

```
-- =====  
-- 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 ProductosController(Controller):
    def cargar_datos(self):
        self.table.setModel(self.modelo)

    def actualizar_producto(self, producto_id, nuevo_stock):
        """Actualizar el stock de un producto en la base de datos"""
        query = "UPDATE productos SET stock = %s WHERE id = %s"
        result = db.execute(query, (nuevo_stock, producto_id))
        if result:
            # Actualizar la visualización inmediatamente
            self.cargar_datos()

    def agregar_producto(self):
        name = self.nombre_input.text().strip()
        price = float(self.precio_input.value())
        cat = self.categoria_input.currentText()
        stock = self.stock_input.value()

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

        query = "INSERT INTO productos (nombre, precio, categoria, stock, disponible, categoria) VALUES (%s, %s, %s, %s, %s, %s)"
        result = db.execute(query, (name, price, cat, stock, 1, 1))

        if result:
            QMessageBox.information(self, "Éxito", "Producto agregado correctamente")
            self.nombre_input.clear()

```

```

class ProductosWindow(QWidget):
    def __init__(self, login_window=None):
        super().__init__()
        self.login_window = login_window
        self.setWindowTitle("Ventana de Productos")
        self.setGeometry(100, 100, 400, 300)

        central = QWidget()
        self.setCentralWidget(central)

        layout = QVBoxLayout()
        layout.addWidget(self)

        title = QLabel("Ventana de Productos")
        title.setStyleSheet("font-size: 12px; font-weight: bold;")
        title.setStyleSheet("color: #4682B4;")
        layout.addWidget(title)

        # Formulario para agregar productos
        form_layout = QHBoxLayout()
        self.nombre_input = QLineEdit()
        self.precio_input = QLineEdit()
        self.categoria_input = QLineEdit()
        self.stock_input = QLineEdit()
        self.agregar_btn = QPushButton("Agregar")
        form_layout.addWidget(self.nombre_input)
        form_layout.addWidget(self.precio_input)
        form_layout.addWidget(self.categoria_input)
        form_layout.addWidget(self.stock_input)
        form_layout.addWidget(self.agregar_btn)

```

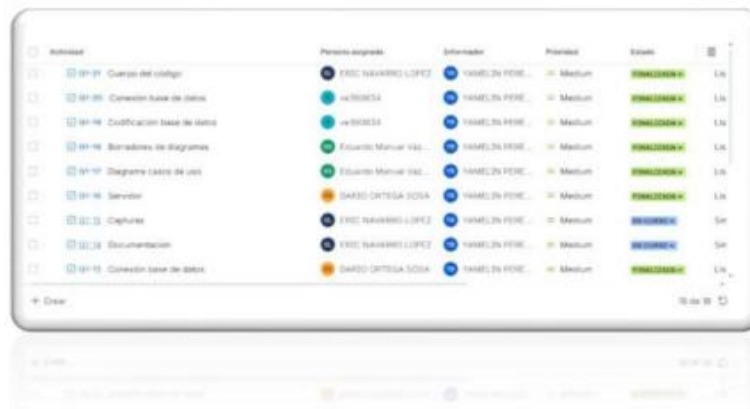
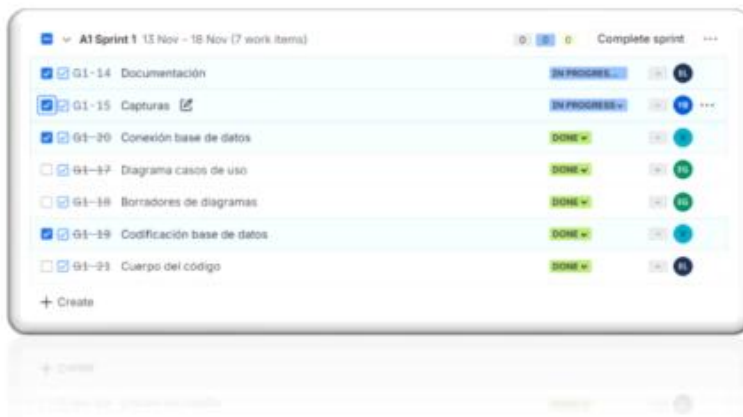
BACKLOG DEL PROYECTO

☐ **Enlace al tablero Jira:**

<https://softwareprojects.atlassian.net/jira/software/projects/G1/boards/2/backlog?atlOrigin=eyJpIjoiNmY5MWE3MDgyZTBiNDkxMWEyMmVkNzVhMTY0MmZhMWMiLCJwIjoi>

١٩

☐ **Lista de tareas por integrante de equipo:**



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"/> Alfredo Hasan Hernandez Hernandez – 5
<input type="checkbox"/> Dario Ortega Sosa – 3
<input type="checkbox"/> Erick Jovani Verdi Baltazar – 5