

### OBJETIVOS

- Manejo de Listas enlazadas.
- Llevar a cabo la codificación de programas utilizando estructuras como un tipo de datos.

### Ejercicios:

- 1) Si todavía no creó el repositorio, copie el siguiente enlace en su navegador: <https://tinyurl.com/tl1-2024-tp4> esto creará el repositorio para poder subir el **Trabajo Práctico Nro. 4**, realice los pasos ya aprendidos para *clonar* el repositorio en su máquina y poder comenzar a trabajar de forma *local*.

**Nota.** No se olvide de incluir el archivo `.gitignore` en la raíz del repositorio para excluir los archivos `.exe`, `.obj` y `.tds` del mismo.

- 2) Considere la siguiente situación: En la misma distribuidora del práctico anterior ahora surgió la necesidad de llevar un control de las tareas realizadas por sus empleados. Usted forma parte del equipo de programación que se encargará de hacer el módulo en cuestión que a partir de ahora se llamará módulo To-Do:

#### Tareas

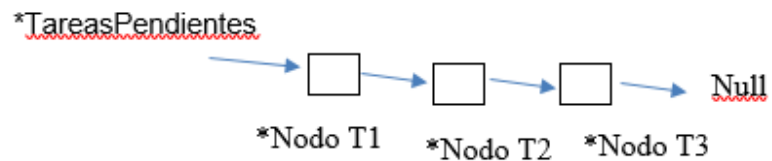
Cada empleado tiene un listado de tareas a realizar y debe indicar en el sistema si fueron realizadas o no. Para ello deberá crear dos listas enlazadas: una para las tareas pendientes y otra para las tareas realizadas. Cada vez que se marque una tarea como realizada deberá mover la tarea de la lista de tareas pendientes a la lista de tareas realizadas.

Las estructuras de datos necesarias son las siguientes:

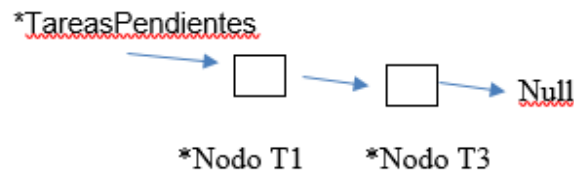
```
struct Tarea{
    int TareaID;//Numérico autoincremental comenzando en 1000
    char *Descripcion; //
    int Duracion; // entre 10 - 100
};

struct Nodo{
    Tarea T;
    Nodo *Siguiete;
};
```

La siguiente gráfica muestra cómo debe realizarse la transferencia de tareas entre listas:



Si la tarea 2 se marca como Realizada



- 1) Desarrolle una interfaz de carga de tareas para solicitar tareas pendientes, en la cual se solicite descripción y duración de la misma (el id debe ser generado automáticamente por el sistema, de manera autoincremental comenzando desde el número 1000). Al cabo de cada tarea consulte al usuario si desea ingresar una nueva tarea o finalizar la carga.
- 2) Implemente una interfaz para elegir qué tareas de la lista de pendientes deben ser transferidas a la lista de tareas realizadas.
- 3) Implemente una funcionalidad que permita listar todas las tareas pendientes y realizadas.
- 4) Implemente una funcionalidad que permita consultar tareas por id o palabra clave y mostrarlas por pantalla, indicando si corresponde a una tarea pendiente o realizada.