

IFC-303D - 1º DAW

2020-2021

Desarrollo de Aplicaciones Web

PROGRAMACIÓN

UNIDAD 4

Tarea: Ejercicios Java – Estructuras
datos estáticas

Contenido

Descripción de la tarea.....	5
Ejercicio 1	6
Código final	6
Resultado final	7
Ejercicio 2	8
Código final	8
Resultado final	8
<i>Ejercicio 3</i>	8
Código final	8
Resultado final	9
Ejercicio 4	10
Código final	10
Resultado final	11
Ejercicio 5	12
Código final	12
Resultado final	14
Ejercicio 6	14
Ejemplo de un tutorial	15
Código de pruebas	16
Código final	17
Resultado final	18
Ejercicio 7	18
Código pruebas	18
Resultado pruebas	21
Código final	21
Resultado final	22
Ejercicio 8	22
Código de pruebas	23
Resultado pruebas	24

Código final	25
Resultado final	26
Ejercicio 9	26
Código final	26
Ejercicio 10	27
Código pruebas	27
Resultado pruebas	29
Código final	29
Resultado final	30
Ejercicio 11	31
Código pruebas	31
Resultado pruebas	31
Código final	32
Ejercicio 12	32
Código final	33
Resultado pruebas	34
Ejercicio 13	34
Código pruebas	34
Ejercicio 14	35
Código pruebas	35
Resultado pruebas	36
Código final	37
Resultado final	38
Ejercicio 15	39
Código pruebas Notas Medias	39
Ejercicio 16	40
Código final	40
Ejercicio 17	41
Ejercicio HUNDIR BARCOS	41
Código final	42

Resultado final	42
------------------------------	-----------

Descripción de la tarea

Instrucciones para resolver los ejercicios

- *Se han de definir clases necesarias para cada ejercicio en el paquete unidad4 dentro del proyecto que has creado en la tarea “Crear proyecto nuevo de Eclipse con control de versiones en servidor de Git”.*
- *Cada vez que se resuelva un ejercicio se realizará un Commit and Push con el mensaje “Ejercicio n de la unidad 4 resuelto”, donde n será el número de ejercicio.*
- *No es obligatorio resolver y confirmar los ejercicios en el orden de numeración.*

Ejercicio 1

Programa que lea una cadena desde el teclado y muestre en la consola el número de veces que se repite cada vocal.

Código final

```
package unidad4;

import java.util.Scanner;

public class Ejercicio1 {

    static Scanner input = new Scanner (System.in); // Objeto Scanner
    para captar datos por teclado

    public static void main(String[] args) {

        System.out.println("Escriba un texto");
        String cadena = input.nextLine(); //Guardamos el texto
        introducido por teclado
        int longitud = cadena.length(); // Almacenamos la longitud de
        la cadena ded texto
        cadena = cadena.toUpperCase(); // Pasamos el texto a mayúsculas
        para que todas las letras sean iguales
        String letra = "";
        int a = 0, e = 0, i = 0, o = 0, u = 0; // Contadores
        for (int j = 0; j < longitud; j++ ) { // i es un contador para
        recorrer los caracteres de la cadena
            letra = Character.toString(cadena.charAt(j)); //
            Almacenamos el primer caracter del texto y lo convertimos a String
            switch (letra) {
                case "A":
                    a++;
                    break;
                case "E":
                    e++;
                    break;
                case "I":
                    i++;
                    break;
                case "O":
                    o++;
                    break;
                case "U":
                    u++;
                    break;
                default:
                    break;
            } // fin de switch
        } // fin de for
        System.out.println(" a: " + a + " veces \n e: " + e + "
        veces \n i: " + i + " veces \n o: " + o + " veces \n u: " + u+ " veces
        \n");

    } // fin de main
```

```
} // fin de clase
```

Resultado final

Escriba un texto

Hola, esto es un texto de prueba

a: 2 veces

e: 5 veces

i: 0 veces

o: 3 veces

u: 2 veces

Ejercicio 2

Programa que lea una cadena en desde el teclado y la muestre invertida en la consola.

Código final

```
package unidad4;

import java.util.Scanner;

public class Ejercicio2 {

    //CLASES
    static Scanner input = new Scanner (System.in); // Objeto Scanner
    para captar datos por teclado

    public static void main(String[] args) {
        System.out.println("Escriba un texto");
        String texto = input.nextLine(); // Almacenamos texto por
    teclado
        StringBuffer invertido = new StringBuffer(texto); // Creamos un
    búfer de cadena con el texto
        invertido = invertido.reverse(); // Usamos el método reverse de
    StringBuffer
        System.out.println(texto + "\n" + invertido);
    }
}
```

Resultado final

```
Escriba un texto
Esto es una prueba
Esto es una prueba
abeurp anu se otsE
```

Ejercicio 3

Programa que lea dos cadenas desde el teclado y muestre el número de veces que la segunda está contenida en la primera.

Código final

```
package unidad4;

import java.util.Scanner;

public class Ejercicio3 {

    static Scanner input = new Scanner (System.in);

    public static void main(String[] args) {
```



```

        System.out.println("Escriba un texto");
        String texto1 = input.nextLine(); // Almacenamos
        texto por teclado
        System.out.println("Escriba otro texto");
        String texto2 = input.nextLine(); // Almacenamos
        texto por teclado

        String texto1Mayus = texto1.toUpperCase();
        String texto2Mayus = texto2.toUpperCase();

        if (texto2Mayus.equals(texto1Mayus))
            System.out.println(texto1 + " y " + texto2 + "
son IGUALES");
        else {
            int index =
texto1Mayus.indexOf(texto2Mayus);
            int contador = 0;
            if (index >=0) {
                //System.out.println(index);
                int i = 0;
                String str = texto1Mayus;
                while ((i < texto1Mayus.length()) &&
(index != -1)) {
                    str =
str.substring(index+texto2Mayus.length());
                    //System.out.println(str);
                    index =
str.indexOf(texto2Mayus);
                    //System.out.println(index);
                    i=index;
                    contador++;
                }
                System.out.println("\n" + texto2 +
"\n se encuentra dentro de \n" + texto1 + "\n " + contador + "
veces");
            } else
                System.out.println(texto1 + " y " +
texto2 + " son DIFERENTES");
        }
    }
}

```

Resultado final

Escriba un texto

Esto es un texto de prueba para un ejercicio

Escriba otro texto

un

"un" se encuentra dentro de "Esto es un texto de prueba para un ejercicio" 2 veces

Ejercicio 4

Programa que, utilizando una cantidad mínima de variables, simule el lanzamiento de un dado N veces, siendo N un número entero que se introducirá por teclado. Antes de finalizar mostrará el número de veces que salió cada una de las caras.

Código final

```
package unidad3;
```

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
public class Dado {
```

```
    public static void main(String[] args) {
```

```
        //CLASES
```

```
        Scanner input = new Scanner (System.in); // Objeto Scanner para captar  
datos por teclado
```

```
        //VARIABLES
```

```
        System.out.println("Introduzca número de tiradas de dado");
```

```
        int tiradas = input.nextInt();
```

```
        int[] numeros = new int[tiradas];
```

```
        int[] repetido = new int[tiradas];
```

```
        int repeticion = 0;
```

```
        //int unicos = 0;
```

```
        //TIRADAS
```

```
        for (int i = 0; i < tiradas; i++) { // i es un contador para realizar n tiradas y
guardar en array

            numeros[i] = (int) Math.floor(Math.random()*(1-6+1)+6); // Número
random entre 1 y 6

            System.out.println("Tirada " + (i+1) + ": " + numeros[i]);

        } // fin de for


        //CONTAR REPETIDOS

        Arrays.sort(numeros); // Se ordena el contenido del array

        int[] copia = numeros; // Copiamos el contenido del array para comparar

        int copiaNum = 0; // Variable que sirve para comprobar si ya hemos leído ese
número

        for (int j = 0 ; j < tiradas ; j++) { // j es un contador para leer array de la copia

            repetido[j] = 0; // Ponemos el contador de repetidos a 0

            if (copia[j] != copiaNum) { // Comprobamos si el número es igual que el anterior para
saltárnoslo (ver línea 39)

                for (int k = 0; k < tiradas; k++) { // k es un contador para leer todo el array de
numeros y comparar con cada posición del array copia

                    if (copia[j] == numeros[k]) { // Se comprueba si el número es igual
que la copia

                        repetido[j]++; // Incrementamos número de repeticiones

                    } // fin de if

                } // fin de for

                System.out.println("El número " + numeros[j] + " se ha repetido " +
repetido[j] + " veces");

            } // fin de if

            copiaNum = numeros[j];

        } // fin de for


    } // fin de main

} // fin de la clase
```

Resultado final

Introduzca número de tiradas de dado

5

Tirada 1: 3

Tirada 2: 3

Tirada 3: 2

Tirada 4: 4

Tirada 5: 5

El número 2 se ha repetido 1 veces

El número 3 se ha repetido 2 veces

El número 4 se ha repetido 1 veces

El número 5 se ha repetido 1 veces

Ejercicio 5

Programa que lea desde el teclado una línea que contenga un NIF completo (número y letra) y a continuación verifique que es correcto. Para obtener la letra de validación del N.I.F. se realiza la división entera de la parte numérica entre 23 y el resto se utiliza como índice de la tabla siguiente:.

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

La validación consiste en comprobar que la letra obtenida en la tabla coincide con la letra introducida por teclado.

Código final

```
package unidad4;

import java.util.Scanner;

public class Ejercicio5 {

    static Scanner input = new Scanner (System.in);

    public static void main(String[] args) {
        System.out.println("Introduzca NIF");
        String cif = input.nextLine();
        String test = "";
        int longitud = cif.length();
        int numeros, resto;
        char letra = cif.charAt(longitud-1); // Guardamos el último
caracter de la cadena cif

        if ( (longitud != 9) || Character.isDigit(letra) ) {
            System.out.println("Formato de CIF no válido");
        } else {
            numeros = Integer.parseInt(cif.substring(0, longitud-
1)); // Almacenamos y pasamos a int el CIF eliminando la letra final
        }
    }
}
```

```
resto = numeros % 23;
switch (resto) {
    case 0:
        test = "T";
        break;
    case 1:
        test = "R";
        break;
    case 2:
        test = "W";
        break;
    case 3:
        test = "A";
        break;
    case 4:
        test = "G";
        break;
    case 5:
        test = "M";
        break;
    case 6:
        test = "Y";
        break;
    case 7:
        test = "F";
        break;
    case 8:
        test = "P";
        break;
    case 9:
        test = "D";
        break;
    case 10:
        test = "X";
        break;
    case 11:
        test = "B";
        break;
    case 12:
        test = "N";
        break;
    case 13:
        test = "J";
        break;
    case 14:
        test = "Z";
        break;
    case 15:
        test = "S";
        break;
    case 16:
        test = "Q";
        break;
    case 17:
        test = "V";
        break;
    case 18:
        test = "H";
```

```

        break;
    case 19:
        test = "L";
        break;
    case 20:
        test = "C";
        break;
    case 21:
        test = "K";
        break;
    case 22:
        test = "E";
        break;
    } // fin switch
    /* Debug
    System.out.println("El num es" + numeros + " letra es: "
+ letra);
    System.out.println(test +
Character.toString(letra).toUpperCase());
    */
    if
(test.equalsIgnoreCase(Character.toString(letra).toUpperCase())) {
        System.out.println("CIF válido");
    } else {
        System.out.println("CIF NO válido");
    } // fin if-else
    } // fin if-else
    } // fin main
} // fin clase

```

Resultado final

Introduzca NIF
71885796a

CIF NO válido

Introduzca NIF
71885796d
CIF válido

Ejercicio 6

Programa que realice las tareas siguientes:

- Crear un vector de números enteros de un tamaño aleatorio comprendido entre 10 y 50 elementos.

- Llenar el vector con números aleatorios comprendidos entre -100 y 100 sin repetir ninguno.
- Crear un segundo vector que contenga los elementos del anterior almacenados en orden inverso.
- Mostrar ambos vectores en la consola

Ejemplo de un tutorial

<https://es.stackoverflow.com/questions/233047/d%C3%B3nde-est%C3%A1-el-fallo-array-n%C3%BAmeros-aleatorios-sin-repetir-en-java>

```
package Pruebas;

public class PruebasEjercicios {

    public static void main(String[] args) {
        // Aquí establecemos la cantidad de elementos aleatorios
        que deseamos
        int cantidad = 10;
        // Esta variable se usará para llenar el arreglo en la
        posición correspondiente
        int index = 0;
        // Arreglo que almacenará los números aleatorios
        int [] aleatorios = new int [cantidad];
        // Nuestro primer bucle que se ejecutará hasta que hayamos
        llenado el arreglo
        while(index < cantidad) {
            // Variable que almacenará el número aleatorio
            propuesto

            int propuesto = (int)(Math.random()*cantidad);
            // Variable que indica si el número propuesto está
            repetido

            // asumimos que aún no está repetido y la establecemos
            a false

            boolean repetido = false;
            //Segundo bucle que se ejecutará siempre que el número
            no esté repetido

            while(!repetido) {
                // Bucle que recorre el arreglo comparando el
                número propuesto con

                // cada uno de los elementos del arreglo
                for(int i=0; i<index; i++) {
                    //realizamos la comparación
                    if(propuesto == aleatorios[i]) {
                        // si el número se repite, establecemos
                        repetido=true

                        // y salimos del bucle for (no es necesario
                        seguir comparando)

                        repetido = true;
                        break;
                    }
                }
            }
        }
    }
}
```

```

// verificamos el estado del valor repetido. Si es
false, significa
index sin encontrar
// coincidencias
if(!repetido) {
    // almacenamos el valor propuesto ya que no
    // incrementamos el índice
    aleatorios[index] = propuesto;
    index++;
}
}

// mostramos los valores por pantalla
for (int i=0;i<aleatorios.length;i++){
    System.out.print(aleatorios[i]+ " ");
}
// Inserta un salto de línea
System.out.println("");
}
}

```

Código de pruebas

```

package unidad4;

import java.util.Arrays;

public class Ejercicio6 {

    public static void main(String[] args) {
        int tamaño = 6;
        //int tamaño = (int) Math.floor(Math.random()*(10-50+1)+50);//
        Queremos que tamaño sea un número entero entre 10 y 50
        int[] vector = new int[tamaño]; // El array tendrá una
        longitud aleatoria entre 10 y 50
        int num; // Almacenará temporalmente el número aleatorio
        int index = 0; // Índice de la posición del número en el array
        while (index < tamaño) {
            num = (int) Math.floor(Math.random()*(1-10+1)+10);//
            Aleatorio entre -100 y 100;
            boolean repetido = false; // Inicializamos en false, el
            primer número no puede estar repetido
            while (repetido == false) { // Comparamos el
            número solo si no se ha comprobado ya que está repetido
                for (int j=0; j < index; j++) { // Recorre
                el array para comprobar si el num ya existe
                    if (num == vector[j]) { //Si el
                    número está repetido establecemos true y salimos del bucle for
                        System.out.println("Num
                        aleatorio:" + num);
                        repetido = true;
                        break;
                    }
                }
            }
            index++;
        }
    }
}

```



```

        }// fin de if
    }// fin de for
    if (repetido == false) {
        vector[index] = num; // Almacenamos
el número
        System.out.println("Num aleatorio:" +
num + "\n Array:" + Arrays.toString(vector));
        System.out.println("Tamaño del array:
" + tamaño + ". Vector " + index + " : " + vector[index] + ". Repetido: " +
repetido + "\n\n");
        index++;
    }// fin de if
    }// fin de while
}// fin de while

}// fin de main

}// fin de class

```

Código final

```

package unidad4;

import java.util.Arrays;

public class Ejercicio6 {

    public static void main(String[] args) {

        int tamaño = (int) Math.floor(Math.random()*(10-50+1)+50);//
Queremos que tamaño sea un número entero entre 10 y 50
        int[] vector = new int[tamaño]; // El array tendrá una
longitud aleatoria entre 10 y 50
        int num; // Almacenará temporalmente el número aleatorio
        int index = 0; // Índice de la posición del número en el array
        while (index < tamaño) {
            num = (int) Math.floor(Math.random()*(-100-
100+1)+100);// Aleatorio entre -100 y 100;
            boolean repetido = false; // Inicializamos en false, el
primer número no puede estar repetido
            while (repetido == false) { // Comparamos el
número solo si no se ha comprobado ya que está repetido
                for (int j=0; j < index; j++) { // Recorre
el array para comprobar si el num ya existe
                    if (num == vector[j]) { //Si el
número está repetido establecemos true y salimos del bucle for
                        repetido = true;
                        break;
                    }// fin de if
                }// fin de for
            if (repetido == false) {
                vector[index] = num; // Almacenamos
el número
                index++;
            }// fin de if
        }// fin de while
    }
}

```

```
        }// fin de while
        System.out.println("Tamaño del array: " + tamaño + ".\nArray:"
+ Arrays.toString(vector) + "\n\n");
    }// fin de main

}// fin de class
```

Resultado final

Tamaño del array: 28.

Array:[24, -36, 0, 16, -97, 65, -15, -42, 23, 5, 44, -52, 3, -59, -14, 36, -10, -1, -56, 11, -63, -98, 32, -6, -70, -51, 62, -96]

Ejercicio 7

Programa que realice las tareas siguientes:

- *Crear un vector de números enteros de un tamaño especificado por el usuario entre 10 y 1.000.000 de elementos.*
- *Llenar el vector con números aleatorios comprendidos entre -999.999 y 1.000.000 sin repetir ninguno.*
- *Mostrar en la consola los datos siguientes:*

o Tamaño del vector.

o Tiempo empleado en llenarlo

o Diferencia entre el menor y el mayor de los números almacenados en el vector.

o Tiempo empleado en calcular la diferencia entre el menor y el mayor de los números almacenados.

Código pruebas

```
package unidad4;
```

```
import java.util.Scanner;
```

```
import java.util.Arrays;
```

```
import java.util.Random;
```

```
public class Ejercicio7 {

    static Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        Random r = new Random();
        Scanner in = new Scanner(System.in);
        int [] vector;

        System.out.println("Introduce la dimensiÃ³n del vector
(10<=dim<=1.000.000):");

        int dim = in.nextInt();
        vector = new int[dim];
        int mayor=-100000;
        int menor=1000001;
        long t0 = System.currentTimeMillis();
        for (int i=0; i<vector.length; i++) {
            //System.out.println("Vuelta " + i);
            int j;
            int n;
            do {
                //n = r.nextInt(15);
                System.out.println("introduzca num");
                n = input.nextInt();
                j = 0;
                System.out.println(j);
                while (j < i && vector[j] != n) { // Si j es menor que i (índice
del array) y el num de la posición j es diferente al número
                    j++;} // incrementa el contador j para comprobar
el siguiente número

                //System.out.println("j " + j);
            } while (true);
        }
    }
}
```

```
//System.out.println(" i: " + i);

} while (j != i);
vector[i] = n;
//System.out.println("numero " + n);
//System.out.println(" mayor: " + mayor + " menor: " + menor);
if (n > mayor){
    mayor = n;
    //System.out.println(" mayor: " + mayor);
}
if ((n < menor)) {
    menor = n;
    //System.out.println(" menor: " + menor);
}
//System.out.println("vector num: " + vector[i]);
//System.out.println(" mayor: " + mayor + " menor: " + menor);
}

long t1 = System.currentTimeMillis() - t0;
System.out.println("Tamaño del vector: " + vector.length);
System.out.println("Tiempo empleado en llenarlo: " + t1 + "
milisegundos");

t0 = System.currentTimeMillis();
int dif = mayor-menor;

t1 = System.currentTimeMillis() - t0;
System.out.println("Diferencia entre el valor mayor y el menor: " + dif);
System.out.println("Tiempo empleado en calcular la diferencia: " + t1 + "
milisegundos");

System.out.println("El vector contiene: " + Arrays.toString(vector));
```

```
}
```

```
}
```

Resultado pruebas

Introduce la dimensiÃ³n del vector (10<=dim<=1.000.000):

3

introduzca num

5

0

introduzca num

9

0

introduzca num

1

0

Tamaño del vector: 3

Tiempo empleado en llenarlo: 4454 milisegundos

Diferencia entre el valor mayor y el menor: 8

Tiempo empleado en calcular la diferencia: 0 milisegundos

El vector contiene: [5, 9, 1]

Código final

SOLUCIÓN DEL PROFE

```
package unidad4;
```

```
import java.util.Random;
```

```
import java.util.Scanner;
```

```
public class Ejercicio7 {
```

```
    public static void main(String[] args) {
```

```
        Random r = new Random();
```

```
        Scanner in = new Scanner(System.in);
```

```
        int [] vector;
```

```
        System.out.println("Introduce la dimensiÃ³n del vector  
(10<=dim<=1.000.000):");
```

```
        int dim = in.nextInt();
```

```
        vector = new int[dim];
```

```
        long t0 = System.currentTimeMillis();
```

```
        for (int i=0; i<vector.length; i++) {
```

```
            int j;
```

```
            int n;
```

```

        do {
            n = r.nextInt(2000000) - 999999;
            j = 0;
            while (j < i && vector[j] != n)
                j++;
            } while (j != i);
            vector[i] = n;
        }
        long t1 = System.currentTimeMillis() - t0;
        System.out.println("Tamaño del vector: " + vector.length);
        System.out.println("Tiempo empleado en llenarlo: " + t1 + "
milisegundos");
        t0 = System.currentTimeMillis();
        int dif = 0;

        t1 = System.currentTimeMillis() - t0;
        System.out.println("Diferencia entre el valor mayor y el menor:
" + dif);
        System.out.println("Tiempo empleado en calcular la diferencia:
" + t1 + "milisegundos");
    }
}

```

Resultado final

Introduce la dimensi3n del vetor (10<=dim<=1.000.000):

10

Tama3o del vector: 10

Tiempo empleado en llenarlo: 0 milisegundos

Diferencia entre el valor mayor y el menor: 0

Tiempo empleado en calcular la diferencia: 0milisegundos

Ejercicio 8

Programa que realice las tareas siguientes:

- Crear un vector de números enteros de un tamaño especificado por el usuario que estará comprendido entre 10 y 200 elementos.
- Llenar el vector con números aleatorios comprendidos entre -100 y 100.
- Mostrar la suma de los números almacenados teniendo en cuenta que hay quien piensa que el número 13 es el número de la mala suerte. Por tanto, si en alguna posición se encuentra almacenado el número 13, no se sumará ni este número ni los que se encuentren almacenados en las 13 posiciones siguientes (o las que haya hasta el final del vector si estas son menos de 13) si la suma de todos ellos es distinta de 7.
- Mostrar el contenido del vector y la cantidad de números que no se han sumado.o Tiempo empleado en calcular la diferencia entre el menor y el mayor de los números almacenados.

Código de pruebas

PRUEBAS REALIZADAS SOBRE LA SOLUCIÓN PROPORCIONADA POR EL PROFE

```
package Pruebas;
```

```
import java.util.Random;
```

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
public class PruebasEjercicios {
```

```
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Introduce un tamaño para el vector entre 1
y 200: ");
        int n = in.nextInt();
        while (n < 1 || n > 200) { // Modificado para que se pueda
probar con menos de 10
            System.out.println("Incorrecto, el tamaño tiene que
estar entre 10 y 200. Vuelve a introducirlo: ");
            n = in.nextInt();
        }
        //in.close();

        int [] vector = new int[n];
        for (int i=0; i<vector.length; i++) { // Contador i para
recorrer el array
            //vector[i] = (int) (Math.round(Math.random() * 200) -
100); // Num aleatorio entre -100 y 100
            System.out.println("Introduce num"); // Para debug
introducimos num por teclado
            vector[i] = in.nextInt(); // Almacenamos en posición i
del array
        }
        // Inicializamos variables a 0
        int suma = 0;
        int parcial = 0;
        int nosumar = 0;
        int nosumados = 0;
        for (int i=0; i<vector.length; i++) { // Contador i para
recorrer el array
            if (vector[i] == 13 && nosumar == 0) { // Comprobamos
cada num si es igual a 13 siempre que no haya habido un 13 anterior
                nosumar = 14; // Se establece un contador a 14,
que serán las posiciones que no se sumarán salvo que el resultado de parcial
sea 7
                parcial = 0; // Al detectar 13 establecemos
parcial a 0 para empezar a sumar
                System.out.print("{ ");
            }
            System.out.print(vector[i] + " ");
        }
    }
}
```

```

        if (nosumar == 0) // Si es 0 es que el número no es 13
ni estamos dentro de las siguientes 13 posiciones a no sumar
            suma += vector[i]; // Se añade el número al
sumatorio
        else {
            parcial += vector[i]; // Si no es 0 es que tenemos
que ir sumando parciales hasta comprobar si suman 7 o no
            System.out.print("--nosumar " + nosumar + "
parcial " + parcial + " "); // Texto debug
            if (--nosumar == 0) { // (--nosumar es igual que
nosumar = nosumar-1) Decrementa en 1 y lo compara con 0 para controlar si
hemos llegado al último de los 13 números
                if (parcial == 7) // Si la suma parcial es
igual a 7
                    suma += parcial; // Sumamos el
parcial a la suma
                else // si no
                    nosumados += 14; // Guardamos el
número de ignorados (el 13 más las siguientes 13 posiciones)
                System.out.print("} ");
            }
        }
    }
    System.out.println();
    if (nosumar > 0) { // Al finalizar el bucle se comprueba si
nosumar no ha llegado a 0
        if (parcial == 7) // Si la suma parcial es igual a 7
            suma += parcial; // Sumamos el parcial a la suma
        else // si no
            nosumados = 14 - nosumar; // Guardamos el número
de ignorados (el 13 + las siguientes 13 posiciones - las posiciones que
restaban hasta llegar a 0)
    }

    System.out.println("Suma: " + suma);
    System.out.println("No sumados: " + nosumados);
}
}

```

Resultado pruebas

Introduce un tamaño para el vector entre 1 y 200:

4

Introduce num

1

Introduce num

13

Introduce num

2

Introduce num

3

1 { 13 --nosumar 14 parcial 13 2 --nosumar 13 parcial 15 3 --nosumar 12
parcial 18

Suma: 1

No sumados: 3

Código final

```

package unidad4;

import java.util.Scanner;

public class Ejercicio8 {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Introduce un tamaño para el vector entre
10 y 200: ");
        int n = in.nextInt();
        while (n < 10 || n > 200) {
            System.out.println("Incorrecto, el tamaño tiene que
estar entre 10 y 200. Vuelve a introducirlo: ");
            n = in.nextInt();
        }
        in.close();

        int [] vector = new int[n];
        for (int i=0; i<vector.length; i++)
            vector[i] = (int) (Math.round(Math.random() * 200) -
100);

        int suma = 0;
        int parcial = 0;
        int nosumar = 0;
        int nosumados = 0;
        for (int i=0; i<vector.length; i++) {
            if (vector[i] == 13 && nosumar == 0) {
                nosumar = 14;
                parcial = 0;
                System.out.print("{ ");
            }
            System.out.print(vector[i] + " ");
            if (nosumar == 0)
                suma += vector[i];
            else {
                parcial += vector[i];
                if (--nosumar == 0) {
                    if (parcial == 7)
                        suma += parcial;
                    else
                        nosumados += 14;
                    System.out.print("} ");
                }
            }
        }
        System.out.println();
        if (nosumar > 0) {
            if (parcial == 7)
                suma += parcial;
            else
                nosumados = 14 - nosumar;
        }
    }
}

```

```

        System.out.println("Suma: " + suma);
        System.out.println("No sumados: " + nosumados);
    }
}

```

Resultado final

Introduce un tamaño para el vector entre 10 y 200:

10

-58 72 -8 -24 -93 53 -20 57 21 -20

Suma: -20

No sumados: 0

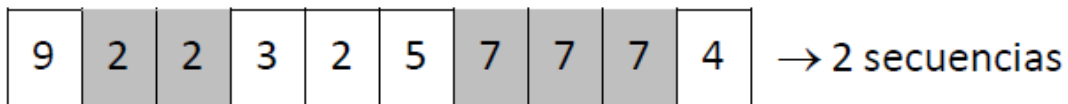
Ejercicio 9

Programa que realice las tareas siguientes:

- Crear un vector de números enteros de un tamaño aleatorio entre 10 y 500 elementos.
- Llenar el vector con números aleatorios comprendidos entre -100 y 100.
- Mostrar el contenido del vector si su tamaño es menor o igual a 50.

Mostrar, independientemente del tamaño del vector, el número de secuencias de números repetidos.

Ejemplo:



Código final

SOLUCIÓN DEL PROFE

```

package unidad4;

public class Ejercicio9 {

    public static void main(String[] args) {
        // int[] vector = new int[(int) (Math.round(Math.random() * 490) +
        // 10)];
        // for (int i = 0; i < vector.length; i++) {
        //     vector[i] = (int) (Math.round(Math.random() * 200) -
        // 100);
        //     if (vector.length <= 50)
        //         System.out.print(vector[i] + " ");
        //     }
        //     if (vector.length <= 50)
    }
}

```

```
//
    System.out.println();
    int [] v = {2,2,3,3,3};
    System.out.println("Se han detectado " +
    secuenciasDeNumerosRepetidos(v) + " secuencias de números repetidos");
}

    static int secuenciasDeNumerosRepetidos(int[] vector) {
        int secuencias = 0;
        boolean serepite = false;
        for (int i = 1; i < vector.length; i++) {
            if (vector[i] == vector[i - 1]) {
                if (!serepite) {
                    serepite = true;
                    secuencias++;
                }
            } else if (serepite)
                serepite = false;
        }
        return secuencias;
    }
}
```

Ejercicio 10

Programa que realice las tareas siguientes:

- Crear un vector de números enteros de un tamaño especificado por el usuario que esté comprendido entre 10 y 20 elementos.
- Llenar el vector con números aleatorios.
- Mostrar el contenido del vector.
- Mostrar la mínima diferencia entre dos valores adyacentes. La diferencia entre dos valores adyacentes se calcula como el valor almacenado en cada posición [i] (excepto la primera) menos el valor almacenado en la posición [i-1].

Código pruebas

```
package Pruebas;

import java.util.Arrays;
import java.util.Scanner;
import java.util.Random;

public class PruebasEjercicios {
```

```

    public static void main(String[] args) {
        Random r = new Random();
        Scanner in = new Scanner(System.in);
        System.out.println("Introduzca el tamaño del vector
(entre 10 y 20):");
        int dim = in.nextInt();
        while ((dim < 10) || (dim > 20)) {
            System.out.println("Valor incorrecto. Debe estar
comprendido entre 10 y 20:");
            dim = in.nextInt();
        }

        int[] vector = new int[dim]; // El tamaño del array
vector es el establecido por el usuarios

        for (int i=0; i<vector.length; i++) { // Recorre el
array

            int num = r.nextInt(); // Num random
//System.out.println("Nº: " + num); //DEBUG
            vector[i] = num;
        }

        System.out.println("Contenido del vector: " +
Arrays.toString(vector));

        int minDif = diferenciaAdyacentes(vector);

        System.out.println("Mínima diferencia entre dos valores
adyacentes: " + minDif);
    }

    static int diferenciaAdyacentes(int[] vector) {
        //PRUEBA CON BUCLE DO WHILE
        int i = 1;
        int minDif = vector[i] - vector[i-1];
        //System.out.println("Resta 1: " + minDif); // DEBUG
        do {
            i++;
            int test = vector[i] - vector[i-1];
            if (test < minDif) {
                minDif = test;
            }
            // System.out.println("Test " + i + ": " + test);
// DEBUG
            // System.out.println("Resta " + i + ": " +
minDif); // DEBUG
        } while (i < vector.length-1);

        //PRUEBA CON BUCLE FOR
        /*int minDif = vector[1] - vector[0];
        int dif;
        for (int i=2; i<vector.length; i++) {
            dif = vector[i] - vector[i-1];
            if (dif < minDif)
                minDif = dif;
            System.out.println("Mínima diferencia parcial: " +
minDif); // DEBUG
        }
    }

```

```

        */
        return minDif;
    }
}

```

Resultado pruebas

Introduzca el tamaño del vector (entre 10 y 20):

10

Nº: 15

Nº: 4

Nº: 19

Nº: 7

Nº: 19

Nº: 10

Nº: 1

Nº: 18

Nº: 13

Nº: 2

Contenido del vector: [15, 4, 19, 7, 19, 10, 1, 18, 13, 2]

Resta 1: -11

Test 2: 15

Resta 2: -11

Test 3: -12

Resta 3: -12

Test 4: 12

Resta 4: -12

Test 5: -9

Resta 5: -12

Test 6: -9

Resta 6: -12

Test 7: 17

Resta 7: -12

Test 8: -5

Resta 8: -12

Test 9: -11

Resta 9: -12

Mínima diferencia entre dos valores adyacentes: -12

Código final

```
package unidad4;
```

```
import java.util.Arrays;
```

```
import java.util.Random;
```

```
import java.util.Scanner;
```

```
public class Ejercicio10 {
```

```
    public static void main(String[] args) {
```

```
        Random r = new Random();
```

```
        Scanner in = new Scanner(System.in);
```

```
        System.out.println("Introduzca el tamaño del vector  
(entre 10 y 20):");
```

```

        int dim = in.nextInt();
        while ((dim < 10) || (dim > 20)) {
            System.out.println("Valor incorrecto. Debe estar
comprendido entre 10 y 20:");
            dim = in.nextInt();
        }

        int[] vector = new int[dim]; // El tamaño del array
vector es el establecido por el usuarios

        for (int i=0; i<vector.length; i++) { // Recorre el
array
            int num =r.nextInt(); // Num random
            vector[i] = num;
        }

        System.out.println("Contenido del vector: " +
Arrays.toString(vector));

        int minDif = diferenciaAdyacentes(vector);

        System.out.println("Mínima diferencia entre dos valores
adyacentes: " + minDif);
    }

    static int diferenciaAdyacentes(int[] vector) {
        //PRUEBA CON BUCLE DO WHILE
        int i = 1;
        int minDif = vector[i] - vector[i-1];
        do {
            i++;
            int test = vector[i] - vector[i-1];
            if (test < minDif) {
                minDif = test;
            }
        } while (i < vector.length-1);

        //PRUEBA CON BUCLE FOR
        /*int minDif = vector[1] - vector[0];
        int dif;
        for (int i=2; i<vector.length; i++) {
            dif = vector[i] - vector[i-1];
            if (dif < minDif)
                minDif = dif;
        }*/

        return minDif;
    }
}

```

Resultado final

Introduzca el tamaño del vector (entre 10 y 20):

5

Valor incorrecto. Debe estar comprendido entre 10 y 20:

10

Contenido del vector: [644862434, 597727732, 1707535477, 244871653, -1672395033, -1329021291, -1819922874, -589155131, -941644481, 1130153256]

Mínima diferencia entre dos valores adyacentes: -1917266686

Ejercicio 11

Definir un método que reciba un vector de cadenas y retorne la cadena de mayor longitud almacenada en dicho vector.

Poner a prueba el método invocándolo desde otro método que cree un vector con los nombres de 10 personas introducidos por teclado.

Código pruebas

```
package unidad4;

import java.util.Scanner;

public class Ejercicio11 {

    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);

        System.out.println("Escriba el nombre de 10 personas.");
        String[] vector = new String[4];
        for (int i=0; i<vector.length; i++) {
            System.out.println("Escriba el nombre nº " + (i+1));
            vector[i] = input.nextLine();
        }
        String cadena = cadenaMayor(vector);

        System.out.println("El nombre más largo es: " + cadena);
    }

    static String cadenaMayor(String[] vector) {
        String cadena = vector[0];
        System.out.println(cadena.length() + cadena);
        for (int i=1; i<vector.length; i++) {
            String nombre = vector[i];

            if (nombre.length() > cadena.length()) {
                System.out.println(nombre.length() + nombre);
                cadena = vector[i];
            }
        }
        return cadena;
    }

}
```

Resultado pruebas

Escriba el nombre de 10 personas.

Escriba el nombre nº 1

Tuko

Escriba el nombre nº 2

Verónica

Escriba el nombre nº 3

Román

Escriba el nombre nº 4

Mamá

4Tuko

8Verónica

El nombre más largo es: Verónica

Código final

```
package unidad4;

import java.util.Scanner;

public class Ejercicio11 {

    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);

        System.out.println("Escriba el nombre de 10 personas.");
        String[] vector = new String[10];
        for (int i=0; i<vector.length; i++) {
            System.out.println("Escriba el nombre nº " + (i+1));
            vector[i] = input.nextLine();
        }
        String cadena = cadenaMayor(vector);

        System.out.println("El nombre más largo es: " + cadena);
    }

    static String cadenaMayor(String[] vector) {
        String cadena = vector[0];
        for (int i=1; i<vector.length; i++) {
            String nombre = vector[i];
            if (nombre.length() > cadena.length()) {
                cadena = vector[i];
            }
        }
        return cadena;
    }

}
```

Ejercicio 12

Consideremos un vector de números enteros con índices entre 0 y n. Se define el centro del vector como el índice c que verifica la siguiente propiedad:

$$\sum_{i=0}^{c-1} (c - i) \cdot V[i] = \sum_{j=c+1}^n (j - c) \cdot V[j]$$

Esta propiedad no siempre se verifica; en ese caso, decimos que el vector no tiene centro.

Ejemplo, consideremos el siguiente vector:

0	1	2	3	4
6	2	3	0	1

El centro de este vector es el índice 1. En efecto, si aplicamos la definición con $c = 1$ y con $n = 4$, obtenemos lo siguiente:

$$\sum_{i=0}^{c-1} (c - i) \cdot V[i] = 1 \cdot 6 = \sum_{j=c+1}^n (j - c) \cdot V[j] = 1 \cdot 3 + 2 \cdot 0 + 3 \cdot 1$$

Por el contrario, el siguiente vector no tiene centro:

0	1	2	3
1	2	1	1

Crear una clase que defina dos métodos:

- Un método llamado `centro` que reciba como parámetro un vector de números enteros y retorne el índice donde se encuentra su centro o `null` si no tiene centro.
- Un método `main` que cree un vector de números enteros leídos por teclado y ponga a prueba el método anterior.

Código final

ES UNA SOLUCIÓN DEL PROFE PERO NO PARECE FUNCIONAR BIEN CON EL EJEMPLO QUE NO TIENE CENTRO (`int [] v = {1, 2, 1, 1}`)

```
package unidad4;
```

```
public class Ejercicio12 {
```

```

    public static void main(String[] args) {
        int [] v = {6, 2, 3, 0, 1};
        Integer c = centro(v);
        System.out.println(c != null ? ("El centro está en la posición " + c) : "No tiene centro");
    }

    static Integer centro(int [] vector) {
        int c = 0;
        int izda;
        int dcha;
        do {
            c++;
            izda = dcha = 0;
            for (int i=0; i<c; i++)
                izda += (c - i) * vector[i];
            for (int j=c+1; j<vector.length; j++)
                dcha += (j - c) * vector[j];
        } while (izda != dcha && c <= vector.length - 1);

        if (c == vector.length - 1)
            return null;
        else
            return c;
    }
}

```

Resultado pruebas

El centro está en la posición 1

Ejercicio 13

Programa que cree una matriz de números enteros aleatorios con un número de filas y de columnas aleatorios que estén comprendidos entre 2 y 20.

Escribir un método que reciba una matriz de números enteros y retorne un vector que contenga la suma de los valores de cada fila.

Escribir otro método que reciba una matriz de números enteros y retorne un vector que contenga la suma de los valores de cada columna.

Mostrar en la consola la matriz, la suma de las filas a su derecha y la suma de las columnas en la parte inferior.

Código pruebas

Texto

Ejercicio 14

Programa que utilice un único array para leer y almacenar los datos de varias secuencias de números enteros. Los datos se introducirán por teclado de la forma siguiente:

- *Se escribe una primera línea que contiene el número de secuencias.*
- *A continuación, se escriben las secuencias a razón de dos líneas por secuencia con el formato siguiente:*
 - o La primera contiene la cantidad de números de la secuencia.*
 - o La segunda contiene los números de la secuencia separados por espacios.*

Código pruebas

```
package unidad4;

import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public class Ejercicio14 {

    public static void main(String[] args) {

        Random r = new Random();

        Scanner in = new Scanner(System.in);

        System.out.println("Introduzca el nº de secuencias:");

        int sec = in.nextInt();

        /*while (isNaN(sec)) {

            System.out.println("Valor incorrecto. Debe introducir un número");

            sec = in.nextInt();

        }*/
    }
}
```

```
int[][] vector = new int[sec][]; // El tamaño del array vector es el establecido
por el usuario
```

```
    for (int i=0; i < sec; i++) {
        System.out.println("Introduzca cantidad de números de la
secuencia:");

        vector[i] = new int[in.nextInt()];
        for (int j=0; j<vector[i].length; j++) {
            System.out.println("Introduzca los números de la secuencia
separados por espacios:");
            vector[i][j] = in.nextInt();
        }
    }
```

```
    for (int [] v: vector)
        System.out.println(Arrays.toString(v));
}
```

```
/*private static boolean isNaN(int dim) {
    // TODO Auto-generated method stub
    return false;
}*/
}
```

Resultado pruebas

Introduzca el nº de secuencias:

3

Introduzca cantidad de números de la secuencia:

2

Introduzca los números de la secuencia separados por espacios:

4

Introduzca los números de la secuencia separados por espacios:

5

Introduzca cantidad de números de la secuencia:

3

Introduzca los números de la secuencia separados por espacios:

9

Introduzca los números de la secuencia separados por espacios:

8

Introduzca los números de la secuencia separados por espacios:

7

Introduzca cantidad de números de la secuencia:

1

Introduzca los números de la secuencia separados por espacios:

5

[4, 5]

[9, 8, 7]

[5]

Código final

SOLUCIÓN DEL PROFE

```
package unidad4;
```

```
import java.util.Arrays;
```

```
import java.util.Scanner;
```

```
public class Ejercicio14 {
```

```
    public static void main(String[] args) {
```

```
        int [][] numeros;
```

```
        Scanner in = new Scanner(System.in);
```

```
        System.out.println("Número de secuencias: ");
```

```
        numeros = new int[in.nextInt()][];
```

```
        for (int i=0; i<numeros.length; i++) {
```

```
            System.out.println("Cantidad de números de la secuencia " + (i + 1) + ": ");
```

```
            numeros[i] = new int[in.nextInt()];
```

```
            System.out.println("Introduce los números de la secuencia:");
```

```
            for (int j=0; j<numeros[i].length; j++)
```

```
                numeros[i][j] = in.nextInt();
```

```
        }
```

```
        for (int [] v: numeros) {  
            for (int n: v)  
                System.out.printf("%-5d", n);  
            System.out.println();  
        }  
  
        for (int [] v: numeros)  
            System.out.println(Arrays.toString(v));  
  
        in.close();  
    }  
  
}
```

Resultado final

Número de secuencias:

3

Cantidad de números de la secuencia 1:

2

Introduce los números de la secuencia:

9

8

Cantidad de números de la secuencia 2:

3

Introduce los números de la secuencia:

7

8

6

Cantidad de números de la secuencia 3:

1

Introduce los números de la secuencia:

5

9 8

7 8 6

5

[9, 8]

[7, 8, 6]

[5]

Ejercicio 15

Programa para la gestión de las calificaciones de los alumnos en las tres evaluaciones de una asignatura. El programa deberá pedir al profesor que introduzca por teclado los nombres de cada alumno junto a las calificaciones de cada evaluación. Después le permitirá realizar las acciones siguientes a través del menú correspondiente:

- 1. Mostrar la nota media de todos los alumnos.*
- 2. Mostrar la nota media de un alumno determinado.*
- 3. Visualizar las notas por evaluación y la nota final de cada evaluación.*
- 4. Visualizar las notas por evaluación y la nota final de un alumno determinado.*
- 5. Calcular la nota media del curso.*
- 6. Calcular la nota más alta y la más baja e indicar a qué alumno y evaluación pertenece.*
- 7. Salir.*

Almacenar los datos en las estructuras de datos estáticas que se consideren necesarias, además de proporcionar una solución basada en la técnica de programación modular.

Código pruebas Notas Medias

```
package unidad4;

import java.util.Arrays;
import java.util.Scanner;

public class NotasMedias {

    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);

        int[][] arrayMedias = new int[3][1];
        int numNotas = 0, notaMedia = 0;

        for (int j=0;j<3;j++) {
            double sumaNotas = 0;
            System.out.println("Introduzca nº de
calificaciones de la " + (j+1) + "ª evaluación: ");
            numNotas = input.nextInt();
            //System.out.println(numNotas);
            double[][][] arrayNotas = new
double[3][3][numNotas];
```

```

        for (int k=0; k<numNotas; k++) {
            System.out.println("Introduzca la nota "
+ (k+1) + " de la " + (j+1) + "ª evaluación: ");
            double nota = input.nextDouble();
            //System.out.println(nota);
            arrayNotas[j][j][k] = nota;
            sumaNotas += nota;
        }
        System.out.println(sumaNotas/numNotas);
        notaMedia = (int)
Math.round(sumaNotas/numNotas);
        System.out.println(notaMedia);
        arrayMedias[j][0] = notaMedia;

        System.out.println(Arrays.toString(arrayMedias[j]));

    }
    for (int l=0; l<arrayMedias.length; l++) {
        System.out.println("Media de la " + (l+1) + "
eval: " + Arrays.toString(arrayMedias[l])); // imprime
directamente el contenido del array pasado a String
    }
    input.close();
}
}

```

Ejercicio 16

Crear una clase llamada `MétodosOrdenación` que defina tres métodos estáticos para ordenar vectores de números enteros implementando la ordenación por inserción directa, por selección directa y por intercambio directo respectivamente.

Código final

SOLUCIÓN DEL PROFE

```

package unidad4;

import java.util.Arrays;

public class MetodosOrdenacion {

    public static void insercionDirecta(int [] v) {
        for (int i=1; i<v.length; i++) {

```



```

        int j = 0;
        while (j < i && v[i] > v[j])
            j++;
        if (j < i) {
            int aux = v[i];
            for (int k=i-1; k>=j; k--)
                v[k+1] = v[k];
            v[j] = aux;
        }
    }
}

public static void seleccionDirecta(int [] v) {

}

public static void intercambioDirecto(int [] v) {

}

public static void main(String[] args) {
    int [] v = {2, 5, 9, 1, 14, 21, 7, 33, 8};
    insercionDirecta(v);
    System.out.println(Arrays.toString(v));
}
}

```

Ejercicio 17

Programa que cree dos vectores de números enteros, cada uno de ellos de una longitud aleatoria entre 10 y 100, los rellene con valores aleatorios, los ordene, los mezcle en un tercer vector manteniendo la ordenación y finalmente muestre el contenido de los tres vectores.

Ejercicio HUNDIR BARCOS

Hacer un programa que permita descubrir en qué posición se encuentra situado un barco que se ha colocado al azar en un array de 10 x10.

Para ello, crear una tabla de 10 x 10 y hacer que el ordenador coloque un barco en una posición al azar.

El usuario ha de intentar descubrir dónde está el barco, de forma que:

- Si falla aparezca el aviso de “AGUA”
- Si acierta aparezca el mensaje “HUNDIDO”.

Contabilizar el número de intentos que se han necesitado.

Código final

```

package unidad4;

import java.util.Random;
import java.util.Scanner;

public class HundirBarcos {

    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);

        String[][] miArray = new String[10][10];
        Random r1 = new Random();
        int posX = r1.nextInt(9)+1; // se pasa como
parámetro número máximo-mínimo y se suma el mínimo
        System.out.print(posX);
        System.out.println();
        Random r2 = new Random();
        int posY = r2.nextInt(9)+1; // se pasa como
parámetro número máximo+mínimo y se suma el mínimo
        System.out.print(posY);
        System.out.println();
        miArray[posX][posY] = "BARCO";
        int intentos = 0;
        int testX = 11;
        int testY = 11;
        do {
            System.out.print("Indique una posición X (0-
9)");

            testX = input.nextInt();
            System.out.print("Indique una posición Y (0-
9)");

            testY = input.nextInt();
            intentos++;
            if (miArray[testX][testY] != "BARCO")
                System.out.print("AGUA! \n");
        } while (miArray[testX][testY] != "BARCO");
        System.out.print("HUNDIDO! \n");
        System.out.print("Intentos: " + intentos);
    }
}

```

Resultado final

5

8

Indique una posición X (0-9)6

Indique una posición Y (0-9)5

AGUA!

Indique una posición X (0-9)5

Indique una posición Y (0-9)8

HUNDIDO!

Intentos: 2