

## Administración de memoria (Recurso valioso, debe administrarse bien)

Ciclo de vida de un programa

1-Edición: construcción de código

2-Copilación: Conversión de código fuente a código objeto

3-Distribución: Se empaqueta en un ejecutable

4-Enlace: se ligan dependencias y otras bibliotecas, hay 2 tipos; estático al copilar y dinámico con bibliotecas externas

5-Carga: Se lleva a memoria principal el programa

6-Ejecución: Se ejecuta

## Sistema operativo

¿Qué es?

Es software que se ejecuta sobre el hardware

¿Para que existe?

Para abstraer los recursos computacionales de modo tal que permita construir programas de aplicación (API)

## Jerarquía de memoria



## Esquemas de administración de memoria

Esquema inexistencia: - Son abstracciones de memoria

- Usado en los mainframes de los 60

- Interacción directa entre el programa y la memoria física

Funcionamiento: El SO copia por completo un programa a la RAM, lo libera hasta finalizarse.

Se subdivide la memoria para esto en bloques. Soluciona los problemas con static relocation

Esquema de redireccionamiento: - Se utiliza Dynamic Relocation

- Cada programa cree que empieza en la dirección 0

- Espacios reservados en memoria, base y límites.
- Si el programa referencia memoria, se le suma la base y se verifica que no sea mayor que el límite
- Se introduce el concepto de Swapping
- Los programas se cargan parcialmente en la memoria. El resto se deja en disco.

Memoria Virtual: Existe una memoria virtual de mayor tamaño que la RAM, el programa opera sobre esta memoria. Nace ante la necesidad de más memoria, pues los programas crecían +rápido que la RAM. El memory management unit, mapea y asocia memoria real (una página) memoria virtual

## Memory Layout

Parámetros de línea de comandos(argumentos) y variables ambientales(variables de solo lectura)
Pila ↓
Heap ↑
Datos sin inicializar
Datos inicializados
Código

-Pila y Heap, son lo más importante para el programador

-Pila y Heap, son dinámicos para que no se toquen

Código: código ejecutable

Datos inicializados: Variables globales y estáticas inicializadas por el programador

Datos sin inicializados: Variables globales no inicializadas

Heap y pila: Secciones específicas

Parámetros de líneas de comandos y variables ambientales: Parámetros pasados al programa y variables definidas en el Shell

## Pila

- Es amigable con el programador, menos pulgas y menos trabajo
- Se compone de Stack Frames
- Un frame tiene: \*storage, para variables globales
- \* Número de líneas para regresar
- \* Storage para parámetros
- La pila hace transparente a la memoria
- Cada vez que llama a un método crea un stack frame
- Cuando el método termina, el frame se limpia por completo y se libera la memoria asociada

## Heap

- No tiene estructura o no se maneja de manera automática
- Hay un api
- No es transparente
- Void \*malloc – memory allocation – (n bytes): Asigna n bytes en memoria y retorna un puntero
- Void free(void): Libera la memoria apuntada para el parámetro dado
- realloc: Me crea más espacio y trasporta la info hacia esa memoria nueva

- calloc: Me aloja la memoria y también me la inicializa el bloque de memoria en zero
- delete: Me borra ese elemento
- new: Me almacena un dato en memoria

## Puntero

- Tipo de dato
- Una variable que me apunta a un espacio de memoria
- En c/c++, se emplea un \* para declarar un puntero

Para poder validar que una función cumpla su complejidad teórica en parámetros prácticos para un programador, se le puede realizar un “benchmark” al algoritmo per se; no obstante, un “benchmark” hace referencia a una prueba de rendimiento o comparativa que llega a presentar un algoritmo, esta técnica radica en tomar el tiempo junto antes y después, de que un algoritmo se ejecute, con ayuda de esto se puede realizar una estrategia que me relaciona estrechamente la complejidad teórica de un algoritmo, con el uso de esta técnica.

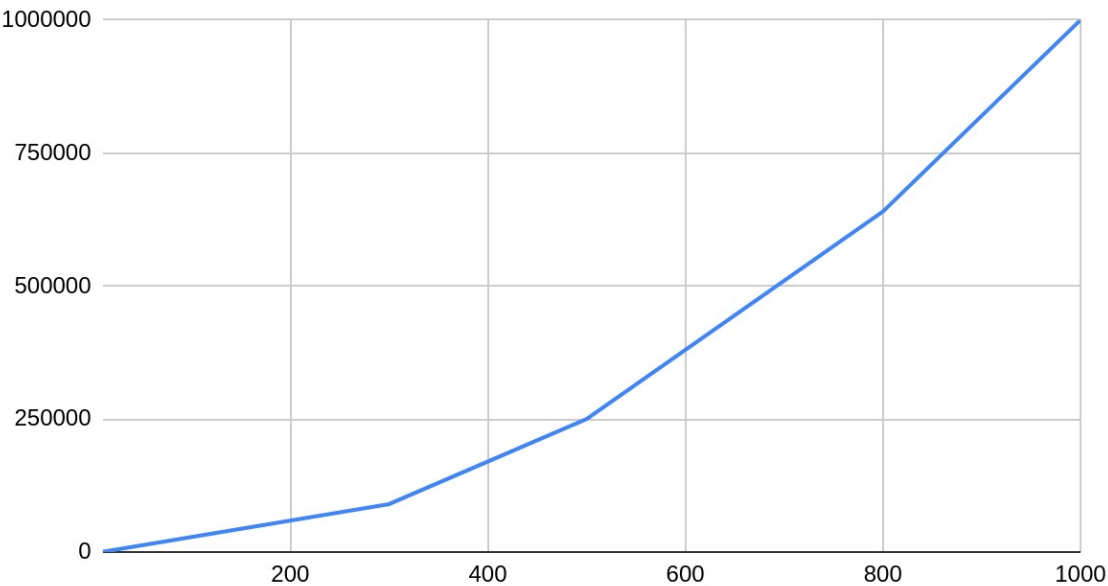
Esta técnica o estrategia hace referencia al uso de graficar una función que se comporta como la función teórica, donde las variables independientes serán las cantidades de datos ingresados y la variable dependiente será el tiempo de diferencia que se reporte, para luego con ayuda de Excel o cualquier programa de graficación, para graficar esta función.

Tabla 1: Datos y tiempo, experimental	
datos	tiempo
10	2.86E-06
300	2.51E-04
500	0.00067579
800	1.69E-03
1000	0.00265804

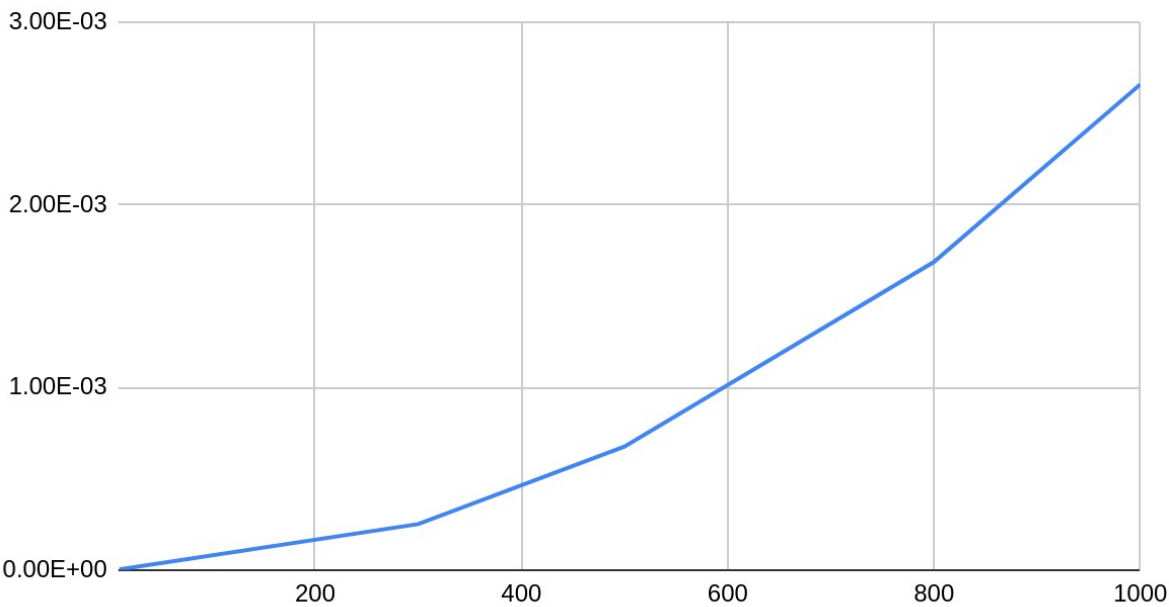
Tabla 2: Datos y tiempo, teórico	
datos	procesos
10	100
300	90000
500	250000
800	640000
1000	1000000

En la tabla 1 y 2, se ve el comportamiento que tubo el algoritmo de bubble sort, para una lista desordenada con varios valores y en las graficas 1 y 2, se puede apreciar el comportamiento que se llega a tener ambas funciones, en este caso que se representa bubble sort tiene un orden de crecimiento de  $O(n^2)$  para el peor caso de prueba, se ve como ambas graficas presentan una función cuadrática sumamente parecida y así comprobando la efectividad de esta técnica.

Gráfica 1:Datos vs tiempo, al nivel teórico



Gráfica 2: Datos vs tiempo, al nivel experimental



```

import java.util.*;
class hash{
    static void crearHash(int arr[]){
        HashMap<Integer, Integer> hmap = new HashMap<Integer, Integer>();
        for (int i = 0; i < arr.length; i++) {
            Integer current = hmap.get(arr[i]);
            if (hmap.get(arr[i]) == null) {
                hmap.put(arr[i], 1);
            }else {
                hmap.put(arr[i], ++current);
            }
        }
    }
}

```

Auto-Evaluación				
Estudiante:				
Josue Daniel Cubero Montero				
Criterios	Regular	Bueno	Excelente	Observaciones
Participa activamente durante la ejecución de la actividad, sin distraerse o perder el tiempo con temas ajenos a la actividad.			A	
Brinda información oportuna para incluir en el entregable de la actividad, asumiendo un rol de responsabilidad con alguno(s) de los componentes del entregable.			A	
Hace uso de una comunicación asertiva tanto para la información que pueda incluirse en el entregable como para la gestión del trabajo entre los diferentes miembros del equipo.			A	
Colabora en el cumplimiento de los lineamientos establecidos para la actividad.			A	
Estudiante:				
Aldo Cambronero Ureña				
Criterios	Regular	Bueno	Excelente	Observaciones
Participa activamente durante la ejecución de la actividad, sin distraerse o perder el tiempo con temas ajenos a la actividad.			A	
Brinda información oportuna para incluir en el entregable de la actividad, asumiendo un rol de responsabilidad con alguno(s) de los componentes del entregable.			A	
Hace uso de una comunicación asertiva tanto para la información que pueda incluirse en el entregable como para la gestión del trabajo entre los diferentes miembros del equipo.			A	
Colabora en el cumplimiento de los lineamientos establecidos para la actividad.			A	
Co-Evaluación				
Para estudiante:				
Josue Daniel Cubero Montero				
De estudiante:				
Aldo Cambronero Ureña				
Criterios	Regular	Bueno	Excelente	Observaciones
Participa activamente durante la ejecución de la actividad, sin distraerse o perder el tiempo con temas ajenos a la actividad.			A	
Brinda información oportuna para incluir en el entregable de la actividad, asumiendo un rol de responsabilidad con alguno(s) de los componentes del entregable.			A	
Hace uso de una comunicación asertiva tanto para la información que pueda incluirse en el entregable como para la gestión del trabajo entre los diferentes miembros del equipo.			A	
Colabora en el cumplimiento de los lineamientos establecidos para la actividad.			A	
Para estudiante:				
Aldo Cambronero Ureña				
De estudiante:				
Josue Daniel Cubero Montero				
Criterios	Regular	Bueno	Excelente	Observaciones
Participa activamente durante la ejecución de la actividad, sin distraerse o perder el tiempo con temas ajenos a la actividad.			A	
Brinda información oportuna para incluir en el entregable de la actividad, asumiendo un rol de responsabilidad con alguno(s) de los componentes del entregable.			A	
Hace uso de una comunicación asertiva tanto para la información que pueda incluirse en el entregable como para la gestión del trabajo entre los diferentes miembros del equipo.			A	
Colabora en el cumplimiento de los lineamientos establecidos para la actividad.			A	