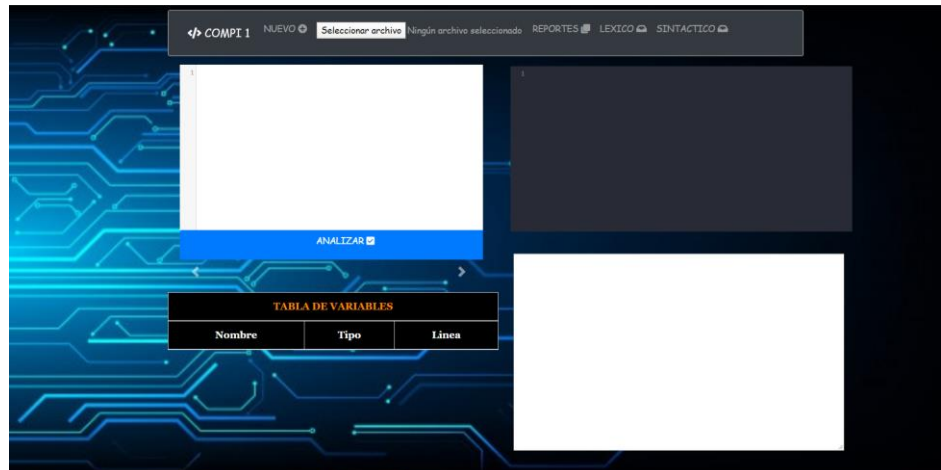
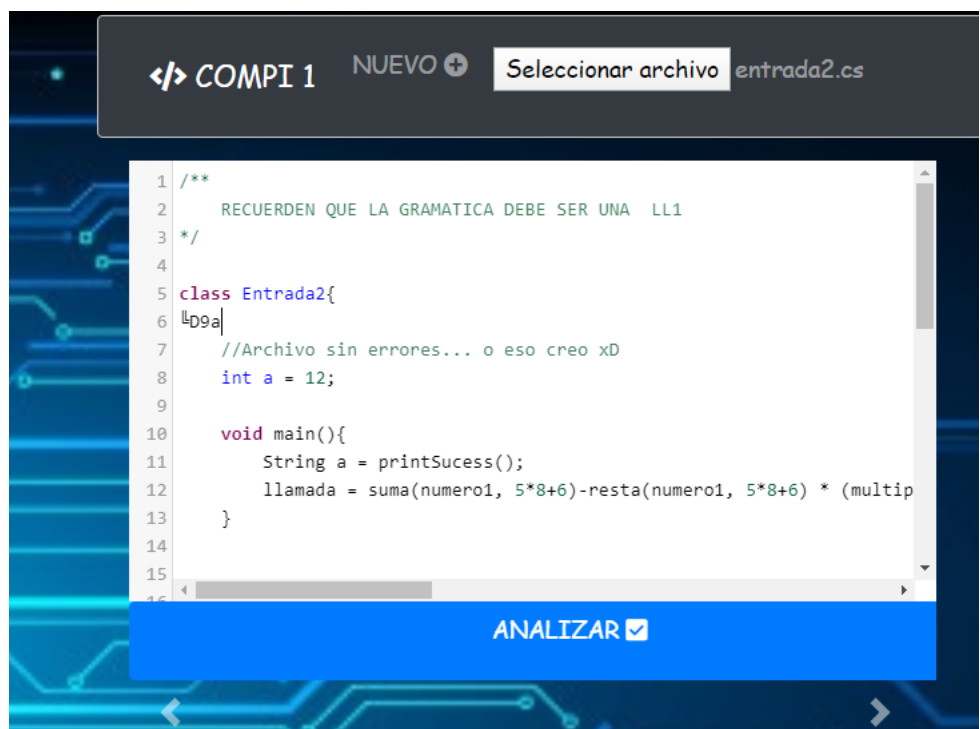


MANUAL DE USUARIO

Se muestra una interfaz agradable al usuario donde podrá encontrar fácilmente el botón para analizar la entrada, obviamente llenando primero el cuadro en blanco encontrado en la parte superior izquierda de la interfaz.



También cuenta con un apartado para cargar archivos el cual aparece a continuación es el botón que dice "seleccionar archivo" este cargara de la siguiente manera.



Si el archivo cuenta con errores léxicos sean estos caracteres no pertenecientes a dicho lenguajes serán mostrados al hacer clic en la ventana superior derecha la cual dice “Léxico”

Esta sección cuenta con los Tokens y los errores a la vez.

175	.	punto	34	10
176	Write	Propiedad Write	34	15
177	(Parentesis_izquierdo	34	16
178	'<html><head><title>Example 1</title></head><body style="background: skyblue"><h2>[OLC1]Practica 2</h2><p>Si sale compi 1 :) html sin errores...!!!</p></body></html>'	Cadena HTML	34	196
179)	Parentesis Derecho	34	197
180	;	Punto_y_coma	34	198
181	}	llave_Derecha	35	2
182	}	llave_Derecha	37	1

LISTA DE ERRORES				
#	Valor	Tipo	Fila	Columna
1	§	ERROR LEXICO	3	1
2	ℓ	ERROR LEXICO	5	1
3	¶	ERROR LEXICO	5	3

En la barra de herramientas se cuenta con un regreso al ingresar a cada opción si se moviera de pestañas, la aplicación posee un apartado para errores sintácticos correspondiente al orden de los tokens.

<div> COMPI 1 <div> BACK LEXICO SINTACTICO </div> </div>				
LISTA DE ERRORES SINTACTICOS				
#	Valor	Tipo	Fila	Columna
1	se esperaba llave_Derecha en lugar de ID	Error Sintactico	5	2

Si la aplicación determina que todo está correcto se pasa a la respectiva traducción del código de c# a Python la cual se muestra en la parte derecha de la interfaz principal.

```
9
10     var a = 12
11     def main():
12
13         var a = printSucess()
14         llamada=suma(numero1,5*8+6)-resta(numero1,5*8+6)*(multiplicacion(numero1
15
16         if __name__ == "__main__":
17             main()
18
19     def suma(n1,n2):
20         return n1+n2
21
22     def resta(n1,n2):
23         return n1-n2
```

A su vez en paralelo ese generara el HTML el cual produce un JSON en la pestaña siguiente.

```
1
2     <HTML>
3
4     <head>
5
6     <TITLE>
7         SOLO PASANDO A LA CONSOLA
8     </TITLE>
9
10    </head>
11
12    <BODY style = "background:white">
13
14        <DIV style = "background:yellow">
15            centro de la app
16        </DIV>
```

Al hacer clic en el botón que corre a la siguiente pestaña se mostrara lo siguiente:

```
"HTML":{
  "head":{
    "TITLE":{
      "TEXT":"SOLO PASANDO A LA CONSOLA"
    }
  },
  "BODY":{
    "STYLE":"background:white",

    "DIV":{
      "STYLE":"background:yellow",
      "TEXT":"centro de la app"
    },
    "ul":{
      "li":{
        "TEXT":"HOLA"
      }
    }
  }
}
```

La aplicación tiene la capacidad de analizar las variables que fueron declaradas adentro del código de c# , esta se muestra abajo del cuadro donde ingreso la entrada de su código.

TABLA DE VARIABLES		
Nombre	Tipo	Línea
a	int	2
b	int	2
c	int	2
c	int	5