Ejemplos de Prolog

- **Clase:** Paradigmas de Programación
- **Alumno:** Fernando Haro Calvo

Ejemplo 1

```
% Hechos
food(burger).
food(sandwich).
food(pizza).
lunch(sandwich).
dinner(pizza).

% Reglas
meal(X) :- food(X).
```

Consultas

```
?- food(pizza).
```

Devuelve `true.`, ya que food(pizza) es un hecho que fue declarado.

```
?- meal(X), lunch(X).
```

Devuelve `X = sandwich.`, ya es el único átomo que cumple con las dos condiciones de la consulta. Prolog llega a esta conclusión comparando uno por uno los hechos y reglas con la consulta.

```
| ?- meal(X), lunch(X).
      1    1  Call: meal(_23) ?
      2    2  Call: food(_23) ?
      2    2  Exit: food(burger) ?
      1    1  Exit: meal(burger) ?
      3    1  Call: lunch(burger) ?
      3    1  Fail: lunch(burger) ?
      1    1  Redo: meal(burger) ?
      2    2  Redo: food(burger) ?
      2    2  Exit: food(sandwich) ?
      1    1  Exit: meal(sandwich) ?
      3    1  Call: lunch(sandwich) ?
      3    1  Exit: lunch(sandwich) ?

X = sandwich ? ;
      1    1  Redo: meal(sandwich) ?
      2    2  Redo: food(sandwich) ?
      2    2  Exit: food(pizza) ?
      1    1  Exit: meal(pizza) ?
      3    1  Call: lunch(pizza) ?
      3    1  Fail: lunch(pizza) ?

no
```

```
?- dinner(sandwich).
```

Devuelve `false.`, ya que no existe un hecho que cumpla con la consulta.

Ejemplo 2

```
% Hechos
studies(charlie, csc135).
studies(olivia, csc135).
studies(jack, csc131).
studies(arthur, cs134).
teaches(kirke, csc135).
teaches(collins, csc131).
teaches(collins, csc171).
teaches(juniper, csc134).

% Reglas
professor(X, Y) :- teaches(X, C), studies(Y, C).
```

Consultas

```
?- studies(charlie, What).
```

Devuelve `What = csc135.`, ya que solo hay un hecho que cumple con la consulta.

```
?- professor(kirke, Who).
```

Primero, devolverá `Who = charlie.`, ya que charlie estudia csc135 y kirke enseña csc135. Luego, si continuamos con la consulta devolverá también `Who = olivia.`, ya que olivia también estudia csc135. El camino que sigue Prolog para llegar a estas conclusiones es el siguiente:

```
{trace}
| ?- professor(kirke, Who).
      1    1  Call: professor(kirke,_23) ?
      2    2  Call: teaches(kirke,_92) ?
      2    2  Exit: teaches(kirke,csc135) ?
      3    2  Call: studies(_23,csc135) ?
      3    2  Exit: studies(charlie,csc135) ?
      1    1  Exit: professor(kirke,charlie) ?

Who = charlie ? ;
      1    1  Redo: professor(kirke,charlie) ?
      3    2  Redo: studies(charlie,csc135) ?
      3    2  Exit: studies(olivia,csc135) ?
      1    1  Exit: professor(kirke,olivia) ?

Who = olivia ? ;
      1    1  Redo: professor(kirke,olivia) ?
      3    2  Redo: studies(olivia,csc135) ?
      3    2  Fail: studies(_23,csc135) ?
      1    1  Fail: professor(kirke,_23) ?

(3 ms) no
{trace}
```