

Ejercicios de Estructuras dinámicas en C++

25 %

1. Diseñe una clase en C++ llamada `ListadoEstudiantil` que implemente una lista enlazada simple. La clase debe incluir:

- Un constructor que inicialice una lista vacía.
- Un método para insertar un nuevo nodo al final de la lista.
- Un método para eliminar un nodo con un valor específico.
- Un método para buscar un valor en la lista y devolver su posición.
- Un método para imprimir todos los elementos de la lista (en archivo de texto `.txt`).
- Un destructor que libere la memoria asignada dinámicamente.
- Sobrecarga del operador de acceso para permitir la indexación de elementos de la lista.
- Un método para invertir la lista enlazada.

El listado debe almacenar información de estudiantes, incluyendo nombre completo, año de nacimiento, número de identificación, promedio académico y email.

50 %

2. Se tiene una señal digital electrocardiográfica (ECG) almacenada en un archivo de texto plano (`ECG.txt`). Cada línea del archivo contiene dos valores numéricos que representan la amplitud de la señal y el instante de tiempo específico. En la Fig.1 se muestra la señal ECG representativa en azul y su versión filtrada en rojo. Diseñe una clase en C++ llamada `SeñalECG` que implemente

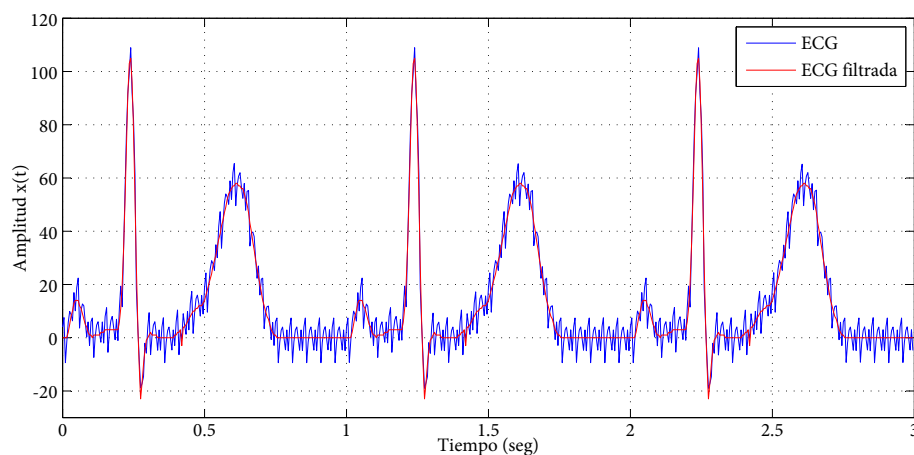


Fig. 1: Señal ECG.

una lista enlazada doble para almacenar y procesar la señal ECG. La clase debe incluir:

- Un constructor que inicialice una lista vacía.

- Un método para cargar la señal desde el archivo de texto.
- Un filtro digital pasa bajos en ecuaciones de diferencias, que logre obtener la versión filtrada similar a la que aparece en la Fig. 1.
- Un método para detectar picos en la señal filtrada.
- Un método para imprimir la señal original y filtrada en un archivo de texto.
- Un destructor que libere la memoria asignada dinámicamente.
- Un método para calcular la frecuencia cardíaca promedio basada en los picos detectados.

25 %

3. Tome el ejercicio del juego de dominó implementado en el trabajo práctico anterior y modifíquelo para que utilice una estructura dinámica de lista enlazada que vaya guardando el historial de movimientos de cada jugador. Al final del juego, el programa debe imprimir el historial completo de movimientos en un archivo de texto llamado `historial_domino.txt`. Cada entrada en el historial debe incluir el nombre del jugador, la ficha jugada y el estado actual del tablero después de cada movimiento.

Entrega

El trabajo práctico debe ser entregado en formato digital a través de un repositorio en la plataforma Github. Asegúrese de incluir un archivo README con instrucciones claras sobre cómo compilar y ejecutar su código. También, el archivo README debe explicar claramente la metodología que utiliza cada programa para resolver el problema dado. La fecha límite para la entrega es el 4 de Diciembre del 2025 a las 23:59 horas. No se aceptarán entregas tardías sin una justificación válida.