

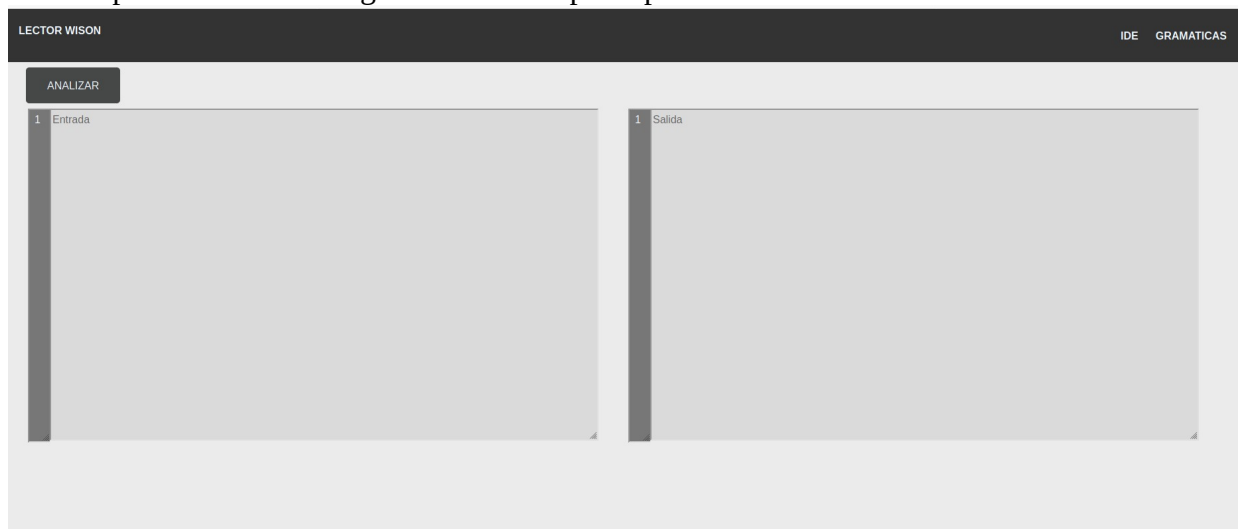
UNIVERSIDAD SAN CARLOS DE GUATEMALA USAC
CENTRO UNIVERSITARIO DE OCCIDENTE
EST: DENILSON FLORENTÍN DE LEÓN AGUILAR
CAR: 201830313
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES
MANUAL DE USUARIO – PRÁCTICA 2

REQUERIMIENTOS DEL SISTEMA:

- Para que la aplicación cliente funcione, se recomienda tener un sistema operativo linux (de preferencia Ubuntu 18.04LTS o superior, ya que fue desarrollada en esa versión). Además de ello tener instalado npm, nvm (opcional), angular (ng), tener un navegador web disponible.
- Se recomienda tener una resolución elevada para disfrutar mejor de la aplicación.

DESCRIPCIÓN:

- La aplicación tiene la siguiente ventana principal:



- donde a la izquierda está la entrada de datos, en ese espacio se ingresa código en el lenguaje Wison.
- El botón analizar manda el texto ingresado y lo analiza.
- Dependiendo de los resultados del análisis en el campo de la salida se agregarán todos los errores detectados, y de no haber errores no se mostrará nada.
- Ejemplo de una entrada:

“

Wison {

Lex {:

/**

Esto es un

Comentario de bloque

*/

Declaración de terminales de la forma:

Terminal \$_NOMBRE <- EXPRESIÓN ;

Terminal \$_Una_A <- 'a' ; # cualquier carácter alfanumérico por separado

Terminal \$_Mas <- '+' ; # cualquier carácter especial por separado

```

Terminal $_Punto    <- '.'; # cualquier carácter especial por separado
Terminal $_P_Ab     <- '('; # cualquier carácter especial por separado
Terminal $_P_Ce     <- ')'; # cualquier carácter especial por separado
Terminal $_FIN      <- 'FIN'; # cualquier palabra reservada
Terminal $_Letra    <- [aA-zZ]; # alfabeto completo en mayúsculas y minúsculas
Terminal $_NUMERO   <- [0-9]; # Dígitos del 0 al 9
Terminal $_NUMEROS  <- [0-9]*; # Estrella de Kleene para hacer 0 o n veces
Terminal $_NUMEROS_2 <- [0-9]+; # Cerradura positiva para hacer 1 o n veces
Terminal $_NUMEROS_3 <- [0-9]?; # reconoce la cláusula ? para hacer 0 o 1 vez
Terminal $_Decimal  <- ([0-9]*)($_Punto)($_NUMEROS_2); # terminal combinado

```

```

:}

```

Syntax { {:

```

# Declaración de no terminales de la forma
# No_Terminal %_Nombre ;

```

```

No_Terminal %_Prod_A;
No_Terminal %_Prod_B;
No_Terminal %_Prod_C;
No_Terminal %_S;

```

```

# Símbolo inicial de la forma
# Initial_Sim %_Nombre ;

```

```

Initial_Sim %_S ;

```

```

# Todo símbolo no terminal debe ser declarado antes de usarse en las producciones
# Las producciones son de la siguiente forma
# %_Initial_Sim <= %_Prod_A ... %_No_terminal_N o $_Terminal_N ... ;

```

```

%_S <= %_Prod_A $_FIN ;
%_Prod_A <= $_P_Ab %_Prod_B $_P_Ce ;
%_Prod_B <= %_Prod_B %_Prod_C | %_Prod_C ;
%_Prod_C <= $_Una_A $_Mas $_Una_A ;

```

```

:}}

```

?Wison

”

La salida de esto será un campo vacío, ya que la entrada es correcta, sin embargo para una entrada como esta:

“

Wison ħ

Lex {:

```
Terminal $_Una_A    <- 'a' ; ; # cualquier carácter alfanumérico por separado
```

```
Terminal $_Mas      <- '+' ; ; # cualquier carácter especial por separado
```

```
Terminal $_Punto    <- '.' ; ; # cualquier carácter especial por separado
```

```
Terminal $_P_Ab     <- '(' ; # cualquier carácter especial por separado
```

;}

Syntax {{:

```
No_Terminal %_Prod_B;
```

```
No_Terminal %_Prod_C;
```

```
No_Terminal %_S;
```

```
Initial_Sim %_S ;
```

```
%_S <= %_Prod_A $_FIN ;
```

```
%_Prod_A <= $_P_Ab %_Prod_B $_P_Ce ;
```

```
%_Prod_B <= %_Prod_B %_Prod_C | %_Prod_C ;
```

```
%_Prod_C <= $_Una_A $_Mas $_Una_A ;
```

:{}}

?Wison

”

nos saldrán los siguientes errores

```
1 Error sintactico linea 4:
...na_A    <---- 'a' ; ; # cualquier caráct
            ^
Esperaba ':', 'TERMINAL(Terminal)', se obtuvo ';'

Error sintactico linea 5:
...as      <---- '+' ; ; # cualquier caráct
            ^
Esperaba ':', 'TERMINAL(Terminal)', se obtuvo ';'

Error sintactico linea 6:
...unto    <---- '.' ; ; # cualquier caráct
            ^
Esperaba ':', 'TERMINAL(Terminal)', se obtuvo ';'

```

Los comandos para compilar la aplicación son las siguientes:

- ng serve: la aplicación enciende el servidor en el puerto 4200, en el <http://localhost:4200/>

La aplicación fue generada en ángular, una herramienta muy poderosa.

TIPOS DE ENTRADAS

- Terminal \$_Una_A;
- No_Terminal %_Prod_B;
- Wison;
- Lex;