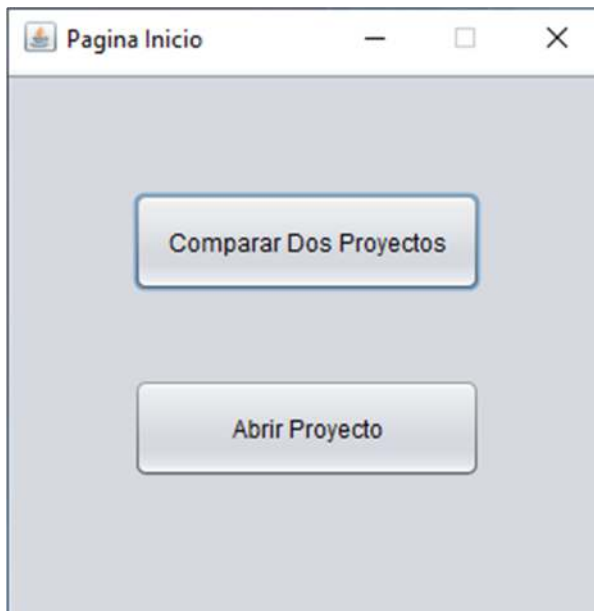


Manual de Usuario Proyecto 1.

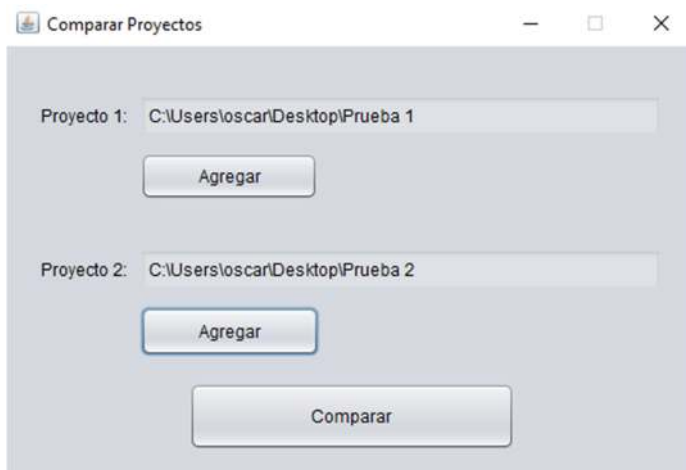
Ejecutar la aplicación del servidor: Servidor-Proyecto1-Compiladores1-jar-with-dependencies y mantenerlo así para poder comparar proyecto.

Ejecutar la aplicación de usuario: Aplicacion-Proyecto1-Compiladores1-jar-with-dependencies.

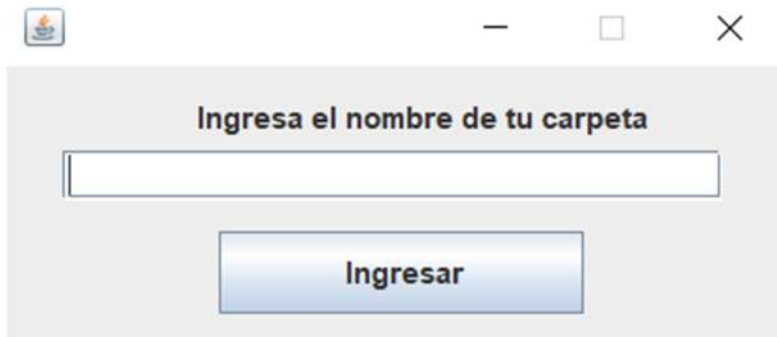
Se abrirá la siguiente ventana.



Al presionar el botón de comparar nos saldrá una ventana donde podremos agregar dos directorios para realizar una comparación.



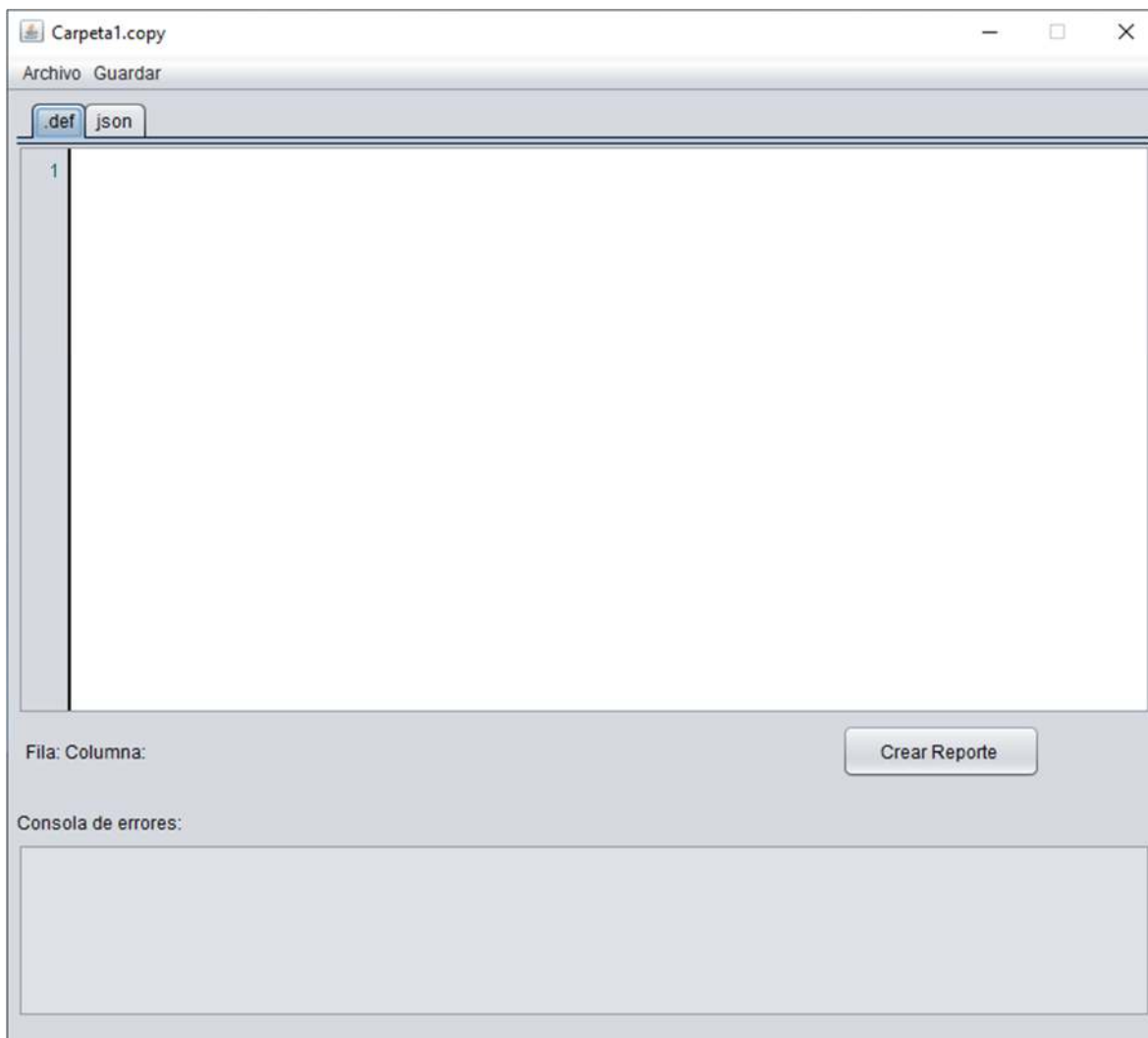
Al comparar. Si hay errores en alguna de las clases java. Saldrá una ventana con los errores enumerados. Sino se abrirán una ventana donde podremos ingresar el nombre de nuestro proyecto.

A screenshot of a small graphical user interface window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area has a light gray background. At the top, the text "Ingresa el nombre de tu carpeta" is displayed in a bold, black font. Below this text is a white rectangular text input field with a thin blue border. At the bottom of the window is a blue rectangular button with the word "Ingresar" written in white text.

Podremos guardar el proyecto en cualquier parte para posteriormente abrirlo desde otra ventana.

Se guardará una carpeta con un archivo. copy, .def. .json. El archivo. copy contiene los directorios de los demás archivos. El archivo .def contiene la estructura del lenguaje copy. Y el archivo. json contiene todas las clases, variables, métodos, comentarios y el score. El cual es el porcentaje de copia entre proyectos.

Regresamos a la ventana inicial. Al presionar en abrir proyecto aparecerá un buscador donde podremos elegir un archivo. copy. Al hacerlo se abrirá la siguiente ventana:



En el menú archivo podremos abrir un nuevo proyecto o comparar dos proyectos java. El menú de Guardar guarda los cambios realizados en el área de texto de la pestaña.def o la pestaña.json.

En la pestaña .def podremos agregar lenguaje copy para generar reportes html de nuestro archivo json.

La estructura es la siguiente:

Para los reportes se tendrán dos secciones una para la declaración de variable que serán usadas dentro del reporte; y una más para la sintaxis html para la definición de nuestros reportes.

Sección de definición de variables

Variable Global Json

El reporte JSON generado en la comparación de los proyectos es una variable global que podrá que al igual que las variables que se declaran podrá ser alcanzable tanto desde la sección de de declaración de variables como la sección del html. Este acceso a la información generada por el archivo JSON podrá ser accedida de la siguiente forma.

Acceso	Tipo
RESULT.Score	String
RESULT.Clases[0].Nombre	String
RESULT.Variables[0].Nombre	String
RESULT.Variables[0].Tipo	String
RESULT.Variables[0].Funcion	String
RESULT.Metodos[0].Nombre	String
RESULT.Metodos[0].Tipo	String
RESULT.Metodos[0].Parametros	Integer
RESULT.Metodos[0].Texto	String

Variables definidas para el lenguaje

Los tipos de variables que podrán 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"

Asignación de valores:

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

<nombre_de_la_variable> = <expresión> ;

Casteo automático de variables:

Los resultados de tipos para las operaciones con diferente tipo en cualquier orden son las siguientes. Para los operandos con tipos iguales se mantiene el tipo. Esto servirá a la hora de asignar un valor y comparar tipos a la variable a asignar.

Operación	Operandos	Resultado	Aplicación
suma	String+ Integer	String	"string"+5
resta	Integer– Integer	Integer	5-4
división	Integer / Integer	Integer	5/2
multiplicación	Integer*Integer	Integer	6*5

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 (/>)

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

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

Tabla No. 18

Descripción de la etiqueta for:

Esta etiqueta hace uso de dos variables de tipo Integer declaradas en el código, la primer (dada por iterador:VARIABLE) es estrictamente una variable definida en la sección de definición de variables la cual irá cambiando su valor en 1 en cada iteración del ciclo, y la segunda (Dada por hasta:Numérico) es estrictamente una variable numérica la cual nos indica hasta qué cantidad podría llegar nuestra variable iterador.

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: $$ ( RESULT.Score ) $$ </h2>
```

Al presionar el botón de Generar Reporte, se abrirá una ventana donde se verá el código utilizado.

También se puede editar el archivo json en su sección correspondiente.

Consola de reporte de errores:

La aplicación de usuario deberá cuenta con una consola en la cual se deben ver reflejados los errores en caso de que el análisis falle.