

```
-- =====
-- TABLA: MESA
-- =====

CREATE TABLE mesa (
    id_mesa INT AUTO_INCREMENT PRIMARY KEY,
    numero INT NOT NULL,
    capacidad INT NOT NULL,
    estado VARCHAR(20) DEFAULT 'Disponible',
    id_cafeteria INT,
    FOREIGN KEY (id_cafeteria) REFERENCES cafeteria(id_cafeteria)
);

-- =====
-- TABLA: SILLA
-- =====

CREATE TABLE silla (
    id_silla INT AUTO_INCREMENT PRIMARY KEY,
    numero INT NOT NULL,
    estado VARCHAR(20) DEFAULT 'Disponible',
    id_mesa INT,
    FOREIGN KEY (id_mesa) REFERENCES mesa(id_mesa)
);
```

```
-- =====
-- TABLA: PRODUCTO
-- =====

CREATE TABLE producto (
    id_producto INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion VARCHAR(200),
    precio DECIMAL(10,2) NOT NULL,
    categoria VARCHAR(50)
);

-- =====
-- TABLA: PEDIDO
-- =====

CREATE TABLE pedido (
    id_pedido INT AUTO_INCREMENT PRIMARY KEY,
    fecha DATETIME NOT NULL,
    estado VARCHAR(30) DEFAULT 'Pendiente',
    id_cliente INT,
    id_empleado INT,
    id_mesa INT,
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),
    FOREIGN KEY (id_empleado) REFERENCES empleado(id_empleado),
    FOREIGN KEY (id_mesa) REFERENCES mesa(id_mesa)
);
```

```
-- =====
-- TABLA: PEDIDO_PRODUCTO (N:M)
-- =====

CREATE TABLE pedido_producto (
    id_pedido INT,
    id_producto INT,
    cantidad INT NOT NULL,
    precio_unitario DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (id_pedido, id_producto),
    FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido),
    FOREIGN KEY (id_producto) REFERENCES producto(id_producto)
);
```

OPERACIONES CRUD IMPLEMENTADAS

Lista de funcionalidades implementadas

- Gestión de cafetería (registro y administración de información general).
- Registro y administración de clientes.
- Registro y administración de empleados con roles.
- Gestión de mesas (número, capacidad, estado).
- Gestión de sillas asociadas a mesas.
- Registro y administración de productos del menú (precio, categoría, descripción).
- Registro de pedidos con fecha, estado y asociaciones (cliente, empleado, mesa).

- Asociación de múltiples productos a un mismo pedido mediante tabla intermedia.
- Control de cantidad y precio unitario por producto dentro del pedido.
- Estructura relacional lista para implementar CRUD completos en todas las tablas.

□ Código:

```

class ProductAndStock(MainWindow):
    def __init__(self, login_window=None):
        super().__init__(login_window)
        self.title = "Login"
        self.login_window = login_window
        self.setWindowTitle("Iniciar de Sesión")
        self.setGeometry(300, 300, 400, 400)

        central = QWidget()
        self.setCentralWidget(central)

        layout = QVBoxLayout()
        layout.setContentsMargins(10, 10, 10, 10)

        title = QLabel("Registro de Productos")
        title.setAlignment(Qt.AlignCenter)
        title.setStyleSheet("color: #4a7e5f;")

        layout.addWidget(title)

        # Formulario para agregar producto
        form_layout = QFormLayout()
        self.nombre_input = QLineEdit()
        self.nombre_input.setPlaceholderText("Nombre del producto")
        form_layout.addRow("Nombre", self.nombre_input)

        self.precio_input = QLineEdit()
        self.precio_input.setPlaceholderText("Precio")
        self.precio_input.setPrefix("$ ")
        form_layout.addRow("Precio", self.precio_input)

        central.setLayout(layout)

```

```

def cargar_stock(self):
    db = Database()
    query = "UPDATE producto SET stock = %s WHERE id = %s"
    result = db.execute_query(query, (nuevo_stock, producto_id))
    if result:
        QMessageBox.information(self, "Success", "Producto agregado exitosamente")
        self.nombre_input.clear()

def actualizar_stock(self, producto_id, nuevo_stock):
    """Actualiza el stock en la base de datos en tiempo real"""
    query = "UPDATE producto SET stock = %s WHERE id = %s"
    result = db.execute_query(query, (nuevo_stock, producto_id))
    if result:
        QMessageBox.information(self, "Success", "Producto actualizado exitosamente")
        self.cargar_stock()

def agregar_producto(self):
    nombre = self.nombre_input.text().strip()
    precio = self.precio_input.text().strip()
    cat = self.categoria_input.currentText()
    stock = self.stock_input.value()

    if not nombre:
        QMessageBox.warning(self, "Error", "Ingrese el nombre del producto")
        return

    query = """INSERT INTO producto (nombre, precio, categoria, stock, disponible, categoria)
VALUES (%s, %s, %s, %s, %s)"""
    result = db.execute_query(query, (nombre, precio, cat, stock, 1))
    if result:
        QMessageBox.information(self, "Success", "Producto agregado exitosamente")
        self.nombre_input.clear()

```

BACKLOG DEL PROYECTO

Enlace al tablero Jira:

<https://softwareprojects.atlassian.net/jira/software/projects/G1/boards/2/backlog?atIOrigin=eyJpljoiNmY5MWE3MDgyZTBINDkxMWEyMmVkJNzVhMTY0MmZhMWMiLCJwljoiaij9>

Lista de tareas por integrante de equipo:

The image shows two screenshots of a Jira project interface. The top screenshot displays the backlog for 'A1 Sprint 1' from Nov 13 to Nov 16, showing 7 work items. The bottom screenshot shows a detailed view of tasks assigned to team members, with columns for Author, Person assigned, Estimated, Priority, and Status.

Backlog View (Top Screenshot):

ID	Tarea	Estado	Prioridad
G1-14	Documentación	In Progress	Medium
G1-15	Capturas	In Progress	Medium
G1-20	Conexión base de datos	Done	Medium
G1-17	Diagrama casos de uso	Done	Medium
G1-18	Borradores de diagramas	Done	Medium
G1-19	Codificación base de datos	Done	Medium
G1-23	Cuerpo del código	Done	Medium

Team Assignment View (Bottom Screenshot):

Author	Person assigned	Estimated	Priority	Status
G1-14	JESÚS NAVARRO LOPEZ	1d	Medium	IN PROGRESS
G1-15	carlos054	1d	Medium	IN PROGRESS
G1-20	carlos054	1d	Medium	IN PROGRESS
G1-17	Eduardo Manuel Valdés	1d	Medium	IN PROGRESS
G1-18	Eduardo Manuel Valdés	1d	Medium	IN PROGRESS
G1-19	DARIO ORTEGA SOSA	1d	Medium	IN PROGRESS
G1-23	JESÚS NAVARRO LOPEZ	1d	Medium	IN PROGRESS
G1-14	JESÚS NAVARRO LOPEZ	1d	Medium	IN PROGRESS
G1-15	DARIO ORTEGA SOSA	1d	Medium	IN PROGRESS

REPOSITORIO DE CÓDIGO

Enlace al repositorio:

<https://github.com/Ismael-Software/proyecto-final-gris2.git>

Commits por integrante:

<input type="checkbox"/> Yamelin Perez Barrera – 5
<input type="checkbox"/> Eric Navarro Lopez – 5
<input type="checkbox"/> Eduardo Manuel Vazquez Gerardo – 7
<input type="checkbox"/> Alirredo Hasan Hernandez Hernandez – 5
<input type="checkbox"/> Dario Ortega Sosa – 3
<input type="checkbox"/> Erick Jovani Verdi Baltazar – 5