

```
<!--Carné 202131782-->
```

```
Manual de  
usuario {
```

```
<Por="Jorge Bravo"/>
```

```
}
```



# Aclaraciones{

Es necesario tener java instalado en el equipo donde se ejecutará el servidor de captchas.

Es necesario tener Node y Angular en la computadora para ejecutar el cliente.

Es necesario tener configurado un servidor tomcat previo a la ejecucion del servidor.

Los programas creados, tanto el servidor como el cliente son multiplataforma gracias a Java y Node, pueden ejecutarse en Windows, Mac y Linux.

Para manejar el servidor web de captchas generados es necesario tener python instalado. Para iniciarlo ejecutar

**python3 -m http.server 8000**

en la carpeta donde se guardan los captchas.

}

Aplicacion web

{

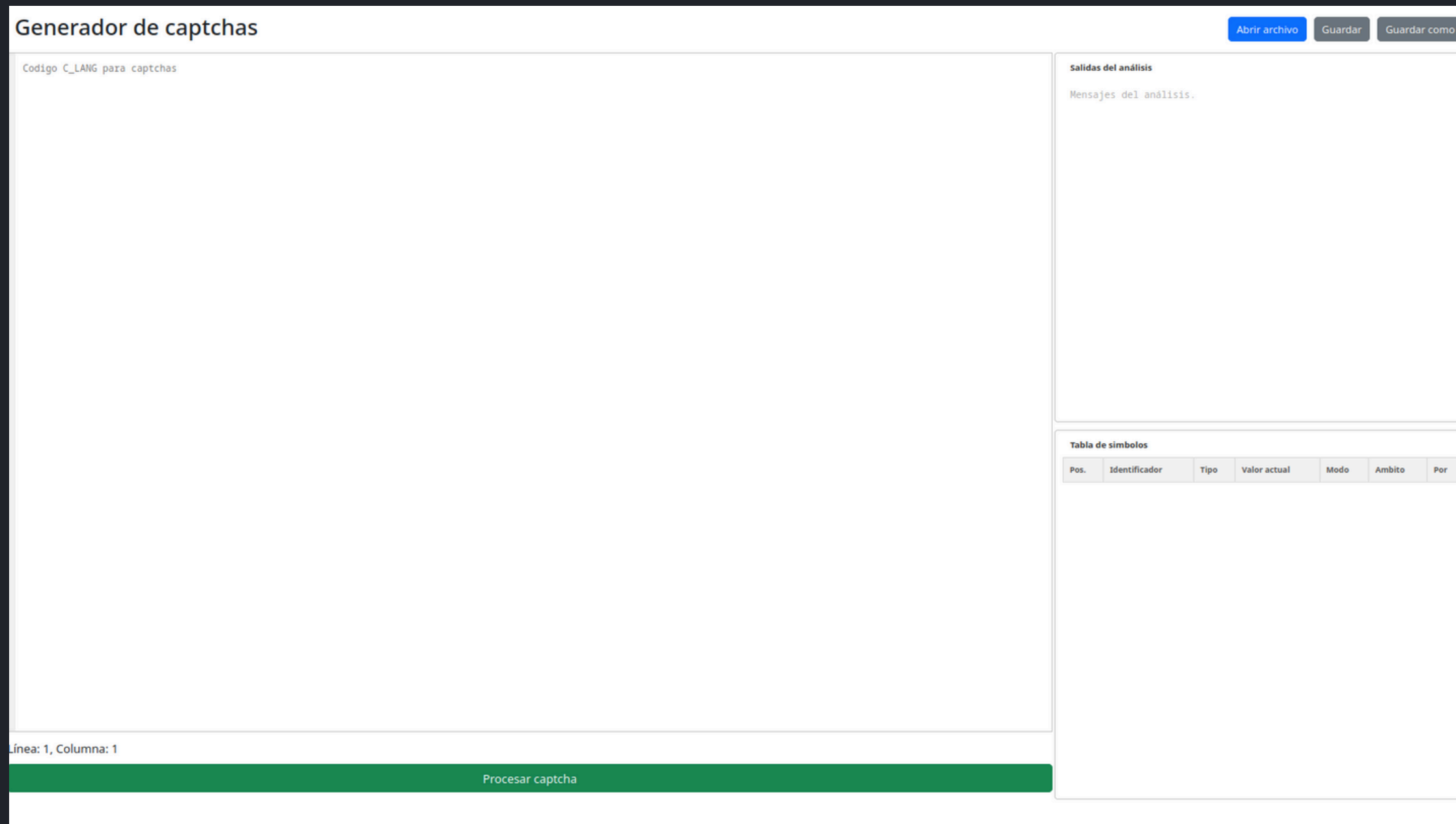
}



# Generador de Captchas {

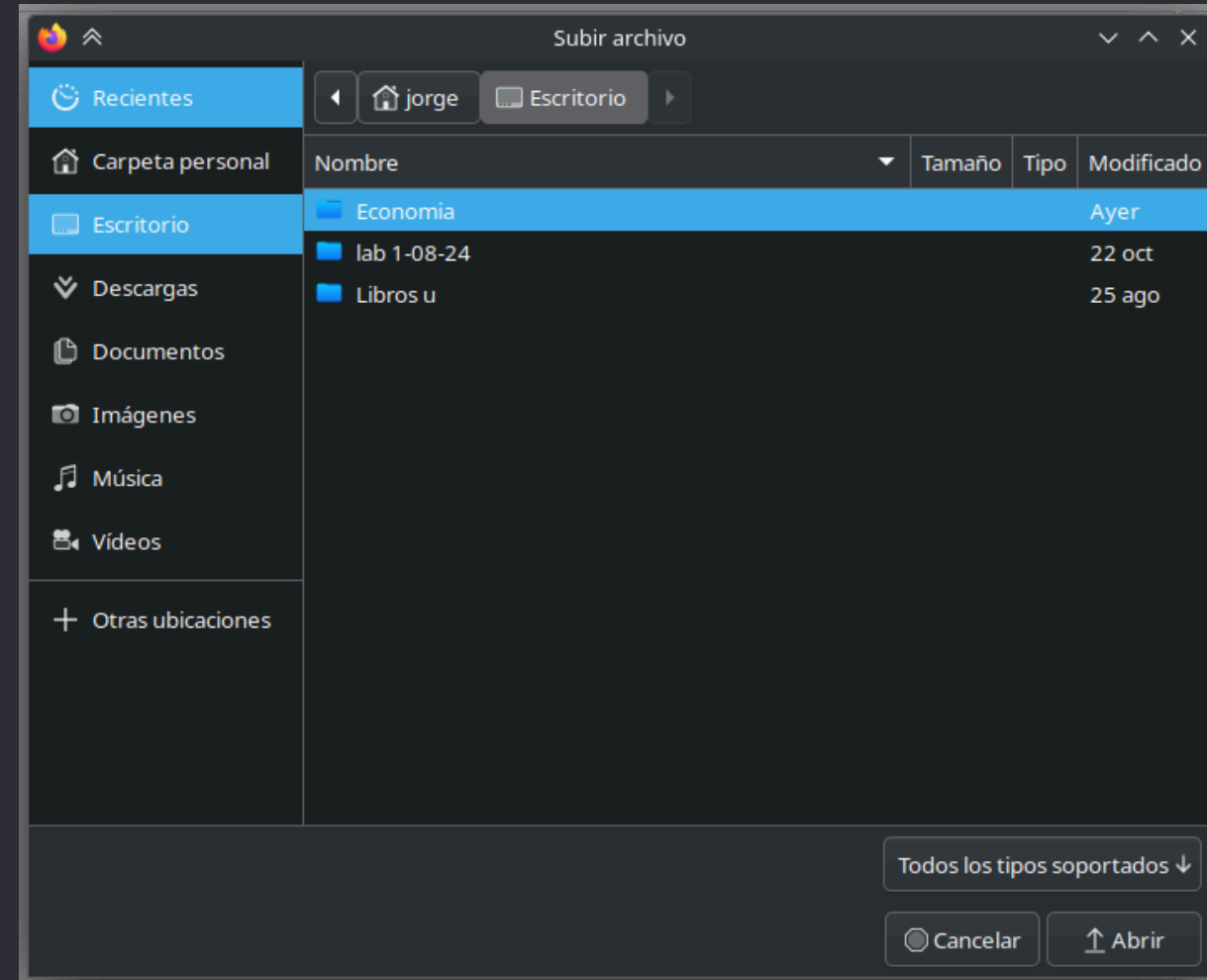
- La pagina principal de la aplicación es donde podremos realizar todo lo relacionado con la escritura de código y generación de captchas.
- Podremos escribir el código directamente en el editor, cargar un archivo previamente creado y guardar el archivo que estamos editando.
- También podremos ver información relevante del captcha generado.

}



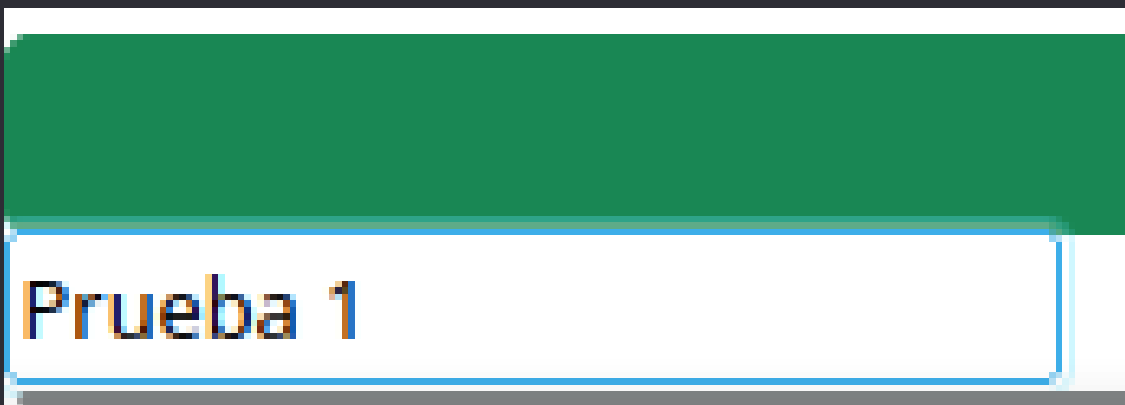
# ABRIR ARCHIVO

Nos permitirá abrir un selector de archivos nativo del sistema con el cual podremos seleccionar un archivo para cargar en el editor.



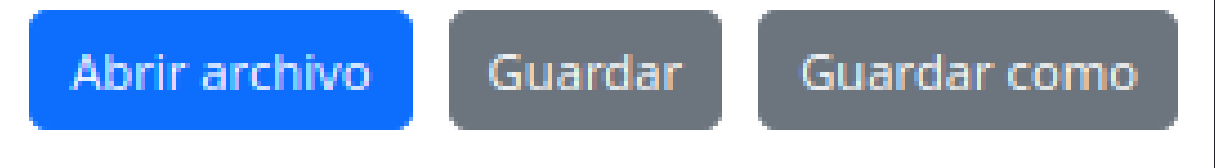
# GUARDAR

Este botón tomará el contenido del editor y lo guardará dentro de la computadora. Para ello se mostrará un input en la parte inferior de la aplicación donde colocar el nombre del archivo.



## Manejo de archivos {

- En la parte superior derecha tendremos acceso a diferentes acciones a realizar con archivos



}

# Editor de código

## Generador de captchas

```
1 <c_CC [id = "ejemplo"] [name = "captcha1"] [type="tipos"]>
2
3 <C_HEAD>
4   <C_TITLE>
5     Hola a todos
6   </C_TITLE>
7 </C_HEAD>
8 !! Esto es un comentario de una linea
9
10 <C_BODY [background="olive"]>
11   !! Esto es otro comentario de una linea
12
13   <C_DIV>
14     <C_SCRIPTING>
15
16       FUNCTION_calc() [
17         !! Estas instrucciones no se ejecutan hasta llamar a FUNCTION_calc()
18         integer @global contador_fallas = 5;
19         string result_caja_texto = getElementById("entrada_1");
20         string @global result = "10 ";
21         result = "30 ";
22
23         no_declarada = 40;
24
25         string @global p_declaracion;
26         string p2_declaracion;
27
28         string mensaje_fallo = "El captcha no fue validado intente otra vez ";
29         string mensaje_acierto = "El captcha fue validado ";
30         string mensaje_final = "El captcha no logró ser validado :( intente mas tarde";
31
32
33         WHILE (true) THENWHILE
34           !! instruccion
35           INSERT('Iteracion infinita');
```

Línea: 124, Columna: 9

- Es un área en donde podremos escribir el código relacionado con el captcha que queremos generar.
- En caso de haber cargado un archivo loca, es en esta área en donde podremos visualizar y modificar su contenido.
- El editor cuenta con señalización de filas y de filas y columnas para hacer más fácil la detección y corrección de errores.

}

# Salidas del análisis {

## Salidas del análisis

Token: STRING | Valor: "30 "

LEXER:

Token: SEMIC | Valor: ;

LEXER:

Token: ID | Valor: no\_declarada

PARSER:

Expresion asignacion terminada correctamente

LEXER:

Token: EQU | Valor: =

LEXER:

Token: INTEGER | Valor: 40

LEXER:

Token: SEMIC | Valor: ;

## Salidas del análisis

Expresion insert termina correctamente.

PARSER:

Expresion while correctamente

LEXER:

Token: REPEAT | Valor: REPEAT

PARSER:

EXPRESION COMO ERROR ->

Error: ; linea: 22 columna: 52

PARSER:

EXPRESION COMO ERROR ->

Error: ; linea: 25 columna: 20

LEXER:

Token: PAROPN | Valor: (

LEXER:

En esta área se podrán visualizar mensajes de salida producto del análisis del código de etiquetas y scripting, como finalizaciones exitosas de las producciones y errores de los cuales se recuperó.

}

# Tabla de símbolos

{ Esta parte de la aplicación mostrará todos los símbolos declarados, su tipo, valor actual y el modo en el que fueron declarados, además de su ámbito.

También indicará los cambios de valor y si los valores fueron asignados por declaración, asignación o si por el contrario se intenta asignar un valor a una variable no declarada.

}

Tabla de simbolos						
Pos.	Identificador	Tipo	Valor actual	Modo	Ambito	Por
1	contador_fallas	integer	5	@global	FUNCTION_calc()	declaracion
2	result_caja_texto	string	getElementById("entrada_1")	-	FUNCTION_calc()	declaracion
3	result	string	"10 "	@global	FUNCTION_calc()	declaracion
4	result		"30 "		FUNCTION_calc()	asignacion
5	no_declarada		40		FUNCTION_calc()	NO DECLARADO
6	p_declaracion	string	undefined	@global	FUNCTION_calc()	declaracion
7	p2_declaracion	string	undefined	-	FUNCTION_calc()	declaracion
8	mensaje_fallo	string	"El captcha no fue validado intente otra vez "	-	FUNCTION_calc()	declaracion
9	mensaje_acierto	string	"El captcha fue validado "	-	FUNCTION_calc()	declaracion
10	mensaje_final	string	"El captcha no logró ser validado :( intente mas tarde"	-	FUNCTION_calc()	declaracion





## Generador de captchas

## <Manejo de archivos>

Abrir archivo

Guardar

Guardar como

```
78      <C_OPTION>b</C_OPTION>
79      <C_OPTION>c</C_OPTION>
80      <C_OPTION>d</C_OPTION>
81  </C_SELECT>
82
83
84
85  <C_SCRIPTING>
86
87      ON_LOAD () [
88          !!Estas instrucciones se ejecutan media vez se entra al scripting !! Insertamos el input con sus parámetros con la instrucción INSERT
89          INSERT('<C_INPUT [type= "text"] [text-align= "center"] [id= "entrada_1"] >');
90          INSERT('</C_INPUT>');
91      ]
92
93      FUNCTION_calcular() [
94          string mensaje_acierto = "El captcha fue validado ";
95      ]
96
97
98  </C_SCRIPTING>
99
100
101
102
103  <C_LINK [href = "https://i.pinimg.com/564x/be/dc/c5/bedcc5715f193ba54f8af170ba98ae93.jpg"] >EsteEsElLink pero con separaciones</C_LINK>
104  <C_BR>
105  <C_SPAM [color = "red"] [font-size = "15px"] [text-align = "center"] [id = "idSpam"]>Este es el spam</C_SPAM>
106  </C_BR>
107  <C_INPUT[type = "text"]>texto y a ver que pasa</C_INPUT> <C_BR>
108  <C_INPUT[type = "number"]>number</C_INPUT> <C_BR>
109  <C_INPUT[type = "radio"]>radio</C_INPUT> <C_BR>
110  <C_INPUT[type = "checkbox"]>checkbox</C_INPUT> <C_BR>
111  <C_BR>
112  <C_TEXTAREA[cols = "40"] [rows = "2"]>Este es un text area con separaciones</C_TEXTAREA>
113  </C_BR>
114  <C_IMG [alt = "imagen de un gato de internet"] [width = "300px"] [height = "450px"] [src = "https://i.pinimg.com/564x/70/b3/ba/70b3bace7a286266f0d05fc5488a"]></C_IMG>
115
116
117  <C_BUTTON [onclick() = "FUNCTION_SUMAR()"] [color = "#FF5733"] [background = "purple"]>Este es un boton</C_BUTTON>
118  <C_H1 [font-size = "20px"] [text-align = "right"] [color = "maroon"]>HACHEUN000 o un titulo</C_H1>
119  <C_P [color = "lime"] [font-size="40px"]>este es un parrafo mis amigos como la ven</C_P>
120
121  </C_DIV>
122
123  </C_BODY>
124  </C_CC>
```

Línea: 124, Columna: 9

Procesar captcha

Prueba\_1|

<Nombre del archivo>

<Editor  
de  
código>

<Salidas  
del  
Análisis>

### Salidas del análisis

Token: REPEAT | Valor: REPEAT

PARSER:  
EXPRESION COMO ERROR ->  
Error: ; línea: 22 columna: 52

PARSER:  
EXPRESION COMO ERROR ->  
Error: ; línea: 25 columna: 20

LEXER:  
Token: PAROPN | Valor: (

LEXER:  
Token: INT | Valor: integer

LEXER:  
Token: ID | Valor: i

LEXER:  
Token: EQU | Valor: =

### Tabla de símbolos

Pos.	Identificador	Tipo	Valor actual	Modo	Ambito	Por
1	contador_fallas	integer	5	@global	FUNCTION_calc()	declaracion
2	result_caja_texto	string	getElementById("entrada_1")	-	FUNCTION_calc()	declaracion
3	result	string	"10 "	@global	FUNCTION_calc()	declaracion
4	result		"30 "		FUNCTION_calc()	asignacion
5	no_declarada		40		FUNCTION_calc()	NO DECLARADO
6	p_declaracion	string	undefined	@global	FUNCTION_calc()	declaracion
7	p2_declaracion	string	undefined	-	FUNCTION_calc()	declaracion
8	mensaje_fallo	string	"El captcha no fue validado intente otra vez "	-		acion
9	mensaje_acierto	string	"El captcha fue validado "	-		acion
10	mensaje_final	string	"El captcha no logró ser validado ¿ intente mas tarde "			
11	i	integer	0			

<Tabla  
de  
símbolos>



# Captchas generados

Generador de captchas		Abrir archivo	Guardar	Guardar como	Captchas generados
<a href="#">captcha1</a>					Visitado: 2 veces
<a href="#">captcha2</a>					Visitado: 4 veces
<a href="#">ejemploCaptcha</a>					Visitado: 2 veces
<a href="#">captcha_4</a>					Visitado: 1 veces
<a href="#">ejemplo2</a>					Visitado: 1 veces

Para acceder de manera fácil a los captchas generados por el generador de captchas hay un menú desplegable que muestra enlaces a las páginas junto a su número de visitas.

}

Servidores  
web{

}



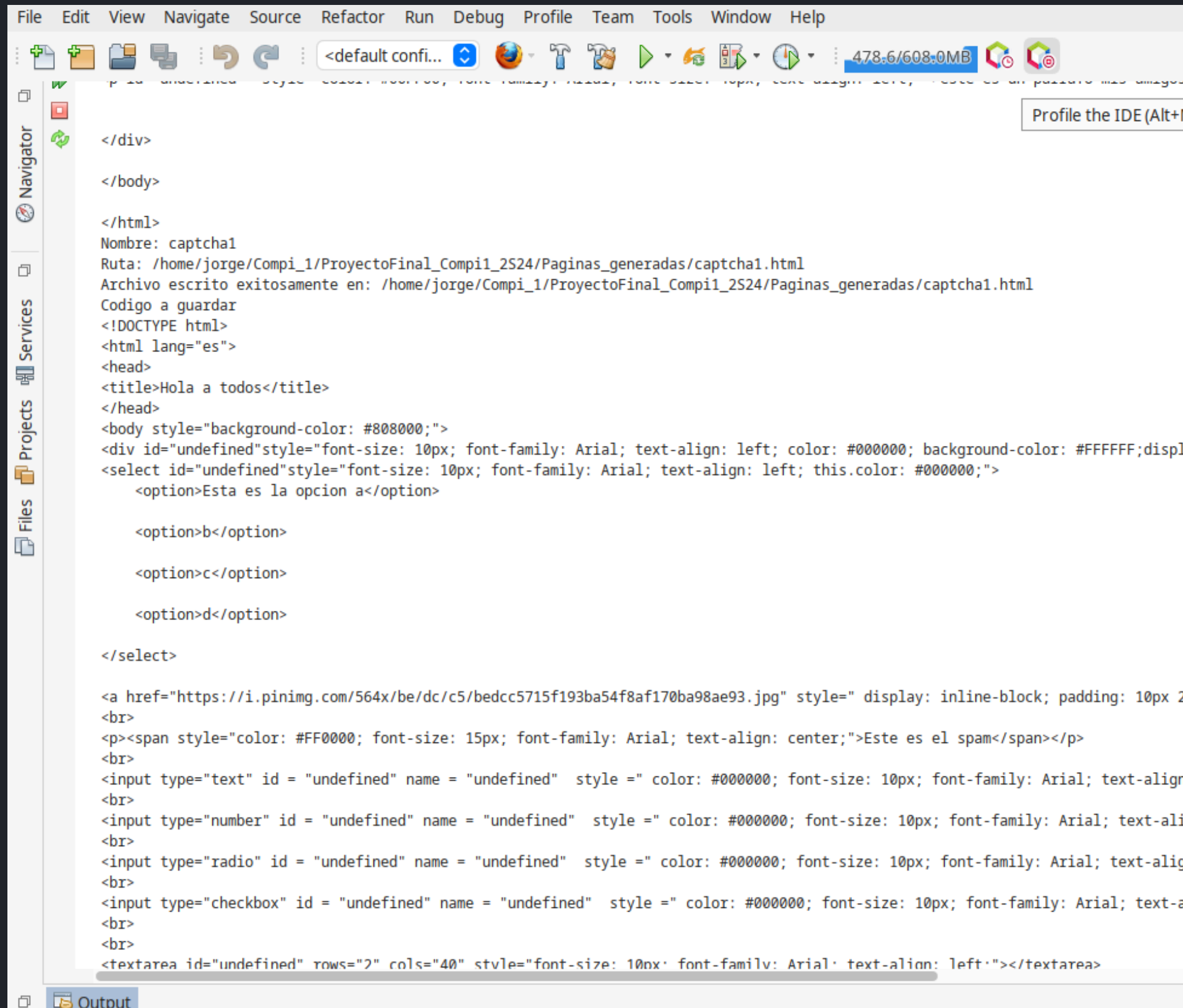
# Inicio del servidor de almacenamiento {

El servidor le permitirá a la aplicación web acceder a los archivos dentro de la computadora, ya que por seguridad el navegador no puede hacerlo.

Para inciarlo, será necesario tener tomcat instalado junto a un entorno que permita su ejecución.

El servidor únicamente indicará por consola las peticiones recibidas por el cliente así como si pudo escribir o no un archivo.

}



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default confi... 478.6/608.0MB
Profile the IDE (Alt+M)

</div>

</body>

</html>
Nombre: captcha1
Ruta: /home/jorge/Compi_1/ProyectoFinal_Compi1_2S24/Paginas_generadas/captcha1.html
Archivo escrito exitosamente en: /home/jorge/Compi_1/ProyectoFinal_Compi1_2S24/Paginas_generadas/captcha1.html
Codigo a guardar
<!DOCTYPE html>
<html lang="es">
<head>
<title>Hola a todos</title>
</head>
<body style="background-color: #808000;">
<div id="undefined" style="font-size: 10px; font-family: Arial; text-align: left; color: #000000; background-color: #FFFFFF; display: inline-block; padding: 5px 10px;">
<select id="undefined" style="font-size: 10px; font-family: Arial; text-align: left; this.color: #000000;">
  <option>Esta es la opcion a</option>

  <option>b</option>

  <option>c</option>

  <option>d</option>
</select>

<a href="https://i.pinimg.com/564x/be/dc/c5/bedcc5715f193ba54f8af170ba98ae93.jpg" style="display: inline-block; padding: 10px 20px;">
<br>
<p><span style="color: #FF0000; font-size: 15px; font-family: Arial; text-align: center;">Este es el spam</span></p>
<br>
<input type="text" id="undefined" name="undefined" style="color: #000000; font-size: 10px; font-family: Arial; text-align: left;">
<br>
<input type="number" id="undefined" name="undefined" style="color: #000000; font-size: 10px; font-family: Arial; text-align: left;">
<br>
<input type="radio" id="undefined" name="undefined" style="color: #000000; font-size: 10px; font-family: Arial; text-align: left;">
<br>
<input type="checkbox" id="undefined" name="undefined" style="color: #000000; font-size: 10px; font-family: Arial; text-align: left;">
<br>
<br>
<textarea id="undefined" rows="2" cols="40" style="font-size: 10px; font-family: Arial; text-align: left;"></textarea>
```

# Inicio del servidor de captchas{

```
Paginas_generadas:python3
jorge on ultramarine ProyectoFinal_Compil_2S24/Paginas_generadas on y enlaces [!]
> python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
127.0.0.1 - - [30/Oct/2024 11:14:08] "GET /captcha1.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:14:08] code 404, message File not found
127.0.0.1 - - [30/Oct/2024 11:14:08] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [30/Oct/2024 11:14:51] "GET /captcha1.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:15:23] "GET /captcha1.html HTTP/1.1" 304 -
127.0.0.1 - - [30/Oct/2024 11:20:46] "GET /captcha1.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:24:11] "GET /captcha1.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:25:02] "GET /captcha2.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:29:40] "GET /captcha1.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:36:13] "GET /captchaGatitos.html HTTP/1.1" 200 -
127.0.0.1 - - [30/Oct/2024 11:36:21] "GET /captchaGa
```

Este servidor permitirá a la aplicación web acceder a las páginas generadas y almacenadas.

Para ejecutarlo, tendremos que ubicarnos en la carpeta donde se almacenan las páginas y ejecutar el comando:

```
python3 -m http.server 8000
```

}