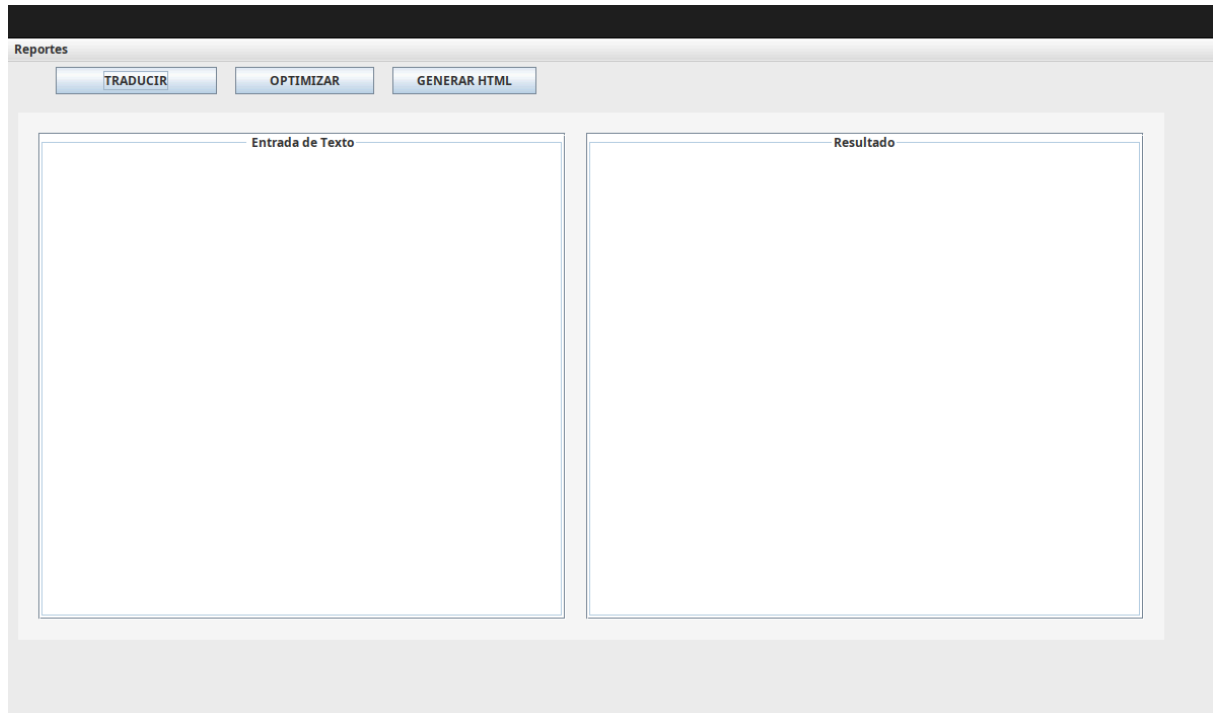
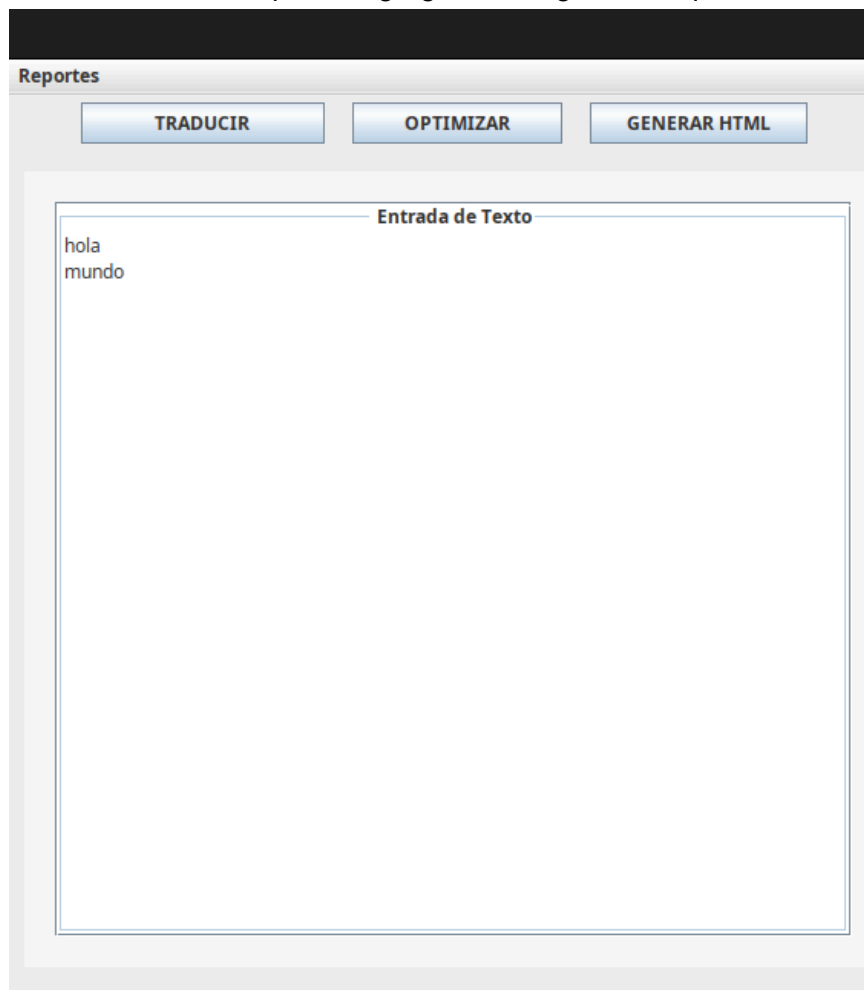


Manual de Usuario

Al abrir la aplicación se muestra la ventana.



En la sección “Entrada de Texto” puede agregar el código con el que se desee trabajar



Para iniciar el análisis puede utilizar los token de cambio de estado:

- >>[html]
- >>[css]
- >>[js]

Después de indicar el formato del código del código se pueden ingresar los valores deseados dependiendo del tipo de lenguaje que se haya seleccionado.

The screenshot shows a web application window titled 'Reportes'. It has three buttons at the top: 'TRADUCIR', 'OPTIMIZAR', and 'GENERAR HTML'. Below these buttons is a large text area labeled 'Entrada de Texto'. The text area contains the following code:

```
>>[js]
function showSection(sectionId) {
  const sections = document.querySelectorAll('#main-content > div');
  sections.forEach(section => {

    section.style.display = 'none';

    section.style.background = 'red';
  });

  // Aplicar display a la section
  document.getElementById(sectionId).style.display = 'block';
}

>>[html]
<contenedor id="showcase">
  <contenedor class="container">
    <titulo1>Bienvenido a Mi Página<titulo1/>
    <parrafo>Explora nuestro contenido y conoce más sobre nosotros.<parrafo/>

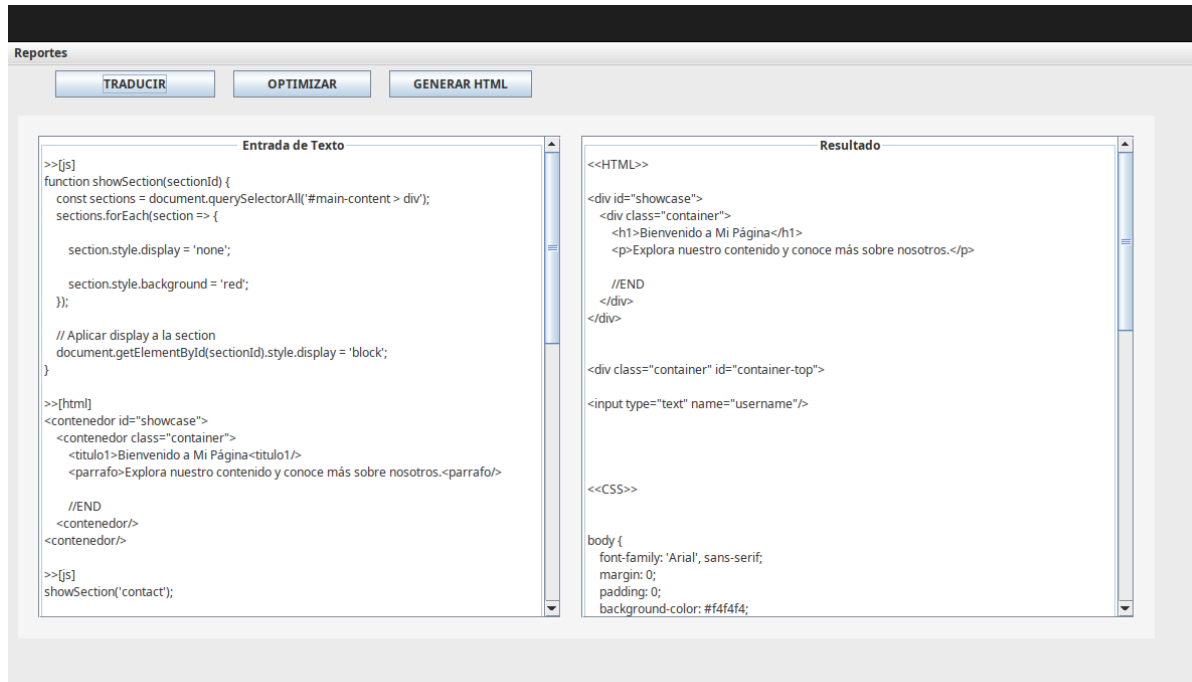
    //END
  <contenedor/>
</contenedor/>

>>[js]
showSection('contact');
```

Cuando se haya terminado de ingresar el código para traducirlo se debe presionar el botón "TRADUCIR".

This is a close-up view of the 'Reportes' window. It shows the 'Reportes' title bar and a single button labeled 'TRADUCIR'.

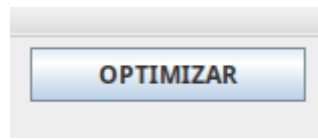
Al presionarlo se mostrará el resultado de la traducción del código del lado derecho de la aplicación en el área “Resultado”.



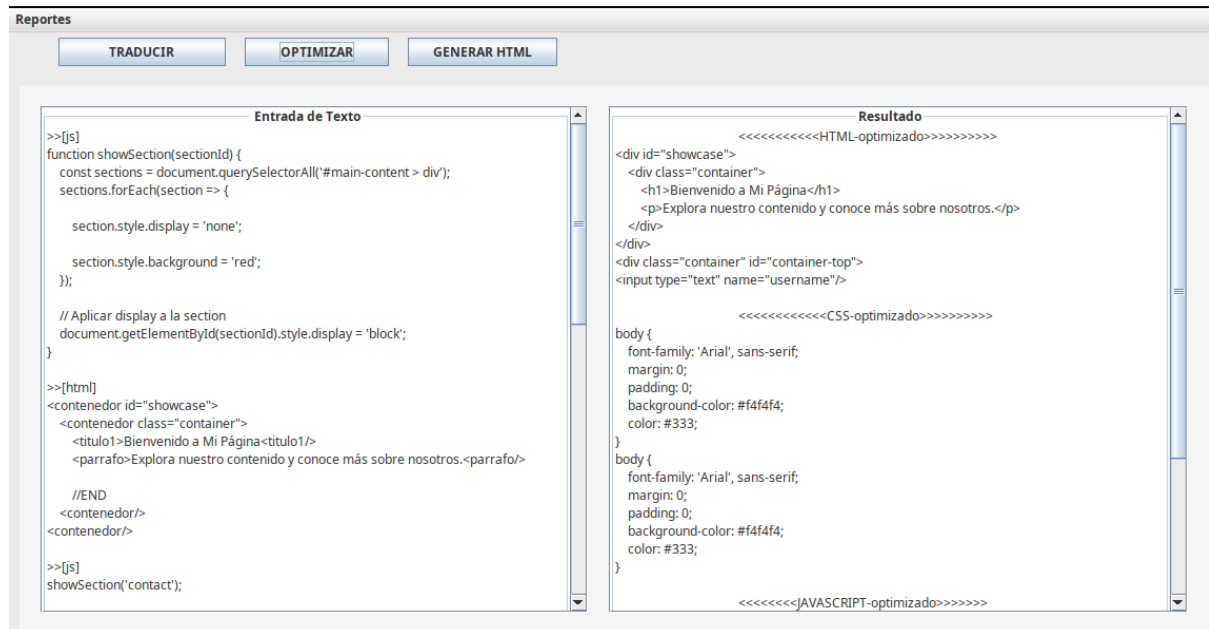
En el área de resultado se mostrará la traducción del código ingresado en el siguiente orden, primero el código html, después el código css y por último el código javascript



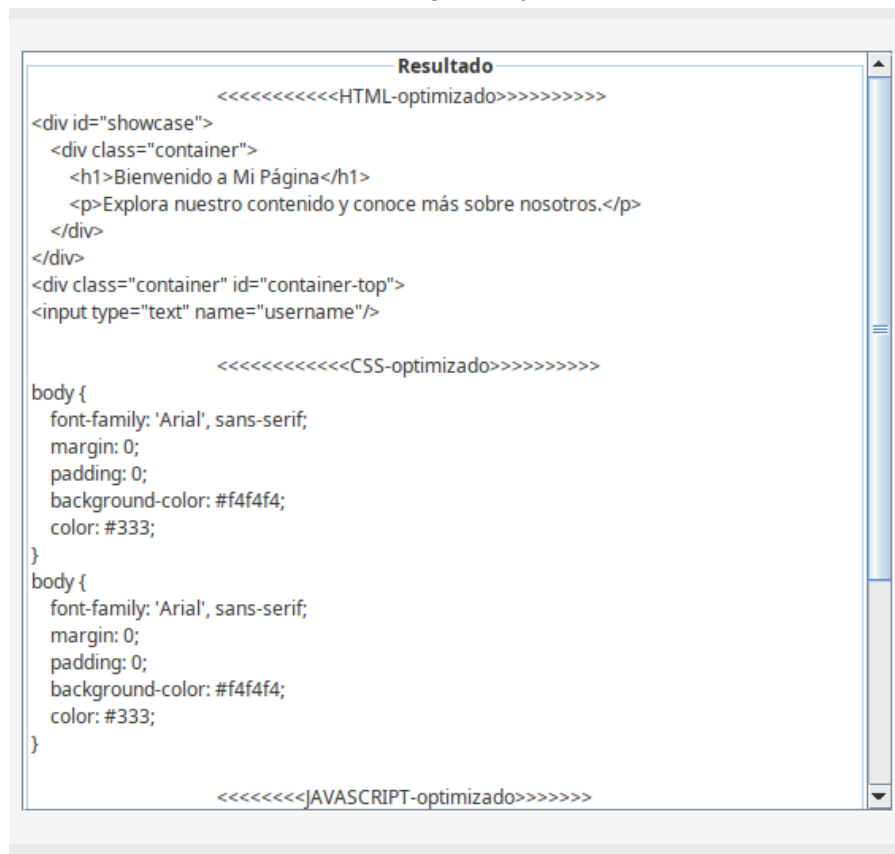
Después de haber traducido el código puede optimizarlo para eliminar comentarios y saltos de línea innecesarios presionando el botón “OPTIMIZAR”



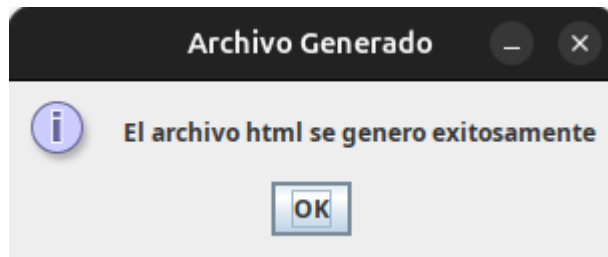
El resultado de la optimización se mostrará en el área de “Resultado”.



El resultado se mostrará en el siguiente orden primero la optimización del código html, después la optimización del código css y por último la optimización del código javascript.



Al haber seleccionado la carpeta aparecerá una notificación informando que el proceso fue exitoso.



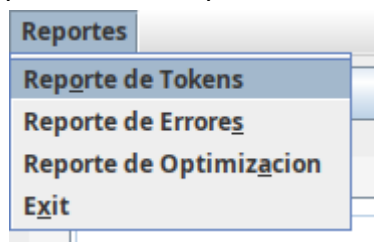
Reportes

La aplicación también genera un archivo html para los siguientes reportes,

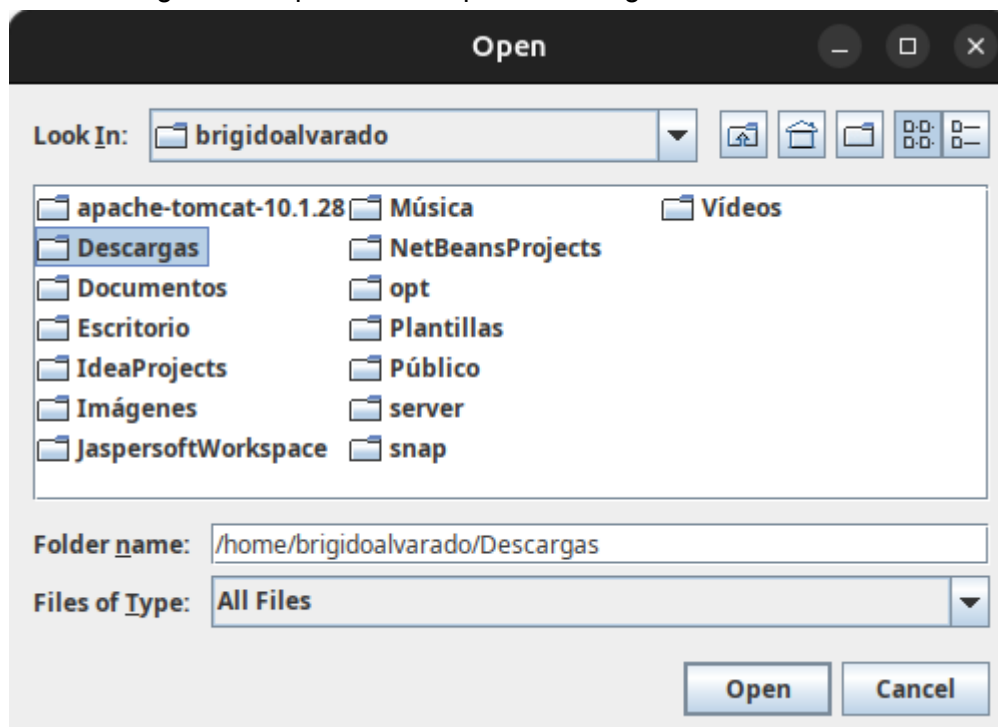
- **Reporte de Tokens:** Una tabla con la información de los tokens que fueron leídos y aceptados por la aplicación.
- **Reporte de Errores:** Una tabla con la información de las palabras que no fueron reconocidas como tokens del lenguaje.
- **Reporte de Optimización:** Una tabla con la información de los tokens que fueron borrados al optimizar el código.

Reporte de Tokens

Para generar un reporte de tokens se debe presionar el botón "Reporte de Tokens" de la pestaña de "Reportes" de la aplicación.




se debe elegir una carpeta destino para el html generado.



Después se notificará si el proceso fue exitoso y la ubicación y nombre de la aplicación.

Reporte generado exitosamente



Se guardo el archivo de reportes en

/home/brigidoalvarado/Descargas/Reporte de Tokens2024-10-09T22:23:17.062393233.html

OK

Ejemplo de archivo html generado.

Reporte de Tokens

Reporte de Errores

Reporte de Errores

Reporte de Tokens

Archivo

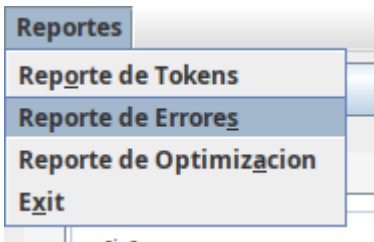
/home/brigidoalvarado/Descargas/Reporte%20de%20Tokens2024-10-09T22:23:17.062393233.html

Reporte de Tokens

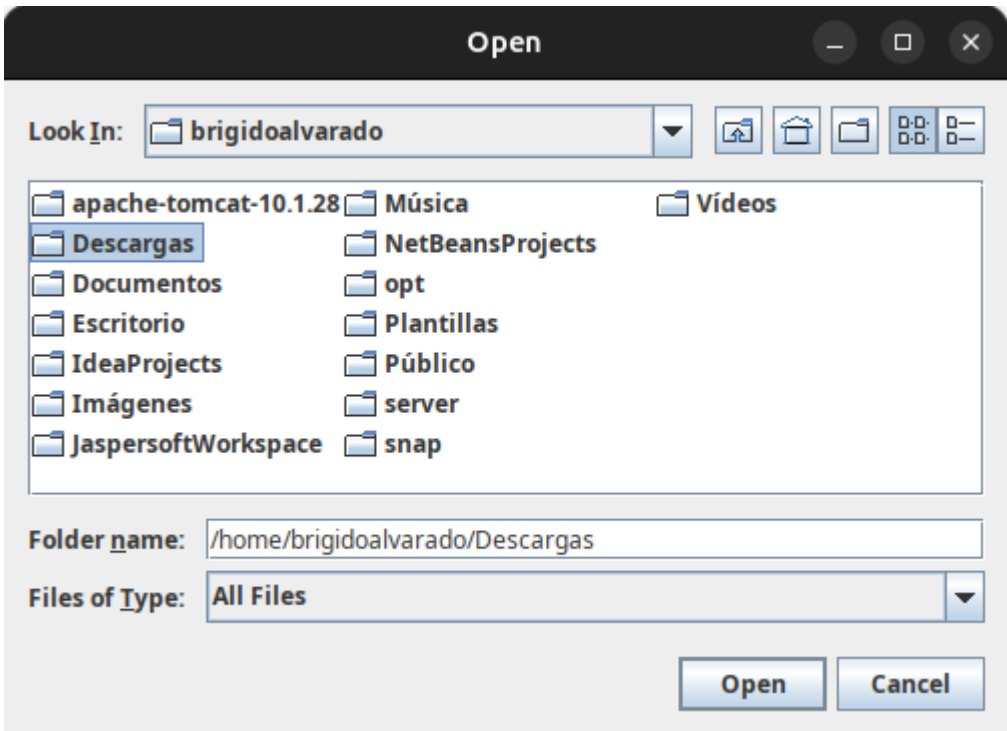
Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
>>[js]	>>[js]		Estado	1	7
function	function	Javascript	Palabra reservada	2	8
showSection	[a-zA-Z]([a-zA-Z] [0-9] [_])*	Javascript	Identificador	2	20
((Javascript	Parentesis_apertura	2	21
sectionId	[a-zA-Z]([a-zA-Z] [0-9] [_])*	Javascript	Identificador	2	30
))	Javascript	Parentesis_cierre	2	31
{	{	Javascript	Llaves_apertura	2	33
const	const	Javascript	Palabra reservada	3	9
sections	[a-zA-Z]([a-zA-Z] [0-9] [_])*	Javascript	Identificador	3	18
=	=	Javascript	Asignacion	3	20
document	document	Javascript	Palabra reservada	3	29
.	.	Javascript	Punto	3	30
querySelectorAll	[a-zA-Z]([a-zA-Z] [0-9] [_])*	Javascript	Identificador	3	46
((Javascript	Parentesis_apertura	3	47
'#main-content > div'	"([^\"]*(\\\"\\\"))*"	Javascript	cadena	3	68

Reporte de Errores

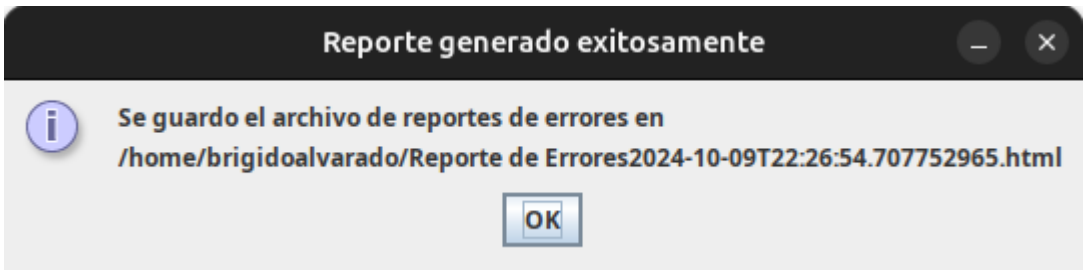
Para generar un reporte de tokens se debe presionar el botón “Reporte de Errores” de la pestaña de “Reportes” de la aplicación.



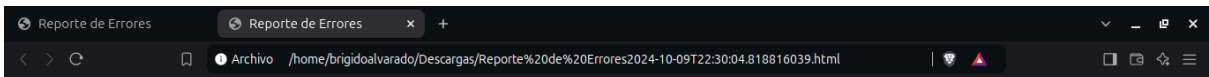
Se debe elegir una carpeta destino para el html generado.



Después se notificará si el proceso fue exitoso y la ubicación y nombre de la aplicación.



Ejemplo de archivo html generado.

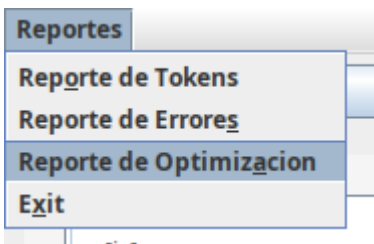


Reporte de Errores

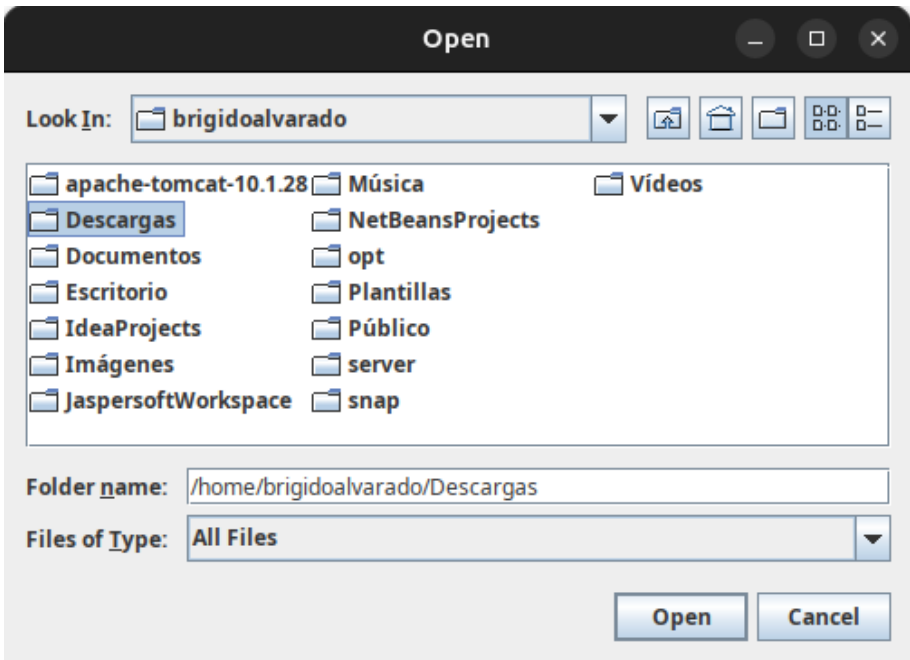
Token	Lenguaje donde se Encontro	Lenguaje sugerido	Fila	Columna
let	HTML	Javascript	54	1
const	HTML	Javascript	6	1
flex	HTML	CSS	45	1

Reporte de Optimización

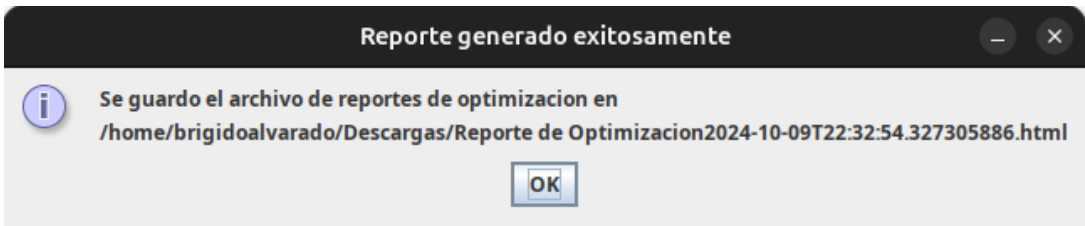
Para generar un reporte de tokens se debe presionar el botón “Reporte de Errores” de la pestaña de “Reportes” de la aplicación.



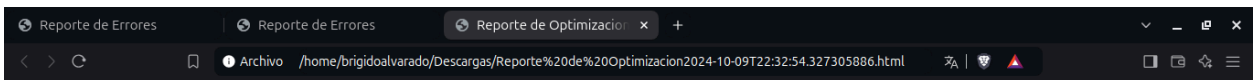
Se debe elegir una carpeta destino para el html generado.



Después se notificará si el proceso fue exitoso y la ubicación y nombre de la aplicación.



Ejemplo de archivo html generado.



Reporte de Optimizacion

Token	Expresion Regular	Lenguaje	Tipo	Fila	Columna
// Aplicar display a la section	// [a-zA-Z][0-9][.]	Javascript	Comentario	11	35
//END	// [a-zA-Z][0-9][.]	HTML	Comentario	21	13
showSection	[a-zA-Z]([a-zA-Z] [0-9] [_])*	Javascript	Identificador	52	11
((Javascript	Parentesis_apertura	52	12
'contact2'	"([^\"]*(\"\\\"))*"	Javascript	cadena	52	22
))	Javascript	Parentesis_cierre	52	23
;	;	Javascript	Punto_y_coma	52	24
//Un Comentario	// [a-zA-Z][0-9][.]	Javascript	Comentario	52	40