

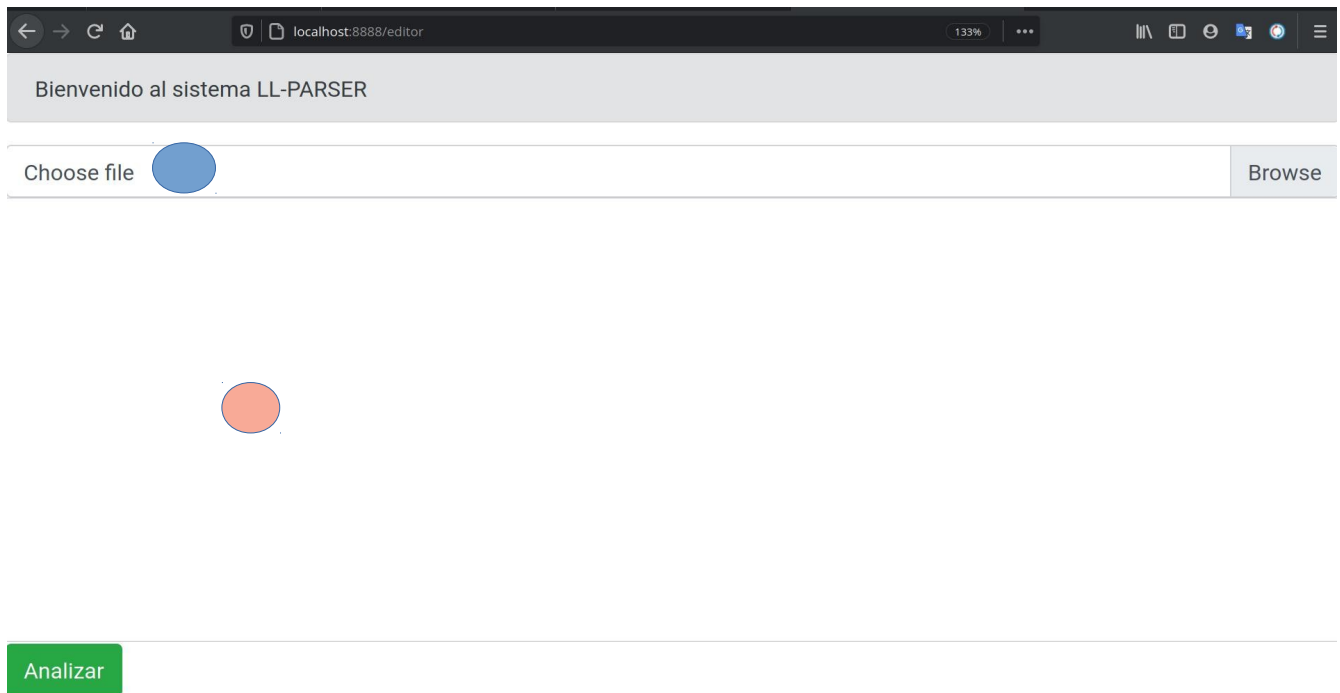
Universidad de San Carlos de Guatemala  
Centro Universitario de Occidente  
Division de Ciencias de la Ingeniería  
Ingeniería en Ciencias y Sistemas

Manua usuario proyecto1: “LL-PARSER”


David Rodolfo Martinez Miranda  
20632145

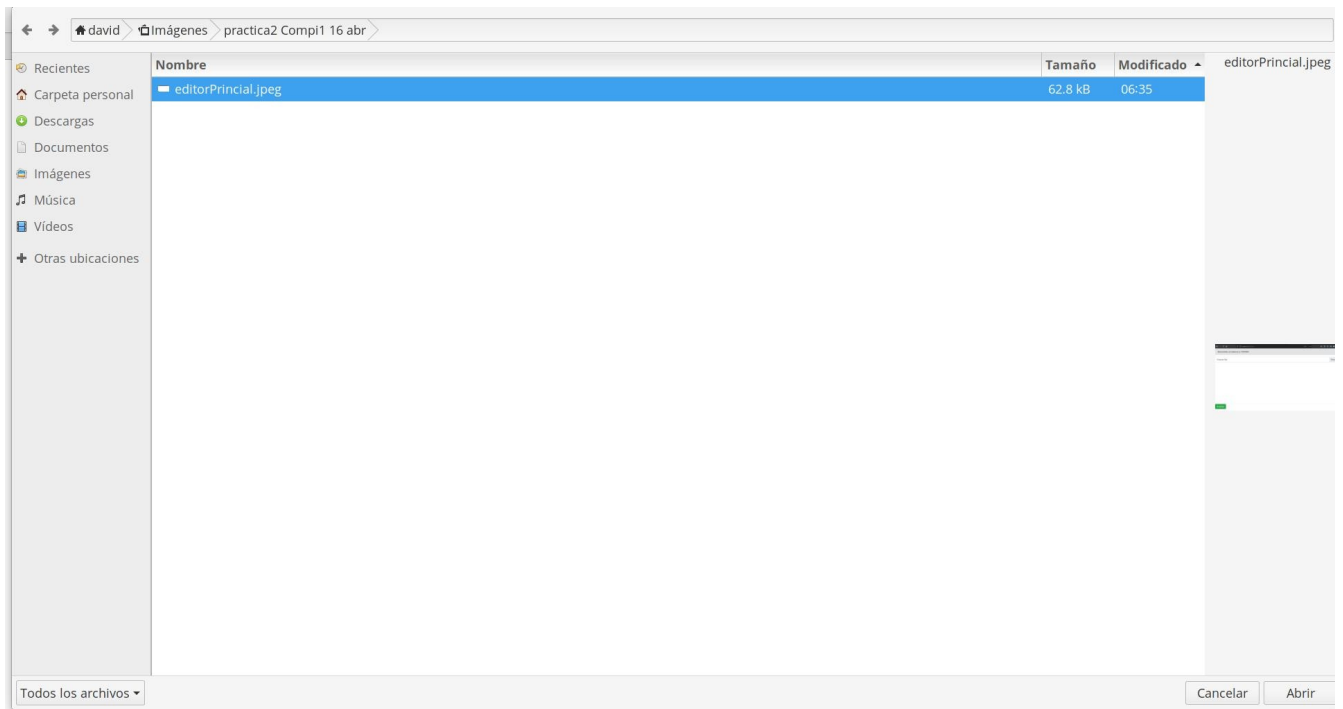
## Manual para el Usuario

Usuario correspondiente gracias por utilizar nuestra app-web LL-PARSER, esta basado en un reconocedor de gramaticas tipo LL, para su analisis correspondiente debe ingresar un formato de entrada que es reconocio por dicho analizador, el formato se indica mas adelante, cuando usted ingrese a la pagina **localhost:8888/editor** tendra la siguiente pantalla.



Se le presenta un area para ingresar texto o poder cargar un archivo.

 Cargar archivo, para que usted pueda cargar el archivo puede dar click en la referencia y se le mostrara el siguiente gestionador basico de archivos.



Usted podra seleccionar el archivo que contenga la estructura requerida por LL-PARSER.

○ Cuando usted cargue el archivo se mosrara el texto en el area de texto.

---

#Esto es un comentario de línea

#Estructura Wison

Wison ζ

Lex {:

/\*\*

Esto es un

Comentario de bloque

\*/

# Declaración de terminales de la forma:

---

## Definicion de gramatica

La estructura basica para reconocer el lenguaje es la siguiente.

- **Comentarios**, para definir comentario se tienen dos formas

- #Esto es un comentario de línea, tiene que iniciar con # seguido de cualquier cosa menos saltos de línea
- /\*\*
- **Esto es un**
- **Comentario de bloque**
- \*/
  - **Esta** definido por el inicio de /\*\* seugido de cualquier cosa menos / o \*, y termina con \*/
- **Inicio de estructura** para iniciar la estrucura se debe seguir el siguiente patrones

Wison ¿

?Wison

Dentro de la estrucutura pueden venir comentarios como tambien afuera de el, para hacer definiciones lexicas se utiliza la siguiente estructura.

- **Definiciones lexicas**

Lex {:

:}

Para definir una expresion regular se debe de seguir la siguiente sintaxis

Terminal \$\_Una\_A   <- 'a' ;

Terminal define que es una terminal el nombre que se le de tiene que se \$\_ seguido de cualquier cosa menos espacios ni saltos de línea, para asignarle un valor se tiene lo siguiente:

- Literal, viene entre comillas ‘literal’
- Caracter, viene en comillas ‘a’

- Alfabeto [aA-zZ]
- Dígito [0-9]

Estas pueden ir acompañadas de las clausuras siguientes

- Cierre \* (cero o más veces)
- Cierre positivo + (una o más veces)
- Cierre ? (Una o cero veces)

Las terminales que se definan en esta área serán utilizadas en el análisis sintáctico que se define más adelante.

## **Análisis sintáctico**

Para hacer el análisis sintáctico se define dentro del siguiente bloque:

```
Syntax { {:
```

:} }

Dentro de este bloque primero debe definir las no terminales a usar, luego el símbolo inicial y seguido las producciones que desee agregar.

**Recuerde que es una gramática LL, por lo tanto debe cumplir estas reglas.**

- No debe ser ambigua.
- No tiene que tener recursividad por la izquierda.
- Debe estar factorizada.

Si la gramática presenta estos problemas, se presentan las siguientes posibles soluciones.

Quitar recursividad por la izquierda

- $A \rightarrow A\alpha$                        $A \rightarrow \beta 1 A'$
- $A \rightarrow A\alpha N$                      $A \rightarrow \beta N A'$
- $A \rightarrow \beta 1$                          $A' \rightarrow \alpha A'$
- $A \rightarrow \beta N$                          $A' \rightarrow \alpha N A'$
- $A' \rightarrow \lambda$

Factorizar por la izquierda

- $A \rightarrow \alpha\beta 1$                      $A \rightarrow \alpha A'$
- $A \rightarrow \alpha\beta 2$                      $A' \rightarrow \beta 1$
- $A \rightarrow \alpha\beta N$                      $A' \rightarrow \beta 2$   
                                       $A' \rightarrow \beta N$

**Declaración de No terminales**, para declarar no terminales se debe seguir la siguiente sintaxis.

No\_Terminal %\_Prod\_A;

Con No\_Terminal se declara el no terminal, y se le asigna el nombre con %\_ seguido de cualquier cosa menos espacios o saltos de línea.

**Declaración inicio símbolo**, para indicar con qué símbolo se inicia se usa la siguiente sintaxis

Initial\_Sim %\_S ;

**Declaración de una producción**, para declarar una producción se utiliza la siguiente forma.

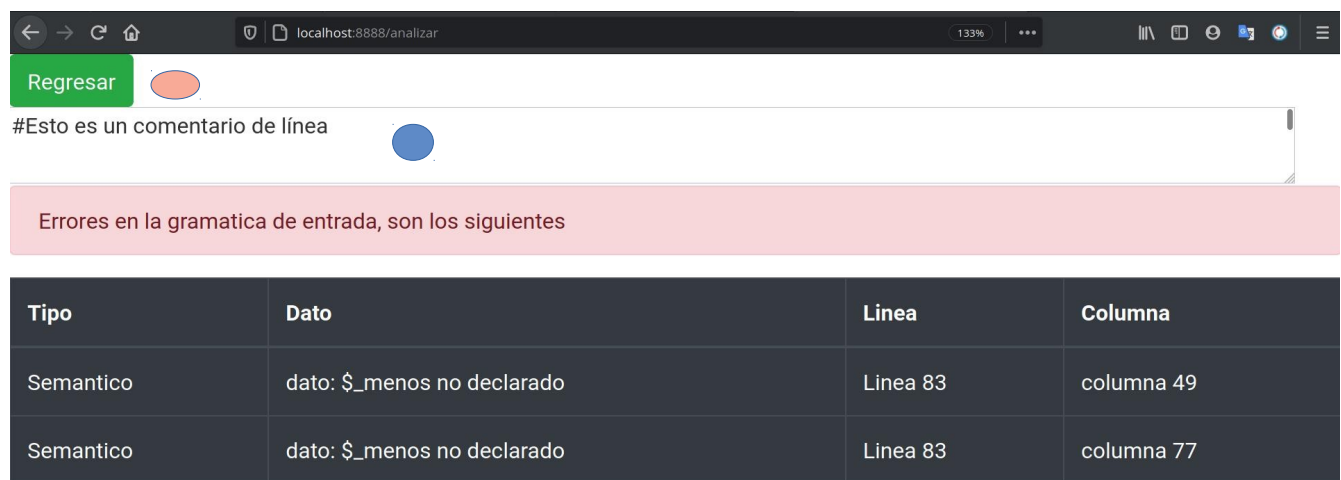
%\_Initial\_Sim <= %\_Prod\_A ... %\_No\_terminal\_N o \$\_Terminal\_N

Se indica que produccion y lo que va a derivar, ademas puede agregar mas subproducciones utilizando

|


**Si desea agregar una produccion lambda debera usar la palabra reservada labmda**

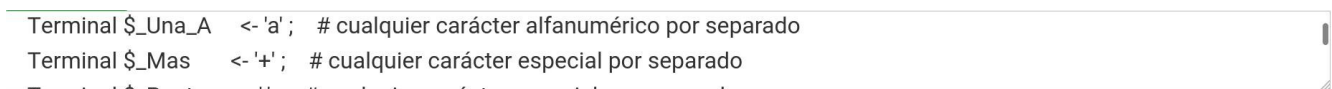
Si todo lo anterior no esta correcto el analizar va a presentar lo siguiente



Tipo	Dato	Linea	Columna
Semantico	dato: \$_menos no declarado	Linea 83	columna 49
Semantico	dato: \$_menos no declarado	Linea 83	columna 77

**En esta parte se indican de errores, lexicos, sintacticos o semanticos**, asi como informacion de el tipo, el dato que es el error y la linea y columna de dicho error.

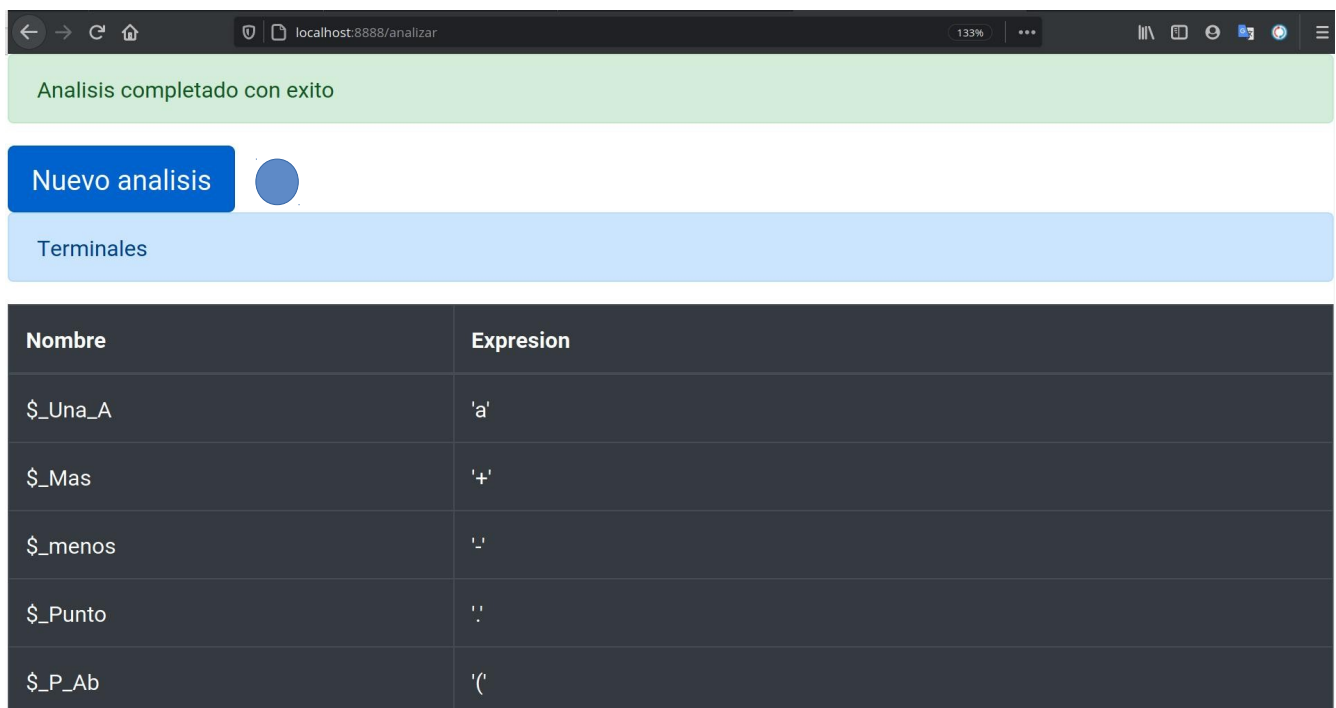
 Regresar le dara la opcion de regresar al editor y poder hacer las correcciones correspondientes



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

- Pequeña area donde podra buscar el error he repararlo sin ir al editor principal, cuando haga esto es necesario que para poder hacer el analisis debe regrear y el texto seguira con las modificaciones que haga en esta parte.

Si por el contrario el analisis tuvo éxito se presentara la siguiente parte.



The screenshot shows a web browser window with the address bar displaying 'localhost:8888/analizar'. The page has a green header bar with the text 'Analisis completado con exito'. Below this, there is a blue button labeled 'Nuevo analisis' and a light blue bar labeled 'Terminales'. At the bottom, there is a table with two columns: 'Nombre' and 'Expresion'.

Nombre	Expresion
\$_Una_A	'a'
\$_Mas	'+'
\$_menos	'-'
\$_Punto	'.'
\$_P_Ab	'('

- Para hacer un nuevo analisis debe, se perdera la informacion del analisis ya realizado.



En esta area tambien se encuentran:

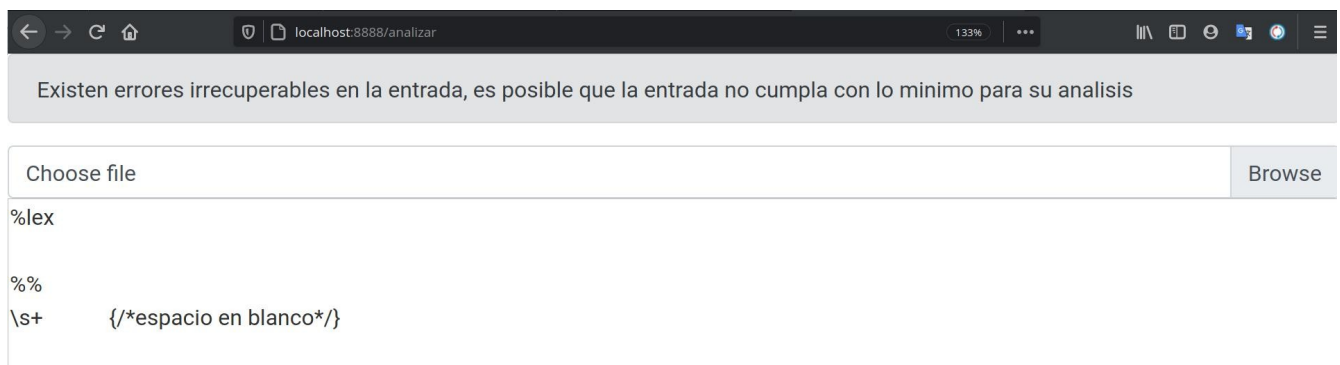
- **Terminales y sus expresiones, como ya se presento en la imagen anterior.**
- **Primeros y siguientes.**

Primeros y Siguients		
No terminal	Primeros	Siguients
%_Prod_A ambigua: false	\$_P_Ab	\$_FIN
%_Prod_B ambigua: true	\$_Una_A,\$_Una_A	\$_P_Ce,\$_Una_A,\$_Una_A,\$_FIN,\$_menos
%_Prod_C ambigua: true	\$_Una_A,\$_Una_A,lambda,\$_FIN,\$_menos	\$_P_Ab,\$_P_Ce,\$_Una_A,\$_Una_A,\$_FIN,\$_menos
%_Prod_D ambigua: false	lambda,\$_FIN	\$_menos,\$_FIN
\$_S ambigua: false	\$_P_Ab	\$

- **Movimientos con base a la tabla de analisis sintactico.**

Listado de mueve con terminal a no terminal		
Prod	Mueve con	Hacia
%_Prod_A	---	---
->	\$_P_Ab	\$_P_Ab,%_Prod_B,\$_P_Ce
%_Prod_B	---	---
->	\$_Una_A	%_Prod_B,%_Prod_C,\$_P_Ab
->	\$_Una_A	%_Prod_B,%_Prod_C,\$_P_Ab
->	\$_Una_A	%_Prod_C

Si por alguna razon no se sigue ni el minimo procedimiento para la estructura se mostrara la siguiente parte.



-----Final manual usuario -----