

# CB100 - Algoritmos y Estructuras de Datos

*Cátedra Ing. Gustavo Schmidt - 2do cuatrimestre 2024*

## Trabajo Práctico 1: Ordenamiento V1.0

### Objetivo

Generar una pieza de software que simule el funcionamiento de la comparativa entre los métodos de ordenamiento.

### Enunciado

Se debe levantar un listado de valores en un archivo de texto ingresado por teclado (la ruta relativa del nombre del archivo), se debe presentar las opciones de ordenamiento:

Burbujeo

QuickSort

RadixSort

Una vez que se elige la opción deseada, se genera un archivo de salida con los valores ordenados (uno por línea) y se imprime en pantalla el tiempo que se demoró en segundos.

Se vuelve a preguntar si se quiere ejecutar nuevamente o termina el programa.

### Interfaz de usuario

Toda la interfaz de usuario debe estar basada en texto. El menú debe ser muy sencillo y solo se imprime el tiempo demorado por pantalla.

No es necesario que se limpie la pantalla, simplemente escribir el menú cada vez que se necesite.

# Cuestionario

Responder el siguiente Cuestionario:

- 1) ¿Qué es un Debug?
- 2) ¿Qué es un "Breakpoint"?
- 3) ¿Qué es "Step Into", "Step Over" y "Step Out"?

## Normas de entrega

Trabajo práctico individual: 1 persona.

Reglas generales: respetar el Apéndice A.

Se deberá subir un único archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

Padron-TP1.zip

Deberá contener los archivos fuentes (no los binarios), los datos personales (padrón, nombre y mail), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF).

La fecha de entrega vence el día viernes 06/09/23 a las 23.59hs.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

## Apéndice A

- 1) Usar las siguientes convenciones para nombrar identificadores.
  - a) Clases: Los nombres de clases y structs siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.
  - b) Métodos: Deben comenzar con letra minúscula, y si está compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().

- c) Variables y objetos: las variables siguen la misma convención que los métodos.  
Por Ejemplo: alumno, padronElectoral.
  - d) Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "\_". Ejemplo: ANCHO, VACIO, COLOR\_BASE.
- 2) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.
  - 3) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.
  - 4) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios.
  - 5) Modularizar el código. No entregar 1 o 2 archivos, separar cada clase.
  - 6) No utilizar variables globales.
  - 7) Si cualquier estructura de control tiene 1 línea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    std::cout << i;  
}
```