

Ejercicios Java 7

70.

• Array.java

```
package codigo;
import java.util.Scanner; //Inporta Scanner de la libreria util de java

public class Array {

    public void crearArray() { //Crea un array y lo muestra

        int[] array = new int[5]; //Inicializa el array

        Scanner obj = new Scanner(System.in); //Inicializa un objeto
Scanner

        for (int x = 0; x < array.length; x++) {

            System.out.println("Introduce un número en el array: ");
            int numero = obj.nextInt(); //Recoge un número introducido
por el usuario

            array[x] = numero; //Guarda el número en el array
            System.out.println();

        }

        obj.close();

        for (int arr : array) { //Imprime el array

            System.out.println(arr);

        }

    }

}
```

• Principal.java

```
package codigo;
public class Principal {

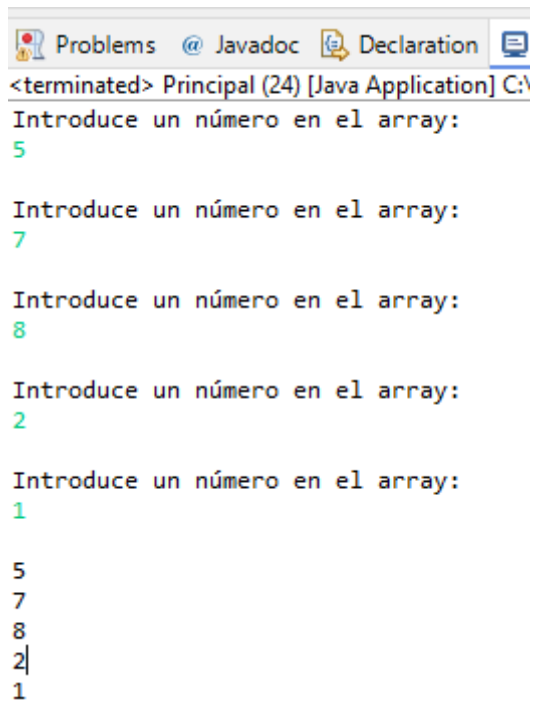
    public static void main(String[] args) {

        Array array = new Array(); //Crea un objeto de la clase array

        array.crearArray(); //Crea y muestra un array

    }

}
```



```
<terminated> Principal (24) [Java Application] C:\
Introduce un número en el array:
5

Introduce un número en el array:
7

Introduce un número en el array:
8

Introduce un número en el array:
2

Introduce un número en el array:
1

5
7
8
2
1
```

71.

• Array.java

```
package codigo;

public class Array {

    String[] array = {"Luis", "María", "Juan"}; //Inicializa el array

    public void mostrarArray() { //Muestra los nombres del array en
        sentido inverso
        for (int x = 2; x >= 0; x--) {
            System.out.println(array[x]);
        }
    }
}
```

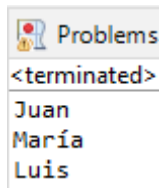
• Principal.java

```
package codigo;
public class Principal {

    public static void main(String[] args) {

        Array array = new Array(); //Crea un objeto de la clase array
        array.mostrarArray(); //Muestra el array en sentido inverso
    }
}
```

```
}
```



72.

• Matriz.java

```
package codigo;
```

```
public class Matriz {
```

```
    int[][] matriz = {
        {2, 7, 4},
        {5, 9, 1},
        {6, 8, 3}
    };
```

```
    public void mostrarMatriz() { //Imprime la matriz
```

```
        for (int x = 0; x < matriz.length; x++) {
            for (int y = 0; y < matriz[x].length; y++) {
                System.out.print(matriz[x][y] + " ");
            }
            System.out.println();
        }
    }
```

```
}
```

• Principal.java

```
package codigo;
```

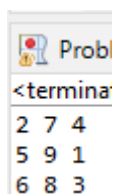
```
public class Principal {
```

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

```
        Matriz matriz = new Matriz(); //Crea un objeto de la clase  
Matriz
```

```
        matriz.mostrarMatriz(); //Muestra la matriz
```

```
    }  
}
```



73.

• **Tareas.java**

```
package codigo;

import java.util.ArrayList;
import java.util.Scanner;

public class Tareas {

    Scanner obj = new Scanner(System.in); //Inicializa un objeto Scanner

    ArrayList<String> tareas = new ArrayList<>();

    public void agregarTarea() { //Añade una tarea a la lista

        System.out.println("Añade una tarea: ");

        String tarea = obj.nextLine(); //Recoge la tarea indicada por el
        usuario

        tareas.add(tarea);

        System.out.println("Tarea " + tarea + " añadida");
        System.out.println();

    }

    public void eliminarTarea() { //Elimina una tarea

        System.out.println("Elimina una tarea: ");

        String tarea = obj.nextLine(); //Recoge la tarea indicada por el
        usuario

        tareas.remove(tarea);

        System.out.println("Tarea " + tarea + " eliminada");
    }
}
```

```

        System.out.println();

    }

    public void mostrarTareas() { //Muestra las tareas

        System.out.println("Tareas: ");

        for (int x = 0; x < tareas.size(); x++) {

            System.out.println(tareas.get(x));

        }

    }

}

```

• Principal.java

```

package codigo;
public class Principal {

    public static void main(String[] args) {

        Tareas tareas = new Tareas();

        tareas.agregarTarea(); //Añade una tarea a la lista
        tareas.agregarTarea(); //Añade una tarea a la lista
        tareas.agregarTarea(); //Añade una tarea a la lista

        tareas.eliminarTarea(); //Elimina una tarea

        tareas.mostrarTareas(); //Muestra las tareas

    }

}

```

```
Problems @ Javadoc
<terminated> Principal (27) [Java]
Añade una tarea:
Limpiar
Tarea Limpiar añadida

Añade una tarea:
Cocinar
Tarea Cocinar añadida

Añade una tarea:
Colada
Tarea Colada añadida

Elimina una tarea:
Cocinar
Tarea Cocinar eliminada

Tareas:
Limpiar
Colada
```

74.

• Edad.java

```
package codigo;

import java.util.HashMap;
import java.util.Scanner;

public class Edad {

    static HashMap<String, Integer> edades = new HashMap<>();

    public static void crearHashMap() { //Guarda datos en un HashMap

        edades.put("Juan", 27);
        edades.put("María", 32);
        edades.put("Luisa", 16);
    }

    public static void buscarEdad() { //Muestra la edad de la persona
introducida

        Scanner obj = new Scanner(System.in); //Inicializa un objeto
Scanner

        System.out.println("Introduce un nombre: ");
        String nombre = obj.nextLine(); //Recoge el nombre indicado por
el usuario

        System.out.println("Edad de " + nombre + ": "
+edades.get(nombre));

        obj.close();
    }
}
```

```
}
```

• Principal.java

```
package codigo;

public class Principal {

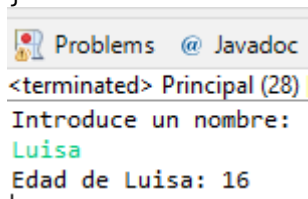
    public static void main(String[] args) {

        Edad.crearHashMap(); //Crea un HashMap

        Edad.buscarEdad(); // Muestra la edad de la persona indicada

    }

}
```



```
Problems @ Javadoc
<terminated> Principal (28)
Introduce un nombre:
Luisa
Edad de Luisa: 16
```

75.

• Colores.java

```
package codigo;

import java.util.HashSet;
import java.util.Scanner;

public class Colores {

    static HashSet<String> colores = new HashSet<>();
    static Scanner obj = new Scanner(System.in); //Inicializa un objeto Scanner

    public static void añadirColor() { //Guarda colores en el HashSet

        System.out.println("Introduce un color: ");
        String color = obj.nextLine();
        colores.add(color);

        System.out.println("");

    }

    public static void verCantidadColores() { //Muestra cuantos colores hay en el HashSet

        System.out.println("Número de colores: " + colores.size());

    }

}
```

• Principal.java

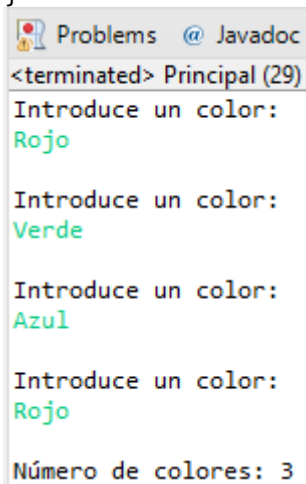
```
package codigo;

public class Principal {

    public static void main(String[] args) {

        Colores.añadirColor(); //Guarda colores en un HashSet
        Colores.añadirColor(); //Guarda colores en un HashSet
        Colores.añadirColor(); //Guarda colores en un HashSet
        Colores.añadirColor(); //Guarda colores en un HashSet

        Colores.verCantidadColores(); // Muestra la cantidad de colores
        en el HashSet
    }
}
```



76.

• Numeros.java

```
package codigo;

import java.util.ArrayList;
import java.util.Collections;

public class Numeros {

    static ArrayList<Integer> numeros = new ArrayList<>();

    public static void crearLista() { //Guarda números en una lista

        numeros.add(5);
        numeros.add(7);
        numeros.add(8);
        numeros.add(2);
        numeros.add(1);
        numeros.add(4);

        System.out.println("Lista desordenada: ");
    }
}
```



```

        System.out.println(numeros);
        System.out.println("");
    }

    public static void ordenarLista() { //Ordena la lista

        Collections.sort(numeros);

        System.out.println("Lista ordenada: ");
        System.out.println(numeros);
    }
}

```

• Principal.java

```

package codigo;

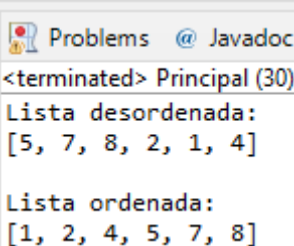
public class Principal {

    public static void main(String[] args) {

        Numeros.crearLista(); //Guarda números en una lista

        Numeros.ordenarLista(); //Ordena la lista
    }
}

```



Problems @ Javadoc
 <terminated> Principal (30)
 Lista desordenada:
 [5, 7, 8, 2, 1, 4]
 Lista ordenada:
 [1, 2, 4, 5, 7, 8]

77.

• Producto.java

```

package codigo;

import java.util.HashMap;

public class Producto {

    static HashMap<String, Integer> productos = new HashMap<>();

    public static void añadirProducto() { //Agrega productos al HashMap

        productos.put("Coche", 1200);
        productos.put("Lavadora", 200);
        productos.put("Pan", 12);
        productos.put("Leche", 7);
    }
}

```

```

    }

    public static void listarProductos() { //Devuelve los productos con
    precio mayor a 50

        for (int x : productos.values()) {

            if (x > 50) {

                for (String y : productos.keySet()) {

                    if (productos.get(y) == x) {

                        System.out.println(y + " : " + x +
"€");
                    }
                }
            }
        }
    }
}

```

• Principal.java

```

package codigo;

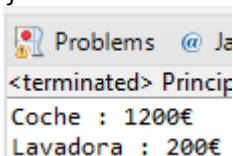
public class Principal {

    public static void main(String[] args) {

        Producto.añadirProducto(); //Agrega productos al HashMap
        Producto.añadirProducto(); //Agrega productos al HashMap
        Producto.añadirProducto(); //Agrega productos al HashMap
        Producto.añadirProducto(); //Agrega productos al HashMap

        Producto.listarProductos(); //Devuelve los productos con precio
mayor a 50
    }
}

```



Problems @ Ja
<terminated> Princip
Coche : 1200€
Lavadora : 200€

78.

• Frutas.java

```

package codigo;

import java.util.ArrayList;
import java.util.Iterator;

```

```

public class Frutas {

    static ArrayList<String> frutas = new ArrayList<>();

    public static void añadirFrutas() { //Crea una lista de frutas

        frutas.add("Pera");
        frutas.add("Manzana");
        frutas.add("Naranja");
        frutas.add("Plátano");

        System.out.println(frutas);
        System.out.println("");
    }

    public static void eliminarFruta(String fruta) { //Elimina la fruta
indicada

        Iterator<String> it = frutas.iterator();

        while (it.hasNext()) {

            String f = it.next();

            if (f.equals(fruta)) {

                it.remove();
            }
        }
        System.out.println(fruta + " eliminada");
        System.out.println(frutas);
    }
}

```

• Principal.java

```

package codigo;

public class Principal {

    public static void main(String[] args) {

        Frutas.añadirFrutas(); //Crea una lista de frutas

        Frutas.eliminarFruta("Naranja"); //Elimina la fruta indicada
    }
}

```

```

<terminated> Principal (32) [Java Application]
[Pera, Manzana, Naranja, Plátano]

Naranja eliminada
[Pera, Manzana, Plátano]

```

79.

• Nombres.java

```
package codigo;

import java.util.ArrayList;

public class Nombres {

    static ArrayList<String> nombres = new ArrayList<>();

    public static void añadirNombres() { //Crea una lista de nombres

        nombres.add("Juan");
        nombres.add("Luis");
        nombres.add("María");
        nombres.add("Antonio");

    }

    public static void mostrarNombres() { //Muestra los nombres de la
    lista en mayúsculas

        for (String nombre : nombres) {

            System.out.println(nombre.toUpperCase());

        }

    }

}
```

• Principal.java

```
package codigo;

public class Principal {

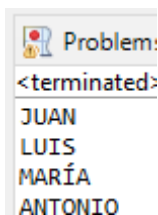
    public static void main(String[] args) {

        Nombres.añadirNombres(); //Crea una lista de nombres

        Nombres.mostrarNombres(); //Muestra los nombres de la lista en
    mayúsculas

    }

}
```



```
Problem:
<terminated>
JUAN
LUIS
MARÍA
ANTONIO
```

80.

• Movable.java

```
package codigo;

public interface Movable {

    void mover(); //Método abstracto
}
```

• Vehiculo.java

```
package codigo;

public abstract class Vehiculo {

    String id;

    public Vehiculo(String id) { //Constructor

        this.id = id;
    }
}
```

• Coche.java

```
package codigo;

public class Coche extends Vehiculo implements Movable {

    public Coche(String id) { //Constructor

        super(id);
    }

    public void mover() { //Muestra un mensaje

        System.out.println("El coche " + id + " se mueve por la
carretera");
    }
}
```

• Bicicleta.java

```
package codigo;

public class Bicicleta extends Vehiculo implements Movable {

    public Bicicleta(String id) { //Constructor

        super(id);
    }
}
```

```

        public void mover() { //Muestra un mensaje

            System.out.println("La bicicleta " + id + " pedalea");

        }
    }
}

```

• Principal.java

```

package codigo;

import java.util.ArrayList;

public class Principal {

    public static void main(String[] args) {

        Coche c1 = new Coche("Toyota");
        Coche c2 = new Coche("Corolla");
        Bicicleta b1 = new Bicicleta("Shimano");
        Bicicleta b2 = new Bicicleta("Stevens");

        ArrayList<Movable> vehiculos = new ArrayList<>();

        vehiculos.add(c1);
        vehiculos.add(c2);
        vehiculos.add(b1);
        vehiculos.add(b2);

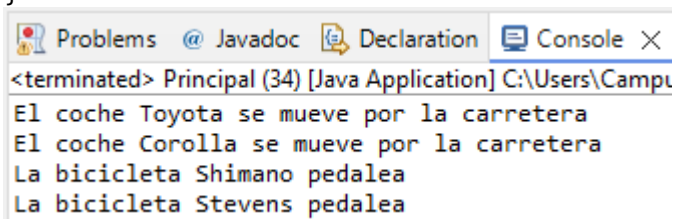
        for (int x = 0; x < vehiculos.size(); x++) { //Muestra un
mensaje dependiente del tipo de vehículo

            vehiculos.get(x).mover();

        }

    }
}

```



```

<terminated> Principal (34) [Java Application] C:\Users\Campu
El coche Toyota se mueve por la carretera
El coche Corolla se mueve por la carretera
La bicicleta Shimano pedalea
La bicicleta Stevens pedalea

```

81.

• Comunicable.java

```

package codigo;

public interface Comunicable {

    abstract String hacerSonido(); //Método abstracto

}

```

• Animal.java

```
package codigo;

public abstract class Animal implements Comunicable {

    String nombre;

    public Animal(String nombre) { //Constructor

        this.nombre = nombre;
    }
}
```

• Perro.java

```
package codigo;

public class Perro extends Animal implements Comunicable {

    public Perro(String nombre) { //Constructor

        super(nombre);
    }

    public String hacerSonido() { //Devuelve el sonido del animal

        return "Guau";
    }
}
```

• Gato.java

```
package codigo;

public class Gato extends Animal implements Comunicable {

    public Gato(String nombre) { //Constructor

        super(nombre);
    }

    public String hacerSonido() { //Devuelve el sonido del animal

        return "Miau";
    }
}
```

• Principal.java

```
package codigo;

import java.util.ArrayList;
```

```

public class Principal {

    public static void main(String[] args) {

        Perro p1 = new Perro("Toby");
        Perro p2 = new Perro("Rex");
        Gato g1 = new Gato("Bigotes");
        Gato g2 = new Gato("Calcetines");

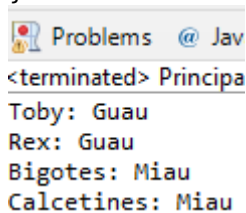
        ArrayList<Animal> animales = new ArrayList<>(); //Crea una lista
de animales

        animales.add(p1);
        animales.add(p2);
        animales.add(g1);
        animales.add(g2);

        for (int x = 0; x < animales.size(); x++) { //Muestra el nombre
y sonido del animal

            System.out.println(animales.get(x).nombre + ": " +
animales.get(x).hacerSonido());
        }
    }
}

```



```

<terminated> Principa
Toby: Guau
Rex: Guau
Bigotes: Miau
Calcetines: Miau

```

82.

• Calculable.java

```

package codigo;

public interface Calculable {

    abstract double calcularArea(); //Método abstracto
}

```

• Figura.java

```

package codigo;

public abstract class Figura implements Calculable {

    String color;

}

```


• Circulo.java

```
package codigo;

public class Circulo extends Figura implements Calculable {

    double radio;

    public Circulo(double radio, String color) { //Constructor

        this.radio = radio;
        this.color = color;
    }

    public double calcularArea() { //Calcula el área del círculo

        return 3.14 * radio * radio;
    }
}
```

• Rectangulo.java

```
package codigo;

public class Rectangulo extends Figura implements Calculable {

    double ancho;
    double alto;

    public Rectangulo(double ancho, double alto, String color) {
//Constructor

        this.ancho = ancho;
        this.alto = alto;
        this.color = color;
    }

    public double calcularArea() { //Calcula el área del rectángulo

        return ancho * alto;
    }
}
```

• Principal.java

```
package codigo;

import java.util.ArrayList;

public class Principal {

    public static void main(String[] args) {

        double total = 0;

        Circulo c1 = new Circulo(3, "rojo");
```

```

Circulo c2 = new Circulo(8, "verde");
Rectangulo r1 = new Rectangulo(4, 2, "azul");
Rectangulo r2 = new Rectangulo(5, 7, "naranja");

ArrayList<Figura> figuras = new ArrayList<>(); //Crea una lista
de figuras

figuras.add(c1);
figuras.add(c2);
figuras.add(r1);
figuras.add(r2);

for (int x = 0; x < figuras.size(); x++) { //Muestra el area de
las figuras

    System.out.println("Area de la figura: " +
figuras.get(x).calcularArea());

    total = total + figuras.get(x).calcularArea();
}

    System.out.println("Area total: " + total); //Muestra el area
total de la figura
}
}

```

83.

• Pagable.java

```

package codigo;

public interface Pagable {

    abstract double calcularSalarioMensual(); //Método abstracto
}

```

• Empleado.java

```

package codigo;

public abstract class Empleado implements Pagable {

    int id;
    String nombre;

}

```

• EmpleadoFijo.java

```

package codigo;

public class EmpleadoFijo extends Empleado implements Pagable{

    double salarioBase;
}

```

```

        public EmpleadoFijo(int id, String nombre, double salarioBase) {
//Constructor

            this.id = id;
            this.nombre = nombre;
            this.salarioBase = salarioBase;
        }

        public double calcularSalarioMensual() { //Calcula el salario mensual

            return salarioBase*31;
        }
    }
}

```

• EmpleadoPorHoras.java

```

package codigo;

public class EmpleadoPorHoras extends Empleado implements Pagable {

    double tarifaHora;
    int horasTrabajadas;

    public EmpleadoPorHoras(int id, String nombre, double tarifaHora, int
horasTrabajadas) { //Constructor

        this.id = id;
        this.nombre = nombre;
        this.tarifaHora = tarifaHora;
        this.horasTrabajadas = horasTrabajadas;
    }

    public double calcularSalarioMensual() { //Calcula el salario mensual

        return horasTrabajadas*tarifaHora;
    }
}

```

• Principal.java

```

package codigo;

import java.util.HashMap;

public class Principal {

    public static void main(String[] args) {

        EmpleadoFijo ef1 = new EmpleadoFijo(1, "Juan", 27.5);
        EmpleadoFijo ef2 = new EmpleadoFijo(2, "Matias", 31.43);
        EmpleadoPorHoras eh1 = new EmpleadoPorHoras(3, "Luis", 7, 12);
        EmpleadoPorHoras eh2 = new EmpleadoPorHoras(4, "Antonio", 5.12,
20);

        HashMap<Integer, Empleado> empleados = new HashMap<>(); //Crea
un diccionario de empleados
    }
}

```

```

empleados.put(ef1.id, ef1);
empleados.put(ef2.id, ef2);
empleados.put(eh1.id, eh1);
empleados.put(eh2.id, eh2);

    for (int x = 0; x < empleados.size(); x++) { //Muestra nombre y
salario de empleados

        System.out.println(empleados.get(x).nombre + ": " +
empleados.get(x).calcularSalarioMensual());
    }
}

```

Problems @ Javadoc Declaration Console X
 <terminated> Principal (37) [Java Application] C:\Users\CampusFP\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe (2 abr 2025, 12:24:59 - 12:25:00)
 Exception in thread "main" java.lang.NullPointerException: Cannot read field "nombre" because the return value of "java.util.HashMap.get(Object)" is null
 at codigo.Principal.main(Principal.java:23)

84.

• Notificable.java

```

package codigo;

public interface Notificable {

    abstract void enviarMensaje(String mensaje);
}

```

• NotificacionEmail.java

```

package codigo;

public class NotificacionEmail implements Notificable {

    String direccionCorreo;

    public NotificacionEmail(String direccionCorreo) { //Constructor

        this.direccionCorreo = direccionCorreo;
    }

    public void enviarMensaje(String mensaje) { //Muestra a quien se envia
el mensaje

        System.out.println("Mensaje " + mensaje + " enviado a: " +
direccionCorreo);
    }
}

```

• NotificacionSMS.java

```

package codigo;

public class NotificacionSMS implements Notificable {

    int numeroTelefono;
}

```

```

    public NotificacionSMS(int numeroTelefono) { //Constructor

        this.numeroTelefono = numeroTelefono;
    }

    public void enviarMensaje(String mensaje) { //Muestra a quien se envia
el mensaje

        System.out.println("Mensaje " + mensaje + " enviado a: " +
numeroTelefono);
    }
}

```

• Principal.java

```

package codigo;

import java.util.LinkedList;

public class Principal {

    public static void main(String[] args) {

        NotificacionEmail ne1 = new NotificacionEmail("correo1@gmail");
        NotificacionEmail ne2 = new NotificacionEmail("correo2@gmail");
        NotificacionSMS ns1 = new NotificacionSMS(456475736);
        NotificacionSMS ns2 = new NotificacionSMS(235469367);

        LinkedList<Notificable> notificaciones = new LinkedList<>();
//Crea una lista de notificaciones

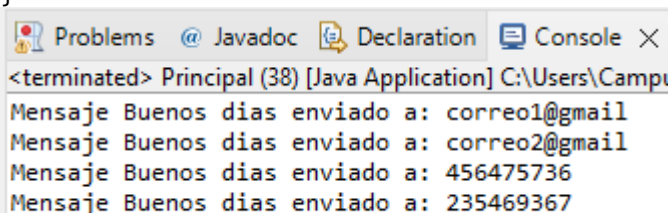
        notificaciones.add(ne1);
        notificaciones.add(ne2);
        notificaciones.add(ns1);
        notificaciones.add(ns2);

        String mensaje = "Buenos días";

        for (int x = 0; x < notificaciones.size(); x++) { //Muestra las
notificaciones

            notificaciones.get(x).enviarMensaje(mensaje);
        }
    }
}

```



```

<terminated> Principal (38) [Java Application] C:\Users\Campi
Mensaje Buenos días enviado a: correo1@gmail
Mensaje Buenos días enviado a: correo2@gmail
Mensaje Buenos días enviado a: 456475736
Mensaje Buenos días enviado a: 235469367

```

85.

• Describable.java

```
package codigo;

public interface Describable {

    abstract String describir();
}
```

• Item.java

```
package codigo;

public abstract class Item implements Describable{

    int id;
}
```

• Libro.java

```
package codigo;

public class Libro extends Item implements Describable{

    String titulo;

    public Libro(int id, String titulo) {

        this.id = id;
        this.titulo = titulo;
    }

    public String describir() {

        return "Libro: " + titulo;
    }
}
```

• Electronico.java

```
package codigo;

public class Electronico extends Item implements Describable{

    String marca;

    public Electronico(int id, String marca) {

        this.id = id;
        this.marca = marca;
    }

    public String describir() {
```

```

        return "Libro electronico marca: " + marca;
    }
}

```

• Principal.java

```

package codigo;

import java.util.HashMap;
import java.util.ArrayList;

public class Principal {

    public static void main(String[] args) {

        Libro l1 = new Libro(1, "Juramentada");
        Libro l2 = new Libro(2, "Palabras radiantes");
        Electronico e1 = new Electronico(3, "apple");
        Electronico e2 = new Electronico(4, "samsung");

        ArrayList<Item> libros = new ArrayList<>();
        ArrayList<Item> electronicos = new ArrayList<>();

        HashMap<String, ArrayList<Item>> items = new HashMap<>();

        items.put("Libros", libros);
        items.put("Electronicos", electronicos);

        libros.add(l1);
        libros.add(l2);
        electronicos.add(e1);
        electronicos.add(e2);

        for (int x = 0; x < items.size(); x++) {
            for (int y = 0; y < items.get(x).size(); y++) {

                System.out.println(items.get(x).get(y).describir());
            }
        }
    }
}

```

Problems Javadoc Declaration Console X
 <terminated> Principal (39) [Java Application] C:\Users\CampusFP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe (2 abr 2025, 13:29:31 - 13:29:31 elapsed: 0:00:00)
 Exception in thread "main" java.lang.NullPointerException: Cannot invoke "java.util.ArrayList.size()" because the return value of "java.util.HashMap.get(Object)" is null
 at codigo.Principal.main(Principal.java:30)

86.

• Tocable.java

```

package codigo;

public interface Tocable {

    abstract void tocarNota(String nota);
}

```

```
}
```

• Instrumento.java

```
package codigo;

public abstract class Instrumento implements Tocable{

    String marca;
}
```

• Guitarra.java

```
package codigo;

public class Guitarra extends Instrumento implements Tocable{

    int numeroCuerdas;

    public Guitarra(String marca, int numeroCuerdas) {

        this.marca = marca;
        this.numeroCuerdas = numeroCuerdas;
    }

    public void tocarNota(String nota) {

        System.out.println("La guitarra toca la nota: " + nota);
    }
}
```

• Piano.java

```
package codigo;

public class Piano extends Instrumento implements Tocable{

    int numeroTeclas;

    public Piano(String marca, int numeroTeclas) {

        this.marca = marca;
        this.numeroTeclas = numeroTeclas;
    }

    public void tocarNota(String nota) {

        System.out.println("El piano toca la nota: " + nota);
    }
}
```

• Principal.java

```
package codigo;
```



```

import java.util.HashSet;

public class Principal {

    public static void main(String[] args) {

        Guitarra g1 = new Guitarra("a", 7);
        Guitarra g2 = new Guitarra("a", 7);
        Piano p1 = new Piano("b", 20);
        Piano p2 = new Piano("b", 20);

        HashSet<Instrumento> instrumentos = new HashSet<>();

        instrumentos.add(g1);
        instrumentos.add(g2);
        instrumentos.add(p1);
        instrumentos.add(p2);

    }

}

```

87.

• Ejecutable.java

```

package codigo;

public interface Ejecutable {

    abstract void ejecutar();

}

```

• Tarea.java

```

package codigo;

public abstract class Tarea implements Ejecutable {

    String descripcion;

    int prioridad;

}

```

• TareaSimple.java

```

package codigo;

public class TareaSimple extends Tarea implements Ejecutable {

    public TareaSimple(String descripcion, int prioridad) {

        this.descripcion = descripcion;

    }

}

```

```

        this.prioridad = prioridad;
    }

    public void ejecutar() {
        System.out.println("Ejecutando tarea: " + descripcion);
    }
}

```

• Principal.java

```

package codigo;

import java.util.ArrayList;

public class Principal {

    public static void main(String[] args) {

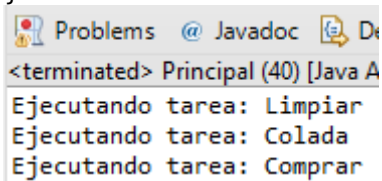
        TareaSimple t1 = new TareaSimple("Limpiar", 1);
        TareaSimple t2 = new TareaSimple("Colada", 2);
        TareaSimple t3 = new TareaSimple("Comprar", 3);

        ArrayList<Tarea> tareas = new ArrayList();
        tareas.add(t1);
        tareas.add(t2);
        tareas.add(t3);

        for (int x = 0; x < tareas.size(); x++) {

            tareas.get(x).ejecutar();
        }
    }
}

```



```

Problems @ Javadoc
<terminated> Principal (40) [Java A
Ejecutando tarea: Limpiar
Ejecutando tarea: Colada
Ejecutando tarea: Comprar

```

88.

• Medible.java

```

package codigo;

public interface Medible {

    abstract double leerValor();

    abstract String getUnidad();
}

```

• Sensor.java

```
package codigo;

public abstract class Sensor {

    int id;
    String ubicacion;
}
```

• SensorTemperatura.java

```
package codigo;

public class SensorTemperatura extends Sensor implements Medible{

    public SensorTemperatura(int id, String ubicacion) {

        this.id = id;
        this.ubicacion = ubicacion;
    }

    public double leerValor() {

        return 54.32;
    }

    public String getUnidad() {

        return "°C";
    }
}
```

• SensorHumedad.java

```
package codigo;

public class SensorHumedad extends Sensor implements Medible{

    public SensorHumedad(int id, String ubicacion) {

        this.id = id;
        this.ubicacion = ubicacion;
    }

    public double leerValor() {

        return 27.32;
    }

    public String getUnidad() {

        return "%";
    }
}
```

• Principal.java

```
package codigo;

import java.util.ArrayList;
import java.util.HashMap;

public class Principal {

    public static void main(String[] args) {

        SensorTemperatura t1 = new SensorTemperatura(1, "Sahara");
        SensorHumedad h1 = new SensorHumedad(2, "Sahara");

        ArrayList<Sensor> sensores = new ArrayList<>();

        sensores.add(t1);
        sensores.add(h1);

        HashMap<String, Double> mediciones = new HashMap<>();

        for (int x=0; x<sensores.size(); x++) {

            System.out.println(sensores.get(x).id + ": " +
sensores.get(x).leerValor() + sensores.get(x).getUnidad());
            mediciones.put(sensores.get(x).ubicacion,
sensores.get(x).leerValor());
        }

    }
}
```



Problems
<terminated>
1: 54.32°C
2: 27.32%

89.

• Auditable.java

• Cuenta.java

• CuentaAhorro.java

• CuentaCorriente.java

• Principal.java

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>