

Manual de Usuario

El siguiente manual es una guía completa para los usuarios del programa JPR 2021 que trata de un Compilador que ejecuta instrucciones con un lenguaje establecido mediante una gramática establecida y un conjunto de normas para que la ejecución y compilación sea mas practica para el usuario programador de JPR2021 en el lenguaje JPR Lenguaje 1.0

El Compilador puede ser descargado en el siguiente repositorio:

www.github.com/jamesg19/OCL2021

deberá ser clonado y ejecutado en cualquier sistema operativo Windows o Linux con Python instalado

A continuación

El compilador JPR cuenta con las siguientes opciones

Boton Archivo

Crear archivos: El editor deberá ser capaz de crear archivos en blanco.

Abrir archivos: El editor deberá abrir archivos .jpr

Guardar: El editor deberá guardar el estado del archivo en el que se estará trabajando.

Guardar Como: El editor deberá guardar el estado del archivo en el que se estará trabajando con un nuevo nombre a elegir.

Nuevo

Abrir

Guardar

Guardar como

Salir

```
5      contadorA = 0;
6      {
7      ackermann(3,4));
8      ECHO");
9  }
10
11 func ackermann(int m, int n){
12     contadorA = contadorA + 1;
13     print(contadora);
14     if (m == 0){
15         print(contadora);
16         return n + 1;
17     } else if (m > 0 && n == 0){
18         print(contadora);
19         return ackermann(m-1, 1)
20     } else{
21         print(contadora);
22         return ackermann(m-1, ackermann(m, n-1))
23     }
24 }
```

Boton Ejecutar

EJECUTAR : hará el llamado al intérprete, el cual se hará cargo de realizar los análisis léxico, sintáctico y semántico, además de ejecutar todas las sentencias.

Debugger: Característica que nos ayudará a ver el flujo de nuestro código al momento de ser ejecutado.

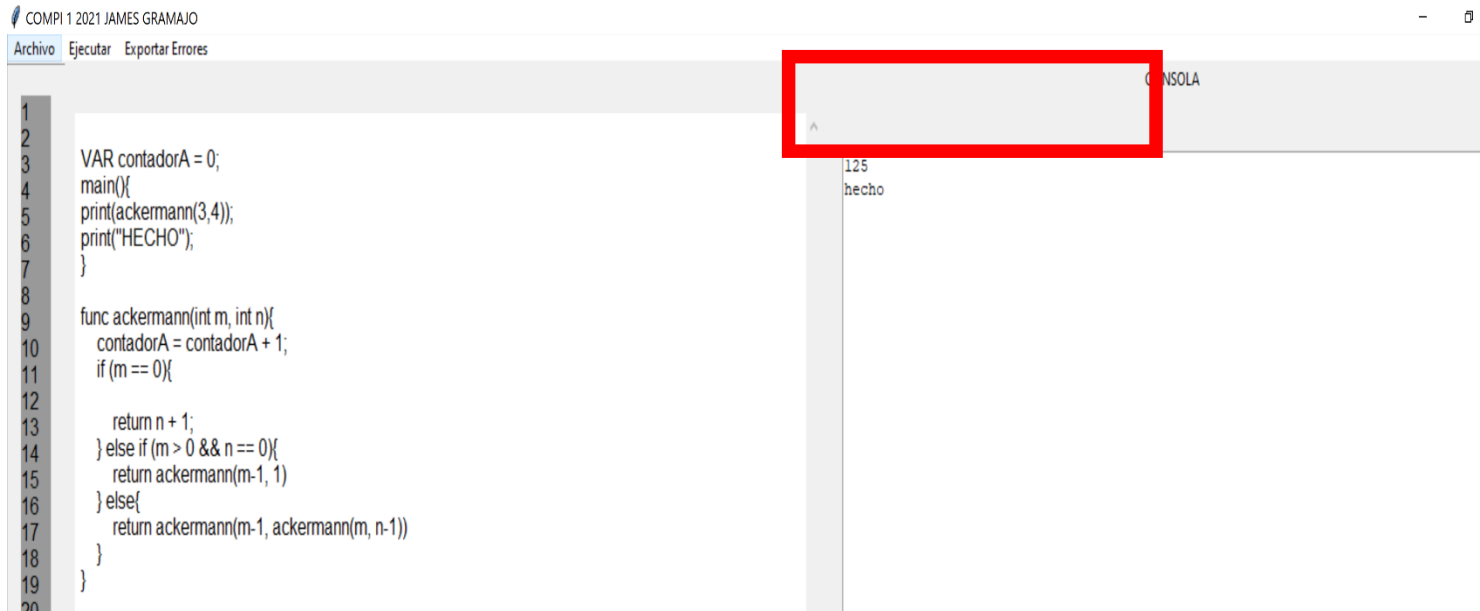
Para ejecutar el código



```
1
2
3  VAR contadorA = 0;
4  main(){
5    print(ackermann(3,4));
6    print("HECHO");
7  }
8
9  func ackermann(int m, int n){
10     contadorA = contadorA + 1;
11     print(contadora);
12     if (m == 0){
13       print(contadora);
14       return n + 1;
15     } else if (m > 0 && n == 0){
16       print(contadora);
17       return ackermann(m-1, 1)
18     } else{
19       print(contadora);
20       return ackermann(m-1, ackermann(m, n-1))
21     }
22   }
23
```

ERRORES: Cursor: 23,0

A continuación se muestra la salida en consola del Compilador JPR.



The screenshot shows the JPR compiler interface. The title bar reads 'COMPI 1 2021 JAMES GRAMAJO'. The menu bar includes 'Archivo', 'Ejecutar', and 'Exportar Errores'. The main window is split into two panes. The left pane displays source code for an Ackermann function, with line numbers 1 through 20 on the left margin. The code is as follows:

```
1  VAR contadorA = 0;
2
3  main(){
4      print(ackermann(3,4));
5      print("HECHO");
6  }
7
8
9  func ackermann(int m, int n){
10     contadorA = contadorA + 1;
11     if (m == 0){
12
13         return n + 1;
14     } else if (m > 0 && n == 0){
15         return ackermann(m-1, 1)
16     } else{
17         return ackermann(m-1, ackermann(m, n-1))
18     }
19 }
20
```

The right pane is titled 'CONSOLA' and shows the output of the program. A red rectangle highlights the top portion of the console output, which contains the number '125'. Below this, the text 'hecho' is visible.

En caso de tener un error se mostrarán en la parte inferior izquierda indicando el tipo de error.

```

1
2
3  VAR contadorA = 0;
4  main(){
5    print(ackermann(3,4));
6    print("HECHO");
7    print("james"~);
8    print(var1);
9  }
10
11 func ackermann(int m, int n){
12   contadorA = contadorA + 1;
13   if (m == 0){
14     return n + 1;
15   } else if (m > 0 && n == 0){
16     return ackermann(m-1, 1)
17   } else{
18     return ackermann(m-1, ackermann(m, n-1))
19   }
20 }

```

ERRORES:

Cursor: 7,12

Sintáctico - Error Sintáctico.) [5,15]
 Lexico - Error léxico.~ [6,14]
 Semantico - Variable (var1) no encontrada. [7,7]

Tabla de Simbolos


Después de ejecutar un código fuente ingresado por el usuario programador

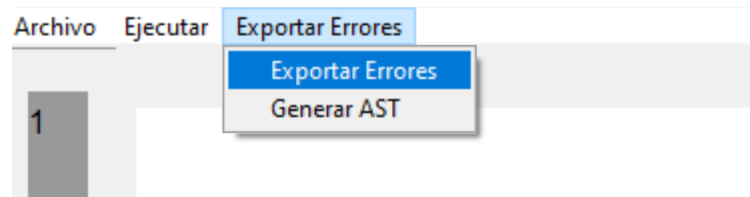
Este puede ver la tabla de símbolos donde puede visualizar con mayor contexto el valor de cada variable declarada y utilizada en el código también las funciones utilizadas.

Nº	ID	Tipo1	Tipo2	Valor	Línea	Columna
1	calculadora	- - -	FUNC	- - -	13	1
2	nombre	TIPO.CADENA	VAR	james	1	5
3	numero1	TIPO.ENTERO	VAR	50	14	9
4	numero2	TIPO.ENTERO	VAR	10	15	9
5	operacion	TIPO.CADENA	VAR	*	16	9

EXPORTAR REPORTE

Se puede visualizar con los reportes de errores en un archivo .html si el programador desea enviarlos a sus colegas para el análisis y mejoramiento del código.

 COMPI 1 2021 JAMES GRAMAJO



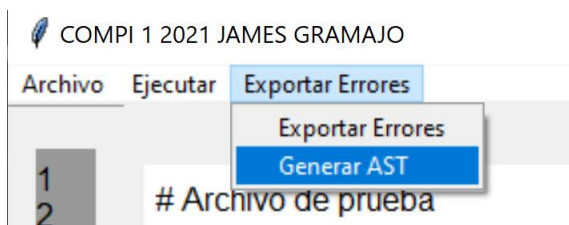
Los reportes seran visibles en un explorador o navegador de internet superior a 8.2.2 version GL o posterior.



	TIPO	DESCRIPCION	LINEA	COLUMNA					
		Semantico - Variable global_a ya existe	19	5					

EXPORTAR AST

Se podrá visualizar el AST generado con el código ingresado de la siguiente forma:



A continuación un ejemplo de un AST pequeno

