

**Ejercicios de la Unidad 02 - Instalación y Uso de ED****Modo consola**

1. Instala o revisa la instalación del JDK
  - ¿En qué carpeta queda instalado?
  - Captura una pantalla que muestre la variable el PATH del sistema
  - Opcional: Captura una pantalla que muestre CLASSPATH del sistema *Variable de entorno que indica a la Máquina Virtual de Java, , dónde buscar paquetes y clases definidas por el usuario a la hora de ejecutar programas*
2. Crea y ejecuta HolaMundo desde consola
  - Captura la pantalla que muestre la edición con el bloc de notas
  - Captura la pantalla que muestre la ejecución en cmd (CommandPrompt)
3. Crear un JAR **desde Netbeans** y ejecútalo desde consola
  - ¿En qué carpeta te genera Netbeans el JAR?
  - Captura la pantalla que muestre la ejecución en cmd

**Redireccionamiento:**

Se trata de redirigir (cambiar) las entradas y salidas de las órdenes del ms-dos. El siguiente cuadro presenta la entrada y la salida por defecto de las órdenes del dos.

Función	Dispositivo
Entrada por defecto	Teclado (CON)
Salida por defecto	Monitor (CON)

Para variar la entrada de una orden, la salida o ambas, se utilizan los operadores de redireccionamiento. Son cuatro:

Operador	Descripción	Sintaxis
<	Redirección de la entrada	orden <fichero/dispositivo
>	Redirección de la salida	orden >fichero/dispositivo
>>	Redirección de la salida a un fichero existente	orden >>fichero
	Redirección de la salida de una orden a entrada de otra orden	orden1  orden2

4. Ejecuta algún programa Java desde consola redireccionando la entrada y la salida

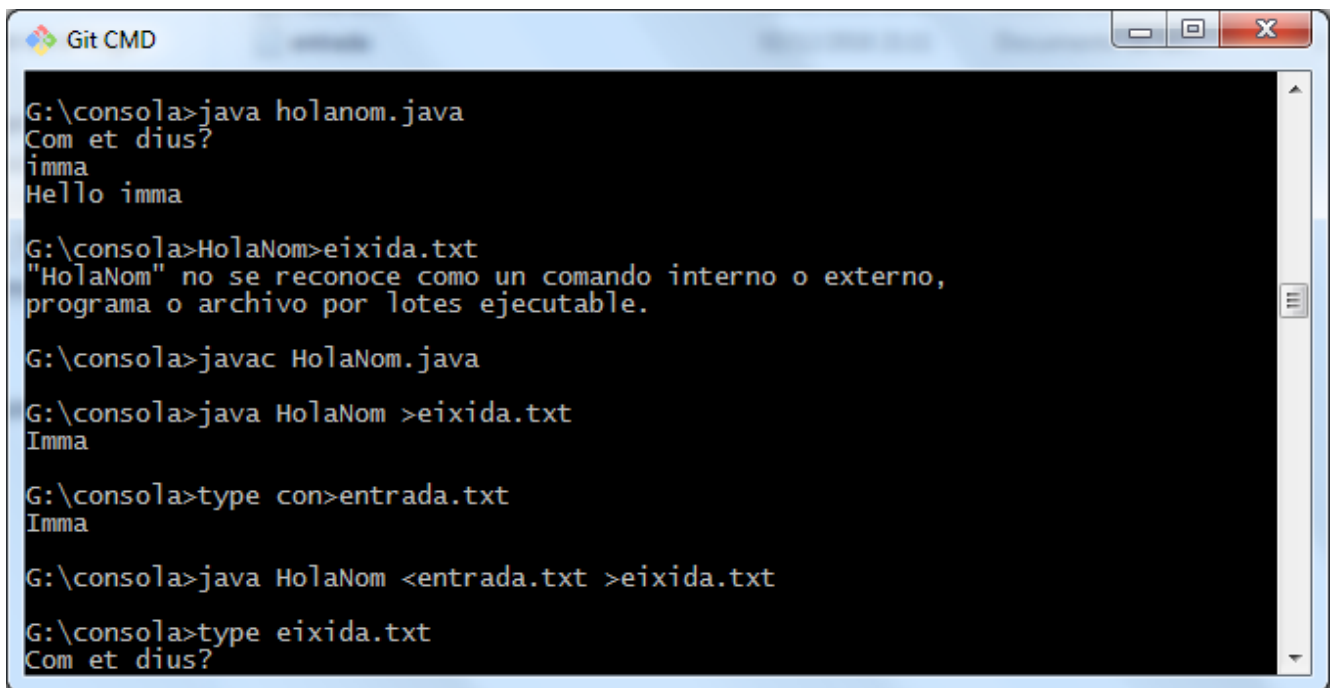
- Captura la pantalla que muestren la ejecución
- Captura las pantallas que muestren (type) el fichero de entrada (llámalo entrada.txt) y el de salida (llámalo salida.txt)

Por ejemplo:

```
import java.util.Scanner;

public class HolaNom {

    public static void main(String[] args) {
        Scanner lector = new Scanner(System.in);
        System.out.println("Com et dius?");
        String nom = lector.nextLine();
        System.out.println("Hello " + nom);
    }
}
```



```
Git CMD
G:\consola>java holanom.java
Com et dius?
imma
Hello imma

G:\consola>HolaNom>eixida.txt
"HolaNom" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

G:\consola>javac HolaNom.java

G:\consola>java HolaNom >eixida.txt
Imma

G:\consola>type con>entrada.txt
Imma

G:\consola>java HolaNom <entrada.txt >eixida.txt

G:\consola>type eixida.txt
Com et dius?
```

### Argumentos en línea de comandos

Uno de los posibles mecanismos para que un programa reciba los datos a procesar son los argumentos de línea de comandos. Normalmente se utilizan estos argumentos para especificar parámetros que regulen qué, cómo, etc. hace el programa, o para proporcionar datos de entrada de una forma distinta a como se hace mediante su entrada estándar.

El usuario proporciona estos argumentos en el momento de invocar el programa.

En Java, el programa los recibe como parámetro del método main, con tipo array de cadenas de texto, donde cada cadena del array contiene un argumento.

5.- Para practicar este concepto, guarda y compila el siguiente programa:

```
public class HelloName {

    public static void main(String[] args) {

        String name = args[0];

        System.out.println("Hello " + name);

    }

}
```

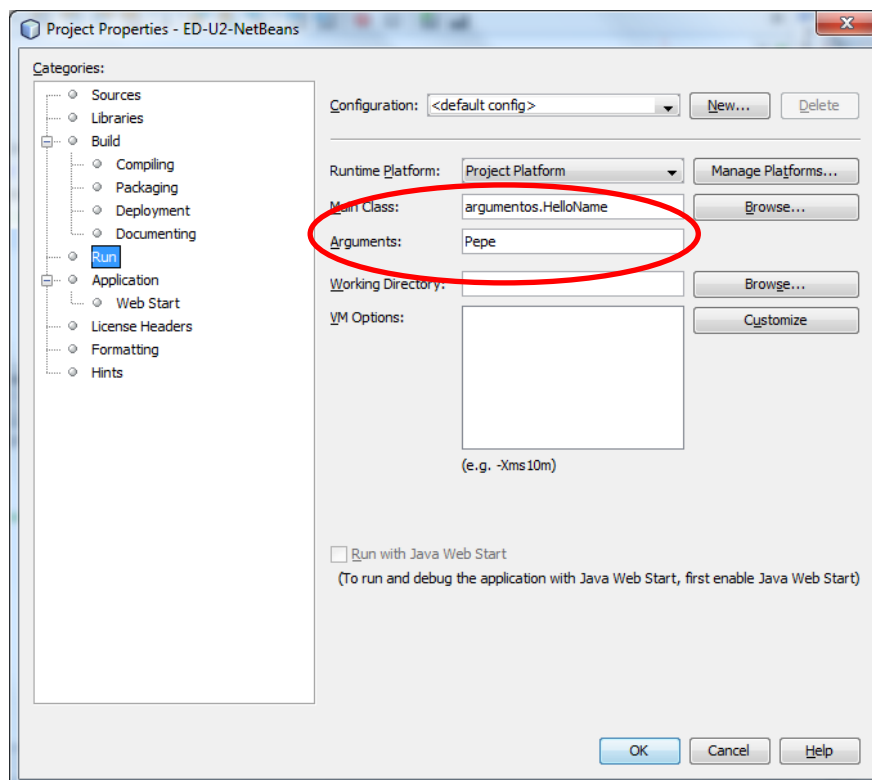
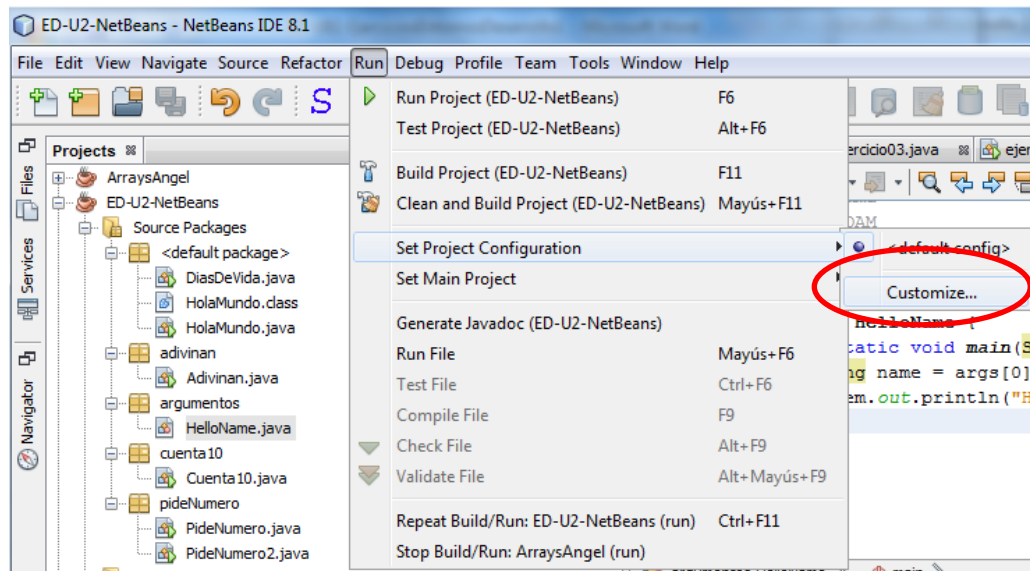
Los argumentos se pasan al programa al invocar su ejecución, justo a continuación del nombre de la clase (con una separación de al menos un espacio en blanco):

```
C:\Users\Alumno>java HelloName Juan
```

```
Hello Juan
```

5b.- Para ejecutar un programa que solicita argumentos, desde NetBeans:

Menu -> Run ->Set Project configuration -> Costumize



Y al ejecutar **el proyecto** (Run Project ó F6) asumirá el argumento “Pepe”

Atención: Observa las diferencias entre

- ejecutar el proyecto: Run Project ó F6
- ejecutar el fichero: Run File ó Mayús+F6

6.- También es posible **pasar más de un argumento** de línea de comandos separando cada uno del anterior mediante al menos un espacio en blanco. Guarda y compila el siguiente programa:

```
public class Adder {  
    public static void main(String[] args) {  
        int sum = 0;  
        for (int i = 0; i < args.length; i++) {  
            sum += Integer.parseInt(args[i]);  
        }  
        System.out.println("suma: "+sum);  
    }  
}
```

El programa recibe un número arbitrario de argumentos de línea de comandos, cada uno de los cuales debe ser un número entero.

Como los programas Java siempre recogen los argumentos de main() como cadenas de texto, se está utilizando Integer.parseInt() para obtener el entero que representa la cadena de texto dada.

Prueba a ejecutarlo pasándole varios números como argumentos. Por ejemplo:

```
C:\Users\Alumno>java Adder 2 -3 6 4
```

```
Suma: 9
```