



# INSTITUTO TECNOLÓGICO SUPERIOR DE CHICONTEPEC

## INGENIERÍA EN SISTEMAS COMPUTACIONALES

**Producto:** Unidad 3, Descarga e Instalación de  
SWI-Prolog.

**Asignatura:** Programación Lógica y Funcional.

**Docente:** Ing. Efrén Flores Cruz.

**Estudiante:** Manuel Zúñiga Hernández.

**Semestre:** Octavo

Chicontepec, Veracruz, a 24 de Abril 2020.

## Contenido

Introducción.....	3
Desarrollo.....	4
Descarga de SWI-Prolog.....	4
Instalación de SWI-Prolog 8.1.29. ....	7
Anexos. ....	11
Conclusión. ....	17

## Introducción.

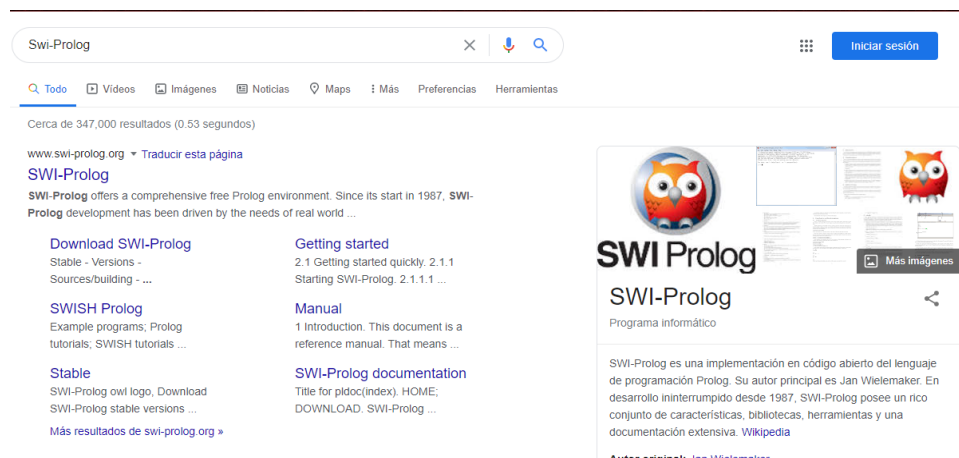
En el siguiente reporte de investigación se describe el procedimiento de descarga e instalación del programa SWI-Prolog, este trabajo pertenece a la asignatura de Programación Lógica.

SWI-Prolog es un programa que es considerado como un lenguaje de programación. El SWI se usa para hacer consultas a partir de un programa.

## Desarrollo.

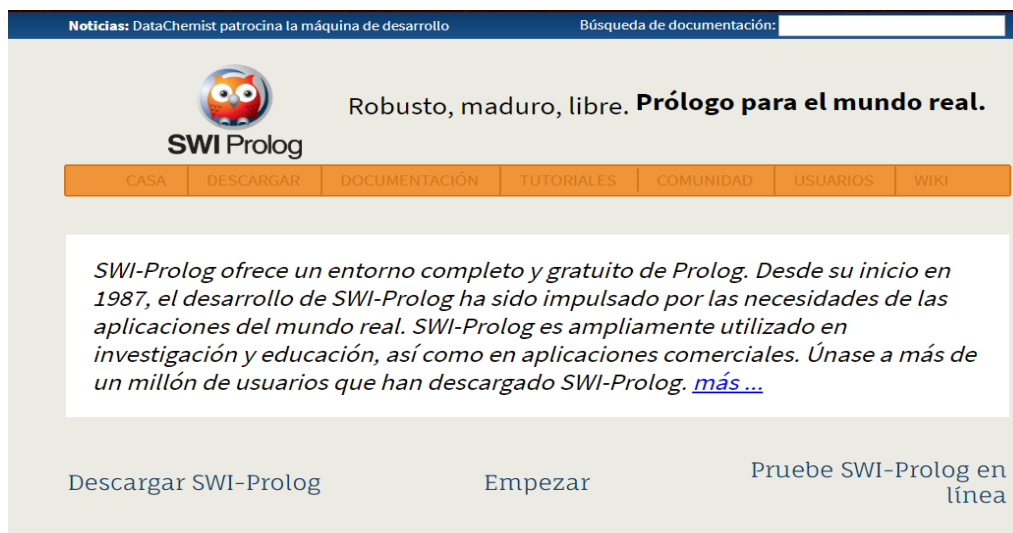
### Descarga de SWI-Prolog.

1. Vamos a empezar la descarga de SWI-Prolog, debemos de ingresar a la página oficial de dicho programa tal y como se muestra en la siguiente imagen.




#### 1. Propuesta propia.

2. Una vez que nos encontremos dentro de la página oficial, deberemos de ingresar a la opción de descargar SWI-Prolog que se encuentra a la izquierda en la parte inferior.



#### 2Propuesta propia.

3. Cuando estemos en la opción de descarga, tendremos que dar clic en la opción que se encuentra subrayado y de color azul que dice versiones.



## Descargas de SWI-Prolog

[CASA](#) | [DESCARGAR](#) | [DOCUMENTACIÓN](#) | [TUTORIALES](#) | [COMUNIDAD](#) | [USUARIOS](#) | [WIKI](#)

### Versiones disponibles

La versión **estable** se actualiza con poca frecuencia. Está bien para ejecutar el código básico de Prolog sin sorpresas. La versión de **desarrollo** se lanza aproximadamente cada dos o cuatro semanas. Esta es la versión recomendada para desarrolladores y usuarios de aplicaciones como [SWISH](#) o [ClioPatria](#). Finalmente, el **GIT** y las versiones **diarias** son para desarrolladores que desean contribuir o tienen acceso inmediato a parches. Estas versiones son generalmente buenas, pero ocasionalmente sufren de regresión.

- [Lanzamiento estable](#)
- [Lanzamiento de desarrollo](#)
- [Compilaciones diarias para Windows](#)
- Examinar el [repositorio](#) GIT

### Leer más sobre

- [Versiones](#) disponibles de SWI-Prolog
- Información sobre [paquetes de Linux y construcción en Linux](#)

### 3. Propuesta propia.

4. En este apartado nos aparecerán las diferentes versiones de SWI-Prolog, existen para diferentes sistemas operativos, pero en este caso vamos a elegir para Windows.

Binarios		
	12,210,795 bytes	<a href="#">SWI-Prolog 8.0.3-1 para Microsoft Windows (64 bits)</a> Ejecutable autoinstalable para las ediciones de Windows de 64 bits de Microsoft. Requiere al menos Windows 7. Consulte el <a href="#">manual de referencia</a> para decidir si usar la versión de 32 o 64 bits. Este binario está vinculado a GMP 6.1.1 que está cubierto por la licencia LGPL. <b>SHA256</b> : f3b4783dd05a746219600185e8404238ee451708f3512c601a2c99cf549b0fa5
	11,907,718 bytes	<a href="#">SWI-Prolog 8.0.3-1 para Microsoft Windows (32 bits)</a> Ejecutable autoinstalable para MS-Windows. Requiere al menos Windows 7. Instala <b>swipl-win.exe</b> y <b>swipl.exe</b> . Este binario está vinculado a GMP 6.1.1 que está cubierto por la licencia LGPL. <b>SHA256</b> : 7b6b7295a8a19d350f16b38b00201725091e5d8272d4694c8df8696eabe45767
	27,270,035 bytes	<a href="#">SWI-Prolog 8.0.3-1 para MacOSX 10.12 (Sierra) y versiones posteriores de Intel</a> Instalador con binarios creados con <a href="#">Macports</a> . Se instala <code>/opt/local/bin/swipl</code> . Necesita <a href="#">xquartz</a> (X11) y las herramientas de desarrollo (Xcode) instaladas para ejecutar las <a href="#">herramientas de desarrollo</a> <b>SHA256</b> : 9d65988b5ca92c8abae4bdeca3aba41901287e7de44574cabe43dff4a17eb5cf
Fuentes		
	10,418,553 bytes	<a href="#">Fuente SWI-Prolog para 8.0.3</a> Fuentes en <code>.tar.gz</code> formato, incluidos paquetes y archivos de documentación generados. Ver <a href="#">instrucciones de construcción</a> . <b>SHA256</b> : cee59c0a477c8166d722703f6e52f962028f3ac43a5f41240ecb45dbdbe2d6ae
Documentación		
	2,505,443 bytes	<a href="#">Manual de referencia de SWI-Prolog 8.0.3 en PDF</a> Manual de referencia de SWI-Prolog como archivo PDF. Esto <i>no</i> incluye la documentación del <a href="#">paquete</a> . <a href="#">Mostrar todos los archivos</a>

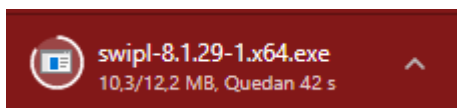
### 4. Propuesta propia.

5. En el siguiente apartado tendremos que habilitar la opción de entiendo, para después dar clic en descargar swipl-8.1.29-1.



5. Propuesta propia.

6. En la siguiente imagen podemos observar el proceso de descarga del programa.



6. Propuesta propia.

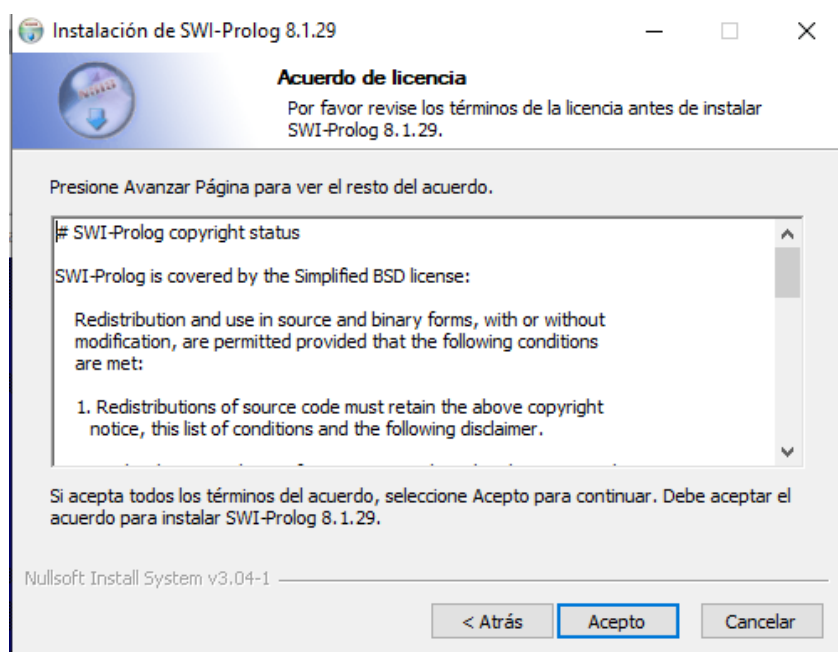
## Instalación de SWI-Prolog 8.1.29.

1. Ahora vamos a explicar el proceso de instalación, lo primero que nos aparecerá es la siguiente ventana en la cual solo tendremos que dar clic en siguiente.



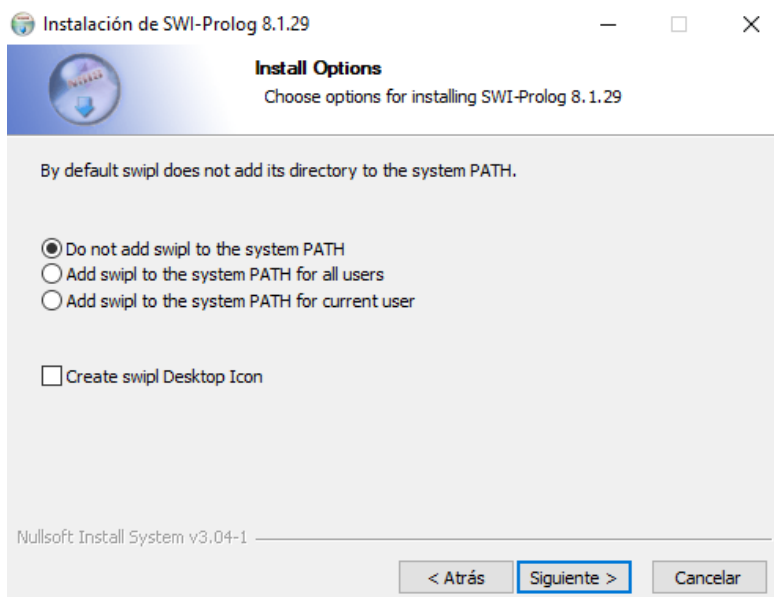
### 7. Propuesta propia.

2. Después tendremos que aceptar el acuerdo de la licencia del programa.



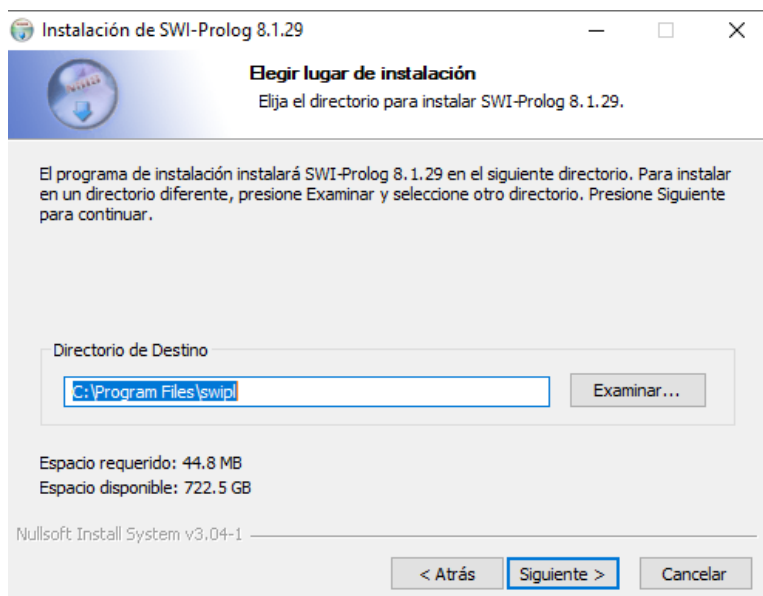
### 8. Propuesta propia.

3. En este apartado no seleccionaremos nada, solo daremos clic en siguiente para seguir con la instalación.



#### 9. Propuesta propia.

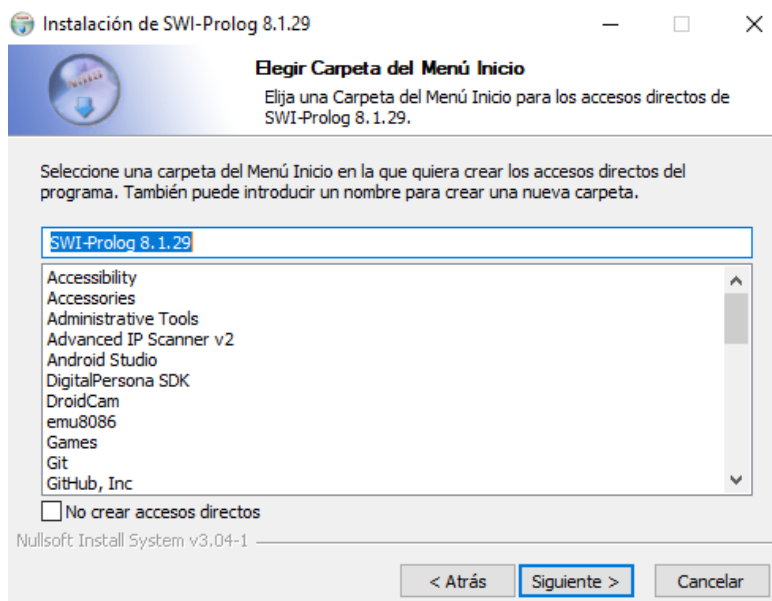
4. En la siguiente ventana ya viene la ruta en la cual se guardaran nuestros trabajos que vallamos realizando. la ruta la podemos cambiar si gustamos.



#### 10. Propuesta propia.

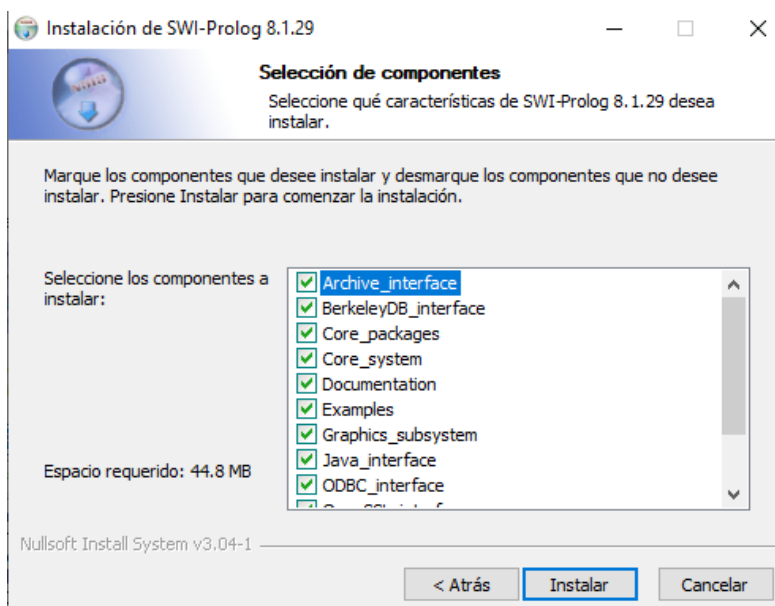


5. Vamos a dejar esta parte tal y como esta, y solo daremos clic en siguiente.



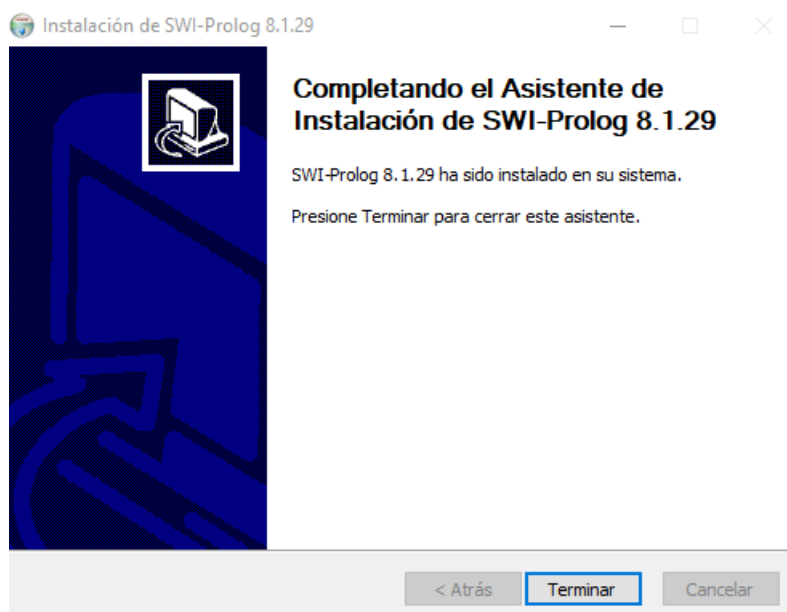
#### 11. Propuesta propia.

6. Para que empiece la instalación daremos clic en instalar, empezara el proceso de instalación una vez que termine de cargar mostrara otra ventana.



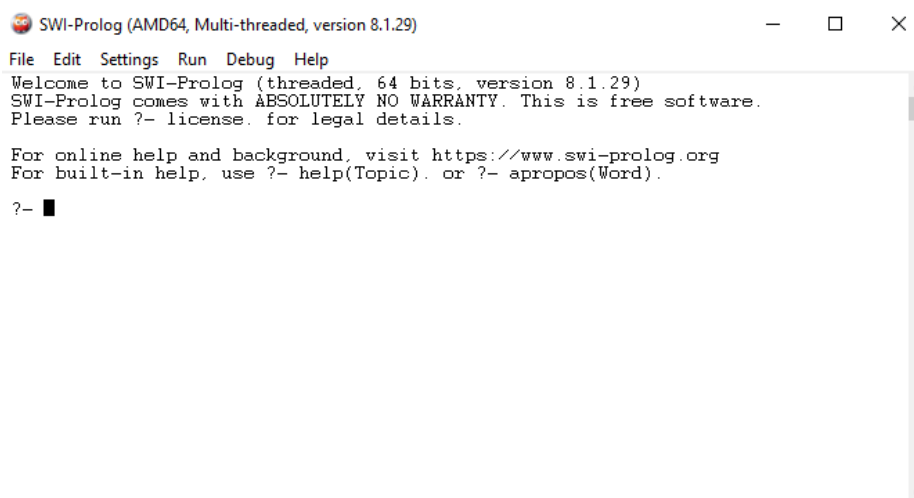
#### 12. Propuesta propia.

7. La siguiente ventana nos indica que el proceso de instalación a finalizado con éxito.



### 13. Propuesta propia.

8. la siguiente ventana muestra el area de trabajo de SWI-Prolog, en la cual estaremos trabajando más adelante.



### 14. Propuesta propia.

## Anexos.

El lenguaje de programación prolog, fue creado por Alain Colmerauer y sus colaboradores alrededor de 1970.

PROLOG usa lógica de predicados de primer orden (restringida a cláusulas de Horn) para representar datos y conocimiento, utiliza encadenamiento hacia atrás y una estrategia de control retroactiva sin información heurística (backtracking).

Uno de las ventajas de la programación lógica es que se especifica qué se tiene que hacer (programación declarativa), y no cómo se debe hacer (programación imperativa). Prolog ofrece un sistema de programación práctico que tiene algunas de las ventajas de claridad y declaratividad que ofrecería un lenguaje de programación lógica y, al mismo tiempo, nos permite un cierto control y operatividad.

### Elementos del lenguaje.

- Hechos (átomos)
- Reglas (cláusulas de Horn)
- Preguntas u. objetivos (conjunciones o disyunciones de átomos).

### Predicados

Se utilizan para expresar propiedades de los objetos, predicados monádicos, y relaciones entre ellos, predicados poliádicos. En Prolog los llamaremos hechos. Los hechos son las sentencias más sencillas.

Debemos tener en cuenta que:

- \* Los nombres de todos los objetos y relaciones deben comenzar con una letra minúscula.





**Números:** Se utilizan para representar números de forma que se puedan realizar operaciones aritméticas. Dependen del orden y la implementación.

- **Enteros:** en la implementación de prolog-2 puede utilizarse

cualquier entero que el intervalo  $[-223, 223-1] = [-8,388,608]$

- **Reales:** decimales con coma flotante, consistentes en al menos un dígito, opcionalmente un punto decimal y más dígitos, opcionalmente E, un «+» o «-» y más dígitos.

Las variables se utilizan para representar objetos cualesquiera del universo u objetos desconocidos en ese momento, es decir, son las incógnitas del problema. Se diferencian de los átomos en que empiezan siempre con una letra mayúscula o con el signo de subrayado (\_).

### Conectivos Lógicos.

Para trabajar con sentencias más complejas, fórmulas moleculares, que constarán de fórmulas atómicas combinadas mediante conectivos.

La **conjunción**, "y", la representamos poniendo una coma entre los objetivos "," y consiste en objetivos separados que Prolog debe satisfacer, uno después del otro.

Por ejemplo  $X, Y$ .

La **disyunción**, "o", tendrá éxito si se cumple alguno de los objetivos que la componen se utiliza un punto y coma ";" Colocado entre los objetivos.

Por ejemplo  $X; Y$



La **negación** lógica no puede ser representada explícitamente en Prolog, sino que se representa implícitamente por la falta de aserción: "no", tendrá éxito si el objetivo  $X$  fracasa. No es una verdadera negación, en el sentido de la lógica, sino una negación "por fallo". La representamos con el predicado predefinido `not` o con `!`.

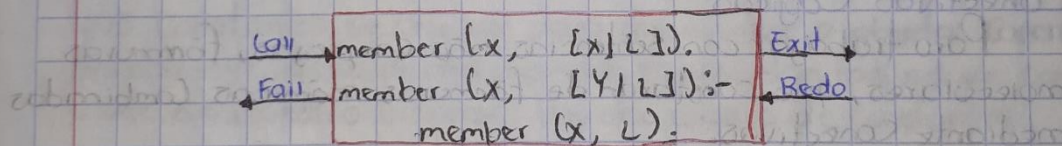
Por ejemplo, `not(X) !t X`.

La **implicación o condicional**, sirve para significar que un hecho depende de un grupo de otros hechos. En Prolog se usa el símbolo ":-" para representar lo que llamamos una regla.

Por ejemplo: cabeza-de-la-regia :- cuerpo-de-la-regia.

El depurador de Prolog.

En Prolog, los predicados se consideran "cajas negras" que tienen cuatro puertos, dos de entrada y dos de salida.



**Puertos:**

**Call** (llamada): Comienza la ejecución del objetivo.

**Exit** (salida): Salida con éxito del objetivo.

**Redo** (reintentar): Reintentar el predicado utilizando otra alternativa.

**Fail** (fallo): Salida con fallo del objetivo. No se encontraron soluciones.

**Estructura de un Prolog.**

El hecho de programar en Prolog consiste en dar al ordenador un universo finito en forma de hechos y reglas, proporcionando los medios para realizar inferencias de un hecho a otro.

La programación de Prolog consiste en:

- \* Declarar algunos **HECHOS** sobre los objetos y sus relaciones.
- \* Definir algunas **REGLAS** sobre los objetos y sus relaciones.
- \* Hacer **PREGUNTAS** sobre los objetos y sus relaciones.

**Programa Prolog:** Conjunto de afirmaciones (hechos y reglas) representando los conocimientos que poseemos en un determinado dominio o campo de nuestra competencia.

### Preguntas

Son las herramientas que tenemos para recuperar la información desde Prolog. Al hacer una pregunta a un programa lógico queremos determinar si esa pregunta es consecuencia lógica del programa. Prolog considera que todo lo que hay en la Base de datos es verdad, y lo que no, es falso.

Cuando se hace una pregunta en Prolog, éste efectuará una búsqueda por toda la Base de Datos intentando encontrar hechos que coincidan con la pregunta.

### Sintaxis.

**Comentarios.** Para escribir un comentario lo encerramos entre los signos /\* y \*/ o desde el símbolo % hasta el final de línea.

/\* ... Comentario ... \*/

% Comentario de una sola línea.

### Caracteres.

Los nombres de Constantes y Variables se construyen a partir de cadenas de caracteres. Prolog reconoce dos tipos de caracteres.



\* **Imprimibles**: hace que aparezca un determinado signo en la pantalla del ordenador.

Se dividen en cuatro categorías:

- Letras mayúsculas: A, B, C, ..., Z.
- Letras minúsculas: a, b, c, ..., z.
- Dígitos numéricos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Signos: ' ! " # \$ % & ' ( ) = - ^ \ | ~ { } [ ] \_ @ + \* ; : < > , . ?

\* **No imprimibles**: no aparecen en forma de signo en la pantalla, pero realizan una determinada acción: nueva línea, retorno de carro, ... Cada carácter tiene un entero entre 0 y 127 asociado a él, este es su código ASCII.

### Estructuras.

Es un único objeto que se compone de una colección de otros objetos, llamados Componentes, lo que facilita su tratamiento. Una estructura se escribe en Prolog especificando su nombre, y sus Componentes (argumentos).

nombre (comp1, comp2, ..., compn).

### Operadores.

En prolog están predefinidos los operadores aritméticos y relacionales típicos, con la precedencia habitual entre ellos.

\* **Posición**: **Prefijo**: el operador va delante de sus argumentos.

**Infijo**: el operador se escribe entre los argumentos.

**Postfijo**: el operador se escribe detrás de sus argumentos.

\* **Precedencia**: Nos indica el orden en que se realizan las operaciones.

\* **Asociatividad**: Sirve para quitar la ambigüedad en las expresiones en los que hay dos operadores, uno a cada lado del argumento.





## Conclusión.

En este reporte se explicó dos procesos que se deben de realizar si queremos trabajar con el programa de SWI-Prolog.

El primer procedimiento que debemos de realizar es visitar la página oficial del programa, para después realizar el proceso de descarga, el segundo proceso es realizar la instalación del programa, para esto fue que se realizó la creación de este pequeño manual de descarga e instalación de SWI-Prolog.