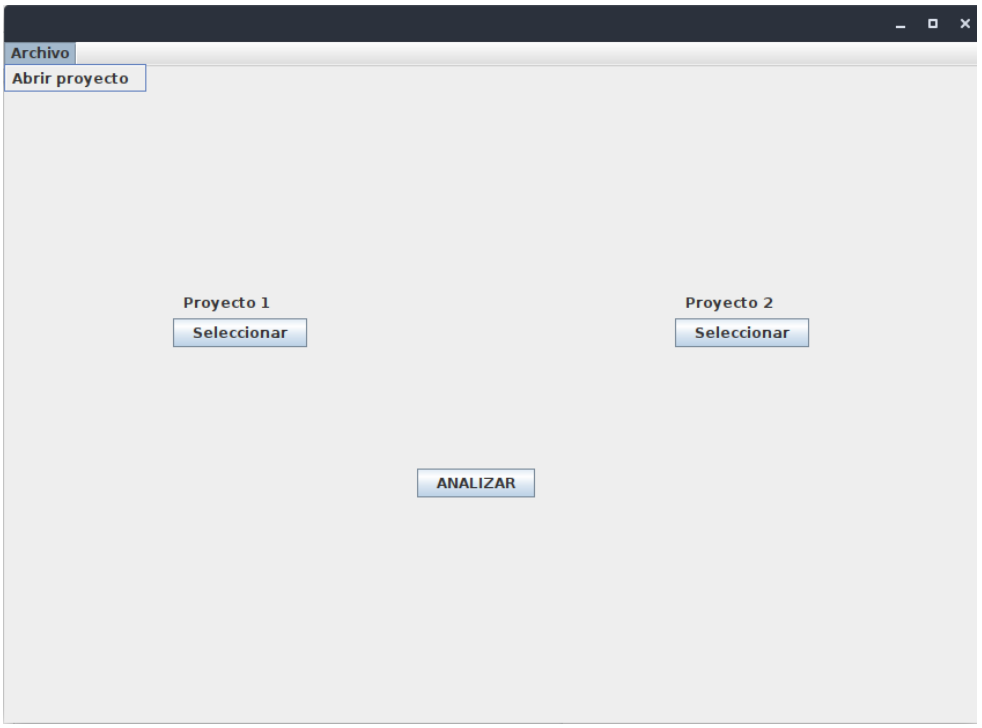
MANUAL DE USUARIO

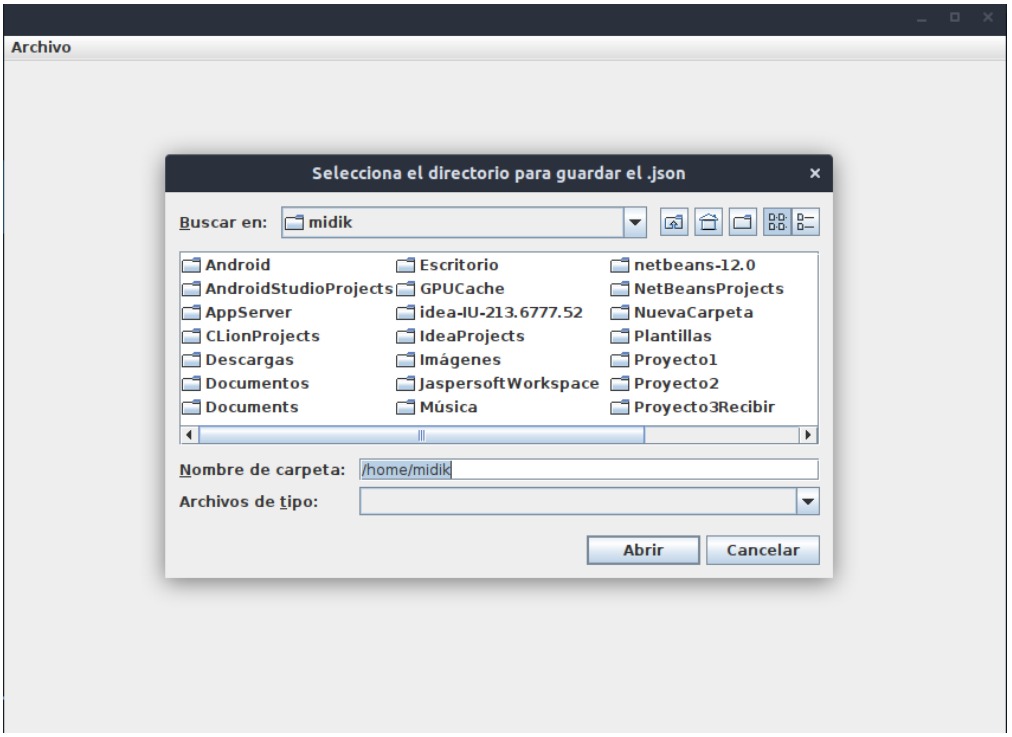
APLICACIÓN - SERVIDOR

ÁREA DE ERRORES: En este área aparecen los errores léxicos y sintácticos que existen en los proyectos. También aparece el nombre de los archivos que fueron analizados correctamente.
BOTÓN "LIMPIAR": Limpia en área de errores.

APLICACIÓN - CLIENTE



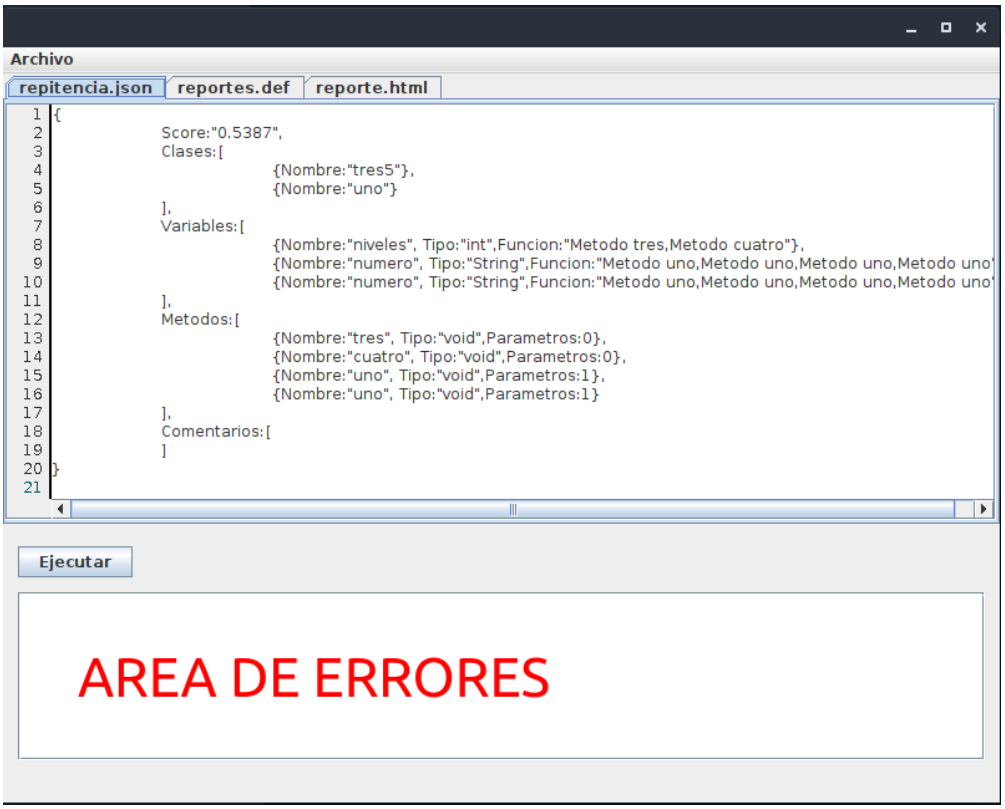
ABRIR PROYECTO: Permite elegir un archivo .copy, estos archivos son generados después de un análisis.
BOTÓN "Seleccionar": Sirve para elegir uno de los proyectos*.
BOTÓN "ANALIZAR": Envía los proyectos al servidor para su análisis.



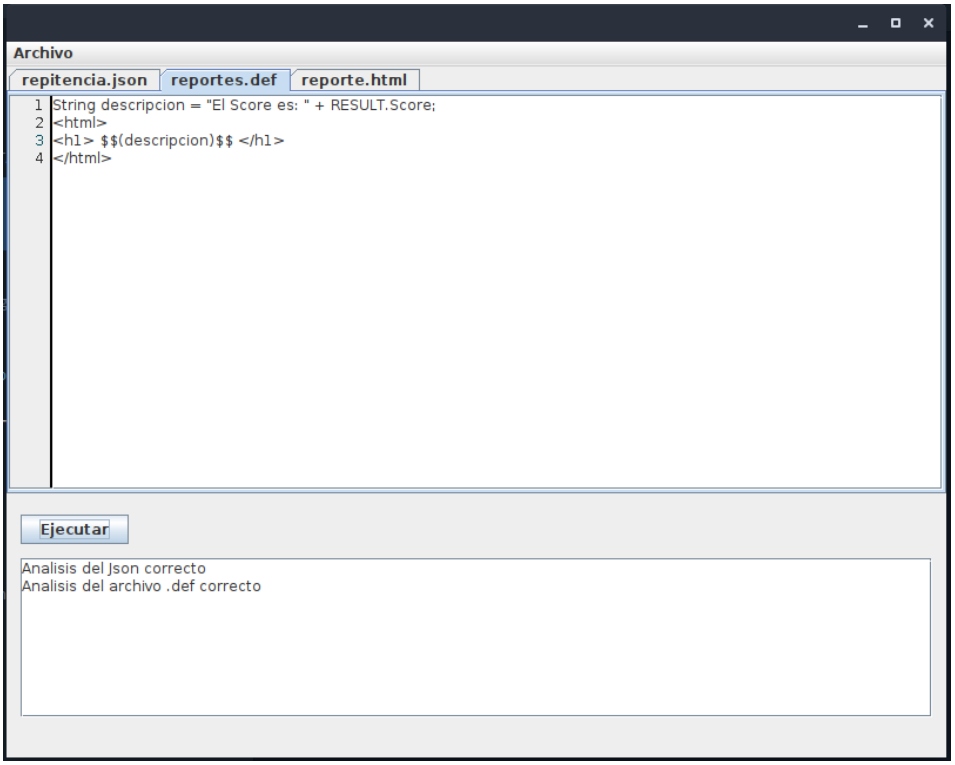
*Proyecto: directorio con cero o muchos archivos.java

Al terminar el análisis se mostrará una ventana que permite elegir en donde se guardarán los resultados del análisis.

Después de elegir el directorio se mostrará el editor y generador de reportes.



ARCHIVO->GUARDAR: Guarda los cambios realizados en el proyecto.
PESTAÑA "repitencia.json": Muestra los resultados del análisis, estos resultados pueden ser modificados siguiendo una determinada sintaxis (Sintaxis repitencia.json).
BOTÓN "Ejecutar": Analiza la sintaxis del código de la pestaña repitencia.json, si esta es correcta procede a analizar el código de la pestaña reportes.def y si esta es correcta genera una vista del reporte en la pestaña reporte.html.
ÁREA DE ERRORES: Muestra los errores léxicos y sintácticos del código de la pestaña repitencia.json, también muestra los errores léxicos, sintácticos y semánticos del código de la pestaña reportes.def. También indica si el análisis de su sintaxis es correcta.



PESTAÑA "reportes.def": En este apartado se puede escribir código para generar reportes ([Generación de reportes](#)).

PESTAÑA "reporte.html": En este apartado se puede visualizar el reporte escrito en reportes.def

Sintaxis repitencia.json

```
{
        “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"}
            ]
}
```

Lo que está en **rojo** es obligatorio.

Agregar clases: Se agrega entre llaves, siguiendo este orden: “Nombre”: “nombre”
Ejemplo:

{ “Nombre”: "clase1"}

Si se define mas de una clase se separa con coma:

{ “Nombre”: "clase1"}, {“Nombre”:"clase2"}

Agregar variables: Se agrega entre llaves, siguiendo este orden: “Nombre”: "var1", “Tipo”:"int", “Funcion”: "funcion1, funcion2"
Ejemplo:

{“Nombre”: "var1", “Tipo”:"int", “Funcion”: "funcion1, funcion2"}

Si se define más de una variable se separa con coma:

{“Nombre”: "var1", “Tipo”:"int", “Funcion”: "funcion1, funcion2"},
{“Nombre”: "var2", “Tipo”:"int", “Funcion”: "funcion2, Clase hola"}

Agregar métodos: Se agrega entre llaves, siguiendo este orden: “Nombre”: "metodo1",”Tipo”:"void" , “Paremetros”: 2
Ejemplo:

{“Nombre”: "metodo1",”Tipo”:"void" , “Parametros”: 2}

Si se define mas de un metodo se separa con coma:

{“Nombre”: "metodo1",”Tipo”:"void" , “Paremetros”: 2},
{“Nombre”: "metodo2",”Tipo”:"String", “Parametros”:0}

Agregar comentarios: Se agregan entre llaves, siguiendo este orden: “Texto”: "hola es un comentario"
Ejemplo:

{ “Texto”: "hola es un comentario"}

Si se definen más de un comentario se separa con comas:

{ “Texto”: "hola es un comentario"},
{ “Texto”: "otro coment"}

Generación de reportes

Está dividida en dos secciones

1. Definición de variables
2. Definición de html

1. Definición de variables

Variables definidas para el lenguaje

Los tipos de variables que podran venir definidas en nuestra sección de definición de variables son:

Tipo	Entrada
Integer	Número entero: 5,45,948, etc.
String	Cadena: “hola es una cadena”

Tabla No. 15

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>;

Ejemplos:

Integer var1;
Integer var1 = 5+6*5;
Integer var1, var2, var3;
String hola=”Hola ”+ “Mundo”

Asignación de valores:

A las variables definidas se les podrá asignar o cambiar su valor mediante la sintaxis:

<nombre_de_la_variable> = <expresión> ;

Ejemplo

entero1 = 1;
cadena1=RESULT.Metodos[entero1].Nombre;
cadena2=“hola ”+”Resultado”
cadena3=RESULT.Score;

Variable Global Json

Podemos acceder a las variables de repitencia.json con esta sintaxis:

Accesso	**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.Comentarios[0].Texto	String

Operadores Aritméticos reconocidos:

Operador	**Operación**	**Aplicación**
+	Suma	5+var2
-	Resta	Var3-var2
/	división	Var3/var2
*	multiplicación	Var3*var6*5.5
()	Agrupación	(5+5)*2/(8-7)

IMPORTANTE: LA SECCIÓN DE DEFINICIÓN DE VARIABLES INICIA EN EL PRINCIPIO Y TERMINA AL ENCONTRAR LA ETIQUETA

<html>

Comentarios:

Los comentarios podrán ser definidos tanto en en la sección de definición de variables como en la sección del html. y tendrán la siguiente estructura:

(</) comentario (/>)

Ejemplo:

</ Este es un comentario
multilínea />

2. Sección de definición de html

Definición de etiquetas permitidas:

Se utilizará una simplificación de html y las etiquetas que se pueden definir son:

Etiqueta	Tipo, Función
<html></html>	Inicio y finalización del html
	String, Tamaño de letra
	String, Tamaño de letra 2
<table></table>	Tabla, Inicio de una taba
<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.

Acceso a variables en la sintaxis html:

Dentro de cualquier etiqueta podremos escribir cualquier texto pero también podremos acceder a los valores dentro de cualquier variable para poder visualizarla de la siguiente forma:

$$ (<Nombre_Variable>) $$

Ejemplo

<h1>El score es: $$(RESULT.Score)$$ </h1>

Ejemplo completo

```
</ Sección de definición de variables />
Integer i = 0;
String titulo1 = “Numero”;
String titulo2 = “Variable”;
String titulo3 = “Tipo”;
String titulo4 = “Funcion”;
String texto=”Su score fue de: ”+RESULT.Score

</ Sección de definición de html />
<html>
        <h1>$$( texto )$$<h1>
        <table>
                <tr>
                        <th>$$( titulo1 )$$<th>
                        <th>$$( titulo2 )$$ <th>
                        <th>$$( titulo3 )$$ <th>
                        <th>$$( titulo4 )$$ <th>
                 </tr>
                <tr>
                        <td> $$( i )$$<td>
                        <td> $$( RESULT.Variables[i].Nombre )$$<td>
                        <td> $$( RESULT.Variables[i].Tipo )$$ <td>
                        <td> $$( RESULT.Variables[i].Funcion )$$ <td>
                </tr>
        </table>
</html>
```

Nota: Los lenguajes de la aplicación de usuario NO son Case sensitive, es decir, html y HtmL significan lo mismo.

Más información sobre las etiquetas:

1. Las etiquetas <tr> </tr> solo pueden ser definidas dentro de las etiquetas <table> </table>
2. Las etiquetas <th></th> solo pueden estar definidas dentro de las etiquetas <tr> </tr>
3. Las etiquetas <td> </td> solo pueden estar definidas dentro de las etiquetas <tr> </tr>
4. Las etiquetas <html> </html> solo pueden estar definidas una vez.
5. Las etiquetas
, <h1></h1>, <h2> </h2> pueden venir en cualquier lugar o dentro de cualquier otra etiqueta.