

**PARTE 2: Ejercicios de aplicación**  
**(se puede tener material MANUSCRITO PERSONAL)**  
**Duración: 60 minutos**

**EJERCICIO 1**

Una empresa constructora se encuentra desarrollando un barrio suburbano residencial, en el que las casas se encuentran dispersas en forma no estructurada sobre un área relativamente extensa. El subcontratista eléctrico ha presentado un plano de conexión eléctrica de las residencias con un costo que ha sido juzgado demasiado alto, ya que, a juicio del CEO de la empresa constructora, no optimiza la cantidad de cable necesaria.

Consultado el Gerente de Sistemas de Información, dice tener la solución necesaria para resolver el problema, solución conocida por cualquier ingeniero o licenciado en informática, y nos encarga el desarrollo de esa funcionalidad. Solamente necesita tener un plano del terreno en el que se puedan medir las distancias entre todas las casas y la sub estación de energía eléctrica.

1. Describe en lenguaje natural la solución al problema (solución general y modelo)
2. Define las estructuras de datos necesarias para representar el TDA y eventualmente las auxiliares necesarias para la resolución.
3. Escribe un pseudocódigo de alto nivel sobre las estructuras definidas.
4. Analiza el orden del tiempo de ejecución de la aplicación escrita.

**EJERCICIO 2**

Una compañía de transporte aéreo ha descubierto que la consulta sobre vuelos más frecuente que hacen los usuarios es la del vuelo con menos escalas entre un origen y un destino. Por ello, se le ha encargado al Departamento de Sistemas que desarrolle una aplicación para que la consulta sobre qué combinación de vuelos une dos aeropuertos con la menor cantidad de escalas posibles se resuelva de la forma más eficiente posible.

Sabiendo que el modelo usado para representar los aeropuertos y los vuelos es un grafo dirigido, desarrolla un método con la siguiente firma, que se ejecute en un **orden del tiempo de ejecución lineal** respecto de la cantidad de vuelos:

En Grafo Dirigido

TLista <aeropuerto> menosEscalas (Comparable origen, Comparable destino)

Nota:

Cada aeropuerto cuenta con un atributo de tipo aeropuerto, que puede ser usado para cargar allí el predecesor (anterior) a él en el camino que se busca.