

MANUAL DE USUARIO

06.04.2021

—

Carlos Benjamín Pac Flores

Centro Universitario de Occidente (CUNOC)

Ingeniería en Ciencias y Sistemas

Abril 2022

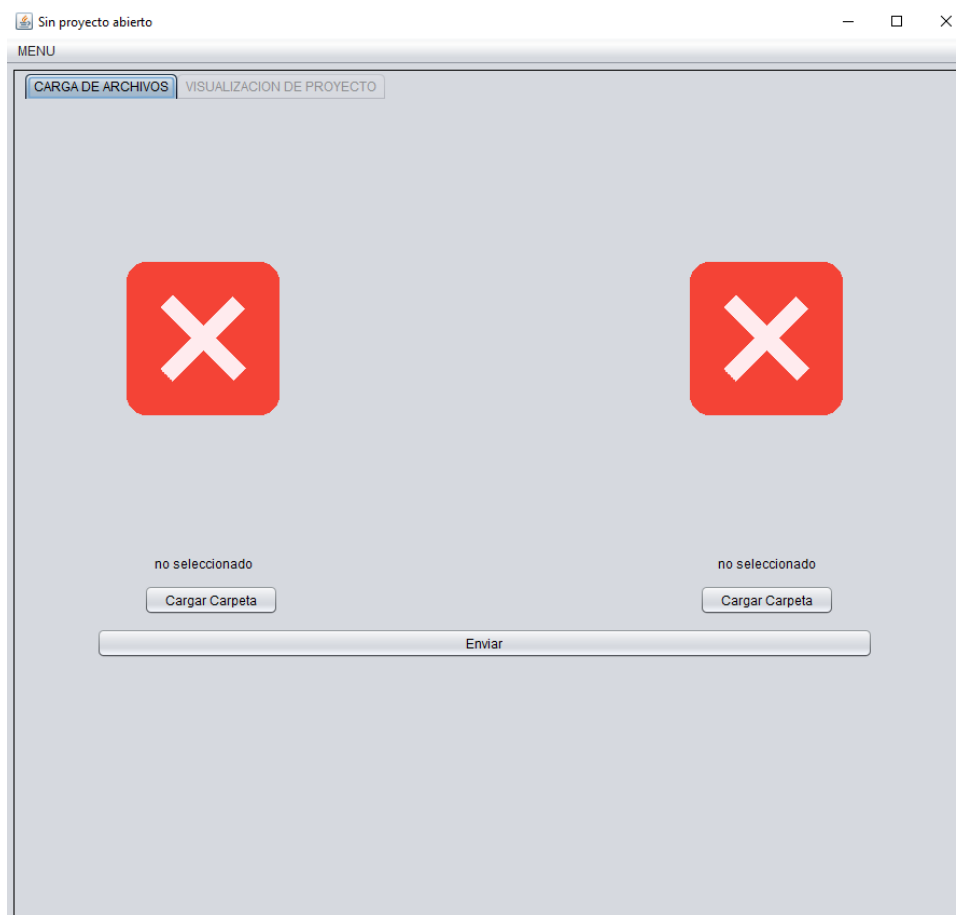
Requisitos del Programa	2
Funcionamiento de la aplicación	2
reportes.def	4
resultado.json	5
Reportes	6
Aspectos Importantes	7
Codigo COPY	7
Archivo Json	9

Requisitos del Programa

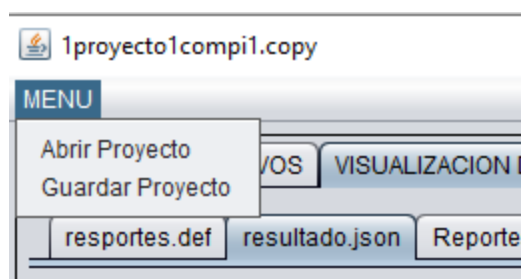
- Java 11
- RAM libre recomendada para uso 2GB
- Almacenamiento libre 100MB

Funcionamiento de la aplicación

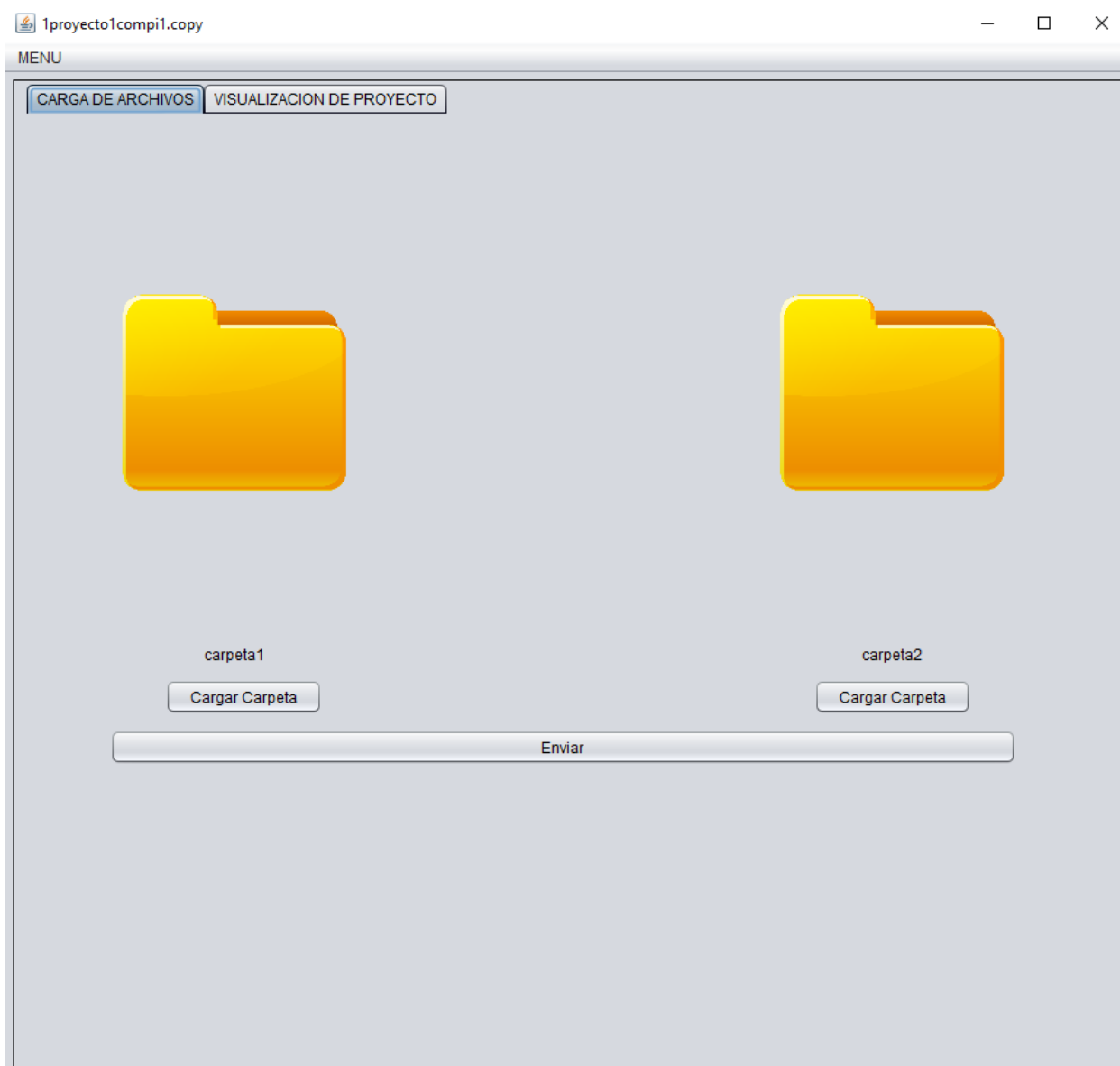
Pantalla principal de la aplicación en donde encontramos un barra principal del menú una ventana en la cual podemos seleccionar las carpetas con los archivos que deseamos enviar y un botón principal en donde enviaremos la información al servidor.



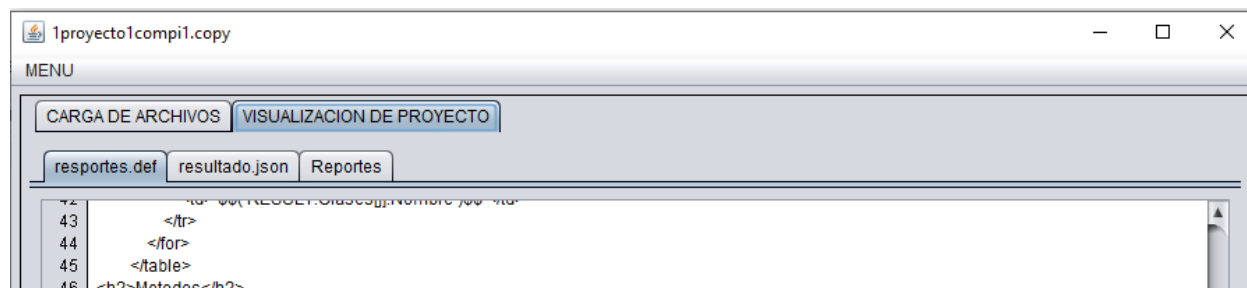
En la barra integrada en la aplicación tenemos la posibilidad de guardar un proyecto por completo o de abrir otro proyecto que hayamos generado con anterioridad.



Ya creado o aperturado un proyecto en la aplicación la barra de título mostrará el nombre del proyecto, y se nos aperturará una nueva pestaña en donde podremos acceder a las demás funciones de la aplicación.

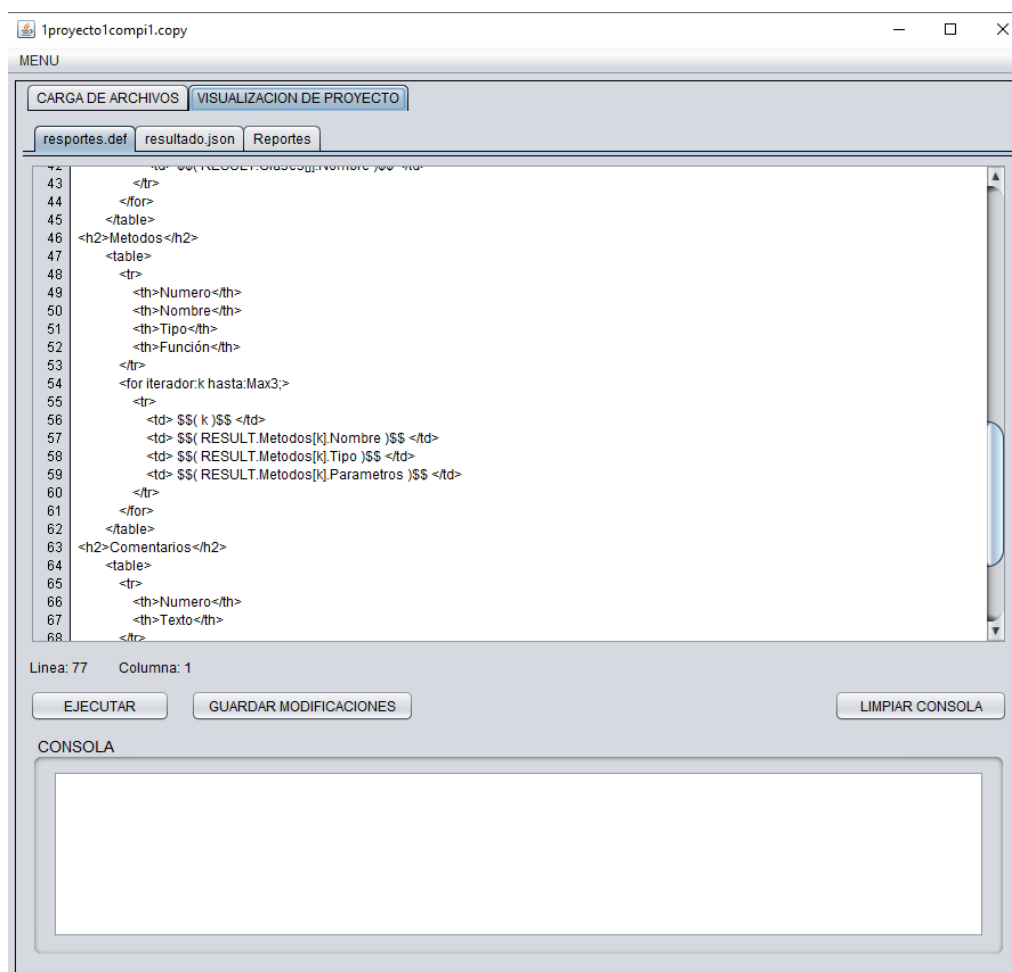


En esta segunda pestaña denominada “VISUALIZACIÓN DE PROYECTO” tendremos una vista que contiene 3 pestañas nuevas, “reportes.def”, “resultado.json” y “Reportes” donde la última de estas es un visualizador de html integrado.



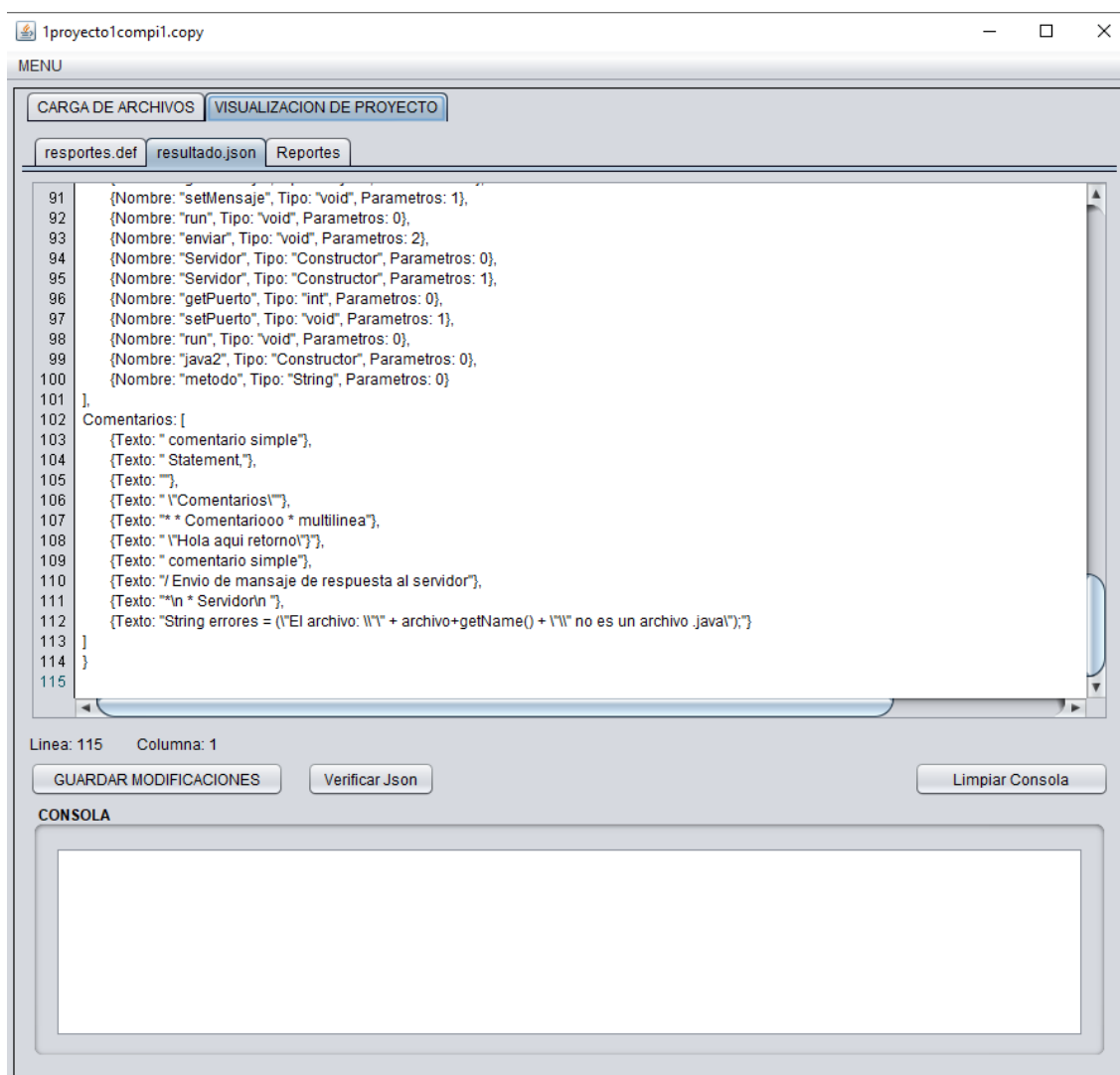
1. reportes.def

En esta pestaña del programa encontraremos un espacio dedicado para la edición de código COPY donde en todo momento se nos dirá nuestra ubicación en el cuadro de edición de texto, luego más abajo hay ubicada una caja de texto dedicada a la consola del lenguaje y en caso de existir un error en el lenguaje nos mostrará la línea y columna donde ocurrió dicho error.



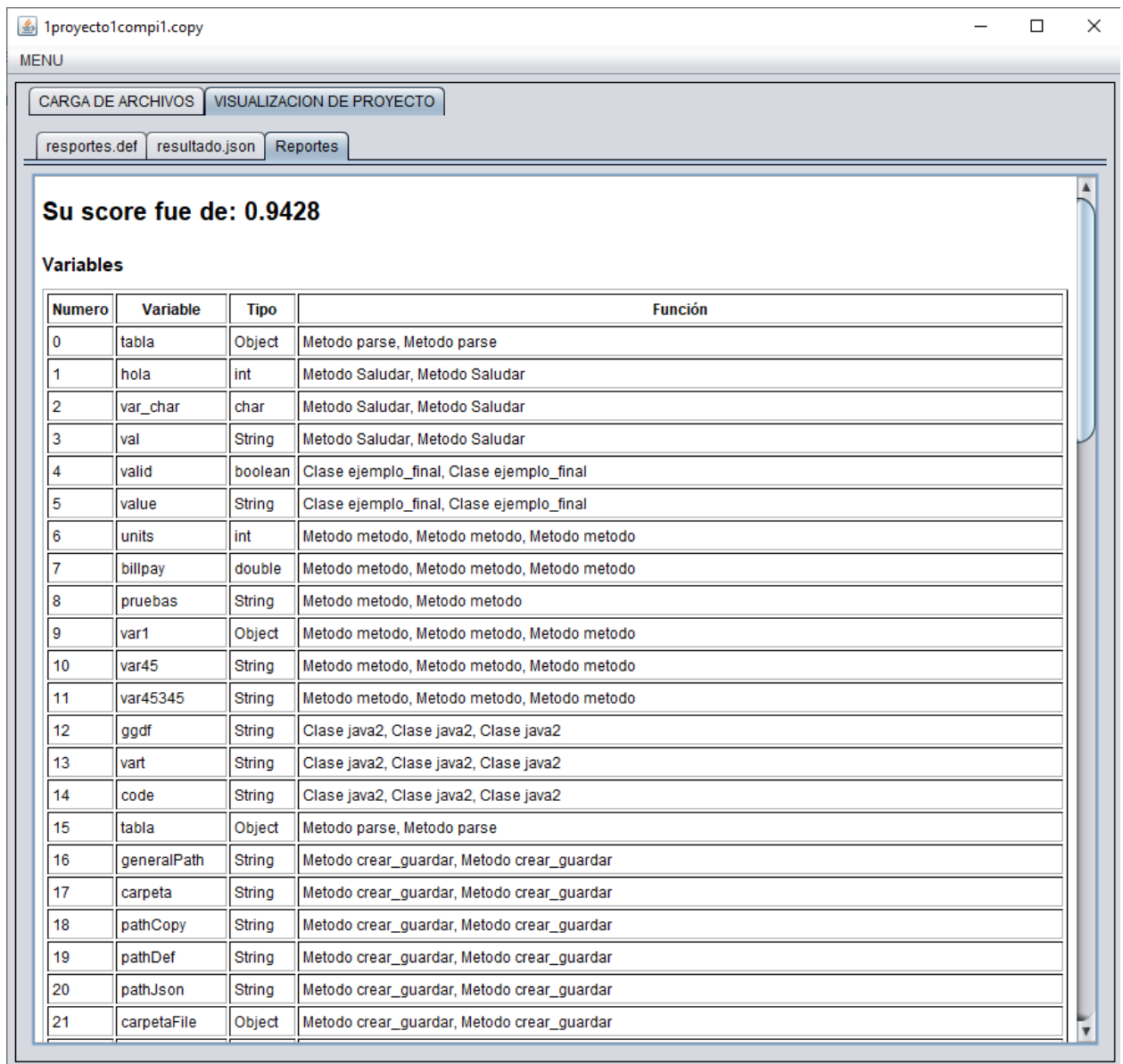
2. resultado.json

En esta pestaña el usuario puede visualizar el json recibido por el servidor además ofrece la edición al usuario en tal caso que sea necesario, al igual que la pantalla anterior el editor de texto nos ofrece la función de indicarnos dónde nos ubicamos dentro del mismo, y más abajo un consola y una serie de botones donde el más importante es el de "verificar Json" dicho botón realiza un pre-análisis del archivo json para saber si existe algún error del mismo y si lo llegara a existir en la consola se mostrará la ubicación del mismo en el contenedor de texto.



3. Reportes

El funcionamiento de esta pestaña está simplemente diseñado para la visualización de código en formato html ya renderizado, esta pestaña entra en funcionamiento cuando se ejecuta el código COPY en la pestaña de reportes.def.



Su score fue de: 0.9428

Variables

Numero	Variable	Tipo	Función
0	tabla	Object	Metodo parse, Metodo parse
1	hola	int	Metodo Saludar, Metodo Saludar
2	var_char	char	Metodo Saludar, Metodo Saludar
3	val	String	Metodo Saludar, Metodo Saludar
4	valid	boolean	Clase ejemplo_final, Clase ejemplo_final
5	value	String	Clase ejemplo_final, Clase ejemplo_final
6	units	int	Metodo metodo, Metodo metodo, Metodo metodo
7	billpay	double	Metodo metodo, Metodo metodo, Metodo metodo
8	pruebas	String	Metodo metodo, Metodo metodo
9	var1	Object	Metodo metodo, Metodo metodo, Metodo metodo
10	var45	String	Metodo metodo, Metodo metodo, Metodo metodo
11	var45345	String	Metodo metodo, Metodo metodo, Metodo metodo
12	ggdf	String	Clase java2, Clase java2, Clase java2
13	vart	String	Clase java2, Clase java2, Clase java2
14	code	String	Clase java2, Clase java2, Clase java2
15	tabla	Object	Metodo parse, Metodo parse
16	generalPath	String	Metodo crear_guardar, Metodo crear_guardar
17	carpeta	String	Metodo crear_guardar, Metodo crear_guardar
18	pathCopy	String	Metodo crear_guardar, Metodo crear_guardar
19	pathDef	String	Metodo crear_guardar, Metodo crear_guardar
20	pathJson	String	Metodo crear_guardar, Metodo crear_guardar
21	carpetaFile	Object	Metodo crear_guardar, Metodo crear_guardar

Aspectos Importantes

Codigo COPY

Este lenguaje es una extracción pequeña de HTML normal dotándolo de uso de variables y de ciclos y acceso al resultado JSON que recibimos del servidor, y que es guardado en un archivo con extensión .def, el lenguaje no es sensible a las mayúsculas y minúsculas en las instrucciones del mismo.

Ya descrito lo anterior se brinda un ejemplo del código de COPY:


```

</ iniciare a definir de alguna manera />
Integer Max1,Max2,Max3,Max4, i,j,k,l;
Max1=40;
Max2=5;
Max3=20;
Max4=6;
i=0;
j=0;
k=0;
l=0;
String texto="Su score fue de: "+RESULT.Score;
</ Aqui defino el html />
<html>
<H1>$$( texto )$$</h1>
<h2>Variables</h2>
    <table>
        <tr>
            <th>Numero</th>
            <th>Variable</th>
            <th>Tipo</th>
            <th>Función</th>
        </tr>
        <for iterador:i hasta:Max1;>
            <tr>
                <td> $$ ( i ) $$ </td>
                <td> $$ ( RESULT.Variables[i].Nombre ) $$ </td>
                <td> $$ ( RESULT.Variables[i].Tipo ) $$ </td>
                <td> $$ ( RESULT.Variables[i].Funcion ) $$ </td>
            </tr>
        </for>
    </table>
<h2>Clases</h2>
    <table>
        <tr>
            <th>Numero</th>
            <th>Nombre</th>
        </tr>
        <for iterador:j hasta:Max2;>
            <tr>
                <td> $$ ( j ) $$ </td>
                <td> $$ ( RESULT.Clases[j].Nombre ) $$ </td>
            </tr>
        </for>
    </table>

```

```

        </for>
    </table>
<h2>Metodos</h2>
    <table>
        <tr>
            <th>Numero</th>
            <th>Nombre</th>
            <th>Tipo</th>
            <th>Función</th>
        </tr>
        <for iterador:k hasta:Max3;>
            <tr>
                <td> $$ ( k ) $$ </td>
                <td> $$ ( RESULT.Metodos[k].Nombre ) $$ </td>
                <td> $$ ( RESULT.Metodos[k].Tipo ) $$ </td>
                <td> $$ ( RESULT.Metodos[k].Parametros ) $$ </td>
            </tr>
        </for>
    </table>
<h2>Comentarios</h2>
    <table>
        <tr>
            <th>Numero</th>
            <th>Texto</th>
        </tr>
        <for iterador:l hasta:Max4;>
            <tr>
                <td> $$ ( l ) $$ </td>
                <td> $$ ( RESULT.Comentarios[l].Texto ) $$ </td>
            </tr>
        </for>
    </table>
</html>

```

Archivo Json

Al igual que el lenguaje COPY el json utilizado en esta aplicación no es igual que un json normal, lo podríamos considerar un pseudo-json donde se comparten algunas características del lenguaje y unas implementaciones al momento de reconocer propiedades del objeto ya que los parámetros no deben ser estrictamente escritos como son dados si no que se pueden mezclar las mayúsculas y minúsculas como **Score** y **score** y **ScOrE** son la misma propiedad; pero que su encapsulamiento es diferente, un ejemplo a continuación:

```

{
Score: "0.9428",
Clases: [
  {Nombre: "ejemplo_final"},
  {Nombre: "java2"}
],
Variables: [
  {Nombre: "tabla", Tipo: "Object", Funcion: "Metodo parse, Metodo
parse"},
  {Nombre: "hola", Tipo: "int", Funcion: "Metodo Saludar, Metodo
Saludar"},
  {Nombre: "var_char", Tipo: "char", Funcion: "Metodo Saludar, Metodo
Saludar"},
  {Nombre: "val", Tipo: "String", Funcion: "Metodo Saludar, Metodo
Saludar"}, {Nombre: "code", Tipo: "String", Funcion: "Clase java2, Clase
java2, Clase java2"}
],
Metodos: [
  {Nombre: "ejemplo_final", Tipo: "Constructor", Parametros: 0},
  {Nombre: "parse", Tipo: "boolean", Parametros: 0},
  {Nombre: "Saludar", Tipo: "String", Parametros: 1},
  {Nombre: "java2", Tipo: "Constructor", Parametros: 0},
  {Nombre: "metodo", Tipo: "String", Parametros: 0},
  {Nombre: "CrearArchivos", Tipo: "Constructor", Parametros: 0},
  {Nombre: "parse", Tipo: "boolean", Parametros: 0},
  {Nombre: "metodo", Tipo: "String", Parametros: 0}
],
Comentarios: [
  {Texto: " comentario simple"},
  {Texto: " Statement,"},
  {Texto: ""},
  {Texto: " \"Comentarios\""},
  {Texto: "* * Comentarioooo * multilinea"},
  {Texto: " \"Hola aqui retorno\""},
  {Texto: " comentario simple"},
  {Texto: "/ Envio de mansaje de respuesta al servidor"},
  {Texto: "*\n * Servidor\n "},
  {Texto: "String errores = (\"El archivo: \"" + archivo+getName() +
\"\" no es un archivo .java\");"}
]
}

```