

# INF-253 Lenguajes de Programación

## Tarea 5: Prolog

15 de junio de 2023

### 1. Objetivo

Deberán implementar el siguiente grafo utilizando Prolog, y a partir de este deberán ser capaces de responder una serie de consultas.

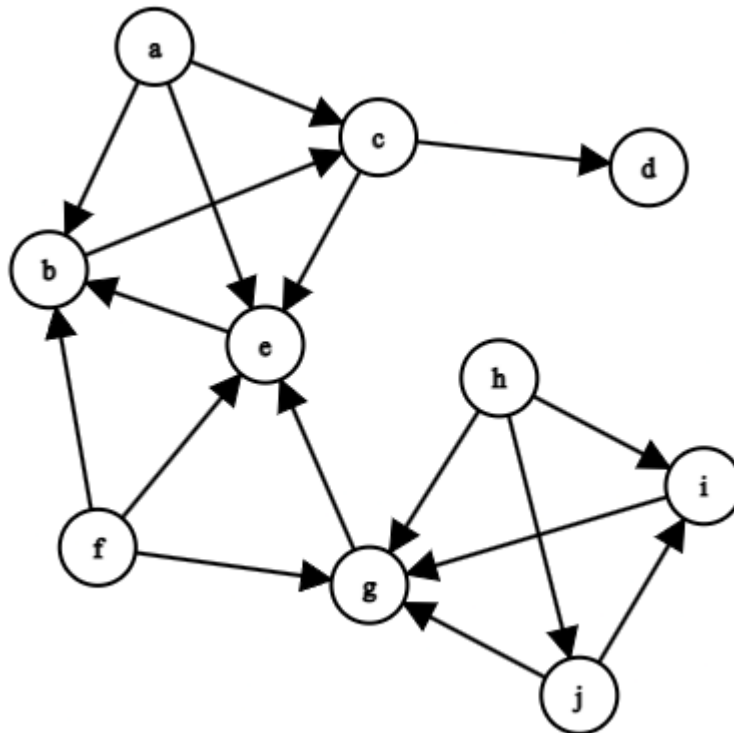


Figura 1: Grafo a implementar en Prolog

El grafo debe ser implementado utilizando predicados, por ejemplo el siguiente código indicaría que a es vecino de b y b es vecino de c:

---

```
1 vecino(a, b).  
2 vecino(b, c).
```

---

Para poder realizar la tarea, se debe utilizar SWI-Prolog, el cual se puede encontrar en: <https://www.swi-prolog.org/download/stable>.

## 2. Consultas a implementar

### 1. Puede llegar?

Debe implementar el predicado `puedellegar(X, Y)` tal que `X` e `Y` y exista un camino desde `X` hasta `Y`. (Cuando hay múltiples posibles opciones el orden en que las entrega no importa, lo importante es que las entregue todas)

---

```
1 ?- puedellegar(a, b)
2 true
3
4 ?- puedellegar(a, X)
5 X = b
6 X = c
7 X = d
8 X = e
9
10 ?- puedellegar(X, g)
11 X = f
12 X = h
13 X = i
14 X = j
```

---

### 2. Vecinos

Debe implementar el predicado `vecinos(X, L)` tal que `X` sea un nodo del grafo y `L` sea una lista con todos los vecinos del nodo `X`.

---

```
1 ?- vecinos(f, [b, e, g])
2 true
3
4 ?- vecinos(f, L)
5 L = [b, e, g]
```

---

### 3. Camino valido

Debe implementar el predicado `caminovalido(L)` tal que `L` sea una lista de nodos y todo par de nodos adyacentes en la lista sean vecinos, es decir, los nodos en la lista forman un camino en el grafo.

---

```
1 ?- caminovalido([a])
2 true
3
4 ?- caminovalido([a, b])
5 true
6
7 ?- caminovalido([a, b, c, e, b])
8 true
9
10 ?- caminovalido([f, b, a])
11 false
```

---

### 4. Camino más corto

Debe implementar el predicado `caminomascorto(X, Y, L)` tal que `X` e `Y` sean nodos en el grafo y `L` sea la lista de nodos que forma el camino más corto desde `X` hasta `Y`.

---

```
1 ?- caminomascorto(a, d, [a, c, d])
2 true
3
4 ?- caminomascorto(h, d, L)
5 L = [h, g, e, b, c, d]
```

---

### 3. Sobre la Entrega

- Se deberá entregar un único archivo con todos los predicados implementados en el orden descrito en el enunciado.
- Cuidado con el orden y la indentación de su tarea, llevará descuento de lo más 20 puntos.
- **Todas las funciones deben ser comentadas. Se descontará por función sin comentar.**
- Se debe trabajar de forma individual obligatoriamente.
- **La entrega debe entregarse en .tar.gz y debe llevar el nombre: Tarea5LP\_RolAlumno.tar.gz**
- **El archivo README.txt debe contener nombre y rol del alumno e instrucciones detalladas para la correcta utilización de su programa. De no incluir README se realizará un descuento.**
- La entrega será vía aula y el plazo máximo de entrega es hasta el **23 de Junio a las 23:59.**
- **Por cada día de atraso se descontarán 20 pts.**
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- **Solo se contestarán dudas realizadas en AULA y que se realicen al menos 48 horas antes de la fecha de entrega original.**

### 4. Calificación

#### 4.1. Entrega mínima

La entrega mínima deberá contener la implementación del grafo en forma de predicados y la implementación de `puedellegar`.

#### 4.2. Entrega

- Entrega minima (30pts)
- vecinos (20pts)
- caminovalido (25pts)
- caminomascorto (25pts)

### 4.3. Descuentos

- Falta de comentarios (-5 pts c/u Max 20 pts)
- Falta de README (-20 pts)
- Falta de alguna información obligatoria en el README (-5 pts c/u)
- Falta de orden (entre -5 y -20 pts dependiendo de que tan desordenado)
- Día de atraso (-20 pts por día, -10 pts dentro de la primera hora)
- Mal nombre en algún archivo entregado (-5 pts c/u)