­­­

**Práctica\_1\_3 v1 .** Estudio empírico del Algoritmo de ordenación por Inserción y comparación con ordenación por Burbuja

**GUIÓN DE LA PRÁCTICA**

**1.- Objetivos de la práctica**

El objetivo principal de la práctica es estudiar empíricamente por conteo de operaciones elementales y por tiempo de procesador el Algoritmo ordenación por inserción sobre *vectores* de enteros generados aleatoriamente y comparar después con otro algoritmo de ordenación. . Los análisis teóricos están incluidos en los contenidos de la asignatura.

1. El objetivo específico de la práctica es:
   * Comprobar el comportamiento del algoritmo de ordenación por inserción.
   * Utilizar distintas instancias de datos y calcular promedios.
   * Razonar sobre datos experimentales.
   * Comparar con el algoritmo de ordenación por burbuja.

Se va a realizar una experimentación para comprobar el comportamiento de un algoritmo en términos de tiempo, operaciones elementales y comportamiento teórico. Para ello hay que generar un conjunto de 5 vectores aleatorios para cada tamaño de 1000 a 5000 con posibilidad de añadir magnitudes variables.

**NO ES NECESARIO VOLVER A GENERAR LOS GRAFICOS, DEBE FUNCIONAR DIRECTAMENTE REFRESCANDO LOS DATOS DE LOS NUEVOS RESULTADOS.**

**2.- Actividades a realizar**

**Actividad 1**: Comprobar el algoritmo de Ordenación.

1. Crear un directorio **Practica\_1\_3**, crear proyecto nuevo “**actividad\_1**” copiar el fichero empirico.cpp a este directorio e incluirlo en el proyecto.
2. Modifíquelo para cambiar el algoritmo en ordena por el algoritmo de inserción:

void insercion(std::vector<int>& vec) {

int n = vec.size();

for (int i = 1; i < n; ++i) {

int clave = vec[i];

int j = i - 1;

while (j >= 0 && vec[j] > clave) {

vec[j + 1] = vec[j];

--j;

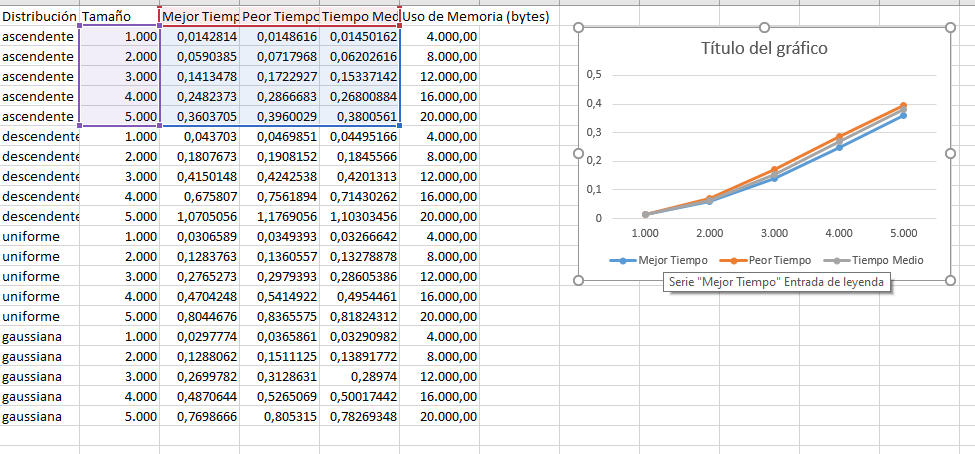
}

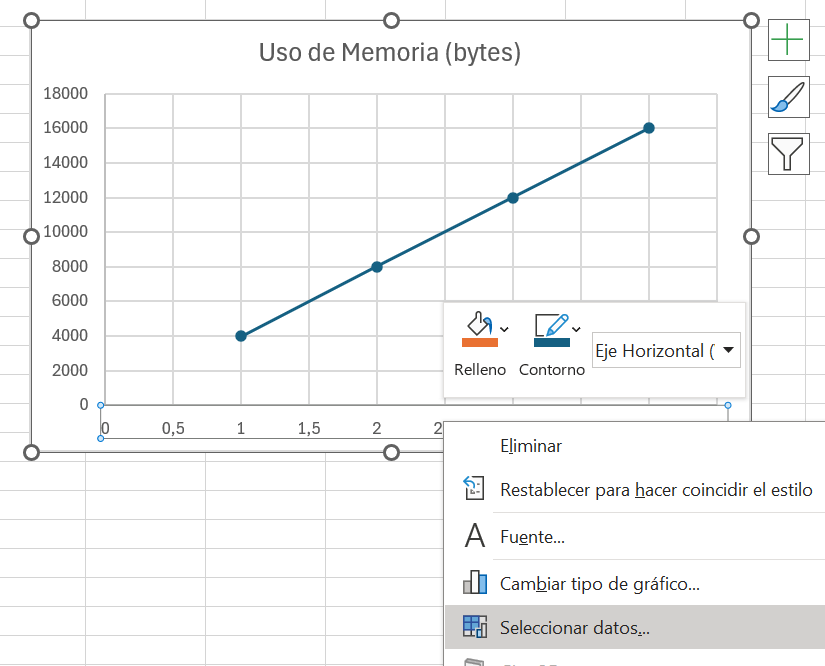
vec[j + 1] = clave;

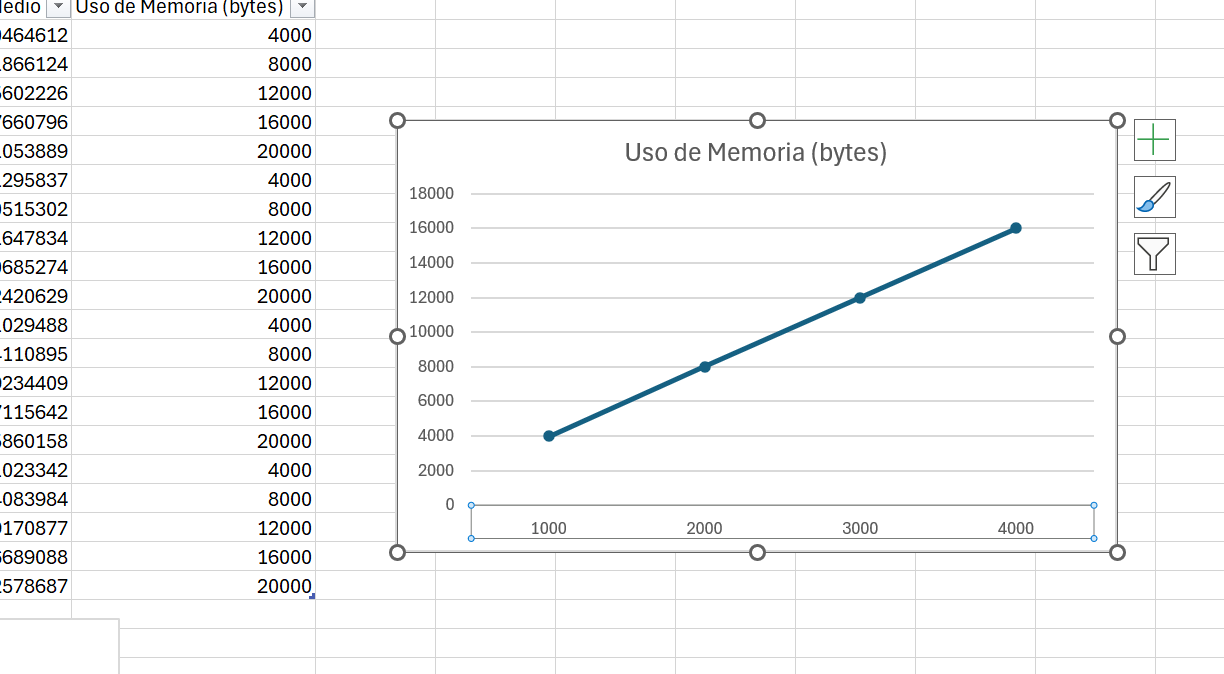
}

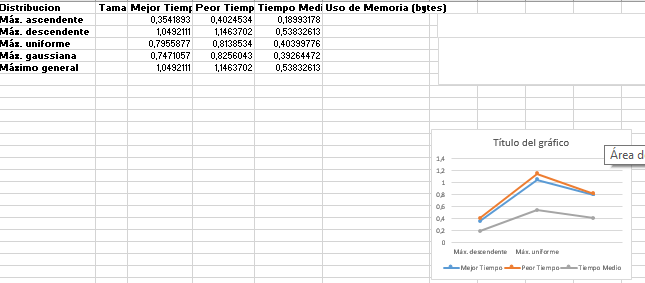
}

1. **COPIAR** **tiempos.xls** de la sesión anterior donde se refrescarán los datos .
2. Copiar los gráficos de tiempo, memoria y distribución a la hoja de trabajo, se muestran los gráficos de resumen de la sesión anterior (burbuja) como guía, los datos para el algoritmo de inserción serán diferentes.









**Actividad 2**: Contar las operaciones elementales de la ordenación

1. **Copiar** el proyecto y hoja Excel OE.xls de la “**actividad\_2**” de la sesión anterior el directorio Practica\_1\_3, modifíquelo para cambiar el algoritmo en ordena por el algoritmo de inserción.
2. Importe los datos de la ejecución al archivo Excel OE.xls
3. Incluya una copia de pantalla en la hoja de trabajo

**Actividad 3**: Calcular el comportamiento teórico del orden

1. Crear un fichero Excel **Teórico.xls** con una tabla y un gráfico de líneas con las tres series teóricas de mejor caso, peor caso y caso medio según lo visto en teoría
2. Añadir líneas de tendencia cuadrática, mostrar las fórmulas sobre el gráfico con las fórmulas del caso peor. medio y mejor según la teoría
3. Compárelo con las formas obtenidas para las operaciones elementales y el tiempo. Que se puede decir respecto al cambio de distribución. ¿Y al tamaño? . Que se puede decir de la comparación de ambos algoritmos en base a su orden en peor/mejor y caso medio?

**Actividad 4:** Comparar dos algoritmos de ordenación:

1. Realice un nuevo código comparación.cpp utilizando lo usado en prácticas anteriores para medir el tiempo medio con cuatro (12345,23451,34521,45123) semillas para la distribución uniforme y 5 tamaños (1000,2000,3000,4000,5000) que genera los tiempos del algoritmo de inserción y burbuja para cada ejecución de Semilla y tamaño , mostrando estas columnas más el tiempo del algoritmo de ordenación por Burbuja , Ordenación por Inserción, y una columna ganador, que muestra +1 si tarda menos burbuja y -1 si tarda menos inserción guardando los resultados en resultado.csv.
2. Cree una hoja Excel Comparación.xls e importe los y cree la casilla calculada sumando la columna ganador y Genere una tabla de subtotales con el promedio por cada salto en Tamaño
3. Cree el grafico de comparación y cópielo en la hoja de trabajo.

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

**3.- Entrega fin sesión**

Los archivos con los programas correspondientes a las actividades se deberán entregar, a través de Moodle, al finalizar la sesión de prácticas, junto con la Hoja de Trabajo del estudiante. Todo el código entregado debe compilar y ejecutar, comente aquel código que está pendiente para la entrega final de la practica para que no de errores de compilación.

Las líneas de código deben estar comentadas con su funcionalidad, tome como ejemplo los comentarios en el código proporcionado y las fórmulas utilizadas para el cálculo de las operaciones elementales en cada línea de código, así como las fórmulas teóricas utilizadas para el cálculo teórico.

**4.- Después de asistir al laboratorio**

* + Completar el código y el documento de trabajo que no haya dado tiempo a realizar en clase.
  + El código y documento entregado en esta entrega debe realizarse, aunque se haya terminado todo el trabajo en clase. El código debe ser una ampliación de lo realizado en clase no puede ser completamente nuevo.
  + Realizar la Entrega\_Practica\_1\_3 en un fichero comprimido con la hoja de trabajo y 4 subcarpetas (actividad\_1, actividad\_2 , actividad\_3 y actividad 4:
    - En la carpeta actividad\_1:
      * fichero empirico.cpp
      * resultados.csv
      * tiempo.xls (o extensión compatible Excel por defecto)
    - En la carpeta actividad\_2:
      * fichero empiricoOE.cpp , modificado para conteo de OE de inserción
      * resultados.csv
      * OE.xls (o extensión compatible Excel por defecto)
    - En la carpeta actividad\_3:
      * Teorico.xls (o extensión compatible Excel por defecto)
    - En la carpeta actividad\_4:
      * Comparacion.cpp
      * Resultados.csv
      * Comparacion.xls
    - Hoja de Trabajo