



# CREAR Y CONSULTAR TABLA DB40

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

|                               |   |
|-------------------------------|---|
| 1. ENUNCIADO - OBJETIVOS..... | 3 |
| 2. DESARROLLO .....           | 3 |

## 1. ENUNCIADO - OBJETIVOS

En esta práctica se pedía comentar y ejecutar el código de un ejemplo sobre la creación y consulta de tablas utilizando la librería DB4O en Java.

Este ejemplo cuenta con dos clases 'Persona' y 'Main', desde las cuales se implementan las variables hasta distintos ejemplos para filtrar objetos.

## 2. DESARROLLO

Primero, se visualiza el código de la clase Persona y se acompaña una breve explicación.

```
package db4o;

public class Persona {

    private String nombre;
    private int edad;
    private double peso;
    private double altura;

    public Persona() {

    }

    public Persona(String nombre, int edad, double peso, double altura) {
        super();
        this.nombre = nombre;
        this.edad = edad;
        this.peso = peso;
        this.altura = altura;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    public double getPeso() {
        return peso;
    }

    public void setPeso(double peso) {
        this.peso = peso;
    }

    public double getAltura() {
        return altura;
    }

    public void setAltura(double altura) {
        this.altura = altura;
    }

}
```

```

        @Override
        public String toString() {
            return "Persona [nombre=" + nombre + ", edad=" + edad + ", peso=" +
            peso + ", altura=" + altura + "]\n";
        }
    }
}

```

En esta clase, se declaran las variables que van a ser usadas al crear objetos de esta misma clase en la otra clase. Para ello, se instancian las mismas, se instancian un constructor vacío y un constructor con todas las variables. Además, se crean los getters y setters correspondientes, además de sobrescribir el método toString con las variables de la clase.

Vista esta clase, se presenta el código de la clase Main, donde se ejecuta el programa, además de una breve explicación.

```

package db4o;
import com.db4o.*;

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

        ObjectContainer db=Db4oEmbedded.openFile("personas.db4o");

        Persona p1=new Persona("Crispulo",30,120,1.59);
        Persona p2=new Persona("Robustiano",40,180,1.99);
        Persona p3=new Persona("Andrea Diaz Navas",20,56,1.70);
        Persona p4=new Persona("Rober Widoll",57,50,1.65);
        Persona p5=new Persona("Arturo",20,70,1.80);

        db.store(p1);
        db.store(p2);
        db.store(p3);
        db.store(p4);
        db.store(p5);

        //Todas las personas
        Persona p=new Persona();
        ObjectSet<Persona>result=db.queryByExample(p);
        System.out.println("Todas las Personas");
        while(result.hasNext()) {
            System.out.println(result.next());
        }

        // Todas las personas con la edad de 30
        p=new Persona(null,30,0,0);
        result=db.queryByExample(p);
        System.out.println("Personas con 30 anos");
        while(result.hasNext()) {
            System.out.println(result.next());
        }
    }
}

```

```

//Todas las personas con la edad de 20 y 1.80 de altura
p=new Persona(null,20,0,1.80);
result=db.queryByExample(p);
System.out.println("Personas con 20 anos y altura de 1.70");
    while(result.hasNext()) {
        System.out.println(result.next());
    }

//Todas las personas llamadas Rober Widoll
p=new Persona("Rober Widoll",0,0,0);
result=db.queryByExample(p);
System.out.println("Personas que se llamen Rober Widoll");
    while(result.hasNext()) {
        System.out.println(result.next());
    }

//BORRAR todas las personas llamadas Arturo
p=new Persona("Arturo",0,0,0);
result=db.queryByExample(p);
System.out.println("Personas borradas que se llamen Arturo");
    while(result.hasNext()) {
        db.delete(result.next());
    }

//Para ver si se han borrado las personas que se llamen Arturo
p=new Persona();
result=db.queryByExample(p);
System.out.println("Todas las personas");
    while(result.hasNext()) {
        System.out.println(result.next());
    }

//Cerramos la base de datos
db.close();
}
}

```

En esta clase Main, en primer lugar, se crea la conexión con la base de datos DB4O: se crea un objeto de la clase ObjectContainer, el cual abre el fichero de la base de datos en cuestión; al no tener creada una, se crea automáticamente al ejecutar este método openFile.

Posteriormente, se crean los objetos Persona que se quieran (en este caso, se han creado cinco, cada uno con ejemplos de diferentes variables).

Hecho esto, se guardan los objetos en la base de datos: a partir del objeto db de la clase ObjectContainer, se llama al método store y se guarda cada objeto.

Guardados los objetos, se empieza a hacer ejemplos para ver como funcionan los db4o. Por ejemplo, para visualizar a todas las personas insertadas y guardadas, se crea un objeto Persona nuevo, se crea un arraylist de la clase ObjectSet y del tipo Persona, que se iguala al método llamado con el db queryByExample, dentro del cual se introduce el objeto Persona creado.

Con un bucle while, se recorre este objeto de la clase ObjectSet y se visualizan todas las personas.

Resultado:

```
<terminated> Main (3) [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (17 r
Todas las Personas
Persona [nombre=Crispulo, edad=30, peso=120.0, altura=1.59]
Persona [nombre=Robustiano, edad=40, peso=180.0, altura=1.99]
Persona [nombre=Andrea Diaz Navas, edad=20, peso=56.0, altura=1.7]
Persona [nombre=Rober Widoll, edad=57, peso=50.0, altura=1.65]
Persona [nombre=Arturo, edad=20, peso=70.0, altura=1.8]
```

El siguiente ejemplo pretende encontrar todas las personas con 30 años de edad. Para ello, al objeto Persona creado en el anterior ejemplo se le instancia el constructor a null y 0 en todas las variables que no vayamos a buscar; en la edad, se pondrá un 30.

Y al igual que en el ejemplo anterior, se recorre el objeto result.

Resultado:

```
Personas con 30 anos
Persona [nombre=Crispulo, edad=30, peso=120.0, altura=1.59]
```

Los siguientes ejemplos son prácticamente iguales a este anterior: uno requiere dos atributos edad y altura y el otro el nombre.

Resultados:

```
Personas con 20 anos y altura de 1.70
Persona [nombre=Arturo, edad=20, peso=70.0, altura=1.8]
Personas que se llamen Rober Widoll
Persona [nombre=Rober Widoll, edad=57, peso=50.0, altura=1.65]
```

Finalmente, hay un ejemplo de eliminación de objetos. Se va a eliminar el objeto con nombre 'Arturo'. Para ello, hay un cambio respecto a los demás ejemplos, ya que en este se va a borrar el objeto según se busque dentro del bucle while. Posteriormente, se imprimen todas las personas para ver si es verdad que se ha eliminado el objeto pretendido.

Resultado:

```
Personas borradas que se llamen Arturo
Todas las personas
Persona [nombre=Crispulo, edad=30, peso=120.0, altura=1.59]
Persona [nombre=Robustiano, edad=40, peso=180.0, altura=1.99]
Persona [nombre=Andrea Diaz Navas, edad=20, peso=56.0, altura=1.7]
Persona [nombre=Rober Widoll, edad=57, peso=50.0, altura=1.65]
```

Por último y muy importante, para evitar problemas, se cierra la base de datos db con el método close.