" completo y a partir de ahí explicar lo que nos encontramos.

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

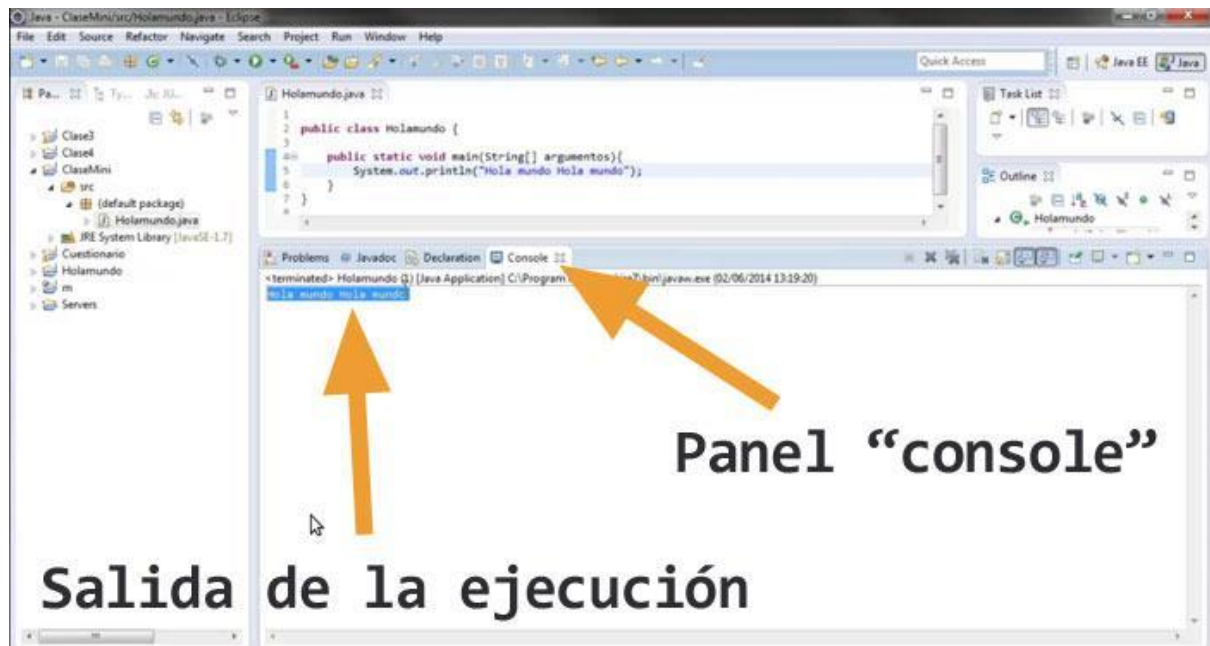
Rápidamente, estos son los elementos que te puedes fijar:

- "public class" es una declaración de la clase.
-
- "HolaMundo" es el nombre de la clase que estamos creando. Por convención los nombres de las clases en Java tienen la primera letra en mayúscula.
-
- "public static void main" es la declaración de un método público y estático llamado "main" que no devuelve valor alguno. De momento olvida todo lo demás y quédate que en Java; cuando quieres ejecutar una clase debes crear este método "main" tal como está. Al ejecutar una clase se pondrá todo en marcha a partir del código que hayas escrito en tu "main".
-
- "String[] argumentos" son los parámetros que recibe todo método main, que sirven para enviar por consola argumentos para que al ejecutar la clase se personalice el comportamiento. De momento no vamos a hacer nada con esos argumentos, por lo que puedes obviar este código.
-
- "System.out.println("Hola mundo");" es la línea de código que hace realmente el "Hola mundo". Lo que indica es que estás enviando una salida, en este caso por pantalla. Es una manera un poco fea o complicada de decir simplemente a Java que muestre algo por pantalla, pero recuerda que todo en Java son clases, por lo que aquí estás accediendo a una jerarquía de clases y al método "println()" que es el encargado de imprimir una línea. A ese método se le envía una cadena, en este caso "Hola mundo" que es aquello que debe mostrar por pantalla.

Ejecutar una clase desde Eclipse

Una vez que hemos escrito el código de nuestra clase estamos listos para ejecutarla. Lo hacemos dentro del propio IDE eclipse, con el botón "run", que tiene forma de un botón de *play*, en la barra de menús de arriba del todo.

Cuando ejecutas te aparece la salida en una consola, en un panel en la parte de abajo de Eclipse, en el que deberías ver el mensaje de salida de la ejecución de tu primer programa.



Más adelante podremos realizar otros ejemplos más complejos, pero de momento podemos quedarnos satisfechos por haber creado nuestro primer programa en Java.