

**Created By Freepik**

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CHETUMAL



**Ing. En Sistemas Computacionales**

**Programación Web**

**Angel Alberto May Catzin**

**Reporte de Actividad**

**Grupo: I6U**

**Junio 29 de 2020**

Contenido

[Reporte de actividad 3](#_Toc44665087)

[Uso de la aplicación 3](#_Toc44665088)

[Consideraciones para elaborar la actividad. 9](#_Toc44665089)

[Validar HTML y CSS 12](#_Toc44665090)

# Reporte de actividad

## Uso de la aplicación

La actual aplicación web, usted podrá abrirlo o visualizarlo en su navegador web (Por ejemplo: Google Chrome). Para utilizar la aplicación web, se debe tener en cuenta que únicamente podemos leer archivos CSV, con lo cual se podrá visualizar su contenido; es decir, si usted carga un archivo que no sea CSV la aplicación la rechazará y no podrá ver su contenido.

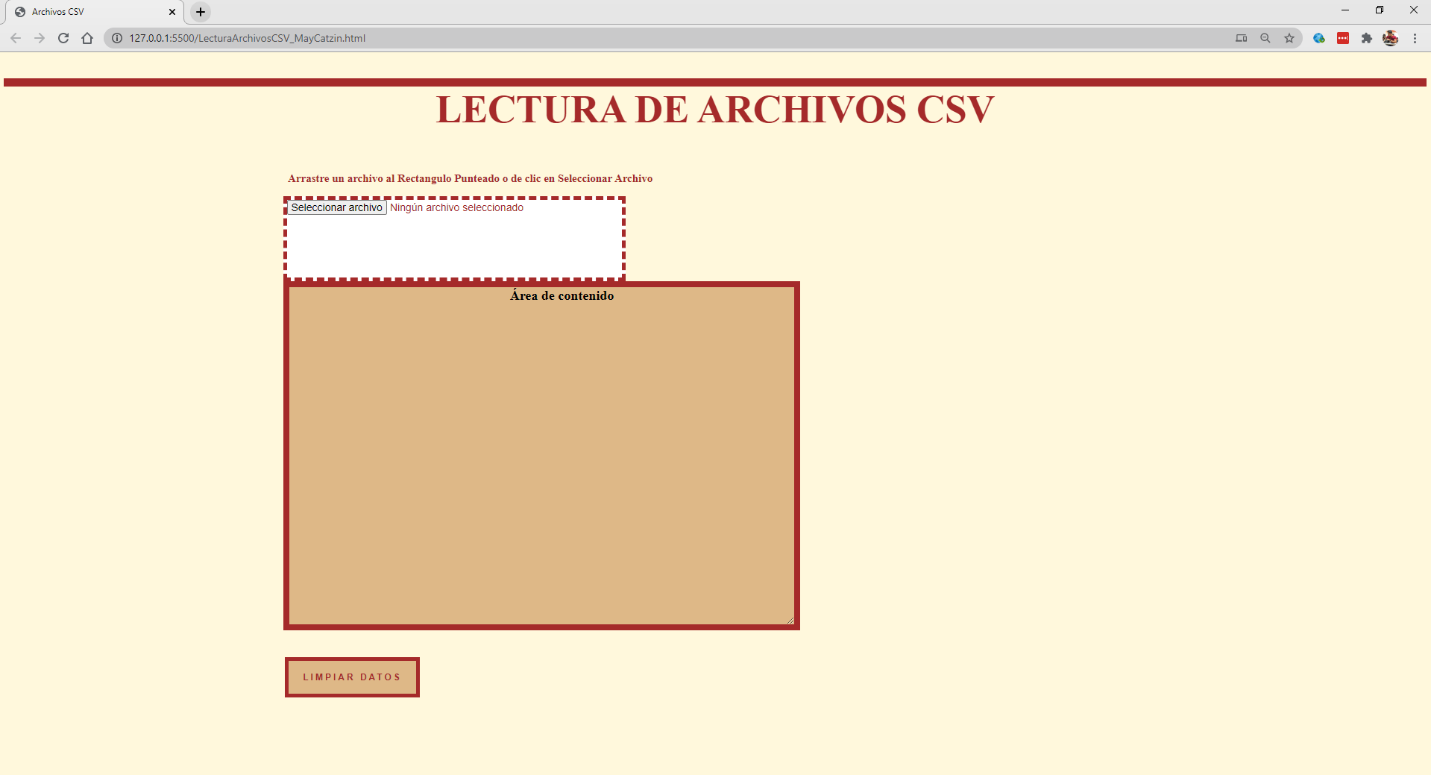
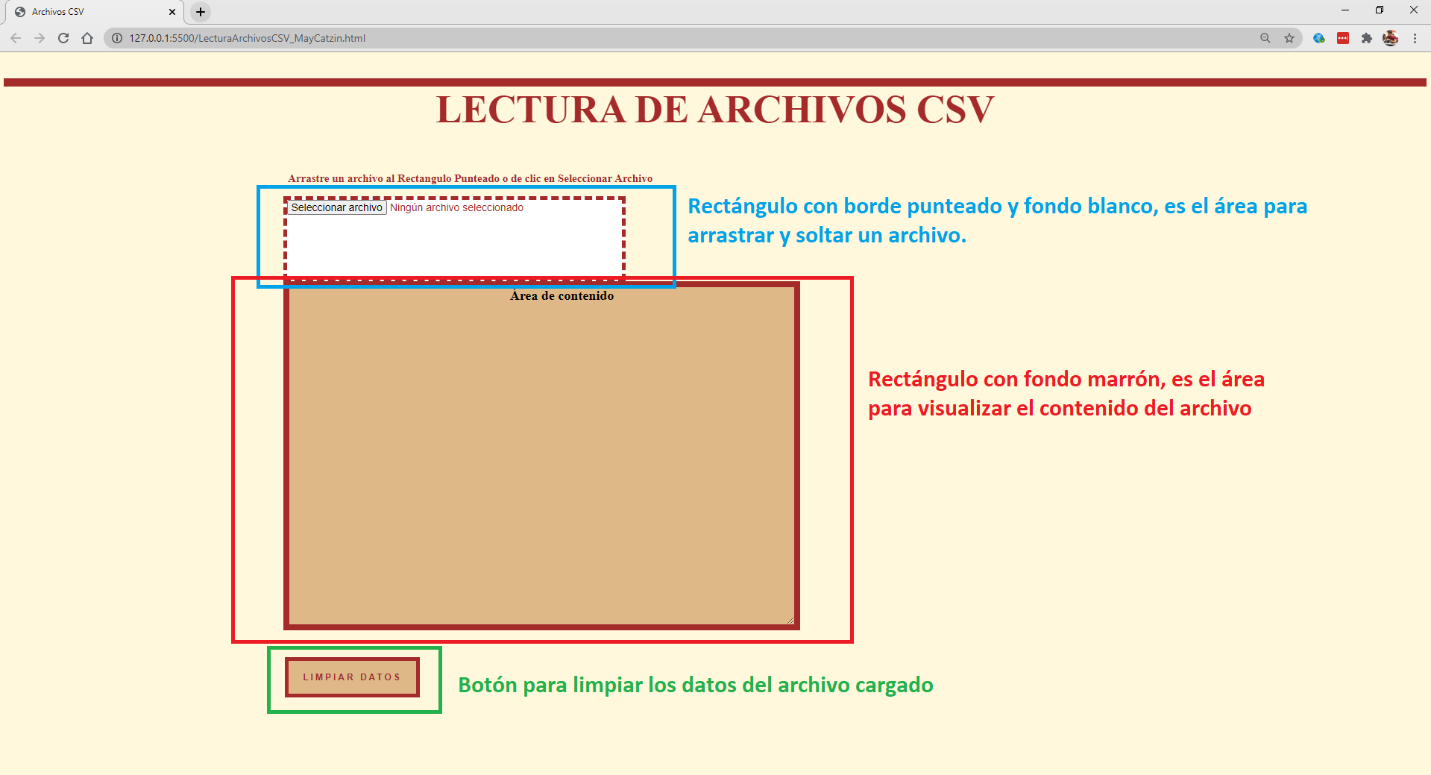


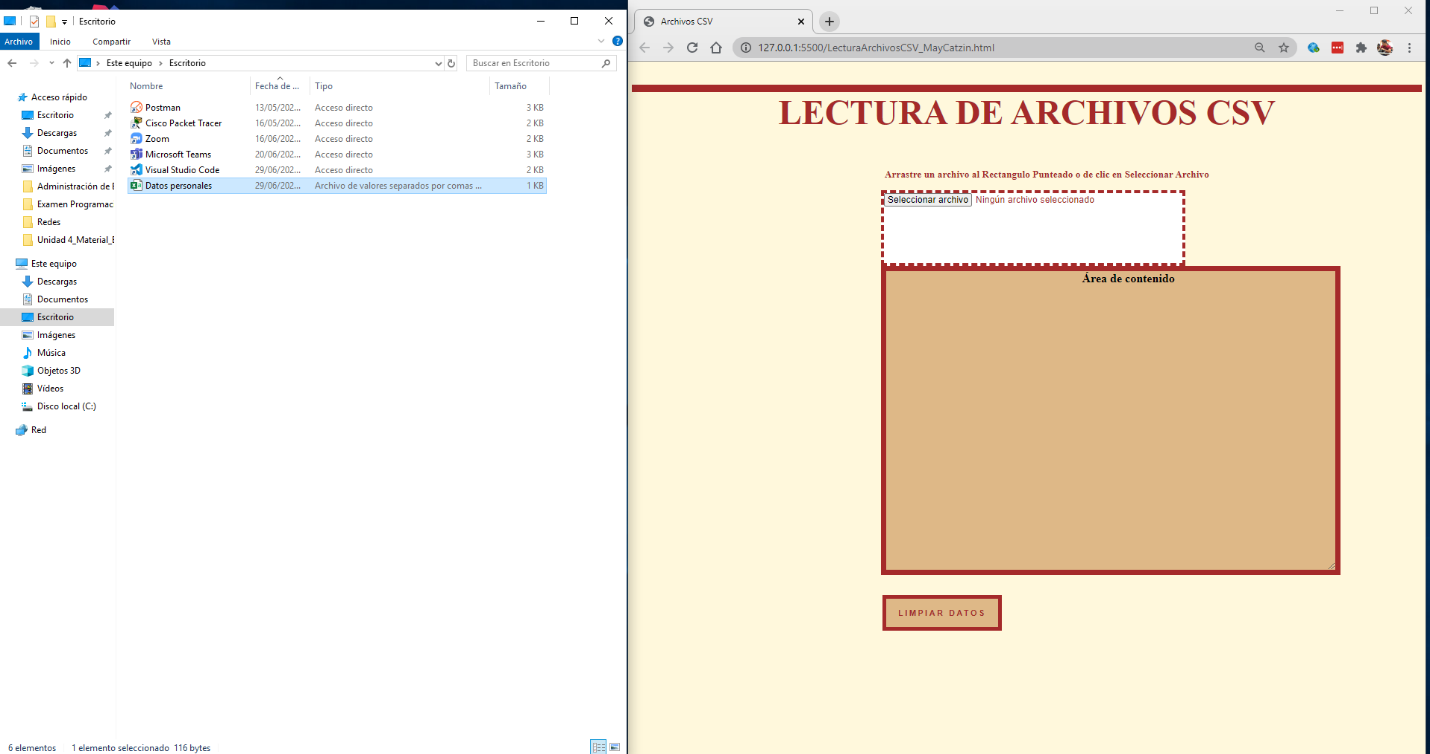
Ilustración 1. Aplicación web

Como podemos ver en la siguiente imagen, se muestran las funcionalidades de cada componente de la aplicación web:

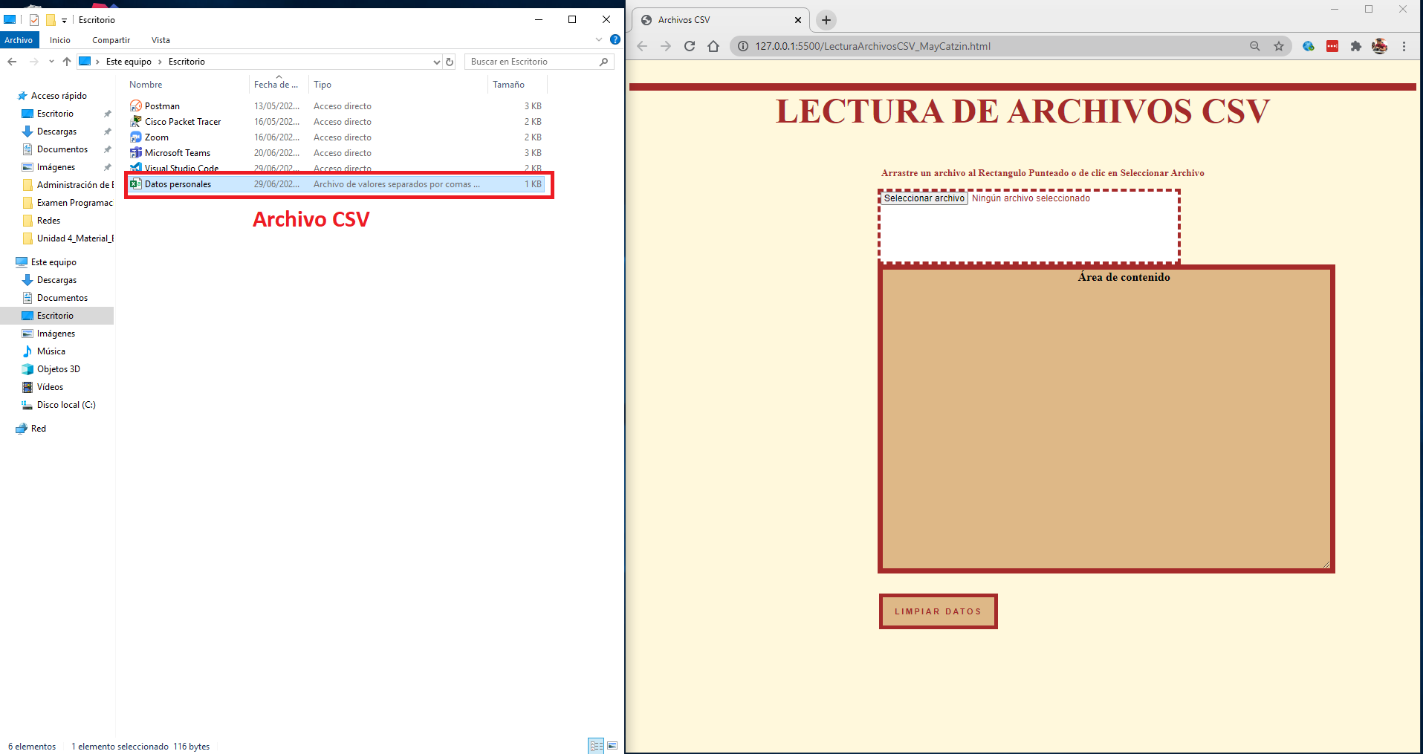


Tenemos dos opciones para leer un archivo CSV: arrastrar el archivo o seleccionarlo desde la página web.

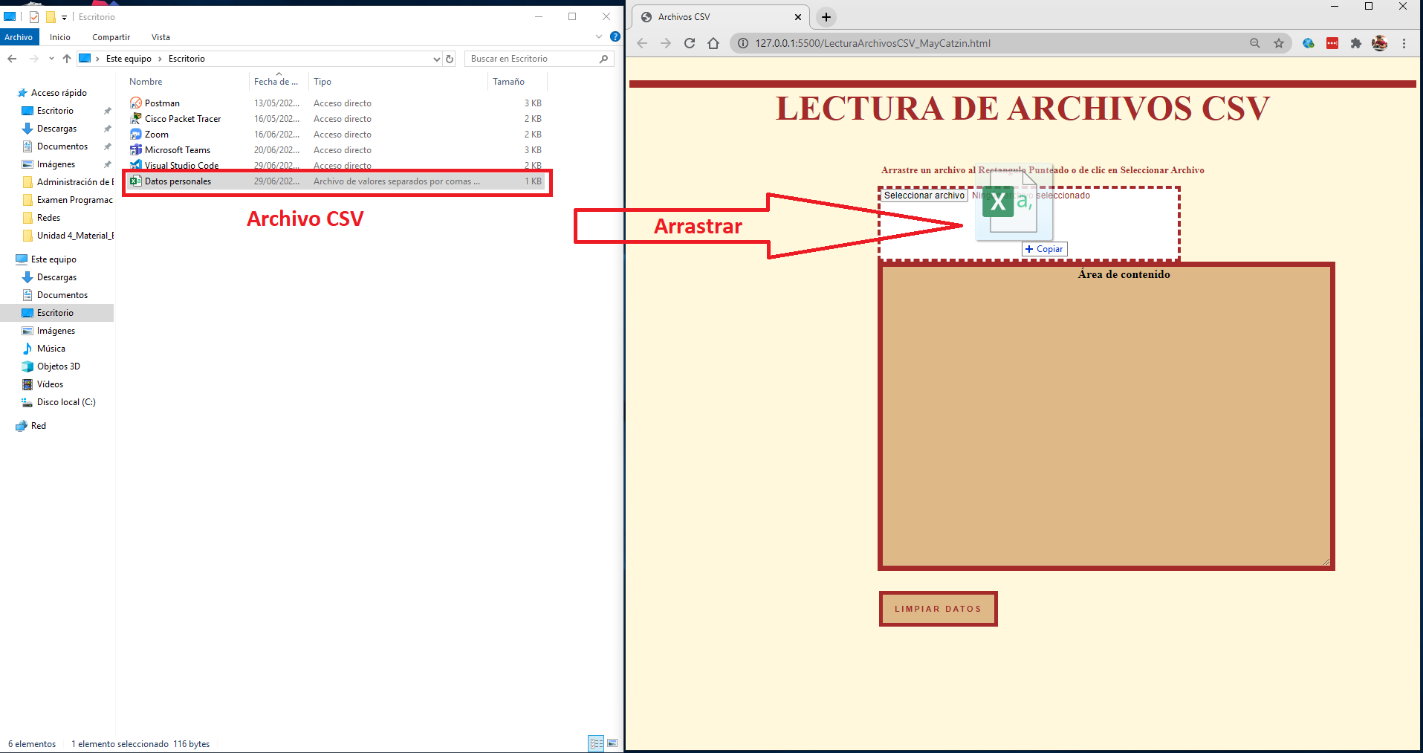
1. Para arrastrar el archivo, debemos ubicar el archivo dentro de nuestro ordenador, podemos poner las ventanas (explorador de archivos y navegador) en paralelo:



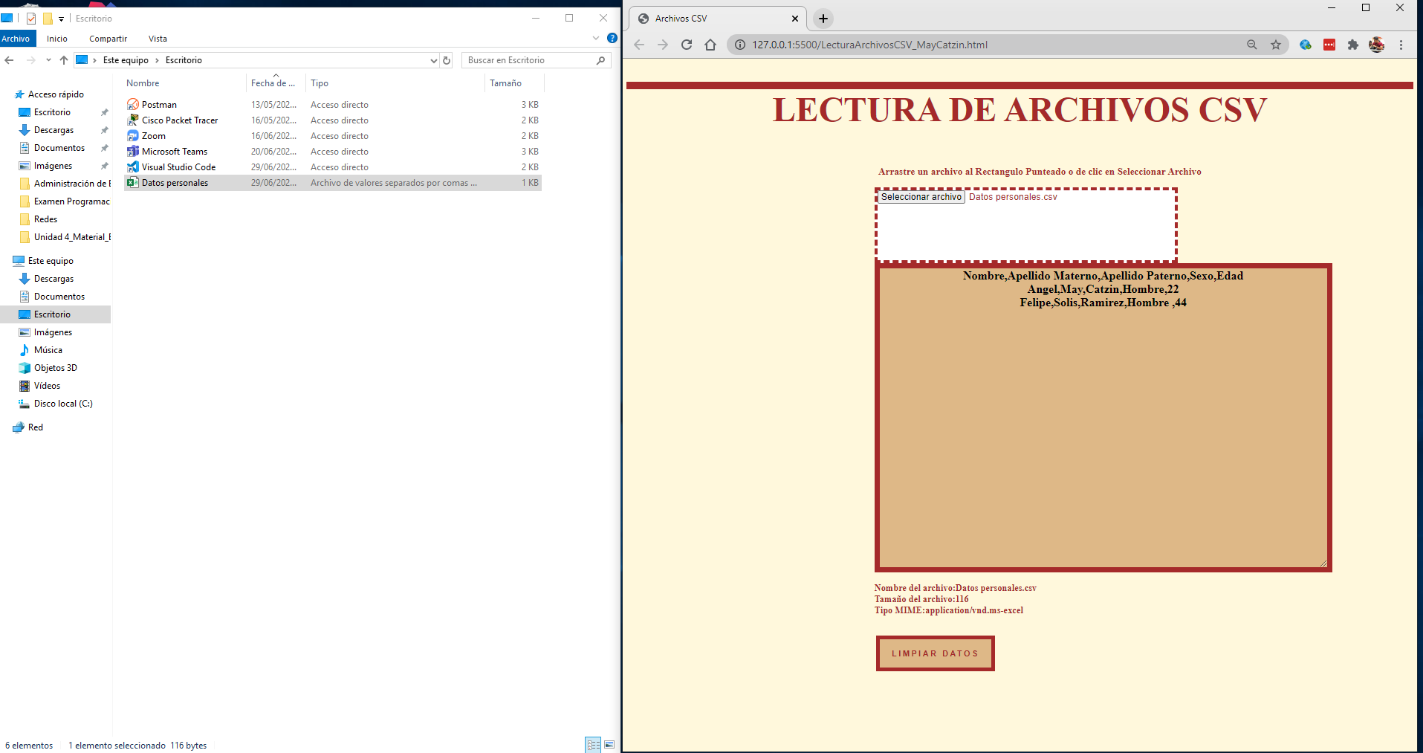
Después, tenemos que ubicar el archivo CSV desde el explorador de archivos, en nuestro caso en el escritorio tenemos un archivo CSV llamado **Datos Personales** (en tipo, podemos ver que es un archivo de valores separados por comas); el cual vamos a arrastrar a la aplicación web para visualizarla.



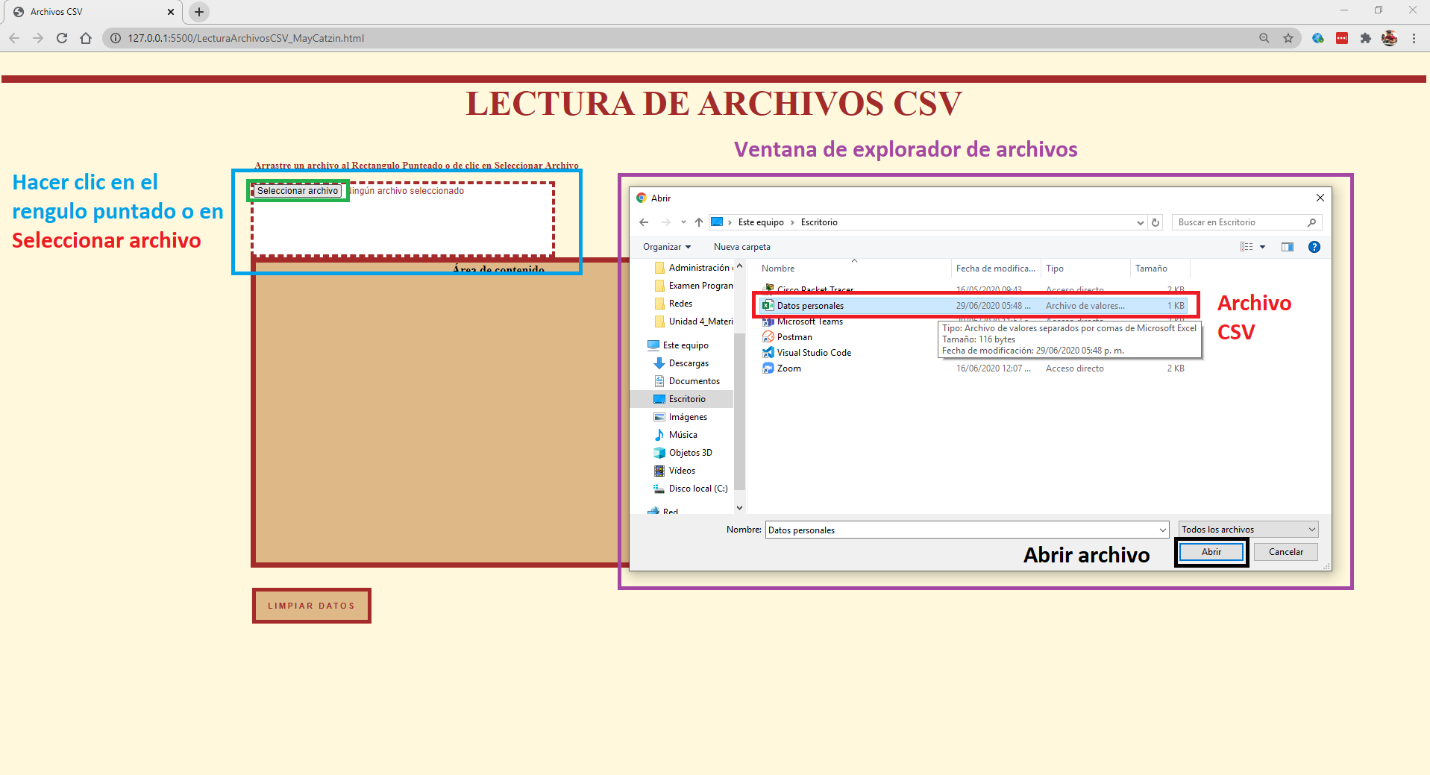
Por último, únicamente tenemos que arrastrar y soltar el archivo ya ubicado al rectángulo punteado con color de fondo blanco, enseguida usted podrá visualizar el contenido de este mismo:



En la visualización del archivo, usted podrá notar que la primera fila hace referencia a los encabezados del archivo (nombre de los campos) y que las demás filas muestran los datos.



1. La segunda forma de cargar el archivo es desde la página, únicamente usted tendrá que dar clic en la parte de **seleccionar archivo o en rectángulo punteado** y nos abrirá la ventana del explorador para ubicar el archivo:

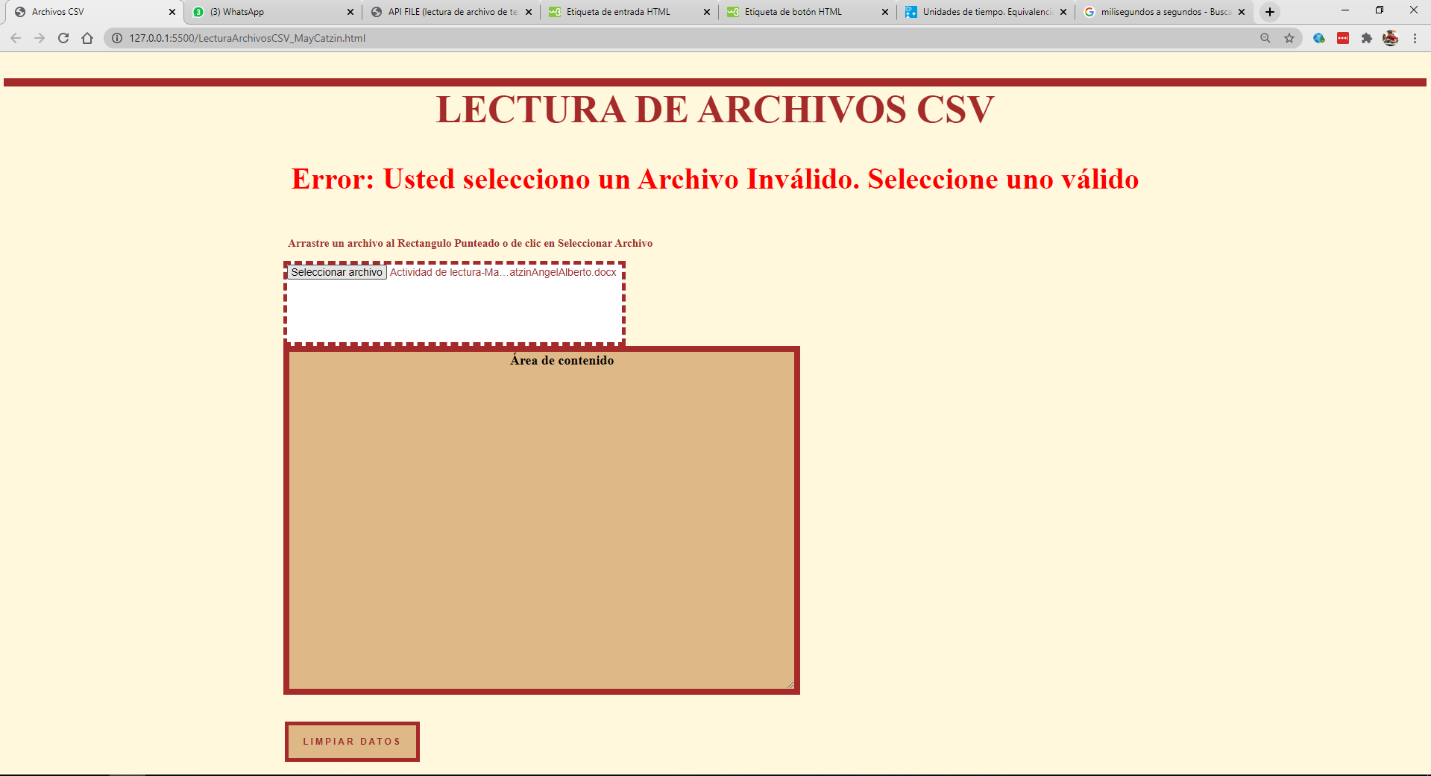


Seleccionamos el archivo y damos en abrir, después se nos desplegara el contenido y los detalles del archivo:



Esas son las dos formas de cargar el archivo CSV que usted desea visualizar.

Por otra parte, si usted carga un archivo inválido, se le mostrara un texto en rojo: Error: Usted selecciono un Archivo Inválido. Seleccione uno válido, y después de 4 segundos, se recargará la página para que usted pueda ingresar un nuevo archivo válido:



## Consideraciones para elaborar la actividad.

Para poder desarrollar la aplicación web, se tienen tres elementos o etiquetas integradas en la interfaz del usuario, se muestran en la siguiente imagen:

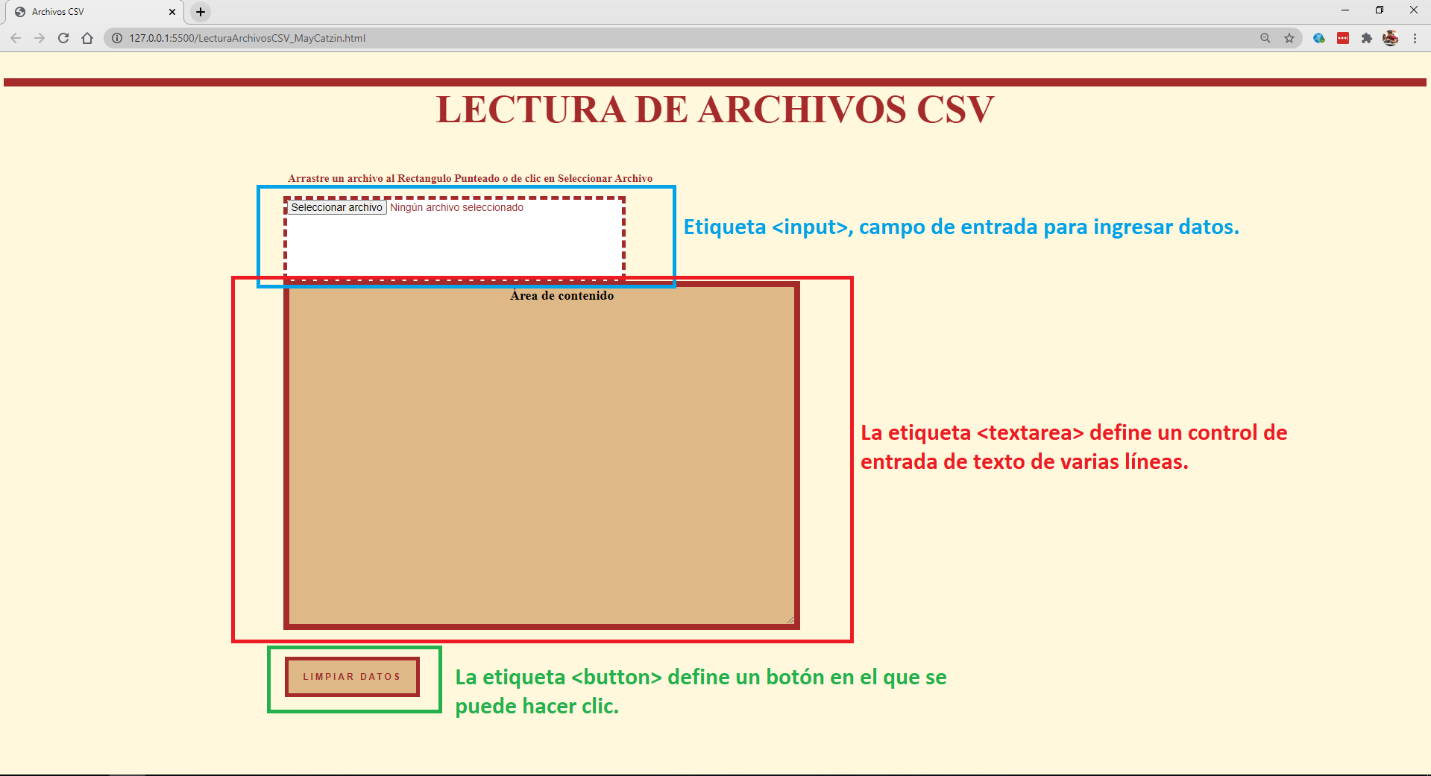


Ilustración 2. Elementos de la aplicación web

1. Definimos una etiqueta input (campo de entrada para ingresar datos) de tipo file que es el encargado de permitirnos seleccionar o arrastrar un archivo:



1. Disponemos una etiqueta textarea (control de entrada de texto de varias líneas) donde mostraremos el contenido del archivo que seleccione el usuario:



1. Además, disponemos de una etiqueta párrafo donde mostraremos el nombre de archivo que se leerá, su tamaño y tipo:



1. Por último, tenemos una etiqueta botón en el que podemos hacer clic para actualizar la página y seleccionar o arrastrar un nuevo archivo:



Para la funcionalidad de los elementos de la interfaz de usuario, definimos un archivo JavaScript externo para separar el HTML del código, con lo cual, para poder usar el script externo, colocamos el nombre del archivo de script en el src atributo (fuente) de una etiqueta <script> ubicada en el <body>:

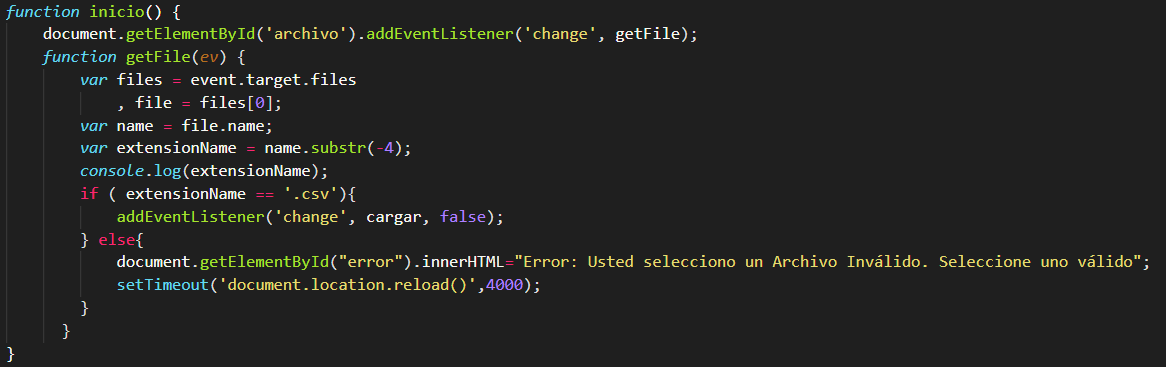


El archivo JavaScript, contiene las funciones de la interfaz de usuario:

1. Tenemos un evento load que se dispara cuando un recurso y sus recursos dependientes han terminado de cargar, es decir, al presionar seleccionar o arrastrar archivo nosotros podemos cargar un recurso (archivo CSV):



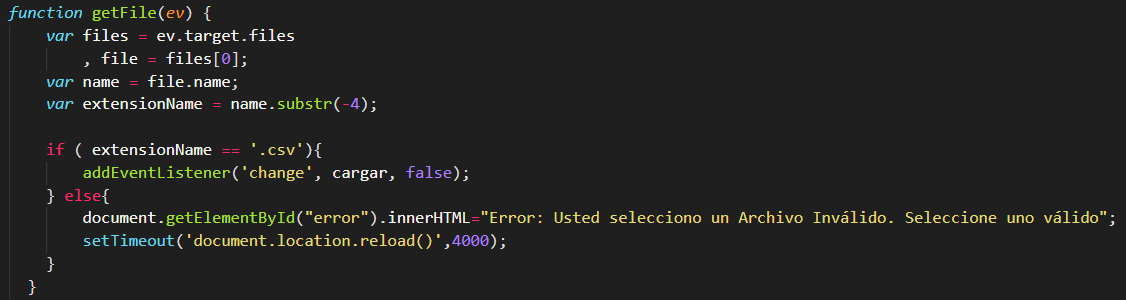
1. Una vez que se dispara el evento load, nos permitirá acceder a la función llamada inicio:



Aquí añadimos dinámicamente un listener tipo change y accedemos a la función llamada getFile:

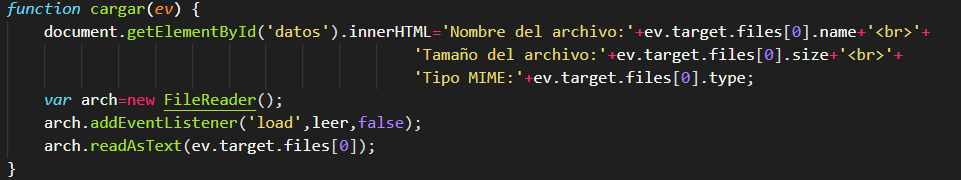


Aquí, nosotros hacemos referencia a nuestra etiqueta input, en la cual cargamos el archivo. La función llamada getFile, esta implementada para validar si el usuario esta cargando un archivo válido (únicamente archivos CSV), para ello se obtiene el nombre del archivo que se cargar y después se obtiene la extensión de este para luego ser validada en una sentencia condicional (if…else):



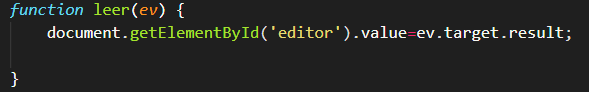
Cuando entramos a la sentencia condicional, validamos o tenemos la condición que nos permitirá analizar si la extensión del nombre del archivo es igual a '.csv', una vez cumplida la condición, accedemos a la función llamada cargar, de lo contrario nos mostrara el texto en la etiqueta que tiene como id error: “Error: Usted selecciono un Archivo Inválido. Seleccione uno válido” y en 4 segundos se actualizará la página para intentar nuevamente cargar un archivo valido.

1. Cuando cargamos el archivo valido, llamamos a la función cargar que recibe un objeto de la clase File que lo accedemos: ev.target.files[0], este objeto tiene tres atributos name (nombre del archivo que acabamos de seleccionar), size (tamaño en bytes del archivo) y type (tipo de archivo):



Luego de mostrar las tres propiedades fundamentales del archivo procedemos a crear un objeto de la clase FileReader para poder acceder al contenido del archivo. Mediante la llamada al método readAsText procedemos a leer el contenido del archivo y registramos el evento load para indicar la función que se disparará cuando tengamos todo el archivo en memoria.

1. Por último, tenemos la función leer; recibe un objeto que almacena todos los datos contenidos en del archivo:



### Validar HTML y CSS

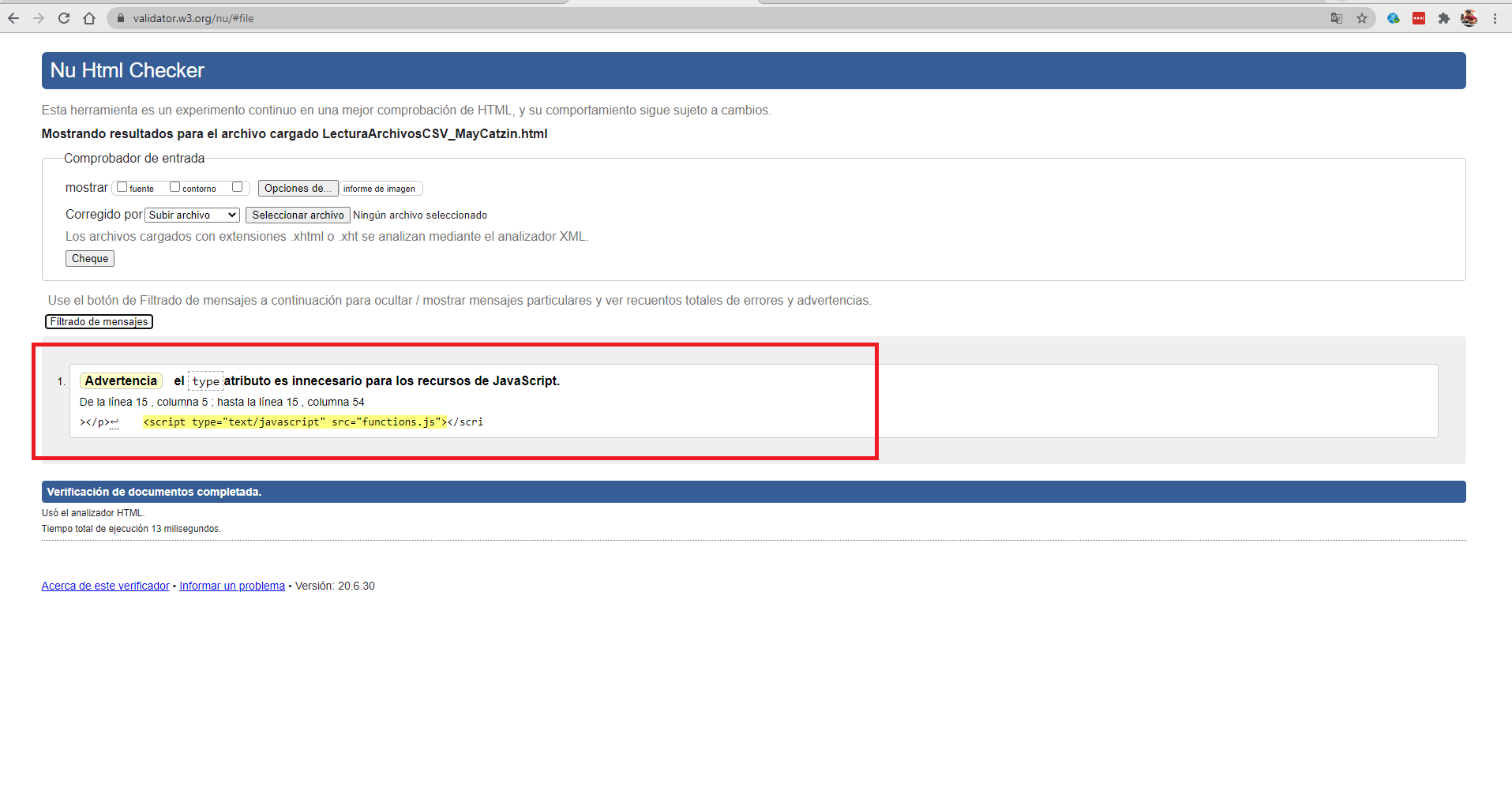
Para comprobar o verificar si nuestro HTML contiene errores y advertencias, podemos hacer uso de Markup Validation Service:



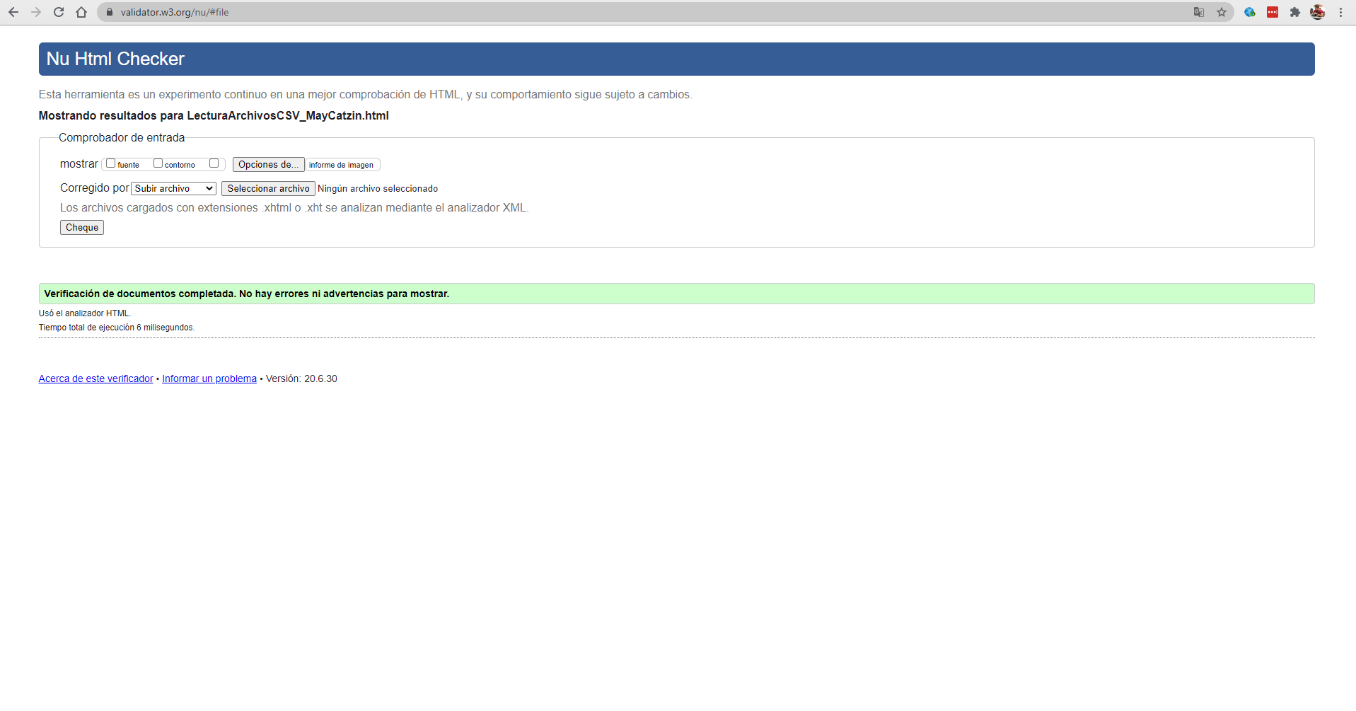
Es una herramienta que nos permite verificar la validez de marcado de los documentos web.

Link: <https://validator.w3.org/>

Al validar nuestro documento HTML nos mostró la siguiente advertencia:

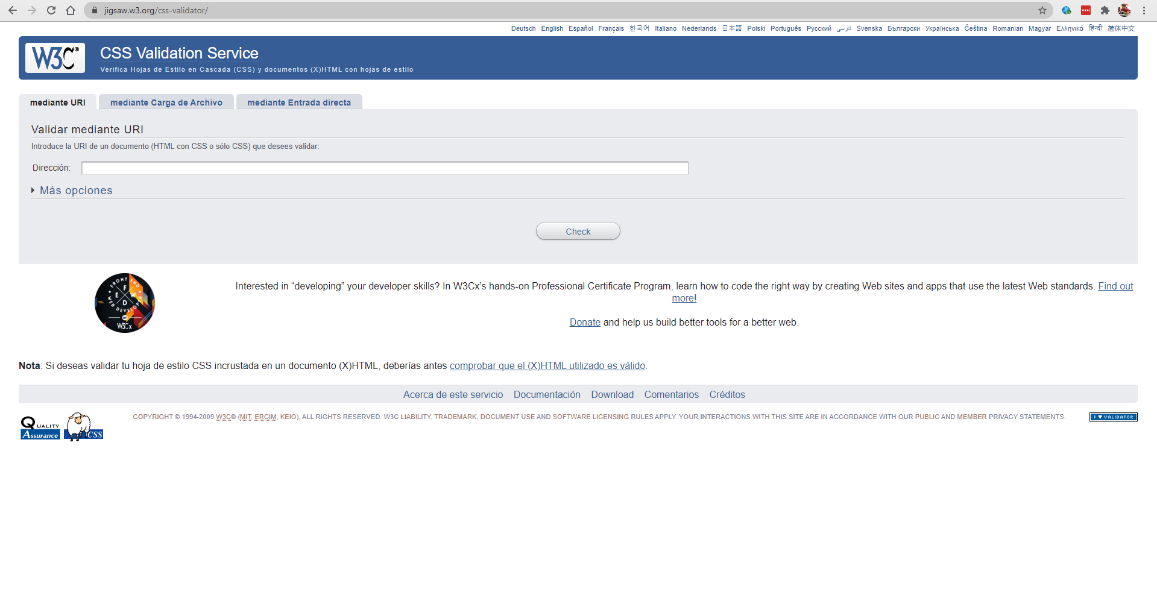


Esto se debe a que en HTML 5 no se requiere el atributo type en la etiqueta <script>, porque en HTML 5 el lenguaje de script predeterminado es JavaScript. Al corregir esta advertencia y volver a validar el documento HTML, nos arrojó que todo está correcto:



\

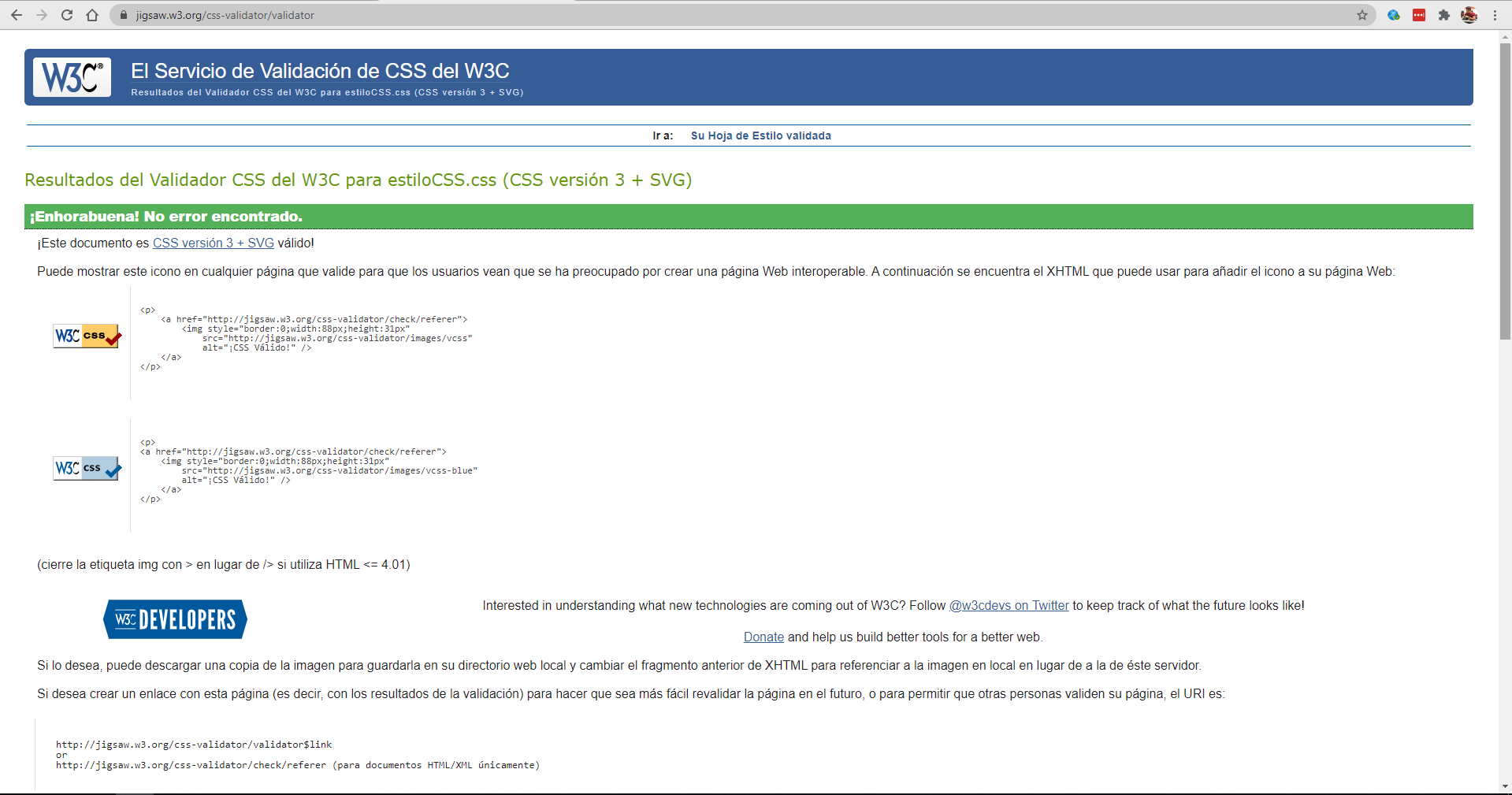
Por otra parte, para comprobar o verificar si nuestro CSS contiene errores y advertencias, podemos hacer uso de CSS Validation Service:



Es una herramienta que nos permite validar nuestra hoja de estilo CSS.

Link: <https://jigsaw.w3.org/css-validator/>

Al validar nuestro documento HTML nos mostró lo siguiente:



Podemos que la validación es exitosa, no tenemos errores y advertencias.

De esta forma, concluimos con las consideraciones para elaborar esta actividad.