

Nombre de la práctica	PRÁCTICA FINAL			No.	1
Asignatura:	ESTRUCTURA DE DATOS	Carrera:	INGENIERIA EN SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	2 HRS

Nombre del alumno:	Arce Gabriel Luz Jimena.
--------------------	--------------------------

Encuadre con CACEI

No. Atributo	Atributos de egreso del PE que impactan en la asignatura	Criterios de desempeño		Indicadores	
A6	El estudiante mantiene la filosofía del autoaprendizaje	CD1	Capacitación externa a través de cursos presenciales o virtuales para permanecer constantemente actualizado	I1	Hacer uso de herramientas virtuales para su capacitación
				I2	Buscar capacitación en horarios fuera de clase
		CD2	Aplica los conocimientos adquiridos de manera autodidacta o en cursos externos en la resolución de problemas y desarrollo de trabajos y proyectos	I1	Aplica los conocimientos adquiridos en cursos extraescolares en su práctica académica
A7	El estudiante desarrolla proyectos y trabajos en equipo basándose en metodologías preestablecidas para lograr mayor calidad y eficiencia	CD2	Asume su responsabilidad en el desarrollo de trabajos y/o proyectos en equipo y en la entrega de resultados	I1	Participación activa en el desarrollo de trabajos y proyectos en equipo
				I2	Dirigir y organizar trabajo en equipo
				I3	Presentación y/o exposición de trabajos y proyectos en equipo

I. Competencia(s) específica(s):

Comprende las técnicas de administración de procesos para crear procesos empleando los mecanismos que presenta el sistema operativo para la comunicación y sincronización.

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado:

- IDE Entorno de desarrollo integrado (Eclipse).
- Wampserver (Servidor local).
- Laptop.

IV. Desarrollo de la práctica:

1. Crear la base de datos: Base: base1.
2. Crear la Tabla: artículos (código, descripción, precio).
3. Código de tipo varchar (3) llave primaria.
4. Descripción de tipo varchar (50) .
5. Precio de tipo Float .
6. Crear la interface, ahora agregaremos el control Jtable.
7. Botón BUSCAR mostrara la información del artículo si existe, en caso contrario mandara el mensaje de "Articulo no Existe".

8. Botón AGREGAR, dará de alta nuevos artículos, al agregar el nuevo código validara si existe mandará el mensaje que “ya existe el código”, y no podrá darlo de alta nuevamente, en caso de que no exista el código permitirá agregarlo.

9. Botón MODIFICAR, primero se buscará el articulo con el botón BUSCAR, si existe el articulo lo mostrará y se podrá modificar solo la descripción o el precio no así el código del artículo. Presionando el botón MODIFICAR guardaran los cambios.

10. Botón ELIMINAR, primero se buscará el articulo con el botón BUSCAR, si existe el articulo lo mostrará y se podrá eliminar.


11. Cuando se agreguen, modifique o elimine datos de artículos se deberán mostrarse de manera inmediata en el Jtable.

12. Calificación de la practica:

A) se revisa el programa en la computadora.

B) entregar manual de prácticas.

IMAGEN (COMO DEBE DE QUEDAR)



CONTROL DE ARTICULOS

CODIGO **BUSCAR**

DESCRIPCION

PRECIO

AGREGAR **MODIFICAR** **BORRAR**

LISTADO DE ARTICULOS

CONTROL: JTABLE

CÓDIGO EN JAVA:

programas_eclipse - basedatos01/src/basedatos02/ControlArticulos.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

*ControlArticulos.java ×

```
1 package basedatos02;
2
3 import javax.swing.*;
4 import javax.swing.table.DefaultTableModel;
5 import java.awt.EventQueue;
6 import java.awt.event.*;
7 import java.sql.*;
8
9 public class ControlArticulos extends JFrame {
10     private static final long serialVersionUID = 1L;
11     private JTextField tfCodigo, tfDescripcion, tfPrecio;
12     private JTable table;
13     private DefaultTableModel model;
14
15     public ControlArticulos() {
16
17         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18         setBounds(100, 100, 600, 400);
19         getContentPane().setLayout(null);
20
21         JLabel lblCodigo = new JLabel("CÓDIGO:");
22         lblCodigo.setBounds(30, 30, 80, 20);
23         getContentPane().add(lblCodigo);
24
25         JLabel lblDescripcion = new JLabel("DESCRIPCIÓN:");
26         lblDescripcion.setBounds(30, 60, 100, 20);
27         getContentPane().add(lblDescripcion);
28
29         JLabel lblPrecio = new JLabel("PRECIO:");
30         lblPrecio.setBounds(30, 90, 80, 20);
31         getContentPane().add(lblPrecio);
32
33         tfCodigo = new JTextField();
34         tfCodigo.setBounds(120, 30, 150, 20);
35         getContentPane().add(tfCodigo);
36
37         tfDescripcion = new JTextField();
38         tfDescripcion.setBounds(120, 60, 150, 20);
39         getContentPane().add(tfDescripcion);
40
41         tfPrecio = new JTextField();
42         tfPrecio.setBounds(120, 90, 150, 20);
43         getContentPane().add(tfPrecio);
44
45         JButton btnBuscar = new JButton("BUSCAR");
46         btnBuscar.setBounds(300, 30, 100, 20);
47         getContentPane().add(btnBuscar);
48
49         JButton btnAgregar = new JButton("AGREGAR");
50         btnAgregar.setBounds(30, 130, 100, 20);
51         getContentPane().add(btnAgregar);
52
53         JButton btnModificar = new JButton("MODIFICAR");
54         btnModificar.setBounds(240, 130, 100, 20);
55     }
```



```

56     getContentPane().add(btnModificar);
57
58     JButton btnBorrar = new JButton("BORRAR");
59     btnBorrar.setBounds(450, 130, 100, 20);
60     getContentPane().add(btnBorrar);
61
62     model = new DefaultTableModel(new Object[][] {}, new String[] { "CÓDIGO", "DESCRIPCIÓN", "PRECIO" });
63     table = new JTable(model);
64     table.setBounds(30, 193, 520, 150);
65     getContentPane().add(table);
66
67     JLabel lblListado = new JLabel("LISTADO DE ARTICULOS");
68     lblListado.setBounds(224, 170, 277, 13);
69     getContentPane().add(lblListado);
70
71     JLabel lblControl = new JLabel("CONTROL DE ARTICULOS");
72     lblControl.setBounds(224, 7, 210, 13);
73     getContentPane().add(lblControl);
74
75     actualizarTabla();
76
77     btnBuscar.addActionListener(e -> {
78         buscarArticulo();
79     });
80
81     btnAgregar.addActionListener(e -> {
82         agregarArticulo();
83     });
84
85     btnModificar.addActionListener(e -> {
86         modificarArticulo();
87     });
88
89     btnBorrar.addActionListener(e -> {
90         borrarArticulo();
91     });
92 }
93
94 private void buscarArticulo() {
95     try (Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/base2", "root", "")) {
96         String codigo = tfCodigo.getText();
97         String query = "SELECT * FROM articulos WHERE codigo = ?";
98         PreparedStatement ps = conexion.prepareStatement(query);
99         ps.setString(1, codigo);
100
101         ResultSet rs = ps.executeQuery();
102         if (rs.next()) {
103             tfDescripcion.setText(rs.getString("descripcion"));
104             tfPrecio.setText(rs.getString("precio"));
105         } else {
106             JOptionPane.showMessageDialog(this, "El articulo no existe");
107             limpiarCampos();
108         }
109     } catch (SQLException ex) {
110         ex.printStackTrace();
111     }
112 }
113
114 private void agregarArticulo() {
115     try (Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/base2", "root", "")) {
116         String codigo = tfCodigo.getText();
117         String descripcion = tfDescripcion.getText();

```



```

118 String precio = tfPrecio.getText();
119
120 String validarQuery = "SELECT * FROM articulos WHERE codigo = ?";
121 PreparedStatement validarPs = conexion.prepareStatement(validarQuery);
122 validarPs.setString(1, codigo);
123 ResultSet rs = validarPs.executeQuery();
124
125 if (rs.next()) {
126     JOptionPane.showMessageDialog(this, "Ya existe el código.");
127 } else {
128     String query = "INSERT INTO articulos (codigo, descripcion, precio) VALUES (?, ?, ?)";
129     PreparedStatement ps = conexion.prepareStatement(query);
130     ps.setString(1, codigo);
131     ps.setString(2, descripcion);
132     ps.setString(3, precio);
133     ps.executeUpdate();
134     JOptionPane.showMessageDialog(this, "Artículo agregado.");
135     actualizarTabla();
136     limpiarCampos();
137 }
138 } catch (SQLException ex) {
139     ex.printStackTrace();
140 }
141 }
142
143 private void modificarArticulo() {
144     try (Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/base2", "root", "")) {
145         String codigo = tfCodigo.getText();
146         String descripcion = tfDescripcion.getText();
147         String precio = tfPrecio.getText();
148
149         String query = "UPDATE articulos SET descripcion = ?, precio = ? WHERE codigo = ?";
150         PreparedStatement ps = conexion.prepareStatement(query);
151         ps.setString(1, descripcion);
152         ps.setString(2, precio);
153         ps.setString(3, codigo);
154
155         int rowsAffected = ps.executeUpdate();
156         if (rowsAffected > 0) {
157             JOptionPane.showMessageDialog(this, "Artículo modificado.");
158             actualizarTabla();
159             limpiarCampos();
160         } else {
161             JOptionPane.showMessageDialog(this, "Artículo no encontrado.");
162         }
163     } catch (SQLException ex) {
164         ex.printStackTrace();
165     }
166 }
167
168 private void borrarArticulo() {
169     try (Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/base2", "root", "")) {
170         String codigo = tfCodigo.getText();
171         String query = "DELETE FROM articulos WHERE codigo = ?";
172         PreparedStatement ps = conexion.prepareStatement(query);
173         ps.setString(1, codigo);
174
175         int rowsAffected = ps.executeUpdate();
176         if (rowsAffected > 0) {
177             JOptionPane.showMessageDialog(this, "Artículo eliminado.");
178             actualizarTabla();
179             limpiarCampos();
180         } else {
181             JOptionPane.showMessageDialog(this, "Artículo no encontrado.");
182         }
183     }
184 }

```



```
182     }
183     } catch (SQLException ex) {
184         ex.printStackTrace();
185     }
186 }
187
188 private void actualizarTabla() {
189     try (Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/base2", "root", "")) {
190         String query = "SELECT * FROM articulos";
191         Statement st = conexion.createStatement();
192         ResultSet rs = st.executeQuery(query);
193
194         model.setRowCount(0);
195         while (rs.next()) {
196             model.addRow(new Object[] { rs.getString("codigo"), rs.getString("descripcion"), rs.getString("precio") });
197         }
198     } catch (SQLException ex) {
199         ex.printStackTrace();
200     }
201 }
202
203 private void limpiarCampos() {
204     tfCodigo.setText("");
205     tfDescripcion.setText("");
206     tfPrecio.setText("");
207 }
208
209 public static void main(String[] args) {
210     EventQueue.invokeLater(() -> {
211         try {
212             ControlArticulos frame = new ControlArticulos();
213             frame.setVisible(true);
214         } catch (Exception e) {
215             e.printStackTrace();
216         }
217     });
218 }
219 }
220 }
```

INTERFACE:

BASE DE DATOS:

localhost/phpmyadmin/index.php?route=/table/structure&db=base2&table=articulos

Servidor: MySQL:3306 » Base de datos: base2 » Tabla: articulos

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	codigo	varchar(3)	utf8mb4_0900_ai_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2	descripcion	varchar(50)	utf8mb4_0900_ai_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	precio	float		No	Ninguna			Cambiar Eliminar Más

☐ Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único

Imprimir Mover columnas Normalizar

Agregar 1 columna(s) después de precio Continuar

PRUEBAS REALIZADAS CON DIFERENTES DATOS:

1.- AGREGAR:

CONTROL DE ARTICULOS

CÓDIGO: 1

BUSCAR

DESCRIPCIÓN: PELOTA

PRECIO: 15

AGREGAR

MODIFICAR

BORRAR

LISTADO DE ARTICULOS


CONTROL DE ARTICULOS

CÓDIGO:

DESCRIPCIÓN:

PRECIO:

Message

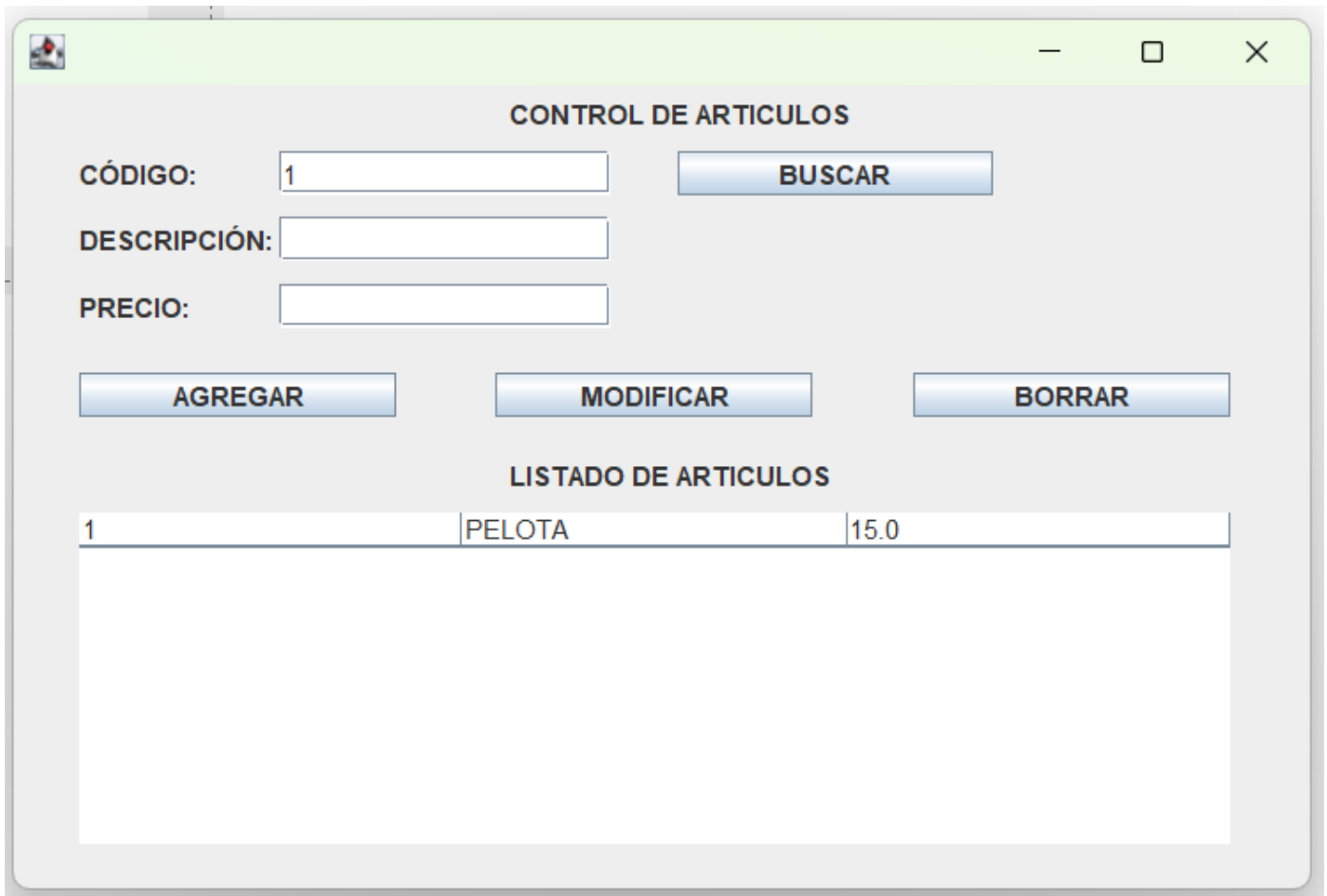
 Artículo agregado.

PRECIO:

LISTADO DE ARTICULOS

1	PELOTA	15.0
---	--------	------

2.- BUSCAR:



CONTROL DE ARTICULOS

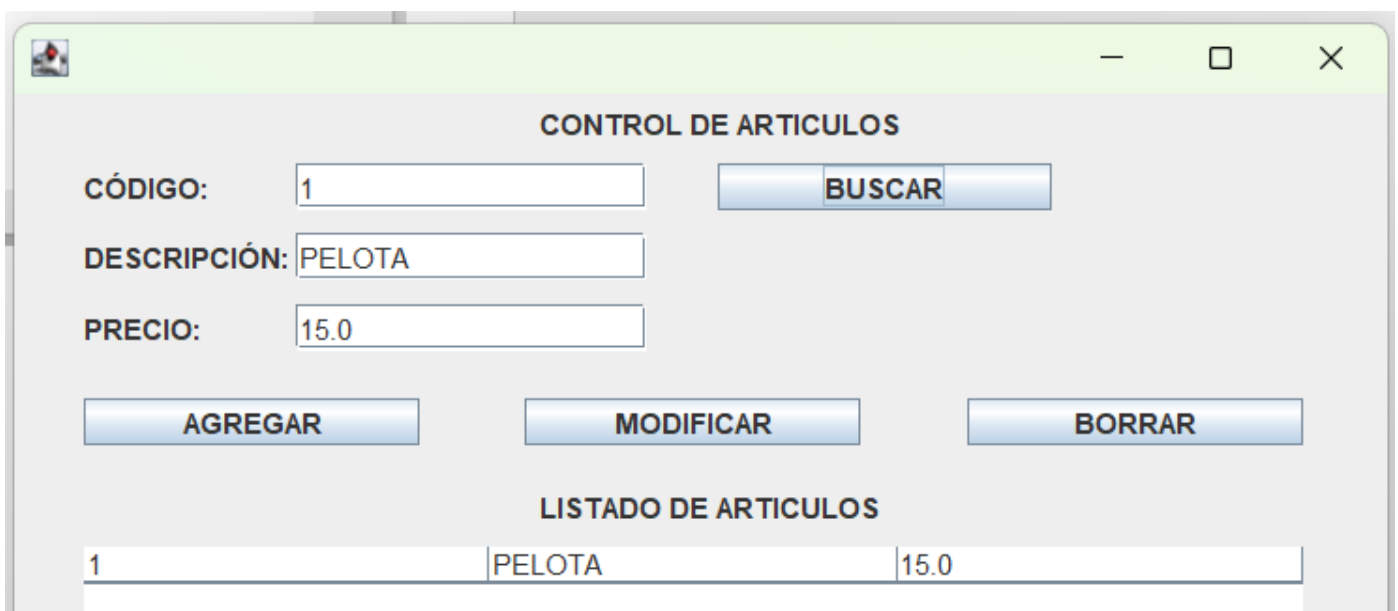
CÓDIGO:

DESCRIPCIÓN:

PRECIO:

LISTADO DE ARTICULOS

1	PELOTA	15.0
---	--------	------



CONTROL DE ARTICULOS

CÓDIGO:

DESCRIPCIÓN:

PRECIO:

LISTADO DE ARTICULOS

1	PELOTA	15.0
---	--------	------

3. MODIFICAR:

CONTROL DE ARTICULOS

CÓDIGO:

DESCRIPCIÓN:

PRECIO:

LISTADO DE ARTICULOS

1	PELOTA	15.0
---	--------	------

CONTROL DE ARTICULOS


CÓDIGO:

DESCRIPCIÓN:

PRECIO:

1

Message

 **Articulo modificado.**

CONTROL DE ARTICULOS

CÓDIGO: **BUSCAR**

DESCRIPCIÓN:

PRECIO:

AGREGAR **MODIFICAR** **BORRAR**

LISTADO DE ARTICULOS

1	BALÓN	50.0
---	-------	------

4.- BORRAR:

CONTROL DE ARTICULOS

CÓDIGO: **BUSCAR**

DESCRIPCIÓN:

PRECIO:

AGREGAR **MODIFICAR** **BORRAR**

LISTADO DE ARTICULOS

1	BALÓN	50.0
---	-------	------

CONTROL DE ARTICULOS


CÓDIGO:

DESCRIPCIÓN:

PRECIO:

1

Message

 **Articulo eliminado.**

CONTROL DE ARTICULOS

CÓDIGO:

DESCRIPCIÓN:

PRECIO:

LISTADO DE ARTICULOS

BUSCAR SI EL ARTICULO NO EXISTE:

CONTROL DE ARTICULOS

CÓDIGO: **BUSCAR**

DESCRIPCIÓN:

PRECIO:

AGREGAR **BORRAR**

Message

El articulo no existe

OK

BASE DE DATOS:

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

codigo	descripcion	precio
1	PELOTA	13
2	LIBRETA	22

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Operaciones sobre los resultados de la consulta

Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

V. Conclusiones:

En conclusión, a través de esta práctica, se logró desarrollar una aplicación de gestión de artículos utilizando Java y una base de datos, que permitió implementar funcionalidades para buscar, agregar, modificar y eliminar artículos de manera eficiente y efectiva. Se demostró la capacidad para diseñar e implementar bases de datos, crear interfaces gráficas de usuario y programar en Java para interactuar con bases de datos, lo que permitió desarrollar habilidades prácticas en el diseño y la implementación de una aplicación de gestión de artículos.

Además, se validó la existencia de códigos de artículos y se mostraron los cambios en la base de datos de manera inmediata en el JTable, lo que permitió una visualización clara y precisa de la información. La aplicación también permitió la modificación y eliminación de artículos de manera segura y controlada.

En general, esta práctica permitió desarrollar habilidades prácticas en el diseño y la implementación de una aplicación de gestión de artículos, lo que puede ser aplicado en diferentes contextos y escenarios. Además, se demostró la importancia de la validación y la seguridad en la implementación de aplicaciones que manejan información sensible.