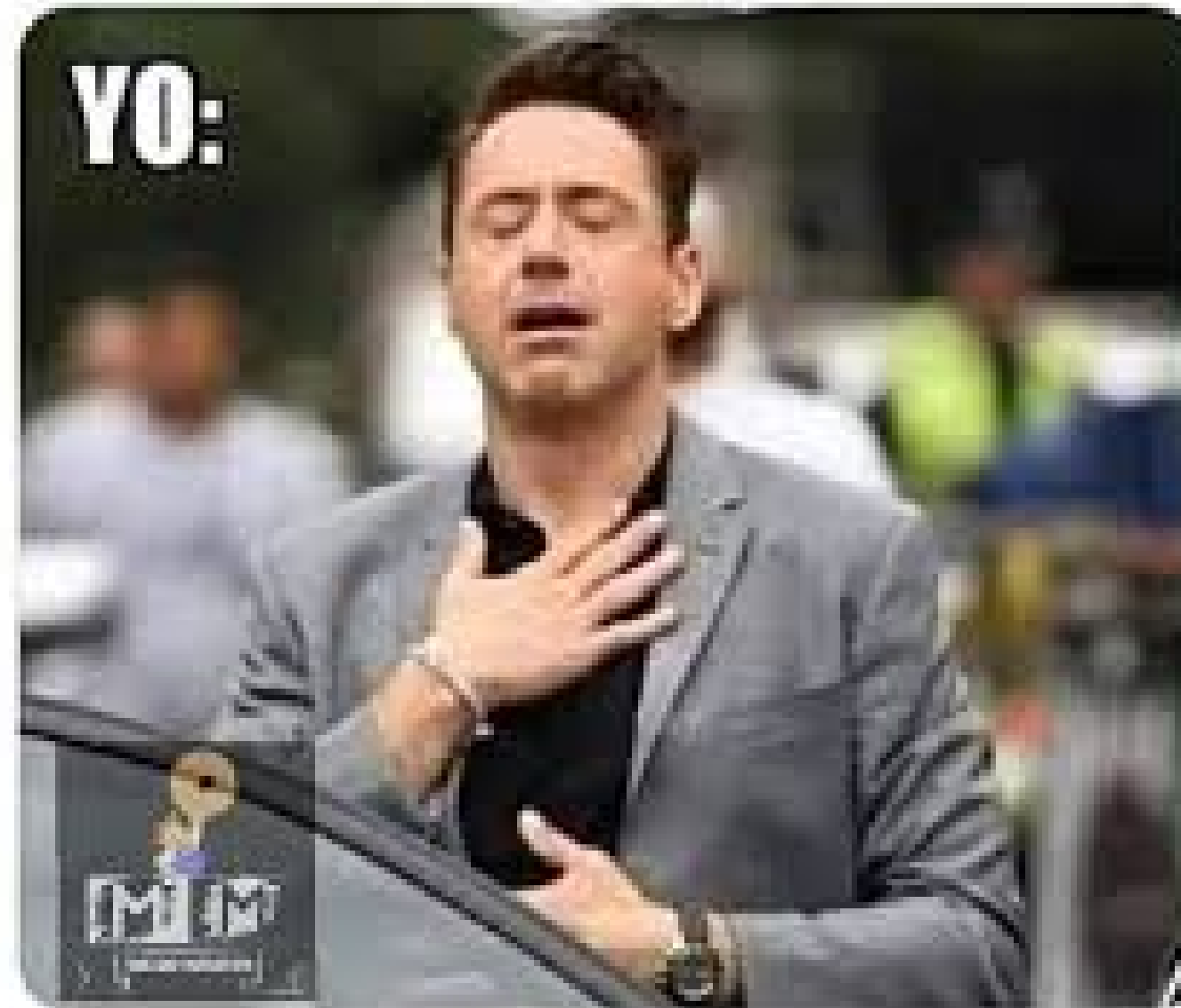


KATHERINE FIALLOS  
MIGUEL FLORES  
SARAH CASTELAR  
OSWALDO VARELA



# FeedMe

***El coronavirus no le da  
a los Meseros***

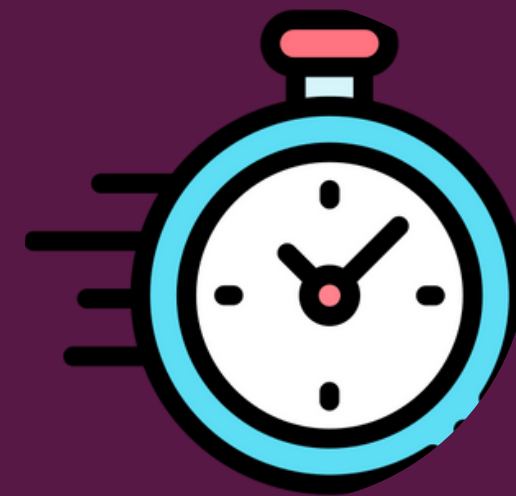


# PROBLEMA

ORDENAR COMIDA EN LOS  
RESTUARANTES ES UN  
PROCESO TEDIOSO EN  
TIEMPOS DE PANDEMIA.



RIESGO DE  
CONTAGIO



PÉRDIDA  
DE TIEMPO



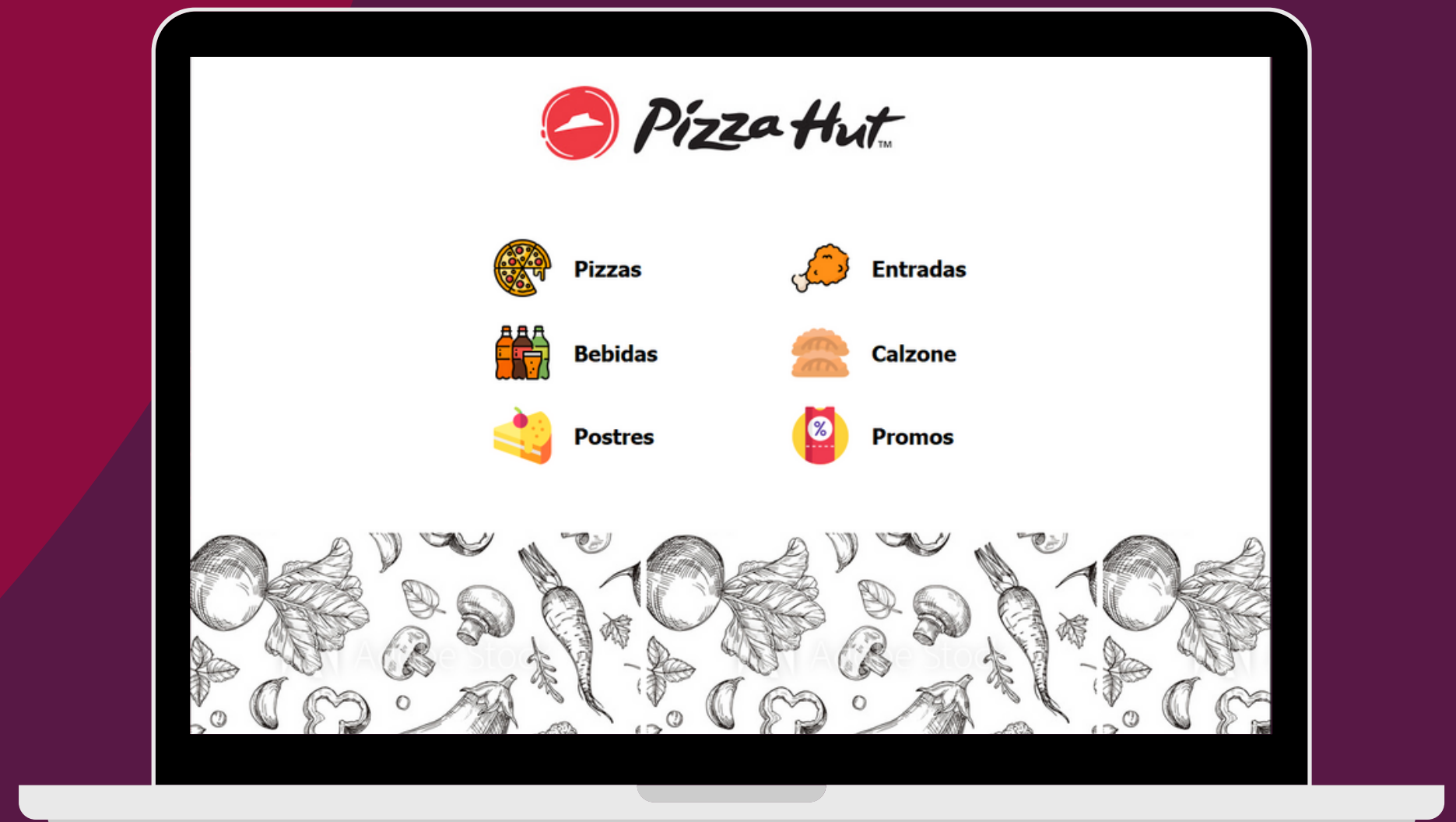
OPCIONES  
ACTUALES  
NO EFICACES

# SOLUCIÓN

## FEEDME OFRECE:

FeedMe ofrece el servicio de logística de ordenes para los negocios, tambien para que los clientes puedan ordenar su pedido y gestionar el pago necesidad sin empleados.

MENÚ INTERACTIVO  
PARA ORDERNAR  
PRODUCTO SIN  
CONTÁCTO CON LOS  
DEMÁS



# HERRAMIENTAS UTILIZADAS



**MICROSOFT  
SQL SEVER**

**JAVA**



**GITHUB**



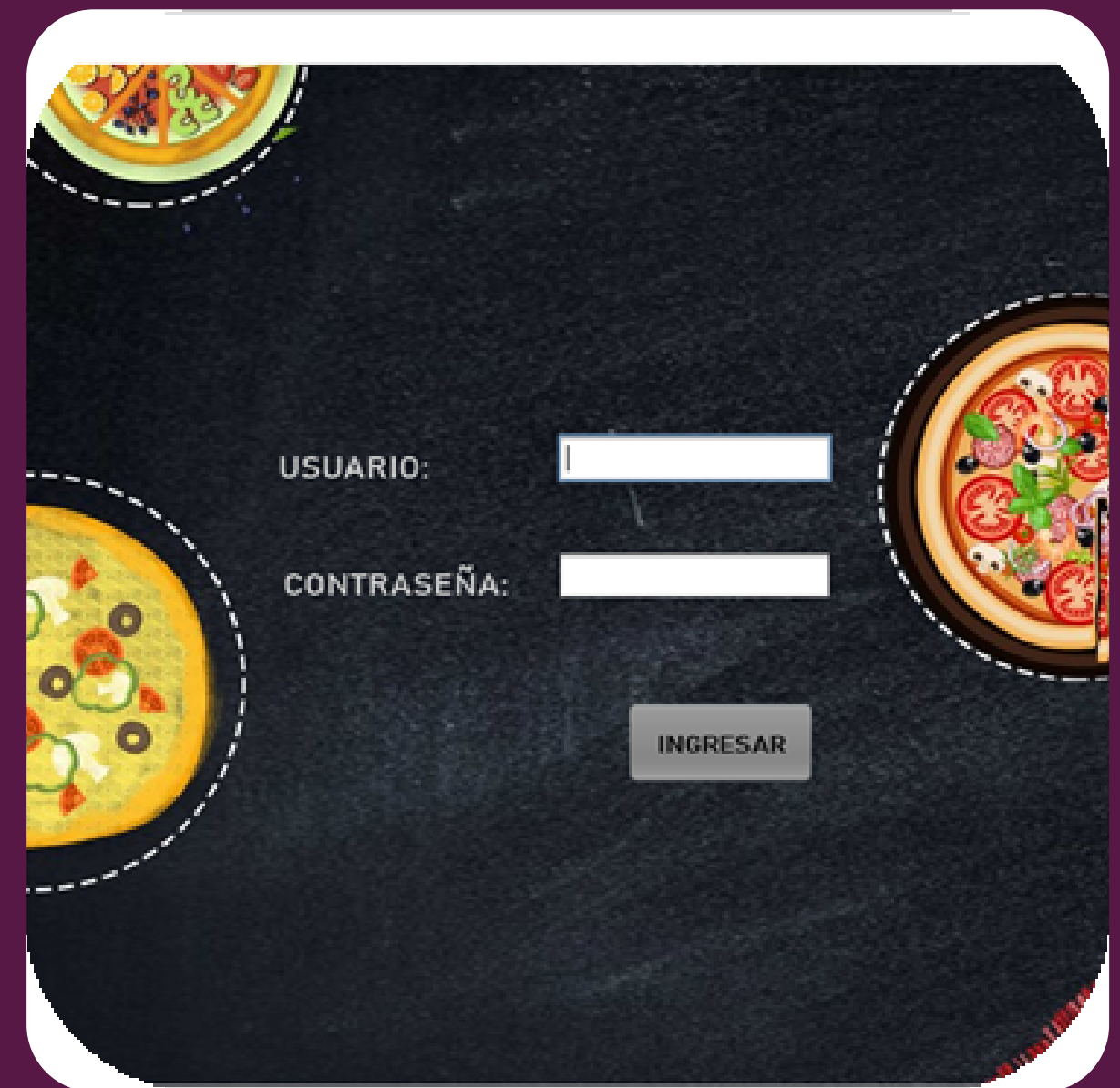
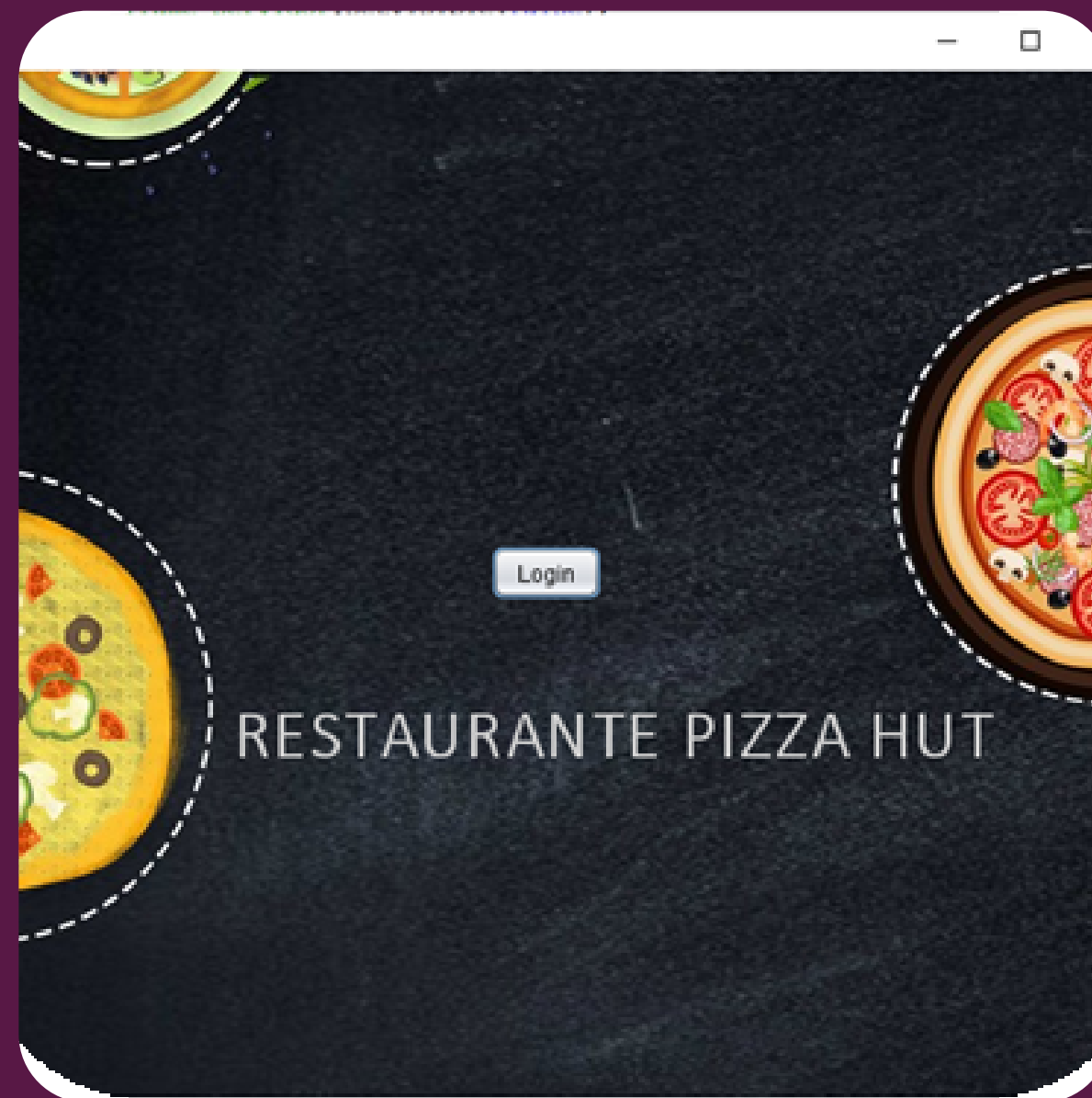
1



FASE:  
SERVIDOR

# Pantalla Login

Parte gráfica de la ventanilla principal para los administradores del restaurante. Solamente acceso autorizado con personas previamente registradas.



# Pantalla Menú Principal Servidor



Pantalla donde se visualiza la apertura y cierre del restuarante. Tambiens se ve en tiempo real las ventas.



# INVENTARIO

Se visualiza cantidad, nombre, precio y descripción de las comidas.

SERVIDOR PIZZA HUT

InfoInventarioMesas

COMIDASBEBIDASPOSTRES

Cantidad	Nombre	Precio	Descripción
50	4 Estaciones	318.0	.
3	444	44.0	44
60	Calzone Meat Lovers	115.0	.
60	Calzone Supremo	115.0	.
50	Canadiense	280.0	.
60	Chicken Buffalo	115.0	.
25	Hut Pack	299.0	.
100	Hut Wings	149.0	.
50	Pan de ajo supremo	89.0	.
25	Pechu Pack	599.0	.
400	Pasta con salsa	100.0	.

Cantidad

Nombre

Precio

Descripción

0

AGREGAR

MODIFICAR

ELIMINAR

TM

# INVENTARIO

Se visualiza cantidad, nombre, precio y descripción de las bebidas.

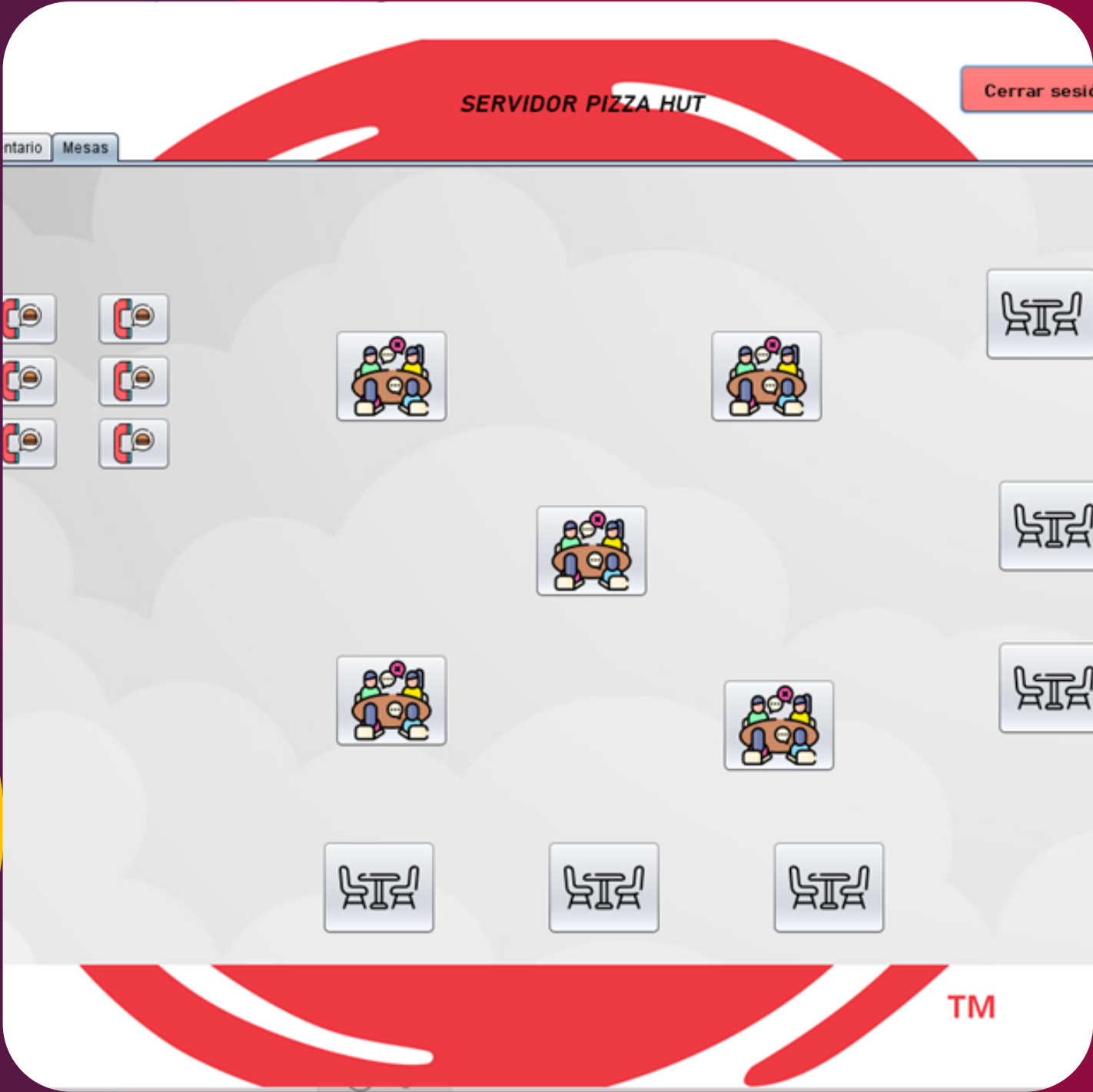


# INVENTARIO

Se visualiza cantidad, nombre, precio y descripción de los postres.



# Distribución mesas y detalle de pedido por mesa



### GESTION DE MESA

Producto	Cantidad	Precio
4 ESTACIONES	2	318.0
CALZONE MEAT LOVERS	1	115.0
CHICKEN BUFFALO RAN...	1	115.0
PEPSI 1/2 L	1	29.0

AGREGARA OR...

GENERAR COBRO

# Tablas DB

SQLQuery1.sql - KATHERINE\MSSQLSERVER01.Pizza (kk (68))\* - Microsoft SQL Server Management Studio

Edit View Query Project Tools Window Help

SQLQuery1.sql - KAT...R01.Pizza (kk (68))\*

SELECT\* FROM Inventario WHERE Categoria='comida'

SELECT\* FROM Inventario WHERE Categoria='bebida'

SELECT\* FROM Inventario WHERE Categoria='postre'

SELECT\* FROM DETALLE\_PEDIDO

100 %

Results Messages

	Nombre	Descripcion	Precio	Cantidad	Categoria
1	4 Estaciones	.	318	11	comida
2	alitas de celdo	.	56.2999992370605	8	comida
3	alitas de res	.	56.2999992370605	8	comida
4	Calzone Meat Lovers	.	115	57	comida
5	Calzone Supremo	.	115	51	comida
6	Canadiense	.	280	29	comida
7	Chicken Buffalo	.	115	57	comida
8	ensalada	44	125	3	comida
9	Hut Pack	.	299	19	comida
10	Hut Wings	.	149	98	comida
11	Pan de ajo supremo	.	89	47	comida
12	Pasta Pack	.	500	16	comida

	Nombre	Descripcion	Precio	Cantidad	Categoria
1	7UP 1/2 L	.	29	100	bebida
2	AGUA 20 OZ	...	20	103	bebida
3	PEPSI 1/2 L	..	29	88	bebida

	Nombre	Descripcion	Precio	Cantidad	Categoria
1	Hersheys Chocolate Chip	.	149	63	postre
2	Pecan Pie	.	89	41	postre

	ID_detalle	Nombre_Producto	Cantidad	Precio	Estado	Categoria	ID_mesa	ID_llevar
1	0	4 ESTACIONES	2	318	1	Comer en restaurante	1	NULL
2	0	CALZONE MEAT LOVERS	1	115	1	Comer en restaurante	1	NULL
3	0	CHICKEN BUFFALO RANCH	1	115	1	Comer en restaurante	1	NULL
4	0	PEPSI 1/2 L	1	29	1	Comer en restaurante	1	NULL

# Código Conexión DB

```
public class Conexion {  
    static Connection contacto=null;  
    public static Connection getConnection() throws ClassNotFoundException, SQLException{  
  
        try {  
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
String url="jdbc:sqlserver://KATHERINE\\MSSQLSERVER01:1433;database=Pizza;user=kk;password=kk;";  
Connection con= DriverManager.getConnection(url);  
System.out.println("se conectó esta papada");  
return con;  
        } catch (Exception e) {  
            System.out.println(e.toString());  
            return null;  
        }  
    }  
}
```



# Código CRUD

```
private void bt_modificar_comidaMouseClicked(java.awt.event.MouseEvent evt) {  
  
    //String nombre2=jt_nombre_comida.getText();  
    if(((Integer)js_cantidad_comida.getValue()<=0) || jt_nombre_comida.getText().equals("") || jt_precio_comida.  
        JOptionPane.showMessageDialog(frame_servidor, "Por favor, introducir todos los datos");  
    }else{  
        //validad si ya esta agregado en la base de datos  
        try {  
            int cantidad=(Integer)js_cantidad_comida.getValue();  
            String nombre=jt_nombre_comida.getText();  
            String descripcion=jt_descripción_comida.getText();  
            float precio=Float.parseFloat(jt_precio_comida.getText());  
  
            //BASE DE DATOS  
            try {  
                Connection con=Conexion.getConnection();  
                PreparedStatement ps=con.prepareStatement("UPDATE Inventario SET Nombre=? ,Descripcion=?,Precio=?  
                    + "WHERE Nombre=? ");  
                ps.setString(1, nombre);  
                ps.setString(2, descripcion);  
                ps.setFloat(3, precio);  
                ps.setInt(4, cantidad);  
                ps.setString(5, nombresito_comida);  
  
                ps.executeUpdate();  
                cargarTablacomida();  
                JOptionPane.showMessageDialog(frame_servidor, "Comida modificada exitosamente");  
  
            } catch (Exception e) {  
                System.out.println(e.toString());  
            }  
        } catch (Exception e) {  
            System.out.println(e.toString());  
        }  
    }  
}
```

```
private void bt_eliminar_comidaActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String nombr2e=jt_nombre_comida.getText();  
  
    //validad si ya esta agregado en la base de datos  
    try {  
        try {  
            Connection con=Conexion.getConnection();  
            PreparedStatement ps=con.prepareStatement("DELETE Inventario WHERE Nombre=?");  
            ps.setString(1, nombr2e);  
            ps.executeUpdate();  
  
            cargarTablacomida();  
            JOptionPane.showMessageDialog(frame_servidor, "Comida modificada exitosamente");  
  
        } catch (Exception e) {  
            System.out.println(e.toString());  
        }  
    } catch (Exception e) {  
        System.out.println(e.toString());  
    }  
  
    js_cantidad_comida.setValue(0);  
    jt_precio_comida.setText("");  
    jt_nombre_comida.setText("");  
    jt_descripción_comida.setText("");  
}
```

```
private void bt_agregar_comidaMouseClicked(java.awt.event.MouseEvent evt) {  
  
    if(((Integer)js_cantidad_comida.getValue()<=0) || jt_nombre_comida.getText().equals("") || jt_precio_comida.  
        JOptionPane.showMessageDialog(frame_servidor, "Por favor, introducir todos los datos");  
    }else{  
        try {  
  
            int cantidad=(Integer)js_cantidad_comida.getValue();  
            String nombre=jt_nombre_comida.getText();  
            String descripcion=jt_descripción_comida.getText();  
            float precio=Float.parseFloat(jt_precio_comida.getText());  
  
            //BASE DE DATOS  
            try {  
                Connection con=Conexion.getConnection();  
                System.out.println("pasoooo");  
                PreparedStatement ps=con.prepareStatement("INSERT INTO Inventario (Nombre,Descripcion,Precio,Ca  
                    + "VALUES (?,?,?,?) ");  
                ps.setString(1, nombre);  
                ps.setString(2, descripcion);  
                ps.setFloat(3, precio);  
                ps.setInt(4, cantidad);  
                ps.setString(5, "comida");  
  
                ps.executeUpdate();  
                cargarTablacomida();  
                JOptionPane.showMessageDialog(frame_servidor, "Comida agregada exitosamente");  
  
            } catch (Exception e) {  
                System.out.println(e.toString());  
            }  
        } catch (Exception e) {  
            System.out.println(e.toString());  
        }  
  
        js_cantidad_comida.setValue(0);  
    }  
}
```

```
private void cargarTablacomida() {  
    DefaultTableModel modeloTabla=(DefaultTableModel)table_comida.getModel();  
    modeloTabla.setRowCount(0);  
    PreparedStatement ps;  
    ResultSet rs;  
    ResultSetMetaData rsmd;  
    int columnas;  
    int[] anchos={100,100,100,100};  
    for (int i = 0; i < table_comida.getColumnCount(); i++) {  
        table_comida.getColumnModel().getColumn(i).setPreferredWidth(anchos[i]);  
    }  
    try {  
        Connection con=Conexion.getConnection();  
        ps=con.prepareStatement("SELECT Cantidad,Nombre,Precio,Descripcion FROM Inventario WHERE Categoria='c  
        rs=ps.executeQuery();  
        rsmd=rs.getMetaData();  
        columnas=rsmd.getColumnCount();  
  
        while(rs.next()){  
            Object[] fila=new Object[columnas];  
            for (int i = 0; i < columnas; i++) {  
                fila[i]=rs.getObject(i+1);  
            }  
            modeloTabla.addRow(fila);  
        }  
  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(frame_servidor, e.toString());  
    }  
}
```



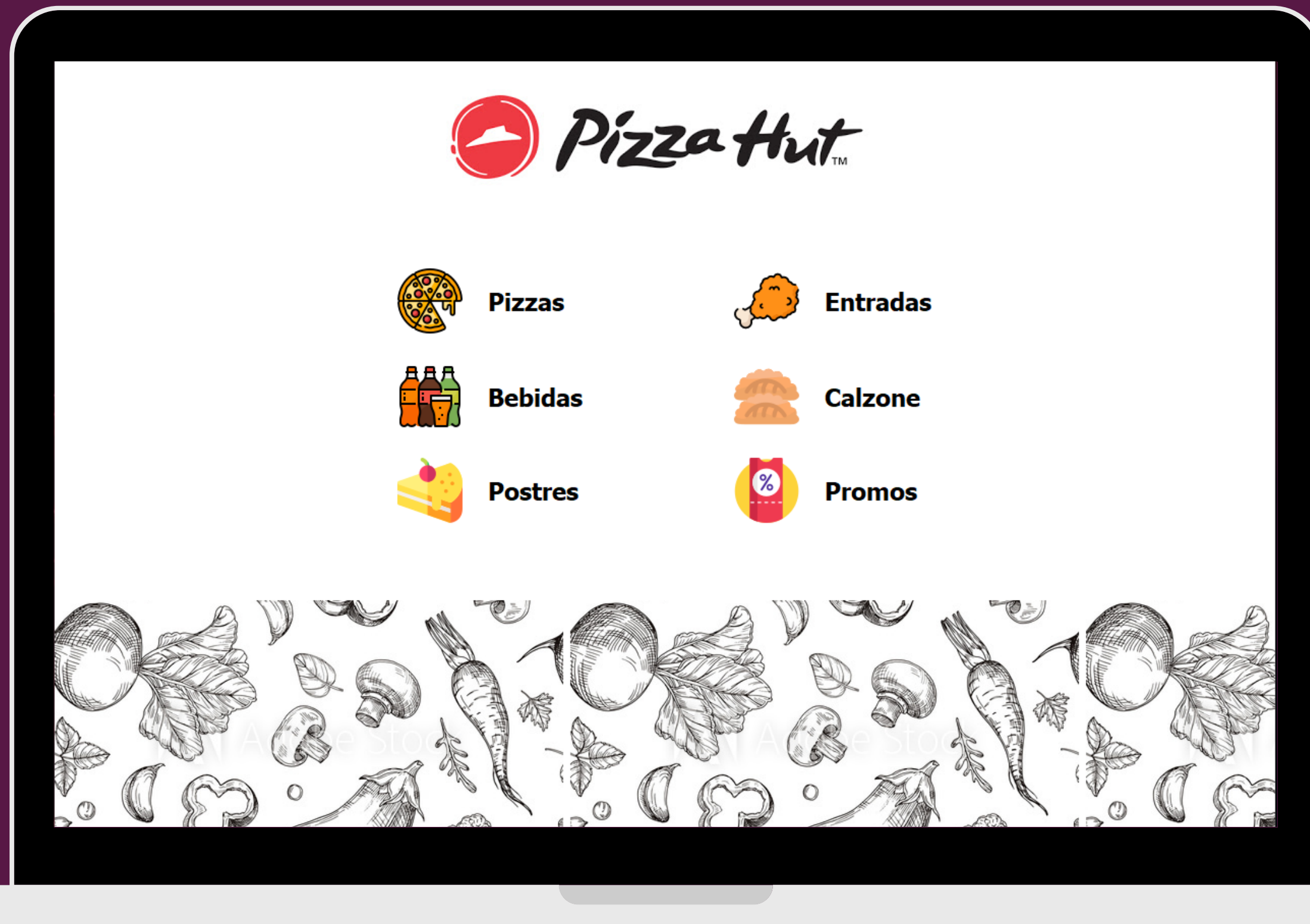
2



FASE:  
CLIENTE

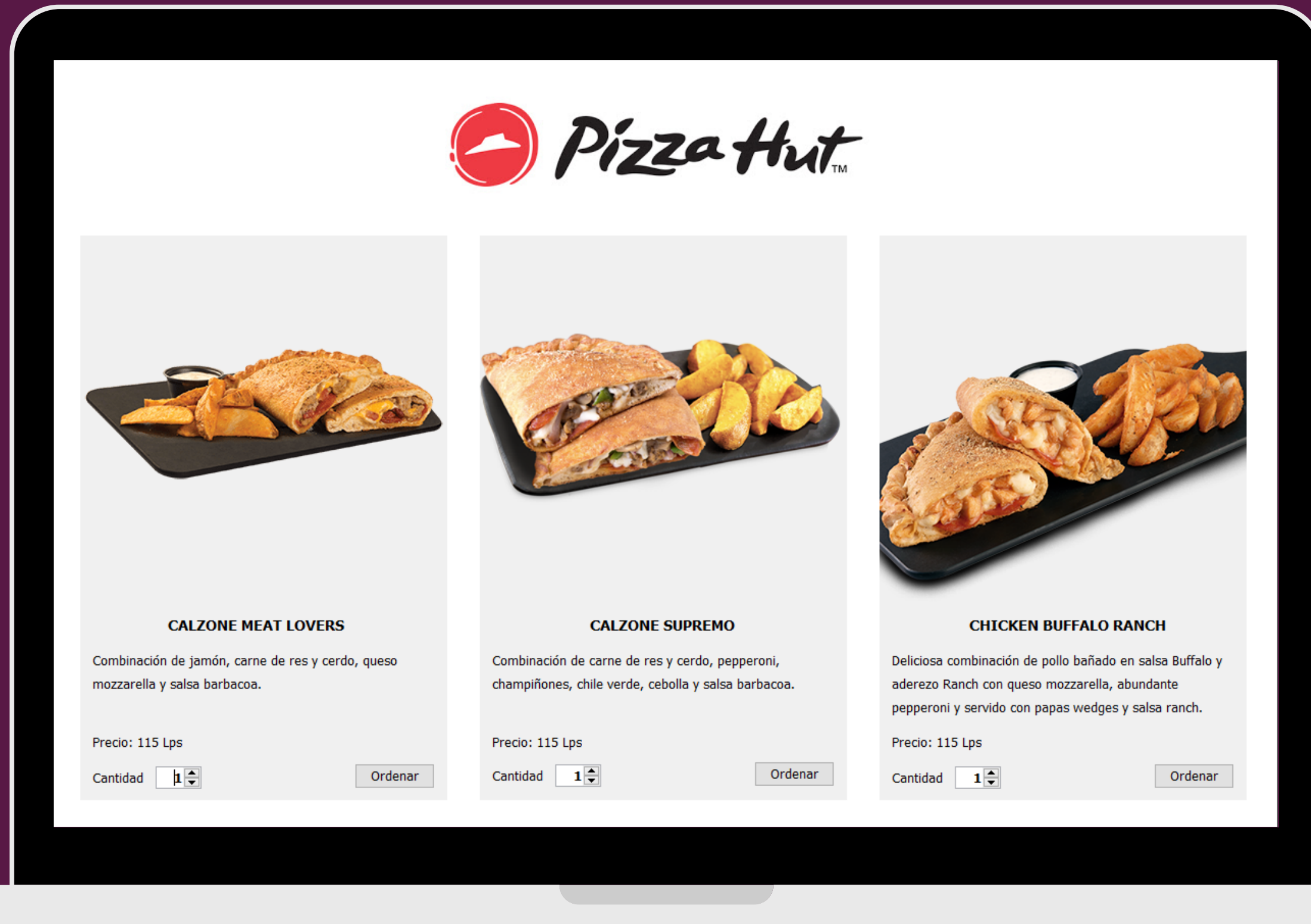


# Menú Principal Cliente



Lista de opciones  
donde puede  
escoger una  
categoría para  
empezar a  
comprar.

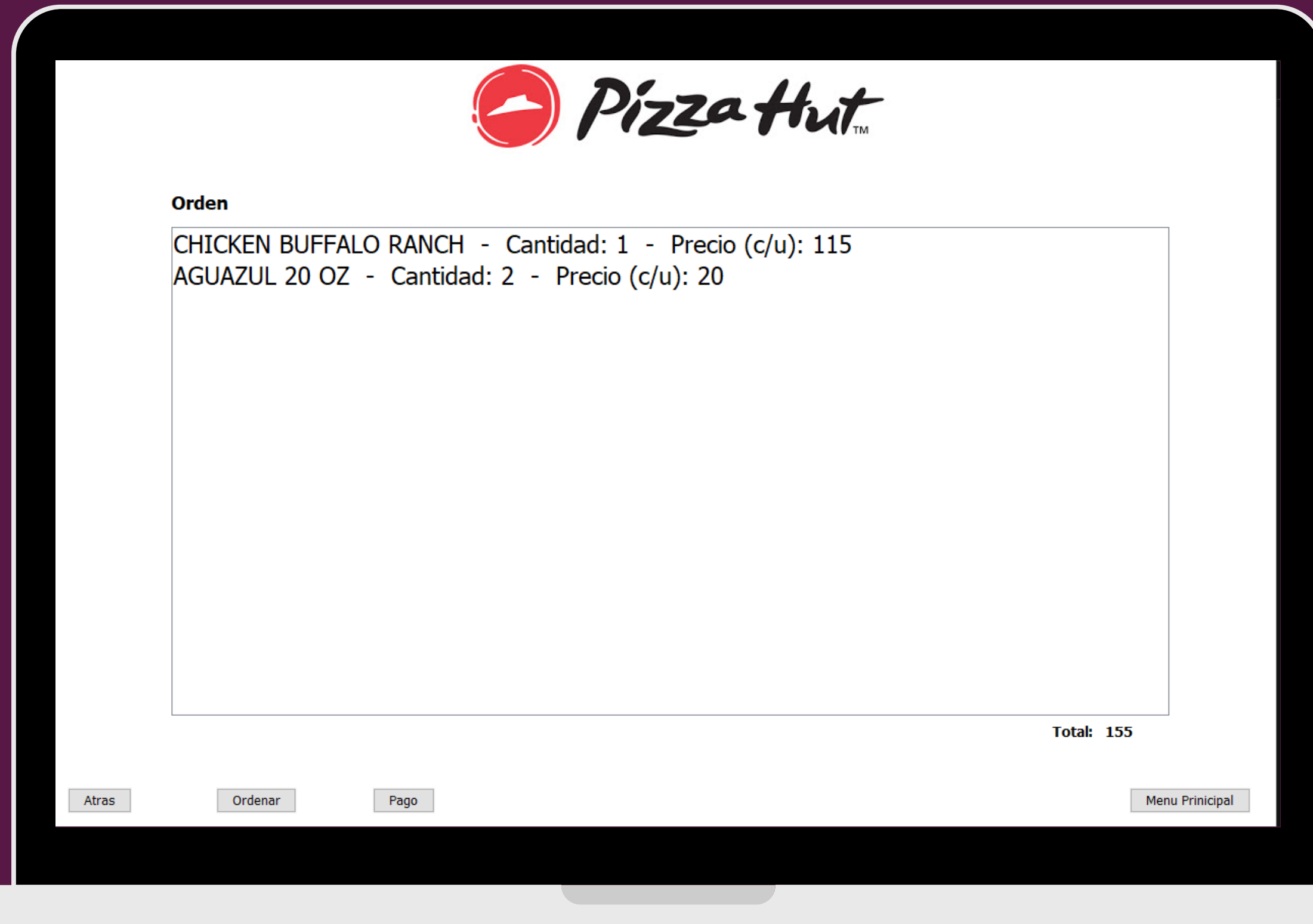
# Productos



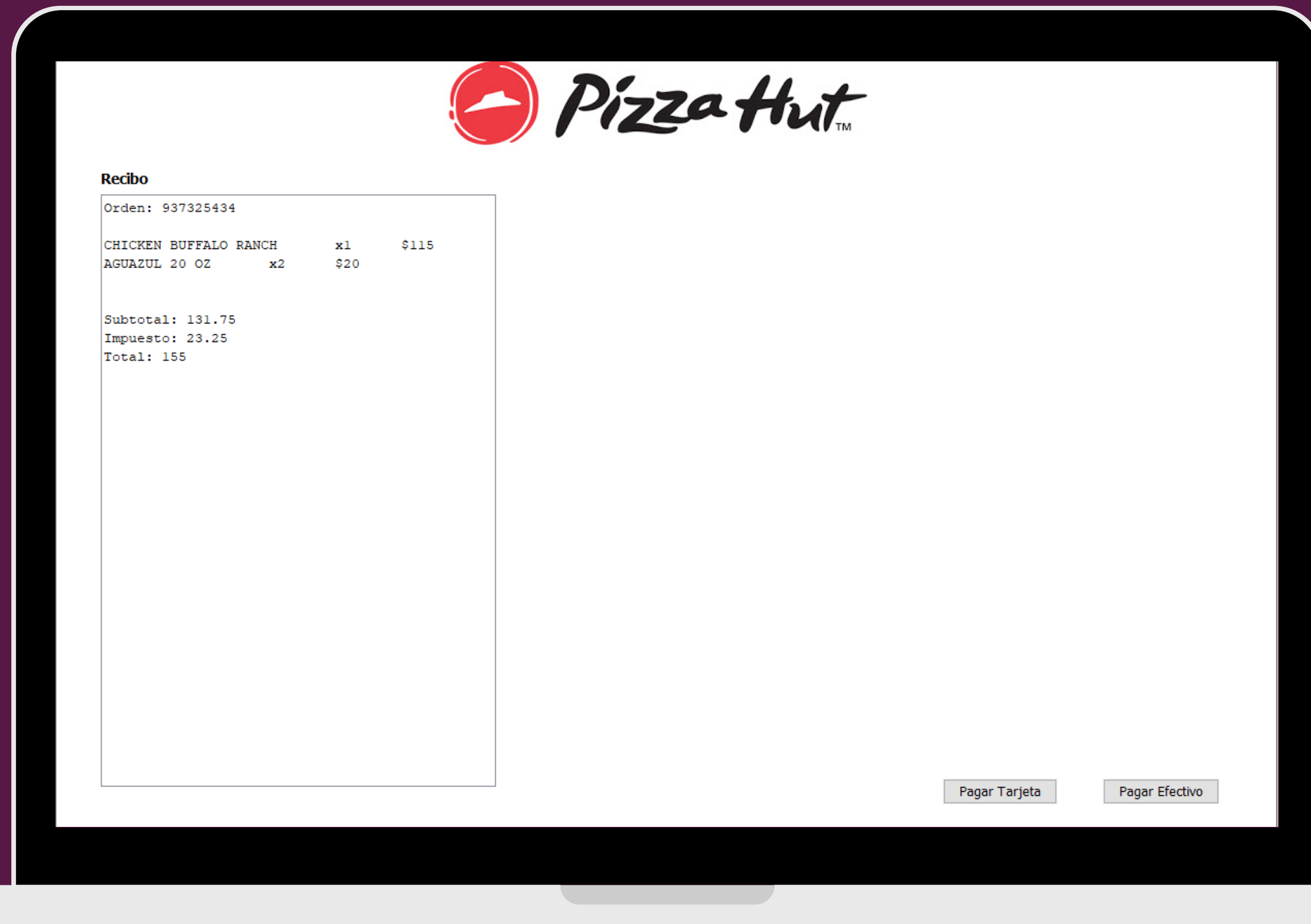
Al ingresar a alguna pestaña se visualiza las opciones disponibles con descripción, precio, imagen y cantidad a llevar.

# Detalle de orden

Lista de  
productos a  
comprar.



# Visualizacion Factura



Total de factura  
con resumen de  
la orden y el  
impuesto  
incluido.

# Código Ordenes

```
private void b_buffaloMouseClicked(java.awt.event.MouseEvent evt) {  
    String nombre = "CHICKEN BUFFALO RANCH";  
    int precio = 115;  
    int cantidad = (int) s_buffalo.getValue();  
    PreparedStatement ps,psl;  
    ResultSet rs;  
    int cant2=0;  
  
    try {  
        Connection con=Conexion.getConnection();  
        ps=con.prepareStatement("SELECT Cantidad FROM Inventario WHERE Nombre='Chiken Buffalo'");  
        rs=ps.executeQuery();  
        while(rs.next()){  
            cant2=rs.getInt("Cantidad");  
        }  
        if(cant2>=cantidad){  
            Orden o = new Orden(nombre, precio, cantidad);  
            pedidos.add(o);  
            JOptionPane.showMessageDialog(j_pizzas, "Se agregaron " + cantidad + " " + nombre + " a la orden!");  
  
            s_buffalo.setValue(1);  
  
            con=Conexion.getConnection();  
            psl=con.prepareStatement("UPDATE Inventario SET Cantidad=? WHERE Nombre='Chiken Buffalo' ");  
            psl.setInt(1, cant2-cantidad);  
            psl.executeUpdate();  
        }else{  
            JOptionPane.showMessageDialog(j_pizzas, "CANTIDAD INVALIDA\nNo hay la cantidad de producto que desea"  
        }  
    }  
  
    } catch (Exception e) {  
    }  
}
```

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    JOptionPane.showMessageDialog(j_parallevar, "Su orden se ha enviado");  
    try {  
        Connection con=Conexion.getConnection();  
        PreparedStatement ps;  
  
        for (int i = 0; i < pedidos.size(); i++) {  
            ps=con.prepareStatement("INSERT INTO DETALLE_PEDIDO (ID_detalle,Nombre_Producto,Cantidad,Precio,"  
                + "Estado,Categoria,ID_mesa) VALUES(?, ?, ?, ?, ?, ?, ?)");  
            ps.setInt(1, id_pedido_actual);  
            ps.setString(2, pedidos.get(i).nombre);  
            ps.setInt(3, pedidos.get(i).getCantidad());  
            ps.setDouble(4, (double)pedidos.get(i).getPrecio());  
            ps.setInt(5, 1);  
            ps.setString(6, "Comer en restaurante");  
            ps.setInt(7, cont_pedido_mesa);  
  
            ps.executeUpdate();  
        }  
    } catch (Exception e) {  
    }  
}
```

*Thank  
you!*