

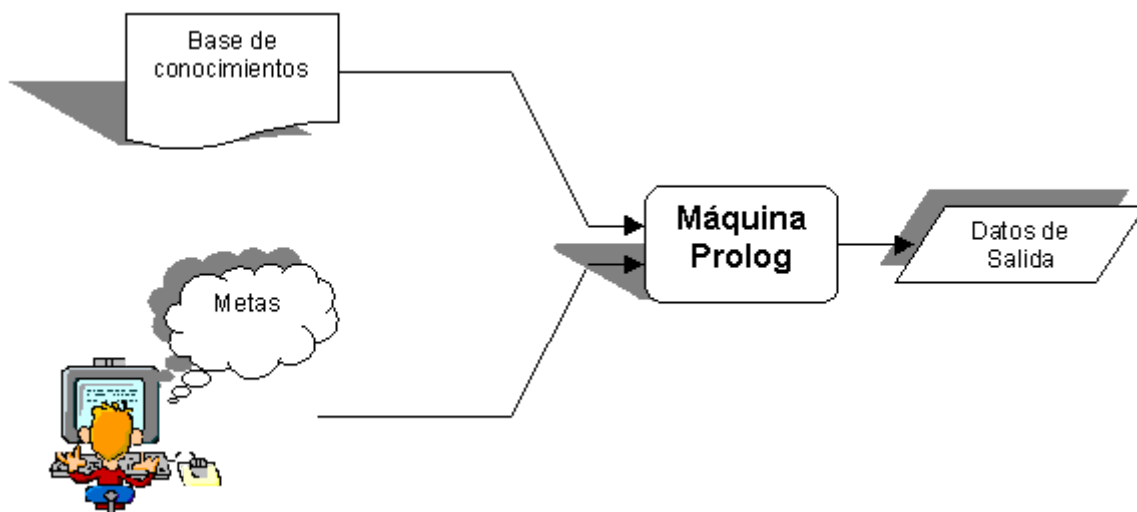


**INSTITUTO TECNOLÓGICO BELTRÁN**  
Centro de Tecnología e Innovación

Informe Práctico

# Trabajo Práctico N.º 4

## Gramáticas de cláusulas definidas en Prolog



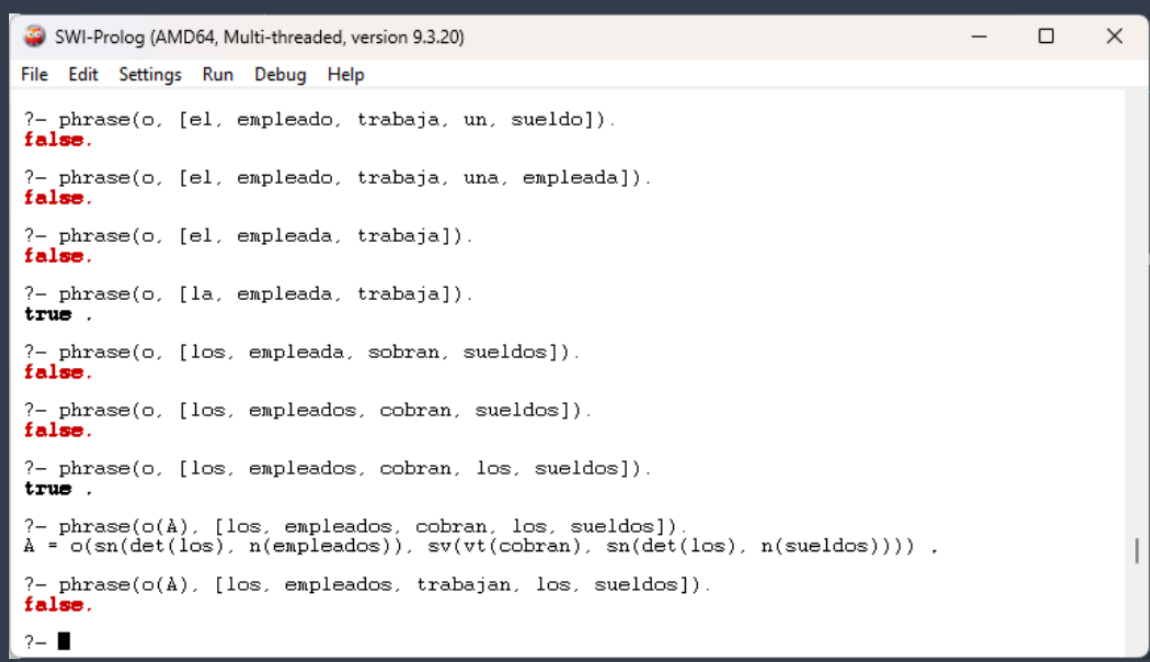
Nombre: Coral Tolazzi  
Tema: Lenguajes y Autómatas - Gramáticas  
Profesora: Yanina Ximena Scudero  
Cuatrimestre y Año: 1 Cuatrimestre del 2025

Instituto tecnológico Beltrán  
Procesamiento del Lenguaje Natural

## Realizar programa en Prolog, base de conocimiento y árbol sintáctico con las siguientes palabras:

*empleada/o, trabaja/n, cobra/n, sueldo/s*

- Programar en Prolog las gramáticas.
- Convertir las categorías gramaticales en predicados de Prolog con argumentos que indican el género y el número.
- Construir el árbol sintáctico.
- Realizar Consultas con los siguientes resultados:



```
SWI-Prolog (AMD64, Multi-threaded, version 9.3.20)
File Edit Settings Run Debug Help

?- phrase(o, [el, empleado, trabaja, un, sueldo]).
false.

?- phrase(o, [el, empleado, trabaja, una, empleada]).
false.

?- phrase(o, [el, empleada, trabaja]).
false.

?- phrase(o, [la, empleada, trabaja]).
true.

?- phrase(o, [los, empleada, sobran, sueldos]).
false.

?- phrase(o, [los, empleados, cobran, sueldos]).
false.

?- phrase(o, [los, empleados, cobran, los, sueldos]).
true.

?- phrase(o(A), [los, empleados, cobran, los, sueldos]).
A = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

?- phrase(o(A), [los, empleados, trabajan, los, sueldos]).
false.

?- 
```

## Como esta formada la base de conocimiento:

Reglas:

$o(o(SN, SV)) \rightarrow sn(SN, \_Gen, Num), sv(SV, Num).$

- Una **oración (o)** está compuesta por un **sintagma nominal (sn)** y un **sintagma verbal (sv)**.
- El género del **sn** se ignora para la oración, pero el número (**Num**) debe coincidir con el del **sv**.

Sintagma Nominal (**sn**):

$sn(sn(DET, N), Gen, Num) \rightarrow det(DET, Gen, Num), n(N, Gen, Num).$

- Un **sn** está compuesto por un **determinante (det)** y un **nombre (n)**.
- El género (**Gen**) y número (**Num**) deben coincidir entre determinante y nombre.

Sintagma Verbal (**sv**):

$sv(sv(VT, SN), Num) \rightarrow vt(VT, Num), sn(SN, \_Gen, \_Num).$

$sv(sv(VI), Num) \rightarrow vi(VI, Num).$

- Puede ser:
  - Verbo transitivo (**vt**) seguido de un **sn** (como "cobra un sueldo").
  - Solo un verbo intransitivo (**vi**) (como "trabaja").

Determinantes (**det**) por género y número:

$det(det(X), f, sg) \rightarrow [la, una].$

$det(det(X), f, pl) \rightarrow [las, unas].$

$det(det(X), m, sg) \rightarrow [el, un].$

$det(det(X), m, pl) \rightarrow [los, unos].$

Verbos:

$vi(vi(trabaja), sg) \rightarrow [trabaja].$

$vi(vi(trabajan), pl) \rightarrow [trabajan].$

$vt(vt(cobra), sg) \rightarrow [cobra].$

$vt(vt(cobran), pl) \rightarrow [cobran].$

Sustantivos:

$n(n(empleada), f, sg) \rightarrow [empleada].$

$n(n(empleadas), f, pl) \rightarrow [empleadas].$

$n(n(X), m, sg) \rightarrow [empleado, sueldo].$

$n(n(X), m, pl) \rightarrow [empleados, sueldos].$

```

C: > prolog > Ejercicio_11.pl
1  o(o(SN,SV)) --> sn(SN,_Gen,Num), sv(SV,Num).
2  sn(sn(DET,N),Gen,Num) --> det(DET,Gen,Num), n(N,Gen,Num).
3  sv(sv(VT,SN),Num) --> vt(VT,Num), sn(SN,_Gen,_Num).
4  sv(sv(VI),Num) --> vi(VI,Num).
5  det(det(X),f,sg) --> [X], {member(X,[la,una])}.
6  det(det(X),f,pl) --> [X], {member(X,[las,unas])}.
7  det(det(X),m,sg) --> [X], {member(X,[el,un])}.
8  det(det(X),m,pl) --> [X], {member(X,[los,unos])}.
9  vi(vi(trabaja),sg) --> [trabaja].
10 vi(vi(trabajan),pl) --> [trabajan].
11 vt(vt(cobra),sg) --> [cobra].
12 vt(vt(cobran),pl) --> [cobran].
13 n(n(empleada),f,sg) --> [empleada].
14 n(n(empleadas),f,pl) --> [empleadas].
15 n(n(X),m,sg) --> [X], {member(X,[empleado,sueldo])}.
16 n(n(X),m,pl) --> [X], {member(X,[empleados,sueldos])}.

```

#### Para las consultas se utilizó:

- `phrase(o(_), [la, empleada, trabaja])`  
Utiliza una variable anónima (\_). Esta consulta solo verifica si la oración es válida según la gramática. Devuelve true o false, sin mostrar la estructura sintáctica.
- `phrase(o(O), Oracion)`  
Usa una variable normal (O). Además de verificar la validez, Prolog unifica O con la estructura sintáctica generada. Es útil para analizar cómo se compone la oración internamente.
- `_` → solo verificación (válida o no).
- `O` → análisis y estructura completa de la oración.

```

?-
% c:/prolog/Ejercicio_11.pl compiled 0.00 sec, 16 clauses
?- phrase(o(_), [el, empleado, trabaja, un, sueldo]).
false.

?- phrase(o(_), [el, empleado, trabaja, una, empleada]).
false.

?- phrase(o(_), [el, empleda, trabaja]).
false.

?- phrase(o(_), [la, empleada, trabaja]).
true.

?- phrase(o(_), [los, empleada, sobran, sueldos]).
false.

?- phrase(o(_), [los, empleados, cobran, sueldos]).
false.

?- phrase(o(_), [los, empleados, cobran, los, sueldos]).
true.

?- phrase(o(A), [los, empleados, cobran, los, sueldos]).
A = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

?- phrase(o(_), [los, empleados, trabajan, los, sueldos]).
false.

?- ■

```