UTN - Mar Del Plata – TUP
Laboratorio 2 - Recuperatorio del Primer Parcial
2021

IMPORTANTE:

- **Crear un proyecto con su Apellido y Nombre.**
- Añadir comentarios a su código identificando cada inciso.

Estructuras:

```
typedef struct {
      char apellido[20];
      int matricula;
}stAlumno;
```

```
typedef struct {
      stAlumno dato;
      struct nodo* siguiente;
}nodo;
```

Obtenido	Valor	**Inciso**
	20	1-A) Crear la funcion **crearEstructuraAlumno:** stAlumno crearEstructuraAlumno(char[20], int); Esta funcion **no debe** pedirle los datos al usuario. 1-B) Crear la funcion **crearNodo** utilizando la estructura **stAlumno** como parametro. 1-C) Crear la funcion **mostrarNodo.** 1-D) Crear una funcion **cargarAlumno** que le pida los datos de un alumno al usuario y luego crear un nodo con los datos y mostrar ese nodo por mantalla utilizando las funciones creadas previamente.
	10	2) Crear una lista con tres nodos. Cada nodo debera guardar un alumno diferente. Puede hardcodear los datos de los alumnos. Los nuevos nodos deberan **insertarse al final de la lista** utilizando la funcion **insertarAlFinal.**
	20	3-A) Cargar un archivo “alumnos.bin” con los nodos de la lista. 3-B) Mostrar los datos cargados en el archivo “alumnos.bin”.
	10	4) Hacer una función que busque un alumno por matricula en la lista. Si lo encuentra devuelve 1, si no lo encuentra devuelve 0.
	15	5) Hacer una función que elimine el nodo de una lista que coincida con el apellido de un alumno pasado por parámetro. La función recibe **un puntero doble a la lista**. ***void eliminarNodo (nodo** lista, char apellido[30])***
	15	6) Hacer una función que **muestre un arreglo de alumnos en forma recursiva.** Recuerde utilizar modularizacion para mostrar cada alumno.
	10	7) Hacer una función main () que invoque a las funciones anteriores y demuestre el correcto funcionamiento del programa.