

# Proyecto DAO

REALIZADO POR

Juan Carlos Tauroni Roldán, Fernando Vaz Carvalho, Javier Menéndez  
Abreu, Ziqiang Chen

Este proyecto se ha realizado con una BBDD mysql, y con el lenguaje de programación JAVA.

Se han creado 6 tablas, Empleado, Empresa, Fabricante, Inventario, Producto y Tienda.

SE ha implementado el método de interfaces dao, que nos permite utilizar una interfaz común durante todo el código, para cada tabla.

```
CREATE TABLE EMPRESA(  
  ID INT(5) NOT NULL AUTO_INCREMENT,  
  NOMBRE VARCHAR(50) NOT NULL,  
  DIRECCION VARCHAR(200) NOT NULL,  
  CP INT(10),  
  PAIS VARCHAR(50) NOT NULL,  
  EMAIL VARCHAR(50) NOT NULL,  
  TELEFONO VARCHAR(30) NOT NULL,  
  PRIMARY KEY (ID)  
);
```

*Empresa.*

```
CREATE TABLE TIENDA(  
  ID INT(5) NOT NULL AUTO_INCREMENT,  
  NOMBRE VARCHAR(50) NOT NULL,  
  DIRECCION VARCHAR(200) NOT NULL,  
  CP INT(5),  
  PAIS VARCHAR(50) NOT NULL,  
  PROVINCIA VARCHAR(50) NOT NULL,  
  EMAIL VARCHAR(50) NOT NULL,  
  TELEFONO VARCHAR(30) NOT NULL,  
  EMPRESASID INT(5) NOT NULL,  
  PRIMARY KEY (ID),  
  FOREIGN KEY (EMPRESASID) REFERENCES EMPRESA(ID)  
);
```

*Tienda.*

```
CREATE TABLE EMPLEADO(  
  ID INT(5) NOT NULL AUTO_INCREMENT,  
  NOMBRE VARCHAR(50) NOT NULL,  
  APELLIDOS VARCHAR(50) NOT NULL,  
  FECNA DATE NOT NULL,  
  DNI VARCHAR(9) NOT NULL,  
  EMAIL VARCHAR(50) NOT NULL,  
  TELEFONO VARCHAR(30) NOT NULL,  
  EMPRESASID INT(5) NOT NULL,  
  TIENDASID INT(5),  
  PRIMARY KEY (ID),  
  FOREIGN KEY (EMPRESASID) REFERENCES EMPRESA(ID),  
  FOREIGN KEY (TIENDASID) REFERENCES TIENDA(ID)  
);
```

*Empleado.*

```
CREATE TABLE PRODUCTO(
  ID INT(5) NOT NULL AUTO_INCREMENT,
  NOMBRE VARCHAR(50) NOT NULL,
  DESCRIPCION VARCHAR(200) NOT NULL,
  PRECIO DECIMAL(7,3) NOT NULL,
  FABRICANTEID INT(5) NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (FABRICANTEID) REFERENCES FABRICANTE(ID)
);
```

*Producto.*

```
CREATE TABLE INVENTARIO (
  ID INT(5) NOT NULL AUTO_INCREMENT,
  PRODUCTOID INT(5) NOT NULL,
  TIENDAID INT(5) NOT NULL,
  CANTIDAD INT(5) NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (PRODUCTOID) REFERENCES PRODUCTO(ID),
  FOREIGN KEY (TIENDAID) REFERENCES TIENDA(ID)
)
```

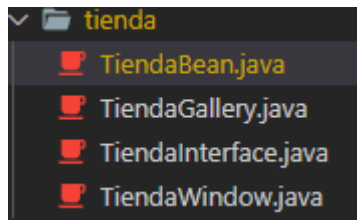
*Inventario.*

Para realizar el proyecto utilizando el modelo DAO, se han creado paquetes independientes con un archivo .java, que contendrá el método main.

Este archivo es el archivo MainWindow.java

Cada paquete corresponde a una tabla de las que se han creado en BBDD.

EJ:



El archivo interface contendrá los principales métodos utilizados por mysql. En este caso son:

```
public void loadJDBC();
public void connect();
public void displayAllInventario(DefaultTableModel model);
public void addInventario() ;
public void findInventarioById(DefaultTableModel model);
public void deleteInventario(DefaultTableModel model, int selectedRow);
public void updateInventario(DefaultTableModel model, int selectedRow);
```

Las funciones sql que van a utilizar.

El bean será el encargado de crear los métodos declarados en la interfaz, de cargar la conexión y el driver,

*Tabla*

Agregar Empresa

?

nombre:

direccion:

cp:

pais:

email:

telefono:

OK Cancel

*Formulario Agregar*

Actualizar Empresa

?

id:

4

nombre:

ASUS

direccion:

15 Li-Te Rd., Peitou, Taipei 112, Taiwan

cp:

112

pais:

Taiwan

email:

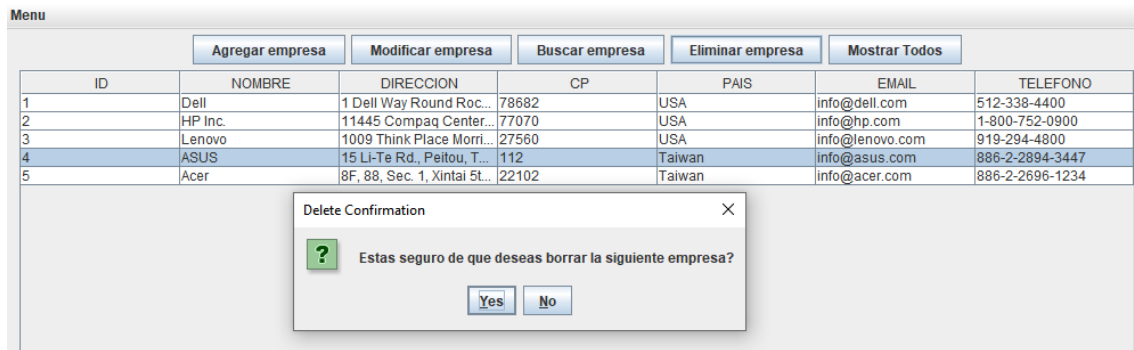
info@asus.com

telefono:

886-2-2894-3447

OK Cancel

*Actualizar empresa*



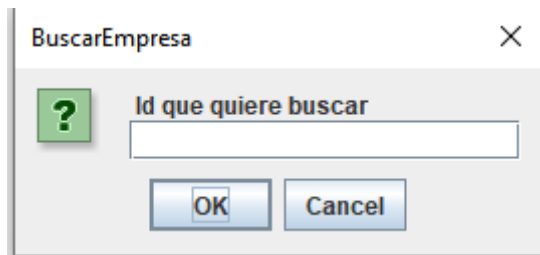
### Eliminar empresa

Tanto las operaciones borrar y editar requiere que la fila este seleccionada, en el caso de borrar aparecerá un dialogo de confirmación para que el usuario decida si quiere borrar esa fila.

El filtrar por id, permitirá mostrar solo una fila con los datos de una sola fila.

<div> Agregar empresa Modificar empresa Buscar empresa Eliminar empresa Mostrar Todos </div>						
ID	NOMBRE	DIRECCION	CP	PAIS	EMAIL	TELEFONO
3	Lenovo	1009 Think Place Morri...	27560	USA	info@lenovo.com	919-294-4800

### Resultado Buscar Empresa.



### Buscar Empresa.

Por último, existe la ventana principal que nos permite cambiar entre las diferentes tablas que se quieran manipular.



Este es el menú desplegable que aparecerá, permitiendo cambiar entre las diferentes tablas del proyecto.

Respecto al código, se ha creado una interfaz con el paquete swing de java.

Cada botón de la interfaz gráfica corresponde a un panel, que se abrirá en segundo plano, que permitirá la acción sql, respecto al botón de mostrar todos, simplemente redibujara toda la tabla.

```
JButton button5 = new JButton(text: "Mostrar Todos");
darkone0112, hace 7 horas | 1 author (darkone0112)
button5.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        scrollPane.setPreferredSize(new Dimension(width: 980, height: 600));
        add(scrollPane);
        inventarioBean.displayAllInventario(model);
        revalidate();
        repaint();
    }
});
add(button5);
```

```
JButton button2 = new JButton(text: "Modificar Inventario");
darkone0112, hace 7 horas | 1 author (darkone0112)
button2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        int selectedRow = table.getSelectedRow();
        if (selectedRow != -1) {
            inventarioBean.updateInventario(model, selectedRow);
        }
    }
});
add(button2);
```

Para la función de actualizar y borrar, se ha creado un panel que depende, si hay una fila seleccionada

```
JTextField textFieldInventarioId = new JTextField(String.valueOf(model.getValueAt(selectedRow, column: 0)));
JTextField textFieldProducto = new JTextField(String.valueOf(model.getValueAt(selectedRow, column: 1)));
JTextField textFieldTienda = new JTextField(String.valueOf(model.getValueAt(selectedRow, column: 3)));
JTextField textFieldCantidad = new JTextField(String.valueOf(model.getValueAt(selectedRow, column: 4)));
```

```
JPanel panel = new JPanel();
panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
panel.add(new javax.swing.JLabel(text: "Inventario Id:"));
panel.add(textFieldInventarioId);
panel.add(new javax.swing.JLabel(text: "Producto:"));
panel.add(textFieldProducto);
panel.add(new javax.swing.JLabel(text: "Tienda:"));
panel.add(textFieldTienda);
panel.add(new javax.swing.JLabel(text: "Cantidad:"));
panel.add(textFieldCantidad);
```

Este código permite identificar cuales son los datos actuales, se accede al modelo, selecciona la fila actual, y guarda el valor. Después se crea un panel que se le mostrará al usuario con los datos recuperados y texto indicando a que campo corresponde.

```
int result = JOptionPane.showConfirmDialog(parentComponent: null, panel, title: "Actualizar Inventario", JOptionPane.OK_CANCEL_OPTION);
```

Mostrará los botones aceptar o cancelar, si el usuario pulsa aceptar se ejecutará la query adecuada, en este caso la operación es actualizar.

Por último, se cambia el valor de la fila seleccionada en la tabla desplegada

```
model.setValueAt(getId(), selectedRow, column: 0);  
model.setValueAt(getProducto(), selectedRow, column: 1);  
model.setValueAt(getTienda(), selectedRow, column: 2);  
model.setValueAt(getCantidad(), selectedRow, column: 3);
```

darkone0112, hace 8 horas • ProyectoTerminadoTodasLasClasesImplemen