



**LCompiler**

---

**Manual de Usuario**

Versión 1.0  
Fecha: 03/09/2020

## **Tabla de Contenidos**

## **1. Descripción del sistema.**

### **1.1. Objetivo**

El software fue desarrollado con la finalidad de ser utilizado como una herramienta flexible capaz de procesar archivos con extensión len cuyo contenido define reglas léxicas, sintácticas y semánticas y así producir un interprete capaz analizar texto y sentenciar las respectivas acciones si el análisis ejecutado es correcto.

### **1.2. Alcance**

La aplicación de escritorio ha sido desarrollada para todas aquellas personas que se encuentran dentro del mundo de la programación, mas específicamente de los lenguajes formales y los compiladores y de sean utilizar una herramienta que les permita crear sus propio lenguajes de alto nivel.

### **1.3. Funcionalidad**

La aplicación basa su funcionalidad en la lectura de archivos .len, los cuales al ser correctos generan un nuevo interprete que es agregado al repositorio de la aplicación para que pueda ser utilizado en cualquier momento.

Posee la funciones básicas de un editor de texto:

- Nuevo
- Abrir
- Guardar
- Guardar Como

Y las funciones específicas de los lenguajes:

- Cargar lenguaje
- Borrar lenguaje
- Compilar
- Ver Tabla LALR
- Ver Pila

Todas se encuentran integradas dentro de un mismo entorno intuitivo y minimista que sera descrito posteriormente.

### **1.4. Instalación**

El software es una aplicación ejecutable por lo cual no necesita de una instalación previa, para su uso se recomienda:

- Crear una nueva carpeta en la ubicación deseada.
- Guardar dentro de la carpeta creada el archivo LCompiler-jar-with-dependencies.jar.

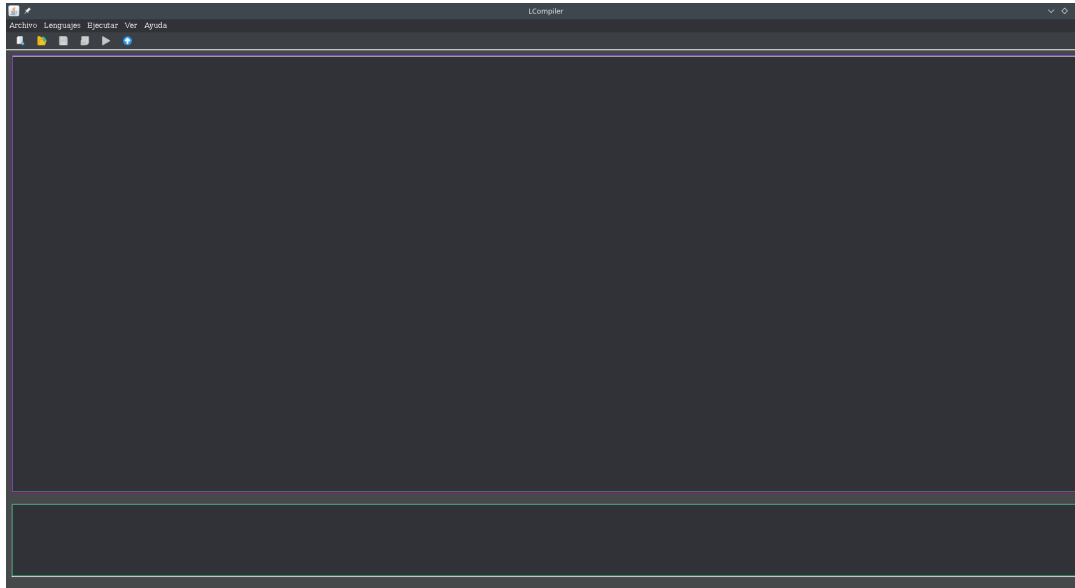
Al terminar los pasos descritos solamente se necesita ingresar a la carpeta creada y dar doble click izquierdo sobre el archivo LCompiler-jar-with-dependencies.jar para poder ejecutarlo.

Dentro de la carpeta se creara una nueva carpeta llamada Lenguajes, esta es utilizada por el programa para almacenar los interpretes generados.

## 2. Especificación del sistema.

El software se divide en 4 secciones principales:

- Barra de menús.
- Área de texto.
- Área de mensaje.
- Barra de herramientas.

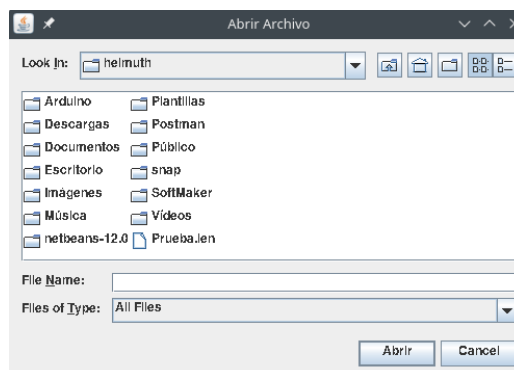


### 2.1. Barra de Menús

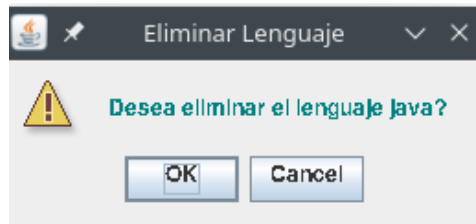
Permite la selección de alguna funcionalidad del sistema, cada submenu posee un shortcut para poder acceder a el de forma mas eficiente(consultar anexo). Las funciones disponibles son las siguientes:

Archivo Lenguajes Ejecutar Ver Ayuda

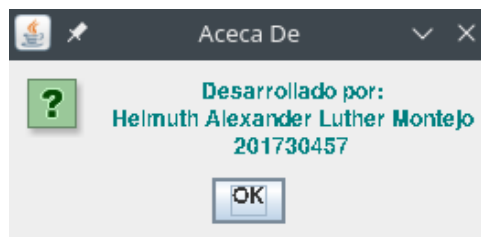
- Menú Archivo
  - ★ **Nuevo:** Despliega una nueva pestaña, en la cual el usuario puede editar un archivo.
  - ★ **Abrir:** Permite abrir un archivo para que este pueda ser editado.
  - ★ **Guardar:** Permite guardar un archivo en la ubicación que el usuario desea.
  - ★ **Guardar Como:** Esta opción permite guardar un archivo previamente almacenado con diferente nombre.
  - ★ **Salir:** Con esta opción se cierra la aplicación.



- **Menú Ejecutar**
  - ★ **Compilar:** A través de esta opción se podrán compilar la o las cadenas de entrada ingresadas por el usuario. Previamente el usuario deberá seleccionar un lenguaje para compilar dichas cadenas. El contenido que se tomará en cuenta para realizar este proceso será el que esté en la pestaña activa.
  - ★ **Cargar Lenguaje:** Por medio de esta opción se podrán agregar nuevos lenguajes al repositorio de la aplicación, los cuales posteriormente aparecen en el menú principal para que puedan ser seleccionados por el usuario para realizar el proceso de compilación. El lenguaje no podrá ser cargado a la aplicación sin estar libre de errores (léxicos, sintácticos y semánticos). El contenido que se tomará en cuenta para realizar este proceso será el que esté en la pestaña activa. Si ya existe una carga previa del mismo lenguaje, entonces se debe reemplazar por la carga más reciente.
  - ★ **Borrar Lenguaje:** Por medio de esta opción se podrá eliminar un lenguaje del repositorio de la aplicación el cual también deberá ser eliminado del menú principal.



- **Menú Lenguajes**  
En este menú muestra todos los lenguajes que la aplicación soporta y permite seleccionar el lenguaje con el que se va a trabajar.
- **Menú Ver**
  - ★ **Tabla LALR:** Esta opción despliega la tabla LALR que fue generada en el momento que el usuario seleccionó un lenguaje para compilar una cadena de entrada.
  - ★ **Pila:** Esta opción despliega en pantalla el contenido de la pila que se utilizó para reconocer una cadena de entrada previamente ingresada por el usuario. En caso contrario la aplicación muestra un mensaje indicando que no se ha reconocido ninguna cadena.
- **Menú Ayuda**  
Acerca De: Muestra información acerca del desarrollado de la aplicación.

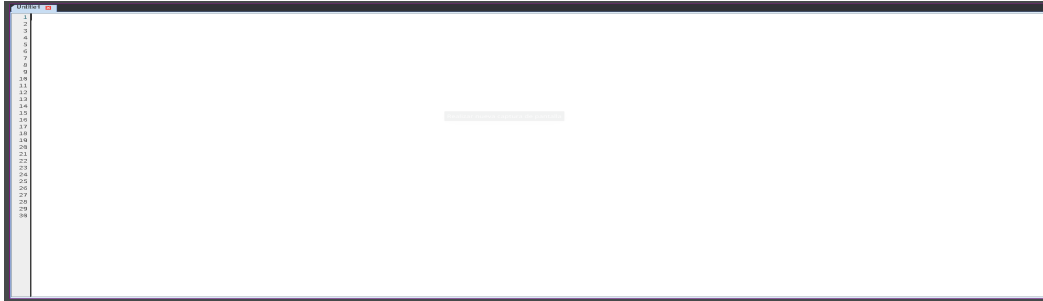


## 2.2. Área de texto

Editor de texto sencillo que permite el ingreso de texto, pose con un indicador de numero de linea en la parte izquierda. Cuenta con un indicador de numero de linea y numero de columna en la parte inferior derecha de la aplicación

Lin: 1 Col: 0

Permite la apertura de múltiples pestañas, cada pestaña tiene en la parte superior de selección el nombre que permite una fácil identificación y navegación entre pestañas.



## 2.3. Área de mensajes

Se encuentra ubicada en la parte inferior, debajo de área de texto, como su nombre lo indica es utilizada para la impresión de mensajes, los mensajes que se verán en esta área son:

- Reporte de errores léxicos, sintácticos y semánticos producidos tras la lectura de un archivo con extensión len.
- Reporte de errores léxicos, sintácticos y semánticos producidos tras el análisis de un texto a través uno de los interpretes disponibles.
- Acciones ejecutadas tras en análisis de un texto a través uno de los interpretes disponible.

```
Token: [palabra] Lexema: [nombre] Filo: 1 Columna: 0
Token: [Error] Lexema: [] Filo: 1 Columna: 6
Token: [palabra] Lexema: [java] Filo: 1 Columna: 8
Token: [Error] Lexema: [] Filo: 2 Columna: 12
Token: [palabra] Lexema: [version] Filo: 2 Columna: 0
Token: [Error] Lexema: [] Filo: 2 Columna: 7
Token: [real] Lexema: [11.8] Filo: 2 Columna: 9
```

Debajo de esta área existe un espacio donde se muestran mensaje varios en la ejecución de la aplicación.

Guardado en: /home/helmuth/Prueba.len

## 2.4. Barra de Herramientas

Brinda acceso de forma rápida a la ejecución de acciones dentro de la aplicación, en esta barra se encuentran las opciones:

- Nuevo
- Abrir
- Guardar
- Guardar Como
- Compilar
- Cargar Lenguaje



## 3. Especificación de la sintaxis del archivo de entrada.

### 3.1. Estructura del archivo

La estructura del archivo para agregar un nuevo lenguaje, está compuesta por cinco secciones: información del lenguaje, código fuente, expresiones regulares, símbolos terminales y no terminales, finalmente se define la gramática.

Información Lenguaje

%%

Código fuente

%%

Expresiones Regulares

%%

Símbolos Terminales y No Terminales

%%

Gramática

Se puede realizar comentarios en cualquier sección del archivo usando de la siguiente forma:

- Una línea: Esto permite hacer un comentario en una línea.  
// Este es un comentario
- Varias líneas: esto permite realizar comentarios en una o más líneas del archivo.  
/\* Este es un comentario \*/

### 3.2. Información Lenguaje

En esta sección se deberá de insertar información acerca del lenguaje que se está definiendo en el archivo, los campos de esta sección son los siguientes:

- Nombre: Nombre del lenguaje
- Versión: la versión del lenguaje
- Autor: el autor del lenguaje
- Lanzamiento: el año del lanzamiento del lenguaje.
- Extensión: extensión de los archivos de cada lenguaje.

Esta información es crucial cuando se lee el archivo, ya que mediante estos campos se obtendrá el nombre del lenguaje y que posteriormente se podrá seleccionar identificándose con el nombre indicado en esta sección, se podrá consultar también la información sobre cada lenguaje y básicamente consiste en mostrar esta información al usuario.

El único campo obligatorio de esta sección es nombre, los demás datos son opcionales.

### 3.3. Sección de Código Fuente

En esta sección deberá ir el código fuente java de las funciones o procedimientos que serán utilizados en las reglas semántica. Esta sección es opcional, es decir puede o no venir en el archivo de entrada.

### 3.4. Sección de Expresiones Regulares

En esta sección deberán ser declaradas las expresiones regulares para establecer la forma que tendrá cada token que es válido para un determinado lenguaje.

$\alpha?$  ->  $\alpha$  puede venir o no puede venir.  
 $\alpha^*$  ->  $\alpha$  puede venir cero o más veces.  
 $\alpha^+$  ->  $\alpha$  puede venir una o más veces.  
 $\alpha\beta$  -> Concatenación implícita entre  $\alpha$  y  $\beta$ .  
 $\alpha|\beta$  -> Puede venir  $\alpha$  o  $\beta$ .  
[0-9] -> Cualquier número entre 0 y 9.  
[a-z] -> Cualquier letra entre a y z.  
 $\backslash n$  -> Nueva línea.  
 $\backslash t$  -> Tabulador.  
 $\backslash b$  -> Espacio en blanco.  
“ $\alpha$ ” -> Agrupa a uno o más caracteres.  
( ) ó [ ] -> Agrupa expresiones regulares.

La sintaxis de esta sección está compuesta por un identificador y la expresión regular que corresponde al identificador.

- Identificador: el identificador será el nombre que tendrá cada token que cumpla con la expresión regular especificada.
- Expresión Regular: Permite establecer la forma que tendrá cada token que es válido para el lenguaje que se está definiendo.

$\langle \text{identificador1} \rangle = \langle \text{expresión regular1} \rangle;$   
 $\langle \text{identificador2} \rangle = \langle \text{expresión regular2} \rangle;$

$\& = [\backslash n \backslash t];$  /\* Significa que cuando se encuentre este token deberá ser ignorado \*/

Se puede emplear el identificador “ $\&$ ” para indicar que la expresión regular deberá ser ignorada, cuando se analice un archivo.

### 3.5. Sección de símbolos terminales y no terminales

En esta sección se deberán declarar los símbolos terminales y no terminales del lenguaje que se esté definiendo.

Sintaxis:

$\langle \text{no terminal} \mid \text{terminal} \rangle [\text{entero} \mid \text{real} \mid \text{cadena}] \text{id1}[, \text{id2}, \dots, \text{idn}];$

### 3.6. Sección de gramática

En esta sección se definirá la gramática del lenguaje y las reglas semánticas asociadas a cada producción. Las reglas semánticas van encerradas entre llaves al final de la producción y antes del punto y coma, y estas son opcionales para cada producción, este código se deberá de ejecutar al momento de reducir la producción. Cada símbolo No Terminal y Terminal puede tener una estructura o variable asociada del tipo declarado previamente, la cual se declarará con “:nombre” y existirá un atributo RESULT asociado a cada lado izquierdo de las producciones.

Sintaxis:

$\text{no terminal} :: [\langle \text{terminal} \mid \text{no terminal} \rangle : \text{nombre}] \dots [\{ \text{reglas semánticas} \}];$





*LCompiler*

**LCompiler  
Manual de Usuario**