

CIFP César Manrique.

Programación 1º de Desarrollo de Aplicaciones Web

Profesor: José David Díaz Díaz

Actividades de la Unidad 2: Condicionales



Esta obra está licenciada bajo la Licencia Creative Commons Atribución 4.0 Internacional. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/> o envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Índice

Actividades.....	1
Desarrollo.....	2
Actividades de comprobación.....	2
2.1. Los operadores lógicos operan con valores booleanos , resultando :.....	2
2.2. La evaluación de una expresión relacional puede generar un valor de tipo.....	2
2.3. La expresión $3 == 3 \ \&\& \ 2 < 3 \ \&\& \ 1 != 2$ resulta.....	2
2.4. La siguiente expresión , donde interviene la variable booleana a : $3 != 3 \ \ a \ \ 1 < 2$, resulta :.....	2
2.5 . Elige los valores de las variables enteras (a , b y c) que permiten que la evaluación de la siguiente expresión sea cierta : $a < b \ \&\& \ b != C \ \&\& \ b < = c$	2
2.6 . El bloque de instrucciones de una sentencia if se ejecutará :.....	3
2.7 . En una sentencia if - else los bloques de instrucciones (bloque true y bloque false) pueden ejecutarse :.....	3
2.8. ¿ Qué valor toma la variable a en la siguiente expresión: $a = 1 < 2 ? 3 : 4 ?$	3
2.9. La cláusula default en la sentencia switch es :.....	3
2.10. Realiza una traza del siguiente fragmento de código y selecciona el valor que toma finalmente la variable a:.....	4
Actividades de Aplicación.....	5
2.11 . Escribe una aplicación que solicite al usuario un número comprendido entre 0 y 9999. La aplicación tendrá que indicar si el número introducido es capicúa5	
2.12 . El DNI consta de un entero de 8 dígitos seguido de una letra que se obtiene a partir del número de la siguiente forma : letra = número DNI módulo 22 Basándote en esta información , elige la letra a partir de la numeración de la siguiente tabla :.....	6

2.13. En una granja se compra diariamente una cantidad (comidaDiaria) de comida para los animales.....	7
2.14. Escribe un programa que solicite al usuario un número comprendido entre 1 y 99. El programa debe mostrarlo con letras, por ejemplo, para 56, se verá : « cincuenta y seis »	9
2.15. Escribe una aplicación que solicite por consola dos números reales que corresponden a la base y la altura de un triángulo. Deberá mostrarse su área, comprobando que los números introducidos por el usuario no son negativos, algo que no tendría sentido	11
2.16. Utiliza el operador ternario para calcular el valor absoluto de un número que se solicita al usuario por teclado.....	12
2.17. Realiza el « juego de la suma », que consiste en que aparezcan dos números aleatorios (comprendidos entre 1 y 99) que el usuario tiene que sumar. La aplicación debe indicar si el resultado de la operación es correcto o incorrecto.....	13
2.18. Modifica la Actividad de aplicación 2.17 para que, además de los dos números aleatorios, también aparezca la operación que debe realizar el jugador : suma , resta o multiplicación	14
2.19. Crea una aplicación que solicite al usuario cuántos grados tiene un ángulo y muestre el equivalente en radianes. Si el ángulo introducido por el usuario no se encuentra en el rango de 0° a 360° , hay que transformarlo a dicho rango. Nota : El operador módulo puede ayudarnos a convertir un ángulo a su equivalente en el rango comprendido de 0° a 360°	15

Actividades

Actividades de la Unidad 2: Condicionales.

En este documento se detallan las actividades a realizar. Se entregará al profesor en la plataforma digital dos ficheros. Un primer fichero pdf con todas las actividades a realizar, el nombre del fichero será “unidad2 + nombre del alumno.pdf”. Añadir en el fichero pdf por cada actividad de programación dos capturas de pantalla, una del código y otra de su ejecución. También en el fichero pdf copiar todas las preguntas y las respuestas correctas de las actividades de comprobación. Además, entregar un segundo fichero comprimido con todos los códigos fuentes de cada actividad de programación realizada.

Todas las actividades resueltas se deberán de analizar y no se entregarán.

A continuación, detallamos las actividades a realizar:

- **Actividades propuestas.** No realizar ninguna.
- **Actividades de comprobación.** Realizarlas todas. Copiar las preguntas y las respuestas correctas.
- **Actividades de aplicación.** Realizar las siguientes **2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17, 2.18, y 2.19.**
- **Actividades de ampliación.** No realizar ninguna.

[Volver al índice](#)

Desarrollo

Actividades de comprobación.

2.1. Los operadores lógicos operan con valores booleanos , resultando :

d) Solo valores booleanos

[Volver al índice](#)

2.2. La evaluación de una expresión relacional puede generar un valor de tipo

c) Booleano

[Volver al índice](#)

2.3. La expresión `3 == 3 && 2 < 3 && 1 != 2` resulta

a) Cierto

[Volver al índice](#)

2.4. La siguiente expresión , donde interviene la variable booleana a : `3 != 3 || a || 1 < 2` , resulta :

a) Dependerá del valor de a

[Volver al índice](#)

2.5 . Elige los valores de las variables enteras (a , b y c) que permiten que la evaluación de la siguiente expresión sea cierta : `a < b && b != C && b <= c`

d) a = 1 , b = 2 , C = 3 .

[Volver al índice](#)

2.6 . El bloque de instrucciones de una sentencia if se ejecutará :

c) Dependerá de la evaluación de la expresión utilizada.

[Volver al índice](#)

2.7 . En una sentencia if - else los bloques de instrucciones (bloque true y bloque false) pueden ejecutarse :

c) Siempre se ejecutará al menos uno y son excluyentes .

[Volver al índice](#)

2.8. ¿ Qué valor toma la variable a en la siguiente expresión: $a = 1 < 2 ? 3 : 4 ?$

c) 3

[Volver al índice](#)

2.9. La cláusula default en la sentencia switch es :

d) Opcional y puede usarse en cualquier lugar.

[Volver al índice](#)

2.10. Realiza una traza del siguiente fragmento de código y selecciona el valor que toma finalmente la variable a:

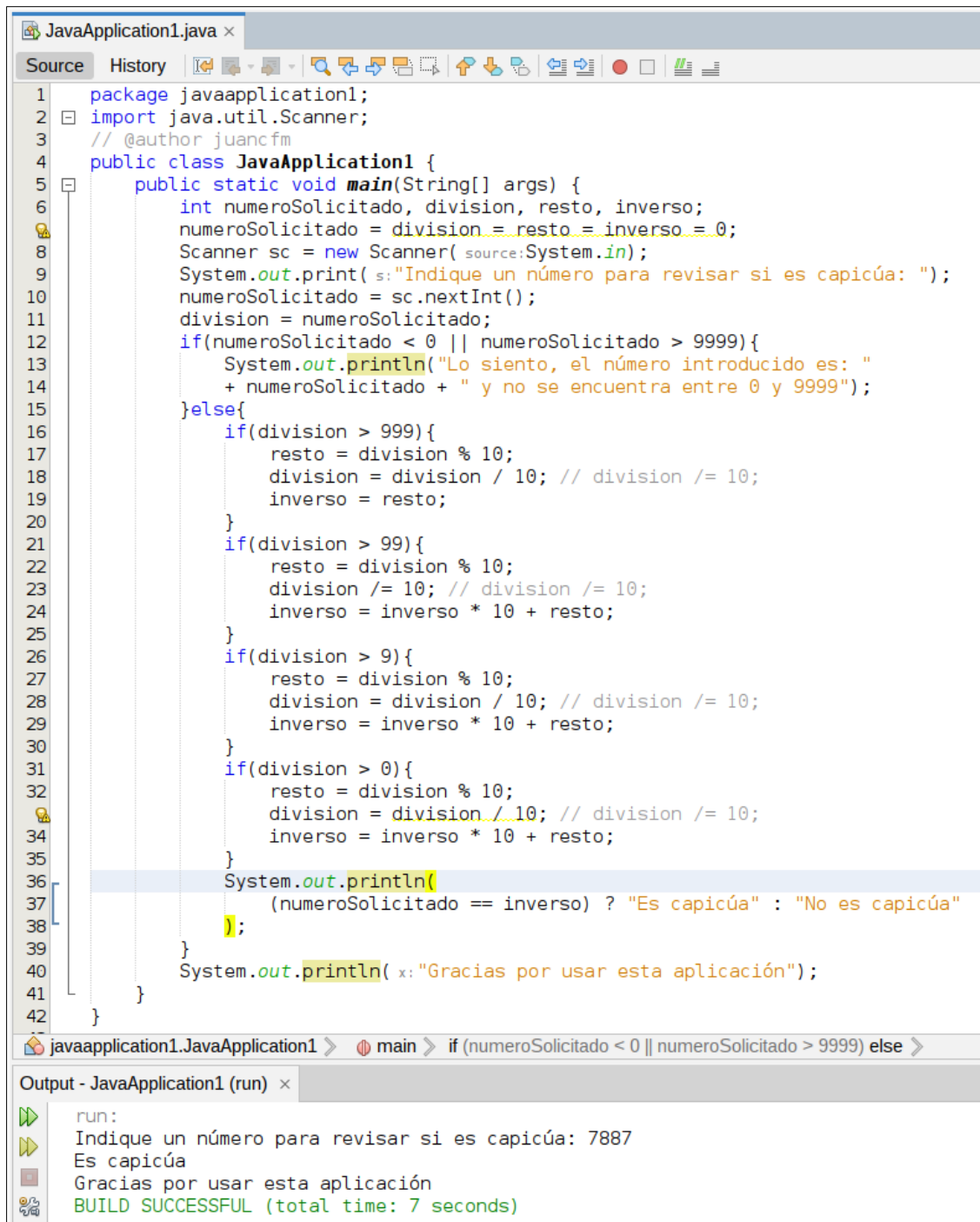
```
a = 0 ;  
switch a + 1 {  
    case 0:  
        a = 2 ;  
    case 1:  
        a = 3;  
    case 2:  
        a ++;  
        break;  
    case 3:  
        a --;  
        break;  
}
```

d) 4

[Volver al índice](#)

Actividades de Aplicación.

2.11 . Escribe una aplicación que solicite al usuario un número comprendido entre 0 y 9999. La aplicación tendrá que indicar si el número introducido es capicúa .



```

1  package javaapplication1;
2  import java.util.Scanner;
3  // @author juancfm
4  public class JavaApplication1 {
5      public static void main(String[] args) {
6          int numeroSolicitado, division, resto, inverso;
7          numeroSolicitado = division = resto = inverso = 0;
8          Scanner sc = new Scanner(System.in);
9          System.out.print("Indique un número para revisar si es capicúa: ");
10         numeroSolicitado = sc.nextInt();
11         division = numeroSolicitado;
12         if(numeroSolicitado < 0 || numeroSolicitado > 9999){
13             System.out.println("Lo siento, el número introducido es: "
14                 + numeroSolicitado + " y no se encuentra entre 0 y 9999");
15         }else{
16             if(division > 999){
17                 resto = division % 10;
18                 division = division / 10; // division /= 10;
19                 inverso = resto;
20             }
21             if(division > 99){
22                 resto = division % 10;
23                 division /= 10; // division /= 10;
24                 inverso = inverso * 10 + resto;
25             }
26             if(division > 9){
27                 resto = division % 10;
28                 division = division / 10; // division /= 10;
29                 inverso = inverso * 10 + resto;
30             }
31             if(division > 0){
32                 resto = division % 10;
33                 division = division / 10; // division /= 10;
34                 inverso = inverso * 10 + resto;
35             }
36             System.out.println(
37                 (numeroSolicitado == inverso) ? "Es capicúa" : "No es capicúa"
38             );
39         }
40         System.out.println("Gracias por usar esta aplicación");
41     }
42 }

```

Output - JavaApplication1 (run) x

```

run:
Indique un número para revisar si es capicúa: 7887
Es capicúa
Gracias por usar esta aplicación
BUILD SUCCESSFUL (total time: 7 seconds)

```

[Volver al índice](#)

2.12 . El DNI consta de un entero de 8 dígitos seguido de una letra que se obtiene a partir del número de la siguiente forma : letra = número DNI módulo 22 Basándote en esta información , elige la letra a partir de la numeración de la siguiente tabla :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

y diseña una aplicación en la que , dado un número de DNI , calcule la letra que le corresponde . Observa que un número de 8 dígitos está dentro del rango del tipo int .

```

1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int dni;
13         String letra = "";
14         Scanner sc = new Scanner(System.in);
15         System.out.print("Introduzca el número del DNI: ");
16         dni = sc.nextInt();
17         switch (dni % 23) {
18             case 0 ->
19                 letra = "T";
20             case 1 ->
21                 letra = "R";
22             case 2 ->
23                 letra = "W";
24             case 3 ->
25                 letra = "A";
26             case 4 ->
27                 letra = "G";
28             case 5 ->
29                 letra = "M";
30             case 6 ->
31                 letra = "Y";
32             case 7 ->
33                 letra = "F";
34             case 8 ->
35                 letra = "P";
36             case 9 ->
37                 letra = "D";
38             case 10 ->

```

```

38         case 10 ->
39             letra = "X";
40         case 11 ->
41             letra = "B";
42         case 12 ->
43             letra = "N";
44         case 13 ->
45             letra = "J";
46         case 14 ->
47             letra = "Z";
48         case 15 ->
49             letra = "S";
50         case 16 ->
51             letra = "Q";
52         case 17 ->
53             letra = "V";
54         case 18 ->
55             letra = "H";
56         case 19 ->
57             letra = "L";
58         case 20 ->
59             letra = "C";
60         case 21 ->
61             letra = "K";
62         case 22 ->
63             letra = "E";
64         }
65         System.out.println("La letra correspondiente a esa numeración es: " + letra);
66         System.out.println("\nSu NIF es: " + dni + letra);
67     }
68 }
69
70

```

```

Output - JavaApplication1 (run) x
run:
Introduzca el número del DNI: 43386436
La letra correspondiente a esa numeración es: H

Su NIF es: 43386436H
BUILD SUCCESSFUL (total time: 17 seconds)

```

[Volver al índice](#)

2.13. En una granja se compra diariamente una cantidad (comidaDiaria) de comida para los animales.

El número de animales que alimentar (todos de la misma especie) es numAnimales , y sabemos que cada animal come una media de kilosPorAnimal. Diseña un programa que solicite al usuario los valores anteriores y determine si disponemos de alimento suficiente para cada animal. En caso negativo, ha de calcular cuál es la ración que corresponde a cada uno de los animales.

Nota : Evitar que la aplicación realice divisiones por cero .

```

1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  * @author juancf
7  */
8
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int numAnimales;
13         double comidaDiaria, kilosPorAnimal;
14         Scanner sc = new Scanner(System.in);
15
16         System.out.print("¿A cuántos animales vamos a alimentar hoy: ");
17         numAnimales = sc.nextInt();
18         System.out.print("¿Cuántos kilos de alimento come "
19             + "cada animal al día?: ");
20         kilosPorAnimal = sc.nextDouble();
21         System.out.print("¿Cuánta comida compramos hoy: ");
22         comidaDiaria = sc.nextDouble();
23         if (numAnimales <= 0) {
24             System.out.println("\nLo sentimos, pero no ha indicado..."
25                 + "correctamente el número de animales.");
26         } else {
27             System.out.println("\nLo sentimos, pero no ha indicado..."
28                 + "correctamente el número de animales.");
29             if ((comidaDiaria / numAnimales) >= kilosPorAnimal) {
30                 System.out.println("Perfecto, hay alimentos en cantidad "
31                     + "suficiente para todos los animales");
32             } else {
33                 System.out.println("Lo sentimos, pero la comida no alcanza.");
34                 System.out.println("¿Me diríamos que...");
35                 + "racionar los alimentos a:");
36                 System.out.println(comidaDiaria / numAnimales + " kilos de "
37                     + "alimento por animal");
38             }
39             System.out.println("\nGracias por usar nuestro servicio.");
40         }
41     }
42 }
  
```

```

run:
¿A cuántos animales vamos a alimentar hoy: 45
¿Cuántos kilos de alimento come cada animal al día?: 2
¿Cuánta comida compramos hoy: 100
Perfecto, hay alimentos en cantidad suficiente para todos los animales

Gracias por usar nuestro servicio.
BUILD SUCCESSFUL (total time: 9 seconds)
  
```

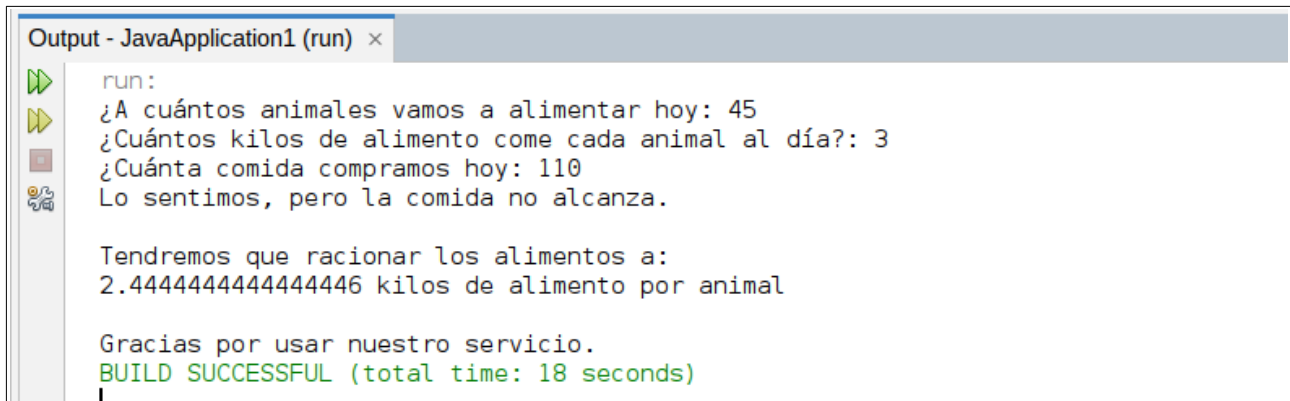
```

run:
¿A cuántos animales vamos a alimentar hoy: 0
¿Cuántos kilos de alimento come cada animal al día?: 3
¿Cuánta comida compramos hoy: 45

Lo sentimos pero no ha indicado correctamente el número de animales.

Gracias por usar nuestro servicio.
BUILD SUCCESSFUL (total time: 24 seconds)
  
```

Actividades de la Unidad 2: Condicionales

A screenshot of an IDE's output window titled "Output - JavaApplication1 (run)". The window contains the following text: "run:", "¿A cuántos animales vamos a alimentar hoy: 45", "¿Cuántos kilos de alimento come cada animal al día?: 3", "¿Cuánta comida compramos hoy: 110", "Lo sentimos, pero la comida no alcanza.", "Tendremos que racionar los alimentos a:", "2.4444444444444446 kilos de alimento por animal", "Gracias por usar nuestro servicio.", and "BUILD SUCCESSFUL (total time: 18 seconds)". On the left side of the output window, there are four icons: a green play button, a yellow play button, a red stop button, and a magnifying glass icon.

```
run:
¿A cuántos animales vamos a alimentar hoy: 45
¿Cuántos kilos de alimento come cada animal al día?: 3
¿Cuánta comida compramos hoy: 110
Lo sentimos, pero la comida no alcanza.

Tendremos que racionar los alimentos a:
2.4444444444444446 kilos de alimento por animal

Gracias por usar nuestro servicio.
BUILD SUCCESSFUL (total time: 18 seconds)
```

[Volver al índice](#)

2.14. Escribe un programa que solicite al usuario un número comprendido entre 1 y 99. El programa debe mostrarlo con letras, por ejemplo, para 56, se verá : « cincuenta y seis » .

```

1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int numUsuario, decenaNumero, unidadNumero;
13         String decenas, unidades, respuesta;
14         Scanner sc = new Scanner(System.in);
15
16         System.out.print("Introduzca un número entre 1 y 99: ");
17         numUsuario = sc.nextInt();
18         if(numUsuario <= 99 && numUsuario >= 1){
19             unidadNumero = numUsuario % 10;
20             decenaNumero = numUsuario / 10;
21             decenas = unidades = respuesta = "";
22             if(decenaNumero > 0 && unidadNumero == 0){
23                 respuesta = switch (decenaNumero){
24                     case 1 -> "Diez";
25                     case 2 -> "Veinte";
26                     case 3 -> "Treinta";
27                     case 4 -> "Cuarenta";
28                     case 5 -> "Cincuenta";
29                     case 6 -> "Sesenta";
30                     case 7 -> "Setenta";
31                     case 8 -> "Ochenta";
32                     case 9 -> "Noventa";
33                     default -> "";
34                 };
35             }
36             if(decenaNumero > 0 && unidadNumero != 0){
37                 decenas = switch (decenaNumero){
38                     case 2 -> "Veinte y ";
39                     case 3 -> "Treinta y ";
40                     case 4 -> "Cuarenta y ";
41                     case 5 -> "Cincuenta y ";
42                     case 6 -> "Sesenta y ";
43                     case 7 -> "Setenta y ";
44                     case 8 -> "Ochenta y ";
45

```

```

45         case 8 -> "Ochenta y ";
46         case 9 -> "Noventa y ";
47         default -> "";
48     };
49
50     if(numUsuario > 10 && numUsuario < 20){
51         respuesta = switch(numUsuario){
52             case 11 -> "Once";
53             case 12 -> "Doce";
54             case 13 -> "Trece";
55             case 14 -> "Catorce";
56             case 15 -> "Quince";
57             case 16 -> "Dieciseis";
58             case 17 -> "Diecisiete";
59             case 18 -> "Dieciocho";
60             case 19 -> "Diecinueve";
61             default -> "";
62         };
63     }
64
65     if(respuesta == ""){
66         unidades = switch (unidadNumero){
67             case 1 -> "uno";
68             case 2 -> "dos";
69             case 3 -> "tres";
70             case 4 -> "cuatro";
71             case 5 -> "cinco";
72             case 6 -> "seis";
73             case 7 -> "siete";
74             case 8 -> "ocho";
75             case 9 -> "nueve";
76             default -> "";
77         };
78         respuesta = decenas + unidades;
79     }
80     System.out.println("El número introducido es: " + respuesta);
81 }else{
82     System.out.println("Lo lamento, pero el número debe "
83         + "estar comprendido entre 1 y 99");
84 }
85
86 System.out.println("¡Gracias por hacer uso de este programa.");
87
88 }
89

```

```

Output - JavaApplication1 (run) x
run:
Introduzca un número entre 1 y 99: 7
El número introducido es: siete
Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 5 seconds)

```

```

Output - JavaApplication1 (run) x
run:
Introduzca un número entre 1 y 99: 14
El número introducido es: Catorce
Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 8 seconds)

```

```

Output - JavaApplication1 (run) x
run:
Introduzca un número entre 1 y 99: 56
El número introducido es: Cincuenta y seis
Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 4 seconds)

```

Actividades de la Unidad 2: Condicionales

```
Output - JavaApplication1 (run) x
run:
Introduzca un número entre 1 y 99: 80
El número introducido es: Ochenta
Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
Output - JavaApplication1 (run) x
run:
Introduzca un número entre 1 y 99: 234
Lo lamento, pero el número debe estar comprendido entre 1 y 99
Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 9 seconds)
```

[Volver al índice](#)

2.15. Escribe una aplicación que solicite por consola dos números reales que corresponden a la base y la altura de un triángulo. Deberá mostrarse su área, comprobando que los números introducidos por el usuario no son negativos, algo que no tendría sentido .

```

1  package javaapplication1;
2
3  import java.util.Scanner;
4
5  /**
6   *
7   * @author juancfm
8   */
9  public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         double base, altura;
13         Scanner sc = new Scanner(System.in);
14
15         System.out.println("Favor introducir la misma unidad de medida.");
16         System.out.print("Introduce la base del triángulo: ");
17         base = sc.nextDouble();
18         System.out.print("Introduce ahora la altura del triángulo: ");
19         altura = sc.nextDouble();
20
21         // Se comprueba la condición de existencia de un triángulo, que nos dice que la
22         // suma de cualquiera de los dos lados de un triángulo será mayor que la
23         // longitud del tercero, para lo cual todos los lados han de ser mayores a 0.
24
25         if (base <= 0 || altura <= 0) {
26             System.out.print("Lo siento, "
27                 + "las medidas no pueden ser negativas ni iguales a cero");
28         } else {
29             System.out.println("\nEl área del triángulo es "
30                 + (base*altura/2) + " unidades cuadradas.");
31         }
32         System.out.println("\nGracias por hacer uso de este programa.");
33     }
34 }

```

```

run:
Favor introducir la misma unidad de medida.
Introduce la base del triángulo: 324
Introduce ahora la altura del triángulo: 30

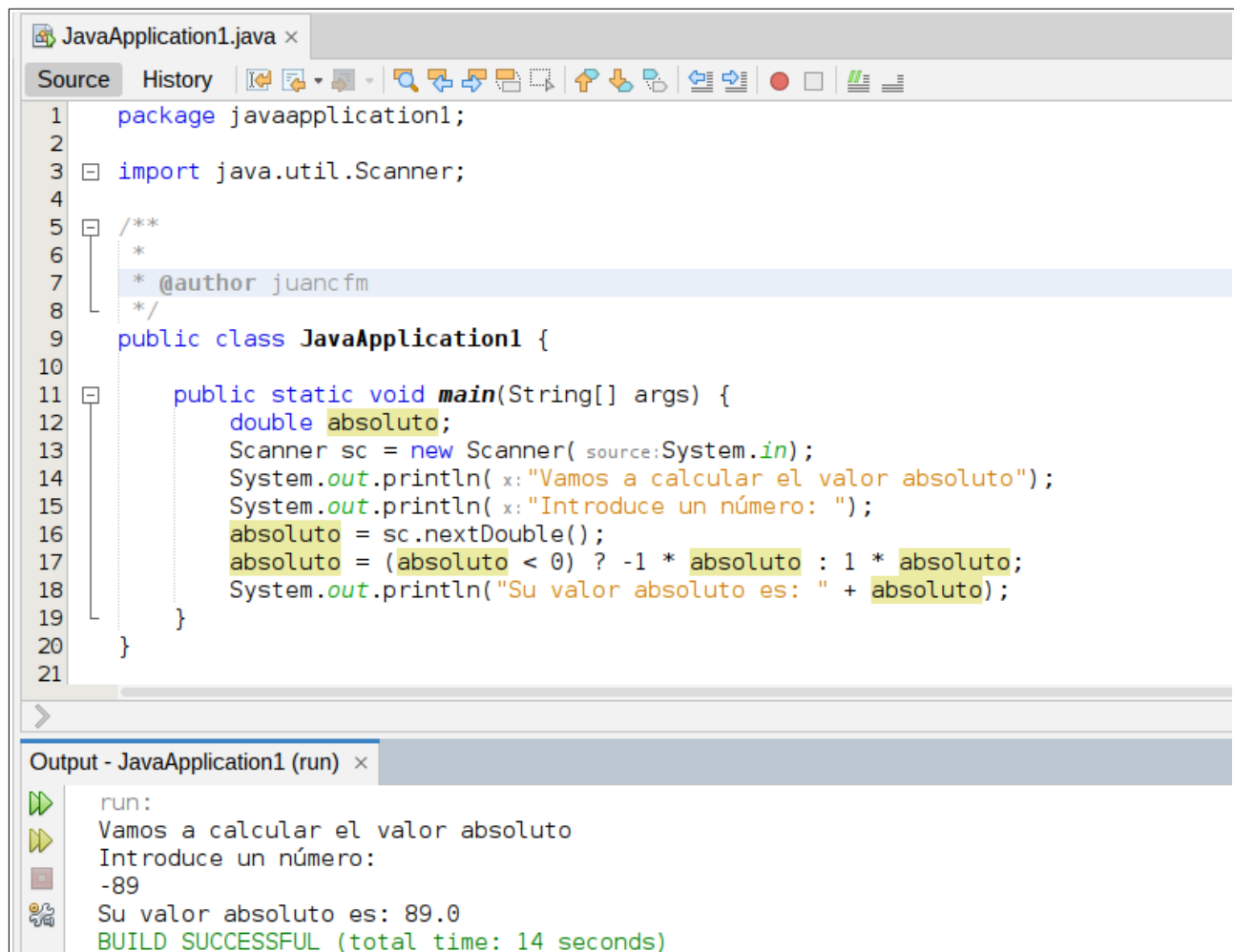
El área del triángulo es 4860.0 unidades cuadradas.

Gracias por hacer uso de este programa.
BUILD SUCCESSFUL (total time: 9 seconds)

```

[Volver al índice](#)

2.16. Utiliza el operador ternario para calcular el valor absoluto de un número que se solicita al usuario por teclado.



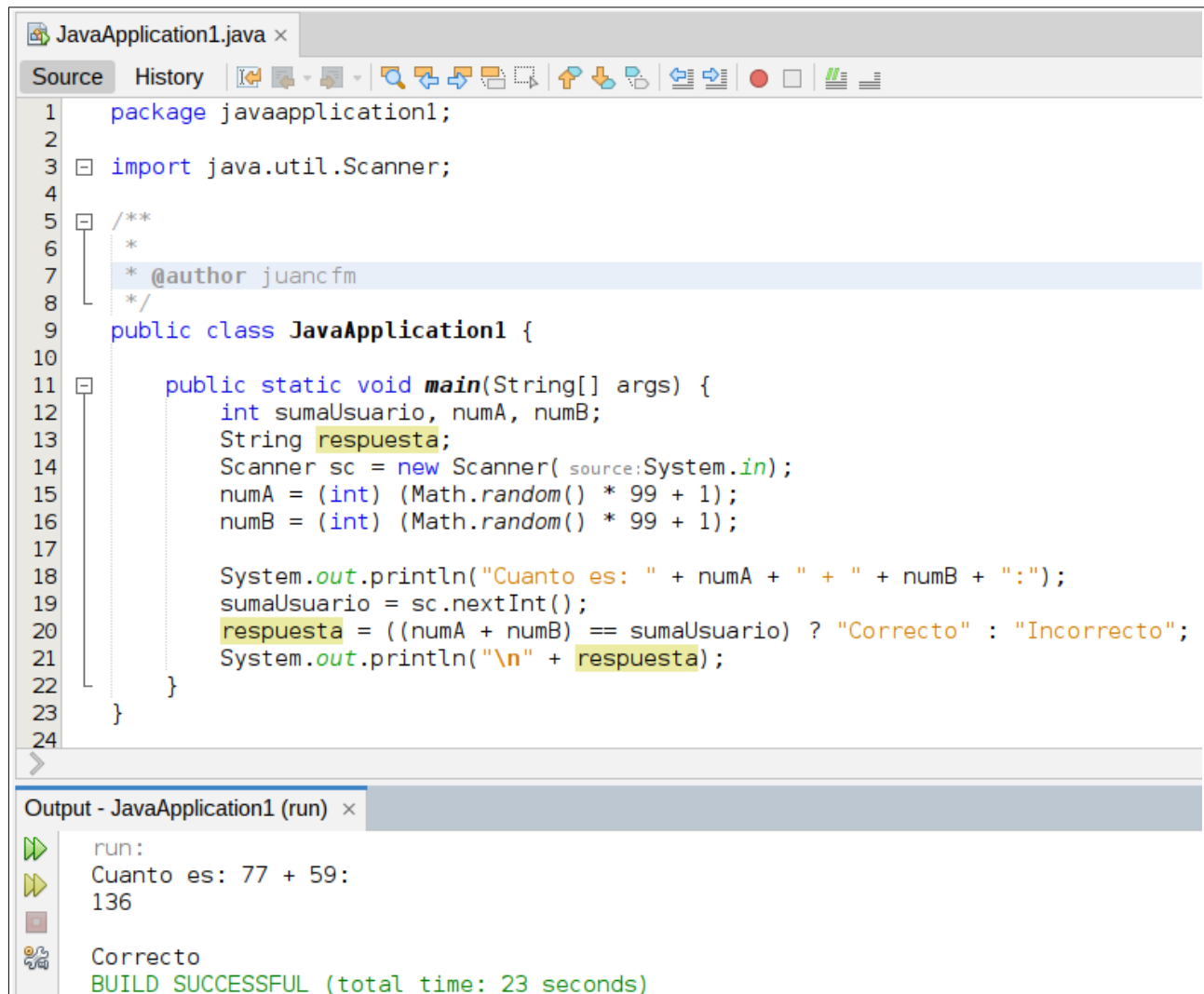
```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         double absoluto;
13         Scanner sc = new Scanner(System.in);
14         System.out.println("Vamos a calcular el valor absoluto");
15         System.out.println("Introduce un número: ");
16         absoluto = sc.nextDouble();
17         absoluto = (absoluto < 0) ? -1 * absoluto : 1 * absoluto;
18         System.out.println("Su valor absoluto es: " + absoluto);
19     }
20 }
21
```

Output - JavaApplication1 (run) ×

```
run:
Vamos a calcular el valor absoluto
Introduce un número:
-89
Su valor absoluto es: 89.0
BUILD SUCCESSFUL (total time: 14 seconds)
```

[Volver al índice](#)

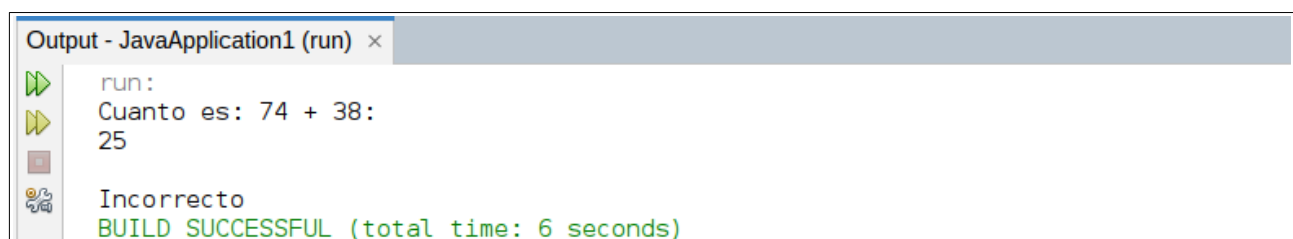
2.17. Realiza el « juego de la suma », que consiste en que aparezcan dos números aleatorios (comprendidos entre 1 y 99) que el usuario tiene que sumar. La aplicación debe indicar si el resultado de la operación es correcto o incorrecto.



```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int sumaUsuario, numA, numB;
13         String respuesta;
14         Scanner sc = new Scanner( source: System.in );
15         numA = (int) (Math.random() * 99 + 1);
16         numB = (int) (Math.random() * 99 + 1);
17
18         System.out.println("Cuanto es: " + numA + " + " + numB + ":");
19         sumaUsuario = sc.nextInt();
20         respuesta = ((numA + numB) == sumaUsuario) ? "Correcto" : "Incorrecto";
21         System.out.println("\n" + respuesta);
22     }
23 }
24
```

Output - JavaApplication1 (run) ×

```
run:
Cuanto es: 77 + 59:
136
Correcto
BUILD SUCCESSFUL (total time: 23 seconds)
```



Output - JavaApplication1 (run) ×

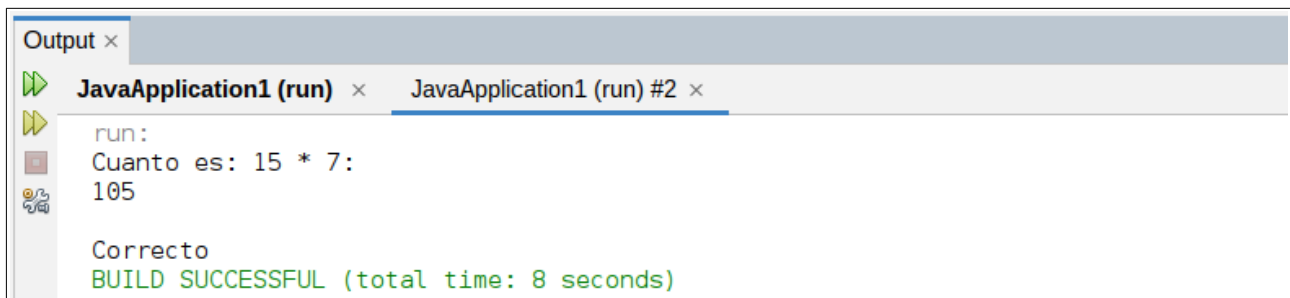
```
run:
Cuanto es: 74 + 38:
25
Incorrecto
BUILD SUCCESSFUL (total time: 6 seconds)
```

[Volver al índice](#)

2.18. Modifica la Actividad de aplicación 2.17 para que, además de los dos números aleatorios, también aparezca la operación que debe realizar el jugador : suma , resta o multiplicación .

```
1 package javaapplication1;
2
3 import java.util.Scanner;
4
5 /**
6  *
7  * @author juancfm
8  */
9 public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         int sumaUsuario, numA, numB, operacion;
13         String respuesta, signo;
14         Scanner sc = new Scanner(System.in);
15         numA = (int) (Math.random() * 99 + 1);
16         numB = (int) (Math.random() * 99 + 1);
17         operacion = (int) (Math.random() * 3 + 1);
18         signo = switch (operacion){
19             case 1 -> "+";
20             case 2 -> "-";
21             case 3 -> "*";
22             default -> "";
23         };
24     }
```

```
24
25     switch (signo){
26         case "+":
27             operacion = numA + numB;
28             break;
29         case "-":
30             operacion = numA - numB;
31             break;
32         case "*":
33             operacion = numA * numB;
34             break;
35     };
36
37     System.out.println("Cuanto es: " + numA + " "
38         + signo + " " + numB + ":");
39     sumaUsuario = sc.nextInt();
40     sc.close();
41     respuesta = (operacion == sumaUsuario) ? "Correcto" : "Incorrecto";
42     System.out.println("\n" + respuesta);
43 }
44 }
```

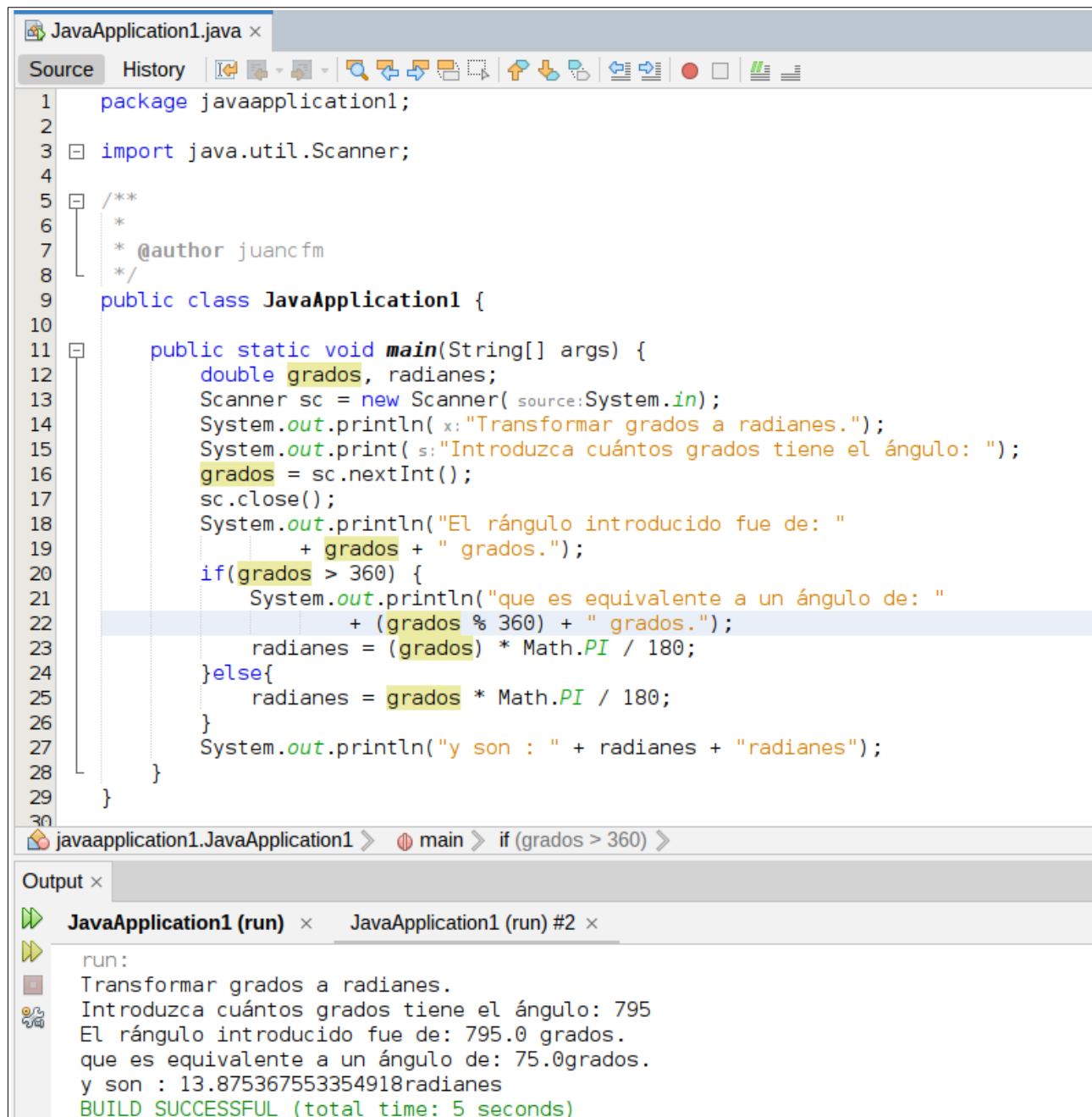


```
Output x
JavaApplication1 (run) x JavaApplication1 (run) #2 x
run:
Cuanto es: 15 * 7:
105

Correcto
BUILD SUCCESSFUL (total time: 8 seconds)
```

[Volver al índice](#)

2.19. Crea una aplicación que solicite al usuario cuántos grados tiene un ángulo y muestre el equivalente en radianes. Si el ángulo introducido por el usuario no se encuentra en el rango de 0 ° a 360 °, hay que transformarlo a dicho rango. Nota : El operador módulo puede ayudarnos a convertir un ángulo a su equivalente en el rango comprendido de 0 ° a 360 °.



The screenshot shows an IDE window titled 'JavaApplication1.java'. The code is as follows:

```

1  package javaapplication1;
2
3  import java.util.Scanner;
4
5  /**
6   *
7   * @author juancfm
8   */
9  public class JavaApplication1 {
10
11     public static void main(String[] args) {
12         double grados, radianes;
13         Scanner sc = new Scanner(System.in);
14         System.out.println("Transformar grados a radianes.");
15         System.out.print("Introduzca cuántos grados tiene el ángulo: ");
16         grados = sc.nextInt();
17         sc.close();
18         System.out.println("El rángulo introducido fue de: "
19             + grados + " grados.");
20         if(grados > 360) {
21             System.out.println("que es equivalente a un ángulo de: "
22                 + (grados % 360) + " grados.");
23             radianes = (grados) * Math.PI / 180;
24         }else{
25             radianes = grados * Math.PI / 180;
26         }
27         System.out.println("y son : " + radianes + "radianes");
28     }
29 }

```

The breadcrumb at the bottom of the editor shows the path: `javaapplication1.JavaApplication1 > main > if (grados > 360)`.

Below the editor is the 'Output' window, which shows the execution results:

```

run:
Transformar grados a radianes.
Introduzca cuántos grados tiene el ángulo: 795
El rángulo introducido fue de: 795.0 grados.
que es equivalente a un ángulo de: 75.0grados.
y son : 13.875367553354918radianes
BUILD SUCCESSFUL (total time: 5 seconds)

```

[Volver al índice](#)