

22-5-2020

PROYECTO 2

ANTENNAE



MANUAL DE USUARIO

Uzzi Libni Aaron Pineda Solorzano

201403541

Antes de usar el software lea cuidadosamente este instructivo

COMO EMPEZAR

La aplicación permite analizar un archivo de texto en formato .java el cual es la extensión de un archivo Java un famoso lenguaje de programación, esta pagina web es capaz de realizar un análisis léxico como sintáctico. Podemos observar en los resultados un apartado donde podemos mostrar los errores de dicho documento, acompañado de un árbol donde muestra la jerarquía de las operaciones ingresadas.

El fin de esta aplicación es poder visualizar en nivel de copia en un código por varios usuarios, al finalizar el análisis se muestra las instrucciones repetidas en dichos códigos.

INTERFAZ GRAFICA

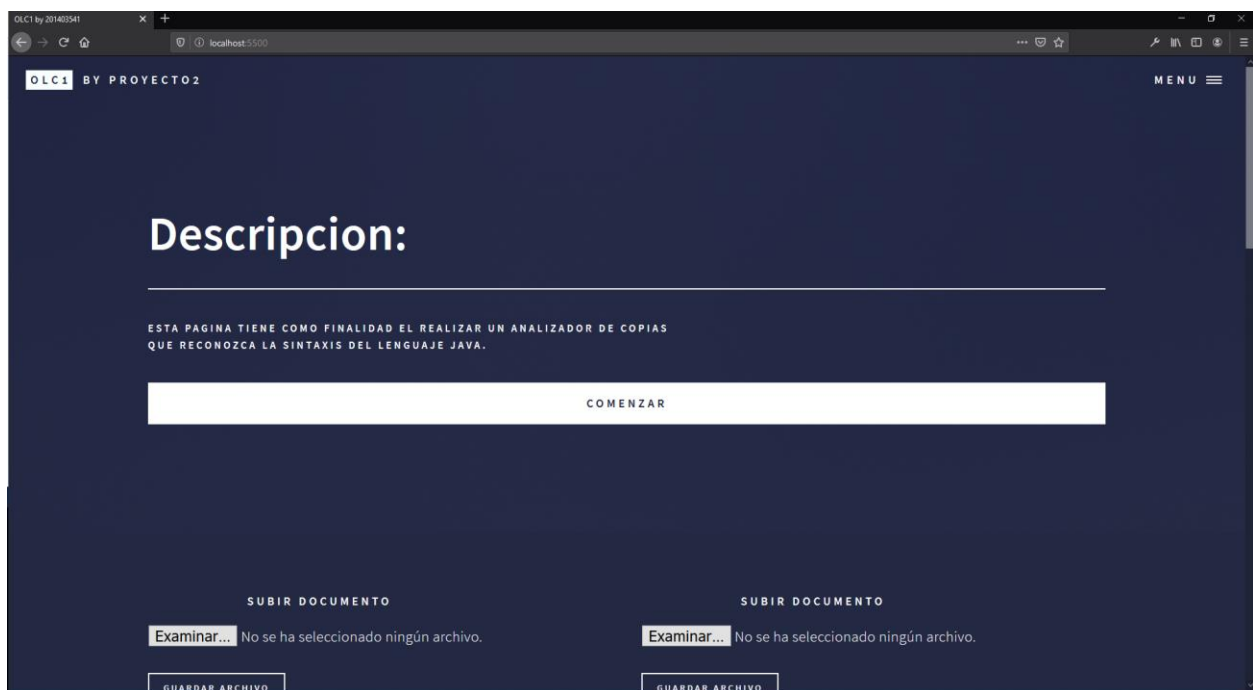


FIGURA NO.1

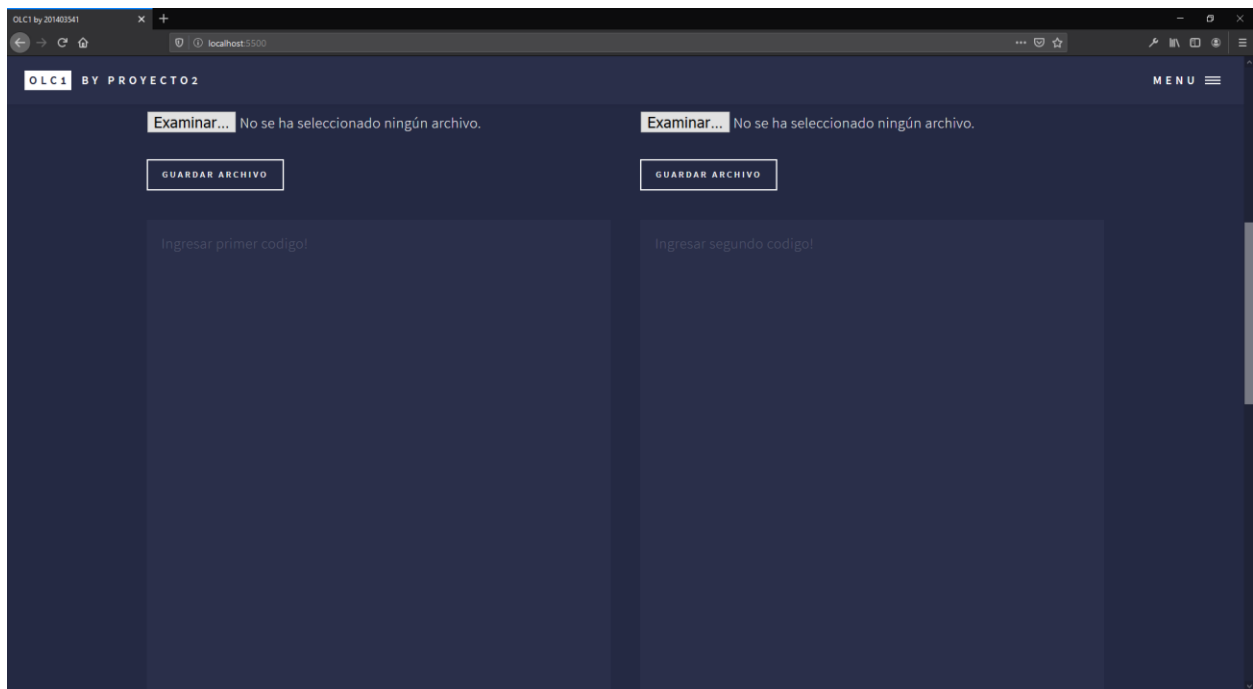


FIGURA NO.2

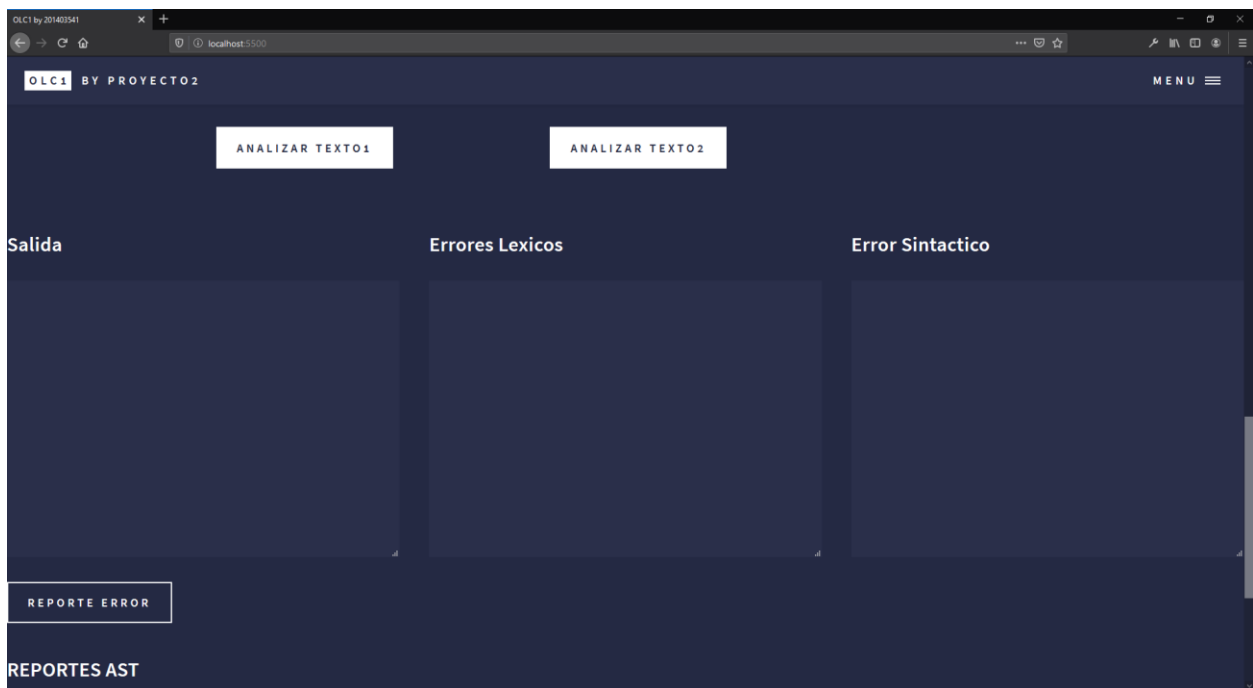


FIGURA NO.3

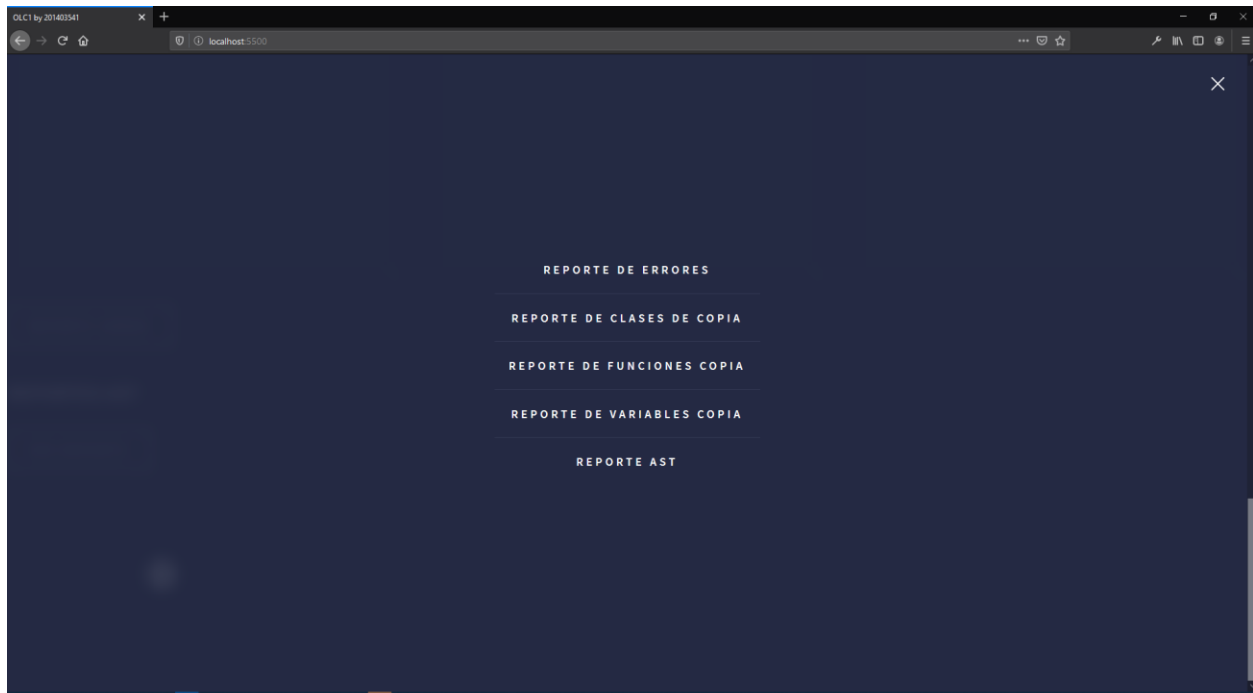


FIGURA NO.4

De las FIGURAS (1-4) encontramos las diferentes funciones que tiene dicha aplicación. A continuación, procedemos a explicar las funciones:

MENU: (FIGURA NO.4)

Al presionar encontramos un listado de reportes que la aplicación puede realizar.

BOTÓN GUARDAR: (FIGURA NO.2)

Permiten guardar el archivo de entrada con extensión .java en dado caso se haya hecho algún cambio.

BOTÓN ANALIZAR: (FIGURA NO.3)

Analiza léxica y sintácticamente los archivos de entrada el cual permite generar distintos reportes mostrados en la pestaña salida, mostrando también sus errores léxicos y sintácticos en sus respectivas áreas (FIGURA NO.3).

REPORTES: (FIGURA NO.3)

El botón reporte AST genera el respectivo árbol de análisis sintáctico del archivo 1 enviado, así como el botón REPORTE ERROR permite descargar un archivo html con una tabla de errores sintácticos y léxicos, el botón COPIA analiza los distintos archivos mostrando sus resultados en la pestaña salida. El botón funciones copia genera el

reporte en la pestaña salida mostrando cada una de sus funciones y métodos que son copia con su nombre y parámetros y cada uno de sus atributos.

EJEMPLO DE ARCHIVO DE ENTRADA:

ARCHIVO DE ENTRADA

*****COMENTARIO*****

[illegible]

\$Esto es un comentario multilinea y puede venir en cualquier parte del archivo de entrada\$

*****/

```
import java.io.util;  
  
import java.util;  
  
import Crear_Arbol.Arbol;  
  
import GraficarArbol.Control;  
  
import GraficarArbol.GraficarA;  
  
import Listas.ListaError;  
  
import Listas.ListaToken;  
  
import java.awt.image.RenderedImage;  
  
import java.io.File;  
  
import java.io.FileOutputStream;  
  
import java.io.IOException;  
  
import java.util.Stack;  
  
import javax.imageio.ImageIO;  
  
import javax.swing.JOptionPane;  
  
/**
```

*

* @author libni

*/

class Token{

String jo;

void Scanner(String cadena){

System.out.println(hola + mundo);

int juan, a, f = 1, g;

}

void Prueba(int a, int b){

System.out.print();

}

int Prueba1(boolean k, String cadena, int b, char car, double pi){

switch(estado)

{

case 1:

System.out.print();

break;

case 2:

break;

case 3:

System.out.print();

```
System.out.print();  
break;  
case 4:  
System.out.print();  
break;  
case 5:  
for(int j = 0; j == 10; i = i+1)  
{  
    System.out.print("LIBNI");  
}  
break;  
case 6:  
System.out.print();  
System.out.print();  
System.out.print();  
break;  
case 7:  
if(false)  
{  
  
}  
break;  
default:  
while(true)  
{
```

```

        System.out.println();
        do{}while(false);
    }
    do{
        if(true)
        {

        }
    }while(false);
    break;
}

    return hola;
}

String Prueba2(char ca){
    if(true)
    {
        if(a == b)
        {
            if(a != c && a > c || true)
            {
                for(int a = 0; 5 < a; i=i+1)
                {
                    for( int j = a; a <= j; j=j+1)

```



```
        {  
            }  
        }  
    }  
    }  
    }  
    }  
    }  
    return null;  
}  
}
```

```
class Prueba{  
  
}
```