

Ejemplo swi-prolog

JAFH

Explicación

Para una empresa de desarrollo de software se requiere un Sistema Experto que ayude a la gestión del personal de informática.

Suponga lo siguientes hechos para el personal de la empresa

ana tiene el nivel de técnico.

juan tiene el nivel de pasante.

luis tiene nivel de titulado.

rosa tiene nivel de maestría.

ana tiene certificado en lenguaje c.

rosa esta certificada en php.

ana esta certificada en js.

juan esta certificado en c.

Hechos representado en swi-prolog:

nivel(ana,tecnico).

nivel(juan,pasante).

nivel(luis,titulado).

nivel(rosa,maestria).

certificado_en(ana,c).

certificado_en(rosa,php).

certificado_en(ana,js).

certificado_en(juan,c).

Se encontraron las siguientes reglas para asignar puestos al personal de la empresa

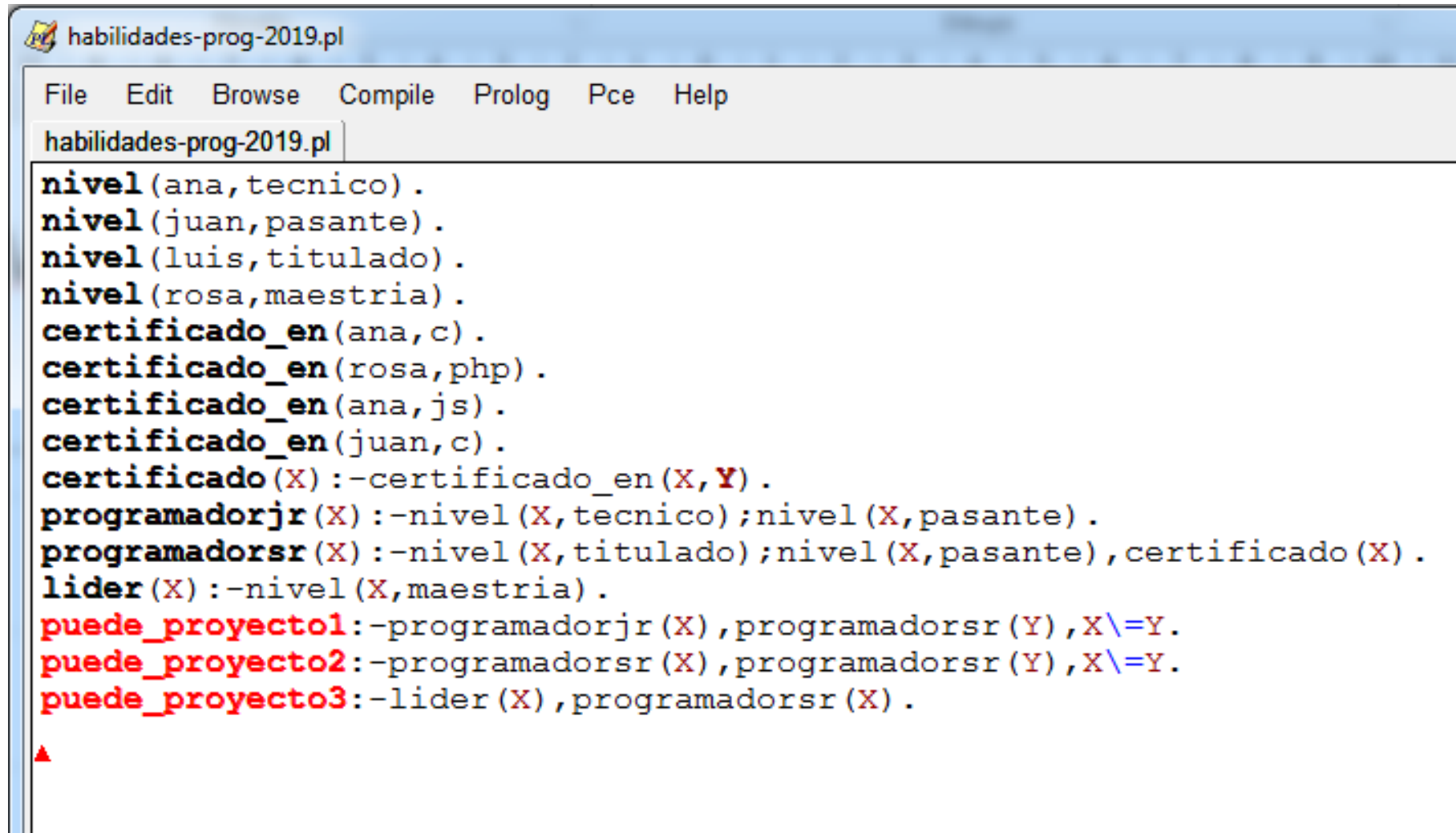
1. Decimos que un empleado esta certificado si el empleado esta certificado en cualquier lenguaje.
2. Decimos que un empleado es programador junior si el empleado tiene nivel técnico o el empleado es pasante.
3. Decimos que un empleado es programador senior si el empleado esta titulado o es un pasante y esta certificado (en lo que sea).
4. Decimos que un empleado es líder de proyecto si el empleado tiene maestría .
5. Decimos que podemos realizar el proyecto 1 si tenemos el siguiente personal: un programador junior y un programador senior.
6. Decimos que podemos realizar el proyecto 2 si tenemos el siguiente personal: dos programadores senior.
7. Decimos que podemos realizar el proyecto 3 si tenemos el siguiente personal: un líder de proyecto y un programador senior.

Reglas representadas en Swi-prolog

1. `certificado(X):-certificado_en(X,Y).`

2. `programadorjr(X):-nivel(X,tecnico);nivel(X,pasante).`
3. `programadorsr(X):-nivel(X,titulado);nivel(X,pasante),certificado(X).`
4. `lider(X):-nivel(X,maestria).`
5. `puede_proyecto1:-programadorjr(X),programadorsr(Y),X\=Y.`
6. `puede_proyecto2:-programadorsr(X),programadorsr(Y),X\=Y.`
7. `puede_proyecto3:-lider(X),programadorsr(X).`

Capturado en swiprolog: archivo habilidades-prog-2019.pl

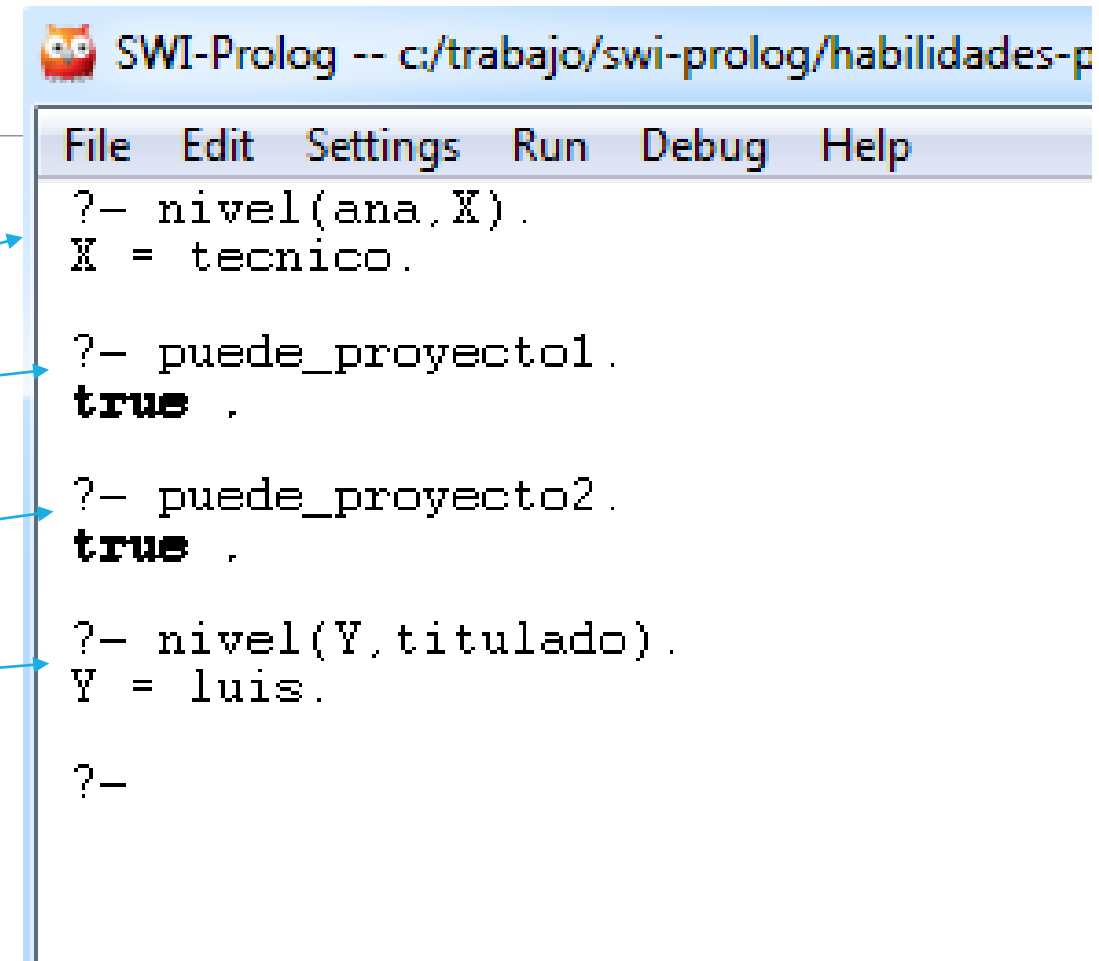


```
File Edit Browse Compile Prolog Pce Help
habilidades-prog-2019.pl
nivel(ana, tecnico) .
nivel(juan, pasante) .
nivel(luis, titulado) .
nivel(rosa, maestria) .
certificado_en(ana, c) .
certificado_en(rosa, php) .
certificado_en(ana, js) .
certificado_en(juan, c) .
certificado(X) :- certificado_en(X, Y) .
programadorjr(X) :- nivel(X, tecnico) ; nivel(X, pasante) .
programadorsr(X) :- nivel(X, titulado) ; nivel(X, pasante), certificado(X) .
lider(X) :- nivel(X, maestria) .
puede_proyecto1 :- programadorjr(X), programadorsr(Y), X \= Y.
puede_proyecto2 :- programadorsr(X), programadorsr(Y), X \= Y.
puede_proyecto3 :- lider(X), programadorsr(X) .
▲
```

Consultas

Hagamos las consultas siguientes:

1. Que nivel tiene el empleado ana?
2. Tenemos personal para realizar el proyecto 1?
3. Tenemos personal para realizar el proyecto 2?
4. Que empleados están titulados?



```
SWI-Prolog -- c:/trabajo/swi-prolog/habilidades-p
File Edit Settings Run Debug Help
?- nivel(ana,X).
X = tecnico.

?- puede_proyecto1.
true .

?- puede_proyecto2.
true .

?- nivel(Y,titulado).
Y = luis.

?-
```


Fin

“De nada”