

# EJERCICIO RESUELTO

## Módulo: Programación

### Consultas con Db4o

#### Descripción:

Realizar un programa java usando el SGBD Db4o, donde hagamos lo siguiente:

- ☐ Crear una clase jugador , con los siguientes atributos:
  - ☐ String nombre,
  - ☐ String deporte,
  - ☐ String ciudad,
  - ☐ int edad
- ☐ Insertar 5 objetos en la BD con los datos deseados.
- ☐ Realizar las siguientes consultas imprimiendo los datos obtenidos:
  - ☐ Buscar con un nombre concreto
  - ☐ Listar todos los datos.
  - ☐ Listar los mayores de una edad concreta
- ☐ Partiendo de la solución creada para la tarea anterior, crear un entorno gráfico que nos permita realizar las pruebas de las clases implementadas. Ahora dispondremos una opción de altas que nos permitirá introducir los objetos deseados.

#### Objetivos:

- ☐ Comprender las BD Orientadas a Objeto.
- ☐ Conocer las características de una BDOO.
- ☐ Explotar una BDOO Db4o haciendo inserciones, consultas simples y consultas resumen

#### Recursos:

- Acceso a Internet.
- Netbeans o Eclipse
- Db4o ObjectDatabase

#### Resolución:

En este ejercicio, modificaremos la clase “Consultas\_con\_Db4o” para poder trabajar mejor con ella:

```

package consultas_con_db4o;

import Database.Jugador;
import com.db4o.Db4oEmbedded;
import com.db4o.ObjectContainer;
import com.db4o.ObjectSet;
import com.db4o.query.Query;

public class Consultas_con_Db4o {
    private ObjectContainer db;
    private String url = "C:\\BasesDatos\\db4o\\Equipos2DB.db";

    public Consultas_con_Db4o() {
        this.db = Db4oEmbedded.openFile(
            Db4oEmbedded.newConfiguration(),
            url
        );
    }

    public void addPlayer(String name, String sport,
        String pais, int edad){
        db.store(new Jugador(name, sport, pais, edad));
    }

    public String buscarJugadores(String name, String sport,
        String pais, Integer edad){
        ObjectSet<Jugador> result = db.queryByExample(
            new Jugador (name, sport, pais, edad)
        );

        return toResultString(result);
    }

    public String buscarJugadoresMayoresQue(int edad){
        Query q = db.query();
        q.constrain(Jugador.class);
        q.descend("edad").constrain(edad).greater();
        ObjectSet result = q.execute();

        return toResultString(result);
    }

    public String toResultString(ObjectSet<Jugador> result){
        String stringResult = "";
        if(result.isEmpty()){
            stringResult+="No hay jugadores en la base de datos";
        }
        else{
            stringResult+="Número de jugadores recuperados: "
                + result.size();
            while(result.hasNext()){
                Jugador player = result.next();
                stringResult+="\n"+

```

```

        player.getNombre()+
        " ( "+player.getEdad()+" )"+
        "\tDeporte: "+player.getDeporte()+
        "\tPais: "+player.getPais();
    }
}
return stringResult;
}
}

```

Después crearemos una interfaz con swing:

The screenshot shows a Java Swing window titled "Fomento Ocupacional". It contains a form with four input fields labeled "Nombre", "Edad", "Deporte", and "Pais". Below these fields is an "Add player" button. Further down, there are two more input fields, and below them are three buttons: "Mostrar todos", "Buscar por nombre", and "Buscar mayores que". At the bottom of the window is a large empty rectangular area, likely for displaying a list of players.

Cuya clase será la siguiente:

```

package UI;

import consultas_con_db4o.Consultas_con_Db4o;

public class MainScreen extends javax.swing.JFrame {
    private Consultas_con_Db4o bd;

    public MainScreen() {
        bd = new Consultas_con_Db4o();
        initComponents();
    }

    @SuppressWarnings("unchecked")
    Generated Code

```

```

private void addPlayerButtonActionPerformed(java.awt.event.ActionEvent evt) {
    String name = this.nameText.getText();
    String age = this.EdadText.getText();
    String sport = this.deporteText.getText();
    String country = this.paisText.getText();
    bd.addPlayer(name, sport, country, Integer.parseInt(age));
    this.showTextArea.setText("Jugador añadido con éxito");
}

private void showAllButtonActionPerformed(java.awt.event.ActionEvent evt) {
    String showText = bd.buscarJugadores(null, null, null, null);
    this.showTextArea.setText(showText);
}

private void findByNameButtonActionPerformed(java.awt.event.ActionEvent evt) {
    String showText = bd.buscarJugadores(this.findByNameText.getText(),
        null, null, null);
    this.showTextArea.setText(showText);
}

private void findGreaterThanButtonActionPerformed(java.awt.event.ActionEvent evt) {
    int edad = Integer.parseInt(this.findGreaterThanOrEqualToText.getText());
    String showText = bd.buscarJugadoresMayoresQue(edad);
    this.showTextArea.setText(showText);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) { ...31 lines }

// Variables declaration - do not modify
private javax.swing.JLabel EdadJLabel;
private javax.swing.JTextField EdadText;
private javax.swing.JButton addPlayerButton;
private javax.swing.JLabel deporteJLabel;
private javax.swing.JTextField deporteText;
private javax.swing.JButton findByNameButton;
private javax.swing.JTextField findByNameText;
private javax.swing.JButton findGreaterThanOrEqualToButton;
private javax.swing.JTextField findGreaterThanOrEqualToText;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JLabel nameJLabel;
private javax.swing.JTextField nameText;
private javax.swing.JLabel paisJLabel;
private javax.swing.JTextField paisText;
private javax.swing.JButton showAllButton;
private javax.swing.JTextArea showTextArea;
// End of variables declaration
}

```