

**Trabajo práctico número 3**  
**"LinkedList"**  
*Programación I – Laboratorio I.*  
*Tecnicatura Superior en Programación.*  
**UTN-FRA**

**Autores:** *Mg. Mauricio Dávila*

**Revisores:** *Esp. Ing. Ernesto Gigliotti*

*Versión : 3*



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](http://creativecommons.org/licenses/by-sa/4.0/).

## **Índice de contenido**

1 Enunciado.....	3
2 Biblioteca LinkedList.....	3
3 Cómo realizar y entregar el trabajo práctico.....	4
3.1 Objetivo.....	4

## 1 Enunciado

Una empresa requiere una aplicación que le permita administrar su nomina de empleados, para lo cual se deberá desarrollar una solución implementando la biblioteca **LinkedList**. Se deberá modelar la entidad que representa un empleado con todos sus datos asociados de tal manera que la misma permita interactuar con las estructuras de datos almacenadas en los archivos.

***El programa deberá tener el siguiente menú de opciones:***

1. Cargar los datos de los empleados desde el archivo data.csv (modo texto).
2. Cargar los datos de los empleados desde el archivo data.csv (modo binario).
3. Alta de empleado
4. Modificar datos de empleado
5. Baja de empleado
6. Listar empleados
7. Ordenar empleados
8. Guardar los datos de los empleados en el archivo data.csv (modo texto).
9. Guardar los datos de los empleados en el archivo data.csv (modo binario).
10. Salir

## 2 Biblioteca LinkedList

Las funciones del LinkedList que deben utilizarse como mínimo son las siguientes:

**LinkedList\* ll\_newLinkedList(void)** -> *Crea y retorna un nuevo LinkedList. Es el constructor, ya que en él crearemos la struct y daremos valores iniciales a los campos.*

**void ll\_deleteLinkedList(LinkedList\* self)** -> *Elimina el LinkedList*

**void ll\_add(LinkedList\* self , void\* element)** -> *Agrega un elemento al final de LinkedList..*

**void ll\_remove(LinkedList\* self , int index)** -> *Elimina un elemento en LinkedList, en el índice especificado.*

**void\* ll\_get(LinkedList\* self , int index)** -> *Retorna un puntero al elemento que se encuentra en el índice especificado.*

**int ll\_len(LinkedList\* self )** -> *Retorna el tamaño del LinkedList.*

### 3 Cómo realizar y entregar el trabajo práctico

El trabajo práctico deberá ser entregado en el repositorio de GIT correspondiente al TP3 de la materia.

El mismo consistirá en el proyecto de CodeBlocks con el programa funcionando y comentado, respetando las reglas de estilo de la cátedra. La compilación no deberá arrojar mensajes de error ni de warnings.

El proyecto deberá contar con la biblioteca LinkedList la cual se proveerá como una biblioteca estática, y se deberá hacer uso de dichas funciones para resolver la lógica del programa.

#### 3.1 Objetivo

El objetivo del siguiente trabajo es que el alumno sea capaz de demostrar que puede integrar los conocimientos aprendidos durante la cursada en un caso real. Los conocimientos necesarios para la realización del TP son los siguientes:

- Manejo de punteros.
- Manejo de linkedList.
- Manejo de estructuras.
- Manejo de memoria dinámica.