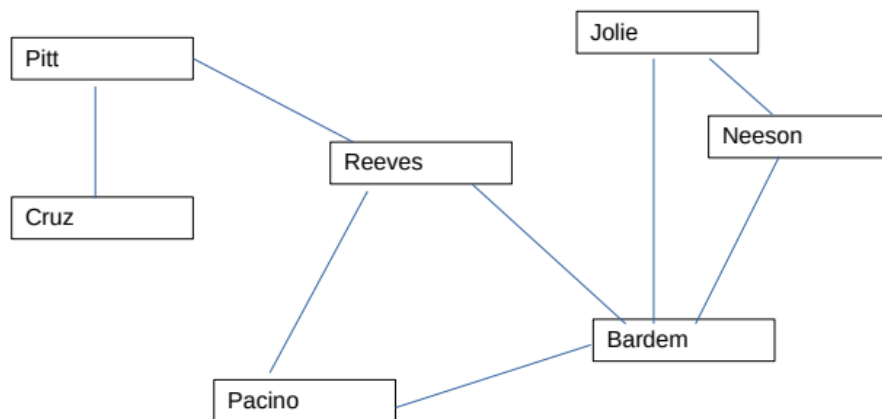


Actividad 3.

Objetivo final: obtener un sistema que permitirá hacer eficientemente (deberemos razonar los motivos de la eficiencia) las siguientes operaciones:

boolean estanConectados(a1, a2)

El resultado será true si hay una cadena de relaciones que une a a1 y a2. Por ejemplo, la llamada a `estanConectados("Neeson", "Cruz")` dará true.



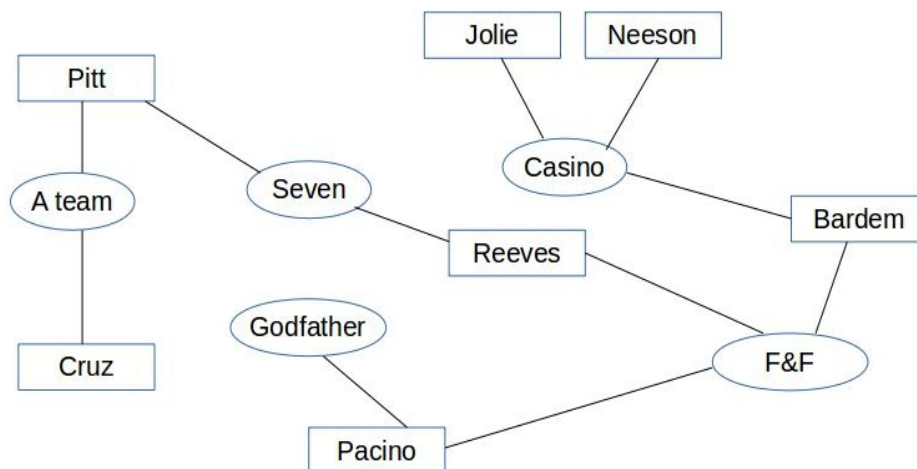
Para poder resolver este problema, previamente, deberemos obtener una estructura de datos adecuada, a partir de los datos de la lista de actores obtenida en la fase 1 de la práctica. Los nodos del grafo estarán representados por los nombres de los actores y dos actores/actrices estarán conectados por medio de un arco si han participado en la misma película. Se utilizará la siguiente estructura de datos:

* Tabla hash con una lista de elementos relacionados para cada elemento:

HashMap<String, ArrayList<String>>

Clave	Dato
Pitt	<Cruz, Reeves>
Cruz	<Pitt>
Reeves	<Pitt, Pacino, Bardem>
Jolie	<Neeson, Bardem>
Neeson	<Jolie, Bardem>
Pacino	<Reeves, Bardem,>
Bardem	<Pacino, Reeves, Jolie, Neeson>

El grafo anterior se debe sacar de la estructura de datos de la práctica 1:



En esa estructura se ve que Cruz y Pitt están relacionadas porque compartieron la película “A team”. Jolie, Neeson y Bardem están relacionados entre sí porque participaron en “Casino” y así sucesivamente.

```

public class GraphHash {

    HashMap<String, ArrayList<String>> g;

    public void crearGrafo(ListaActricesActores lActores)
        // Post: crea el grafo desde la lista de actores
        //       Los nodos son nombres de actores

        // COMPLETAR CÓDIGO

    public void print(){
        int i = 1;
        for (String s: g.keySet()){
            System.out.print("Element: " + i++ + " " + s + " --> ");
            for (String k: g.get(s)){
                System.out.print(k + " ### ");
            }
            System.out.println();
        }
    }

    public boolean estanConectados(String a1, String a2)
        // COMPLETAR CÓDIGO
}
  
```

OPCIONAL: `ArrayList<String> estanConectados(a1, a2)`

El resultado será una lista de relaciones desde a1 hasta a2, donde cada relación indica que esas 2 personas están conectadas por haber trabajado en una misma película. Por ejemplo, la llamada a `estanConectados("Neeson", "Cruz")` producirá:

<Neeson>, <Bardem>, <Reeves>, <Pitt>, <Cruz>

Se deberá entregar:

- Programas que implementen lo pedido (ejecutados correctamente). **Se deberá demostrar que el programa funciona realmente con conjuntos de datos no triviales (es decir, procesando todo el fichero original).**
- Se deberá especificar de manera clara **el número de relaciones** (llamadas a estarConectados) que se calculan **en un periodo de tiempo** (un minuto, una hora, ...).
- Documentación describiendo el problema planteado, las alternativas examinadas, implementaciones, y eficiencia.
- Además debéis rellenar y entregar el Checklist para verificar que habéis realizado todo lo que se os pide.