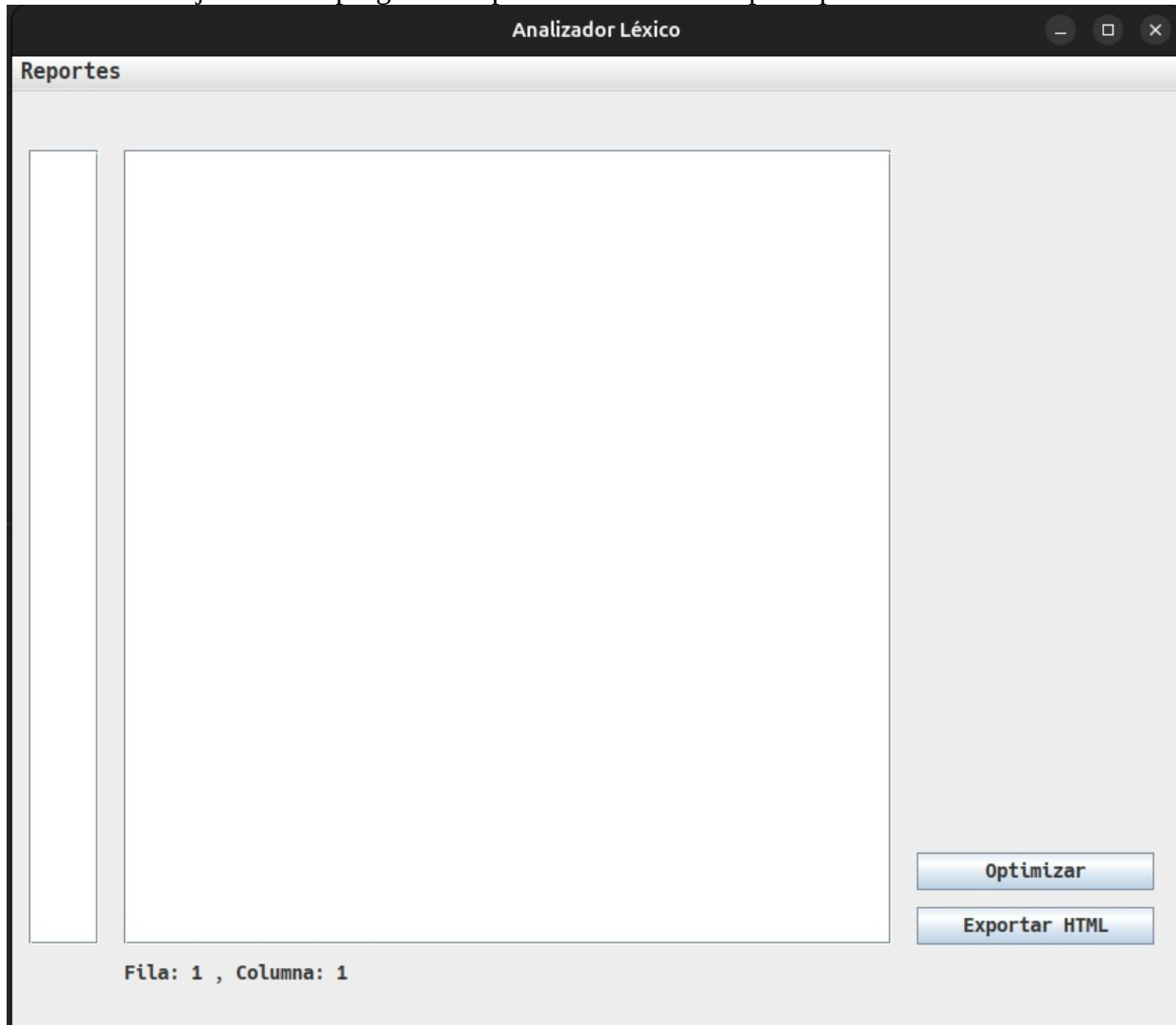


# MANUAL DE USUARIO.

**Fredy Jose Gabriel Herrera Funes. Carnet: 202130478.**

1. Ejecutar con el comando en la terminal o consola: `java -jar AnalizadorLexico—Lenguajes-1.0-SNAPSHOT.jar` para utilizar el programa.
2. Una vez ejecutado el programa se puede ver la ventana principal la cual es:



3. Ingresar el código como el siguiente:

>>[html]

```
<contenedor class="container">
  <titulo1>To-Do List</titulo1>
  <contenedor class="input-section">
    <entrada type="text" id="taskInput" placeholder="Add a new task...">
    <boton id="addButton">Add Task</boton>
  </contenedor>
  <listaordenada id="taskList"></listaordenada>
</contenedor>
```

>>[css]

```
body {
  font-family: 'Arial', sans-serif;
  background-color: #f0f0f0;
```

```
display: flex;
justify-content: center;
align-items: center;
height: 100 vh;
margin: 0;
}

.container {
background-color: #ffffff;
padding: 20 px;
border-radius: 10 px;
box-shadow: 0 0 10 px rgba(0, 0, 0, 0.1);
width: 300 px;
}

h1 {
text-align: center;
}

.input-section {
display: flex;
justify-content: space-between;
margin-bottom: 20 px;
}

#taskInput {
width: 70%;
padding: 10 px;
border: 1 px solid #ccc;
border-radius: 5 px;
}

#addButton {
padding: 10 px;
border: none;
background-color: #28a745;
color: #fff;
border-radius: 5 px;
cursor: pointer;
}

#addButton:hover {
background-color: #218838;
}

#taskList {
list-style-type: none;
padding: 0;
}

#taskList li {
background-color: #f8f9fa;
```

```
padding: 10 px;
border: 1 px solid #ddd;
border-radius: 5 px;
display: flex;
justify-content: space-between;
margin-bottom: 10 px;
}
```

```
#taskList li .delete-button {
  border: none;
  background: transparent;
  cursor: pointer;
  color: #dc3545;
  font-size: 16 px;
}
```

```
#taskList li .delete-button:hover {
  color: #c82333;
}
```

```
>>[js]
```

```
// script.js
```

```
// Referencias a los elementos de la interfaz
```

```
const taskInput = document.getElementById('taskInput');
const addButton = document.getElementById('addButton');
const taskList = document.getElementById('taskList');
```

```
// Evento para agregar una nueva tarea
```

```
addButton.addEventListener('click', addTask);
```

```
// Función para agregar la tarea a la lista
```

```
function addTask() {
  const taskValue = taskInput.value.trim();
  if (taskValue === "") return;
```

```
  const listItem = document.createElement('li');
  listItem.textContent = taskValue;
```

```
  const deleteButton = document.createElement('button');
  deleteButton.textContent = 'X';
  deleteButton.classList.add('delete-button');
  deleteButton.addEventListener('click', () => listItem.remove());
```

```
  listItem.appendChild(deleteButton);
  taskList.appendChild(listItem);
  taskInput.value = "";
```

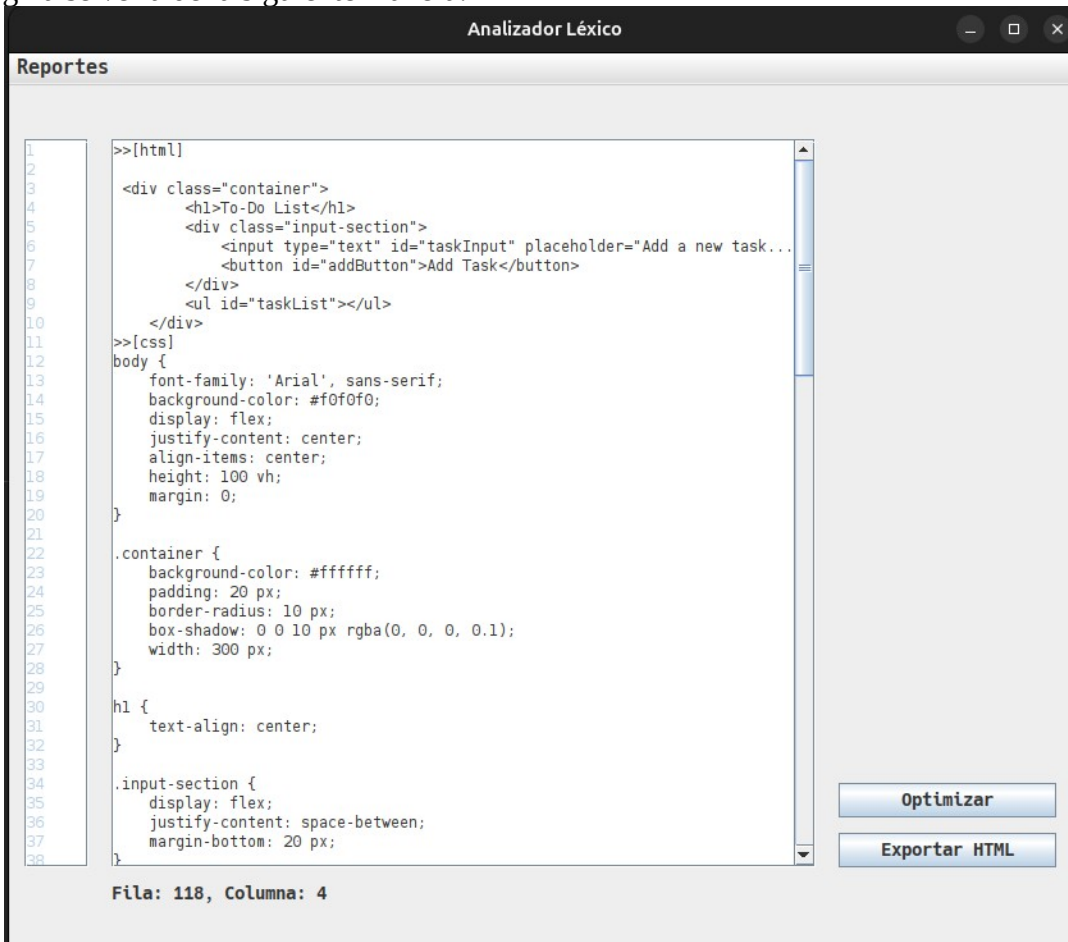
```
}
```

```
// Permitir agregar tareas con la tecla Enter
```

```
taskInput.addEventListener('keyup', (e) => {
  if (e.key === 'Enter') addTask();
```

});

4. La pagina se vería de la siguiente manera:



The screenshot shows the 'Analizador Léxico' window with the 'Reportes' tab selected. The code editor displays the following HTML and CSS code with line numbers 1 through 38 on the left. The code includes comments and empty lines. At the bottom, it shows 'Fila: 118, Columna: 4'. On the right side, there are two buttons: 'Optimizar' and 'Exportar HTML'.

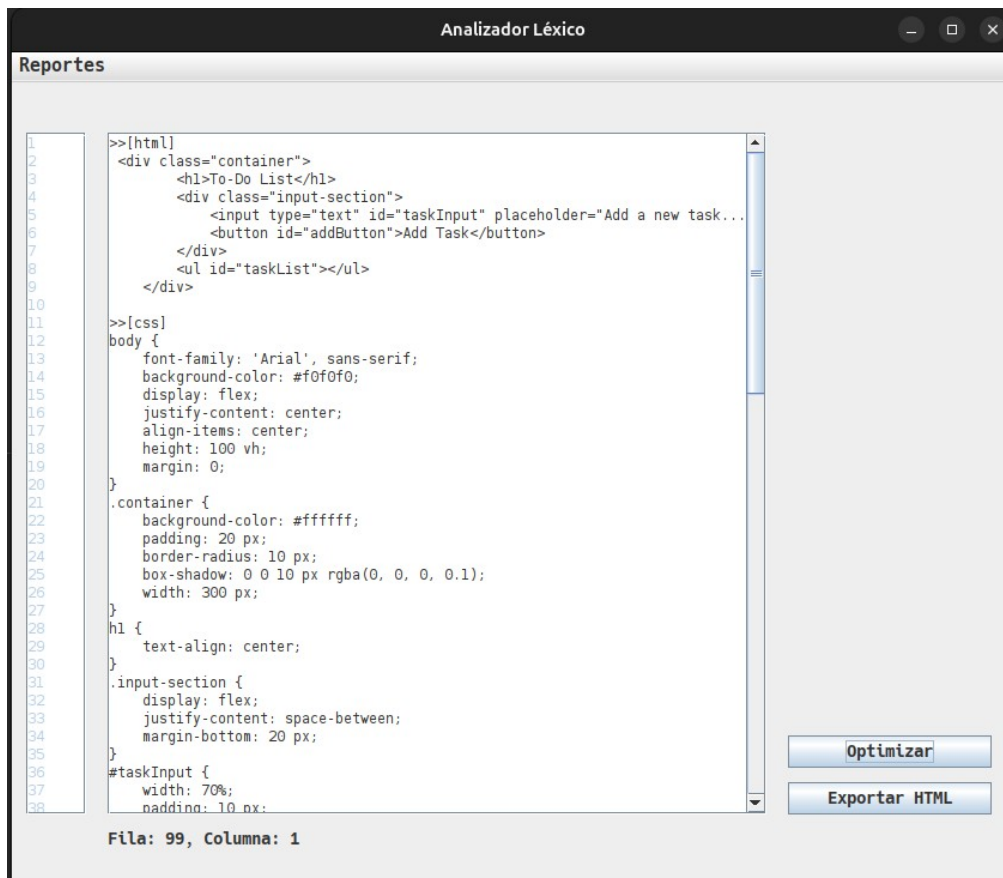
```
1 >>[html]
2
3 <div class="container">
4 <h1>To-Do List</h1>
5 <div class="input-section">
6 <input type="text" id="taskInput" placeholder="Add a new task..."
7 <button id="addButton">Add Task</button>
8 </div>
9 <ul id="taskList"></ul>
10 </div>
11 >>[css]
12 body {
13   font-family: 'Arial', sans-serif;
14   background-color: #f0f0f0;
15   display: flex;
16   justify-content: center;
17   align-items: center;
18   height: 100 vh;
19   margin: 0;
20 }
21
22 .container {
23   background-color: #ffffff;
24   padding: 20 px;
25   border-radius: 10 px;
26   box-shadow: 0 0 10 px rgba(0, 0, 0, 0.1);
27   width: 300 px;
28 }
29
30 h1 {
31   text-align: center;
32 }
33
34 .input-section {
35   display: flex;
36   justify-content: space-between;
37   margin-bottom: 20 px;
38 }
```

Fila: 118, Columna: 4

Optimizar

Exportar HTML

5. Al dar click en el boton “Optimizar” se eliminarian las lineas vacias y las lineas que contengan comentario:



The screenshot shows the 'Analizador Léxico' window with the 'Reportes' tab selected. The code editor displays the optimized code with line numbers 1 through 38 on the left. The code is now compact, with all comments and empty lines removed. At the bottom, it shows 'Fila: 99, Columna: 1'. On the right side, there are two buttons: 'Optimizar' and 'Exportar HTML'.

```
1 >>[html]
2 <div class="container">
3   <h1>To-Do List</h1>
4   <div class="input-section">
5     <input type="text" id="taskInput" placeholder="Add a new task..."
6     <button id="addButton">Add Task</button>
7   </div>
8   <ul id="taskList"></ul>
9 </div>
10
11 >>[css]
12 body {
13   font-family: 'Arial', sans-serif;
14   background-color: #f0f0f0;
15   display: flex;
16   justify-content: center;
17   align-items: center;
18   height: 100 vh;
19   margin: 0;
20 }
21
22 .container {
23   background-color: #ffffff;
24   padding: 20 px;
25   border-radius: 10 px;
26   box-shadow: 0 0 10 px rgba(0, 0, 0, 0.1);
27   width: 300 px;
28 }
29
30 h1 {
31   text-align: center;
32 }
33
34 .input-section {
35   display: flex;
36   justify-content: space-between;
37   margin-bottom: 20 px;
38 }
39
40 #taskInput {
41   width: 70%;
42   padding: 10 px;
43 }
```

Fila: 99, Columna: 1

Optimizar

Exportar HTML

6. Para ver los reportes y exportarlos ir al apartado de reportes en la parte superior, luego dar click en el boton “Reporte optimizacion”, lo cual desplegara la siguiente ventana, y para exportar a html, click en “Exportar HTML”.

Texto	Lenguaje	Expresión Regular	Tipo	Fila	Columna
// script.js	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// script.js	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Referencias a los el...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Referencias a los el...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Evento para agrega...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Evento para agrega...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Función para agreg...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Función para agreg...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Permitir agregar ta...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0
// Permitir agregar ta...	JavaScript	//[a-zA-Z][0-9][.]	Comentario	0	0

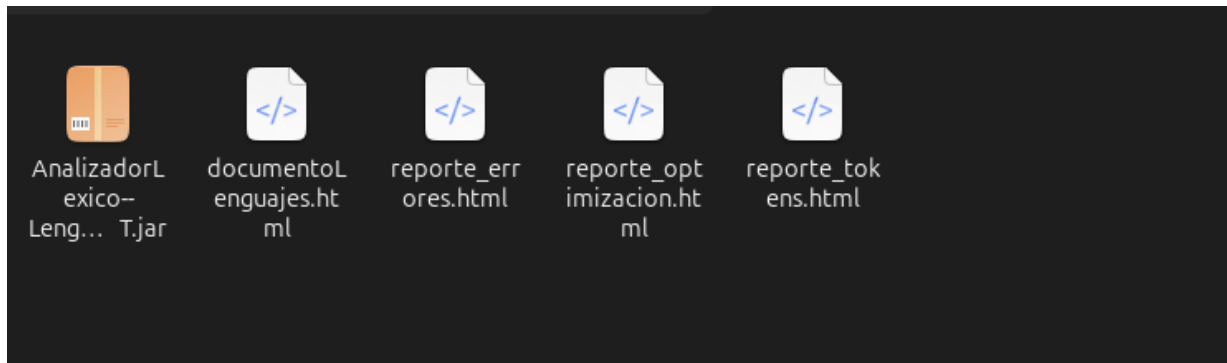
Exportar HTML

7. al dar click en el boton “Exportar HTML” de la pagina principal, se exportara el HTML funcional del codigo ingresado si todos los tokens son validos, si no son validos se desplegara la ventana de reportes de error:

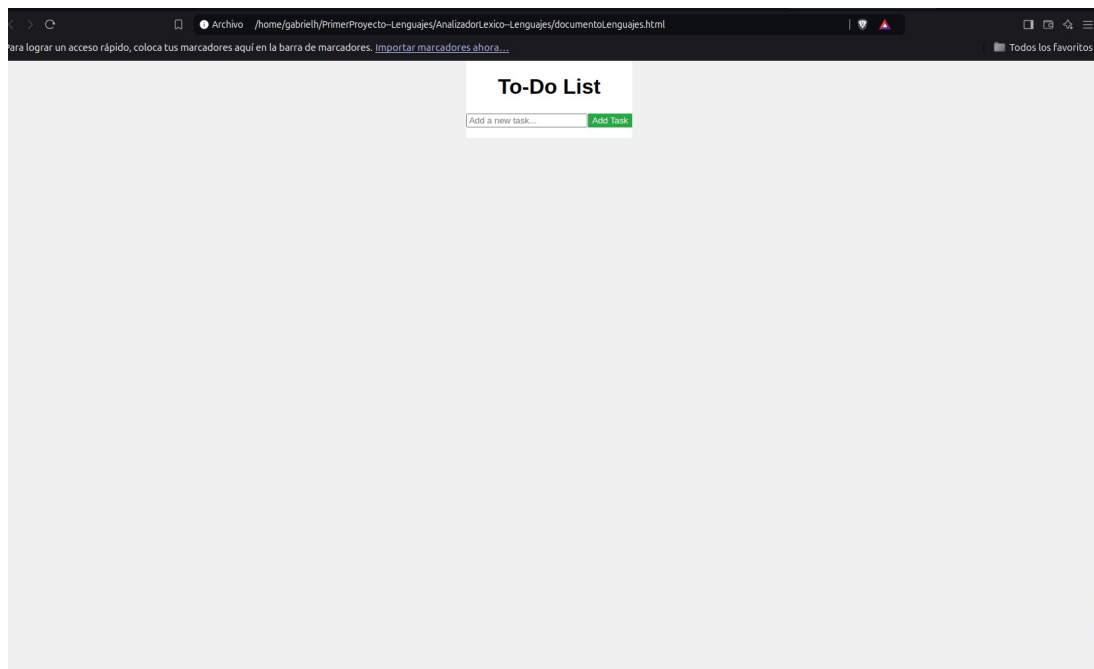
Token	Lenguaje Encontrado	Lenguaje Sugerido	Fila	Columna
<button id="addButton">	Html	Html	0	0
</button>	Html	Html	0	0
</div>	Html	Html	0	0
<ul id="taskList">	Html	Html	0	0
</ul>	Html	Html	0	0
</div>	Html	Html	0	0

Exportar HTML

8. Se podran ver los html exportados en la carpeta donde se encuentra el archivo JAR.



9. Al doble click en los archivos HTML, se abriran las paginas en el navegador:



10. Los reportes se exportan de esta manera:

Archivo

/home/gabrielh/PrimerProyecto-Lenguajes/AnalizadorLexico-Lenguajes/reporte\_errores.html

Para lograr un acceso rápido, coloca tus marcadores aquí en la barra de marcadores. [Importar marcadores ahora...](#)

Todos los favoritos

Reporte de Errores

Token	Lenguaje Encontrado	Lenguaje Sugerido	Fila	Columna
<titulo1/>	Html	Html	0	0
<parrafo/>	Html	Html	0	0
<contenedor/>	Html	Html	0	0
<contenedor/>	Html	Html	0	0