

*Universidad de San Carlos de Guatemala USAC.  
División de Ciencias de la Ingeniería.  
Centro Universitario de Occidente CUNOC.  
Organización de lenguajes y compiladores  
Ing. Moises*



***Estudiante***

*Elvis Lizandro Aguilar Tax*

***Carnet***

*201930304*

***Manual de Usuario***

## 1. Ejecución:

Debe tener descargada ambas aplicaciones, de extensión .jar, para poder descargarlas y ejecutarlas en la consola, claro esta que es una aplicación java por lo que necesita tener JVM para que funcione correctamente.

### Comando de ejecución:

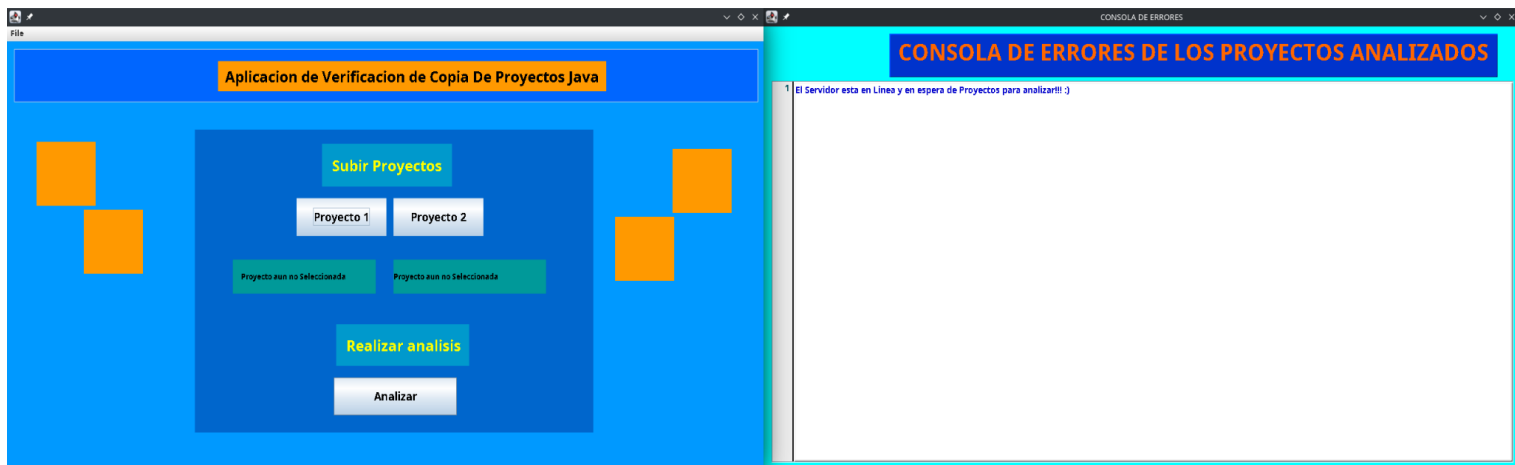
`java -jar Cliente-whit-dependences`

`java -jar servidor-whit-dependences`

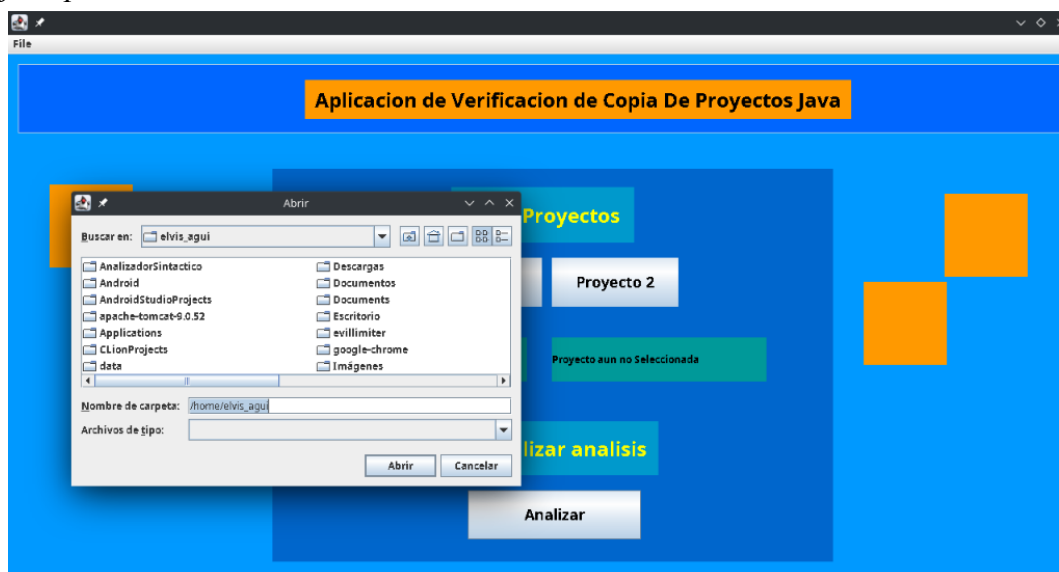
este comando ejecutarlos en la misma ruta donde se encuentran los programas ejecutables.

## 2. funcionamiento

Debes tener en cuenta, que las dos aplicaciones esten ejecutándose en el mismo momento



en la aplicación Cliente (el de lado izquierdo) podrás seleccionar los proyectos con sintaxis java para analizar.

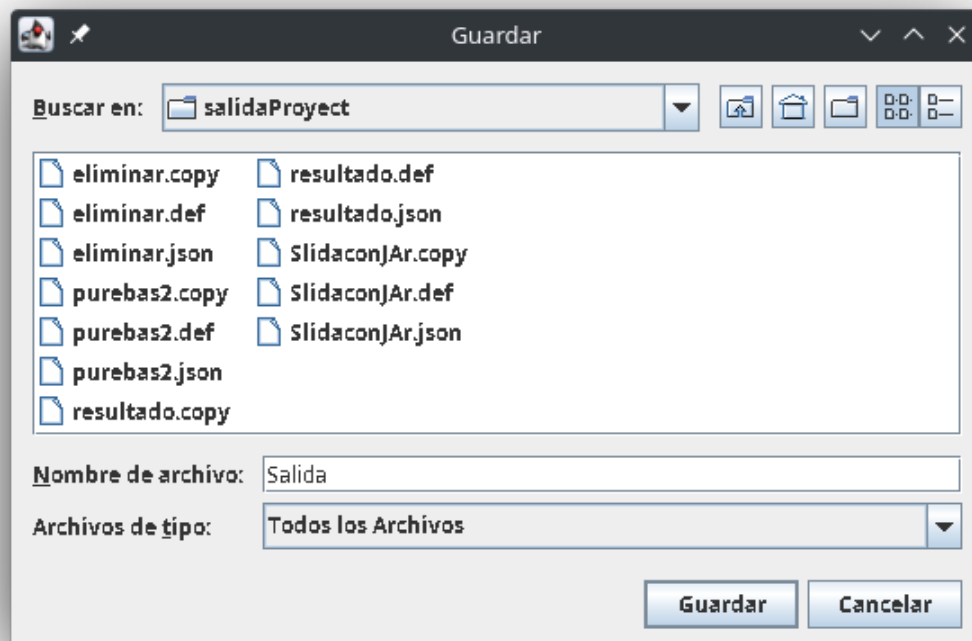


Verifica que ambos proyectos están en el apartado de proyecto seleccionado  
El siguiente paso es Analizar los proyectos

*Caso 1:*

***que tu proyecto no tenga ningún error de sintaxis***

*Se te pedirá que en una carpeta guardes los proyectos generados con este análisis como los archivos de json y dej y copy para poder cargarlos de nueva cuenta.*

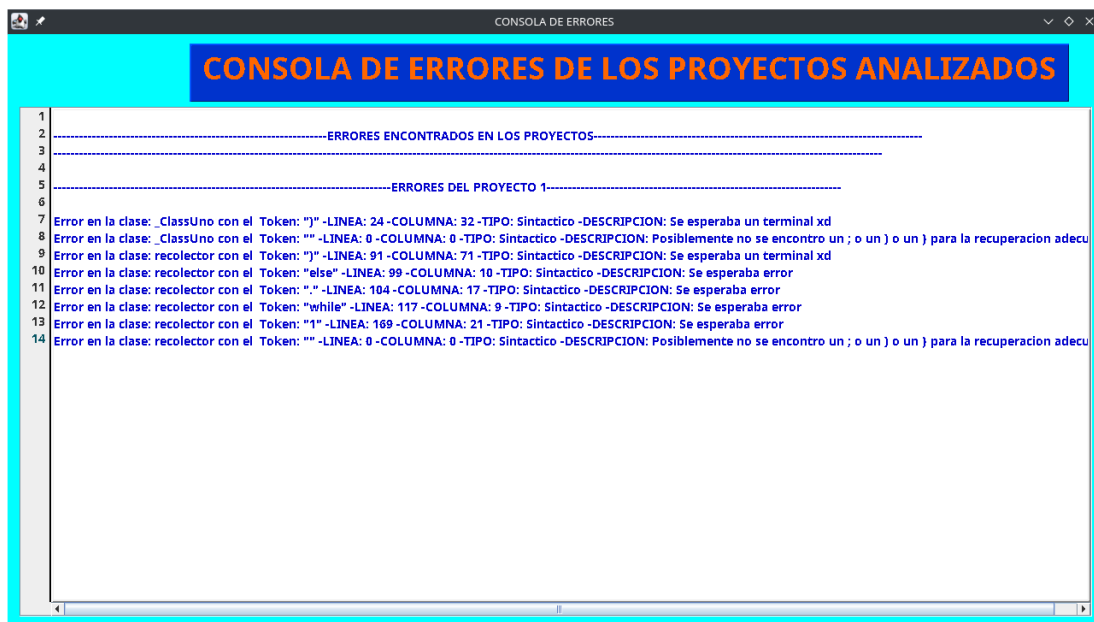


*con ponerle un nombre es más suficiente*

*Caso 2:*

***Tus proyectos tengan errores de sintaxis***

*En el apartado de la consola del servidor encontrarás los errores*



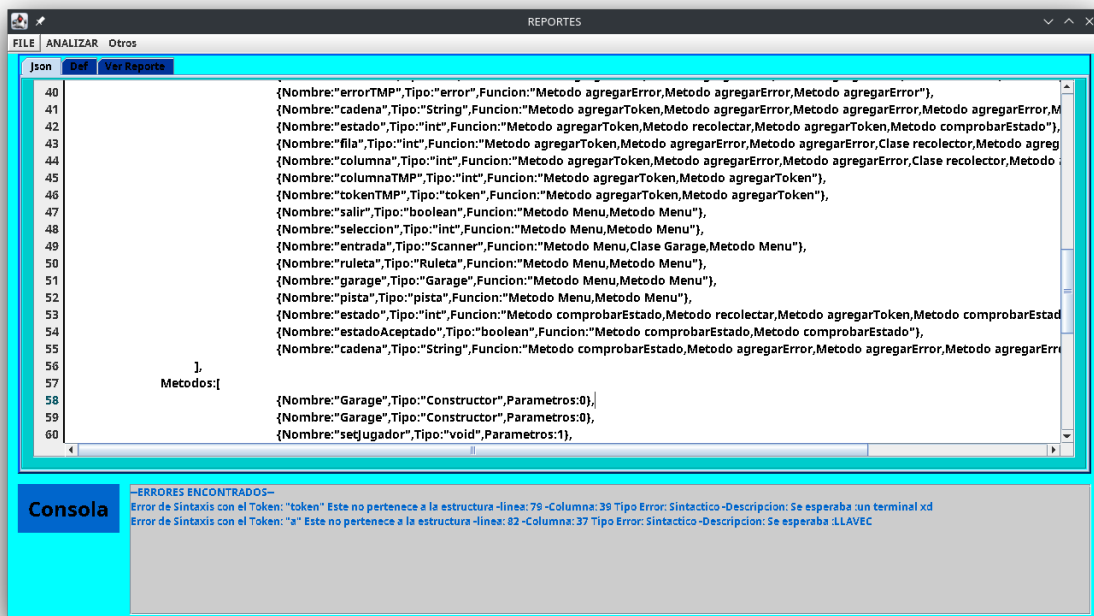
*deberás arreglar los errores para pasar a las siguiente pantalla.*

### **3. definicion del lenguaje DEF**

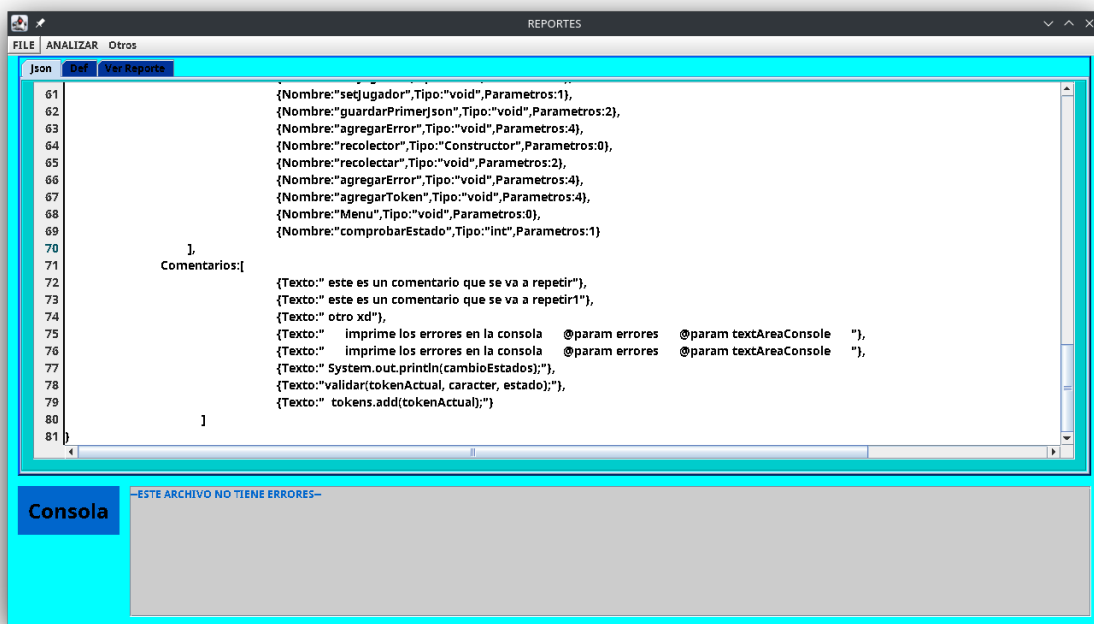
*donde podras generar reportes*

*deberás analizar el json para poder obtener*

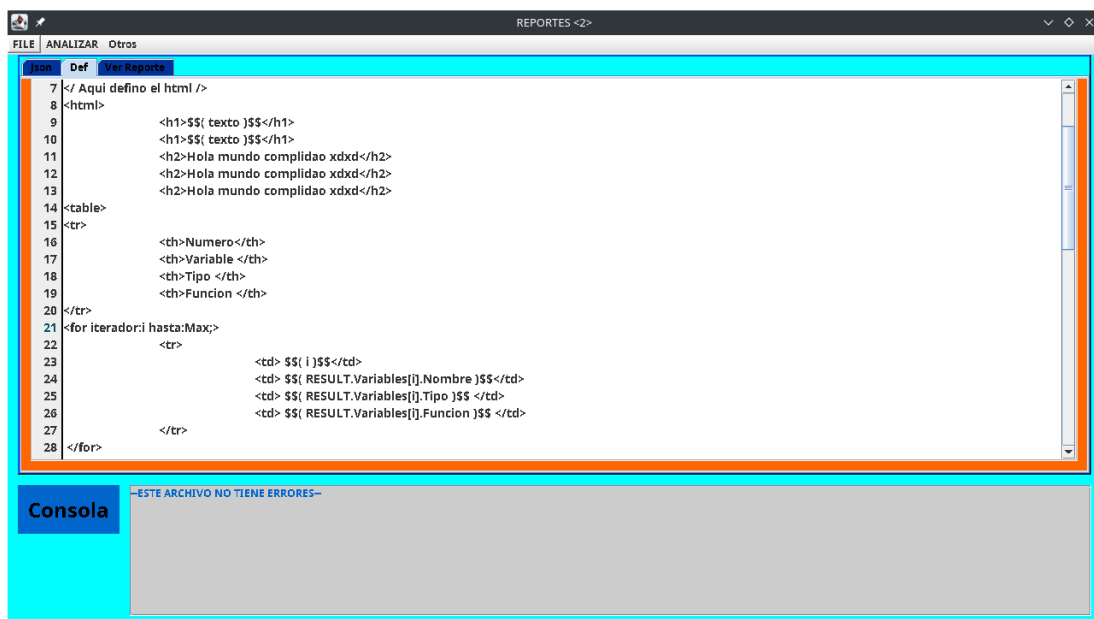
*Si el archivo tiene errores los encontrarás en el apartado de consola*



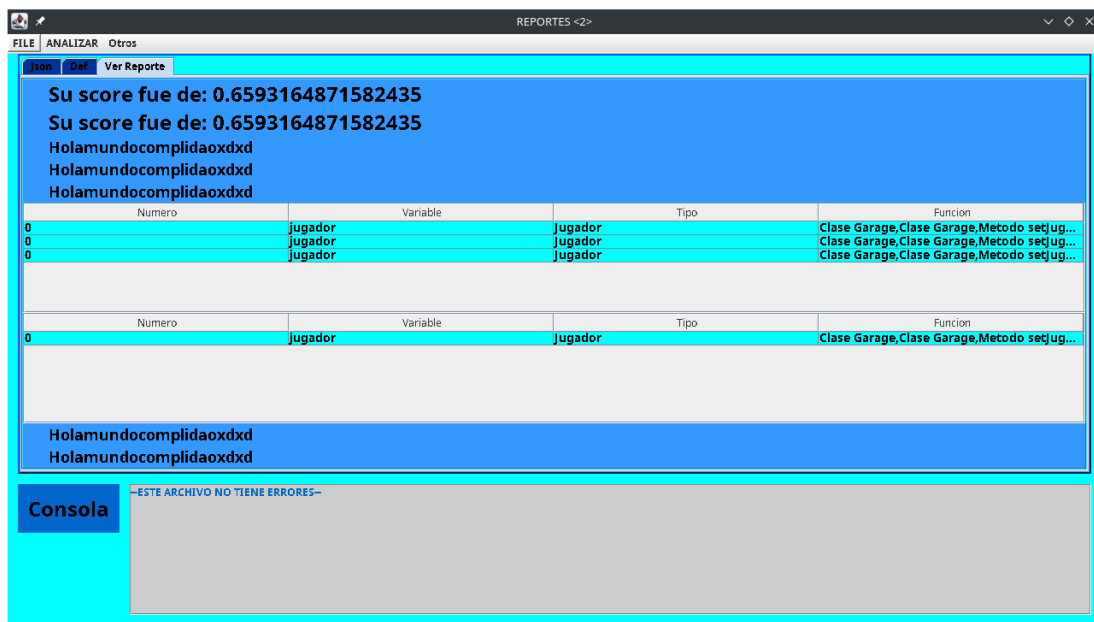
*una vez que arregles los errores podrás generar su sintaxis del proyecto*



*definicion DEF*



*de igual manera si tiene sintaxis invalida se mostrar en la consola, de lo contrario podra tener tu reporte gráficamente*



### Opciones Extras

en la barra de herramientas podrás encontrar las opciones de analizar, y también de abrir otro proyecto, así mismo también podrás guardar cambios, podrás regresar a la pantalla inicial.

Agregados:

Forma de definir cada sintaxis

Sentencias de java válidas

### 1. Imports

Descripción	Ejemplo
Importar Clases Java o librerías	<code>Import java.util.* ;</code> <code>Import analisis.AST ;</code>

### 2. Clases

Descripción	Ejemplo
Clases java simples.	<code>Visibilidad class clase_ejemplo{</code> <code>//Sentencias</code> <code>}</code>

Tabla No. 2

### 3. Métodos y Funciones

Descripción	Ejemplo
Declaración de métodos y funciones.	<b>Visibilidad tipo</b> método_funcion(con_o_sin_parametros){ //sentencias }

Tabla No. 3

#### 4. Declaración de variables

Descripción	Ejemplo
Declaración de variables.	<b>Visibilidad tipo</b> Lista_de_variables ;

#### 5. Asignaciones

Descripción	Ejemplo
La asignación puede ser al momento de declarar o después de declarar.	<b>Visibilidad tipo</b> Lista_de_variables = <b>expresión</b> ; O Lista_de_variables = <b>expresión</b> ;

Tabla No. 5

#### 6. Tipos de datos:

Descripción	Ejemplo
Los tipos de datos a utilizar son.	Enteros = <b>int</b> Booleanos = <b>boolean</b> Cadenas = <b>String</b> Carácter = <b>char</b> Decimales = <b>double</b> Objetos = <b>Object</b> Donde los objetos son instancias de otras clases.

Tabla No. 6

#### 7. Sentencias de control.

Descripción	Ejemplo
Las sentencias de control son son.	<b>If ,if else, for, while, do while, switch.</b>

Tabla No. 7

#### 8. Operadores aritméticos

Descripción	Ejemplo
Las operadores posibles son	<b>Relacionales, lógicos, incremento/decremento, agrupación.</b>

## 9. Comentarios

Descripción	Ejemplo
Los dos tipos de comentarios, simples y multilínea	//este es un comentario simple.  /*este Es multi línea */

Tabla No. 9

## 10. Constructor

Descripción	Ejemplo
La clase puede tener su constructor donde se inicializan las variables	Public Nombre_clase(Con_o_Sin_Parámetros){ //sentencias }

Tabla No. 10

## 11. Visibilidad de variables y clases:

Descripción	Ejemplo
La visibilidad de las clases y variables.	Private, public, protected, final

Tabla No. 11

## 12. Sentencias de salida.

Descripción	Ejemplo
Las dos sentencias de flujo que pueden venir dentro de un while, for, switch, y funciones.	Break , Return

### *sintaxis de json válidas*

```
{  
  Score: "0.75" ,  
  Clases:[  
    { Nombre: "clase1"}, {"Nombre":"clase2"}  
  ],  
  Variables:[  
    {Nombre: "var1", "Tipo":"int", "Funcion": "funcion1, funcion2"},  
    {Nombre: "var2", "Tipo":"int", "Funcion": "funcion2, Clase hola"}  
  ],  
  Metodos:[  
    {Nombre: "metodo1","Tipo":"void" , "Paremetros": 2},  
    {Nombre: "metodo2","Tipo":"String", "Parametros":0}  
  ],  
}
```



```

    Comentarios:[
      { "Texto": "hola es un comentario"},
      { "Texto": "otro coment"}
    ]
  }

```

Para que sea válida, los Keys no tiene que tener comillas solo el contenido de las mismas.

### Sintaxis valida del DEF

Acceder a variables del json

Acceso	Tipo
RESULT.Score	String
RESULT.Clases	Lista de objetos Clase
RESULT.Clases[0]	Objeto Clase en la posición 0 del arreglo
RESULT.Clases[0].Nombre	String
RESULT.Variables	Lista de objetos Variable
RESULT.Variables[0]	Objeto Variable en la posición 0 del arreglo
RESULT.Variables[0].Nombre	String
RESULT.Variables[0].Tipo	String
RESULT.Variables[0].Funcion	String
RESULT.Metodos	Lista de objetos Método
RESULT.Metodos[0]	Objeto Método en la posición 0 del arreglo
RESULT.Metodos[0].Nombre	String
RESULT.Metodos[0].Tipo	String
RESULT.Metodos[0].Parametros	Integer
RESULT.Comentarios	Lista de objetos Comentario
RESULT.Comentarios[0]	Objeto Comentario en la posición 0 del arreglo
RESULT.Metodos[0].Texto	String

Las variables serán declarables de la siguiente forma, las opciones dentro de corchetes son opcionales.

<nombre\_del\_tipo\_de\_dato> <nombre\_de\_la\_variable> [ = <expresión> ];

<nombre\_del\_tipo\_de\_dato> <nombre\_de\_la\_variable\_0>,  
<nombre\_de\_la\_variable\_1>,<nombre\_de\_la\_variable\_n>;

Etiquetas validas

Etiqueta	Tipo, Función
<html></html>	Inicio y finalización del html
<h1></h1>	String, Tamaño de letra
<h2></h2>	String, Tamaño de letra 2
<table></table>	Tabla, Inicio de una taba
<for iterador:i hasta:Entero1;> </for>	Iterador, repetir una secuencia desde un punto hasta otro
<tr> </tr>	Fila Columna, Indica desde donde a donde abarca una fila de la columna
<th></th>	Columna título, crea una columna del tipo título para una tabla
<td> </td>	Columna datos, crea una columna de tipo datos para una tabla.
 	Espacio, deja un enter de espacio entre los componentes.