|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Práctica de Laboratorio 02**  **Escuela Profesional de Ciencia de la Computación Lenguajes de Programación - 2022-2** | | | | |
| **TIPO DE PRÁCTICA:** | **INDIVIDUAL** | *Mauricio Carazas Segovia* | | |
| **GRUPAL** |  | **MÁXIMO DE ESTUDIANTES** |  |
| **Objetivo:** Reforzar los conceptos de programación lógica.  **Método de entrega:** Muéstreselo al profesor una vez que hayas terminado. | | | | |

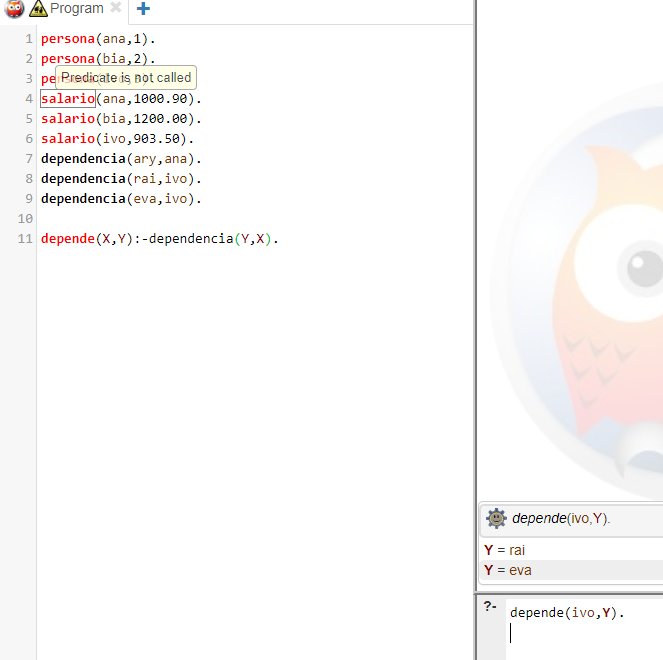
# EJERCICIO Nro 1

La siguiente tabla relaciona a cada empleado de una empresa su código, su salario y sus dependientes.

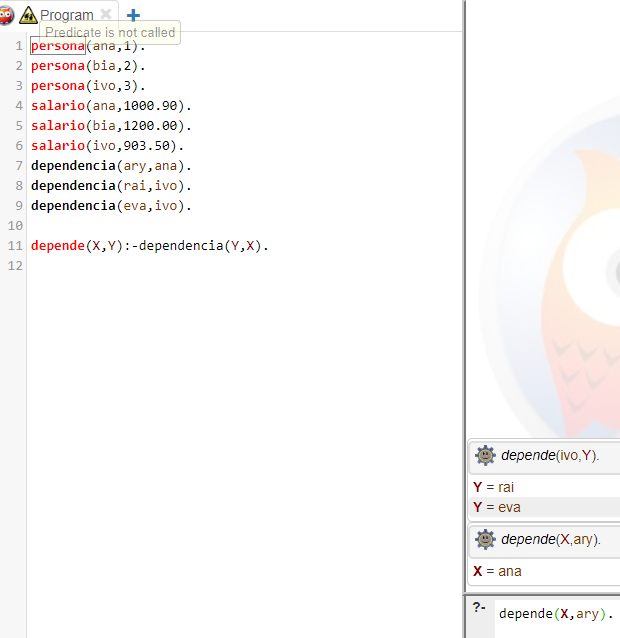
|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Nombre** | **Salario** | **Dependientes** |
| 1 | Ana | $ 1000,90 | Ary |
| 2 | Bia | $ 1200,00 | ---------- |
| 3 | Ivo | $ 903,50 | Raí, Eva |

Cree una base de conocimientos que represente esta tabla.

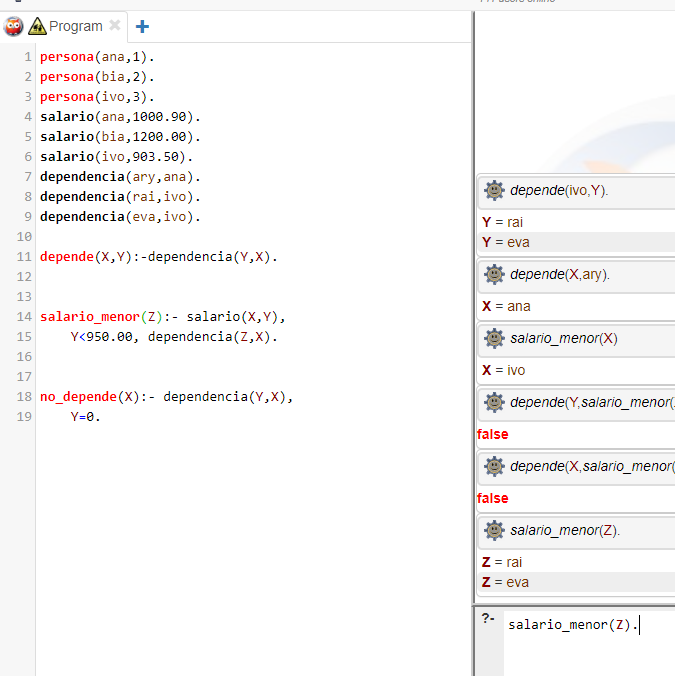
1. Consultar el sistema para recuperar dependientes de Ivo



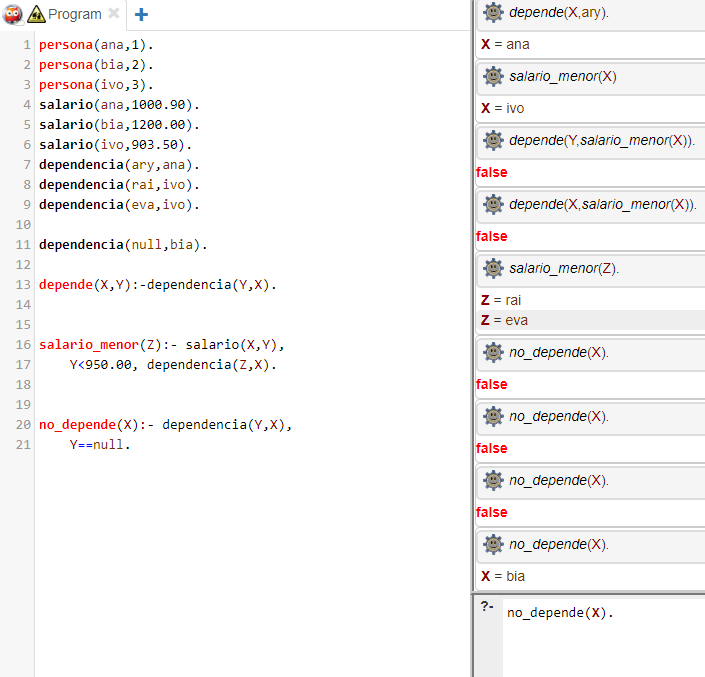
1. Crear consultas para descubrir de quién depende Ary



1. Crear reglas para descubrir quién depende de funcionario con salario inferior a $ 950,00



1. Consulte el sistema para encontrar funcionarios que no tengan dependientes ( tal vez usted necesite del operador de negación: not )

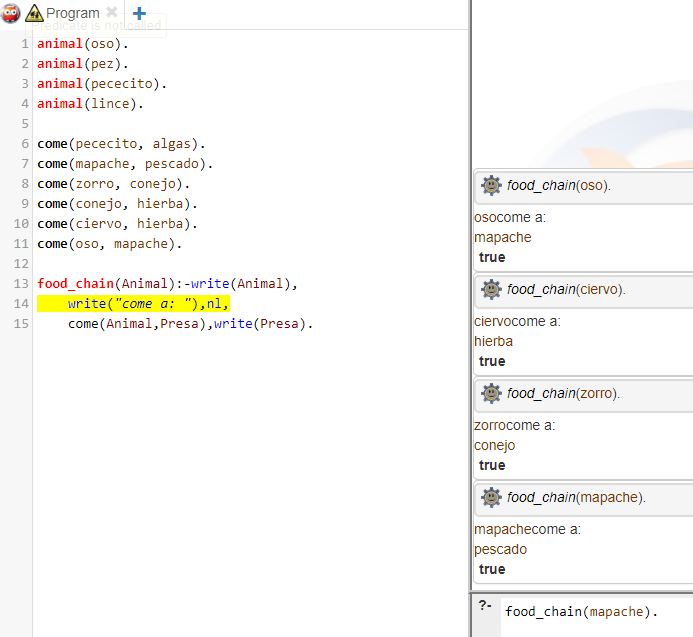


# EJERCICIO Nro 2

Describa una regla para determinar cuáles animales pertenecen a la cadena alimenticia de otro:

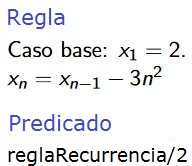
animal(oso). animal(pez). animal(pececito). animal(lince).

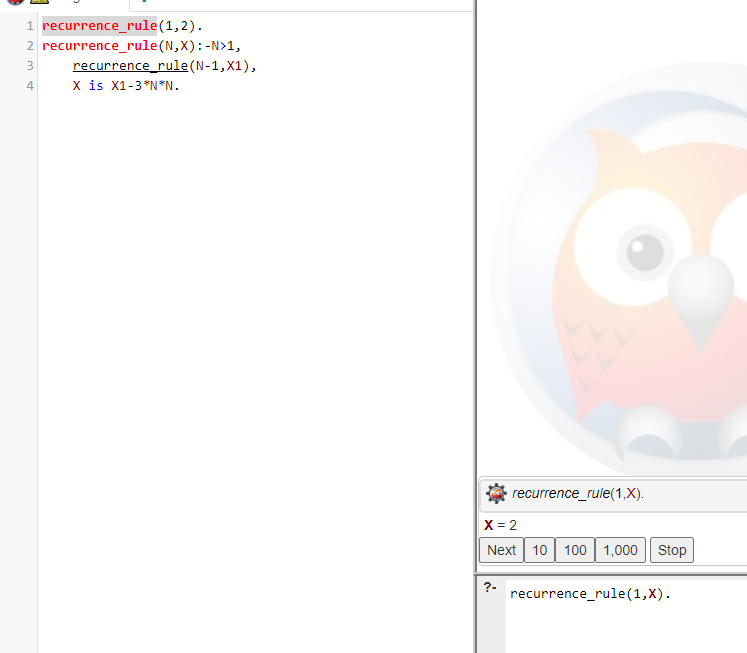
come(pececito, algas). come(mapache, pescado). come(zorro, conejo). come(conejo, hierba). come(ciervo, hierba). come(oso, mapache).



# EJERCICIO Nro 3

Cree un programa en Prolog para resolver la ecuación de recurrencia de aridad 2:





# EJERCICIO Nro 4

Considere el programa:

m(1). m(2):- !. m(3).

m1(X,Y) :- m(X),m(Y).

m2(X,Y) :- m(X),!,m(Y).

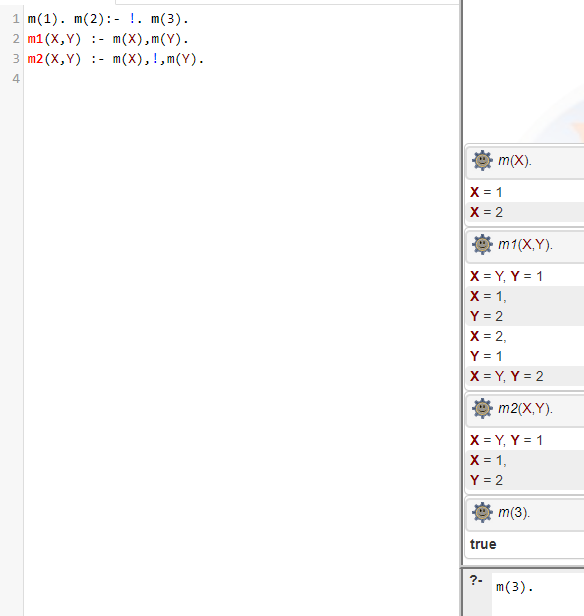
Diga cuales son todas las respuestas de Prolog a los siguientes objetivos:

1.?- m(X).

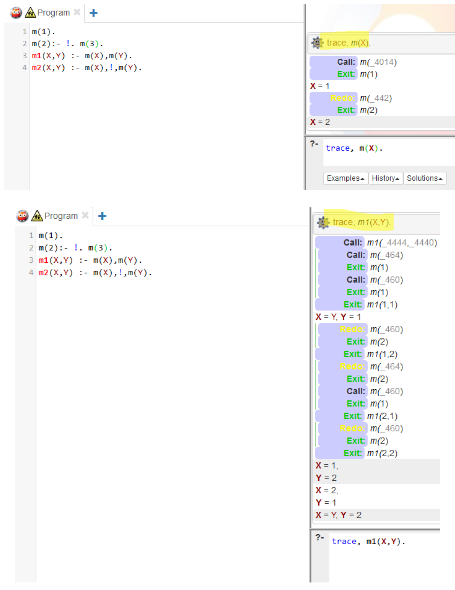
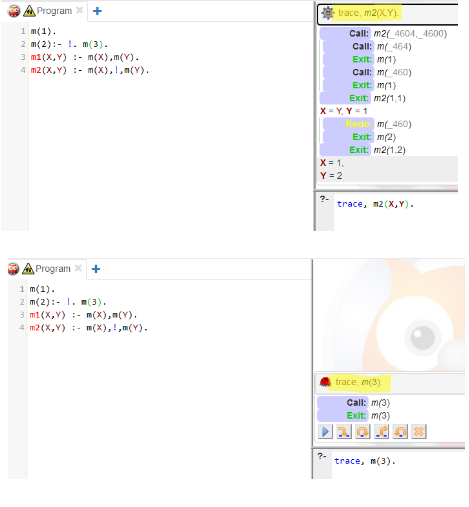
2.?- m1(X,Y).

3.?- m2(X,Y).

4.?- m(3).



Después de intentar responder a las consultas, utilice SWI-Prolog con el comando de trace activado y siga paso a paso la solución de la consulta.

# EJERCICIO Nro 5

La siguiente relación clasifica los números en 3 clases: positivos, cero y negativos. Define una relación equivalente que sea más eficiente, usando corte. clase(Número,positivo):- Numero>0.

clase(0,cero). clase(Numero,negativo):- Numero<0

# Ejercicio Nro 6

La siguiente regla calcula el índice de masa corporal (IMC), la regla recibe el peso y la altura.

imc(Peso,Altura) :- X is Peso/(Altura\*Altura),

write(‘Su IMC es: ’), write(X).

Modifique el programa para clasificar el IMC (use corte): IMC < 20, 'IMC abajo de lo normal'.

IMC >= 20 y < 30, 'IMC está normal'. IMC >= 30 y < 40, 'Obesidad grado 1'.

IMC >= 30 y < 40, 'Obesidad grado 2’.

IMC > 40, 'Obesidad grado 3.'