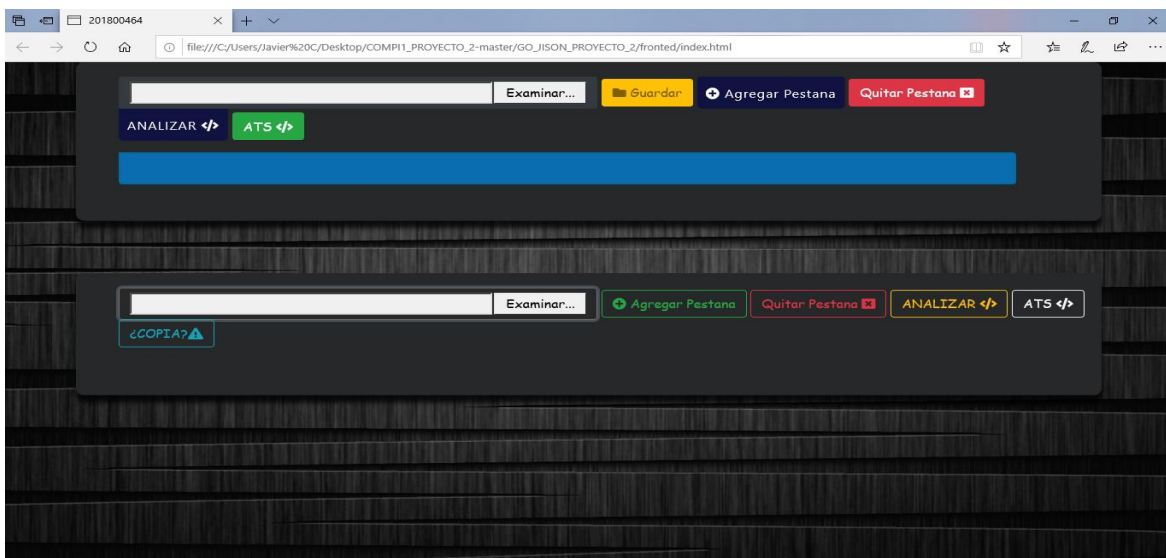
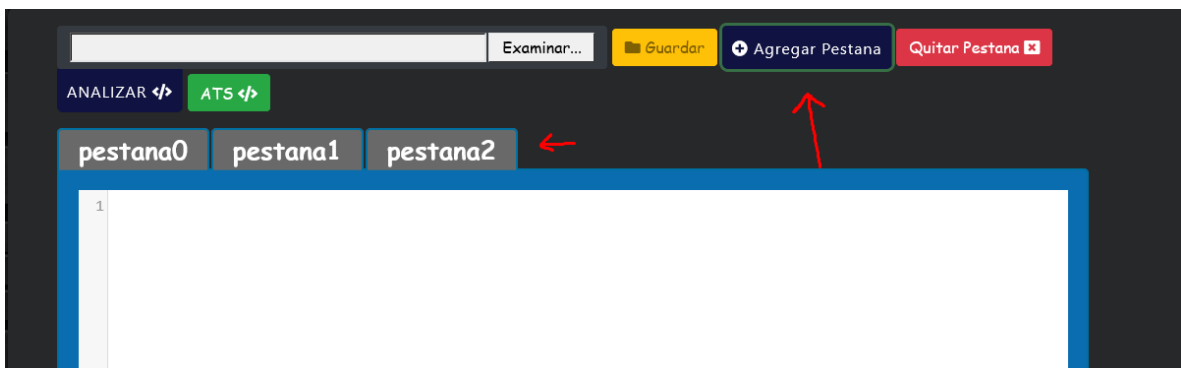


# MANUAL DE USUARIO

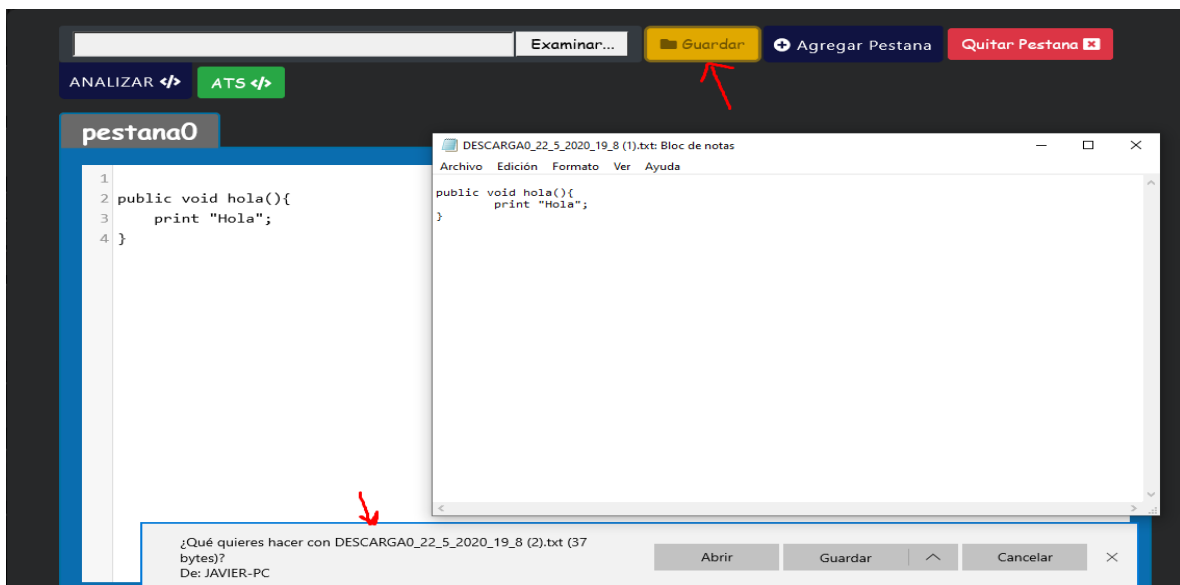
La aplicación presente es utilizada para manejar archivos.java y hacer el respectivo análisis semántico y sintáctico, para realizar dicho análisis se presiona el botón que dice análisis el cual es capaz de desplegar las listas de errores si los tuviera. Se muestra la interfaz inicial de la aplicación con algunas de las opciones que se tienen al cargar la página.



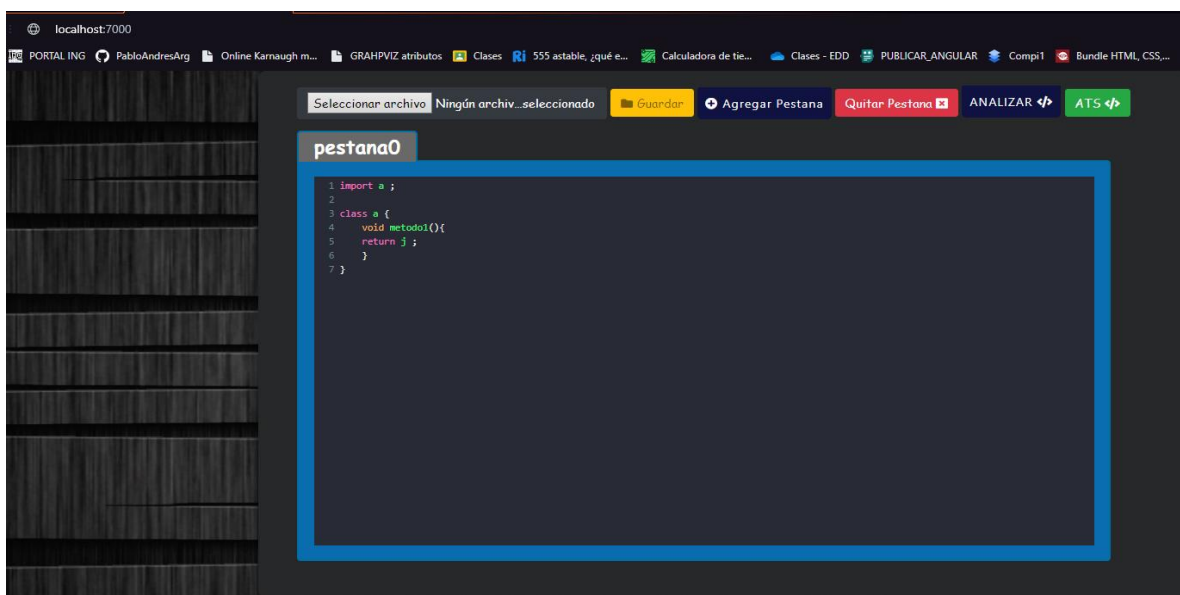
La aplicación cuenta con un apartado para agregar pestañas o quitar pestañas según el usuario lo necesite para que trabaje según los requerimientos.

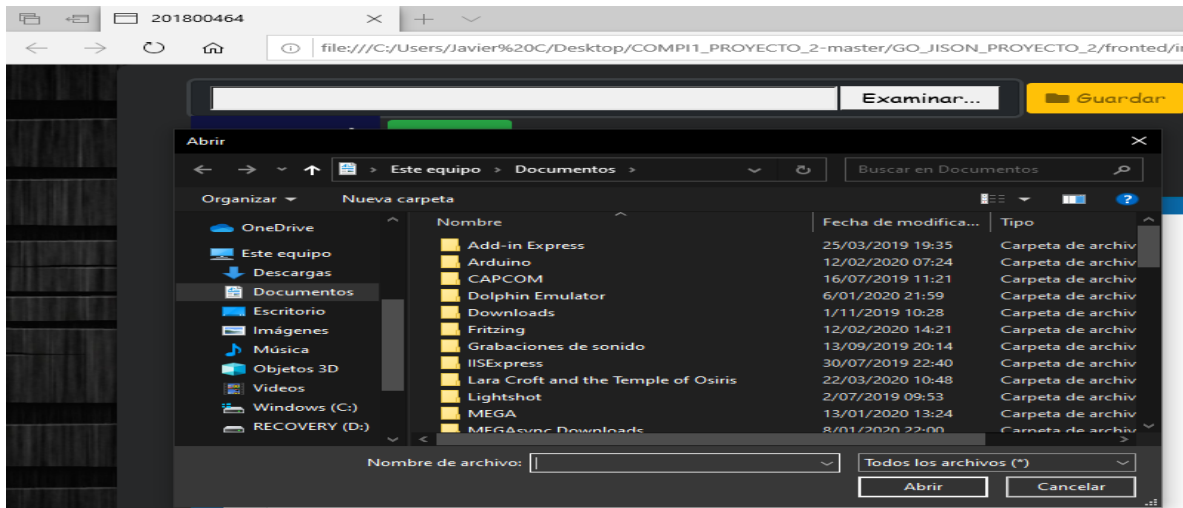


Al darle en el apartado Guardar se procede a extraer el texto contenido en la pestaña actual y se guardará en un archivo txt para que pueda ser editado posteriormente o manipularse según las necesidades del usuario.

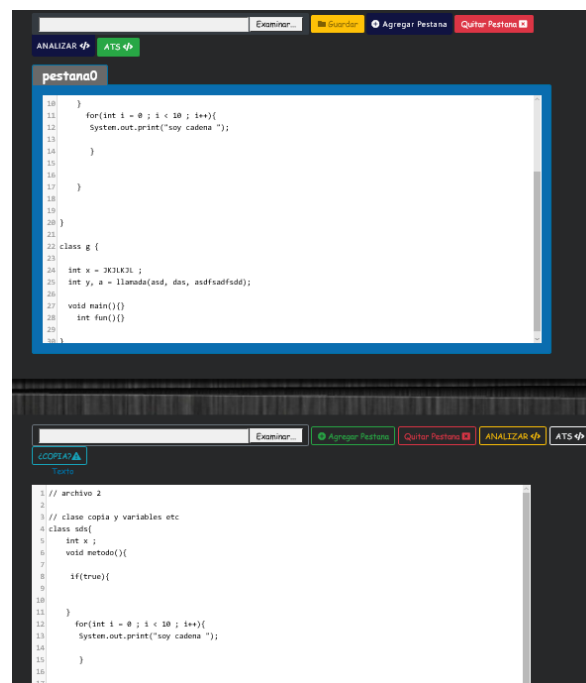


La aplicación posee una interfaz agradable al usuario y fácil de usar pues le proporciona abrir los archivos desde la misma app presionando el botón de “seleccionar archivo”.





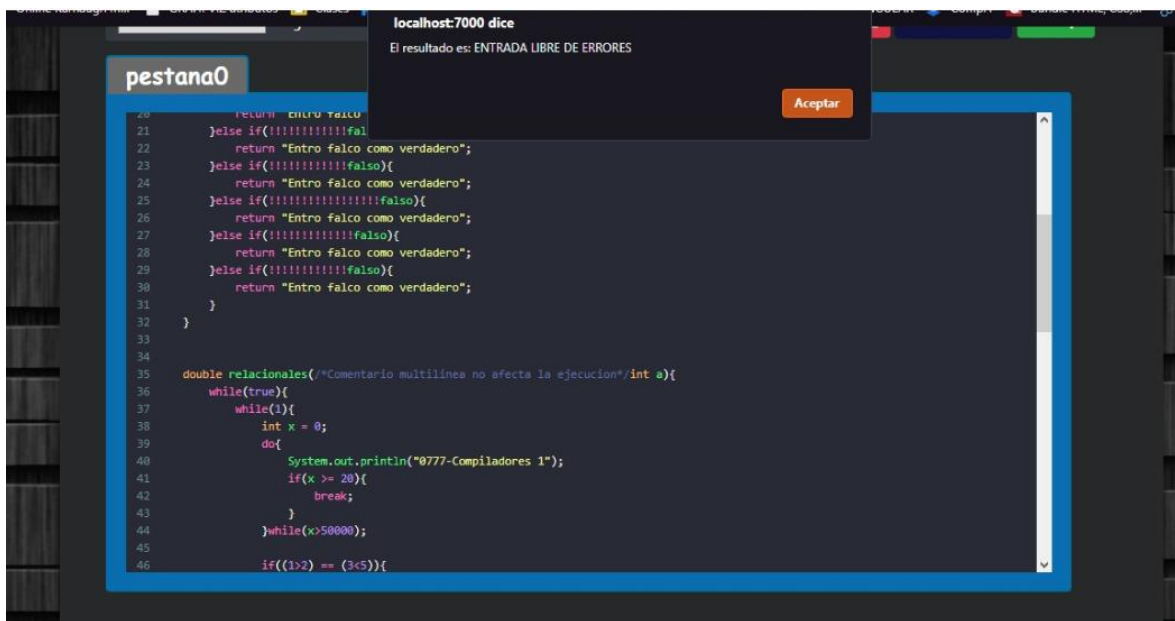
Se tienen dos secciones de pestañas en las cuales se puede ingresar texto para su análisis y el reporte de error si existieran siendo mostrados por medio de una tabla la cual indica la columna y la línea que contiene el error y el tipo de error presente, la aplicación cuenta con 3 tipos de errores que pueden ser identificados mediante el análisis léxico, sintáctico o semántico, aunque este último solo es aplicado para 4 sentencias las cuales son el break, continue y los retornos. Además, se tiene el apartado el ATS que muestra un árbol de visualización según los elementos incluidos informáticos que se encuentran en el editor de texto.



Al darle analizar a cualquiera de los dos apartados muestra un mensaje indicando el texto contenido en la pestaña.



Suponiendo que se cuenta con una entrada libre de errores entonces la página procederá a mostrar un mensaje indicando que la entrada se encuentra libre de errores.

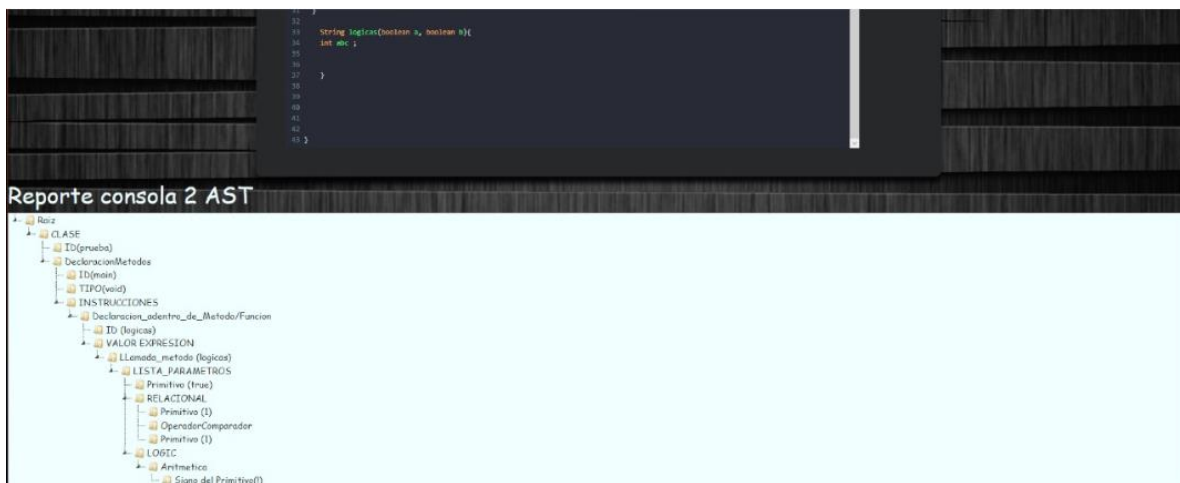


Caso contrario, si la entrada posee errores entonces se procederá a mostrar los errores encontrados durante la compilación, ya sea del tipo sintáctico, semántico o léxico descritos anteriormente con la columna y la fila en la cual ocurrió el error.

### Reporte de Errores de Compilacion

TIPO DE ERROR	DESCRIPCION	LINEA
Sintactico	error --> ; Columna:6	7
Sintactico	error --> errr Columna:15	7
Sintactico	error --> { Columna:6	9
Sintactico	error --> if Columna:7	9
Sintactico	error --> else Columna:0	11
Sintactico	error --> { Columna:4	11
Sintactico	error --> { Columna:0	13
Sintactico	error --> } Columna:4	14
Sintactico	error --> } Columna:0	15
Sintactico	error --> } Columna:1	15

Al solicitar el reporte de ATS de una entrada se mostrará un árbol de visualización con los apartados definidos en orden jerárquico.



Además, la aplicación cuenta con un apartado para la detección de copias entre dos archivos de entrada los cuales son los dos textos que se tienen en las pestañas de los apartados de la página principal.

```

36         return "Entro falso como verdadero";
37     }else if(!!!!!falso){
38         return "Entro falso como verdadero";
39     }else if(!!!!!falso){
40         return "Entro falso como verdadero";
41     }else if(!!!!!falso){
42         return "Entro falso como verdadero";
43     }else if(!!!!!falso){
44         return "Entro falso como verdadero";
45     }else if(!!!!!falso){
46         return "Entro falso como verdadero";

```

### REPORTE DE CLASES COPIA

#	NOMBRE DE LA CLASE	CANTIDAD DE METODOS
1	viendo	3

### REPORTE DE FUNCIONES COPIA

#	Nombre de la clase	Nombre Metodo/funcion	TIPO	Lista de Parametros	Tipo de retorno
1	viendo	x	double		no hay :double
2	viendo	relacionales	double	int ID	no hay :double
3	viendo	metodo_2	string	boolean ID	retorno de funcion (return EXPRESION;) :string

### REPORTE DE VARIABLES COPIA

#	TIPO	ID	NOMBRE METODO	NOMBRE CLASE
1	x	int	relacionales	viendo