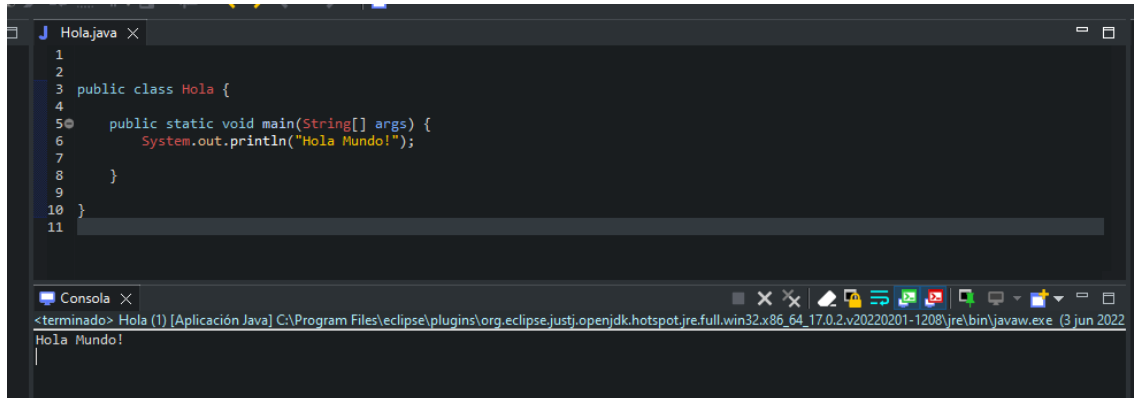


1 Ejercicios Java 1 . [Enlace Repo Ejercios 1 java](#)

1.1 Ejercicio 1: ejecutar un HolaMundo en java para comprobar el correcto funcionamiento del IDE y JDK .



```
1
2
3 public class Hola {
4
5     public static void main(String[] args) {
6         System.out.println("Hola Mundo!");
7     }
8 }
9
10
11
```

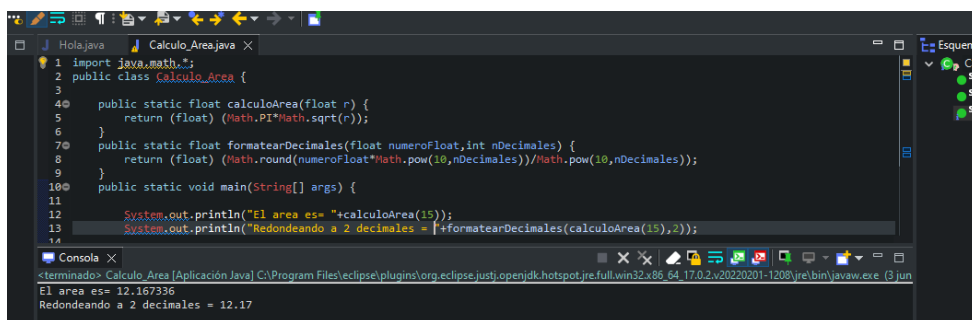
Console

<terminado> Hola (1) [Aplicación Java] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (3 jun 2022)

Hola Mundo!

Ilustración 1 Hola Mundo!!!!

1.2 Ejercicio 2: Dado el radio, (15), calcular el área del círculo y la longitud de la circunferencia.



```
1 import java.math.*;
2 public class Calculo_Area {
3
4     public static float calculoArea(float r) {
5         return (float) (Math.PI*Math.sqrt(r));
6     }
7
8     public static float formatearDecimales(float numeroFloat,int nDecimales) {
9         return (float) (Math.round(numeroFloat*Math.pow(10,nDecimales))/Math.pow(10,nDecimales));
10    }
11
12    public static void main(String[] args) {
13        System.out.println("El area es= "+calculoArea(15));
14        System.out.println("Redondeando a 2 decimales = "+formatearDecimales(calculoArea(15),2));
15    }
16}
```

Console

<terminado> Calculo_Area [Aplicación Java] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (3 jun 2022)

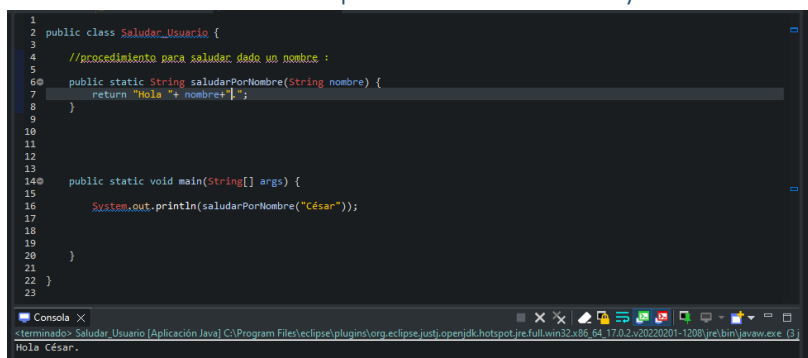
El area es= 12.167336

Redondeando a 2 decimales = 12.17

Ilustración 2 Calcular el área de un Circulo

1.3 Ejercicio 3:

1.3.1 Crear un método que reciba un nombre y salude a dicha persona.



```
1
2 public class Saludar_Usuario {
3
4     //procedimiento para saludar: dada un nombre :
5
6     public static String saludarPorNombre(String nombre) {
7         return "Hola "+ nombre+"!";
8     }
9
10
11
12
13
14
15
16
17
18
19
20
21
22 }
23
```

Console

<terminado> Saludar_Usuario [Aplicación Java] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (3 jun 2022)

Hola César.

Ilustración 3 Saludar a César

1.3.2 Crear un método con un mensaje de error (“usuario no encontrado”).

```
16
17 //Método que da un mensaje de error.
18 public static void mensajeError() {
19     Logger error=Logger.getLogger("err");
20     error.log(Level.INFO, "!Usuario no encontrado!");
21 }
22
23 public static void main(String[] args) {
24     System.out.println(saludarPorNombre("César"));
25     mensajeError();
26 }
27
28
29 }
30
```

Consola X

<terminado> Saludar Usuario [Aplicación Java] C:\Program Files\ eclipse\plugins\org.eclipse.j

Hola César.

Jun 03, 2022 6:25:32 P. M. Saludar_Usuario mensajeError

INFORMACIÓN: !Usuario no encontrado!.

Ilustración 4 mensaje de error

1.3.3 Crear un procedimiento que reciba en nombre de una persona y verifique si es un usuario valido o no (crear con una constante un usuario valido ficticio).

```
23 //Método que reciba el nombre y verifica si es válido o no
24 public static String comprobarUsuario(String user) {
25
26     return (Objects.equals(usr, user))?"Usuario válido":"Usuario inválido";
27 }
28
29
30 public static void main(String[] args) {
31     System.out.println(saludarPorNombre("César"));
32     System.out.println("*****");
33     mensajeError();
34     System.out.println("*****");
35     System.out.println(comprobarUsuario("pedro"));//no válido
36     System.out.println(comprobarUsuario("CesarBouzas"));//válido
37     System.out.println("*****");
38 }
39
40 }
```

Consola X

<terminado> Saludar Usuario [Aplicación Java] C:\Program Files\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full

Hola César.

Jun 03, 2022 6:44:41 P. M. Saludar_Usuario mensajeError

INFORMACIÓN: !Usuario no encontrado!.

Usuario inválido

Usuario válido

Ilustración 5 control de usuario

1.3.4 Hacer pruebas introduciendo distintos nombres de usuarios (podéis ajustarlo más introduciendo nombre y password, por ejemplo).

```
6 public static final String USR="CesarBouzas";
7 public static final String PASS="manuelmatos";
8
9 //Método para saludar , dado un nombre :
10 public static String saludarPorNombre(String nombre) {
11     return "Hola "+ nombre+";";
12 }
13
14 //Método que da un mensaje de error.
15 public static void mensajeError() {
16     Logger error=Logger.getLogger("err");
17     error.log(Level.INFO, "!Usuario no encontrado!");
18 }
19
20 //Método que reciba el nombre y verifica si es válido o no
21 public static String comprobarUsuario(String user) {
22
23     return (Objects.equals(USR, user))?"Usuario válido":"Usuario inválido";
24 }
25
26 public static String comprobarPassword(String pass) {
27     return (Objects.equals(PASS, pass))?"Contraseña correcta":"Contraseña incorrecta";
28 }
29
30 public static void comprobarAcceso(String usr,String pass) {
31     if (Objects.equals(usr,USR)&& Objects.equals(pass,PASS)) {
32         System.out.println ("Acceso permitido");
33     }else {
34         System.out.println(comprobarUsuario(usr));
35         System.out.println(comprobarPassword(pass));
36     }
37 }
38
39 public static void main(String[] args) {
40     System.out.println(saludarPorNombre("César"));
41     System.out.println("*****");
42     mensajeError();
43     System.out.println("*****");
44     System.out.println(comprobarUsuario("pedro"));//no válido
45     System.out.println(comprobarUser in("CesarBouzas"));//válido
46     System.out.println("*****");
47     comprobarAcceso("Pedro","manuelmatos");
48     comprobarAcceso("CesarBouzas","manuelmatos");
49     System.out.println("*****");
50     comprobarAcceso("CesarBouzas","manuelmatos");
51 }
52 }
```

Consola X

<terminado> Saludar Usuario [Aplicación Java] C:\Program Files\ eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\bin\java.exe [17.0.2]

Hola César.

Jun 03, 2022 6:58:17 P. M. Saludar_Usuario mensajeError

INFORMACIÓN: !Usuario no encontrado!.

Usuario inválido

Usuario válido

Usuario inválido

Contraseña correcta

Usuario válido

Contraseña incorrecta

Acceso permitido

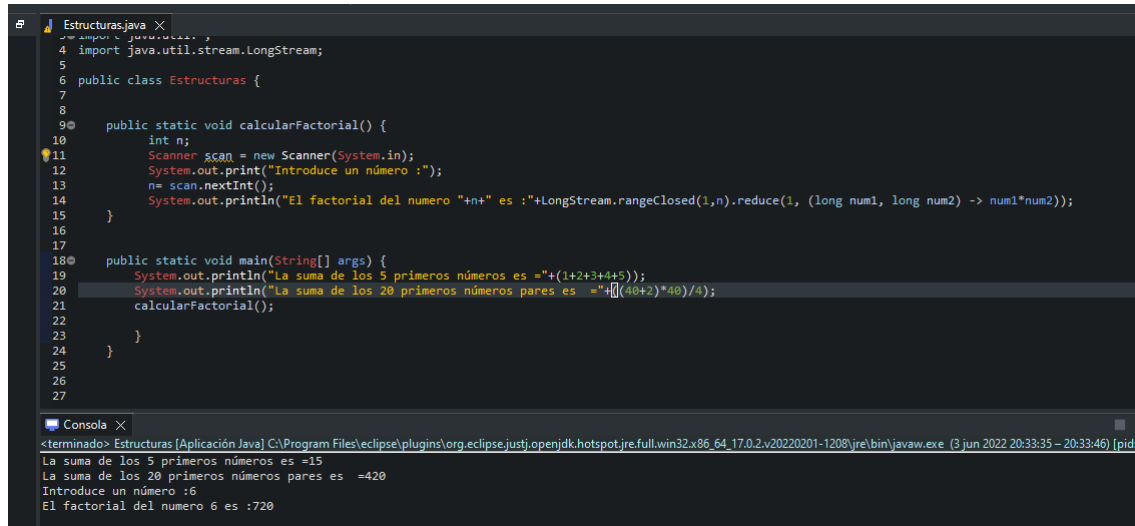
Ilustración 6 control de usuario y password

1.4 Ejercicio 4: Ejercicio para que creen ejemplos sin estructuras de control

1.4.1 Suma los 5 primeros números.

`System.out.println(1+2+3+4+5);`

1.4.2 Suma los 20 primeros números pares.



```
1  import java.util.Scanner;
2
3  import java.util.stream.LongStream;
4
5  public class Estructuras {
6
7
8      public static void calcularFactorial() {
9          int n;
10         Scanner scan = new Scanner(System.in);
11         System.out.print("Introduce un número :");
12         n = scan.nextInt();
13         System.out.println("El factorial del numero "+n+" es :"+LongStream.rangeClosed(1,n).reduce(1, (long num1, long num2) -> num1*num2));
14     }
15
16
17     public static void main(String[] args) {
18         System.out.println("La suma de los 5 primeros números es =(1+2+3+4+5));");
19         System.out.println("La suma de los 20 primeros números pares es =(0+(40+2)*40)/4);");
20         calcularFactorial();
21     }
22 }
23
24
25
26
27
```

Console X

```
<terminado> Estructuras [Aplicación Java] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (3 jun 2022 20:33:35 - 20:33:46) [pid:
La suma de los 5 primeros números es =15
La suma de los 20 primeros números pares es =420
Introduce un número :6
El factorial del numero 6 es :720
```

Ilustración 7 sin Econtrol suma 5 primeros numero pares.

1.4.3 Calcula el factorial del número que introduzcas.



```
1
2
3 import java.util.*;
4 import java.util.stream.LongStream;
5
6 public class Estructuras {
7
8
9     public static void calcularFactorial() {
10         int n;
11         Scanner scan = new Scanner(System.in);
12         System.out.print("Introduce un número :");
13         n = scan.nextInt();
14         System.out.println("El factorial del numero "+n+" es :"+LongStream.rangeClosed(1,n).reduce(1, (long num1, long num2) -> num1*num2));
15     }
16
17
18     public static void main(String[] args) {
19         System.out.println("La suma de los 5 primeros numeros es =(1+2+3+4+5));");
20         calcularFactorial();
21     }
22 }
23
24
25
```

Console X

```
<terminado> Estructuras [Aplicación Java] C:\Program Files\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.2.v20220201-1208\jre\bin\javaw.exe (3 jun 2022 20:18:03 - 20:18:35) [pid:
La suma de los 5 primeros numeros es =15
Introduce un número :5
El factorial del numero 5 es :120
```

Ilustración 8 SinEcontrol factorial de un número

1.4.4 Hacer lo mismo usando estructuras de control.

```

1 import java.util.Scanner;
2 public class Estructuras_Control {
3
4     @SuppressWarnings("resource")
5     public static void sumarNPrimeros(int n) {
6         String txt="";
7         int resultado=0;
8         System.out.println("Suma de los n numeros iniciales ");
9         for(int i=1;i<=n;i++) {
10             resultado+=i;
11             if(i<n) {
12                 txt=txt.concat(Integer.toString(i)+" ");
13             }else if(i==n){
14                 txt=txt.concat(Integer.toString(i)+"=");
15             }
16         }
17         System.out.println("La suma de los "+n+" números primeros es "+txt+resultado);
18     }
19     public static int leerPorEnteroTeclado()
20     {
21         Scanner teclado=new Scanner(System.in);
22         System.out.print("Introduce un número entero :");
23         int n=teclado.nextInt();
24         teclado.close();
25         return n;
26     }
27     @SuppressWarnings("resource")
28     public static void main(String args[]) {
29         int numeros=leerPorEnteroTeclado();
30         sumarNPrimeros(numeros);
31     }
32 }
33

```

Problems Javadoc Declaration Consola

<terminado> Estructuras_Control [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.open

Introduce un número entero :5
 Suma de los n numeros iniciales
 La suma de los 5 números primeros es 1+2+3+4+5=15

Ilustración 9 suma de los n números primeros

```

1 Scanner teclado=new Scanner(System.in);
2 System.out.print("Introduce un número entero :");
3 int n=teclado.nextInt();
4 teclado.close();
5 return n;
6 }
7 public static void sumarNprimerosPares(int n) {
8     String txt="";
9     int contador=0;
10    int i=0;
11    int resultado=0;
12    while (contador<=n) {
13        if(i%2==0) {
14            resultado+=i;
15            if(contador<n) {
16                txt=txt.concat(Integer.toString(i)+" ");
17            }else if(contador==n) {
18                txt=txt.concat(Integer.toString(i)+"");
19            }
20            i++;
21            contador++;
22        }else {
23            i++;
24            continue;
25        }
26    }
27    System.out.println("La suma de los "+n+" números pares es :"+txt+resultado);
28 }
29 public static void main(String args[]) {
30     int numeros=leerPorEnteroTeclado();
31     sumarNprimeros(numeros);
32     sumarNprimeroPares(numeros);
33 }
34 }

```

Problems Javadoc Declaration Console

<terminado> Estructuras_Control [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot

Introduce un número entero :20
 Suma de los n numeros iniciales
 La suma de los 20 números primeros es 1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20=210
 La suma de los 20 números pares es :0+2+4+6+8+10+12+14+16+18+20+22+24+26+28+30+32+34+36+38+40=420

Ilustración 10 suma de los N primeros pares

```

47     }
48
49
50     public static void factorialN(int n) {
51         double resultado=1; // muy facil pasarse de rango
52         String txt="";
53         for(int i=n;i>0;i--) {
54             resultado*=i;
55             if(i>1) {
56                 txt=txt.concat(Integer.toString(i)+"X");
57             } else if(i==1) {
58                 txt=txt.concat(Integer.toString(i)+"=");
59             }
60         }
61         System.out.println("El factorial de "+n+" es :"+txt+resultado);
62     }
63
64     public static void main(String args[]) {
65         int numero=leerPorEnteroTeclado();
66         sumarNprimeros(numero);
67         sumarNprimeroPares(numero);
68         factorialN(numero);
69     }
70 }
71
72 }

```

Problems Javadoc Declaration Console

<terminado> Estructuras_Control [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win

Introduce un número entero :20
Suma de los n números iniciales
La suma de los 20 números primeros es 1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20=210
La suma de los 20 números pares es :0+2+4+6+8+10+12+14+16+18+20+22+24+26+28+30+32+34+36+38+40=420
El factorial de 20 es :20X19X18X17X16X15X14X13X12X11X10X9X8X7X6X5X4X3X2X1=2.43290200817664E18

Ilustración 11 Factorial de un número.

- 1.5 Ejercicio 5: Crear ejercicios de ejemplo para usar las estructuras de control (sentencias if) Comprobar si un número es positivo o negativo, comprobar si un número es múltiplo de otro comprobar que un número es menor a otro.

```

*Sentecias_If.java
1 import java.util.Scanner;
2
3 public class Sentecias_If {
4
5     public static int leerEteroTeclado() {
6         Scanner teclado=new Scanner(System.in);
7         System.out.print("Introduce un número:");
8         return teclado.nextInt();
9     }
10
11     public static void isPositivo(int n){
12         String txt=((n>0)?"El número "+n+" es positivo":"El número "+n+" es negativo");
13         System.out.println(txt);
14     }
15
16     public static void isMultiplo(int n1, int n2) {
17         String txt=((n1%n2==0)?"Los números son multiplos":"Los números no son multiplos");
18         System.out.println(txt);
19     }
20
21     public static void isMenor(int n1, int n2) {
22         String txt=((n1<n2)?"El numero "+n1+" es menor que "+n2:"El numero "+n2+" es menor que "+n1);
23         System.out.print(txt);
24     }
25
26     public static void main(String args[]) {
27         int numero1=leerEteroTeclado();
28         int numero2=leerEteroTeclado();
29         isPositivo(numero1);
30         isPositivo(numero2);
31         isMultiplo(numero1, numero2);
32         isMenor(numero1, numero2);
33     }
34 }
35
36 }

```

Problems Javadoc Declaration Console

<terminado> Sentecias_If [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win

Introduce un número:10
Introduce un número:-2
El número 10 es positivo
El número -2 es negativo
Los números son multiplos
El numero -2 es menor que 10

Ilustración 12 multiplos , mayor y negativo

ERROR CON SCANNER CLOSE

```
Sentecias_If.java X
1 import java.util.Scanner;
2
3 public class Sentecias_If {
4
5 public static int leerEteroTeclado() {
6     Scanner teclado=new Scanner(System.in);
7     System.out.print("Introduce un número:");
8     teclado.close();
9     return teclado.nextInt();
10 }
11 public static void isPositivo(int n){
12     String txt=((n>0)?"El número "+n+" es positivo":"El número "+n+" es negativo");
13     System.out.println(txt);
14 }
15 public static void isMultiplo(int n1, int n2) {
16     String txt=((n1%n2==0)?"Los números son multiplos":"Los números no son multiplos");
17     System.out.println(txt);
18 }
19 public static void isMenor(int n1, int n2) {
20     String txt=((n1<n2)?"El numero "+n1+" es menor que "+n2:"El numero "+n2+" es menor que "+n1);
21     System.out.print(txt);
22 }
23
24 public static void main(String args[]) {
25     int numero1=leerEteroTeclado();
26     int numero2=leerEteroTeclado();
27     isPositivo(numero1);
28     isPositivo(numero2);
29     isMultiplo(numero1, numero2);
30     isMenor(numero1, numero2);
31 }
32 }
```

Problems Javadoc Declaration Console X

terminado> Sentecias_If [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Introduce un número:Exception in thread "main" java.lang.IllegalStateException: Scanner closed
at java.base/java.util.Scanner.ensureOpen(Scanner.java:1154)
at java.base/java.util.Scanner.next(Scanner.java:1577)
at java.base/java.util.Scanner.nextInt(Scanner.java:2263)
at java.base/java.util.Scanner.nextInt(Scanner.java:2217)
at Sentecias_If.leerEteroTeclado(Sentecias_If.java:9)
at Sentecias_If.main(Sentecias_If.java:25)

1.6 Ejercicio 6: Crear ejercicios de ejemplo para usar las estructuras de control (sentencia switch case) .

Este paquete java.time incluye muchas clases, pero las básicas son:

LocalDate: representa a fechas sin la hora y nos facilita su manejo para declararlas, sumar y restar fechas y compararlas.

LocalTime: es idéntica a la anterior pero para el manejo de horas, sin ninguna fecha asociada, pudiendo así compararlas, sumar o restar tiempo a las mismas...

LocalDateTime: como puedes suponer, es una combinación de las dos anteriores, que permite hacer lo mismo con fechas y horas simultáneamente.

Instant: es muy parecida a la anterior pero a la vez muy diferente. Se usa para almacenar un punto determinado en el tiempo, o sea con fecha y hora, pero guarda su valor como un timestamp de UNIX, es decir, en nanosegundos desde el epoch de UNIX (1/1/1970 a las 00:00) y usando la zona horaria UTC. Es muy útil para manejar momentos en el tiempo de manera neutra e intercambiarlo entre aplicaciones y sistemas, por lo que lo verás utilizado muy a menudo.

ZonedDateTime: esta clase es como la `LocalDateTime` pero teniendo en cuenta una zona horaria concreta, ya que las anteriores no la tienen en cuenta.

Period: esta clase auxiliar nos ayuda a obtener diferencias entre fechas en distintos periodos (segundos, minutos, días...) y también a añadir esas diferencias a las fechas.

Duration: esta es muy parecida a la anterior pero para manejo de horas exclusivamente..

Construyendo fechas y horas con `java.time`

Estas clases producen instancias inmutables, al contrario de lo que pasaba con las antiguas clases `Date` de Java, por lo que son thread-safe. Dado que carecen de constructores públicos, se instancian usando métodos de tipo "factoría", es decir, tienen métodos que construyen estas clases a partir de posibles parámetros que le pasemos.

En concreto, todas las de manejo de fechas y horas disponen de tres métodos importantes, que son:

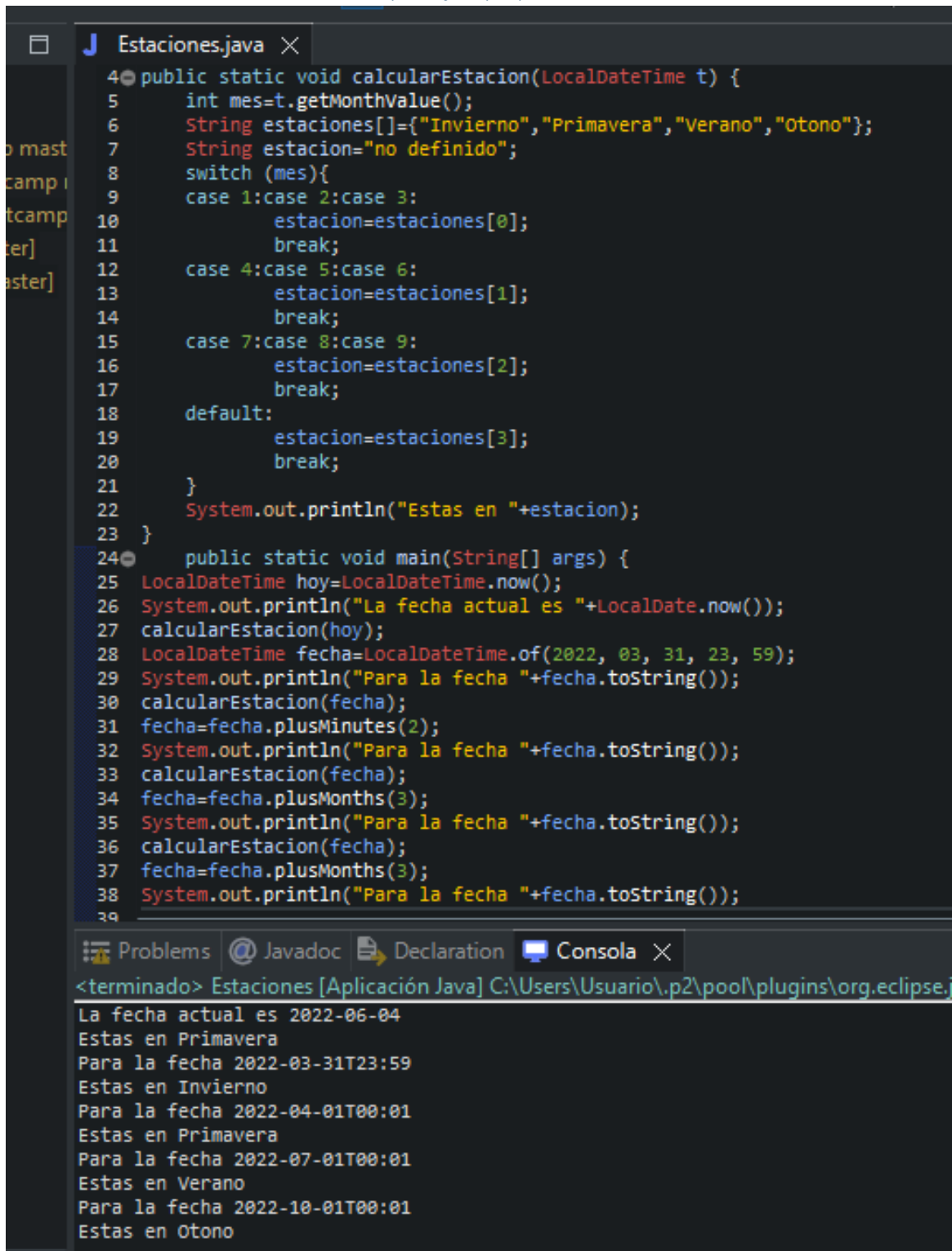
`now()`: crean instancias nuevas a partir de la fecha y hora actual.

```
System.out.println("La fecha actual es: " + LocalDate.now());
System.out.println("La hora actual es: " + LocalTime.now());
System.out.println("La fecha y hora actuales son: " + LocalDateTime.now());
System.out.println("El instante actual es: " + Instant.now());
System.out.println("La fecha y hora actuales con zona horaria son: " +
ZonedDateTime.now());
```

`of()`: construyen fechas y horas a partir de sus partes.

`with()`: modifican la fecha u hora actual en función del parámetro que se le pase, con alguna cantidad (años, días, horas...) o alguna clase de ajuste.

- 1.6.1 Comprobar cuál es la estación del año (se pueden hacer por meses... Enero, Febrero, Marzo es INVIERNO, por ejemplo) .



```
Estaciones.java X
4 public static void calcularEstacion(LocalDate t) {
5     int mes=t.getMonthValue();
6     String estaciones[]={"Invierno","Primavera","Verano","Otono"};
7     String estacion="no definido";
8     switch (mes){
9         case 1:case 2:case 3:
10             estacion=estaciones[0];
11             break;
12         case 4:case 5:case 6:
13             estacion=estaciones[1];
14             break;
15         case 7:case 8:case 9:
16             estacion=estaciones[2];
17             break;
18         default:
19             estacion=estaciones[3];
20             break;
21     }
22     System.out.println("Estas en "+estacion);
23 }
24 public static void main(String[] args) {
25     LocalDateTime hoy=LocalDateTime.now();
26     System.out.println("La fecha actual es "+LocalDate.now());
27     calcularEstacion(hoy);
28     LocalDateTime fecha=LocalDateTime.of(2022, 03, 31, 23, 59);
29     System.out.println("Para la fecha "+fecha.toString());
30     calcularEstacion(fecha);
31     fecha=fecha.plusMinutes(2);
32     System.out.println("Para la fecha "+fecha.toString());
33     calcularEstacion(fecha);
34     fecha=fecha.plusMonths(3);
35     System.out.println("Para la fecha "+fecha.toString());
36     calcularEstacion(fecha);
37     fecha=fecha.plusMonths(3);
38     System.out.println("Para la fecha "+fecha.toString());
39 }
```

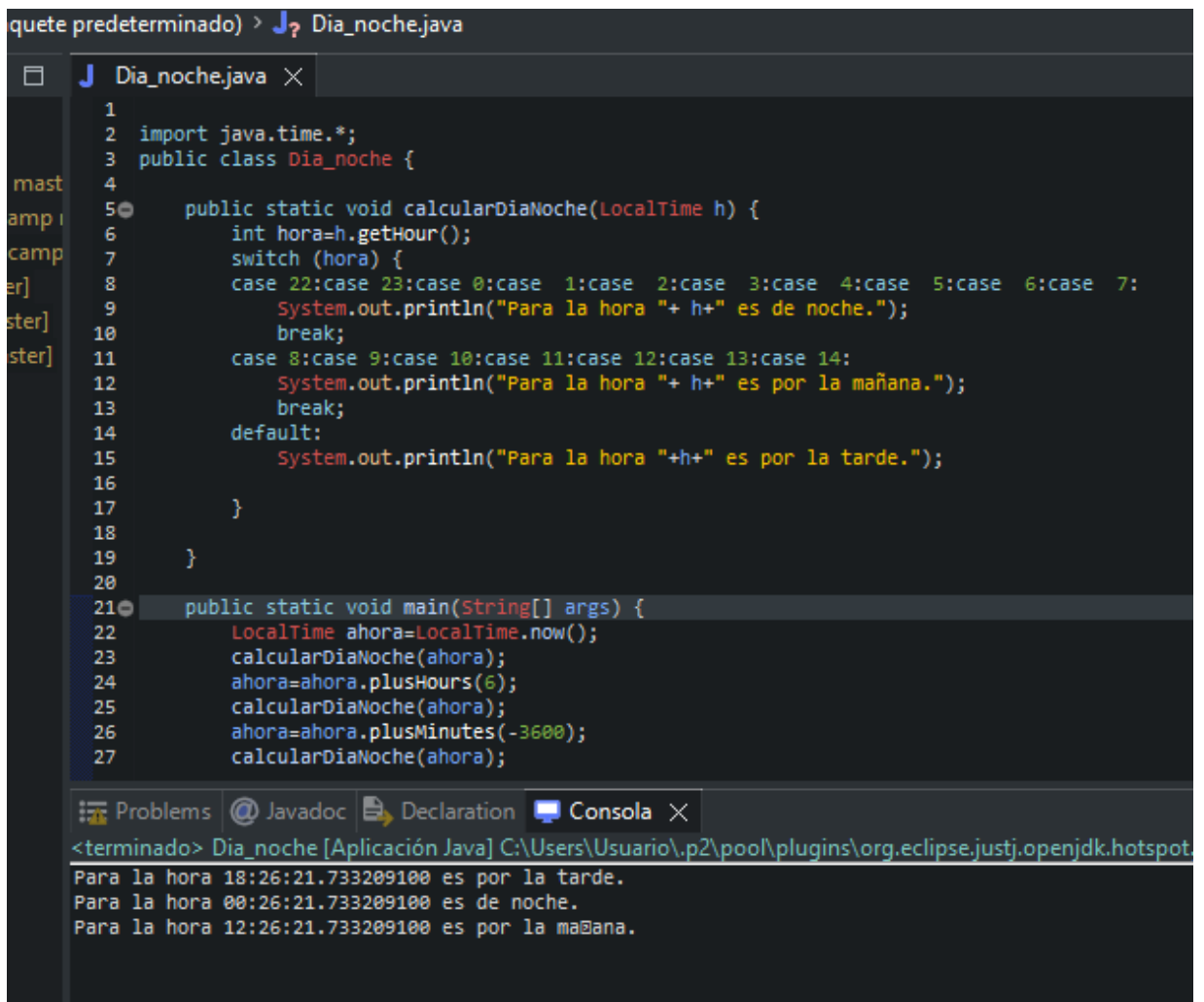
Problems Javadoc Declaration Consola X

<terminado> Estaciones [Aplicación Java] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.j

La fecha actual es 2022-06-04
Estas en Primavera
Para la fecha 2022-03-31T23:59
Estas en Invierno
Para la fecha 2022-04-01T00:01
Estas en Primavera
Para la fecha 2022-07-01T00:01
Estas en Verano
Para la fecha 2022-10-01T00:01
Estas en Otono

Ilustración 13 Cálculo de estaciones.

1.6.2 Comprobar si es de mañana tarde o noche :

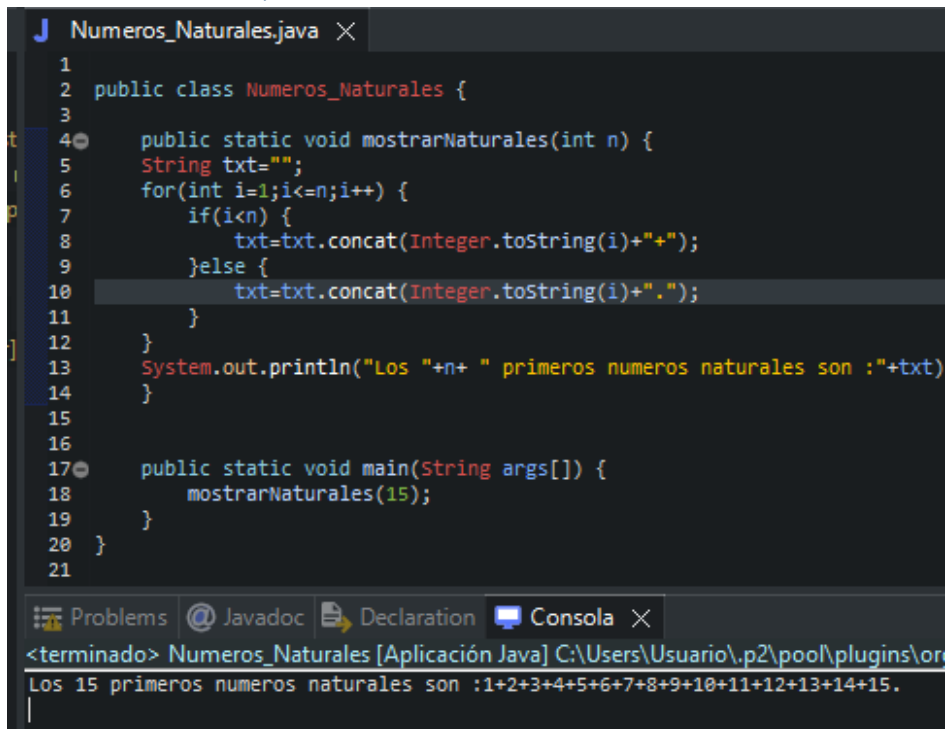


```
quete predeterminado) > J? Dia_noche.java
J Dia_noche.java X
1
2 import java.time.*;
3 public class Dia_noche {
4
5     public static void calcularDiaNoche(LocalTime h) {
6         int hora=h.getHour();
7         switch (hora) {
8             case 22:case 23:case 0:case 1:case 2:case 3:case 4:case 5:case 6:case 7:
9             System.out.println("Para la hora "+ h+" es de noche.");
10            break;
11            case 8:case 9:case 10:case 11:case 12:case 13:case 14:
12            System.out.println("Para la hora "+ h+" es por la mañana.");
13            break;
14            default:
15            System.out.println("Para la hora "+h+" es por la tarde.");
16
17        }
18
19    }
20
21    public static void main(String[] args) {
22        LocalTime ahora=LocalTime.now();
23        calcularDiaNoche(ahora);
24        ahora=ahora.plusHours(6);
25        calcularDiaNoche(ahora);
26        ahora=ahora.plusMinutes(-3600);
27        calcularDiaNoche(ahora);
28    }
29 }
30
31 Problems Javadoc Declaration Console X
32 <terminado> Dia_noche [Aplicación Java] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
33 Para la hora 18:26:21.733209100 es por la tarde.
34 Para la hora 00:26:21.733209100 es de noche.
35 Para la hora 12:26:21.733209100 es por la mañana.
```

Ilustración 14 cálculo de día , tarde o noche

1.7 Ejercicio 7: estructuras de control:

1.7.1 Mostrar los primeros 15 números naturales.



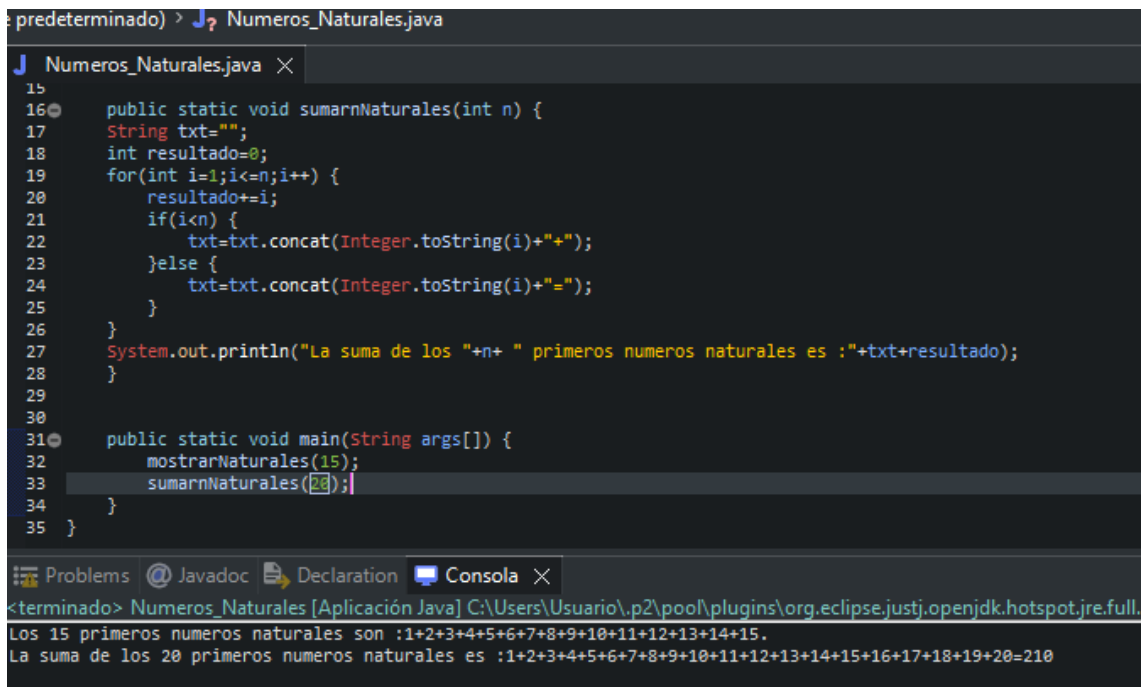
```
1 public class Numeros_Naturales {
2
3
4     public static void mostrarNaturales(int n) {
5         String txt="";
6         for(int i=1;i<=n;i++) {
7             if(i<n) {
8                 txt=txt.concat(Integer.toString(i)+"");
9             }else {
10                txt=txt.concat(Integer.toString(i)+".");
11            }
12        }
13        System.out.println("Los "+n+ " primeros numeros naturales son :"+txt)
14    }
15
16
17    public static void main(String args[]) {
18        mostrarNaturales(15);
19    }
20 }
21
```

Problems Javadoc Declaration Console X

<terminado> Numeros_Naturales [Aplicación Java] C:\Users\Usuario\.p2\pool\plugins\or
Los 15 primeros numeros naturales son :1+2+3+4+5+6+7+8+9+10+11+12+13+14+15.
|

Ilustración 15 n primeros naturales

1.7.2 Sumar los primeros 100 números naturales.



```
15
16     public static void sumarNaturales(int n) {
17         String txt="";
18         int resultado=0;
19         for(int i=1;i<=n;i++) {
20             resultado+=i;
21             if(i<n) {
22                 txt=txt.concat(Integer.toString(i)+"");
23             }else {
24                 txt=txt.concat(Integer.toString(i)+"=");
25             }
26         }
27         System.out.println("La suma de los "+n+ " primeros numeros naturales es :"+txt+resultado);
28     }
29
30
31     public static void main(String args[]) {
32         mostrarNaturales(15);
33         sumarNaturales(20);
34     }
35 }

```

Problems Javadoc Declaration Console X

<terminado> Numeros_Naturales [Aplicación Java] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.
Los 15 primeros numeros naturales son :1+2+3+4+5+6+7+8+9+10+11+12+13+14+15.
La suma de los 20 primeros numeros naturales es :1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20=210

Ilustración 16 suma de los n números naturales

1.7.3 Buscar la posición de un elemento concreto en un array .

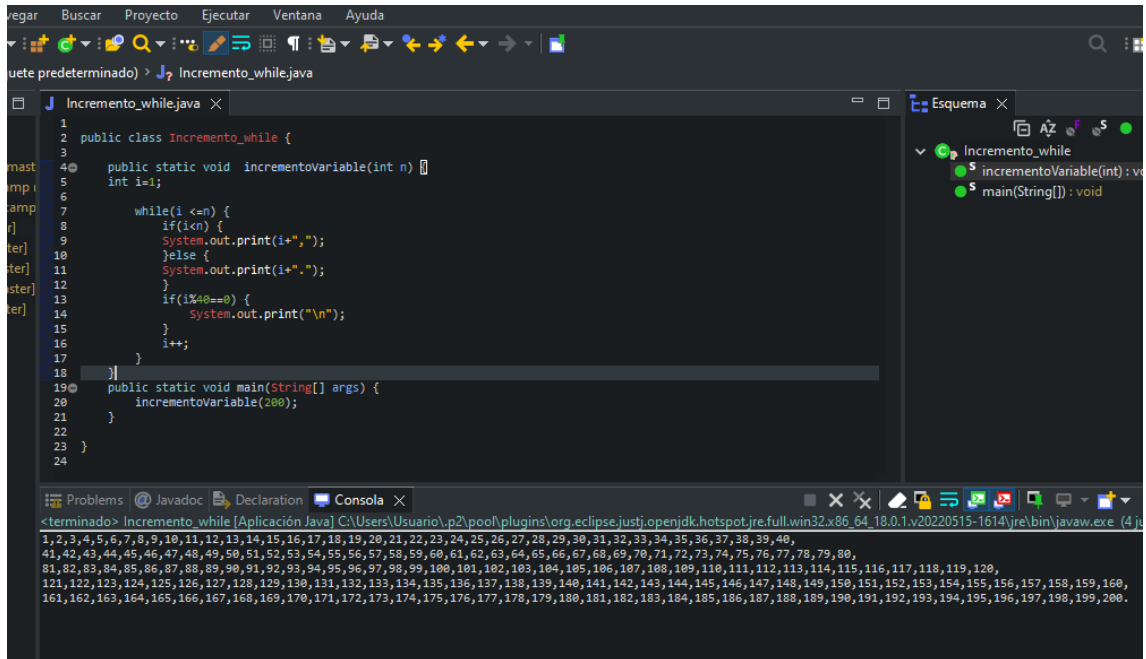
```

J Numeros_Naturales.java X
29 }
30 public static void mostrarMatiz(int a[]) {
31     System.out.print("");
32     for(int j=0;j<a.length;j++) {
33         if(j<a.length-1) {
34             System.out.print(a[j]+",");
35         } else {
36             System.out.println(a[j]+"");
37         }
38     }
39 public static void buscarElementoenArray(int ele,int a[]) {
40     int pos[]=new int [a.length];
41     int sum=0;
42     mostrarMatiz(a);
43     for(int i=0;i<a.length;i++) {
44         if(a[i]==ele) {
45             pos[i]=1;
46             System.out.println("Elemento "+ele+" encontrado en posicion :"+(i+1));
47         } else {
48             pos[i]=0;
49         }
50         sum+=pos[i];
51     }
52     System.out.println("Elemento "+ele+" se ha encontrada :"+sum+" veces");
53     mostrarMatiz(a);
54     mostrarMatiz(pos);
55 }
56
57
58 public static void main(String args[]) {
59     mostrarNaturales(15);
60     sumarnNaturales(20);
61     int s[]= {44,44,2,1,2,2,2,2,3,4,1,44,44,2};
62
Problems Javadoc Declaration Console X
<terminado> Numeros_Naturales [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Los 15 primeros numeros naturales son :1+2+3+4+5+6+7+8+9+10+11+12+13+14+15.
La suma de los 20 primeros numeros naturales es :1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20=210
{44,44,2,1,2,2,2,2,3,4,1,44,44,2}
Elemento 44 encontrado en posicion :1
Elemento 44 encontrado en posicion :2
Elemento 44 encontrado en posicion :12
Elemento 44 encontrado en posicion :13
Elemento 44 se ha encontrada :4 veces
{44,44,2,1,2,2,2,2,3,4,1,44,44,2}
{1,1,0,0,0,0,0,0,0,0,0,1,1,0}

```

Ilustración 17 Búsqueda en un Array

1.8 Ejercicio 8: estructuras de control. Realizar un bucle mientras una variable incrementa su valor hasta un tope (usar while) .



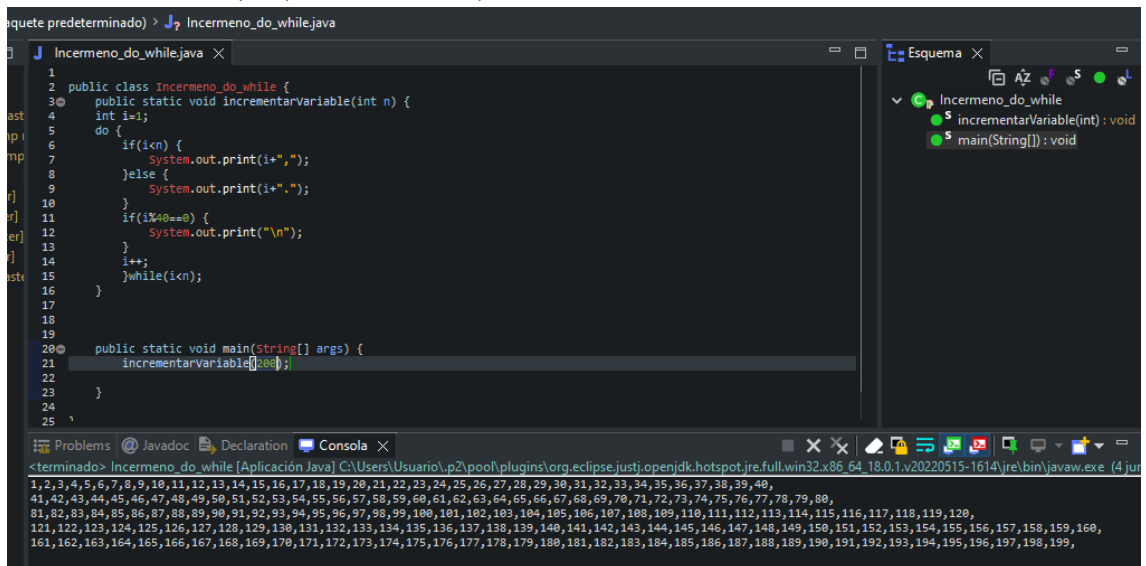
```
1 public class Incremento_while {
2
3
4     public static void incrementoVariable(int n) {
5         int i=1;
6
7         while(i <=n) {
8             if(i<n) {
9                 System.out.print(i+",");
10            }else {
11                System.out.print(i+",");
12            }
13            if(i%40==0) {
14                System.out.print("\n");
15            }
16            i++;
17        }
18    }
19    public static void main(String[] args) {
20        incrementoVariable(200);
21    }
22 }
23
24
```

Console output:

```
<terminado> Incremento_while [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.18.0.1.v20220515-1614\jre\bin\javaw.exe (4 j
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,
121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,
161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200.
```

Ilustración 18 lista while

1.9 Ejercicio 9: Realizar un bucle mientras una variable incrementa su valor hasta un tope (usar do while) .



```
1 public class Incerneno_do_while {
2
3
4     public static void incrementarVariable(int n) {
5         int i=1;
6         do {
7             if(i<n) {
8                 System.out.print(i+",");
9             }else {
10                System.out.print(i+",");
11            }
12            if(i%40==0) {
13                System.out.print("\n");
14            }
15            i++;
16        }while(i<n);
17    }
18
19
20    public static void main(String[] args) {
21        incrementarVariable(200);
22    }
23 }
24
25
```

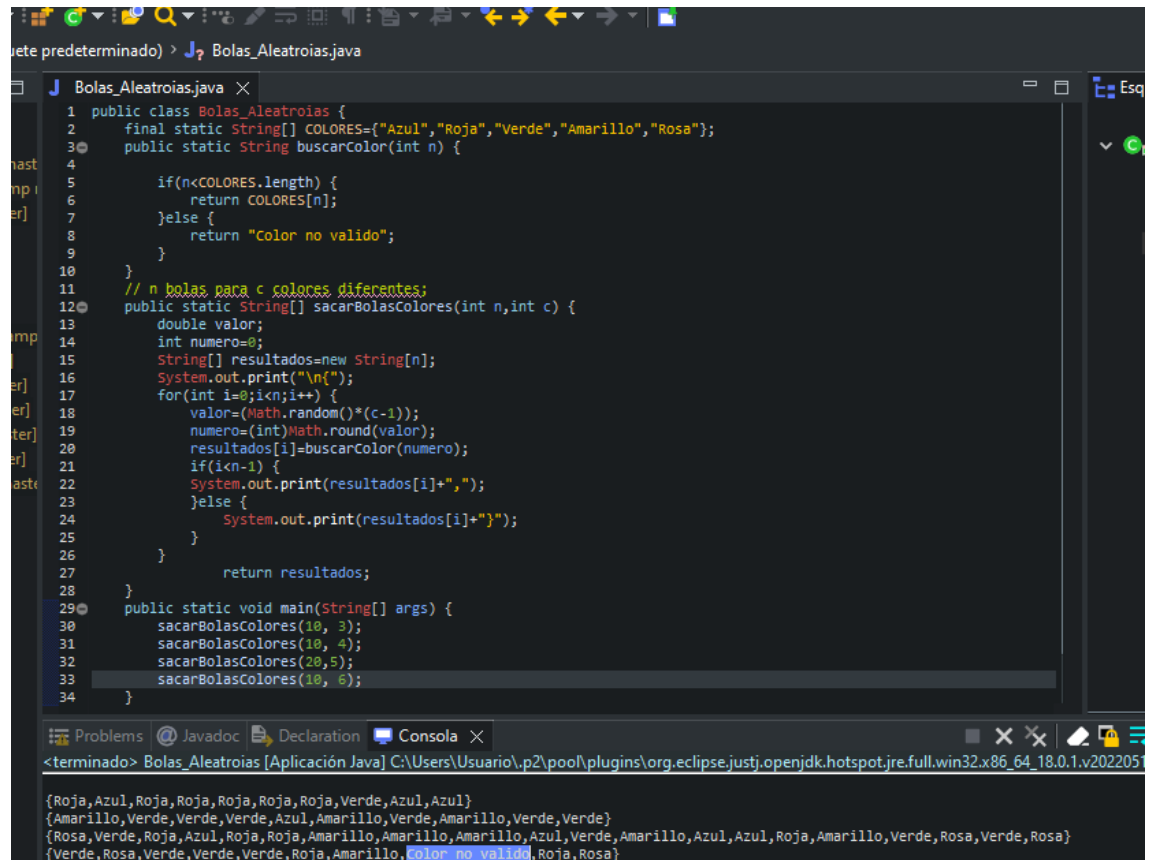
Console output:

```
<terminado> Incerneno_do_while [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.18.0.1.v20220515-1614\jre\bin\javaw.exe (4 j
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,
121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,
161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,
```

Ilustración 19 do-while

1.10 Ejercicio 10:

1.11 Sacar 10 bolas de color aleatoriamente (azul, rojas y verdes).



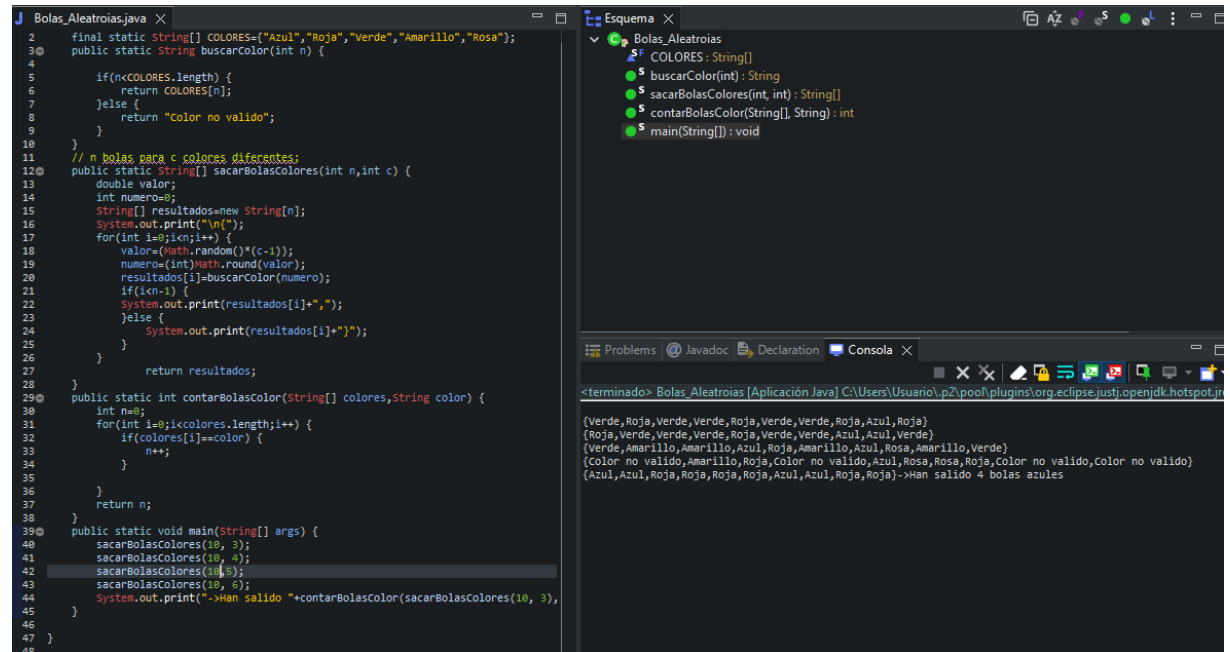
```
1 public class Bolas_Aleatorias {
2     final static String[] COLORES={"Azul","Roja","Verde","Amarillo","Rosa"};
3     public static String buscarColor(int n) {
4
5         if(n<COLORES.length) {
6             return COLORES[n];
7         }else {
8             return "Color no valido";
9         }
10    }
11    // n bolas para c colores diferentes;
12    public static String[] sacarBolasColores(int n,int c) {
13        double valor;
14        int numero=0;
15        String[] resultados=new String[n];
16        System.out.print("\n");
17        for(int i=0;i<n;i++) {
18            valor=(Math.random()*(c-1));
19            numero=(int)Math.round(valor);
20            resultados[i]=buscarColor(numero);
21            if(i<n-1) {
22                System.out.print(resultados[i]+",");
23            }else {
24                System.out.print(resultados[i]+"");
25            }
26        }
27        return resultados;
28    }
29    public static void main(String[] args) {
30        sacarBolasColores(10, 3);
31        sacarBolasColores(10, 4);
32        sacarBolasColores(20,5);
33        sacarBolasColores(10, 6);
34    }
```

Console Output:

```
<terminado> Bolas_Aleatorias [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_18.0.1.v20220518\jre\bin\java.exe
{Roja,Azul,Roja,Roja,Roja,Roja,Roja,Verde,Azul,Azul}
{Amarillo,Verde,Verde,Verde,Azul,Amarillo,Verde,Amarillo,Verde,Verde}
{Rosa,Verde,Roja,Azul,Roja,Roja,Amarillo,Amarillo,Azul,Verde,Amarillo,Verde,Rosa,Verde,Rosa}
{Verde,Rosa,Verde,Verde,Verde,Roja,Amarillo,color no valido,Roja,Rosa}
```

Ilustración 20 Sacar Bolas Aleatorias

1.12 Contar cuantas bolas azules salen en total.



```
1 final static String[] COLORES={"Azul","Roja","Verde","Amarillo","Rosa"};
2 public static String buscarColor(int n) {
3
4     if(n<COLORES.length) {
5         return COLORES[n];
6     }else {
7         return "Color no valido";
8     }
9 }
10 // n bolas para c colores diferentes;
11 public static String[] sacarBolasColores(int n,int c) {
12     double valor;
13     int numero=0;
14     String[] resultados=new String[n];
15     System.out.print("\n");
16     for(int i=0;i<n;i++) {
17         valor=(Math.random()*(c-1));
18         numero=(int)Math.round(valor);
19         resultados[i]=buscarColor(numero);
20         if(i<n-1) {
21             System.out.print(resultados[i]+",");
22         }else {
23             System.out.print(resultados[i]+"");
24         }
25     }
26     return resultados;
27 }
28 public static int contarBolasColor(String[] colores,String color) {
29     int n=0;
30     for(int i=0;icolores.length;i++) {
31         if(colores[i]==color) {
32             n++;
33         }
34     }
35     return n;
36 }
37 public static void main(String[] args) {
38     sacarBolasColores(10, 3);
39     sacarBolasColores(10, 4);
40     sacarBolasColores(10, 5);
41     sacarBolasColores(10, 6);
42     System.out.print("\nHan salido "+contarBolasColor(sacarBolasColores(10, 3),
43     "Azul"));
44 }
```

Console Output:

```
<terminado> Bolas_Aleatorias [Aplicación Java] C:\Users\Usuario\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_18.0.1.v20220518\jre\bin\java.exe
{Verde,Roja,Verde,Verde,Roja,Verde,Verde,Roja,Azul,Roja}
{Roja,Verde,Verde,Verde,Roja,Verde,Verde,Azul,Azul,Verde}
{Verde,Amarillo,Amarillo,Azul,Roja,Amarillo,Azul,Rosa,Amarillo,Verde}
{Color no valido,Amarillo,Roja,color no valido,Azul,Rosa,Rosa,Roja,color no valido,color no valido}
{Azul,Azul,Roja,Roja,Roja,Roja,Azul,Azul,Roja,Roja}->Han salido 4 bolas azules
```

Ilustración 21 contar bolas de color Azul

1.13 Mostrar qué color sale hasta que aparezca la primera azul (incluida la azul).

The image shows a screenshot of an IDE with two main panels. The left panel displays the source code for a Java class named `Bolas_Aleatoires`. The right panel shows the execution output in the console.

Source Code (Bolas_Aleatoires.java):

```
1 public class Bolas_Aleatoires {
2     final static String[] COLORES={"Azul","Roja","Verde","Amarillo","Rosa"};
3     public static String buscarColor(int n) {
4
5         if(n<COLORES.length) {
6             return COLORES[n];
7         } else {
8             return "Color no valido";
9         }
10    }
11    // n bolas saca c colores aleatorios
12    public static String[] sacarBolasColores(int n,int c) {
13        double valor;
14        int numero=0;
15        String[] resultados=new String[n];
16        System.out.print("\n");
17        for(int i=0;i<n;i++) {
18            valor=(Math.random()*(c-1));
19            numero=(int)Math.round(valor);
20            resultados[i]=buscarColor(numero);
21            if(i<n-1) {
22                System.out.print(resultados[i]+",");
23            } else {
24                System.out.print(resultados[i]+"");
25            }
26        }
27        return resultados;
28    }
29    public static int contarBolasColor(String[] colores,String color) {
30        int n=0;
31        for(int i=0;i<colores.length;i++) {
32            if(colores[i]==color) {
33                n++;
34            }
35        }
36        return n;
37    }
38    public static void sacarBolasHastaColor(int n,int ncolores,String c) {
39        double valor;
40        int numero=0;
41        System.out.print("\n");
42        String[] resultados=new String[n];
43        int i=0;
44        do {
45            valor=(Math.random()*(ncolores-1));
46            numero=(int)Math.round(valor);
47            resultados[i]=buscarColor(numero);
48            if(i<n-1) {
49                System.out.print(resultados[i]+",");
50            } else {
51                System.out.print(resultados[i]+"");
52            }
53            i++;
54        }while((c!=resultados[i-1])&&(i<n));
55    }
56
57    public static void main(String[] args) {
58        sacarBolasColores(10, 3);
59        sacarBolasColores(10, 4);
60        sacarBolasColores(10,5);
61        sacarBolasColores(10, 6);
62        System.out.print("\nHan salido "+contarBolasColor(sacarBolasColores(10, 3), COLORES[0])+" bolas azules");
63        sacarBolasHastaColor(10, 5,"Azul");
64        sacarBolasHastaColor(20, 5,"Amarillo");
65    }
66 }
67
68 }
```

Execution Output (Console):

```
<terminado> Bolas_Aleatoires [Aplicación Java] C:\Users\Usuari...
(Verde,Azul,Roja,Roja,Roja,Verde,Roja,Roja,Verde,Roja)
(Roja,Amarillo,Roja,Roja,Amarillo,Roja,Amarillo,Verde,Amarillo,Amarillo)
(Azul,Amarillo,Amarillo,Verde,Verde,Roja,Verde,Verde,Rosa,Amarillo)
(Color no valido,Roja,Azul,Roja,Rosa,Color no valido,Rosa,Color no valido,Roja,Amarillo)
(Azul,Roja,Roja,Roja,Roja,Azul,Azul,Roja,Verde,Roja)->Han salido 3 bolas azules
(Roja,Verde,Roja,Verde,Roja,Roja,Azul,
(Azul,Rosa,Roja,Roja,Rosa,Verde,Roja,Verde,Amarillo,
```