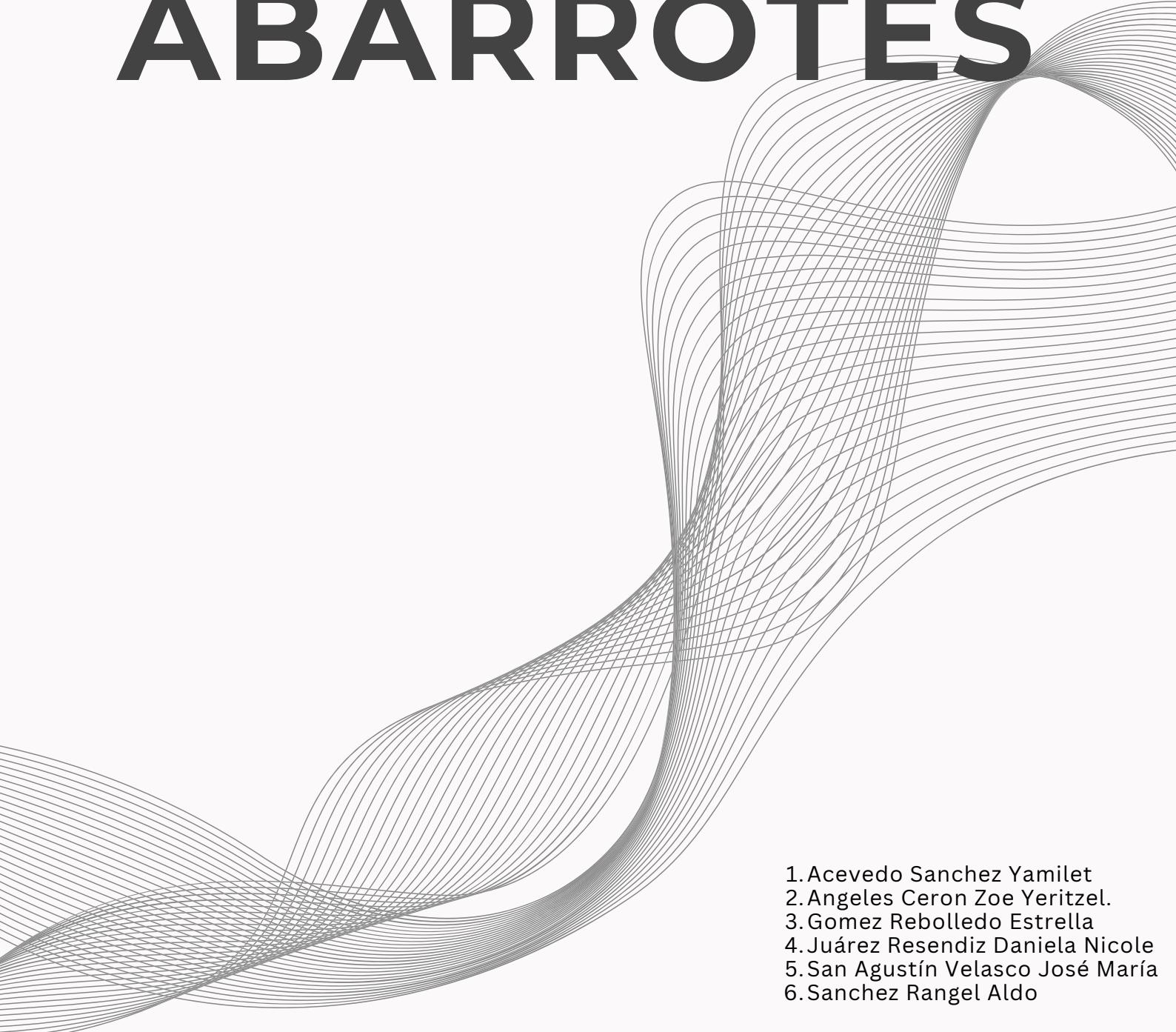


PROYECTO FINAL

# TIENDA DE ABARROTES

- 
- A large, abstract graphic element consisting of numerous thin, dark grey lines that curve and overlap to create a sense of depth and motion, resembling waves or architectural structures.
- 1. Acevedo Sanchez Yamilet
  - 2. Angeles Ceron Zoe Yeritzel.
  - 3. Gomez Rebollo Estrella
  - 4. Juárez Resendiz Daniela Nicole
  - 5. San Agustín Velasco José María
  - 6. Sanchez Rangel Aldo

# ÍNDICE

1. Descripción	3
2. Diagramas UML	4
2.1 Diagrama de objetos	4
2.2 Diagrama de clases	5
2.3 Diagrama de casos de uso (empleado y cliente)	6
2.4 Diagrama de casos de uso (proveedor)	7
3. Diseño de la base de datos	8
3.1 Diagrama Entidad-Relación	8
3.2 Diccionario de Datos	9
3.2.1 Tabla de clientes	9
3.2.2 Tabla de empleados	9
3.2.3 Tabla de productos	9
3.2.4 Tabla de proveedor	10
3.2.5 Tabla de ventas	10

# 1. DESCRIPCIÓN



## Proveedor y vendedor:

El proveedor es quien abastece los productos a la tienda, asegurando que haya suficiente mercancía para la venta. El vendedor es quien atiende al cliente, utiliza el sistema para registrar las ventas y mantener actualizado el inventario.



## Consumidor:

El consumidor es la persona que compra los productos de la tienda. Gracias al sistema, recibe una atención más rápida y precisa, ya que los precios y existencias están actualizados.



## Registro de ventas:

El sistema permite registrar cada venta realizada, mostrando la fecha, el producto y el total vendido. Esto ayuda a llevar un control diario de ingresos y a conocer qué productos se venden más.

## DESCRIPCIÓN GENERAL

El Sistema de Gestión para una Tienda de Abarrotes es una herramienta **creada para** facilitar la administración y el control de los productos de consumo básico. Su **función principal** es ayudar al dueño o encargado de la tienda a llevar un registro ordenado de los artículos que tiene disponibles, las ventas que realiza cada día y el inventario que queda en existencia.

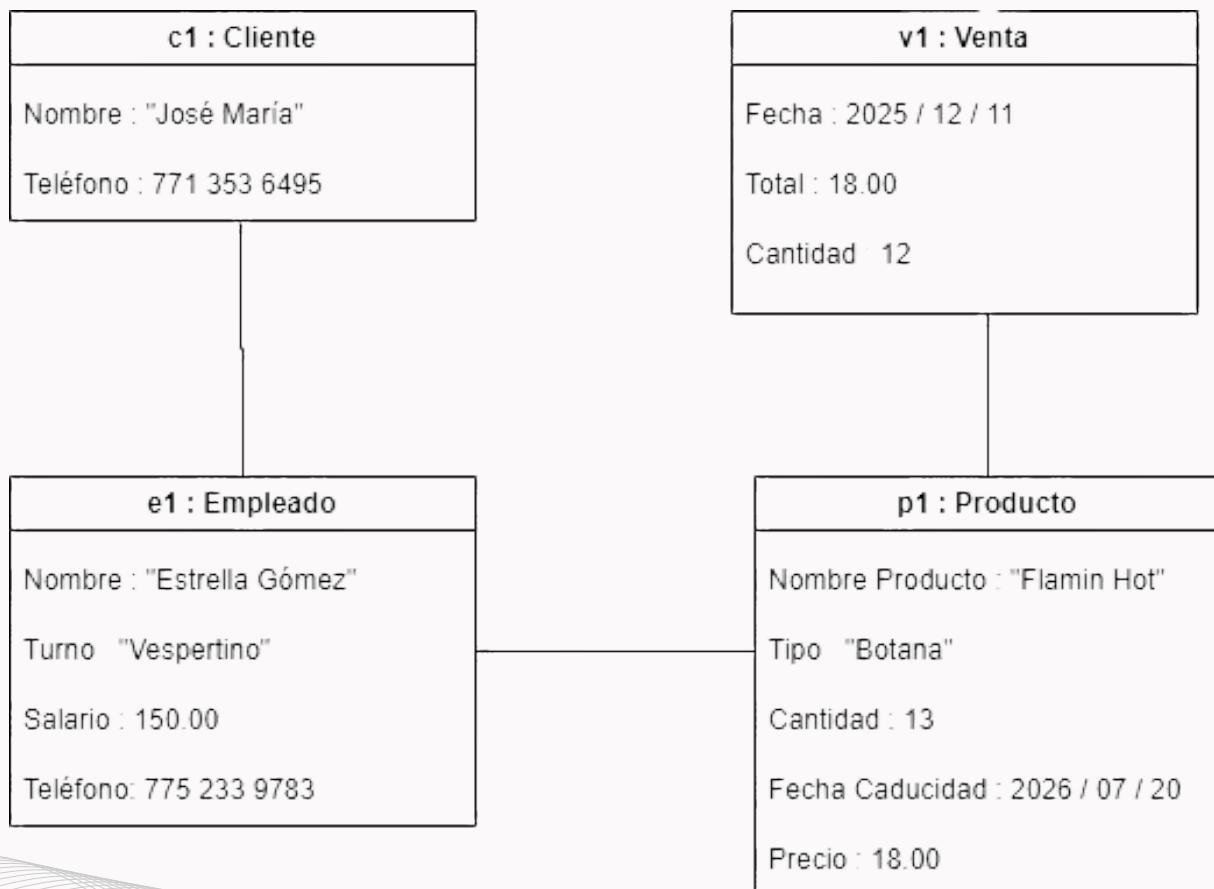
Con este sistema se **puede registrar** la información de **cada producto**, como su nombre, precio, cantidad disponible y proveedor. Además, permite registrar las ventas diarias, calcular los ingresos y actualizar automáticamente el inventario, lo que hace más fácil la organización y el control del negocio.



# 2. DIAGRAMAS UML

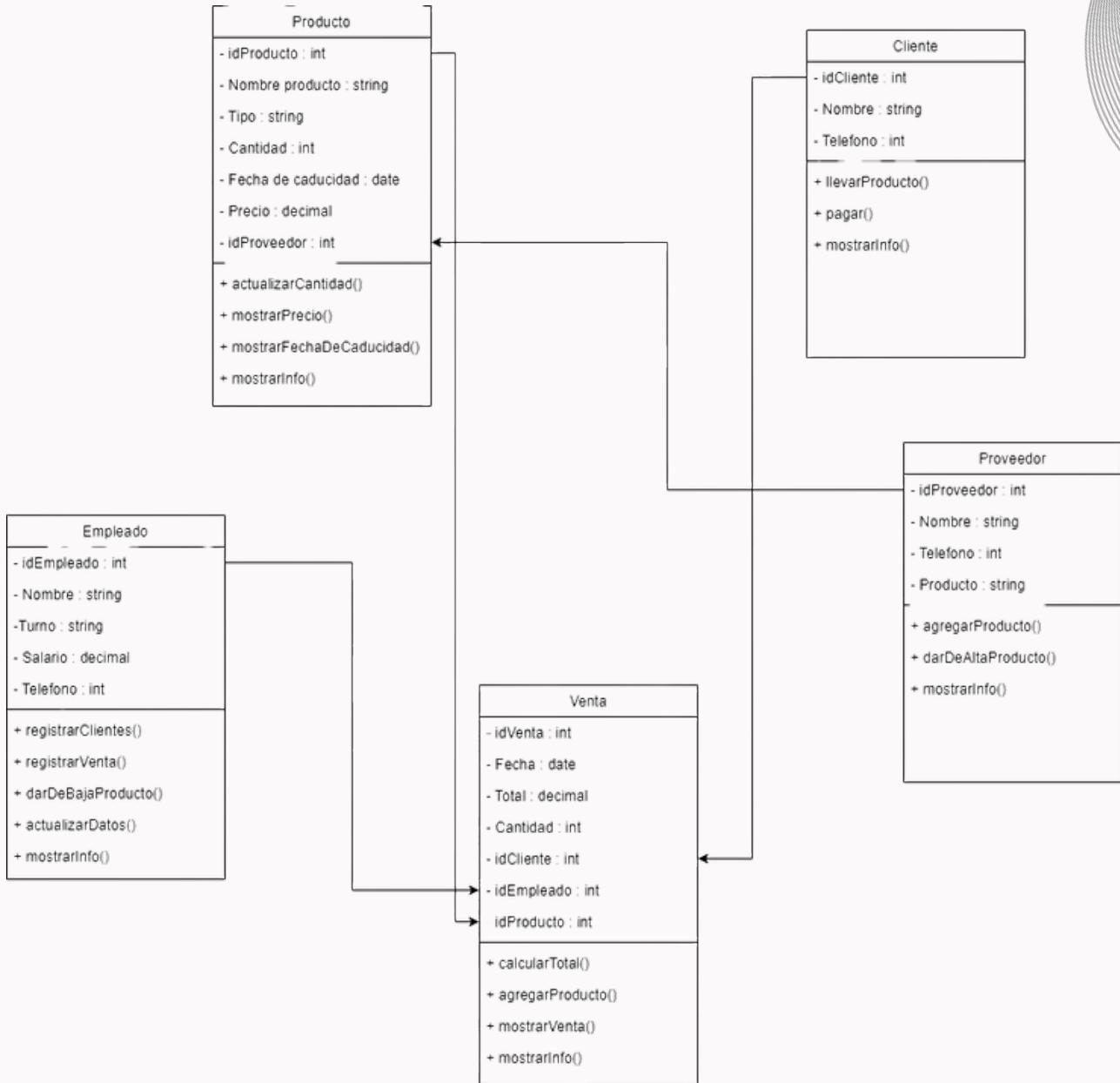
## 2.1 DIAGRAMA DE OBJETOS

Ejemplo de instancias en un escenario real



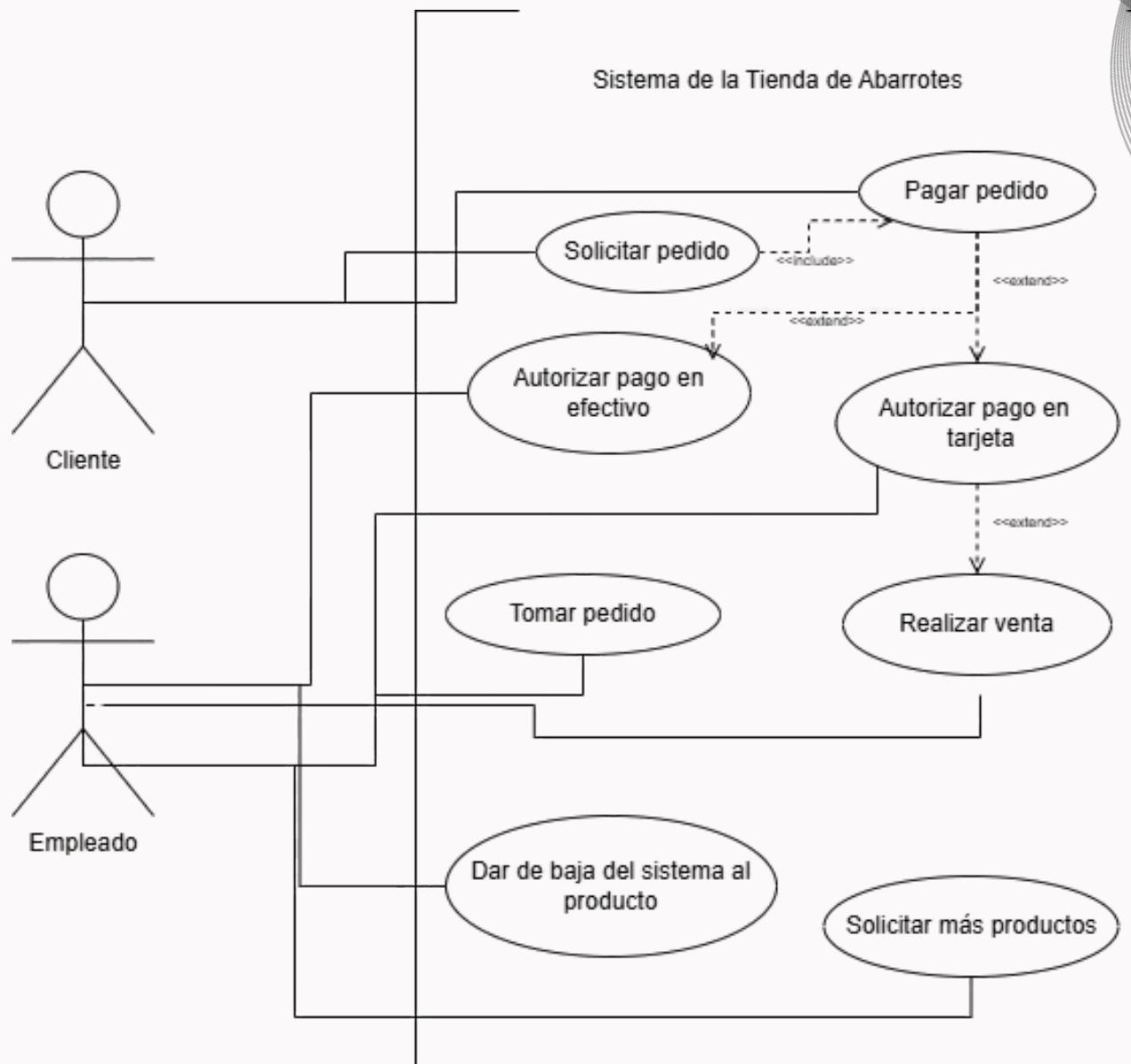
## 2.2 DIAGRAMA DE CLASES

Diagrama con atributos y casos principales



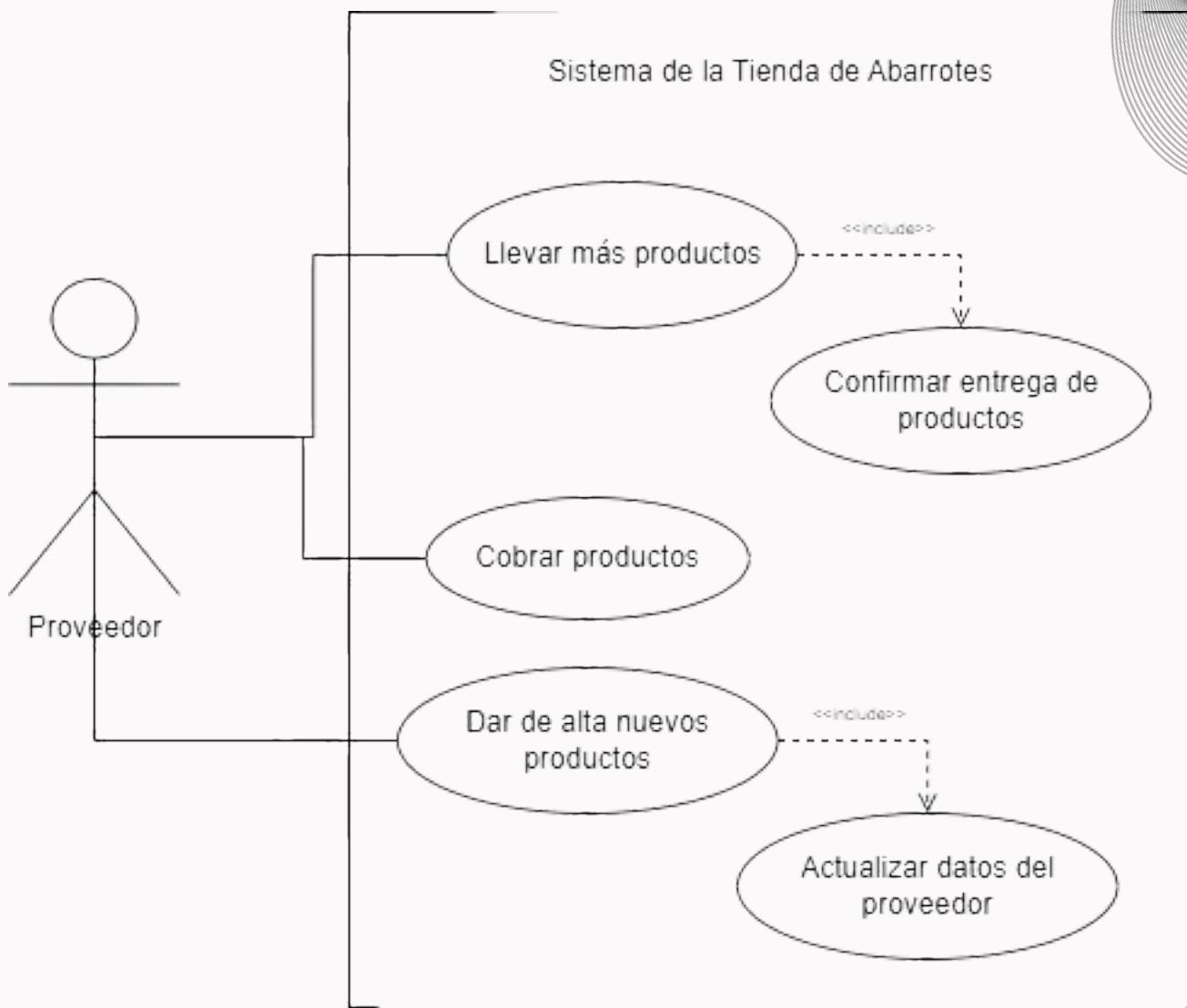
## 2.3 DIAGRAMA DE CASOS DE USO

Diagrama cliente empleado



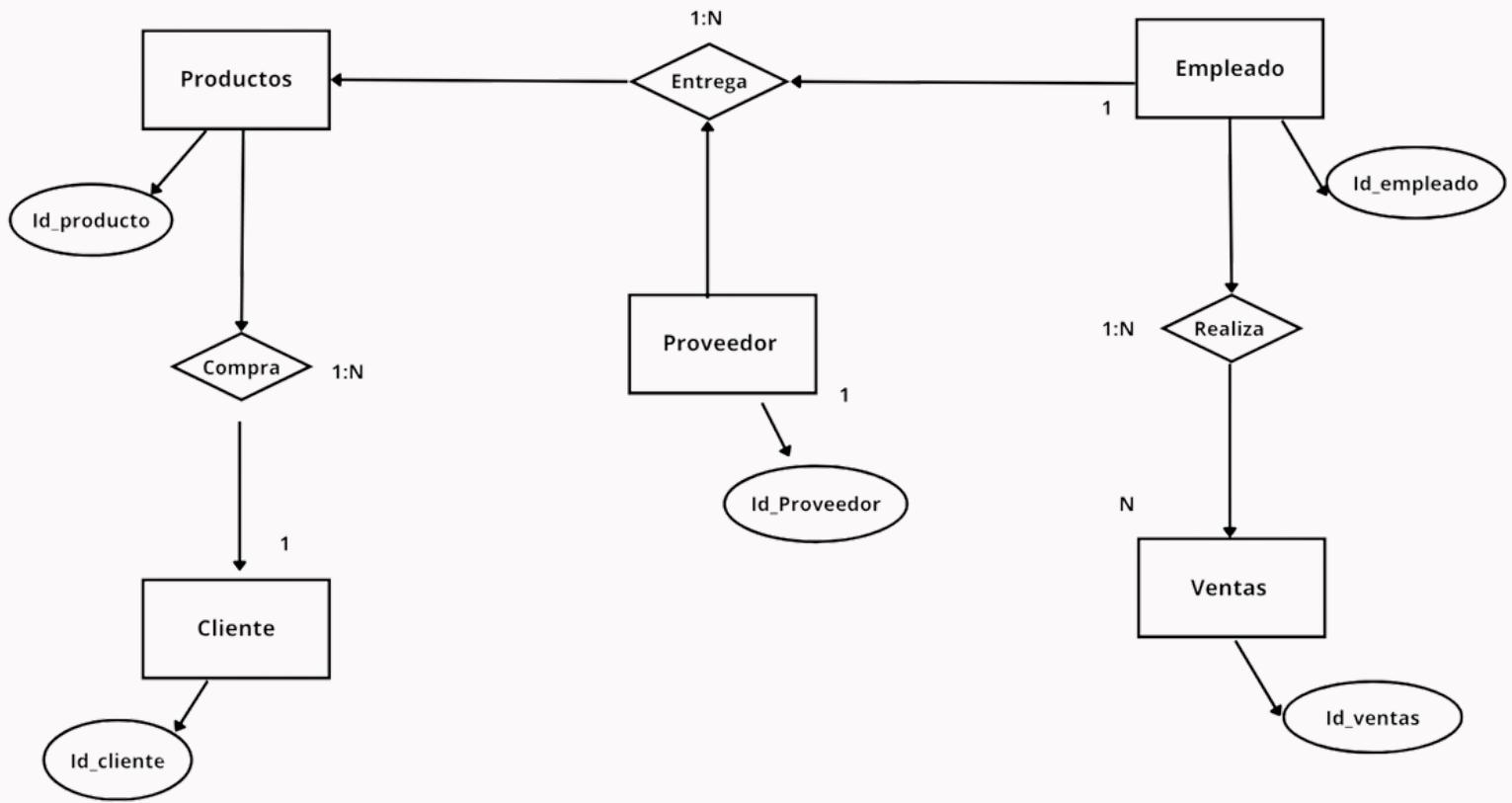
## 2.3 DIAGRAMA DE CASOS DE USO

### Diagrama del proveedor



# 3. DISEÑO DE LA BD

## 3.1 DIAGRAMA ENTIDAD RELACIÓN



## 3.2 DICCIONARIO DE DATOS

### 3.2.1 Tabla de Clientes

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 <b>idCliente</b>	int(50)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 <b>Nombre</b>	varchar(50)	utf8mb4_general_ci		No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 <b>Teléfono</b>	decimal(50,0)			No	Ninguna		Cambiar  Eliminar  Más	

### 3.2.2 Tabla de Empleados

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 <b>idEmpleado</b>	int(50)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 <b>Nombre</b>	varchar(100)	utf8mb4_general_ci		No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 <b>Turno</b>	varchar(100)	utf8mb4_general_ci		No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 <b>Salario</b>	decimal(50,0)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 <b>Telefono</b>	double			No	Ninguna		Cambiar  Eliminar  Más	

### 3.2.3 Tabla de Productos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 <b>idProducto</b>	int(50)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 <b>Producto</b>	varchar(100)	utf8mb4_general_ci		No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 <b>Tipo</b>	varchar(50)	utf8mb4_general_ci		No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 <b>Cantidad</b>	int(50)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 <b>Fecha de Caducidad</b>	date			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	6 <b>Precio</b>	decimal(10,2)			No	Ninguna		Cambiar  Eliminar  Más	
<input type="checkbox"/>	7 <b>idProveedor</b>	int(50)			No	Ninguna		Cambiar  Eliminar  Más	

### 3.2.4 Tabla de Proveedor

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 <b>idProveedor</b>	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 <b>Nombre</b>	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 <b>Telefono</b>	double			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 <b>Producto</b>	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	

### 3.2.5 Tabla de Ventas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 <b>idVenta</b>	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 <b>Fecha</b>	date			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 <b>Total</b>	decimal(50,0)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 <b>Cantidad</b>	int(100)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 <b>idCliente</b>	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	6 <b>idEmpleado</b>	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	7 <b>idProducto</b>	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	

### **3.3 SCRIPT DE LA BD**

---

# 4. CÓDGO RELEVANTE

## Y OPERACIONES CRUD

### 4.1 CONEXIONES A LA BD

#### 4.1.1 Conexión a la tabla Cliente

```
python conexionAbarrotesBDClientes.py ×  
1 import pymysql  
2 try:  
3     conexion = pymysql.connect(  
4         host = "localhost",  
5         user = "root",  
6         password = "",  
7         database = "tiendadeabarrotes"  
8     )  
9     print("Conexión exitosa :)")  
10  
11     consulta = conexion.cursor()  
12     textoConsulta = "SELECT * FROM clientes"  
13     consulta.execute(textoConsulta)  
14     resultado = consulta.fetchall()  
15     for x in resultado:  
16         print(x)  
17  
18 except Exception as err:  
19     print("Hubo un error en la conexión",err)
```

#### 4.1.2 Conexión a la tabla Empleados

```
py conexionAbarrotesBDEmpleados.py ×

1 import pymysql
2 try:
3     conexion = pymysql.connect(
4         host = "localhost",
5         user = "root",
6         password = "",
7         database = "tiendadeabarros"
8     )
9     print("Conexión exitosa :)")
10
11     consulta = conexion.cursor()
12     textoConsulta = "SELECT * FROM empleados"
13     consulta.execute(textoConsulta)
14     resultado = consulta.fetchall()
15     for x in resultado:
16         print(x)
17
18 except Exception as err:
19     print("Hubo un error en la conexión",err)
```

#### 4.1.3 Conexión a la tabla Productos

```
conectionAbarrotesBDProductos.py ✘
1 import pymysql
2 try:
3     connection = pymysql.connect(
4         host = "localhost",
5         user = "root",
6         password = "",
7         database = "tiendadeabarros"
8     )
9     print("Conexión exitosa :)")
10
11     consulta = connection.cursor()
12     textoConsulta = "SELECT * FROM productosdelatienda"
13     consulta.execute(textoConsulta)
14     resultado = consulta.fetchall()
15     for x in resultado:
16         print(x)
17
18 except Exception as err:
19     print("Hubo un error en la conexión",err)
```

#### 4.1.4 Conexión a la tabla Proveedor

```
pyc conexionAbarrotesBDProveedor.py ×
1 import pymysql
2
3
4 try:
5     conexion = pymysql.connect(
6         host = "localhost",
7         user = "root",
8         password = "",
9         database = "tiendadeabarros"
10    )
11    print("Conexión exitosa :)")
12
13    consulta = conexion.cursor()
14    textoConsulta = "SELECT * FROM proveedor"
15    consulta.execute(textoConsulta)
16    resultado = consulta.fetchall()
17    for x in resultado:
18        print(x)
19
20    except Exception as err:
21        print("Hubo un error en la conexión",err)
```

#### 4.1.5 Conexión a la tabla Ventas

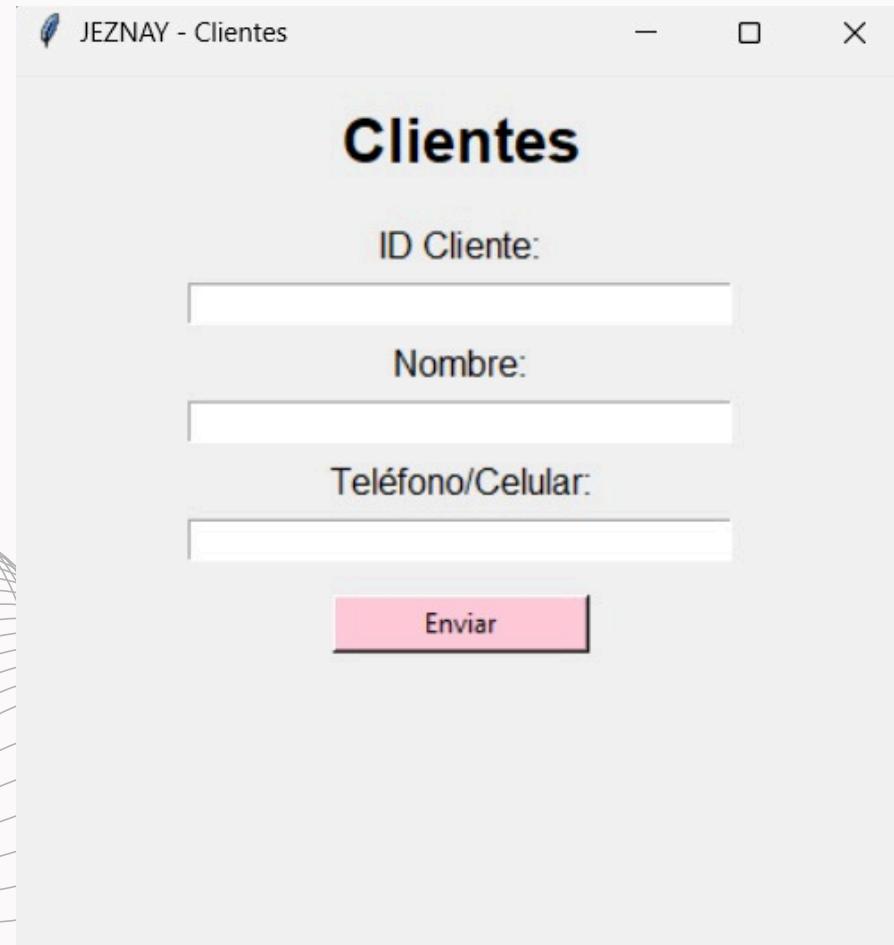
```
pyc conexionAbarrotesBDVentas.py ×

1 import pymysql
2
3
4 try:
5     conexion = pymysql.connect(
6         host = "localhost",
7         user = "root",
8         password = "",
9         database = "tiendadeabarrotes"
10    )
11    print("Conexión exitosa :)")
12
13    consulta = conexion.cursor()
14    textoConsulta = ("SELECT * FROM tablaventas")
15    consulta.execute(textoConsulta)
16    resultado = consulta.fetchall()
17    for x in resultado:
18        print(x)
19
20 except Exception as err:
21     print("Hubo un error en la conexión",err)
```

## 4.2 CÓDIGO DE INTERFÁZ

### 4.2.1 Interfaz y Código : Cliente

```
clientes.py ×  
1 import tkinter as tk  
2  
3 root = tk.Tk()  
4 root.title("JEZNAY - Clientes")  
5 root.geometry("400x400")  
6  
7 tk.Label(root, text="Clientes", font=("Arial", 20, "bold")).pack(pady=15)  
8  
9 campos = ["ID Cliente", "Nombre", "Teléfono/Celular"]  
10 for campo in campos:  
11     tk.Label(root, text=campo + ":", font=("Arial", 12)).pack()  
12     tk.Entry(root, width=40).pack(pady=5)  
13  
14 tk.Button(root, text="Enviar", bg="black", fg="white", width=15).pack(pady=10)  
15  
16 root.mainloop()
```



## 4.2.2 Interfaz y Código : Empleados

```
Empleados.py ×
1 import tkinter as tk
2
3 root = tk.Tk()
4 root.title("JEZNAY - Empleados")
5 root.geometry("400x400")
6
7 tk.Label(root, text="Empleados", font=("Arial", 20, "bold")).pack(pady=15)
8
9 campos = ["ID Empleado", "Nombre", "Turno", "Salario"]
10 for campo in campos:
11     tk.Label(root, text=campo + ":", font=("Arial", 12)).pack()
12     tk.Entry(root, width=40).pack(pady=5)
13
14 tk.Button(root, text="Enviar", bg="black", fg="white", width=15).pack(pady=10)
15
16 root.mainloop()
```



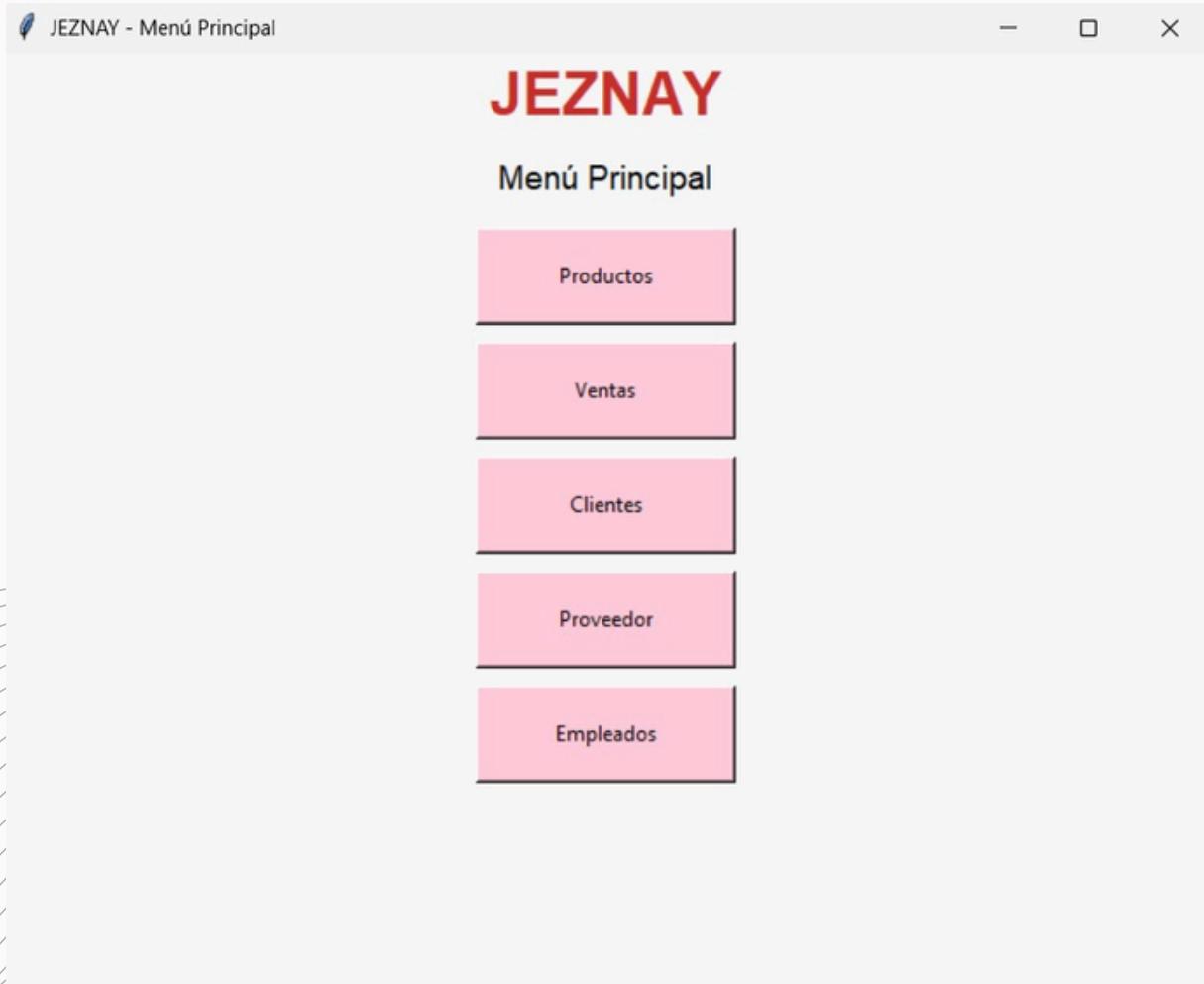
### 4.2.3 Interfaz y Código : Login

```
LogIn.py x
1 import tkinter as tk
2 from tkinter import messagebox
3
4 def verificar_login(): usage & Estrella Gomez
5     usuario = entry_usuario.get()
6     contrasena = entry_contrasena.get()
7
8     if usuario == "JEZNAY" and contrasena == "1234":
9         messagebox.showinfo( title: "Acceso concedido", message: ";Bienvenido, admin!")
10        root.destroy()
11    else:
12        messagebox.showerror( title: "Error", message: "Usuario o contraseña incorrectos")
13
14 root = tk.Tk()
15 root.title("JEZNAY - Login")
16 root.geometry("600x500")
17 root.geometry("350x350")
18 root.resizable( width: False, height: False)
19 root.configure(bg="#fdf1f1")
20
21 tk.Label(root, text="Inicio de Sesión \n JEZNAY", font=("Arial", 18, "bold"), bg="#fdf1f1").pack(pady=15)
22
23 tk.Label(root, text="Usuario:", font=("Arial", 12), bg="#fdf1f1").pack(pady=5)
24 entry_usuario = tk.Entry(root, width=30)
25 entry_usuario.pack(pady=5)
26
27 tk.Label(root, text="Contraseña:", font=("Arial", 12), bg="#fdf1f1").pack(pady=5)
28 entry_contrasena = tk.Entry(root, width=30, show="*")
29 entry_contrasena.pack(pady=5)
30
31 tk.Button(root, text="Ingresar", bg="#e74c3c", fg="white", font=("Arial", 11, "bold"),
32 width=20, command=verificar_login).pack(pady=20)
33
34 root.mainloop()
```



#### 4.2.4 Interfaz y Código : Menú

```
Menu.py ×  
1 import tkinter as tk  
2  
3 root = tk.Tk()  
4 root.title("JEZNAY - Menú Principal")  
5 root.geometry("600x500")  
6 root.configure(bg="#f7f7f7")  
7  
8 tk.Label(root, text="JEZNAY", font=("Arial", 26, "bold"), fg="#c9302c", bg="#f7f7f7").pack(pady=30)  
9 tk.Label(root, text="Menú Principal", font=("Arial", 14), bg="#f7f7f7").pack(pady=10)  
10  
11 botones = ["Productos", "Ventas", "Clientes", "Proveedor", "Empleados"]  
12 for texto in botones:  
13     tk.Button(root, text=texto, bg="#e74c3c", fg="white", width=20, height=3).pack(pady=5)  
14  
15 root.mainloop()
```



## 4.2.5 Interfaz y Código : Productos

```
Products.py x
1  import tkinter as tk
2  from tkinter import ttk
3
4  root = tk.Tk()
5  root.title("JEZNAY - Productos")
6  root.geometry("400x450")
7
8  tk.Label(root, text="Producto", font=("Arial", 20, "bold")).pack(pady=15)
9
10 tk.Label(root, text="Nombre del producto:", font=("Arial", 12)).pack()
11 tk.Entry(root, width=40).pack(pady=5)
12
13 tk.Label(root, text="Tipo de producto:", font=("Arial", 12)).pack()
14 tipos = ["Abarrotes", "Bebidas", "Botanas", "Lácteos", "Productos de Limpieza", "Higiene Personal"]
15 tipo_var = tk.StringVar()
16 combo_tipo = ttk.Combobox(root, textvariable=tipo_var, values=tipos, state="readonly", width=37)
17 combo_tipo.pack(pady=5)
18
19 tk.Label(root, text="Cantidad:", font=("Arial", 12)).pack()
20 tk.Entry(root, width=40).pack(pady=5)
21
22 tk.Label(root, text="Fecha de caducidad:", font=("Arial", 12)).pack()
23 tk.Entry(root, width=40).pack(pady=5)
24
25 tk.Button(root, text="Enviar", bg="black", fg="white", width=15).pack(pady=10)
26
27 root.mainloop()
```



## 4.2.6 Interfaz y Código : Proveedor

```
proveedor.py ×
1  import tkinter as tk
2  from tkinter import ttk
3
4  root = tk.Tk()
5  root.title("JEZNAY - Proveedor")
6  root.geometry("400x480")
7
8  tk.Label(root, text="Proveedor", font=("Arial", 20, "bold")).pack(pady=15)
9
10 tk.Label(root, text="ID Proveedor:", font=("Arial", 12)).pack()
11 tk.Entry(root, width=40).pack(pady=5)
12
13 tk.Label(root, text="Nombre:", font=("Arial", 12)).pack()
14 tk.Entry(root, width=40).pack(pady=5)
15
16 tk.Label(root, text="Teléfono:", font=("Arial", 12)).pack()
17 tk.Entry(root, width=40).pack(pady=5)
18
19 tk.Label(root, text="Tipo de producto:", font=("Arial", 12)).pack()
20 tipos = ["Abarrotes", "Bebidas", "Botanas", "Lácteos", "Productos de Limpieza", "Higiene Personal"]
21 tipo_var = tk.StringVar()
22 combo_tipo = ttk.Combobox(root, textvariable=tipo_var, values=tipos, state="readonly", width=37)
23 combo_tipo.pack(pady=5)
24
25 tk.Label(root, text="Producto que provee:", font=("Arial", 12)).pack()
26 tk.Entry(root, width=40).pack(pady=5)
27
28 tk.Button(root, text="Enviar", bg="black", fg="white", width=15).pack(pady=10)
29
30 root.mainloop()
```



#### 4.2.7 Interfaz y Código : Ventas

```
 Ventas.py x
1  import tkinter as tk
2
3  root = tk.Tk()
4  root.title("JEZNAY - Ventas")
5  root.geometry("400x400")
6
7  tk.Label(root, text="Ventas", font=("Arial", 20, "bold")).pack(pady=15)
8
9  campos = ["Nom. Venta", "Fecha", "Total", "Cantidad"]
10 for campo in campos:
11     tk.Label(root, text=campo + ":", font=("Arial", 12)).pack()
12     tk.Entry(root, width=40).pack(pady=5)
13
14 tk.Button(root, text="Enviar", bg="black", fg="white", width=15).pack(pady=10)
15
16 root.mainloop()
```

