

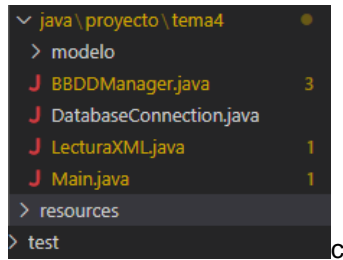


PROYECTO MYSQL - XML ACCESO A DATOS

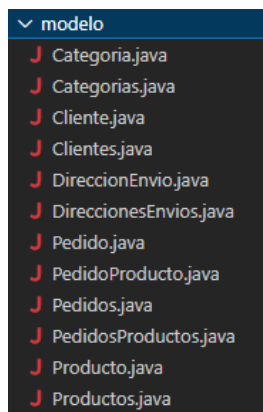
Rubén Díaz Marey

Estructura técnica

El proyecto está estructurado de la siguiente manera:



Modelo: Es el paquete donde irán los diferentes Plain Old Java Object (POJO) de la aplicación, las tablas de la aplicación son: Categorías, Clientes, DireccionesEnvios, Pedidos, PedidosProductos y Productos, todo ello configurado utilizando la anotación JAXB ya que la aplicación es capaz de cargar y guardar los diferentes POJOS.



Clase DatabaseConnection.java: Es la clase responsable de gestionar la conexión a la BBDD relacional utilizando jdbc, tiene los 3 atributos necesarios y un constructor para inicializar la conexión a la base de datos. También tiene el método `getCon()` que devuelve la conexión a la base de datos y permite establecer la conexión. En caso de que salte una excepción el objeto Connection sería nulo y no habría ninguna conexión a la base de datos.

```
package proyecto.tema4;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private String url;
    private String username;
    private String password;

    public DatabaseConnection(String url, String username, String password) {
        this.url = url;
        this.username = username;
        this.password = password;
    }

    public Connection getCon() {
        try {
            return DriverManager.getConnection(url, username, password);
        } catch (SQLException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Clase LecturaXML.java: Es la clase responsable de leer y guardar los documentos XML utilizando JAXB (Java Architecture for XML Binding). Tiene una constante RUTA_XML que es donde se guardarán y cargarán los diferentes ficheros XML a BBDD y viceversa utilizando los diferentes POJOs configurados. Para diferenciar entre los ficheros guardados y cargados se utiliza una nomenclatura con la fecha actual de cada guardado de XML. En caso de que salte una excepción se imprime la traza de excepción.

```
public static Pedidos cargarPedidos() {
    File fichero = new File(RUTA_XML+"pedidos.xml");

    JAXBContext jaxbContext;
    try {

        jaxbContext = JAXBContext.newInstance(...classesToBeBound:Pedidos.class);
        Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();

        if (fichero.exists()) {
            return (Pedidos) jaxbUnmarshaller.unmarshal(fichero);
        }
    } catch (JAXBException e) {
        e.printStackTrace();
    }
    return new Pedidos();
}
```

Clase BBDDManager.java: Es la clase principal responsable de realizar las consultas en la base de datos relacional, contiene diferentes métodos diversos para realizar las diferentes funcionalidades que el programa ofrece, desde listar las tablas, buscar clientes por email, productos por ID, también realiza las operaciones de actualización de stock y de inserción de datos desde el XML. Por último también tiene los métodos requeridos para realizar un pedido. Se utiliza JDBC y la clase DatabaseConnection para gestionar las conexiones de Base de Datos de manera óptima.

```
public void realizarPedido(Cliente cliente, HashMap<Producto, Integer> productosCantidad, DireccionEnvio dirEnvio) {
    try {
        LocalDate fechaActual = LocalDate.now();

        String sqlInsertPedido = "INSERT INTO pedido (EstadoPedido, FechaPedido, Email_Cliente, ID_Direccion) VALUES ('"+""+"Pendiente"+""+"', "+
            " "+fechaActual.toString()+"', '"+cliente.getEmail()+"', "+dirEnvio.getId()+"')";

        Statement sentencia = getCon().createStatement();
        sentencia.executeUpdate(sqlInsertPedido);

        int idPedido = 0;
        ResultSet resultSet = sentencia.executeQuery("SELECT LAST_INSERT_ID()");
        if (resultSet.next()) {
            idPedido = resultSet.getInt(1);
        }

        for (Map.Entry<Producto, Integer> cantidadProductos : productosCantidad.entrySet()) {
            Producto producto = cantidadProductos.getKey();
            int cantidad = cantidadProductos.getValue();

            String sqlUpdateStock = "UPDATE producto SET CantidadEnStock = CantidadEnStock - " + cantidad + " WHERE Codigo = " + producto.getCodigo();
            sentencia.executeUpdate(sqlUpdateStock);

            String sqlInsertDetallePedido = "INSERT INTO detalle_pedido (ID_Pedido, ID_Producto, Cantidad) VALUES (" + idPedido + ", " + producto.getCodigo() + ", " + cantidad +
            sentencia.executeUpdate(sqlInsertDetallePedido);
        }

        sentencia.close();
        getCon().close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Clase Main.java: Es la vista de la aplicación final, la responsable de mostrar el menú y las diferentes opciones de la aplicación que se explicaran en detalle más adelante, cada opción llama a un método diferente para realizar cada opción de manera independiente. También se va informando al usuario con mensaes informativos y también se informa de posibles errores que pueda cometer el usuario para evitar posibles excepciones innecesarias a la hora de llamar a los métodos de la clase principal de BBDDManager

```
Run | Debug
public static void main(String[] args) {
    int opcion;
    do {
        System.out.println("Menú:");
        System.out.println("1. Consultar tabla");
        System.out.println("2. Realizar nuevo pedido");
        System.out.println("3. Modificar precio producto");
        System.out.println("4. Cargar XML a BBDD");
        System.out.println("5. Backup de BBDD a XML");
        System.out.println("6. Salir");
        System.out.println();
        System.out.println("Seleccione una opción: ");
        opcion = sc.nextInt();

        switch (opcion) {
            case 1:
                mostrarTabla();
                break;
            case 2:
                realizarPedido();
                break;
            case 3:
                modificarStock();
                break;
            case 4:
                insertarDesdeXML();
                break;
            case 5:
                backupXML();
                break;
            case 6:
                System.out.println("Saliendo del programa.");
                break;
            default:
                System.out.println("Opción no válida.");
        }
    } while (opcion != 6);
}
```

Funcionalidades

Como funcionalidades la aplicación presenta un menú que se explicará en detalle:

```
Menú:
1. Consultar tabla
2. Realizar nuevo pedido
3. Modificar stock producto
4. Cargar XML a BBDD
5. Backup de BBDD a XML
6. Salir

Seleccione una opción:
█
```

Consulta de tablas

Como primera opción la aplicación nos permite realizar una consulta simple para listar todos los resultados de las diferentes tablas de la base de datos como podemos observar a continuación:

Una vez terminado el usuario introducirá -1 y se habría realizado el pedido, también mostrará un desglose de precios del pedido:

```

Productos [listaProductos=[Producto [codigo=1, nombre=Placa base ASUS Z590, precio=199.99, marca=ASUS,
Procesador Intel i9-12900K, precio=499.99, marca=Intel, descripcion=Procesador de 12 núcleos y 24 hilos,
.99, marca=HP, descripcion=PC empresarial con procesador Intel Core i7, cantidadEnStock=20, categoria=Computadoras],
Producto [codigo=5, nombre=Tarjeta grafica RTX 3080, precio=699.99, marca=NVIDIA, descripcion=Tarjeta grafica de 10GB,
con pantalla táctil, cantidadEnStock=15, categoria=Portátiles], Producto [codigo=5, nombre=Tarjeta grafica RTX 3080, precio=699.99,
categoria=Tarjetas gráficas]]]

Introduzca códigos de productos a pedir (-1 para finalizar compra, -2 para mostrar lista)
1
Producto seleccionado: Producto [codigo=1, nombre=Placa base ASUS Z590, precio=199.99, marca=ASUS,
Introduzca cantidad que desea comprar:
    
```

Como podemos observar en la base de datos se han introducido en las tablas respectivas, también se ha actualizado y restado el stock a los productos:

ID	EstadoPedido	FechaPedido	Email_Cliente	ID_Direccion
1	En proceso	2023-09-26	ruben@gmail.com	1
2	Entregado	2023-09-25	alba@outlook.com	2
3	Pendiente	2023-09-24	pedro@gmail.com	3
4	En proceso	2023-09-23	laura@hotmail.es	4
5	Entregado	2023-09-22	miguel@gmail.com	5
10	Pendiente	2024-02-05	alba@outlook.com	2

ID	EstadoPedido	FechaPedido	Email_Cliente	ID_Direccion
1	En proceso	2023-09-26	ruben@gmail.com	1
2	Entregado	2023-09-25	alba@outlook.com	2
3	Pendiente	2023-09-24	pedro@gmail.com	3
4	En proceso	2023-09-23	laura@hotmail.es	4
5	Entregado	2023-09-22	miguel@gmail.com	5
10	Pendiente	2024-02-05	alba@outlook.com	2

Modificar stock de producto

La tercera opción como bien indica nos permite añadir stock a un producto del catálogo a través del código:

```

3
Catálogo:
Productos [listaProductos=[Producto [codigo=1, nombre=Placa base ASUS Z590, precio=199.99, marca=ASUS,
Procesador Intel i9-12900K, precio=499.99, marca=Intel, descripcion=Procesador de 12 núcleos y 24 hilos,
.99, marca=HP, descripcion=PC empresarial con procesador Intel Core i7, cantidadEnStock=20, categoria=Computadoras],
Producto [codigo=5, nombre=Tarjeta grafica RTX 3080, precio=699.99, marca=NVIDIA, descripcion=Tarjeta grafica de 10GB,
con pantalla táctil, cantidadEnStock=15, categoria=Portátiles], Producto [codigo=5, nombre=Tarjeta grafica RTX 3080, precio=699.99,
categoria=Tarjetas gráficas]]]

Introduzca código de producto a modificar el stock:
1
Introduzca cuanto stock desea añadir al producto:
100
    
```

Como podemos observar se han añadido 100 de stock al producto1:

Codigo	Nombre	Precio	Marca	Descripción	CantidadEnStock	Categoría
1	Placa base ASUS Z590	199,99	ASUS	Placa base para Intel 10th y 11th Gen	170	Placas base
2	Procesador Intel i9-12900K	499,99	Intel	Procesador de 12 núcleos y 24 hilos	10	Procesadores
3	PC de escritorio HP EliteDesk 800 G7	899,99	HP	PC empresarial con procesador Intel Core i7	20	Ordenadores
4	Portátil Dell XPS 13	1.299,99	Dell	Portátil ultraligero con pantalla táctil	15	Portátiles
5	Tarjeta gráfica NVIDIA RTX 3080	799,99	NVIDIA	Tarjeta gráfica de alto rendimiento	40	Tarjetas gráficas

Insertar a BBDD desde XML

La penúltima opción nos permite introducir desde un XML diferentes registros a la BBDD, en mi caso te permite utilizar XML de clientes y productos ya que es lo más útil:

Seleccione una opción:

4

Introduzca la tabla que desee insertar desde un XML: clientes, productos

Una vez insertado podemos observar que se han añadido los registros del XML a la base de datos:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<listaProductos>
  <producto>
    <cantidadEnStock>21</cantidadEnStock>
    <categoria>Placas base</categoria>
    <codigo>6</codigo>
    <descripcion>Placa madre Intel® Z590 LGA 1200 ATX con PCIe 4.0 , 14 + 2 etapas de potencia en equipo</descripcion>
    <marca>ASUS ROG</marca>
    <nombre>ASUS ROG STRIX Z590-F GAMING WIFI</nombre>
    <precio>281.08</precio>
  </producto>
  <producto>
    <cantidadEnStock>2</cantidadEnStock>
    <categoria>Placas base</categoria>
    <codigo>7</codigo>
    <descripcion>Las placas base MSI le permiten administrar velocidades y temperaturas para todos los ventiladores de su sistema
    <marca>MSI</marca>
    <nombre>MSI MPG X570S EDGE MAX WIFI</nombre>
    <precio>348.56</precio>
  </producto>
  <producto>
    <cantidadEnStock>10</cantidadEnStock>
    <categoria>Tarjetas gráficas</categoria>
    <codigo>8</codigo>
    <descripcion>RTX. IT'S ON. Disfruta de los mayores éxitos de ventas de hoy como nunca antes con la fidelidad visual del traz
    <marca>EVGA</marca>
    <nombre>EVGA GeForce RTX 3080 Ti FTW3 ULTRA 12GB GDDR6X</nombre>
    <precio>727.65</precio>
  </producto>
  <producto>
    <cantidadEnStock>2</cantidadEnStock>
    <categoria>Tarjetas gráficas</categoria>
    <codigo>9</codigo>
    <descripcion>La ROG Strix GeForce RTX® 4080 le da un significado completamente nuevo a ir con la corriente.</descripcion>
    <marca>ASUS ROG</marca>
    <nombre>ASUS ROG Strix GeForce RTX 4080 OC Edition 16GB GDDR6X DLSS3</nombre>
    <precio>1673.52</precio>
  </producto>
</listaProductos>
```

Codigo	Nombre	Precio	Marca	Descripción	CantidadEnStock	Categoría
1	Placa base ASUS Z590	199,99	ASUS	Placa base para Intel 10th y 11th Gen	170	Placas base
2	Procesador Intel i9-12900K	499,99	Intel	Procesador de 12 núcleos y 24 hilos	10	Procesadores
3	PC de escritorio HP EliteDesk 800 G7	899,99	HP	PC empresarial con procesador Intel Core i7	20	Ordenadores
4	Portátil Dell XPS 13	1.299,99	Dell	Portátil ultraligero con pantalla táctil	15	Portátiles
5	Tarjeta gráfica NVIDIA RTX 3080	799,99	NVIDIA	Tarjeta gráfica de alto rendimiento	40	Tarjetas gráficas
6	ASUS ROG STRIX Z590-F GAMING WIFI	281,08	ASUS ROG	Placa madre Intel® Z590 LGA 1200 ATX con PCIe 4.0 , 1...	21	Placas base
7	MSI MPG X570S EDGE MAX WIFI	348,56	MSI	Las placas base MSI le permiten administrar velocidades ...	2	Placas base
8	EVGA GeForce RTX 3080 Ti FTW3 ULTRA 12GB GDDR6X	727,65	EVGA	RTX. IT'S ON. Disfruta de los mayores éxitos de ventas ...	10	Tarjetas gráficas
9	ASUS ROG Strix GeForce RTX 4080 OC Edition 16GB GDD...	1.673,52	ASUS ROG	La ROG Strix GeForce RTX® 4080 le da un significado co...	2	Tarjetas gráficas

Volcar desde BBDD a XML

La última opción del programa te permite volcar una tabla de la BBDD a un fichero XML, en mi caso las tablas de productos, pedidos y clientes:

Seleccione una opción:

5

Introduzca la tabla que desee guardar en un XML: clientes, productos, pedidos
clientes

Una vez exportado podemos acceder al fichero XML recién creado, para diferenciarlo de otros XML le he puesto un formato característico tal que:
FechaHoraMinutoExported(NombreTabla).xml:

```

202402052357exportedClientes.xml X
src > main > resources > 202402052357exportedClientes.xml
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <listaClientes>
3      <cliente>
4          <apellidos>Ontalba</apellidos>
5          <email>alba@outlook.com</email>
6          <nombre>Alba</nombre>
7          <password>clave456</password>
8          <telefono>987654321</telefono>
9      </cliente>
10     <cliente>
11         <apellidos>López</apellidos>
12         <email>laura@hotmail.es</email>
13         <nombre>Laura</nombre>
14         <password>securepass</password>
15         <telefono>111222333</telefono>
16     </cliente>
17     <cliente>
18         <apellidos>Rodríguez</apellidos>
19         <email>miguel@gmail.com</email>
20         <nombre>Miguel</nombre>
21         <password>secreto</password>
22         <telefono>444333222</telefono>
23     </cliente>
24     <cliente>
25         <apellidos>Sánchez</apellidos>
26         <email>pedro@gmail.com</email>
27         <nombre>Pedro</nombre>
28         <password>mipass</password>
29         <telefono>555555555</telefono>
30     </cliente>
31     <cliente>
32         <apellidos>Díaz</apellidos>
33         <email>ruben@gmail.com</email>
34         <nombre>Ruben</nombre>
35         <password>contraseña123</password>
36         <telefono>123456789</telefono>
37     </cliente>
38 </listaClientes>

```