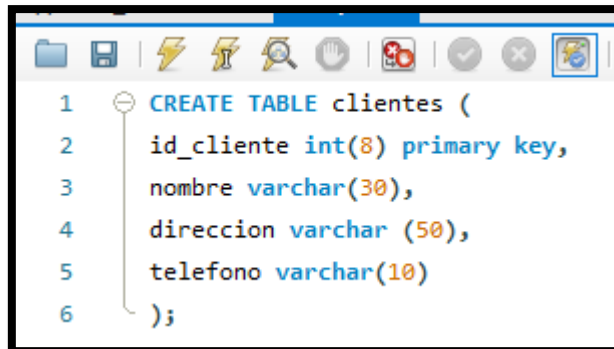


Evaluación inicial: Acceso a Datos

BBDD

1.-



```
1 CREATE TABLE clientes (  
2     id_cliente int(8) primary key,  
3     nombre varchar(30),  
4     direccion varchar (50),  
5     telefono varchar(10)  
6 );
```

2.-

```
INSERT INTO clientes VALUES (44654789,'Angel Rodriguez','Rosario 23','967451278');  
INSERT INTO clientes VALUES (16145125,'Enrico Prieto','alamenda 23','967458923');
```

3.-

```
select * from clientes
```

4.-

```
SELECT id_cliente FROM clientes WHERE nombre LIKE 'pepe';
```

5.-

```
SELECT nombre, telefono FROM clientes
```

PROGRAMACIÓN:

1.-

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

2.-

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

```
@Override  
public String toString() {  
    return "ID de cliente: "+this.id_cliente+"\nNombre: "+this.nombre  
        +"\nDirección: "+this.direccion+"\nTeléfono: "+this.telefono;  
}
```

4.-

```
public void pedirDatosCliente () {
    Scanner teclado = new Scanner (source: System.in);

    System.out.println( x: "Introduzca los datos del cliente:");
    System.out.print( s: "Identificador del cliente --> ");
    this.id_cliente = teclado.nextInt();
    teclado.nextLine();
    System.out.print( s: "Nombre del cliente --> ");
    this.nombre = teclado.nextLine();
    System.out.print( s: "Direccion del cliente --> ");
    this.direccion = teclado.nextLine();
    System.out.print( s: "Teléfono del cliente --> ");
    this.telefono = teclado.nextLine();
}
```

5.-

```
public void guardarDatos() {
    String ruta = "src/clases/misclientes.txt";
    File archivo = new File (pathname: ruta);

    if (archivo.exists() && archivo.isFile()) {
        try(BufferedWriter buferEscritura = new BufferedWriter(new FileWriter( fileName: ruta, append: true))) {
            buferEscritura.write( str: this.toString());
            buferEscritura.newLine();
        } catch (IOException ex) {
            System.out.println( x: "Error al crear o escribir en el archivo.");
        }
    } else if (!archivo.isDirectory()) {
        try(BufferedWriter buferEscritura = new BufferedWriter(new FileWriter( fileName: ruta))) {
            buferEscritura.write( str: this.toString());
            buferEscritura.newLine();
        } catch (IOException ex) {
            System.out.println( x: "Error al crear o escribir en el archivo.");
        }
    } else {
        System.out.println( x: "Imposible escribir en esa ruta, la ruta es un directorio. ");
    }
}
```

PROGRAMACION PARTE 2:

1.-

```
private static void dividirDosnumeros(){
    Scanner teclado = new Scanner (source: System.in);
    int dividendo, divisor;
    System.out.println(x: "=====  
Calcule la division de dos números =====");
    System.out.print(s: "Introduzca el dividendo --> ");
    dividendo = teclado.nextInt();
    System.out.print(s: "Introduzca el divisor --> ");
    divisor = teclado.nextInt();
    System.out.println("El resultado de la división es :"+ ((double)dividendo/divisor));
}
```

2.-

```
private static void dividirDosnumerosExcepciones1(){
    Scanner teclado = new Scanner (source: System.in);
    int dividendo=0, divisor=0;
    boolean datosIncorrectos = false;
    System.out.println(x: "=====  
Calcule la division de dos números =====");
    System.out.print(s: "Introduzca el dividendo --> ");
    try{
        dividendo = teclado.nextInt();
    }catch (InputMismatchException ex){
        datosIncorrectos = true;
        teclado.next();
    }
    System.out.print(s: "Introduzca el divisor --> ");
    try{
        divisor = teclado.nextInt();
    }catch (InputMismatchException ex){
        datosIncorrectos = true;
        teclado.next();
    }
    if(datosIncorrectos)
        System.out.println(x: "No se ha podido realizar la operacion");
    else
        System.out.println("El resultado de la división es :"+ ((double)dividendo/divisor));
}
```

3.-

```
private static void dividirDosnumerosExcepciones2() {
    Scanner teclado = new Scanner ( source: System.in);
    int dividendo=0, divisor=0;
    boolean datosIncorrectos = false;
    System.out.println( x: "===== Calcule la division de dos números =====");
    System.out.print( s: "Introduzca el dividendo --> ");
    try{
        dividendo = teclado.nextInt();
    }catch (InputMismatchException ex){
        datosIncorrectos = true;
        teclado.next();
    }
    System.out.print( s: "Introduzca el divisor --> ");
    try{
        divisor = teclado.nextInt();
        while (divisor == 0){
            System.out.println( x: "No es posible divisor por 0, introduzca un nuevo divisor.");
            System.out.print( s: "Introduzca el divisor --> ");
            divisor = teclado.nextInt();
        }
    }catch (InputMismatchException ex){
        datosIncorrectos = true;
        teclado.next();
    }
    if(datosIncorrectos)
        System.out.println( x: "No se ha podido realizar la operacion");
    else
        System.out.println("El resultado de la división es :"+ ((double) dividendo/divisor) );
}
```