

En este artículo presentamos un sencillo programa elaborado en haskell que realiza el conteo de digitos, la suma y el producto.

Codigo fuente

```
digitos ::Int->Int
digitos 0 = 0
digitos n =1+digitos (div n 10)
suma::Int->Int
suma 0=0
suma n=(mod n 10)+suma(div n 10)
prod::Int->Int
prod 0=1
prod n=(mod n 10)*prod(div n 10)
resultado::Int->String
resultado 0=""
resultado n="numero de digitos:"++show(cont n)++"    "++"la suma es:"+
+show(suma n)++"    "++"el producto es:"++show(prod n)
```

3.1.1 Uso en una terminal

Una vez ejecutado el método de instalación correspondiente, hay varias formas de acceder a Prolog. Desde una terminal del sistema operativo, SWI-Prolog puede ejecutarse mediante el comando `swipl`:

```
1 > swipl
2 Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 7.4.2)
3 Copyright (c) 1990-2015 University of Amsterdam, VU Amsterdam
4 SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
5 and you are welcome to redistribute it under certain conditions.
6 Please visit http://www.swi-prolog.org for details.
7
```

-
- 1 https://brew.sh/index_es.html
 - 2 <https://www.xquartz.org>

```
8 | For help, use ?- help(Topic). or ?- apropos(Word).
9 |
10 | ?-
```

Prolog se ejecuta en un ciclo REPL (Read-Eval-Print Loop), lo que significa que una vez ejecutado, el sistema está a la espera de una instrucción para evaluarla e imprimir el resultado. Eso se indica con el prompt `?-`. Asumiendo que existe el siguiente programa bajo el nombre `abuelo.pl`:

```
1 | %%%padre(X,Y): X es padre de Y
2 |
3 | padre(luis,ana).
4 | padre(juan,luis).
5 |
6 | %%%abuelo(X,Y): X es abuelo de Y
7 |
8 | abuelo(X,Y) :-
9 |     padre(X,Z), padre(Z,Y).
```

el programa puede cargarse en Prolog de la siguiente manera:

```
1 | ?- [abuelo].
2 | true.
```

Una vez cargado un programa, se le pueden plantear metas al mismo. Por ejemplo:

```
1 | ?- padre(luis,ana).
2 | true.
3 |
4 | ?- padre(luis,X).
5 | X = ana.
6 |
7 | ?- abuelo(X,ana).
8 | X = juan.
```

Si el programa se encuentra en otro directorio, puede ser cargado mediante una cadena de texto que indique el camino completo hasta el programa:

```
1 | ?- ["Dropbox/code/prolog/ia2/cap03/abuelo.pl"].
2 | true.
3 |
4 | ?- abuelo(X,ana).
5 | X = juan.
```

Para salir de Prolog, use el comando `halt`:

```
1 | ?- halt.
2 | >
```