

Apartado 2.1

“En esta captura se pueden ver los errores sintácticos”

The screenshot shows the Eclipse IDE interface. On the left, the code editor displays a Java file named 'PiramideSumas.java'. The code contains several syntax errors, indicated by red squiggly underlines. The right side of the interface shows the 'Problems' view, which lists 6 warnings and 0 errors. The status bar at the bottom indicates 'Vie 31 oct 14:09'.

```
4
5  blic class PiramideSumas {
6
7  public static void main(String[] args) {
8      Scanne sc = new Scanne(System.in);
9      String resp;
10
11     do {
12         int num = leerEntero(sc, "Introduzca un número: "); // RGC20251031 -
13         while (num < 0 && num > 20) {
14             num = leerEntero(sc, "**Valor fuera de rango** Introduzca un núm
15         }
16
17         System.out.println("\nSu pirámide de sumas es la siguiente:\n" + pir
18
19         System.out.print("¿Quiere hacer otra pirámide? (s/n) ");
20         resp = sc.next().trim().toUpperCase(); // RGC20251031 - Duplicación
21
22     } while (resp.equals("S")); // RGC20251031 - Error sintáctico. " Falta
23     borrarConsola();
24     System.out.println("||||PROGRAMA FINALIZADO!!!!");
25
26     sc.close();
27 }
28
29 private static int leerEntero(Scanner sc, String mensaje) {
30     System.out.print(mensaje);
31     while (!sc.hasNextInt()) {
32         System.out.print("**Valor no válido** Introduzca un número entero: "
33         sc.nextInt();
34     }
35     return sc.nextInt();
36 }
37
38 public static void borrarConsola() {
39     for (int i = 0; i < 50; i++) { // RGC20251031 - es una , en vez de ;
40         System.out.println();
41     }
}
```

“Despues de haber corregido los errores sintácticos”

Apartado 2.2

- ¿Qué es un error sintáctico?
- Un error sintáctico es un fallo gramatical de un lenguaje de programación o de un comando; estos errores hacen que no sea posible ejecutar o compilar el programa.
- ¿Cuándo los visualizamos?

El IDE nos muestra a través de una señal de peligro en la línea de código donde nos podemos encontrar el error sintáctico. Asimismo, nos muestra en concreto en qué parte de la línea está el error, que muchas veces es que falta un ; o un).

- ¿Podemos depurar con errores de sintaxis?

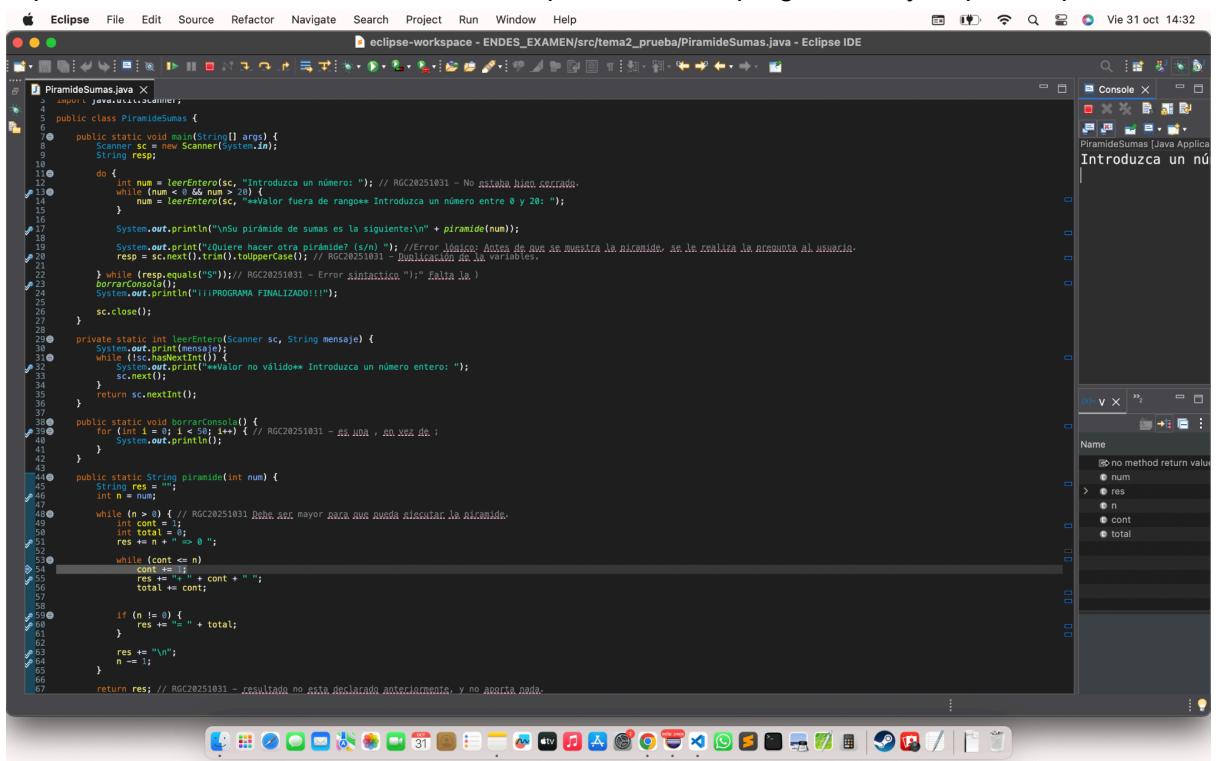
Sí, aunque la mayoría de las veces estos errores se corrigen en la codificación, puesto que, como hemos mencionado anteriormente, son visibles para el programador a diferencia de los errores lógicos.

Apartado 2.3

- Proporciona una mini guía de cómo realizar la depuración de un programa (si te es más fácil, elige uno de los fallos lógicos y úsalos de ejemplo)

Para poder depurar un código, debemos entender que debe hacer el programa, luego procederemos a ejecutar el programa para verificar si este se ejecuta de manera correcta, de lo contrario, deberemos establecer unos puntos de ruptura en el código para posteriormente utilizar el depurador, que nos ayudara a ir ejecutando el código paso a paso y ver como las variables van cambiando, según los métodos y funciones que hayamos declarado anteriormente. También, debemos verificar las variables estén bien declaradas y que estén adquiriendo los valores esperados, de no ser así, es donde empezaremos a corregir las líneas de código y lo volveremos a ejecutar para cerciorarnos de que, ahora sí, se esté ejecutando correctamente.

- Agrega capturas con los puntos de ruptura y las vistas que consideres más importantes a la hora de depurar un programa y por qué.



```

1 package com.example;
2 import java.util.Scanner;
3
4 public class PiramideSumas {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         String resp;
9
10        do {
11            int num = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No se admite ningún carácter
12            while (num < 0 || num > 20) {
13                num = leerEntero(sc, "Valor fuera de rango! Introduzca un número entre 0 y 20: ");
14            }
15
16            System.out.println("Su pirámide de sumas es la siguiente:\n" + piramide(num));
17            System.out.print("Quiere hacer otra pirámide? (s/n)"); // Error lógico: Antes de que se muestra la pregunta se le realiza la respuesta al usuario.
18            resp = sc.nextLine().trim(); // RGC20251031 - Utilizar trim() en vez de trim()
19            sc.nextLine();
20        } while (!resp.equals("s")); // RGC20251031 - Error sintáctico: "Falta la "
21        barrarConsola();
22        System.out.println("¡¡PROGRAMA FINALIZADO!!!");
23    }
24
25    private static void leerEntero(Scanner sc, String mensaje) {
26        System.out.print(mensaje);
27        while (!sc.hasNextInt()) {
28            System.out.print("Valor no válido! Introduzca un número entero: ");
29            sc.next();
30        }
31        return sc.nextInt();
32    }
33
34    public static void barrarConsola() {
35        for (int i = 0; i < 50; i++) {
36            System.out.print(" ");
37        }
38    }
39
40    public static String piramide(int num) {
41        String res = "";
42        int n = num;
43
44        while (n > 0) { // RGC20251031 La que sea mayor para que pueda imprimir la pirámide.
45            int cont = 1;
46            int total = 0;
47            res += n + " ";
48            total += n;
49            while (cont <= n) {
50                cont++;
51                res += " " + cont + " ";
52                total += cont;
53            }
54            if (n == 0) {
55                res += "\n";
56            }
57            n--;
58        }
59
60        return res; // RGC20251031 - resultado no está declarado anteriormente, y no apunta nada.
61    }
62
63    public static void main() {
64        System.out.println("Introduzca un número");
65    }
66}

```

“Breakpoints”

```

4
5 public class PiramideSumas {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String resp;
10
11        do {
12            int num = leerEntero(sc, "Introduzca un número");
13            while (num < 0 && num > 20) {
14                num = leerEntero(sc, "**Valor fuera de rango");
15            }
16
17            System.out.println("\nSu pirámide de sumas es ");
18
19            System.out.print("¿Quiere hacer otra pirámide?");
20            resp = sc.next().trim().toUpperCase(); // RGC20251031 - Error ;
21            borrarConsola();
22            System.out.println("!!!PROGRAMA FINALIZADO!!!"); 
23
24            sc.close();
25
26        } while (resp.equals("S")); // RGC20251031 - Error ;
27
28
29    private static int leerEntero(Scanner sc, String mensa;
30        System.out.print(mensaje);
31        while (!sc.hasNextInt()) {
32            System.out.print("**Valor no válido** Introduzca ");
33            sc.next();
34

```

Introduzca un número: 40

Name	Value
num	40
res	"40 => 0 " (id=59)
n	40
cont	2
total	0

“Vistas”

Estas vistas nos permiten observar cómo van cambiando las variables según vayamos depurando, lo que nos facilita a la hora de ver errores lógicos, al igual que nos permiten ver cuántos breakpoints tenemos, al igual que podemos observar las variables y como cambian según se vaya avanzando en las líneas de código.

- ¿Dónde colocaste los breakpoints y qué valores viste que confirmaron el fallo?
Coloque los breakpoints en cada “System.out.” y debajo de cada “While” y “for” para poder observar el comportamiento de las variables según las condiciones del While y For y así poder darme cuenta de los errores lógicos.
- ¿En qué situación debemos usar Step Into (F5), Step Over (F6), Step Return (F7) y Resume (F8)?

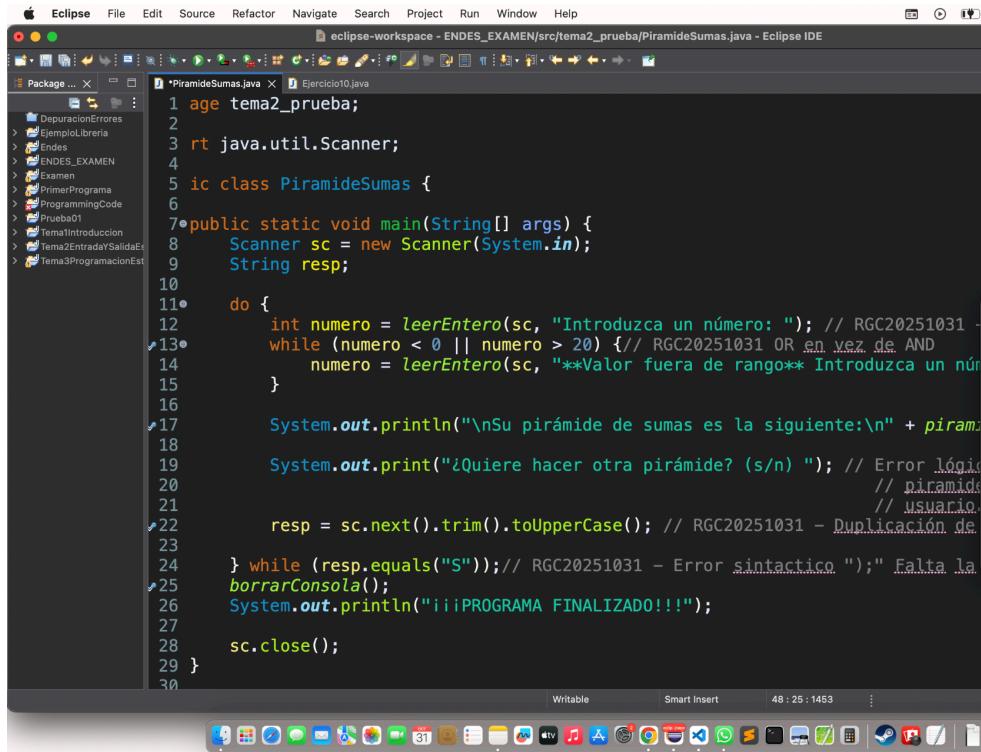
Utilizamos Step Into cuando queremos entrar al método o función llamado de la línea de código.

Utilizamos el Step Over cuando se ejecuta la línea sin entrar al método.

Utilizamos Step Return para salir de la función actual y volver a la línea anterior.

Utilizamos Resume para continuar con la ejecución sin interrupción hasta el próximo breakpoint.

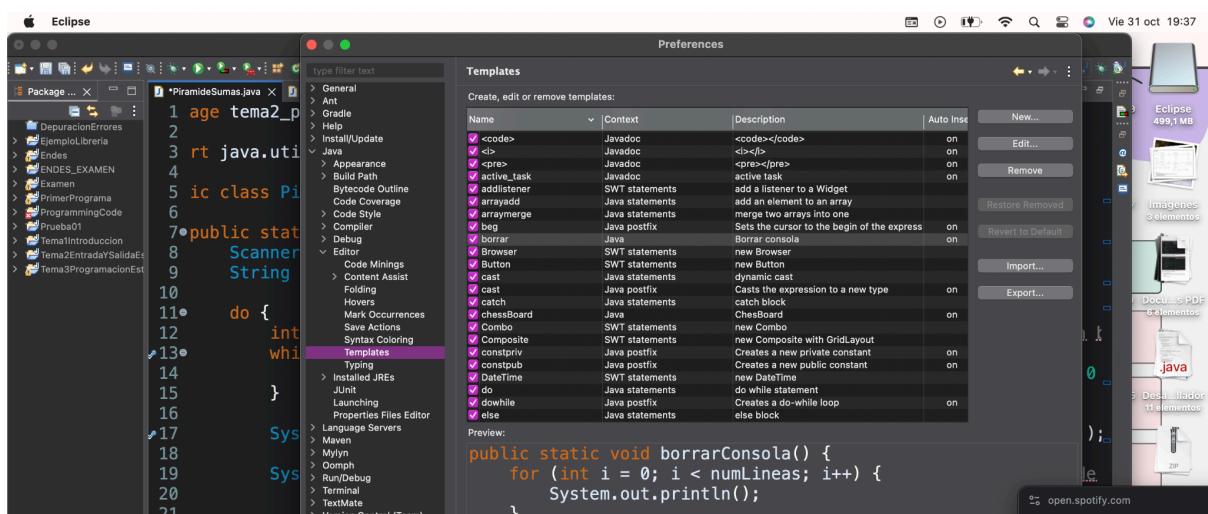
Apartado opcional:



```
1 age tema2_prueba;
2
3 rt java.util.Scanner;
4
5 ic class PiramideSumas {
6
7•public static void main(String[] args) {
8     Scanner sc = new Scanner(System.in);
9     String resp;
10
11•    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 -
13•        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un númer
15        }
16
17        System.out.println("\nSu pirámide de sumas es la siguiente:\n" + piramid
18
19        System.out.print("¿Quiere hacer otra pirámide? (s/n) "); // Error lógico
20                                         // pirámide
21                                         // usuario
22        resp = sc.next().trim().toUpperCase(); // RGC20251031 - Duplicación de
23
24    } while (resp.equals("S")); // RGC20251031 - Error sintáctico );" Falta la
25•    System.out.println("!!!PROGRAMA FINALIZADO!!!"); // RGC20251031 - Error lógico
26
27
28    sc.close();
29 }
30
```

- ¿Cómo lo hiciste? ¿Usaste alguna herramienta integrada del IDE para no equivocarte? Expícame cuál, cómo lo hiciste (brevemente) y por qué lo consideras una buena ayuda.

Utilice cmd+option+r. Para poder renombrar variables sin errores. Ya que esta función cambia todos los nombres automáticamente en el código, evitando confusiones y manteniendo el programa coherente. Esto es muy útil porque ahorra tiempo y previene fallos de referencia.

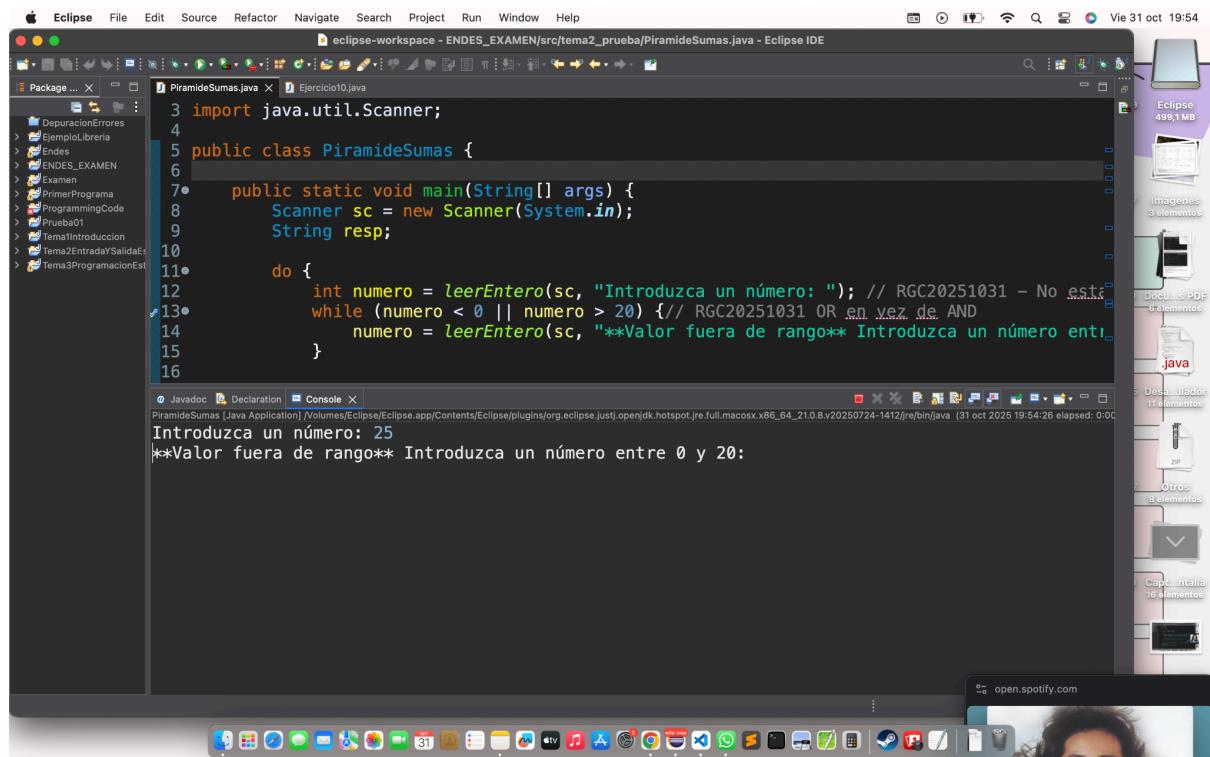


Name	Context	Description	Auto Insert
<code>	Javadoc	<code></code>	on
<pre>	Javadoc	<pre></pre>	on
<active_task>	Javadoc	active-task>	on
> addlistener	SWT statements	add a listener to a Widget	on
> arrayadd	Java statements	add an element to an array	on
> arraymerge	Java statements	merge two arrays into one	on
> beg	Java postfix	Sets the cursor to the begin of the expression	on
> borrar	Java	Borrar consola	on
> Browser	SWT statements	new Browser	on
> cast	Java statements	new icon	on
> cast	Java postfix	dynamic cast	on
> catch	Java statements	Casts the expression to a new type	on
> chessBoard	Java	catch block	on
> Combo	SWT statements	ChessBoard	on
> Composite	SWT statements	new Composite with GridLayout	on
> constpriv	Java postfix	Creates a new private constant	on
> constpub	Java postfix	Creates a new public constant	on
> Dateitime	SWT statements	new DateTime	on
> do	Java statements	do while statement	on
> dowhile	Java postfix	Creates a do-while loop	on
> else	Java statements	else block	on

- ¿Para qué sirve una plantilla?

Sirve para poder ahorrar tiempo al escribir código repetitivo, a su vez, mantiene un formato uniforme en varios proyectos o clases.

Ejecución final del programa:



A screenshot of the Eclipse IDE interface on a Mac OS X desktop. The central window shows a Java code editor with the file 'PiramideSumas.java' open. The code implements a simple input validation loop using a Scanner object to read from System.in. The 'Console' tab at the bottom displays the output of running the program, which asks for a number and then prints an error message if the input is invalid. The desktop background shows a folder icon for 'Eclipse' and a Spotify window.

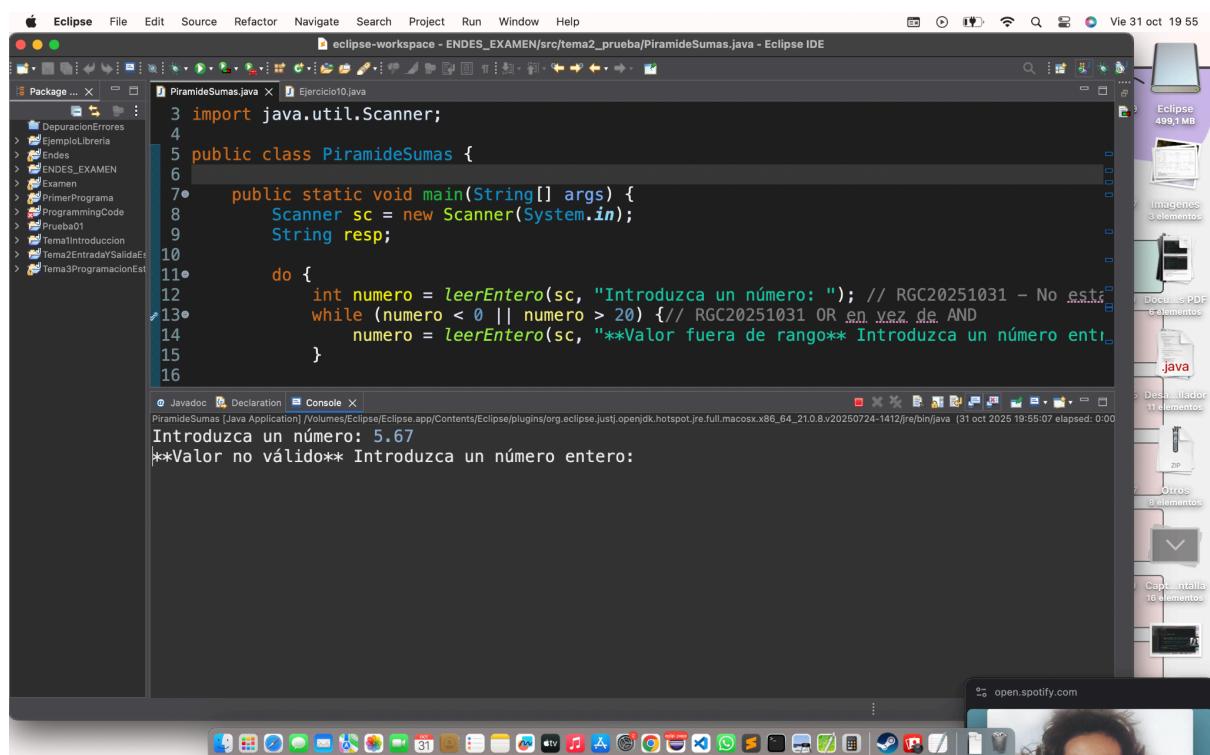
```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No est
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un número enti
15        }
16    }
```

PiramideSumas [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.jst.java.core/bin/java (31 oct 2025 19:54:26 elapsed: 0:00)

Introduzca un número: 25
Valor fuera de rango Introduzca un número entre 0 y 20:

"Al introducir un número fuera de rango nos sale el mensaje correspondiente"

2



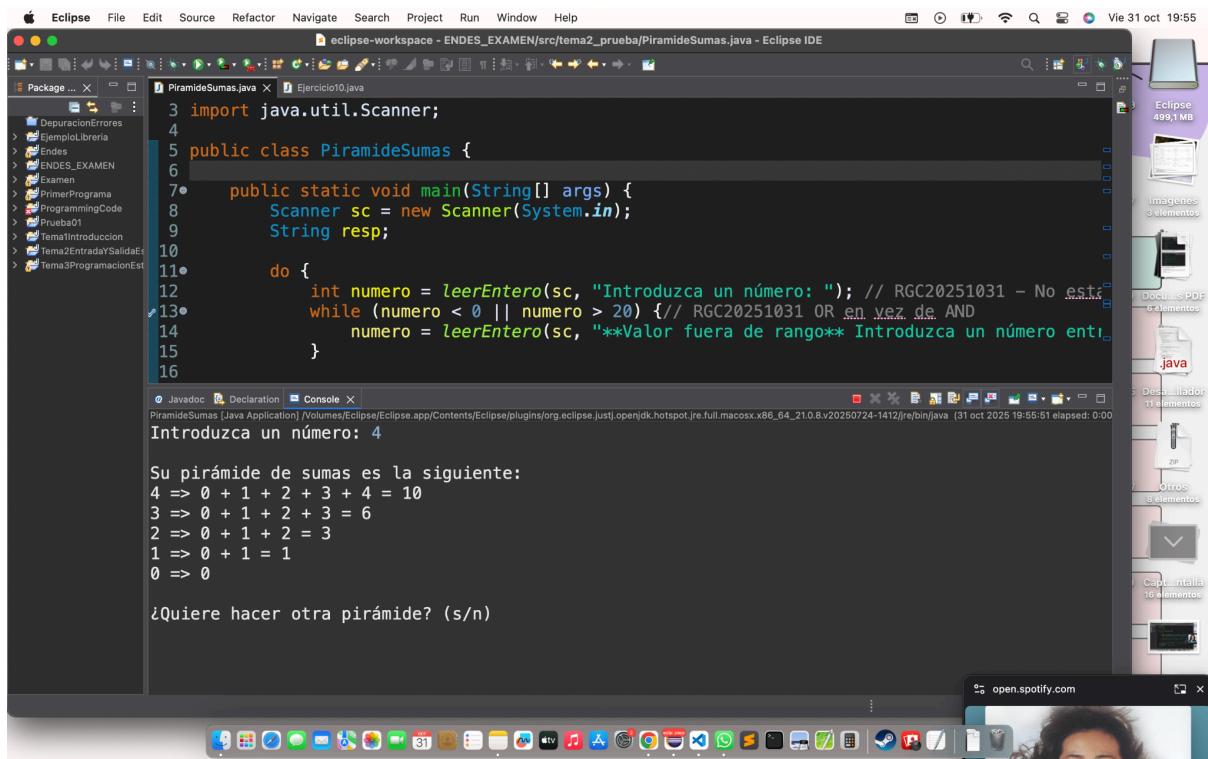
A second screenshot of the Eclipse IDE interface on the same Mac OS X desktop. The code and setup are identical to the first screenshot, but the user has entered a valid integer value (5.67) instead of an invalid one. The console output reflects this, showing the program asking for a valid integer input. The desktop background elements remain the same.

```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No est
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor no válido** Introduzca un número enti
15        }
16    }
```

PiramideSumas [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.jst.java.core/bin/java (31 oct 2025 19:55:07 elapsed: 0:00)

Introduzca un número: 5.67
Valor no válido Introduzca un número entero:

“Al introducir un número NO entero nos solicita que volvamos a introducir un número entero”



```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No esta bien el while
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un número entero: ");
15        }
16    } while (true);
17
18    System.out.println("Su pirámide de sumas es la siguiente:");
19    for (int i = 0; i < numero; i++) {
20        System.out.print(i + " => " + (0 + i));
21    }
22
23    System.out.println("\n¿Quiere hacer otra pirámide? (s/n)");
24
25    resp = sc.nextLine();
26
27    if (resp.equalsIgnoreCase("s")) {
28        continue;
29    } else {
30        break;
31    }
32
33    System.out.println("Adiós!");
34
35}
```

Introduzca un número: 4

Su pirámide de sumas es la siguiente:

4 => 0 + 1 + 2 + 3 + 4 = 10

3 => 0 + 1 + 2 + 3 = 6

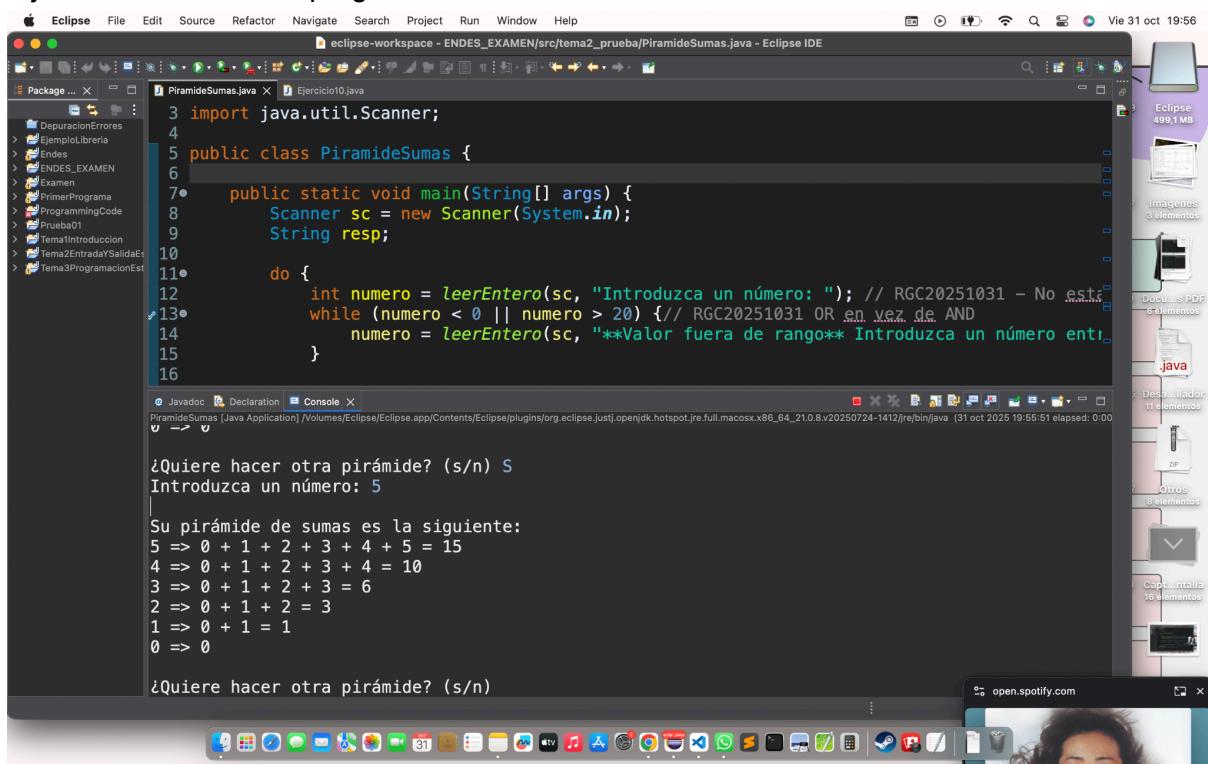
2 => 0 + 1 + 2 = 3

1 => 0 + 1 = 1

0 => 0

¿Quiere hacer otra pirámide? (s/n)

“Ejecución correcta del programa”



```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No esta bien el while
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un número entero: ");
15        }
16    } while (true);
17
18    System.out.println("Su pirámide de sumas es la siguiente:");
19    for (int i = 0; i < numero; i++) {
20        System.out.print(i + " => " + (0 + i));
21    }
22
23    System.out.println("\n¿Quiere hacer otra pirámide? (s/n)");
24
25    resp = sc.nextLine();
26
27    if (resp.equalsIgnoreCase("s")) {
28        continue;
29    } else {
30        break;
31    }
32
33    System.out.println("Adiós!");
34
35}
```

v -- v

¿Quiere hacer otra pirámide? (s/n) S

Introduzca un número: 5

Su pirámide de sumas es la siguiente:

5 => 0 + 1 + 2 + 3 + 4 + 5 = 15

4 => 0 + 1 + 2 + 3 + 4 = 10

3 => 0 + 1 + 2 + 3 = 6

2 => 0 + 1 + 2 = 3

1 => 0 + 1 = 1

0 => 0

¿Quiere hacer otra pirámide? (s/n)

“Ejecución correcta del programa”

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "eclipse-workspace - ENDES_EXAMEN" containing several source files like "PiramideSumas.java", "Ejercicio10.java", and "Prueba01.java".
- Editor:** Displays the code for "PiramideSumas.java".
- Console:** Shows the output of the program execution:

```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No esta bien el uso de AND
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un número entero: ");
15        }
16    }
```

```
PiramideSumas [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java [31 oct 2025 19:55:51 elapsed: 0:00]
1 => 0 + 1 + 2 + 3
1 => 0 + 1 + 1
0 => 0

¿Quiere hacer otra pirámide? (s/n) s
Introduzca un número: 3

Su pirámide de sumas es la siguiente:
3 => 0 + 1 + 2 + 3 = 6
2 => 0 + 1 + 2 = 3
1 => 0 + 1 = 1
0 => 0

¿Quiere hacer otra pirámide? (s/n)
```
- File Bar:** Shows the date and time: "Vie 31 oct 19:56"
- System Tray:** Shows icons for battery, signal, and other system status.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "eclipse-workspace - ENDES_EXAMEN" containing several source files like "PiramideSumas.java", "Ejercicio10.java", and "Prueba01.java".
- Editor:** Displays the code for "PiramideSumas.java".
- Console:** Shows the output of the program execution:

```
3 import java.util.Scanner;
4
5 public class PiramideSumas {
6
7    public static void main(String[] args) {
8        Scanner sc = new Scanner(System.in);
9        String resp;
10
11    do {
12        int numero = leerEntero(sc, "Introduzca un número: "); // RGC20251031 - No esta bien el uso de AND
13        while (numero < 0 || numero > 20) { // RGC20251031 OR en vez de AND
14            numero = leerEntero(sc, "**Valor fuera de rango** Introduzca un número entero: ");
15        }
16    }
```

```
<terminated> PiramideSumas [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java [31 oct 2025 19:55:51 elapsed: 0:00]
```

```
iii PROGRAMA FINALIZADO!!!
```
- File Bar:** Shows the date and time: "Vie 31 oct 19:57"
- System Tray:** Shows icons for battery, signal, and other system status.

“Ejecución colorrectal del programa al escribir “N o n” para salir del programa”