

PROGRAMACION EN:



## ▀ QUE ES EL PROLOG ?

Esta orientado a la resolución de problemas mediante el calculo de predicados los mismos que están basados en:

- ❑ Preguntas a la base de datos
- ❑ Pruebas matemáticas

## HISTORIA

- ❖ Coulmerauer (1970) desarrolló un lenguaje para hacer deducciones de texto.
- ❖ Se aplica un mecanismo de resolución sobre predicados especiales, cláusulas de Horn, llamado unificación.
- ❖ La difusión del lenguaje se produce en los 80, pero de forma muy limitada debido a la falta de aplicaciones en dicho lenguaje.

## CARACTERISTICAS

- PROLOG especifica como debe ser la solución en vez de dar el algoritmo para su resolución.
- La solución se obtiene mediante búsqueda aplicando la lógica de predicados.

# ELEMENTOS DE PROLOG

- ✓ ATOMOS
- ✓ PREDICADOS
- ✓ HECHOS
- ✓ VARIABLES
- ✓ REGLAS

## ATOMOS

Definición genérica de un objeto del mundo que queremos representar

## PREDICADOS

Nos permite especificar características de los objetos de nuestro mundo o las relaciones entre ellos.



## ■ USO DE LOS ELEMENTOS

HECHOS = Describe una propiedad de un objeto

Ejemplo

- oro
- Hecho = valioso(oro).

Se puede definir interpretaciones de los objetos y las relaciones entre ellos

Ejemplo

Objetos = Juan, libro

Relación = tiene(juan,libro).

## USO DE LOS ELEMENTOS

VARIABLES = Representa el valor de un Atomo, esta puede estar instanciada o no instanciada.

Instanciada: cuando existe un objeto determinada representado por la variable

- ❑ Los nombres de variables comienzan siempre por una letra mayúscula
- ❑ Para variable anónima se representa el carácter subrayado (“\_”)



## ■ USO DE LOS ELEMENTOS

REGLAS: Usadas para significar que un hecho depende de uno o mas hechos.

- ❖ Una regla consiste en una cabeza y un cuerpo, unidos por el signo ":-".
- ❖ La cabeza está formada por un único hecho.
- ❖ El cuerpo puede ser uno ó mas hechos (conjunción de hechos), separados por una coma (","), que actúa como el "y" lógico.
- ❖ Las reglas finalizan con un punto (".").

## Operadores lógicos en PROLOG

- ❖ ; = or (o)
- ❖ , = and (y)
- ❖ \+ = not (no)
- ❖ :- = if (si)

## Operadores de comparación

- ❖  $X == Y$  son el mismo numero
- ❖  $X != Y$  son diferentes números
- ❖  $X < Y$  X es menor que Y
- ❖  $X > Y$  X es mayor que Y
- ❖  $X \leq Y$  X es menor o igual que Y
- ❖  $X \geq Y$  X es mayor o igual que Y

## Operadores Aritméticos

- ❖  $X + Y$  la suma de X e Y
- ❖  $X - Y$  la resta de X e Y
- ❖  $X * Y$  la multiplicación de X e Y
- ❖  $X / Y$  la división de X e Y
- ❖  $X // Y$  el cociente entero de X entre Y
- ❖  $X \text{ mod } Y$  el resto de dividir X entre Y

## Operadores de Entrada y Salida

- ❖ `write()` para números
- ❖ `write(' ')` para cadenas de texto
- ❖ `nl` Salto de línea en flujo de salida
- ❖ `read()` Captura de términos de entrada
- ❖ `put( )` Caracteres del código ASCII salida
- ❖ `get( )` Captura el numero de letra en código ASCII

## Comandos para usar en prolog


- halt: Para salir
- % Para comentarios
- ["Directorio"]. Para ejecutar un proceso especificado la ruta del archivo



# INSTALACIÓN


## WINDOWS:

- ✓ <https://www.swi-prolog.org/download/stable/bin/swipl-8.2.3-1.x64.exe.envelope>

**SWI Prolog**

## Download binary

HOME	DOWNLOAD	DOCUMENTATION	TUTORIALS	COMMUNITY	USERS	WIKI
------	----------	---------------	-----------	-----------	-------	------

 Windows antivirus software works using *signatures* and *heuristics*. Using the huge amount of viruses and malware known today, arbitrary executables are often [falsily classified as malicious](#). [Google Safe Browsing](#), used by most modern browsers, therefore often classifies our Windows binaries as malware. You can use e.g., [virustotal](#) to verify files with a large number of antivirus programs.

Our Windows binaries are cross-compiled on an isolated Linux container. The integrity of the binaries on the server is regularly verified by validating its SHA256 fingerprint.

Please select the checkbox below to enable the actual download link.

☒ I understand

[Download swipl-8.2.3-1.x64.exe](#) (SHA256: b4297d1d60ce050daf0d07764487370ef475cad70b54a96b074f83138d792cd6)

[VIRUSTOTAL Scan Result](#)

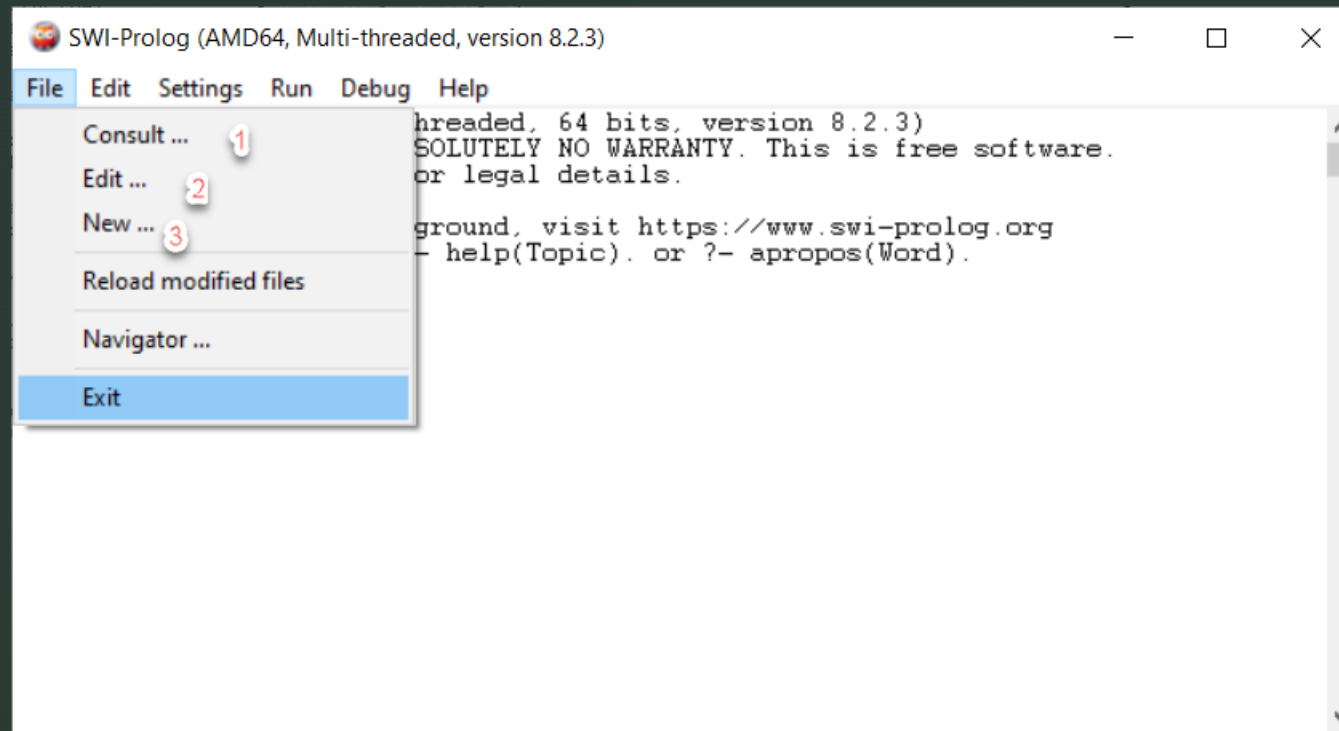
# EJECUCIÓN



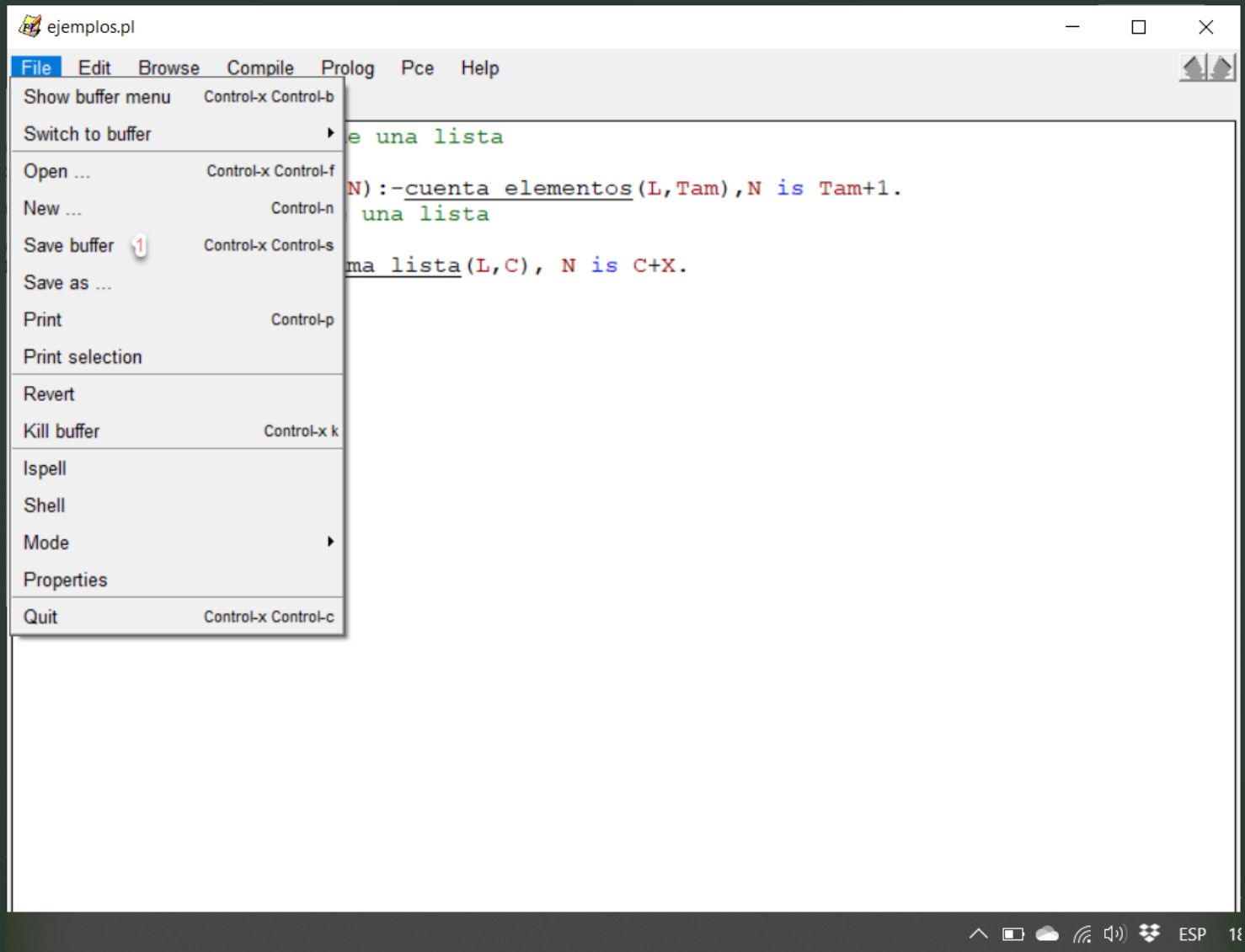
```
SWI-Prolog (AMD64, Multi-threaded, version 8.2.3)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

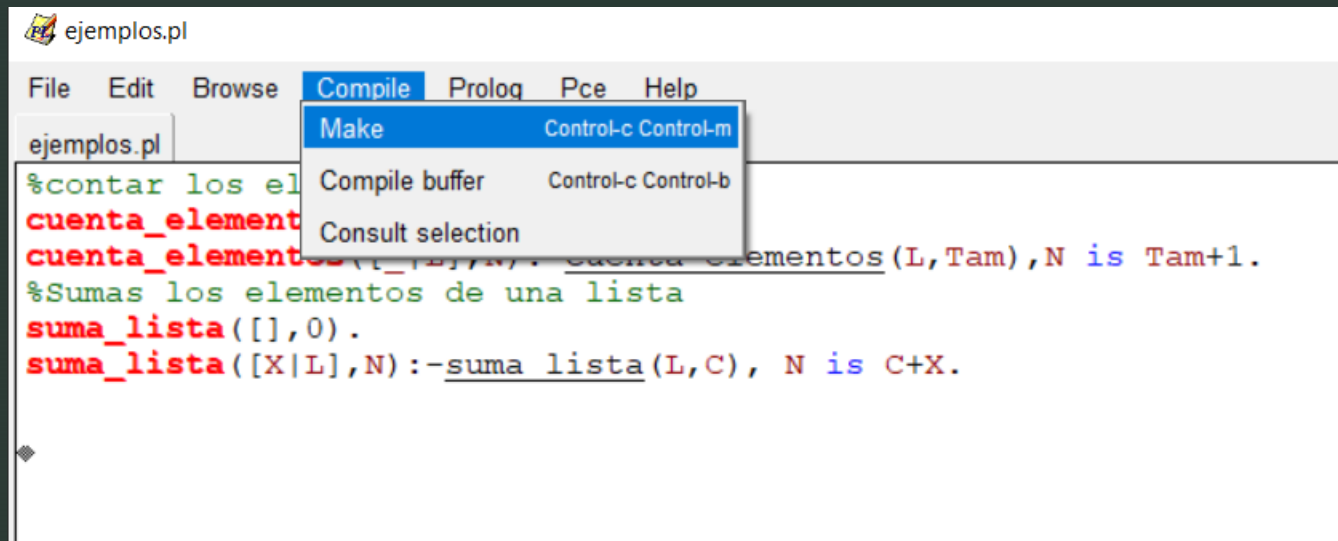
?-
```



- ☐ 1 abre documentos a ejecutar .pl
- ☐ 2 edita documentos .pl
- ☐ 3 crea nuevos documentos .pl



- ❖ guardamos el archivo en un espacio en el disco para ejecutar



The screenshot shows a Prolog IDE window titled "ejemplos.pl". The menu bar includes "File", "Edit", "Browse", "Compile", "Prolog", "Pce", and "Help". The "Compile" menu is open, showing options: "Make" (with shortcuts "Control-c Control-m"), "Compile buffer" (with shortcuts "Control-c Control-b"), and "Consult selection". The code editor contains the following Prolog code:

```
%contar los el  
cuenta_element  
cuenta_elementos(L,Tam),N is Tam+1.  
%Sumas los elementos de una lista  
suma_lista([],0).  
suma_lista([X|L],N):-suma_lista(L,C), N is C+X.
```

- ❑ prepara el código para ejecutar