



Prueba escrita 9 de Octubre de 2025

Unidad 2: Condicionales

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

- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- e) Se ha modificado el código de un programa para crear y utilizar variables.
- i) Se han introducido comentarios en el código.

RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

- b) Se ha identificado la sintaxis del lenguaje de programación.

RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

- a) Se ha escrito y probado código que haga uso de estructuras de selección.
- e) Se han creado programas ejecutables utilizando diferentes estructuras de control.

RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

- a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.



1. Posible solución

(2,5 puntos) Ejercicio 1: Idear un programa que solicite al usuario un número comprendido entre 1 y 7, correspondiente a un día de la semana. Se debe mostrar el nombre del día de la semana al que corresponde. Por ejemplo, el número 1 corresponde a «lunes» y el 6 a «sábado».

```
package examen;
import java.util.Scanner;
public class Ejercicio1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Solicita un número del 1 al 7 y muestra el día correspondiente
        System.out.print("Introduce un número (1-7): ");
        int n = sc.nextInt();
        String dia;
        switch (n) {
            case 1 -> dia = "lunes";
            case 2 -> dia = "martes";
            case 3 -> dia = "miércoles";
            case 4 -> dia = "jueves";
            case 5 -> dia = "viernes";
            case 6 -> dia = "sábado";
            case 7 -> dia = "domingo";
            default -> dia = "Valor fuera de rango (debe ser 1-7)";
        }
        System.out.println(dia);
        sc.close();
    }
}
```

(2,5 puntos) Escribe un programa en Java que solicite al usuario su **edad** y determine si puede **entrar a una sala de cine**.

- Si tiene **menos de 13 años**, no puede entrar solo.
- Si tiene entre **13 y 17 años**, puede entrar acompañado.
- Si tiene **18 o más**, puede entrar sin restricciones.

El programa debe mostrar un **mensaje personalizado** según el caso.

Debes usar:

- **Una constante literal de texto** para el nombre del cine.
- **Variables** para almacenar la edad y el mensaje.
- **Estructuras condicionales** para decidir la salida.

```
package examen;
import java.util.Scanner;  public class Ejercicio2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Variables
        // Nombre del cine
        final String CINE = "Cine Estrella";
```



```
String mensaje;
int edad;
System.out.print("Introduce tu edad: ");
edad= sc.nextInt();

if (edad < 0) {
    mensaje = "Edad no válida.";
} else if (edad < 13) {
    // Menores de 13 años deben venir con su representante legal.
    mensaje = "No puedes entrar solo.";
} else if (edad <= 17) {
    // Entre 13 y 17 años pueden entrar acompañados (por un amigo
    o adulto).
    mensaje = "Puedes entrar acompañado.";
} else {
    // Mayores de edad: sin restricciones.
    mensaje = "Puedes entrar sin restricciones.";
}

System.out.println("[ " + CINE + " ] " + mensaje);
sc.close();
}
```



(5 puntos) Ejercicio 3: Selecciona la respuesta correcta

<p>2.1. Los operadores lógicos operan con valores booleanos, resultando:</p> <ul style="list-style-type: none">a) Valores enteros.b) Valores enteros y booleanos.c) Otros tipos de valores.d) Sólo valores booleanos.	<p>2.2. La evaluación de una expresión relacional puede generar un valor de tipo:</p> <ul style="list-style-type: none">a) Entero.b) Real.c) Booleano.d) Todos los anteriores.
<p>2.3. La expresión</p> <pre>3 == 3 && 2 < 3 && 1 != 2</pre> <p>resulta:</p> <ul style="list-style-type: none">a) Cierto.b) Falso.c) No se puede evaluar.d) No genera un booleano, ya que la expresión es aritmética.	<p>2.4. La siguiente expresión, donde interviene la variable booleana a:</p> <pre>3 != 3 a 1 < 2</pre> <p>resulta:</p> <ul style="list-style-type: none">a) Dependerá del valor a.b) Cierto.c) Falso.d) No se puede evaluar.
<p>2.5. Elige los valores de las variables enteras (a, b y c) que permiten que la evaluación de la siguiente expresión sea cierta:</p> <pre>a < b && b != c && b <= c</pre> <ul style="list-style-type: none">a) a = 1, b = 1, c = 2.b) a = 1, b = 2, c = 2.c) a = 1, b = 2, c = 3.d) a = 2, b = 2, c = 3.	<p>2.6. El bloque de instrucciones de una sentencia if se ejecutará:</p> <ul style="list-style-type: none">a) Siempre.b) Nunca.c) Dependerá de la evaluación de la expresión utilizada.d) Todas las respuestas anteriores son correctas.
<p>2.7. En una sentencia if-else los bloques de instrucciones (bloque true y bloque false) pueden ejecutarse:</p> <ul style="list-style-type: none">a) Simultáneamente.b) Siempre.c) Si es posible, dependiendo de la condición utilizada, que no se ejecute ninguno.d) Sólo se ejecuta al menos uno y son excluyentes.	<p>2.8. Realiza una traza del siguiente fragmento de código y selecciona el valor que toma finalmente la variable a:</p> <pre>a = 0; switch (a) { case 0: a = 1; case 1: a = 3; case 2: a++; break; case 3: a--; break; }</pre> <ul style="list-style-type: none">a) 1b) 2c) 3d) 4
<p>2.9. La cláusula default en la sentencia switch es:</p> <ul style="list-style-type: none">a) Obligatoria y tiene que ser la última que aparezca.b) Obligatoria, pero puede aparecer en cualquier lugar.c) Opcional y tiene que ser la última que aparezca.d) Opcional y puede usarse en cualquier lugar.	



C.F.G.S. Desarrollo de Aplicaciones Web/Multiplataforma

MÓDULO: Programación

IES CAMAS Curso 2025/26



2. Notas

	2'5 puntos							2'5 puntos							5 puntos							Total		
	Ejercicio 1							Ejercicio 2							Ejercicio 3									
	a	b	c	d	e	f	g	a	b	c	d	e	f	g	1	2	3	4	5	6	7	8	9	
ialcrod1802	1	1	0,5	1	1	1	0	1	1	0,5	1	1	1	0	1	0	1	0	1	1	1	0	1	7,6
jblagon1804	0,5	0,75	0,5	0,5	0,5	0,5	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1	0	4,6
jcalgar480	1	0,75	1	1	1	1	0	1	0,75	1	1	1	1	0	1	0	1	0	1	1	1	0	0	7,0
lcascan0310																								NP
acobfra1611	1	0,75	0,5	1	1	1	1	1	0,75	0,5	1	1	1	1	1	1	1	0	1	1	1	1	0	8,4
mdiajim407	1	0,75	0,5	1	1	1	0	1	0,75	0,5	1	1	1	0	1	0	1	0	1	1	1	0	0	6,8
adorcum696	0,5	0,5	0,5	0,5	0,5	0,5	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	2,8
rfrabar2206	1	0,75	0,5	1	1	1	1	0,75	0,75	0,5	1	1	1	1	0	0	1	1	1	0	1	0	0	6,5
ggarlob532	0,5	0,75	0,5	1	1	1	1	0,5	0,75	0,5	1	1	1	1	1	0	1	0	1	1	1	1	1	7,6
sherace1405	1	0,75	0,5	1	1	1	0	1	0,75	0,5	1	1	1	0	0	0	1	1	1	1	1	0	1	7,3
mhuecob081	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	7,8
gllepen645	0	0	0,5	1	1	1	0	0	0	0,5	1	1	1	0	1	1	1	1	1	1	1	0	0	5,6
amacpac0204	0,5	0,5	0	1	0,75	0,5	0	0,5	0,5	0	1	0,75	0,5	0	1	0	1	0	1	1	1	0	0	5,2
amarloz3112	1	0,75	0,5	1	1	1	0	1	0,75	0,5	1	1	1	0	0	0	1	0	1	1	0	0	0	5,7
amarboc																								NP
rrodsan855	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	2,8
asanesp2907	0	0,5	0,5	1	1	1	1	0	0,5	0,5	1	1	1	1	0	0	1	1	1	1	1	0	0	5,5
jsersan1204	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	1	1	0	0	0	7,3
msieuafa855	1	1	1	1	1	1	0	1	0,75	0,5	1	1	1	0	1	0	1	1	1	1	0	0	0	7,0
ptormar1403	1	0,5	0,5	1	1	1	0	1	1	0,5	1	1	1	0	1	1	1	1	1	1	1	0	0	7,9



sfad

3. Comentarios

<p>RA1.i). -> Hay que comentar el código RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo.</p>	ialcrod1802
<p>RA1.i). -> Hay que comentar el código. RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo. 0.5 RA3. e:). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código</p>	jblagon1804
<p>RA1.i). -> Hay que comentar el código. RA3. e:). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código</p>	jcalgar480
	lcascan0310
<p>RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo. RA3. e). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la</p>	acobfra1611



claridad y mantenimiento del código	
RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo. RA3. e). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código RA1.i). -> Hay que comentar el código.	mdiajim407
RA5.a) → En el ejercicio 1 no lees desde teclado, y no finaliza imprimiendo el resultado. RA1.i). -> Hay que comentar el código.	adorcum696
RA2.b) ->Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo RA3. e:). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código.	rfrabar2206
RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo. RA3. e). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código En el ejercicio 1, dentro del switch, faltan las sentencias break ; en los distintos case. Esto puede provocar que el flujo de ejecución continúe hacia el siguiente caso de forma no intencionada Lo correcto sería	ggarlob532



```
switch (dia) {  
    case 1:  
        mensajeDiaSemana="Lunes";  
        break;  
    case 2:  
        mensajeDiaSemana ="Martes";  
        break;  
    ...  
    default:  
        mensajeDiaSemana ="Valor fuera de rango";  
    }  
System.out.println(mensajeDiaSemana)
```

RA2.b) ->Falta el identificador del nombre de la clase. En Java, tras la palabra clave `class` debe ir un nombre válido (identificador), y si la clase es `public` debe coincidir con el nombre del archivo

RA3. e).
-> Se sugiere guardar el texto en una variable intermedia (por ejemplo, `mensaje`) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código.

RA1.i).
-> Hay que comentar el código

sherace140
5

RA1.i).
-> Hay que comentar el código

mhuecob08
1

RA2.b) RA3.a RA3.e)

-> Has puesto:

```
if (n > 1) && (n<=7)  
    Switch(n) {
```

- Evita redundancia en el código:

Usa solo una estructura de selección para la misma lógica: o bien `if/else`, o bien `switch`, pero no ambas. Esto mejora la claridad,

O usas una estructura u otra

RA2.b)

Falta el identificador del nombre de la clase. En Java, tras la palabra clave `class` debe ir un nombre válido (identificador), y si la clase es `public` debe coincidir con el nombre del archivo.

gllepen645



RA1.i). -> Hay que comentar el código.	
RA1.i). -> Hay que comentar el código RA2.b) Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo. RA2.b)	amacpac02 04
Has escrito: <pre>... }else(num 3){ String.println("es martes");</pre>	
No es correcto. Recuerda que la condición sólo puede evaluarse como “cierta” o “falsa”.	
Si no se cumple, el programa puede evaluar otra condición mediante else if, y si ninguna se cumple, ejecutará el bloque else <pre>if(<condición>){ } }else if(<condición>){ } }else{ } }</pre>	
has escrito <pre>case 1: a>13 a <=17 ...</pre>	
No es correcto. Usa solo una estructura de selección para la misma lógica: o bien if/else, o bien switch, pero no ambas. Esto mejora la claridad, Lo correcto sería <pre>switch (dia) { case 1: sys("Lunes"); break; case 2: sys("Martes"); break; default: sys("Valor fuera de rango"); }</pre>	



RA2.b)

Falta el identificador del nombre de la clase. En Java, tras la palabra clave `class` debe ir un nombre válido (identificador), y si la clase es `public` debe coincidir con el nombre del archivo.

RA1.i).

-> Hay que comentar el código

RA3. e)

-> Se sugiere guardar el texto en una variable intermedia (por ejemplo, `mensaje`) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código.

amarloz311

2

amarboc

RA2.b) ->Falta el identificador del nombre de la clase. En Java, tras la palabra clave `class` debe ir un nombre válido (identificador), y si la clase es `public` debe coincidir con el nombre del archivo.

RA3.a) -> Recuerda <condición> solo puede ser 'true' o 'false'.

```
if(<condición>){  
    }else if(<condicion>){  
    }else{  
    }
```

rrodsan855

RA3. e). -> Se sugiere guardar el texto en una variable intermedia (por ejemplo, `mensaje`) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código

RA2.b) ->Falta el identificador del nombre de la clase. En Java, tras la palabra clave `class` debe ir un nombre válido (identificador), y si la clase es `public` debe coincidir con el nombre del archivo.

asanesp290

7

RA3. e).

-> Se sugiere guardar el texto en una variable intermedia (por ejemplo, `mensaje`) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código

Has escrito

```
if { edad < 13 ){  
    ...
```



```
if { (edad > 13 && edad<17);
```

No es correcto-

Recuerda que la **condición** sólo puede evaluarse como “cierta” o “falsa”.

Si no se cumple, el programa puede evaluar otra condición mediante else if, y si ninguna se cumple, ejecutará el bloque else

```
if(<condición>){
```

```
}else if(<condición>){
```

```
}else{
```

```
}
```

RA1.i).

-> Hay que comentar el código

jsersan1204

RA3.a) RA2.b) ->

has escrito

```
if(switch (dia))  
..  
case 1 (sys0 ("Lunes"));  
case 2 (sys0("Martes"));
```

No es correcto.

Lo correcto sería

```
switch (dia) {  
case 1:  
    sys0("Lunes");  
    break;  
case 2:  
    sys0("Martes");  
    break;  
default:  
    sys0("Valor fuera de rango");  
}
```

msieufa855

Has escrito

```
if { edad < 13 ;  
...  
else if { (edad > 13);
```

No es correcto-

Recuerda que la **condición** sólo puede evaluarse como “cierta” o “falsa”.

Si no se cumple, el programa puede evaluar otra condición mediante else if,



y si ninguna se cumple, ejecutará el bloque else

```
if(<condición>){  
}  
}else if(<condición>){  
}  
}else{  
}  
}
```

RA3. e).

-> Se sugiere guardar el texto en una variable intermedia (por ejemplo, mensaje) y mostrarlo al final, en un único punto de salida, para mejorar la claridad y mantenimiento del código.

RA1.i).

-> Hay que comentar el código

RA2.b) -> Falta el identificador del nombre de la clase. En Java, tras la palabra clave class debe ir un nombre válido (identificador), y si la clase es public debe coincidir con el nombre del archivo.

RA3.a) RA2.b) ->

has escrito

```
int numDia = sc.nextInt();  
switch(numDia){  
    case 1 ->"Lunes";  
    case 2 ->"Martes";  
    ...  
    default -> "Error";  
};
```

No es correcto porque no lo guardas en ninguna variable.

Lo correcto sería

```
int numDia = sc.nextInt();  
String nombreDia = switch (numDia) {  
    case 1 -> "Lunes";  
    case 2 -> "Martes";  
    ...  
    default -> "Valor fuera de rango";  
};
```

ptormar140

3

RA1.i).

-> Hay que comentar el código



Ponderaciones

Reparto de 2,5 puntos (ajustado proporcionalmente):	Peso Ejercicio 1 y 2		Escala de logro		
			1	Excelente	Cumple totalmente el criterio con precisión y claridad.
RA3.a: Se ha escrito y probado código que haga uso de estructuras de selección.	0,75	a	0,75	Adequado	Cumple de forma correcta, con mínimos errores.
RA3.e:) Se han creado programas ejecutables utilizando diferentes estructuras de control.	0,5	b	0,5	Básico	Cumple parcialmente o con fallos leves.
RA2.b) Se ha identificado la sintaxis del lenguaje de programación	0,25	c	0	Insuficiente	No alcanza el criterio o presenta errores graves.
RA5.a) Se ha utilizado la consola para realizar operaciones de entrada y salida de información.	0,25	d			
RA1.d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.	0,25	e			
RA1.e) Se ha modificado el código de un programa para crear y utilizar variables.	0,25	f			
RA1.i) Se han introducido comentarios en el código.	0,25	g			