

INSTITUTO TECNOLÓGICO SUPERIOR DE CHICONTEPEC.

INGENIERIA EN SISTEMAS COMPUTACIONALES.

ASIGNATURA:

Programación Lógica.

ALUMNO:

Flor Hernández Cruz.

DOCENTE:

Ingeniero Efrén Flores Cruz.

UNIDAD 3:

Programación Lógica.

TRABAJO:

Resumen y manual de instalación de Swi-Prolo.

SEMESTRE:

8º

FECHA DE ENTREGA:

24 de abril 2020.

Contenido

INTRODUCCION	3
DESARROLLO.....	4
RESUMEN DE PROLOG	4
DESCARGAR E INSTALAR SWI-PROLOG.....	11
SWI-PROLOG	11
FUNCIONAMIENTO	12
DESCARGA DE SWI-PROLOG	14
INSTALACCION DE SWI-PROLOG.....	16
CONCLUSION.....	21

INTRODUCCION

En el siguiente trabajo se muestra el desarrollo de la unidad 3: Programación lógica, de la asignatura de Programación Lógica y funcional, se desarrollan los siguientes temas en el temario:

- 3.1 Repaso de la lógica de primer orden.
- 3.2 Unificación y resolución.
- 3.3 Cláusulas de Horn, resolución SLD.
- 3.4 Programación lógica con cláusulas Horn.

En el desarrollo de este trabajo se muestra un resumen del lenguaje de programación de PROLOG, así como también muestra la descarga e instalación de Swi-Prolog.

Este software funciona mediante la intervención e interacción del software y el programador, el programador le realiza preguntas al programa, este responde si (verdad), no (falso), obviamente deben ser preguntas adecuadas de lógica, no se debe preguntar cosas incoherentes, tiene reglas especiales para su funcionamiento, por ejemplo, cada que termina la instrucción se debe agregar el punto. de lo contrario no se ejecuta la instrucción.

Finalmente se encuentra la conclusión de este documento, en donde se describe lo que entendí al realizar este trabajo.

DESARROLLO

RESUMEN DE PROLOG

Programación lógica y funcional 22-Abril-2020

PROLOG

Introducción

PROLOG es un lenguaje de programación (Programming en LOGique fue creado por Alain Colmerauer y sus colaboradores alrededor de 1970 en la universidad de Marseille-Aix1. PROLOG usa lógica de predicados de Primer Orden para representar datos y conocimiento, utiliza encadenamiento hacia atrás y una estrategia de control retroactiva sin información heurística (backtracking)

Elementos del lenguaje

- Hechos (átomos)
- Reglas (Cláusulas de Horn)
- Preguntas u objetos (conjunctiones ó disyunciones de átomos)

Predicados

Expresan Propiedades de los objetos, predicados monádicos y relaciones entre ellos, predicados poliádicos. En Prolog las llamamos hechos. Estos son las sentencias más sencillas. Un hecho es una fórmula atómica o átomo $P(t_1, \dots, t_n)$ e indica que se verifica la relación.

Se debe tomar en cuenta:

- Los nombres de todas las objetos y relaciones deben comenzar con una letra minúscula.
- Primero se escribe la relación o propiedad o predicado.
- Y los objetos se escriben separándolos mediante comas y encerrados entre parentesis: argumentos.
- Al final del hecho debe ir un punto (".").

Ejemplo: símbolo - de - predicado (arg1, arg2... argn).

Unificación

Es el algoritmo que se encarga de resolver las igualdades lógicas. Aunque la unificación en Prolog sustituye a la asignación de los lenguajes procedurales, no hay que confundirlo con esta, son operaciones totalmente distintas. La operación de unificación se representa mediante el operador $=$.

Terminos

Pueden ser constantes o variables, y suponemos definido un dominio no vacío en el cual toman valores (universo del discurso).

Las constantes se utilizan para dar nombre a objetos concretos del dominio, dicha de otra manera, representan individuos conocidos de nuestro universo. Además, las constantes atómicas de Prolog se utilizan para representar propiedades y relaciones entre los objetos del dominio. Hay 2 clases de constantes:

Átomos: Existen 3 clases de constantes atómicas:

- Cadenas de letras, dígitos y subrayados ($_$) empezando por letra minúscula.
- Cualquier cadena de caracteres encerrada entre comillas simples ($'$).
- Combinaciones especiales de signos $"?", "-", ":", ":", "$

Números: Representan números de forma que se puedan realizar operaciones aritméticas. Depende

del ordenador y la implementación.

- **Enteros:** Implementación del Prolog-2 puede utilizarse cualquier entero que el intervalo $[-223, 223-1] = [-8,388,608, 8,388,607]$.

- **Reales:** decimales en coma flotante, consisten al menos un dígito, opcionalmente un punto decimal y más dígitos, opcionalmente E, un $\langle\langle + \rangle\rangle$ o $\langle\langle - \rangle\rangle$ y más dígitos.

Las variables representan objetos del universo u objetos desconocidos en ese momento, son las incógnitas del problema. Empezan siempre con letra mayúscula o con el signo de subrayado (_).

Colectivas lógicas

Se utilizan en la lógica de Primer Orden son: conjunción, disyunción, negación e implicación.

La conjunción, "y", la representamos poniendo una coma entre las objetivos " , " y consiste en objetivos separados que Prolog debe satisfacer, uno después otro. Ejemplo X, Y .

La disyunción, "o" tendrá éxito si se cumple alguno de los objetivos que la componen. Se utiliza un $;$ colocado entre los objetivos. Ejemplo $X; Y$.

La negación lógica puede ser representada explícitamente en Prolog, sino que se representa implícitamente por la falta de aserción: "no", tendrá éxito si el objetivo X falla. 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 !

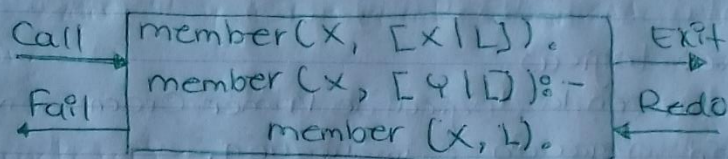
Ejemplo: $\text{not}(X) \text{ ! } X$.

La implicación o condicional funciona cuando un hecho depende de un grupo de otras hechas. Ejemplo es español decimos "Si... entonces..." En Prolog se usa el símbolo ":-" para representar a lo que llamamos regla. Ejemplo.

Cabeza-de-la-regla :- cuerpo-de-la-regla.

Depurador de Prolog

En Prolog, los predicados se consideran "cajas negras" que tienen 4 puertos, 2 de entrada y 2 de salida.



Puertas:

- * Call (llamada): Comienza la ejecución del objetivo.
- * Exit (salida): Salida con éxito de objetivo.
- * Redo (reintentar): Reintentar el predicado usando otra alternativa.
- * Fail (fallo): Salida con fallo del objetivo. No se encontraron soluciones.

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. Programación en Prolog consiste en:

- * Declarar algunas 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.

Ejecución del programa: Demostración de un Teorema en es Universo. Demostración de que una conclusión se deduce de las premisas (Afirmaciones previas).

Preguntas

Son herramientas para recuperar información desde Prolog. Al hacer una pregunta a un programa lógico queremos determinar si esa pregunta es consecuencia lógica del programa. Prolog responde "yes" a lo que puede demostrar y "no" lo que no se ha demostrado. Cuando se hace una pregunta esta se realiza en búsqueda toda la BD.

Sintaxis

Comentarios

Se usa los signos /* y */ para comentar o desde el simbolo % hasta el final de línea

Caracteres

Los nombres de constantes y variables se construyen a partir de cadenas de caracteres. Prolog reconoce 2 tipos de caracteres:

② **Imprimibles** = Hacen que aparezca un determinado signo de pantalla del ordenador. Se dividen en 4 categorías:

- * letras Mayúsculas.
- * letras minúsculas.
- * dígitos numéricos
- * Signos.

③ **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 uno 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 nombre, componentes (argumentos). Las componentes están encerradas entre parentesis (), separados por comas; el nombre se escribe antes del parentesis.

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

Operadores

En Prolog están predefinidos los operadores aritméticos y relacionales típicos, con la precedencia

habitual entre ellos:

\wedge
mod
* /
+ -
 $= \neq < > \leq \geq$

Para poder leer expresiones que contengan operadores se necesitan conocer los atributos:

- * **Posición:** Prefijo: el operador va delante de sus argumentas.
Infixo: El operador se escribe entre los argumentos.
Postfijo: El operador se escribe detrás de sus argumentos.
- * **Precedencia:** Indica el orden en que se realizan las operaciones. El operador más prioritario tendrá una precedencia 1 y el menos, 1201 (depende de la implementación).
- * **Asociatividad:** Sirve para quitar ambigüedad en las expresiones en las que hay 2 operadores uno a cada lado del argumento, que tienen la misma precedencia.

DESCARGAR E INSTALAR SWI-PROLOG.

SWI-PROLOG

Es una implementación versátil del lenguaje Prolog . Aunque SWI-Prolog ganó su popularidad principalmente en educación, su desarrollo se debe principalmente a las necesidades de desarrollo de aplicaciones. Esto se ve facilitado por una interfaz rica para otros componentes de TI al admitir muchos tipos de documentos y protocolos (de red), así como una interfaz integral de bajo nivel para C que es la base para interfaces de alto nivel para C ++, Java (incluido), C #, Python, etc. (disponible externamente). Las extensiones de tipo de datos, como los dictados y las cadenas , así como la compatibilidad total con Unicode y los enteros sin límites, simplifican el intercambio sin problemas de datos con otros componentes.

SWI-Prolog apunta a la escalabilidad. Su sólido soporte para subprocesos múltiples explota el hardware multinúcleo de manera eficiente y simplifica la integración en aplicaciones concurrentes. Su Just In Time Indexing (JITI) proporciona soporte transparente y eficiente para predicados con millones de cláusulas.

SWI-Prolog unifica muchas extensiones del lenguaje central que se han desarrollado en la comunidad de Prolog, como tabulación, restricciones, variables globales, asignación destructiva, continuaciones delimitadas e interactuadores.

SWI-Prolog ofrece una variedad de herramientas de desarrollo, la mayoría de las cuales se pueden combinar a voluntad. El sistema nativo proporciona un editor escrito en Prolog que es un clon cercano de Emacs. Proporciona resaltado semántico basado en el análisis en tiempo real del código por el propio sistema Prolog. Las herramientas complementarias incluyen un depurador gráfico, generador de perfiles y referencias cruzadas. Alternativamente, hay un modo para GNU-Emacs y el complemento Eclipse llamado PDT y un complemento VSC , cada uno de los cuales puede combinarse con las herramientas gráficas nativas. Finalmente, SWISH proporciona un cuaderno

computacional y un IDE basado en la web . SWISH es una herramienta versátil que se puede configurar y ampliar para adaptarse a muchos escenarios diferentes.

SWI-Prolog proporciona un mecanismo adicional de distribución e instalación llamado paquetes. Un paquete es un directorio con convenciones organizativas mínimas y un archivo de control que describe el origen, la versión, las dependencias y el soporte de actualización automática. Los paquetes se pueden instalar desde un archivo, repositorio GIT o URL usando `pack_install / 1` . Los paquetes se usan para compartir código en la comunidad. El sistema de paquetes ha desarrollado un par de ecosistemas para tratar tipos, coroutining, etc.

Prolog es un lenguaje adecuado para lenguajes específicos de dominio (DSL). SWI-Prolog ha mejorado este soporte mediante extensiones sintácticas al lenguaje Prolog. Ejemplos de sintaxis válida que pueden procesarse sin ambigüedades son `[1][2]`, `point {x:1, y:1}` y `function ()`. Además, las cuasi citas permiten una integración segura y limpia con la sintaxis de lenguajes arbitrarios. Por ejemplo, la siguiente instrucción especifica un fragmento de JavaScript con un seguro de interpolación del valor Prolog `X`. `{| JavaScript (X) || var x = X; |}`

FUNCIONAMIENTO

Prolog es un lenguaje conversacional; es decir, el sistema Prolog mantiene un diálogo continuo con el programador desde el inicio de la sesión hasta el final de la misma. Este diálogo toma generalmente la forma de un interrogatorio, a lo largo del cual el programador planteará preguntas al sistema Prolog. Por su parte, el sistema Prolog responderá cada una de las preguntas formuladas por el programador en la medida en que esto sea posible.

Prolog le indica al programador que está esperando a que éste le formule una pregunta mostrando en pantalla el siguiente símbolo

?-

Tras este símbolo, el programador puede teclear una pregunta (terminada en un punto) y pulsar el retorno de carro. Con ello, el programador solicita al sistema Prolog que responda a la pregunta recién formulada. Una vez procesada la pregunta el sistema Prolog mostrará en pantalla la respuesta correspondiente. Por ejemplo, si queremos preguntar a Prolog si 5 es igual a $2+3$ podemos teclear la pregunta

?- 5 is 2+3.

Yes

Después de pulsar el retorno de carro, Prolog comprobará que efectivamente 2 y 3 suman 5 y, por lo tanto, responderá afirmativamente (Yes). Prolog puede dar también respuestas negativas a las preguntas

?-

1 is 1+1.

No

Es importante recordar que todas las preguntas formuladas a Prolog deben terminar en un punto. Si se olvida incluir el punto, por más veces que se presione retorno de carro, Prolog considerará que la pregunta no está formulada en su totalidad y, por lo tanto, seguirá esperando a que se termine de formular la pregunta. Por ejemplo, si olvidamos teclear el punto en la pregunta

?- 5 is 2+3

|

Prolog mostrará el símbolo |, indicando que está esperando a que se termine de formular la pregunta, para lo que basta teclear un punto seguido de un retorno de carro

?- 5 is 2+3 |.

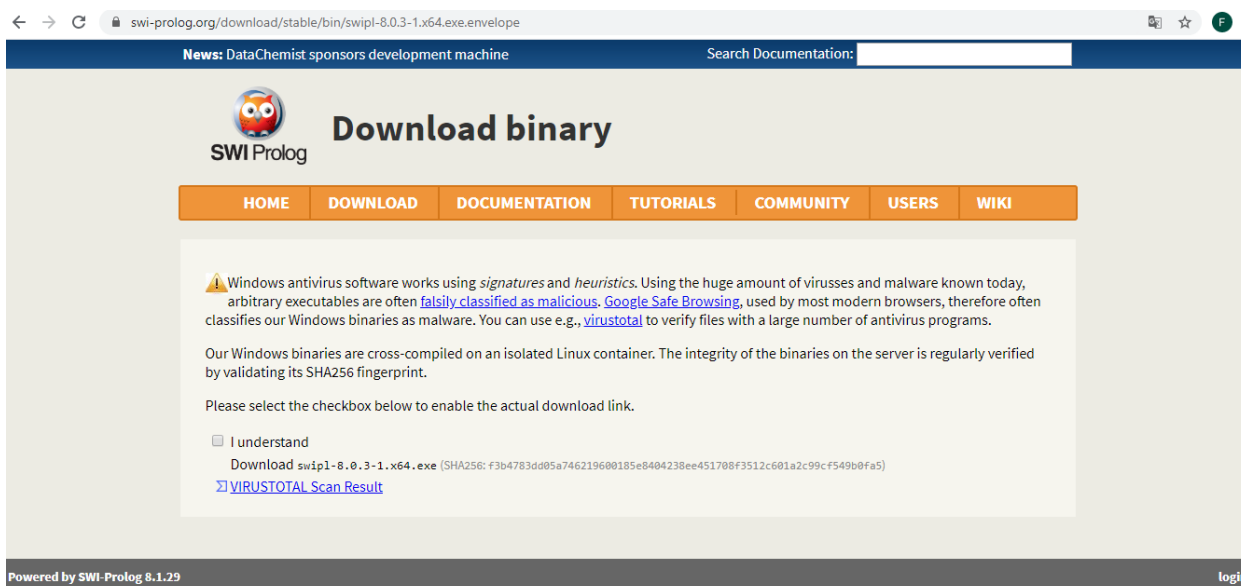
Yes

DESCARGA DE SWI-PROLOG

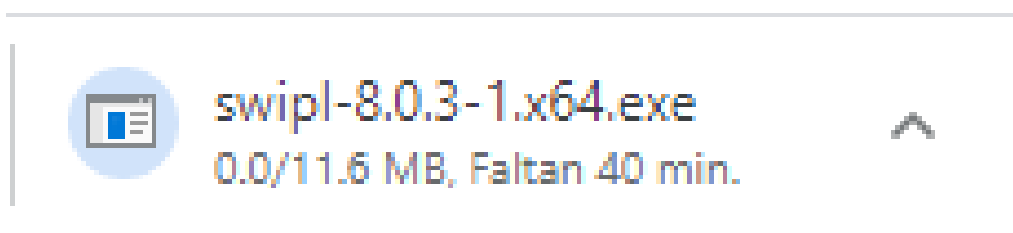
1. Se accede a la página oficial de SWI-PROLOG

<https://www.swi-prolog.org/download/stable/bin/swipl-8.0.3-1.x64.exe.envelope>

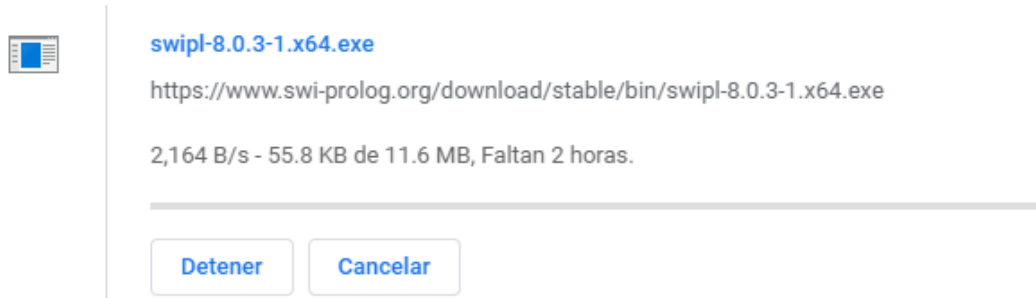
Se activa la casilla de **I understand** para poder realizar la descarga, posteriormente se activa se da clic en el apartado de Download swipl-8.0.3-1x64.exe



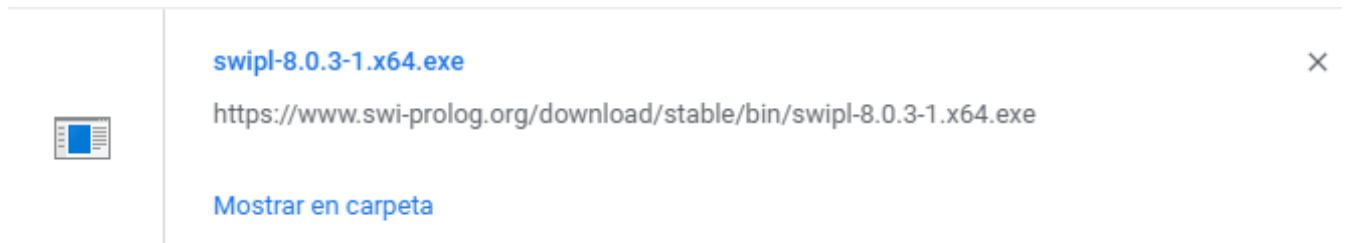
2. En la parte inferior de la pantalla se muestra el proceso de descarga del software.



3. Se debe tener paciencia con la internet, y dependiendo del ancho de banda que se tenga es el tiempo que se va a tardar en descargar.

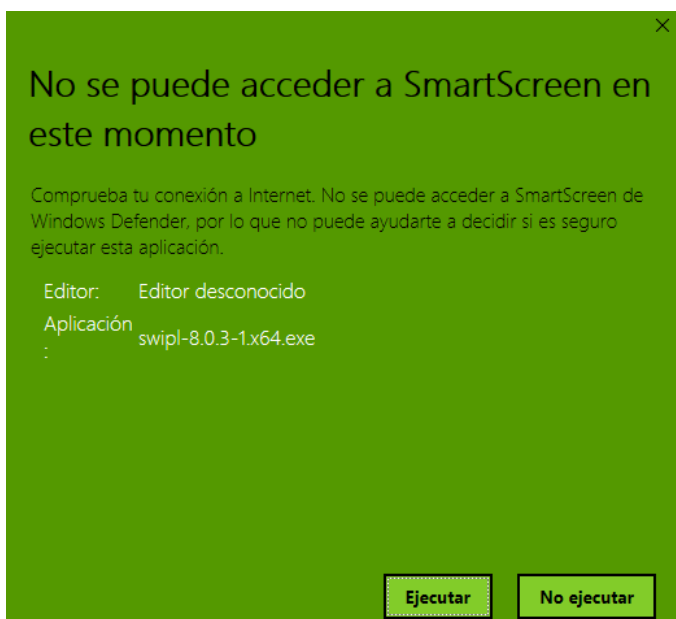


4. Al finalizar la descarga se muestra la siguiente imagen, para la ejecución del software.

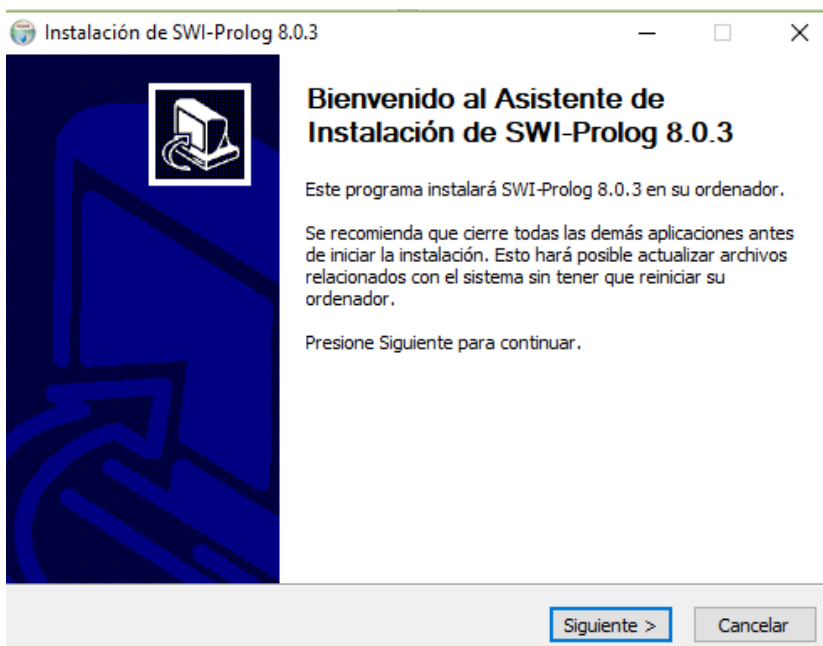


INSTALACION DE SWI-PROLOG

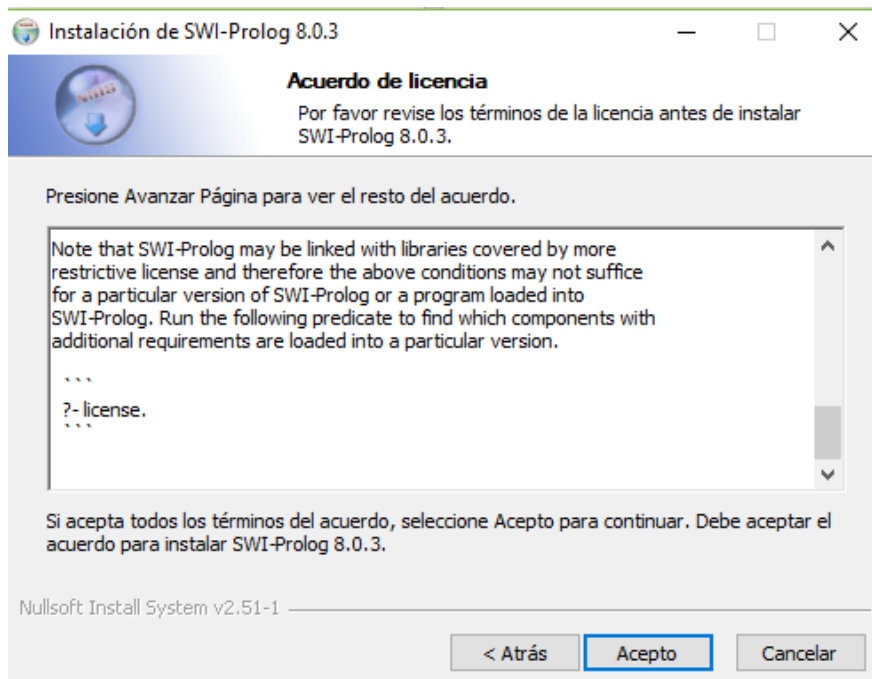
5. Se muestra la siguiente ventana, se da clic en la opción de ejecutar, para comenzar con la instalación.



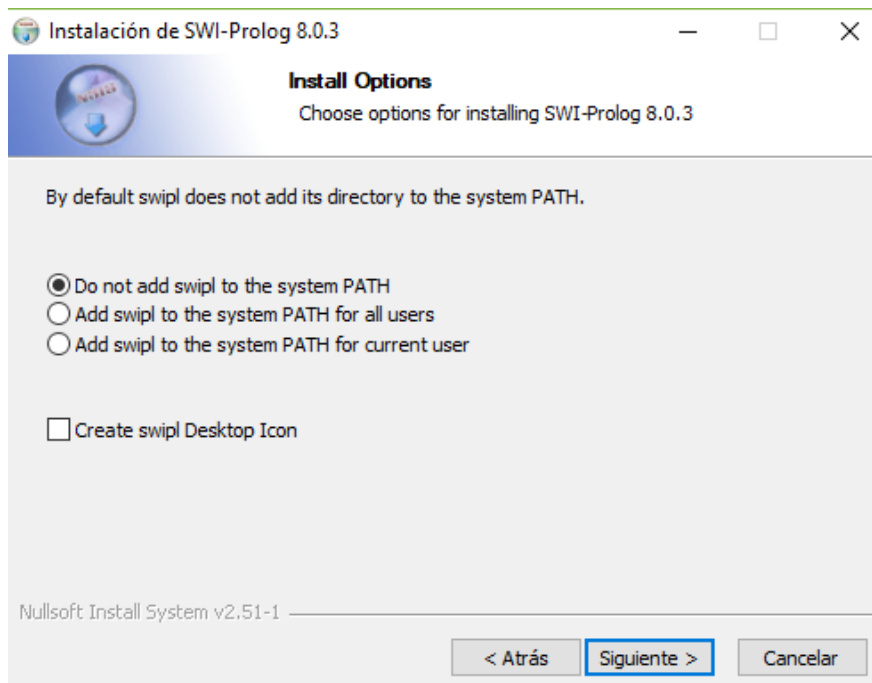
6. Se da clic en el botón de siguiente.



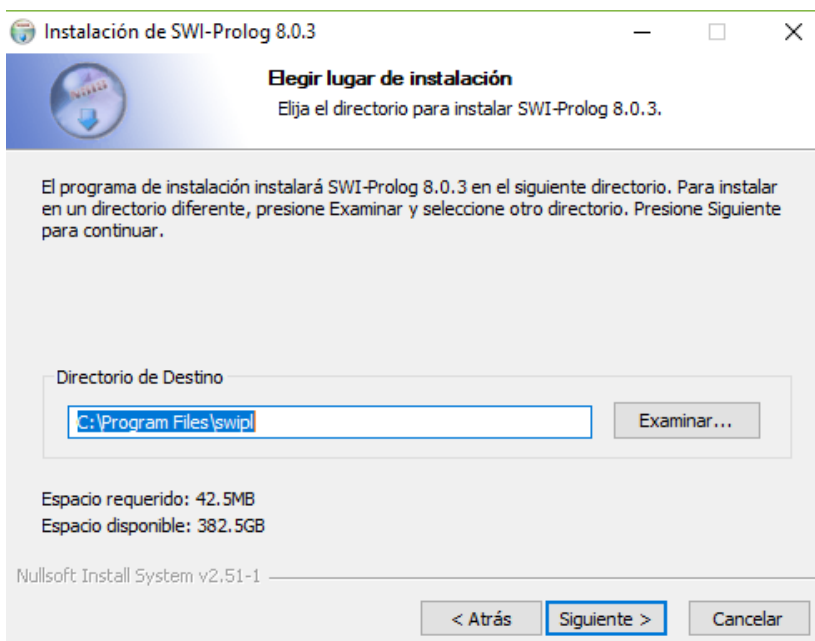
7. Se aceptan los términos y condiciones del software.



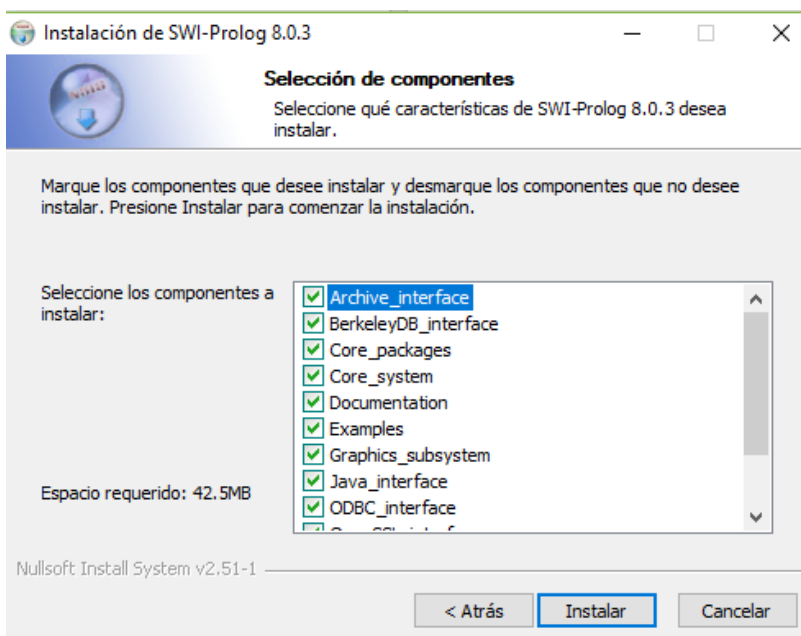
8. Se da clic en la opción de siguiente.



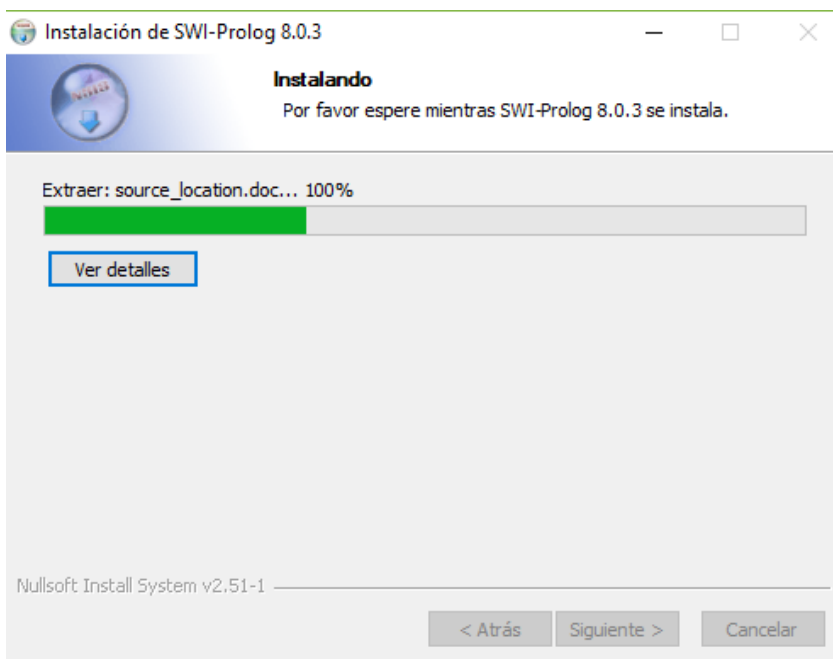
9. Se busca el lugar donde se va a instalar el archivo de instalación, en la parte de examinar, posteriormente clic en la opción de siguiente.



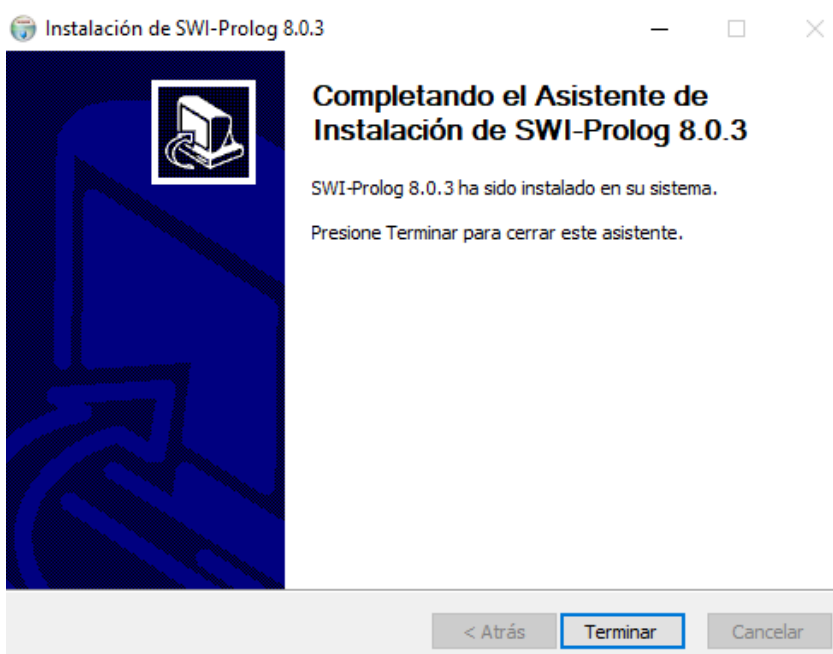
10. Se activan los componentes que se van a instalar, posteriormente clic en el botón de instalar.



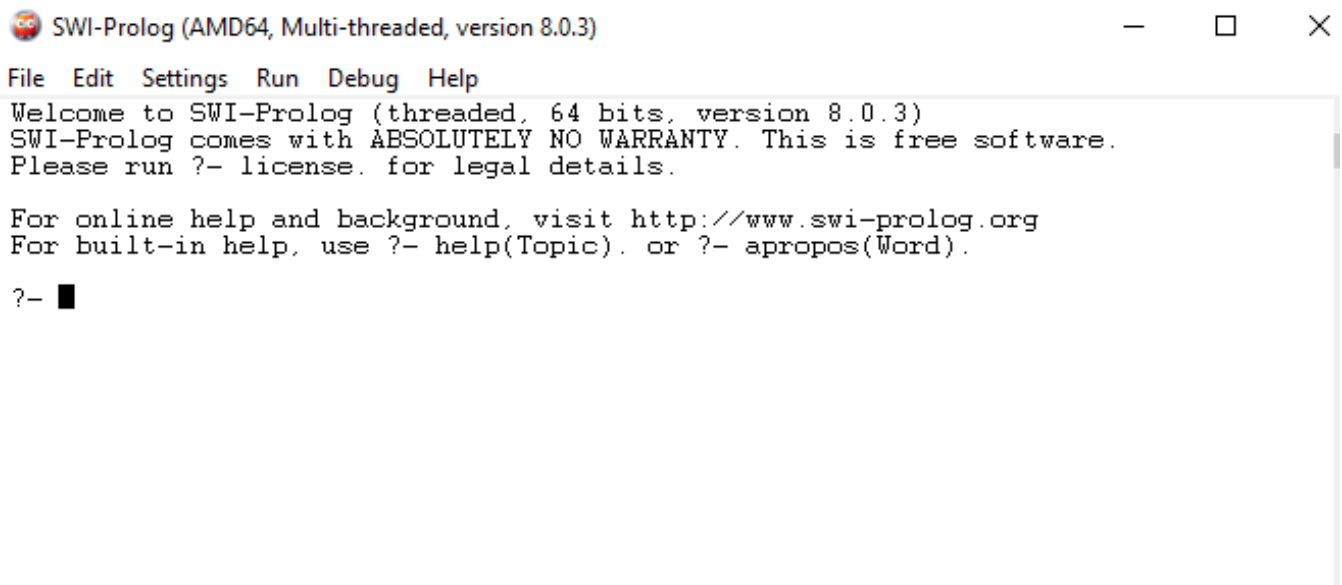
11. La siguiente imagen muestra el proceso de instalación.



12. La siguiente interfaz se muestra cuando ha finalizado la instalación, se da clic en el botón de Terminar.



13. Posteriormente se muestra la imagen del área de trabajo de SWI-PROLOG.



```
SWI-Prolog (AMD64, Multi-threaded, version 8.0.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- █
```


CONCLUSION

Este trabajo me pareció muy interesantes es con respecto a la unidad 3 de programación lógica funcional, es como un proceso para el entendimiento de la asignatura, porque primero se analizaron conceptos fundamentales, posteriormente el modelo de programación funcional.

Para la primera actividad realice un resumen del lenguaje de programación PROLOG, y para la segunda actividad se descargó un software llamado Swi-Prolo; la única complicación para la segunda actividad fue el ancho de banda, por lo que tardo en descargar.