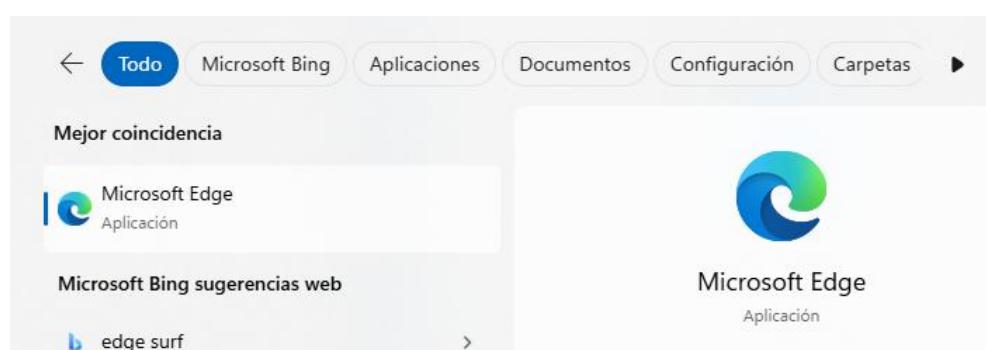
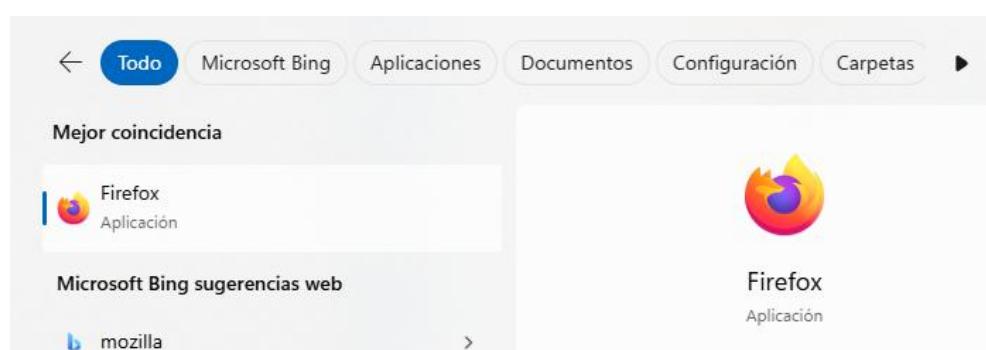
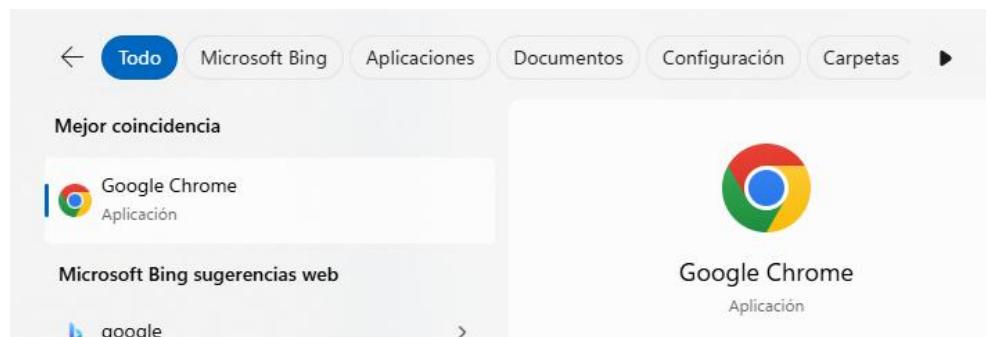
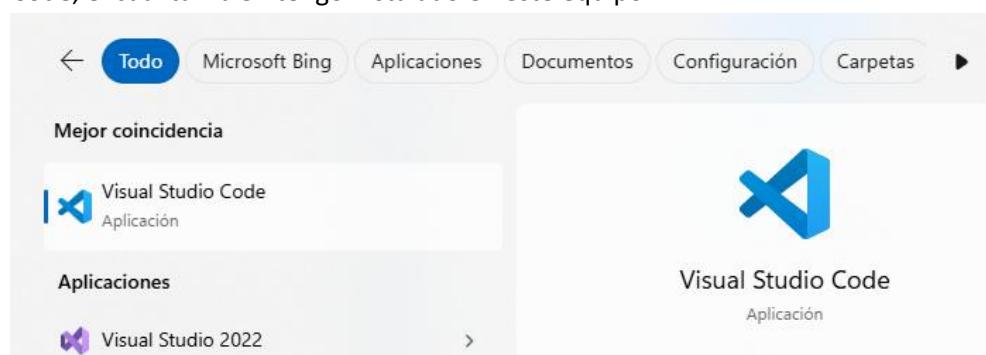


PRÁCTICA UNIDAD 1

- Para este ejercicio usaré los siguientes 3 navegadores que ya tengo instalados en este ordenador.



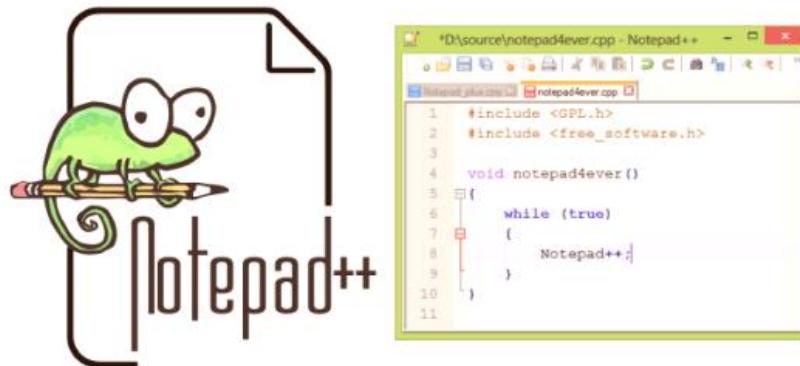
- Por otro lado, el entorno de desarrollo utilizado para esta práctica será Visual Studio Code, el cual también tengo instalado en este equipo.



3. Los editores de texto son programas informáticos que nos permiten crear y modificar archivos digitales compuestos por texto sin formato. Algunos editores de texto y sus principales características son los siguientes:

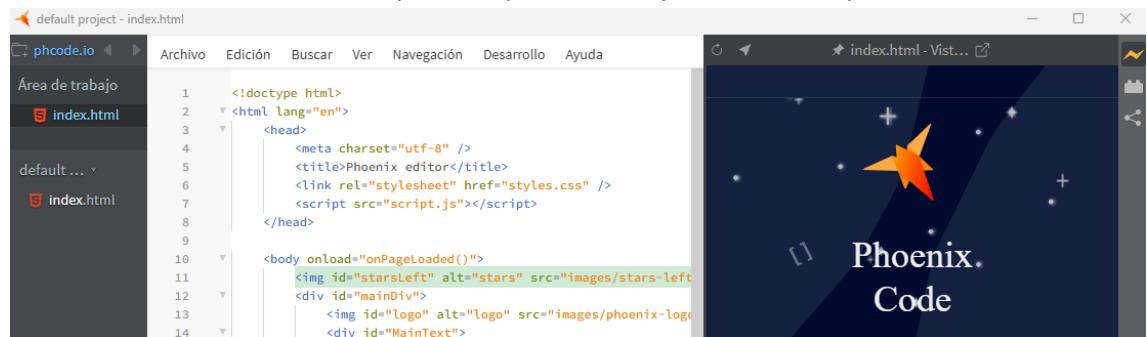
- NOTEPAD++

Fue creado en 2003, cuenta con soporte para varios lenguajes de programación y cuenta con resaltado de sintaxis. Disponible para Windows, Linux y MacOS.



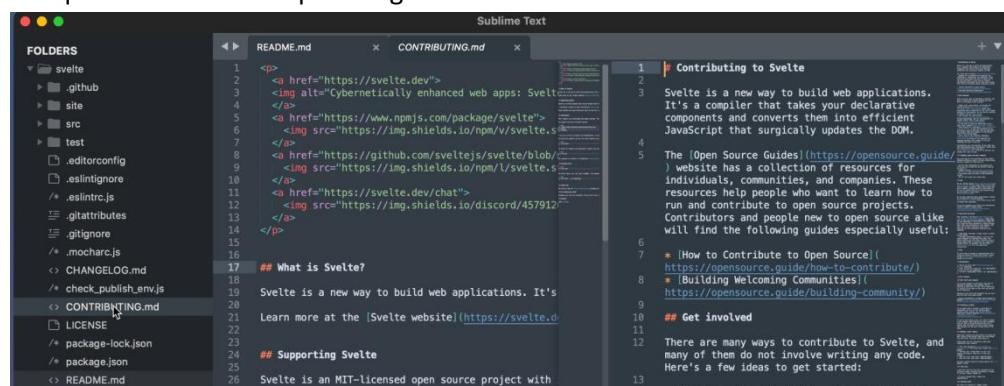
- BRACKETS / PHOENIX CODE

Es un editor más enfocado al desarrollo web y también cuenta con soporte para Linux, MacOS y Windows. Fue creado por adobe en 2014. A día de hoy se le conoce como Phoenix Code, después de que Adobe dejara de darle soporte.

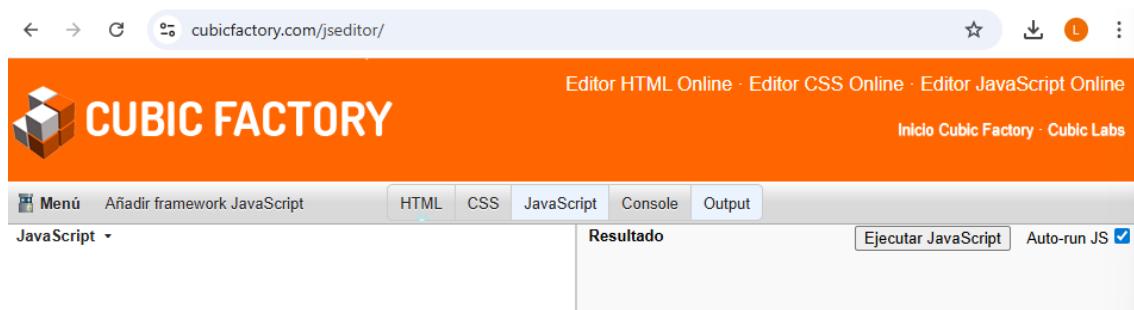


- SUBLIME TEXT

Fue inicialmente una extensión de VIM (orto editor de código), cuenta con extensiones, resaltado de sintaxis y está disponible para los tres principales sistemas operativos. No obstante, se necesita comprar una licencia para su uso, aunque cuenta con una prueba gratuita.



4. Por último, el editor online con soporte para JavaScript que usaré será CUBIC FACTORY: <https://www.cubicfactory.com/jseditor/>



EXPLICACIÓN DE LOS ENTORNOS ELEGIDOS

[ENTORNO DE DESARROLLO - VISUAL STUDIO CODE]

Elegí este entorno principalmente por las siguientes características que presenta:

- Tiene soporte integrado para múltiples lenguajes de programación: Me permitirá ejecutar tanto los scripts con extensión .js como los html de ambos ejemplos.
- Uso de extensiones importantes como Live Server o Live Share con lo que podré ir visualizando en tiempo real lo que voy desarrollando.
- Es gratuito, lo que lo hace muy accesible.
- Uso familiarizado, ya que hemos estado utilizando este entorno a lo largo de este y el curso pasado, lo que es más cómodo para el trabajo.
- Alta personalización para adaptar el trabajo a las necesidades de cada tarea o proyecto.

[EDITOR DE TEXTO PLANO – PHOENIX CODE]

Por otro lado, como editor de texto plano me he decidido a usar Brackets. A pesar de que en este equipo ya tenía instalado otro editor de texto como Notepad++, elegí el primero debido a su interfaz más moderna y atractiva. Otras características que tiene son:

- Ofrece vistas previas en tiempo real, al igual que la extensión de Live Preview que tiene Visual Studio Code.
- Gratuito e instalación sencilla, no pide ninguna suscripción, pago o creación de cuenta, lo que hace que su uso exclusivamente para este ejercicio sea ideal.
- Está enfocado al desarrollo web, lo que significa que sus herramientas están orientadas a las necesidades de este campo.

[EDITOR ONLINE – CUBIC FACTORY]

Por último, como editor de texto online usaré Cubic Factory, ya que tiene soporte para JavaScript y cubre perfectamente las necesidades de esta práctica. Algunas características que presenta son:

- Está diseñado para ser intuitivo y fácil de usar.
 - Se puede utilizar como prueba de compatibilidad y test para ver si el código se integra bien en otras plataformas.
 - Bajo consumo de recursos locales ya que parte del procesamiento del programa se realiza en servidores remotos por lo que el propio ordenador solo necesita recursos básicos, además el diseño básico de la web favorece a ello.
 - Acceso desde diversos dispositivos como ordenadores o portátiles.
 - No requiere de ninguna instalación permitiendo que el acceso web a este editor de texto sea realmente más rápido y sencillo.
5. Ahora vamos a proceder a integrar el código de prueba en cada uno de los entornos elegidos.

VISUAL STUDIO CODE

Para este entorno ejecutaré el código mediante el archivo externo.html con su correspondiente script.js



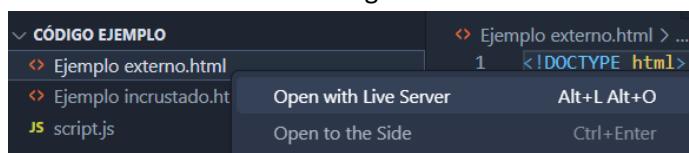
```

EXPLORER
CÓDIGO EJEMPLO
Ejemplo externo.html
Ejemplo incrustado.html
JS script.js

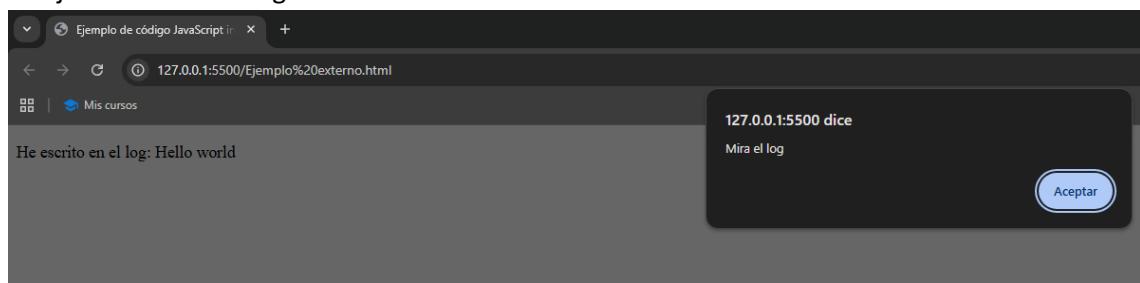
Ejemplo externo.html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
<p id="msg1">Prueba</p>
<!-- El script se llama después porque si se hace en el head no se ha cargado aún la página
    Esto se estudiará más adelante --&gt;
&lt;script type="text/javascript" src="script.js"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

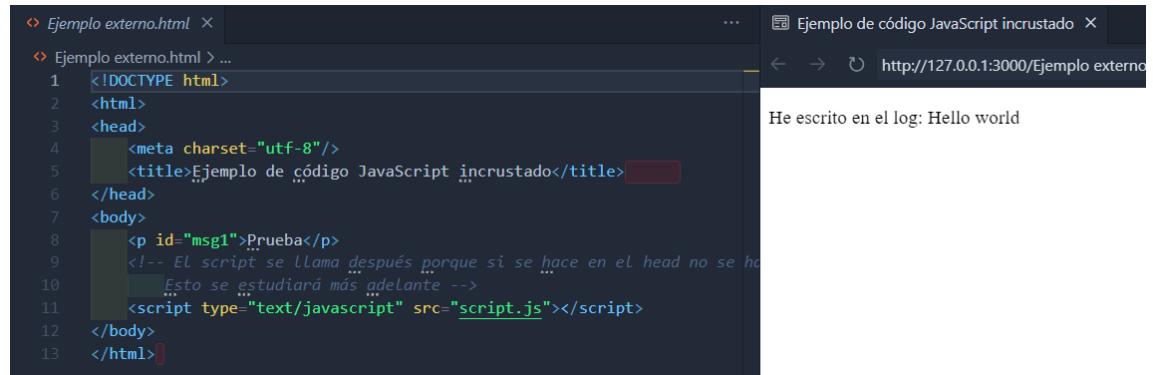
La primera forma de ejecutarlo será mediante la herramienta de “Live Server” que nos llevará a una ventana de navegador externa desde donde correrá el programa.



La ejecución sería la siguiente.



Otra forma de ejecutar el código, sería mediante la extensión de Live Preview, no obstante, tiene algunas limitaciones como el de no mostrar la ventana emergente del inicio y solo mostrar el contenido de la página.

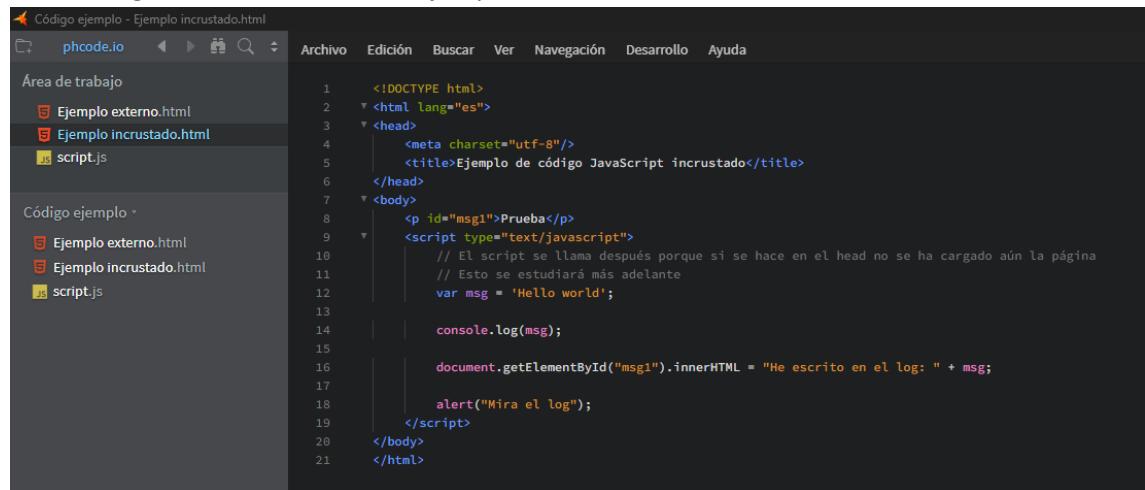


The screenshot shows a browser window with two tabs. The left tab is titled 'Ejemplo externo.html' and contains the source code for an external HTML file. The right tab is titled 'Ejemplo de código JavaScript incrustado' and shows the result of running the embedded script. The output is a single line of text: 'He escrito en el log: Hello world'.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
<p id="msg1">Prueba</p>
<!-- El script se llama después porque si se hace en el head no se ha cargado aún la página
    Esto se estudiará más adelante -->
<script type="text/javascript" src="script.js"></script>
</body>
</html>
```

PHOENIX CODE

En este segundo entorno usaré el ejemplo incrustado.html.



The screenshot shows the Phoenix Code IDE interface. On the left, there's a sidebar labeled 'Área de trabajo' (Workspace) containing files: 'Ejemplo externo.html', 'Ejemplo incrustado.html', and 'script.js'. Below it, another sidebar labeled 'Código ejemplo' (Example Code) also lists the same three files. The main central area is a code editor showing the source code for 'Ejemplo incrustado.html'. The code includes a script that logs 'Hello world' to the console and updates a paragraph element with the message 'He escrito en el log: Hello world'.

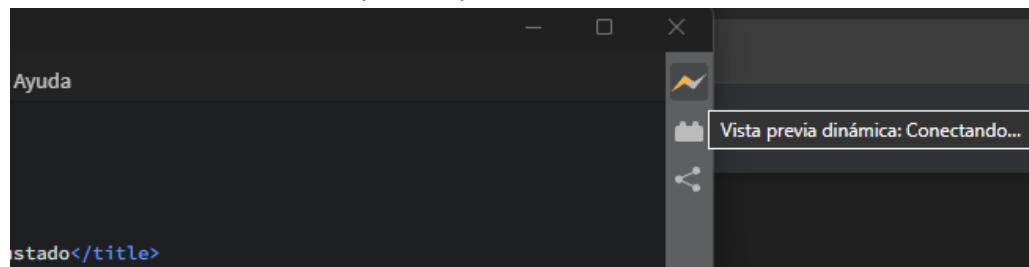
```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8"/>
<title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
<p id="msg1">Prueba</p>
<script type="text/javascript">
    // El script se llama después porque si se hace en el head no se ha cargado aún la página
    // Esto se estudiará más adelante
    var msg = 'Hello world';

    console.log(msg);

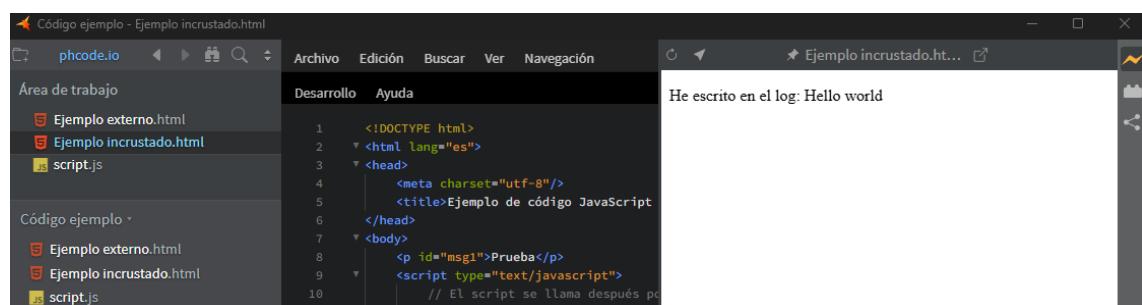
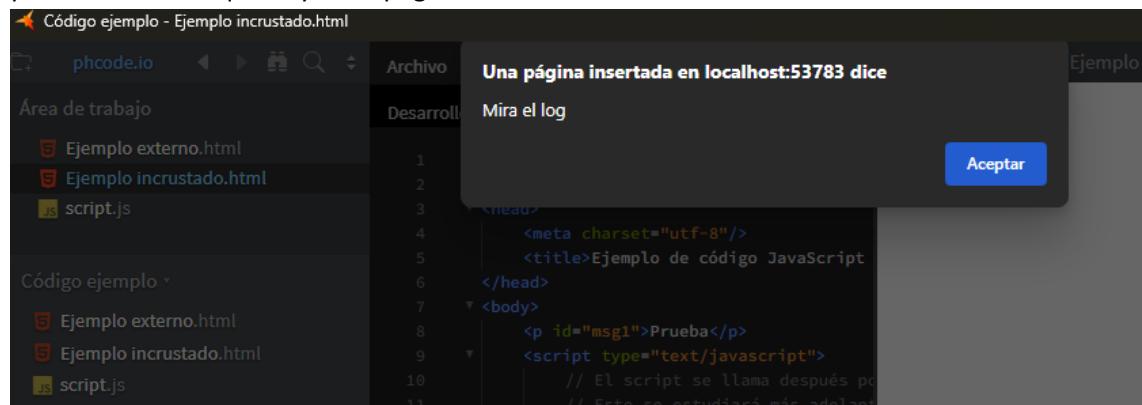
    document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;

    alert("Mira el log");
</script>
</body>
</html>
```

Phoenix ofrece una herramienta similar a la extensión de Visual Studio, pero más dinámica. Se encuentra en la parte superior derecha.



La ejecución sería la siguiente. Primero lanzaría una ventana emergente y al aceptar previsualizaría lo que hay en la página.



CUBIC FACTORY

Para el último entorno, usaré tanto el ejemplo incrustado como el ejemplo externo.

Para ejecutarlo únicamente habría que tener abierta la ventana de “Output”.

Con el ejemplo incrustado.html tendríamos la siguiente ejecución.

The screenshot shows the CUBIC FACTORY interface. On the left, there is a code editor with the following HTML code:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8"/>
    <title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
    <p id="msg1">Prueba</p>
    <script type="text/javascript">
        // El script se llama después porque si se hace en el head no se ha cargado aún la página
        // Esto se estudiará más adelante
        var msg = 'Hello world';
        console.log(msg);

        document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;

        alert("Mira el log");
    </script>
</body>
</html>

```

On the right, there is a "Resultado" (Result) panel showing the output: "He escrito en el log: Hello world".

Con el ejemplo externo.html ejecutaría de la misma forma. Es importante tener abierta la venta de JavaScript para integrar cada código por separado.

```

Menú Añadir framework JavaScript
HTML JavaScript Resultado
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8"/>
    <title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
    <p id="msg1">Prueba</p>
    <!-- El script se llama después porque si se hace en el head no se ha cargado aún la página
        Esto se estudiará más adelante -->
    <script type="text/javascript" src="script.js">
</script>
</body>
</html>

```

```

var msg = 'Hello world';
console.log(msg);

document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;

alert("Mira el log");

```

He escrito en el log: Hello world

En ninguno de los dos casos lanza el alert con la ventana emergente y solo muestra el contenido de la página.

6. Apertura del código con cada uno de los navegadores.

GOOGLE CHROME

Haciendo clic derecho en la página y seleccionando “inspeccionar”, podremos acceder al código mediante el navegador de Google Chrome. Otra forma de acceder es presionando la tecla “F12” en la página que queramos ver su código fuente.

```

DevTools - file:///C:/Users/Luisfer/Downloads/C%3B3digo%20ejemplo%20para%20pr%C3%A1ctica%20Unidad%201/C%3B3digo%20ejemplo%20incrustado.html
Elementos Consola Fuentes Red Rendimiento Memoria Aplicación Lighthouse
<!DOCTYPE html>
...<html lang="es"> == $0
<head>
    <meta charset="utf-8">
    <title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
    <p id="msg1">He escrito en el log: Hello world</p>
    <script type="text/javascript">
        // El script se llama después porque si se hace en el head no se ha cargado aún la página
        // Esto se estudiará más adelante
        var msg = 'Hello world';

        console.log(msg);

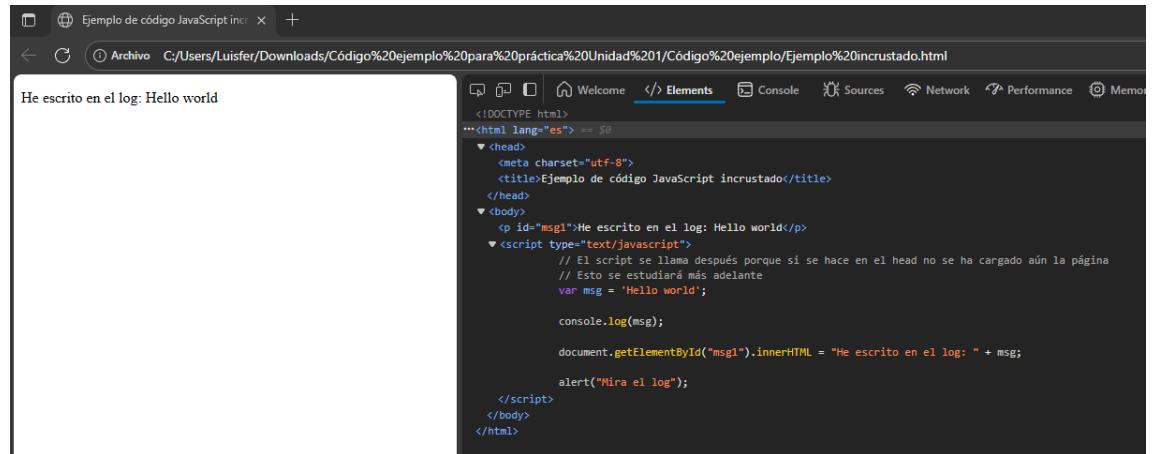
        document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;

        alert("Mira el log");
    </script>
</body>
</html>

```

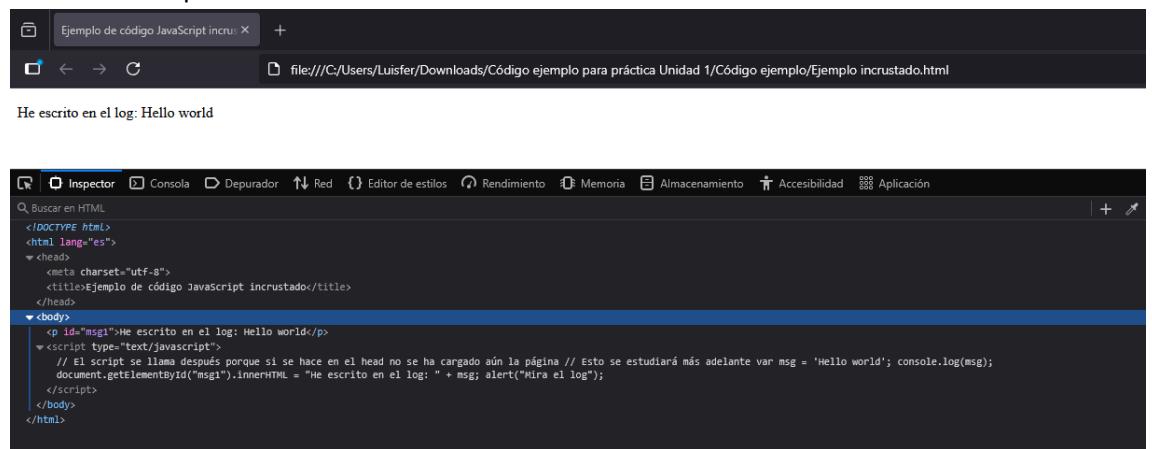
MICROSOFT EDGE

Accedemos de la misma forma desde este navegador. Clic derecho e “inspeccionar” o con la tecla “F12”.

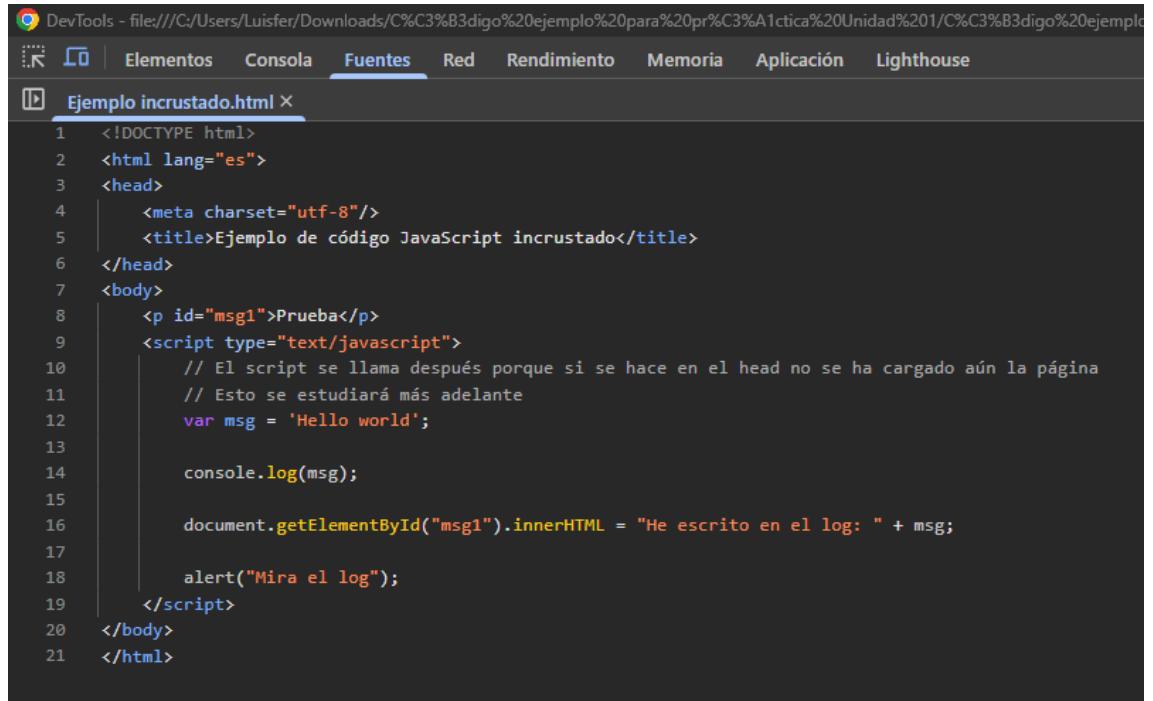


MOZILLA FIREFOX

Al igual que los otros dos navegadores, se abre de la misma forma, con “F12” o clic derecho e “inspeccionar”.



7. Para la depuración del código es necesario acceder la pestaña “Sources” o “Debugger” del navegador que vayamos a usar. En este primer caso usará **Google Chrome**.

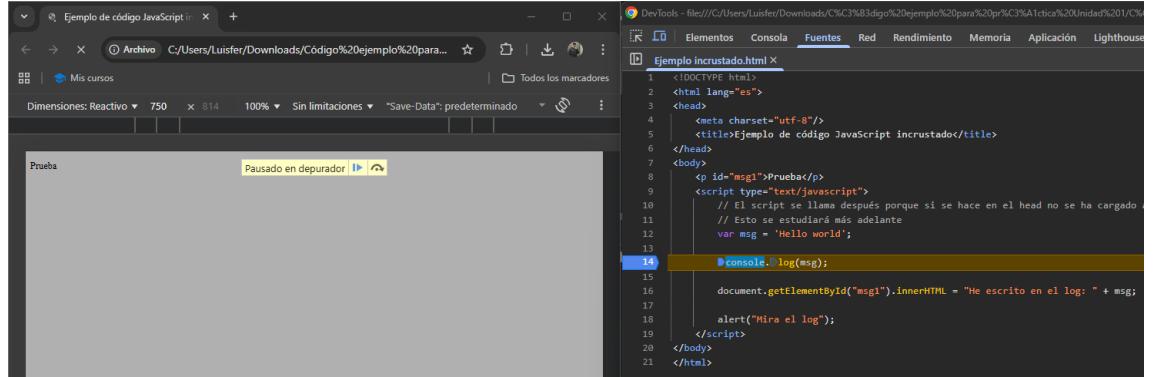


```

1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="utf-8"/>
5      <title>Ejemplo de código JavaScript incrustado</title>
6  </head>
7  <body>
8      <p id="msg1">Prueba</p>
9      <script type="text/javascript">
10         // El script se llama después porque si se hace en el head no se ha cargado aún la página
11         // Esto se estudiará más adelante
12         var msg = 'Hello world';
13
14         console.log(msg);
15
16         document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;
17
18         alert("Mira el log");
19     </script>
20 </body>
21 </html>

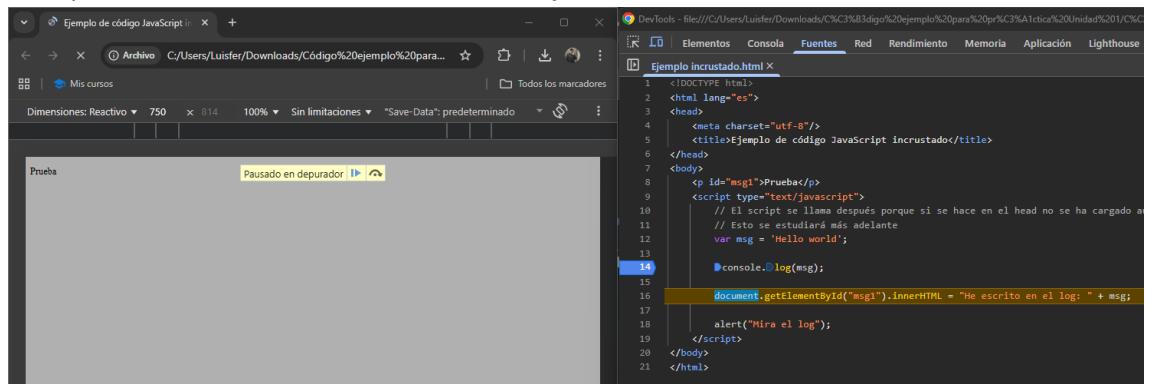
```

Para establecer un punto de ruptura lo haremos haciendo clic izquierdo en la línea deseada, aquí se parará la ejecución del código una vez recarguemos la página.

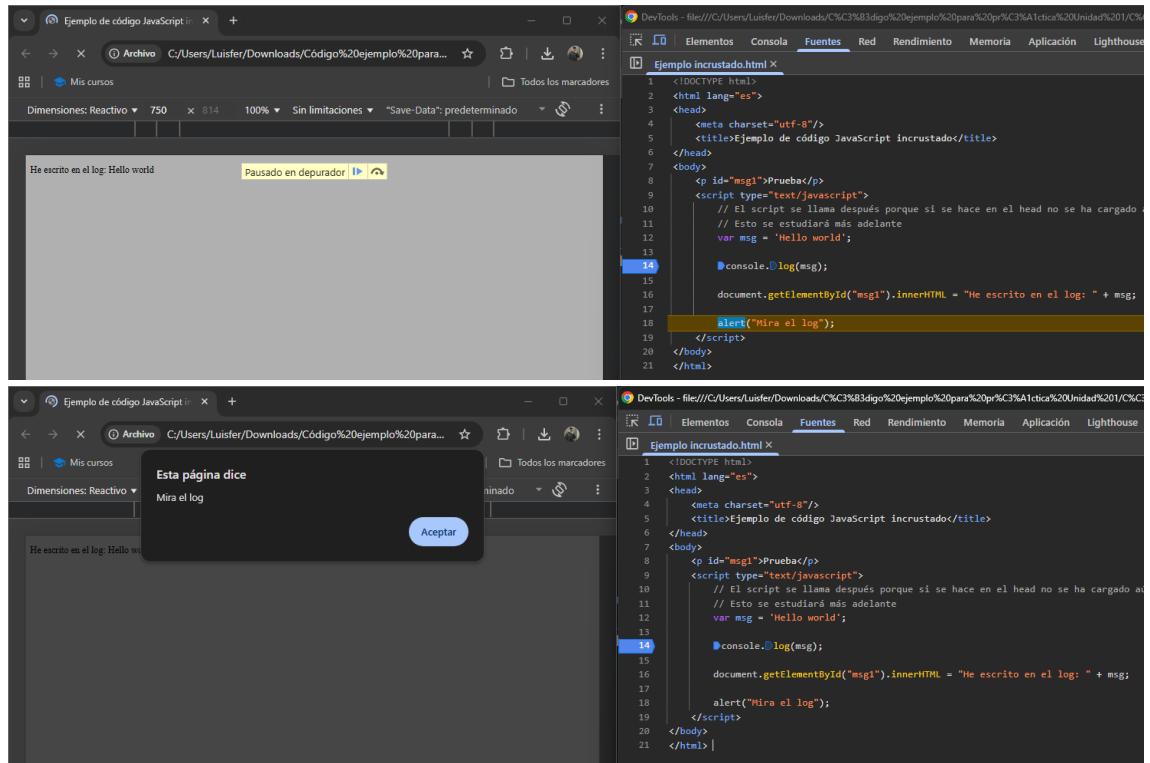


The screenshot shows the Google Chrome browser window with the URL "C:/Users/Luisfer/Downloads/Código%20ejemplo%20para%20pr%C3%A1ctica%20Unidad%201/C%C3%B3digo%20ejemp..." and the DevTools Sources tab open. The code editor highlights line 14: `14 console.log(msg);` with a yellow background. The browser window shows the page content "Prueba" and the status bar indicates "Pausado en depurador".

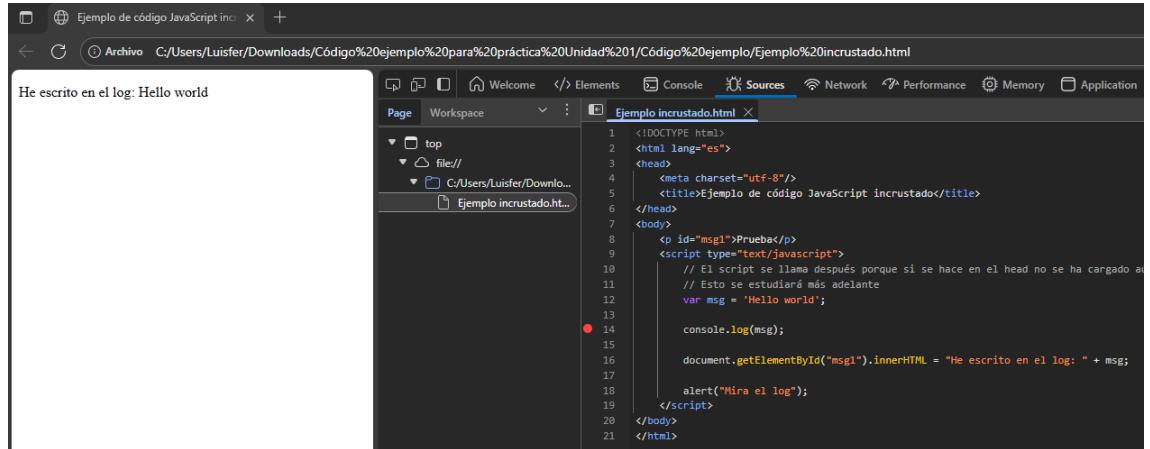
Para ir ejecutando el código paso a paso desde este breakpoint, pulsaremos la tecla “F9” y observaremos como irá avanzando la ejecución en cada línea.



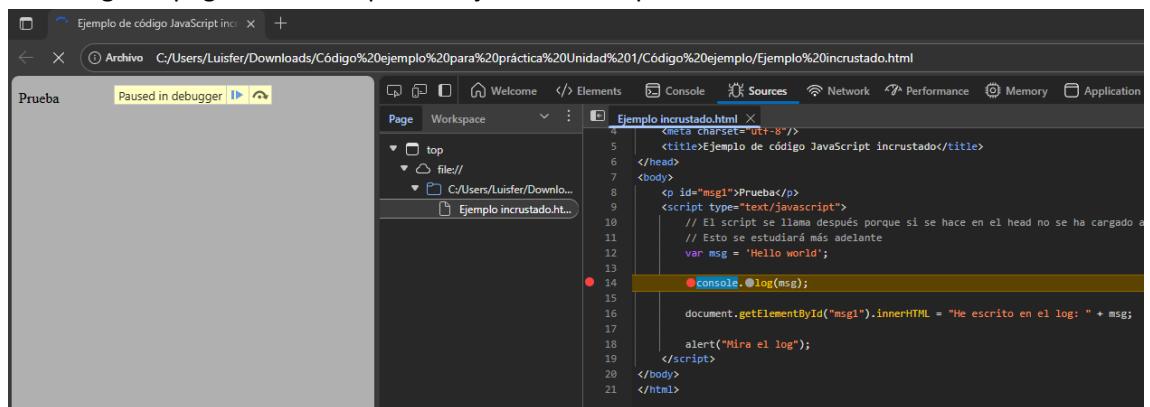
The screenshot shows the Google Chrome browser window and DevTools Sources tab. The code editor highlights line 14: `14 console.log(msg);` with a yellow background. The browser window shows the page content "Prueba" and the status bar indicates "Pausado en depurador". The DevTools sidebar shows the execution progress: "Linea 14" and "Pausado en depurador".



Para los otros dos navegadores la forma de depurar es similar. Seguiremos con **Microsoft Edge** donde empezaremos añadiendo el punto de ruptura en la línea 14.



Al recargar la página se habrá pausado justo en este punto.



Para ir avanzando línea a línea el proceso es igual, presionaremos “F9” e iremos avanzando en cada una de ellas.

The image consists of three vertically stacked screenshots of a browser's developer tools, specifically the debugger panel. Each screenshot shows a different state of the execution of a JavaScript script named 'Ejemplo incrustado.html'.

Screenshot 1: The code is at line 14, where a red dot marks the current line. The line contains a call to 'console.log(msg);'. The code continues with 'document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;' followed by an alert box. The alert box is visible in the foreground, displaying the message 'He escrito en el log'.

```
4 <meta charset="utf-8"/>
5 <title>Ejemplo de código JavaScript incrustado</title>
6 </head>
7 <body>
8 <p id="msg1">Prueba</p>
9 <script type="text/javascript">
10 // El script se llama después porque si se hace en el head no se ha cargado al
11 // Esto se estudiará más adelante
12 var msg = 'Hello world';
13
14 ●console.log(msg);
15
16 document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;
17
18 alert("Mira el log");
19 </script>
20 </body>
21 </html>
```

Screenshot 2: The code has moved to line 18, where the alert box is now closed. The code continues with 'alert("Mira el log");' followed by the closing script tag and body tag.

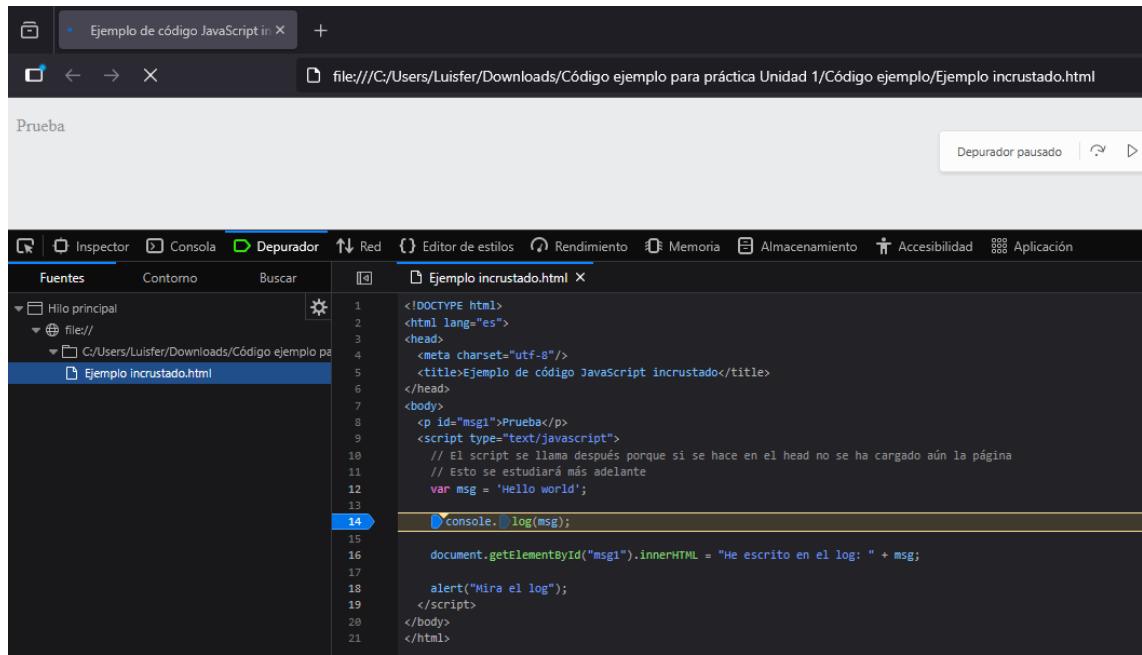
```
4 <meta charset="utf-8"/>
5 <title>Ejemplo de código JavaScript incrustado</title>
6 </head>
7 <body>
8 <p id="msg1">Prueba</p>
9 <script type="text/javascript">
10 // El script se llama después porque si se hace en el head no se ha cargado al
11 // Esto se estudiará más adelante
12 var msg = 'Hello world';
13
14 ●console.log(msg);
15
16 document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;
17
18 alert("Mira el log");
19 </script>
20 </body>
21 </html>
```

Screenshot 3: The code has moved to line 21, where the browser is prompting the user to close the dialog box. The message 'Esta página dice' is displayed above the code, and a blue 'Aceptar' (Accept) button is visible.

```
4 <meta charset="utf-8"/>
5 <title>Ejemplo de código JavaScript incrustado</title>
6 </head>
7 <body>
8 <p id="msg1">Prueba</p>
9 <script type="text/javascript">
10 // El script se llama después porque si se hace en el head no se ha cargado al
11 // Esto se estudiará más adelante
12 var msg = 'Hello world';
13
14 ●console.log(msg);
15
16 document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;
17
18 alert("Mira el log");
19 </script>
20 </body>
21 </html>
```

Por último, repetiremos este mismo proceso con **Mozilla Firefox**.

Establecemos el punto de ruptura y recargamos la página.

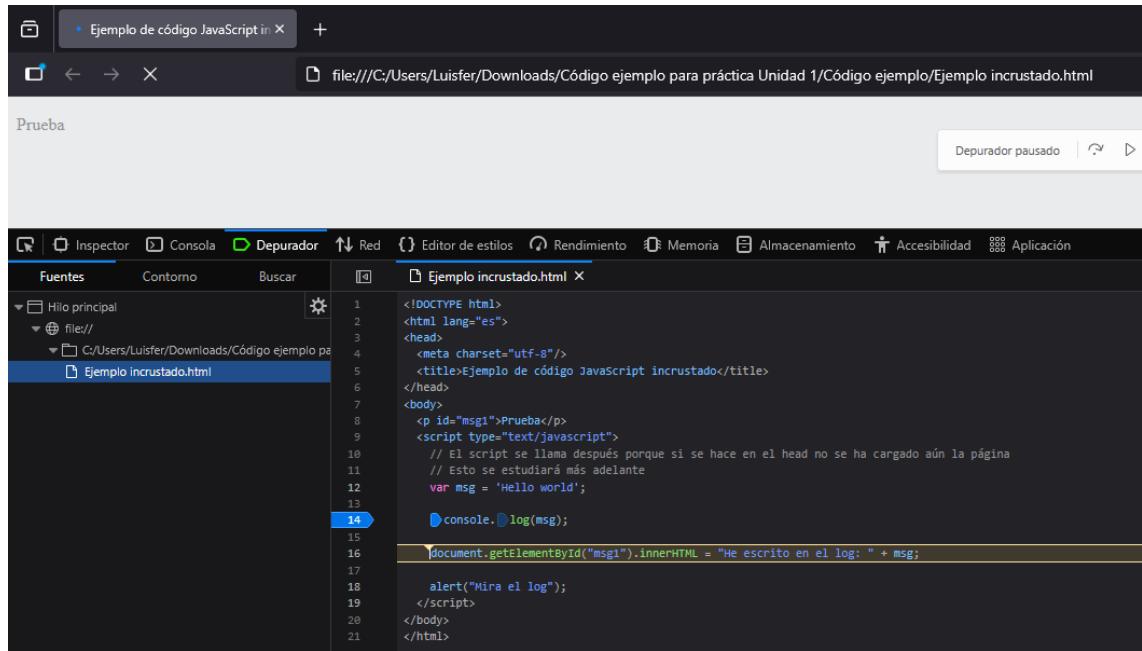


The screenshot shows the Mozilla Firefox developer tools interface. The title bar says "Ejemplo de código JavaScript incrustado.html". The main area displays the following HTML code:

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8"/>
<title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
<p id="msg1">Prueba</p>
<script type="text/javascript">
    // El script se llama después porque si se hace en el head no se ha cargado aún la página
    // Esto se estudiará más adelante
    var msg = 'Hello world';
    console.log(msg);
    document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;
    alert("Mira el log");
</script>
</body>
</html>
```

The line number 14 is highlighted with a blue arrow, indicating it is the current line being debugged. The status bar at the bottom right says "Depurador pausado".

En este caso, para avanzar en cada línea usaremos la tecla “F10”, ya que la tecla “F9” en este buscador no está configurada para ello.



The screenshot shows the Mozilla Firefox developer tools interface again. The title bar says "Ejemplo de código JavaScript incrustado.html". The main area displays the same HTML code as before, but now the line 14 has been executed. The status bar at the bottom right says "Depurador pausado".

The code execution has reached the line 14, which contains the `console.log` statement. The output of this log statement is visible in the browser's developer tools console, showing the message "He escrito en el log: Hello world".

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8"/>
<title>Ejemplo de código JavaScript incrustado</title>
</head>
<body>
<p id="msg1">Prueba</p>
<script type="text/javascript">
    // El script se llama después porque si se hace en el head no se ha cargado aún la página
    // Esto se estudiará más adelante
    var msg = 'Hello world';

    console.log(msg);

    document.getElementById("msg1").innerHTML = "He escrito en el log: " + msg;

    alert("Mira el log");
</script>
</body>
</html>

```

- Para encontrar la consola, nos tenemos que situar en la pestaña “console” de las herramientas de desarrollador del navegador. Esto es lo que se muestra en cada uno.

GOOGLE CHROME

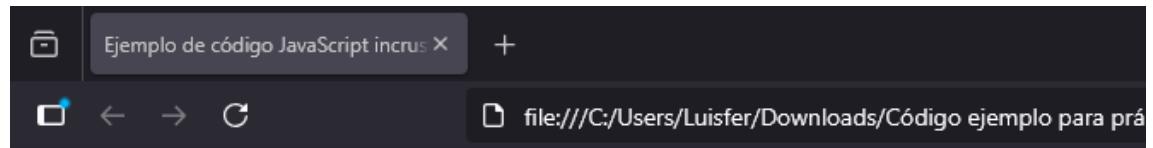
DevTools - file:///C:/Users/Luisfer/Downloads/C%C3%B3digo... | Elementos | Consola | Fuentes | Red

top | Todos los marcadores | Filtrar

Hello world

>

He escrito en el log: Hello world

MOZILLA FIREFOX**MICROSOFT EDGE**