

Algoritmos y Estructuras de Datos II

**PROYECTO INDIVIDUAL – PARTE I**

Gustavo Germán Ríos

Contenido

[Introducción 3](#_Toc417411916)

[Problema planteado 3](#_Toc417411917)

[Análisis de alternativas 3](#_Toc417411918)

[Algoritmos 3](#_Toc417411919)

[Selección y justificación de alternativa a implementar 3](#_Toc417411920)

[Conclusiones 3](#_Toc417411921)

[Guía del usuario 3](#_Toc417411922)

# Introducción

Se realizará la implementación de un software para la aerolínea AlasUcu, qué en base a la información proporcionada de aeropuertos, trayectos, horarios y costos sea capaz de facilitar al usuario la ruta optima entre dos aeropuertos.

## Problema planteado

Dada una base de datos, que contiene información de vuelos entre aeropuertos, los horarios de dichos vuelos, y el costo; el usuario deberá de escoger en pantalla el origen y el destino, y se le deberá ofrecer entre todos los vuelos que tengan el lapso horario menor, el que tenga menor costo.

# Análisis de alternativas

Para la implementación del software solicitado, se evalúan dos alternativas como estructura de datos principal. Ambas son utilizando un grafo dirigido, con un arco entre el aeropuerto A y el B, si es que existe un vuelo directo de A a B, teniendo alcance a su vez al costo que implica dicho vuelo, la hora de salida y la hora de llegada.

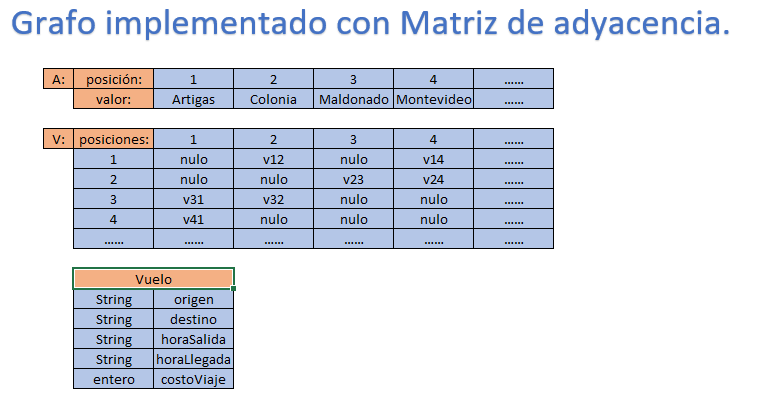
### Alternativa 1:

Implementación de grafo con Matriz de vuelos:

Dada la cantidad de aeropuertos **N** se crea el grafo, el cual consta de una matriz de vuelos **V** de **NxN**, donde en la posición **V[i,j]** tenemos nulo si no existe un vuelo entre el aeropuerto **i** y el **j** y un objeto **Vuelo** si es que existe dicho trayecto de forma directa.

La clase **Vuelo** consta de la información necesaria para obtener los costos, y horarios de dichos vuelos.

A su vez necesitamos un vector **A**, que asocie los aeropuertos con los índices respectivos de la matriz.



### Alternativa 2:

* Descripción de al menos 2 alternativas de estructuras de datos para representar el problema. Se sugiere utilizar dibujos explicativos

### Algoritmos

* Descripción en seudocódigo (de acuerdo a los estándares de la asignatura) de las diferentes funcionalidades que el programa debe cumplir, para cada alternativa de representación indicada arriba.
* Análisis del orden del tiempo de ejecución de cada funcionalidad, para cada alternativa de representación indicada arriba.

# Selección y justificación de alternativa a implementar

Breve fundamentación de la elección.

Descripción detallada de la alternativa implementada (dibujos, diagramas, aclaraciones sobre clases e interfaces).

# Conclusiones

Resumen de las características más relevantes del producto. ¿Por qué debería el cliente elegir mi solución?

# Guía del usuario

* Instrucciones (breves) para el uso de la aplicación.
* Consideraciones de instalación (si corresponde) y requerimientos no funcionales para la operación.