



VERACRUZ  
GOBIERNO  
DEL ESTADO



SEV  
Secretaría  
de Educación



DET  
Dirección de Educación  
Tecnológica del Estado  
de Veracruz



# **Instituto Tecnológico Superior de Chicontepec**

**Ingeniería en Sistemas Computacionales**

**Nombre:**

**Camelia Bautista Hernández**

**Docente:**

**Ing. Efrén Flores Cruz**

**Asignatura:**

**Programación Lógica y Funcional**

**Unidad 4**

**Modelo de Programación Lógica**

**Trabajo:**

**Ejercicios en Prolog uso de Interfaz**

**8º SEMESTRE**

**Fecha de entrega:**

**28 de Mayo del 2020**









## CONTENIDO

<b>INTRODUCCION .....</b>	<b>3</b>
<b>FUNCIONAMIENTO.....</b>	<b>4</b>
<b>RESULTADOS.....</b>	<b>5</b>
1. Ventanas en prolog .....	5
2. Crear ventana y etiqueta .....	6
3. Crear ventana y botón .....	7
4. Caja de texto .....	8
5. Copiar dato de un numero .....	9
<b>TAREA .....</b>	<b>11</b>
6. Suma usando ventanas .....	11
7. Calcular la edad.....	13
8. Menú grafico.....	15
Operaciones básicas .....	17
Área de Figuras .....	22
<b>CONCLUSIÓN .....</b>	<b>26</b>

## INTRODUCCION

En el presente documento nos muestra sobre los ejercicios que se realizar atreves de del software en SWI-Prolog. Está basado en dos ventanas: La ventana principal, con una línea donde se ejecutan los objetivos o La ventana de edición, donde se editan y compilan los programas o La ventana principal siempre existe, la de edición sólo cuando se está usando.

PROLOG busca automáticamente en la base de datos si existe un hecho que se puede unificar (es decir, tiene el mismo nombre de predicado, el mismo número de argumentos en los siguientes ejercicios que muestra son sobre prolog uso de interfaz las cuales son las siguientes:

-  Ventanas en prolog.
-  Crear ventana y etiqueta.
-  Crear ventana y botón.
-  Caja de texto.
-  Copia un número.
-  Suma usando ventanas.
-  Ventana para calcular la edad.
-  Ventana con operaciones básicas.

## FUNCIONAMIENTO

Los programas Prolog hay que editarlos con un editor de texto. Se puede usar el Notepad, pero no conviene, es bastante limitado ya que en este caso se utiliza el mismo editor del programa SWI-Prolog Editor la cual se debe de descargar.

También se pueden usar editores de texto de uso general, p.ej. el UltraEdit. En la página de editores contamos algunos tipos sobre cómo trabajar con los editores. Conviene guardar los archivos con extensión .pl.

El SWI se usa para hacer consultas a partir de un programa. Hay varias formas de "cargar" un programa en Prolog, mencionamos algunas:

1. darle doble click a la fuente .pl en el administrador de archivos o similar (p.ej. Windows commander), abre el SWI y carga el programa.
2. abrir el SWI, y después File -> Consult.
3. escribir en la línea de comando prolog nombreArchivo.pl
4. abrir el SWI, y en la consola utilizar el predicado consult/1: ?- consult("clase2.pl").

Si hay errores en el programa, aparecen en la ventana de SWI indicando la línea de cada error. También pueden aparecer warnings, esos no importan. Si hay errores (te das cuenta porque la línea correspondiente empieza con **ERROR:**) el programa no se carga, hay que corregir los errores.

Una vez que se cargó el programa, las consultas se hacen escribiéndolas en la ventana del SWI, al lado del prompt ?- . Recordar que las consultas también tienen que terminar con punto.

## RESULTADOS

### 1. Ventanas en prolog

En el primer ejercicio se realizó la creación de etiqueta en ventana la cual es lo principal para poder crear lo siguiente que se desea agregar.

**New**=creando un nuevo objeto

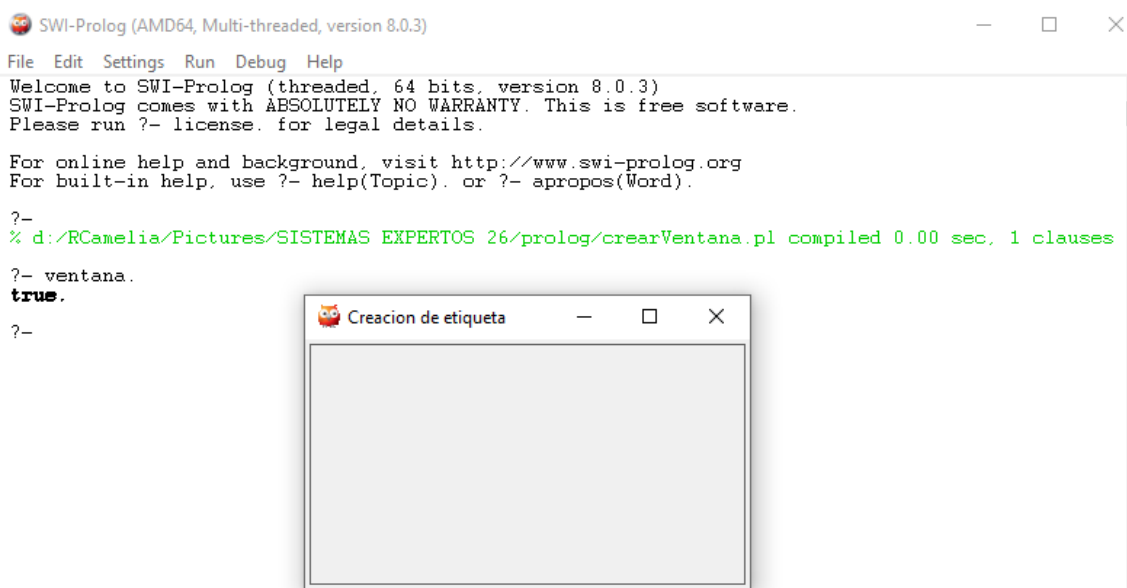
**Ventana, dialog**= ventana es una variable y dialogo es un objeto también se agrega el nombre del título en la ventana.

**Send (Ventana, open) = send** es colocar y abrir la ventana.

Así como nos muestra en la parte del código que es utilizado solo para crear la ventana .

```
ventana:-new(Ventana,dialog('Creacion de etiqueta')),  
send(Ventana,open) .
```

Una vez ya realizado la programación ejecutamos el programa presionamos la opción el **FILE** y seleccionamos **CONSULT** una vez que haya compilado nos muestra lo siguiente lo cual agregamos el nombre que le ayamos agregado la cual es ventana. Y nos muestra que es **true**. la cual es verdadero y al final nos muestra la ejecución la cual es la creación de la etiqueta .

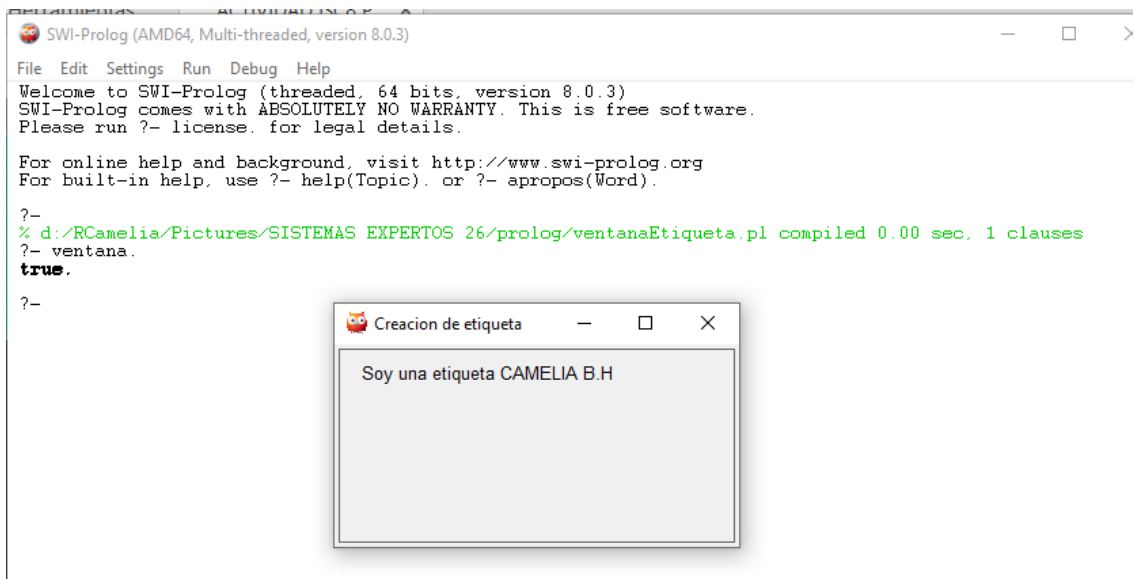


## 2. Crear ventana y etiqueta

En el siguiente ejercicio nos muestra el código de la programación para al crear una ventana nos muestre una etiqueta en la cual podemos agregar un texto ya que en este caso nos muestra “Soy una etiqueta CAMELIAB.H”.

```
1 ventana:-new(Ventana,dialog('Creacion de etiqueta')),
2   new(Etiqueta,label(etiqueta,'Soy una etiqueta CAMELIA B.H')),
3   send(Ventana,append,Etiqueta),
4   send(Ventana,open).
5
```

Una vez que hayamos realizado la programación para la muestra de la creación de ventana con etiqueta la ejecutamos y escribimos el nombre ventana. La cual es la variable que está escrita al consultarlo nos muestra enseguida la ventana de etiqueta.



### 3. Crear ventana y botón

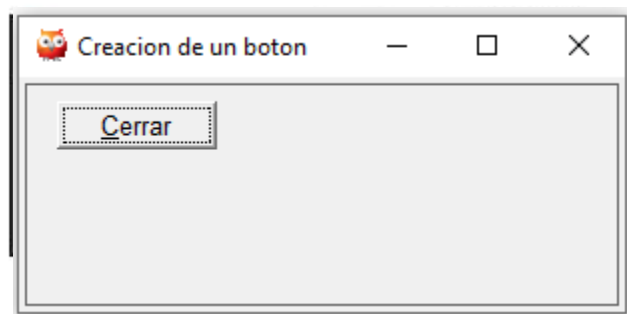
En el siguiente ejercicio nos muestra sobre la creación de un botón en una ventana la cual da la funcionalidad de cerrar.

**name:** indica el nombre del botón (si no se especifica la etiqueta que queramos que tenga el botón, entonces adoptará una etiqueta con el mismo texto que name)

**message:** indica el mensaje o acción que queremos que sea ejecutado cuando pulsemos sobre el botón con el ratón. label: Indica la etiqueta que queremos que se muestre en el botón.

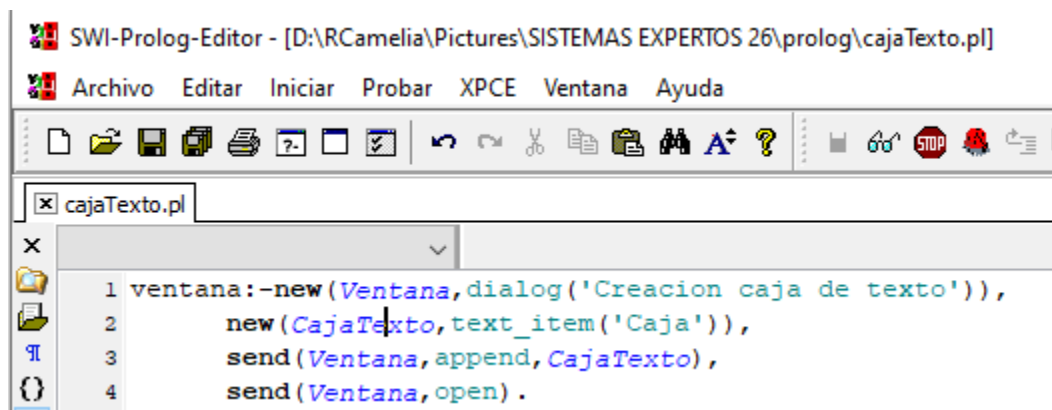
```
1 ventana:-new(Ventana,dialog('Creacion de un boton')),  
2   new(Boton,button('Cerrar',message(@prolog,cerrarVentana,Ventana))),  
3   send(Ventana,append,Boton),  
4   send(Ventana,open).  
5  
6 cerrarVentana(Ventana):-send(Ventana,destroy).
```

```
% d:/Camelia/Pictures/SISTEMAS EXPERTOS 26/prolog/crearBoton.pl compiled 0.00 sec. 1 clauses  
?- ventana.  
true.  
?-
```



## 4. Caja de texto

En el cuarto ejercicio es sobre realizar en la ventana agregar una caja de texto para la creación de la caja ya que al realizarlo nos muestra por ello podemos escribir en ella.

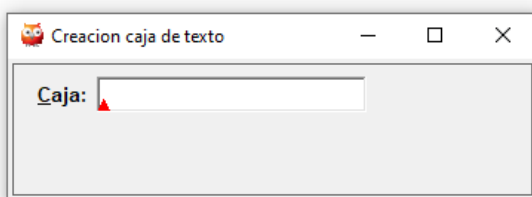


```
SWI-Prolog-Editor - [D:\RCamelia\Pictures\SISTEMAS EXPERTOS 26\prolog\cajaTexto.pl]
Archivo  Editar  Iniciar  Probar  XPCE  Ventana  Ayuda

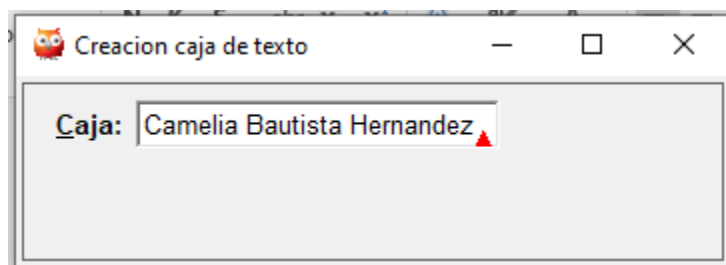
cajaTexto.pl
1 ventana:-new(Ventana,dialog('Creacion caja de texto')),
2   new(CajaTexto,text_item('Caja')),
3   send(Ventana,append,CajaTexto),
4   send(Ventana,open).
```

Una vez realizado la programación ejecutamos el programa escribiremos ventana ya que es el nombre de la variable de la que se ejecuta y nos muestra la siguiente caja de texto la cual creamos.

```
% d:\RCamelia\Pictures\SISTEMAS EXPERTOS 26\prolog\cajaTexto.pl compiled 0.00 sec, 1 clauses
?- ventana.
true.
?-
```



permite ver el promedio, por lo que cuenta con una caja de texto para introducir cualquier palabra que deseemos agregar ya que en este caso decidí agregar mi nombre para verificar que si se puede agregar cierto texto.





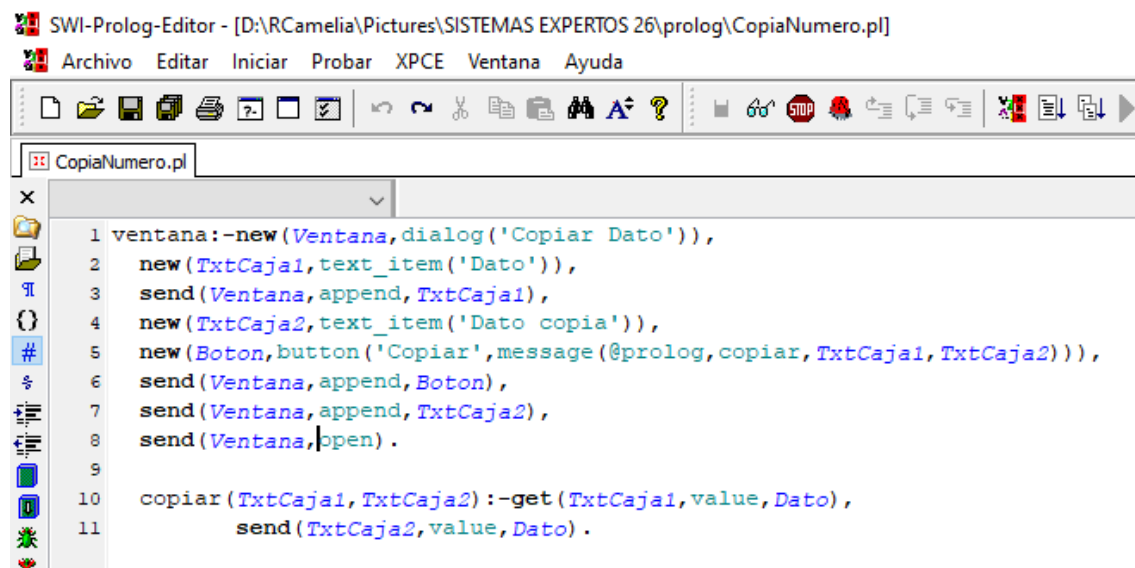
## 5. Copiar dato de un numero

existen 3 predicados con los que se vamos a trabajar. Estos predicados sirven para crear objetos, enviar mensajes a objetos, recibir mensajes de objetos.

**new:** Este predicado recoge dos parámetros, el primero recogería la referencia que se le asigna al nuevo objeto, ya que new se usa para crear objetos. El segundo parámetro le indicaría el objeto que se quiere crear.

**send:** El primer parámetro del predicado es una referencia al objeto al que deseamos enviarle un mensaje. El segundo parámetro indicará el método al que queremos invocar, lo cual indicaremos junto a los argumentos que queremos enviarle al método.

**get:** Los dos primeros parámetros tienen el mismo significado que para send, sin embargo, el último parámetro sirve para recoger el valor que nos devuelva el método que hallamos invocado.

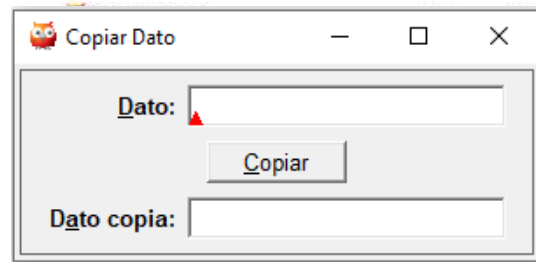


```
SWI-Prolog-Editor - [D:\RCamelia\Pictures\SISTEMAS EXPERTOS 26\prolog\CopiaNumero.pl]
Archivo  Editar  Iniciar  Probar  XPCE  Ventana  Ayuda

1 ventana:-new(Ventana,dialog('Copiar Dato')),
2   new(TxtCaja1,text_item('Dato')),
3   send(Ventana,append,TxtCaja1),
4   new(TxtCaja2,text_item('Dato copia')),
5   new(Boton,button('Copiar',message(@prolog,copiar,TxtCaja1,TxtCaja2))),
6   send(Ventana,append,Boton),
7   send(Ventana,append,TxtCaja2),
8   send(Ventana,open).
9
10 copiar(TxtCaja1,TxtCaja2):-get(TxtCaja1,value,Dato),
11   send(TxtCaja2,value,Dato).
```

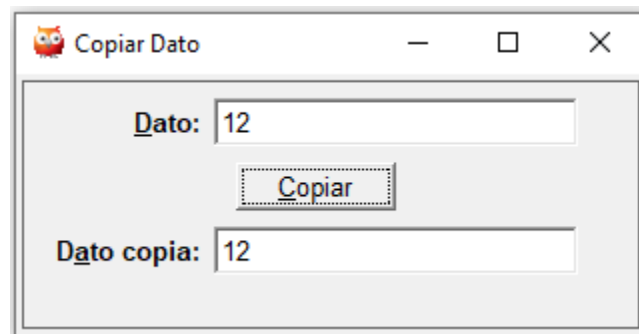
En la siguiente imagen nos muestra sobre el código la cual es la programación sobre copiar un dato en la ventana ahora en este siguiente ejercicio nos muestra dos tipos de cajas que una será donde se ingresa copiar dato y el otro sería el dato de la copia, así como también contiene un botón la cual es llamada copiar.

```
% d:/RCamelia/Pictures/SISTEMAS EXPERTOS 26/prolog/CopiaNumero.pl compiled 0.02 sec. 0 clauses  
?- ventana.  
true.  
?-
```



The screenshot shows a window titled "Copiar Dato" with a standard Windows title bar (minimize, maximize, close buttons). Inside the window, there are two text input fields. The first field is labeled "Dato:" and is currently empty. Below it is a button labeled "Copiar". The second field is labeled "Dato copia:" and is also empty.

Como podemos observar nos muestra los resultados que se tiene al realizar la programación de la copia de un dato la cual cuenta con dos cajas de texto una es el dato que vamos a ingresar en la parte del medio nos muestra el botón y para finalizar nos muestra el dato copia en la cual nos muestra el resultado de esta actividad.



The screenshot shows the same "Copiar Dato" window. The "Dato:" field now contains the number "12". The "Copiar" button is still present. The "Dato copia:" field also contains the number "12", indicating that the value from the first field has been copied.

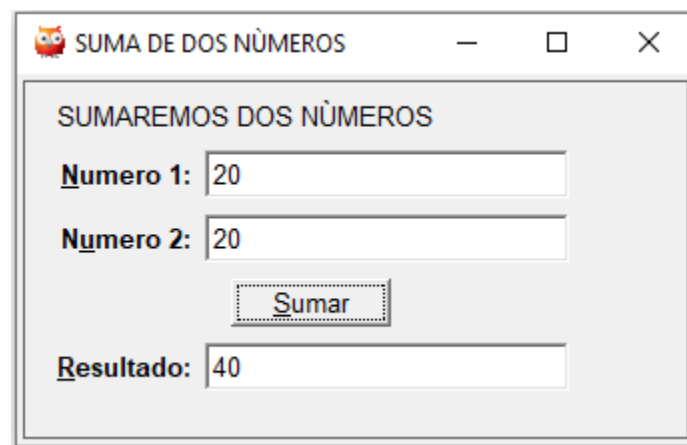
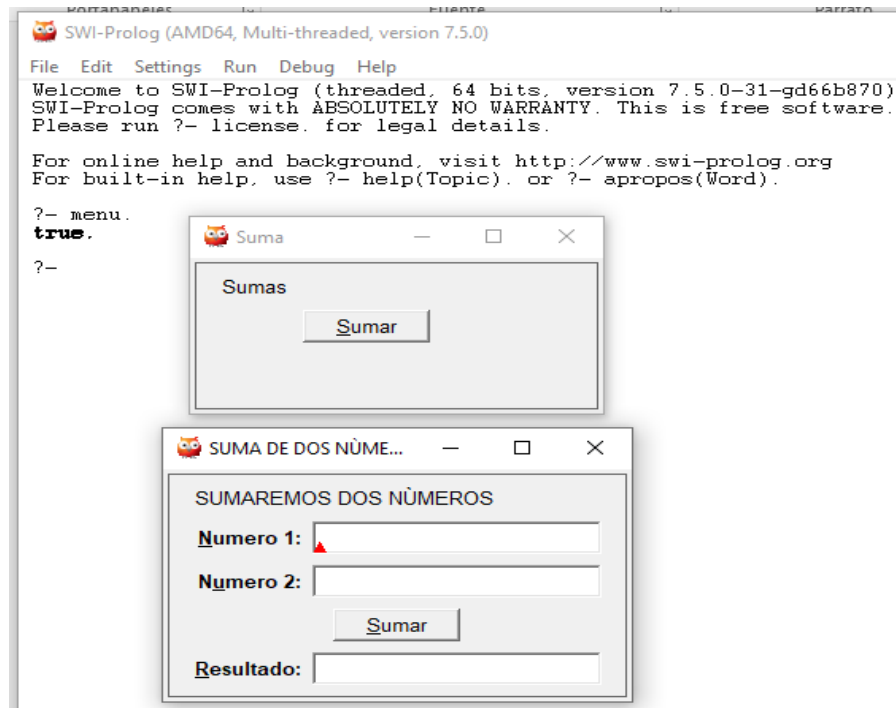
## TAREA

### 6. Suma usando ventanas

En el siguiente ejercicio nos muestra la suma de dos números. Primero se creamos la ventana, así como también lo que va a contener en el primer apartado nos muestra el código, después con el **send** nos permitir realizar la función dada. En la programación de funcionamiento se agrega la suma del valor uno con la suma del valor 2.

```
Suma.pl
1 menu:-new(Ventana,dialog('Suma')),
2 new(Opba, label(opba, 'Sumas')),
3 new(Sumar,button('Sumar',message(@prolog,vsumar))),
4 send(Ventana,append,Opba),
5 send(Ventana,append,Sumar),
6 send(Ventana,open).
7 vsumar:-new(Vsumar, dialog('SUMA DE DOS NÚMEROS')),
8 new(Etisuma, label(etiqueta, 'SUMAREMOS DOS NÚMEROS')),
9 new(Caja1,text_item('Numero 1')),
10 new(Caja2,text_item('Numero 2')),
11 new(Caja3,text_item('Resultado')),
12 new(Boton,button('Sumar',message(@prolog,sumar,Caja1,Caja2,Caja3))),
13 send(Vsumar,append, Etisuma),
14 send(Vsumar,append, Caja1),
15 send(Vsumar,append, Caja2),
16 send(Vsumar,append,Boton),
17 send(Vsumar,append,Caja3),
18 send(Vsumar,open).
19
20
21 sumar(Caja1,Caja2,Caja3):-get(Caja1,value,Valor1),
22 get(Caja2,value,Valor2),
23 send(Caja3,value,Valor1+Valor2).
```

Una vez ejecutando el programa nos muestra la primera ventana en donde se encuentra el botón de sumar. Al presionar en el botón nos muestra la siguiente ventana en donde nos muestra la suma de dos números la cual contiene dos cajas de texto para poder ingresar el número 1 y el numero 2 la cual contiene también el botón de sumar y para verificar nos muestra el resultado como nos muestra a continuación.



## 7. Calcular la edad

En el siguiente ejercicio nos muestra el código para calcular la edad, así como también se agregan las etiquetas, las cajas de texto y el botón. Posteriormente se abren con la palabra de send. Finalmente, el programa con una resta, del primer valor que hace referencia al valor de la fecha actual, después el valor 2 con la de nacimiento.

```
Suma.pl | MenuGrafico.pl | CalcularEdad.pl

1 menu:-new(Ventanita, dialog('Edad')),
2 new(Opba, label(opba, 'Calcular edad ')),
3 new(Edad, button('Edad', message(@prolog, vedad))),
4 send(Ventanita, append, Opba),
5 send(Ventanita, append, Edad),
6 send(Ventanita, open).
7 vedad:-new(Vedad, dialog('EDAD')),
8 new(Etiedad, label(etiqueta, 'CALCULAR LA EDAD')),
9 new(Caja1, text_item('Año actual')),
10 new(Caja2, text_item('Año de nacimiento')),
11 new(Caja3, text_item('Resultado')),
12 new(Boton, button(' Calcular ', message(@prolog, edad, Caja1, Caja2, Caja3))),
13 send(Vedad, append, Etiedad),
14 send(Vedad, append, Caja1),
15 send(Vedad, append, Caja2),
16 send(Vedad, append, Boton),
17 send(Vedad, append, Caja3),
18 send(Vedad, open).
19
20 edad(Caja1, Caja2, Caja3):-get(Caja1, value, Valor1),
21                             get(Caja2, value, Valor2),
22                             send(Caja3, value, Valor1-Valor2).|
```

Una vez creado la interfaz con el código ejecutamos el programa ingresamos la variable menú. Y nos muestra la palabra true la cual significa que es verdadero y nos muestra la ventana en donde se va a calcular la edad la cual contiene un botón que nos muestra una siguiente ventana.

 SWI-Prolog (AMD64, Multi-threaded, version 7.5.0)

File Edit Settings Run Debug Help

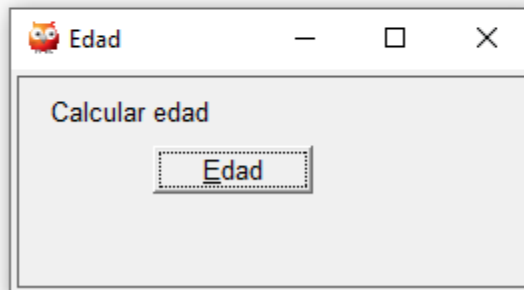
Welcome to SWI-Prolog (threaded, 64 bits, version 7.5.0-31-gd66b870)  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  
Please run `?- license.` for legal details.

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

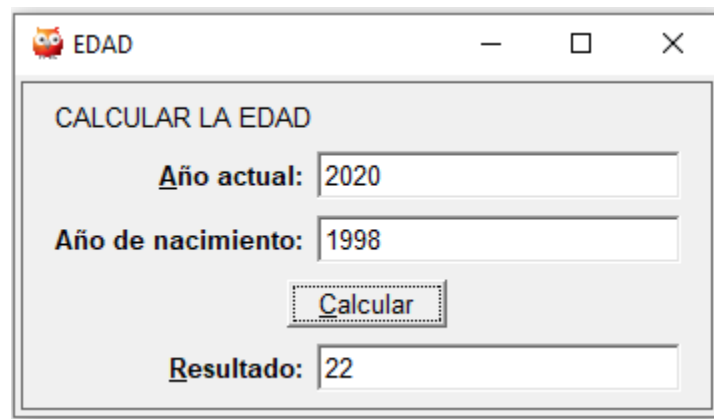
`?- menu.`

**true.**

`?-`



Una vez que hallamos abierto la primera ventana nos muestra la siguiente ventana la cual es calcular la edad la cual nos muestra tres etiquetas una es donde se agrega el año actual, año de nacimiento, botón de calcular y una vez ingresado los datos correspondientes nos muestra el resultado de la edad la cual deseamos saber.



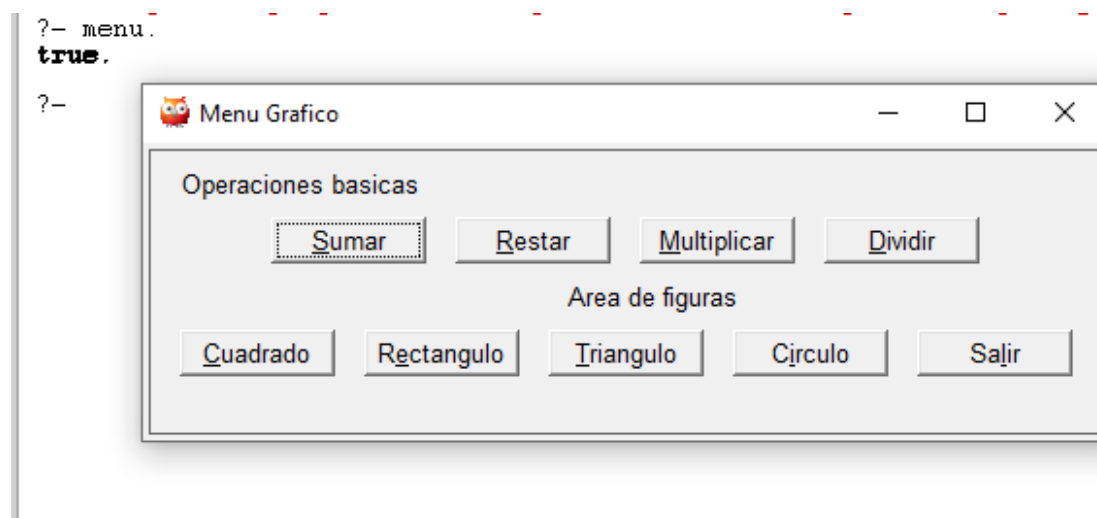
## 8. Menú grafico

En el siguiente ejercicio la cual se realiza el menú grafico de operaciones básicas (suma, multiplicación, dividir, resta). áreas de las figuras (cuadrado, rectángulo, triangulo, circulo, botón salir) para poder realizar el siguiente ejercicio se realiza a través de código.

```
Suma.pl | X MenuGrafico.pl | X CalcularEdad.pl |
1 menu:-new(Ventanita, dialog('Menu Grafico')),
2 new(Opba, label(opba, 'Operaciones basicas')),
3 send(Ventanita,append,Opba),
4 new(Sumar, button('Sumar',message(@prolog, vsumar))),
5 send(Ventanita,append,Sumar),
6 new(Restar, button('Restar',message(@prolog, vrestar))),
7 send(Ventanita,append,Restar),
8 new(Multiplicar, button('Multiplicar',message(@prolog, vmulti))),
9 send(Ventanita,append,Multiplicar),
10 new(Dividir, button('Dividir',message(@prolog, vdiv))),
11 send(Ventanita,append,Dividir),
12
13 new(AreFig, label(areafig, 'Area de figuras')),
14 send(Ventanita,append,AreFig),
15 send(Ventanita,append,Opba),
16 new(Cuadrado, button('Cuadrado',message(@prolog, vcuadrado))),
17 send(Ventanita,append,Cuadrado),
18 new(Rectangulo, button('Rectangulo',message(@prolog, vrectangulo))),
19 send(Ventanita,append,Rectangulo),
20 new(Triangulo, button('Triangulo',message(@prolog, vtriangulo))),
21 send(Ventanita,append,Triangulo),
22 new(Circulo, button('Circulo',message(@prolog, vcirculo))),
23 send(Ventanita,append,Circulo),
24 new(Boton,button('Salir',message(@prolog, cerrarVentana, Ventanita))),
25 send(Ventanita,append,Boton),
26 send(Ventanita,open).
27 cerrarVentana(Ventanita):-send(Ventanita,destroy).
28
```



Una vez terminado con la creación de ventana del menú grafico ejecutamos el programa la cual agregamos la variable menú. Y nos muestra la palabra true la cual significa que es verdadero y correcto. Primero se crea la ventana del menú, posteriormente la etiqueta de operaciones básicas con la palabra de new, y con el send se abren para visualizar, se crean las de sumar, restar, multiplicar, dividir. Posteriormente se crea la etiqueta de área de figuras, la de cuadrado, rectángulo, triángulo, círculo y un botón de salir.





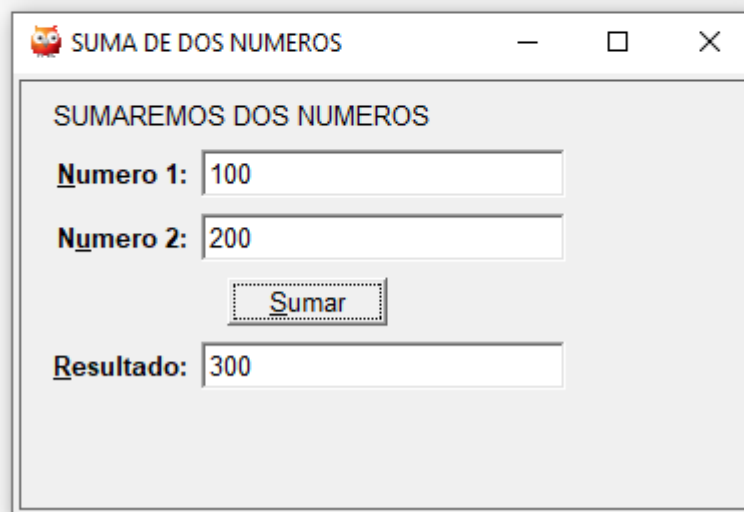
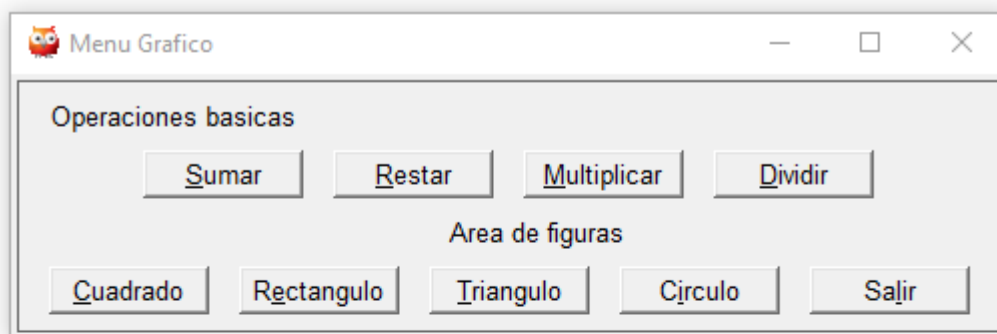
## Operaciones básicas

### ❖ Sumar

En el siguiente ejercicio nos muestra la suma de dos números. Primero se creamos la ventana, así como también lo que va a contener en el primer apartado nos muestra el código, después con el **send** nos permitir realizar la función dada. En la programación de funcionamiento se agrega la suma del valor uno con la suma del valor 2.

```
vsumar:-new(Vsumar, dialog('SUMA DE DOS NUMEROS')),  
new(Etisuma,label(etiqueta,'SUMA DE DOS NUMEROS')),  
new(Caja1, text_item('Numero 1:')),  
new(Caja2, text_item('Numero 2:')),  
new(Caja3, text_item('Resultado')),  
new(Boton, button('Sumar',message(@prolog,sumar,Caja1,Caja2,Caja3))),  
send(Vsumar, append, Etisuma),  
send(Vsumar, append, Caja1),  
send(Vsumar, append, Caja2),  
send(Vsumar, append, Boton),  
send(Vsumar, append, Caja3),  
send(Vsumar, open).  
sumar(Caja1,Caja2,Caja3):-get(Caja1,value,Valor1),  
get(Caja2,value,Valor2),  
send(Caja3,value,Valor1+Valor2).
```

Una vez ejecutando el programa en el menú grafico en el botón de operaciones básicas nos muestra en el primer botón, nos muestra la primera ventana en donde se encuentra el botón de sumar. Al presionar en el botón nos muestra la siguiente ventana en donde nos muestra la suma de dos números la cual contiene dos cajas de texto para poder ingresar el número 1 y el numero 2 la cual contiene también el botón de sumar y para verificar nos muestra el resultado como nos muestra a continuación.



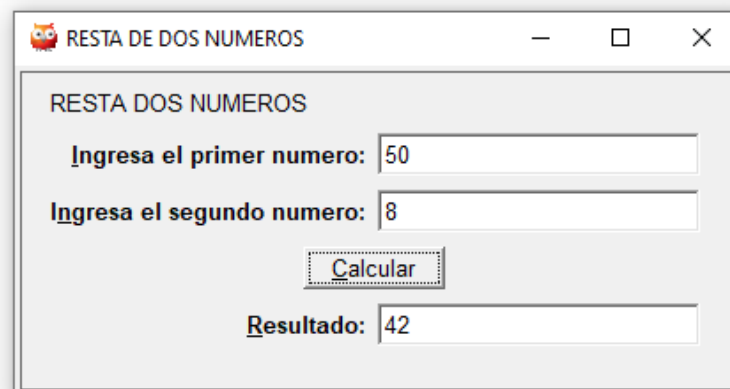
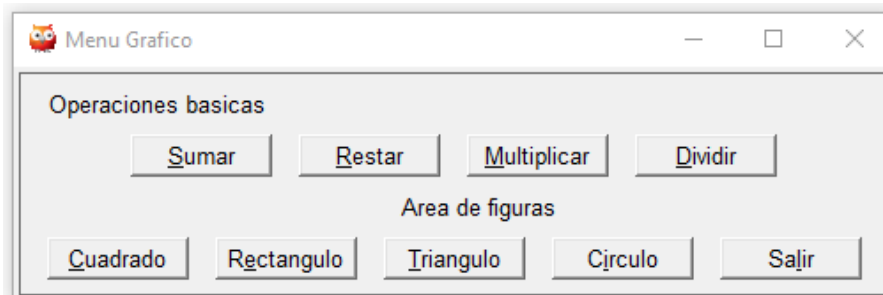
## ❖ Restar

En el botón número dos la cual es el botón de restar agregamos la etiqueta para restar dos números, ingresamos el primer y el segundo número para poder realizar la siguiente operación. La siguiente imagen muestra la interfaz de la resta. Se verifica que los datos son correctos.

```

1 vrestar:=new(Vrestar, dialog('RESTA DE DOS NUMEROS')),
2 new(Etirestar, label(etiqueta, 'RESTA DOS NUMEROS')),
3 new(CajaRes, text_item('Ingresa el primer numero')),
4 new(CajaRes2, text_item('Ingresa el segundo numero')),
5 new(CajaRes3, text_item('Resultado')),
6 new(Boton, button(' Calcular ', message(@prolog, restar, CajaRes, CajaRes2, CajaRes3))),
7 send(Vrestar, append, Etirestar),
8 send(Vrestar, append, CajaRes),
9 send(Vrestar, append, CajaRes2),
10 send(Vrestar, append, Boton),
11 send(Vrestar, append, CajaRes3),
12 send(Vrestar, open).
13 restar(CajaRes, CajaRes2, CajaRes3) :-get(CajaRes, value, Valor1),
14 get(CajaRes2, value, Valor2),
15 send(CajaRes3, value, Valor1-Valor2).

```

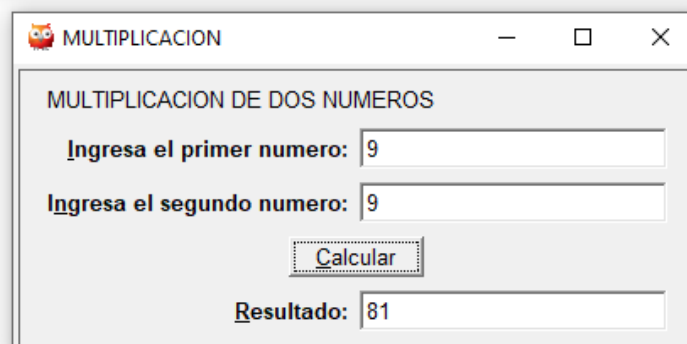
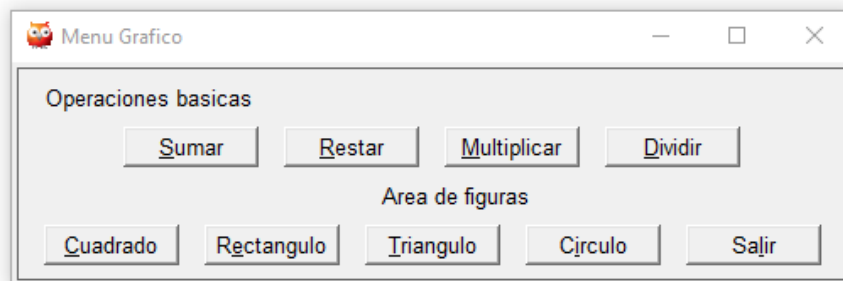


## ❖ Multiplicar

Para poder realizar el siguiente ejercicio la cual es en el botón de multiplicar debemos de realizarla dentro del menú grafico de las operaciones básicas, agregamos la etiqueta para poder ingresar el primer y segundo número y para finalizar nos debe de mostrar el resultado ya que por ello debemos ejecutar el programa una vez presionamos la opción de multiplicación, una vez ingresado los datos podemos calcular los datos de los números ingresados.

```

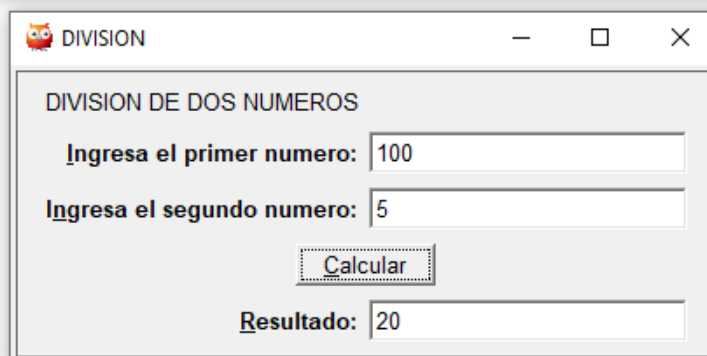
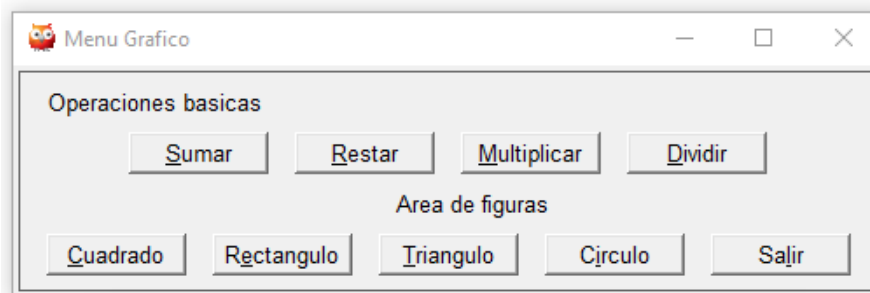
vmulti:--new(Vmulti, dialog('MULTIPLICACION')),
new(Etimulti, label(etiqueta, 'MULTIPLICACION DE DOS NUMEROS')),
new(Cajamulti, text_item('Ingresa el primer numero')),
new(Cajamulti2, text_item('Ingresa el segundo numero')),
new(Cajamulti3, text_item('Resultado')),
new(Boton, button(' Calcular ', message(@prolog, multiplicar, Cajamulti, Cajamulti2, Cajamulti3))),
send(Vmulti, append, Etimulti),
send(Vmulti, append, Cajamulti),
send(Vmulti, append, Cajamulti2),
send(Vmulti, append, Boton),
send(Vmulti, append, Cajamulti3),
send(Vmulti, open).
multiplicar(Cajamulti, Cajamulti2, Cajamulti3) :- get(Cajamulti, value, Valor1),
get(Cajamulti2, value, Valor2),
send(Cajamulti3, value, Valor1*Valor2).
  
```



## ❖ Dividir

En el siguiente botón la cual es sobre la división de dos numero para poder elaborar la siguiente operación agregamos, una caja de texto para poder ingresar el primer número, así como también agregamos otra caja de texto para poder ingresar el siguiente número, también se agregó un botón para poder calcular y al final se agregó una caja de texto para que nos muestre el resultado para poder realizar el siguiente ejercicio agregamos en el get y send ahí es en donde se agrega el código para que realice el ejercicio indicado sobre la división.

```
vdiv:-new(Vdiv, dialog('DIVISION')),
new(Etdiv, label(etiqueta, 'DIVISION DE DOS NUMEROS')),
new(Cajadiv, text_item('Ingresa el primer numero')),
new(Cajadiv2, text_item('Ingresa el segundo numero')),
new(Cajadiv3, text_item('Resultado')),
new(Boton, button(' Calcular ', message(@prolog, dividir, Cajadiv, Cajadiv2, Cajadiv3))),
send(Vdiv, append, Etdiv),
send(Vdiv, append, Cajadiv),
send(Vdiv, append, Cajadiv2),
send(Vdiv, append, Boton),
send(Vdiv, append, Cajadiv3),
send(Vdiv, open).
dividir(Cajadiv, Cajadiv2, Cajadiv3):-get(Cajadiv, value, Valor1),
get(Cajadiv2, value, Valor2),
send(Cajadiv3, value, Valor1/Valor2).
```

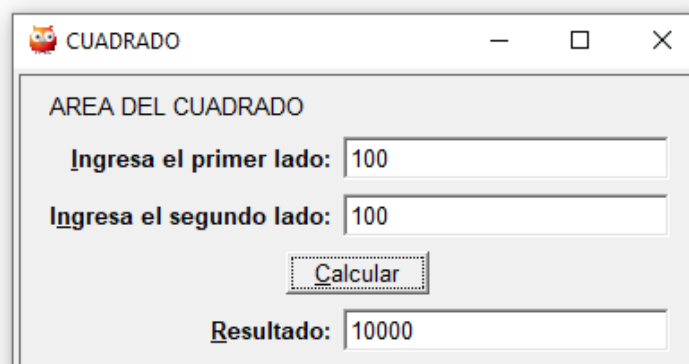
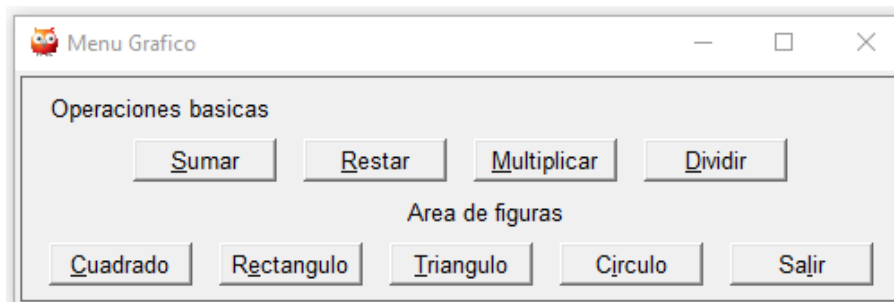


## Área de Figuras

### ❖ Cuadrado

En el área de figuras se realiza en el botón del cuadrado se necesita saber sobre el área de un cuadrado para poder realizar este ejercicio, se agregó la caja de textos del primer número del lado, así como también agregamos para poder ingresar el segundo número, y así poder calcular los datos ingresados por ello también se agregó una caja de texto la cual nos muestra el resultado de un área del cuadrado.

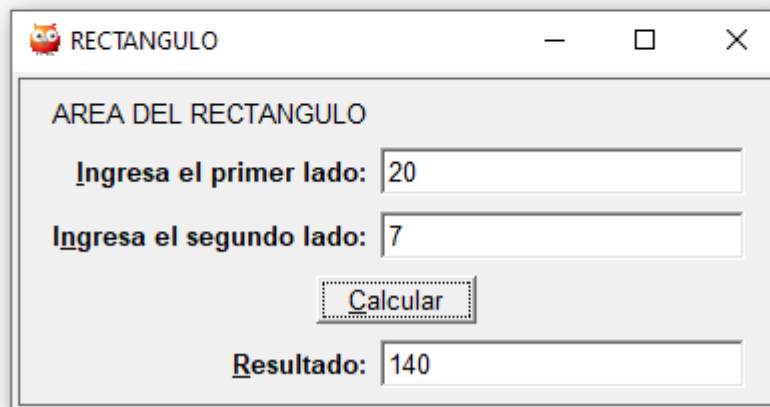
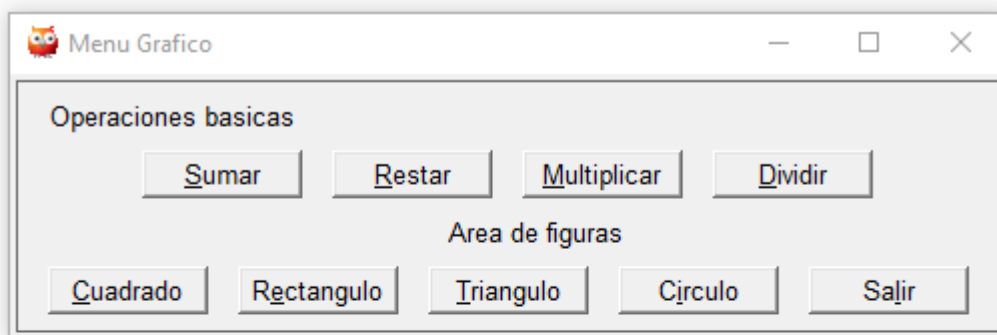
```
vcuadrado:-new(Vcuadrado, dialog('CUADRADO')),
new(Eticuadrado,label(etiqueta,'AREA DEL CUADRADO')),
new(Cajacua, text_item('Ingresa el primer lado')),
new(Cajacua2, text_item('Ingresa el segundo lado')),
new(Cajacua3, text_item('Resultado')),
new(Boton, button(' Calcular ',message(@prolog, cuadrado, Cajacua, Cajacua2, Cajacua3))),
send(Vcuadrado, append, Eticuadrado),
send(Vcuadrado, append, Cajacua),
send(Vcuadrado, append, Cajacua2),
send(Vcuadrado, append, Boton),
send(Vcuadrado, append, Cajacua3),
send(Vcuadrado, open).
cuadrado(Cajacua, Cajacua2, Cajacua3):-get(Cajacua,value,Valor1),
get(Cajacua2,value,Valor2),
send(Cajacua3,value,Valor1*Valor2).
```



## ❖ Rectángulo

Para poder realizar el siguiente botón la cual es el rectángulo se realiza en la programación de la interfaz gráfica el nombre de la etiqueta y para poder ingresar los datos, se agrega caja texto para poder el primer lado, así también se agrega un botón para poder calcular el resultado ya que para este ejercicio agregamos en el área del rectángulo es la multiplicación para poder saber el área de un rectángulo.

```
vrectangulo:-new(Vrectangulo, dialog('RECTANGULO')),
new(Etirectangulo,label(etiqueta,'AREA DEL RECTANGULO')),
new(Cajarec, text_item('Ingresa el primer lado')),
new(Cajarec2, text_item('Ingresa el segundo lado')),
new(Cajarec3, text_item('Resultado')),
new(Boton, button('Calcular ',message(@prolog,rectangulo,Cajarec,Cajarec2,Cajarec3))),
send(Vrectangulo, append, Etirectangulo),
send(Vrectangulo, append, Cajarec),
send(Vrectangulo, append, Cajarec2),
send(Vrectangulo, append, Boton),
send(Vrectangulo, append, Cajarec3),
send(Vrectangulo, open).
rectangulo(Cajarec,Cajarec2,Cajarec3):-get(Cajarec,value,Valor1),
get(Cajarec2,value,Valor2),
send(Cajarec3,value,Valor1*Valor2).
```





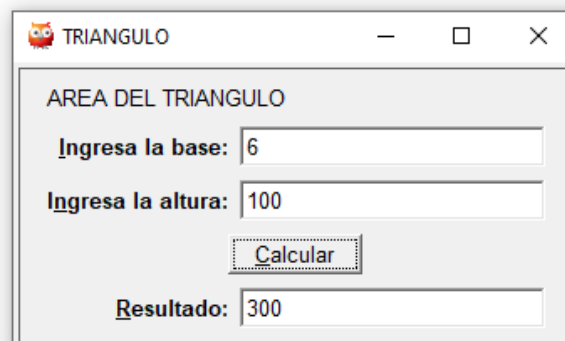
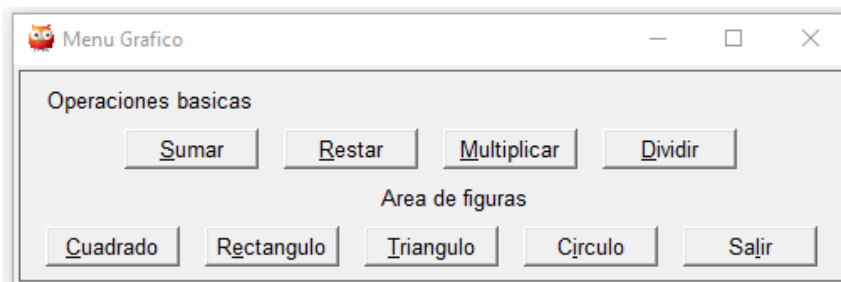
## ❖ Triangulo

Para poder realizar la interfaz del triángulo agregamos la etiqueta del nombre ,agregamos la caja de texto para poder ingresar el número de la base ,ya que también en el siguiente caja de texto también agregamos el número de altura y para poder calcular se agrega un botón y para verificar el resultado nos muestra en la caja de texto .Después se observa la venta del área del triángulo, esta se multiplican los dos datos de las cajas de texto, el resultado se divide entre 2, así como se indica en la fórmula del triángulo.

```

1 vtriangulo:-new(Vtriangulo, dialog('TRIANGULO')),
2 new(Etitriangulo,label(etiqueta,'AREA DEL TRIANGULO')),
3 new(Cajatri, text_item('Ingresa la base')),
4 new(Cajatri2, text_item('Ingresa la altura')),
5 new(Cajatri3, text_item('Resultado')),
6 new(Boton, button(' Calcular ',message(@prolog,triangulo,Cajatri,Cajatri2,Cajatri3))),
7 send(Vtriangulo, append, Etitriangulo),
8 send(Vtriangulo, append, Cajatri),
9 send(Vtriangulo, append, Cajatri2),
10 send(Vtriangulo, append, Boton),
11 send(Vtriangulo, append, Cajatri3),
12 send(Vtriangulo, open).
13 triangulo(Cajatri,Cajatri2,Cajatri3):-get(Cajatri,value,Valor1),
14 get(Cajatri2,value,Valor2),
15 send(Cajatri3,value,Valor1*Valor2/2).
16

```

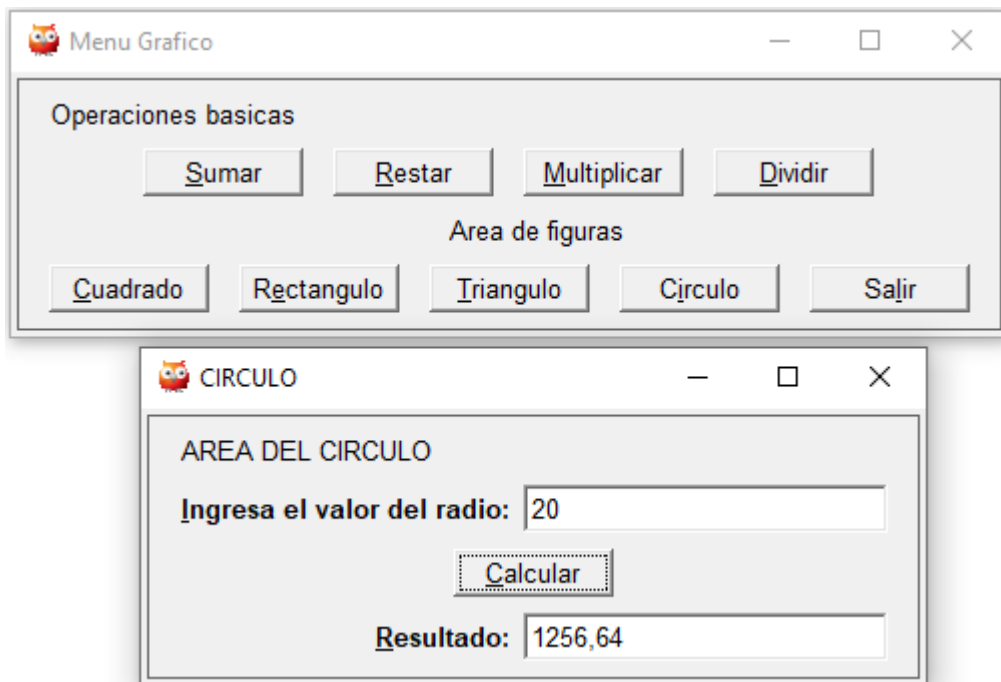




## ❖ Circulo

En el área de figura el ultimo botón es calcular el área del círculo para crear la siguiente interfaz se agrega el nombre de la etiqueta, así como también agregamos una caja de texto la cual es para ingresar el valor del radio, la cual solo presionamos la opción del botón de calcular y nos muestra el resultado en la caja de texto. Finalmente se muestra la interfaz para sacar el área del círculo, solo se agrega el dato del radio, debido a que la fórmula es  $\pi * r^2$  por lo tanto el radio se repite multiplicándose por su mismo y el valor de pi ya se establece por predeterminado en la codificación.

```
vcirculo:-new(Vcirculo, dialog('CIRCULO')),
new(Eticirculo,label(etiqueta,'AREA DEL CIRCULO')),
new(Cajacir, text_item('Ingresa el valor del radio')),
new(Cajacir2, text_item('Resultado')),
new(Boton, button(' Calcular ',message(@prolog,circulo,Cajacir,Cajacir2))),
send(Vcirculo, append, Eticirculo),
send(Vcirculo, append, Cajacir),
send(Vcirculo, append, Boton),|
send(Vcirculo, append, Cajacir2),
send(Vcirculo, open).
circulo(Cajacir,Cajacir2):-get(Cajacir,value,Valor1),
get(Cajacir,value,Valor1),
send(Cajacir2,value,Valor1*Valor1*3.1416).
```



## CONCLUSIÓN

se trabaja con un esquema orientado a objetos, donde podremos crear clases y trabajar con diferentes objetos, pudiendo llamar a métodos de dichos objetos (pasándoles los correspondientes parámetros) o llamar a métodos que nos devuelvan algún valor, y obviamente si creamos objetos, también podremos destruirlos.

Prolog usa principalmente dos tipos de referencias, una que sería mediante las variables típicas de prolog (una cadena que empiece por mayúscula, como Variable), y la otra forma es definir referencias con nombre, las cuales una vez definidas no podremos volver a crear otra referencia para dicho nombre, ya que nos dará error en ejecución. Es importante conocer como está conformado y de cómo utilizar el programa ya que es de mucha ayuda para el proceso de trabajo.