



ESTRUCTURA DE DATOS

NRC / ASIGNATURA: 23225

PROFESORA: Mayra Isabel Alvarez
Jimenez

PERÍODO ACADÉMICO:

PREGRADO S-I ABR 25 - AGO 25

PROYECTO 1ER PARCIAL



TÍTULO:

**Visualizador de operaciones básicas de estructuras
dinámicas lineales**

ESTUDIANTE(S)

Jose Villa
Andres Sandoval
Gabriel Calvache

FECHA DE ENTREGA: 2 /06/ 2025

CALIFICACIÓN OBTENIDA:

Objetivos

- Desarrollar una aplicación interactiva en C++ que permita representar y manipular estructuras de datos lineales de forma dinámica e intuitiva, utilizando el entorno de desarrollo Code Blocks.
- Implementar las operaciones fundamentales de cada estructura de datos aplicando principios de programación orientada a objetos y plantillas genéricas.
- Fomentar el aprendizaje práctico de la programación en C++ mediante la creación de una interfaz gráfica, que facilite la visualización y manipulación de las estructuras de datos.

Desarrollo

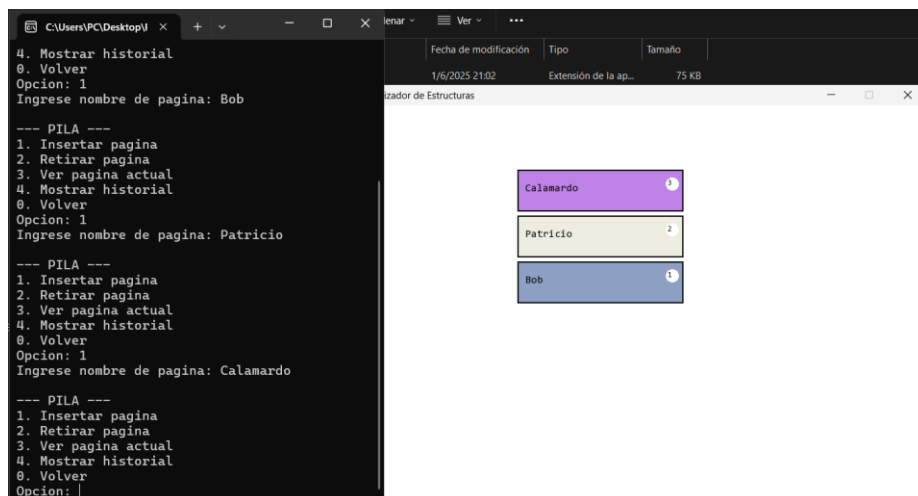
El proyecto fue implementado en el entorno de desarrollo Code Blocks utilizando el lenguaje C++. El objetivo principal fue construir una aplicación interactiva capaz de representar de manera visual el funcionamiento de tres estructuras de datos lineales: pila, cola y lista. Para lograr una experiencia interactiva y didáctica, se empleó la biblioteca Simple and Fast Multimedia Library (SFML), la cual permitió generar animaciones gráficas sincronizadas con las operaciones realizadas sobre las estructuras.

Arquitectura del Proyecto

El proyecto está estructurado en módulos, organizados mediante archivos de cabecera .h y un archivo principal main.cpp. Cada estructura cuenta con su propia lógica de operaciones insertar, eliminar y mostrar. Cada cambio que se realiza con las funciones es representado gráficamente por la ventana renderizada por SFML.

La ejecución se divide en dos partes principales:

- **Consola:** En esta parte se presenta al usuario un menú inicial desde el cual puede seleccionar la estructura a gestionar. Posteriormente, se muestran submenús con las operaciones disponibles para la estructura seleccionada.
- **Ventana gráfica:** A la par se abre una ventana gráfica donde se visualizan los cambios en la estructura seleccionada de una manera dinámica. Cada operación realizada en consola tiene su correspondiente representación visual, mostrando cómo se insertan o eliminan nodos en tiempo real.



SFML en el Proyecto

SFML es una biblioteca multimedia escrita en C++ que facilita la creación de aplicaciones con gráficos, sonido y manejo de eventos de entrada. En este proyecto, se utilizó SFML principalmente para:

- **Renderizado de gráficos en 2D:** Se emplearon clases como `sf::RenderWindow`, `sf::RectangleShape`, `sf::CircleShape`, y `sf::Text` para construir visualmente los elementos de cada estructura.
- **Gestión de eventos:** A través de `sf::Event`, se controló el cierre de la ventana gráfica y la interacción con el usuario en la interfaz visual, aunque la mayoría de las operaciones se realizan desde la consola.
- **Texto:** Se utilizó la clase `sf::Font` para cargar fuentes personalizadas que permitieran mostrar información clara y estilizada dentro de los nodos .

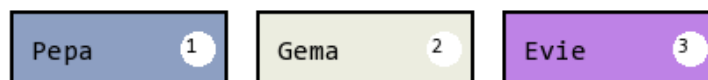
Implementación Visual de las Estructuras

Cada estructura fue representada visualmente de forma diferente, adaptándose a sus características:

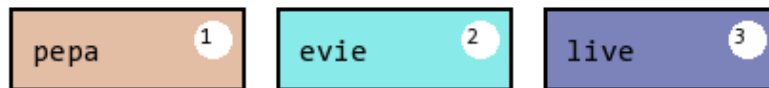
- **Pila:** Los nodos se apilan verticalmente desde la base hacia arriba. Cada inserción coloca un nuevo nodo en la parte superior, y cada eliminación lo retira visualmente con una animación.



- **Cola:** Los nodos se disponen horizontalmente, mostrando el orden de entrada y salida. La inserción ocurre al final y la eliminación al inicio, respetando el comportamiento First In - First Out.



- **Lista:** Los nodos de la lista aparecen de forma horizontal en pantalla, uno al lado del otro, de manera ordenada conforme al orden en que se insertan. Aunque la interfaz no muestra visualmente las conexiones entre nodos, su posición permite observar el orden de la estructura y cómo cambian tras alguna operación.



Conclusiones:

1. La implementación visual de las estructuras de datos lineales facilitó significativamente la comprensión de su funcionamiento interno. Observar cómo se agregan y eliminan elementos en tiempo real permite reforzar los conceptos teóricos, especialmente en cuanto a la lógica secuencial y la manipulación de nodos.
2. El uso de la biblioteca SFML fue importante para desarrollar una interfaz visual interactiva que permita representar cada estructura con claridad y que a la vez esta se pueda actualizar de una manera dinámica con cada operación realizada, esto no solamente ayuda a mejorar nuestro conocimiento sobre las estructuras dinámicas, si no también desarrollar conocimiento sobre la implementación de un entorno visual en C++.
3. El proyecto fomentó el trabajo modular y organizado, al dividir el código en archivos .h y un archivo principal main.cpp. Asimismo, fortaleció habilidades de programación orientada a objetos, uso de menús en consola y control de flujo. Además, se adquirió experiencia en integrar librerías externas como SFML dentro de entornos como Code Blocks.

Recomendaciones

1. Un problema que se halló al trabajar con la librería SFML en Code Blocks es la incompatibilidad entre la arquitectura de la biblioteca y la del compilador MinGW. Para solucionar este problema, se debe verificar que ambas arquitecturas coincidan desde el inicio del proyecto. Es decir, si se está utilizando un compilador MinGW de 64 bits, debe descargarse y configurarse la versión de SFML también para 64 bits.
2. Otro problema fue que, al ejecutar el archivo .exe fuera del entorno de Code Blocks o al trasladarlo a otro equipo, la aplicación no iniciaba correctamente debido a la ausencia de archivos DLL u otros recursos requeridos por SFML. Para solucionar esto se identificó que el problema no sucedía si se copiaba todos los archivos necesarios que se encuentran en la carpeta bin, principalmente los archivos .dll, junto con el ejecutable. Si el programa utiliza recursos adicionales como fuentes, imágenes o sonidos, estos también deben ser incluidos en la misma carpeta.
3. Un tercer problema que se identificó fue que la ventana gráfica generada por SFML dejaba de responder al intentar interactuar directamente con ella. Aunque no se encontró una solución efectiva, esta situación permitió comprender la importancia de gestionar adecuadamente los eventos de la ventana en SFML, especialmente cuando se combinan interfaces gráficas con interacción por consola. Sería recomendable explorar otras formas de estructura del programa que separen o sincronicen mejor ambas interfaces para evitar conflictos de ejecución.

Bibliografía

- [1] “SFML Graphics Library | Quick Tutorial,” GeeksforGeeks, [En línea]. Disponible en: <https://www.geeksforgeeks.org/sfml-graphics-library-quick-tutorial/>
- [2] “SFML GitHub repository,” GitHub, [En línea]. Disponible en: <http://github.com/SFML/SFML>
- [3] “SFML 3.0.1 Documentation,” SFML, [En línea]. Disponible en: <https://www.sfm1-dev.org/documentation/3.0.1/annotated.html>