

## area 11 – Repaso UD2 II – While – for - Switch

### Enunciado

1. Escribe un método llamado `imprimeNumeroEnLetra`. El método tiene un parámetro llamado `numero` que es entero. El método debe imprimir "CERO", "UNO", "DOS", ... "NUEVE", "OTRO" si el parámetro `numero` es 0, 1, 2, .... 9 u otro para cualquier otro número incluidos los números negativos. Usa la estructura `switch`.

Ayuda: El método `imprimeNumeroEnLetra` debe ser `public static`.

```
public class ImprimeNumeroEnLetra {  
    public static void main(String[] args) {  
        System.out.println("Vamos a convertir numeros en letras por un  
método.");  
        System.out.println("Por favor, introduce un numero:");  
        Scanner sc=new Scanner(System.in);  
        int numeroNumero = sc.nextInt();  
        System.out.println(imprimeNumeroEnLetra(numeroNumero));  
    }  
    public static String imprimeNumeroEnLetra(int numeroNumero) {  
        String texto="Nada";  
        if (numeroNumero < 0 || numeroNumero > 9) {  
            return texto="OTRO";  
        }else{  
            switch (numeroNumero) {  
                case 0:  
                    return texto="CERO";  
  
                case 1:  
                    return texto="UNO";  
                case 2:  
                    return texto="DOS";  
                case 3:  
                    return texto="TRES";  
                case 4:  
                    return texto="CUATRO";  
                case 5:  
                    return texto="CINCO";  
                case 6:  
                    return texto="SEIS";  
                case 7:  
                    return texto="SIETE";  
                case 8:  
                    return texto="OCHO";  
                case 9:  
                    return texto="NUEVE";  
            }  
        }  
        return texto;  
    }  
}
```

<pre>Vamos a convertir numeros en letras por un método. Por favor, introduce un numero: -222 OTRO</pre>	<pre>Vamos a convertir numeros en letras por un método. Por favor, introduce un numero: 8 OCHO</pre>
---	--

2. Escribe un método llamado `esImpar` con un parámetro `int` y llámalo `numero`. El método debe devolver un valor booleano.

Verifica que el número sea  $> 0$ , si no es falso, el método devuelve falso.

Si el número es impar, devuelve verdadero; de lo contrario, devuelve falso.

Escribe un segundo método llamado `sumarImpares` que tenga 2 parámetros, uno `int` inicio y otro `int` fin, que representan un rango de números. El método devuelve un `int`.

El método debe usar un bucle `for` para sumar todos los números impares en ese rango, incluido el final y devolver la suma. El bucle debe llamar al método `esImpar` para comprobar si cada número que se itera dentro del bucle es impar.

El parámetro fin debe ser mayor o igual que el inicio y los parámetros de inicio y fin deben ser mayores que 0.

Si esas condiciones no se cumplen, el método devuelve -1 para indicar una entrada no válida.

Ejemplo input/output:

\* `sumarImpares(1, 100)`; → debe devolver 2500

\* `sumarImpares(-1, 100)`; → debe devolver -1

\* `sumarImpares(100, 100)`; → debe devolver 0

\* `sumarImpares(13, 13)`; → debe devolver 13 (este conjunto contiene un número, 13, y es impar)

\* `sumarImpares(100, -100)`; → debe devolver -1

\* `sumarImpares(100, 1000)`; → debe devolver 247500

Ayuda: para saber si un número es par se usa: `i % 2 == 0`

Ambos métodos deben definirse como `public static`, como lo hemos estado haciendo hasta ahora en el curso.

El método `esImpar` se utiliza dentro del método `sumarImpares`.

```
public class MetodoImparOPar {
    public static void main(String[] args) {
        System.out.println("Vamos a sumar números impares por un método.");
        System.out.println("Por favor, introduce un numero inicial:");
        Scanner sc=new Scanner(System.in);
        int inicio = sc.nextInt();
        System.out.println("Por favor, introduce hasta que número quieres llegar:");
        int fin = sc.nextInt();
        System.out.println(sumarImpares(inicio, fin));
    }
    public static int sumarImpares (int inicio, int fin){
        int sumaFinalImpares=0;
        if(inicio<0 || fin<0) {
            sumaFinalImpares = -1;
        }else {
            for (int i = inicio; i <= fin; i++) {
                if (esImpar(i) == true) {
                    sumaFinalImpares = sumaFinalImpares + i;
                }
            }
        }
        return sumaFinalImpares;
    }
}
```

```

    }
}
}
return sumaFinalImpares;
}
public static boolean esImpar(int i){
    boolean salida=false;
    if (i<1){
        salida=false;
    } else if (i%2!=0) {
        salida=true;
    }
    return salida;
}
}
}

```

```

▶ public class MetodoImparOPar {
▶     public static void main(String[] args) {
        System.out.println("Vamos a sumar números impares por un método.");
        System.out.println("Por favor, introduce un número inicial:");
        Scanner sc=new Scanner(System.in);
        int inicio = sc.nextInt();
        System.out.println("Por favor, introduce hasta que número quieres llegar:");
        int fin = sc.nextInt();
        System.out.println(sumarImpares(inicio, fin));
    }
1 usage
    public static int sumarImpares (int inicio, int fin){
        int sumaFinalImpares=0;
        if(inicio<0 || fin<0) {
            sumaFinalImpares = -1;
        }else {
            for (int i = inicio; i <= fin; i++) {
                if (esImpar(i) == true) {
                    sumaFinalImpares = sumaFinalImpares + i;
                }
            }
        }
        return sumaFinalImpares;
    }
1 usage
    public static boolean esImpar(int i){
        boolean salida=false;
        if (i<1){
            salida=false;
        } else if (i%2!=0) {
            salida=true;
        }
        return salida;
    }
}

```

ImprimeNumeroEnLetra x MetodoImparOPar x

```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intel
Vamos a sumar números impares por un método.
Por favor, introduce un número inicial:
-2
Por favor, introduce hasta que número quieres llegar:
10
-1

```

```
public class MetodoImparOPar {
    public static void main(String[] args) {
        System.out.println("Vamos a sumar números impares por un método.");
        System.out.println("Por favor, introduce un número inicial:");
        Scanner sc=new Scanner(System.in);
        int inicio = sc.nextInt();
        System.out.println("Por favor, introduce hasta que número quieres llegar:");
        int fin = sc.nextInt();
        System.out.println(sumarImpares(inicio, fin));
    }

    1 usage
    public static int sumarImpares (int inicio, int fin){
        int sumaFinalImpares=0;
        if(inicio<0 || fin<0) {
            sumaFinalImpares = -1;
        }else {
            for (int i = inicio; i <= fin; i++) {
                if (esImpar(i) == true) {
                    sumaFinalImpares = sumaFinalImpares + i;
                }
            }
        }
        return sumaFinalImpares;
    }

    1 usage
    public static boolean esImpar(int i){
        boolean salida=false;
        if (i<1){
            salida=false;
        } else if (i%2!=0) {
            salida=true;
        }
        return salida;
    }
}

ImprimeNumeroEnLetra x  MetodoImparOPar x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea-agent-1.0.0.jar" -Dfile.encoding=UTF-8
Vamos a sumar números impares por un método.
Por favor, introduce un número inicial:
100
Por favor, introduce hasta que número quieres llegar:
1000
247500
```

3. Escribe un método llamado `imprimirFactores` que recibe un parámetro entero llamado `n`.

Si el `n < 1` el método imprime "Valor inválido". El método debe imprimir todos los factores del número. Un factor de un número es un entero que es divisible con resto 0 por ese número.

Por ejemplo, 3 es un factor de 6, porque  $6/3=2$  resto 0.

Ejemplo input/output

`imprimirFactores(10)` → devuelve:

1  
2  
5  
10

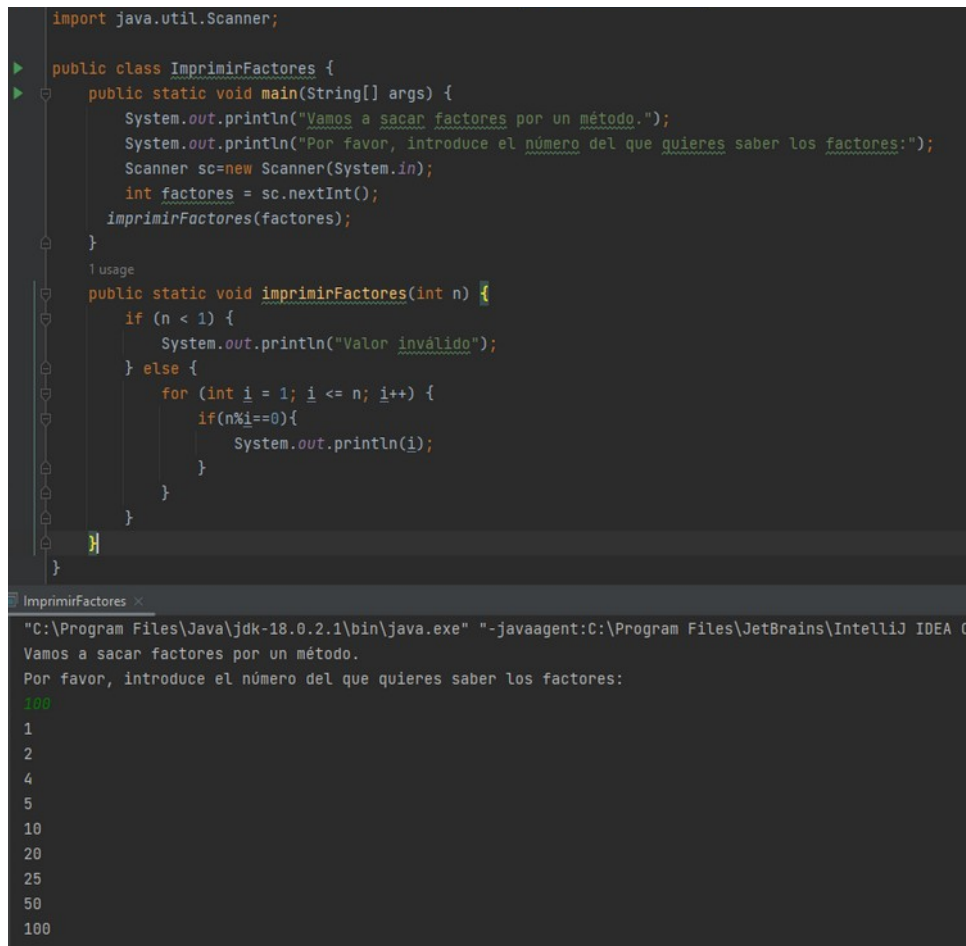
`imprimirFactores(6)`; → devuelve: 1 2 3 6 (en diferentes líneas)

`imprimirFactores(-4)`; → devuelve: Valor inválido

```

public class ImprimirFactores {
    public static void main(String[] args) {
        System.out.println("Vamos a sacar factores por un método.");
        System.out.println("Por favor, introduce el número del que
quieres saber los factores:");
        Scanner sc=new Scanner(System.in);
        int factores = sc.nextInt();
        imprimirFactores(factores);
    }
    public static void imprimirFactores(int n) {
        if (n < 1) {
            System.out.println("Valor inválido");
        } else {
            for (int i = 1; i <= n; i++) {
                if(n%i==0){
                    System.out.println(i);
                }
            }
        }
    }
}

```



```

import java.util.Scanner;

public class ImprimirFactores {
    public static void main(String[] args) {
        System.out.println("Vamos a sacar factores por un método.");
        System.out.println("Por favor, introduce el número del que quieres saber los factores:");
        Scanner sc=new Scanner(System.in);
        int factores = sc.nextInt();
        imprimirFactores(factores);
    }
    public static void imprimirFactores(int n) {
        if (n < 1) {
            System.out.println("Valor inválido");
        } else {
            for (int i = 1; i <= n; i++) {
                if(n%i==0){
                    System.out.println(i);
                }
            }
        }
    }
}

```

ImprimirFactores x

```

"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 0
Vamos a sacar factores por un método.
Por favor, introduce el número del que quieres saber los factores:
100
1
2
4
5
10
20
25
50
100

```

```
public class ImprimirFactores {
    public static void main(String[] args) {
        System.out.println("Vamos a sacar factores por un método.");
        System.out.println("Por favor, introduce el número del que quieres saber los factores:");
        Scanner sc=new Scanner(System.in);
        int factores = sc.nextInt();
        imprimirFactores(factores);
    }
}

1 usage
public static void imprimirFactores(int n) {
    if (n < 1) {
        System.out.println("Valor inválido");
    } else {
        for (int i = 1; i <= n; i++) {
            if(n%i==0){
                System.out.println(i);
            }
        }
    }
}
}
```

ImprimirFactores x

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE
Vamos a sacar factores por un método.
Por favor, introduce el número del que quieres saber los factores:
-10
Valor inválido
```

4. Escribe un método llamado esPalindromo con un parámetro entero llamado n. El método devuelve un booleano. El valor devuelto será true cuando el número es un palíndromo, en otro caso devolverá false.

¿Qué es un número palíndromo? Es un número que si le damos la vuelta, es decir, que si lo leemos de atrás hacia adelante es el mismo número. Por ejemplo, 12321, 121, 1001, etc.

Ayuda. Para verificar si un número es palíndromo deberemos calcular el número en reverso y almacenarlo en una variable para al final comparar si los dos números son iguales.

Cálculo de un número reverso. Primero crea e inicializa otra variable llamada reverso = 0.

Extrae el último dígito del número n utilizando el módulo y almacénalo en otra variable: ultimoDigito = n % 10.

Aumenta el valor posicional de reverso en uno, para hacer esto tienes que multiplicar la variable reverso por 10: reverso = reverso \* 10;  
Suma a reverso el último dígito: reverso +=ultimoDigito;  
Como el último dígito del número n ha sido procesado, elimínalo del número utilizando la / 10:

n = n / 10;

Repite estos pasos hasta que el número n sea 0.

Utiliza la estructura while.

Fuera del bucle, compara el número original introducido con reverso. No te olvides de guardar el número original introducido en alguna variable ya que n se va modificando.

```

public class JugandoConPalindromos {
    public static void main(String[] args) {

        System.out.println(esPalindromo(1221));
    }
    public static boolean esPalindromo(int n) {
        int reverso = 0;
        int nCalcular=n;
        while (nCalcular > 0) {
            int ultimoDigito = nCalcular % 10;

            reverso = reverso* 10 + (ultimoDigito);
            nCalcular = nCalcular / 10;
        }
        if(reverso==n){
            return true;
        }else {
            return false;
        }
    }
}

```

5. Escribe un método llamado insertaImprimeSumaYMedia sin parámetros y sin devolver nada.

El método leerá números introducidos por el usuario por pantalla hasta que el usuario inserte algo que no es un número. Entonces el programa saldrá del while e imprimirá la suma y la media de los números introducidos con formato: SUM = XX AVG = YY, donde XX representa la suma de los datos introducidos y YY la media.

Ejemplo input:

1  
2  
3  
4  
5  
a

Output

SUM = 15 AVG = 3

Input

Prueba

Output

SUM = 0 AVG = 0

Ayuda.

Utiliza un while infinito. Usa break cuando el usuario inserte algo distinto a un número para salir del bucle. Si el usuario no inserta ningún número válido muestra SUM = 0 AVG = 0.

Utiliza Scanner para leer los números. Ayúdate del método hasNextInt() de Scanner para saber si ha insertado un número o no. Utiliza Math.round para redondear el cálculo de la media.

Necesitarás una variable contadora para contar el número de dígitos introducidos y que te ayude luego a hacer la media. Otra variable suma que almacene el resultado de sumar los dígitos insertados.

```
public class insertaNumerosSumaYMedia {
    public static void main(String[] args) {
        insertaImprimeSumaYMedia();
    }
    public static void insertaImprimeSumaYMedia(){
        Scanner sc= new Scanner(System.in);
        System.out.println("introduce numeros para la suma y la media. Para salir introduce algo que no sea un número:");
        boolean salida=false;
        int sum=0;
        int contador=0;
        while(salida==false){

            if (sc.hasNextInt()==false){
                int AVG=sum/contador;
                salida=true;
                System.out.println("SUM= "+sum + "AVG= "+AVG);
                break;
            }else{
                sum=sum+sc.nextInt();
                contador++;
            }
        }
    }
}
```