

Manual de Usuaio

El siguiente proyecto Generador de Gramaticas Wison utilizando la herramienta jison para poder analizar la gramática generada por el usuario por el formato Wison.

El proyecto web tiene como principal funcionalidad analizar gramáticas en formato Wison donde puede analizar su transcripción y determinara si es ambiguar, si tiene resurvidad por la izquierda para que el usuario la pueda modificar a su criterio.

Para generar gramáticas con Wison es necesario saber su estructura, de deja a disposición del usuario desarrollador la estructura:

#Esto es un comentario de línea

#Estructura Wison

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 mayusculas y minusculas
Terminal $_NUMERO  <- [0-9] ; # Digitos 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 on 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;

```

```

# Simbolo 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 ;

```

:}}

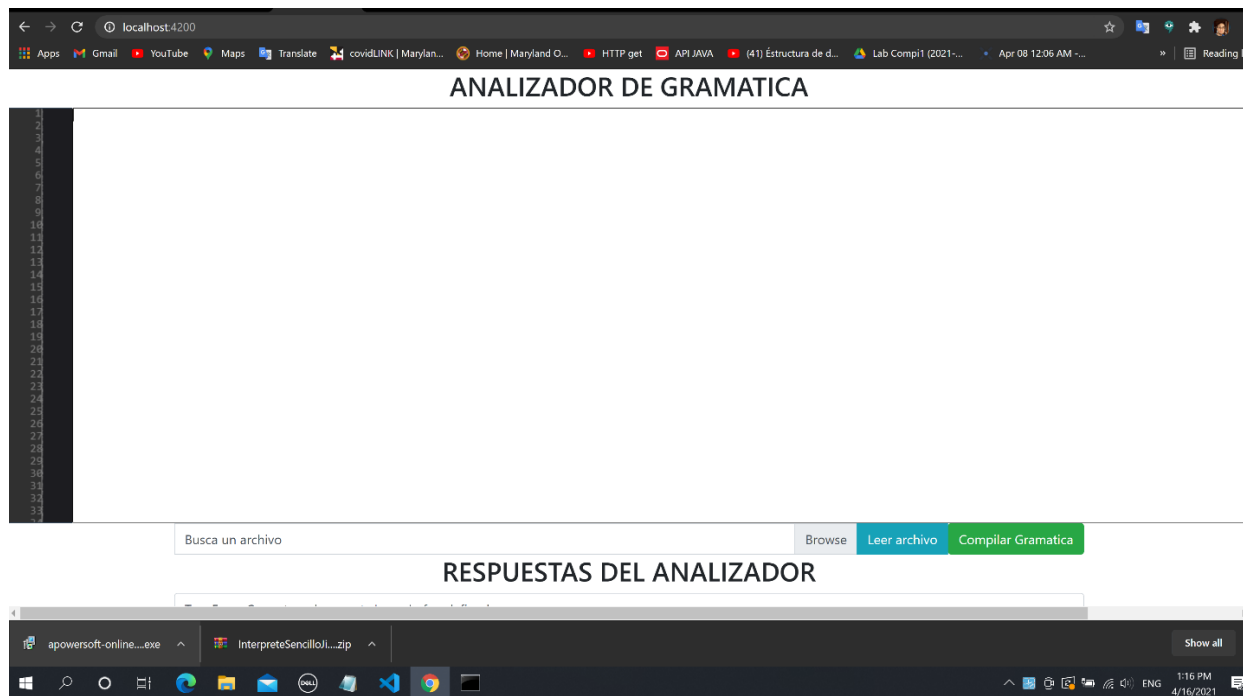
?Wison

Fin de estructura Wison

Para ejecutar la aplicación web se necesita ir a la URL:

<https://localhost:4200>

A continuacion el usuario desarrollador puede visualizar las opciones de :



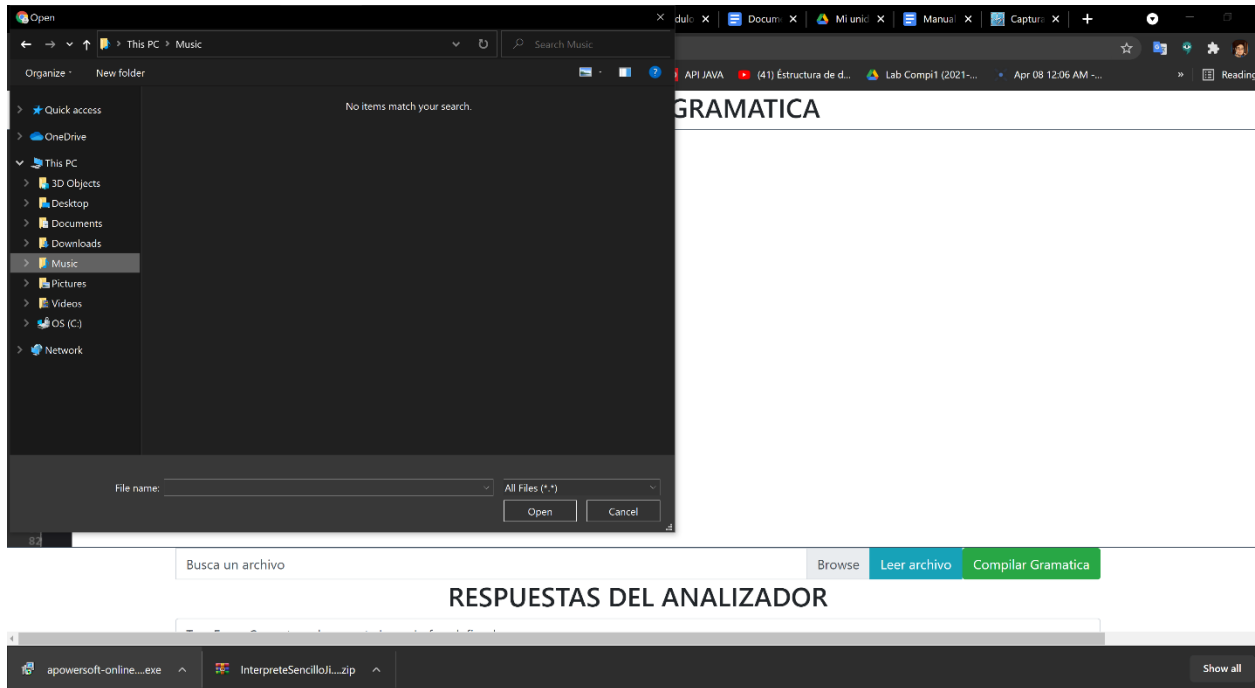
Subir Archivo de texto

Leer Archivo

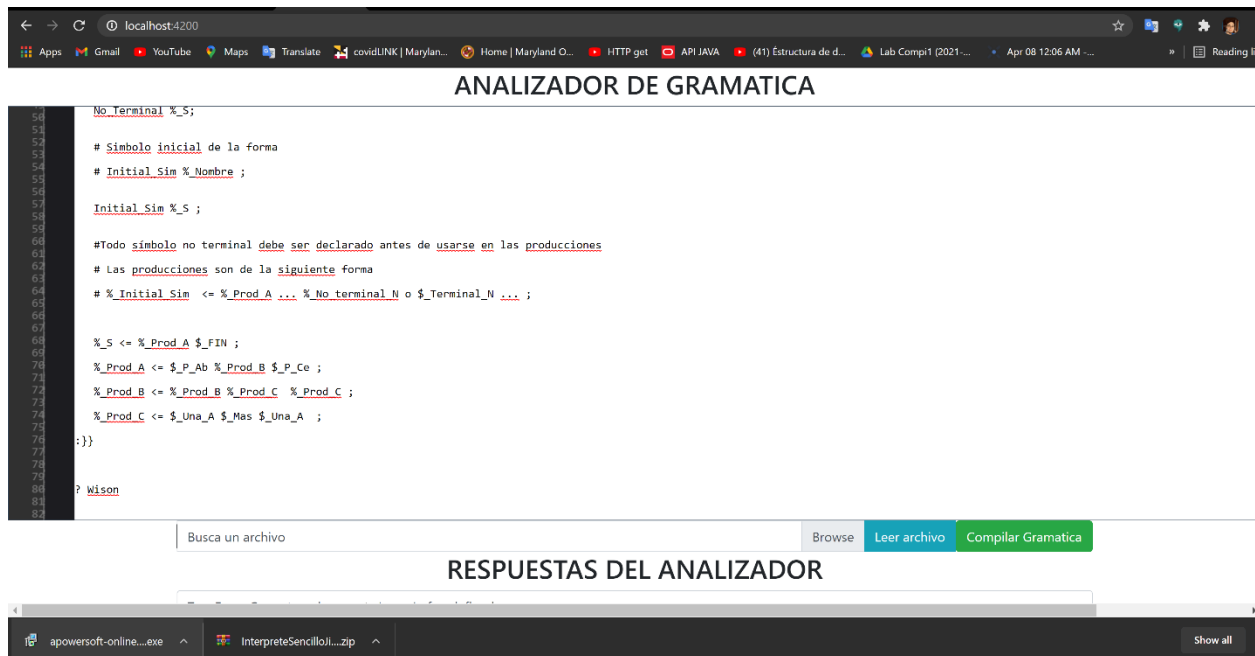
Compilar Gramatica

[Browse](#)[Leer archivo](#)[Compilar Gramatica](#)

El usuario puede buscar un archivo que desea cargar en el Analizador Wison.



Y puede visualizar el explorador de archivos y seleccionarlo



El usuario puede visualizar su archivo cargado de la siguiente manera



analizador dara las respuestas en el siguiente pampo de texto en donde podra informar si tiene ambigüedad, si tiene recursividad por la izquierda y si existen errores lexicos o sintacticos.

La aplicación fue desarrollada por el estudiante James Osmin Gramajo Carcamo con carne 201731172 del curso de Compiladores 1 1er semestre 2021