

Sección 1: Hello World

Paso 1) Abre un editor de texto sencillo.

- En Windows, puedes utilizar **Notepad**.
- En Mac OS X, **TextEdit**
- En Linux, **gedit**

Cualquiera de los programas listados anteriormente deben estar instalados por default.

Paso 2) Crea un archivo de código fuente para tu programa. Recuerda que el código fuente (*source code*) es un archivo que contiene las instrucciones de código en un cierto lenguaje de programación.

- Declara una clase con el nombre HelloWorld
- Declara un método main public static void main(String args[])
- Escribe la instrucción System.out.println("Hello World")

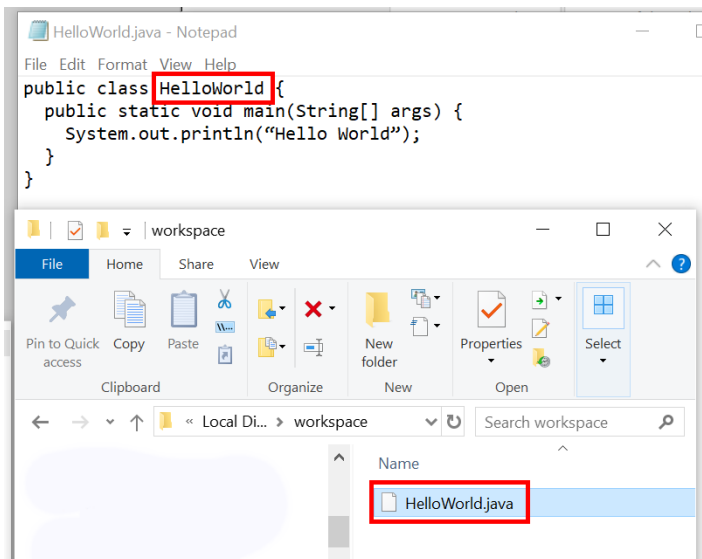
```
public class HelloWorld{  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

Paso 3) Crea un folder llamado **workspace** en el directorio

- En Windows: **C:**
- En Mac OS X: **/Users/<tu_usuario>/**
- En Linux: **/usr/<tu_usuario>/**

Paso 3) Graba el archivo como **HelloWorld.java** en el folder **workspace** que acabas de crear.

NOTA: Es importante notar que el nombre del archivo debe coincidir con el nombre de la clase en el programa. En este caso, ambos se llaman **HelloWorld**.



Paso 4) Abre una sesión de línea de comandos

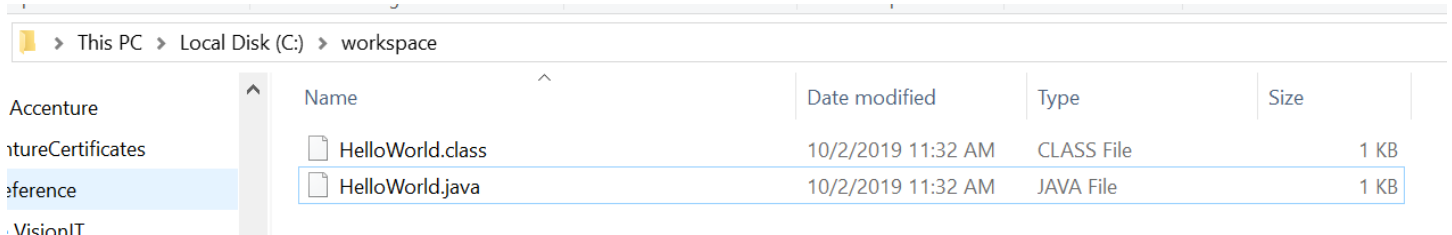
- En Windows, abre el programa **Command Prompt**.
- En Mac OS X o Linux, abre el programa **Terminal**

Paso 5) Compila el programa a través de la instrucción **javac**. Recordemos al compilar un programa de Java, estamos revisando que el programa esté correctamente codificado, y generando un archivo con terminación **class** que contiene el bytecode del programa. En la terminal, escribe `javac <ruta>/HelloWorld.java`. Sustituye <ruta> por la dirección del folder que elegiste en el Paso 3:

- En Windows: **C:\workspace**
- En Mac OS X: **/Users/<tu_usuario>/workspace/**
- En Linux: **/usr/<tu_usuario>/workspace/**

Nota: Windows utiliza diagonales invertidas \ para indicar un salto a un nuevo folder. Puedes insertar este ícono presionando Alt + 092.

Paso 6) Navega hacia el folder y debes de ver dos archivos:



Name	Date modified	Type	Size
HelloWorld.class	10/2/2019 11:32 AM	CLASS File	1 KB
HelloWorld.java	10/2/2019 11:32 AM	JAVA File	1 KB

Paso 7) Ejecuta el programa utilizando la instrucción **java -cp**. Aquí, queremos pedirle a la máquina virtual de Java que cargue el archivo `HelloWorld.class` y lo ejecute. Para esto, en la terminal, escribe: `java -cp <ruta>/HelloWorld`. Sustituye <ruta> por:

- En Windows: **C:\workspace**
- En Mac OS X: **/Users/<tu_usuario>/workspace/**
- En Linux: **/usr/<tu_usuario>/workspace/**

Nota: No es necesario agregar la terminación “.class”.

Sección 2: Impresiones en consola

Mediante la instrucción `System.out.println(<argument>)` le indicamos al programa que queremos darle salida a un texto en la consola, y terminando con un salto de línea. Dentro de los paréntesis debemos incluir un argumento, que puede ser una variable o un texto encasillado entre comillas.

Existe una segunda variación de la instrucción que es `System.out.print(<argument>)`. Esta instrucción de igual forma sirve para imprimir un texto, pero al final posiciona el cursor justo después del argumento impreso.

Por ejemplo, ambos bloques siguientes son equivalentes:

```
System.out.println("Hoy es día lunes");
```

```
System.out.print("Hoy ");
System.out.print("es ");
System.out.print("día ");
System.out.print("lunes");
```

Crea una nueva clase de Java en el folder <workspace>. Nómbrala con tu matrícula A0XXXXX. El programa deberá imprimir en pantalla un ASCII Art de mínimo 5 líneas. Incluye también tu matrícula como parte del dibujo.

¡Pónte creativo!

Ejemplo:

```
A01135459

..
$. ,o$$$$o.
$. $$$$$$$$o. ..
$. '$$$$$$ ,o"
$. $ '$$$$$$,o',.oo'
$. $. $$$$' ,, .o'.
$. '$o. 'O$ ..ooo"',oo'
$. .o$ '$$' ,,o'
.%,,,,,,ooO' ' ,,o"
$o. ,o' $o ..oo'
"O",, '$$.o'
'$ $ '$,o'
'$ $ .o $
'$ $ .$$
'$ $ .o$$
'$ $ ,o' $
$. '$. $,oooo"o,
$. $. 'o' '$
$. $. ,ooo, $
. " 'oo...o' $'o $
    $ $ "
    $ $
    $ $
    $ %
    $ $
    $ $
    "
```