PROYECTO FINAL

SISTEMAS DE PROGRAMACIÓN II

RESPONSABLE:
Génesis Anahí Mejía Díaz
Matrícula: 19191838
8°-ISC

índice

Base de datos

```
Use Master
Go

Create DataBase BD_Sistema
Go

Use BD_Sistema
Go

Create Table Usuario
(
	id			int			Identity Primary Key,
	usuario		varchar(8)	Not Null,
	contrasena	varchar(8)	Not Null,
	idAcceso	int			Not Null,
)

/*
idAcceso		Acceso
*****************************
	1		Administrador
	2		Supervisor
	3		Vendedor
*****************************
*/

Insert Into Usuario Values ('jmendez','123abc',1)
Insert Into Usuario Values ('rgonzalo','abc123',2)
Insert Into Usuario Values ('mtorres','def562',3)
Insert Into Usuario Values ('ljimenez','123ghi',3)


Select * From Usuario

Create table lenguaje (
idlenguaje char (2) primary key,
nombrelenguaje Varchar (max),
Fenchaentrada Varchar (max),
hora char (5))
```

```
Create table registro(
idregistro char (2) primary key,
cod_usuario char (10) foreign key reference usuario(cod_idusuario),
idlenguaje char (2) foreign key reference lenguaje (idlenguaje),
fechasalida smalldatetime)
```

Diccionario de datos

Variable	Nombre	Descripcion	Tipo	Longitud	
Id	Identificador	Este es para saber el objeto que se esta consultando	char	2	
Nombre	Identificador	esto es para saber el nombre de las personas	varchar	10	
ApellidoP	Identificador	Este es para saber el apellido de las personas	varchar	10	
ApellidoM	Identificador	Apellido de las personas	vacrhar	10	
Lenguaje	El lenguaje	Este es el lenguaje que se va a usar para consultar las palabras	varchar	10	
Fecha	La fecha	La fecha a la hora que se consulto el lenguaje	Smalldatetime	10	
Hora	Hora de entrada y salida	y la hora de entrada y salida del lenguaje	datetime	4	

Modelo Relacional

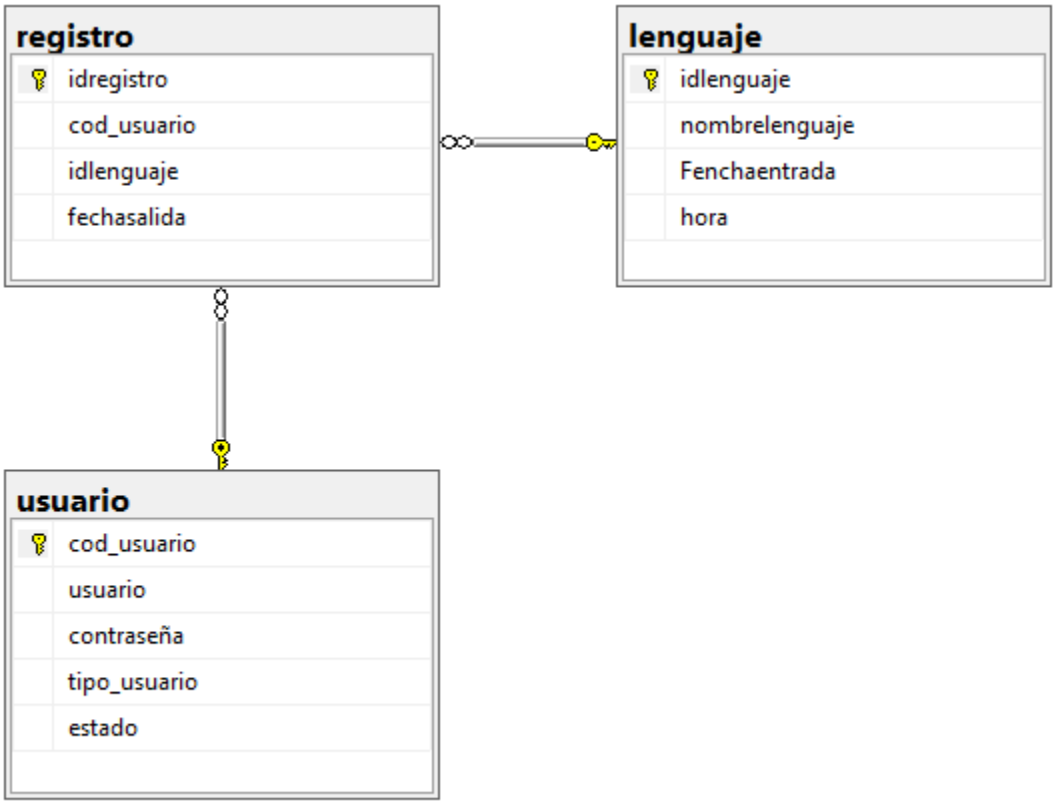
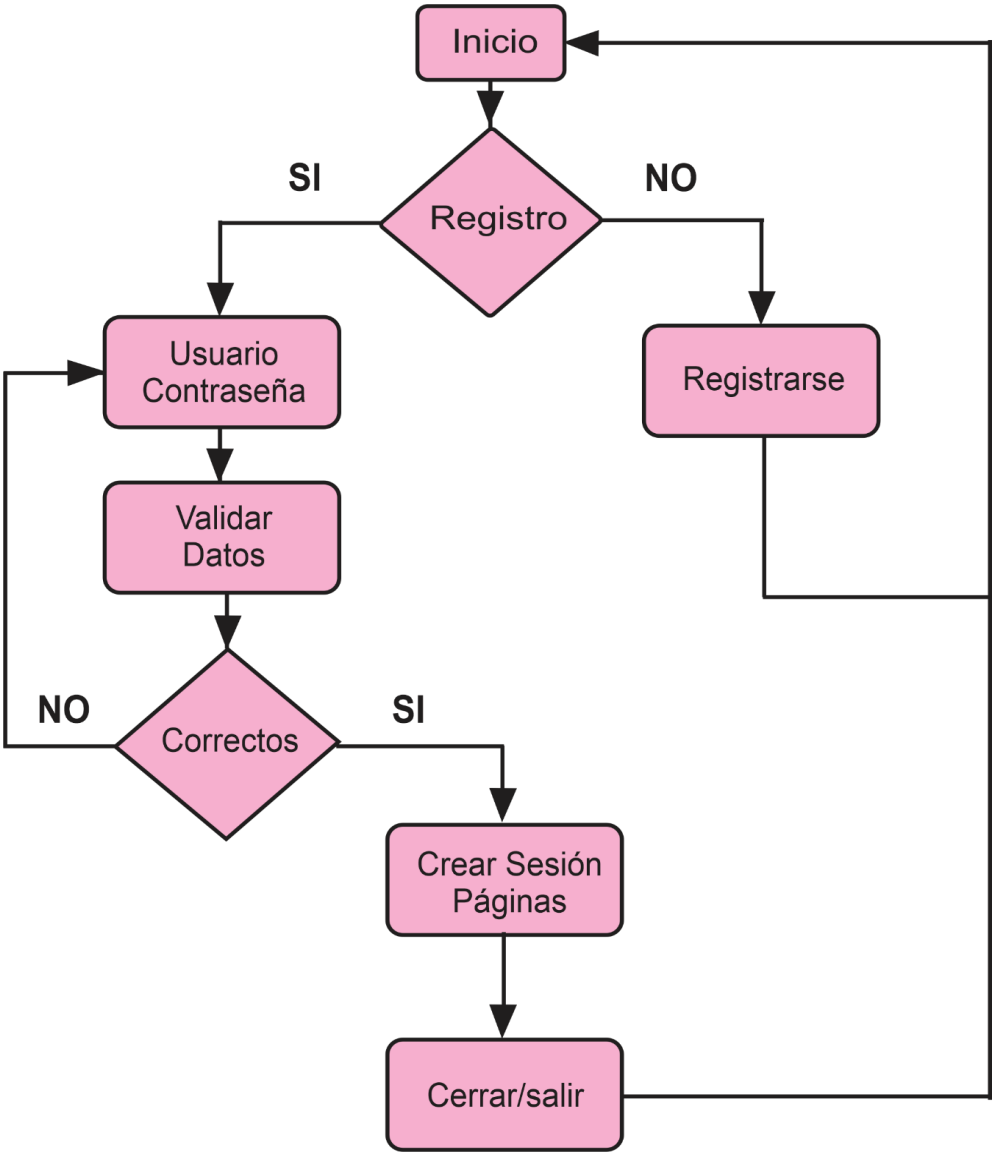
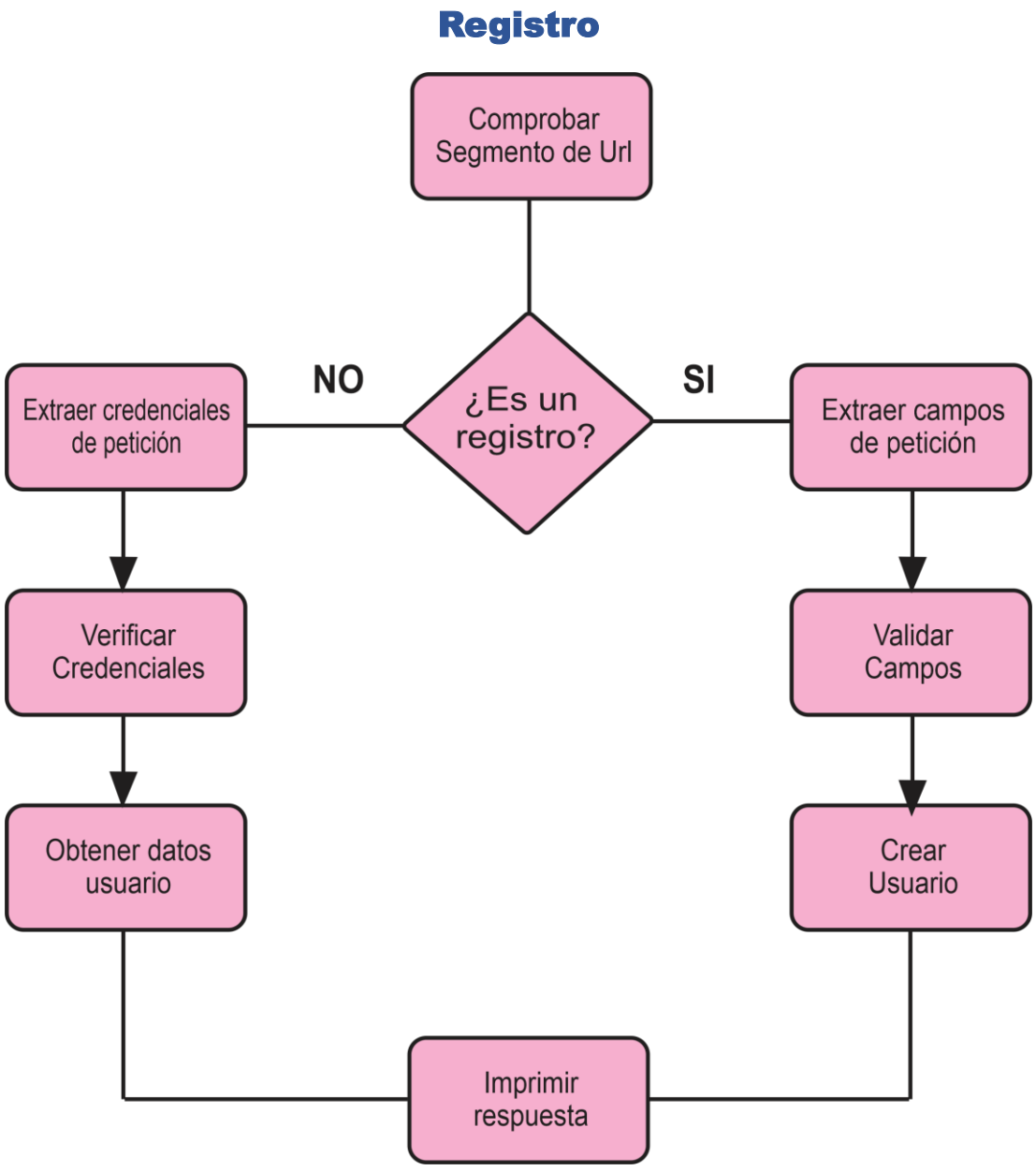


Diagrama de flujo



Un diagrama de flujo de inicio de sesión de usuario proporciona una representación visual clara y ordenada de los pasos necesarios para que un usuario acceda al sistema. Es útil para identificar posibles puntos de falla o mejorar la experiencia del usuario al optimizar el flujo de inicio de sesión.



Un diagrama de creación de usuario ayuda a comprender el flujo y los pasos necesarios para que un usuario pueda registrarse en un sistema o plataforma. Es útil para identificar posibles problemas en el proceso, mejorar la experiencia del usuario y garantizar la recopilación correcta de datos necesarios para el funcionamiento del sistema.

Compilador

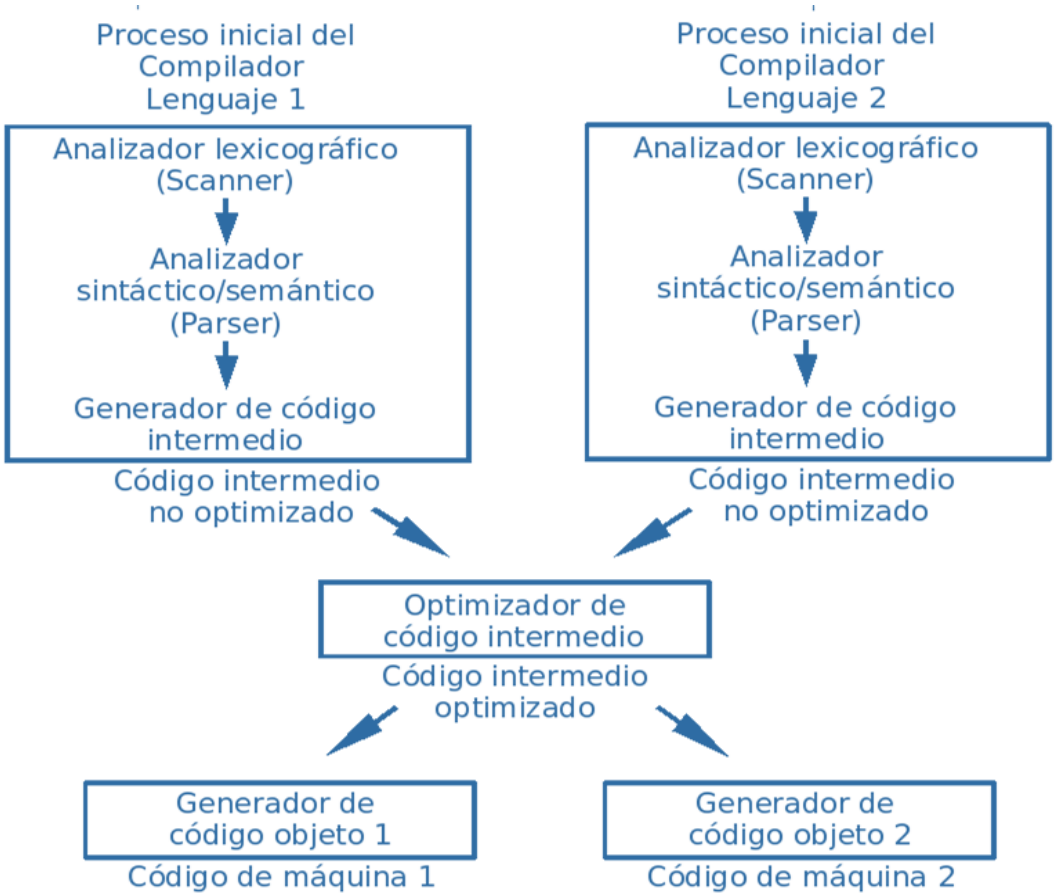
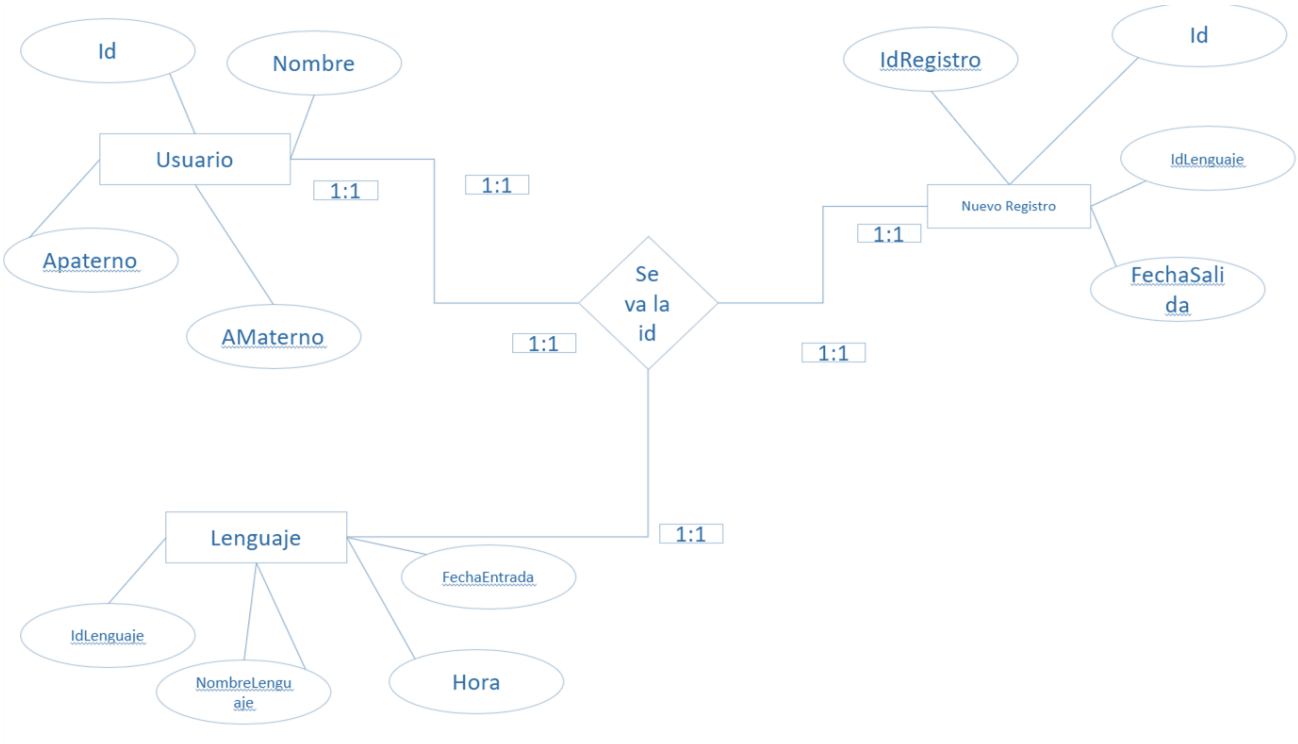

Proceso inicial del Compilador Lenguaje 1
Analizador lexicográfico (Scanner)
Analizador sintáctico/semántico (Parser)
Generador de código intermedio
Código intermedio no optimizado
Proceso inicial del Compilador Lenguaje 2
Analizador lexicográfico (Scanner)
Analizador sintáctico/semántico (Parser)
Generador de código intermedio
Código intermedio no optimizado
Optimizador de código intermedio
Código intermedio optimizado
Generador de código objeto 1
Código de máquina 1
Generador de código objeto 2
Código de máquina 2

Diagrama Entidad-Relación



El diagrama entidad-relación sirve para visualizar y diseñar la estructura de una base de datos, comunicar de manera efectiva el diseño a los miembros del equipo, analizar los requisitos y las dependencias del sistema, y optimizar el rendimiento de las operaciones en la base de datos.

Desempeña un papel fundamental en el diseño de bases de datos y la programación. Ayuda a crear una estructura coherente y eficiente para la base de datos, facilita la comunicación entre los miembros del equipo, identifica requisitos y oportunidades de optimización, y guía la programación de la aplicación que interactúa con la base de datos.

Matrices

Visual Basic

```
1,3,4,6,104,105,106,107,108,7,8,9,300,0,114,115
1,1,100,100,100,100,100,100,100,100,100,100,2,100,100,100
1,1,300,300,300,300,300,300,300,300,300,300,300,300,300,300
301,3,4,101,101,101,101,101,101,101,101,101,300,101,101,101
302,5,302,302,302,302,302,302,302,302,302,302,302,302,302,302
302,5,102,102,102,102,102,102,102,102,102,102,102,102,102,102
6,6,6,103,6,6,6,6,6,6,6,6,6,6,6,6
7,7,7,7,7,7,7,7,7,109,7,7,7,7,7,7
110,110,110,110,110,110,110,110,110,110,110,110,110,110,110,110
113,113,113,113,113,113,113,113,113,113,113,113,113,113,113,113
```

Cobol

```
1,3,4,6,7,121,106,107,108,109,110,111,112,113,114,115,10,11,122,300
1,1,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,2,100
1,1,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301
101,3,4,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101
302,5,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302
102,5,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102
6,6,6,103,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6
104,104,104,104,104,8,104,104,104,104,104,104,104,104,104,104,104,104,104,104
8,8,8,8,8,9,8,8,8,8,8,8,8,8,8,8,8,8,8,8
303,303,303,303,105,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303
117,117,117,117,117,117,117,117,117,117,117,117,117,117,116,117,118,117,117
120,120,120,120,120,120,120,120,120,120,120,120,120,120,119,120,120,120,120
```

C++

```
0,0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 0,10
1,0 1,1 1,2 1,3 1,4 1,5 1,6 1,7 1,8 1,9 1,10
2,0 2,1 2,2 2,3 2,4 2,5 2,6 2,7 2,8 2,9 2,10
3,0 3,1 3,2 3,3 3,4 3,5 3,6 3,7 3,8 3,9 3,10
4,0 4,1 4,2 4,3 4,4 4,5 4,6 4,7 4,8 4,9 4,10
5,0 5,1 5,2 5,3 5,4 5,5 5,6 5,7 5,8 5,9 5,10
6,0 6,1 6,2 6,3 6,4 6,5 6,6 6,7 6,8 6,9 6,10
7,0 7,1 7,2 7,3 7,4 7,5 7,6 7,7 7,8 7,9 7,10
8,0 8,1 8,2 8,3 8,4 8,5 8,6 8,7 8,8 8,9 8,10
9,0 9,1 9,2 9,3 9,4 9,5 9,6 9,7 9,8 9,9 9,10
10,0 10,1 10,2 10,3 10,4 10,5 10, 6 10,7 10,8 10,9 10,10
```

Fortran

```
1 1 1 101 302 102 104 7 7 108 10 11 115 118
300 2 300 101 302 102 104 7 7 308 10 11 315 317
3 1 1 3 5 5 104 7 7 108 10 11 115 118
4 100 300 4 112 102 104 7 7 308 10 11 315 317
0 100 300 101 302 102 104 7 7 308 10 11 315 317
6 100 300 101 302 102 104 7 103 308 10 11 315 317
105 100 300 101 302 102 7 8 7 308 10 11 315 317
106 100 300 101 302 102 104 7 7 308 10 11 315 317
107 100 300 101 302 102 104 7 7 308 10 11 315 317
9 100 300 101 302 102 104 7 7 109 10 11 116 117
110 100 300 101 302 102 104 7 7 308 10 11 315 317
111 100 300 101 302 102 104 7 7 308 10 11 315 317
10 100 300 101  302 102 104 7 7 308 113 11 315 317
11 100 300 101  302 102 104 7 7 308 10 114 315 317
12 100 300 101  302 102 104 7 7 308 10 11 315 317
13 100 300 101  302 102 104 7 7 308 10 11 315 317
119 100 300 101 302 102 104 7 7 308 10 11 315 317
120 100 300 101 302 102 104 7 7 308 10 11 315 317
121 100 300 101 302 102 104 7 7 308 10 11 315 317
122 100 300 101 302 102 104 7 7 308 10 11 315 317
124 100 300 101 302 102 104 7 7 308 10 11 315 317
123 100 300 101 302 102 104 7 7 308 10 11 315 317
```

Java

```
1,2,1,3,5,7,9,109,107,108,110,111,112,113,114,115,116,117,118,119,120,121,122,0
1,1,1,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
101,2,101,3,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101
302,4,101,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302
302,4,101,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102
6,6,6,6,303,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6
303,303,303,303,103,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303
8,8,8,8,8,304,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8
8,8,8,8,8,104,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8
106,106,106,106,106,106,106,10,106,106,106,106,106,106,106,106,106,106,106,106,106,106,106,106
10,10,10,10,10,10,10,11,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10
10,10,10,10,10,10,105,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10
```

Pascal

```
0,1,1,3,4,300,105,106,10,11,300,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122
100,1,2,1,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
302,1,302,1,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302
101,303,303,3,4,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101
304,304,304,5,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304,304
102,305,305,5,305,6,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102
306,306,306,306,306,306,7,8,306,306,306,306,306,306,306,306,306,306,306,306,306,306,306,306,306,306
308,308,308,9,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308,308
309,309,309,9,309,309,102,102,309,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102
10,10,10,10,10,10,10,10,103,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10
11,11,11,11,11,11,11,11,11,104,11,11,11,11,11,11,11,11,11,11,11,11,11,11
312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,312,121,312,312
313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,313,122,123,313
```

Python

```
1,3,0,300,4,6,7,105,106,107, 108, 8, 9, 10
1,1,100,2,100,100,100,100,100,100, 100, 100, 100, 100
1,1,300,300,300,300,300,300,300,300, 300, 300, 300, 300
301,3,101,101,4,101,101,101,101,101, 101, 101, 101, 101
302,5,302,302,302,302,302,302,302,302, 302, 302, 302, 302
303,5,102,102,102,102,102,102,102,102, 102, 102, 102, 102
6,6,6,6,6,103,6,6,6,6, 6, 6, 6, 6
7,7,7,7,7,7,104,7,7,7, 7, 7, 7, 7
109,109,109,109,109,109,109,109,109,109, 109, 109, 109, 110
112,112,112,112,112,112,112,112,112,112, 112, 111, 112, 113
304, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304,304,304,114
```

Foxpro

```
0,1,3,4,6,7,103,106,107,108,109,110,111,112,113,114,115,116,300
1,1,1,100,100,100,100,100,100,100,100,100,100,100,100,100,100,2,100
2,1,1,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301,301
3,101,3,4,101,101,101,101,101,101,101,101,101,101,101,101,101,101,101
4,302,5,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302,302
5,102,5,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102,102
6,6,6,6,103,6,6,6,6,6,6,6,6,6,6,6,6,6,6
7,7,7,7,7,104,7,7,7,7,7,7,7,7,7,7,7,7,7
8,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303,303
```

Clipper

```
1, 3, 0, 4, 6, 7, 10, 11, 107, 105, 106, 114, 115, 116, 117, 118, 119, 120, 121, 300
1, 1, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 2
1, 1, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300, 300
101, 3, 101, 4, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101, 101
302, 5, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302, 302
102, 5, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102, 102
6, 6, 6, 6, 103, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6
108, 108, 108, 108, 108, 108, 108, 108, 8, 108, 108, 108, 108, 108, 108, 108, 108, 108, 108, 108
8, 8, 8, 8, 8, 8, 8, 8, 9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
304, 304, 304, 304, 304, 104, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304, 304
111, 111, 111, 111, 111, 111, 111, 109, 111, 111, 111, 111, 111, 111, 111, 111, 111, 111, 110, 111
113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 113, 112, 113
```

DBase

13	12	4	5	102	103	104	105	1	3	112	110	2	113	10	114	115	116	117	7	8	9	301	0	302
108	108	108	108	108	108	108	107	108	106	108	108	108	108	108	108	108	108	108	108	108	108	108	108	108
300	300	300	300	300	300	300	300	300	300	300	300	111	300	300	300	300	300	300	300	300	300	300	300	300
119	119	119	119	119	119	119	118	119	119	119	119	119	119	119	119	119	119	119	119	119	119	119	119	119
4	4	100	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
109	109	109	6	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109
6	6	6	15	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
303	303	303	303	303	303	303	303	303	303	303	303	303	303	303	303	303	303	606	120	303	303	303	303	303
304	304	304	304	304	304	304	121	304	304	304	304	304	304	304	304	304	304	304	304	304	304	304	304	304
122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	122	123	122	122	122
305	11	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305	305
124	11	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124	124
125	12	125	125	125	125	125	125	125	125	125	125	125	125	10	125	125	125	125	125	125	125	125	125	125
13	13	126	126	126	126	126	126	126	126	126	126	126	126	126	126	126	126	126	126	126	14	126	126	126
13	13	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306	306
6	6	6	101	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Visual Basic

```
save
list
load
run
dim
list
tablas
byte
int
word
float
print
input
tab
locate
goto
if
for
sub
screen
pset
line
draw
paint
get
put
beep
sound
play
random
```

Cobolt

ACCEPT
ADD
AND
BY
COMPUTE
DISPLAY
DIVIDE
ELSE
FROM
GIVING
GO
IF
MOVE
MULTIPLY
NOT
OR
PERFORM
PICTURE
PROMPT
SET
SUBTRACT
THRU
TIMES
TO
UNTIL
VALUE
VARYNG

C++

```
alignas
alignof
and
and_eq
asm
atomic_cancel
atomic_commit
atomic_noexcept
auto
bitand
bitor
bool
break
case
catch
char
char8_t
char16_t
char32_t
class
compl
concept o
const
consteval
constexpr
const_cast
continue
co_await
co_return
co_yield
decltype
default
```

Fortran

```
Allocatable
Allocate
Case
Contains
Cycle
Deallocate
Elsewhere
Exit
Include
Interface
Intent
Module
Namelist
Nullify
Only
Operator
Optional
Pointer
Private
Procedure
Public
Recursive
Result
Select
Sequence
Target
Use
While
Where
Elemental
```

Java

break,
case,
catch,
continue,
default,
delete,
do,
else,
finally,
for,
function,
if,
in,
instanceof,
new,
return,
switch,
this,
throw,
try,
typeof,
var,
void,
while,
with.

Pascal

```
Absolute
Asm
Circle
Downto
External
Forward
If
Label
Not
Procedure
Record
String
Type
Uses
While
And
Begin
Const
Else
File
Function
Implementation
Mad
Of
Program
Rectangle
Then
Unity
Var
With
Array
Case
```

Python

```
And
As
Assert
Async
Await
Break
Class
Continue
Def
Del
Elif
Else
Except
Finally
For
From
Global
If
Import
In
Is
Lambda
Nonlocal
Not
Or
Pass
Raise
Return
Try
While
With
Yield
```

```
For
End
While
Seek
Loop
Dim
Skip
With
Scan
If
Do
Class
Else
Void
Break
Catch
Return
Int
Double
Native
=
<
>
+
-
;
/
\
(
```

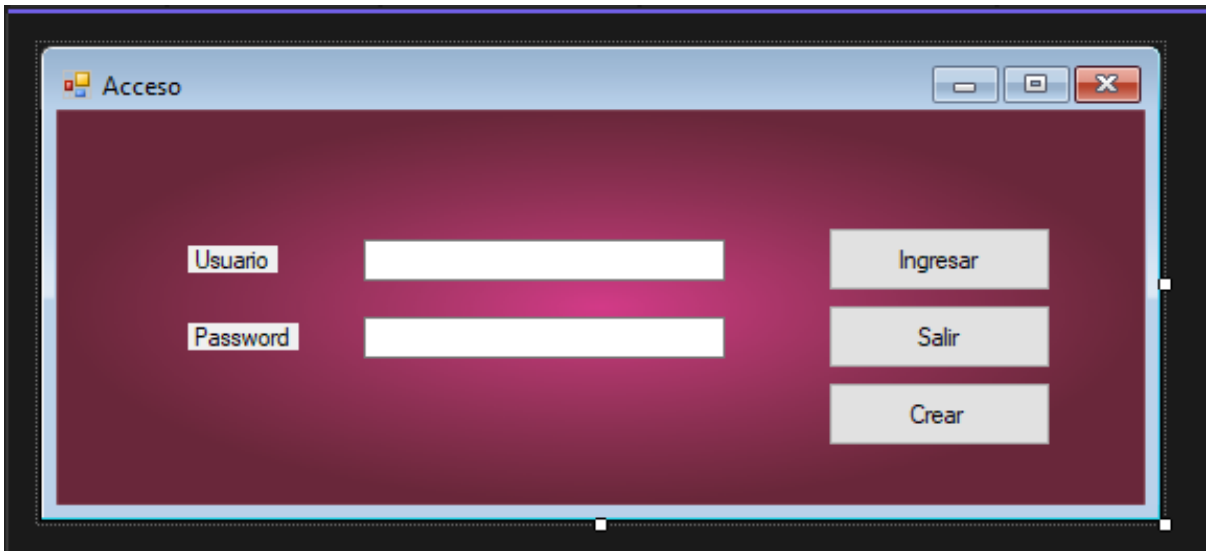
Clipper

```
IF          |
AADD
CTOD
EXP
INT
MONTH
ROW
TIME
ELSE        |
ABS
DATE
FCOUNT
LASTREC
PCOL
RTRIM
TRANSFORM
ELSEIF      |
ASC
DAY
FIELDNAME
LEN
PCOUNT
SECONDS
TRIM
ENDIF       |
AT
DELETED
FILE
LOCK
PROW
SELECT
TYPE
```

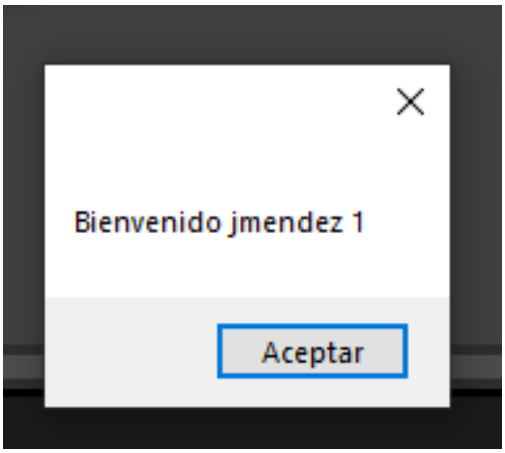
Dbase

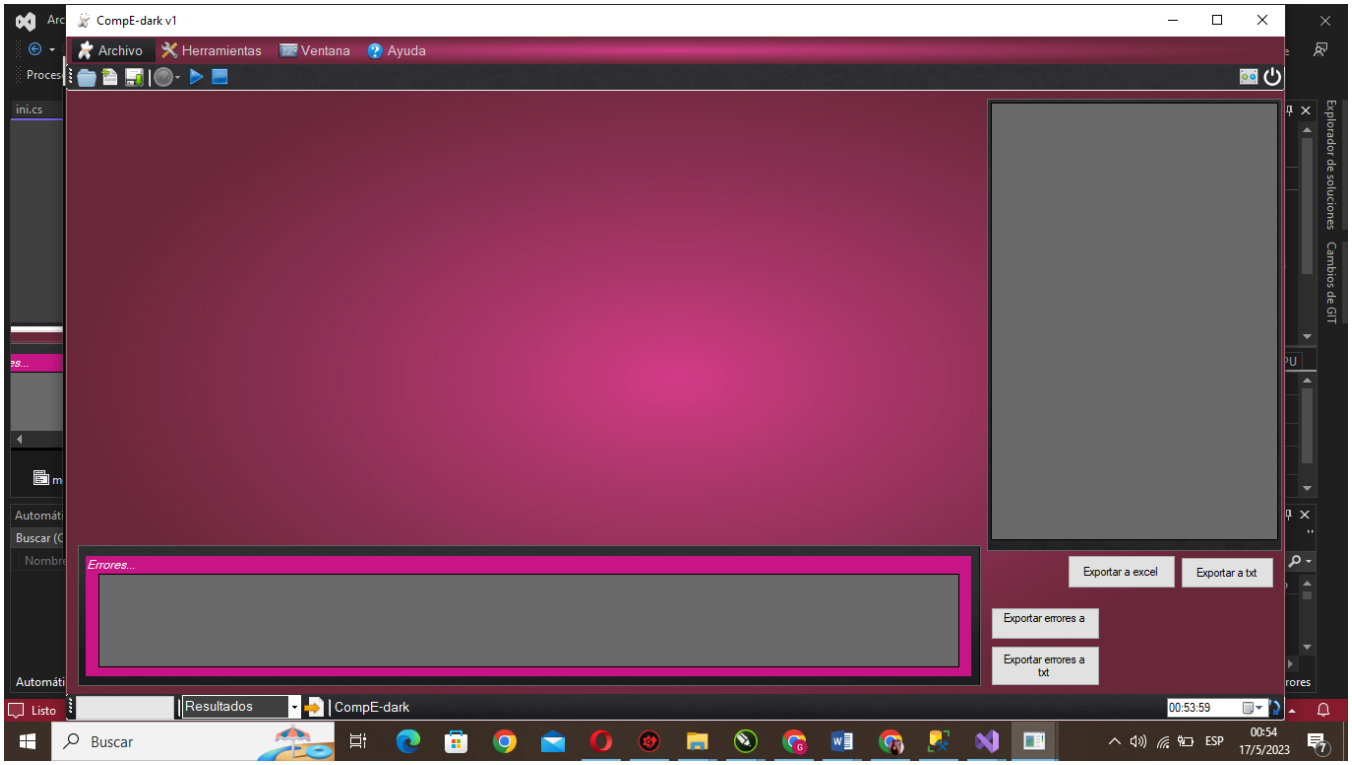
```
FROM,
ARRAY,
BEGIN,
TRANSACTION,
END,
CANCEL,
CHANGE,
CLEAR,
ALL,
CLOSE,
FORMAT,
COMPILE,
COPY,
CREATE,
MODIFY,
DIR,
EDIT,
ERASE,
EXPORT,
HELP,
IMPORT,
INSERT,
KEY,
SAVE,
UPDATE,
INT,
DO,
STATUS,
PAUSE,
GETS,
READ,
DEBUG,
FLOAT,
SELECT,
SQRT,
STRING,
WHILE
```

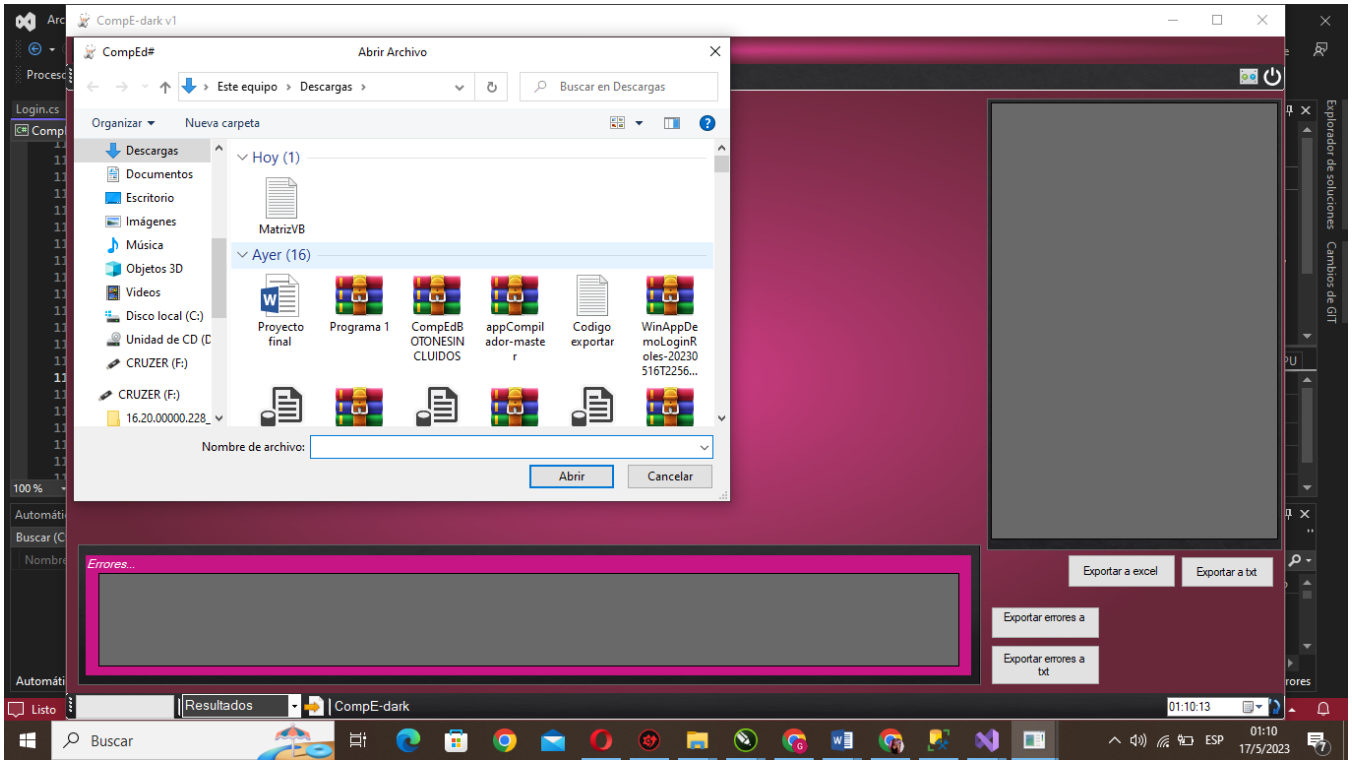
Programa

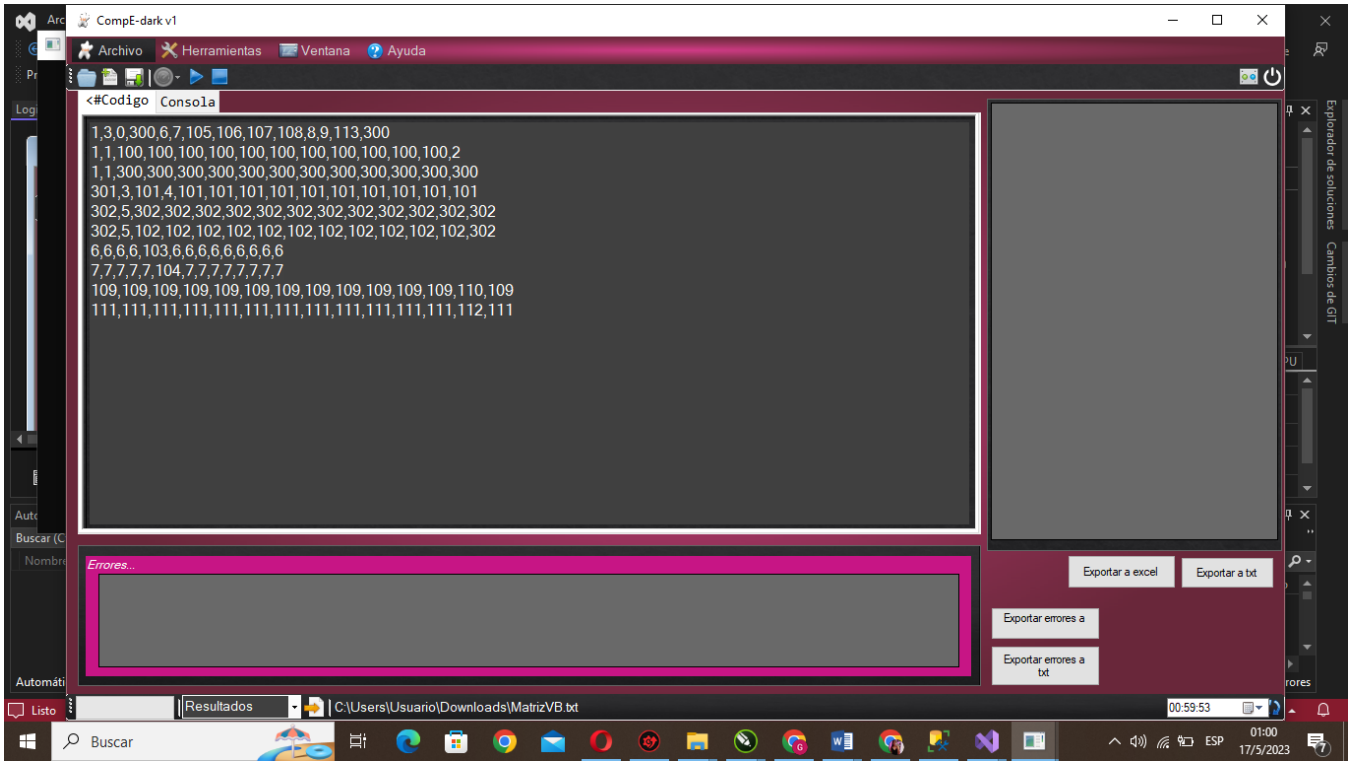

Acceso
Usuario
Password
Ingresar
Salir
Crear

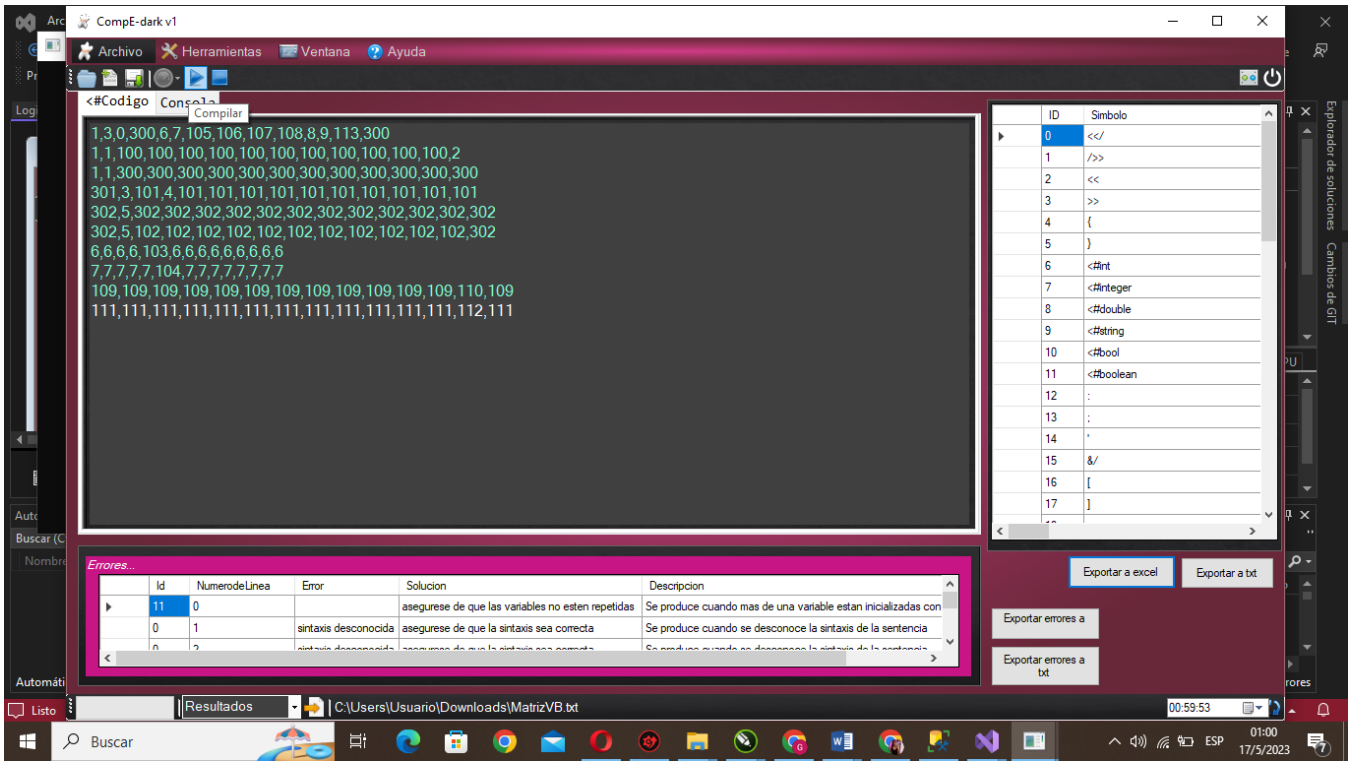

COMPILADOR
DE ARCHIVOS
2023
Iniciando sistema ...

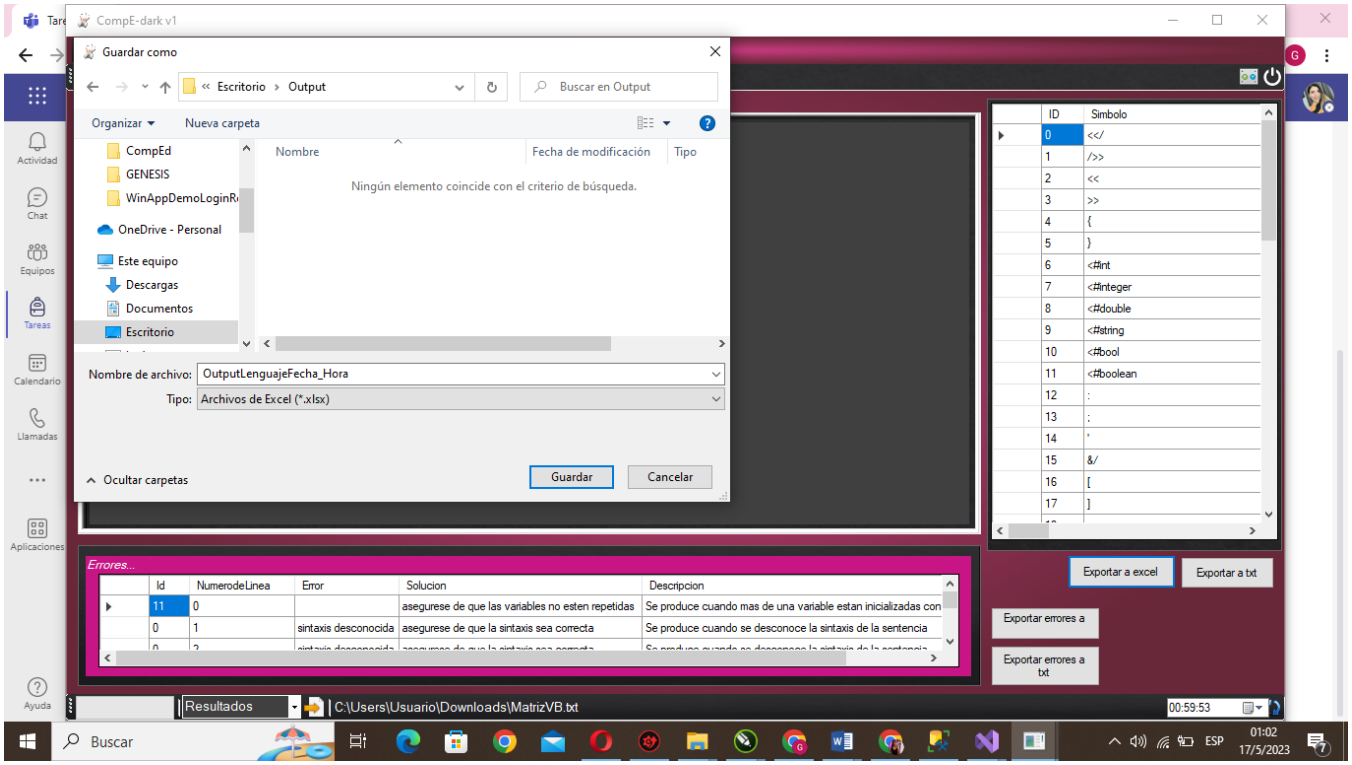

Bienvenido jmendez 1
Aceptar

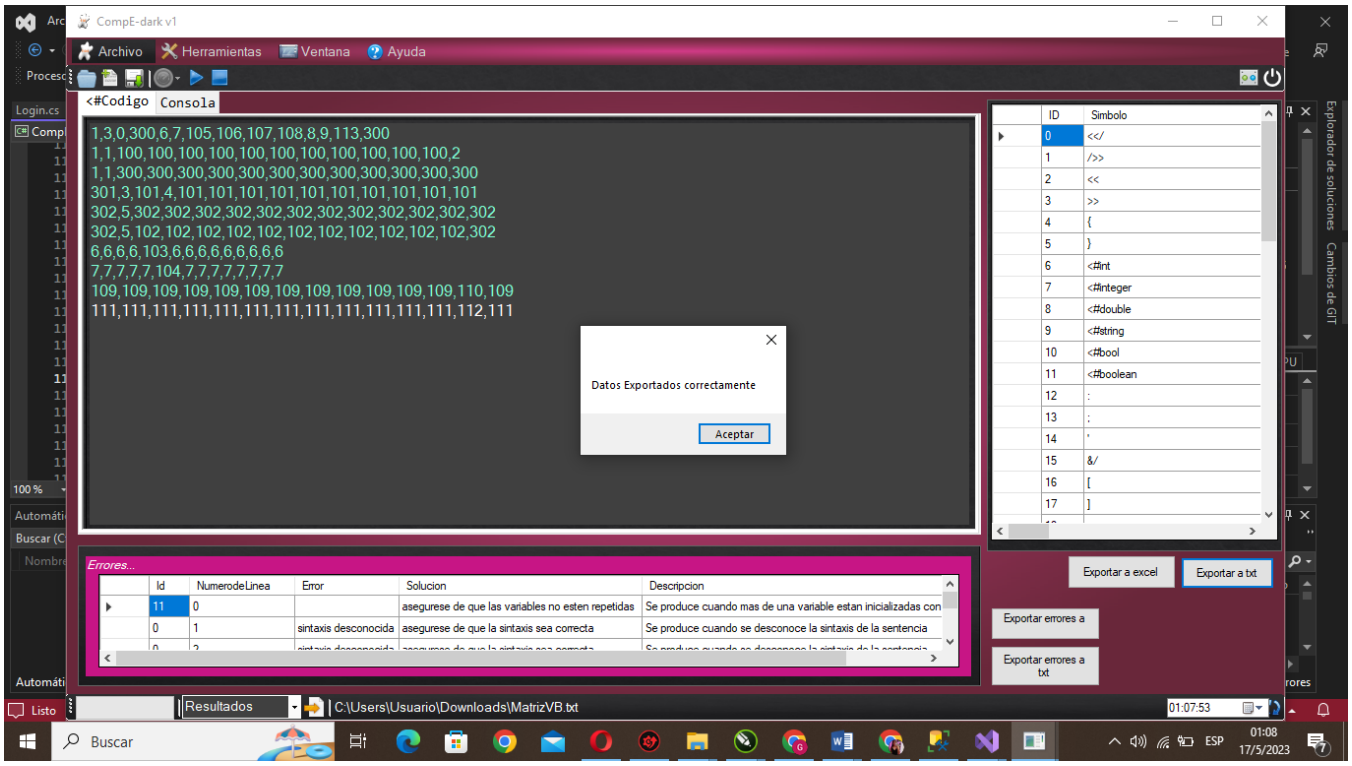

CompE-dark v1
Archivo
Herramientas
Ventana
Ayuda
Errores...
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
CompE-dark
00:53:59
Buscar
ESP
00:54
17/5/2023

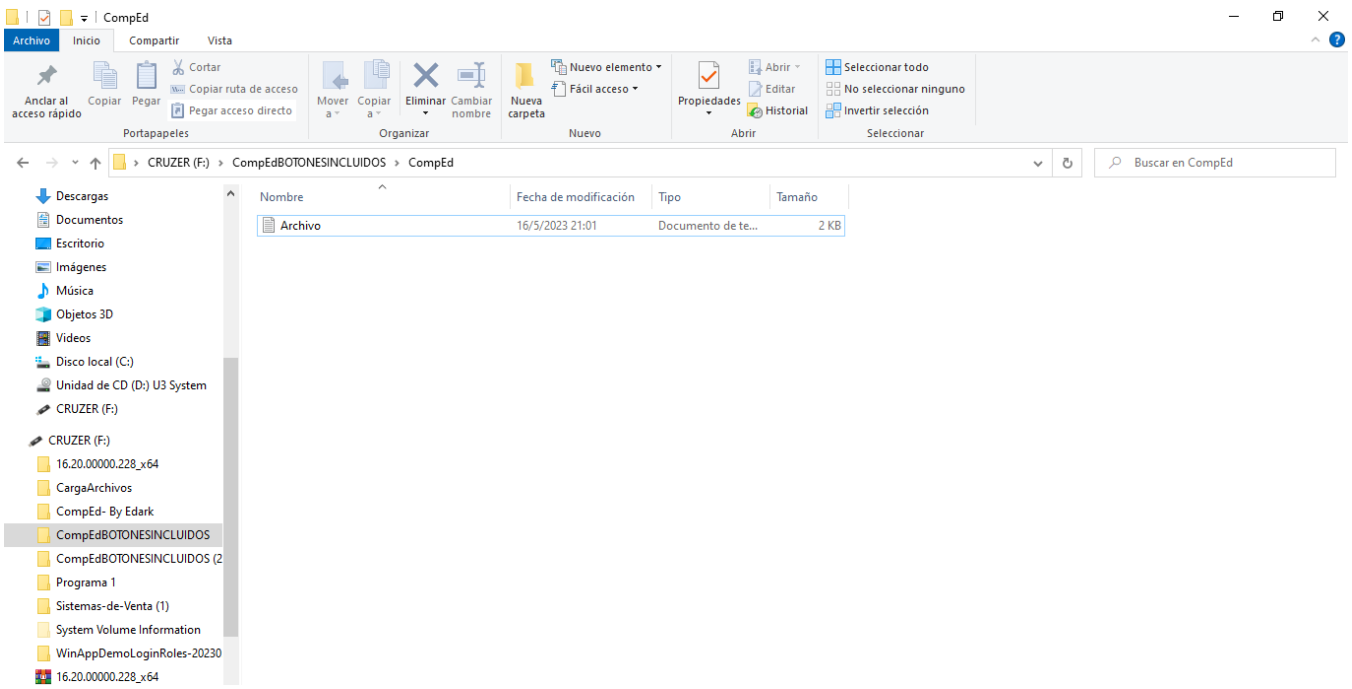

CompE-dark v1
CompEd#
Abrir Archivo
Este equipo › Descargas ›
Buscar en Descargas
Organizar
Nueva carpeta
Descargas
Documentos
Escritorio
Imágenes
Música
Objetos 3D
Videos
Disco local (C:)
Unidad de CD (D
CRUZER (F:)
CRUZER (F:)
16.20.00000.228_
Hoy (1)
MatrizVB
Ayer (16)
Proyecto final
Programa 1
CompEdBOTONESINCLUIDOS
appCompilador-master
Codigo exportar
WinAppDemoLoginRoles-20230516T2256...
Nombre de archivo:
Abrir
Cancelar
Errores...
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
CompE-dark
01:10:13
Buscar
ESP
01:10
17/5/2023


CompE-dark v1
Archivo
Herramientas
Ventana
Ayuda
<#Codigo
Consola
1,3,0,300,6,7,105,106,107,108,8,9,113,300
1,1,100,100,100,100,100,100,100,100,100,100,100,2
1,1,300,300,300,300,300,300,300,300,300,300,300,300
301,3,101,4,101,101,101,101,101,101,101,101,101,101
302,5,302,302,302,302,302,302,302,302,302,302,302,302
302,5,102,102,102,102,102,102,102,102,102,102,102,302
6,6,6,6,103,6,6,6,6,6,6,6,6,6
7,7,7,7,7,104,7,7,7,7,7,7,7,7
109,109,109,109,109,109,109,109,109,109,109,109,110,109
111,111,111,111,111,111,111,111,111,111,111,111,112,111
Errores...
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
C:\Users\Usuario\Downloads\MatrizVB.txt
00:59:53
Buscar
ESP
01:00
17/5/2023


CompE-dark v1
Archivo
Herramientas
Ventana
Ayuda
<#Codigo
Consola
Compilar
1,3,0,300,6,7,105,106,107,108,8,9,113,300
1,1,100,100,100,100,100,100,100,100,100,100,2
1,1,300,300,300,300,300,300,300,300,300,300,300
301,3,101,4,101,101,101,101,101,101,101,101,101
302,5,302,302,302,302,302,302,302,302,302,302,302
302,5,102,102,102,102,102,102,102,102,102,102,302
6,6,6,6,103,6,6,6,6,6,6,6,6
7,7,7,7,7,104,7,7,7,7,7,7,7
109,109,109,109,109,109,109,109,109,109,109,110,109
111,111,111,111,111,111,111,111,111,111,111,112,111
ID
Simbolo
0 <<!
1 />>
2 <<
3 >>
4 {
5 }
6 <#int
7 <#integer
8 <#double
9 <#string
10 <#bool
11 <#boolean
12 :
13 ;
14 '
15 &!
16 [
17]
Errores...
Id
NumerodeLinea
Error
Solucion
Descripcion
11 0 asegurese de que las variables no esten repetidas Se produce cuando mas de una variable estan inicializadas con
0 1 sintaxis desconocida asegurese de que la sintaxis sea correcta Se produce cuando se desconoce la sintaxis de la sentencia
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
C:\Users\Usuario\Downloads\MatrizVB.txt
00:59:53
Buscar
ESP
01:00
17/5/2023


CompE-dark v1
Guardar como
Escritorio › Output
Buscar en Output
Organizar
Nueva carpeta
CompEd
GENESIS
WinAppDemoLoginR
OneDrive - Personal
Este equipo
Descargas
Documentos
Escritorio
Nombre
Fecha de modificación
Tipo
Ningún elemento coincide con el criterio de búsqueda.
Nombre de archivo: OutputLenguajeFecha_Hora
Tipo: Archivos de Excel (*.xlsx)
Ocultar carpetas
Guardar
Cancelar
ID
Simbolo
0 <<!
1 />>
2 <<
3 >>
4 {
5 }
6 <#int
7 <#integer
8 <#double
9 <#string
10 <#bool
11 <#boolean
12 :
13 ;
14 '
15 &!
16 [
17]
Errores...
Id
NumerodeLinea
Error
Solucion
Descripcion
11 0 asegurese de que las variables no esten repetidas Se produce cuando mas de una variable estan inicializadas con
0 1 sintaxis desconocida asegurese de que la sintaxis sea correcta Se produce cuando se desconoce la sintaxis de la sentencia
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
C:\Users\Usuario\Downloads\MatrizVB.txt
00:59:53
Buscar
ESP
01:02
17/5/2023


CompE-dark v1
Archivo
Herramientas
Ventana
Ayuda
<#Codigo
Consola
1,3,0,300,6,7,105,106,107,108,8,9,113,300
1,1,100,100,100,100,100,100,100,100,100,100,2
1,1,300,300,300,300,300,300,300,300,300,300,300
301,3,101,4,101,101,101,101,101,101,101,101,101,101
302,5,302,302,302,302,302,302,302,302,302,302,302
302,5,102,102,102,102,102,102,102,102,102,102,302
6,6,6,6,103,6,6,6,6,6,6,6,6,6
7,7,7,7,7,104,7,7,7,7,7,7,7,7
109,109,109,109,109,109,109,109,109,109,109,110,109
111,111,111,111,111,111,111,111,111,111,111,111,112,111
Datos Exportados correctamente
Aceptar
ID
Simbolo
0 <</
1 />>
2 <<
3 >>
4 {
5 }
6 <#int
7 <#integer
8 <#double
9 <#string
10 <#bool
11 <#boolean
12 :
13 ;
14 '
15 &/
16 [
17]
Errores...
Id
NumerodeLinea
Error
Solucion
Descripcion
11 0 asegurese de que las variables no esten repetidas Se produce cuando mas de una variable estan inicializadas con
0 1 sintaxis desconocida asegurese de que la sintaxis sea correcta Se produce cuando se desconoce la sintaxis de la sentencia
Exportar a excel
Exportar a txt
Exportar errores a
Exportar errores a txt
Resultados
C:\Users\Usuario\Downloads\MatrizVB.txt
01:07:53
Listo
Buscar
ESP
01:08
17/5/2023


CompEd
Archivo
Inicio
Compartir
Vista
Anclar al acceso rápido
Copiar
Pegar
Cortar
Copiar ruta de acceso
Pegar acceso directo
Portapapeles
Mover a
Copiar a
Eliminar
Cambiar nombre
Nueva carpeta
Nuevo elemento
Fácil acceso
Organizar
Nuevo
Propiedades
Abrir
Editar
Historial
Abrir
Seleccionar todo
No seleccionar ninguno
Invertir selección
Seleccionar
CRUZER (F:) › CompEdBOTONESINCLUIDOS › CompEd
Buscar en CompEd
Descargas
Documentos
Escritorio
Imágenes
Música
Objetos 3D
Videos
Disco local (C:)
Unidad de CD (D:) U3 System
CRUZER (F:)
CRUZER (F:)
16.20.00000.228_x64
CargaArchivos
CompEd- By Edark
CompEdBOTONESINCLUIDOS
CompEdBOTONESINCLUIDOS (2
Programa 1
Sistemas-de-Venta (1)
System Volume Information
WinAppDemoLoginRoles-20230
16.20.00000.228_x64
Nombre
Fecha de modificación
Tipo
Tamaño
Archivo
16/5/2023 21:01
Documento de te...
2 KB

Código

Login

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WinAppDemoLoginRoles
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }

        SqlConnection conex = new
SqlConnection("Server=SERV_REDES\\SQLEXPRESS;DataBase=BD_Sistema;Trusted_Connection=True")
;


        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                conex.Open();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnIngresar_Click(object sender, EventArgs e)
        {
            String usua = txtUsuario.Text;
            String pass = txtPassword.Text;

            try
            {
                SqlCommand cmd = new SqlCommand("Select * From Usuario Where usuario='" +
usua + "' and contrasena ='" + pass + "'", conex);
                SqlDataReader dr = cmd.ExecuteReader();
                if (dr.Read())
                {
                    MessageBox.Show("Bienvenido " + dr.GetValue(1) + " " + dr.GetValue(3));
```

```
                CompEd.carga frmShared = new CompEd.carga();

                frmShared.ShowDialog();
            }
            else
            {
                MessageBox.Show("Usuario no Valido");
            }
            dr.Close();
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message);

        }
    }

    private void btnSalir_Click(object sender, EventArgs e)
    {
        Close();
    }
  }
}
```

Compilador

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using Tsimbolos;
using System.Text.RegularExpressions;//using necesario , llama ala referencia de la
libreria de expresiones regulares
using Microsoft.Office.Interop.Excel;
using ManejoDeErrores;

using Excel = Microsoft.Office.Interop.Excel;



namespace CompEd
{
    public partial class Ide : Form
    {
        int cantLineas =0 ;
        string nomarchivox;
        TS tabla_simbolos = new TS();
```

```
TE tabla_errorres = new TE();



public Ide()
{
    InitializeComponent();


}

private void salirToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Windows.Forms.Application.Exit();

}

private void toolStripButton1_Click(object sender, EventArgs e)
{
    //AnlzdrSntctc();

}

private void toolStripButton7_Click(object sender, EventArgs e)
{
    System.Windows.Forms.Application.Exit();
}

private void Ide_Load(object sender, EventArgs e)//---------------
{
    tabla_errorres.inicialestaE();
    tabla_simbolos.inicialista();
    tabControl1.Visible = false;
    PagCodigo.Select();
    PagCodigo.DetectUrls = true;
    #region  area de notificacion
    notifyIcon1.Text = " CompEd# 2013";
    notifyIcon1.BalloonTipTitle = " <# Hello World";
    notifyIcon1.BalloonTipText = "Bienvenido a CompEd# 2013";
    notifyIcon1.BalloonTipIcon = ToolTipIcon.Info;
    this.Click += new EventHandler(notifyIcon1_Click);

    notifyIcon1.Visible = true;
    notifyIcon1.ShowBalloonTip(3000);
    #endregion
}

private void notifyIcon1_Click(object sender, EventArgs e)
{
    if (this.WindowState == FormWindowState.Minimized)
        this.WindowState = FormWindowState.Maximized;

    this.Activate();
}
```

```
        private void notifyIcon1_BalloonTipShown(object sender, EventArgs e)
        {

        }

        private void toolStripContainer1_TopToolStripPanel_Click(object sender, EventArgs
e)
        {

        }

        private void Ide_FormClosing(object sender, FormClosingEventArgs e)
        {
            DialogResult dialogo = MessageBox.Show("¿ Desea cerrar CompE-dark ?","Cerrar
Compilador", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);
            if (dialogo == DialogResult.OK) {
                System.Windows.Forms.Application.Exit();
            }
            else {
                e.Cancel = true;
            }

        }

        private void acercaDeCompEdToolStripMenuItem_Click(object sender, EventArgs e)
        {


        }

        private void abrirToolStripMenuItem_Click(object sender, EventArgs e)
        {
            abrirarchivo();
        }


        //-----------------------------METODOS DE ARCHIVOS --------------------------------
---------

        public void exportaraexcel(DataGridView tabla)
        {

            Microsoft.Office.Interop.Excel.Application excel = new
Microsoft.Office.Interop.Excel.Application();

            excel.Application.Workbooks.Add(true);

            int ColumnIndex = 0;

            foreach (DataGridViewColumn col in tabla.Columns)
            {

                ColumnIndex++;

                excel.Cells[1, ColumnIndex] = col.Name;
```

```
            }

            int rowIndex = 0;

            foreach (DataGridViewRow row in tabla.Rows)
            {

                rowIndex++;

                ColumnIndex = 0;

                foreach (DataGridViewColumn col in tabla.Columns)
                {

                    ColumnIndex++;

                    excel.Cells[rowIndex + 1, ColumnIndex] = row.Cells[col.Name].Value;

                }

            }

            excel.Visible = true;

            Worksheet worksheet = (Worksheet)excel.ActiveSheet;

            worksheet.Activate();



        }

        public void abrirarchivo()
        {

             try{
            OpenFileDialog ofd = new OpenFileDialog();

            ofd.Title = "CompEd#
Abrir Archivo                                                                ";
            ofd.ShowDialog();
           // ofd.Filter = "Archivos ed#(*.ed)|*.ed";
            if (File.Exists(ofd.FileName))
            {
                using (Stream stream = ofd.OpenFile())
                {
                    //MessageBox.Show("archivo encontrado:  "+ofd.FileName);
                    leerarchivo(ofd.FileName);
                    nomarchivox = ofd.FileName;

                    txt_direccion.Text =ofd.FileName;
                    tabControl1.Visible = true;
                }
```

```
            }
            }catch(Exception){

                MessageBox.Show("El archivo no se abrio correctamente");

                tabla_errorres.addliste(2);
            }

        }

        public void leerarchivo(string nomarchivo)
        {
            StreamReader reader = new StreamReader(nomarchivo,
System.Text.Encoding.Default);
            //string read = reader.ReadLine();
            string texto;
           // while (read != null)
            //{
                texto = reader.ReadToEnd();
               // read = read + "\n";

                reader.Close();

                PagCodigo.Text = texto;
               // read =reader.ReadLine();

            //}


        }

        public bool revisasiarchivoexiste(string nomarchivo)
        {

            bool existe;

            if (File.Exists(nomarchivo))
            {
                // el archivo existe
                existe = true;
            }
            else
            {
                // el archivo no extiste
                existe = false;
            }
            return existe;
        }

        public void guardaArchivo()
        {
            SaveFileDialog saveFile = new SaveFileDialog();
            saveFile.Filter = "Archivos ed|*.ed";
            if (saveFile.ShowDialog() == DialogResult.OK)
            {
```

```
            if (File.Exists(saveFile.FileName))
            {
                // el archivo existe
                //------------------- para un log que agrega eventos ..................
                //StreamWriter writer = File.AppendText(nomarchivo);
                //writer.WriteLine("\n <</ ---Actualizacion del " +
DateTime.Now.ToString() + " />>");
                //writer.Write(PagCodigo.Text);
                //writer.Close();

                //------------------ para sobrescribir el texto ...................
                StreamWriter codigonuevo = File.CreateText(saveFile.FileName);
                codigonuevo.Write(PagCodigo.Text);
               codigonuevo.Flush();
                codigonuevo.Close();
                nomarchivox = saveFile.FileName;
                txt_direccion.Text = saveFile.FileName;

            }
            else
            {
                // el archivo no extiste

                    StreamWriter codigonuevo = File.CreateText(saveFile.FileName);
                    codigonuevo.Write(PagCodigo.Text);
                    codigonuevo.Write("\n \n <</ Archivo creado el: " +
DateTime.Now.ToString() + " />> \n ");
                    codigonuevo.Flush();
                    codigonuevo.Close();
                    nomarchivox = saveFile.FileName;
                    txt_direccion.Text = saveFile.FileName;
            }


        }

    }

    public void guardaArchivo2(string nomarchivo)
    {
        try
        {
            if (nomarchivo == null)
            {
                guardaArchivo();

            }
            else
            {
                // el archivo nuevo
                StreamWriter codigonuevo = File.CreateText(nomarchivo);
                codigonuevo.Write(PagCodigo.Text);
                codigonuevo.Flush();
                codigonuevo.Close();
```

```
        }
    }
    catch (Exception)
    {

        MessageBox.Show("error al guardar");

    }


}

public void leer_archivo_al(string nomarchivo)
{

    int contador_Ambitoi =0;
    int contador_Ambitf = 0;
    int ambito = 0;
    try
    {
        StreamReader reader = new StreamReader(nomarchivo);
        string[] Palabras_Separadas;
        string read;
        int numero_de_lineas = 0;
        PagCodigo.Select(0, PagCodigo.SelectionStart);

        while (reader != null)
        {
            numero_de_lineas = numero_de_lineas + 1;
            read = reader.ReadLine();


            if (reader.EndOfStream)
            {
                //MessageBox.Show("ultima linea");

                break;
            }
            else
            {


                Palabras_Separadas = read.Split(' ');
                foreach (var palabra in Palabras_Separadas)
                {
                    #region Medicion del ambito

                    if (palabra == "{")
                    {
                        contador_Ambitoi = contador_Ambitoi + 1;
                    }
                    if (palabra == "}")
                    {
                        contador_Ambitf = contador_Ambitf + 1;
                    }
```

```
                        ambito = contador_Ambitoi;


                        #endregion


                       //-----------------------------------------------------------------
------


                        if (tabla_simbolos.compararAL(palabra.ToString()) && palabra !=
null)// se manda a comparar la palabra con la tabla de simbolos
                        {
                            //                                                  simb  ,val,
nunlin          ,tam,ambit,              id_,          tipo,      descrip
                            //uneSentencias();

                            tabla_de_simbolos objnuevo = new tabla_de_simbolos(palabra, "",
numero_de_lineas, -0, ambito, tabla_simbolos.compararALRef(palabra.ToString()), "palabra
nueva", "palabra que coincide con la Tabla de simbolos", "");
                            tabla_simbolos.añadir_obj(objnuevo);

                            PagCodigo.SelectionStart = PagCodigo.Find(palabra);
                            PagCodigo.SelectionColor = Color.DodgerBlue;

                        }
                        else//de no estar en la tabla de simbolos se agrega a un campo
nuevo
                        {
                            if (Regex.IsMatch(palabra, @"[a-zA-Z]") && palabra !=
null)//sentencia que revisa los dos texbox
                            {
                                // System.Windows.Forms.MessageBox.Show("esto es una
palabra");
                                tabla_de_simbolos objnuevo = new tabla_de_simbolos(palabra,
"", numero_de_lineas, -0, ambito, tabla_simbolos.contlineas() + 1, "palabra nueva",
"palabra que no coincide con la Tabla de simbolos,pero no se considera error","");
                                tabla_simbolos.añadir_obj(objnuevo);
                            }
                            else if (Regex.IsMatch(palabra,
@"\d{1}|\d{2}|\d{3}|\d{4}|\d{5}") && palabra != null)
                            {
                                //System.Windows.Forms.MessageBox.Show("esto es un
numero");
                                tabla_de_simbolos objnuevo = new tabla_de_simbolos(palabra,
palabra, numero_de_lineas, -0, ambito, tabla_simbolos.contlineas() + 1, "numero nuevo",
"numero","");
                                tabla_simbolos.añadir_obj(objnuevo);

                                PagCodigo.SelectionStart = PagCodigo.Find(palabra);
                                PagCodigo.SelectionColor = Color.Aquamarine;

                            }
                            else
                            {
```

```
                                // System.Windows.Forms.MessageBox.Show("Error en la
expresion \n no cumple con un formato correcto ");

                            }
                        }
                    }//fin del analisis lexico

                }
                Palabras_Separadas = null;
                cantLineas = numero_de_lineas;
            }

            if (contador_Ambitf != contador_Ambitoi)
            {
                //MessageBox.Show("error de ambito");
                tabla_errorres.addliste(8);


            }


            reader.Close();
        }
        catch (ArgumentNullException)
        {

            MessageBox.Show("El archivo no se abrio correctamente");

            tabla_errorres.addliste(2);
        }
        catch (Exception)
        {
            MessageBox.Show("error");
        }


    }

    public string[] uneSentencias (){
        string sentencia = null;
        string[] sentencias = new string[cantLineas];
        int bandera = 0;
        string tipov = "";

        for (int i = 1; i < cantLineas; i++) //une los token de cada linea
        {
            foreach (var token in tabla_simbolos.llamatabla())
            {
                if (token.NumLinea == i && token != null)
                {
                    if (bandera == 0 && Regex.IsMatch(token.Simbolo,
@"(<#int|<#integer|<#double|<#bool|<#string|<#real|<#boolean)$"))
                    {
                        token.TipoVar = token.Simbolo;
                        tipov = token.Simbolo;
```

```
                    }


                    if (bandera != 0)
                    {
                        sentencia = sentencia + " " + token.simbolo.ToString();
                        token.TipoVar = tipov;
                    }
                    else
                    {
                        sentencia = sentencia + token.simbolo.ToString();
                        bandera = 1;
                    }

                }
            }
            sentencias[i] = sentencia;
            sentencia = null;
            bandera = 0;
            tipov = "";
        }

        return sentencias;
    }
    #region analizador sintactico
    public void AnlzdrSntctc(string[] sentencias)
    {

       for (int i = 1; i < sentencias.Length; i++)
       {
           //MessageBox.Show(sentencias[i]);
           #region  Expresiones regulares
           if (sentencias[i] != null)
           {
               if (Regex.IsMatch(sentencias[i], @"^<#int|<#integer\s+[a-
z](1,15)(\s+:\s+\d(0,32000))*;$"))
               {

                   System.Windows.Forms.MessageBox.Show("esto es una sentencia int");
                   #region parte semantica
                   string[] separanum;
                   separanum = sentencias[i].Split(' ');
                   try{
                       int num ;
                      num = int.Parse(separanum[3]);

                      MessageBox.Show("si es un numero entero");

                   }catch(FormatException e){

                       MessageBox.Show("no es un numero entero");
                       tabla_errorres.addliste(0,i);

                    }catch(IndexOutOfRangeException e){
```

```
                    tabla_errorres.addliste(10, i);
                    MessageBox.Show("error de escritura");

                }
                #endregion


            }
            else if (Regex.IsMatch(sentencias[i], @"^<#double|<#real\s+[a-
z](1,15)(\s+:\s+\d(0,32000))*;$"))
            {
                System.Windows.Forms.MessageBox.Show("esto es una sentencia double");
                #region parte semantica
                string[] separanum;
                separanum = sentencias[i].Split(' ');
                try
                {
                    double num;
                    num = double.Parse(separanum[3]);

                    MessageBox.Show("si es un numero double");

                }
                catch (FormatException e)
                {
                    tabla_errorres.addliste(0,i);
                    MessageBox.Show("no es un numero double");

                }catch(IndexOutOfRangeException e){

                    tabla_errorres.addliste(10, i);
                    MessageBox.Show("error de escritura");

                }


                #endregion

            }
            else if (Regex.IsMatch(sentencias[i], @"^<#string|<#texto\s+[a-
z](1,15)(\s+:\s+[a-z](1,15)')*;$"))
            {
                System.Windows.Forms.MessageBox.Show("esto es una sentencia string");

            }
            else if (Regex.IsMatch(sentencias[i], @"^<#bool|<#boolean\s+[a-
z](1,15)(\s+:\s+(true|false))*;$"))
            {
                System.Windows.Forms.MessageBox.Show("esto es una sentencia bool");

                #region parte semantica
                string[] separavar;
                separavar = sentencias[i].Split(' ');
                try
                {
```

```
                    bool var;
                    var = bool.Parse(separavar[3]);

                    MessageBox.Show("si es una variable bool");

                }
                catch (FormatException e)
                {

                    MessageBox.Show("no es una variable bool");
                    tabla_errorres.addliste(0, i);
                }
                #endregion

            }
            else if (Regex.IsMatch(sentencias[i], @"<<*.*>>$"))
            {
                MessageBox.Show("Esto es un comentario");
            }
            else if (Regex.IsMatch(sentencias[i], @"[a-z]\s+:\s[a-
z]|(\w)*\s\+\s(\w)*|\d(0,32000)*\s;$"))
            {
                MessageBox.Show("esto es una sentecia de asignacion");

                #region parte semantica
                string tpv1 = "";
                string tpv2 = "";
                string tpv3 = "";

                string[] separavar;
                separavar = sentencias[i].Split(' ');

                if (Regex.IsMatch(sentencias[i], @"[a-
z]\s+:\s(\w)*\s\+\s(\w)*\s;$"))//-- asignacion del tipo monto : num1 + num2 ;
                {

                    foreach (var token in tabla_simbolos.llamatabla())
                    {
                        if (token.Simbolo == separavar [0])
                        {
                            tpv1 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[2])
                        {
                            tpv2 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[4])
                        {
                            tpv3 = token.TipoVar;
                        }

                    }//-- fin del foreach
                    if (tpv1 == tpv2 && tpv2 == tpv3 && tpv1 != "")
                    {
                        MessageBox.Show("el tipo de las variables son el mismo");
```

```
                    }
                }
                if (Regex.IsMatch(sentencias[i], @"[a-z]\s+:\s(\w)*\s\-
\s(\w)*\s;$"))//-- asignacion del tipo monto : num1 - num2 ;
                {

                    foreach (var token in tabla_simbolos.llamatabla())
                    {
                        if (token.Simbolo == separavar[0])
                        {
                            tpv1 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[2])
                        {
                            tpv2 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[4])
                        {
                            tpv3 = token.TipoVar;
                        }

                    }//-- fin del foreach
                    if (tpv1 == tpv2 && tpv2 == tpv3 && tpv1 != "")
                    {
                        MessageBox.Show("el tipo de las variables son el mismo");
                    }
                }
                if (Regex.IsMatch(sentencias[i], @"[a-
z]\s+:\s(\w)*\s\/\s(\w)*\s;$"))//-- asignacion del tipo monto : num1 / num2 ;
                {

                    foreach (var token in tabla_simbolos.llamatabla())
                    {
                        if (token.Simbolo == separavar[0])
                        {
                            tpv1 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[2])
                        {
                            tpv2 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[4])
                        {
                            tpv3 = token.TipoVar;
                        }

                    }//-- fin del foreach
                    if (tpv1 == tpv2 && tpv2 == tpv3 && tpv1 != "")
                    {
                        MessageBox.Show("el tipo de las variables son el mismo");
                    }
                }
                if (Regex.IsMatch(sentencias[i], @"[a-
z]\s+:\s(\w)*\s\*\s(\w)*\s;$"))//-- asignacion del tipo monto : num1 * num2 ;
                {
```

```
                    foreach (var token in tabla_simbolos.llamatabla())
                    {
                        if (token.Simbolo == separavar[0])
                        {
                            tpv1 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[2])
                        {
                            tpv2 = token.TipoVar;
                        }
                        if (token.Simbolo == separavar[4])
                        {
                            tpv3 = token.TipoVar;
                        }

                    }//-- fin del foreach
                    if (tpv1 == tpv2 && tpv2 == tpv3 && tpv1 != "")
                    {
                        MessageBox.Show("el tipo de las variables son el mismo");
                    }
                }


                #endregion


            }
            else if (Regex.IsMatch(sentencias[i], @"^{$"))
            {
                MessageBox.Show("inicio de ambito");
            }
            else if (Regex.IsMatch(sentencias[i], @"^}$"))
            {
                MessageBox.Show("fin de ambito");
            }
            else if (Regex.IsMatch(sentencias[i],
@"<<si_\s\(\s+\w+\s(<|>|<:|>:|::|!:)\s\w+\s\)\s\{$"))//--
            {
                MessageBox.Show("comienzo de if");
            }
            else if (Regex.IsMatch(sentencias[i],
@"<<ysi_\s\(\s+\w+\s(<|>|<:|>:|::|!:)\s\w+\s\)\s\{$"))//--
            {
                MessageBox.Show("comienzo de else if");
            }
            else if (Regex.IsMatch(sentencias[i], @"<<sino\s*\{$"))//--
            {
                MessageBox.Show("comienzo de else");
            }
            else if (Regex.IsMatch(sentencias[i],
@"#ncasd\s\(\s\w+\s(<|>|<:|>:|::|!:)\s\w+\s\)\s\{$"))//--
            {
                MessageBox.Show("comienzo del switch");
            }
```

```
                else if (Regex.IsMatch(sentencias[i], @"casd\s\(\s(\w+|\d+)\s\)\s{$"))//--
                {
                    MessageBox.Show("comienzo de case");
                }
                else if (Regex.IsMatch(sentencias[i], @"fcasd\s;$"))//--
                {
                    MessageBox.Show("break del case");
                }
                else if (Regex.IsMatch(sentencias[i],
@"#mintrs\s\(\s\w+\s(<|>|<:|>:|::|!:)\s\w+\s\)\s{$"))
                {
                    MessageBox.Show("inicio de un while");
                }

                else if (Regex.IsMatch(sentencias[i], @"#mostrar\s\(\s(\w*)|'\w*'\)\s;$"))
                {
                    MessageBox.Show("mostrar por pantalla \n" + sentencias[i]);

                }
                //else if (Regex.IsMatch(sentencias[i],@""))
                //{
                //MessageBox.Show("");
                //}
                //else if (Regex.IsMatch(sentencias[i],@""))
                //{
                //MessageBox.Show("");
                //}
                //else if (Regex.IsMatch(sentencias[i],@""))
                //{
                //MessageBox.Show("");
                //}
                //else if (Regex.IsMatch(sentencias[i],@""))
                //{
                //MessageBox.Show("");
                //}
                else
                {
                    if (sentencias[i] != null)
                    {
                        tabla_errorres.addliste(9, i);
                        //MessageBox.Show("Expresion invalida");
                    }
                }
            }

            #endregion

            //System.Windows.Forms.MessageBox.Show("" + sentencias[i]);
        }




    }
```

```
#endregion



private void guardarToolStripMenuItem_Click(object sender, EventArgs e)
{
    guardaArchivo();
}

private void toolStripButton3_Click(object sender, EventArgs e)
{
    guardaArchivo2(nomarchivox);


}

private void guardarToolStripMenuItem1_Click(object sender, EventArgs e)
{
    guardaArchivo2(nomarchivox);

}

private void nuevoToolStripMenuItem_Click(object sender, EventArgs e)
{
    tabControl1.Visible = true;

}

private void toolStripButton8_Click(object sender, EventArgs e)
{
    dataGridView1.DataSource = null;
    dataGridView2.DataSource = null;
    dataGridView1.DataSource = tabla_simbolos.llamatabla();
    dataGridView2.DataSource = tabla_errorres.llamatablaE();

}

private void analizadorLexicoToolStripMenuItem_Click(object sender, EventArgs e)
{
    leer_archivo_al(nomarchivox);
}

private void toolStripButton9_Click(object sender, EventArgs e)
{
   tabControl1.Visible = true;

}

private void cerrarProyectoToolStripMenuItem_Click(object sender, EventArgs e)
{
    tabControl1.Visible = false ;

}
```

```
        private void toolStripButton6_Click(object sender, EventArgs e)
        {
            abrirarchivo();
        }

        private void maximizarVentanaToolStripMenuItem_Click(object sender, EventArgs e)
        {
               this.WindowState = FormWindowState.Maximized;
        }

        private void minimizarVentanaToolStripMenuItem_Click(object sender, EventArgs e)
        {
               this.WindowState = FormWindowState.Normal;
        }

        private void minimizarVentanaToolStripMenuItem1_Click(object sender, EventArgs e)
        {
                this.WindowState = FormWindowState.Minimized;
        }

        private void opcionesToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }

        private void colorDeLaFuenteToolStripMenuItem_Click(object sender, EventArgs e)
        {

            var cl = colorDialog1.ShowDialog();
            if (cl == System.Windows.Forms.DialogResult.OK)
            {
                //PagCodigo.SelectionColor = colorDialog1.Color; <..... esto para una parte
del texto
                PagCodigo.ForeColor = colorDialog1.Color;
            }



        }

        private void colorDeConsolaToolStripMenuItem_Click(object sender, EventArgs e)
        {
            var cl = colorDialog1.ShowDialog();
            if (cl == System.Windows.Forms.DialogResult.OK)
            {
                //PagCodigo.SelectionColor = colorDialog1.Color; <..... esto para una parte
del texto
                PagCodigo.BackColor = colorDialog1.Color;
            }

        }

        private void formatoToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
            var fm = fontDialog1.ShowDialog();
            if (fm == DialogResult.OK)
            {
                //PagCodigo.SelectionColor = colorDialog1.Color; <..... esto para una parte
del texto
                PagCodigo.Font = fontDialog1.Font;
            }


        }

        private void toolStripButton2_Click(object sender, EventArgs e)
        {
            if (Cb_resultados.Text == "ver log de errores")
            {
                exportaraexcel(dataGridView2);
            }
            else if (Cb_resultados.Text == "ver log de simbolos")
            {
                exportaraexcel(dataGridView1);
            }

        }

        private void toolStripButton5_Click(object sender, EventArgs e)
        {
            guardaArchivo2(nomarchivox);
            tabla_simbolos.reinicialista();
            tabla_errorres.reinicialista();
            tabla_errorres.inicialestaE();
            tabla_simbolos.inicialista();
            leer_archivo_al(nomarchivox);
            string[] sent = uneSentencias();
            tabla_simbolos.compararALsemantic();

            if (tabla_simbolos.revisar_duplicados())
            {
                tabla_errorres.addliste(11);
            }

            AnlzdrSntctc(sent);
            dataGridView1.DataSource = null;
            dataGridView2.DataSource = null;
            dataGridView1.DataSource = tabla_simbolos.llamatabla();
            dataGridView2.DataSource = tabla_errorres.llamatablaE();
            System.Media.SystemSounds.Asterisk.Play();




        }

        private void PagCodigo_TextChanged(object sender, EventArgs e)
        {
```

```
        }

        private void button1_Click(object sender, EventArgs e)
        {


                // Crear una instancia de Excel
                Excel.Application excel = new Excel.Application();
                Excel.Workbook workbook = excel.Workbooks.Add(Type.Missing);
                Excel.Worksheet worksheet = null;

                try
                {
                    worksheet = workbook.ActiveSheet;

                    int columnIndex = 1;
                    int rowIndex = 1;

                    // Recorrer las columnas del DataGridView y escribir los encabezados en
Excel
                    foreach (DataGridViewColumn column in dataGridView1.Columns)
                    {
                        worksheet.Cells[rowIndex, columnIndex] = column.HeaderText;
                        columnIndex++;
                    }

                    rowIndex++;

                    // Recorrer las filas del DataGridView y escribir los datos en Excel
                    foreach (DataGridViewRow row in dataGridView1.Rows)
                    {
                        columnIndex = 1;

                        foreach (DataGridViewCell cell in row.Cells)
                        {
                            worksheet.Cells[rowIndex, columnIndex] = cell.Value;
                            columnIndex++;
                        }

                        rowIndex++;
                    }

                    // Guardar el archivo de Excel
                    SaveFileDialog saveDialog = new SaveFileDialog();
                    saveDialog.Filter = "Archivos de Excel (*.xlsx)|*.xlsx";
                    saveDialog.FileName = "ExportedData.xlsx";

                    if (saveDialog.ShowDialog() == DialogResult.OK)
                    {
                        workbook.SaveAs(saveDialog.FileName);
                        MessageBox.Show("Datos exportados correctamente a Excel.");
                    }
                }
                catch (Exception ex)
```

```
            {
                MessageBox.Show("Error al exportar los datos a Excel: " + ex.Message);
            }
            finally
            {
                // Cerrar y liberar recursos
                workbook.Close();
                excel.Quit();

                releaseObject(worksheet);
                releaseObject(workbook);
                releaseObject(excel);
            }
        }

        private void releaseObject(object obj)
        {
            try
            {
                System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
                obj = null;
            }
            catch (Exception ex)
            {
                obj = null;
                MessageBox.Show("Error al liberar el objeto: " + ex.ToString());
            }
            finally
            {
                GC.Collect();
            }
        }

    private void button2_Click(object sender, EventArgs e)
    {
        // Crear una instancia de Excel
        Excel.Application excel = new Excel.Application();
        Excel.Workbook workbook = excel.Workbooks.Add(Type.Missing);
        Excel.Worksheet worksheet = null;

        try
        {
            worksheet = workbook.ActiveSheet;

            int columnIndex = 1;
            int rowIndex = 1;

            // Recorrer las columnas del DataGridView y escribir los encabezados en
Excel
            foreach (DataGridViewColumn column in dataGridView2.Columns)
            {
                worksheet.Cells[rowIndex, columnIndex] = column.HeaderText;
                columnIndex++;
            }
```

```
            rowIndex++;

            // Recorrer las filas del DataGridView y escribir los datos en Excel
            foreach (DataGridViewRow row in dataGridView2.Rows)
            {
                columnIndex = 1;

                foreach (DataGridViewCell cell in row.Cells)
                {
                    worksheet.Cells[rowIndex, columnIndex] = cell.Value;
                    columnIndex++;
                }

                rowIndex++;
            }

            // Guardar el archivo de Excel
            SaveFileDialog saveDialog = new SaveFileDialog();
            saveDialog.Filter = "Archivos de Excel (*.xlsx)|*.xlsx";
            saveDialog.FileName = "ExportedData.xlsx";

            if (saveDialog.ShowDialog() == DialogResult.OK)
            {
                workbook.SaveAs(saveDialog.FileName);
                MessageBox.Show("Datos exportados correctamente a Excel.");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error al exportar los datos a Excel: " + ex.Message);
        }
        finally
        {
            // Cerrar y liberar recursos
            workbook.Close();
            excel.Quit();

            releaseObject2(worksheet);
            releaseObject2(workbook);
            releaseObject2(excel);
        }
    }

    private void releaseObject2(object obj)
    {
        try
        {
            System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
            obj = null;
        }
        catch (Exception ex)
        {
            obj = null;
            MessageBox.Show("Error al liberar el objeto: " + ex.ToString());
        }
```

```
            finally
            {
                GC.Collect();
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {

            if (!Directory.Exists(@"F:\CompEdBOTONESINCLUIDOS"))
            {
                Directory.CreateDirectory(@"F:\CompEdBOTONESINCLUIDOS");
            }

            TextWriter sw = new StreamWriter(@"F:\CompEdBOTONESINCLUIDOS\Archivo.txt");
            int rowcount = dataGridView1.Rows.Count;
            for (int i = 0; i < rowcount - 1; i++)
            {
                sw.WriteLine(dataGridView1.Rows[i].Cells[0].Value.ToString() + "\t"
                           + dataGridView1.Rows[i].Cells[1].Value.ToString() + "\t"
                            + dataGridView1.Rows[i].Cells[2].Value.ToString() + "\t"
                             + dataGridView1.Rows[i].Cells[3].Value.ToString() + "\t");
            }
            sw.Close();
            MessageBox.Show("Datos Exportados correctamente");

        }

        private void button4_Click(object sender, EventArgs e)
        {
            if (!Directory.Exists(@"F:\CompEdBOTONESINCLUIDOS"))
            {
                Directory.CreateDirectory(@"F:\CompEdBOTONESINCLUIDOS");
            }

            TextWriter sw = new StreamWriter(@"F:\CompEdBOTONESINCLUIDOS\Archivo.txt");
            int rowcount = dataGridView2.Rows.Count;
            for (int i = 0; i < rowcount - 1; i++)
            {
                sw.WriteLine(dataGridView2.Rows[i].Cells[0].Value.ToString() + "\t"
                           + dataGridView2.Rows[i].Cells[1].Value.ToString() + "\t"
                            + dataGridView2.Rows[i].Cells[2].Value.ToString() + "\t"
                             + dataGridView2.Rows[i].Cells[3].Value.ToString() + "\t");
            }
            sw.Close();
            MessageBox.Show("Datos Exportados correctamente");
        }

        private void toolStrip2_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
        {

        }
    }
}
```

Carga

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace CompEd
{
    public partial class carga : Form
    {


        int total = 0;
        public carga()
        {
            InitializeComponent();



        }

        private void ini_DoubleClick(object sender, EventArgs e)
        {
            Ide id = new Ide();
            this.Hide();

            id.Show();

        }

        private void label2_DoubleClick(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void ini_Load(object sender, EventArgs e)
        {
            progressBar1.Maximum = 100;
```

```
            //minimum indica el valor mínimo de la barra
            progressBar1.Minimum = 0;

            //value indica desde donde se va a comenzar a llenar la barra, la nuestra
iniciara desde cero
            progressBar1.Value = 0;

            //Por ejemplo podemos hacer que la barra inicie desde la mitad
            //la siguiente instrucción indica que inicie cargando desde la mitad del tamaño
de la barra
            //progressBar1.Value = progressBar1.Maximum / 2;

            //step indica el paso de la barra, entre más pequeño sea la barra tardará más en
cargar
            progressBar1.Step = 5;
            //// maximum indica el valor máximo de la barra
            //progressBar1.Maximum = 100;

            ////minimum indica el valor mínimo de la barra
            //progressBar1.Minimum = 0;

            ////value indica desde donde se va a comenzar a llenar la barra, la nuestra
iniciara desde cero
            //progressBar1.Value = 0;

            ////Por ejemplo podemos hacer que la barra inicie desde la mitad
            ////la siguiente instrucción indica que inicie cargando desde la mitad del
tamaño de la barra
            ////progressBar1.Value = progressBar1.Maximum / 2;

            ////step indica el paso de la barra, entre más pequeño sea la barra tardará más
en cargar
            //progressBar1.Step = 5;

            //el ciclo for cargará la barra
            //for (int i = progressBar1.Minimum; i < progressBar1.Maximum; i = i +
progressBar1.Step)
            //{
            //    //esta instrucción avanza la posición actual de la barra
            //    progressBar1.PerformStep();
            //    label1.Text = "cargando complementos..";
            //}
            //label1.Text = "sistema cargado.";
            timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            int i = progressBar1.Minimum;

            total = total + 1;
            if (total == 2 && i < progressBar1.Maximum)
            {
                progressBar1.PerformStep();
                progressBar1.PerformStep();
```

```
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            label1.Text = "cargando complementos.";
        }

        if (total == 3 && i < progressBar1.Maximum)
        {
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            label1.Text = "cargando complementos..";
        }

        if (total == 4 && i < progressBar1.Maximum)
        {
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            progressBar1.PerformStep();
            label1.Text = "cargando complementos...";
        }

        if (total == 5 && i < progressBar1.Maximum)
        {
            label1.Text = "sistema cargado";

        for (int isf = i; isf < progressBar1.Maximum; isf = isf + progressBar1.Step)
        {
            //esta instrucción avanza la posición actual de la barra
            progressBar1.PerformStep();
            label1.Text = "sistema cargado";

        }


        }

        if (total == 6 )
        {
            Ide id = new Ide();
        this.Hide();

        id.Show();
        }




    }
  }
}
```