**Descripción del Proyecto**

**Objetivo general**

Desarrollar una aplicación interactiva que implemente y visualice varias estructuras de datos: **lista enlazada (LL)**, **stack**, **queue**, **lista doblemente enlazada (DLL)**, **árbol binario de búsqueda (BST)** y **árbol AVL**. El sistema debe permitir ejecutar las operaciones fundamentales de cada estructura, animarlas visualmente y soportar persistencia en archivos para recargar estados previamente guardados. Además, debe incluir la conversión de un **BST a AVL**.

**Estructuras y funcionalidades**

**1. Listas y estructuras lineales (LinkedList, stack, queue, Double Linked List)**

* Operaciones de inserción, eliminación, búsqueda y recorrido.
* Animaciones para cada operación y visualización de punteros.
* Implementación de estructuras derivadas como **stack** (LIFO) y **queue** (FIFO).
* Indicar el head, tail y nullptrs.

**2. Árboles (BST y AVL)**

* Mostrar metadata de nodos: balance, altura, profundidad
* **BST**: inserción, eliminación, búsqueda, recorridos (preorder, inorder, postorder) y balance.
* **AVL**: inserción y eliminación con rebalanceo automático (rotaciones simples y dobles). Predecesor, sucesor
* Conversión directa de **BST a AVL**, mostrando visualmente las rotaciones aplicadas.
* Indicar raiz

**3. Interfaz de Usuario (UI) y Animaciones**

* Panel visual para interactuar con cada estructura.
* Controles para crear, modificar y visualizar las estructuras de datos.

**4. Persistencia**

* Guardar y cargar estructuras en archivos binarios o de texto.
* Restaurar el estado de la aplicación a partir de un archivo previamente guardado.

**Presentaciones parciales**

Se realizarán **3 presentaciones obligatorias** del avance del proyecto:

* **Semana 4**: Presentación inicial del diseño de la interfaz de usuario y lista simple.
* **Semana 6**: Presentación intermedia, demostrando estructuras lineales completas y parte del árbol.
* **Semana 10**: Presentación final del proyecto completo.

**Penalización:** Si el alumno no presenta en alguna de estas semanas, se aplicará una **penalidad de 2 puntos por día** faltante.

**Distribución de puntos**

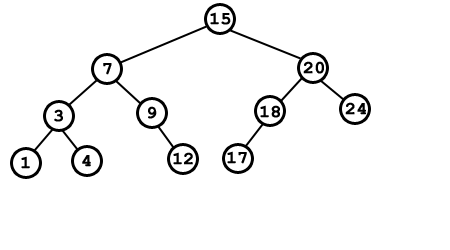
* **Árboles (BST y AVL):** 10 puntos
* **Listas (LL, DLL, stack, queue):** 7 puntos
* **UI y Animaciones:** 5 puntos
* **Persistencia:** 3 puntos  
  **Total:** 25 puntos

**Validación del proyecto y Políticas**

* Durante la evaluación, el docente podrá solicitar **cambios en el código** para validar la autoría del proyecto.
* Si el estudiante no puede realizar los cambios solicitados, se le **deducirá el 50% de la nota final**.
* Un proyecto que **no compila equivale a 0**.
* Proyecto copiado **equivale a 0**.
* Cada vez que la aplicación presente un **error de ejecución** se aplicará una penalización del **10%** sobre la nota final.
* En caso de usar repositorios en GitHub deberá crearlos como privados.
* Para todas las estructuras de datos es obligatorio utilizar su propia implementación y no utilizar las de la librería std de c++ ni las librerías de Qt.
* Separar la lógica de la aplicación (Backend) de la interfaz del usuario (Frontend).

**Política de entrega tardía**

* **No se aceptarán proyectos entregados fuera de la fecha estipulada.**
* **No se aceptarán reclamos** en caso de que el alumno no se presente a las evaluaciones.



Profundidad: 2  
Altura: 0  
Balance: 0