MANUAL DE USUARIO

ANALIZADOR LEXICO

Requisitos de sistema

Requisitos de Hardware

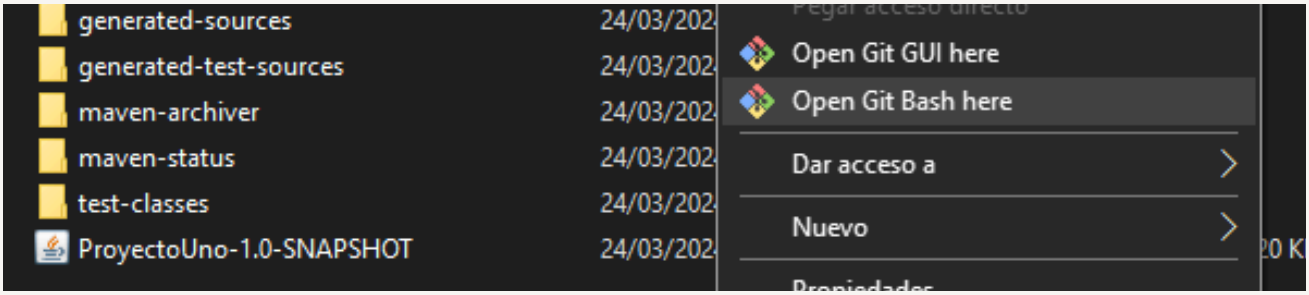
- Procesador: Cualquier procesador compatible con los sistemas operativos actuales.
- Memoria RAM: 2 GB o más recomendado.
- Tener instalado GraphVIz Version dot
- Tener instalado Java 17 o posterior

Requisitos de Software, cualquier sitema operativo:

- Windows (7, 8, 10)
- macOS
- Linux

Iniciar Analizador Lexico

En el paquete proporcionado por el desarrollador encontrara un archivo con nombre como el de la imagen. tiene que tener instalado la consolo gitbash de preferencia,



PRESIONA DOS VECES SOBRE EL ARCHVIO DE PROGRAMA Y SE EJECUTARA EL PROGRAMA

Pantalla de Edición de Código

Esta pantalla permite al usuario ingresar y editar código en los lenguajes HTML, CSS y JavaScript. En la parte superior, se muestra la ubicación del cursor en el código, indicando la línea y columna actuales.

Elementos principales:

Campo de texto de código: Ocupa la mayor parte de la pantalla y permite escribir código de varios lenguajes. En este ejemplo, se ha ingresado código HTML, CSS y JavaScript.

Botón "ANALIZAR": Situado en la parte superior junto a la indicación de línea y columna, este botón ejecuta una acción para analizar el código que se ha ingresado.

Botón "Guardar Lienzo": En la esquina superior derecha, este botón permite guardar los cambios realizados en el código.

Código de ejemplo:

HTML: Un simple documento HTML con un título ("Prueba de Analizador") y un encabezado "Hola Mundo".

CSS: Estilos básicos que cambian los colores de los encabezados y párrafos.

JavaScript: Un comentario y un código que registra en la consola un mensaje cuando la página se carga.

Linea: 31 Columna 1 ANALIZAR

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba de Analizador</title>
</head>
<body>
    <h1>Hola Mundo</h1>
    <p>Este es un ejemplo de prueba.</p>
</body>
</html>

>>[css]
body {
    background-color: #f0f0f0;
    font-family: Arial, sans-serif;
}

h1 {
    color: blue;
}

p {
    color: green;
}

>>[JS]
//comentario 1
// comentario
document.addEventListener('DOMContentLoaded', function() { // coemntario
    console.log('Página cargada y lista para usarse.');
});
```

Generacion De Reportes

La interfaz presenta una sección de generación de reportes que muestra un análisis detallado de los paneles de colores generados en la aplicación. Esta sección se divide en tres partes principales:

Tabla de "Tokens Encontrados": Muestra un listado con las siguientes columnas:

Token: Identifica palabras reservadas, variables, o colores que han sido detectados durante el análisis.

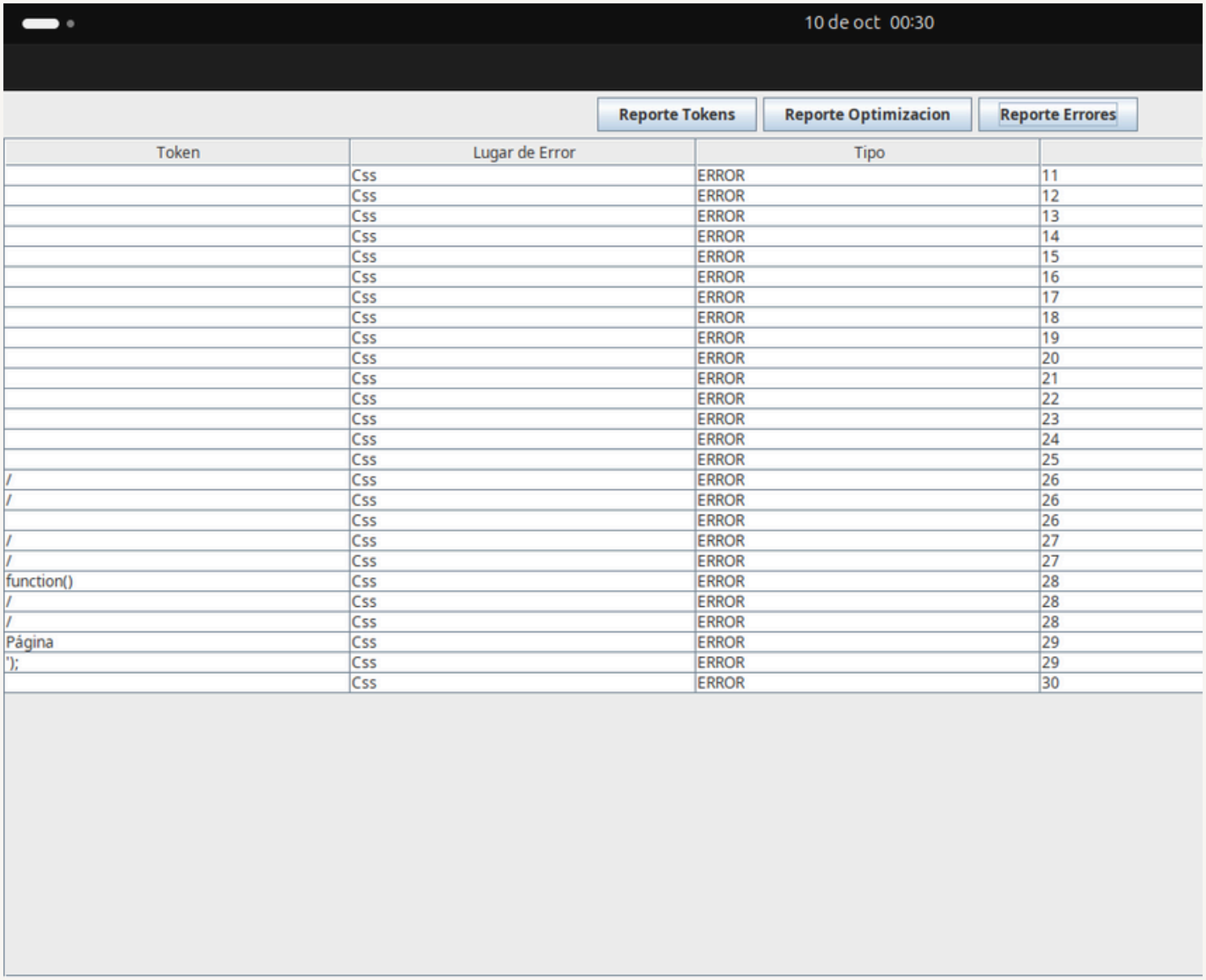
Lexema: Representa el contenido o el valor específico del token identificado.

Fila y Columna: Indican la posición en términos de fila y columna del token en el documento o área analizada.

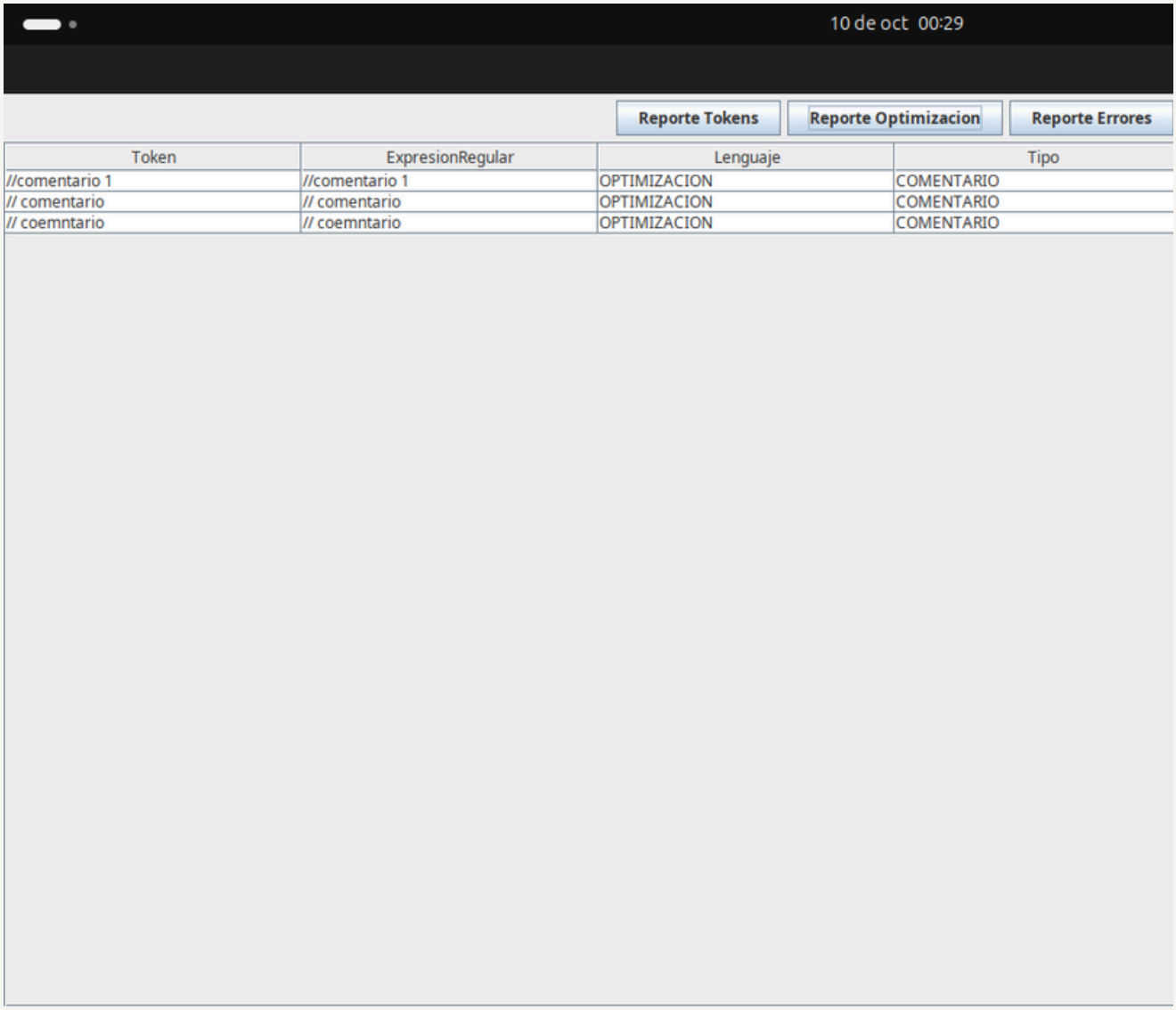
Fila Cuadro y Columna Cuadro: Definen la ubicación específica del cuadro correspondiente al token en la cuadrícula generada.

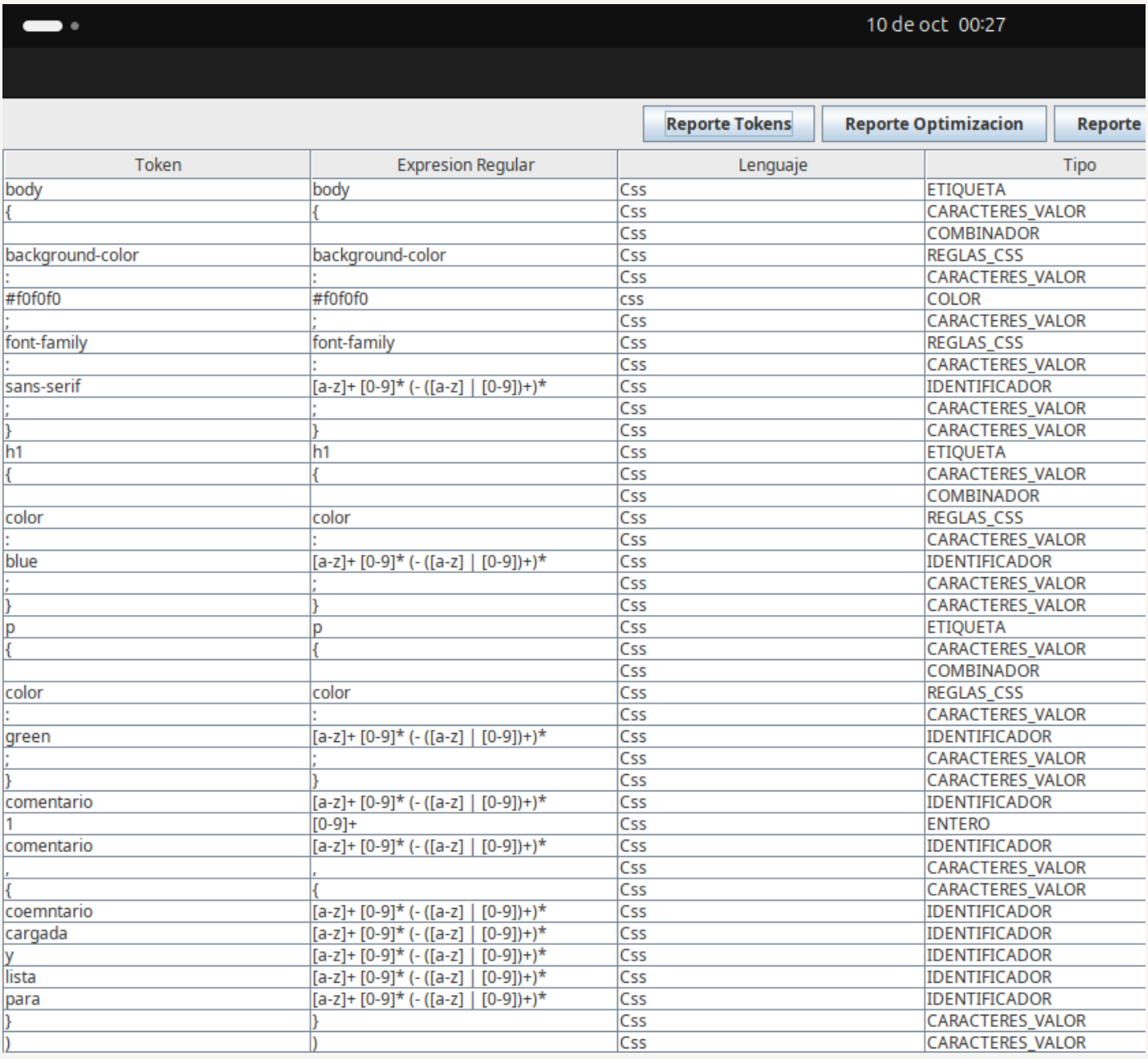
Color: Muestra el código hexadecimal del color asociado con el token, como #60A917 para verde o #FFD300 para amarillo.

Esta pantalla presenta los resultados del análisis del código en tres tipos de reportes: Tokens, Optimización, y Errores. El usuario puede cambiar entre estos reportes a través de las pestañas situadas en la parte superior de la tabla.
Elementos principales:
Pestañas de Reporte:
Reporte Tokens: Muestra un desglose detallado de cada token identificado en el código, incluyendo su expresión regular, lenguaje, tipo y su ubicación exacta en el archivo (fila y columna).
Reporte Optimización: (Sin contenido visible en la imagen, pero esta pestaña permitirá visualizar recomendaciones para optimizar el código).
Reporte Errores: (Sin contenido visible en la imagen, pero aquí se mostrarán los errores encontrados durante el análisis).
Tabla de Resultados:
Token: Muestra el nombre de cada token identificado, como etiquetas, selectores CSS, o elementos dentro del código.
Expresión Regular: Representa la expresión regular utilizada para identificar el token.
Lenguaje: Indica el lenguaje al que pertenece el token (en este caso, todo es CSS).
Tipo: Clasificación del token, como "ETIQUETA", "REGLAS_CSS", o "CARACTERES_VALOR".
Fila y Columna: Posición exacta en el código donde se encuentra el token, lo que ayuda al usuario a localizar el elemento en cuestión.



Token	Lugar de Error	Tipo	
	Css	ERROR	11
	Css	ERROR	12
	Css	ERROR	13
	Css	ERROR	14
	Css	ERROR	15
	Css	ERROR	16
	Css	ERROR	17
	Css	ERROR	18
	Css	ERROR	19
	Css	ERROR	20
	Css	ERROR	21
	Css	ERROR	22
	Css	ERROR	23
	Css	ERROR	24
	Css	ERROR	25
/	Css	ERROR	26
/	Css	ERROR	26
	Css	ERROR	26
/	Css	ERROR	27
/	Css	ERROR	27
function()	Css	ERROR	28
/	Css	ERROR	28
/	Css	ERROR	28
Página	Css	ERROR	29
');	Css	ERROR	29
	Css	ERROR	30


10 de oct 00:29
Reporte Tokens
Reporte Optimizacion
Reporte Errores
Token
ExpresionRegular
Lenguaje
Tipo
//comentario 1
//comentario 1
OPTIMIZACION
COMENTARIO
// comentario
// comentario
OPTIMIZACION
COMENTARIO
// coemntario
// coemntario
OPTIMIZACION
COMENTARIO



Token	Expresion Regular	Lenguaje	Tipo
body	body	Css	ETIQUETA
{	{	Css	CARACTERES_VALOR
		Css	COMBINADOR
background-color	background-color	Css	REGLAS_CSS
:	:	Css	CARACTERES_VALOR
#f0f0f0	#f0f0f0	css	COLOR
;	;	Css	CARACTERES_VALOR
font-family	font-family	Css	REGLAS_CSS
:	:	Css	CARACTERES_VALOR
sans-serif	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
;	;	Css	CARACTERES_VALOR
}	}	Css	CARACTERES_VALOR
h1	h1	Css	ETIQUETA
{	{	Css	CARACTERES_VALOR
		Css	COMBINADOR
color	color	Css	REGLAS_CSS
:	:	Css	CARACTERES_VALOR
blue	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
;	;	Css	CARACTERES_VALOR
}	}	Css	CARACTERES_VALOR
p	p	Css	ETIQUETA
{	{	Css	CARACTERES_VALOR
		Css	COMBINADOR
color	color	Css	REGLAS_CSS
:	:	Css	CARACTERES_VALOR
green	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
;	;	Css	CARACTERES_VALOR
}	}	Css	CARACTERES_VALOR
comentario	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
1	[0-9]+	Css	ENTERO
comentario	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
,	,	Css	CARACTERES_VALOR
{	{	Css	CARACTERES_VALOR
coemntario	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
cargada	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
y	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
lista	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
para	[a-z]+ [0-9]* (- ([a-z] | [0-9])+)*	Css	IDENTIFICADOR
}	}	Css	CARACTERES_VALOR
)	)	Css	CARACTERES_VALOR