**Objetivos**

**Unidad 4: Grafos**

OE4.1. Explicar los conceptos básicos sobre la teoría de grafos.

OE4.2. Modelar la información de un problema utilizando un grafo como estructura de datos.

OE4.3. Aplicar los recorridos en profundidad y por niveles de los grafos en el contexto de un problema dado.

OE4.4. Aplicar los algoritmos de Dijkstra y Floyd-Warshall para resolver problemas de búsqueda de caminos más cortos en el contexto de un problema dado.

OE4.5. Aplicar los algoritmos de Prim y Kruskal para resolver problemas de árboles de recubrimiento mínimo en el contexto de un problema dado.

OE4.6. Diseñar y construir un grafo representado por matrices de adyacencias y listas de adyacencias.

OE4.7. Implementar los algoritmos de recorridos sobre grafos y búsqueda de caminos más cortos.

OE4.8. Diseñar y construir las pruebas unitarias de cada uno de los grafos implementados.

**Enunciado**

Usted debe desarrollar (analizar, diseñar e implementar) un programa que resuelva un problema específico que sea modelado utilizando grafos y que para su solución se apliquen al menos dos (2) de los algoritmos de grafos que se estudiarán durante el curso: Recorridos sobre Grafos (BFS, DFS), Caminos de Peso Mínimo (Dijkstra, Floyd-Warshall), Árbol de Recubrimiento Mínimo -MST- (Prim, Kruskal).

El problema debe ser definido por usted y su grupo de máximo 3 personas y los requisitos son los siguientes:

* Desarrollar 2 versiones de Grafo (su solución debe funcionar sin problema con las dos versiones, es decir, el programa debe admitir el cambio de la implementación utilizada en cualquier momento y funcionar bien indistintamente de la que se esté usando). Cada grafo debe ser desarrollado desde el TAD hasta las pruebas unitarias automáticas.
* Llevar a cabo y documentar cada una de las fases del método de la ingeniería para la solución del problema planteado.
* Documentar apropiadamente las fases de análisis y diseño con el documento de especificación de requerimientos, el diseño del TAD, diagramas de clase y objetos, y el diseño de los casos de pruebas de las pruebas unitarias automáticas.
* Su programa debe contar con una interfaz gráfica de usuario que permita utilizar las funcionalidades que responden a los requerimientos del problema.
* Implementar todos los algoritmos de grafos vistos en clase así no sean utilizados en su proyecto para la solución del problema.

Su proyecto puede ser un juego, un programa que gestione la solución de un problema de la vida real, entre otros. Otra posibilidad es que su problema sea de programación competitiva como los siguientes: [Ejemplo 1](https://drive.google.com/open?id=190gSaCJ7aHqFN2x5FZeCs3DDjiKJt6X3qMZFk9P1hWw), [Ejemplo 2](https://drive.google.com/open?id=1tu-SkR043v12yGPUHMw-lBDwCFR-CyOAkR9sB0g2Wd8).

**Primera Entrega. Martes 6 de Noviembre de 2018.**

La estructura Grafo completamente analizada, diseñada, implementada y probada, más una explicación detallada del problema a solucionar. Su entrega debe incluir:

1. Especificación del TAD Grafo. Nombre, representación, invariante, operaciones y la especificación de cada una de las operaciones en términos de entrada y salida (tal como se revisó en la Unidad 2 del curso).
2. Diseño de Diagrama de Clases del TAD Grafo incluyendo las dos implementaciones y las mejores prácticas de diseño (no olvidar desacoplamiento y generics).
3. Un diagrama de objetos con al menos un (1) objeto de cada uno de las clases de su diseño del grafo. En el caso de las clases asociadas a los nodos y las aristas si debería haber más de un objeto (no van a mostrar un grafo con un solo nodo y un loop!).
4. Diseño de pruebas unitarias:
   1. De las operaciones estructurales del grafo (agregar, eliminar y consultar).
   2. De los algoritmos vistos en clase.
5. El problema a solucionar:
   1. Enunciado suficientemente claro de **la situación problemática** que solucionarán.
   2. Especificación de Requerimientos Funcionales del programa que darán solución al problema.

**Segunda Entrega. Sábado 17 de Noviembre de 2018.**

1. Informe del seguimiento del Método de la ingeniería sobre el problema que están abordando.
2. Implementación completa de:
3. Los grafos incluyendo los algoritmos vistos en clase.
4. Las pruebas unitarias automáticas diseñadas sobre los grafos.

**Entrega Final. Lunes 26 de Noviembre de 2018.**

Los entregables de la primera y segunda entrega más el diseño, la implementación y las pruebas unitarias del programa que da solución al problema. Esta es la rúbrica de evaluación de la entrega final: [Rúbrica Proyecto](https://docs.google.com/spreadsheets/d/1iJ6ltcr4L4gE4tkgP6OBb1iAxTJEd_ykb-pYhYX3bjE/edit?usp=sharing).