

## Instalación y configuración de un entorno de trabajo JS



Para poder trabajar con JavaScript (JS) es necesario utilizar herramientas de edición de texto. Estas pueden ser muy básicas como el Notepad en Windows, editores de código como SublimeText y otro pueden ser IDE adaptados al lenguaje como Eclipse.

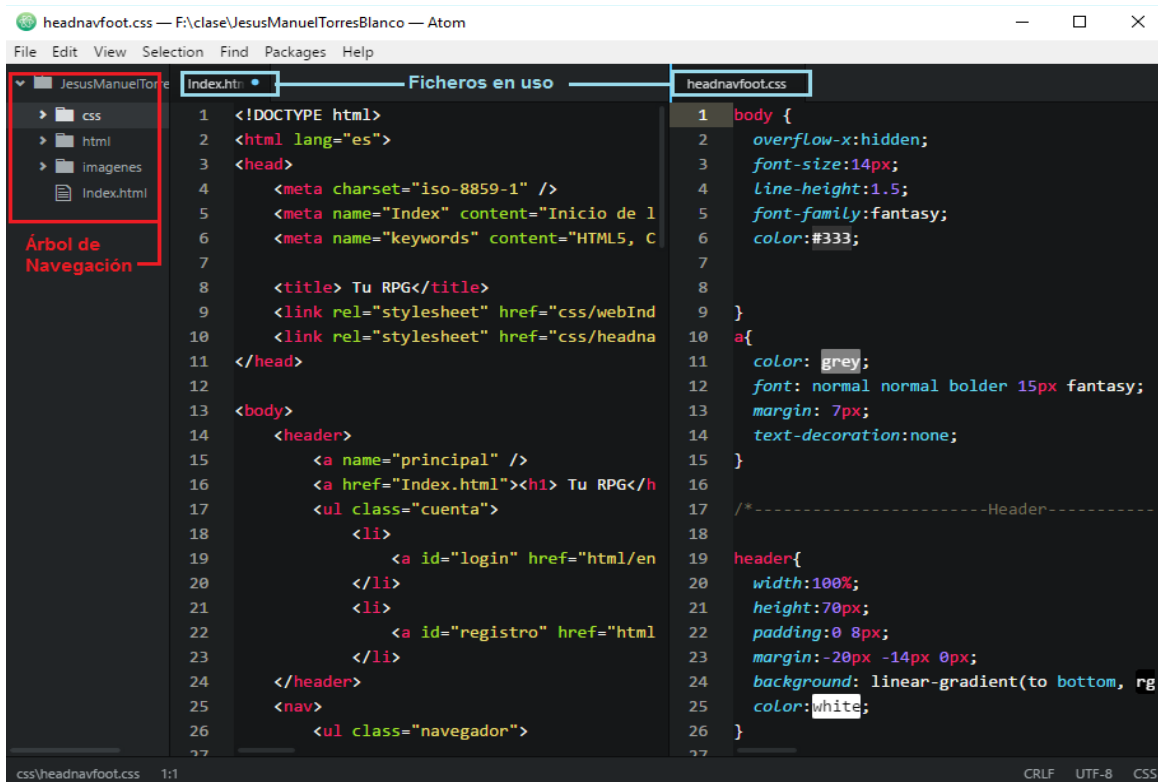
En este documento vamos a ver 3 entornos de desarrollo diferentes para trabajar con JS.

- **Atom**

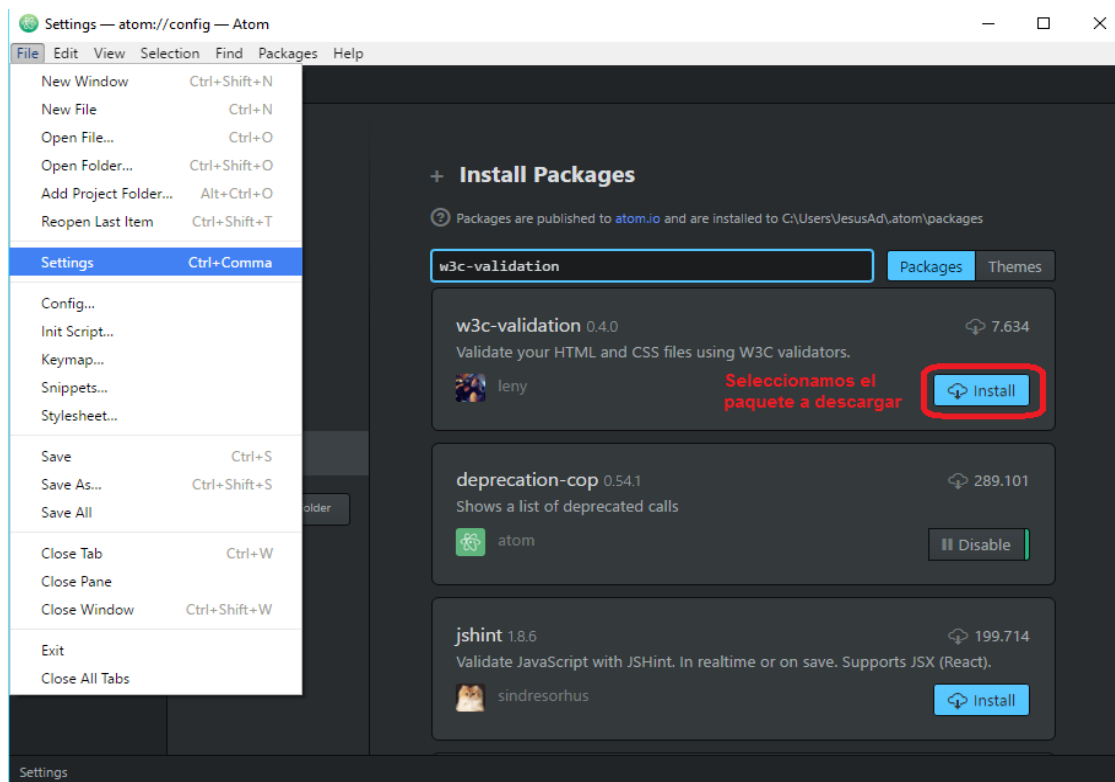


Atom es un editor de código de código abierto para Windows, Linux y OS X. Desarrollado por GitHub, el cual está desarrollado mediante Electron (un framework que utiliza Chromium y Node.js). Además de ser portable su principal característica es la personalización que tiene mediante paquetes.

Atom tiene por defecto algunos paquetes preinstalados que te permiten programar en varios lenguajes con su herramienta de autocompletado. También dispone agrupación del código, emparejamiento de llaves y atajos predefinidos en el editor (Buscar, Sustituir, Ir a línea...).



También dispone de paquetes externos creados por la comunidad que te permiten añadir atajos para programar en distintos lenguajes, personalizar el entorno, selector de colores hexadecimal, etcétera.



Realizado por: Jesús Manuel Torres Blanco

Para instalar estos paquetes en la interfaz de Atom seleccionamos File>Settings y nos dirigimos a la pestaña Install. Desde el buscador escribimos el paquete que deseamos y lo instalamos.

Como recomendación de paquetes he de destacar:

- Emmet – plugin que te permite realizar escritura rápida de archivos html, css y otros lenguajes estructurados.
- Color-picker – plugin para escribir el código de color sin recurrir a herramientas externas.
- atom-beautify – plugin para tabular el código.
- w3c-validation – plugin que te permite validar tu código como en la página <https://validator.w3.org>

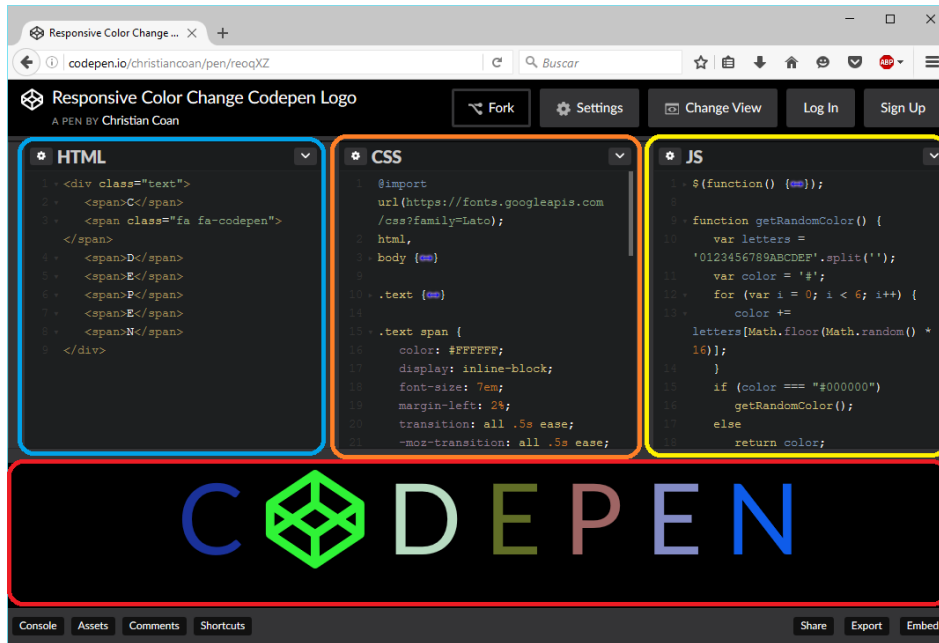
### ¿Por qué usar Atom?

La gran cantidad de paquetes personalizables de Atom le da mucha utilidad en varios sentidos, desde la edición del entorno hasta la implementación de distintos frameworks, siendo una herramienta open source.

- **CodePen**



Codepen es una plataforma de edición de código online creada por Chris Coyier el cual se despliega desde navegador y permite el testeo de html, css y js. Su interfaz es muy sencilla y agradable para el desarrollador. Destaca por no necesitar instalación y la gran cantidad de proyectos distribuidos por la comunidad. Además también dispone de herramientas para añadir recursos externos como librerías o frameworks.



Como muestra la imagen anterior se puede diferenciar los distintos archivos html (azul), css (naranja) y js (amarillo) y el entorno donde se visualizará el resultado (rojo).

Aunque solamente incluye los lenguajes html, css y js, Codepen dispone de Emmet ya integrado permitiéndonos escribir de forma rápida nuestro código. No dispone de chequeo de la sintaxis ni de autocompletado. Una de las herramientas más útiles es la Tidy que nos permite indentar y organizar el código. También incluye una propiedad que permite añadir los perfijos de archivos css.

Otra propiedad es la posibilidad de exportar nuestro trabajo en .zip o a un repositorio Github.

De momento CodePen es soportado por los navegadores más distribuidos (Chrome, Firefox, Edge...).

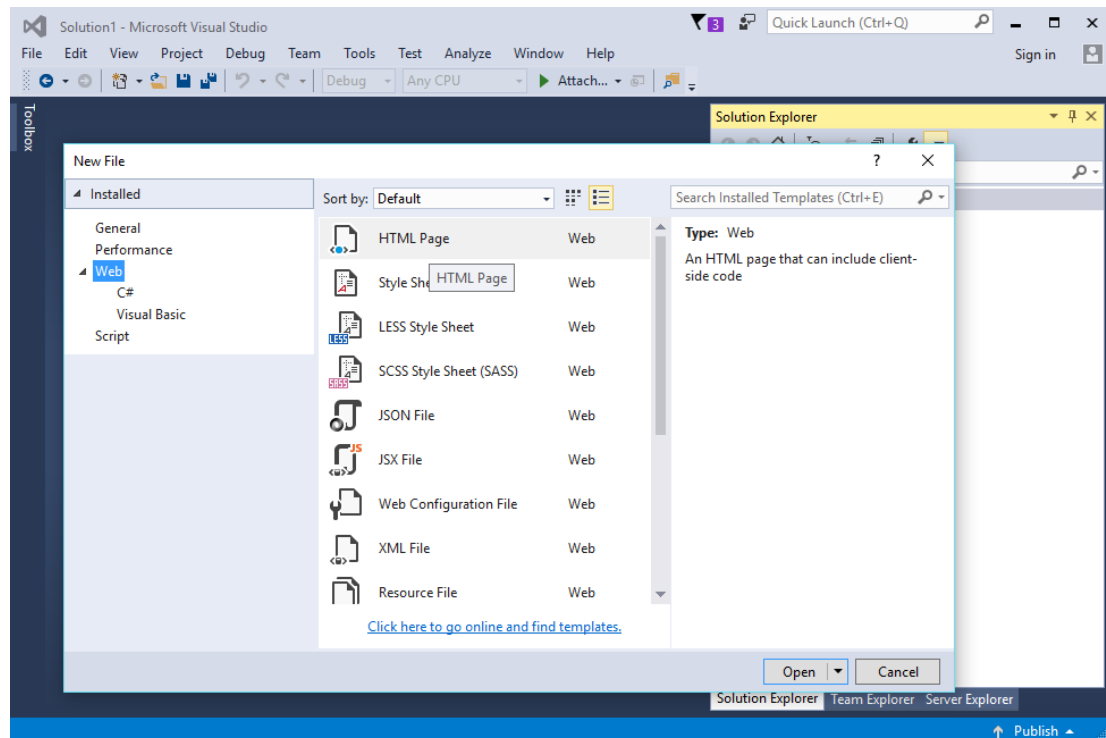
## ¿Por qué usar CodePen?

Siempre que se disponga de conexión a red y no tengamos un editor de código CodePen es una buena opción. La gran cantidad de ejemplos subidos a la plataforma y la posibilidad de exportar nuestros proyectos sin la necesidad de instalar un programa es el punto fuerte de esta plataforma.

- **Visual Studio**

Visual Studio (VS) es el IDE gratuito de Microsoft para sistemas operativos de Windows. VS se puede usar como entorno de desarrollo para nuestras páginas web.

Después de descargar VS desde la [página oficial de Microsoft](#) e instalarlo, podremos trabajar con el de forma sencilla ya que nos crea de forma automática la estructura básica de una página html.



Uno de los fuertes de VS es que permite programar en múltiples lenguajes, nos da una herramienta de autocompletado, permite el agrupamiento y formateo del código así como el chequeo de la sintaxis del lenguaje, el emparejamiento de llaves y atajos del teclado.

Uno de los puntos fuertes de VS es el continuo soporte que tiene y la optimización de sus frameworks.

### ¿Por qué usar Visual Studio?

Microsoft es uno de los grandes de la industria por lo que garantiza un producto de calidad y un buen soporte. Tiene una gran potencia como IDE para desarrollo web. Aunque de momento está para sistemas operativos Windows con VS puedes realizar aplicaciones para otros sistemas así como dispositivos móviles.

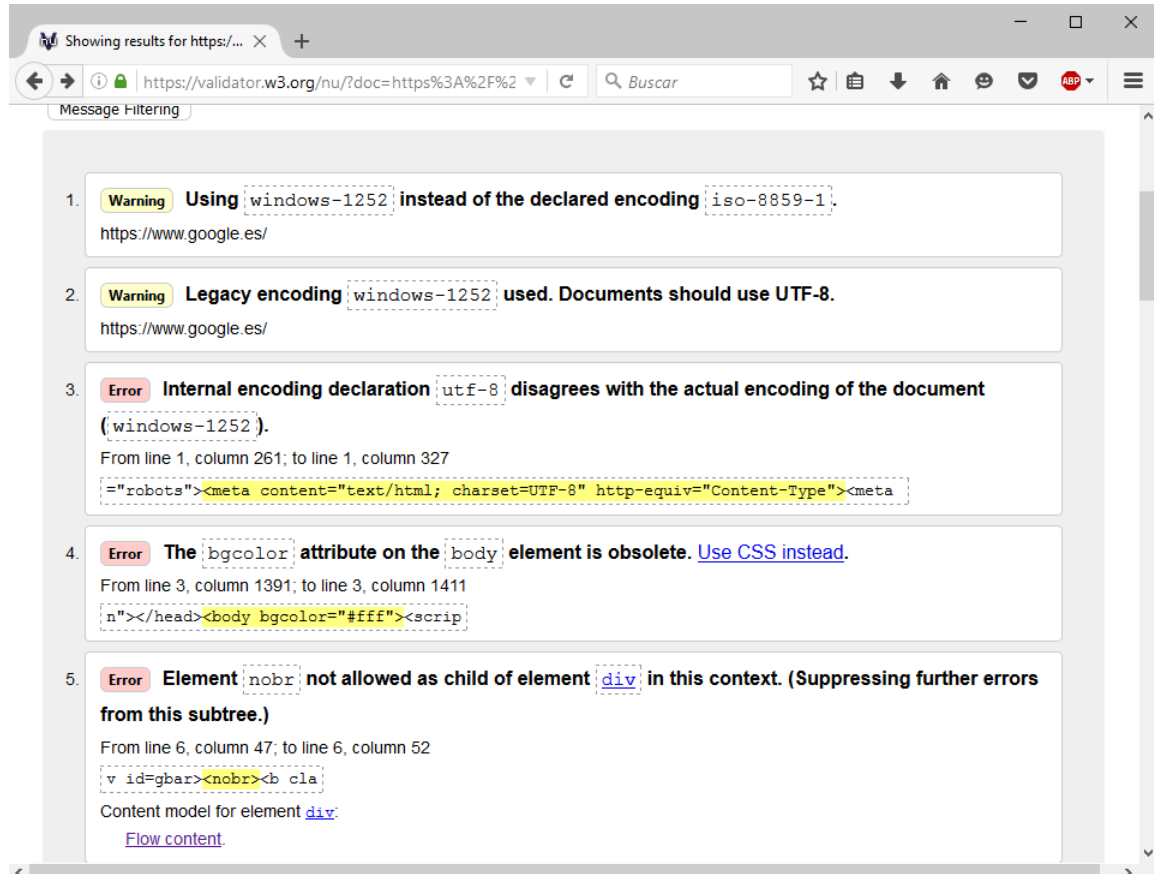
## Conclusiones:

Como podemos ver existen entornos muy diferentes dependiendo de la situación en la que se encuentre el desarrollador, algunas limitadas por el sistema operativo pero muy potentes como Visual Studio, otras más personalizables mediante paquetes externos o creados por la comunidad como Atom y otros los cuales solo hace falta un navegador para poder programar nuestro sitio web y publicarlos en la red como CodePen.

Aunque existen cientos de entornos lo recomendable es escoger el que mejor se adapte a nuestras necesidades y más nos guste. No por escoger el mas complejo vamos a ser mejores programadores, ni por negarse a utilizar distintos entornos va a avanzar, en resumen uno tiene que **adaptarse** a las distintas tecnologías con lo que lleva también a adaptarse a distintos entornos.

## Uso de W3C Validator

Usando la dirección de [validación del W3C](https://validator.w3.org/), validaremos la página principal de Google.



Al validar dicha página nos muestra 2 avisos ( warning ) y 21 errores. ¿Cómo es posible que una página como Google tenga tantos errores? Esto se debe a que existen elementos obsoletos en la misma, errores de formato de codificación y elementos no permitidos en algunas etiquetas, pero aún así la página seguirá funcionando con normalidad.

Los errores de elementos obsoletos comúnmente son atributos que se pueden definir en un archivo .css así que para eliminar dichos errores solamente habría que hacer referencia a esos atributos en un archivo externo css.

Los errores de formato de codificación ocurren cuando el formato es diferente al definido en los metadatos. En este caso el archivo está definido como UTF-8 y la codificación utilizada es windows-1252. Para arreglar este error habrá que cambiar la información definida en los metadatos UTF-8 a windows-1252.

El último error hace referencia a elementos no permitidos en algunas etiquetas. Esto puede ocurrir porque la codificación puede estar obsoleta y etiquetas que antes tenían un atributo, ya no lo tienen. Estos errores se solucionan suprimiéndolos del html.