



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

## Práctica 17

### Datos generales:

Nombre de la Práctica	Listas simplemente ligadas
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	III. Estructuras lineales.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	30-septiembre-2021
Fecha de entrega con extensión y penalización	1-octubre-2021
Fecha de elaboración	29-septiembre-2021

Objetivo de la Práctica	Profundizar con las operaciones en las listas enlazadas.
Tiempo aproximado de realización	3 horas
Introducción	La memoria dinámica es un elemento importante en el manejo de información abundante donde no se sabe de antemano cuantos datos son los requeridos, por tanto solventa las limitaciones de la memoria estática. Las listas enlazadas permiten la manipulación de la memoria dinámica a través de la liga de nodos sucesivos.

### Referencias que debe consultar el alumno (si se requieren):

#### Referencia 1:

1. Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

## Referencia 2:

2. Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.

## Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.

## Actividades que debe realizar el alumno:

### Actividad inicial:

Generar el reporte en formato IDC.

### Actividad 1:

Primero genere la **Introducción**.

### Actividad 2:

Realice las implementaciones de los métodos de la clase ListaEncadenadaOrden (para eso se debe usar el enumerado que hacía uso de ASCENDENTE y DESCENDENTE para indicar el orden que queremos en nuestra lista, tal como se hizo en los arreglos), la cual implementa directa/indirectamente la interface ListaDatos:

**NOTA: No debe utilizar métodos de ordenamiento.**

- Buscar de manera ordenada.
- Imprimir en orden natural.
- Imprimir al revés.
- Insertar un elemento X en una lista enlazada ordenada.
- Eliminar un elemento X en una lista enlazada ordenada.
- Insertar un elemento al final.
- Insertar un elemento al inicio.
- Eliminar un elemento al final.
- Eliminar un elemento al inicio.
- Insertar elementos pasados como argumento en forma de un arreglo (puede ser ordenado y desordenado). De esta manera se tiene que ir insertando de uno por uno en la posición que les va tocando a cada uno de ellos.
- Insertar elementos pasados como argumento en forma de un Tabla2D. De esta manera se tiene que ir insertando de uno por uno en la posición que les va tocando a cada uno de ellos.



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

- Eliminar elementos pasados como argumento en forma de un arreglo (puede ser ordenado y desordenado). De esta manera se tiene que ir eliminando de uno por uno los elementos del arreglo pasado como argumento.
- Eliminar elementos pasados como argumento en forma de un Tabla2D. De esta manera se tiene que ir eliminando de uno por uno los elementos pasados como argumento.
- Adicionalmente termine de implementar todos los métodos que vienen heredados desde ListaDatos, pero cada uno apropiado a una lista ordenada. El método de cambiar es un método particular que lo que debe hacer es reacomodar el elemento nuevo que se ha substituido si no conserva el orden de la posición actual, encontrándole la nueva posición en donde debe ir. No debe usar métodos de ordenamiento como tal. Reutilice los métodos ya hechos para lograr esto.

NOTA: el elemento a procesar debe almacenarse en la lista de tal forma que la lista siga siendo ordenada. No olvide que la implementación de una **lista encadenada con orden NO es igual que la de una lista encadenada común y corriente**. Recuerde que se pueden insertar objetos numéricos, cadenas u objetos completos como Alumno, Casa, etc. Considere estas situaciones, similar a arreglo ordenado.

Haga el programa (actividad 2, la cual es el **Desarrollo** del programa, junto con la captura de pantalla del programa funcionando).

## Actividad 3:

Pruebe el funcionamiento del programa de la actividad 2 con todo y sus capturas de pantalla.

## Actividad 4:

Realice la sección de **Código agregado** (diagrama de clases UML).

## Actividad 5:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).

## Actividad 6:

Finalmente haga las **Conclusiones**.



# Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

---

## Actividad 7:

Enviar en <http://ingsoftware.reduaz.mx/moodle>

## Archivo anexo que se requiere para esta tarea (opcional):

Dudas o comentarios: [a7donso@gmail.com](mailto:a7donso@gmail.com)