

**IC4700**

*Lenguajes de Programación*

**II Proyecto**

*(Gestión de Inventarios)*

**Integrantes:**

Mynell Jemuel Myers Hall

Andrew Denilson López Herrera

Ingeniería en Computación

Prof. Allan Rodríguez Dávila

Semestre 1, 2024

24/04/2024

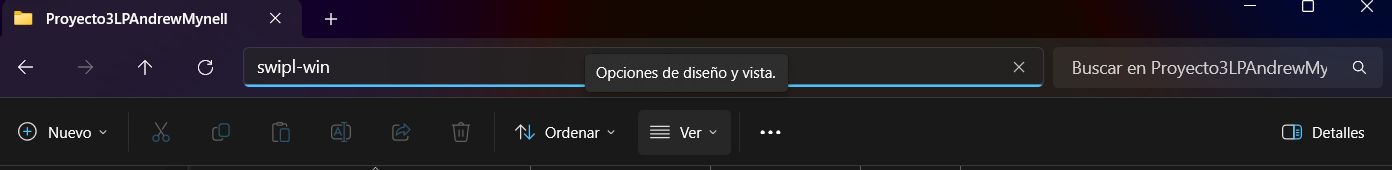
# Manual de usuario

Este proyecto fue construido en Windows usando la herramienta SWI-PROLOG, por lo que el incumplimiento de este entorno puede ocasionar fallas o errores en el programa. (Si no posee el entorno, puede consultar una compilación Online)

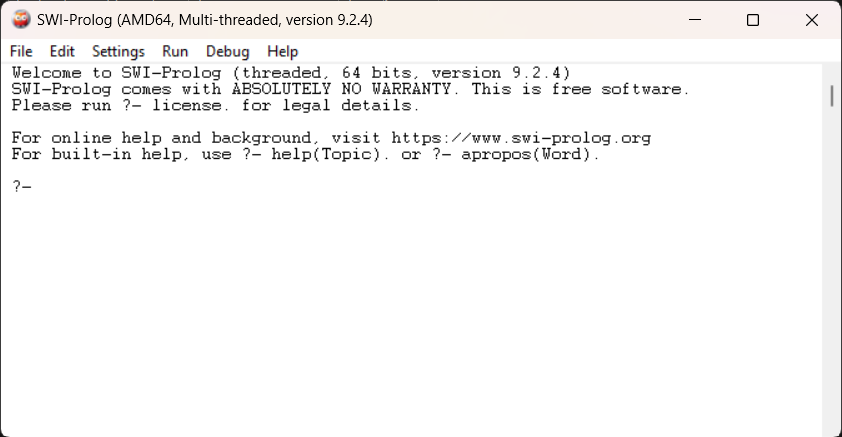
## Compilación

Ante todo, descargaremos el proyecto desde el repositorio de GitHub (Puede encontrar el link del repositorio en la sección de final).

Luego de descargar el proyecto nos posicionaremos en la carpeta programa y abriremos el swipl-win desde la ruta en la posición de la carpeta.



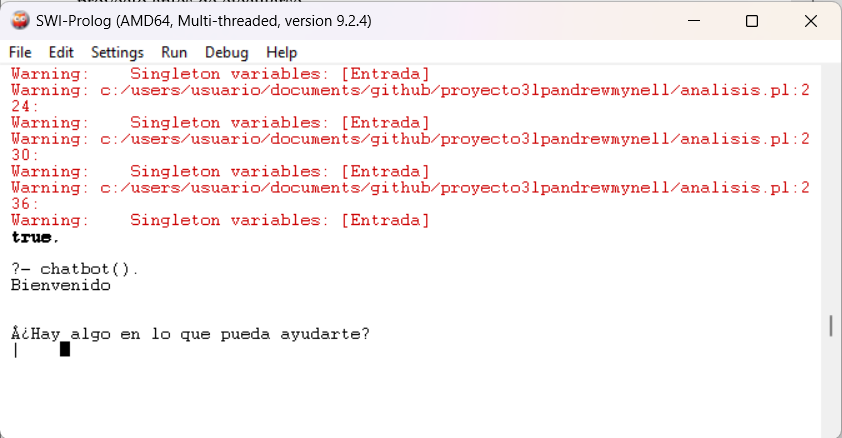
Luego de eso se abrirá una consola de Prolog, lista para comenzar la ejecución.



## Ejecución

En la consola escribir los siguiente: consult('conocimiento.pl'). . Esto hasta mostrar algunas advertencias que marcaran la carga del programa al sistema.

Luego de eso escribir el siguiente comando: chatbot(). Con esto el programa entrara en ejecución dando la bienvenida.



## Uso del programa

Al comenzar la ejecución del programa será recibido por una bienvenida más una pregunta para obtener las consultas del usuario. El programa esperara consultas naturales con respecto a matemática básica, chistes y generación de código en base al conocimiento.

# Pruebas de funcionalidad

La suma funciona con la entrada “Hazme la suma de 2+3+4+5”

Texto

Descripción generada automáticamente  
 La resta funciona con la entrada “Quiero restar 4-5-6”

Texto

Descripción generada automáticamente con confianza media  
 La multiplicación funciona con la entrada “multiplica 9\*2\*2”

Gráfico

Descripción generada automáticamente

La división funciona con la entrada “Chatbot, divida 8/1/2/3”

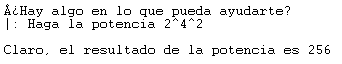
Texto

Descripción generada automáticamente

El modulo funciona con la entrada “Hazme el modulo 10 % 3”

Texto

Descripción generada automáticamente con confianza baja  
 La potencia funciona con la entrada “Haga la potencia 2^4^2”



La raíz cuadrada funciona con la entrada “Por favor, calcula la raiz cuadrada de 25”

Texto

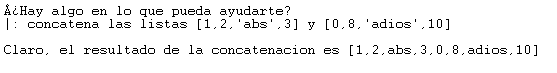
Descripción generada automáticamente con confianza media

Contar la longitud funciona con la entrada “cuenta los elementos de [1,2,3,4,5,6]”

Texto

Descripción generada automáticamente con confianza media

La concatenación de listas funciona con la entrada “concatena las listas [1,2,'abs',3] y [0,8,'adios',10]”



El reverso de la lista funciona con la entrada “Haga el inverso de la lista [1,2,3,4,5,6,7,8,9,10]”

Texto

Descripción generada automáticamente

Extraer el primer elemento funciona con la entrada “Dame el primer elemento de la lista [2,6,45,8,6,9]”

Texto

Descripción generada automáticamente

Extraer el ultimo elemento funciona con la entrada “Dame el ultimo elemento de la lista [2,6,45,8,6,9]”

Texto

Descripción generada automáticamente con confianza media

Extraer el máximo elemento funciona con la entrada “Dame el elemento mas grande de la lista [1,2,3,4,48,6]”

Texto

Descripción generada automáticamente

Extraer el mínimo elemento funciona con la entrada “Dame el elemento minimo de la lista [-1,2,3,4,48,6]”

Diagrama, Texto

Descripción generada automáticamente con confianza media

El n’ésimo elemento funciona con la entrada “Dame el elemento en la posicion 3 de la lista [7,6,5,4,3,2,1]”

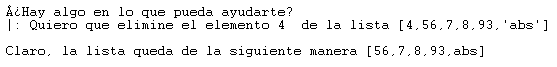
Texto

Descripción generada automáticamente

El elemento existe funciona con la entrada “Digame si existe el elemento 2 en la lista [7,9,5,4,21,3,4,2,5,7,6]”

Texto

Descripción generada automáticamente con confianza media

Eliminar elemento funciona con la entrada “Claro, la lista queda de la siguiente manera [56,7,8,93,’abs’]”

Los chistes funcionan con la entrada “cuentame un chiste”

Texto

Descripción generada automáticamente

Los refranes funcionan con la entrada “dime un refran”

Texto

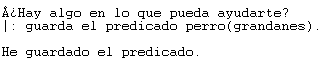
Descripción generada automáticamente

Los consejos de estudio funcionan con la entrada “Dame un consejo de estudio”

Texto

Descripción generada automáticamente

El guardado de nuevos predicados funciona con la entrada “guarda el predicado perro(grandanes).



La impresión de predicados funciona con la siguiente entrada “

Texto

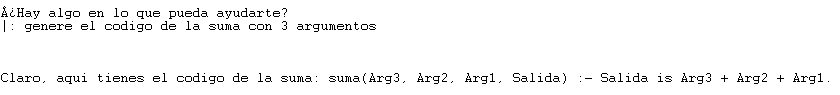
Descripción generada automáticamente

La despedida funciona con la entrada “hasta luego mi lindo chatbot”

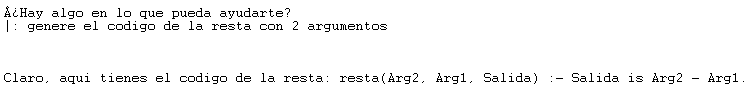
Texto

Descripción generada automáticamente

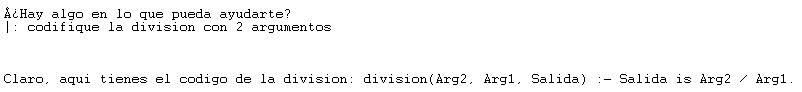
La generación del código de suma funciona con la siguiente entrada “genere el codigo de la suma con 3 argumentos”



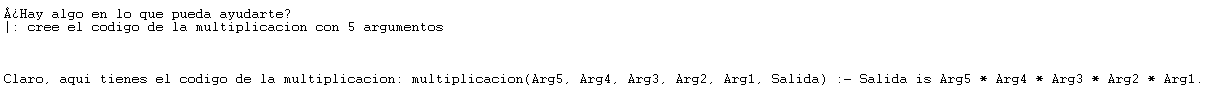
La generación del código de resta funciona con la siguiente entrada “genere el codigo de la resta con 2 argumentos”



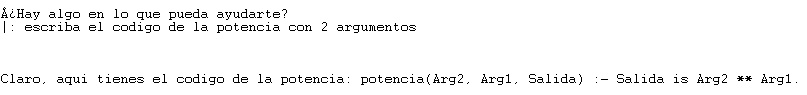
La generación del código de la división funciona con la siguiente entrada “codifique la division con 2 argumentos”



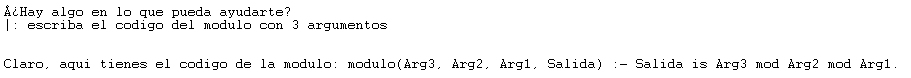
La generación del código para la multiplicación funciona con la entrada “cree el codigo de la multiplicacion con 5 argumentos”



La generación del código para la potencia funciona con la siguiente entrada “escriba el codigo de la potencia con dos argumentos”



La generación del código del módulo funciona con la siguiente entrada “escriba el codigo del modulo con 3 argumentos”



La generación del código para la raíz cuadrada funciona con la siguiente entrada “escriba el codigo de la raiz cuadrada”

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Descripción del problema

Su trabajo consiste en implementar una aplicación de escritorio (consola) de un chatbot con conocimiento aritmético de listas, que le permita generar código y resolver operaciones, utilizando el lenguaje Prolog y los principios de la Programación Lógica. El sistema inicia con un saludo al usuario y debe permitir una interacción constante, sin cortarla con el usuario salvo que el usuario indique una frase de despedida similar o “hasta luego” o “adiós mi lindo chatbot”.

## Temas de conocimiento

El chatbot tendrá las siguientes temáticas:

Operaciones aritméticas, para enteros y flotantes con n argumentos input:

* Suma
* Resta
* Multiplicación
* División
* Módulo (entero)
* Potencia
* Raiz cuadrada (1 argumento)

Listas con uno, dos o n argumentos input:

* Longitud (n argumentos)
* Concatenación (n argumentos)
* Reverso (1 argumento)
* Primer y último elemento (1 argumento)
* Máximo y mínimo (1 argumento)
* N’ésimo elemento (2 argumentos)
* Existe elemento (2 argumentos)
* Eliminar un elemento (2 argumentos)

Nota: en los casos en que se pasa n argumentos se debe “retornar” una única lista

Manejo de la base de conocimiento:

* Insertar predicados (n argumentos)
* Imprimir todos

## Consultar resultado

El chatbot debe ser capaz de generar resultados al evaluar los argumentos que le suministra el usuario. Generación de código El chatbot será capaz de generar código prolog de los temas que conoce, según los argumentos permitidos (1, 2 o n).

## Conversación

El chatbot deberá mantener la conversión en todo momento, con salida alterna en caso de no saber algún tema. Se pude establecer algunas reglas, para el paso de argumentos tanto para generar código como para resolver, pero no se le podrá solicitar al usuario que indiqué la cantidad previamente. El chatbot deberá discernir sobre lo que está solicitando el usuario.

No todas las entradas del usuario podrían ser para requerimientos, podría solicitar temas triviales como: chistes, refranes o consejos de estudio.

## Inicio y Salida del sistema

Al iniciar el sistema se carga a la BC, desde un archivo, los hechos que mantienen registros del sistema. Al salir (o en el momento de ejecución a conveniencia) se debe guardar en el archivo los nuevos hechos, generados en la ejecución del programa.

# Diseño del programa

Se empleo un enfoque de tokens, utilizando diferentes métodos para reconocer palabras que clave que encajasen en los comandos básicos del programa.

También se manejó la permanencia de los predicados generados por los usuarios en un archivo especifico que se tomara en cuenta para la impresión de todos los predicados en programa.

# Módulos utilizados

* regex.

# Análisis de resultados

|  |  |  |  |
| --- | --- | --- | --- |
| Objetivos | Sí | No | Observaciones |
| Documentación interna | X |  |  |
| Documentación externa | X |  |  |
| Operaciones Aritméticas | X |  |  |
| Listas | X |  |  |
| Manejo de Base de Conocimiento | X |  |  |
| Generación de código | X |  |  |
| Permanencia de predicados | X |  |  |

# Repositorio

<https://github.com/Mynell2022/Proyecto3LPAndrewMynell.git>