

Diseño del proyecto 3.3 Gestor de Inventario para Hogar o Laboratorio

Presentado por:

- Daniel Valderrama Lopez
- Jesús Estiven Vargas Calderón
- Mateo Felipe Amaya Novoa

Repositorio en Git Hub:

https://github.com/mateo-amaya-eci/Proyecto_gestor_de_-inventario.git

Diseño de la aplicación con UML:

El diagrama UML representa un **sistema de gestión de inventario de componentes electrónicos**, dividido en varias clases que colaboran entre sí para manejar datos, mostrarlos en una interfaz y generar reportes.

- **MainWindow**
Es la interfaz gráfica principal. Permite al usuario agregar, editar, buscar, eliminar y visualizar componentes. Se comunica indirectamente con el inventario a través de **InventoryManager** para realizar todas las operaciones.
- **InventoryManager**
Es el **controlador principal** del sistema. Maneja la lógica del inventario: agregar, actualizar, eliminar y consultar componentes. Se compone de un **DatabaseManager**, del cual depende directamente para almacenar y recuperar información.
- **DatabaseManager**
Administra la base de datos SQLite permitiendo conectar, desconectar, ejecutar consultas, insertar, actualizar y obtener componentes.
- **Component**
Representa cada componente individual del inventario (ID, nombre, tipo, cantidad, ubicación, fecha de compra) por lo que contiene los métodos para acceder y modificar estos atributos. Es el objeto central que manejan todas las demás clases.
- **ReportGenerator**
Genera reportes en varios formatos (CSV, XML, HTML) usando listas de componentes. También produce reportes de baja existencia.

Main Window

```

-tableView: QTableWidget
-nameEdit: QLineEdit
-typeCombo: QComboBox
-quantitySpin: QSpinBox
-locationEdit: QLineEdit
-dateEdit: QDateEdit
-searchEdit: QLineEdit
 addButton: QPushButton
 updateButton: QPushButton
 deleteButton: QPushButton
 searchButton: QPushButton
 reportButton: QPushButton
 statusLabel: QLabel
 dbManager: DatabaseManager
 inventoryManager: InventoryManager
 selectedId: int

+MainWindow(parent: QWidget* = nullptr)
+~MainWindow()
+~MainWindow() : void
+loadComponents() : void
+clearForm(): void
+populateForm(component: Component & ) : void
+getFormData() : Component
+addComponent() : void
+updateComponent() : void
+deleteComponent() : void
+searchComponents() : void
+onTableSelectionChanged() : void
+generateReport() : void
+checkLowStock() : void

```

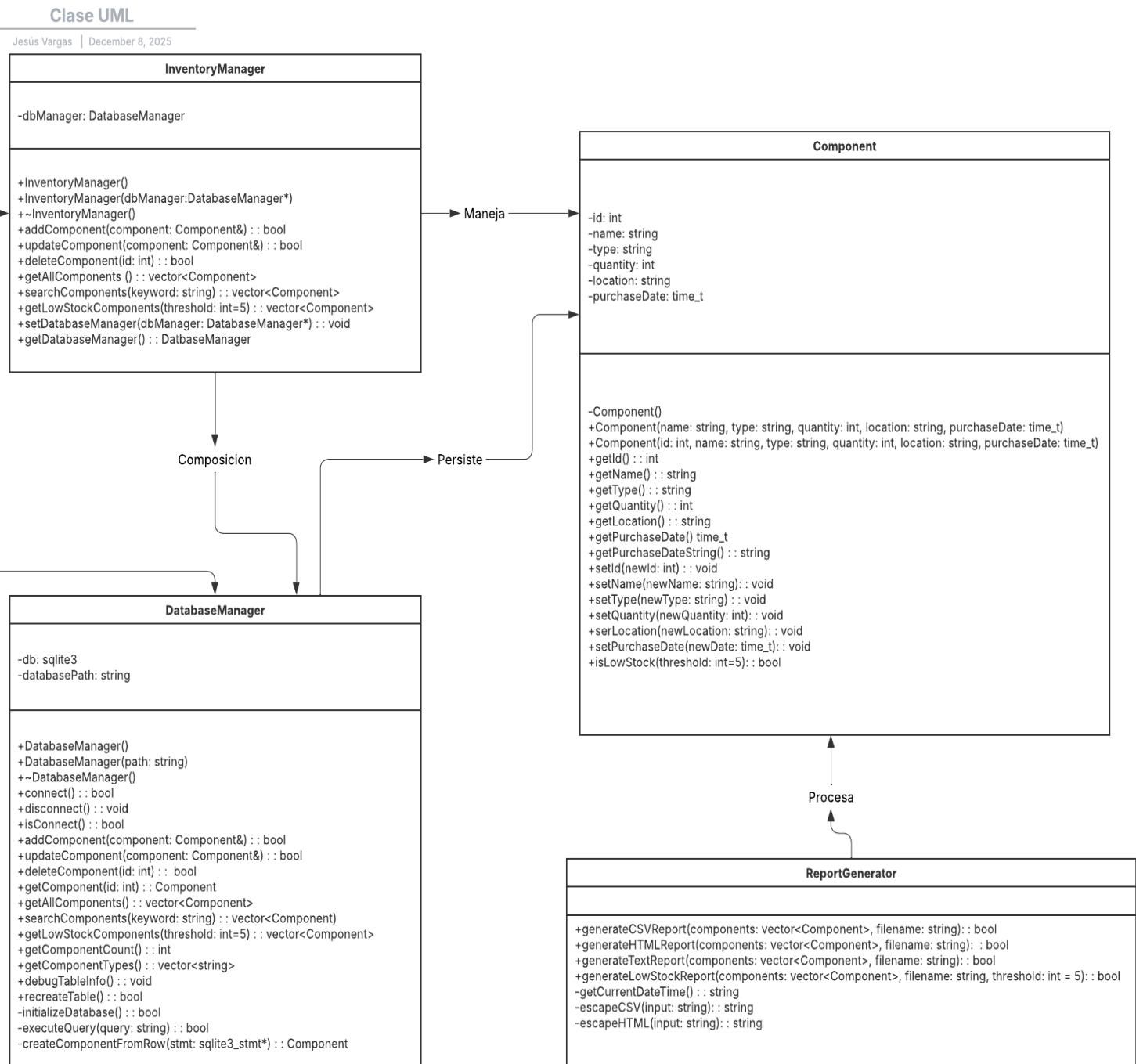


Diagrama entidad-relación:

Este diagrama representa un **sistema de gestión de inventario de componentes electrónicos**.

Las entidades clave son:

- **COMPONENTS**: almacena información de cada componente disponible (nombre, tipo, cantidad, ubicación, fecha de compra).
- **STOCK_ALERTS**: registra **alertas generadas cuando la cantidad de un componente baja de un umbral definido**. Cada alerta está vinculada a un componente mediante una clave foránea.
- **REPORTS**: guarda datos de los **reportes generados** (HTML, CSV, PDF, TXT), incluyendo fecha de creación y ruta del archivo

Donde los componentes pueden generar alertas de stock y los reportes incluyen información sobre los componentes o su inventario.

