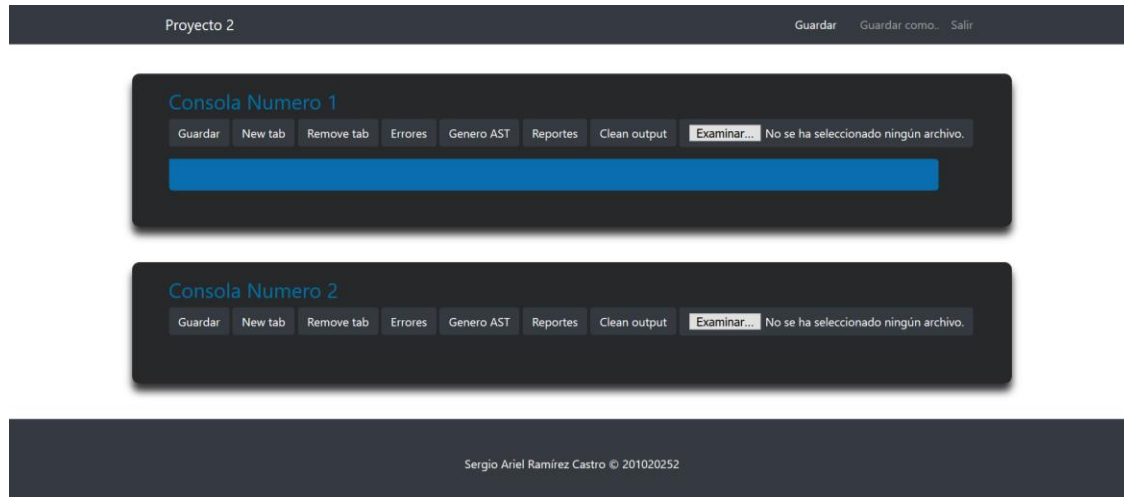
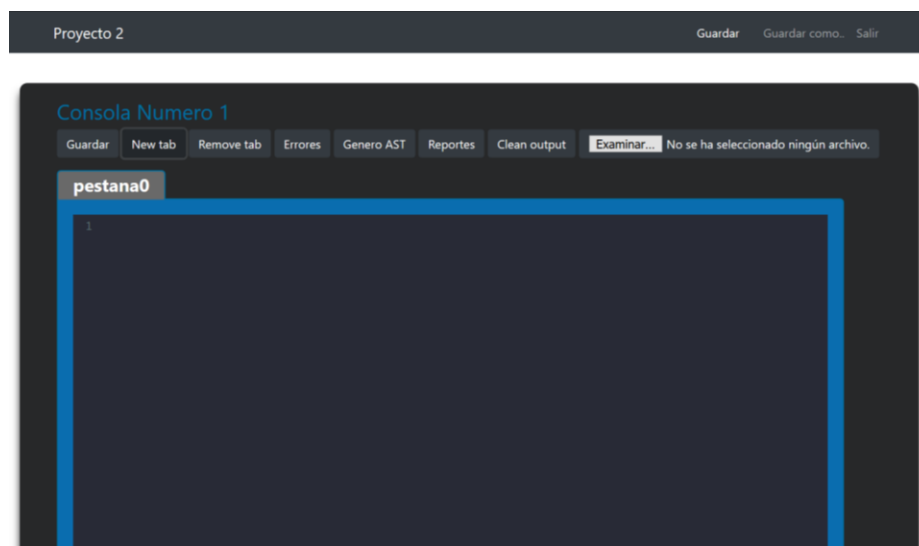


Manual de Usuario

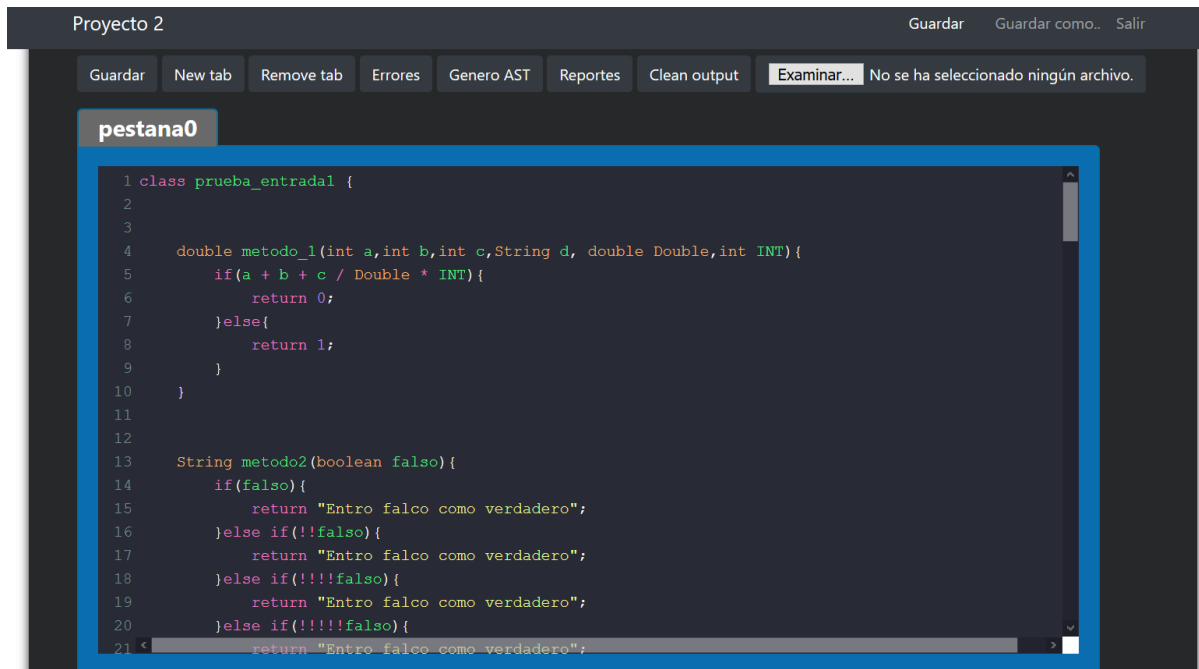
Ingresando a la direccion web podemos ver que tenemos dos consolas, las cuales podemos agregar pestaña, quitar pestaña, guardar y abrir documento



Agregamos una pestaña nueva pestaña y colocamos el texto que deseamos analizar primero

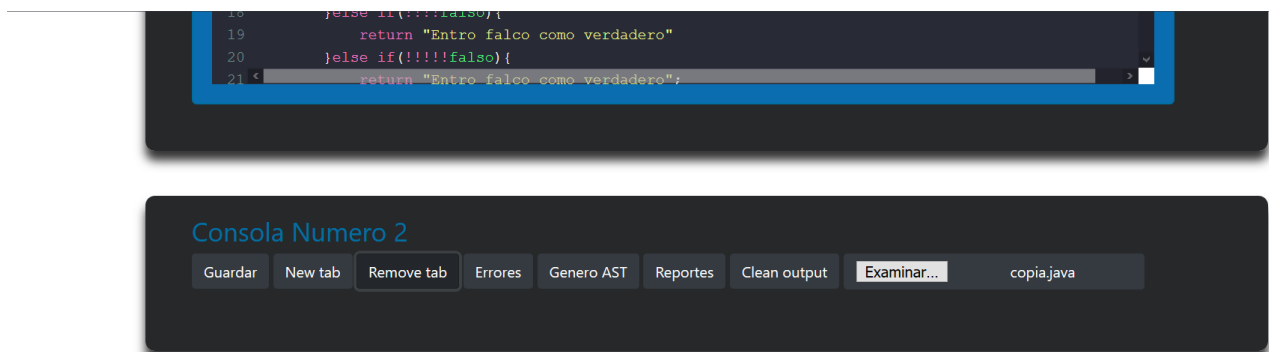


Codigo en Java que se va a analizar



```
1 class prueba_entrada1 {
2
3
4     double metodo_1(int a,int b,int c,String d, double Double,int INT){
5         if(a + b + c / Double * INT){
6             return 0;
7         }else{
8             return 1;
9         }
10    }
11
12
13    String metodo2(boolean falso){
14        if(falso){
15            return "Entro falco como verdadero";
16        }else if(!falso){
17            return "Entro falco como verdadero";
18        }else if(!falso){
19            return "Entro falco como verdadero";
20        }else if(!falso){
21            return "Entro falco como verdadero";
22    }
```

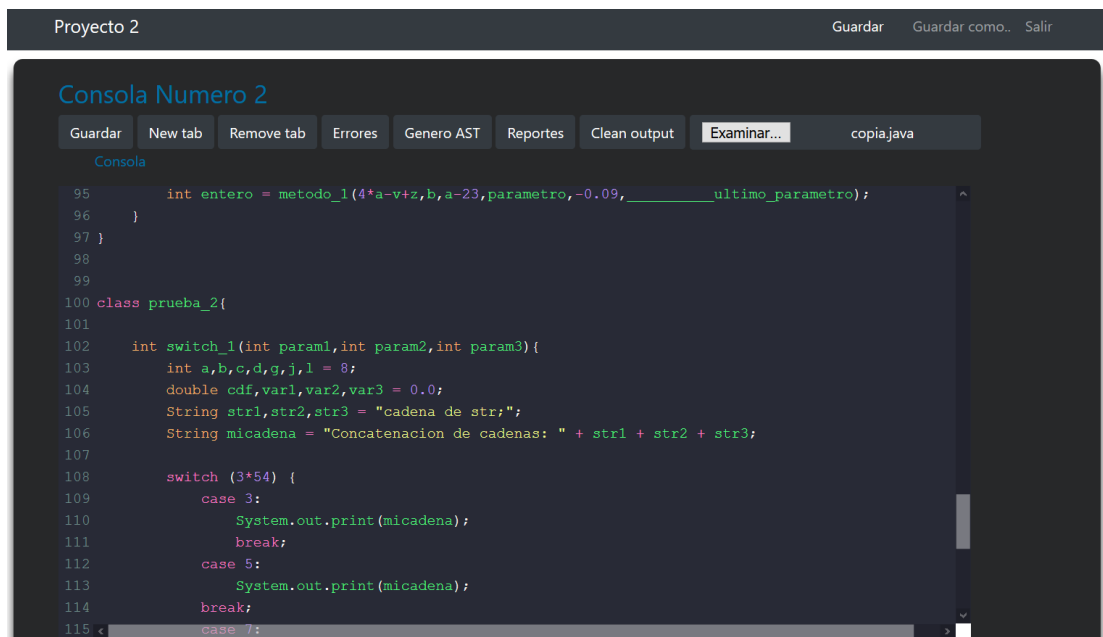
Acontinuacion presionamos el botón Errores , el cual va a analizar el código y principalmente a detectar errores dentro de código, si se ha detectado errores los mostrara en una tabla en la parte inferior de la pagina web



Errores Sintácticos y lexicos

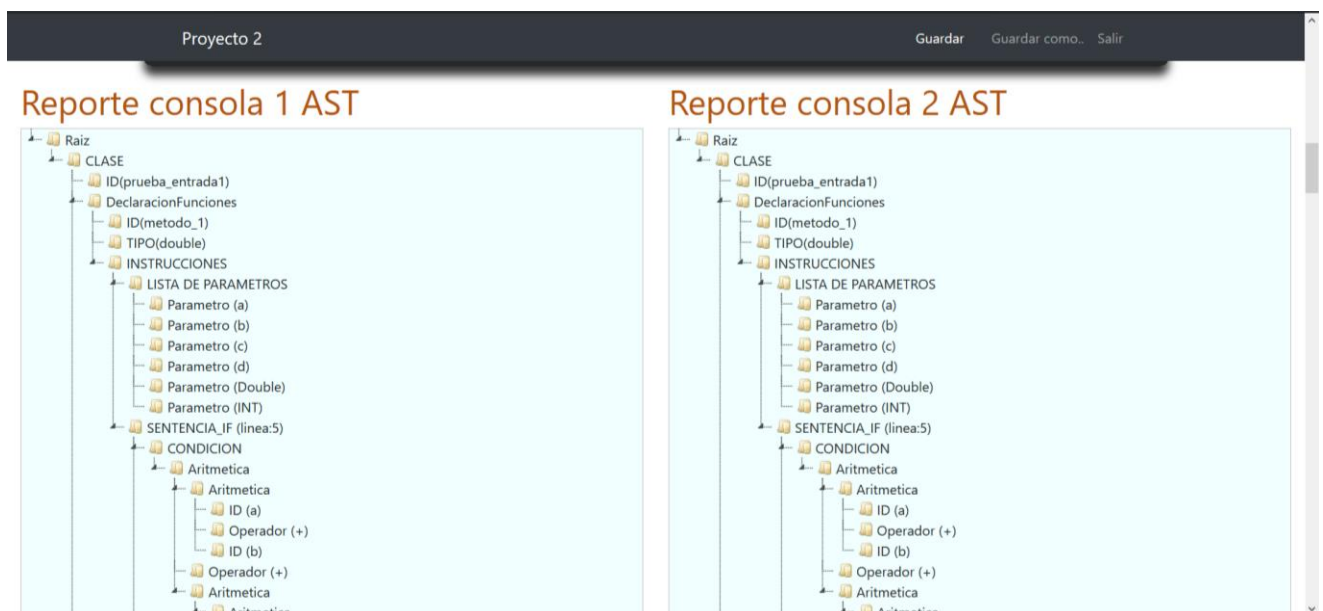
Tipo	Observacion	Linea
Sintactico	error::= "Entro falco como verdadero" Columna:8	20

Realizamos las correcciones en la línea que se detecto, posteriormente vamos a ingresar nuestro segundo código en la segunda consola



```
95     int entero = metodo_1(4*a-v+z,b,a-23,parametro,-0.09,ultimo_parametro);
96 }
97 }
98
99
100 class prueba_2{
101
102     int switch_1(int param1,int param2,int param3){
103         int a,b,c,d,g,j,l = 8;
104         double cdf,var1,var2,var3 = 0.0;
105         String str1,str2,str3 = "cadena de str;";
106         String micadena = "Concatenacion de cadenas: " + str1 + str2 + str3;
107
108         switch (3*54) {
109             case 3:
110                 System.out.print(micadena);
111                 break;
112             case 5:
113                 System.out.print(micadena);
114                 break;
115             case 7:
```

Podemos analizar el código y obtener de igual manera sus errores, si los codigos en las distintas consolas no tienen mas errores podemos proceder a presionar nuestro botón Generar AST el cual en la parte inferior de las consolas reflejara un árbol sintáctico el cual reflejara com esta conformado el código de una manera visual, este quedaría de la siguiente manera. Presionamos Generar AST en ambas consolas y visualizaremos el árbol de cada consola correspondiente



Podemos desplegar los nodos o ver de una manera reducida al dar click en cada carpeta

Proyecto 2

GuardarGuardar como...Salir

```
123 }
124
125 String potencia(double n1, double n2,String mensaje){
126     double resultado = n1^ n2;
127     return mensaje + resultado;
128 }
129
130 void main() {
131     potencia(0.1,5,"Este es el mensaje ");
132 }
133 }
```

Reporte consola 1 AST

Entrada Sin Errores

Reporte consola 2 AST

Entrada Sin Errores

Sergio Ariel Ramírez Castro © 201020252

Para nuestra tercera fase procedemos a realizar los reportes al presionar el botón de reportes

Proyecto 2

GuardarGuardar como...Salir

Copy class report

#	Name Class	No metodos
1	prueba_entrada1	5

Copy function report

#	Name Class	Name Method/Function	TIPO	List of Parameters	Type Return
1	prueba_entrada1	metodo_1	double	int ID int ID int ID string ID double ID int ID	retorno de funcion (return EXPRESION;) :double
2	prueba_entrada1	metodo2	string	boolean ID	retorno de funcion (return EXPRESION;) :string
3	prueba_entrada1	relacionales	double	int ID	retorno de funcion (return EXPRESION;) :double
4	prueba_entrada1	logicas	double	boolean ID boolean ID	DONT EXIST :double
5	prueba_entrada1	main	void		DONT EXIST :void

Copy function report

#	Name Class	Name Method/Function	TIPO	List of Parameters	Type Return
1	prueba_entrada1	metodo_1	double	int ID int ID int ID string ID double ID int ID	retorno de funcion (return EXPRESION;) :double
2	prueba_entrada1	metodo2	string	boolean ID	retorno de funcion (return EXPRESION;) :string
3	prueba_entrada1	relacionales	double	int ID	retorno de funcion (return EXPRESION;) :double
4	prueba_entrada1	logicas	double	boolean ID boolean ID	DONT EXIST :double
5	prueba_entrada1	main	void		DONT EXIST :void