

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE
AREQUIPA

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



LENGUAJES DE PROGRAMACIÓN
LABORATORIO

TAREA 03

Programación en lenguaje de programación Prolog

Alumno:

Lipe Huamaní, Brayan Alexander

Docente:

Cardenas Talavera, Rolando Jesús

Arequipa - Perú

28 de Octubre del 2020

Índice

Índice de figuras	1
1. Introducción	2
2. Objetivos	2
3. Materiales	2
3.1. Lenguaje de programación	2
3.2. Compilador	2
3.3. Entorno de desarrollo	2
4. Problemas	2
4.1. Problemas	2
4.1.1. Ejecución de las sentencias	4
4.2. Resultados	6
5. Conclusiones	6
5.1. Dificultades	6
Referencias	7

Índice de figuras

1. Compilación de la sentencia a.	4
2. Compilación de la sentencia b.	4
3. Compilación de la sentencia c.	5
4. Compilación de la sentencia d.	5
5. Compilación de la sentencia f.	6
6. Compilación de la sentencia g.	6

Listings

1. clase creadora de objetos - alumnos	3
2. clase creadora de objetos - alumnos	3
3. clase creadora de objetos - alumnos	3
4. clase creadora de objetos - alumnos	3
5. clase creadora de objetos - alumnos	3
6. clase creadora de objetos - alumnos	3
7. clase creadora de objetos - alumnos	3
8. clase creadora de objetos - alumnos	4

Programación de Lenguaje Ensamblador

1. Introducción

El presente documento tratará sobre como se realizan consultas sobre una base de datos ya establecida. Como ya sabemos, Prolog es un lenguaje de programación declarativo, creado por Alain Colmerauer y Robert Kowalski en 1972.

Básicamente este está basado en declaraciones, proposiciones, condiciones, afirmaciones o negaciones. El programa PROLOG especifica cómo debe ser la solución, en vez de dar el algoritmo para su resolución. La solución se obtiene mediante búsqueda aplicando la lógica de predicados.

En la sección 2 se declaran los objetivos respecto a este lenguaje determinados en este documento, luego en sección 3 se definen los materiales utilizados para desarrollar los lenguaje en dicho lenguaje. Posteriormente, en la sección 4 tenemos las preguntas que necesitan ser resueltas mediante consultas hechas en Prolog. Finalmente en la sección 5 tenemos las conclusiones acerca de este lenguaje.

2. Objetivos

- Comprender las sentencias realizadas en lenguaje en programación Prolog.
- Implementar consultas en el lenguaje mencionado.
- Interpretar y evaluar los resultados obtenidos después de la ejecución de las sentencias.

3. Materiales

3.1. Lenguaje de programación

- Prolog

3.2. Compilador

- Swi Prolog

3.3. Entorno de desarrollo

- Visual Studio Code 2020

4. Problemas

4.1. Problemas

- a. ¿En que año se realizo la película American Beauty?
- b. Encontrar las películas realizadas en el año 2000.
- c. Encontrar las películas realizadas antes del año 2000.
- d. Encontrar las películas realizadas después del año 1990.
- e. Encontrar el actor que aparece en varias películas.
- f. Encontrar al director que haya dirigido a Scarlett Johansson.
- g. Encontrar al actor que dirigió la misma película.
- h. Encontrar la película donde John Goodman y Jeff Bridges fueron co-estrellas.

Solución a Código:

```
1 get_year_pelicula_american_beauty(M) :-  
2   pelicula(M, Y),  
3   M == american_beauty,  
4   write('Year: '),  
5   write(Y).
```

Listing 1: clase creadora de objetos - alumnos

Solución b Código:

```
1 get_movies_hecha_dosmil(Y) :-  
2   write('Películas hechas en el 2000: '), nl,  
3   pelicula(M, Y),  
4   Y == 2000,  
5   write(M).
```

Listing 2: clase creadora de objetos - alumnos

Solución c Código:

```
1 get_movies_antes_dosmil(Y) :-  
2   write('Películas hechas antes del 2000: '), nl,  
3   pelicula(M, Y),  
4   Y < 2000,  
5   write(M).
```

Listing 3: clase creadora de objetos - alumnos

Solución d Código:

```
1 get_movies_despues_dosmil(Y) :-  
2   write('Películas hechas despues de 1990: '), nl,  
3   pelicula(M, Y),  
4   Y > 1990,  
5   write(M).
```

Listing 4: clase creadora de objetos - alumnos

Solución e Código:

```
1 get_actor_varias_movies(A) :-  
2   write('Actor que aparece en varias peliculas es: '),  
3   actor(M, A, _),  
4   actor(M2, B, _),  
5   M == M2.
```

Listing 5: clase creadora de objetos - alumnos

Solución f Código:

```
1 get_director_scarlett(D) :-  
2   write('Directores: '),  
3   director(M, D),  
4   actriz(Movie, scarlett_johansson, _),  
5   Movie == M.
```

Listing 6: clase creadora de objetos - alumnos

Solución g Código:

```
1 get_actor_director(A):-  
2   write('El actor que dirigio y actuo en su propia pelicula es: '),  
3   actor(M, A, _),  
4   director(Movie, D),  
5   M == Movie,  
6   A == D.
```

Listing 7: clase creadora de objetos - alumnos

Solución h Código:

```
1 get_movies_john_jeff(M) :-  
2   write('La película donde John Goodman y Jeff Bridges actuaron fue: '),  
3   actor(M, john_goodman, _),  
4   actor(Movie2, jeff_bridges, _),  
5   M = Movie2.
```

Listing 8: clase creadora de objetos - alumnos

4.1.1. Ejecución de las sentencias

```
PS C:\Users\BRAYAN LIPE\Documents\Prolog> swipl -s prolog_problema.pl  
Warning: c:/users/brayan lipe/documents/prolog/prolog_problema.pl:3032:  
Warning: Singleton variables: [8]  
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.1)  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  
Please run ?- license. for legal details.  
  
For online help and background, visit https://www.swi-prolog.org  
For built-in help, use ?- help(Topic). or ?- apropos(Word).  
  
1 ?- get_year_pelicula_american_beauty(american_beauty).  
Year: 1999  
true.
```

Figura 1: Compilación de la sentencia a.

```
2 ?- get_movies_hecha_dosmil(2000).  
Peliculas hechas en el 2000:  
down_from_the_mountain  
true ;  
o_brother_where_art_thou  
true ;  
ghost_world  
true.
```

Figura 2: Compilación de la sentencia b.

```
1 ?- get_movies_antes_dosmil(Y).
Películas hechas antes del 2000:
american_beauty
Y = 1999 ;
anna
Y = 1987 ;
barton_fink
Y = 1991 ;
the_big_lebowski
Y = 1998 ;
blade_runner
Y = 1997 ;
blood_simple
Y = 1984 ;
the_cotton_club
Y = 1984 ;
crimewave
Y = 1985 ;
fargo
Y = 1996 ;
the_firm
Y = 1993 ;
frankenweenie
Y = 1984 ;
ghost_busters
Y = 1984 ;
the_godfather
Y = 1972 ;
the_godfather_part_ii
Y = 1974 ;
the_godfather_part_iii
Y = 1990 ;
groundhog_day
Y = 1993 ;
hearts_of_darkness_a_filmaker_s_apocalypse
Y = 1991 ;
the_hudsucker_proxy
Y = 1994 ;
inside_monkey_zetterland
Y = 1992 ;
lick_the_star
```

Figura 3: Compilación de la sentencia c.

```
2 ?- get_movies_despues_1990(Y)
.
Películas hechas despues de 1990:
american_beauty
Y = 1999 ;
barton_fink
Y = 1991 ;
the_big_lebowski
Y = 1998 ;
blade_runner
Y = 1997 ;
cq
Y = 2001 ;
down_from_the_mountain
Y = 2000 ;
fargo
Y = 1996 ;
the_firm
Y = 1993 ;
girl_with_a_pearl_earring
Y = 2003 ;
groundhog_day
Y = 1993 ;
hail_caesar
Y = 2006 ;
hearts_of_darkness_a_filmaker_s_apocalypse
Y = 1991 ;
the_hudsucker_proxy
Y = 1994 ;
inside_monkey_zetterland
Y = 1992 ;
intolerable_cruelty
Y = 2003 ;
the_ladykillers
Y = 2004 ;
lick_the_star
Y = 1998 ;
lost_in_translation
Y = 2003 ;
the_man_who_wasn_t_there
```

Figura 4: Compilación de la sentencia d.

```

3 ?-
  get_director_scarlett(D) .
Directores:
D = peter_webber ;
D = sofia_coppola ;
D = ethan_coen ;
D = joel_coen ;
D = va_g_rdos ;
D = brian_de_palma ;
D = eric_schaeffer ;
D = ellory_elkayem ;
D = terry_zwigoff ;
D = mike_barker ;
D = eric_schaeffer ;
D = raja_gosnell ;
D = robert_redford ;
D = paul_weitz ;
D = arne_glimcher ;
D = michael_bay ;
D = shainee_gabel ;
D = lisa_krueger ;
D = woody_allen ;
D = erik_fleming ;
D = rob_reiner ;
D = brian_robbins ;
D = stephen_hillenburg ;
D = woody_allen ;
false.

```

Figura 5: Compilación de la sentencia f.

```

4 ?- get_actor_director(A).
El actor que dirigió y actuó en su propia película es:
A = joseph_d_kucan ;
A = ivan_reitman ;
A = harold_ramis ;
A = spike_jonze ;
A = eric_schaeffer ;
A = eric_schaeffer ;
A = robert_redford ;
A = stephen_hillenburg ;
A = woody_allen ;
false.

```

Figura 6: Compilación de la sentencia g.

4.2. Resultados

Como vemos, las consultas han sido favorables y el tiempo de respuesta ha sido rápido y sin complicaciones [SWI20].

5. Conclusiones

5.1. Dificultades

- Fue recreativo volver a tocar el tema de lógica de predicados, ya que Prolog se basa en este tópico que pertenece a la matemática discreta.
- A pesar de no haber implícitamente bucle, estos se pueden sustituir con recursividad.
- Cabe recalcar, que en caso no se efectúen correctamente las relaciones y sentencias entre las variables cuando se realicen las sentencias.

Referencias

- [SWI20] SWISH. *Swi-Prolog Tutorial*. 2020. URL: <https://swish.swi-prolog.org/p/Tutorial%5C%20de%5C%20prolog.swinb>.