

Nombre: \_\_\_\_\_ Matrícula: \_\_\_\_\_

**Completa las siguientes actividades. Al terminar, sube los tres archivos de código fuente generados (.java) a Blackboard en la actividad correspondiente.**

### Sección 1: Hello World

**Paso 1)** Abre un editor de texto sencillo.

- En Windows, **Notepad**.
- En Mac OS X, **TextEdit**. Asegúrate de crear un archivo en formato “plain text”, seleccionando Format → Make plain text.
- En Linux, **gedit**

NOTA: Cualquiera de los programas listados anteriormente debe estar instalados por default.

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

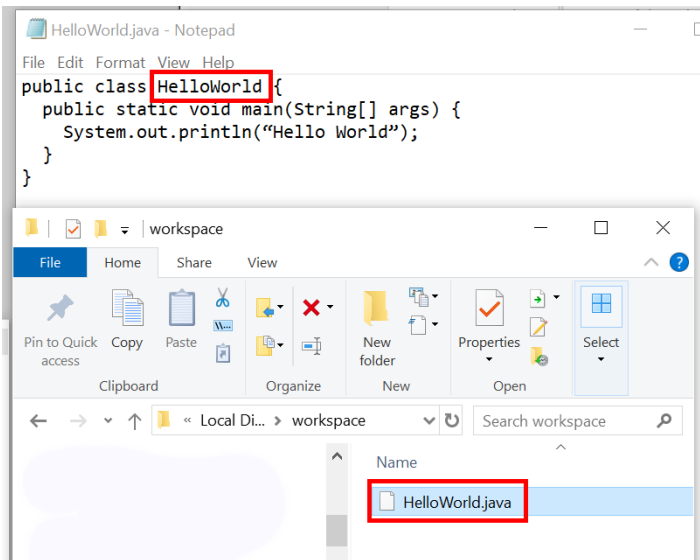
- Declara una clase con el nombre HelloWorld
- Declara un método main public static void main(String args[])
- Escribe la instrucción System.out.println(“Hello World”)

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

NOTA: Para entender este programa, puedes referirte a las últimas diapositivas de la presentación *INFO1 MOD4-Intro a Java* que se encuentra en Blackboard, o a través de este [link](#).

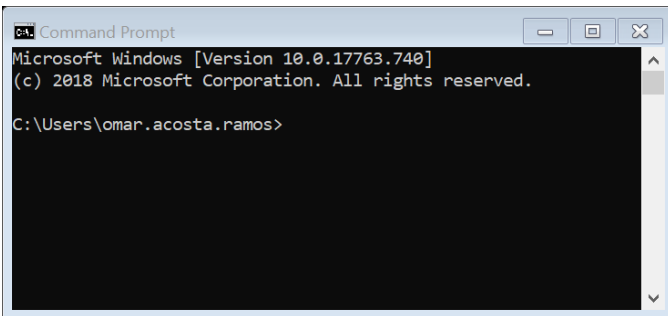
**Paso 4)** Graba el archivo como **HelloWorld.java** en el folder **workspace** que creaste en el Paso 2.

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**

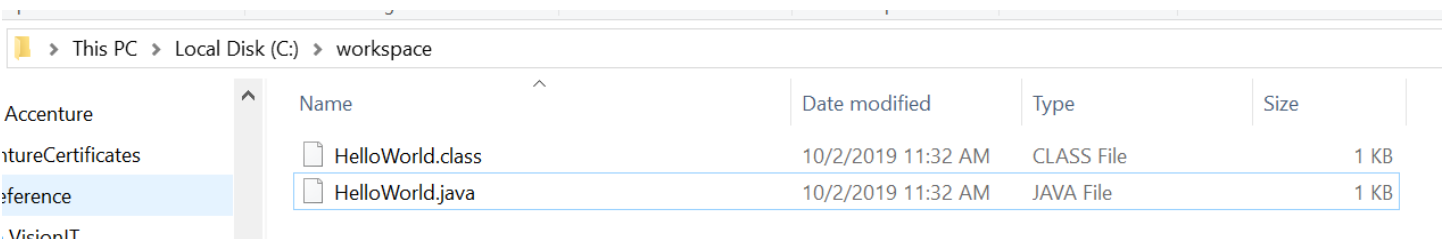


**Paso 5)** Compila el programa a través de la instrucción **javac**. Recordemos al compilar un programa de Java, estamos revisando que el programa esté correctamente codificado, y generando un archivo con terminación **.class** que contiene el bytecode del programa. En la terminal, escribe `javac <ruta>/HelloWorld.java`. 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, a diferencia de Mac y Linux, 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 verifica que se haya generado el archivo **HelloWorld.class**:



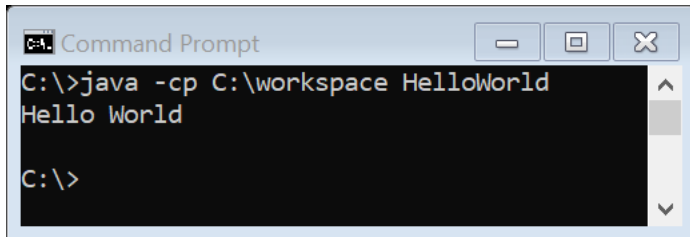
**Paso 7)** En la terminal, ejecuta el programa utilizando la instrucción **java -cp**. Aquí, buscamos pedirle a la máquina virtual de Java que cargue el archivo **HelloWorld.class** y lo ejecute.

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”, pues la instrucción **java -cp** automáticamente está diseñada para buscar archivos con esa terminación.

**Paso 8)** En la terminal, el texto “Hello World” deberá aparecer.



```
Command Prompt
C:\>java -cp C:\workspace HelloWorld
Hello World
C:\>
```

## 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 dia lunes");
```

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

**Problema 1:** Crea una nueva clase de Java en el folder `<workspace>`. Nómbrala con tu matrícula `A0XXXXX_P1.java`. 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 $
$ $ "
$ $
$ %
$, $
$ $'
"
```

Hay ciertos caracteres especiales que requieren una representación especial para que puedan ser impresos en la consola. Estos caracteres son los siguientes:

Display	Special character
Comilla sencilla	\'
Comillas dobles	\"
Diagonal invertida	\\
Tabulador ↵	\t
Backspace ←	\b
Carriage return	\r
Form feed	\f
Newline (salto de línea)	\n
Caracteres especiales Unicode	\uXXXX reemplazando XXXX por el código Unicode

Por ejemplo, la siguiente instrucción:

```
System.out.println("1\n2\n3\n4\n5");
```

Imprimiría el siguiente resultado:

```
1
2
3
4
5
```

**Problema 2.** Crea una nueva clase de Java en el folder <workspace>. Nómbrala con tu matrícula A0XXXXX\_P2.java. El programa deberá imprimir en pantalla el siguiente texto:

```
Las compras que realizó el usuario "A01135459" son:
Botella de agua          $8.50
Doritos Flamin' Hot     ¥70.62
Chicles \Trident/       $17.20
```

Asegúrate de que todos los caracteres especiales se impriman correctamente, y que los precios de los elementos estén alineados mediante tabuladores. Es decir, entre el texto “Botella de agua” y el precio “\$8.50” hay que insertar dos tabuladores ↵.