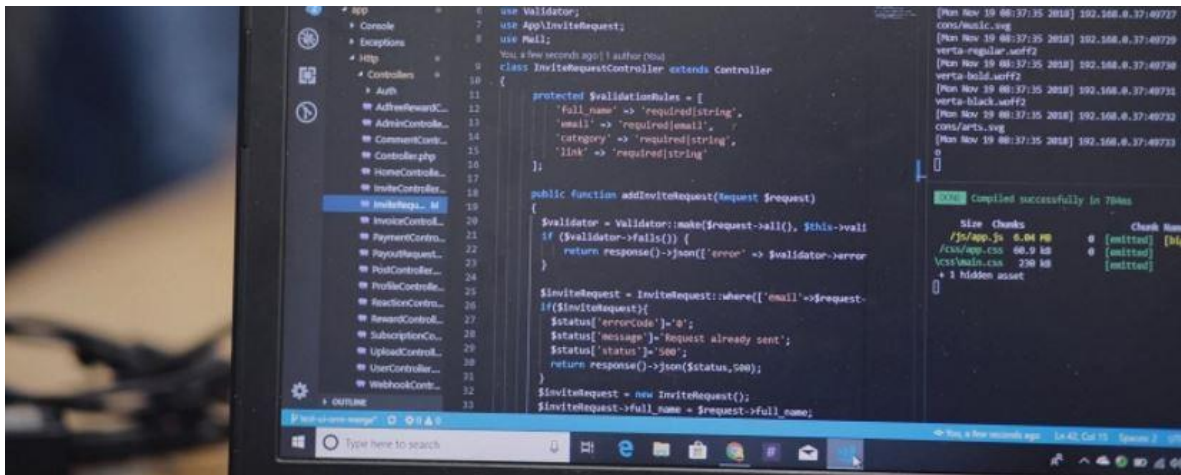


Los mejores editores gratuitos de HTML, CSS y JavaScript (2019)

Por José Manuel Alarcón. Publicado el 28 de mayo de 2019 a las 10:00

👉 El mejor framework de JavaScript es el que mejor conoces. 👉 Haz que sea [Angular](#) [Ver curso](#)



Hace hoy exactamente 5 años, el 28 de mayo de 2014, escribí un artículo que enseguida se convirtió en uno de los más populares de nuestro blog y de los más populares sobre el tema en Internet. Se titulaba "[Los 10 mejores editores gratuitos \(o casi\) de HTML, CSS y JavaScript](#)" y, como su propio nombre indica, nos mostraba una selección de editores de código para la Web que podíamos utilizar gratuitamente.

Desde entonces **las cosas han cambiado mucho**. Ya en 2013 la revista **Mashable** sentenció que era [el año del Responsive Web Design](#), aunque en España empezó a explotar en 2014, más o menos cuando escribí ese artículo. La realidad actual es que [el RWD es inherente al diseño web](#) y hoy en día es inimaginable diseñar una web sin usarlo.

¿Y qué tiene esto que ver con los editores? Pues que hace 5 años, sobre todo si eras un principiante, podías permitirte el lujo de **utilizar algún tipo de diseñador visual para crear tus páginas** o aplicaciones web. **Hoy en día es inviable**. Vamos, puedes hacerlo, y de hecho la mayor parte de los editores que puse en aquel artículo siguen funcionando, **pero no te van a servir de mucho**.

Adiós al diseño visual

Los diseños web actuales se realizan de manera que **los elementos se distribuyen de forma totalmente diferente según el ancho** del [viewport del dispositivo](#) en el que se estén visualizando. Ello implica que no podemos hacer un diseño "visual" distribuyendo los elementos y dejándolos en una posición fija, sino que el diseño parte normalmente de los tamaños de pantalla más pequeños (móviles) y se van variando a medida que crecen éstos:

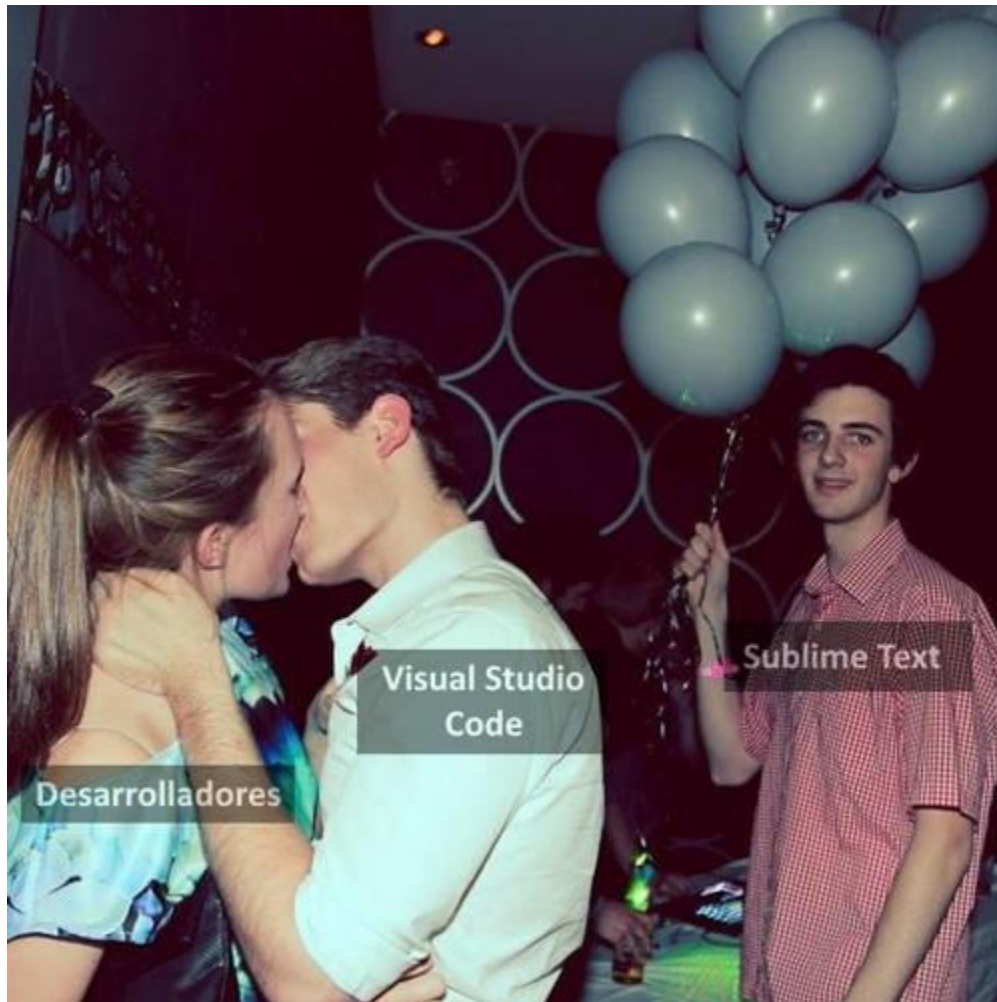
(Animación vía [Giphy](#))

El **diseño web responsivo** (y el diseño web adaptativo) **requiere escribir el código** en el editor, tanto el HTML como el CSS y las *media queries* correspondientes, y luego ir comprobando en un navegador real el efecto de cada cambio. Gran parte de las veces lo que haces es retocar en tiempo real propiedades CSS usando las herramientas del desarrollador del navegador para afinarlo (por suerte hoy en día los navegadores te dejan enlazar estas herramientas con los archivos originales, y que esos cambios se vean al mismo tiempo en aquéllos también, y por lo tanto en el editor).

Hoy, más que nunca, el diseño/desarrollo web es [una disciplina técnica](#) que requiere [escribir código](#), y no arrastrar y soltar o ver en tiempo real cómo quedan las cosas.

Este es el principal motivo de que aquella lista hoy en día sea obsoleta y sobren la mitad de las opciones que proponía por aquel entonces, especialmente los que hacían hincapié en el diseño visual como **Kompozer** (cuya página ya ni funciona) o **BlueGriffon** (que se ha convertido en una herramienta para crear eBooks y ya no es gratuito).

Además, algunos que en aquella época eran lo máximo han ido cayendo un poco en desgracia por la aparición de nuevos productos que se han hecho con el interés de los desarrolladores. En mi opinión, el gran ganador en ese sentido ha sido **Visual Studio Code**, y los perdedores **Sublime Text** y **Atom**.

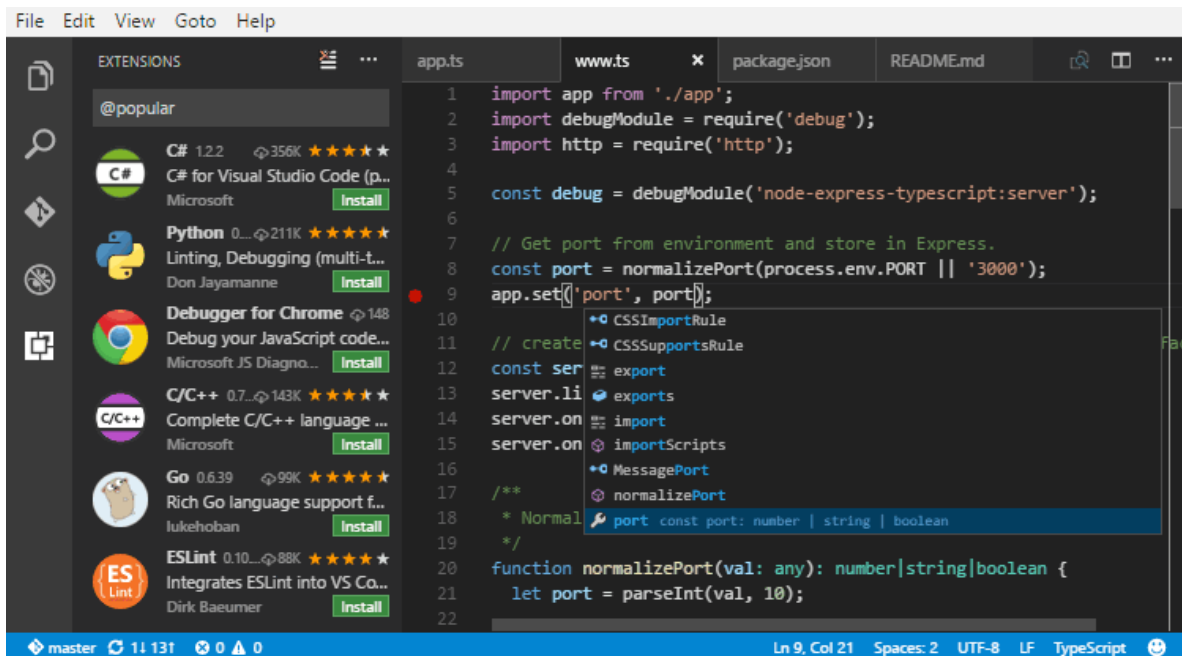


Con todo esto en mente os voy a dejar a continuación una lista, más corta, de mis **editores de código gratuitos preferidos para HTML/CSS y JavaScript según tres categorías** (uso general, aprendizaje y *cloud*). En mi humilde opinión son **los mejores** en cada caso, dentro de los **editores gratuitos**. Existen algunos de pago, fundamentalmente [WebStorm](#) de JetBrains y quizá [Dreamweaver](#) de Adobe solo para diseño web, que son buenísimos pero se salen de esta premisa de la gratuidad.

Nota: antes de que me digas que este o aquel editor que te gusta tanto no está en la lista ten en cuenta una cosa: es **mi** lista, la de **mis** editores favoritos por diversos motivos, y puede que no los conozca todos o quizá no veo las cosas de la misma manera que tú, pero no es por mala intención o dejadez, sino que es **mi** lista personal e intransferible. Ahora bien, estaré encantado de que dejes un comentario en la parte de abajo hablando de tu editor favorito que no está en la lista y por qué consideras que debería estar. ¡Gracias!

¡Allá vamos!

Uso general: Visual Studio Code - El mejor para el día a día y para hacer de todo



He de confesar que cuando salió no hubiera apostado casi nada por él y pensé que iba a ser otro de tantos productos fallidos de Microsoft intentando ser "cool". No podía estar más equivocado. Con el tiempo VSCode se ha convertido en **el editor**, y ya es el más utilizado por desarrolladores de todo tipo.

Es gratuito, de código abierto, multiplataforma (Windows, Linux, macOS, ARM y ahora también Web), ágil, rápido y súper-potente. Además tiene infinitas extensiones para poder ampliar su funcionalidad sin que pierda la ligereza que lo caracteriza, e integra soporte para las herramientas más habituales (desde Git o la línea de comandos integrada, hasta Docker).

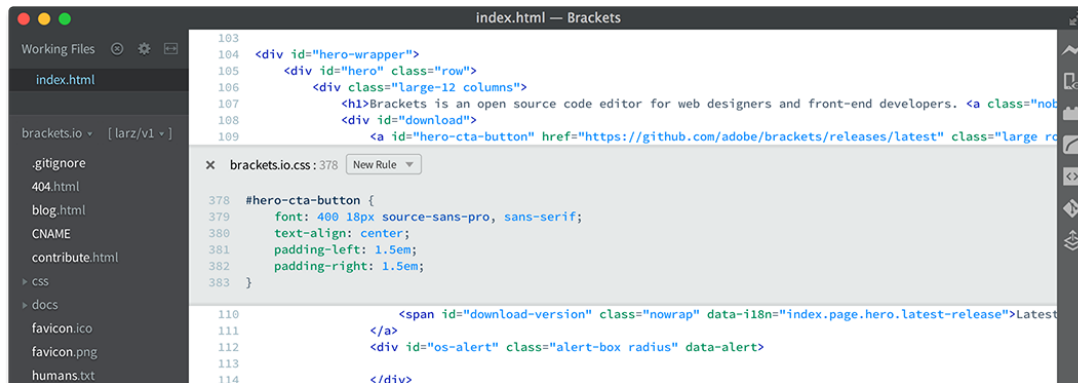
No solo se puede usar para programar HTML, CSS y JavaScript, sino que tiene un gran soporte (nativo o con extensiones) para infinidad de lenguajes: [plataforma .NET y C#](#), Node.js, PHP, Java, Python, C++, Go... además de buen soporte para bibliotecas JavaScript como [Angular](#), Vue.js o React, y no solo para programación. Por ejemplo, este artículo lo estoy escribiendo en Markdown usando Visual Studio Code. Prácticamente lo uso para todo en la actualidad, y además lo sincronizo entre todos mis equipos. Su soporte para [Live Share](#) incluso te permite trabajar con otros desarrolladores en tiempo real en el mismo código.

En lo que se refiere a HTML, CSS y JavaScript en concreto ofrece grandes características como, por supuesto, la completitud automática de código, soporte para emmet/zencoding, ayuda contextual a medida que escribes, selectores de colores, refactorización de código,

depuración sin salir de la herramienta, soporte para Live Server (servir y recargar automáticamente las páginas mientras las editas, con una extensión) y mil cosas más. Demasiadas para enumerarlas.

Por todo ello, hoy por hoy, Visual Studio Code es mi editor principal y el que recomendaría en casi todos los casos a cualquiera.

Aprender: Brackets - Ideal para principiantes



Aunque este editor creado por Adobe no es tan conocido como otros, se lo suelo recomendar a mis alumnos que están empezando con HTML y CSS porque es un editor muy enfocado en estos lenguajes, además de en JavaScript y con algunas facilidades especialmente interesantes para enfocarte mientras comienzas.

Aparte de las características habituales (ayuda contextual a la escritura, auto-cierre de etiquetas, etc...) posee tres características que lo hacen especialmente interesante para este perfil de iniciación:

- **Live preview:** integra una combinación de teclas, **CTRL+E**, que si la pulsas cuando estás sobre una clase CSS en una etiqueta o sobre una función JavaScript, te muestra directamente en línea, sin cambiar de editor, la definición de la misma permitiéndote además editarla:

```

49     </style>
50 </head>
51 ▼ <body>
52 ▼ <ul class="flex flex1">
53     <li></li>
54     <li></li>
55     <li></li>
56 </ul>
57
58 ▼ <ul class="flex flex2">
59     <li></li>
60     <li></li>
61     <li></li>
62 </ul>
63
64 ▼ <script>
65     // Este script pone los anchos finales
66     var filasFlex = document.getElementsByTagName("ul");
67     for(var filas=0; filas<filasFlex.length; filas++)
68         var coldes = filasFlex[filas].getElementsByClassName("flex");

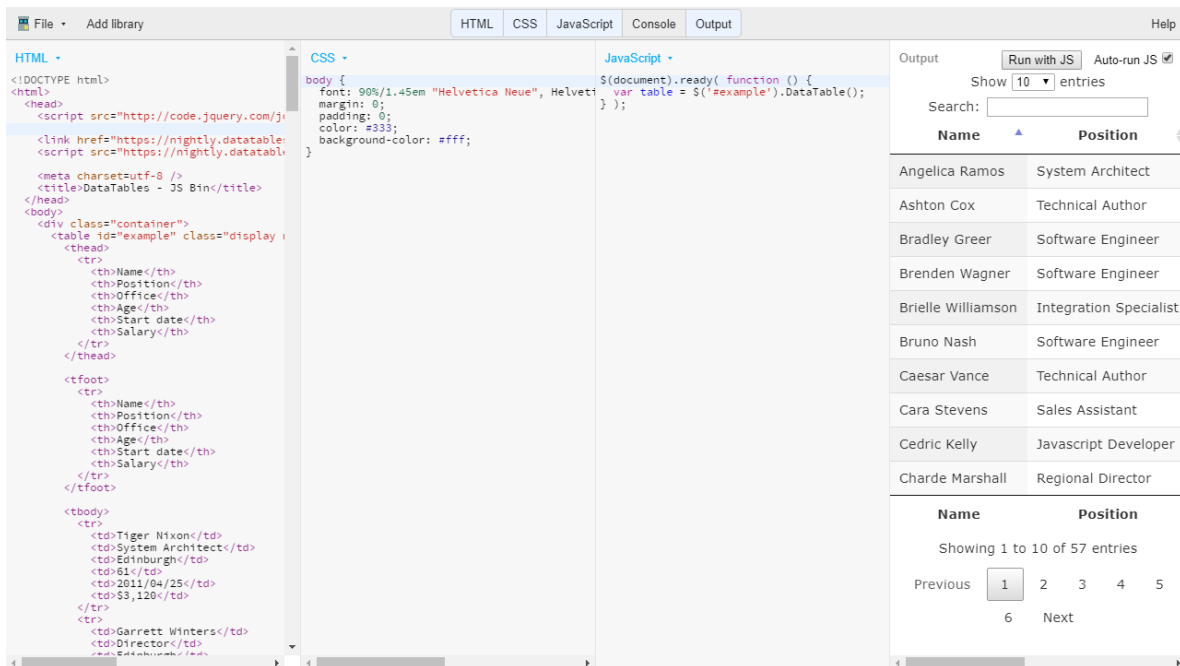
```

Fueron unos de los pioneros en esto y, aunque pueda parecer algo menor, el hecho de que lo traiga integrado resulta muy cómodo y ayuda mucho cuando estás empezando (en VScode lo puedes conseguir también con la extensión [CSS Peek](#) aunque no está tan logrado, y es nativo en el caso de JavaScript).

- **Vista previa de los cambios en tiempo real.** Basta pulsar un botón para que se abra el navegador y se empiece a servir la página, quedando ésta enlazada con el editor. De este modo, cada vez que hagas cualquier cambio (aún sin grabarlo), se verá reflejado en tiempo real en el navegador, sin necesidad de refrescar. Esto se puede hacer también con otros editores como VSCode, pero necesitas una extensión y es más complicado. Nuevamente un "win" para los principiantes.
- **Soporte de serie para preprocesadores CSS (SASS y Less):** si además de CSS estás aprendiendo a sacarle partido a estas potentes herramientas, te vendrá muy bien, y nuevamente sin necesidad de extensiones o herramientas externas.

Tiene también posibilidades de refactorización de código JavaScript, y bastantes extensiones (aunque ni de lejos las de VScode) pero quitando dos o tres (Emmet, Git o Beautify, por ejemplo) no merece la pena instalarlas ya que su virtud está precisamente en lo que trae de serie para principiantes.

Cloud: JSBin - Editor online para pruebas, experimentos y compartir ejemplos



En la actualidad es casi tan importante escribir código como compartirlo. Por eso, los editores "en la nube" se usan cada vez más. No los usarás para crear proyectos completos (aunque en teoría podrías) sino más bien para usarlos desde cualquier lugar para pruebas rápidas, experimentos y compartir código con otros desarrolladores. Esto último es especialmente útil si estás comenzando.

Aunque mi favorito en general es [repl.it](#) porque además soporta muchos lenguajes diferentes, en este caso debo centrarme en alguno especialmente pensado para código Front-End con HTML, CSS y JavaScript. Dentro de esta categoría existen muchos, siendo quizá los más conocidos [JSFiddle](#) y [Codepen](#). Sin embargo el que te recomiendo hoy, [JSBin](#), es uno mucho menos conocido y con un aspecto más espartano. Sin embargo, aunque los tres son buenos, creo que es el que seguramente ofrece las mejores características en conjunto:

- Permite utilizar multitud de preprocesadores, HTML (Markdown, Jade), CSS (Sass, Less, Myth, Stylus), JavaScript (TypeScript, Babel, LiveScript, Coffeescript).
- Puedes añadir, simplemente seleccionándolas de una lista, decenas de bibliotecas JavaScript populares, como jQuery, Bootstrap, QUnit, Angular, React, Vue.js... más que en el caso de los otros (aunque siempre puedes añadir más manualmente).
- Puedes crear plantillas de las que partir en lugar de unos documentos en blanco. También arrastrar archivos directamente desde tu disco.
- Compartir es muy fácil gracias a sus [URLs "pronunciables"](#)

- Tiene la posibilidad de hacer "[Code Casting](#)", es decir, empezar a "emitir" todo lo que haces en el editor mientras escribes tu código y que otra gente se suscriba y lo vea en tiempo real para aprender o para ayudarte mientras habláis a través de Skype o similar.
- Lo puedes [descargar e instalar localmente](#) si no quieres usarlo (sólo) online.

Tal vez JSFiddle es más conocido y más bonito y tiene algunas características muy chulas, pero JSBin me ha conquistado por su sencillez y porque tiene algunas características que lo hacen único. Cualquiera de los dos que elijas será una gran decisión.