

### Indicaciones específicas:

- Esta evaluación contiene 4 páginas (incluyendo esta página) con 1 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - p1.cpp
- Deberás subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), uno en cada ejercicio. También puedes crear un .zip

### Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
  - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)
  - Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)
  - Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
  - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
  - Capacidad de aplicar conocimientos de ingeniería(nivel 2)
  - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	20	
Total:	20	

**1. (20 points) Mapa Tridimensional**

Recientemente Juan estuvo fascinado con el uso de los mapas para registrar ubicaciones importantes. Sabiendo su la posición estas ubicaciones, tiene una leve idea de qué tanto debería recorrer a cada punto partiendo desde una posición inicial. El problema es que pueden ser varias posiciones y varios los trayectos dependiendo de dónde quiere empezar su búsqueda. Luego de pensar mucho en ello, ha decidido calcular el punto en el espacio obtenido al promediar la coordenadas registradas en el mapa. Este punto es conocido como el centroide. La razón por la que quisiera conocer el centroide es porque tiene la idea de que es un punto de partida ideal para empezar la búsqueda de todos los puntos registrados en el Mapa. Ayude a Juan a conocer el centroide de los puntos guardados en el mapa.

Desarrolle los siguientes puntos en un modelo Orientado a objetos para simular la interacción de un Mapa:

1. Genere una clase **Mapa**, con atributo(s) **Nombre**, y **vector de Coordenadas**. La clase **Coordenada** tiene 2 atributos: **Descripcion**, y 3 doubles que representarán las coordenadas **X,Y** y **Z**. (4 pts.)
2. Así mismo, la clase Mapa deriva **MapaDelTesoro**, con atributo **cantidadTesoro**, la clase derivada **MapaATuCorazon**, con atributo **cantidadAmor**, y la clase derivada **MapaDeEsferasDelDragon**, con atributo **deseo**. La clase Mapa tiene un método **ImprimirDatosBasicos**, y las clases derivadas deben implementarlo de forma distinta. (4 pts.)
3. Defina un método que **Centroide** que determine el Centroides de los puntos del mapa. Recordemos que el centroide de un conjunto de puntos es el **promedio de las coordenadas**. Utilice una sobrecarga del operador **+** para sumar las coordenadas. (3 pts.)
4. Ahora genera 1 objeto tipo **MapaDelTesoro** con 3 Coordenadas como mínimo en el vector de coordenadas. Para el caso de **MapaATuCorazon**, crear 1 objeto con 4 Coordenadas como mínimo. Para el caso de **MapaDeEsferasDelDragon**, crear 1 objeto con exactamente 7 Coordenadas. Para agregar Coordenadas a los Mapas se debe hacer uso de sobrecarga del operador **<<**. Incluir un método que logre devolver el centroide de un Mapa (5 pts.)
5. **Crear un documento .txt** que se guarde la siguiente forma.

Nombre Mapa, Año mapa, cantidad de coordenadas, y centroide de las coordenadas. Son libres de decidir como guardan esta información en el archivo.

Mapa1, 3, (3, 4, 12),

Mapa2, 4, (1, 1.1, 1.2)

Solicitar al usuario el nombre del archivo .txt (4 pts.)

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100% corresponde al puntaje indicado en cada punto

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	La ejecución es correcta (1pts).	La ejecución no es correcta (0.5pts)
Código	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts).
Eficiencia	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts).	El código no está optimizado pero la ejecución no es deficiente (1pts).	El código no está optimizado y la ejecución es deficiente (0pts).