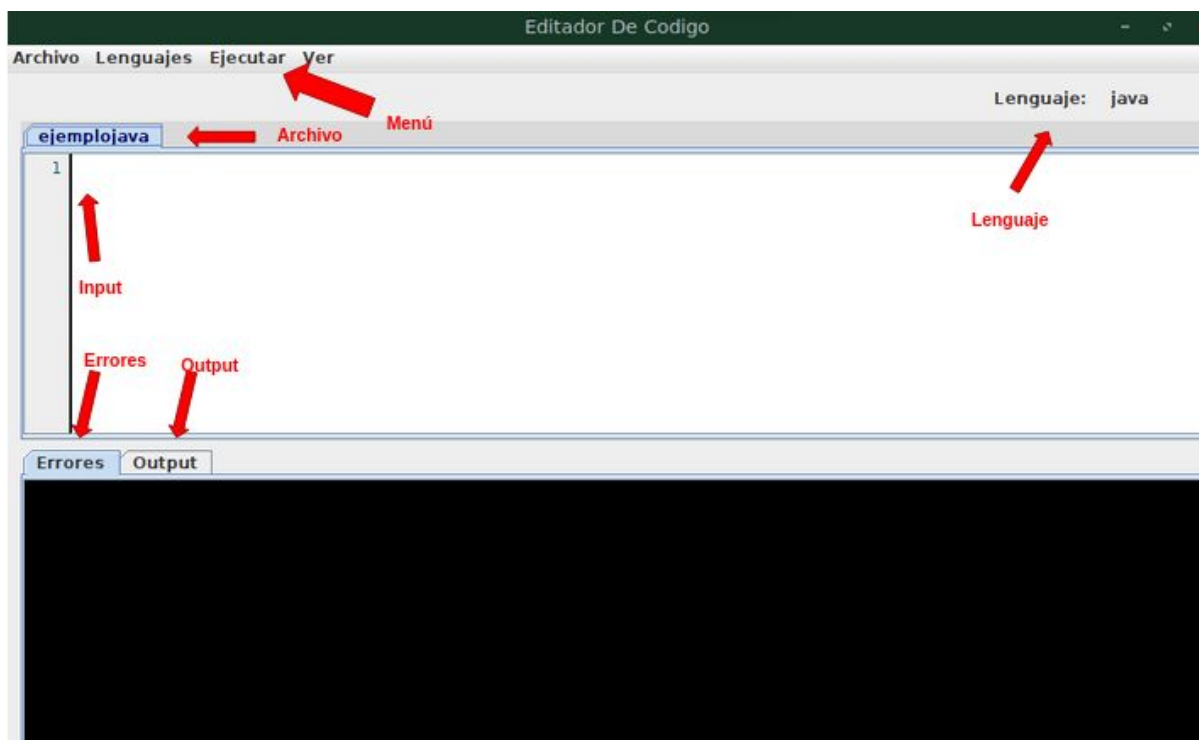


Manual De Usuario (Compilador De Lenguajes)

Antes de entrar a uso y funcionamiento de la aplicación, si tienes algún problema con la instalación y ejecución del programa puedes pasar por el “Manual De Instalación y Desinstalación (Compilador De Lenguajes)”.

Al iniciar la aplicación nos aparece primero la pantalla principal, la cual contiene varios elementos con distintas funcionalidades que se explican posteriormente.



1.Menu:

Para la aplicación existen cuatro menús para diferentes opciones, el primero es el de Archivo en donde se encuentra lo siguiente:

- Nuevo: Despliega una nueva pestaña, en donde se puede crear un archivo nuevo de un lenguaje y nombre especificado.
- Abrir: Permite abrir un archivo para que este pueda ser editado.
- Guardar: Permite guardar un archivo en la ubicación que el usuario desea, el archivo deberá de guardarse con la extensión del lenguaje seleccionado.
- Guardar Como: Esta opción permite guardar un archivo previamente almacenado con diferente nombre, el archivo deberá de guardarse con la extensión del lenguaje seleccionado.
- Salir: Con esta opción se cierra la aplicación.

Luego existe la opción de Lenguaje en donde permitirá ver un listado de lenguajes existentes las cuales se pueden usar para la compilación de código. En donde se encuentra el nombre, autor, fecha de lanzamiento, versión y extensión si se encuentra con la información. Además las producciones que conforman el lenguaje.



SubMenús:

- **Compilar:** A través de esta opción se podrán compilar la o las cadenas de entrada ingresadas en el espacio de texto del archivo. La compilación dependerá del lenguaje que se escogió al momento de crear el archivo así será la compilación.
- **Cargar Lenguaje:** Por medio de esta opción se podrán agregar nuevos lenguajes al repositorio de la aplicación, los cuales posteriormente deberán aparecer en el menú principal para que puedan ser seleccionados por el usuario para realizar el proceso de compilación. Se deberá tomar en cuenta que el lenguaje no podrá ser cargado a la aplicación sin estar libre de errores (léxicos, sintácticos y semánticos). Para cargar un lenguaje, aparecerá una pestaña con un buscador de archivos del equipo, tendrá que elegir un archivo que represente el lenguaje con extensión (.len). 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. Para esto se mostrará el conjunto de lenguajes existentes para que se pueda elegir el que se desee eliminar.

Ver

- Tabla LALR: Este opción mostrará una pestaña con la tabla LALR, la cual fue generada en el momento de cargar un lenguaje.

Tabla LALR: java											
	entero	mas	palabra	parenA	parenB	por	real	\$	E	F	T
1	s6			s5					g2	g4	g3
2		s7						A			
3		r2				s8		r2			
4		r4				r4		r4			
5	s13			s12					g9	g11	g10
6		r6				r6		r6			
7	s6			s5						g4	g14
8	s6									g15	
9		s17			s16						
10		r2			r2	s18					
11		r4			r4	r4					
12	s13			s12					g19	g11	g10
13		r6			r6	r6					
14		r1				s8		r1			
15		r3				r3		r3			
16		r5				r5		r5			
17	s13			s12						g11	g20
18	s13									g21	
19		s17			s22						
20		r1			r1	s18					
21		r3			r3	r3					
22		r5			r5	r5					

- Pila: Esta opción deberá despliega una pantalla con el contenido de la pila que se utilizó para reconocer la cadena de entrada que se compiló, dependiendo del lenguaje que se eligió.

#	Pila	Simbolos	Entrada	Accion
1	1		entero +2	Desplazar
2	1 6	entero	mas 2	Reducir Mediante F -> entero
3	1 4	F	mas 2	Reducir Mediante T -> F
4	1 3	T	mas 2	Reducir Mediante E -> T
5	1 2	E	mas 2	Desplazar
6	1 2 7	E mas	entero	Desplazar
7	1 2 7 6	E mas entero	\$	Reducir Mediante F -> entero
8	1 2 7 4	E mas F	\$	Reducir Mediante T -> F
9	1 2 7 14	E mas T	\$	Reducir Mediante E -> E mas T
10	1 2	E		Aceptar

2.Archivo

Aquí se muestra el conjunto de archivos con los que se está trabajando, indicando su nombre y su extensión. Se podrá navegar entre archivos con la simple acción de darle click a los tabs con los nombres.

3.Input

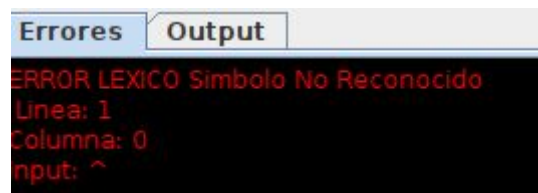
En este espacio se podrá escribir código conforme a la gramática del archivo que se creo, este código será el que se ejecute al momento de compilar, cada archivo que se encuentra abierto tendrá su propio input.

4.Lenguaje

En este espacio aparecerá el lenguaje con el que se está trabajando, dependiendo del archivo que se encuentra abierto.

5. Errores

Aquí se mostrarán los posibles errores si existieran al cargar un lenguaje o al compilar código, indicará la línea y la columna del texto en donde se genera el error.



6. Output

Aquí se mostrará la salida de la ejecución de código que se compilo, si es que no existieron errores.

Información sobre el archivo del lenguaje a cargar:

El archivo (.len) deberá estar conformado por 5 partes separadas por (%%), estas partes son las siguientes:

1. Información Lenguaje:

Aquí se tiene que describir atributos del lenguaje como nombre, versión, autor, lanzamiento, como la siguiente manera por ejemplo:

nombre: java;
versión: 11.0.7;
autor: sun Microsystems;
lanzamiento: 1996;
extensión: java;

2.Código Fuente:

En este espacio deberá ir el código fuente java de las funciones o procedimientos que serán utilizados en la gramática del lenguaje.

3.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, indicando su nombre y la expresión que lo representa. Se puede emplear el identificador "&" para indicar que la expresión regular deberá ser ignorada, cuando se analice un archivo.

4. Símbolos Terminales y No Terminales

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

5. 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.

Ejemplo De Entrada:

//Información del lenguaje

```
nombre: java;  
version: 11.0.7;  
autor: sun Microsystems;  
lanzamiento: 1996;  
extension: java
```

%% //codigo fuente

```
List<String> listString=new ArrayList<>();
```

```
public void addString(String valor){  
    listString.add(valor);  
}
```

```
public Integer suma(Integer numero1,Integer numero2){  
    return numero1+numero2;  
}
```

%% //Sección de expresiones regulares

```
palabra = [a-z]+;  
entero = [0-9]+;  
real = [0-9]+((.)([0-9]+))?;  
más = “+”;  
menos = “-”;  
& = [\n\t]; /* Significa que cuando se encuentre este token deberá ser ignorado */
```

%% //Sección de símbolos terminales y no terminales

```
terminal por, div;  
terminal más, menos;  
terminal entero;
```

```
no terminal A, B;
```

no terminal S;

%%

S :: E:val {println("Resultado = "+val);};

E :: E:val menos E:val2 {RESULT=val - val2};;

E :: E:val mas E:val2 {RESULT=val + val2};;

E :: E:val por E:val2 {RESULT=val * val2};;

E :: E:val div E:val2 {RESULT=val / val2};;

E :: entero:val {RESULT=val};;