TE0101 Tecnologías, lenguajes y entorno de trabajo

Iker Abadia Conejos

11/09/2021

INDICE

# NAVEGADORES

El navegador por defecto de mi sistema operativo, al ser Windows, es Microsoft Edge, así que adicionalmente a este, voy a usar Chrome, por ser el mas usado por los usuarios, y Mozilla Firefox, por ser el que mejor respeta los estándares.

Como ya los tengo instalados en mi equipo, no voy a poner capturas referentes a la instalación de los mismos.

# EDITOR WEB

Como editor web voy a usar VSCode. Es el editor web mas usado por los usuarios, entre otras cosas por la cantidad de comunidad que tiene detrás, aportando de esta forma una cantidad inmensa de plugins al programa, lo que facilita enormemente el desarrollo.

Este editor, además, soporta una gran cantidad de lenguajes diferentes, lo que hace realmente cómodo trabajar con ellos.

# GESTOR DE ESPACIOS DE TRABAJO, PROYECTOS Y CARPETAS.

Como voy a usar VSCode, el propio editor lleva un gestor de espacios de trabajo integrado:

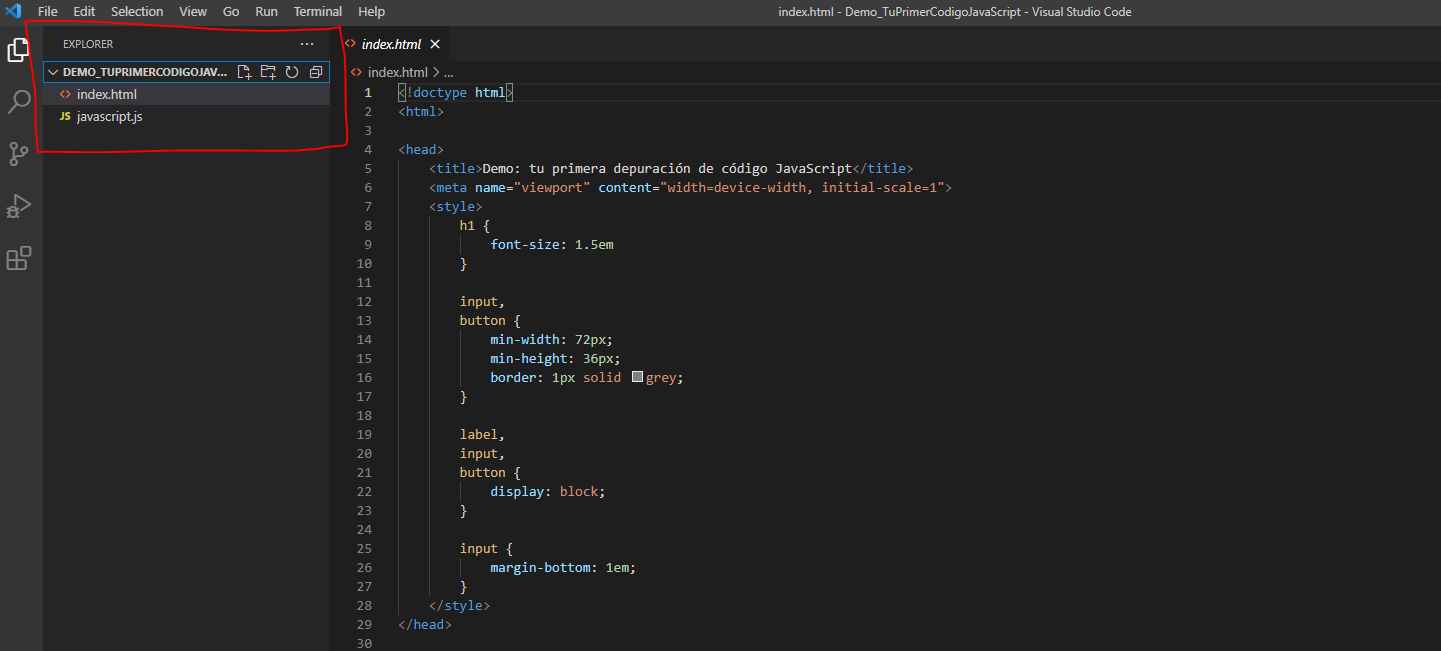


Ilustración 1- Gestor de trabajo integrado de VSCode

Como se puede ver en la imagen, el gestor integrado de VSCode te permite administrar los archivos de tu entorno de trabajo de la forma que desees desde la propia aplicación.

# GESTOR DE VERSIONES Y COPIAS DE SEGURIDAD CON GITHUB.

Este apartado se puede realizar de dos maneras diferentes, o bien con un gestor integrado en vscode, o bien con una aplicación externa a VSCode que te permita administrar tu proyecto en git.

En mi caso me voy a decantar por la segunda opción, usare un programa externo llamado Fork, es un programa de pago, pero ofrece una prueba gratis por tiempo ilimitado.

Una vez con fork instalado, creo un repositorio en guit y lo clono desde fork.

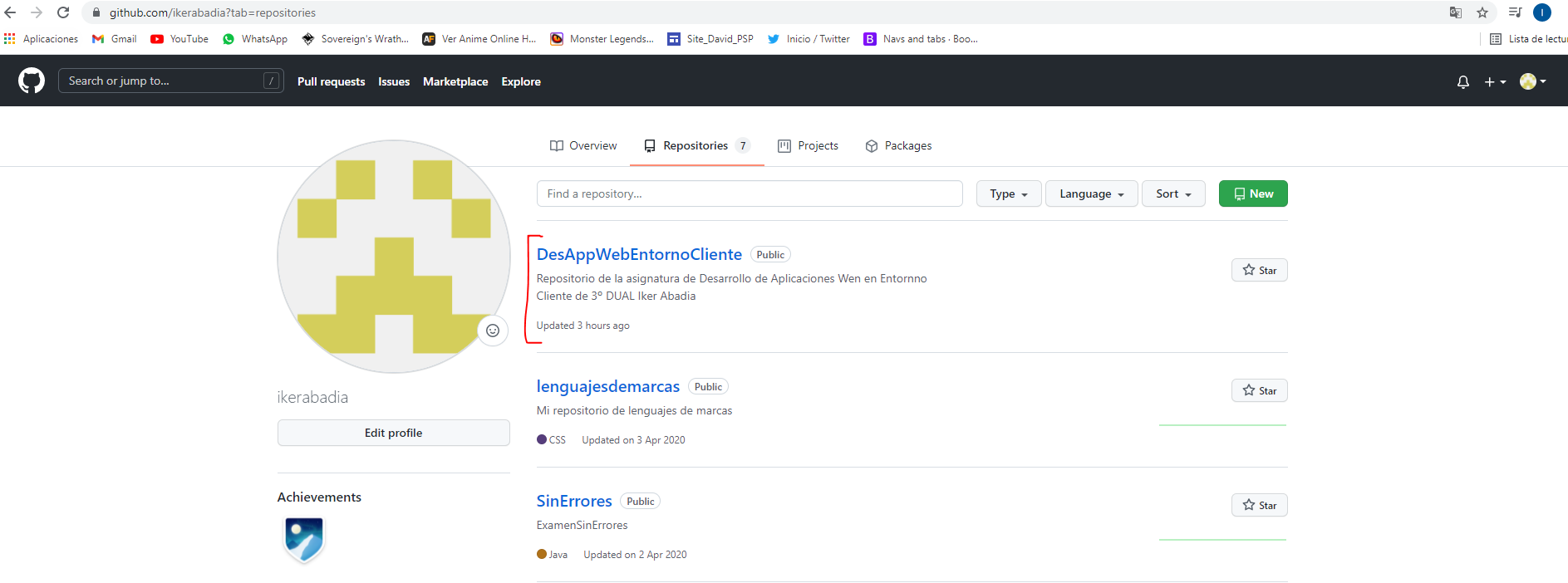


Ilustración 2-Repositorio desde github

En este repositorio voy a incluir la asignatura entera, de forma que todo lo relacionado con la asignatura estará siempre en guithub, así evitare perdidas indeseadas de datos.

Una vez hecho esto voy a hacer un push de lo que tengo hasta ahora en la carpeta de la asignatura.

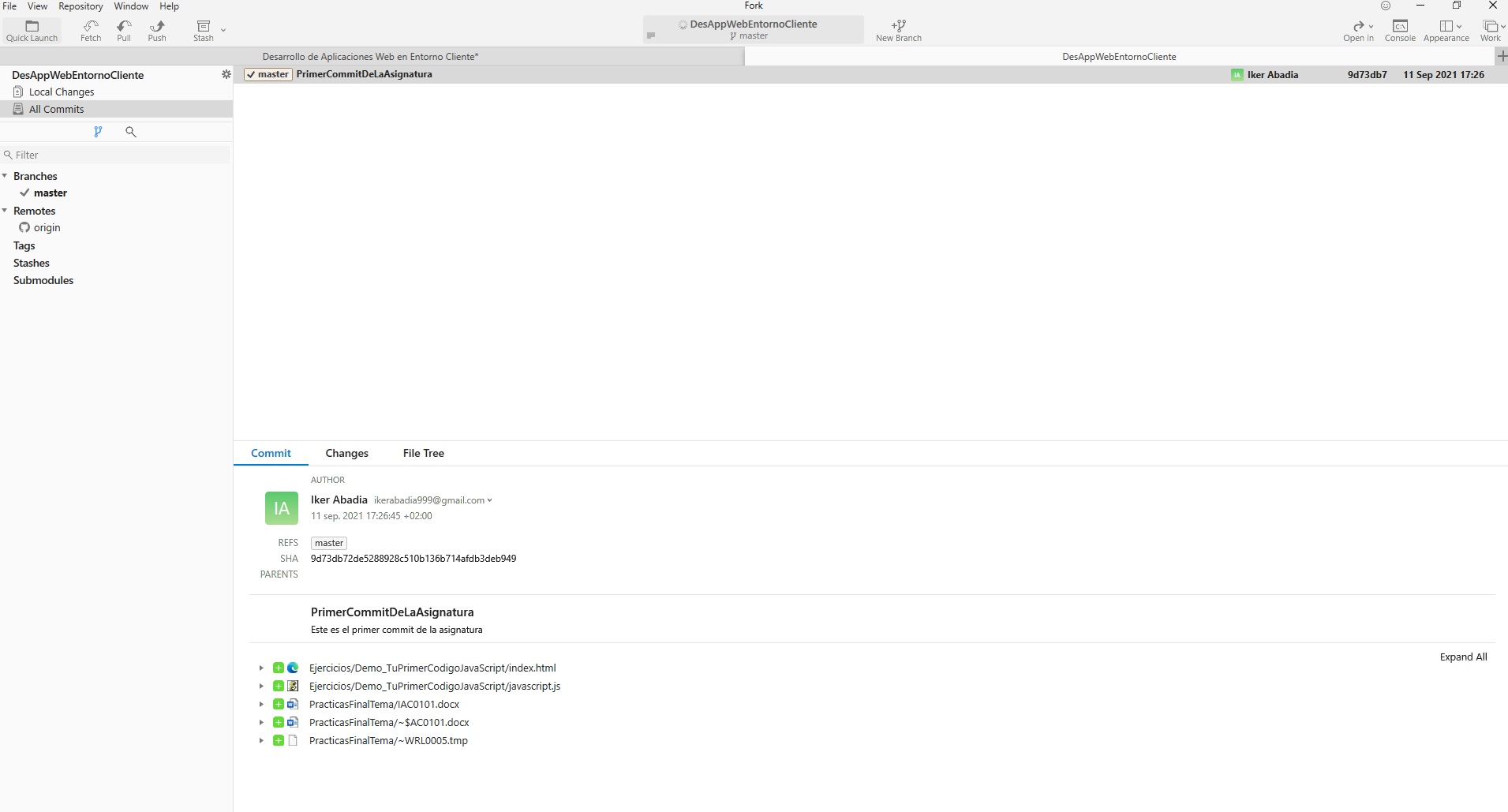


Ilustración 3- Primer commit de la asignatura

Este es el primer commit visto desde fork se pueden ver cada uno de los archivos que fueron commiteados.

# MOSTRAR EN TIEMPO REAL LOS CAMBIOS EN EL HTML Y EL CSS.

Para hacer esto barajábamos varias opciones, por un lado podíamos hacer que los cambios se actualizasen en el navegador externo, sin tener que refrescar, y por otro lado teníamos la opción de incrustar el navegador en el propio VSCode para no tener que estar cambiando de ventanas continuamente, como en clase solo tenemos una pantalla, me decante por la segunda opción ya que me parecía la mas comoda en el caso de que solo tengas una pantalla, pero desafortunadamente no he logrado incrustar Chrome en VSCode, así que he pasado a completar este apartado de la primera forma mencionada.

Para ello voy a instalar el plugin Live Server en VSCode, que permitirá que cada vez que modifique el código, se vea reflejado en el navegador sin necesidad de F5.

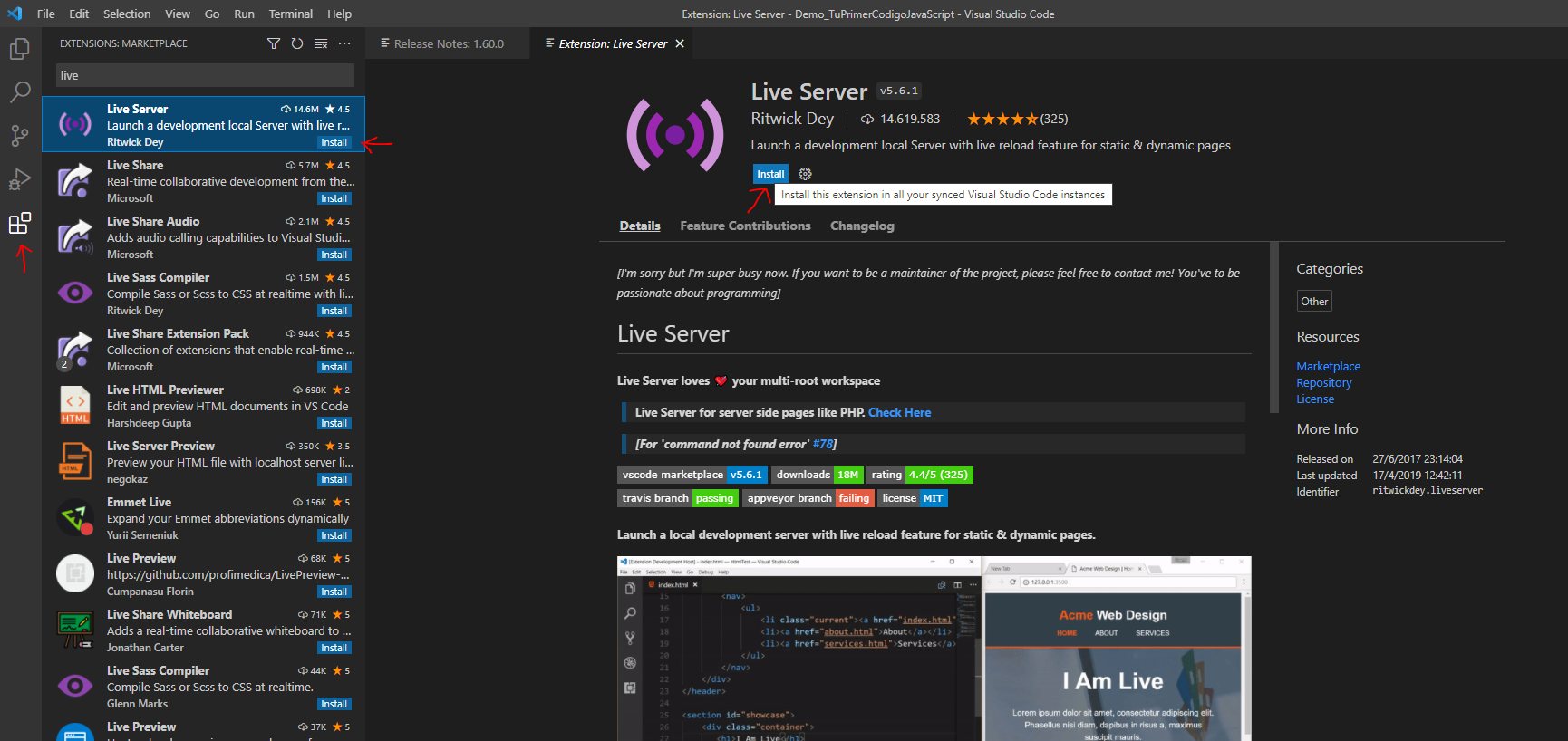


Ilustración 4.Instalacion Live Server

Instalar Live server, es sencillo, como cualquier otro plugin, vas al apartado de instalación de plugins, buscas Live Server, y le das a instalar.

Una vez instalado debemos configurarlo para que cuando lo usemos, funcione con Chrome por defecto.

Para ello nos dirigiremos a “file – preferences - settings – extensions – Live Server Config”

Y cambiaremos el navegador predeterminado a “chrome”.

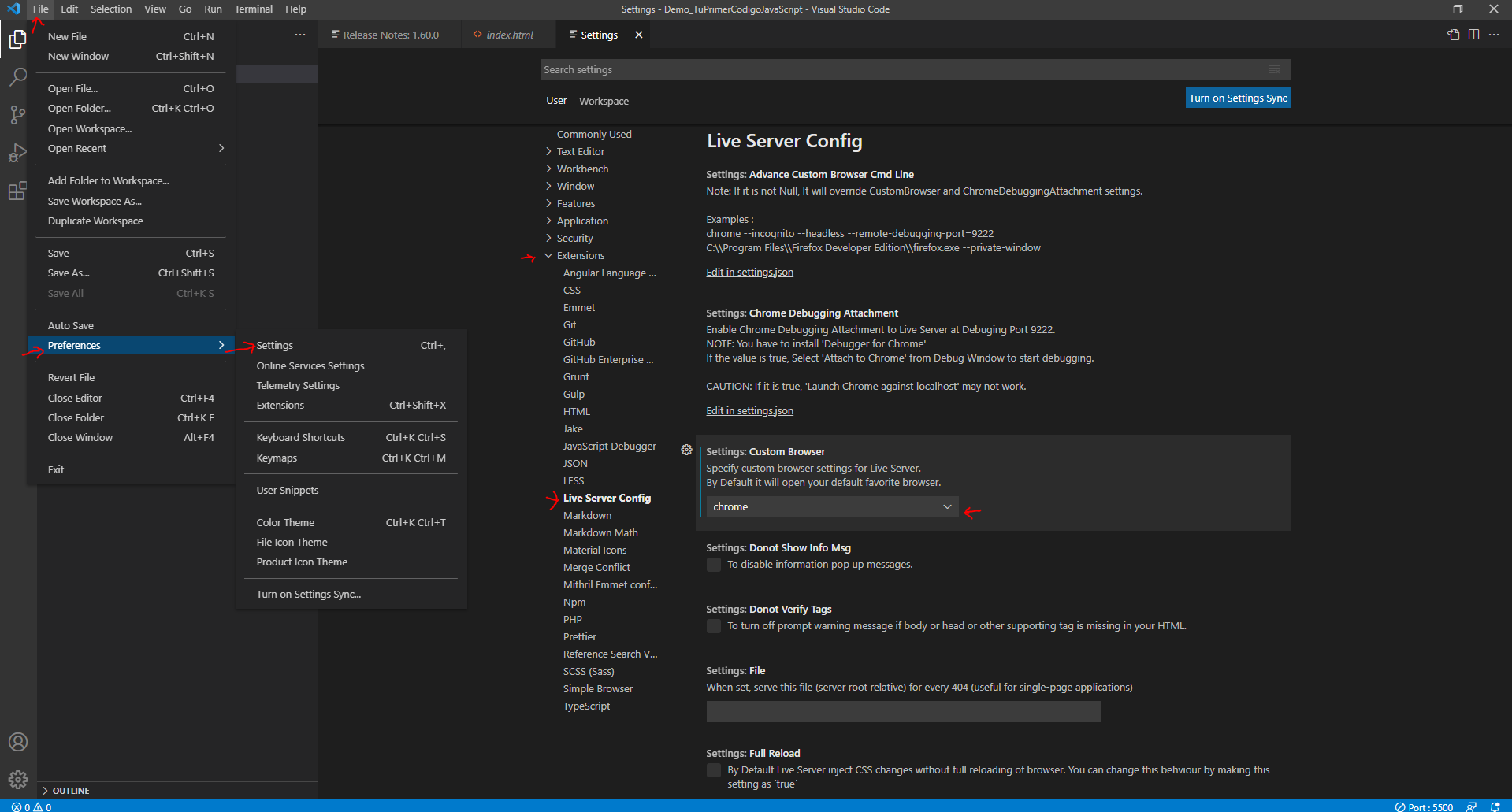


Ilustración 5. Configurador navegador Live Server

Listo ahora cada vez que arranquemos con Live Server un archivo, se arrancara con Chrome.

Ahora, para arrancar Live Server, es tan sencillo como ir al archivo que queremos abrir en el navegador y click derecho “Abrir con live server”

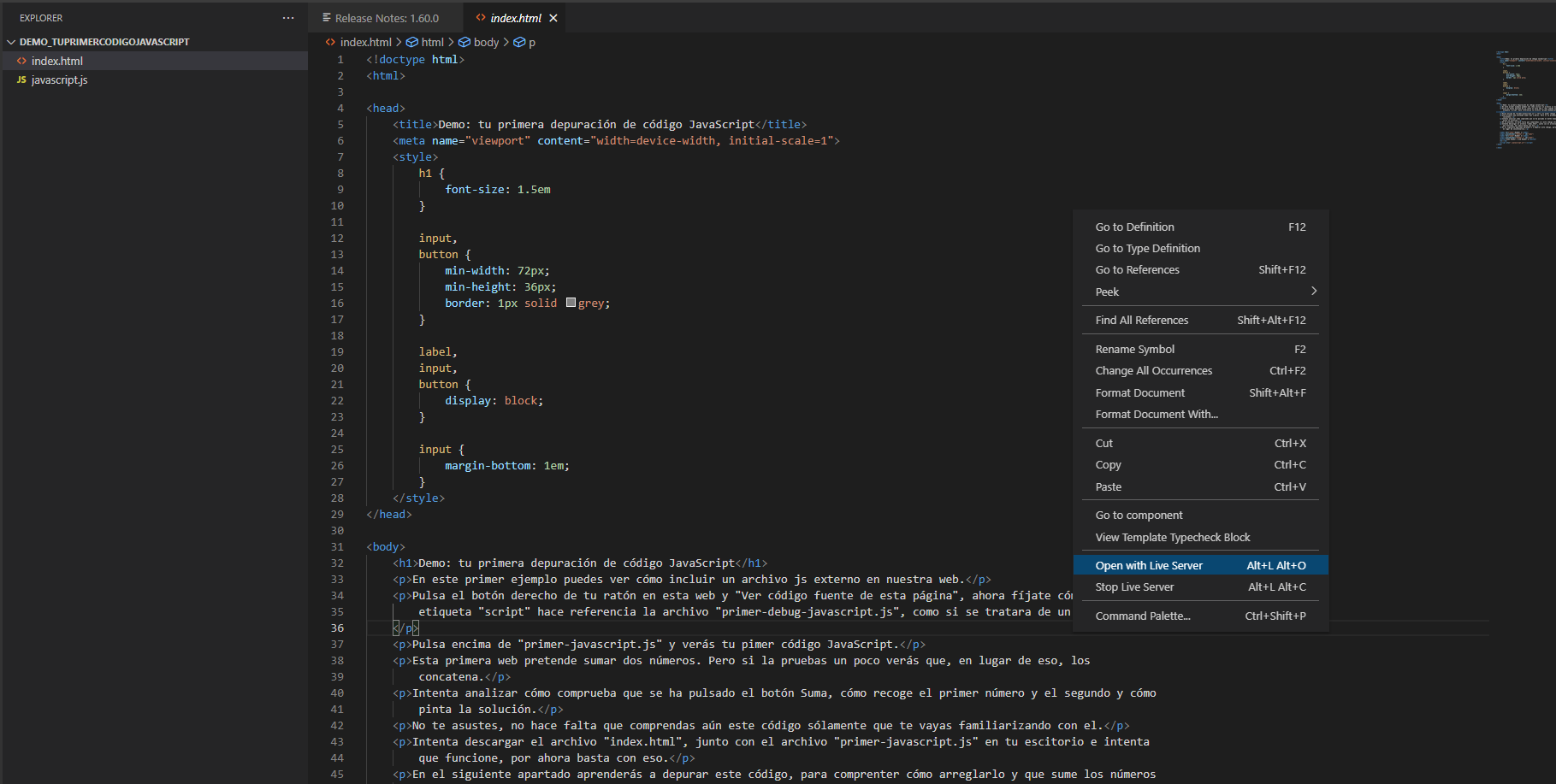


Ilustración 6. Apertura Live Server

Una vez abierto con live server, solo tendrás que guardar el documento y se vera reflejado en el navegador sin tener que actualizar.

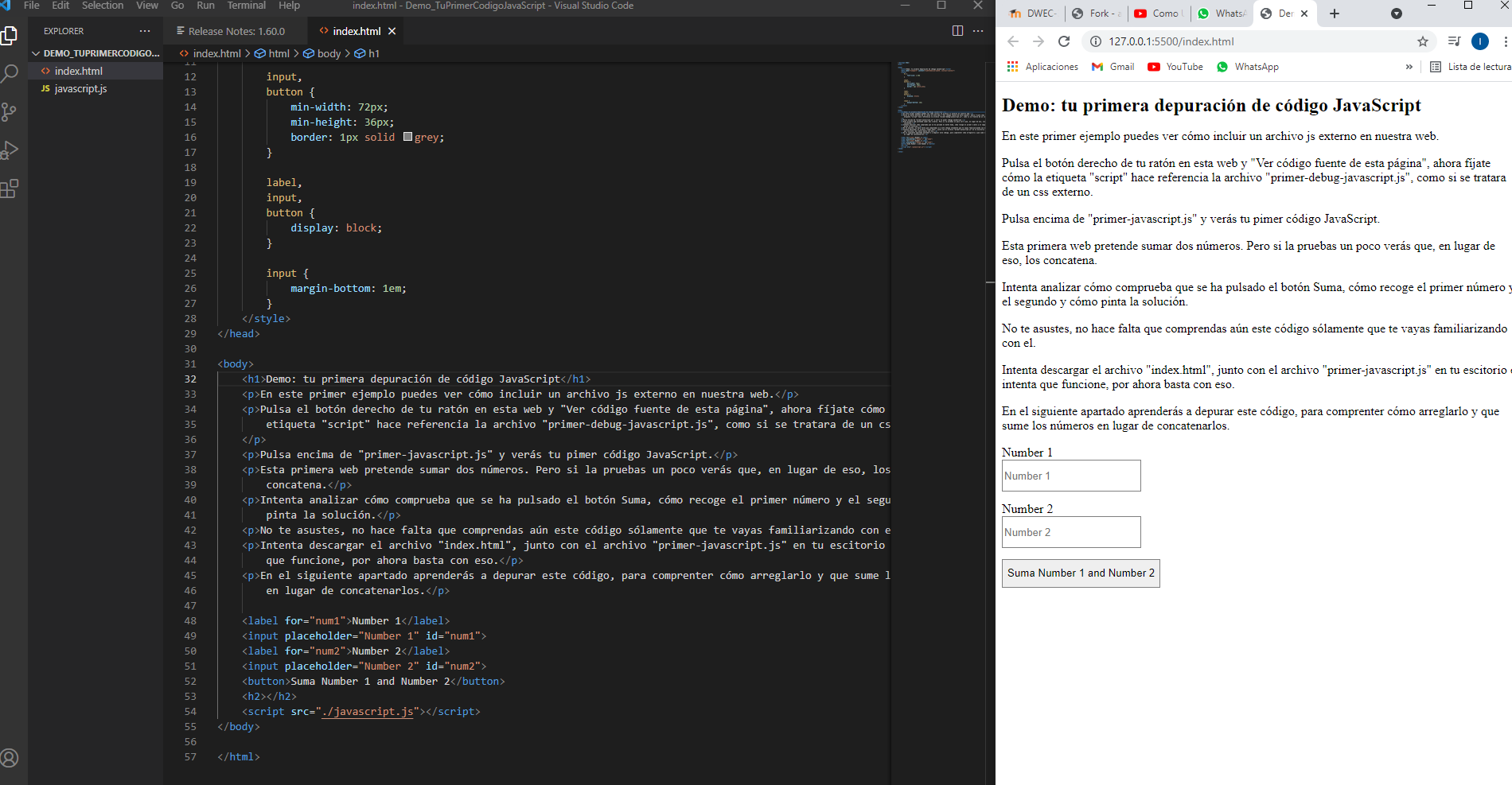


Ilustración 7. Ejemplo de uso de Live Server

Esta imagen es un ejemplo de como lo uso yo, con la pantalla dividida puedo ver ambas cosas a la vez y es casi como si el navegador estuviese incrustado en el VSCode.

# EDITOR DE TEXTO CON MACROS Y AUTOCOMPLETAR CON LOS MÉTODOS DE LAS CLASES DEL API DE JAVASCRIPT. SNIPPETS

Este apartado ya viene integrado en el propio VSCode, tanto las macros como los autocompletados, a continuación, voy a poner un ejemplo de cada uno:

En el caso de las macros, la principal y la que mas uso es la macro para formatear el código -> Alt + Shift + f

Otra macro bastante útil es (Ctrl + K) + (Ctrl + C) para comentar código, y (Ctrl + K) + (Ctrl + U) para descomentarlo.

También puedes crear tus propias macros, pero es algo muy específico, en mi caso nunca lo he usado.

El autocompletado también viene integrado en VSCode, por ejemplo, si escribes “fu” en un archivo .js y presionas Ctrl + Espacio te ofrecerá diferentes opciones de autocompletado:

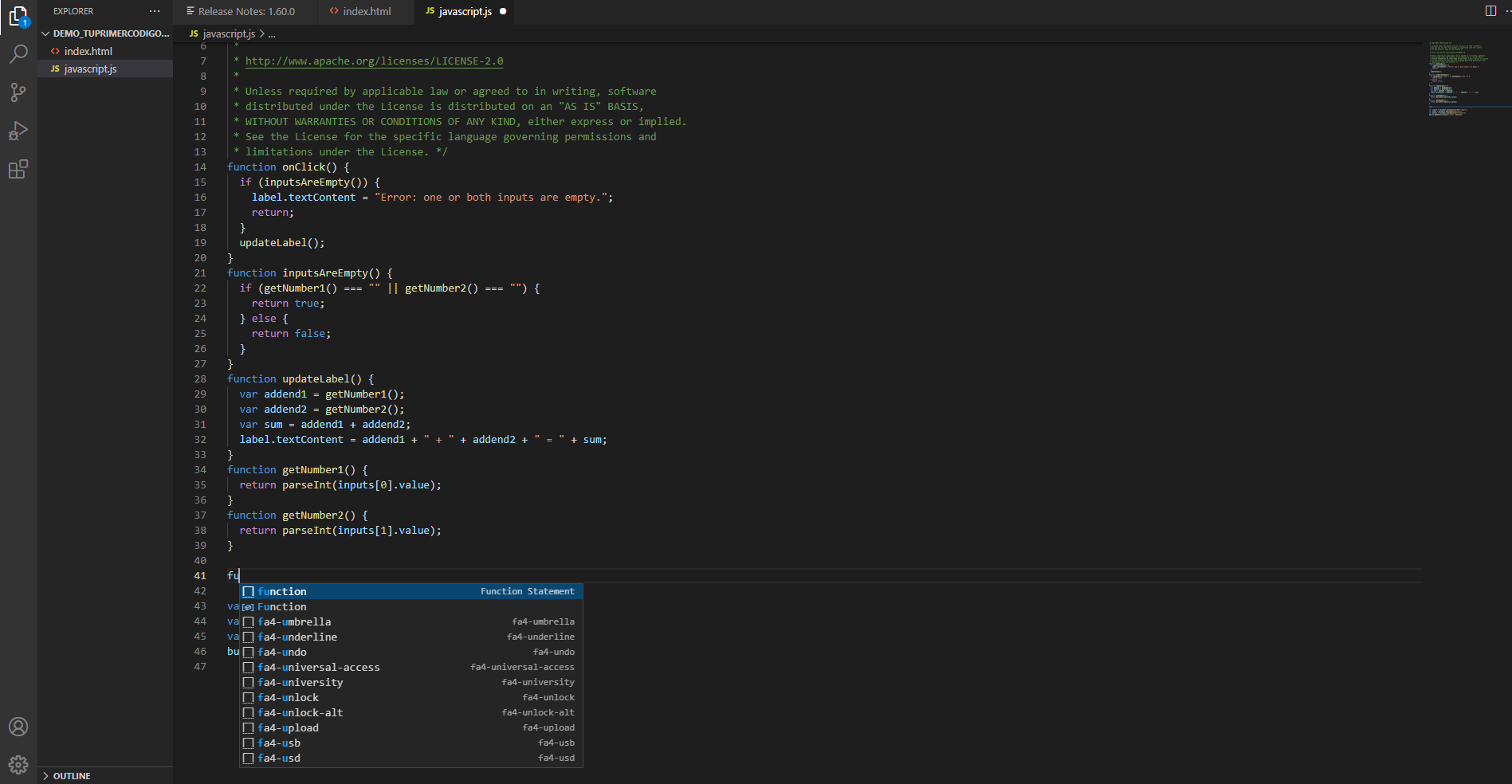


Ilustración . Autocompletado en VSCode

En este caso te ofrece dos opciones para function uno de ellos simplemente te autocompletara la palabra, que es lo que buscamos, esto se vería mas claro con una variable pues autocompletaría su nombre.

La otra opción, lo que hará es generar automáticamente toda la estructura base de una función, esto es conocido como Snippets.

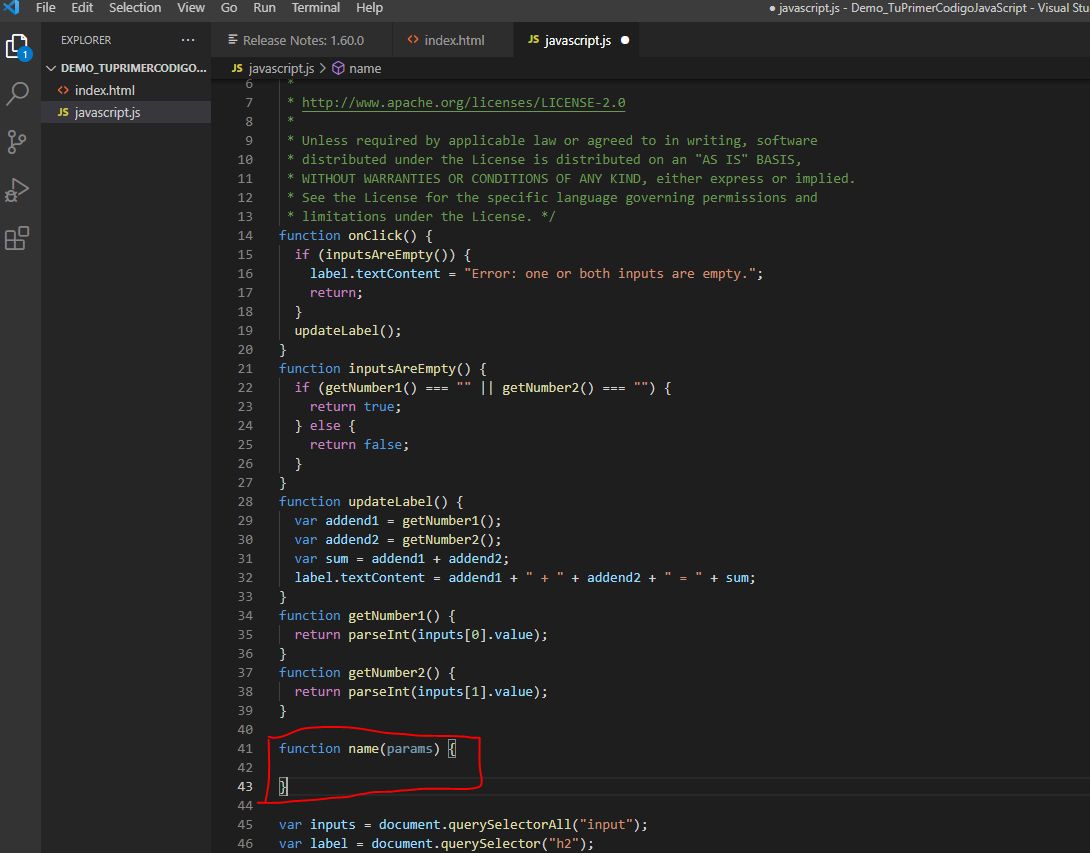


Ilustración . Estructura Función generada mediante Snippets

En el caso de html funciona levemente distinto el tema de los snippets, tendrás que escribir el nombre de la etiqueta, por ejemplo “body”, después pulsa intro y se generará tanto la etiqueta de apertura como la de cierre.

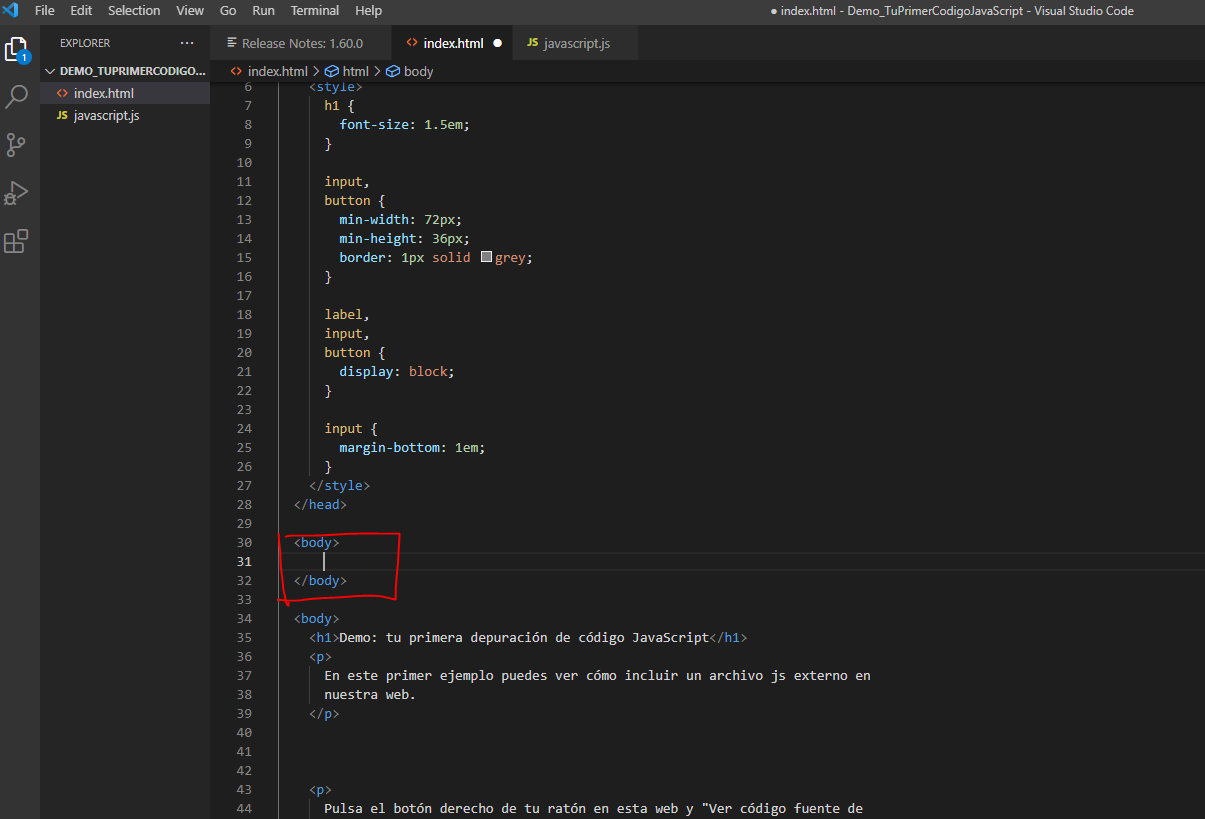


Ilustración . Estructura <body> generada por snippets html

Esta es la estructura body que he generado mediante los snippets html integrados en VSCode.

# CONFIGURACIÓN DE PARÁMETROS DEL NAVEGADOR: RESPONSIVE, MÓVIL, GEOLOCALIZACIÓN...

Todas estas configuraciones pertenecen al navegador, y son fáciles de configurar a través de la interfaz de la tecla F12:

* En el caso de responsive y móvil utilizaremos el siguiente botón de la interfaz del F12 que mostrara la siguiente interfaz:

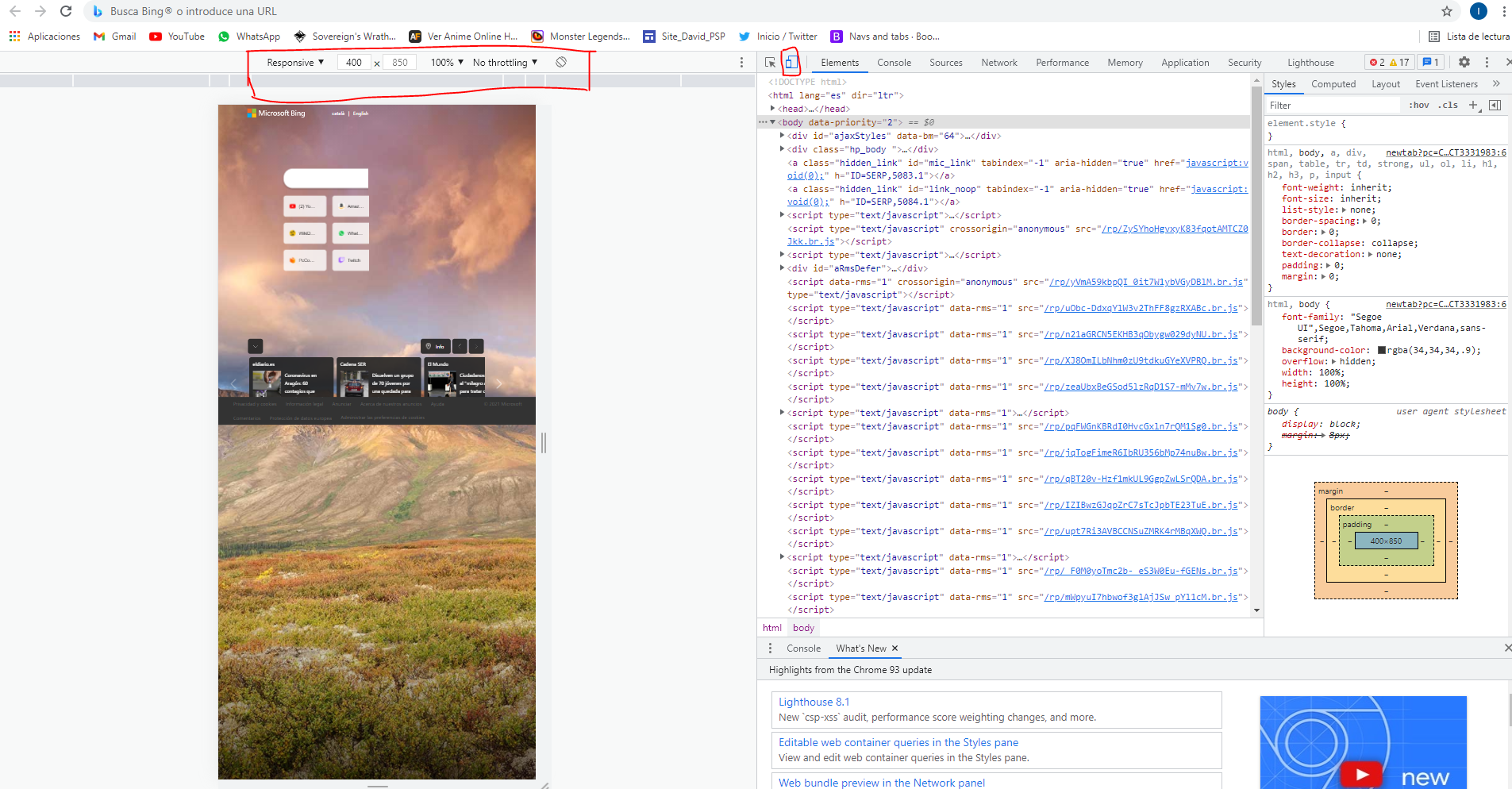


Ilustración . Interfaz responsive y movil del F12

Como se puede ver, esta interfaz te permite como desarrollador modificar las dimensiones de la web a placer para ajustarte a aquel dispositivo objetivo en el que desarrollas y comprobar que tu web es responsive.

El desplegable en el que pone “Responsive” permite seleccionar un dispositivo concreto para simular que la web se esta lanzando en dicho dispositivo, por ejemplo, un Samsung Galaxy S5.

El botón que se encuentra mas a la derecha te permite cambiar la orientación vertical / horizontal.

# DEPURACION

Para depurar js usaremos el propio navegador. Iremos al menú del F12 mencionado previamente y a sources. Seleccionaremos el archivo javascript y pondremos el punto de ruptura donde queramos y ya podremos debuggar este archivo.

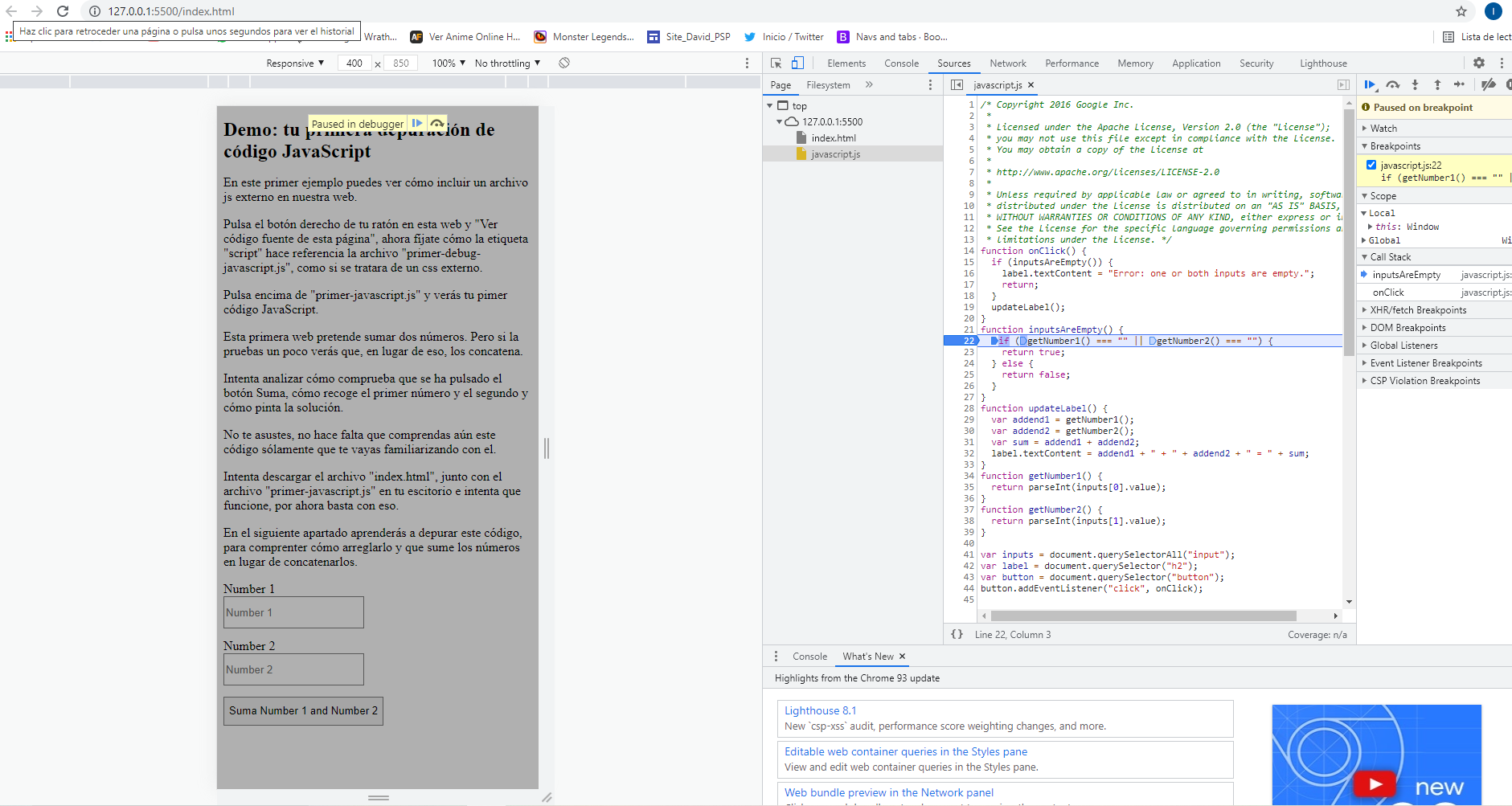


Ilustración . Menu de depuracion en Chrome

Ahora mismo en la imagen se ve que esta parado en el punto de ruptura.

Te permite avanzar línea a línea tanto entrando en cada método como saltando sin entrar en él.

Además te permite ver el valor de las variables, si las hubiere.