

FRONT-END

○ ZALDE ALDAY ELIZABETH
VIRIDIANA

○

○ ZENDEJAS MENDEZ EDWIN
ANTONIO

○

○ ROSALES JUAN DE DIOS ISAAC

○

○ MARTINEZ CORTES CRISTOPHER
HERNAN

○



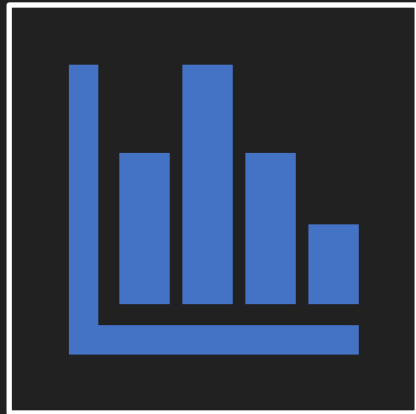
¿Qué es un Front-End?

- El Front end es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos y la que permite a los visitantes navegar por las diferentes páginas mientras lo deseen.

Problemática

- Es una distribuidora de ropa, que lleva su inventario de productos y entregar en papel, lo que genera que ocurran ciertas pérdidas de información por que registrarse de forma correcta o bien por que se traspapelan los reportes.

Solución del problema



Se realizará una interfaz gráfica amigable con el usuarios (Enviex) estará conectada con una base de datos en la que se almacenara la información necesaria para llevar un control exacto de los productos entregados a los destinatarios correspondientes.



Si en algún momento existe un inconveniente con la entrega de un producto, Se realizará aún checklist para confirmar que los datos fueron ingresado de manera correcta.

A dark gray, abstract, rounded triangular shape pointing towards the right, occupying the left side of the slide.

Desarrollo del programa

Diseño y Programacion del software

```
mirror_mod = modifier_ob.  
Set mirror object to mirror.  
mirror_mod.mirror_object  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
@selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
print("please select exactly
```

--- OPERATOR CLASSES ---

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

```
context):  
context.active_object is not
```

Requisitos para las programación

- Datos del cliente (Nombre, apellidos, teléfono, etc.)
- Destino (Calle, colonia, código postal)
- Datos financieros (Método de pago, monto a pagar por el envío, etc.)


```
<default config>
INTERFAZ.java x pedidosDAO.java x pedidosVO.java x
History
public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public String getCalle() {
    return calle;
}

public void setCalle(String calle) {
    this.calle = calle;
}

public String getColonia() {
    return colonia;
}

public void setColonia(String colonia) {
    this.colonia = colonia;
}

public String getAlcaldia() {
    return alcaldia;
}
```

```
<default config>
Start Page x INTERFAZ.java x pedidosDAO.java x pedidosVO.java x
Source History
1 package classVO;
2
3
4 import javax.accessibility.AccessibleContext;
5
6
7 public class pedidosVO {
8
9     private int id_pedido;
10    private String nombre;
11    private String apellido;
12    private String telefono;
13    private String calle;
14    private String colonia ;
15    private String alcaldia;
16    private String codigo_postal;
17    private String descripcion_pedido;
18    private String Nombre_bazar;
19    private String monto_pagar;
20    private String tipo_pago;
21    private String resultado;
22
23    public String getResultado() {
24        return resultado;
25    }
26
27    public void setResultado(String resultado) {
28        this.resultado = resultado;
29    }
30
31    public int getId_pedido() {
32        return id_pedido;
33    }
}
```



```

public class pedidosDAO {
    public static String registrarpedidos (pedidosVO ped) throws SQLException{

        String resultado=null, last=null;
        conectar cc= new conectar ();
        Connection cn= cc.getConnection();
        PreparedStatement pst = null;
        String sql = "INSERT INTO pedidos VALUES (null, ?,?, ?,?, ?,?, ?,?, ?,?, ?)";

        try{
            pst=cn.prepareStatement(sql);
            pst.setString(1, ped.getNombre());
            pst.setString(2, ped.getApellido());
            pst.setString(3, ped.getTelefono());
            pst.setString(4, ped.getCalle());
            pst.setString(5, ped.getColonia());
            pst.setString(6, ped.getAlcaldia());
            pst.setString(7, ped.getCodigo_postal());
            pst.setString(8, ped.getDescripcion_pedido());
            pst.setString(9, ped.getNombre_bazar());
            pst.setString(10, ped.getMonto_pagar());
            pst.setString(11, ped.getTipo_pago());

            pst.execute();
            pst=cn.prepareStatement("SELECT MAX (id_pedido) AS id_ped FROM pedidos");
            ResultSet rs= pst.executeQuery();

            if(rs.next()){
                last = rs.getString(1);
            }

            resultado= "El pedido ha sido añadido id"+ last;
        }
    }
}

```

```

package clases;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;

public class conectar {

    Connection conectar=null;

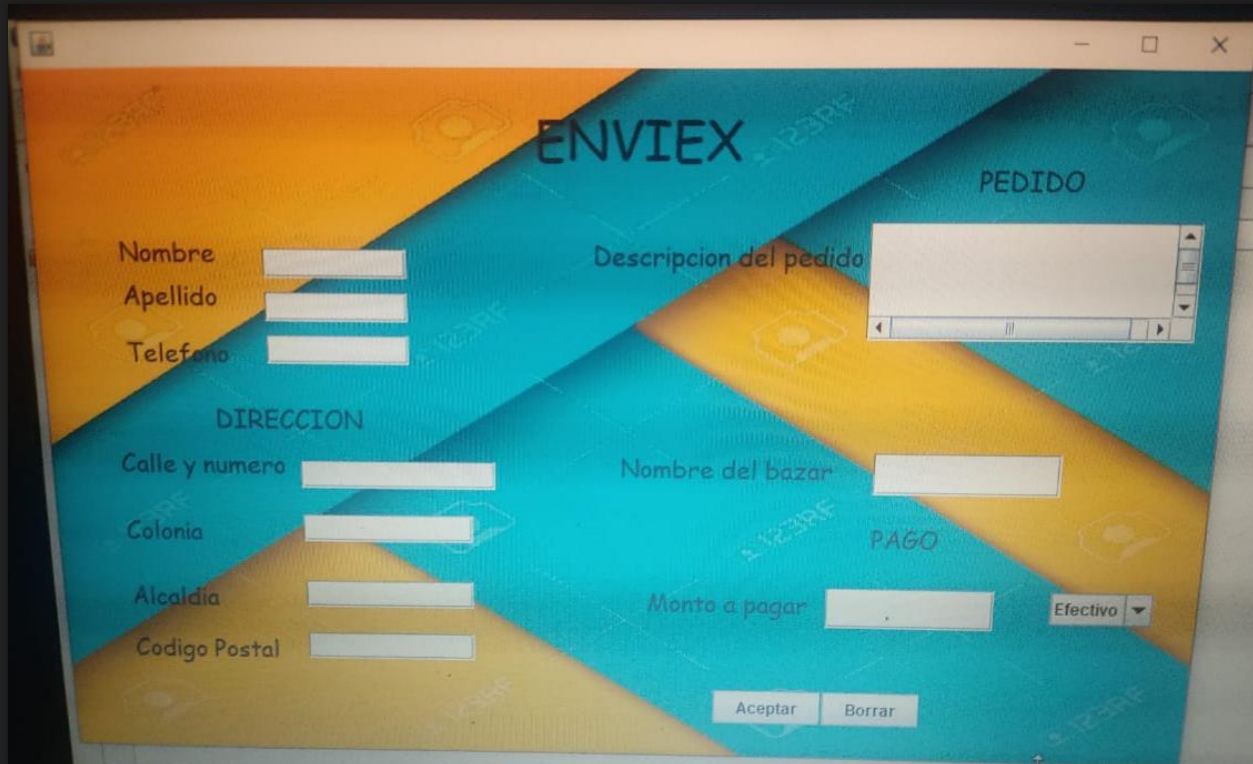
    public conectar(){

        try {
            Class.forName("com.mysql.jdbc.Driver");
            conectar= (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/enviex", "root", "");
            if (conectar !=null){
                JOptionPane.showMessageDialog(null, "La conexión ha sido exitosa");
            }
        }
        catch (Exception e){
            System.out.println("Error al conectar"+ e);
        }
    }

    public Connection getConnection () {
        return conectar;
    }
}

```

Resultado Final



The image shows a screenshot of a web application window titled "ENVIEX PEDIDO". The interface has a blue and yellow geometric background. It contains several input fields and buttons. On the left, under the heading "DIRECCION", there are fields for "Calle y numero", "Colonia", "Alcaldia", and "Codigo Postal". Above these, there are fields for "Nombre", "Apellido", and "Telefono". In the center, there is a large text area for "Descripcion del pedido". To the right of this, there is a section for "PAGO" with a field for "Monto a pagar" and a dropdown menu currently set to "Efectivo". At the bottom center, there are two buttons: "Aceptar" and "Borrar".

ENVIEX PEDIDO	
Nombre <input type="text"/>	Descripcion del pedido <input type="text"/>
Apellido <input type="text"/>	
Telefono <input type="text"/>	
DIRECCION	
Calle y numero <input type="text"/>	Nombre del bazar <input type="text"/>
Colonia <input type="text"/>	PAGO
Alcaldia <input type="text"/>	
Codigo Postal <input type="text"/>	
Monto a pagar <input type="text"/> Efectivo ▼	
Aceptar Borrar	

Aquí podemos observar el resultado final de la interfaz para el usuario ENVIEX distribuidora, aquí podemos observar que de los datos recopilados creamos un programa adecuado a las necesidades del cliente