

EJERCICIOS JS

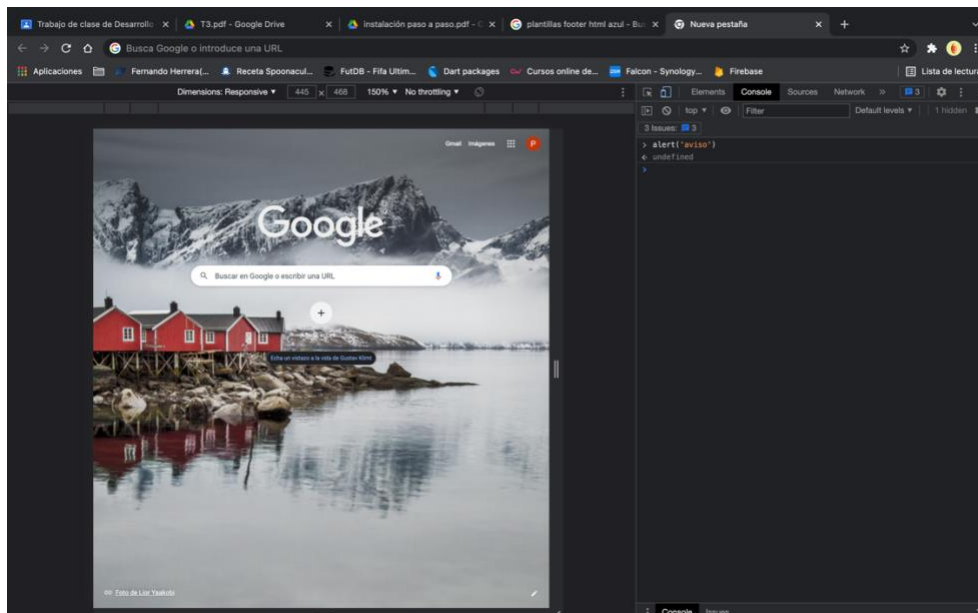
1.- ¿Qué es un transpilador? Indica algunos ejemplos.

Un transpilador es un tipo de compilador, cuya función es traducir de un lenguaje fuente a otra fuente. Sin embargo, se diferencia de los compiladores tradicionales en que ellos reciben ficheros con contenido código fuente y generan el código en una máquina de bajo nivel.

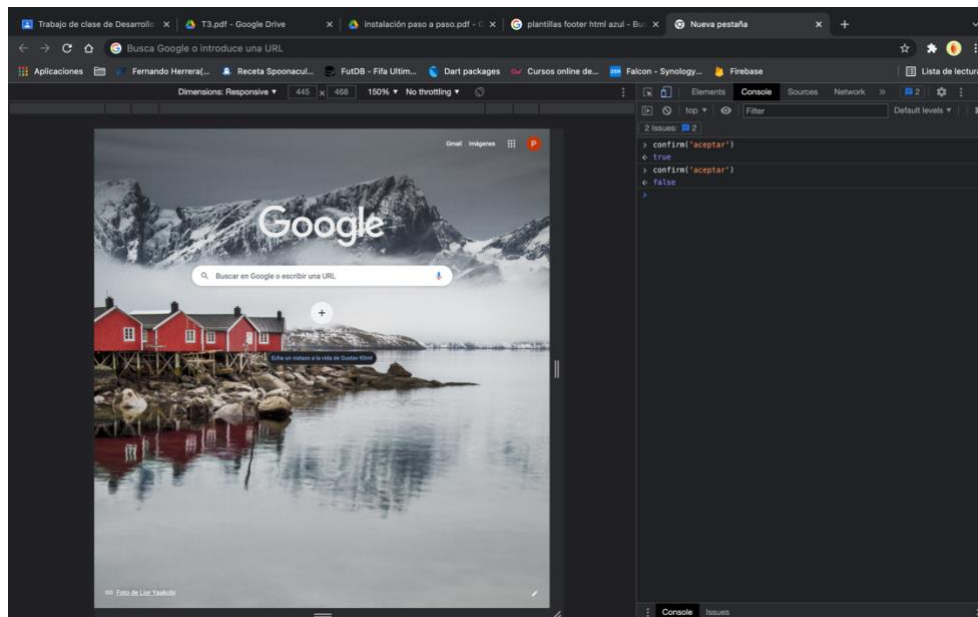
Ejemplo: **Babel** y **Tranceur** (Compilador de las últimas sintaxis de JS).

2.- Abre la consola del navegador y prueba las funciones alert, confirm y prompt. Indicando el valor que devuelve cada una en todas sus posibles interacciones con el usuario (escribe algo, pulsa aceptar, pulsa cancelar...)

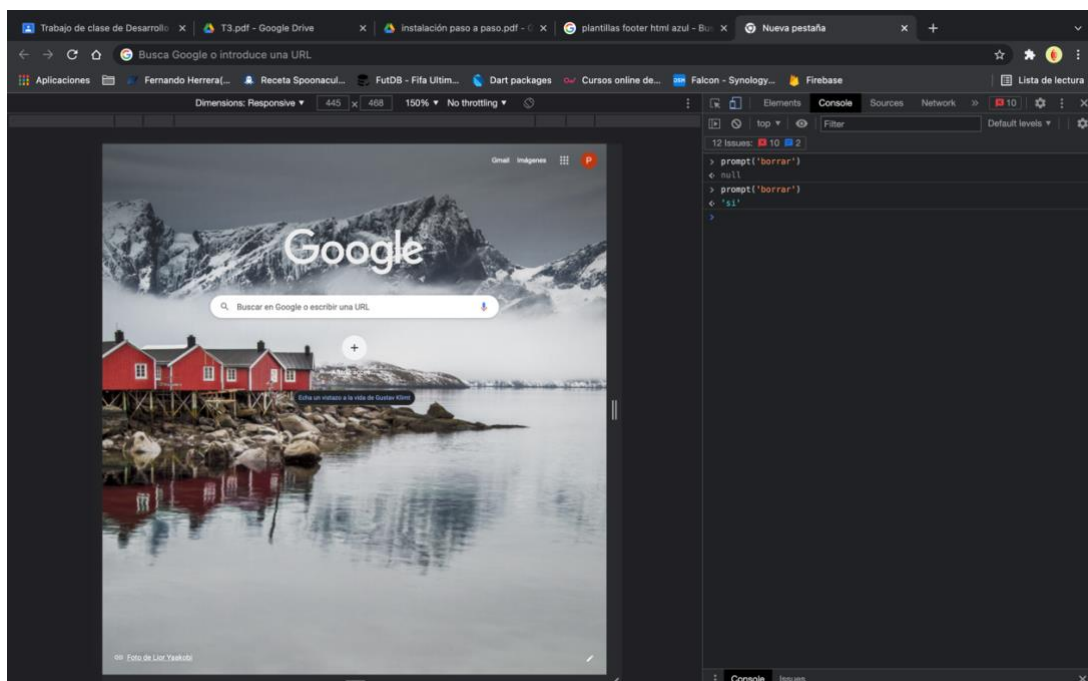
La **función alert** no devuelve nada.



La **función confirm** devuelve **true** si pulsamos el botón de aceptar y **false** si pulsamos el de cancelar.

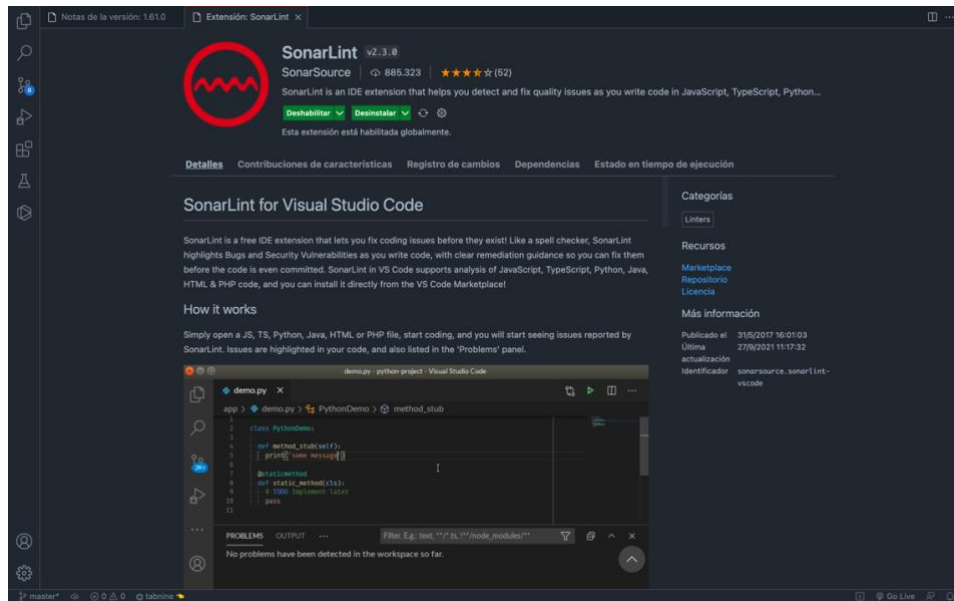


La función **prompt** devuelve null si no escribimos nada en la caja de texto y en cambio si escribimos algo en la caja de texto nos devuelve lo que hemos escrito.

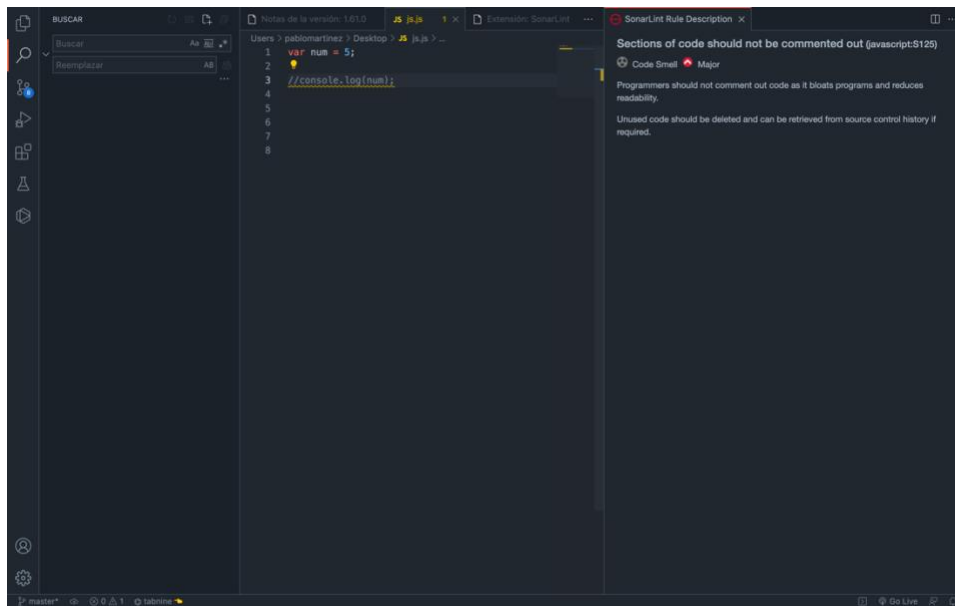


3.- Instala el editor VS Code o actualízalo a su última versión.

4.- Un linter es una herramienta que nos ayuda a seguir las buenas prácticas o guías de estilo de nuestro código fuente. Añade a VS Code la extensión SonarLint, indicando cuáles son sus principales funciones y cómo se utiliza.

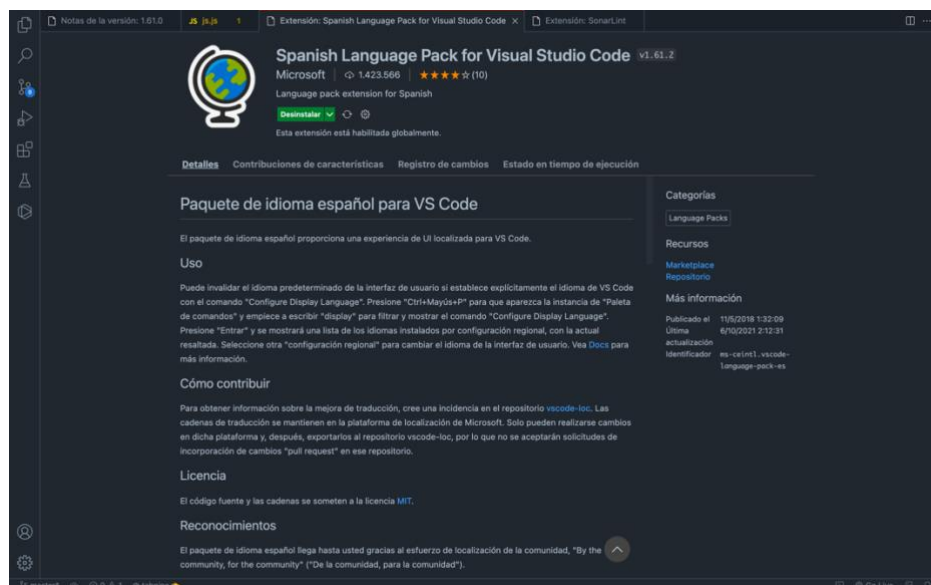


SonarLint es una extensión para optimizar mejor tu código fuente, sus principales características son darte como unas pequeñas ayudas o mejores a tiempo real de como puedes mejorar tu código en diferentes partes donde lo detecte. Es decir, SonarLint busca en tu código fuente las opciones que lo puedes mejorar con unas pequeñas a notaciones.

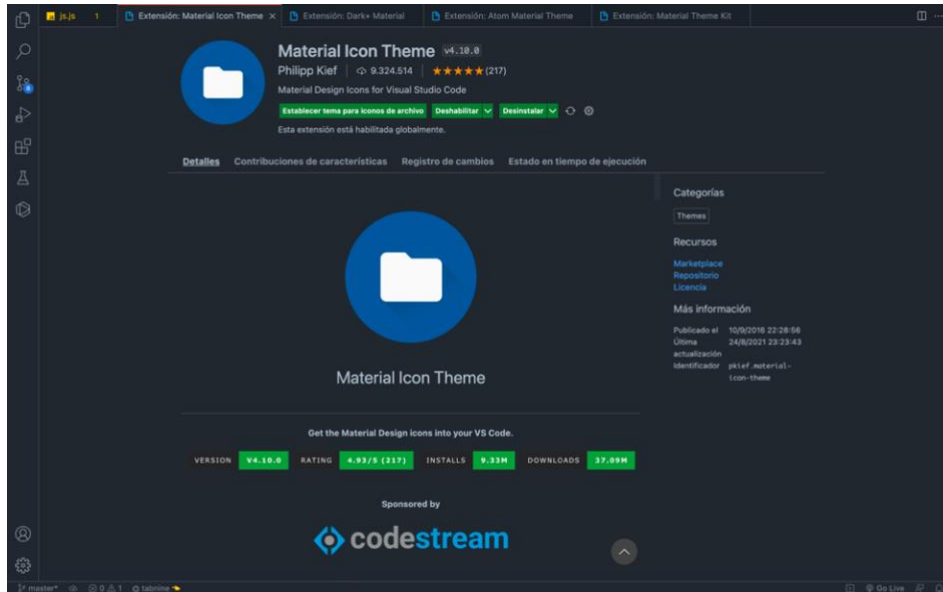


Por ejemplo, he comentado console log y la advertencia o a notación que me dice SonarLint es que no comentemos el código, ya que el código que no vamos a utilizar lo podemos borrar o recuperar de un control de versiones.

5.- Instala en VS Code la extensión Spanish Language Pack for Visual Studio Code, para tener la interfaz disponible en español.



6.- Instala en VS Code la extensión Material Theme Icons, mediante el cual tendremos disponible un nuevo tema de iconos de archivo más descriptivo.

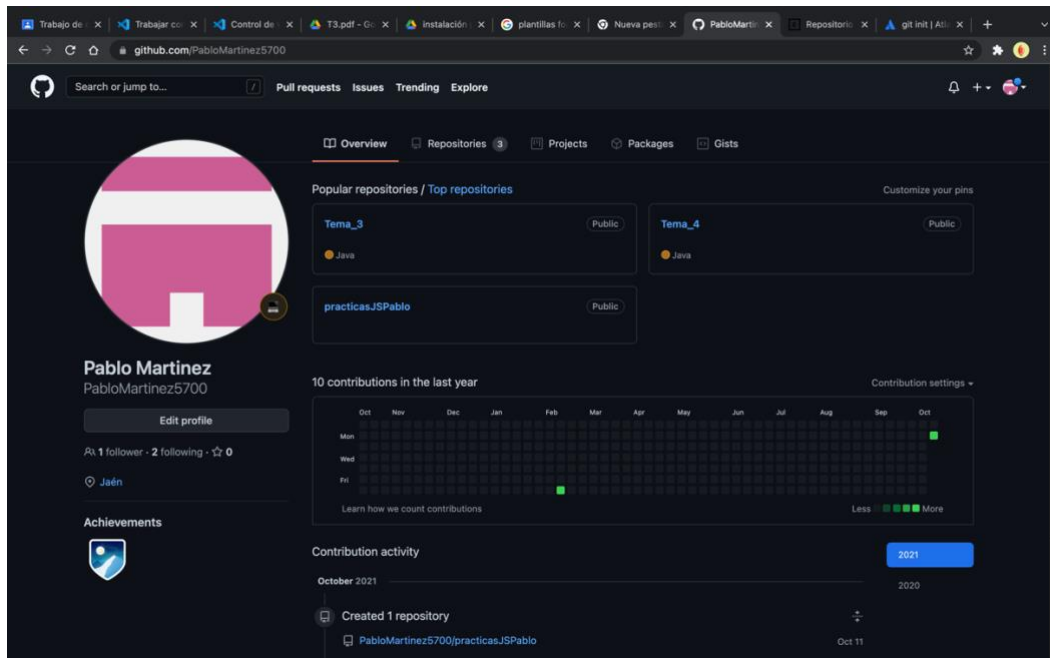


7.- Necesitaremos el framework Node.js y su gestor de paquetes npm. Instala Node.js en tu sistema. La herramienta npm se encargará de descargar y poner a disposición de nuestra aplicación todas las librerías JS que vayamos a utilizar. Comprueba desde consola (en VS Code, menú Ver – Terminal) la versión instalada tanto de node como de npm.

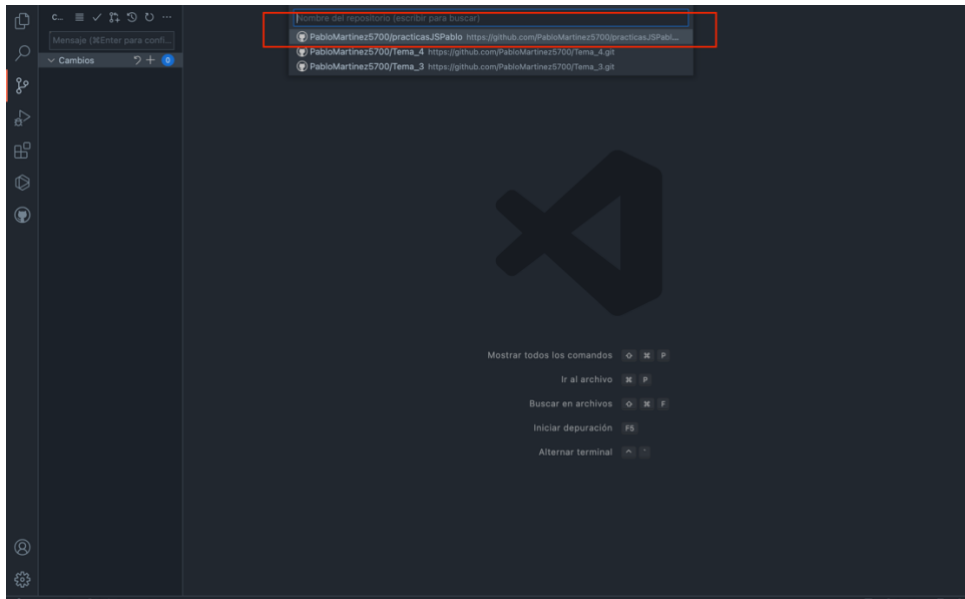
```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL
MacBook-Pro:Users pablmartinez$ node -v
v16.10.0
MacBook-Pro:Users pablmartinez$ npm -v
7.24.0
MacBook-Pro:Users pablmartinez$
```

8.- Instala en VS Code una extensión para gestionar el control de versiones con git. Crea un repositorio en GitHub llamado practicasJSNombre, clónalo en VS Code, crea un archivo de prueba en tu repositorio local y sube los cambios al repositorio de GitHub.

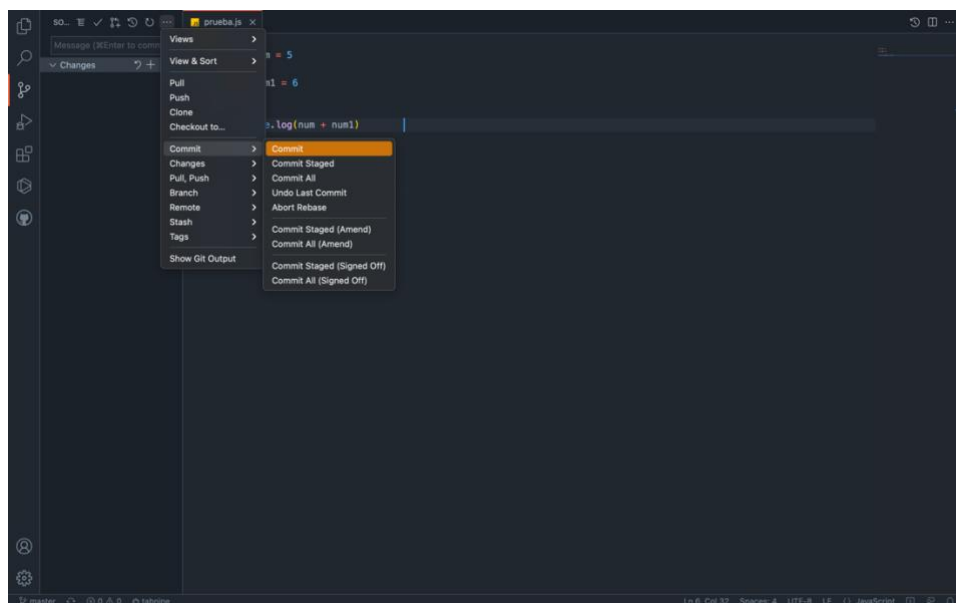
Creado el repositorio

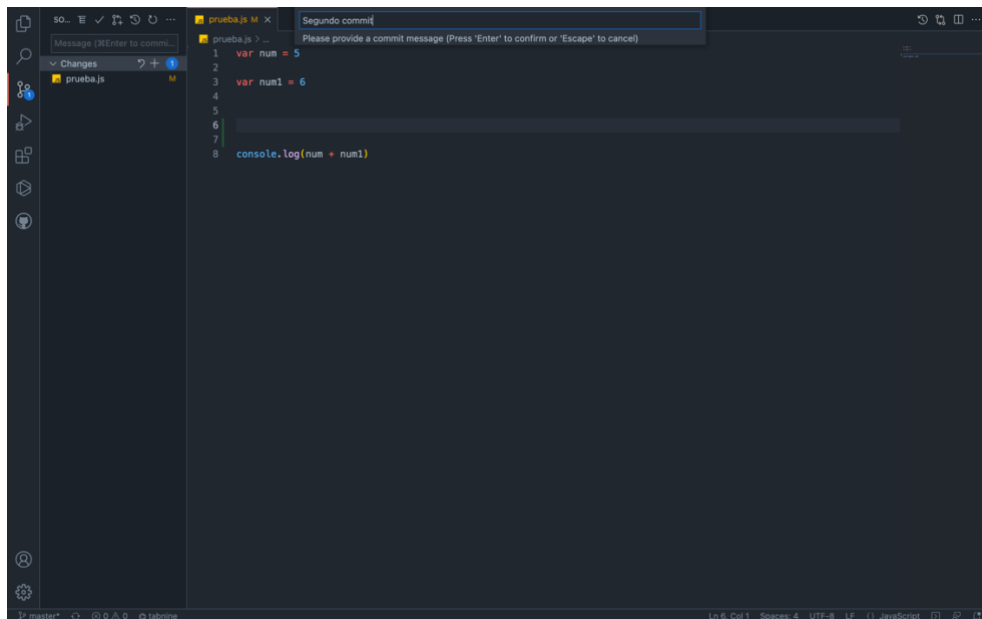


Clonamos el repositorio

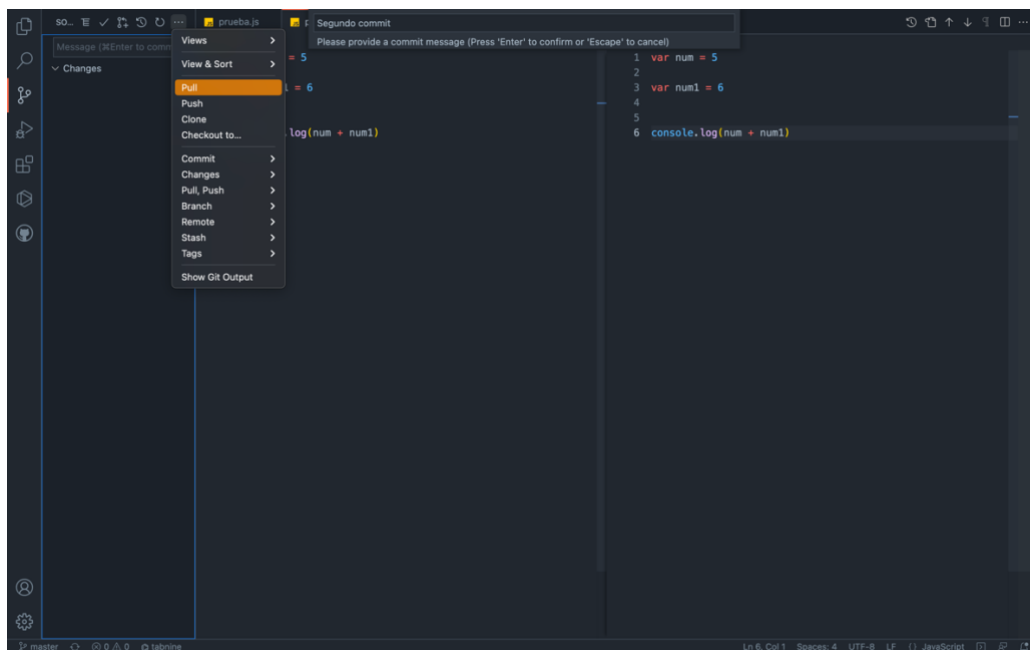


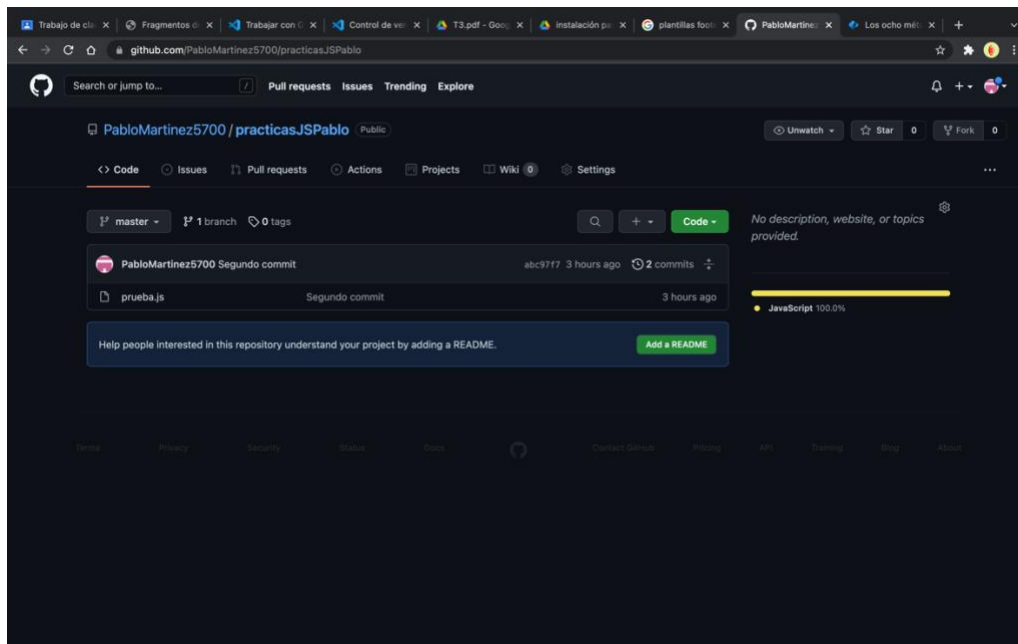
Una vez clonado el repositorio, hacemos un commit para guardar el estado actual.





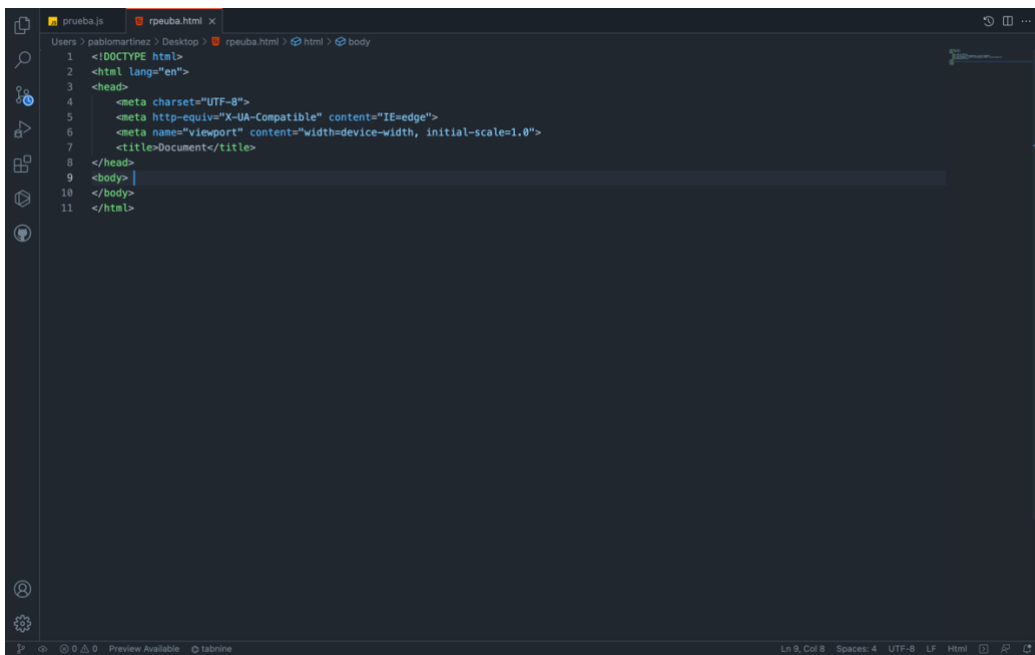
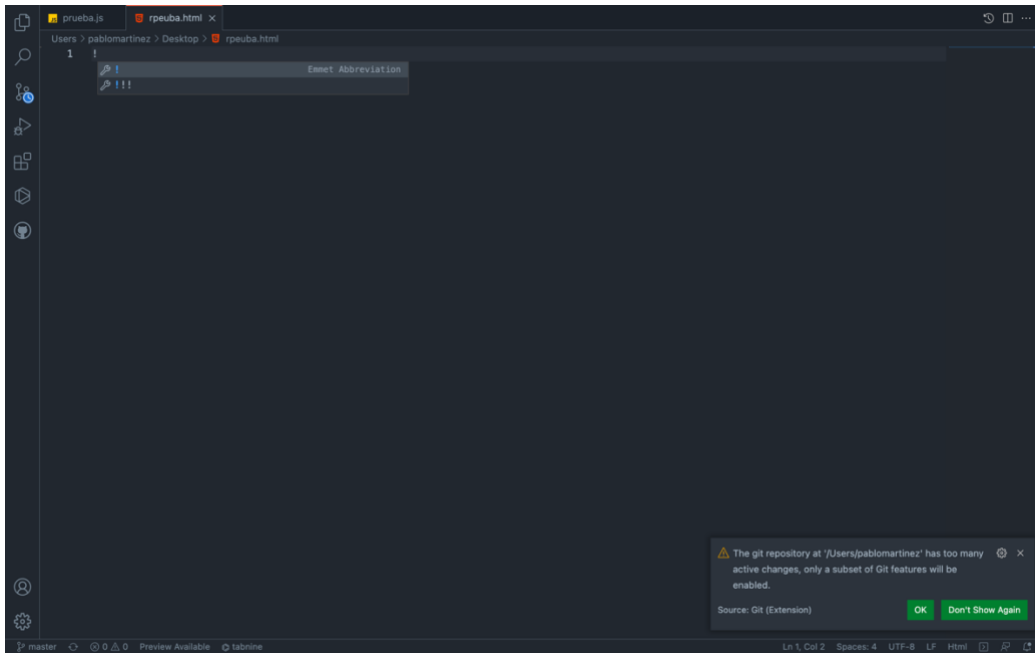
Finalmente, hacemos un Pull para subirlo a nuestro repositorio remoto.





9.- Prueba el complemento Emmet (integrado en VS Code) para agilizar el desarrollo web. Info básica y uso de expresiones compuestas. Escribe las expresiones necesarias para crear una lista de 10 elementos y una tabla de 8 filas y 5 columnas. Prueba a añadir todos los atributos que puedas, como clases e ids.

Cabecera

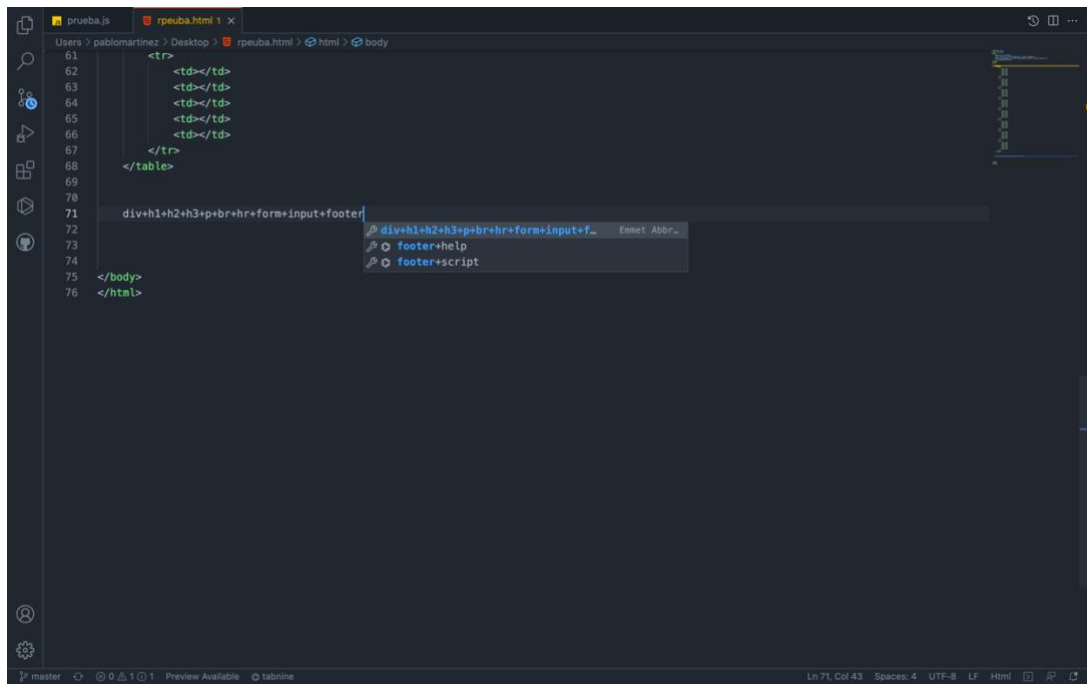


Tablas de 8 filas y 5 columnas

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10   <table>tr<td>5
11   <table>tr<td>5 Emmet Abbreviation
12 </body>
13 </html>
```

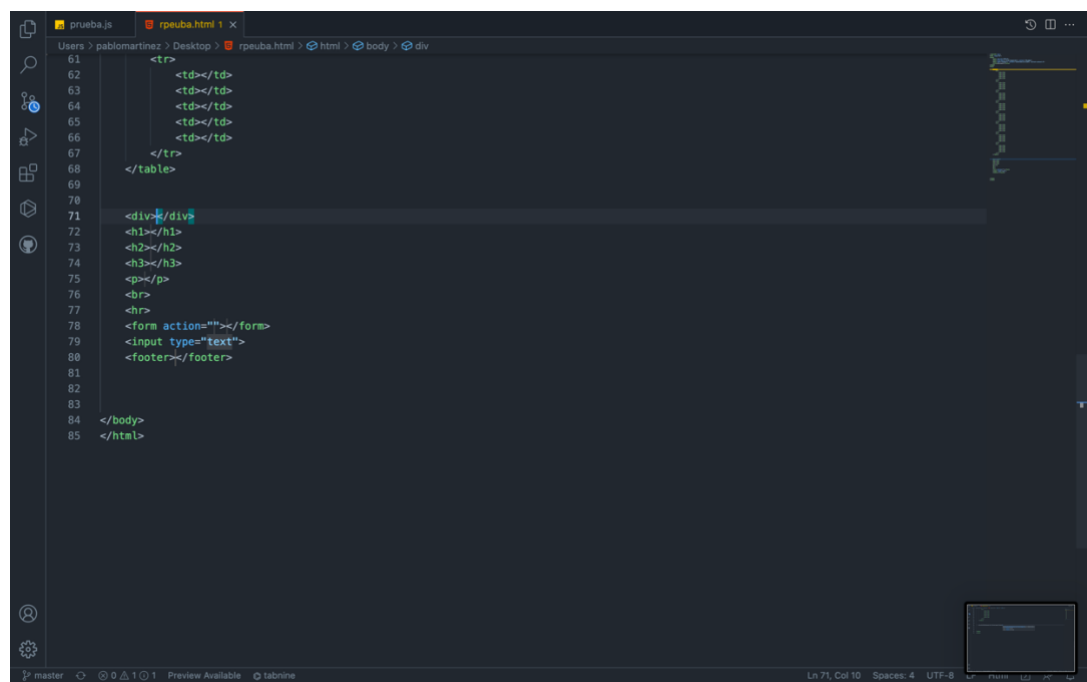
```
11 <table>
12   <tr>
13     <td></td>
14     <td></td>
15     <td></td>
16     <td></td>
17     <td></td>
18   </tr>
19   <tr>
20     <td></td>
21     <td></td>
22     <td></td>
23     <td></td>
24     <td></td>
25   </tr>
26   <tr>
27     <td></td>
28     <td></td>
29     <td></td>
30     <td></td>
31     <td></td>
32   </tr>
33   <tr>
34     <td></td>
35     <td></td>
36     <td></td>
37     <td></td>
38     <td></td>
39   </tr>
40   <tr>
41     <td></td>
42     <td></td>
43     <td></td>
44     <td></td>
45     <td></td>
46   </tr>
47   <tr>
48     <td></td>
49     <td></td>
```

Elementos



The screenshot shows a VS Code editor window with a file named `rpeuba.html`. The code is in HTML and includes a table structure. At line 71, the text `div+h1+h2+h3+p+br+hr+form+input+footer` is entered. An Emmet snippet suggestion box appears, showing options like `div+h1+h2+h3+p+br+hr+form+input+f...`, `footer+help`, and `footer+script`. The status bar at the bottom indicates the cursor is at line 71, column 43.

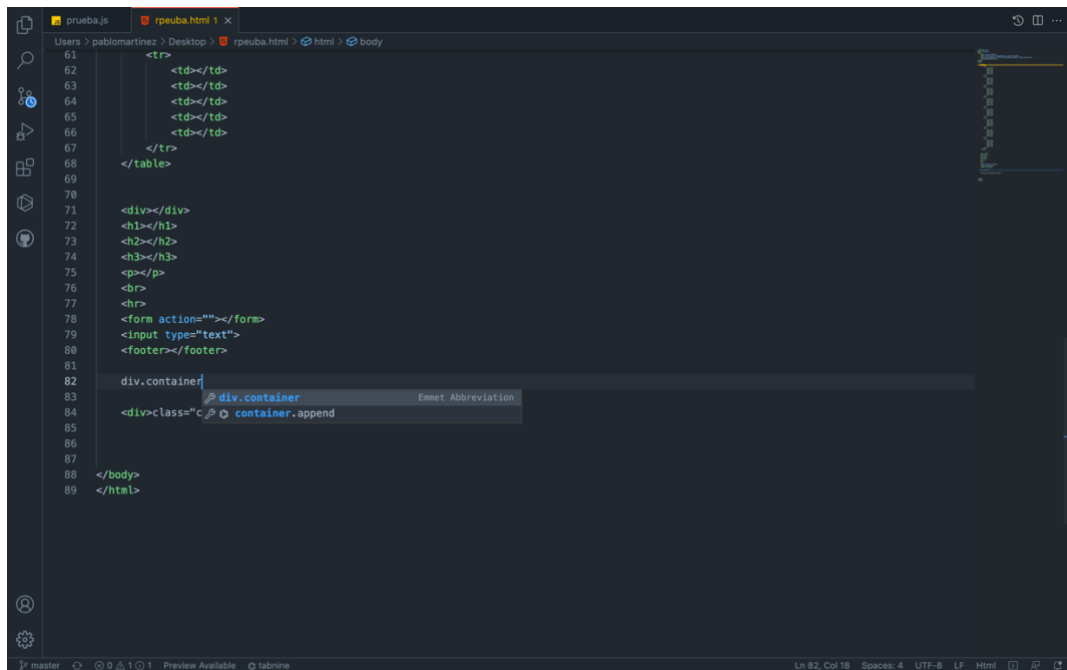
```
61 <tr>
62   <td></td>
63   <td></td>
64   <td></td>
65   <td></td>
66   <td></td>
67 </tr>
68 </table>
69
70
71 div+h1+h2+h3+p+br+hr+form+input+footer
72
73
74
75 </body>
76 </html>
```



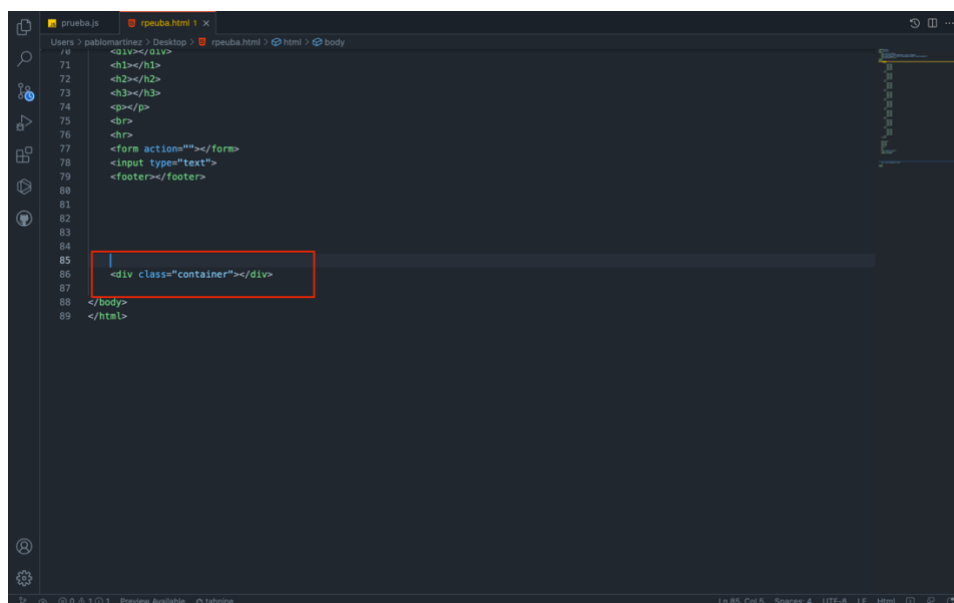
The screenshot shows the same VS Code editor window, but now the HTML code is complete. The `div` element at line 71 has been expanded into a full HTML structure including `h1`, `h2`, `h3`, `p`, `br`, `hr`, `form`, `input`, and `footer` tags. The status bar at the bottom indicates the cursor is at line 71, column 10.

```
61 <tr>
62   <td></td>
63   <td></td>
64   <td></td>
65   <td></td>
66   <td></td>
67 </tr>
68 </table>
69
70
71 <div></div>
72 <h1></h1>
73 <h2></h2>
74 <h3></h3>
75 <p></p>
76 <br>
77 <hr>
78 <form action=""></form>
79 <input type="text">
80 <footer></footer>
81
82
83
84 </body>
85 </html>
```

Div con una clase

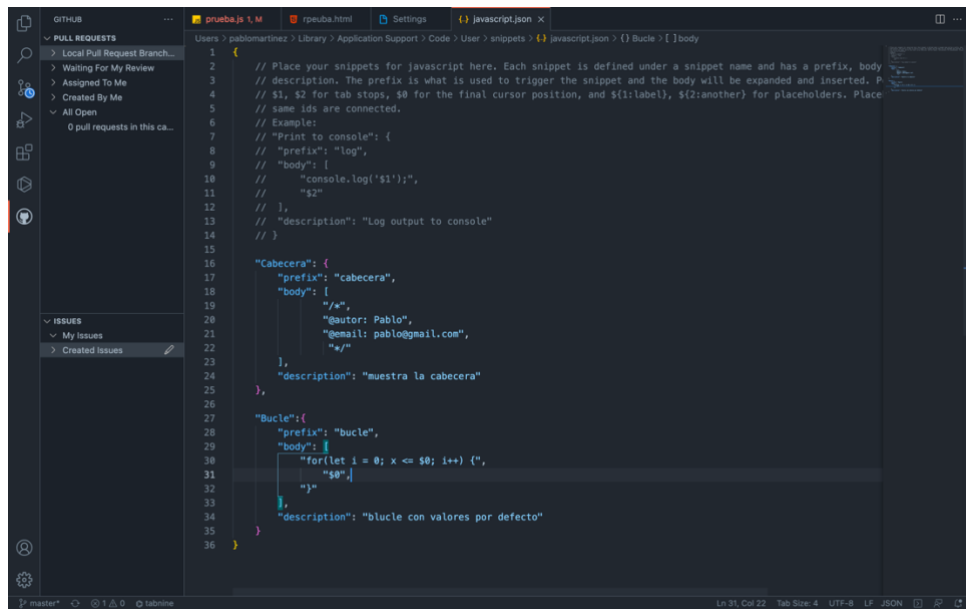


The screenshot shows the VS Code editor with a file named `rpeuba.html` open. The code is in HTML format and includes a table, several heading tags (`<h1>`, `<h2>`, `<h3>`), a paragraph, a form, and a footer. At line 82, the text `div.container` is entered, and an Emmet abbreviation suggestion box appears with the options `div.container` and `<div>class="container"></div>`. The status bar at the bottom indicates the cursor is at line 82, column 18.



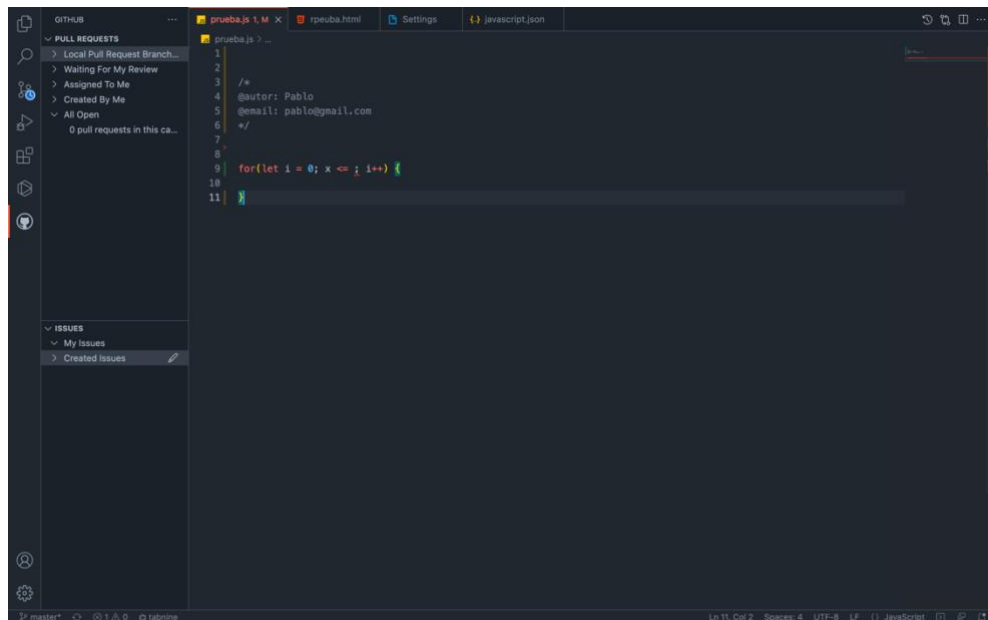
The screenshot shows the same VS Code editor with the `rpeuba.html` file. The code is now complete, and the `<div class="container"></div>` tag has been added at line 85. The status bar at the bottom indicates the cursor is at line 85, column 5.

10.- Crea 2 fragmentos de código personalizados en VS Code. INFO El primero será la típica cabecera comentada donde se indica el autor (tú), correo, web, licencia del código... Se debe lanzar al escribir el código “cabecera”. El segundo, será un bucle con valores por defecto, los que quieras. Se debe lanzar al escribir el código “bucle”.



The screenshot shows the VS Code editor with a file named 'javascript.json' open. The file contains a JSON array of JavaScript snippets. The first snippet is named 'Print to console' and has a prefix 'log'. The second snippet is named 'Cabecera' and has a prefix 'cabecera'. The third snippet is named 'Bucle' and has a prefix 'bucle'. The snippets are defined as follows:

```
1 {
2   // Place your snippets for javascript here. Each snippet is defined under a snippet name and has a prefix, body
3   // description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. P
4   // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label}, ${2:another} for placeholders. Place
5   // same ids are connected.
6   // Example:
7   // "Print to console": {
8   //   "prefix": "log",
9   //   "body": [
10    //     "console.log('$1');",
11    //     "$2"
12    // ],
13    // "description": "Log output to console"
14  // }
15
16  "Cabecera": {
17    "prefix": "cabecera",
18    "body": [
19      /*
20       * @autor: Pablo
21       * @email: pablo@gmail.com
22       */
23    ],
24    "description": "muestra la cabecera"
25  },
26
27  "Bucle": {
28    "prefix": "bucle",
29    "body": [
30      "for(let i = 0; i <= $0; i++) {",
31      "  \"$0\"",
32      "}"
33    ],
34    "description": "bucle con valores por defecto"
35  }
36 }
```



The screenshot shows the VS Code editor with a file named 'prueba.js' open. The file contains a JavaScript snippet that is a comment block with a for loop. The snippet is defined as follows:

```
1
2
3 /*
4  * @autor: Pablo
5  * @email: pablo@gmail.com
6  */
7
8
9 for(let i = 0; i <= i; i++) {
10
11 }
```

11.- Indica las formas de incluir código Javascript en un archivo HTML. ¿Qué diferencias hay entre incluirlo en el <head> o en el <body>? Indica además en qué momento se descarga y ejecutaría utilizando los atributos defer y async.

Las dos formas que podemos incluir código JavaScript en un archivo HTML:

- Mediante la etiqueta <script>

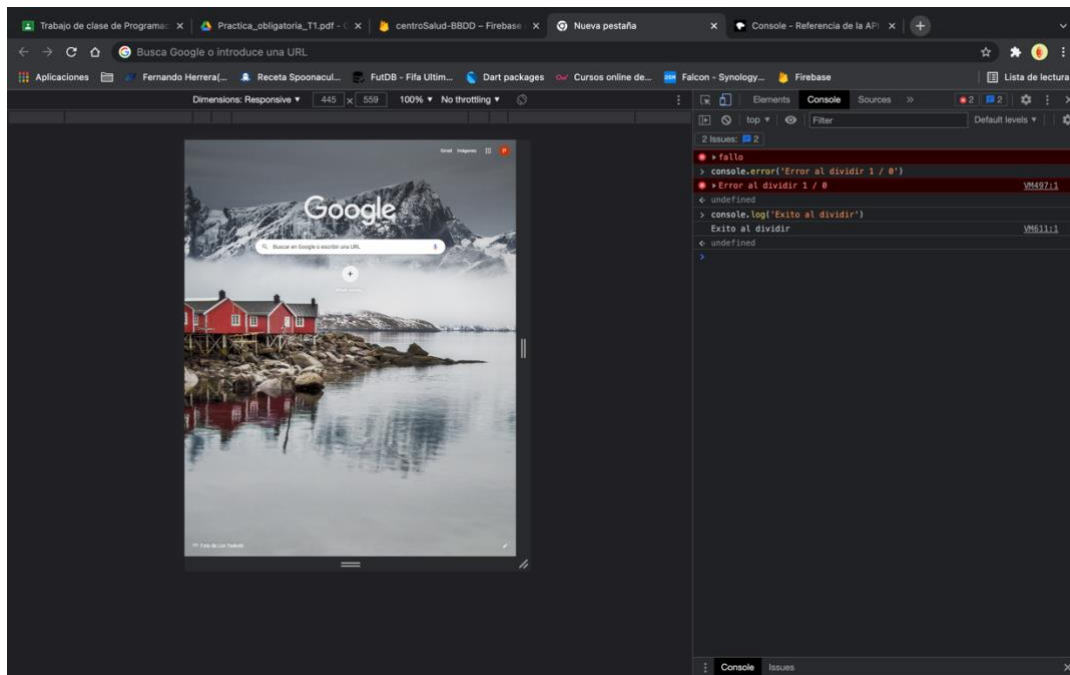
- Importando archivos externos, que hacen referencia a esos archivos. Por ejemplo, tener una carpeta “js” y dentro de esa carpeta tener los ficheros .js, y desde hay enlazarlos con nuestra pagina html.

La diferencia que hay entre ponerlo en el <head> o <body> es, que cuando lo incluimos en el <body> ayuda a mejorar la velocidad de cargar del sitio web, en cambio incluirlo en el <head> se recomienda agregarlo hay ya que podemos visualizar la separación entre los archivos js y html.

Finalmente, una vez que declaramos un <script> de **forma async**, va a descargarse de forma que no va a detener el análisis del código html y una vez descargado se detendrá para ejecutarse el script y una vez ejecutado se volverá a analizar el código html.

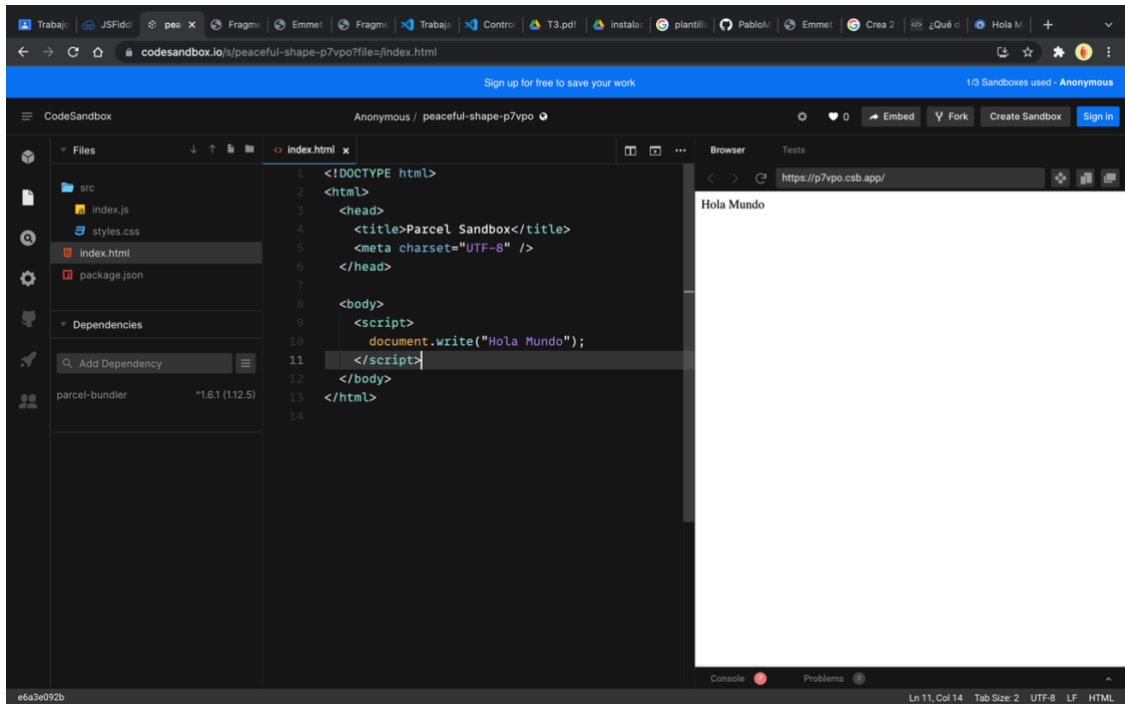
En cambio, de **forma defer**, script se descarga en paralelo con el análisis HTML y su ejecución se aplaza hasta que se haya analizado todo el html.

12.- Utiliza los métodos console.log y console.error directamente en consola con un mensaje que simule su uso como depuración.



13.- Haz un “Hola Mundo” en al menos 2 editores online como los siguientes: Codesandbox, Fiddle, Plunker, CodePen.

Codesandbox



Fiddle

