**Profesor:** Felipe Reyes

**Mail:** [felipe.reyes@usach.cl](mailto:felipe.reyes@usach.cl)

**Eximición:** 5.0

**Notas:**

Cátedra (C)

|  |  |  |
| --- | --- | --- |
| Evaluación | Porcentaje | Fecha |
| Solemne 1 | 25% | 1/4/20 |
| Solemne 2 | 25% | 6/5/20 |
| Solemne 3 | 25% | 10/6/20 |
| Actividades y controles | 25% |  |

Laboratorio (L)

|  |  |  |
| --- | --- | --- |
| Evaluación | Porcentaje | Fecha |
| Controles | 60% |  |
| Proyecto | 40% |  |

Nota presentación = C\*0.6 + L\*0.4

**\*Importante**

Laboratorio estructuras de datos (Los miércoles en la tarde) comienza a final de mes

# Clase 1

Análisis asintótico (Velocidad a la cuál se ejecuta un algoritmo)

Formas de ordenamiento:

inSort

extrapolación

mergeSort

quickSort

# Clase 2

Lenguaje de alto nivel:

Lenguaje de bajo nivel:

Lenguaje fuertemente tipado: Exige que se declaren todos los tipos de datos (Ej: C)

Lenguaje debilmente tipado: No exige que se declaren (Ej: Python)

Lenguaje interpretado:

Lenguaje compilado:

Puntero: Se utilizan para reservar un espacio de memoria

## Algoritmos de ordenamiento

### Algoritmo de la burbuja

### Algoritmo de selección

### Algoritmo de QuickSort

# Clase 3

El arreglo tiene la misma memoria por cada elemento y el tiempo de acceso a cada elemento es constante.

Desventajas:

-Todos los datos están uno al lado de otro. Está la necesidad de solicitar el espacio en memoria para el peor de los casos cuando el arreglo es estático.

-Necesidad de memoria contigua: Puede ocurrir que, pese a haber suficiente memoria libre, no haya un bloque contiguo suficientemente grande.

Las listas enlazas son estructuras de datos dinámicas.

Al contrario de las estructuras estáticas, en las que su tamaño en memoria se establece durante una compilación y permanece inalterable durante la ejecución del programa.

Las estructuras de datos dinámicas crecen y se contraen a medida que se ejecuta el programa.

Son una colección de elementos, denominados nodos, dispuestos uno a continuación de otro.

Cada uno de ellos conectado al siguiente elemento por un enlace o puntero.

Las listas enlazadas son estructuras muy flexibles y con numerosas aplicaciones en el mundo de la programación.

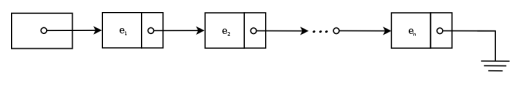
**Fundamentos teóricos**

La idea básica consiste en construir una lista cuyos elementos llamados nodos se componen de dos partes o campos:

La pimera parte consiste en la información.

La segunda parte o campo es un puntero que apunta al siguiente nodo de la lista.

La representación más extendida es aquella que utiliza un rectángulo con dos secciones al interior, valor del dato y el enlace. El último nodo apunta NULL.



**Clasificación de las listas enlazadas**

Listas simplemente enlazadas: Cada node contienen un único enlace con el nodo siguiente.

Listas doblemente enlazadas: Cada nodo contienen dos enlaces, uno con su predecesor y otro con su sucesor.

Listas circulares simplemente enlazadas: Es una lista simplemente enlazada en la que el último elemento se enlaza al primer elemento.

Listas circulares doblemente enlazadas: Es una lista doblemente enlazada en la que el último elemento se enlaza al primer elemento.

**Operaciones en la listas enlazadas**

Inicialización o creación, con declaración de los nodos.

Insertar elementos en una lista.

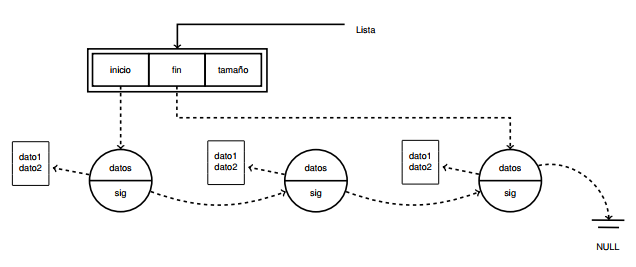
Eliminar elementos de una lista.

Buscar elementos de una lista (comprobar la existencia de elementos en una lista).

Recorrer una lista enlazada.

Comprobar si la lista enlazada está vacía.

**Declaración de una lista**



**La lista**

Una lista vacía no contiene nodos.

Se debe conocer el comienzo y el final.

Se debe conocer el tamaño actual de la lista.