

## **BDD**

1. Escribe la instrucción para crear una tabla "clientes". Esta tabla tiene los siguientes campo:

- a. id\_cliente: entero (este campo es el campo clave)
- b. nombre: cadena de caracteres de tamaño 30
- c. dirección: cadena de caracteres de tamaño 50
- d. teléfono: cadena de caracteres de tamaño 10

```
CREATE TABLE clientes (  
    id_cliente INT PRIMARY KEY,  
    nombre VARCHAR(30),  
    dirección VARCHAR(50),  
    teléfono VARCHAR(10)  
);
```

2. Escribe la instrucción para insertar 2 clientes en esa tabla (Invéntate los valores)

```
INSERT INTO clientes (id_cliente, nombre, direccion, telefono)  
VALUES  
    (1, 'Juan Perez', 'Calle 123, Ciudad A', '123456789'),  
    (2, 'Maria Rodriguez', 'Avenida XYZ, Ciudad B', '098765432');
```

3. Escribe una instrucción para visualizar todos los registros y toda la información de la tabla clientes

```
SELECT * FROM CLIENTES
```

4. Escribe una instrucción para visualizar el nif de un cliente llamado "pepe"

```
SELECT nif FROM CLIENTES WHERE NOMBRE='pepe';
```

5. Escribe una instrucción para visualizar el nombre de cliente y el teléfono de todos los clientes de la tabla

```
SELECT nombre, telefono FROM CLIENTES
```

## PROGRAMACIÓN

1. Escribe el código necesario para mostrar por pantalla el contenido del siguiente array

Int numeros={5,3,1,0,7}

```
public static void main(String[] args) {  
    int[] numeros = {5, 3, 1, 0, 7};  
  
    System.out.println("Contenido del array numeros:");  
    for (int i = 0; i < numeros.length; i++) {  
        System.out.println(numeros[i]);  
    }  
}
```

2. Crea una clase "Cliente" que tenga las propiedades:

- a. id\_cliente: entero
- b. nombre: cadena de caracteres
- c. dirección: cadena de caracteres
- d. teléfono: cadena de caracteres

```
public class Cliente {  
    int id_cliente;  
    String nombre;  
    String direccion;  
    String telefono;  
  
    public Cliente(int id_cliente, String nombre, String direccion, String telefono) {  
        this.id_cliente = id_cliente;  
        this.nombre = nombre;  
        this.direccion = direccion;  
        this.telefono = telefono;  
    }  
}
```

3. En la clase "Cliente", añade un método para visualizar la información de un cliente

```
public void mostrarInformacion() {  
    System.out.println("ID Cliente: " + id_cliente);  
    System.out.println("Nombre: " + nombre);  
    System.out.println("Dirección: " + direccion);  
    System.out.println("Teléfono: " + telefono);  
}
```

4. En la clase cliente, crea un método void pedirDatos() que pida por pantalla todos los datos de un cliente. No hay que hacer control de excepciones.

```
public void pedirDatos() {  
    Scanner teclado = new Scanner( source: System.in);  
  
    System.out.print( s: "ID: ");  
    this.id_cliente = teclado.nextInt();  
    teclado.nextLine();  
  
    System.out.print( s: "Nombre: ");  
    this.nombre = teclado.nextLine();  
  
    System.out.print( s: "Dirección: ");  
    this.direccion = teclado.nextLine();  
  
    System.out.print( s: "Teléfono: ");  
    this.telefono = teclado.nextLine();  
}
```

5. En la clase cliente, crea un método void guardarDatos() que escriba los datos de un cliente en un archivo misclientes.txt.

```
public void guardarDatos() {  
    try {  
        FileWriter escritura = new FileWriter( fileName: "misclientes.txt", append: true);  
        escritura.write("ID Cliente: " + this.id_cliente + "\n");  
        escritura.write("Nombre: " + this.nombre + "\n");  
        escritura.write("Dirección: " + this.direccion + "\n");  
        escritura.write("Teléfono: " + this.telefono + "\n\n");  
        escritura.close();  
        System.out.println( x: "Datos guardados correctamente");  
    } catch (IOException e) {  
        System.out.println( x: "Error al guardar el archivo.");  
        e.printStackTrace();  
    }  
}
```

## PARTE II

1. Crea un programa en el que pidas dos números enteros por pantalla y calcules su división, para mostrar el resultado por pantalla.

```
public static void main(String[] args) {
    // TODO code application logic here
    int num1, num2;
    double resultado;

    Scanner scanner=new Scanner( source:System.in);
    System.out.print( s:"Introduzca un numero: ");
    num1=scanner.nextInt();
    System.out.print( s:"Introduzca un segundo numero: ");
    num2=scanner.nextInt();

    resultado=num1/num2;
    System.out.println("El resultado de la division es: "+resultado);
}
```

2. Repite el programa con control de Excepciones: controla la excepción de que el usuario escriba algo que no sea un número para que el programa no aborte mostrando un mensaje que diga "No se ha podido realizar la operación".

```
try{

    System.out.print( s:"Introduzca un numero: ");
    num1=scanner.nextInt();
    System.out.print( s:"Introduzca un segundo numero: ");
    num2=scanner.nextInt();

    resultado=num1/num2;
    System.out.println("El resultado de la division es: "+resultado);
}catch (Exception e) {
    System.out.println( s:"No se ha podido realizar la operación");
}
```

3. En el código anterior, controla la excepción que se produce si el segundo número es un 0. El control debe hacer que se vuelva a pedir otra vez el número.