

PRÁCTICA No. 2 EDA

Listas

1. Implemente de manera dinámica una lista enlazada simple (TAD Lista) con dos punteros (inicial y actual), nodo fantasma y las operaciones definidas en la clase teórica.

a) Para la implementación recuerde aplicar los conceptos de abstracción y encapsulamiento en su código. Documentelo con comentarios.

b) La estructura básica para definir un nodo perteneciente a una lista es:

```
struct registro{  
    int valor;  
    struct registro *sigNodo;  
};
```

Nota: Esta estructura es básica, depende de los valores que se quieran almacenar en cada nodo.

c) Definida la estructura o esqueleto para un nodo establezca el nuevo tipo de dato con

```
typedef nombreTipoDeDato identificador
```

Ej: typedef struct registro _nodo;

d) Las operaciones a definirse serán:

```
crearLista  
listaVacía  
insertarNuevoNodo  
eliminarNodo  
primerElemento  
ultimoElemento  
recorrerLista  
buscarElemento  
modificarElemento
```

2. En base al ejercicio 1 implemente una lista ordenada que almacena los datos de estudiantes:

- DNI (int)
- Edad (int)
- PromedioCalif (decimal)

a) Cada nodo representa un estudiante de la Lista

- b) El TAD ListaEstudiantes debe estar ordenado de manera creciente según el código (DNI) de cada estudiante (i.e. el primer nodo de la lista contiene el estudiante con DNI 1).
- c) Crear una lista de n ($n \geq 5$) estudiantes, hacer una búsqueda con un DNI dado, imprimir los datos de los estudiantes, eliminar el último estudiante y modificar el promedio de calificaciones de 2 estudiantes (volver a imprimir la lista actualizada).

- **Fecha de presentación: 31 de octubre de 2014**
- **Código será evaluado únicamente si se ejecuta**