### **UD1\_AC6 - Cuadernillo ejercicios básicos de Java**

| **RA1 - CE a,b,c,d,e,f,g,h,i**  **Objetivo**: familiarizarse con la estructura básica de un programa en Java y desarrollar competencias iniciales en el uso de variables, constantes, tipos primitivos, identificadores y comentarios, aplicando buenas prácticas de estilo y comenzando a utilizar operaciones de entrada/salida por consola. |
| --- |

**ENTREGAS:** Sube todos los archivos creados a tu repositorio de GitHub de esta tarea, con los nombres que aparecen en cada ejercicio

## **Bloque A. Programa principal y sintaxis básica**

**Ej01.java**

Escribe un programa mínimo en Java que muestre por pantalla tu nombre.

public class EJ01{

public static void main(String[] args) {

System.out.print("Marcos Hidalgo García");

}

}

**Ej02.java**

Modifica el programa anterior para mostrar tu nombre y tu edad en dos líneas distintas.

public class EJ02 {

public static void main(String[] args) {

System.out.println("Marcos Hidalgo García.");

System.out.print("18 años.");

}

}

**Ej03.java**

Hasta ahora hemos visto cómo mostrar mensajes por pantalla usando ***System.out.println***. Ahora aprenderemos a leer datos introducidos por el usuario con el teclado, usando la clase ***Scanner***. En Java podemos leer datos introducidos por el usuario usando la clase ***Scanner***, que se encuentra en el paquete ***java.util***. Para poder usarla, hay que importarla al inicio del programa:

***import java.util.Scanner;***

Luego se crea un objeto de tipo ***Scanner*** para leer desde el teclado ([***System.in***](http://system.in)):

***Scanner sc = new Scanner(System.in);***

Para leer un número entero, utilizamos el método ***nextInt():***

***int edad = sc.nextInt();***  
Tu tarea:  
1. Escribe un programa que pida al usuario su edad por teclado.  
2. Guarda ese valor en una variable entera.  
3. Muestra en pantalla un mensaje con el texto: Tienes *XX* años (donde *XX* es la edad introducida).

import java.util.Scanner;

public class EJ03 {

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

int edad = 0;

System.out.println("Introduce tu edad: ");

edad = sc.nextInt();

System.out.print("Tienes " + edad + " años.");

sc.close();

}

}

**Ej04.java**

Crea un programa que muestre por pantalla la suma de dos números enteros.

public class EJ04 {

public static void main(String[] args) {

int n1 = 2;

int n2 = 2;

int suma = 0;

suma = n1 + n2;

System.out.print("La suma de " + n1 + " + " + n2 + " es " + suma);

}

}

## **Bloque B. Identificadores y estilo**

**Ej05.txt**

De los siguientes nombres de variables, indica cuáles son válidos y cuáles no: **2edad**, **\_curso**, **while**, **Nombre**, **precioTotal**.

2edad - no puede comenzar en número.

\_curso - si.

while - no, al ser una funciona esta reservada esa palabra.

Nombre - si.

precioTotal - si.

**Ej06.java**

Escribe tres variables con nombres correctos siguiendo la notación **camelCase** y muestra sus valores.

public class EJ06 {

public static void main(String[] args) {

int ejemploVariable1 = 1;

int ejemploVariable2 = 2;

int ejemploVariable3 = 3;

System.out.print(ejemploVariable1 + ", " + ejemploVariable2 + ", " + ejemploVariable3);

}

}

**Ej07.java**

Escribe el nombre de una clase y de un método respetando la convención Java (PascalCase para clases, camelCase para métodos).

public class EJ07 {

public class MiEjemplo {

int miEjemplo = 1;

}

}

## **Bloque C. Variables y tipos primitivos**

**Ej08.java**

Declara e inicializa variables de tipo **int**, **double**, **char** y **boolean**, y muestra su valor por pantalla.

public class EJ08 {

public static void main(String[] args) {

int ejemplo1 = 1;

double ejemplo2 = 1.23;

boolean suspendido = true;

char ejemplo3 = 'M';

System.out.println("Entero: " + ejemplo1);

System.out.println("Doble: " + ejemplo2);

System.out.println("Suspendido: " + suspendido);

System.out.println("Inicial: " + ejemplo3);

}

}

**Ej09.java**

Declara una variable **int** sin inicializar y observa qué ocurre si intentas usarla en un **System**.**out.println**. Escríbelo en un comentario dentro del código.

public class EJ09 {

public static void main(String[] args) {

int ejemplo;

System.out.print(ejemplo);

//Mensaje de error: "The local variable ejemplo may not have been initialized"

//Si declaras una variable sin inicializarla, no puedes usarla

//porque el compilador no sabe qué valor tiene y lanza un error.

}

}

**Ej10.java**

Declara variables de tipo **byte** y **short**. Asígnales valores dentro de su rango y muestra el resultado.

public class EJ10 {

public static void main(String[] args) {

byte b = 125; //valor entre -128 a 127.

short s = 32700; //valor entre -32768 a 32767.

System.out.println("Byte: " + b);

System.out.print("Short: " + s);

}

}

**Ej11.java**

Declara una variable **long** y asígnale un número muy grande (ejemplo: la población mundial) y muestra el resultado.

public class EJ11 {

public static void main(String[] args) {

long poblacion = 800000000000L;

System.out.print(poblacion);

}

}

**Ej12.java**

Declara una variable **float** con el valor de π (3.14) y muestra el error si no añades la f.

public class EJ12 {

public static void main(String[] args) {

float ejemplo = 3.14f;

System.out.print(ejemplo);

}

}

## **D. Constantes**

**Ej13.java**

Declara una constante con el valor de IRPF (0.15) y úsala para calcular el sueldo neto de una nómina a partir de su sueldo bruto.

public class EJ13 {

public static void main(String[] args) {

final double IRPF = 0.14;

double sueldo = 2000;

double sueldoNeto = 0;

sueldoNeto = sueldo - (sueldo \* IRPF);

System.out.println("El sueldo neto es: " + sueldoNeto);

}

}

**Ej14.java**

Declara una constante con el número de días de la semana y úsala en un programa que lo muestre por pantalla: “La semana tiene X días”.

public class EJ14 {

public static void main(String[] args) {

final int diasdelasemana = 7;

System.out.print("La semana tiene " + diasdelasemana + " días.");

}

}

## 

## **E. Desbordamiento de memoria**

**Ej15.java**

Declara un **int** con el valor máximo (2147483647) y súmale 1. ¿Qué ocurre?. Escríbelo en un comentario dentro del código.

public class EJ15 {

public static void main(String[] args) {

int n = 2147483647;

int suma = 0;

suma = n + 1;

System.out.print(suma);

// Sumar 1 hace que pase al valor más pequeño (-2147483648) porque se pasa del límite que puede guardar un int de 32 bits.

}

}

**Ej16.java**

Declara una variable de tipo **short** con valor 30 000. Multiplícala por 10 y muestra el resultado. ¿Qué observas? Explica por qué ocurre en un comentario dentro del código.

public class EJ16{

public static void main(String[] args

int shorts n = 30000;

short resultado = n \* 10;

System.out.print(resultado);

// Multiplicar 30000 por 10 da un número raro porque supera el límite de un short (−32768 a 32767) y se produce desbordamiento.

}

}

**Ej17.txt**

Reflexiona: ¿por qué es importante elegir el tipo de dato adecuado según el rango de valores que necesitas manejar?

Es importante usar el tipo de dato correcto porque si es muy pequeño puede desbordarse al superar su límite, y si es demasiado grande desperdicias memoria.

## **F. Comentarios**

**Ej18.java**

Escribe un programa con el contenido que quieras, con al menos tres comentarios: uno de línea, uno multilínea y uno al inicio con tu nombre, la fecha y el propósito del programa (de cabecera).

/\*Nombre: Marcos

Fecha: 03/10/2025

Propósito: suma de números\*/

public class EJ18 {

public static void main(String[] args) {

// Comentario de línea: se declaran las variables de la suma

int n1 = 2;

int n2 = 1;

int resultado = 0;

;

/\*

Comentario multilínea:

sumamos 2 + 1

\*/

resultado = n1 + n2;

System.out.println("La suma de 2 + 1 es: " + resultado);

}

}

**Ej19.java**

Copia este código e incluye comentarios explicando qué hace cada línea:

int radio = 5;  
double area = 3.1416 \* radio \* radio;  
System.out.println(area);

public class ejercicio19 {

public static void main(String[] args) {

int radio = 5; // Declaramos una variable entera llamada 'radio' y le asignamos el valor 5  
 double area = 3.1416 \* radio \* radio;// Calculamos el área de un círculo usando la fórmula π \* radio2  y lo guardamos en 'area'

System.out.println(area); // Mostramos el valor del área en la consola

}

## **G. JavaDoc**

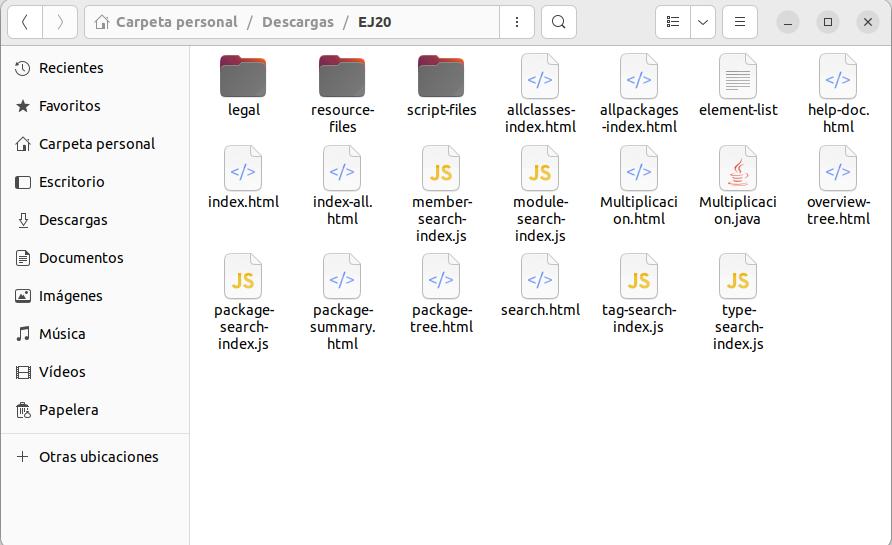
**Ej20.java**

Escribe un programa que muestre por pantalla el resultado de una multiplicación. Añade un comentario JavaDoc a la clase con:

* una breve descripción del programa
* las etiquetas **@author** y **@version**

Genera la documentación con **javadoc** y revisa el archivo HTML resultante.

Resultado de lo generado con javadoc:



Al entrar en el index me sale lo siguiente:

