

PROGRAMACIÓN

UT1: INTRODUCCIÓN A LA PROGRAMACIÓN (II)

UT1: INTRODUCCIÓN A LA PROGRAMACIÓN (II)	7
1. Tipos de datos simples.....	7
2. Constantes y variables	8
2.1 Constantes	8
2.2 Variables	9
3. Comentarios y operadores	10
3.1 Comentarios.....	10
3.2 Operadores aritméticos	10
3.3 Operadores relacionales	10
3.4 Operadores lógicos	11
4. JDK e IDE.....	12
4.1 Instalación del IDE	12
4.2 Instalación del JDK.....	12
4.3 Variables de entorno	13
4.4 Herramientas del IDE	15
5. Hola Mundo!	16

1. Tipos de datos simples

Java, al tratarse de un lenguaje de alto nivel tiene una sintaxis fácilmente relacionable con el lenguaje humano.

Vamos a ver que tipos de datos vienen ya definidos en el programa y son directamente usables por el usuario. A estos tipos de datos les llamaremos primitivos.

Tipo de dato	Información	Rango
byte	Datos enteros	-128 – 127
short	Datos enteros	-32768 – 32767
int	Datos enteros	32 bits de longitud

long	Datos enteros	64 bits de longitud
char	Datos enteros y caracteres	0 – 65535
float	Datos en coma flotante de 32 bits	Precisión de 7 dígitos
double	Datos en coma flotante de 64 bits	Precisión de 16 dígitos
boolean	Valores booleanos	true/ false

¿Cómo se utilizan estos datos?

Tipo de dato	Código
byte	byte a;
short	short a=1;
int	int b = -3;
long	long c = 493843;
char	char caracter = 'a';
float	float pi = 3.1416;
double	double millon = 1e6;
boolean	boolean accion = false;

NOTA:

AL ACABAR CADA INSTRUCCIÓN EN JAVA, ESTA SIEMPRE DEBE FINALIZAR EN ;

SINO EL PROGRAMA FALLARÁ.

2. Constantes y variables

2.1 Constantes

Las constantes son datos que definiremos al inicio de un programa para poder usarlos siempre que queramos.

Debemos darles un valor al ser declarados, y este valor NO puede ser cambiado.

Aunque no es obligatorio, a las constantes las declararemos y daremos nombre usando sólo letras mayúsculas.

Las constantes se declararán siguiendo el siguiente formato:

```
final <tipo_de_dato> <NOMBRE_CONSTANTE> = <valor>;
```

```
final double PI = 3.141592;  
final double IVA = 0.21;
```

2.2 Variables

Las variables son, a nivel interno, zonas de memoria donde se puede almacenar información del tipo que desee el programador.

Las variables se declaran dentro de un bloque determinado (ya veremos que tipos de bloques existen), y sólo pueden usarse dentro de este.

Es decir, si tenemos 3 bloques (cada uno dentro del otro), las variables declaradas en el bloque exterior pueden usarse en los 3 bloques, pero las variables declaradas en el bloque interior no pueden usarse en los otros dos.

Aunque no es obligatorio, a las variables las declararemos y daremos nombre usando sólo letras minúsculas.

Las variables se declararán siguiendo el siguiente formato:

```
<tipo_de_dato> <nombre_variable> = <valor>;
```

```
int n1 =50;  
boolean prueba = false;
```

Siendo n1 y prueba variables que podrán ir cambiando de valor.

Nota:

Las variables de clase se inicializan por defecto (las numéricas a 0, y el resto a null;

Las variables locales no se inicializan por defecto, por lo que si no lo hacemos podemos producir un error.

Nota: EN JAVA, Y NORMALMENTE EN CASI TODOS LOS LENGUAJES DE PROGRAMACIÓN, EL = SE USA COMO ASIGNACIÓN, ES DECIR, PARA DARLE EL VALOR A UNA VARIABLE O UNA CONSTANTE. SI QUEREMOS COMPARAR DOS OBJETOS, SE USARÁ SIEMPRE ==

3. Comentarios y operadores

3.1 Comentarios

Los comentarios son líneas de código que el programa ignora (al compilar se las salta), por lo que podemos escribir lo que queramos en ellas.

Se utilizan para explicar los que se está programando e ir documentando el proceso para que sea legible y perdurable en el tiempo, además de para que, en un proyecto grande, cuando otra persona continúe con el código sepa cómo se ha trabajado con este.

En Java los comentarios pueden realizarse de dos formas.

- Si solo se quiere comentar una línea se usará “//”

```
// esto es un comentario
```

- Si quieren comentarse varias líneas, se abrirá el comentario con “/*” en la primera línea del comentario y se cerrará con “*/” en la última línea

```
/* esto es un comentario
```

```
De varias líneas
```

```
Para que sea un ejemplo */
```

3.2 Operadores aritméticos

Son utilizados para realizar operaciones matemáticas.

Operador	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (Resto)

3.3 Operadores relacionales

Se utilizan para evaluar la igualdad y la magnitud.

Devuelven verdadero o falso.

Operador	Operación
<	Menor
>	Mayor
<=	Menor o igual
>=	Mayor o igual
!=	Distinto
==	Igual

3.4 Operadores lógicos

Se utilizan para realizar operaciones lógicas.

Devuelven verdadero o falso.

Operador	Operación
&&	AND (y)
	OR (o)
!	NOT (contrario)

4. JDK e IDE

Para poder trabajar cómodamente vamos a usar una serie de herramientas que nos ayudaran a programar correctamente poco a poco.

Para ello necesitaremos usar un JDK y un IDE.

¿Qué es un JDK?

El JDK o Java Development Kit es un software que provee herramientas de desarrollo para la creación de programas en Java como puede ser el intérprete Java, clases Java, el compilador o el depurador entre otros.

¿Qué es un IDE?

Un IDE o entorno de desarrollo integrado (IDE) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar software en una aplicación fácil de usar.

En nuestro caso usaremos el IDE Visual Studio Code, aunque cada uno puede elegir el que más le guste.

4.1 Instalación del IDE

<https://code.visualstudio.com/download>

Iremos a la página oficial del Visual Studio Code y nos descargaremos la última versión para el sistema operativo que usemos.

Luego simplemente seguiremos los pasos de instalación.

4.2 Instalación del JDK

<https://www.oracle.com/es/java/technologies/downloads/>

Iremos a la página oficial de Oracle y nos descargaremos la última versión para el sistema operativo que usemos.

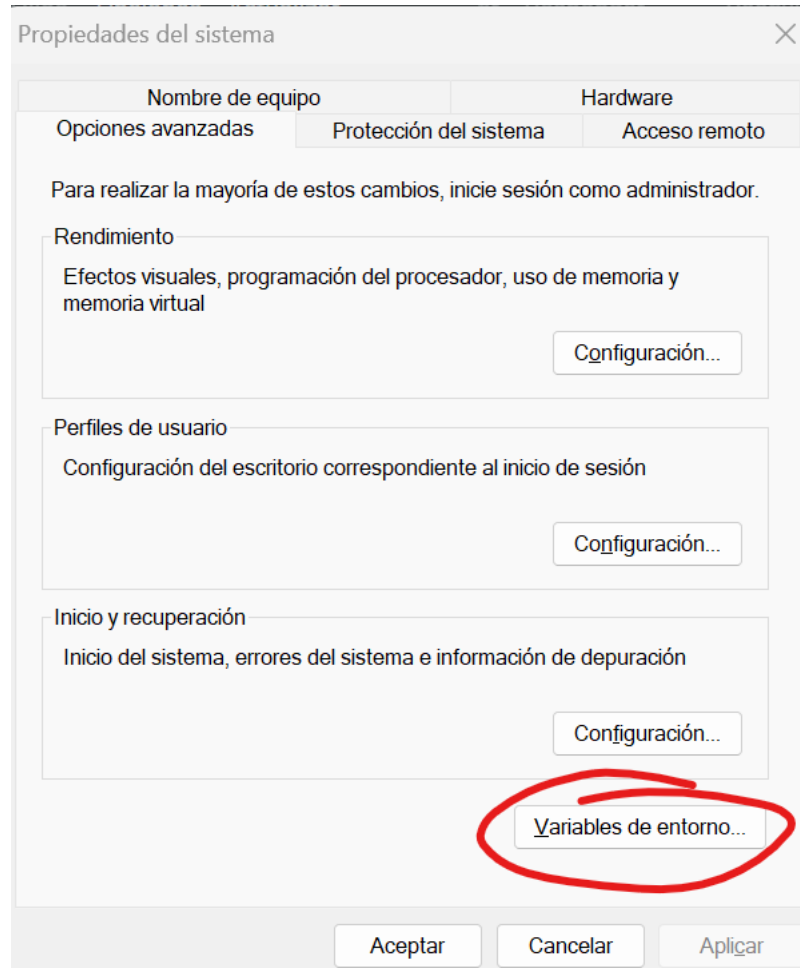
Luego simplemente seguiremos los pasos de instalación.

GUARDAREMOS LA RUTA DONDE HEMOS INSTALADO EL JDK

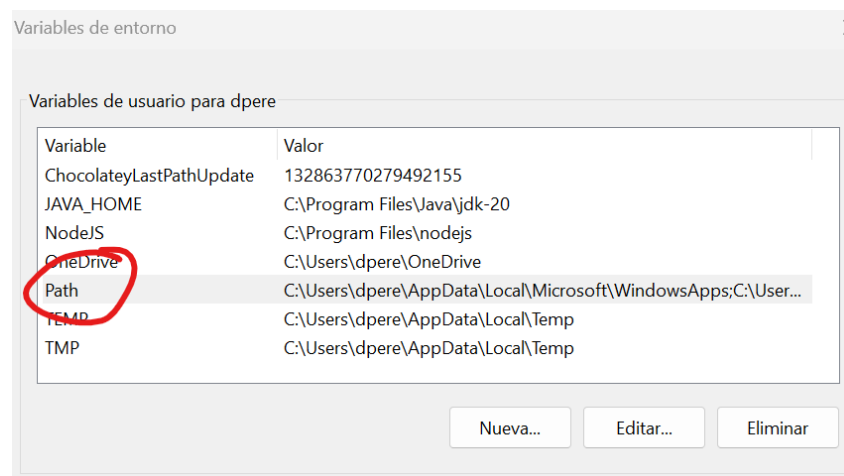
4.3 Variables de entorno

Una vez hemos instalado el JDK y el IDE que vamos a emplear vamos a realizar los cambios necesarios para el correcto funcionamiento de Java.

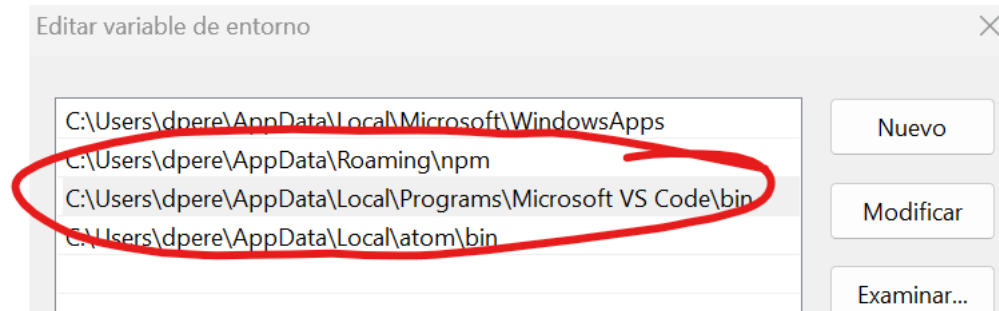
Abriremos las variables de entorno (Windons – escribimos “variables de entorno”)



En variables de usuario seleccionaremos la que se llama Path



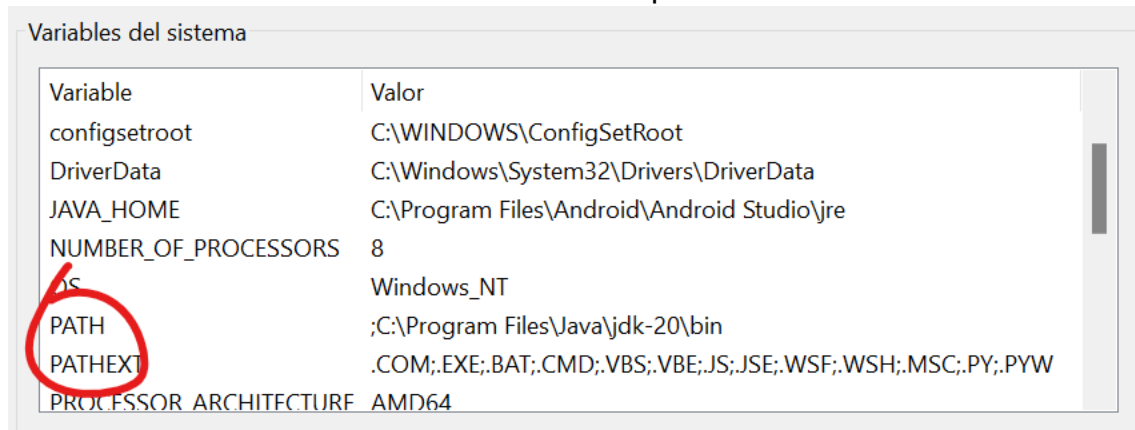
Y escribiremos la ruta donde se encuentra el archivo *bin* de nuestro Visual Studio:



En mi caso la ruta es:

C:\Users\dpere\AppData\Local\Programs\Microsoft VS Code\bin

En variables de sistema seleccionaremos la que se llama PATH



Y AÑADIREMOS, que no borrar, la ruta donde esta el *bin* de nuestro JDK, precedida de un ;

En mi caso será:

;C:\Program Files\Java\jdk-20\bin

Una vez hechos estos pasos, reiniciamos el ordenador y abrimos el cmd.

Escribimos en el cmd "javac – versión" y si obtenemos lo siguiente todo habrá salido correctamente.

```
C:\Users\dpere>javac -version
javac 20.0.2
```


4.4 Herramientas del IDE

Visual Studio no es un IDE por defecto, por lo que vamos a convertirlo en uno nosotros mismos.

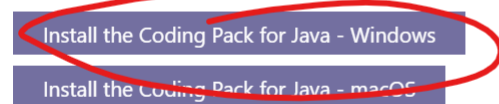
Simplemente deberemos ir a la página oficial de documentación de Visual Studio, y en el apartado JAVA encontraremos lo necesario.

<https://code.visualstudio.com/docs/java/java-tutorial>

Setting up VS Code for Java development

Coding Pack for Java

To help you set up quickly, you can install the **Coding Pack for Java**, which includes VS Code, the Java Development Kit (JDK), and essential Java extensions. The Coding Pack can be used as a clean installation, or to update or repair an existing development environment.



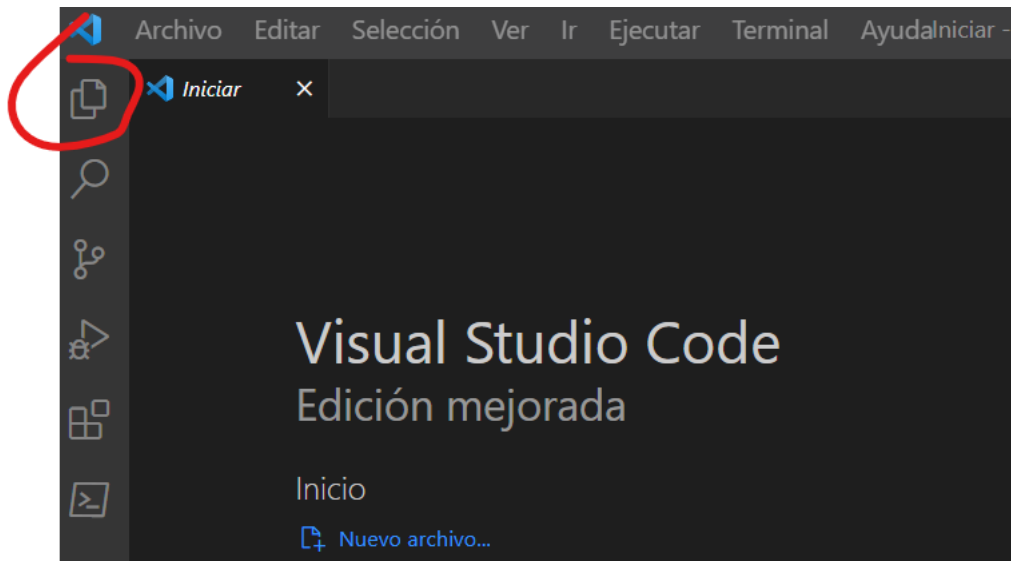
Note: The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions.

Nos descargamos el fichero y lo ejecutamos, siguiendo los pasos de instalación.

¡Con todo esto ya tenemos configurado nuestro IDE con el JDK correctamente, por lo que ya podemos empezar a programar!

5. Hola Mundo!

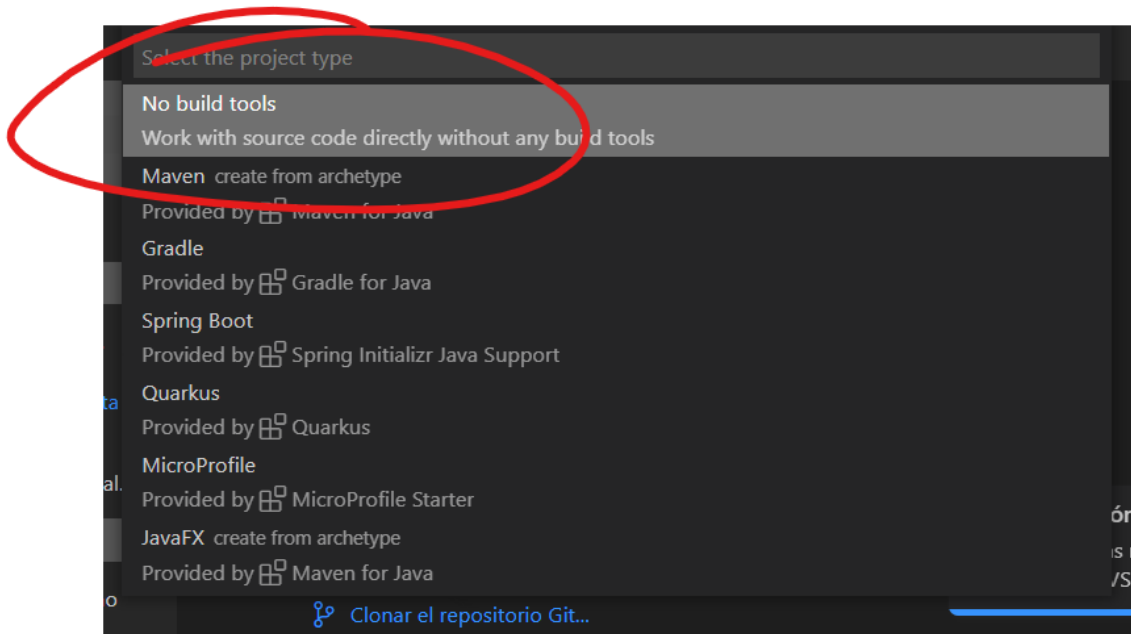
Para la creación de nuestro primer programa vamos a abrir el VSCode.



Seleccionaremos la opción "Create New Java Project".

Al haber instalado el pack de extensión, nos saldrá automáticamente en la barra central múltiples opciones.

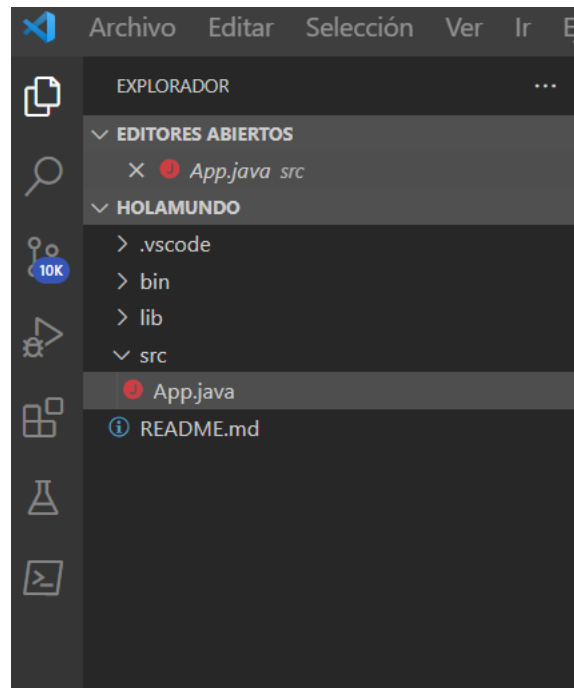
De momento seleccionaremos la primera "No build tools".



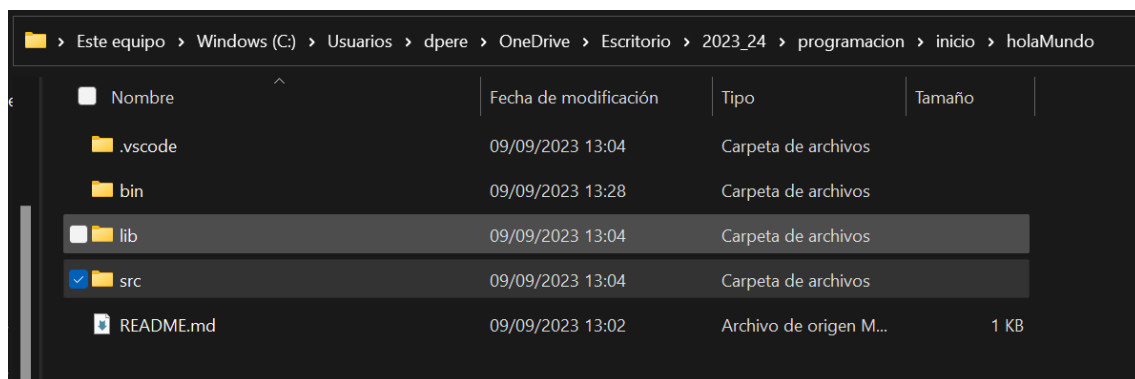
Nos preguntará donde queremos guardar el proyecto, así que elegiremos la carpeta que vayamos a usar para la asignatura y la seleccionaremos.

Después nos pedirá el nombre del proyecto, en este caso le pondremos HOLAMUNDO.

Una vez hecho esto podremos ver a la izquierda el explorador de archivos de nuestro proyecto HOLAMUNDO.



De momento nos vamos a centrar solo en la carpeta **src**, que es donde se encuentra el archivo **App.java**, con el que vamos a trabajar.



Si queremos ver los ficheros en el explorador en forma de carpetas podemos hacerlo pulsando click derecho "Mostrar en el explorador".

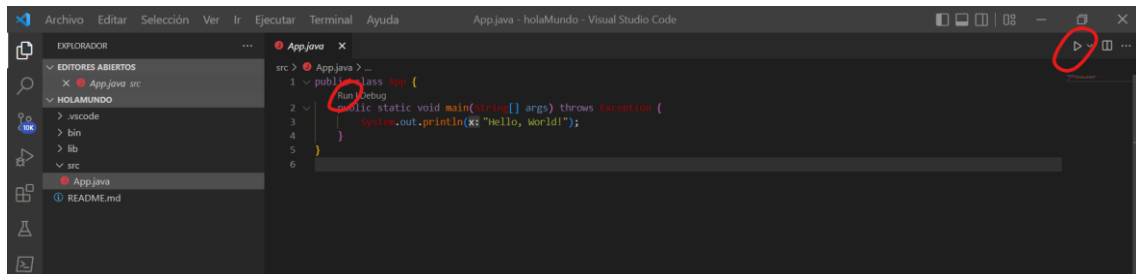
Si ahora en el VSCode pulsamos sobre App.java, se nos abrirá en el centro el archivo y podremos ver su código, el cual es el siguiente:

```
public class App {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello, World!");  
    }  
}
```

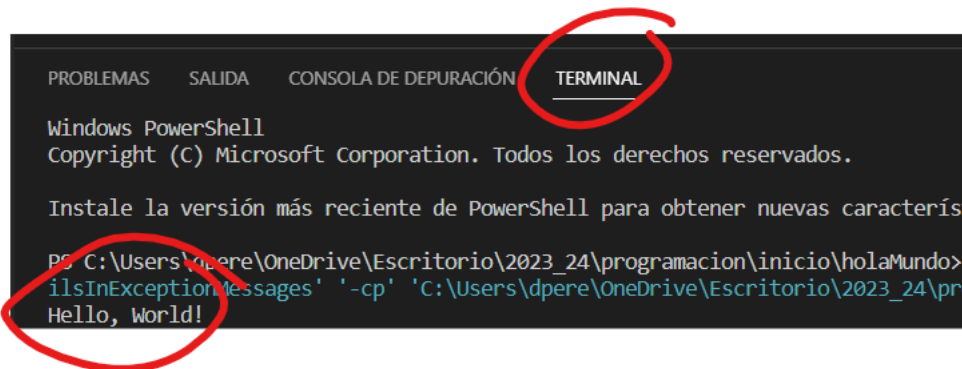
En las siguientes clases veremos que significa cada una de las palabras, métodos, clases, etc

De momento vamos a probar este código.

La instrucción `System.out.println()`; indica que se imprimirá o mostrará por pantalla (POR LA CONSOLA DE COMANDOS) lo que indiquemos dentro de los paréntesis.



Por tanto, si pulsamos el botón “Run” se ejecutará nuestro programa y veremos la salida.



Otra opción de poder ejecutar nuestro programa es hacerlo a mano, ya sea en la misma terminal del VSCode, o en la terminal de Windows.

Deberemos cambiar la ruta a donde se encuentre nuestro archivo App.java

```
cd .\src\
```

y después ejecutar la sentencia

```
java .\App.java
```