

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
ÁREA PROFESIONAL
Organización de Lenguajes y Compiladores 2



Manual de Usuario y Técnico

"IDE PASCAL"

Carné:	Nombre:
201430801	Luis Basilio Baquiaux Sic

Contenido

MANUAL DE USUARIO	3
Requisitos del sistema	3
¿Cómo usar esta aplicación IDE Pascal?	3
Ejecutar programa	3
Ventana principal.....	3
Abrir archivo	4
Nuevo archivo	6
Guardar archivo	6
Guardar archivo como	6
Cerrar archivo	6
Ejecutar.....	6
Reporte de errores.....	8
Tabla de símbolos	8
Tabla de tipos.....	9
MANUAL TÉCNICO	11
Carpeta necesaria	11
Lenguajes usados y tecnologías usadas	11
DIAGRAMAS	12
Diagrama de clases	12
Gramática:	12

MANUAL DE USUARIO

Requisitos del sistema

Para el funcionamiento del programa debe ser tener presente lo siguiente:

1. Tener instalado Java:

- a. Versión (recomendado): java 17.0.7 2023-04-18 LTS

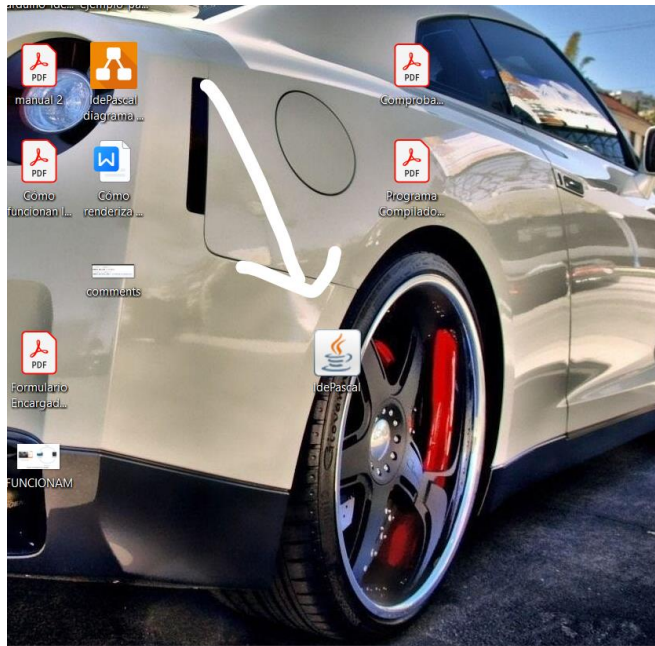
2. Memoria RAM:

- a. Tamaño (recomendado): 2Gb

¿Cómo usar esta aplicación IDE Pascal?

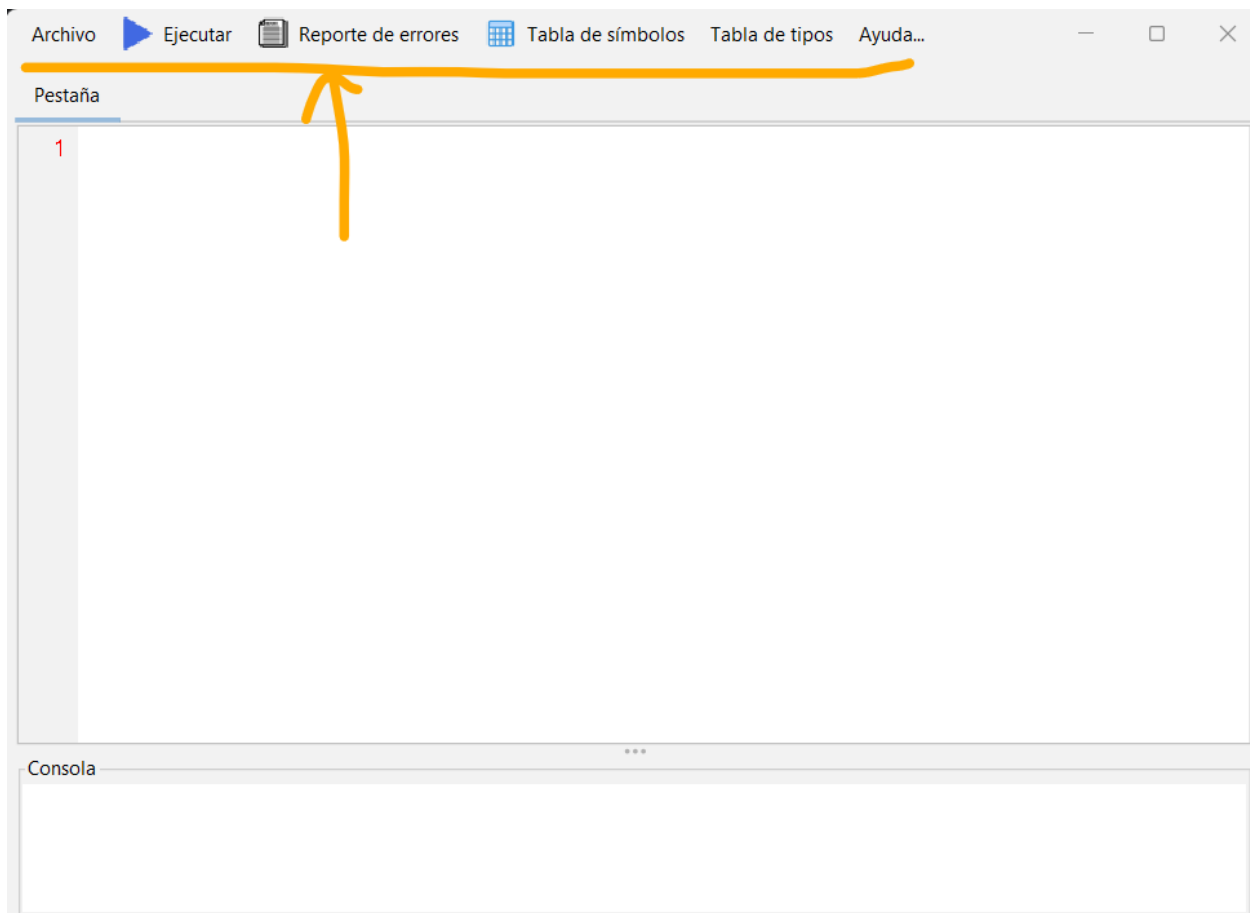
Ejecutar programa

Para ejecutar el programa debe hacer doble clic sobre el ejecutable de java.

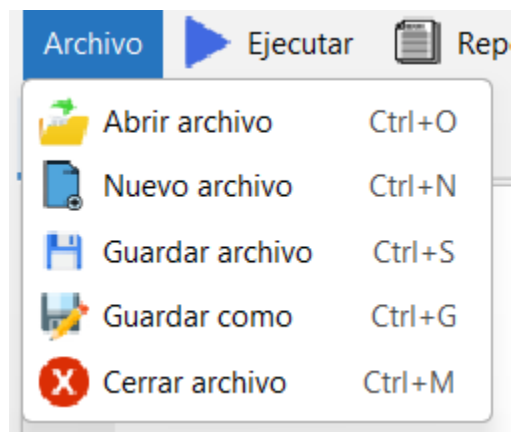


Ventana principal

La pantalla principal muestra las opciones de archivo, ejecutar, reporte de errores, tabla de símbolos, tabal de tipos, y ayuda. Además, tiene un espacio para escribir el código fuente, modificar. En la parte inferior (consola) hay un espacio donde se informará el estado de ejecución del archivo.

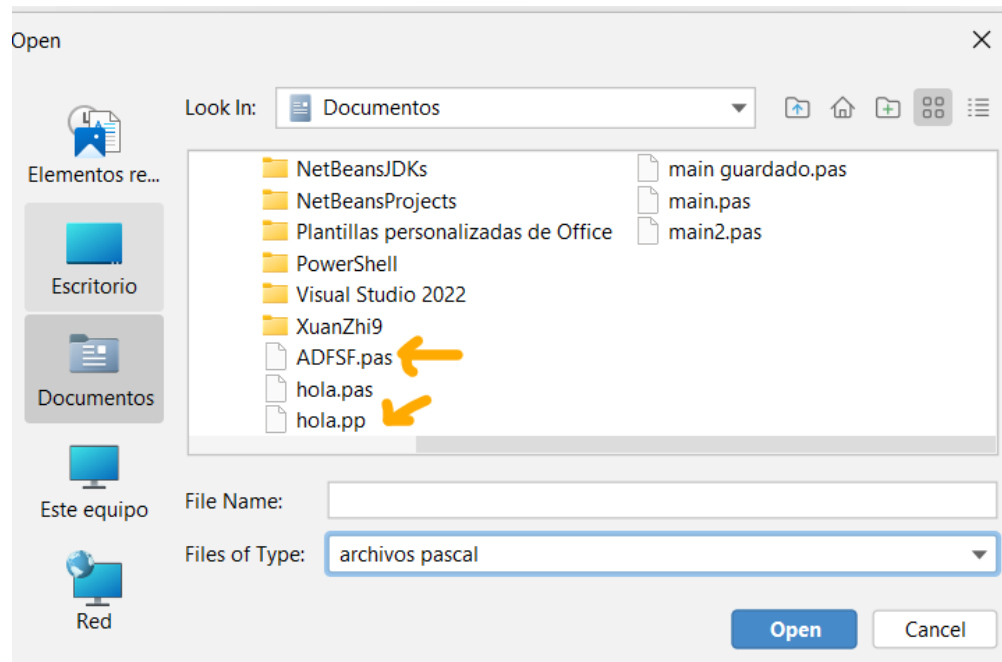


Opciones en el sub menú archivo:



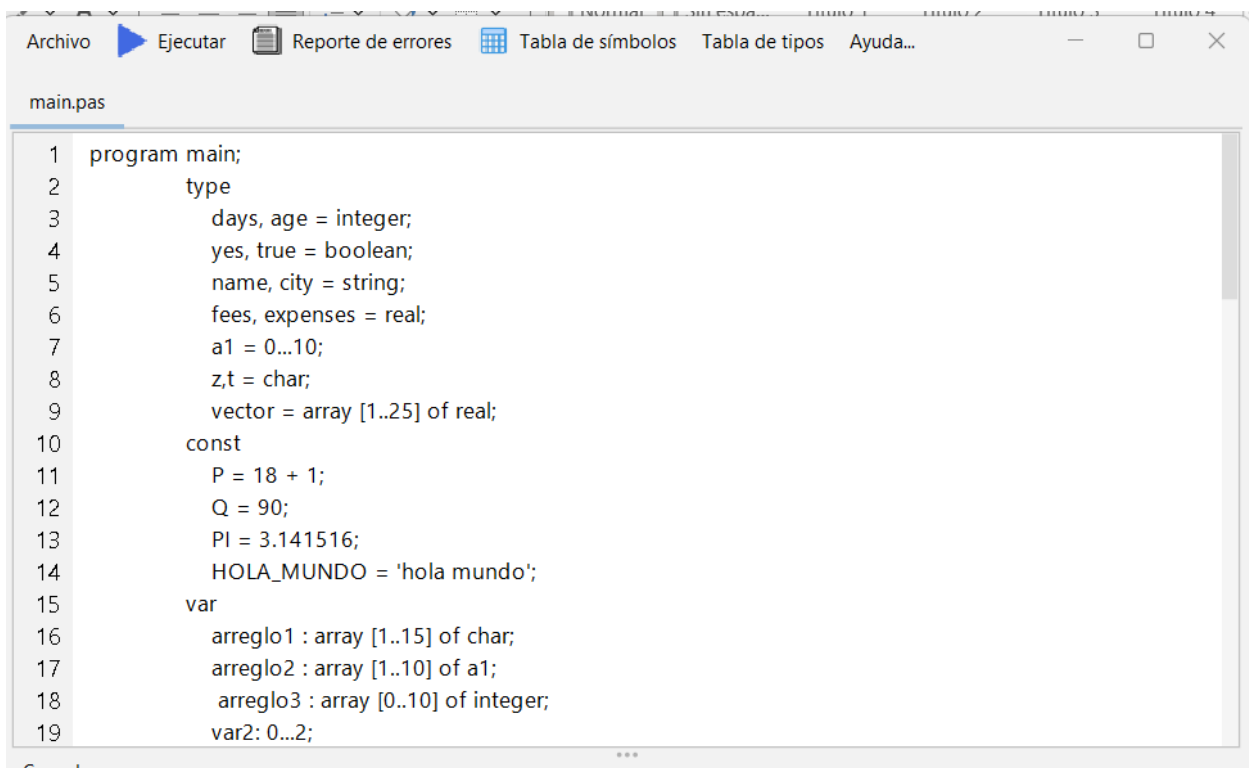
Abrir archivo

Abrirá un seleccionador de archivos. Seleccione el archivo que quiera abrir, los archivos que deben abrirse tendrán la extensión pp o pss.



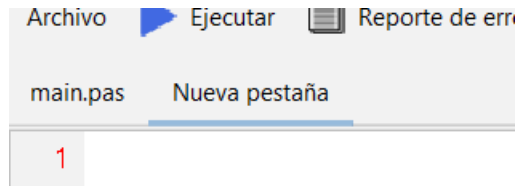
Esto mostrara el contenido del archivo pascal, listo para ejecutarse y ser analizado.

Comando: ctrl + o



Nuevo archivo

Abrirá una nueva pestaña para poder escribir código desde cero.



Comando: ctrl + n

Guardar archivo

Esta opción guarda el archivo abierto con los cambios que se han realizado en ella.

Comando: ctrl + s

Guardar archivo como

Abrirá un selector de archivos para guardar el archivo abierto con otro nombre y diferente ubicación.

Comando: ctrl + g

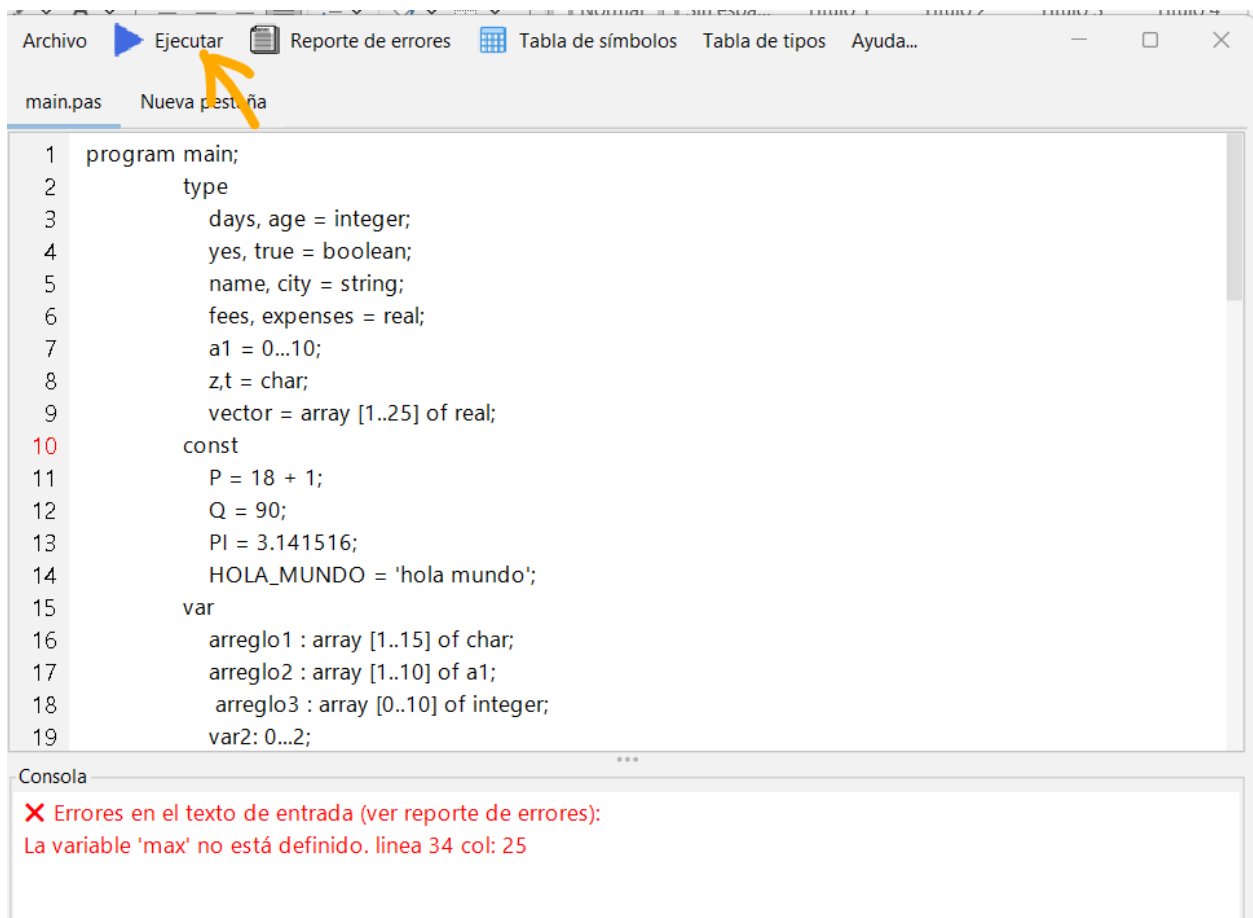
Cerrar archivo

Cierra el archivo abierto.

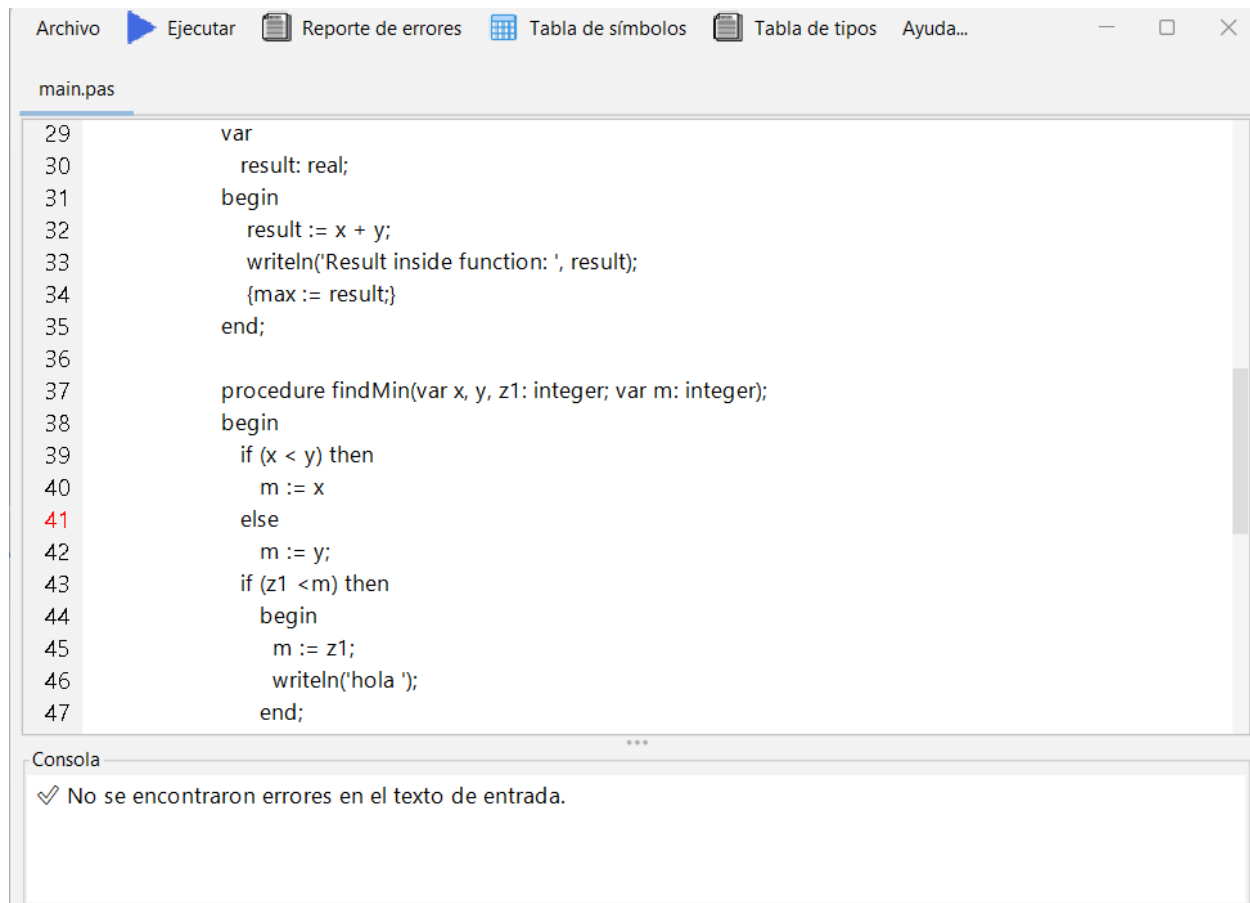
Comando: ctrl + m

Ejecutar

Analizará el archivo y además informará en consola si existe error durante la compilación.



Después de las correcciones del código se mostrará el mensaje de que todo está bien:



Reporte de errores

Esta opción mostrará más a detalle los errores mostrados en consola.

Reporte de Errores			
Todos los errores			
#	TIPO	DESCRIPCION	FILA - COLUMNA
1	SEMANTICO	La variable 'y' al que intentas acceder no está definido.	[41 - 32]
2	SEMANTICO	La variable 'm' al que intentas acceder no está definido.	[45 - 32]
3	SEMANTICO	La variable 'm' al que intentas acceder no está definido.	[50 - 44]
4	LEXICO	El carácter % no es aceptado en el lenguaje.	[37 - 12]
5	LEXICO	El carácter ^ no es aceptado en el lenguaje.	[37 - 14]
6	RECUPERABLE	No se esperaba el componente:) lin,col: 39 69	[39 - 69]
7	RECUPERABLE	No se esperaba el componente: ; lin,col: 51 24	[51 - 24]
8	IRECUPERABLE	No se esperaba el componente: null	[72 - 1]

Tabla de símbolos

Mostrará la tabla de símbolos generado después de la compilación, siempre que se hayan corregido los errores.

Tabla de símbolos					
ID	Categoría	Tipo	Parametros	Dirección	Ambito
a11	VARIABLE	SUBRANGO			GLOBAL
b	VARIABLE	BOOLEAN			GLOBAL
resultado	VARIABLE	BOOLEAN			GLOBAL
martes	VARIABLE	ENTERO			GLOBAL
i	VARIABLE	ENTERO			GLOBAL
arreglo1	VARIABLE	CARACTER			GLOBAL
arreglo2	VARIABLE	PERSONALIZA...			GLOBAL
arreglo3	VARIABLE	ENTERO			GLOBAL
p	CONSTANTE	ENTERO			GLOBAL
result	VARIABLE	REAL			Llamada: max
q	CONSTANTE	ENTERO			GLOBAL
grade	VARIABLE	CADENA			GLOBAL
num1	VARIABLE	ENTERO			Llamada: max
x	VARIABLE	REAL			Llamada: max
pi	CONSTANTE	REAL			GLOBAL
y	VARIABLE	REAL			Llamada: max
lunes	VARIABLE	ENTERO			GLOBAL
v1	VARIABLE	ARRAY			GLOBAL
var2	VARIABLE	ENTERO			GLOBAL
hola_mundo	CONSTANTE	CADENA			GLOBAL
num2	VARIABLE	ENTERO			Llamada: max
max	FUNCION	REAL	2		GLOBAL
num1	PARAMETRO	ENTERO			max

Tabla de tipos

Mostrará la tabla de tipos generado después de la compilación, siempre que se hayan corregidos los errores.

Tabla de tipos					
Nombre	Tipo	Tipo base	Dimension	Indice Min-Max	Ambito
a1	SUBRANGO	ENTERO	1	[0 - 10]	GLOBAL
fees	REAL	REAL	1	[0 - 0]	GLOBAL
t	CARACTER	CARACTER	1	[0 - 0]	GLOBAL
city	CADENA	CADENA	1	[0 - 0]	GLOBAL
yes	BOOLEAN	BOOLEAN	1	[0 - 0]	GLOBAL
true	BOOLEAN	BOOLEAN	1	[0 - 0]	GLOBAL
name	CADENA	CADENA	1	[0 - 0]	GLOBAL
days	ENTERO	ENTERO	1	[0 - 0]	GLOBAL
z	CARACTER	CARACTER	1	[0 - 0]	GLOBAL
vector	ARRAY	REAL	25	[1 - 25]	GLOBAL
age	ENTERO	ENTERO	1	[0 - 0]	GLOBAL
expenses	REAL	REAL	1	[0 - 0]	GLOBAL

MANUAL TÉCNICO

Carpeta necesaria

Se debe crear una carpeta llamada "árbol-llamadas" en el escritorio del usuario para el funcionamiento completo del programa.

En caso de que no se haya terminado de desarrollar la parte de generación del árbol de llamadas de funciones, esta carpeta no es necesaria crearla.

Lenguajes usados y tecnologías usadas

1. Java

a. Versión: java 17.0.7 2023-04-18 LTS

b. Link de descarga:

<https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>



2. Netbeans: NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

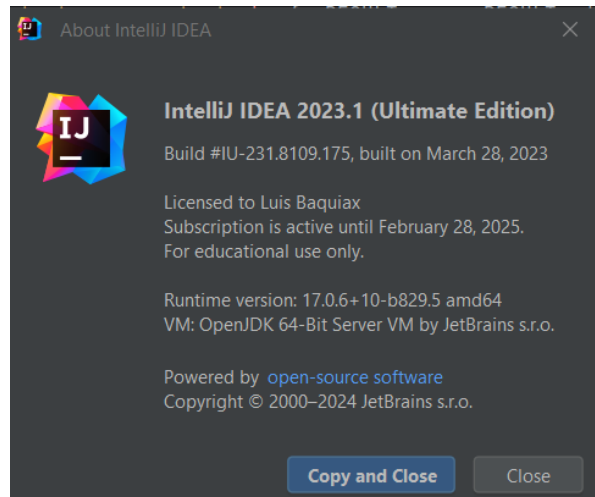
a. Versión: 21

b. Link de descarga:

<https://netbeans.apache.org/front/main/download/nb21/>

3. IntelliJ:

a. Link de descarga: <https://www.jetbrains.com/es-es/idea/download/?section=windows>



DIAGRAMAS

A continuación, se presenta los diagramas:

Diagrama de clases

En la carpeta de documentación se encuentra el diagrama de clases en formato png.

Gramática:

A continuación, se reescribe la gramática usada en cup en formato BNF.

```
<inicio> ::= "PROGRAM" IDENTIFICADOR ";" <components_program>

<components_program> ::= <components_program> <component_program>
                        | <component_program>

<component_program> ::= <staments> "BEGIN" <bloque_codigo> "END" "." ;

<staments> ::= <staments> <stament>
            | <stament>

<stament> ::= <declaracion_tipos>
            | <declaracion_contantes>
            | <declaracion_variable>
            | <funciones>

<funciones> ::= "FUNCTION" IDENTIFICADOR "(" <parametros> ")" ":" <data_type>
              ";" <declaracion_variable> "BEGIN" <bloque_codigo> "END" ";"
```

```

<procedimiento> ::= "PROCEDURE" IDENTIFICADOR "(" <parametros> ")" ";"
<declaracion_variable> "BEGIN" <bloque_codigo> "END" ";"

<parametros> ::= <parametros> "," <parametro>
                | <parametro> ;

<parametro> ::= "VAR" <ids_variables> ":" <data_type>
                | "VAR" <ids_variables> ":" IDENTIFICADOR
                | <ids_variables> ":" <data_type>
                | <ids_variables> ":" IDENTIFICADOR

<llamada_funcion> ::= IDENTIFICADOR "(" <list_params> ")"

<list_params> ::= <list_params> "," <expresion>
                | <expresion> ;

<bloque_codigo> ::= <bloque_codigo> <stament_bloque>
                | <stament_bloque>

<stament_bloque> ::= <un_stament> ";"

<un_stament> ::= <writeline>
                | <readline>
                | <asignacion>
                | <sentencia_if>
                | <setencia_case>
                | <sentencia_while>
                | <sentencia_for>
                | <repeat_until>
                | <llamada_funcion>

<declaracion_tipos> ::= "TYPE" <tipos>

<tipos> ::= <tipos> <tipo>
          | <tipo>

<tipo> ::= <ids_variables> "=" <data_type> ";"
          | <ids_variables> "=" <expresion> ".." <expresion> ";"
          | <ids_variables> "=" "ARRAY" "[" <expresion> ".." <expresion> "]"
          "OF" <data_type> ";"

<declaracion_contantes> ::= "CONST" <constantes>

<constantes> ::= <constantes> <constante>
                | <constante> ;

```

```

<constante> ::= IDENTIFICADOR "=" <expresion> ";"

<declaracion_variable> ::= "VAR" <variables>

<variables> ::= <variables> <variable>
               | <variable> ;

<variable> ::= <ids_variables> ":" <data_type> ";"
               | <ids_variables> ":" <data_type> "=" <expresion> ";"
               | <ids_variables> ":" IDENTIFICADOR ";"
               | <ids_variables> ":" <expresion> ".." <expresion> ";"
               | <ids_variables> ":" "ARRAY" "[" <expresion> ".." <expresion> "]"
"OF" <data_type> ";"
               | <ids_variables> ":" "ARRAY" "[" <expresion> ".." <expresion> "]"
"OF" IDENTIFICADOR ";" ;

<variable_record> ::= "VAR" IDENTIFICADOR ":" "RECORD" <atributos_record> "END"
";" ;

<atributos_record> ::= <atributos_record> <atributo_record>
                       | <atributo_record> ;

<atributo_record> ::= <ids_variables> ":" "PACKED" "ARRAY" "[" <expresion> ".."
<expresion> "]" "OF" <data_type> ";"
                       | <ids_variables> ":" <data_type> ";"
                       | <ids_variables> ":" IDENTIFICADOR ";" ;

<ids_variables> ::= <ids_variables> "," <id_variable>
                  | <id_variable>

<id_variable> ::= IDENTIFICADOR

<asignacion> ::= IDENTIFICADOR "=" <expresion>
                | IDENTIFICADOR "[" <expresion> "]" "=" <expresion>

<sentencia_if> ::= "IF" "(" <expresion> ")" "THEN" <un_stament>
                  | "IF" "(" <expresion> ")" "THEN" "BEGIN" <bloque_codigo>
"END"
                  | "IF" "(" <expresion> ")" "THEN" <un_stament> "ELSE"
<un_stament>
                  | "IF" "(" <expresion> ")" "THEN" "BEGIN" <bloque_codigo>
"END" "ELSE" <un_stament>
                  | "IF" "(" <expresion> ")" "THEN" "BEGIN" <bloque_codigo>
"END" "ELSE" "BEGIN" <bloque_codigo> "END"

```

```

<setencia_case> ::= "CASE" "(" <expresion> ")" "OF" <casos> "ELSE"
<stament_bloque> "END"
                | "CASE" "(" <expresion> ")" "OF" <casos> "END"

<casos> ::= <casos> <caso>
          | <caso>

<caso> ::= <expresion> ":" <stament_bloque>
          | <expresion> ":" "BEGIN" <bloque_codigo> "END" ";"

<sentencia_while> ::= "WHILE" "(" <expresion> ")" "DO" "BEGIN" <bloque_codigo>
"END"

<sentencia_for> ::= "FOR" <asignacion> "TO" <expresion> "DO" "BEGIN"
<bloque_codigo> "END"

<repeat_until> ::= "REPEAT" <bloque_codigo> "UNTIL" <expresion>

<writeline> ::= "WRITELN" "(" <parametros_write> ")"

<readline> ::= "READLN" "(" <parametros_write> ")"

<parametros_write> ::= <parametros_write> "," <expresion>
                    | <expresion>

<expresion> ::= <expresion> "+" <expresion>
                | <expresion> "-" <expresion>
                | <expresion> "*" <expresion>
                | <expresion> "/" <expresion>
                | <expresion> "DIV" <expresion>
                | <expresion> "MOD" <expresion>
                | <expresion> "=" <expresion>
                | <expresion> "<" <expresion>
                | <expresion> ">" <expresion>
                | <expresion> "<" <expresion>
                | <expresion> ">=" <expresion>
                | <expresion> "<=" <expresion>
                | <expresion> "AND" <expresion>
                | <expresion> "OR" <expresion>
                | <expresion> "AND THEN" <expresion>
                | <expresion> "OR ELSE" <expresion>
                | "(" <expresion> ")"
                | "NOT" <expresion>
                | "-" <expresion>

```

```
        | <valor>

<valor> ::= ENTERO
        | DECIMAL
        | CADENA
        | "TRUE"
        | "FALSE"
        | IDENTIFICADOR
        | IDENTIFICADOR "[" <expresion> "]"

<data_type> ::= "INTEGER"
        | "REAL"
        | "BOOLEAN"
        | "STRING"
        | "TKCHAR"
```