

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell



PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell



PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

03

A V A N D E D E L P R O Y E C T O F I N A L 3

Uso de recursos Java

04

A V A N D E D E L P R O Y E C T O F I N A L 3

Uso de control de versiones

05

PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT

01

02

03

04

05

06

07

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT



A V A N D E D E L P R O Y E C T O F I N A L 3
Construcción del product final



A V A N D E D E L P R O Y E C T O F I N A L 3
Conclusiones



"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT



A V A N D E D E L P R O Y E C T O F I N A L 3
Conclusiones



A V A N D E D E L P R O Y E C T O F I N A L 3
Recomendaciones



"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell

PREV

NEXT



"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell



PREV

NEXT

"Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho"



Universidad
Tecnológica
del Perú

A V A N D E D E L P R O Y E C T O F I N A L 3

Sistema de Comercio digital para la floristería "El y Ella Detalles" en Java.

Curso: Curso Integrador: Sistemas – Software

Docente: Mg. Ing. Edwin Palomino Iriarte

E Q U I P O D E T R A B A J O N ° 0 1



U22202494

De la Cruz Asencios
Sandro Cadmo



U22234069

Loa Denegri
Joaquin Alfonso



U22229151

Membrillo Gaitán
Jorge Brandy



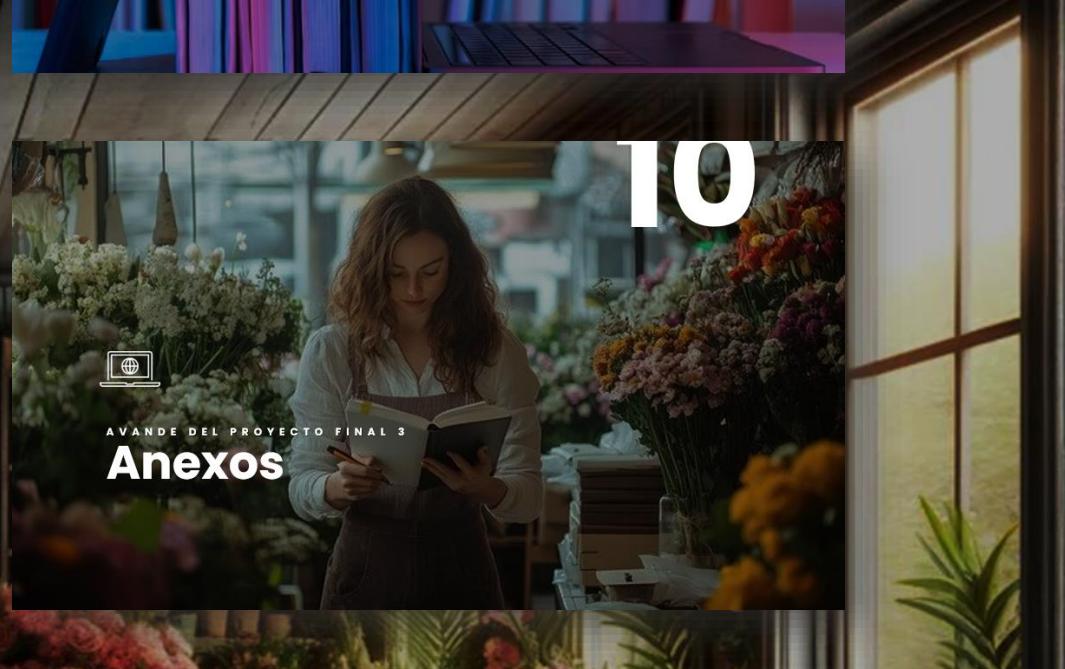
U22220722

Pernia De La Cruz
Jefferson Eudes



U22201636

Santi Saavedra
Kiara Mishell



PREV

NEXT

01



AVANDE DEL PROYECTO FINAL 3

Introducción

02



AVANDE DEL PROYECTO FINAL 3

Diseño de la solución



03



A V A N D E D E L P R O Y E C T O F I N A L 3

Uso de recursos Java

04



AVANZADO DEL PROYECTO FINAL 3

Uso de control de versiones

05



AVANZAR DEL PROYECTO FINAL 3

Implementación de las interfaces gráficas



06



AVANDE DEL PROYECTO FINAL 3

Construcción del product final

07



AVANDE DEL PROYECTO FINAL 3

Conclusiones



08



AVANZANDO DEL PROYECTO FINAL 3

Recomendaciones

09



AVANZANDO DEL PROYECTO FINAL 3

Referencias

10



AVANZANDO DEL PROYECTO FINAL 3

Anexos



A V A N D E D E L P R O Y E C T O F I N A L 3

Despedida



Digitalización para "El y Ella Detalles"

El comercio electrónico es clave para el éxito empresarial global, con ventas proyectadas en 7,4 billones de dólares para 2025. En Perú, el comercio electrónico creció un 50% en 2020, impulsado por la digitalización acelerada. En Lima, la floristería "El y Ella Detalles" busca adaptarse a esta realidad mediante la implementación de un sistema de comercio digital basado en tecnología Java para mejorar sus operaciones y ampliar su alcance.

Este proyecto incluye:

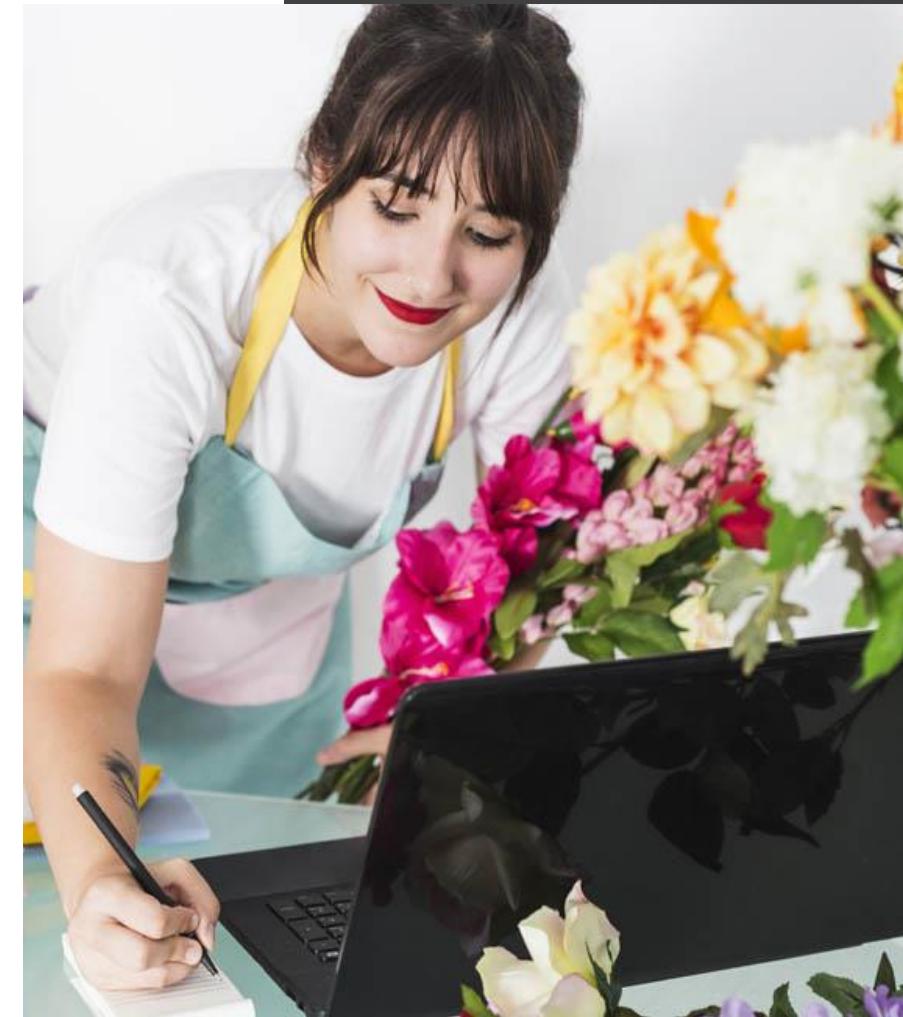
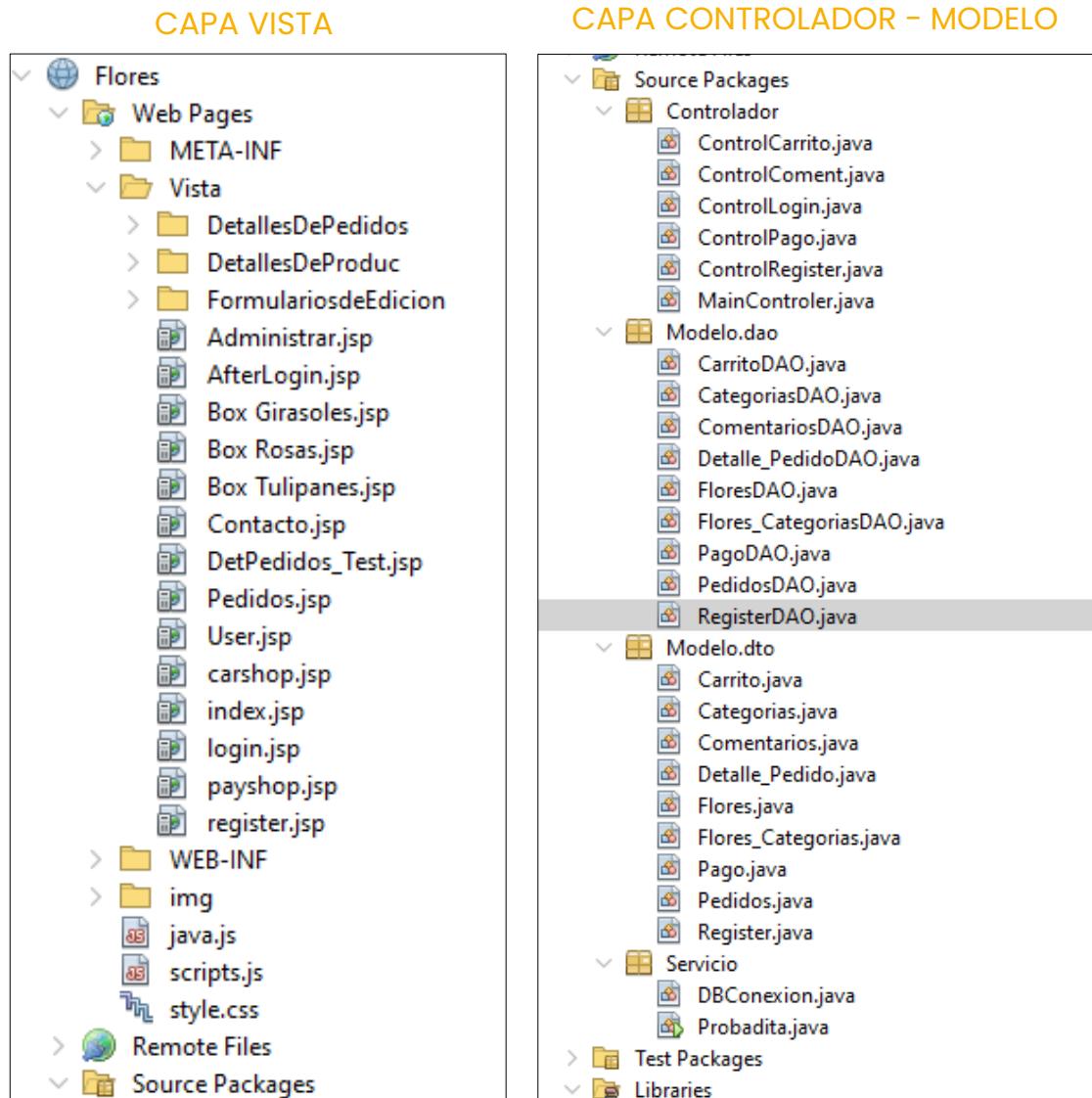
- Análisis del contexto de la floristería y definición del problema.
- Alternativas de solución y diseño de prototipo con pantallas y tecnologías propuestas.
- Desarrollo bajo el patrón MVC, usando JSTL, MySQL Connector, JUnit, y control de versiones en GitHub.
- Avances en arquitectura del sistema, base de datos, y diseño UX/UI.
- Pull Requests y pruebas iniciales para asegurar calidad.

Se espera que este sistema aumente las ventas, mejore la eficiencia y expanda la base de clientes de la floristería, marcando un progreso significativo hacia la digitalización exitosa.





Aplicación del patrón MVC

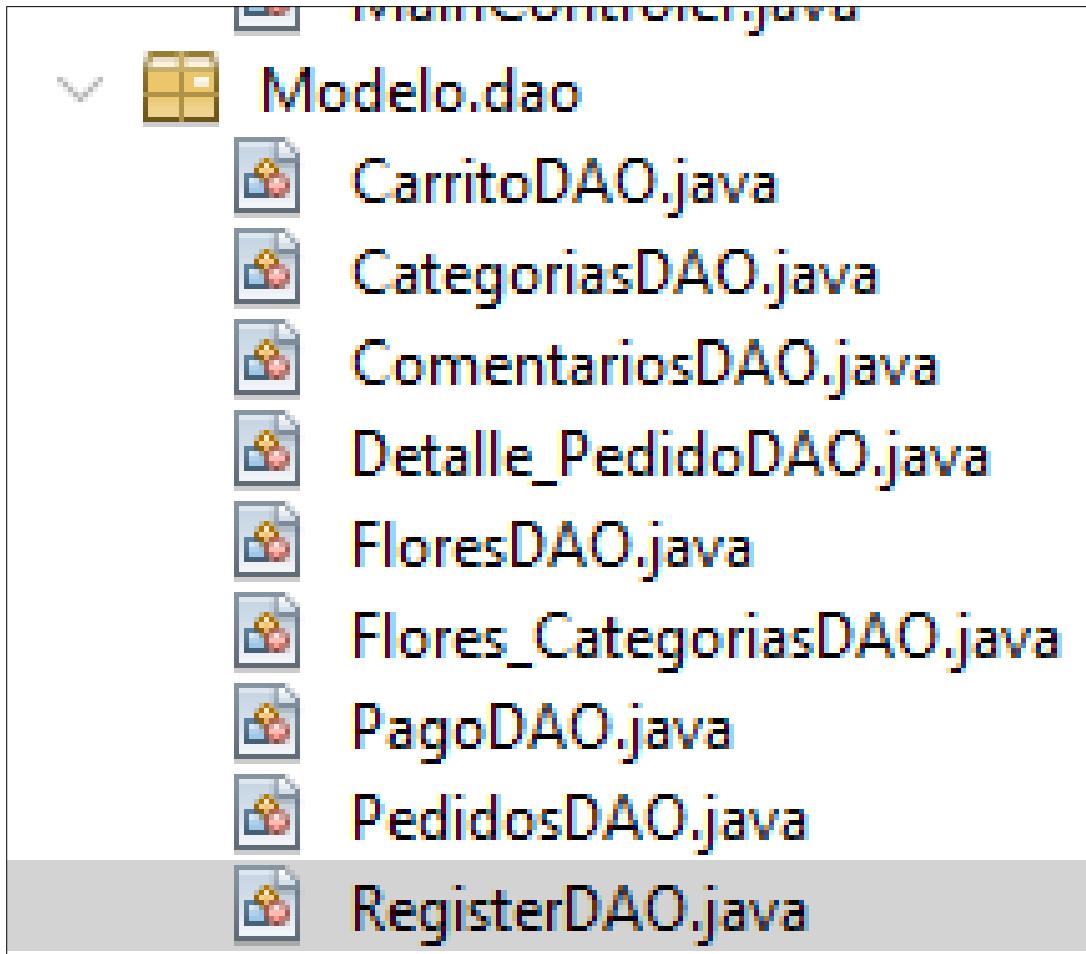


01

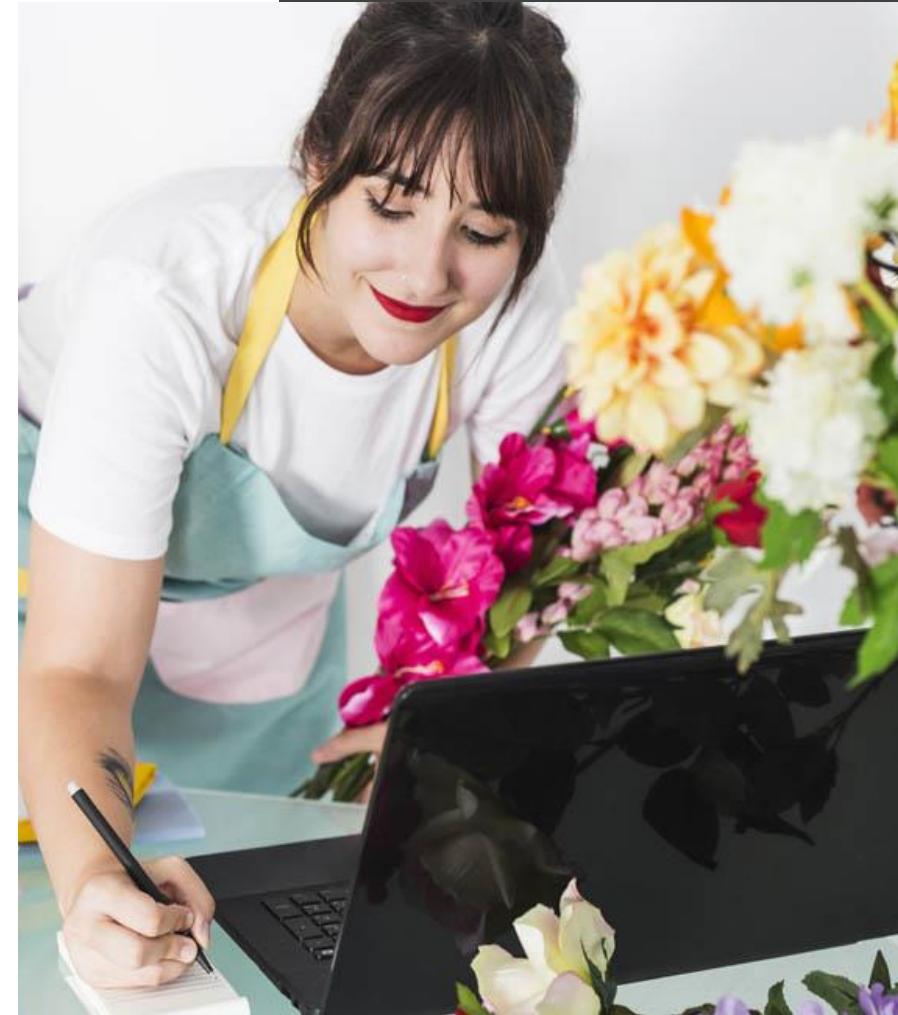




Aplicación del patrón DAO



Para garantizar la integridad y eficiencia en la gestión de los datos del sistema, se utiliza el patrón DAO, el cual permite una interacción directa y segura con las fuentes de datos de la floristería, como productos, clientes y pedidos.



02



Pruebas orientadas a TDD

- ETAPA RED:

```
public class CarritoDAOTest {  
    private CarritoDAO carritoDAO;  
    private Connection mockConnection;  
    private PreparedStatement mockPreparedStatement;  
    private ResultSet mockResultSet;  
  
    private CarritoDAOTest(Connection mockConnection) {  
    }  
  
    @BeforeEach  
    public void setUp(CarritoDAOTest carritoDAOTest) throws Exception {  
        mockConnection = Mockito.mock(Connection.class);  
        mockPreparedStatement = Mockito.mock(PreparedStatement.class);  
        mockResultSet = Mockito.mock	ResultSet.class);  
        carritoDAOTest = new CarritoDAOTest(mockConnection);  
    }  
  
    @Test  
    public void testGetCarritosByUserId() throws Exception {  
        // Configurar el comportamiento del ResultSet simulado  
        Mockito.when(mockConnection.prepareStatement(string: Mockito.anyString())).thenReturn(t: mockPreparedStatement);  
        Mockito.when(mockPreparedStatement.executeQuery()).thenReturn(t: mockResultSet);  
        Mockito.when(mockResultSet.next()).thenReturn(t: true).thenReturn(t: false); // Solo un resultado  
        Mockito.when(mockResultSet.getInt(string: "carrito_id")).thenReturn(t: 1);  
        Mockito.when(mockResultSet.getInt(string: "flor_id")).thenReturn(t: 101);  
        Mockito.when(mockResultSet.getInt(string: "cantidad")).thenReturn(t: 2);  
        Mockito.when(mockResultSet.getDouble(string: "precio")).thenReturn(t: 15.99);  
  
        // Llamar al método a probar  
        List<Carrito> carritos = carritoDAO.getCarritosByUserId(userId: 1);  
  
        // Verificar el resultado  
        assertEquals(actual: carritos, message: "La lista de carritos no debería ser nula");  
        assertEquals(expected: 1, actual: carritos.size(), message: "La lista de carritos debería tener un elemento");  
        assertEquals(expected: 1, actual: carritos.get(index: 0).getIdCarrito(), message: "El ID del carrito no coincide");  
        assertEquals(expected: 101, actual: carritos.get(index: 0).getFlorId(), message: "El ID de la flor no coincide");  
        assertEquals(expected: 2, actual: carritos.get(index: 0).getCantidad(), message: "La cantidad no coincide");  
        assertEquals(expected: 15.99, actual: carritos.get(index: 0).getPrecio(), delta: 0.01, message: "El precio no coincide");  
    }  
}
```

Es la primera etapa del ciclo de TDD, donde se escribe una prueba que inicialmente falla, estableciendo el comportamiento esperado del método o componente antes de implementar cualquier código funcional.



03

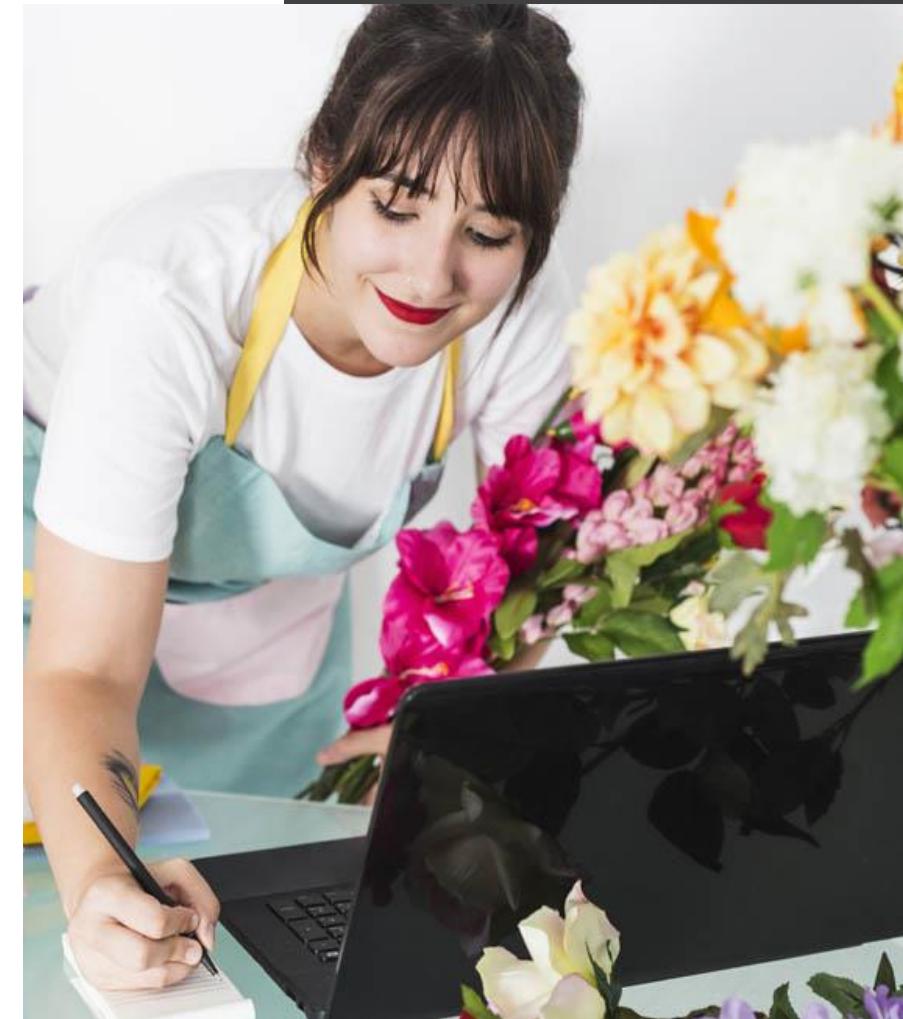


Pruebas orientadas a TDD

- ETAPA GREEN:

```
public class CarritoDAOTest {  
  
    // Declarar la conexión  
    private Connection cnx;  
  
    // Constructor que recibe la conexión  
    public CarritoDAOTest(Connection connection) {  
        this.cnx = connection;  
    }  
  
    public List<Carrito> getCarritosByUserId(int userId) {  
        List<Carrito> carritos = new ArrayList<>();  
        PreparedStatement ps;  
        ResultSet rs;  
  
        String sql = "SELECT ca.carrito_id, ca.cantidad,"  
            + "f.flor_id, f.precio "  
            + "FROM carrito ca "  
            + "INNER JOIN flores f ON ca.flor_id = f.flor_id "  
            + "WHERE ca.user_id = ?";  
  
        try {  
            ps = cnx.prepareStatement(string: sql);  
            ps.setInt(i: 1, ii:userId);  
            rs = ps.executeQuery();  
  
            while (rs.next()) {  
                Carrito carrito = new Carrito();  
                carrito.setIdCarrito(idCarrito: rs.getInt(string: "carrito_id"));  
                carrito.setFlorId(florId: rs.getInt(string: "flor_id"));  
                carrito.setCantidad(cantidad: rs.getInt(string: "cantidad"));  
                carrito.setPrecio(precio: rs.getDouble(string: "precio"));  
  
                carritos.add(e: carrito);  
            }  
            rs.close();  
        } catch (SQLException ex) {  
            ex.printStackTrace();  
        }  
        return carritos;  
    }  
}
```

En esta etapa, se implementa el código mínimo necesario para que la prueba pase, desarrollando solo la funcionalidad suficiente para hacer que la prueba deje de fallar.



04



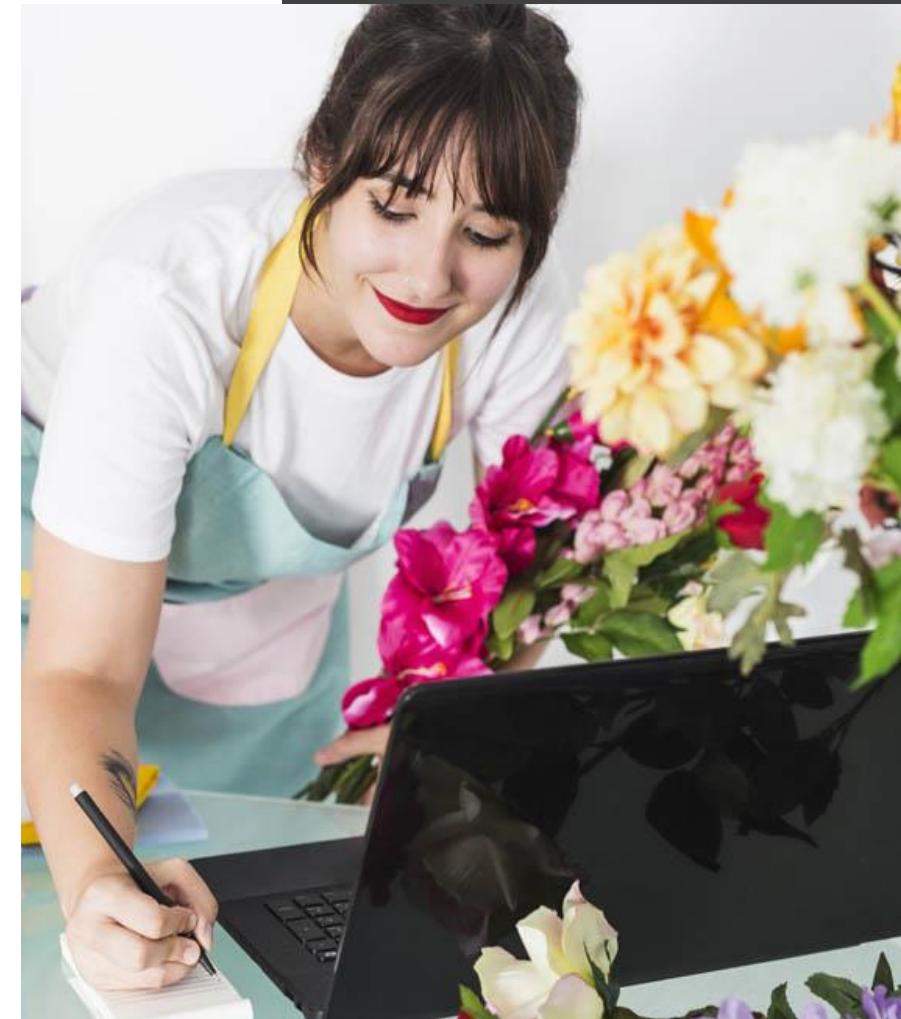


Pruebas orientadas a TDD

- ETAPA REFACTOR:

```
public List<Carrito> getCarritosByUserId(int userId) {  
    List<Carrito> carritos = new ArrayList<>();  
    PreparedStatement ps;  
    ResultSet rs;  
  
    String sql = "SELECT ca.carrito_id, ca.cantidad, "  
        + "f.flor_id, f.precio "  
        + "FROM carrito ca "  
        + "INNER JOIN flores f ON ca.flor_id = f.flor_id "  
        + "WHERE ca.user_id = ?";  
  
    try {  
        ps = cnx.prepareStatement(string: sql);  
        ps.setInt(i: 1, ii: userId);  
        rs = ps.executeQuery();  
  
        while (rs.next()) {  
            Carrito carrito = new Carrito();  
            carrito.setIdCarrito(idCarrito: rs.getInt(string: "carrito_id")); // Actualiza el nombre de la columna aquí  
            carrito.setFlorId(florId: rs.getInt(string: "flor_id"));  
            carrito.setCantidad(cantidad: rs.getInt(string: "cantidad"));  
            carrito.setPrecio(precio: rs.getDouble(string: "precio"));  
  
            carritos.add(e: carrito);  
        }  
        rs.close();  
    } catch (SQLException ex) {  
        System.err.println("Error al obtener los carritos: " + ex.getMessage());  
    }  
    return carritos;  
}
```

La última etapa del ciclo de TDD se centra en mejorar el código sin cambiar su comportamiento externo, limpiando la estructura y optimizando la lógica del código.



05





Uso de JUnit para manejo de errores y validación de métodos

- Implementación de Pruebas en Detalle_PedidoDAO

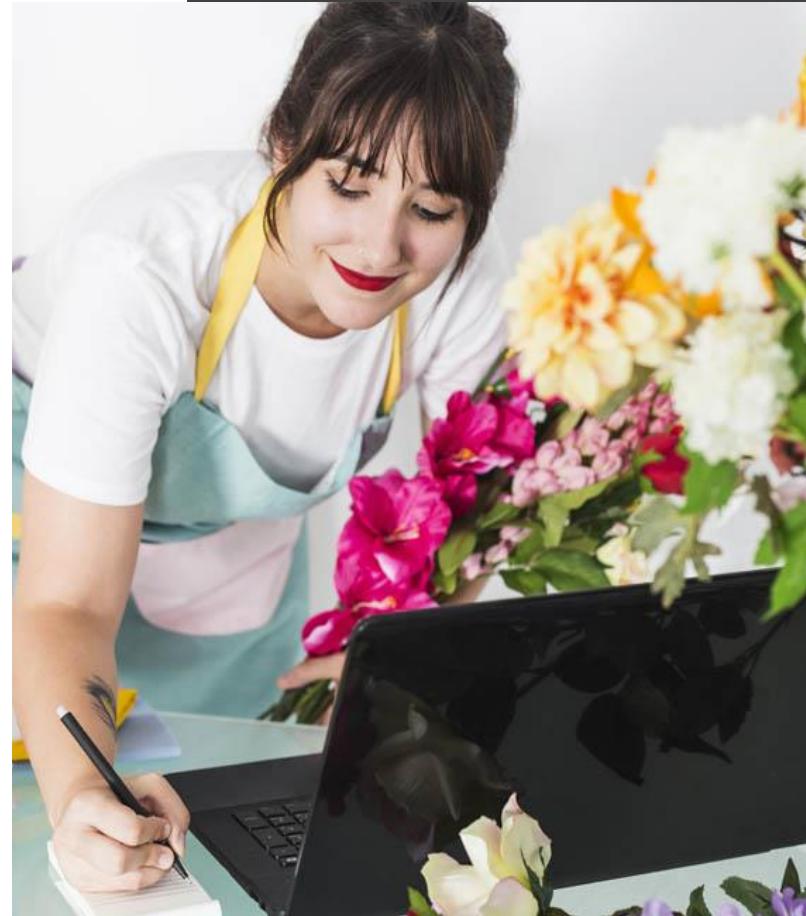
Output - Test (Flores_Test) x
cd C:\Users\Joaquin\Documents\NetBeansProjects\Flores_Test; "JAVA_HOME=C:\\Program Files\\Java\\jdk-20" cmd /c "\"C:\\Program Files\\NetBeans-18\\netbeans\\java\\maven\\bin\\mvn.cmd\" \"-Dmaven.ext.class.path=C:\\Users\\Joaquin\\Documents\\NetBeansProjects\\Flores_Test\\src\\main\\resources\\resources:3.3.0:resources (default-resources) @ Flores_Test ---
Scanning for projects...
Building Flores_Test 1.0-SNAPSHOT
from pom.xml
[jar]
resources:3.3.0:resources (default-resources) @ Flores_Test ---
skip non existing resourceDirectory C:\Users\Joaquin\Documents\NetBeansProjects\Flores_Test\src\main\resources

Test Results x
com.mycompany.Flores_Testjar:1.0-SNAPSHOT (Unit) x
Tests passed: 0,00 %
No test passed, 1 test caused an error. (0,003 s)
Tests.Detalle_PedidoDAOTest Failed
 tests.getDetallePedidosByPedidoId caused an ERROR: Cannot invoke 'java.sql.Connection.prepareStatement(String)' because 'this.cnx' is null
 Cannot invoke 'java.sql.Connection.prepareStatement(String)' because 'this.cnx' is null
 java.lang.NullPointerException
 at Modelo.dao.Detalle_PedidoDAO.getDetallePedidosByPedidoId(Detalle_PedidoDAO.java:73)
 at Tests.Detalle_PedidoDAOTest.testGetDetallePedidosByPedidoId(Detalle_PedidoDAOTest.java:22)
 at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104)
 at java.base/java.lang.reflect.Method.invoke(Method.java:578)
 at org.junit.platform.commons.util.ReflectionUtils.invokeMethod(ReflectionUtils.java:686)
 at org.junit.jupiter.engine.execution.MethodInvocation.proceed(MethodInvocation.java:60)

Error Inicial Detectado

- **Error:** La conexión a la base de datos (Connection) era null en el método getDetallePedidosByPedidoId.
- **Motivo del Error:** La conexión no estaba establecida antes de invocar el método, lo que generaba un NullPointerException al intentar crear un PreparedStatement.
- **Información proporcionada por JUnit:**

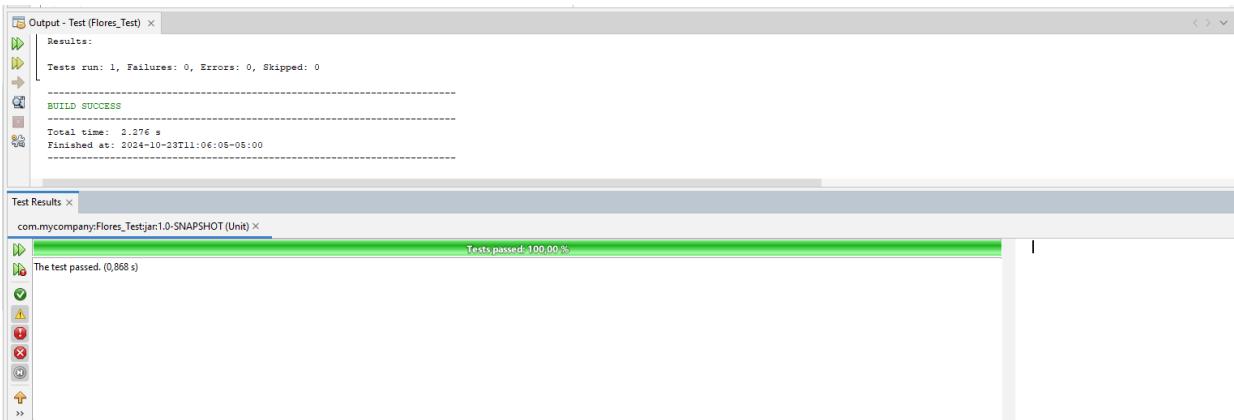
JUnit proporcionó un informe claro del error, señalando la línea específica del código donde se produjo el fallo y mostrando la traza completa del error, lo que facilitó la identificación precisa del problema.





Uso de JUnit para manejo de errores y validación de métodos

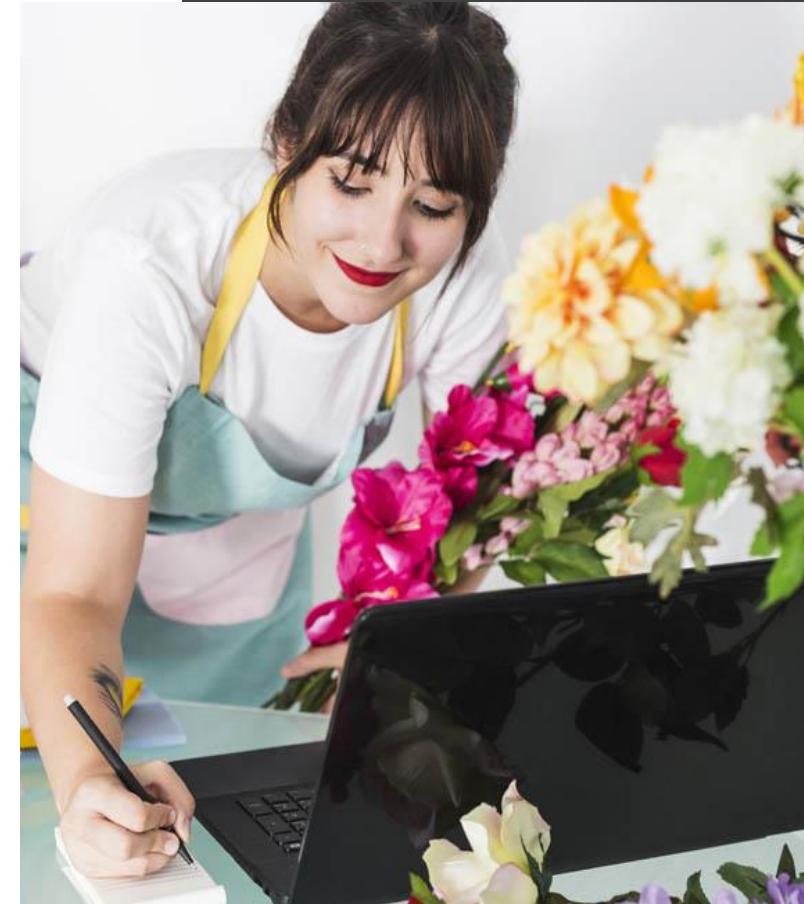
- Implementación de Pruebas en Detalle_PedidoDAO



Resultados Finales de la Prueba

Después de aplicar las correcciones, se volvió a ejecutar la prueba con JUnit, y esta vez el resultado fue exitoso:

- **Pruebas Pasadas:** JUnit mostró que todas las pruebas unitarias se ejecutaron correctamente, logrando una tasa de éxito del 100%.
- **Validación Completa:** La prueba verificó que el método devolviera los resultados esperados y que la conexión funcionara adecuadamente durante toda la ejecución del método.



07



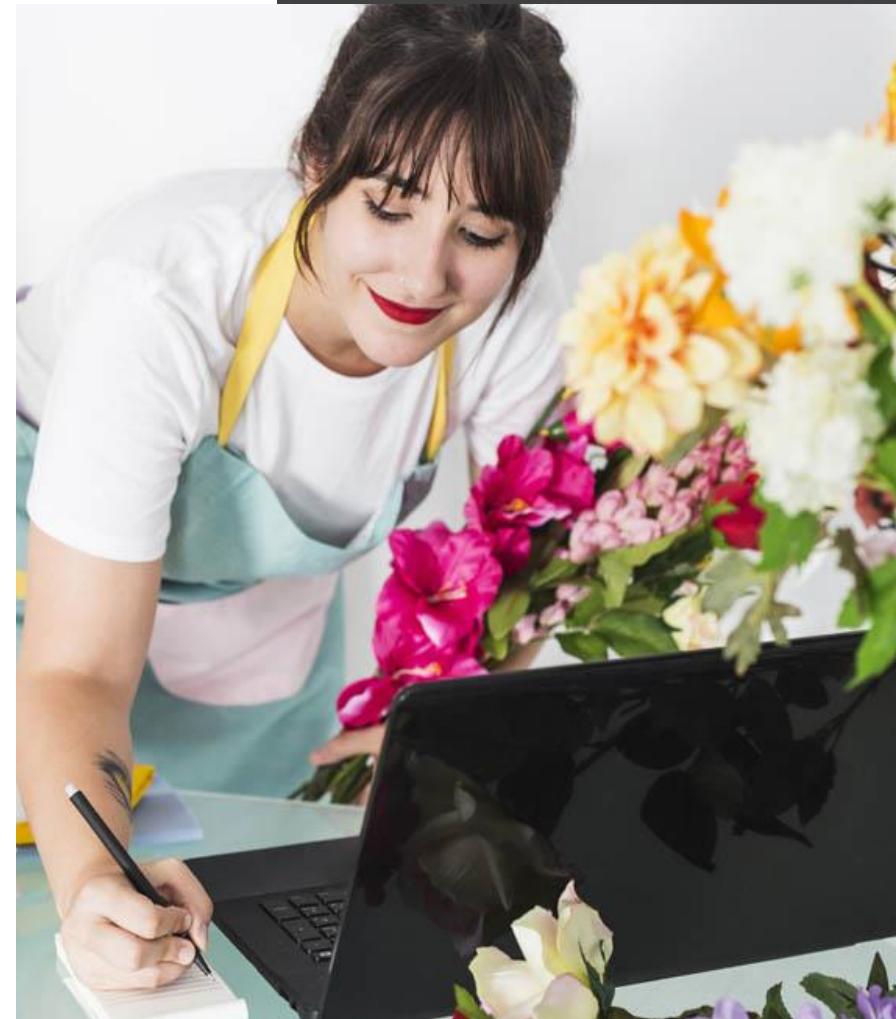


Operaciones CRUD

- Método update() para Actualización de Registros

```
// Método para actualizar una flor existente
public boolean update(Flores f) {
    String cadSQL = "UPDATE flores SET nombre = ?, descripcion = ?, precio = ?, stock = ?, imagen_url = ? WHERE flor_id = ?";
    try (PreparedStatement ps = cnx.prepareStatement(string: cadSQL)) {
        ps.setString(i: 1, string: f.getNombre());
        ps.setString(i: 2, string: f.getDescripcion());
        ps.setDouble(i: 3, d: f.getPrecio());
        ps.setInt(i: 4, il: f.getStock());
        ps.setString(i: 5, string: f.getImagenUrl());
        ps.setInt(i: 6, il: f.getFlorId());
        ps.executeUpdate();
        return true;
    } catch (SQLException ex) {
        System.out.println("Error al actualizar flor: " + ex.getMessage());
        ex.printStackTrace();
        return false;
    }
}
```

La figura muestra la implementación del método update() para actualizar los datos de una flor en la base de datos. Este método utiliza una consulta SQL preparada para modificar los atributos de una flor específica, como su nombre, descripción, precio, stock e imagen. En caso de error, se captura la excepción y se muestra un mensaje informativo con System.out.println(), además de imprimir la traza del error para facilitar la depuración. El método retorna un valor booleano que indica si la actualización fue exitosa o no.



08



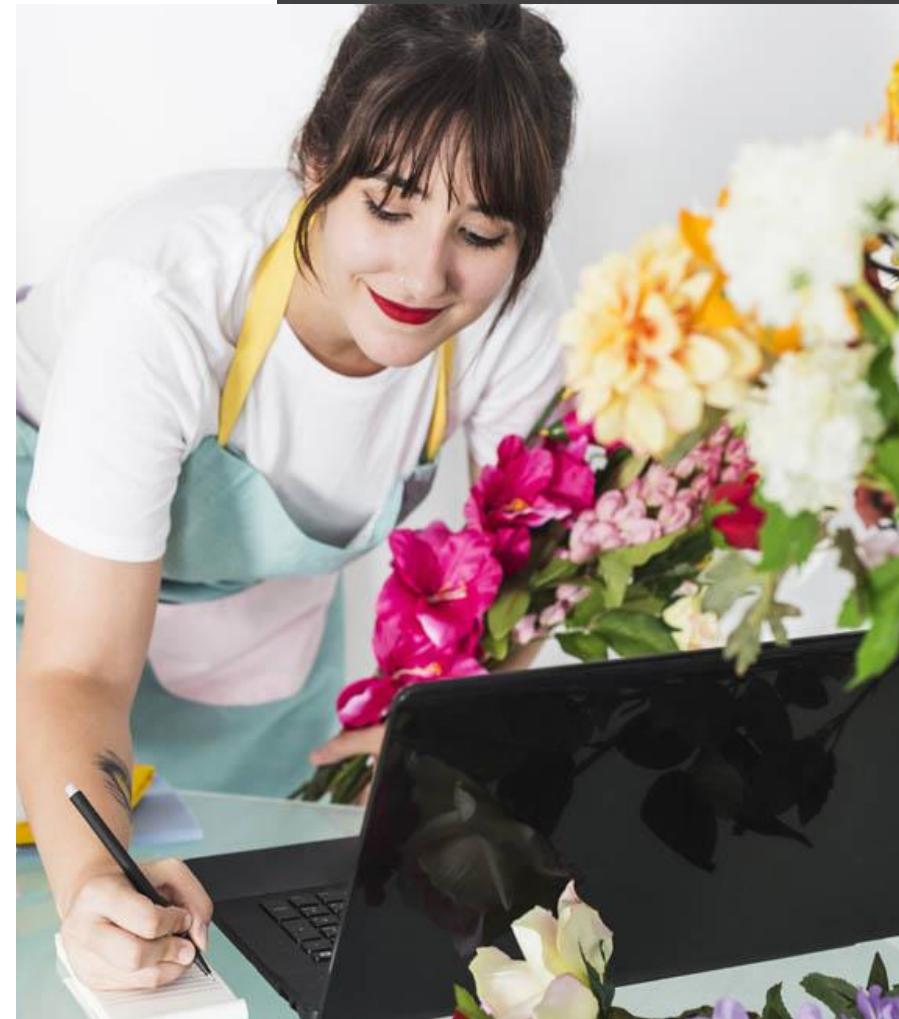


Operaciones CRUD

- Método get() para Recuperación de Registros

```
public Flores get(int id) {  
    Flores f = null;  
    PreparedStatement ps;  
    ResultSet rs;  
    String cadSQL = "SELECT * FROM flores WHERE flor_id = ?";  
    try {  
        ps = cnx.prepareStatement(string: cadSQL);  
        ps.setInt(i: 1, il: id);  
        rs = ps.executeQuery();  
  
        if (rs.next()) {  
            f = new Flores(  
                florId: rs.getInt(string: "flor_id"),  
                nombre: rs.getString(string: "nombre"),  
                descripcion: rs.getString(string: "descripcion"),  
                precio: rs.getDouble(string: "precio"),  
                stock: rs.getInt(string: "stock"),  
                imagenUrl: rs.getString(string: "imagen_url")  
            );  
        }  
        rs.close();  
    } catch (SQLException ex) {  
        ex.printStackTrace();  
    }  
    return f;  
}
```

Este método realiza una consulta SQL preparada y, si encuentra el registro correspondiente, crea un objeto Flores con los valores obtenidos de la base de datos, como su nombre, descripción, precio, stock y URL de la imagen.



09



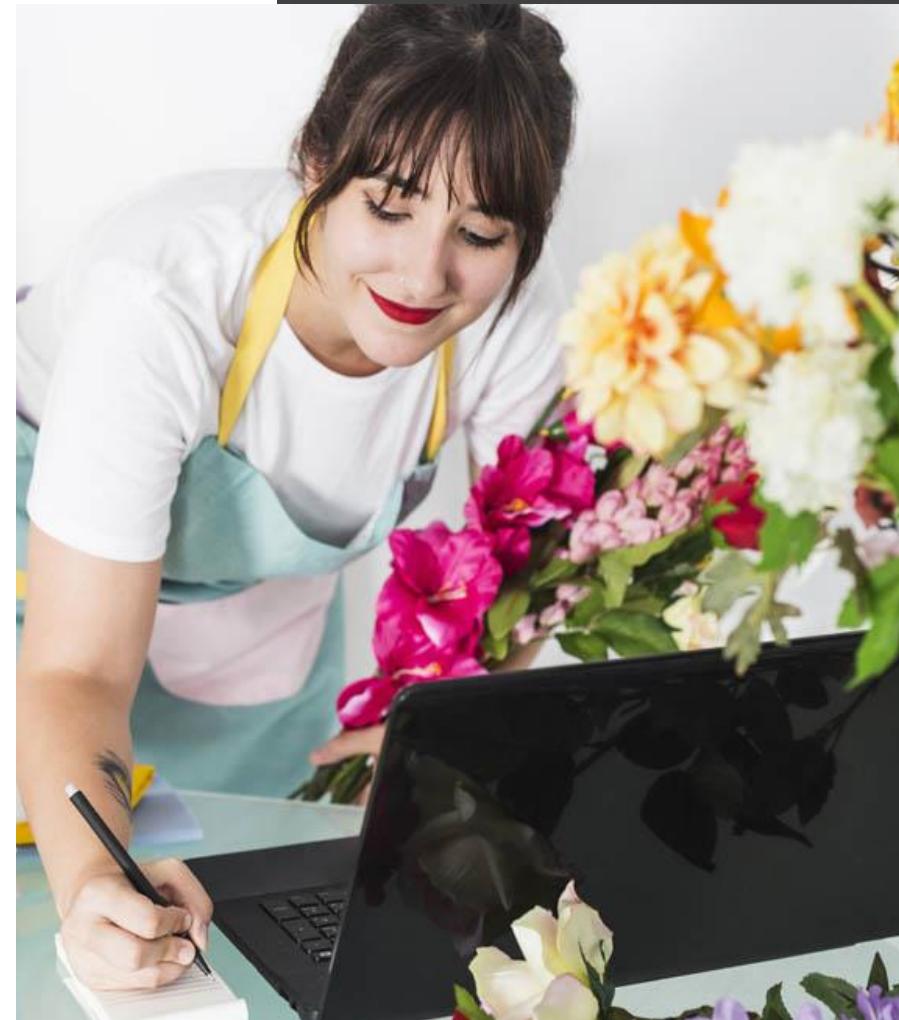


Operaciones CRUD

- Método eliminar() para Eliminación de Registros

```
// Método para eliminar una flor por ID
public boolean eliminar(int id) {
    String cadSQL = "DELETE FROM flores WHERE flor_id = ?";
    try (PreparedStatement ps = cnx.prepareStatement(string: cadSQL)) {
        ps.setInt(i: 1, il: id);
        ps.executeUpdate();
        return true;
    } catch (SQLException ex) {
        System.out.println("Error al eliminar flor: " + ex.getMessage());
        ex.printStackTrace();
        return false;
    }
}
```

La figura muestra la implementación del método eliminar() para borrar una flor de la base de datos mediante su ID. El método utiliza una consulta SQL preparada para asegurar la correcta eliminación del registro. Si la operación es exitosa, retorna true; en caso de error, captura la excepción y muestra un mensaje informativo utilizando System.out.println(), además de imprimir la traza del error para facilitar su depuración. Retorna false si la eliminación no se completa.



10



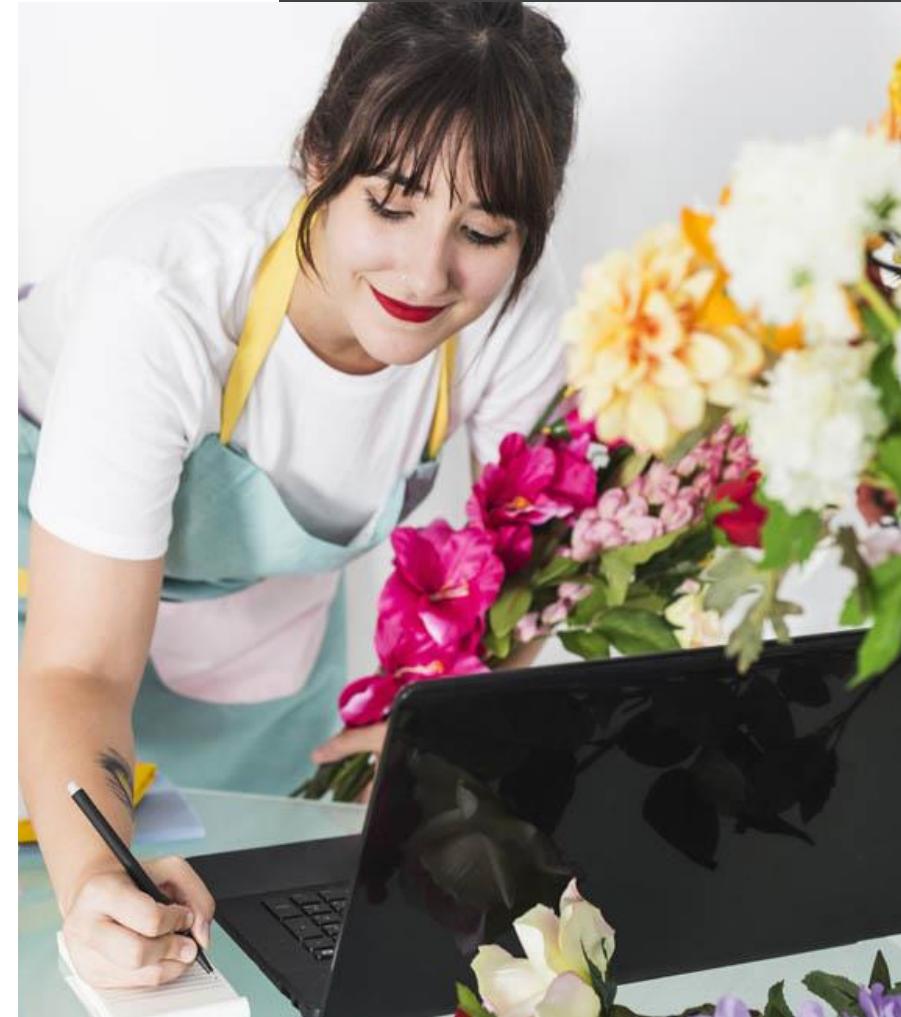


Operaciones CRUD

- Método agregar() para Inserción de Registros

```
public boolean agregar(Flores f) {
    String cadSQL = "INSERT INTO flores (nombre, descripcion, precio, stock, imagen_url) VALUES (?, ?, ?, ?, ?)";
    try (PreparedStatement ps = cnx.prepareStatement(string: cadSQL)) {
        ps.setString(i: 1, string: f.getNombre());
        ps.setString(i: 2, string: f.getDescripcion());
        ps.setDouble(i: 3, d: f.getPrecio());
        ps.setInt(i: 4, i1: f.getStock());
        ps.setString(i: 5, string: f.getImagenUrl());
        ps.executeUpdate();
        return true;
    } catch (SQLException ex) {
        System.out.println("Error al agregar flor: " + ex.getMessage());
        ex.printStackTrace();
        return false;
    }
}
```

La figura muestra la implementación del método agregar() para insertar un nuevo registro de flor en la base de datos. Utiliza una consulta SQL preparada para añadir los datos proporcionados, como nombre, descripción, precio, stock y URL de imagen. Si la inserción se realiza correctamente, el método retorna true; en caso de error, se captura la excepción, se imprime un mensaje informativo con System.out.println() y la traza del error para facilitar la depuración, retornando false si la operación no se completa.





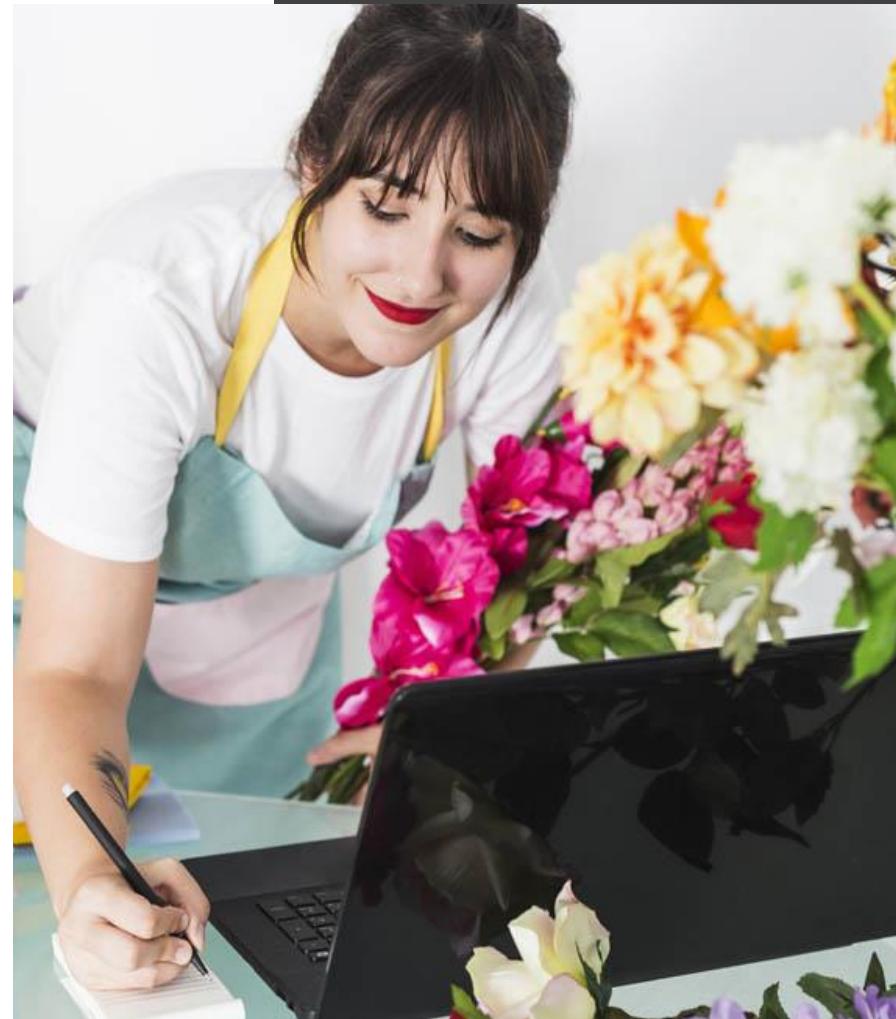
Medidas de seguridad

Para proteger el acceso a la base de datos, se utilizan Prepared Statements, evitando que los datos de entrada se interpreten como parte de la consulta SQL, como se observa en el método save() de RegisterDAO.java:

```
String cadSQL = "INSERT INTO register (nombre, email, contrasena, direccion, telefono) VALUES (?, ?, ?, ?, ?)";
try (PreparedStatement ps = cnx.prepareStatement(cadSQL)) {
    ps.setString(1, r.getNombre());
    ps.setString(2, r.getEmail());
    ps.setString(3, r.getContrasena());
    ps.setString(4, r.getDireccion());
    ps.setString(5, r.getTelefono());
    ps.executeUpdate();
}
```

```
public RegisterDAO() {
    cnx = new DBConexion().getConexion();
}

public boolean save(Register r) {
    String cadSQL = "INSERT INTO register (nombre, email, contrasena, direccion, telefono) VALUES (?, ?, ?, ?, ?)";
    try (PreparedStatement ps = cnx.prepareStatement(string: cadSQL)) { // try-with-resources para cerrar ps automáticamente
        ps.setString(i: 1, string: r.getNombre());
        ps.setString(i: 2, string: r.getEmail());
        ps.setString(i: 3, string: r.getContrasena());
        ps.setString(i: 4, string: r.getDireccion());
        ps.setString(i: 5, string: r.getTelefono());
        ps.executeUpdate();
        return true; // Registro exitoso
    } catch (SQLException ex) {
        System.out.println("Error al registrar usuario: " + ex.getMessage());
        ex.printStackTrace();
        return false;
    }
}
```





Librerías utilizadas en el proyecto



guava-33.2.1-jre.jar

Google Guava es una librería que ofrece utilidades avanzadas para manejar colecciones, cadenas, concurrencia y más.



poi-5.2.3.jar

Apache POI es una librería para trabajar con documentos de Microsoft Office, especialmente hojas de cálculo de Excel (.xls y .xlsx).



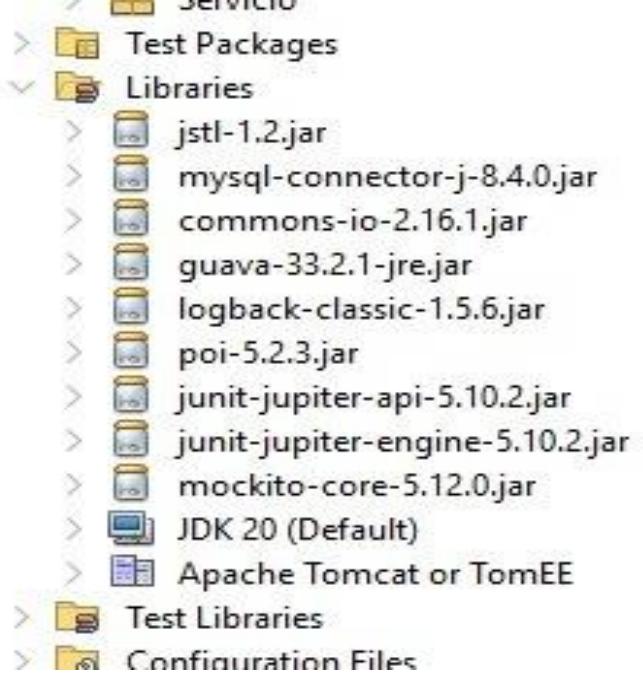
commons-io-2.16.1.jar

Apache Commons IO es una librería de utilidades para trabajar con operaciones de entrada/salida (I/O) en Java, proporcionando métodos para manipular y leer archivos, flujos de datos, y sistemas de archivos.



logback-classic-1.5.6.jar

Logback es una librería de logging (registro de eventos) para Java que permite registrar mensajes en la consola, archivos o sistemas de gestión de registros.



Estructura del Repositorio

01



epalominoi / CIUTP2024

Type ⌘ to search | + - ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

Code Issues 2 Pull requests Actions Projects Wiki Security Insights

Files

main + 🔎

Go to file t

Grupa 01 Curso Integrador I Siste... 1 avance APF1 PPT.pdf APF1.pdf El-y-Ella-Detalles-1.11.zip 2 avance 3 avance (en proceso)

CIUTP2024 / Grupo 01 Curso Integrador I Sistemas Software / 1 avance /

LOAD-13 Add files via upload 6b89284 · yesterday History

Name	Last commit message	Last commit date
..	Add files via upload	yesterday
APF1 PPT.pdf	Add files via upload	yesterday
APF1.pdf	Add files via upload	yesterday
El-y-Ella-Detalles-1.11.zip	Add files via upload	yesterday

epalominoi / CIUTP2024

Type ⌘ to search | + - ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

Code Issues 2 Pull requests Actions Projects Wiki Security Insights

Files

main + 🔎

Go to file t

Grupa 01 Curso Integrador I Siste... 1 avance APF1 PPT.pdf APF1.pdf El-y-Ella-Detalles-1.11.zip 2 avance APF2 PPT.pdf APF2.pdf El-y-Ella-Detalles-2.22.rar db_flores 1.11.sql 3 avance (en proceso)

CIUTP2024 / Grupo 01 Curso Integrador I Sistemas Software / 2 avance /

LOAD-13 Add files via upload 5b43114 · 18 minutes ago History

Name	Last commit message	Last commit date
..	Add files via upload	18 minutes ago
APF2 PPT.pdf	Add files via upload	18 minutes ago
APF2.pdf	Add files via upload	18 minutes ago
El-y-Ella-Detalles-2.22.rar	Add files via upload	18 minutes ago
db_flores 1.11.sql	Add files via upload	18 minutes ago

- **1er Avance:** Incluye archivos y documentación inicial del proyecto.
- **2do Avance:** Contiene documentación más detallada y archivos relacionados con la segunda etapa del desarrollo, tales como presentaciones, archivos de base de datos y otros documentos relevantes.
- **3er Avance (en proceso):** Actualmente, esta carpeta está en proceso de actualización con nuevos entregables.



02



Documentación y Trazabilidad de Cambios

The screenshot shows a GitHub repository's commit history. The commits are listed from most recent at the top to oldest at the bottom. Each commit includes the author, the branch it was pushed to, the commit hash, and the timestamp. The messages describe the changes made in each commit.

Commit Message	Author	Branch	Hash	Timestamp
Implementación del paquete servicio con la clase DBConexion y otra cl...	LOAD-13	master	b0db9d8...3746de3	14 seconds ago
Implementación del paquete servicio con la clase DBConexion y otra cl...	LOAD-13	master	0634c79...3746de3	47 seconds ago
Implementación del modelo dto	Borderline06	master	5b521fb...b0db9d8	6 minutes ago
Implementación del modelo dto	Borderline06	Jorge_Membrillo	b0db9d8	6 minutes ago
Implementación del modelo dao	Kiara21052001	master	b1d4538...5b521fb	10 minutes ago
Implementación del modelo dao	Kiara21052001	Kiara_santi	5b521fb	11 minutes ago
Implementación del Controlador	Sandro222	master	0b34c79...b1d4538	19 minutes ago
Implementación del Controlador	Sandro222	Sandro_DelaCruz	b1d4538	20 minutes ago
Implementación de las Interfaces de Usuario Principales y su codigo c...	LOAD-13	master	d5560e8...0b34c79	25 minutes ago
Implementación de las Interfaces de Usuario Principales y su codigo c...	LOAD-13	Joaquin_Loa	4d20f1b...0b34c79	26 minutes ago
*****Detalles de pedidos, productos y edición form*****	iTaichi-bit	master	4d20f1b...d5560e8	41 minutes ago
*****Detalles de pedidos, productos y edición form*****	iTaichi-bit	Jefferson_Pernia	d5560e8	42 minutes ago
Primera implemetacion de una interfaz grafica para el perfil de usuario	LOAD-13	master	2d3a93e...4d20f1b	57 minutes ago
Primera implemetacion de una interfaz grafica para el perfil de usuario	LOAD-13	Joaquin_Loa	4d20f1b	1 hour ago
Iniciando repositorio	Borderline06	master	2d3a93e	1 hour ago

- El historial de commits en GitHub, como se muestra en la imagen, proporciona una visión clara y detallada de cada modificación realizada en el proyecto. Cada commit está acompañado de un mensaje descriptivo, lo que facilita entender qué cambios se han realizado, quién los ha hecho y en qué momento.



03



Evidencia de Uso

División del Trabajo en el Proyecto

The screenshot shows a GitHub commit page for the repository "Borderline06 / El-y-Ella-Detalles". The commit was made by "Jefferson Pernia" 11 hours ago. It has 17 files changed, with 3649 additions and 0 deletions. The main file shown is "Vista/DetallesDePedidos/DetallePedidoRosa.jsp". The commit message is: "@ -0,0 +1,392 @@". The code snippet includes JSP tags like <%>, <@>, and <meta charset="UTF-8">.

```
1 + <%--  
2 + Document : DetallePedidoRosa  
3 + Created on : 23 sept 2024, 13:51:51  
4 + Author : Joaquin  
5 + --%>  
6 +  
7 + <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  
8 + <!DOCTYPE html>  
9 + <html lang="es">  
10 + <head>  
11 + <meta charset="UTF-8">
```

Activar Windows
Ve a Configuración para activar Windows.

- Jefferson Pernía – Implementación de Detalles de Pedidos, Productos y Edición.



04



Evidencia de Uso

División del Trabajo en el Proyecto

The screenshot shows a GitHub commit page for a repository named 'Borderline06 / El-y-Ella-Detalles'. The commit message is 'Implementación del paquete servicio con la clase DBConexion y otra clase para hacer el Test de Conexion'. It was authored and committed by 'Joaquin Loa' 10 hours ago. The commit has 2 files changed (+58 -0 lines) and includes a diff view of the 'Servicio/DBConexion.java' file:

```
@@ -0,0 +1,41 @@
1 + /*
2 + * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 + * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 + */
5 + package Servicio;
6 +
7 + import java.sql.Connection;
8 + import java.sql.DriverManager;
9 + import java.sql.SQLException;
```

A note at the bottom right says 'Activar Windows' and 'Ve a Configuración para activar Windows.'



U S O D E C O N T R O L D E V E R S I O N E S

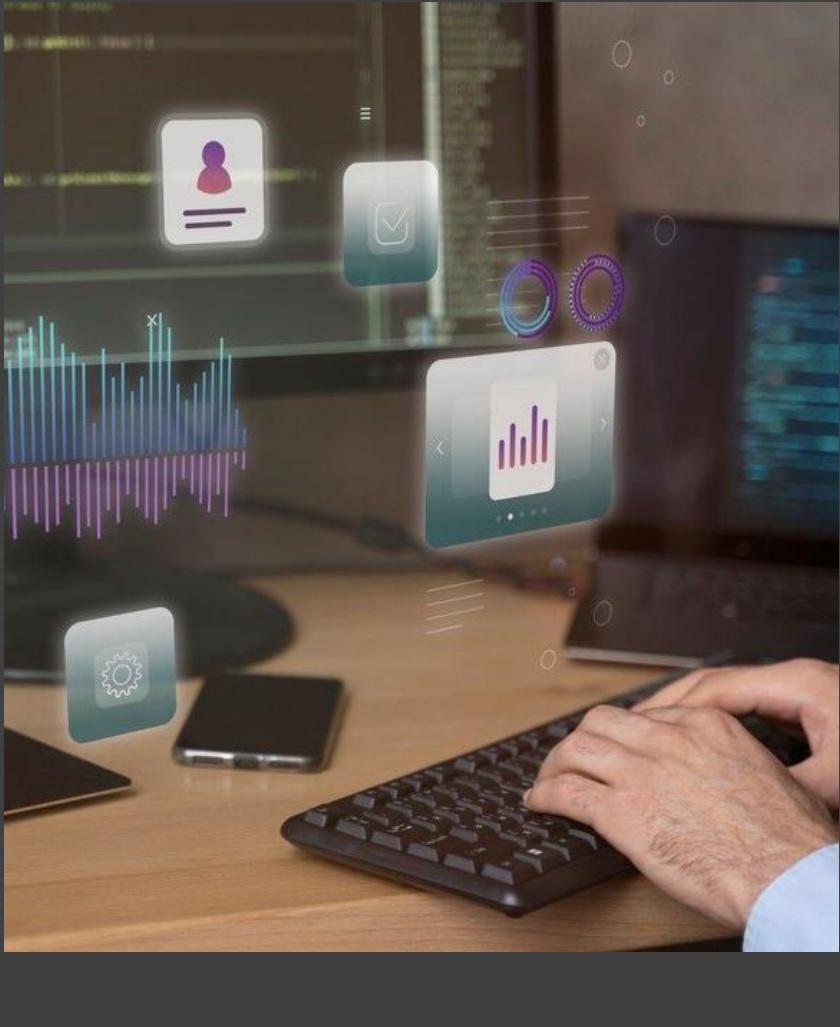
- Joaquín Loa - Implementación del Paquete de Conexión y las Interfaces de Usuario.



Evidencia de Uso

División del Trabajo en el Proyecto

05



Screenshot of a GitHub commit page showing the implementation of a DTO model.

Commit Details:
Borderline06 / El-y-Ella-Detalles
Implementación del modelo dto
Membrillo Gaitan Jorge committed 10 hours ago
1 parent 5b521fb commit b8db9d8

File Changes: 9 files changed +721 -0 lines changed

(dto/Carrito.java)

```
@@ -0,0 +1,70 @@
 1 + /*
 2 + * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 3 + * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 4 + */
 5 + package Modelo.dto;
 6 +
 7 + /**
 8 + * @author Joaquin
 9 + */
10 + */
11 +
```

- Jorge Membrillo – Implementación del Modelo DTO.

06



Evidencia de Uso

División del Trabajo en el Proyecto

The screenshot shows a GitHub commit page for the repository "Borderline06 / El-y-Ella-Detalles". The commit is titled "Implementación del modelo dao" and was made by "Kiara Santti" 10 hours ago. It has 1 parent commit, b1d4538, and a commit hash of 5b521fb. The commit message is "Implementación del modelo dao". The file changed is "dao/CarritoDAO.java" with 10 files changed and +1,106 lines added. The code snippet shows the beginning of the class definition:

```
+ /*  
+ * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.  
+ * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template  
+ */  
+ package Modelo.dao;  
+  
+ import Servicio.DBConexion;  
+ import Modelo.dto.Carrito;  
+ import java.sql.*;  
+ import java.util.ArrayList;
```

At the bottom right, there is a note: "Activar Windows Ve a Configuración para activar Windows."

- Kiara Santti – Implementación del Modelo DAO.





07



Evidencia de Uso

División del Trabajo en el Proyecto

Borderline06 / El-y-Ella-Detalles

Code Issues Pull requests Actions Projects Wiki Security Insights

Implementación del Controlador

Sandro De la Cruz committed 10 hours ago

1 parent 0b34c79 commit b1d4538

master

Filter files...

Controlador

- ControlCarrito.java
- ControlComent.java
- ControlDetallePedido.java
- ControlLogin.java
- ControlPago.java
- ControlRegister.java
- MainController.java

nbproject

project.properties

8 files changed +861 -95 lines changed

Controlador/ControlCarrito.java

```
1  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
3  *
4  */
5  package Controlador;
6  +
7  import Modelo.dao.CarritoDAO;
8  import Modelo.dto.Carrito;
9  import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
```

Activar Windows
Ve a Configuración para activar Windows.

- Sandro De la Cruz – Implementación del Controlador.



01

INTERFACES GRÁFICAS

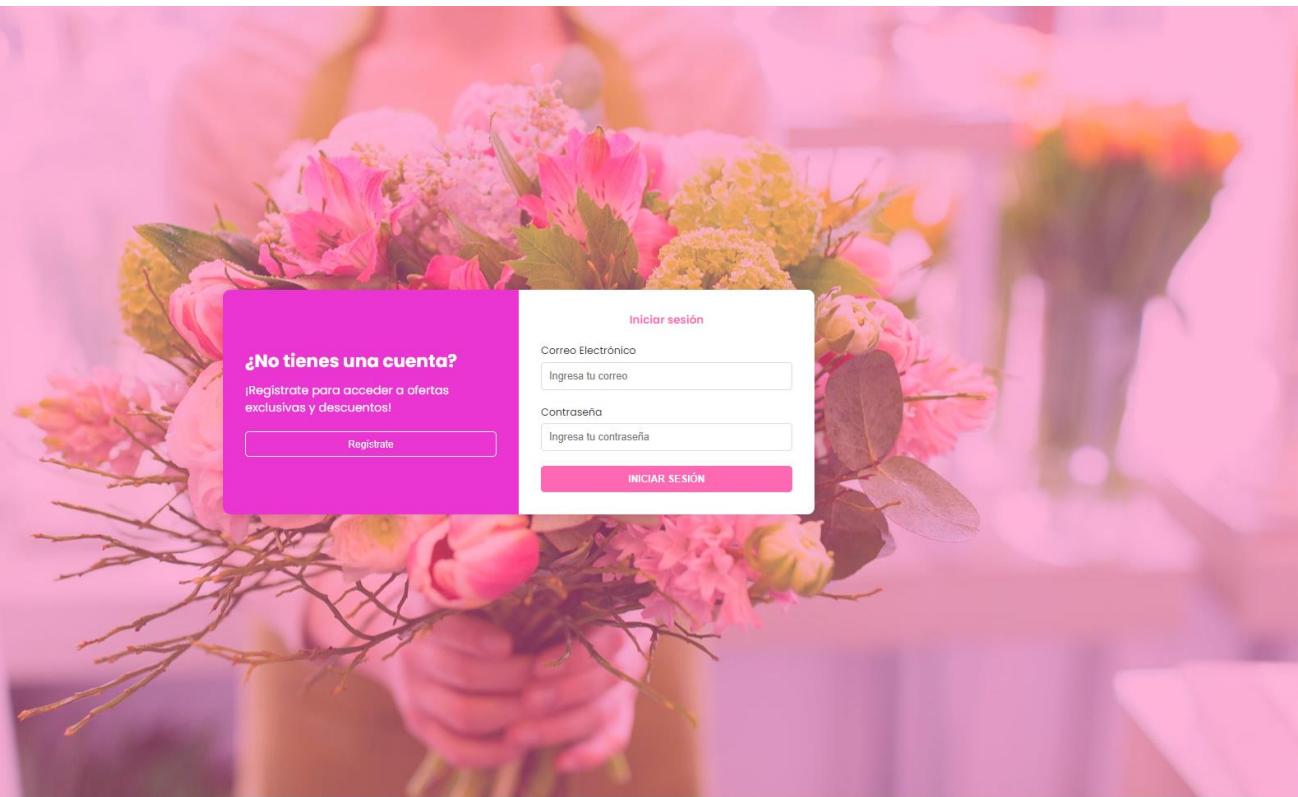
Página de inicio



The screenshot shows a website for "EL Y ELLA DETALLES". The header features a pink navigation bar with links to INICIO, ROSAS, GIRASOLES, TULIPANES, MÁS, and BLOG. On the right, there are user icons for Soporte al cliente (+51 994 802 601), EL Y ELLA DETALLES, and Corriente (0). A search bar is also present. The main content area has a large background image of pink roses and eucalyptus branches. Overlaid on this image is the text "100% Natural Arreglos de Flores" and a "COMPRAR AHORA" button. At the bottom, there are four service icons: Envío rápido, Distintos medios de Pago, Perfecto para todo tipo de regalos, and Servicio al cliente 24/7. Below these icons, the text "Mejores Categorías" is displayed.

La primera vista que el usuario tiene al ingresar a la plataforma es la página de inicio. Esta interfaz presenta una imagen atractiva de bienvenida y permite al usuario acceder de manera rápida a las principales secciones del sitio. Tiene características funcionales específicas como proporcionar una navegación inicial sencilla para el usuario.

Formulario de Inicio de Sesión



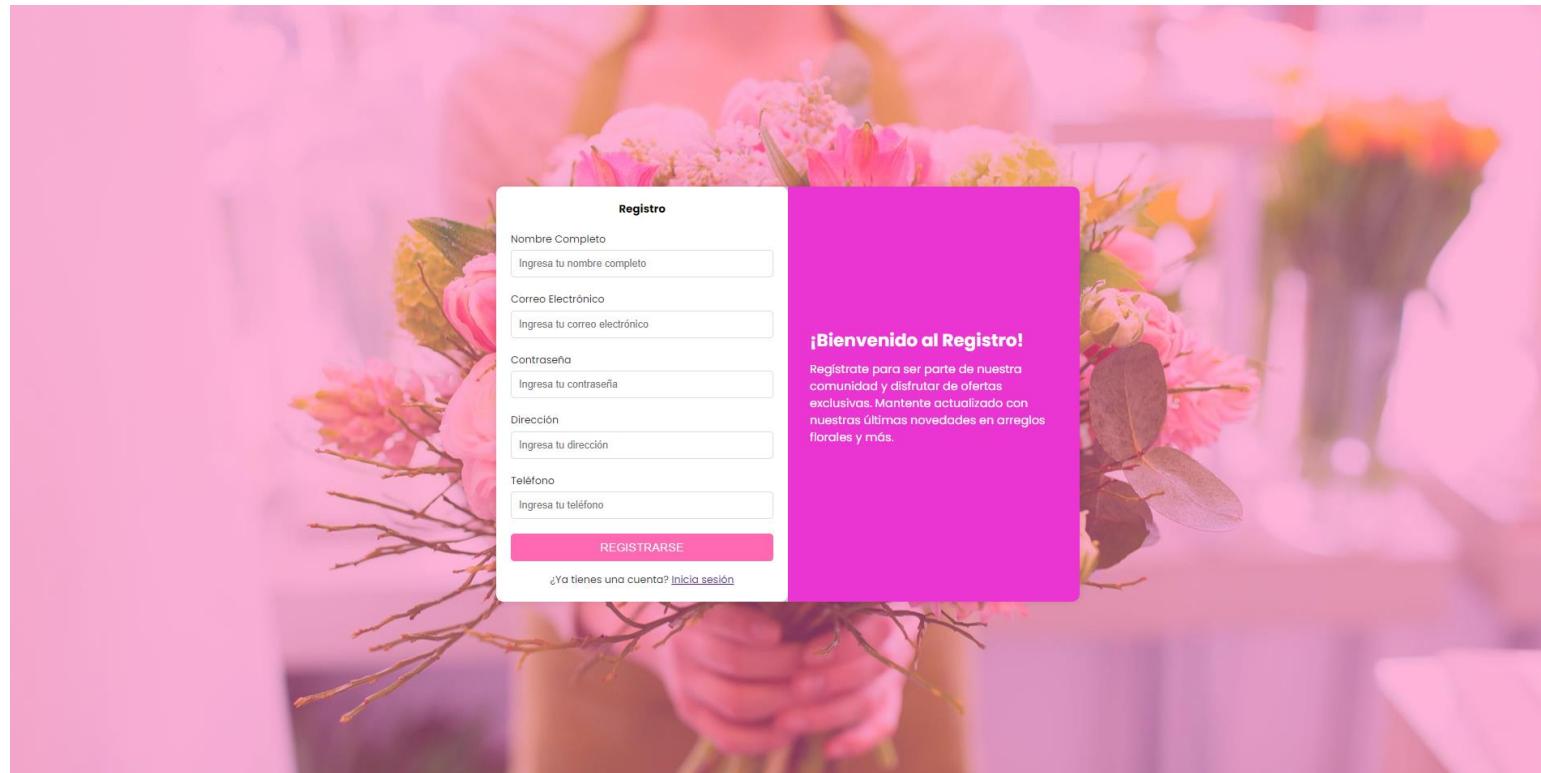
El formulario de inicio de sesión permite a los usuarios autenticarse ingresando su correo electrónico y contraseña. Este formulario realiza una consulta a la base de datos para verificar la validez de las credenciales proporcionadas. Si los datos coinciden con un usuario registrado, se activa una sesión que almacena el user_id del usuario autenticado, permitiendo un acceso seguro y personalizado.



03

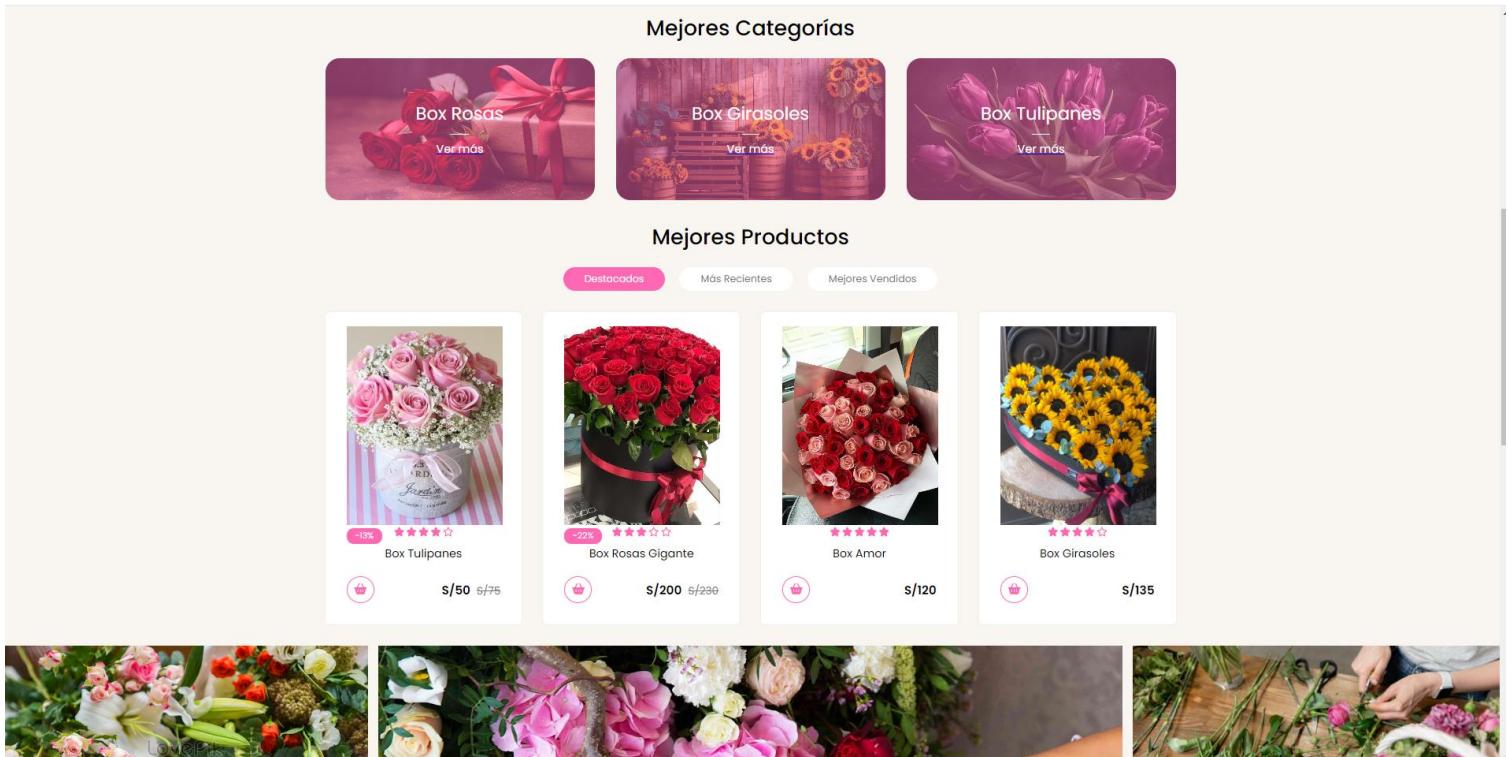
INTERFACES GRÁFICAS

Formulario de Registro



El formulario de registro se utiliza para crear una nueva cuenta de usuario. Los campos solicitados incluyen nombre completo, correo electrónico, contraseña, dirección y teléfono. Al completar el formulario y enviarlo, se registran los datos en la base de datos, creando un nuevo usuario. Esta función permite a los usuarios acceder a ofertas exclusivas y realizar compras en la tienda.

Catálogo de Productos

The screenshot displays a user interface for a flower catalog. At the top, there's a section titled "Mejores Categorías" featuring three flower boxes: "Box Rosas", "Box Girasoles", and "Box Tulipanes", each with a "Ver más" button. Below this is a section titled "Mejores Productos" with three tabs: "Destacados", "Más Recientes", and "Mejores Vendidos". Under "Destacados", four products are shown: "Box Tulipanes" (pink roses), "Box Rosas Gigante" (red roses), "Box Amor" (red and pink roses), and "Box Girasoles" (yellow sunflowers). Each product card includes a discount badge (e.g., "-13%", "-22%"), a star rating, and a price (e.g., S/50, S/200, S/120, S/135). At the bottom of the catalog, there are two images: one showing a variety of flowers and another showing a florist arranging flowers.

El catálogo de productos muestra todos los arreglos florales disponibles para la venta, organizados en categorías como rosas, girasoles y tulipanes. Esta vista permite a los usuarios explorar las diferentes opciones de productos, con información básica como el nombre del producto, precio y una imagen. Cada producto está conectado a la base de datos, lo que permite que se muestren las últimas actualizaciones de stock y precios en tiempo real.





05

INTERFACES GRÁFICAS

Detalle del Producto



The screenshot shows a product detail page for a "Caja de Amor" (Love Box) containing 12 roses. The page includes a large image of the bouquet, a brief description, price (\$120), reviews (18 reviews), delivery information, and a return policy section. At the bottom, there's a section for recommended products.

EL Y ELLA DETALLES

Símbolo de cliente | +51 994 802 801

INICIO ROSAS GIRASOLES TULIPANES MÁS BLOG

Buscar... | Corriente (0)

Código del Producto: AMOR45GOI78

S/120

..... (18 reviews)

Esta hermosa caja especial de San Valentín incluye:
- 12 Rosas rojas premium
- 12 Rosas rosadas de alta calidad
- Follaje delicado con distantes dorados
- Topper con frase romántica como "Te Amo" o "Feliz día".
- Tarjeta personalizada para tu mensaje especial.
Perfecto para una ocasión especial con ese ser amado.

Cantidad: 1 | Añadir al carrito

Información de Entrega
Entregamos en todo el país con opciones de envío exprés por San Valentín. Haz tu pedido antes de las 10:00 a.m. para entregar el mismo día.

Políticas de Devoluciones
Ofrecemos devoluciones dentro de las 48 horas posteriores a la entrega. El producto debe estar en condiciones originales. Contáctanos para más detalles.

Productos Recomendados

Al hacer clic en un producto en el catálogo, el usuario es redirigido a una página de detalles del producto. En esta sección se muestra información detallada sobre el producto seleccionado, incluyendo descripción, precio, imágenes adicionales y políticas de entrega y devolución. Un elemento clave de esta interfaz es el botón "Añadir al carrito", que inserta el `product_id` y el `user_id` en la tabla del carrito en la base de datos, registrando así la selección del usuario para futuras compras.

Formulario de Reseñas



Deja tu comentario y reseña

Comentario:
Escribe tu comentario...

Calificación:
★★★★★

Enviar Comentario

INFORMACIÓN DE CONTACTO

Dirección: Urbanización Edwin Vasquez Carabayllo Enlace
Teléfono: +51 949 234 615
Fax: 55555300
Email: elyelladetalles@gmail.com

INFORMACIÓN

Acerca de Nosotros
Información Delivery
Políticas de Privacidad
Términos y condiciones
Contactáenos

MI CUENTA

Mi cuenta
Historial de ordenes
Lista de deseos
Reembolsos

REGISTRATE

No tienes cuenta? Regístrate ahora fácil y rápido y empieza a disfrutar de los mejores arreglos florales.

REGISTER

Debajo de la sección de detalles del producto se encuentra el formulario de reseñas, donde los usuarios pueden enviar comentarios y calificaciones sobre el producto. El formulario recoge el texto de la reseña, la calificación (de 1 a 5 estrellas), el user_id y el product_id, almacenando toda la información en la base de datos. Esto permite generar un feedback valioso para otros usuarios interesados en el producto.



07

INTERFACES GRÁFICAS

Perfil de Usuario



The screenshot shows a user profile page for 'Joaquin'. At the top, there's a navigation bar with links for 'INICIO', 'ROSAS', 'GIRASOLES', 'TULIPANES', 'MÁS', and 'BLOG'. The main content area has a pink header with the text 'EL Y ELLA DETALLES'. Below this, there's a profile picture placeholder with a heart icon. The profile information section includes the name 'Joaquin', email 'joaquin@gmail.com', and a 'Cerrar Sesión' button. The 'Mis Órdenes' section lists two orders: 'Pedido #1' (Fecha: 2024-10-04, Total: \$/260.0) and 'Pedido #2' (Fecha: 2024-10-07, Total: \$/140.0), each with a 'Ver Detalles' button.

El perfil de usuario muestra la información personal del usuario registrado, utilizando el `user_id` almacenado en la sesión para obtener los datos de la base de datos. Se incluyen detalles como nombre, correo electrónico, dirección y teléfono, además de un historial de pedidos realizados. La interfaz permite a los usuarios editar su información y ver el estado de sus pedidos.

Carrito de Compras



EL Y ELLA DETALLES

Continuar comprando

Carrito de compras

Tienes 1 productos en tu carrito

Flor ID: 4
1 unidades

Actualizar

S/120.00

Subtotal
Envío
Total

Joaquin

Realizar compra

El carrito de compras muestra los productos que el usuario ha añadido para su compra. La interfaz realiza una consulta a la base de datos para obtener los productos vinculados al user_id del usuario autenticado. Los usuarios pueden modificar la cantidad de productos en el carrito o eliminarlos antes de proceder al pago.

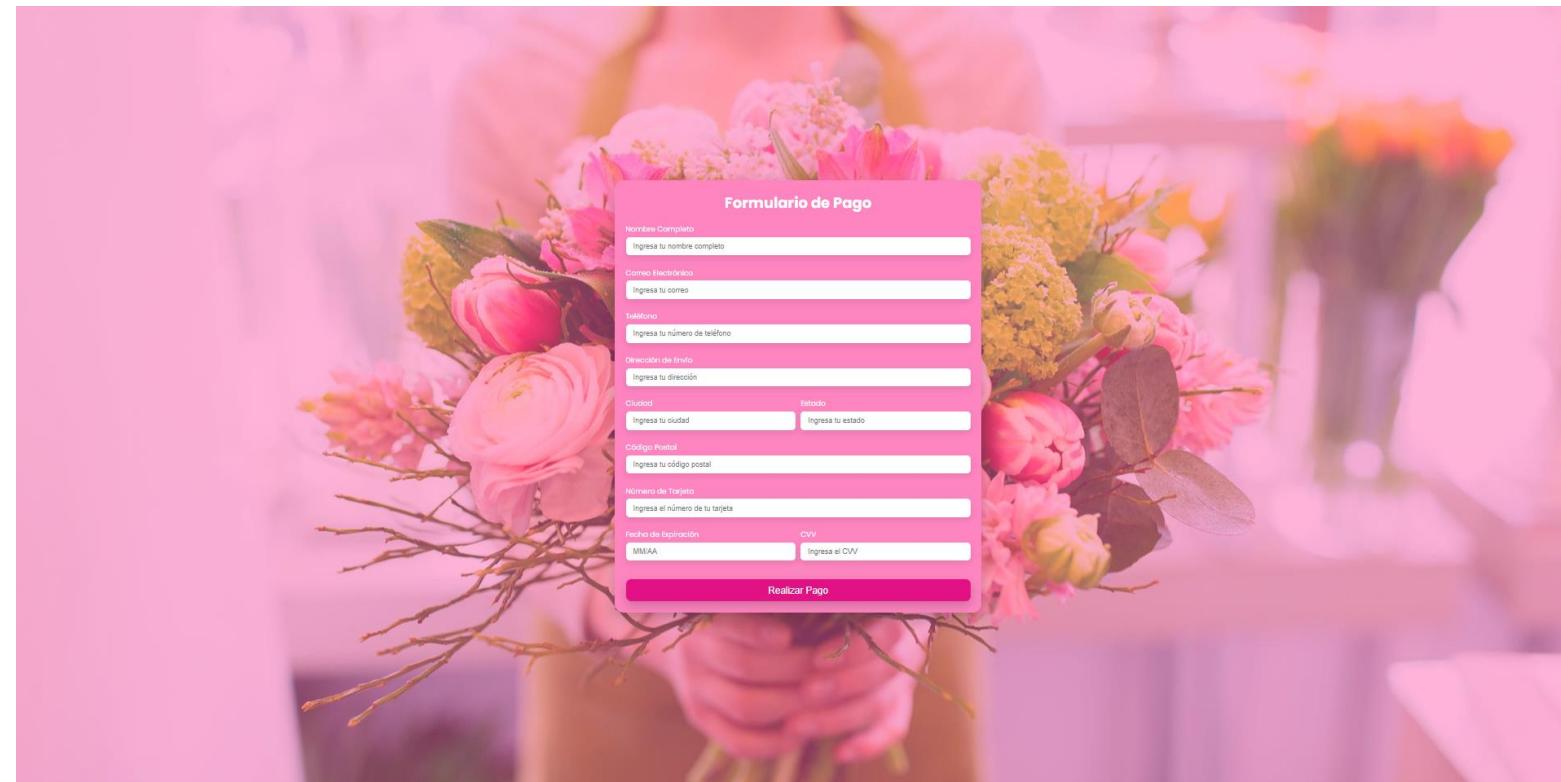




09

INTERFACES GRÁFICAS

Formulario de Pago



El formulario de pago permite a los usuarios ingresar los datos de la tarjeta para procesar la compra. Los datos incluyen nombre del titular, número de tarjeta, fecha de expiración y CVV. Esta información se guarda de manera segura en la base de datos, cumpliendo con las normativas de seguridad requeridas para el manejo de información sensible. Al completar el formulario, se confirma la compra y se emite un recibo digital.



Pedidos



The screenshot shows a web application interface titled "EL Y ELLA DETALLES". At the top, there is a navigation bar with links to "INICIO", "ROSAS", "GIRASOLES", "TULIPANES", "MÁS", and "BLOG". On the right side of the header are icons for "Soporte al cliente" (+51 994 802 601), a user profile, and a shopping cart (0 items). A search bar with the placeholder "Buscar..." and a magnifying glass icon is also present.

The main content area displays a list of orders. Each order item includes a small heart icon, the date it was placed ("Pedido realizado el: 2024-10-04" and "2024-10-07"), a "Descripción del Pedido" section, the total amount ("Total: S/260.0" and "S/140.0"), and two buttons: "Detalles del pedido" and "Borrar".

On the left side of the main content area, there is a sidebar with the following menu items: General, Pedidos, Pago, Reembolsos y devoluciones, Valoraciones, Ajustes, and Dirección de envío.

La vista de pedidos muestra un historial de todos los pedidos realizados por el usuario. Esta interfaz recoge los datos de la base de datos y muestra los pedidos que tienen el mismo user_id que el usuario autenticado, agrupando así los productos del carrito de compras en un solo pedido después de que se ha completado el proceso de pago.





INTERFACES GRÁFICAS

Detalle de Pedido



The screenshot shows a web-based order detail interface. At the top, there's a navigation bar with links for 'INICIO', 'ROSAS', 'GIRASOLES', 'TULIPANES', 'MÁS', and 'BLOG'. On the left, a sidebar titled 'Mi Cuenta' lists options like 'General', 'Pedidos', 'Pago', 'Reembolsos y devoluciones', 'Valoraciones', 'Ajustes', 'Dirección de envío', 'Centro de mensajes', and 'Centro de ayuda'. The main content area is titled 'Detalle del Pedido' and displays the following information:

- Pedido Finalizado**: Pedido terminado. Pedido n° 8190038135507098.
- Pedido efectuado el:** 17 Jun, 2024.
- Método de pago:** Tarjeta de crédito/débito.
- Dirección de Envío**: Nombre: Joaquín Los. Dirección: Calle Los Tulipanes 123, Lima, Perú. Teléfono: +51 949234515.
- Detalles del Pedido**: Una caja llena de rosas rosadas.
- Lista de Productos del Pedido** (Table):

Producto	Cantidad	Precio
Caja de Tulipanes	1	PEN 120.00

On the right side of the main content, there's a small image of a bouquet of pink roses in a white box. At the bottom right of the main content area, there are three buttons: 'Volver a comprar', 'Ver solicitud', and 'Información sobre el reembolso'.

En esta vista, se muestran los siguientes elementos:

- Estado del Pedido: Se indica si el pedido está finalizado o en proceso.
- Fecha y Hora del Pedido: Se muestra la fecha y la hora en que se realizó el pedido.
- Dirección de Envío: Se presentan los detalles de la dirección a la que se enviará el pedido.
- Lista de Productos del Pedido: Se despliega una tabla con los productos incluidos en el pedido, mostrando la cantidad y el precio de cada uno.
- Opciones de Acciones: Hay botones para volver a comprar (redirigiendo al detalle del producto) o rastrear el pedido.

Interfaz Administrativa

**Tablas**

- Usuarios Registrados
- Flores Disponibles
- Carrito de Compras
- Historial de Pedidos
- Detalles de Pedidos
- Categorías
- Relación entre flores y categorías
- Comentarios de usuarios
- Pagos realizados

Administrar Tablas



Selecciona qué tabla quieres administrar



La interfaz administrativa está diseñada para permitir el manejo completo de las tablas que interactúan con el sistema del proyecto. El acceso a esta interfaz está restringido a un usuario con privilegios de administrador, cuyas credenciales están predefinidas en el sistema. Una vez autenticado, el administrador es redirigido automáticamente a esta interfaz, donde puede ver una lista de tablas en una barra lateral para una navegación más sencilla. El administrador puede seleccionar cualquier tabla para ver y gestionar los datos, lo que facilita el mantenimiento y la supervisión del sistema de manera centralizada.





Ejemplo de Administración de Tabla



Tablas

- Usuarios Registrados
- Flores Disponibles
- Carrito de Compras
- Historial de Pedidos
- Detalles de Pedidos
- Categorías
- Relación entre flores y categorías
- Comentarios de usuarios
- Pagos realizados

Administrar Tablas

Buscar...

Usuarios Registrados

ID	Nombre	Email	Dirección	Teléfono	Contraseña	Acciones
1	Joaquin	jioadenegri@gmail.com	mi casa	949234515	12345	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
2	Kiara	kiara_santi@gmail.com	Otro lio	999999222	Kiara	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
3	Raul	raul_escualaya@gmail.com	casa de raul	112345678	raul	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
27	Pedro Castillo	petercastle@gmail.com	carcel	080012200	pedro	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
28	Dina Paucar	DinaBoluarte@gmail.com	Palacio de Gobierno	999222333	Dina	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>



En el ejemplo de la tabla de "Usuarios Registrados", se muestran columnas con el ID, nombre, correo electrónico, dirección, teléfono y contraseña de cada usuario. Además, hay botones de acción para "Editar" o "Eliminar" cada registro, permitiendo al administrador modificar los datos o eliminar usuarios de acuerdo a las necesidades de gestión del sistema.

Esta funcionalidad ofrece un control completo sobre los datos del proyecto, permitiendo realizar ajustes, correcciones o eliminación de registros cuando sea necesario para mantener la integridad y actualización de la información almacenada en la base de datos.

CONSTRUCCIÓN DEL PRODUCTO FINAL



ALCANCE CUBIERTO EN EL AVANCE

Gestión de Inventario

Tablas

- Usuarios Registrados
- Flores Disponibles
- Catálogo de Compras
- Historial de Pedidos
- Detalles de Pedidos
- Categorías
- Relación entre flores y categorías
- Comentarios de usuarios
- Pagos realizados

Administrador Tablas

Buscar... Buscar

Flores Disponibles

Agregar Flor

ID Flor	Nombre	Descripción	Precio	Stock	Imagen URL	Acciones
1	CGirasoles	Caja de Girasoles	135.0	20	/img/Box 14.png	<button>Editar</button> <button>Eliminar</button>
2	CTulipanes	Caja de Tulipanes	50.0	35	Box 4.jpg	<button>Editar</button> <button>Eliminar</button>
3	CRosas	Caja de Rosas	120.0	15	Box 3.jpg	<button>Editar</button> <button>Eliminar</button>
4	CAmor	Caja de dia de San Valentin contiene Rosas de distintos colores	120.0	10	Box 8.jpeg	<button>Editar</button> <button>Eliminar</button>

Relación entre Flores y Categorías

ID Flor	ID Categoría	Acciones
1	1	<button>Editar</button> <button>Eliminar</button>
2	2	<button>Editar</button> <button>Eliminar</button>
3	3	<button>Editar</button> <button>Eliminar</button>
4	3	<button>Editar</button> <button>Eliminar</button>

Este módulo permite registrar, actualizar y consultar las flores disponibles en el sistema. La gestión del inventario es fundamental para mantener un control adecuado del stock, evitando faltantes o excesos de productos. A través de esta funcionalidad, el administrador puede gestionar las flores disponibles en el catálogo de la tienda, asignándoles nombre, descripción, precio, stock e imagen.



01





Carrito de Compras

The screenshot shows the website's header with a support icon and phone number (+51 994 802 801). Below the header is a navigation bar with links to INICIO, ROSAS, GIRASOLES, TULIPANES, MÁS, and BLOG. A search bar and a user account icon are also present. The main content area displays a product card for a flower named 'Joaquin' with a price of \$120.00. A sidebar on the left shows a list of products with a total count of 1. At the bottom, there is a large image of a bouquet of pink roses, overlaid with a payment form titled 'Formulario de Pago'. The payment form includes fields for Nombre Completo, Correo Electrónico, Teléfono, Dirección de Envío, Ciudad, Estado, Código Postal, Número de Tarjeta, Fecha de Expiración, and CVV. A 'Realizar Pago' button is at the bottom of the form.

Este módulo permite a los usuarios seleccionar productos del catálogo y agregarlos al carrito de compras. La funcionalidad del carrito permite modificar las cantidades de los productos seleccionados y proceder al pago, ofreciendo al usuario una experiencia de compra sencilla y eficiente.



02





Historial de pedidos

Administrador de la aplicación.

Administración de tablas:

- Detalles de Pedidos:

ID Detalle	ID Pedido	ID Flor	Cantidad	Precio	Acciones
1	1	4	1	120.0	<button>Editar</button> <button>Eliminar</button>
2	1	4	1	120.0	<button>Editar</button> <button>Eliminar</button>
3	2	4	1	120.0	<button>Editar</button> <button>Eliminar</button>

- Categorías:

ID Categoría	Nombre	Acciones
1	Girasoles	<button>Editar</button> <button>Eliminar</button>
2	Tulipanes	<button>Editar</button> <button>Eliminar</button>
3	Rosas	<button>Editar</button> <button>Eliminar</button>

El historial de pedidos permite tanto a los usuarios como a los administradores consultar los pedidos realizados, junto con información sobre la fecha, estado y monto total. Los usuarios pueden ver un resumen de sus compras anteriores, mientras que los administradores pueden gestionar los pedidos, confirmarlos o cancelarlos según sea necesario.



03





Registro y Gestión de Usuarios

Administrador de Tablas						
Buscar...						
Usuarios Registrados						
ID	Nombres	Email	Dirección	Teléfono	Contraseña	Acciones
1	Joaquin	joadenegr@gmail.com	mi casa	949234515	12345	<button>Editar</button> <button>Eliminar</button>
2	Kara	kara_santi@gmail.com	Otro lco.	999999222	Kara	<button>Editar</button> <button>Eliminar</button>
3	Raul	raul_escutaya@gmail.com	casa de raul	112345678	raul	<button>Editar</button> <button>Eliminar</button>
27	Pedro Castillo	petercastle@gmail.com	cancel	080012200	pedro	<button>Editar</button> <button>Eliminar</button>
28	Dina Pauca	DinaBotuarte@gmail.com	Palacio de Gobierno	999222333	Dina	<button>Editar</button> <button>Eliminar</button>

Se ha implementado un módulo que permite registrar nuevos usuarios en la plataforma, así como gestionar y administrar la información de los clientes. El administrador puede editar o eliminar usuarios registrados, asegurando que la base de datos de usuarios esté siempre actualizada y libre de inconsistencias.



04





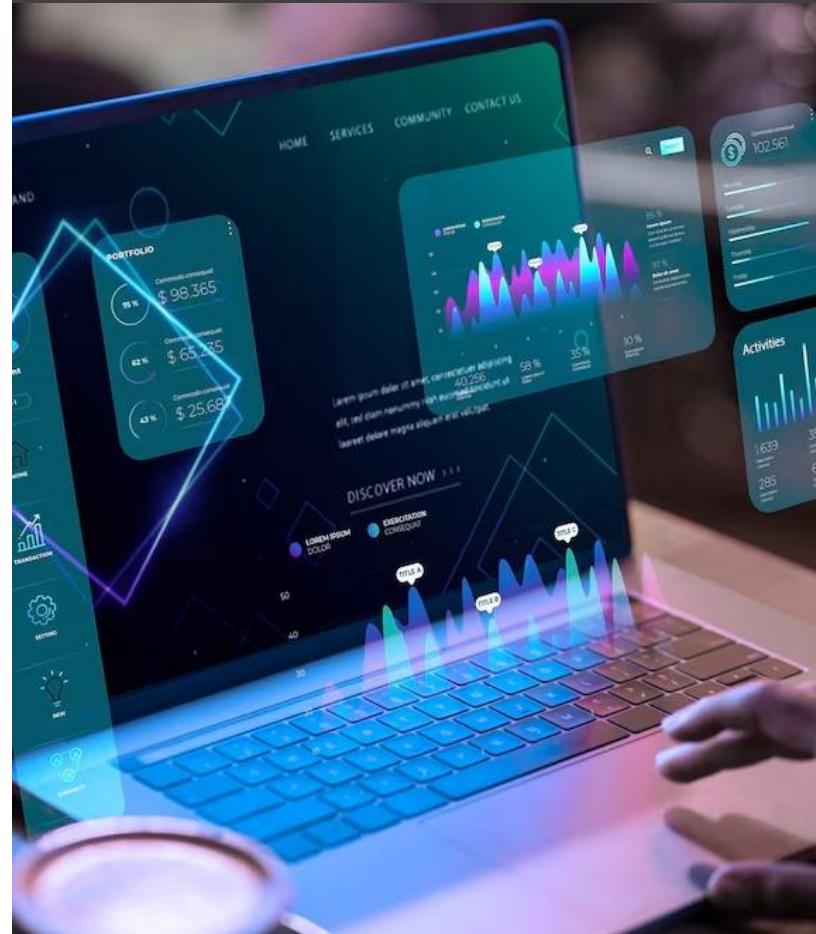
Comentarios de Usuarios sobre Productos

Administrador Tablas

Comentarios de Usuarios

ID Comentario	ID Usuario	ID Flor	Comentario	Calificación	Fecha	Acciones
1	2	1	Muy bueno le gusto mucho a mi tia	5	2024-09-20 13:35:30.0	<button>Editar</button> <button>Eliminar</button>
2	27	1	Muy bonito le gusto a mi novia	5	2024-09-20 18:46:38.0	<button>Editar</button> <button>Eliminar</button>
3	1	1	a	3	2024-09-22 12:37:33.0	<button>Editar</button> <button>Eliminar</button>
4	1	1	God	5	2024-09-22 12:43:32.0	<button>Editar</button> <button>Eliminar</button>
7	28	3	Soy Dina muy Bonito	4	2024-09-22 15:11:14.0	<button>Editar</button> <button>Eliminar</button>
8	2	4	hola	5	2024-09-28 15:54:22.0	<button>Editar</button> <button>Eliminar</button>
9	2	1	buenas	5	2024-09-28 15:55:44.0	<button>Editar</button> <button>Eliminar</button>
10	2	2	adios	3	2024-09-28 15:56:01.0	<button>Editar</button> <button>Eliminar</button>

Los usuarios pueden dejar comentarios y calificaciones sobre los productos adquiridos, lo que permite evaluar la satisfacción del cliente y ofrecer retroalimentación directa a la tienda. Estos comentarios también funcionan como referencias para otros usuarios al momento de realizar sus compras.



05





Pagos Realizados

Tablas

- Usuarios Registrados
- Florés Disponibles
- Camino de Compras
- Historial de Pedidos
- Detalles de Pedidos
- Categorías
- Relación entre flores y categorías
- Comentarios de usuarios
- Pagos realizados

Administrador Tablas

Buscar... Buscar

Pagos Realizados

ID Pago	ID Usuario	Nombre Completo	Correo	Teléfono	Dirección de Envío	Ciudad	Estado	Código Postal	Número de Tarjeta	Fecha de Expiración	CVV	Monto	Fecha Pago	Acciones
1	2	Klara Michell Santi Saavedra	klara_santi@gmail.com	999222353	casa de klara	Lima	S M P	012332	4534534543534543	07/12	324	1940.0	2024-09-28 16:09:04.0	<button>Edit</button> <button>Delete</button>
2	1	Joaquín Alfonso Loa	joadenegril@gmail.com	949234515	mi casa	Lima	Comas	12312312	4524234234234	07/14	432	260.0	2024-10-04 15:34:22.0	<button>Edit</button> <button>Delete</button>
3	1	Joaquín Loa	joadenegril@gmail.com	949234515	Av Felipe Príncipe Alva 308	Lima	Comas	01504	345643543534543	07/12	432	140.0	2024-10-07 15:28:26.0	<button>Edit</button> <button>Delete</button>
4	1	Joaquín Loa	joadenegril@gmail.com	949234515	Av Felipe Príncipe Alva 309	Lima	Comas	01504	345643543534543	07/12	432	140.0	2024-10-07 15:31:53.0	<button>Edit</button> <button>Delete</button>

El sistema registra todos los pagos realizados por los clientes, junto con información relevante como el monto, el método de pago, el número de tarjeta y la fecha de pago. Esta funcionalidad permite al administrador llevar un control preciso de las transacciones efectuadas.



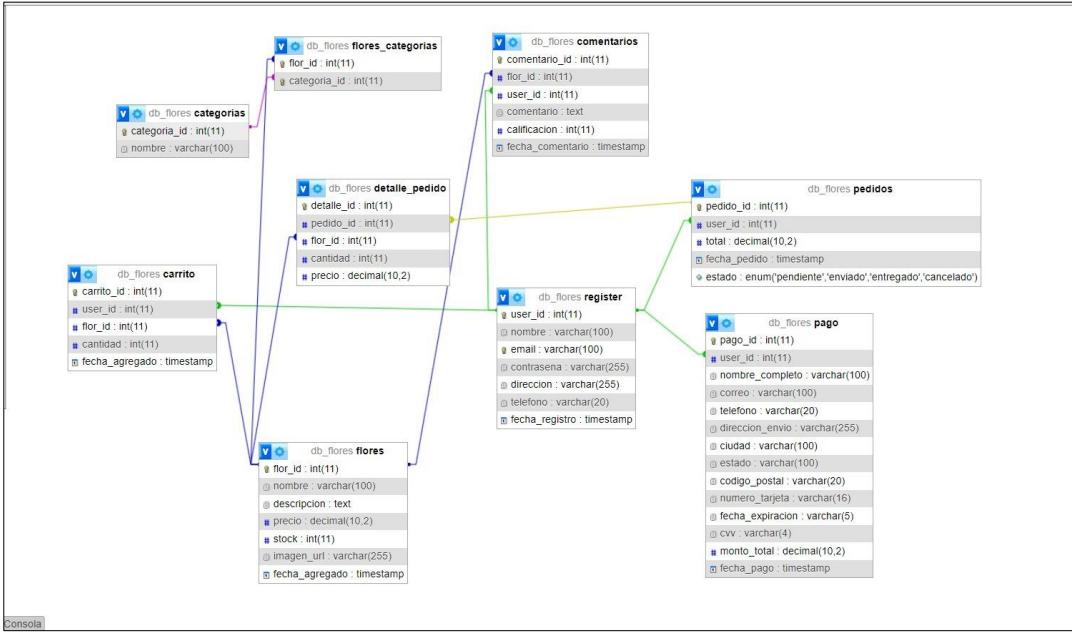
06





Informe Técnico: Decisiones de Diseño del Proyecto

- Justificación del Modelo de Base de Datos



El diseño del modelo de datos se llevó a cabo considerando las necesidades funcionales del sistema y asegurando la correcta relación entre las entidades.





Informe Técnico: Decisiones de Diseño del Proyecto

- Justificación del Modelo de Base de Datos

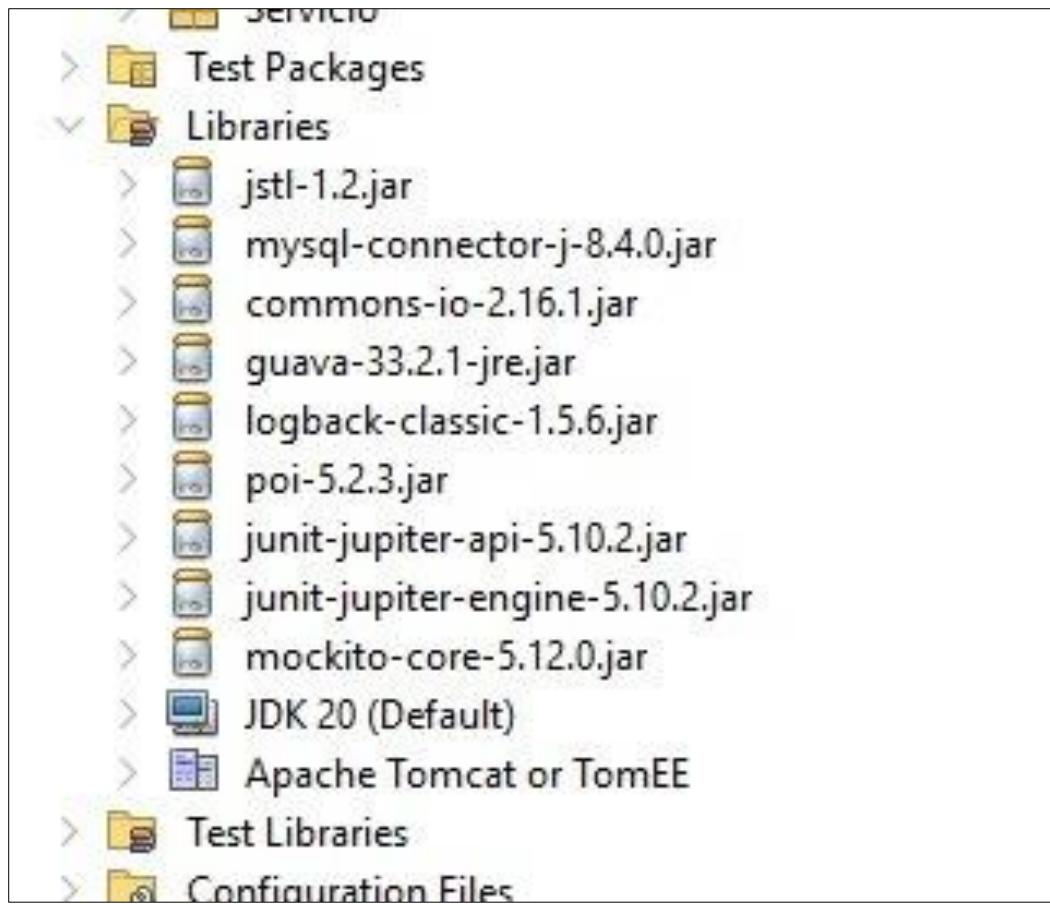
```
4   -- Crear tabla register (Usuarios registrados)
5 • CREATE TABLE register (
6     user_id INT AUTO_INCREMENT PRIMARY KEY,
7     nombre VARCHAR(100) NOT NULL,
8     email VARCHAR(100) UNIQUE NOT NULL,
9     contraseña VARCHAR(255) NOT NULL,
10    direccion VARCHAR(255),
11    telefono VARCHAR(20),
12    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
13 );
```

```
15  -- Crear tabla flores (Productos disponibles)
16 • CREATE TABLE flores (
17     flor_id INT AUTO_INCREMENT PRIMARY KEY,
18     nombre VARCHAR(100) NOT NULL,
19     descripcion TEXT,
20     precio DECIMAL(10, 2) NOT NULL,
21     stock INT NOT NULL,
22     imagen_url VARCHAR(255),
23     fecha_agregado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
24 );
```





Uso de librerías adecuadas



Cada una de estas librerías juega un papel clave en el desarrollo del proyecto. Gracias a su integración, se asegura un rendimiento óptimo, la reducción de errores y una fácil adaptación del sistema a futuras modificaciones o ampliaciones.



09

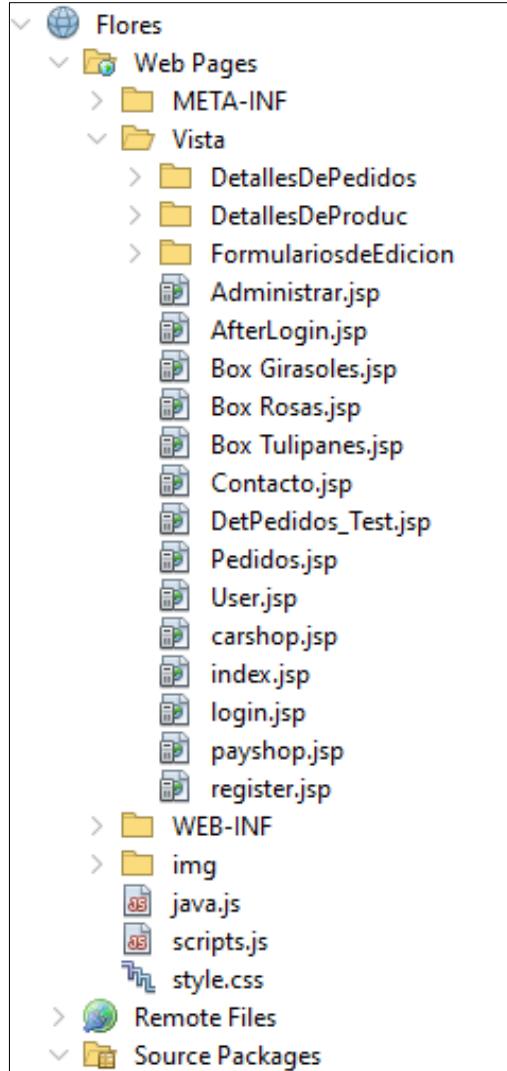




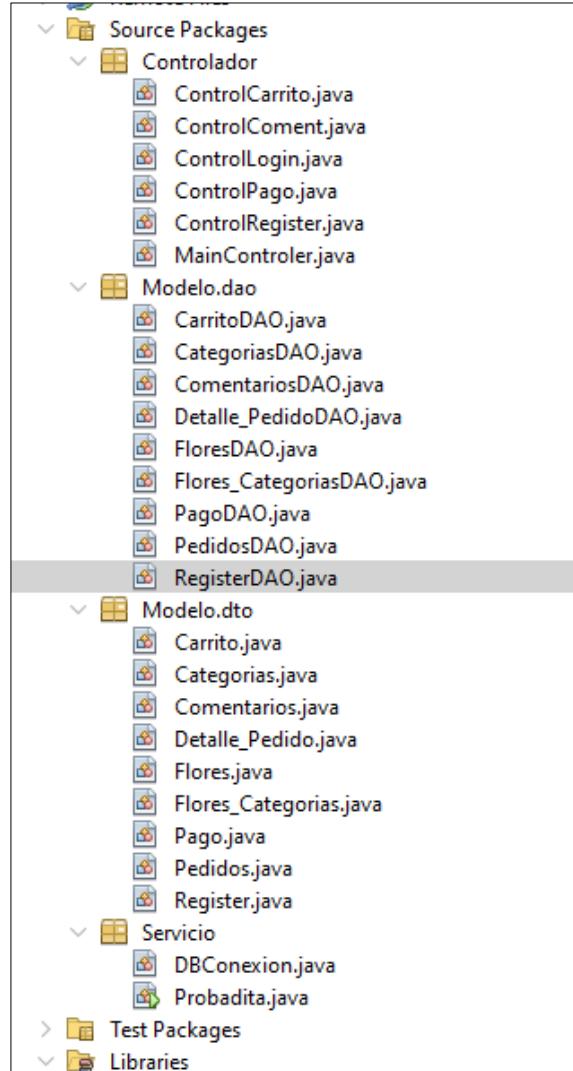
Uso de patrones de diseño

- Patrón MVC

CAPA VISTA



CAPA CONTROLADOR - MODELO



10





Software de control de versiones

Para el desarrollo de este proyecto se utilizó GitHub como software de control de versiones, el cual permitió llevar una gestión eficiente del código fuente y un control riguroso sobre cada etapa del desarrollo. A través de GitHub, fue posible organizar el trabajo en ramas, documentar cambios con mensajes claros en los commits y asegurar una integración continua que minimizara errores en las versiones finales.

The screenshot shows a GitHub repository interface. On the left, there's a sidebar with a 'Files' section containing a 'main' folder. Below it are three branches: '1 avance', '2 avance', and '3 avance (en proceso)'. The '1 avance' branch contains files like 'APF1 PPT.pdf', 'APF1.pdf', and 'El-y-Ella-Detalles-1.11.zip'. The '2 avance' branch contains 'APF2 PPT.pdf', 'APF2.pdf', 'El y Ella Detalles 2.22.rar', and 'db_flores 1.11.sql'. The '3 avance (en proceso)' branch is currently selected. The main area displays a commit history for 'LOAD-13' with four entries, all from yesterday. The commits are: 'Add files via upload' (with file names 'APF1 PPT.pdf', 'APF1.pdf', and 'El-y-Ella-Detalles-1.11.zip'), 'Add files via upload' (with file name 'APF1.pdf'), and 'Add files via upload' (with file name 'El-y-Ella-Detalles-1.11.zip'). The commit messages are all 'Add files via upload' and the dates are all 'yesterday'.

- 1 avance: Contiene las primeras versiones del sistema, junto con documentos preliminares como archivos PDF y versiones empaquetadas en ZIP.
- 2 avance: Abarca actualizaciones significativas del sistema, como nuevas versiones del proyecto en RAR y archivos SQL para la base de datos.
- 3 avance (en proceso): Indica que se encuentran en desarrollo las fases finales del proyecto, con nuevas funcionalidades aún en implementación.





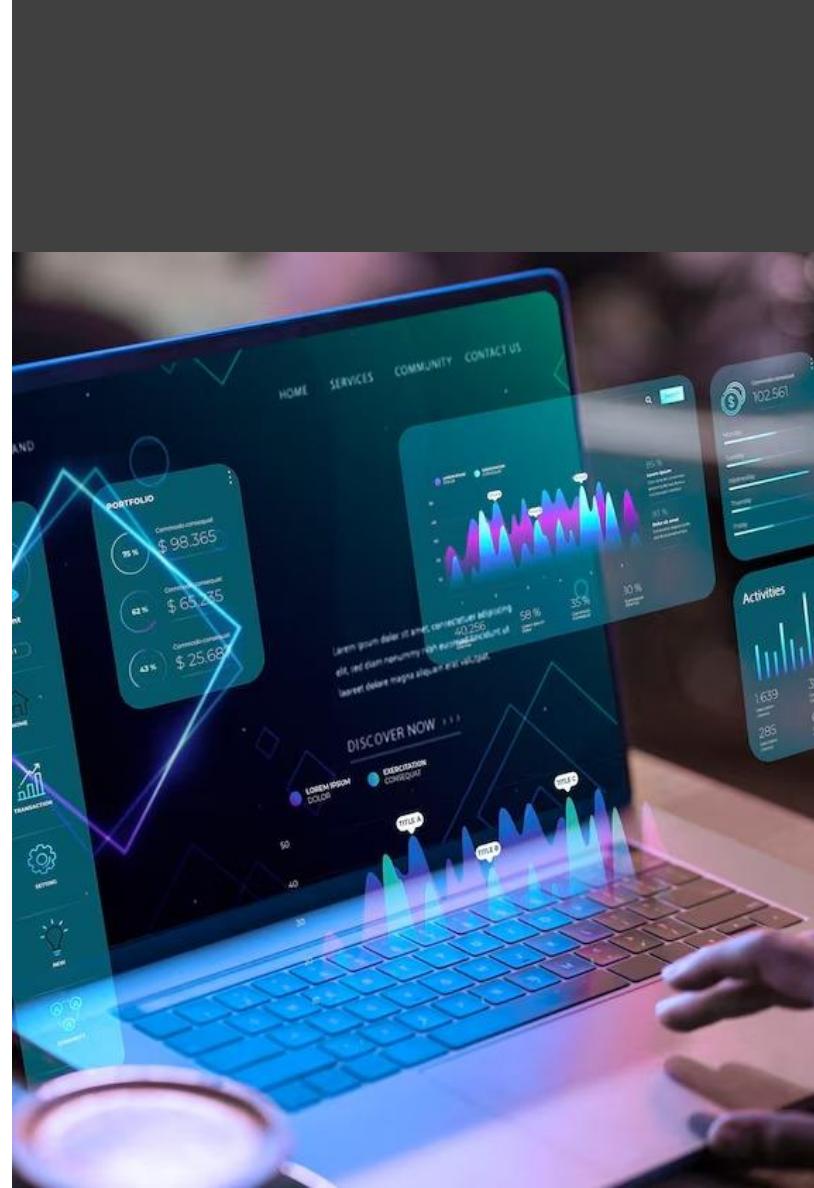
AUTORÍA DEL PROYECTO Y DOMINIO DEL CÓDIGO

Evidencia de autoría y conocimiento del sistema

El dominio del sistema se evidencia en la capacidad del equipo para explicar y justificar cada componente desarrollado. Cada miembro ha trabajado de forma autónoma en la implementación de funcionalidades específicas, y se ha realizado una revisión continua del trabajo mediante sesiones de codificación compartida para asegurar que todos los miembros comprendan el funcionamiento del sistema en su totalidad.

The screenshot shows a GitHub repository interface with a dark theme. It displays a list of commits from different users over time. The commits are related to the implementation of various components of the system, such as service packages, DTO models, DAO models, controllers, and user interfaces. Each commit includes the author's name, the branch it was pushed to, the commit message, and the timestamp.

- Implementación del paquete servicio con la clase DBConexion y otra cl...
- Implementación del paquete servicio con la clase DBConexion y otra cl...
- Implementación del modelo dto
- Implementación del modelo dto
- Implementación del modelo dao
- Implementación del modelo dao
- Implementación del Controlador
- Implementación del Controlador
- Implementación de las Interfaces de Usuario Principales y su código c...
- Implementación de las Interfaces de Usuario Principales y su código c...
- *****Detalles de pedidos, productos y edición form*****
- *****Detalles de pedidos, productos y edición form*****
- Primera implementación de una interfaz gráfica para el perfil de usuario
- Primera implementación de una interfaz gráfica para el perfil de usuario
- Iniciando repositorio



12





01



1

El sistema desarrollado bajo el patrón MVC facilitó la separación de responsabilidades, permitiendo que cada componente (modelo, vista y controlador) se mantuviera modular y fácil de gestionar, lo que asegura una mayor escalabilidad y mantenimiento a futuro.

2

La utilización de GitHub como plataforma de control de versiones fomentó una colaboración efectiva dentro del equipo, permitiendo realizar un seguimiento detallado de los cambios, así como facilitar las revisiones continuas y correcciones en tiempo real.

3

Las decisiones de diseño tomadas en la creación del modelo de datos garantizaron una estructura clara y funcional para la gestión de información, lo que permitió una integración eficiente entre los diferentes módulos del sistema, como los pedidos, usuarios y productos.

4

El uso de librerías especializadas como MySQL Connector y JUnit contribuyó a optimizar tanto la conectividad con la base de datos como las pruebas unitarias del sistema, asegurando la fiabilidad del software desarrollado.

5

Finalmente, la documentación detallada y las revisiones de Pull Request garantizaron que el código entregado cumpliera con las especificaciones iniciales del proyecto, minimizando errores y permitiendo que el equipo mantuviera un alineamiento constante con los objetivos planteados desde el inicio.

CONCLUSIONES

RECOMENDACIONES



- 1 Para mantener y mejorar el sistema en el futuro, se recomienda seguir con la estructura modular basada en el patrón MVC, promoviendo así la escalabilidad y facilitando el mantenimiento continuo del software.
- 2 Se sugiere continuar utilizando GitHub como herramienta de control de versiones, fomentando prácticas de integración continua y revisiones colaborativas del código para garantizar la calidad del proyecto a lo largo del tiempo.
- 3 Es recomendable llevar a cabo sesiones periódicas de revisión y optimización del modelo de datos, asegurando que cualquier expansión del sistema mantenga la eficiencia y coherencia de la gestión de información.
- 4 Se aconseja incorporar nuevas librerías y frameworks según las necesidades futuras del proyecto, así como realizar pruebas automatizadas más extensivas para detectar posibles errores antes de pasar a producción.
- 5 Finalmente, se recomienda mantener la documentación siempre actualizada y alineada con los cambios realizados en el código, asegurando así una fácil comprensión por parte de los futuros desarrolladores o equipos de soporte.



01





01



Chacon, S., & Straub, B. (2021). Pro Git (2nd ed.). Apress.
<https://git-scm.com/book/en/v2>

Choshin, M., y Ghaffari, A. (2023). El impacto del comercio electrónico en el desempeño de las pymes en un país en desarrollo. *Journal of Organizational Change Management*, 36(3), 477-491. <https://doi.org/10.1108/JOCM-07-2022-0200>

Dingsøyr, T., Moe, N. B., & Seim, E. A. (2018). Coordinating Knowledge Work in Multiteam Programs: Findings From a Large-Scale Agile Development Program. *Project Management Journal*, 49(6), 64–77. <https://doi.org/10.1177/8756972818798980>

Kumar, V. y Raheja, G. (2021). Adopción del comercio electrónico entre empresas: una investigación empírica de la teoría del comportamiento planificado. *Technology in Society*, 65, 101575. <https://doi.org/10.1016/j.techsoc.2021.101575>

REFRENCIAS

ANEXOS

Integración de GitHub con NetBeans

Documentación en Javadoc

Documentación del Proyecto con Javadoc

¿Qué es Javadoc y para qué sirve?

Javadoc es una herramienta incluida en el JDK (Java Development Kit) de Java que se utiliza para generar documentación en formato HTML a partir de comentarios especiales en el código fuente de Java. Estos comentarios describen el comportamiento y la funcionalidad de clases, métodos y atributos, proporcionando una referencia clara y organizada para los desarrolladores.

¿Cómo se genera la documentación con Javadoc?

La documentación se genera añadiendo comentarios Javadoc en el código fuente antes de las declaraciones de clases, métodos y atributos. Una vez añadidos estos comentarios, se utiliza una herramienta como NetBeans para generar el HTML de la documentación. En NetBeans, esto se hace seleccionando el proyecto y utilizando la opción "Generate Javadoc..." en el menú contextual.

Organización de la Documentación Javadoc

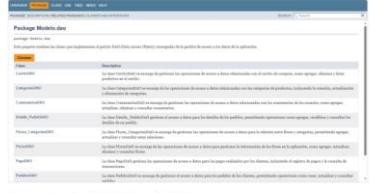
Visión General de los Paquetes

La documentación generada incluye una página principal que muestra todos los paquetes del proyecto, con una breve descripción de cada uno de ellos. Esto proporciona una visión general de la estructura del proyecto y facilita la navegación entre los distintos componentes. Como se observa en la imagen, cada paquete tiene su propia descripción que detalla su propósito en la aplicación.



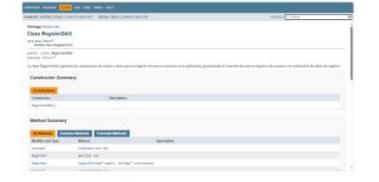
Detalles de las Clases en un Paquete

Dentro de cada paquete, la documentación incluye una tabla con todas las clases que contiene, acompañada de una breve descripción de cada una. Esta tabla permite identificar rápidamente la funcionalidad de cada clase y entender cómo se relacionan entre sí dentro del paquete. En la imagen correspondiente, se puede ver cómo se presentan estas clases y sus descripciones, lo cual facilita la exploración de la lógica del proyecto.



Documentación de Métodos y Atributos

La documentación Javadoc también proporciona una detalle más específico dentro de cada método o atributo, conocido como nivel de detalle. Para cada método, se ofrece una descripción detallada junto con los parámetros que acepta, los valores que devuelve, si corresponde. Este nivel de detalle permite a los desarrolladores comprender el uso adecuado de cada método y atributo de las clases. En la imagen se muestra un ejemplo de cómo se presenta esta información en el HTML generado por Javadoc.



Documentación del Proyecto con Javadoc

Screenshot of the GitHub repository history showing frequent commits from various users related to the implementation of packages, models, controllers, and interfaces.

Showing most recent first

- Implementación del paquete servicio con la clase DBConexion y otra c...
LOAD-13 pushed 1 commit to master · b0d6968...37463e3 · 14 seconds ago
- Implementación del paquete servicio con la clase DBConexion y otra c...
LOAD-13 pushed 4 commits to Joaquin_Los · 0b4c79...37463e3 · 47 seconds ago
- Implementación del modelo dto
Borderline06 pushed 1 commit to master · 5b521fb...b0db9d8 · 6 minutes ago
- Implementación del modelo dto
Borderline06 created Jorge_Membrillo · b0db9d8 · 6 minutes ago
- Implementación del modelo dao
Kira2052001 pushed 1 commit to master · b1d4538...5b521fb · 10 minutes ago
- Implementación del modelo dao
Kira2052001 created Kira_Sentii · 5b521fb · 11 minutes ago
- Implementación del Controlador
Sandronic22 pushed 1 commit to master · 0b34c79...bf1d4538 · 19 minutes ago
- Implementación del Controlador
Sandronic22 created Sandro_de_la_ruz · bf1d4538 · 20 minutes ago
- Implementación de las Interfaces de Usuario Principales y su codigo ...
LOAD-13 pushed 1 commit to master · d5560e8...fb34c79 · 25 minutes ago
- Implementación de las Interfaces de Usuario Principales y su codigo ...
LOAD-13 pushed 2 commits to Joaquin_Los · 4c011b...0b34c79 · 26 minutes ago
- *****Detalles de pedidos, productos y edición form*****
Itachi-bit pushed 1 commit to master · 4e220f1b...d5560e8 · 41 minutes ago
- *****Detalles de pedidos, productos y edición form*****
Itachi-bit created Jefferson_Penla · d5560e8 · 42 minutes ago
- Primera implementación de una interfaz gráfica para el perfil de usuario
LOAD-13 pushed 1 commit to master · 2d3a95e...4620fb · 57 minutes ago
- Primera implementación de una interfaz gráfica para el perfil de usuario
LOAD-13 created Joaquin_Los · 4620fb · 1 hour ago
- Iniciando repositorio
Borderline06 created master · 2d3a95e · 1 hour ago

Share feedback about this page



01



**Sistema de Comercio digital para la
floristería “El y Ella Detalles” en Java.**

**MUCHAS
GRACIAS**