

RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
Criterios de evaluación:

- Se han identificado los bloques que componen la estructura de un programa informático.
- Se han creado proyectos de desarrollo de aplicaciones.
- Se han utilizado entornos integrados de desarrollo.
- Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- Se ha modificado el código de un programa para crear y utilizar variables.
- Se han creado y utilizado constantes y literales.
- Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
- Se han introducido comentarios en el código.

(2,5 puntos) Ejercicio 1: Un economista te ha encargado un programa para realizar cálculo con el IVA. La aplicación debe solicitar la base imponible y el IVA que se debe aplicar.

Muestra en pantalla el importe correspondiente al IVA y al total.

Nota: IVA = 21.0 ; Importe IVA es igual a base imponible por IVA / 100; Total es igual a base imponible más importe IVA

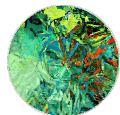
```
package examen.ra1;
import java.util.Scanner;

4 public class Ejercicio1 {
5     public static void main(String[] args) {
6         // 1) Entrada/Salida con Scanner (bloque de E/S)
7         Scanner sc = new Scanner(System.in);
8
9         // 2) Constantes y variables
10        // - CONSTANTE: valor que no cambia. Usamos mayúsculas por convención
11        final double IVA = 0.21;
12        // - VARIABLES: guardan datos que sí pueden cambiar.
13        double baseImponible;
14
15        System.out.print("Introduce la base imponible (€): ");
16        baseImponible = sc.nextDouble(); // tipo primitivo double
17
18        // 3) Cálculo
19        double importeIva = baseImponible * IVA;
20        double total = baseImponible + importeIva;
21
22        // 4) Salida con formato
23        System.out.println("Base imponible: " + baseImponible + " €");
24        System.out.println("TOTAL: " + total + " €");
25
26        sc.close(); // buena práctica
27    }
28 }
```

(2,5 puntos) Ejercicio 2: Sin usar condicionales. Pide dos números al usuario: a y b. Deberá mostrarse **true** si ambos números son iguales y **false** en caso contrario.

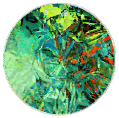
```
package examen.ra1;
import java.util.Scanner;

4 public class Ejercicio2 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         // Variables. (también podría ser double; aquí usamos int para claridad)
8         int a, b;
9
10        // Leemos dos enteros
11        System.out.print("Introduce a: ");
12        a = sc.nextInt();
13        System.out.print("Introduce b: ");
14        b = sc.nextInt();
15
16        // Sin condicionales: el operador == devuelve un boolean directamente
17        boolean sonIguales = (a == b); // true si a y b tienen el mismo valor
18
19        // Mostramos el resultado booleano tal cual pide el enunciado
20        System.out.println(sonIguales);
21
22        sc.close();
23    }
24 }
25 }
```



(5 puntos) Ejercicio 3: Selecciona las respuestas correctas

<p>1.1. ¿Cuál de los siguientes identificadores no puede emplearse para una variable?</p> <p>a) lenguaje b) \$firma c) int d) final</p>	<p>1.2. De todos los tipos primitivos disponibles en Java, selecciona cuál o cuáles son los que tienen un mayor tamaño y, por lo tanto, pueden albergar un mayor número de valores:</p> <p>a) long b) long y double c) long y float d) En Java todos los tipos primitivos tienen el mismo tamaño</p>
<p>1.3. Mediante qué símbolos es posible añadir un comentario en nuestro código:</p> <p>a) // b) /* */ c) ++ d) Cualquiera de los anteriores</p>	<p>1.4. ¿Qué puede usarse de importe automáticamente en cualquier programa sin necesidad de tener que utilizar una sentencia import?</p> <p>a) java.util b) java.lang c) java.swing d) javax.sql</p>
<p>1.5. ¿Cuáles de las siguientes instrucciones nos permiten mostrar información por consola?</p> <p>a) System() b) System.out() c) System.out.print() d) System.out.println()</p>	<p>1.6. ¿Qué instrucción es equivalente a: $i = i + 1$?</p> <p>a) $i += 1$; b) $++i$; c) $i++$; d) Cualquiera de los anteriores</p>
<p>1.7. Si evalúas la siguiente expresión: $z < 1 \ \ z++ == 1 \ \&\& \ z == 1$ el resultado de dicha expresión es:</p> <p>a) true b) false (Nota: Para $z \rightarrow 2$) c) 0 d) Ninguna de las anteriores</p>	<p>1.8. ¿Qué valor toma la variable a, tras la ejecución de la instrucción: $\text{int } a = 1 < 2 ? 3 : 4$;</p> <p>a) 1 b) 2 c) 3 d) 4</p>
<p>1.9. Selecciona la expresión cuya evaluación resulta 3:</p> <p>a) $3 * 2 / 5$ b) $3 + 2 * 6 / 5$ c) $(3 + 2) * 6 / 5$ d) $(3 + 2 * 6) / 5$</p>	<p>1.10. En las siguientes conversiones de tipo, ¿cuál de ellas produce un error?</p> <p>a) $\text{int } a = (\text{int}) 1.23$; b) $\text{int } a = 12.3$; c) $\text{double } a = (\text{double}) 123$; d) $\text{double } a = 123$;</p>



En varias preguntas hay **más de una correcta**. Se indican la razón:

1.1. Identificadores no válidos

- ☒ c) `int` y d) `final` (palabras reservadas; no pueden ser nombres de variables).

1.2. Tipos primitivos con mayor tamaño

- ☒ b) `long` y `double`

`long` (entero 64 bits) y `double` (coma flotante 64 bits).

1.3. Símbolos para comentarios

- ☒ a) `//` y b) `/**`

1.4. Paquete importado automáticamente

- ☒ b) `java.lang` Se importa por defecto (`String`, `Math`, `System`...).

1.5. Mostrar por consola

- ☒ c) `System.out.print()` y d) `System.out.println()`

`System()` y `System.out()` solos no imprimen.

1.6. Equivalente a `i = i + 1;`

- ☒ d) Cualquiera de los anteriores

`i += 1;`, `++i;` e `i++;` incrementan 1. (Ojo: el valor **devuelto** por `++i/i++` difiere, pero como sentencia aislada, todas incrementan en 1).

1.7. Expresión: `z < 1 || z++ == 1 && z == 1`

- 👉 Si `z = 2`, resulta **false**.

`z < 1 || z++ == 1 && z == 1`

`2 < 1 || 3 == 1 && 2 == 1`

`false || false && false`

1.8. Ternaria

`1 < 2 ? 3 : 4` → condición verdadera → **3**

- ☒ c) **3**

1.9. Expresión que vale 3 (enteros)

- ☒ d) `(3 + 2 * 6) / 5`

1.10. Conversión que da error

a) `int a = (int) 1.23;` ☒ (cast explícito)

b) `int a = 12.3;` ☒ (falta cast)

c) `double a = (double) 123;` ☒ (innecesario pero válido)

d) `double a = 123;` ☒ (ampliación implícita)

- ☒ b) `int a = 12.3;`