

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN CIENCIAS DE LA INGENIERIA

ESTRUCTURA DE DATOS



BIBLIOTECA MAGICA

BRANDON GUSTAVO GÜINAC ROMÁN

201931217

FECHA DE ENTREGA: 02/11/2025

DIAGRAMAS ESTRUCTURALES

1. Introducción

Este documento presenta los **diagramas estructurales** que describen la arquitectura del sistema *Biblioteca Mágica Alrededor del Mundo*.

Los diagramas muestran la relación entre los módulos principales, las clases del sistema y el flujo de procesos que permiten gestionar libros, bibliotecas y transferencias entre sedes mediante estructuras de datos avanzadas.

2. Diagrama General de Arquitectura

El sistema está compuesto por **cuatro módulos principales**:

1. Interfaz Gráfica (GUI):

- Desarrollada en Python, permite al usuario visualizar, agregar y administrar bibliotecas y libros.
- Gestiona la carga de archivos CSV y la generación de reportes.

2. Módulo Lógico (Controladores):

- Contiene las clases que coordinan operaciones entre estructuras de datos.
- Maneja inserciones, búsquedas, transferencias y simulaciones de envío.

3. Módulo de Estructuras de Datos:

- Implementa los TAD's: Listas, Árboles, Tablas Hash, Pilas, Colas y Grafos.
- Cada estructura incluye su método `graficar()` para exportar visualizaciones con Graphviz.

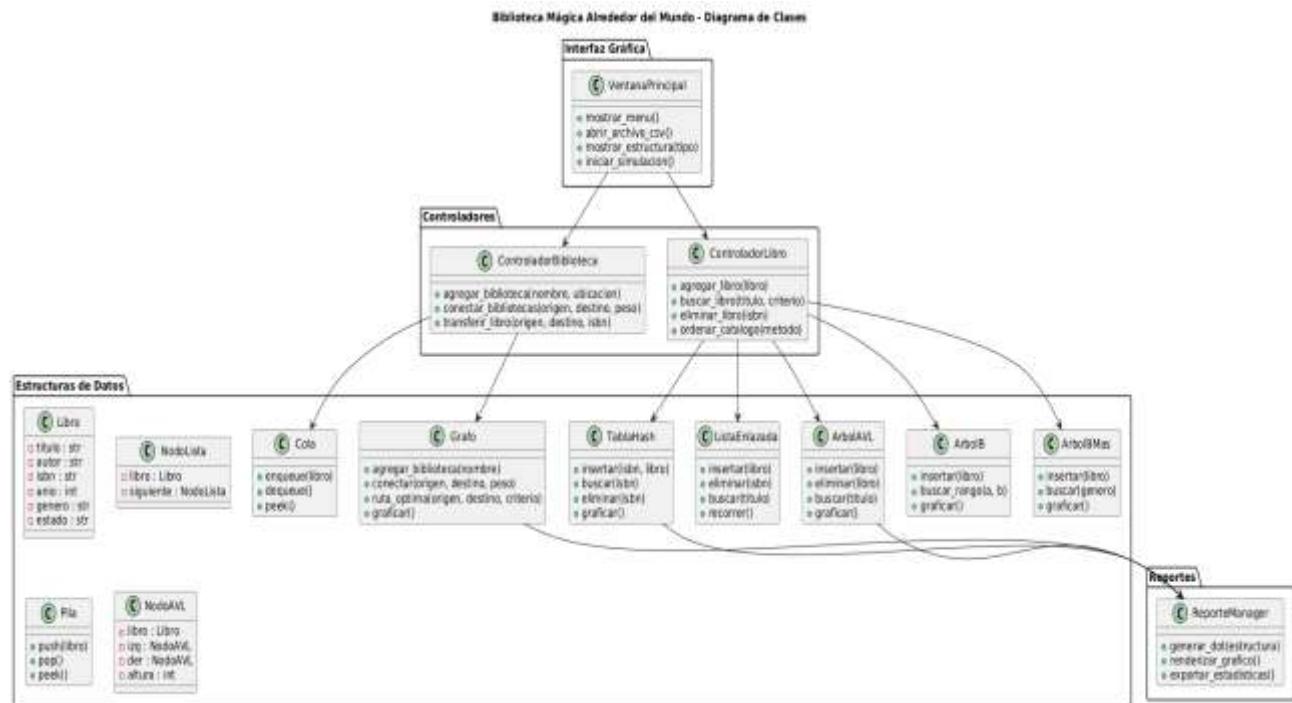
4. Módulo de Reportes y Visualización:

- Genera archivos .dot, imágenes renderizadas, estadísticas de rendimiento y comparaciones de tiempos de búsqueda.

3. Diagrama UML de Clases Principales

Descripción:

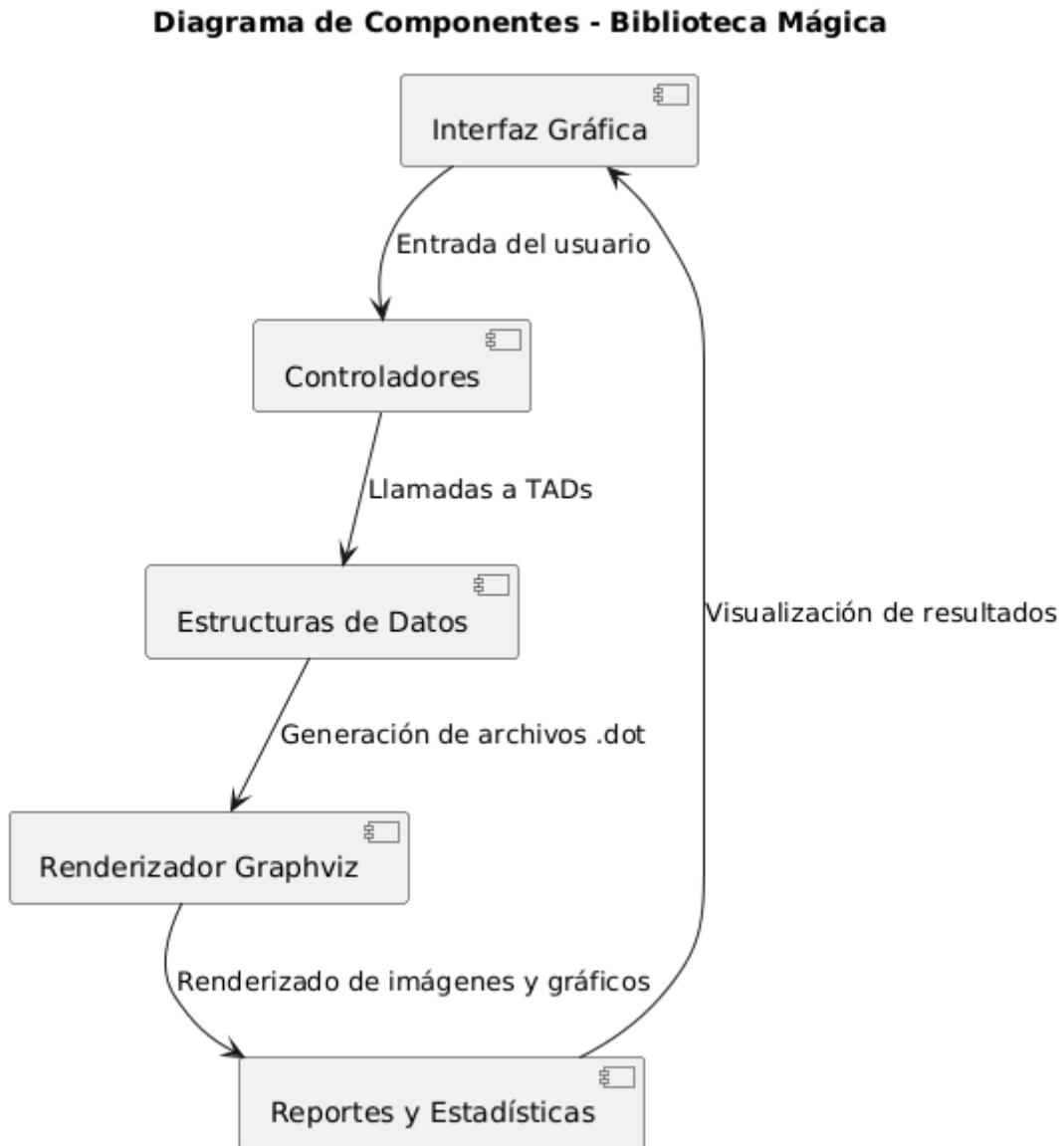
Este diagrama muestra las clases principales del sistema y sus relaciones lógicas.



4. Diagrama de Componentes

Descripción:

El siguiente diagrama muestra cómo los diferentes módulos se comunican entre sí:



5. Diagrama de Flujo General del Sistema

Descripción:

Representa el flujo de operaciones principales dentro de la aplicación.



6. Descripción del Renderizado con Graphviz

El sistema utiliza **Graphviz** para generar las representaciones visuales de las estructuras de datos. Cada clase (ArbolAVL, TablaHash, Grafo, etc.) incluye un método graficar() que:

1. Crea un archivo .dot con la estructura actual.
2. Llama al motor de Graphviz para generar un .png o .svg.
3. Envía la imagen al módulo de la interfaz para su visualización.

De esta forma, el sistema logra representar de manera dinámica:

- Árboles AVL balanceados.
- Árboles B y B+.
- Tabla Hash con colisiones.
- Grafo de bibliotecas con pesos en tiempo o costo.
- Colas de despacho simuladas.

7. Conclusión

Los diagramas estructurales presentados permiten comprender la organización interna del sistema y la interacción entre sus componentes.

El uso de módulos independientes facilita la mantenibilidad y escalabilidad del proyecto, mientras que **Graphviz** aporta una visualización clara del comportamiento de las estructuras dinámicas durante la ejecución.