

# Tarea Programada 1: La lista enlazada simpática... ¡Tirando a chistosa!

Aurelio Sanabria  
Lenguajes de programación  
II semestre, 2017

---

## 1 Descripción

Esta tarea consiste en implementar una lista enlazada simple en archivo que permita almacenar enteros.

El archivo se debe manejar en formato binario con registros de tamaño fijo de 5bytes cada uno. De los 5 bytes, 4 se utilizan para almacenar el valor entero y el byte adicional se utilizará como apuntador (*puntero*) para indicar la posición del archivo en la cuál esta localizado un nodo determinado. Como restricción adicional en memoria no se pueden cargar los nodos y luego escribir la lista al archivo. No se puede, tampoco, cargar el nodo en memoria modificarlo y escribir el nodo completo, se debe trabajar directamente en el archivo para cambiar valores utilizando las funciones de *fseek*, *ftell*, *fwrite* y *fread*. Sin embargo, si se pueden tener múltiples apuntadores para realizar las operaciones.

Al eliminar un nodo se debe liberar ese espacio para futuras inserciones, el manejo de los huecos o secciones libres se debe hacer mediante una estructura de datos en memoria que **NO debe** ser estática y solo irá incluyendo los espacios de memoria en uso, **No los espacios libres**.

Se debe manejar con gracia y elegancia el caso cuando el espacio disponible en disco este lleno.

### Nota:

- Al final del archivo sobraré un byte, utilice ese byte para colocar el puntero a inicio de la lista.
- Esta progra no lleva interfaz, todo se probará desde el **main**, con pruebas que programará el profesor.
- Debe aplicar lo visto en clase sobre manejo de memoria y modularidad.

- Las posiciones de la lista inician en 0.
- Recuperarse graciosamente de una caída del sistema (terminar el programa y volver a arrancar con la misma lista) corresponde a puntos extra.
- Como recomendación implementen la función de imprimir\_lista tan pronto como puedan, para verificar que todo este bien.

## 2 Funcionalidad

1. Se deben implementar las siguientes funciones:

- **Insertar al final:** Se inserta un nuevo elemento al final de la lista.
- **Insertar el elemento en la posición n:** Se inserta en una posición específica de la lista (no del archivo).
- **Eliminar el último elemento:** Se elimina el último elemento de la lista.
- **Eliminar el elemento en la posición n:** Se elimina el elemento en la posición indicada de la lista (no del archivo).

## 3 Fecha de entrega

- La tarea se entrega a más tardar: el 28 de Agosto a las 10 p.m. (GMT -6).

## 4 Documentación

1. El documento debe contener las siguientes secciones

- Portada.
- Lecciones aprendidas (al menos 10, cada una de tres o más oraciones).

2. Se debe presentar el documento en formato *Portable Document Format (PDF)*

## 5 Evaluación

- 75% Revisión del software.
  - Archivo en formato binario 10
  - Inserción al final 15

- Inserción en posición 20
- Eliminar el final 10
- Eliminar en posición 20
- 15% Documentación.
  - Interna
  - Externa

## 6 Aspectos administrativos

1. La tarea se debe entregar en formato digital en un archivo compreso en formato tar.gz con el nombre: **nombre\_apellido.tar.gz**
2. El archivo debe descomprimirse a una carpeta que con el siguiente formato: **nombre\_apellido** y debe contener los archivos que el estudiante considere necesarios.
3. La tarea se debe entregar a la dirección aurelio.itcr@gmail.com
4. Es responsabilidad del estudiante asegurarse de que la tarea llegue a la dirección indicada, si la tarea no llega antes de la fecha límite se asigna nota de 0.
5. El asunto del correo debe ser el siguiente: **(2017 IIS)(Lenguajes)(Tarea 1) nombre apellido (paréntesis cuadrados...** Aunque no lo parezcan)
6. Cualquier correo con un asunto distinto no se revisará.
7. El código se revisa en la computadora del profesor (*Debian GNU/Linux inestable*), si programa bajo Gündou\$ hágalo bajo su propio riesgo (después de todo es su nota la que usted se esta jugando).