

ANALIZADOR LEXICO

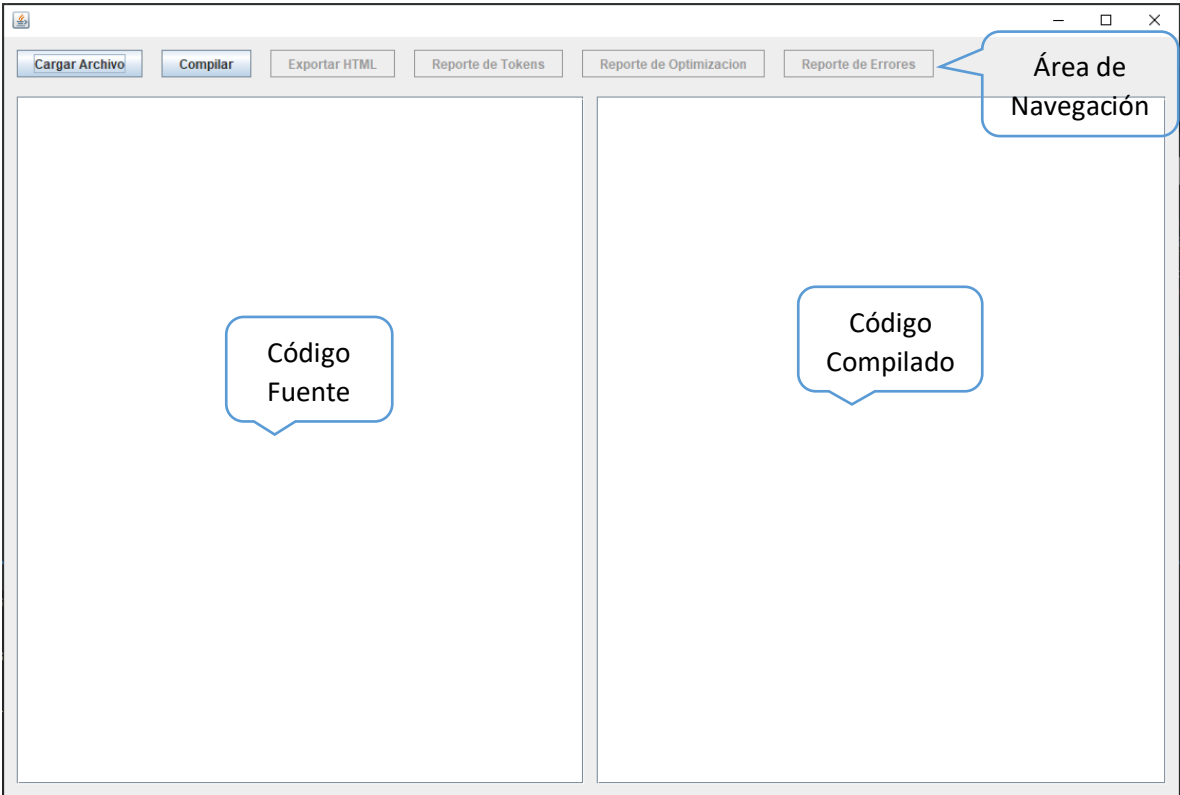
MANUAL DE USUARIO

Descripción de la Aplicación

La aplicación consiste en una Aplicación de Analizador Léxico Inspirado en Visual Basic, dentro de la aplicación se puede introducir “código fuente”, y se podrá hacer acciones tales como pintar un Lienzo de acuerdo a ciertos parámetros de Colores, como también poder exportar el Lienzo como una Imagen y un par de funcionalidades más que se explicaran más adelante.

Inicio de la Aplicación

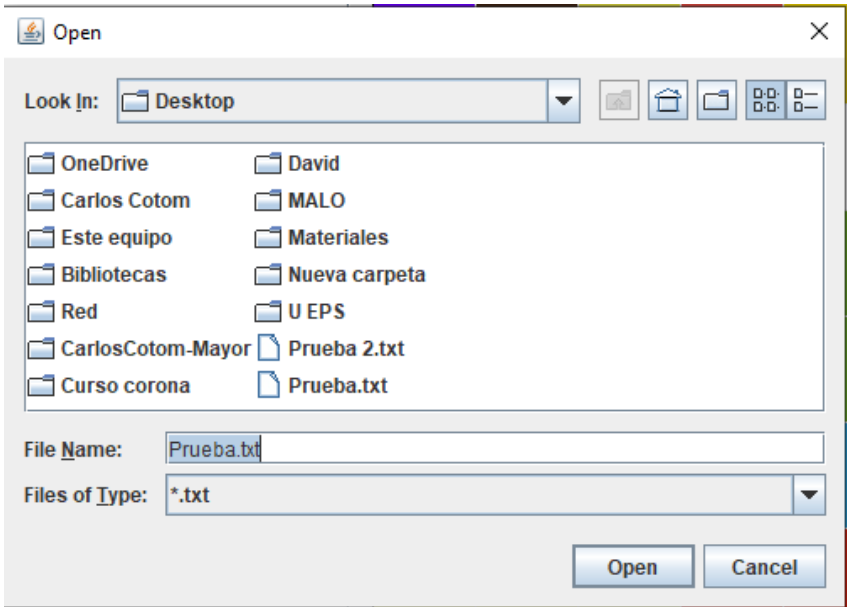
Al iniciar la aplicación se mostrara la siguiente ventana:



Área de Navegación

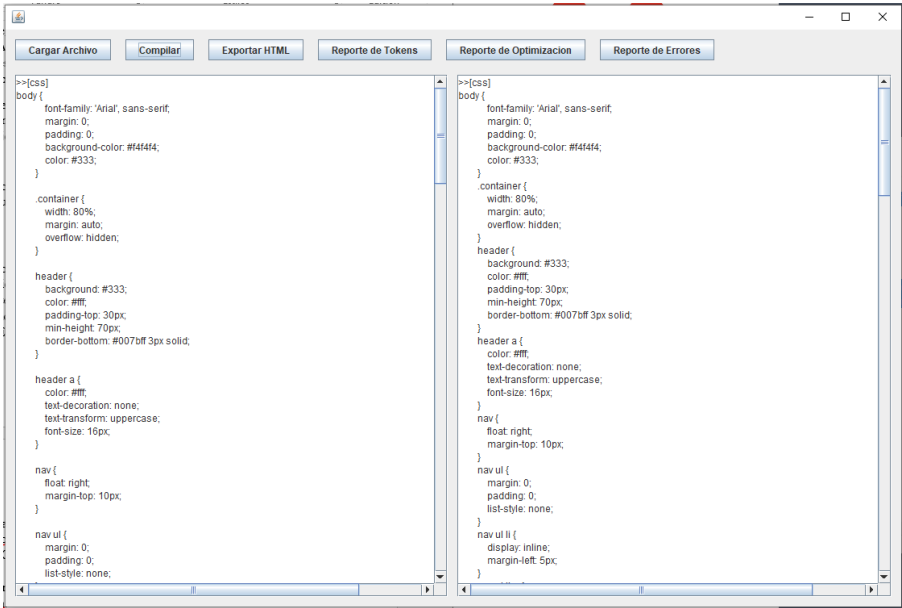
El área de navegación consiste de las siguientes opciones:

- Carga de Archivo: En esta opción se desplegara la siguiente ventana:

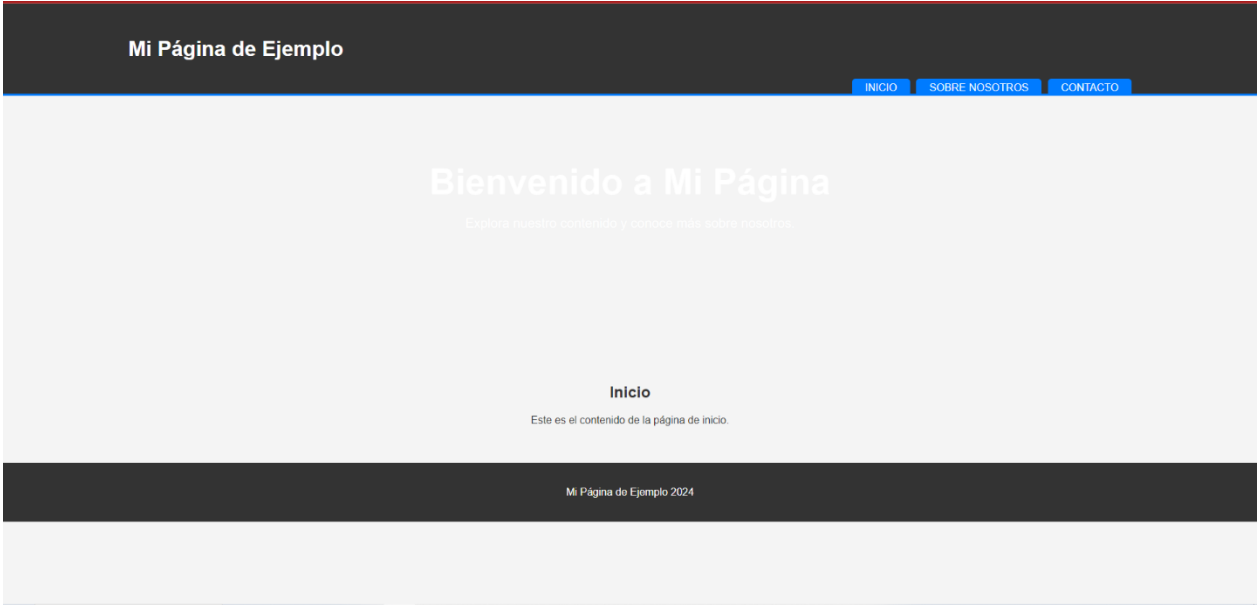


Aquí es en donde se debe elegir el Archivo de Texto que contenga el código fuente para poder introducirlo dentro de la Aplicación y así poder hacer las acciones necesarias que se requieran con el mismo. Cabe resaltar que el archivo aceptado por la Aplicación es de texto plano, es decir, un archivo con extensión .txt

- **Compilar:** Al presionar sobre esta opción, se podrá analizar el código fuente que se haya introducido o cargado, al terminar con el análisis y si el código fuente ingresado cumple con la sintaxis de los lenguajes, entonces se mostrara el código compilado



- **Exportar HTML:** En esta opción se podrá exportar el código compilado a un archivo HTML funcional para la visualización en la web.



- **Reporte de Tokens:** En esta opción se desplegara una tabla con información relevante sobre los tokens que fueron identificados en el código fuente ya compilado. Las columnas informativas son las siguientes:
 - **Token:** Es el Contenido del Token que fue identificado.
 - **Expresión Regular:** Es la representación del Token en forma de Expresión Regular
 - **Lenguaje:** Identifica a que Lenguaje pertenece el Token analizado
 - **Tipo:** Es el tipo de Token que fue identificado y que corresponde a cada lenguaje
 - **Fila:** Es la Fila en el Código Fuente ya Compilado en donde fue identificado el Token.
 - **Columna:** Es la Columna en el Código Fuente ya Compilado en donde fue identificado el Token.

Reporte Tokens

Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
>>[css]	>>[css]		Estado	0	0
body	body	CSS	ETIQUETA_TIPO	1	0
{	{	CSS	OTROS	1	5
font-family	font-family	CSS	REGLA	2	12
:	:	CSS	OTROS	2	23
'Arial'	'([a-zA-Z][0-9][.])'	CSS	CADENA	2	25
,	,	CSS	OTROS	2	32
sans-serif	[a-z]+[0-9]*(-(a-z)[0-9])...	CSS	IDENTIFICADOR	2	34
;	;	CSS	OTROS	2	44
margin	margin	CSS	REGLA	3	12
:	:	CSS	OTROS	3	18
0	[0-9]+	CSS	ENTERO	3	20
;	;	CSS	OTROS	3	21
padding	padding	CSS	REGLA	4	12
:	:	CSS	OTROS	4	19
0	[0-9]+	CSS	ENTERO	4	21
;	;	CSS	OTROS	4	22

Exportar a HTML

Se tiene la opción de **Exportar HTML** para que se pueda exportar el reporte a formato html para su visualización en la web.

Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
>>[css]	>>[css]		Estado	0	0
body	body	CSS	ETIQUETA_TIPO	1	0
{	{	CSS	OTROS	1	5
font-family	font-family	CSS	REGLA	2	12
:	:	CSS	OTROS	2	23
'Arial'	'([a-zA-Z][0-9][.])'	CSS	CADENA	2	25
,	,	CSS	OTROS	2	32
sans-serif	[a-z]+[0-9]*(-(a-z)[0-9])+*	CSS	IDENTIFICADOR	2	34
;	;	CSS	OTROS	2	44
margin	margin	CSS	REGLA	3	12
:	:	CSS	OTROS	3	18
0	[0-9]+	CSS	ENTERO	3	20
;	;	CSS	OTROS	3	21
padding	padding	CSS	REGLA	4	12
:	:	CSS	OTROS	4	19
0	[0-9]+	CSS	ENTERO	4	21
;	;	CSS	OTROS	4	22
background-color	background-color	CSS	REGLA	5	12
:	:	CSS	OTROS	5	28
#f4f4f4	#f4f4f4	CSS	COLOR	5	30
;	;	CSS	OTROS	5	37
color	color	CSS	REGLA	6	12

- Reporte de Optimización: En esta opción se desplegara una tabla con información relevante sobre los tokens que fueron identificados para la optimización en el código fuente. Las columnas informativas son las siguientes:
 - Token: Es el Contenido del Token que fue identificado.
 - Expresión Regular: Es la representación del Token en forma de Expresión Regular
 - Lenguaje: Identifica a que Lenguaje pertenece el Token analizado
 - Tipo: Es el tipo de Token que fue identificado y que corresponde a cada lenguaje
 - Fila: Es la Fila en el Código Fuente ya Compilado en donde fue identificado el Token.
 - Columna: Es la Columna en el Código Fuente ya Compilado en donde fue identificado el Token.

Reporte de Errores:

Reporte de Optimizacion

Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
background	background	CSS	REGLA	58	12
:	:	CSS	OTROS	58	22
url(url(CSS	ERROR	58	24
https	[a-z]+[0-9]*(-([a-z]([0-9]))+)*	CSS	IDENTIFICADOR	58	29
:	:	CSS	OTROS	58	34
//via.placeholder.com/1...	//([a-zA-Z]([0-9]) .)*	CSS	COMENTARIO	58	35

Exportar a HTML

Se tiene la opción de **Exportar HTML** para que se pueda exportar el reporte a formato html para su visualización en la web.

Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
background	background	CSS	REGLA	58	12
:	:	CSS	OTROS	58	22
url(url(CSS	ERROR	58	24
https	[a-z]+[0-9]*(-([a-z]([0-9]))+)*	CSS	IDENTIFICADOR	58	29
:	:	CSS	OTROS	58	34
//via.placeholder.com/1200x400) no-repeat center center/cover;	//([a-zA-Z]([0-9]) .)*	CSS	COMENTARIO	58	35

SINTAXIS DE LENGUAJES

LENGUAJE HTML

Etiquetas

Etiqueta	Traducción	Tipo	Observación
<principal>	<main>	Apertura	
<principal/>	</main>	Cierre	
<encabezado>	<header>	Apertura	
<encabezado/>	</header>	Cierre	
<navegacion>	<nav>	Apertura	
<navegacion/>	</nav>	Cierre	
<apartado>	<aside>	Apertura	
<apartado/>	</aside>	Cierre	
<listaordenada>		Apertura	
<listaordenada/>		Cierre	
<listadesordenada>		Apertura	
<listadesordenada/>		Cierre	
<itemlista>		Apertura	
<itemlista/>		Cierre	
<anclaje>	<a>	Apertura	
<anclaje/>		Cierre	
<contenedor>	<div>	Apertura	
<contenedor/>	</div>	Cierre	
<seccion>	<section>	Apertura	
<seccion/>	</section>	Cierre	
<articulo>	<article>	Apertura	
<articulo/>	</article>	Cierre	
<titulo#>	<h#>	Una Línea	Solo se deben admitir de h1 a h6. Donde # es el valor jerárquico del título
<parrafo>	<p>	Apertura	
<parrafo/>	</p>	Cierre	
		Apertura	
		Cierre	
<entrada/>	<input/>	Una Línea	
<formulario>	<form>	Apertura	
<formulario/>	</form>	Cierre	
<label>	<label>	Apertura	
<label/>	</label>	Cierre	
<area/>	<textarea/>	Una Línea	
<boton>	<button>	Apertura	
<boton/>	</button>	Cierre	
<piepagina>	<footer>	Apertura	
<piepagina/>	</footer>	Cierre	

Palabras Reservadas

Datos

Token	Traducción	Observación	Nombre	Ejemplo	Observación
Palabra Reservada	<ul style="list-style-type: none">class=hrefonClickidstyletypeplaceholderrequiredname	Solo pueden estar dentro de una etiqueta de apertura o de una línea.	cadena	"Hola", "1234", "cualquier conjunto caracteres"	
			Texto	Puede ser cualquier conjunto de caracteres	<p>Este token solo es válido si no está dentro de las propiedades de la etiqueta, de lo contrario se tomaría como un error.</p> <p>Ejemplo:</p> <p><contenedor Esto es texto>: Esto sería un error</p> <p><contenedor> Esto es texto </contenedor>: Esto no sería un error</p>

LENGUAJE CSS

Selectores

Nombre	Token	Observación
Universal	*	
Etiqueta o Tipo	<ul style="list-style-type: none">bodyheadermainnavasidedivulolliah1, h2, ..., h6pspanlabel	
	<ul style="list-style-type: none">textareabuttonsectionarticlefooter	
De clase	•[a-z]+ [0-9]* (- ([a-z] [0-9]))+)*	<p>El selector de clase tiene estas condiciones</p> <ul style="list-style-type: none">Inicia siempre con un punto ('.').Seguido de una letra minúsculaSolo acepta letras minúsculasLuego de la primera letra, pueden seguir números o letras.Guión medio para separar las palabras de la clase. <p>Ejemplo: .container, .elemento, .danger-white1, .badge1-red-gren200</p> <p>Será una notación kebab-case.</p>
De Id	#[a-z]+ [0-9]* (- ([a-z] [0-9]))+)*	<p>El selector de id tiene estas condiciones.</p> <ul style="list-style-type: none">Inicia siempre con un punto ('#').Seguido de una letra minúsculaSolo acepta letras minúsculasLuego de la primera letra, pueden seguir números o letras.Guión medio para separar las palabras de la clase. <p>Ejemplo: #id-del-elemento, #identifier, #danger-white1, #badge1-red-gren200</p>

Combinadores	<ul style="list-style-type: none">• >• +• ~• Espacio en blanco, este combinador tiene solo un espacio en blanco entre dos selectores, ejemplo:<ul style="list-style-type: none">◦ div p◦ nav li◦ p span	Ejemplos: <ul style="list-style-type: none">• div > p• nav + li• p ~ span• .h1 span
Reglas	<ul style="list-style-type: none">• color• background-color• background• font-size• font-weight• font-family• font-align• width• height• min-width• min-height• max-width• max-height• display• inline• block• inline-block• flex• grid• none• margin• border• padding• content• border-color• border-style• border-width• border-top, border-bottom, border-left, border-right• box-sizing• border-box	
	<ul style="list-style-type: none">• position• static, relative, absolute, sticky, fixed• top• bottom• left• right• z-index• justify-content• align-items• border-radius• auto• float• list-style• text-align• box-shadow•	

Otros	<ul style="list-style-type: none">• px• %• rem• em• vw• vh• :hover• :active• :not()• :nth-child()• odd• even• ::before• ::after,• :• ;• ,• '[A-Za-z]'• (•)	
Cadena	'Arial', `Roman-Is`	Una cadena debe de estar entre comillas simples y dentro puede contener cualquier conjunto de caracteres.
Colores	<ul style="list-style-type: none">• #fff, #ffffff• rgba(3, 0, 1)• rgba(2, 2, 2, 0.1)	<ul style="list-style-type: none">• Los colores pueden ser hexadecimales, tanto de tres valores
		como de seis. <ul style="list-style-type: none">• El segundo formato es el rgba, que debe contener tres parámetros enteros, y uno extra opcional que puede ser decimal o entero.
Identificador	[a-z]+ [0-9]* (- ([a-z] [0-9]))+)*	

LENGUAJE JAVA SCRIPT

Nombre	Token o Expresión Regular	Observación
Identificador	[a-zA-Z]([a-zA-Z] [0-9] [_])*	
Aritméticos		
Suma	+	
Resta	-	
Multiplicación	*	
División	/	
Relacionales		
Igual	==	
Menor que	<	
Mayor que	>	
Menor o igual que	<=	
Mayor o Igual que	>=	
Diferente	!=	
Lógicos		
Or		
And	&&	
Not	!	

Incrementales		
Incremento	++	
Decremento	--	
Tipos de Datos		
Entero	[0-9]+	
Decimal	[0-9]+ . [0-9]+	
Cadena	“Cualquier grupo Caracteres”, 'Cualquier grupo Caracteres', `Cualquier grupo Caracteres`	Una cadena puede estar entre comillas simples, dobles y `` , y dentro puede tener cualquier tipo de carácter
Booleano	true	
Booleano	false	
Palabras Reservadas		
Palabra Reservada	<ul style="list-style-type: none">• function• const• let• document• event• alert• for• while• if, else• return• console.log• null	
Otros		
Paréntesis	<ul style="list-style-type: none">• (•)	
Corchetes	<ul style="list-style-type: none">• [•]	
Llaves	<ul style="list-style-type: none">• {• }	
Asignación	=	
Punto y Coma	;	
Coma	,	
Punto	.	
Dos Puntos	:	

La sintaxis en Común para los tres lenguajes será el del Comentario

Nombre	Ejemplo	Observación
Comentario	// Este es un comentario	El comentario inicia con dos diagonales //, seguido de cualquier carácter.