

Fase 3.

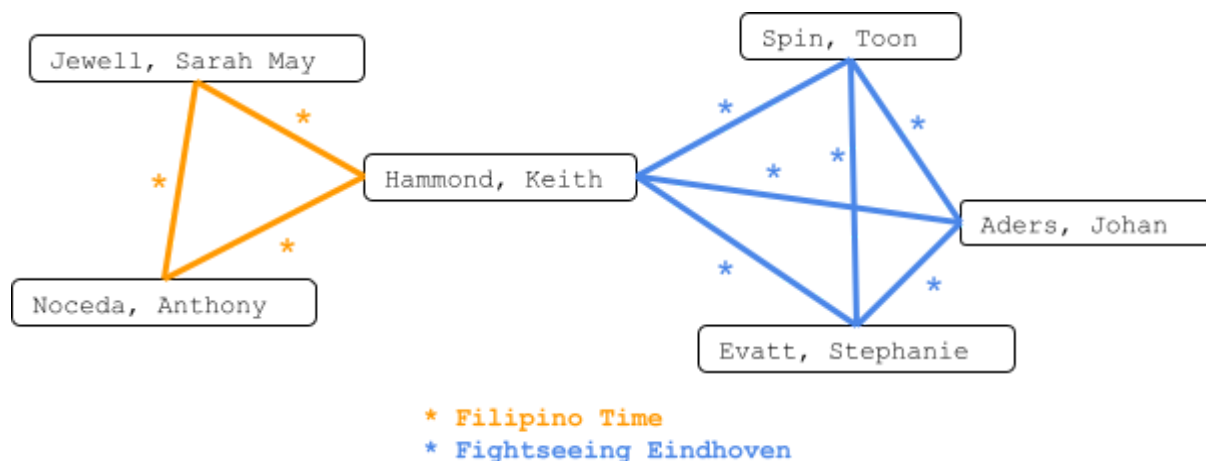
Objetivo:

- Utilizar las estructuras tabla hash y grafo, así como algoritmos analizados en clase en el contexto de un proyecto.
- Desarrollar algoritmos eficientes que utilicen dichas estructuras.

Enunciado:

En esta fase vamos a trabajar en dos aspectos:

- (1) Queremos representar el conjunto de intérpretes del catálogo mediante una nueva representación: una **tabla hash**.
- (2) De las fases anteriores sabemos que cada intérprete tiene asociada la lista de películas en que participa, y cada película la lista de intérpretes que participan en ella. Consideraremos que dos intérpretes son adyacentes si participan juntos en alguna película. Así, implícitamente, la información que tenemos almacenada forma un grafo de intérpretes:



Por ejemplo, en el grafo superior (no es el grafo real), “Noceda, Anthony” y “Hammond, Keith” son adyacentes, puesto que ambos participan en la película “Filipino Time”.

Queremos incorporar al sistema dos nuevas funcionalidades:

- Dados los nombres de dos intérpretes, calcular su distancia mínima. En el ejemplo, la distancia mínima entre “Noceda, Anthony” y “Evatt, Stephanie” es **2** (ambos han trabajado con “Hammond, Keith”).
- Dados los nombres de dos intérpretes, imprimir el camino más corto entre ambos, si es que existe. En el ejemplo, el camino más corto entre “Noceda, Anthony” y “Evatt, Stephanie” es: [**Noceda, Anthony ; Hammond, Keith ; Evatt, Stephanie**]

IMPORTANTE:

- NO tenéis que crear un nuevo grafo, sino manejar como si fuera un grafo la información que ya tenéis almacenada.
- En esta fase se recomienda utilizar los ficheros “small” para agilizar la realización de pruebas.

Tareas:

- a) Crear la clase **HashMapInterpretes**, necesaria para representar los intérpretes del catálogo como una tabla hash. Implementar en ella los métodos necesarios para gestionar los intérpretes, determinando el orden temporal de cada uno de ellos. Realizar las modificaciones pertinentes en el código de la Fase 2 para utilizar esta nueva representación.
- b) Implementar el siguiente método en la clase **Interprete** y determinar su orden temporal:

```
/**
 * Devuelve un HashSet con todos los adyacentes del intérprete, es decir,
 * aquellos intérpretes con los que ha participado en alguna película.
 * @return: el HashSet con los intérpretes que son adyacentes.
 */
public HashSet<Interprete> obtenerAdyacentes();
```

- c) Desarrollar los siguientes métodos en la clase **CatalogoIMDB** y determinar el orden temporal de cada uno de ellos:

```
/**
 * Devuelve la distancia mínima entre dos intérpretes dados.
 * @param inter1: nombre del primer intérprete
 * @param inter2: nombre del segundo intérprete
 * @return: distancia mínima entre ambos intérpretes. En caso de que no
 *          estén conectados, devuelve -1.
 */
public int distancia(String inter1, String inter2)

/**
 * Imprime el camino más corto entre dos intérpretes. Si no existe camino,
 * imprime un mensaje indicando este hecho.
 * @param inter1: nombre del primer intérprete
 * @param inter2: nombre del segundo intérprete
 */
public void imprimirCamino(String inter1, String inter2)
```

- d) Modificar la clase **AplicacionIMDB** para que trabaje con la nueva implementación de la interface **InterfaceInterpretes**. Además, se le ofrecerán dos nuevas funcionalidades al usuario:
- i) Cuando escoja la opción 5, se le pedirá que introduzca dos nombres de intérprete y se le indicará cuál es la distancia mínima entre dichos intérpretes.
 - ii) Cuando escoja la opción 6, se le pedirá que introduzca dos nombres de intérprete y se le mostrará el camino más corto entre dichos intérpretes.
- e) Responder a la siguiente pregunta justificando tu respuesta: *¿Dónde más habría sido adecuado utilizar una tabla hash? ¿Qué beneficio se habría obtenido con ello?*

Entregables:

- Código completo
- Documentación que incluya:
 - Portada, con nombres de los miembros del grupo y fecha
 - Diagrama de clases completo
 - Por cada método implementado:
 - Algoritmo
 - Casos de prueba considerados
 - Orden temporal
 - Respuesta al apartado (e)

Fecha de entrega: 9 de enero