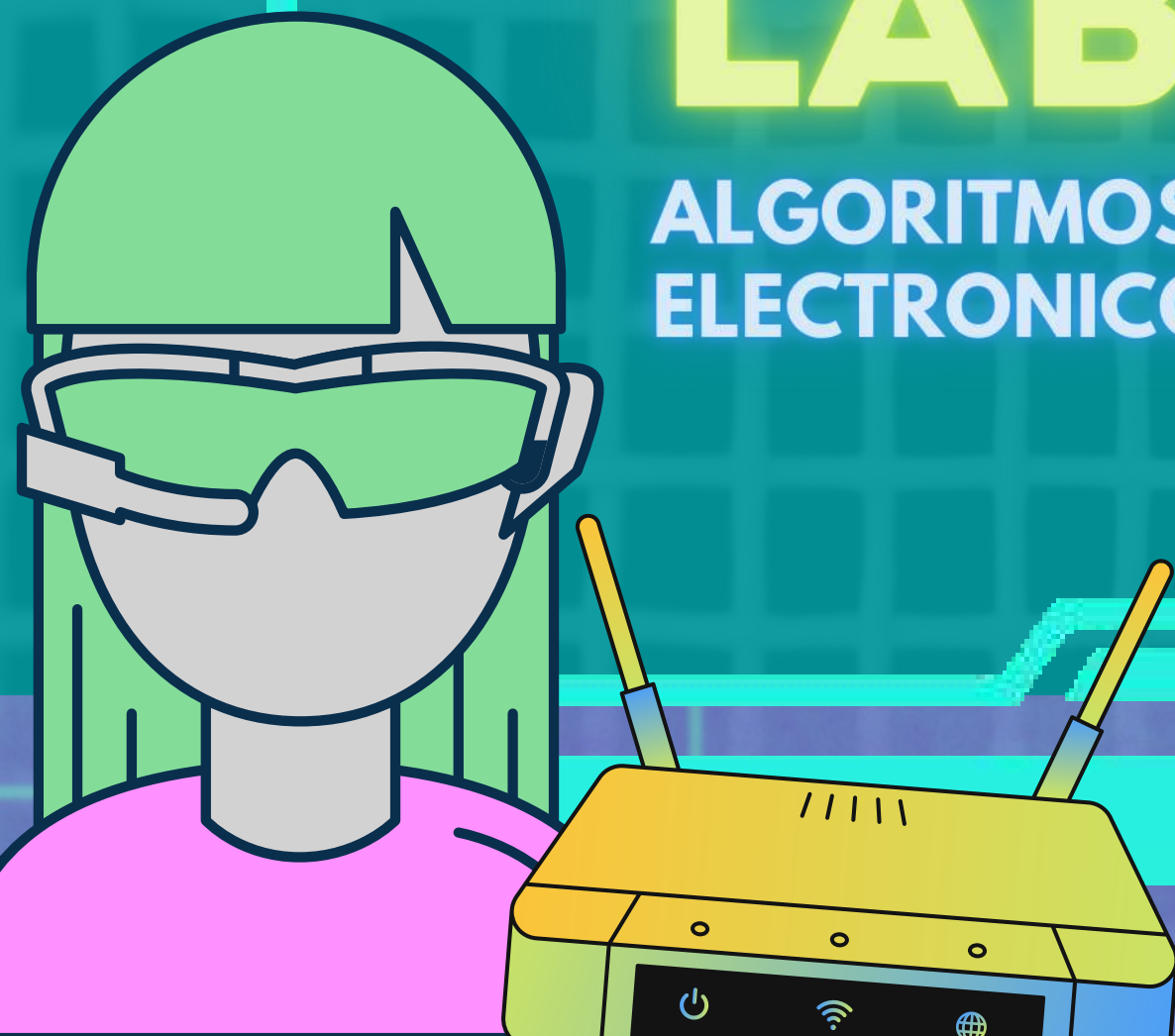


Jose Miguel Sedano
Angye Valentina Garcia

INVENTARIO DE LABORATORIO

ALGORITMOS EN SISTEMAS
ELECTRONICOS



OBJETIVOS Y HERRAMIENTAS

OBJETIVOS

Desarrollar un sistema de inventario en C++ y Qt capaz de:

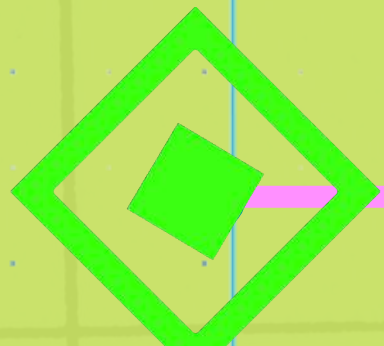
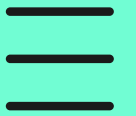
- Registrar componentes
- Consultar stock
- Editar/Eliminar items
- Generar reportes
- Guardar todo en SQLite

HERRAMIENTAS

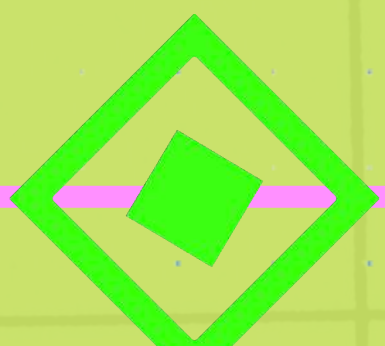
- C++17
- Qt 6.10.1 (Widgets + SQL)
- Qt Creator
- SQLite
- POO / MVC básico

New Tab

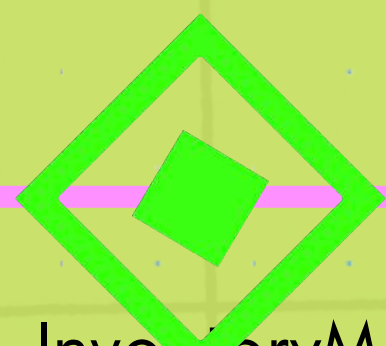
FUNCIONAMIENTO INTERNO



USUARIO:
Usuario interactúa con la GUI



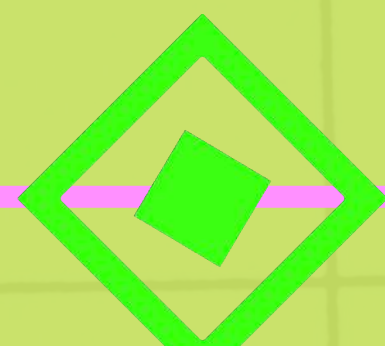
GUI:
GUI llama a InventoryManager



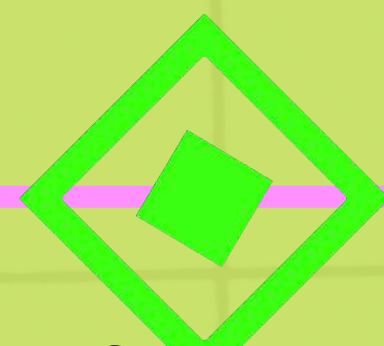
InventoryManager:
InventoryManager pide datos a DatabaseManager



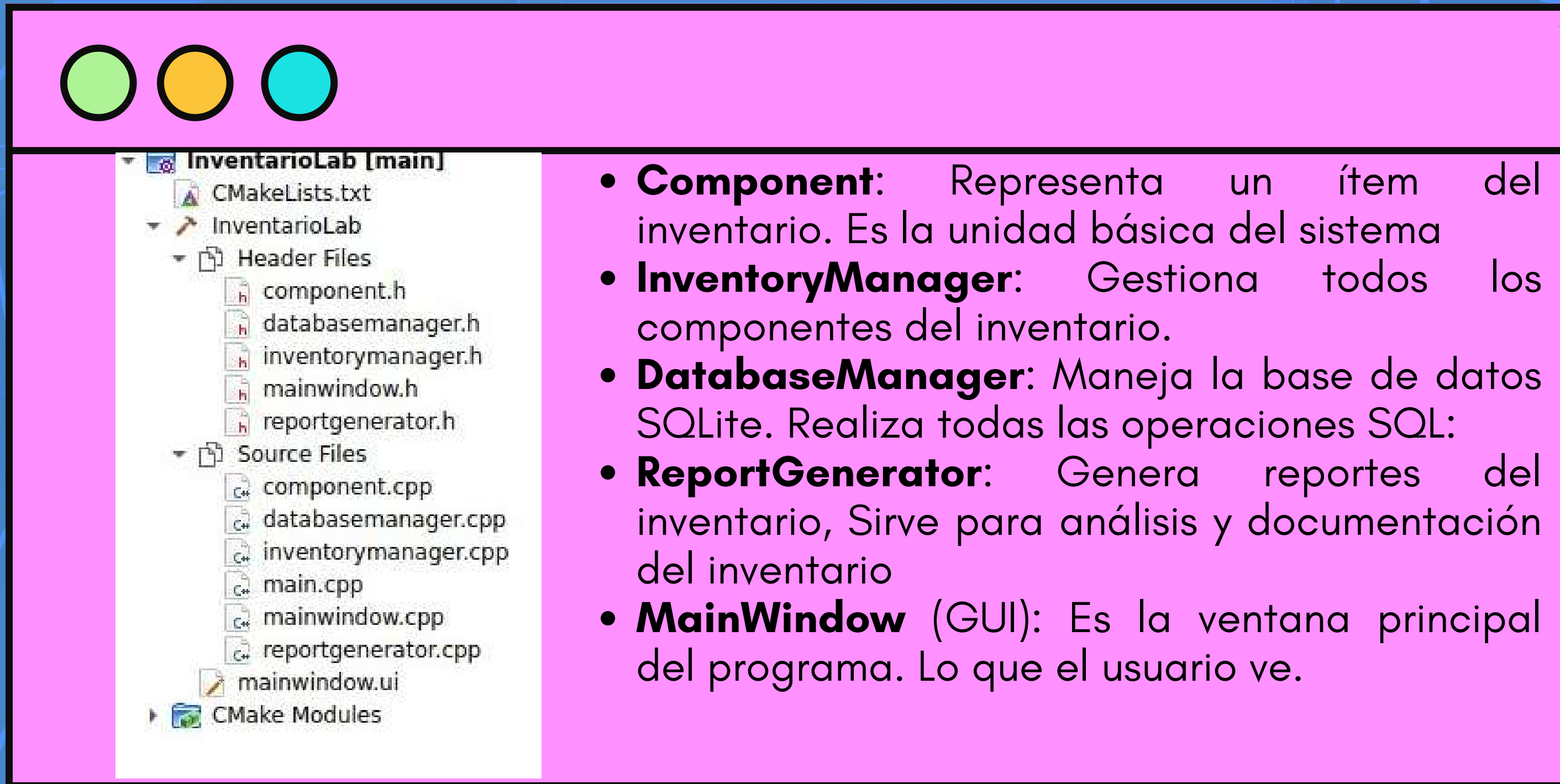
DatabaseManager:
DatabaseManager ejecuta consultas SQLite



SQLite:
Se actualiza la tabla en la GUI



GUI:
fin del ciclo

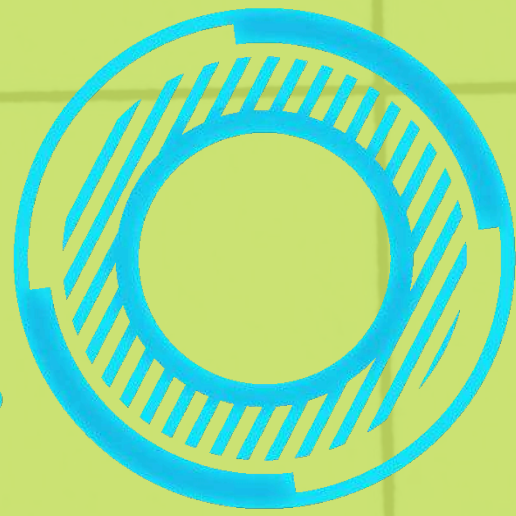


- **Component:** Representa un ítem del inventario. Es la unidad básica del sistema
- **InventoryManager:** Gestiona todos los componentes del inventario.
- **DatabaseManager:** Maneja la base de datos SQLite. Realiza todas las operaciones SQL:
- **ReportGenerator:** Genera reportes del inventario, Sirve para análisis y documentación del inventario
- **MainWindow** (GUI): Es la ventana principal del programa. Lo que el usuario ve.

DISEÑO ORIENTADO A OBJETOS



BASE DE DATOS SQL



```
databasemanager.cpp > ...
#include "databasemanager.h"
#include <QSqlQuery>
#include <QSqlError>
#include <QDir>
#include <QDebug>

DatabaseManager::DatabaseManager() {
    openDatabase();
    createTables();
}

bool DatabaseManager::openDatabase() {
    QDir().mkpath("database");

    db = QSqlDatabase::addDatabase("QSQLITE");
    db.setDatabaseName("database/inventario.db");

    if (!db.open()) {
        qDebug() << "Error abriendo BD:" << db.lastError();
        return false;
    }
    return true;
}

void DatabaseManager::createTables() {
    QSqlQuery query;
    query.exec(
        "CREATE TABLE IF NOT EXISTS components ("
        "id INTEGER PRIMARY KEY AUTOINCREMENT,"
        "name TEXT,"
        "type TEXT,"
        "quantity INTEGER,"
        "location TEXT,"
        "purchase_date TEXT)"
    );
}
```

Guarda todos los elementos del inventario con los siguientes campos:

- id: identificador único del componente
- nombre: del componente
- tipo: categoría o tipo (resistencia, sensor, herramienta, etc.)
- cantidad: disponible en stock
- Ubicación: ubicación física dentro del laboratorio
- Fecha de compra: o ingreso

SQLite permite que el sistema:

- almacene los datos de forma permanente,
- pueda consultar y actualizar información rápidamente

INTERFAZ GRÁFICA

Una barra de búsqueda, que permite filtrar componentes por nombre, tipo o ubicación.

Buscar por nombre o tipo...

ID	Nombre	Tipo	Cantidad	Ubicación	Fecha Compra
1 4	Play Satation 5	Entretenimie...	1	Zona entrenimiento	2025-09-14
2 5	Televisor	Entretenimie...	2	Zona Entretenimie...	2025-05-23
3 6	Nevera	Electrodome...	1	Cocina	2025-06-10
4 7	Lavadora	Electrodome...	1	Cocina	2026-06-13
5 8	Aspiradora Inteligente	Electrodome...	2	Sala y comedor	2025-12-03

Nombre:

Tipo:

Cantidad:

Ubicación:

Fecha Compra:

Agregar Actualizar Eliminar Limpiar Exportar CSV

Una tabla principal donde se muestran los componentes almacenados.

Items de búsqueda

Botones de acción (Agregar, Editar, Eliminar) para gestionar los elementos del inventario..

La interfaz gráfica del sistema fue desarrollada con Qt Widgets.

Está diseñada para ser simple, intuitiva y permitir al usuario gestionar el inventario fácilmente.

iGRACIAS!

