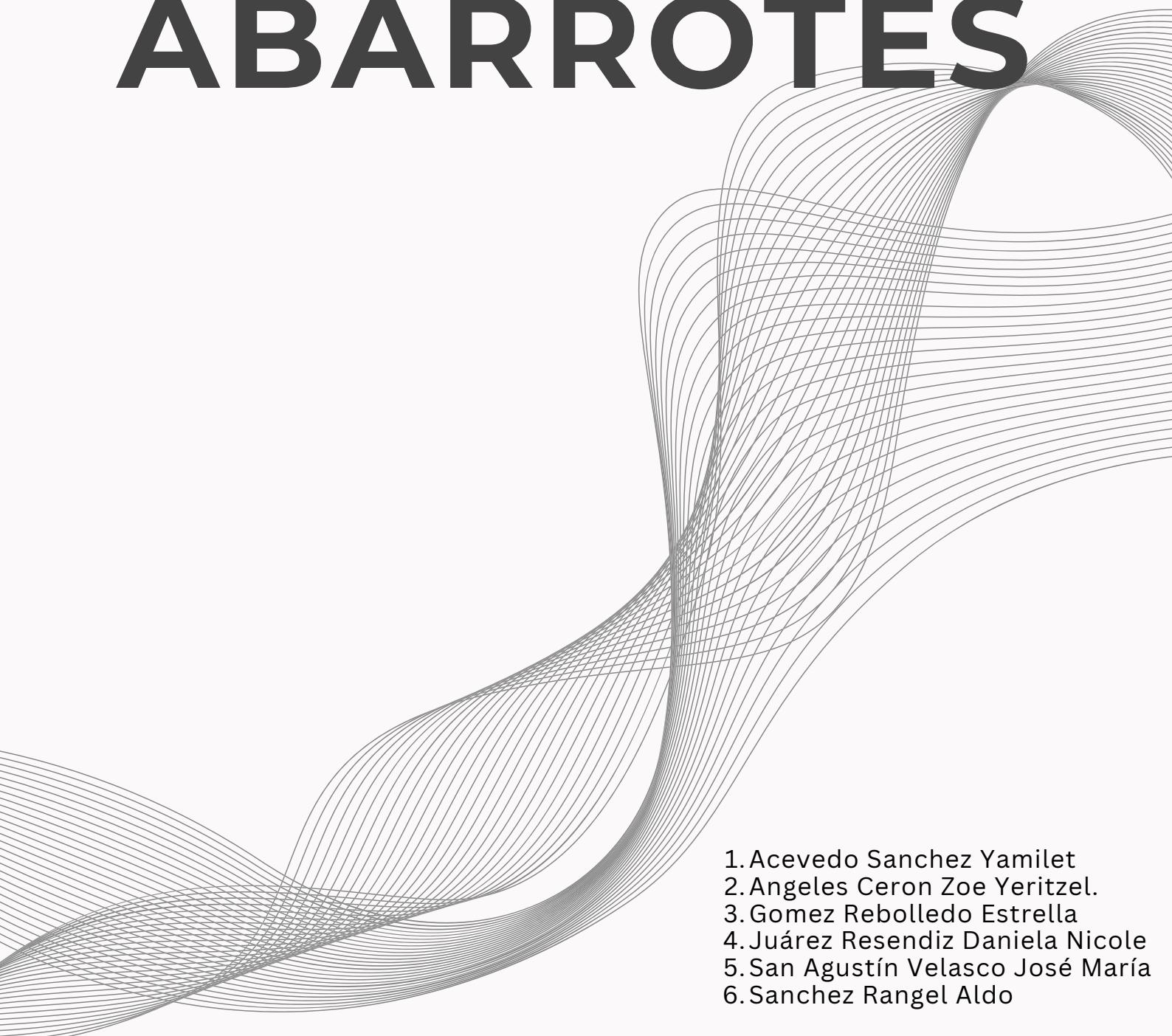


PROYECTO FINAL

TIENDA DE ABARROTES

- 
- 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
3.3 Script de la Base de Datos	11
3.3.1 Script Tabla Clientes	11 , 12
3.3.2 Script Tabla Empleados	12, 13
3.3.3 Script Tabla Ventas	14,15
3.3.4 Script Tabla Productos	15,16
3.3.5 Script Tabla Proveedor	17,18
4. Código relevante y operaciones crud	19
4.1.1 Conexión a la tabla Cliente	19
4.1.2 Conexión a la tabla Empleados	20
4.1.3 Conexión a la tabla Productos	21
4.1.4 Conexión a la tabla Proveedor	22
4.1.5 Conexión a la tabla Ventas	23
4.2 Código de interfáz	24
4.2.1 Interfaz y Código : Cliente	24
4.2.2 Interfaz y Código : Empleados	25
4.2.3 Interfaz y Código : Login	26
4.2.4 Interfaz y Código : Menú	27
4.2.5 Interfaz y Código : Productos	28
4.2.6 Interfaz y Código : Proveedor	29
4.2.7 Interfaz y Código : Ventas	30
5. Backlog	31
6. Repositorio de Git	31
7. Anexos	32
8. Conclusiones finales	33

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.

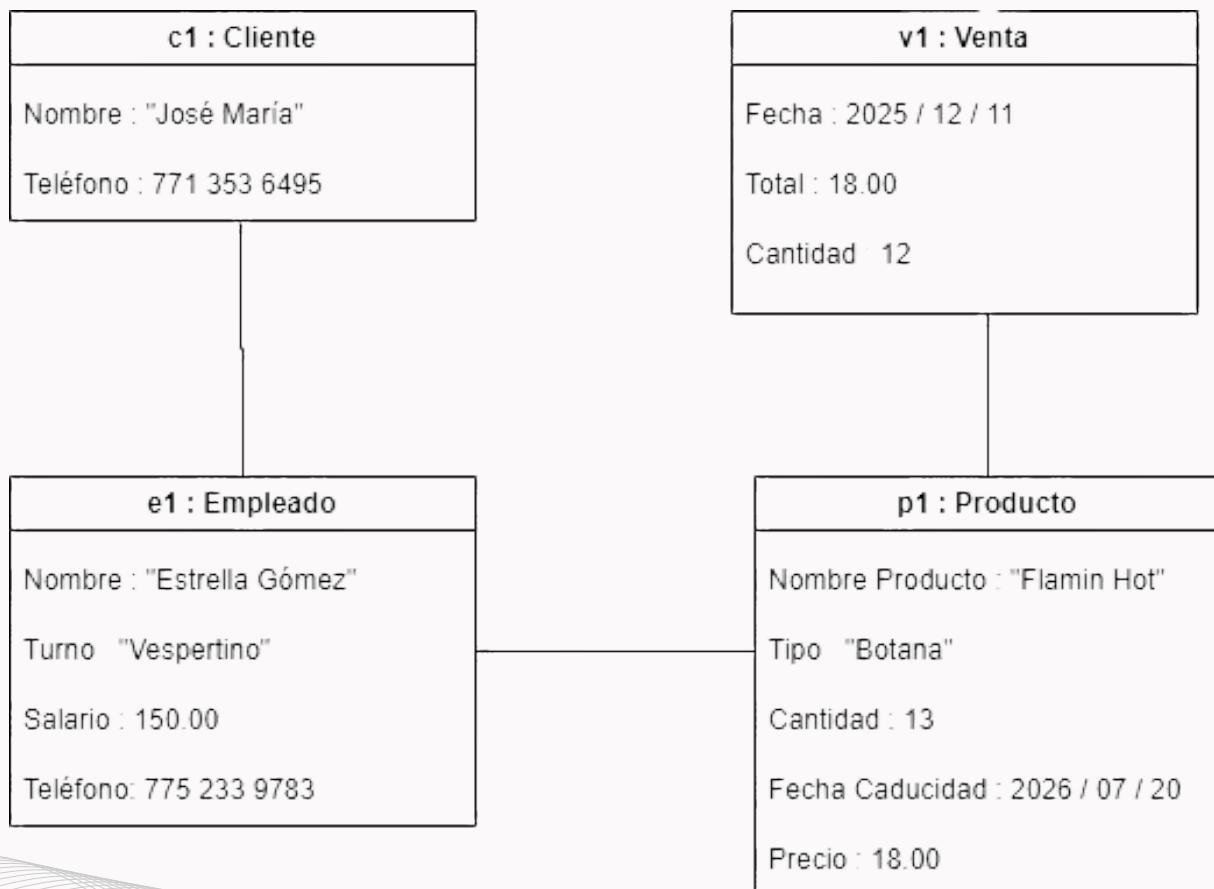
Con este sistema se **puede registrar** la información de **cada producto**, como su nombre, precio, cantidad disponible y proveedor. Además, permite registrar las ventas diarias, calcular los ingresos y actualizar automáticamente el inventario, lo que hace más fácil la organización y el control del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



2. DIAGRAMAS UML

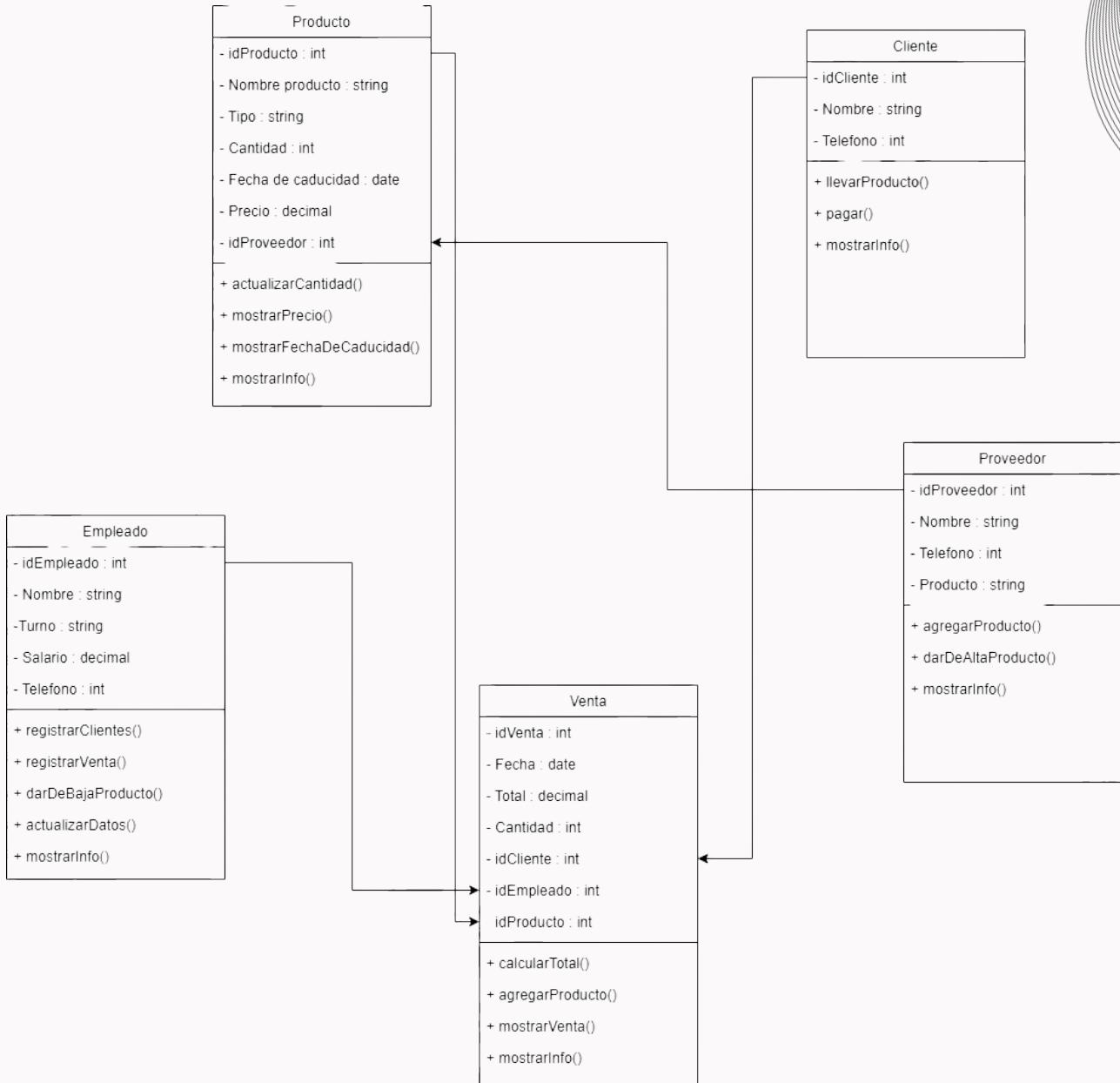
2.1 DIAGRAMA DE OBJETOS

Ejemplo de instancias en un escenario real



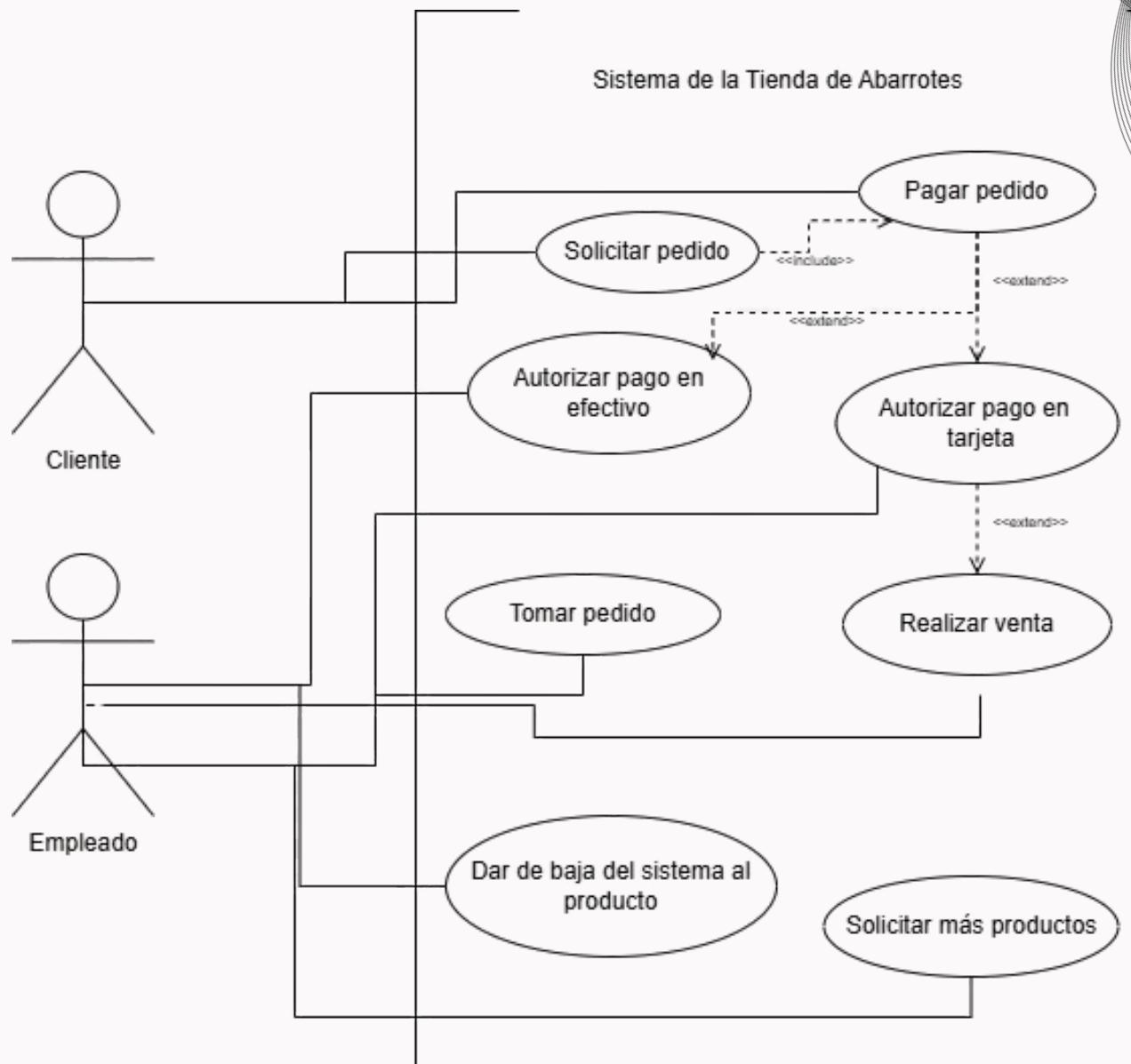
2.2 DIAGRAMA DE CLASES

Diagrama con atributos y casos principales



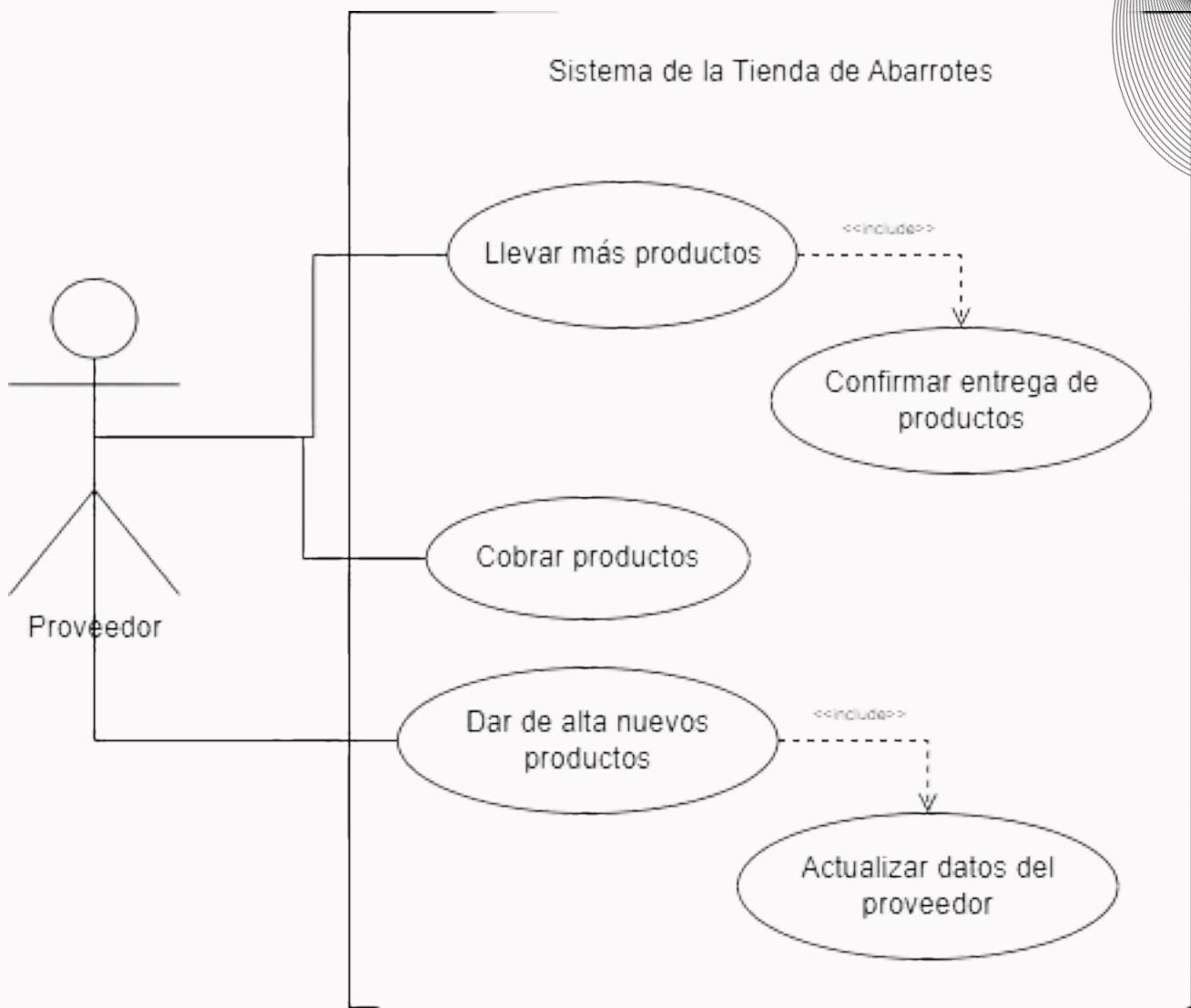
2.3 DIAGRAMA DE CASOS DE USO

Diagrama cliente empleado



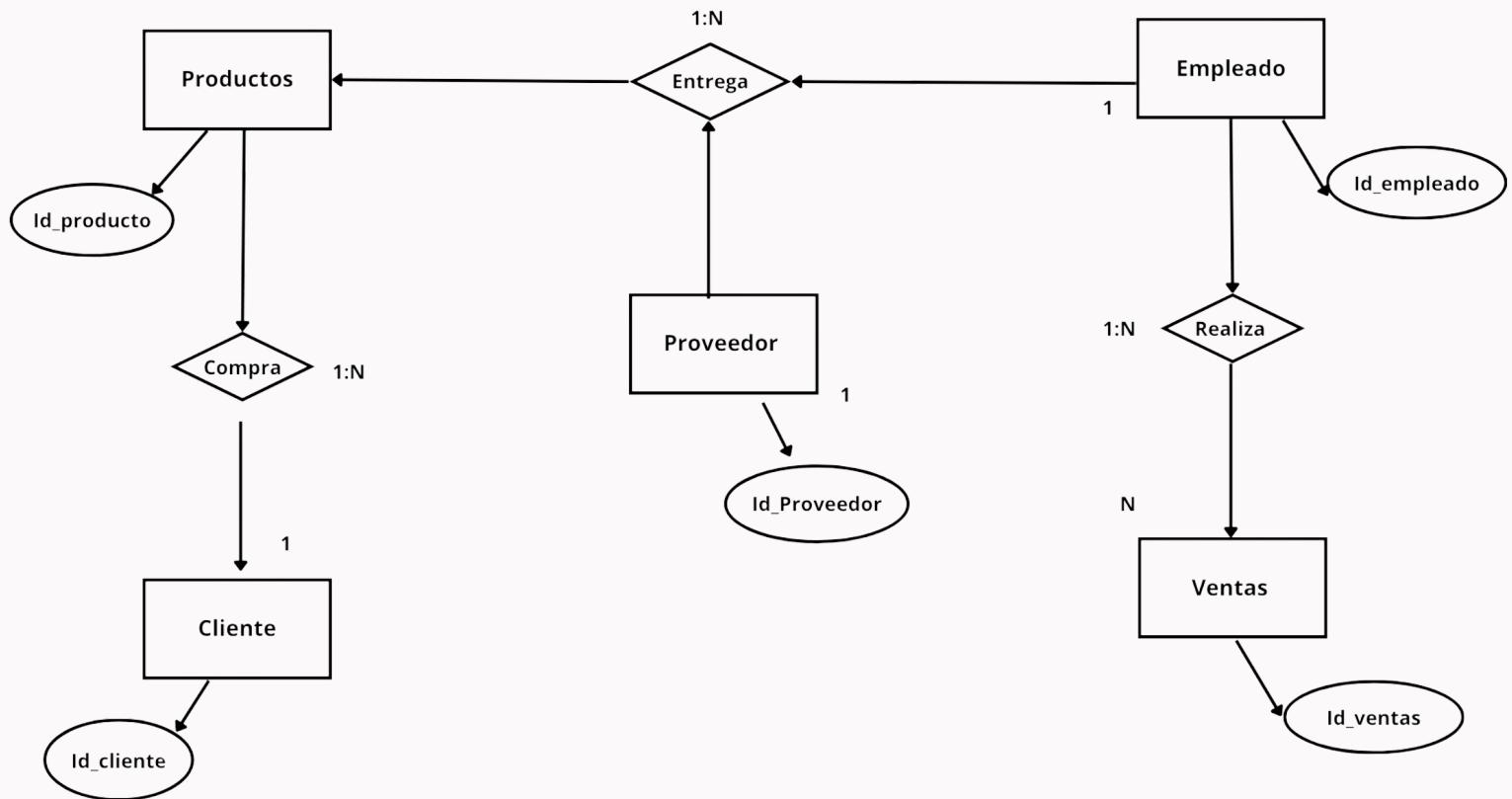
2.3 DIAGRAMA DE CASOS DE USO

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 idCliente	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 Nombre	varchar(50)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 Teléfono	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 idEmpleado	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 Nombre	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 Turno	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 Salario	decimal(50,0)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 Telefono	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 idProducto	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 Producto	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 Tipo	varchar(50)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 Cantidad	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 Fecha de Caducidad	date			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	6 Precio	decimal(10,2)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	7 idProveedor	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 idProveedor	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 Nombre	varchar(100)	utf8mb4_general_ci		No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 Telefono	double			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 Producto	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 idVenta	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	2 Fecha	date			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	3 Total	decimal(50,0)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	4 Cantidad	int(100)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	5 idCliente	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	6 idEmpleado	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	
<input type="checkbox"/>	7 idProducto	int(50)			No	Ninguna		 Cambiar  Eliminar  Más	

3.3 SCRIPT DE LA BD

3.3.1 Script Tabla Clientes

```
clientes <--> Limite de 1000 filas | Estrella | Pencil | Search | Print | Copy

23
24  --
25
26  --
27  -- Estructura de tabla para la tabla `clientes`
28  --
29
30 • CREATE TABLE `clientes` (
31     `idCliente` int(50) NOT NULL,
32     `Nombre` varchar(50) NOT NULL,
33     `Teléfono` decimal(50,0) NOT NULL
34 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
35
36  --
37  -- Volcado de datos para la tabla `clientes`
38  --
39
40 • INSERT INTO `clientes` (`idCliente`, `Nombre`, `Teléfono`) VALUES
41     (1, 'Ana López', 5512345678),
42     (2, 'José Martínez', 5523456789),
43     (3, 'María Hernández', 3245678311),
44     (4, 'Nicole Juárez', 5592711355),
45     (5, 'Carlos Ruiz', 5545678901),
46     (6, 'Lucia Gómez', 5567324674),
```

clientes x

Limit to 1000 rows

```

33   `Teléfono` decimal(50,0) NOT NULL
34 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
35
36 --
37 -- Volcado de datos para la tabla `clientes`
38 --
39
40 • INSERT INTO `clientes` (`idCliente`, `Nombre`, `Teléfono`) VALUES
41   (1, 'Ana López', 5512345678),
42   (2, 'José Martínez', 5523456789),
43   (3, 'María Hernández', 3245678311),
44   (4, 'Nicole Juárez', 5592711355),
45   (5, 'Carlos Ruiz', 5545678901),
46   (6, 'Lucia Gómez', 5567324674),
47   (7, 'Aldo Resendiz', 5556789012),
48   (8, 'Diana Torres', 5234556632),
49   (9, 'Sofia Vergara', 5524563223),
50   (10, 'Estrella Cebolla', 5234567891);
51 • COMMIT;
52
53 • /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
54 • /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
55 • /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
56

```

3.3.2 Script Tabla Empleados

empleados x

Limit to 1000 rows

```

1   -- phpMyAdmin SQL Dump
2   -- version 5.2.1
3   -- https://www.phpmyadmin.net/
4
5   -- Servidor: 127.0.0.1
6   -- Tiempo de generación: 17-11-2025 a las 03:48:24
7   -- Versión del servidor: 10.4.32-MariaDB
8   -- Versión de PHP: 8.2.12
9
10 • SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 • START TRANSACTION;
12 • SET time_zone = "+00:00";
13
14
15 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 • /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Base de datos: `tiendadeabarros`
22 --
23
24

```

```
empleados x
Limit to 1000 rows
21 -- Base de datos: `tiendadeabarrotes`
22 --
23
24 -----
25
26 --
27 -- Estructura de tabla para la tabla `empleados`
28 --
29
30 • CREATE TABLE `empleados` (
31   `idEmpleado` int(50) NOT NULL,
32   `Nombre` varchar(100) NOT NULL,
33   `Turno` varchar(100) NOT NULL,
34   `Salario` decimal(50,0) NOT NULL,
35   `Telefono` double NOT NULL
36 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
37
38 --
39 -- Volcado de datos para la tabla `empleados`
40 --
41
42 • INSERT INTO `empleados` (`idEmpleado`, `Nombre`, `Turno`, `Salario`, `Telefono`) VALUES
43   (1, 'Juan Herrera', 'Matutino', 8500, 7751235467),
44   (2, 'Zoe Zanahorias', 'Matutino', 8500, 5246573453),
```

3.3.3 Script Tabla Ventas

tablaventas <-->

Limit to 1000 rows

```
1 -- phpMyAdmin SQL Dump
2 -- version 5.2.1
3 -- https://www.phpmyadmin.net/
4 --
5 -- Servidor: 127.0.0.1
6 -- Tiempo de generación: 17-11-2025 a las 03:49:00
7 -- Versión del servidor: 10.4.32-MariaDB
8 -- Versión de PHP: 8.2.12
9
10 • SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 • START TRANSACTION;
12 • SET time_zone = "+00:00";
13
14
15 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 • /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Base de datos: `tiendadeabarrotes`
22 --
23
24 -- -----
```

```
tablaventas x
| Limite a 1000 filas | Buscar | Navegar | Eliminar | Insertar | Actualizar | Recuperar | Exportar | Imprimir |
19
20    --
21    -- Base de datos: `tiendadeabarrotes`
22    --
23
24    -- -----
25
26    --
27    -- Estructura de tabla para la tabla `tablaventas`
28    --
29
30 • CREATE TABLE `tablaventas` (
31     `idVenta` int(50) NOT NULL,
32     `Fecha` date NOT NULL,
33     `Total` decimal(50,0) NOT NULL,
34     `Cantidad` int(100) NOT NULL,
35     `idCliente` int(50) NOT NULL,
36     `idEmpleado` int(50) NOT NULL,
37     `idProducto` int(50) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39
40    --
41    -- Volcado de datos para la tabla `tablaventas`
42    --
```

tablaventas x

```

37     `idProducto` int(50) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39
40 --
41 -- Volcado de datos para la tabla `tablaventas`
42 --
43
44 • INSERT INTO `tablaventas` (`idVenta`, `Fecha`, `Total`, `Cantidad`, `idCliente`, `idEmpleado`, `idProducto`) VALUES
45 (1, '2025-11-18', 10, 1, 1, 2, 1),
46 (2, '2025-11-11', 24, 2, 2, 1, 6),
47 (3, '2025-11-10', 18, 1, 3, 2, 2),
48 (4, '2025-11-09', 17, 1, 4, 1, 9),
49 (5, '2025-11-08', 24, 1, 5, 1, 4),
50 (6, '2025-11-15', 90, 5, 6, 3, 3),
51 (7, '2025-11-15', 15, 1, 7, 4, 7),
52 (8, '2025-11-15', 20, 1, 8, 5, 5),
53 (9, '2025-11-15', 84, 3, 9, 5, 8),
54 (10, '2025-11-02', 34, 2, 10, 4, 10);
55 • COMMIT;
56
57 • /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
58 • /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
59 • /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
60

```

3.3.4 Script Tabla Productos

productosdelatienda x

```

1 -- phpMyAdmin SQL Dump
2 -- version 5.2.1
3 -- https://www.phpmyadmin.net/
4 --
5 -- Servidor: 127.0.0.1
6 -- Tiempo de generación: 17-11-2025 a las 03:48:35
7 -- Versión del servidor: 10.4.32-MariaDB
8 -- Versión de PHP: 8.2.12
9
10 • SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 • START TRANSACTION;
12 • SET time_zone = "+00:00";
13
14
15 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 • /*!40101 SET NAMES utf8mb4 */;
19
20 --
21 -- Base de datos: `tiendadeabarrotes`
22 --
23
24 -- -----

```

productosdelatienda

```

19
20 Open a script file in this editor
21 -- Base de datos: `tiendadeabarroses`
22 --
23
24 --
25
26 --
27 -- Estructura de tabla para la tabla `productosdelatienda`
28 --
29
30 • CREATE TABLE `productosdelatienda` (
31     `idProducto` int(50) NOT NULL,
32     `Producto` varchar(100) NOT NULL,
33     `Tipo` varchar(50) NOT NULL,
34     `Cantidad` int(50) NOT NULL,
35     `Fecha de Caducidad` date NOT NULL,
36     `Precio` decimal(10,2) NOT NULL,
37     `idProveedor` int(50) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39
40 --
41 -- Volcado de datos para la tabla `productosdelatienda`
42 --

```

productosdelatienda

```

37     `idProveedor` int(50) NOT NULL
38 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
39
40 --
41 -- Volcado de datos para la tabla `productosdelatienda`
42 --
43
44 • INSERT INTO `productosdelatienda` (`idProducto`, `Producto`, `Tipo`, `Cantidad`, `Fecha de Caducidad`, `Precio`, `idPr
45 (1, 'Queso Oaxaca (1/4)', 'Abarrote', 10, '2026-10-13', 10.00, 2),
46 (2, 'Coca Cola', 'Bebida', 30, '2026-05-07', 18.00, 4),
47 (3, 'Flamin Hot', 'Botana', 13, '2026-07-20', 18.00, 2),
48 (4, 'Arroz (kg)', 'Abarrote', 20, '2026-03-09', 23.50, 3),
49 (5, 'Jugo Del Valle', 'Bebida', 20, '2026-06-18', 20.00, 4),
50 (6, 'Paleta Payaso', 'Dulce', 40, '2026-11-16', 12.00, 5),
51 (7, 'Gansito', 'Botana', 35, '2026-11-09', 15.00, 2),
52 (8, 'Frijol (kg)', 'Abarrote', 25, '2026-11-14', 28.00, 3),
53 (9, 'Pepsi', 'Bebida', 28, '2026-11-23', 17.00, 4),
54 (10, 'Pam Bimbo', 'Abarrote', 20, '2026-11-27', 17.00, 2);
55 • COMMIT;
56
57 • /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
58 • /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
59 • /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
60

```

3.3.5 Script Tabla Proveedor

```
proveedor x
Limit to 1000 rows | + | - | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24

1 -- phpMyAdmin SQL Dump
2 -- version 5.2.1
3 -- https://www.phpmyadmin.net/
4 --
5 -- Servidor: 127.0.0.1
6 -- Tiempo de generación: 17-11-2025 a las 03:48:50
7 -- Versión del servidor: 10.4.32-MariaDB
8 -- Versión de PHP: 8.2.12
9
10 • SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 • START TRANSACTION;
12 • SET time_zone = "+00:00";
13
14
15 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
16 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
17 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
18 • /*!40101 SET NAMES utf8mb4 */;

19
20 --
21 -- Base de datos: `tiendadeabarrotes`
22 --
23
24 ---
```


4. CÓDIGO 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  
3     try:  
4         conexión = pymysql.connect(  
5             host = "localhost",  
6             user = "root",  
7             password = "",  
8             database = "tiendadeabarrotes"  
9         )  
10        print("Conexión exitosa :)")  
11  
12        consulta = conexión.cursor()  
13        textoConsulta = "SELECT * FROM clientes"  
14        consulta.execute(textoConsulta)  
15        resultado = consulta.fetchall()  
16        for x in resultado:  
17            print(x)  
18  
19    except Exception as err:  
20        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

```
conexionAbarrotesBDProductos.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 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

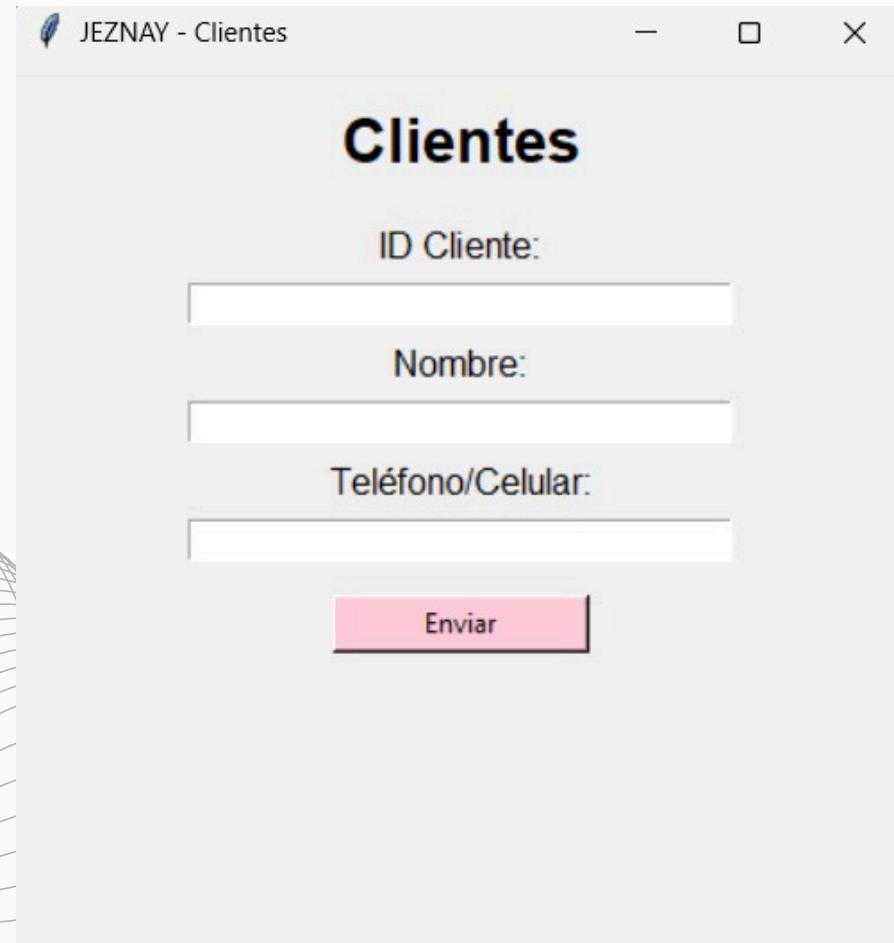
```
py conexionAbarrotesBDVentas.py ×

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()
```



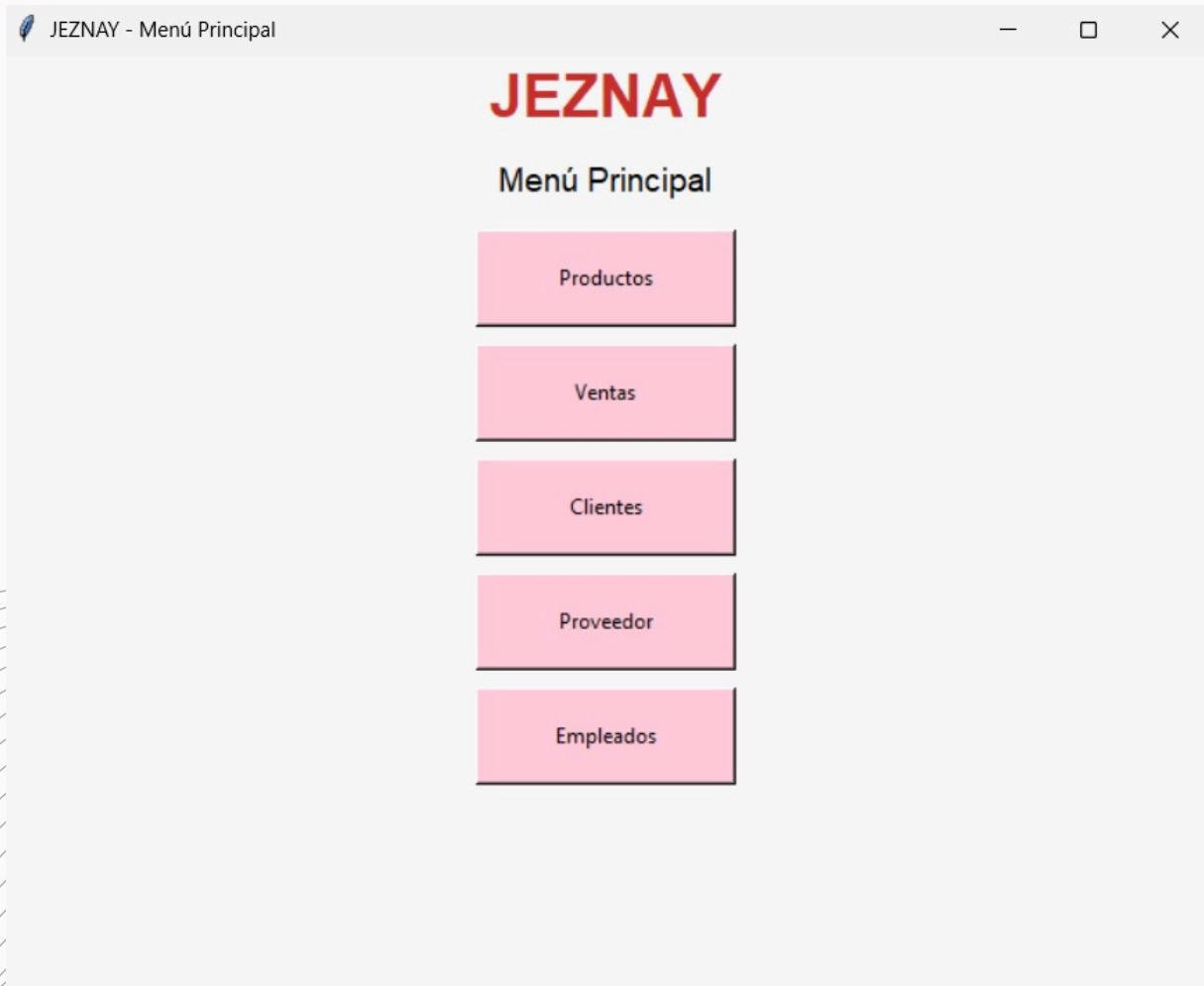
4.2.3 Interfaz y Código : Login

```
LogIn.py x
1 import tkinter as tk
2 from tkinter import messagebox
3
4 def verificar_login(): 1 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 ×
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()
```



5. BACKLOG

En este enlace se encuentra la página de Taiga.io

<https://tree.taiga.io/project/ismael-codes-azul/backlog>

Aquí se encuentra la lista de tareas, los roles del equipo y todo lo que se hizo con el proyecto

6. REPOSITORIO DE GIT

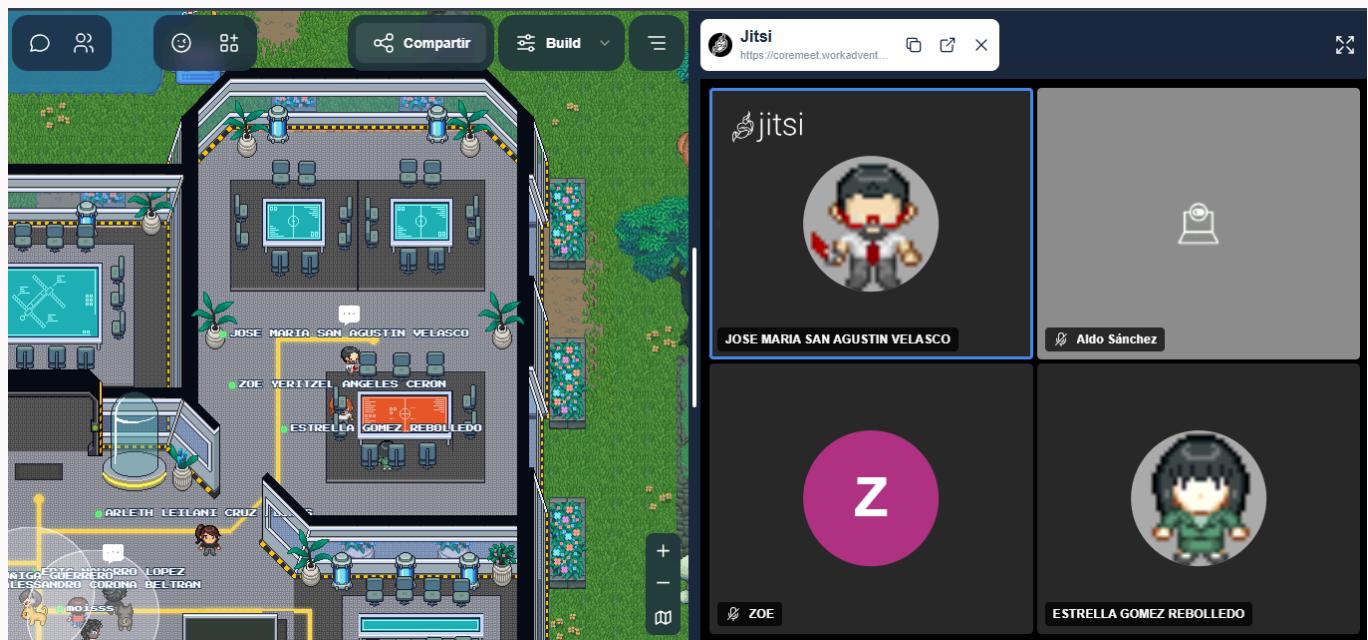
En este enlace se encuentra la página del Git de nuestro equipo

[Enlace a Git](#)

En el Git está todo nuestro progreso, el número de commits por cada integrante y todo el trabajo realizado

7. ANEXOS

Esto es algo extra, pero agregamos las fotos que tomamos en nuestro Daily Scrum, dentro del juego WorkAdventure



8. CONCLUSIONES FINALES

Gracias a este proyecto nos permitió aplicar de manera integral y colaborativa todos los conocimientos que adquirimos en este semestre, desde el análisis a las diagramas UML, el uso de las Bases de Datos y demás herramientas de apoyo como GitHub y Pycharm.

Pero lo importante, es en la manera en la que usamos estos recursos para hacer este proyecto, un "Un sistema de Gestión para una Tienda de Abarrotes", el trabajo en equipo que manejamos fue fundamental para el avance de este proyecto, en la parte de Backlog con la página de Taiga.io nos dividimos las tareas necesarias entre los integrantes para que esto no sea muy pesado, nos ayudó a fortalecer nuestras habilidades con estos programas y nuestra comunicación en equipo.

Nuestro sistema cumple con su objetivo principal, que es facilitar el control y la administración en lo productos, clientes, empleados y demás que ocurre día a día en una tienda. La estructura del sistema permite llevar un inventario actualizado, la información se guarda en la base de datos local y las relaciones entre la tienda y el cliente se ven detalladas e los diagramas UML antes presentados.

Todo el sistema en conjunto, ofrece una solución práctica y funcional para las necesidades básicas que ocupa un negocio de abarrotería, permitiendo una administración más ordenada, un manejo de la información eficiente y mejora productiva en el proceso de registro y consulta de datos. Nuestro proyecto demuestra una manera más fácil de optimizar las actividades diarias de un negocio como este, y con un software sencillo y complejo a la vez, es posible lograr grandes cosas.

