

# **MANUAL TÉCNICO**

## **Creación de un compilador**

**Jhonny García Hernández**

**Versión 1.0**

## **OBJETIVOS**

- General

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo.

- Específicos

1. Representar la a funcionalidad técnica de la estructura, diseño y definición del aplicativo.
2. Definir claramente el procedimiento de instalación del aplicativo.
3. Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
4. Describir las herramientas utilizadas para el diseño y desarrollado del prototipo.

## **INTRODUCCIÓN**

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas, pueda realizar la instalación del software creado para el desarrollado de prototipos de lenguaje de programación basado en gramáticas ascendente.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación del aplicativo.

## **REQUISITOS**

Requisitos para el IDE (Opcional):

### Configuraciones de Hardware Mínimas

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:
- Procesador: intel Pentium III o equivalente a 800 MHz
- Memoria: 512 MB
- Espacio en disco: 750 MB de espacio libre en el disco
- Ubuntu 9.10:
- Procesador: Intel Pentium III o equivalente a 800 MHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Solaris OS versión 10 (SPARC):
- Procesador: UltraSPARC II a 450 MHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Solaris OS versión 10 (x86/x64 Platform Edition):
- Procesador: AMD Opteron serie 1200 a 1,8 GHz
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco
- Macintosh OS X 10.5 Intel:
- Procesador: Intel Dual-Core (32 o 64 bits)
- Memoria: 512 MB
- Espacio en disco: 650 MB de espacio libre en el disco

## Configuraciones de Hardware Recomendadas

- Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:

- Procesador: Intel Pentium IV o equivalente a 2,6 GHz

- Memoria: 2 GB

- Espacio en disco: 1 GB de espacio libre en el disco

- Ubuntu 9.10:

- Procesador: Intel Pentium IV o equivalente a 2,6 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- Solaris OS versión 10 (SPARC):

- Procesador: UltraSPARC IIIi a 1 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- Solaris OS versión 10 (x86/x64 Platform Edition):

- Procesador: AMD Opteron serie 1200 a 2,8 GHz

- Memoria: 2 GB

- Espacio en disco: 850 MB de espacio libre en el disco

- OpenSolaris 2010.03 (x86/x64 platform edition):

- Procesador: AMD Opteron serie 1200 a 2,8 GHz

- Memoria: 2 GB

- Espacio en disco: 650 MB de espacio libre en el disco

- Macintosh OS X 10.6 Intel:

- Procesador: Intel Dual-Core (32 o 64 bits)

- Memoria: 2 GB
- Espacio en disco: 850 MB de espacio libre en el disco

### Configuraciones de Hardware Mínimas (Compilador)

Pentium III o equivalente, 1 ghz o superior

- 512 MB RAM, se recomienda 1 GB o más
- 1,5 GB NTFS o 3 GB FAT o más de espacio disponible en disco
- Windows 2000 SP4 o superior

### **INSTALARCIO (IDE)**

Para instalar el software:

Microsoft Windows, Solaris OS y Linux

1. Una vez finalizada la descarga, ejecute el instalador.
  - El archivo del instalador de Windows tiene la extensión .exe. Haga doble clic en él para ejecutarlo.
  - El archivo del instalador de las plataformas Solaris y Linux tiene la extensión .sh. En estas plataformas debe convertir los archivos del instalador en ejecutables mediante el siguiente comando: `chmod +x <nombre-archivo-instalador>`
2. Para seleccionar las herramientas y tiempos de ejecución que se van a instalar, siga estos pasos en la página de bienvenida del asistente de instalación:
  - Haga clic en Personalizar.
  - En el cuadro de diálogo Personalizar la instalación, realice las selecciones.
  - Haga clic en Aceptar.
3. En la página de bienvenida del asistente para la instalación, haga clic en Siguiente.

4. En la página del contrato de licencia, revise el contrato, marque la casilla de aceptación y haga clic en Siguiente.
5. En la página de instalación de NetBeans IDE, siga estos pasos:
  - Acepte el directorio de instalación predeterminado de NetBeans IDE o especifique otro directorio. Nota: el directorio de instalación debe estar vacío y el perfil de usuario que utilice para ejecutar el instalador debe disponer de permisos de lectura/escritura en dicho directorio.
6. Haga clic en Instalar para comenzar la instalación.

Nota: si surgen problemas para instalar el software correctamente, consulte Resolución de problemas para obtener descripciones y sugerencias para solucionar problemas no resueltos que pueden afectar al proceso de instalación.

## Mac OS X

Para instalar el software:

- Una vez finalizada la descarga, ejecute el instalador. La extensión del archivo del instalador es .dmg.
- En los paneles que se abren, haga clic en el icono del paquete. La extensión del paquete es .mpkg. El asistente de instalación comienza.
- En la página de bienvenida del asistente para la instalación, haga clic en Continuar.
- Revise el contrato de licencia y haga clic en Continuar. Haga clic en Aceptar en la ventana emergente para aceptar la licencia.
- En la página de selección de destino, elija la unidad y haga clic en Continuar.
- Si ha descargado los paquetes de "All" (Todo), Java o Ruby, puede personalizar la instalación. En el último panel del asistente de instalación, pulse el botón Personalizar de la parte inferior izquierda del panel. Aparece el árbol de productos.
- Seleccione los productos que desea instalar.

- Especifique el nombre y la contraseña del administrador del sistema y haga clic en Aceptar para comenzar la instalación.

## **INSTALACIÓN (JDK11)**

Para la instalación de la maquina virtual, para este aplicativo se uso el JDK11 por lo que se recomienda, dependiendo de su sistema operativo puede descargarlo en el siguiente link:

<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

Asi mismo al momento de descargarlo, si su computador tiene un software de instalación como es windows, o linux para archivos .deb , puede proceder a instalarlo de manera facil, de lo contrario leer la documentación en el archivo descargado.

## **CONFIGURACIÓN DEL IDE PARA CREAR LAS GRAMÁTICAS**

### **JLEX (Análisis léxico):**

JLex es un generador de analizadores léxicos, escrito en Java, para Java. JLex fue desarrollado por Elliot Berk en la Universidad de Princeton. Para más información visitar la página oficial de JLex:

<https://www.cs.princeton.edu/~appel/modern/java/JLex/>

La principal tarea de un analizador léxico es leer los caracteres de entrada del programa fuente, agruparlos en lexemas y producir como salida una secuencia de tokens.

- Un token es un par que consiste en un nombre de token y un valor de atributo opcional.
- Un lexema es una secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de este tóken.
- Un patrón es una descripción de la forma que pueden tomar los lexemas de un token.

En Jlex se definen los patrones de los diferentes tokens que se desean reconocer, estos patrones pueden definirse a través de expresiones regulares. Además Jlex cuenta con múltiples opciones, una muy importante es su capacidad para integrarse con generadores de analizadores sintácticos como Cup

## Cup:

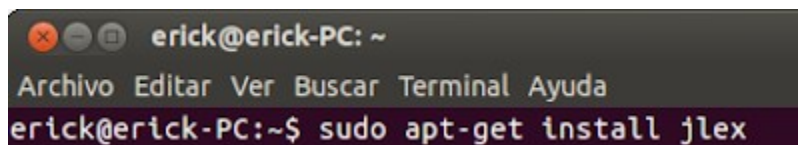
Cup es un generador de analizadores sintácticos de tipo LALR para Java. Para más información visitar la página oficial de Cup:

<http://www2.cs.tum.edu/projects/cup/>

El analizador sintáctico obtiene una cadena de tokens del analizador léxico y verifica que dicha cadena pueda generarse con la gramática para el lenguaje fuente. Una gramática proporciona una especificación precisa y fácil de entender de un lenguaje de programación.

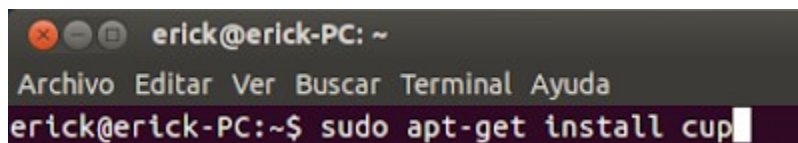
## Instalación de jlex y cup:

Lo primero que haremos será instalar JLex, para ello abrimos una terminal, en Ubuntu puede hacerse con la combinación de teclas Ctrl+Alt+t o en Aplicaciones → Accesorios → Terminal, una vez abierta la terminal ingresamos el comando “sudo apt-get install jlex”, autenticamos ingresando nuestra contraseña y aceptamos la descarga e instalación, con esto quedará instalado JLex.

A screenshot of a terminal window titled 'erick@erick-PC: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command prompt shows 'erick@erick-PC:~\$' followed by the command 'sudo apt-get install jlex' which is currently being typed.

```
erick@erick-PC: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
erick@erick-PC:~$ sudo apt-get install jlex
```

Luego instalamos cup, ejecutando en la terminal el comando “sudo apt-get install cup”, autenticamos ingresando nuestra contraseña y aceptamos la descarga e instalación, con esto quedará instalado Cup.

A screenshot of a terminal window titled 'erick@erick-PC: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command prompt shows 'erick@erick-PC:~\$' followed by the command 'sudo apt-get install cup' which is currently being typed.

```
erick@erick-PC: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
erick@erick-PC:~$ sudo apt-get install cup
```



Para mas información sobre la creacion de archivos jflex e incorporar cup, asi mismo como ejecutar estos componentes para realizar el analisis de una entrada:

<http://e-navarro.blogspot.com/2015/09/mi-primer-proyecto-utilizando-jflex-y-cup.html>  
<https://openfecks.wordpress.com/jflex-y-cup/>

### **ANÁLISIS DE REQUERIMIENTOS:**

1. **Abrir un archivo:** Abre un archivo, a través de un buscador
2. **Guardar un archivo:** Toma el documento abierto y seleccionado, y guarda los datos.
3. **Guardar como un archivo:** Toma el documento abierto y seleccionado, luego solicita la ruta donde se guardara el archivo
4. **Nuevo archivo:** crea un archivo en blanco
5. **Salir:** Cierra la aplicación
6. **Compilar :** Toma la cadena de texto del documento abierto y selecciona en baje al lenguaje seleccionado lo compila para ese lenguaje.
7. **Generar código intermedio:** Toma el documento abierto y seleccionado, luego lo analiza y transforma a un lenguaje de programación nuevo.
8. **Optimizar código intermedio:** Elimina del repositorio y aplicación el lenguaje.
9. **Generar código assembler:** Muestra la tabla LALR de un lenguaje seleccionado.

## CASOS DE USO ALTO NIVEL:

<b>Numero</b>	CU01
<b>Caso de Uso:</b>	Abrir un archivo
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un menú “archivo” dentro un ítem abrir archivo, que despliega un buscador de archivos, luego de seleccionar el archivo se visualiza en el software.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU02
<b>Caso de Uso:</b>	Guardar un archivo
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un menú “archivo” dentro un ítem guardar archivo, que actualiza el archivo y lo guarda.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU03
<b>Caso de Uso:</b>	Guardar como un archivo
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un menú “archivo” dentro un ítem guardar archivo como, permite guardar archivo además con un nuevo nombre y nueva ruta.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU04
<b>Caso de Uso:</b>	Nuevo archivo
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un menú “archivo” dentro un ítem Nuevo archivo, abre un nuevo documento en blanco.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU05
<b>Caso de Uso:</b>	Salir
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Un usuario, estando en el aplicativo se dirige a la barra de menú superior, en el menú archivo existe un menú “archivo” dentro un ítem salir, cierra la aplicación.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU06
<b>Caso de Uso:</b>	Compilar
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Toma la pestaña abierta, y mediante el texto escrito genera un análisis léxico, sintáctico y semántico.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU07
<b>Caso de Uso:</b>	Generar código intermedio
<b>Actores:</b>	Usuario
<b>Descripción:</b>	El usuario, luego de que se halla compilado la cadena de entrada le da click en el boton se genera el código intermedio.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU08
<b>Caso de Uso:</b>	Optimizar código intermedio
<b>Actores:</b>	Usuario
<b>Descripción:</b>	Si el usuario desea un código más optimo, le da click en el botón y este optimiza el codigo y lo muestra.
<b>Tipo:</b>	Primario

<b>Numero</b>	CU09
<b>Caso de Uso:</b>	Generar código assembler
<b>Actores:</b>	Usuario
<b>Descripción:</b>	El usuario, luego de que se halla compilado la cadena de entrada le da click en el botón se genera el código assembler.
<b>Tipo:</b>	Primario

## ESTRUCTURA PARA CODIGO VISUAL BASIC

Método	Public Function ID (parámetros o nada) As tipo_variable ..... End Function
Procedimiento	Public Sub ID (parámetros o nada) ..... End Sub
Declaración de variable	Dim ID As tipo_variable o Dim ID=valor As tipo_variable
tipo_variable	Double,Integer y Char
parámetros	ByVal ID As tipo_variable
Condición if, if-else, if-elseif-else	If condición Then ..... Else If Then .... End If
Estructura while	While condición .... End While
Estructura do while	Do .... Loop While condición
Estructura switch	Select Case ID Case valor_constante .... Exit Select o sin el Exit Select End Select
Estructura for	For ID As tipo_variable = valor To valor (nada o Step valor) .... End For

Cabe mencionar que aunque el lenguaje no trabaje en base a que todo este bien tabulado, si se respeta los salto de linea de cada instrucción, ese decir por cada instrucción se debe de dar un salto de linea para poder agregar otro.

## ESTRUCTURA PARA CODIGO PYTHON

Funciones o procedimiento	def ID (argumentos o nada): .....
variables	id=valor
argumentos	id
Estructura while	while condición : ....
Estructura if-else	if condición : .... elif condición: .... else: ....
Estructura for	for id in range(valor inicial,valor final, valor de aumento) ....

Para esta lenguaje si se considera de que todo este tabulado por ejemplo:

```
def suma(x,y):  
    suma=0  
    if x>y:  
        suma=x*y  
    else:  
        suma=y+x  
    print("resultado",suma)
```