

Manual Usuario
Luis Regalado 202131920
Analizador Léxico y Sintáctico

Dinámica

El proyecto es un Analizador Léxico y Sintáctico que permite a los usuarios procesar código fuente, identificar tokens válidos, errores léxicos y sintácticos, generar reportes detallados y mostrar las operaciones matemáticas. La aplicación opera a través de una interfaz web interactiva.

Requerimientos del Sistema

- Node.js: Versión 18.0 o superior.
- Navegador Web: Google Chrome, Mozilla Firefox o equivalente.
- RAM: 2 GB mínimo.
- Espacio en Disco Duro: 100 MB.
- Sistema Operativo Compatible: Linux Fedora 40, Windows 10+, macOS

Cómo Iniciar el Programa

1. Asegúrate de que Node.js esté instalado en tu sistema. Puedes verificarlo ejecutando:
`node -v`
2. Navega al directorio del proyecto donde clonaste el repositorio/descargarlo y abrirlo en VSCode o en cualquier IDE de preferencia
3. Ejecuta el siguiente comando para instalar paquetes:
`npm install`
4. Ejecuta el siguiente comando para iniciar el servidor:
`npm run dev`

Por la dependencia que tenemos en package.json que es “vite”

5. Abre tu navegador web y accede a <http://localhost:3000> o bien puedes ingresar lo que te de la terminal

```

Re-optimizing dependencies because vite config has changed

VITE v6.0.2 ready in 138 ms

➔ Local:   http://localhost:5173/
➔ Network: use --host to expose
➔ press h + enter to show help
11:40:50 PM [vite] (client) hmr update /src/pages/Operation
Page.jsx
11:41:36 PM [vite] (client) hmr update /src/pages/Report.js
11:41:37 PM [vite] (client) hmr update /src/pages/Report.js
(x2)
11:41:53 PM [vite] (client) hmr update /src/pages/Report.js

```

Desde la interfaz web, podrás:

- Cargar archivos fuente para análisis.
- Visualizar tokens reconocidos y errores.
- Generar reportes detallados.

Al iniciar el proyecto te saldra lo siguiente:

En el boton “**Home**” es el analizador Léxico

Boton “Analizador Sintáctico”

En este botón tendremos el Scanner parser

[Home](#) [Analizador Sintactico](#) [Grafica Arbol](#) [Operaciones](#) [Tabla](#) [Archivo](#)

Analizador Sintáctico

Escribe o pega el código aquí...

Analizar Sintáctico

Árbol Sintáctico

No se ha generado el árbol.

Errores Sintácticos

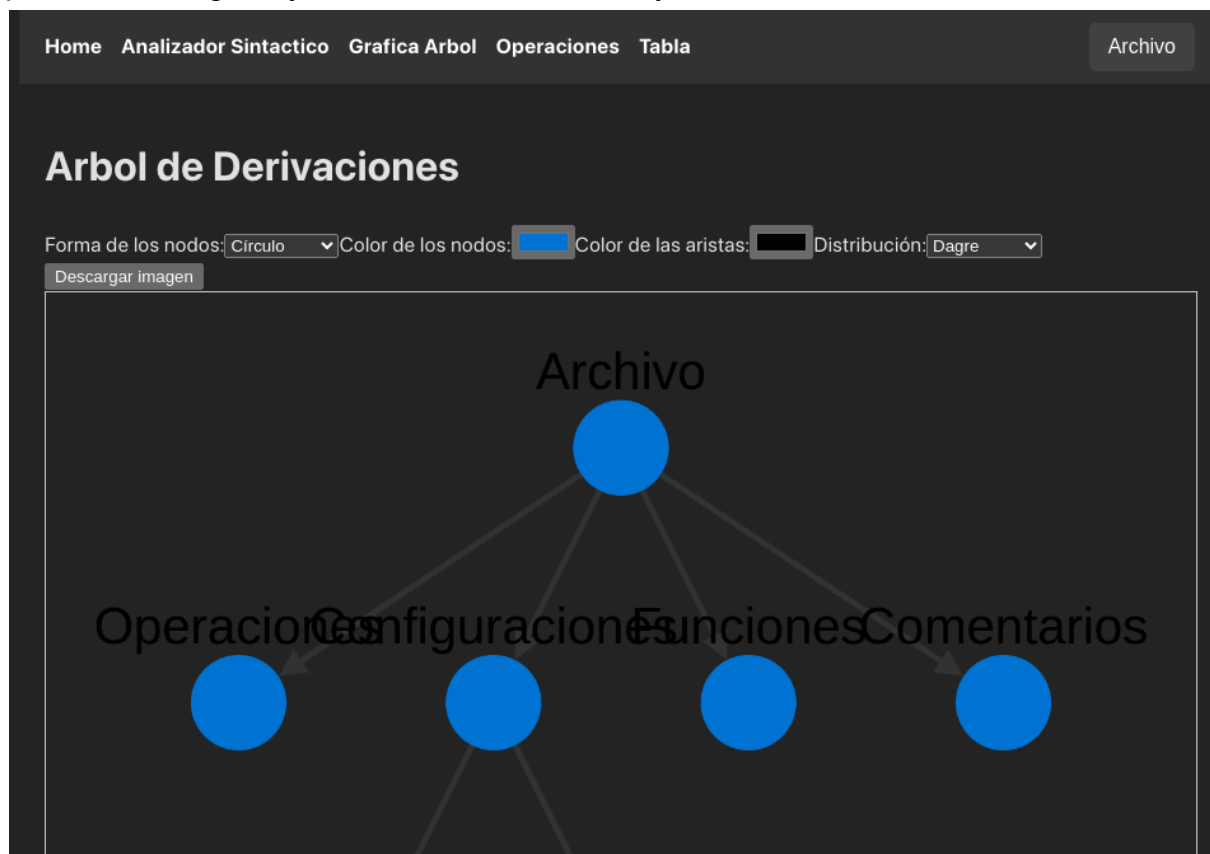
Error

No se encontraron errores.

Boton “Gráfica Arbol”

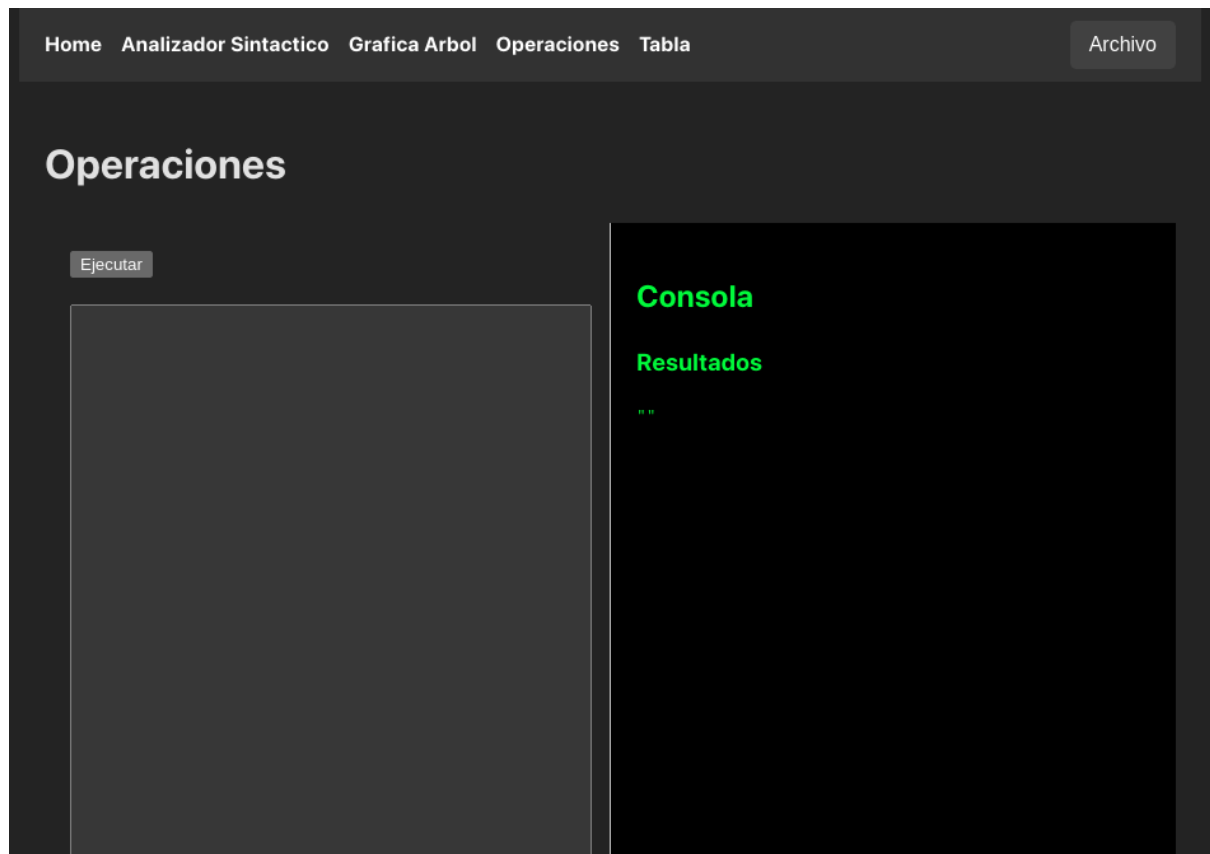
En este boton se te generara automaticamente un arbol de derivaciones donde

podrás descargarlo y cambiarle formas, color y distribución



Boton “Operaciones”

En este botón podrás ejecutar las operaciones que te han proporcionado en un archivo .nlex o .NLEX



Botón “Tabla”

En este botón se genera el HTML con la tabla de Tokens y Errores Encontrados, así mismo tendrás el código HTML para tener una idea de cómo realizarlo y adentrarte

al mundo de la programación (si te interesa)

[Home](#) [Analizador Sintactico](#) [Grafica Arbol](#) [Operaciones](#) [Tabla](#) [Archivo](#)

Reporte de Tokens y Errores

Código HTML

```
<h3>Tokens</h3>
<table>
  <thead>
    <tr>
      <th>Tipo</th>
      <th>Lexema</th>
      <th>Línea</th>
      <th>Columna</th>
    </tr>
  </thead>
  <tbody>

</tbody>
</table>

<h3>Errores</h3>
<table>
  <thead>
    <tr>
      <th>Lexema</th>
      <th>Línea</th>
      <th>Columna</th>
    </tr>
  </thead>
  <tbody>

</tbody>
</table>
```

Tokens

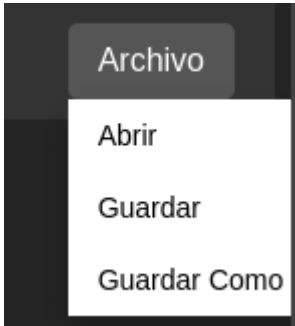
Tipo	Lexema	Línea	Columna
------	--------	-------	---------

Errores

Lexema	Línea	Columna
--------	-------	---------

Botón “Archivo”

En este botón podrás “Abrir” un archivo .NLEX o .nlex y “Guardar” lo que tengas en el Scanner Lexico o Sintactico



Estructura de Entrada

```

Operaciones=[
  {"operacion": "resta", "nombre": "operacion1", "valor1": 6.5, "valor2": 3.5},
  {"operacion": "multiplicacion", "nombre": "operacion2", "valor1": 2.3, "valor2": [{"operacion": "seno", "valor1": 90}]},
  {"operacion": "suma", "nombre": "operacion3", "valor1": 2.3, "valor2": [{"operacion": "multiplicacion", "valor1": 10, "valor2": [{"operacion": "raiz", "valor1": "valor2": 2}]]}]
]

ConfiguracionesLex = [
  fondo: "#000000",
  fuente: "#FFFFFF",
  forma: "circle",
  tipoFuente: "Times-Roman"
]

ConfiguracionesParser = [
  fondo: "#f3ff00",
  fuente: "#000000",
  forma: "box",
  tipoFuente: "Arial"
]

// Comentario simple
/* Comentario de
multiple linea */

imprimir("Lenguajes Formales y de Programacion")
conteo()
promedio("suma")
max("multiplicacion")
min("raiz")

//Generar reportes tipo null
generarReporte("tokens")
generarReporte("errores", "201700761")
generarReporte("arbol", "Nueva derivacion")

```

- **Operaciones:**
 - Lista de operaciones matemáticas.
 - Cada operación debe tener Operación, Valor1, y Valor2.
 - Operaciones matemáticas anidadas
- **Configuraciones:**
 - Opciones para personalizar los diagramas generados.

Soporte

- En caso de problemas, consulta el manual técnico o contacta al desarrollador