

Usamos cookies para optimizar el diseño de nuestro sitio web. Si continua la visita, da su consentimiento para el uso de estas cookies.

✕

[HOME](#)[MASTER](#)[BOOTCAMP BACKEND](#)[BOOTCAMP DEVOPS](#)[BOOTCAMP JS](#)[LEMONCODE TV](#)[EMPRESAS](#)[BLOG](#)[SERVICIOS](#)[CONTACTO](#)[ACCESO
CAMPUS](#)

Git y Visual Studio Code

by **Braulio Diez** | **33 Likes**

VSCode y Git

En estos últimos años unos han manejado Git desde la consola (línea de comandos), otros han tirado de herramientas gráficas independientes como SourceTree o GitKraken, todos huyendo un poco de los IDE's integrados que al final te hacían cliente cautivo.

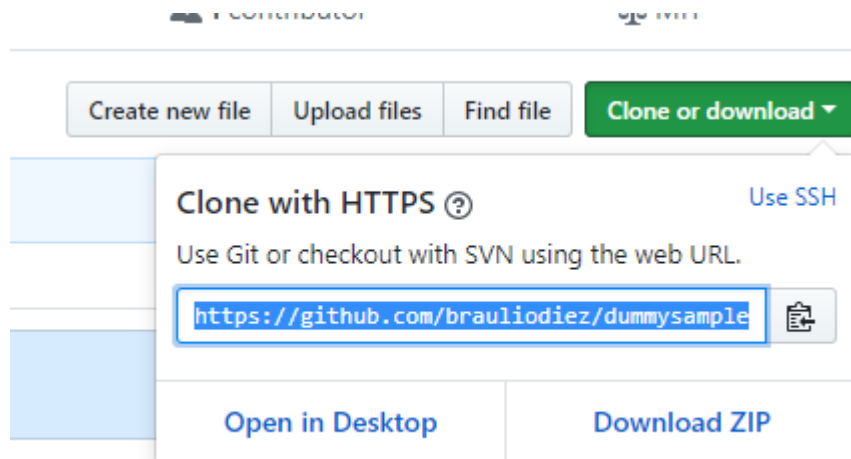
Se un tiempo a esta parte se ha levantado *Visual Studio Code*: un editor ligero, potente, multiplataforma y open source, que va mejorando por semanas y además se ha ganado la simpatía de muchos sectores. Se nota que detrás de esta herramienta hay personalidades del mundo de la programación como Erich Gamma.

En este post vamos a aprender manejarnos con este cliente de Git, para ello nos clonaremos un repo de Github que contiene una aplicación web del tipo "Hola Mundo", e iremos trabajando con ella. Cada paso irá acompañado de gifs animados que nos harán más fácil aprender a como usar esta herramienta.

En este artículo asumimos que ya tienes conocimientos básicos de Git, si necesitas aprender fundamentos de Git tenemos este [curso](#) en español disponible, si eres de lo que les gusta leer, este [libro](#) (disponible online de forma gratuita) está muy bien.

Clonar un repo

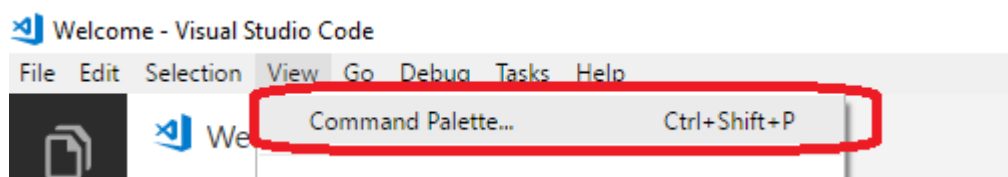
Arrancamos por clonar un repo para ello copiaremos la URL del repo (en este caso tenemos el repo en github pero funcionaría desde bitbucket, gitlab, vsts u otro).



Vamos a clonar un proyecto hospedado en GitHub.

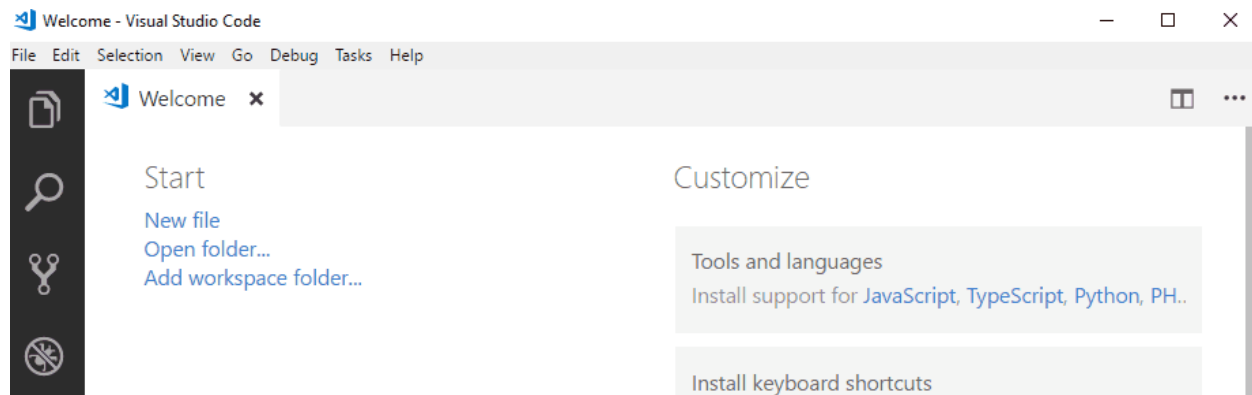
Para clonar un repo, lo podemos hacer desde la paleta de comandos.

- Para mostrar la paleta de comandos: podemos pulsamos la combinación de teclas **CTRL+SHIFT+P** o desde el menú view seleccionamos "command palette".



Mostrar la paleta de comandos (CTRL+SHIFT+P)

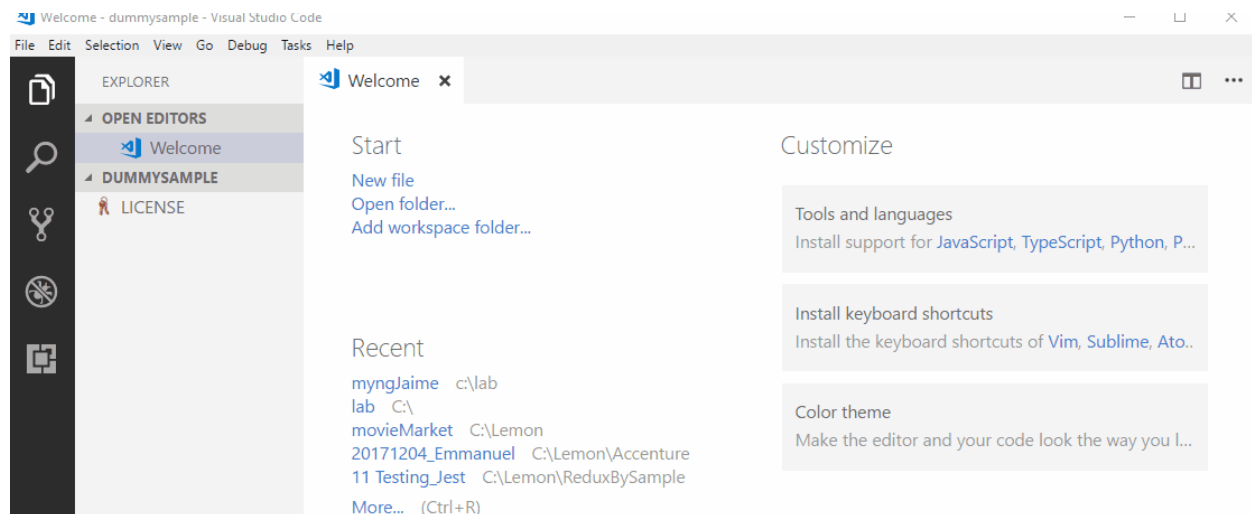
- Desde ahí tecleamos *git clone* nos pide la ruta del repositorio y ya ponemos clonar (si es necesario nos pide credenciales).



Clonando un repo

Introduzco cambios, commit

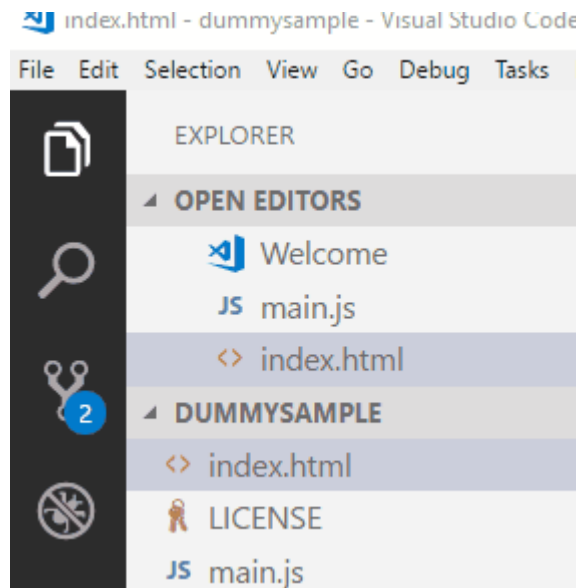
Conforme vamos introduciendo cambios en los ficheros estos se nos marcan como modificados o añadidos.



Los ficheros se van marcando conforme modificamos

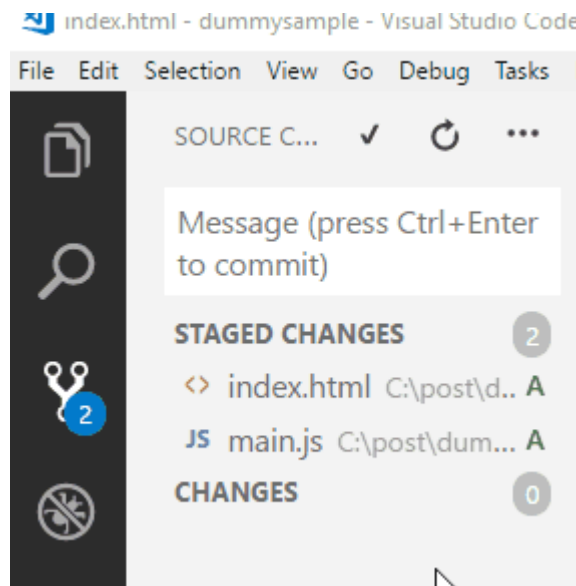
Cuando queramos subir cambios nos vamos a la pestaña de git y vemos de que ficheros queremos hacer stage (podemos ir pulsando en "+" fichero a ficheros o en todos,

también podemos directamente darle a commit y decirle que haga stage de todos los ficheros).



Añadiendo a stage

Una vez que estamos listos le damos al botón de commit (esto sólo guarda en local, tenemos que hacer un push para llevar los cambios a servidor).

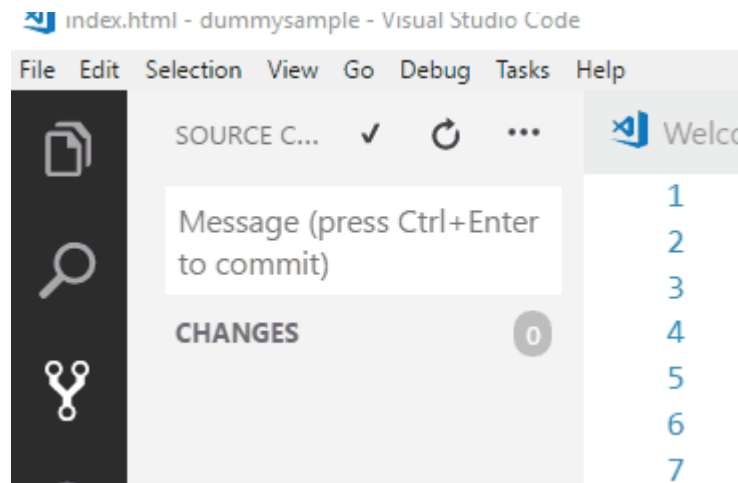


Commit

Hacer push de los cambios

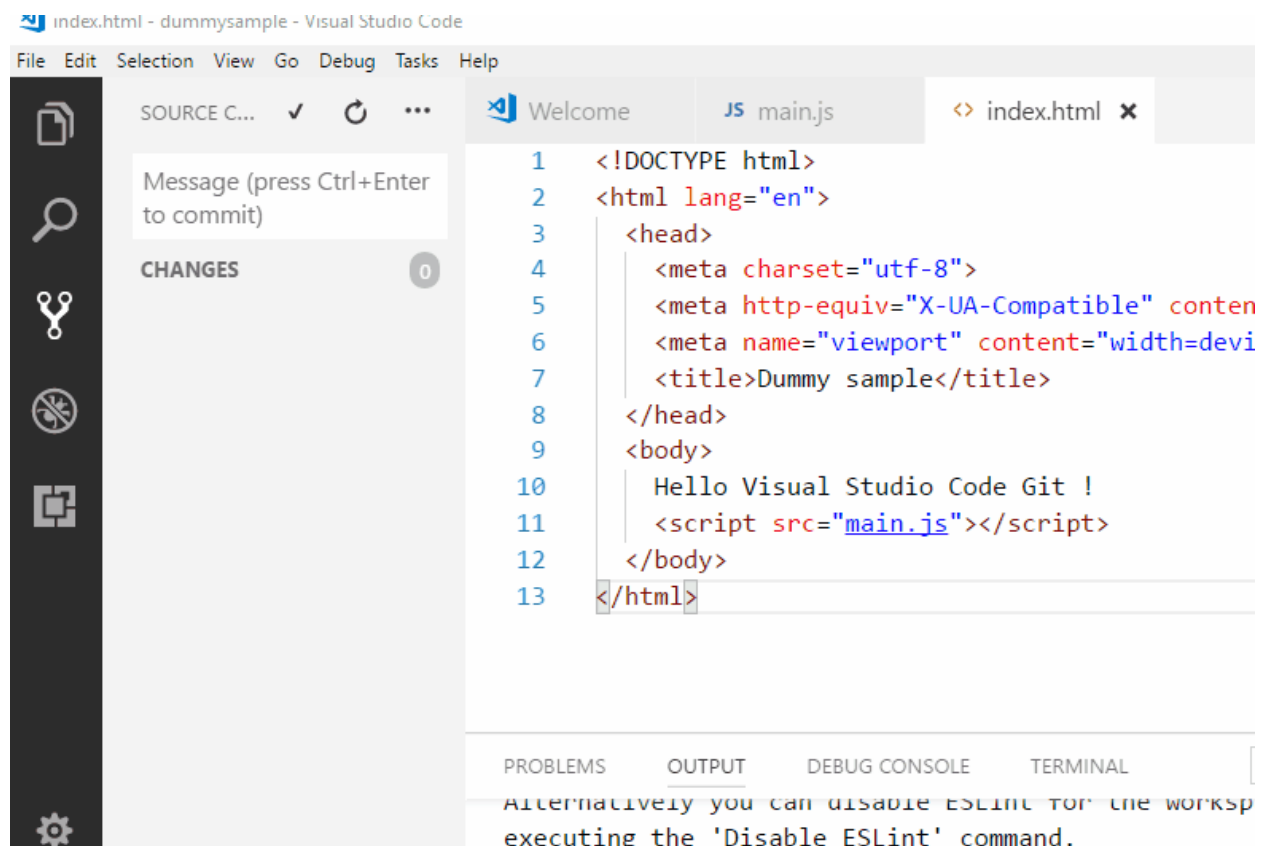
Para hacer push de los cambios podemos hacerlo de dos maneras:

- O bien pulsando en la parte derecha de la barra de git (icono tres puntos) y eligiendo la opción *push*



Push

- O bien en la parte inferior pulsando sobre sincronizar (ojo esto haría push y pull).

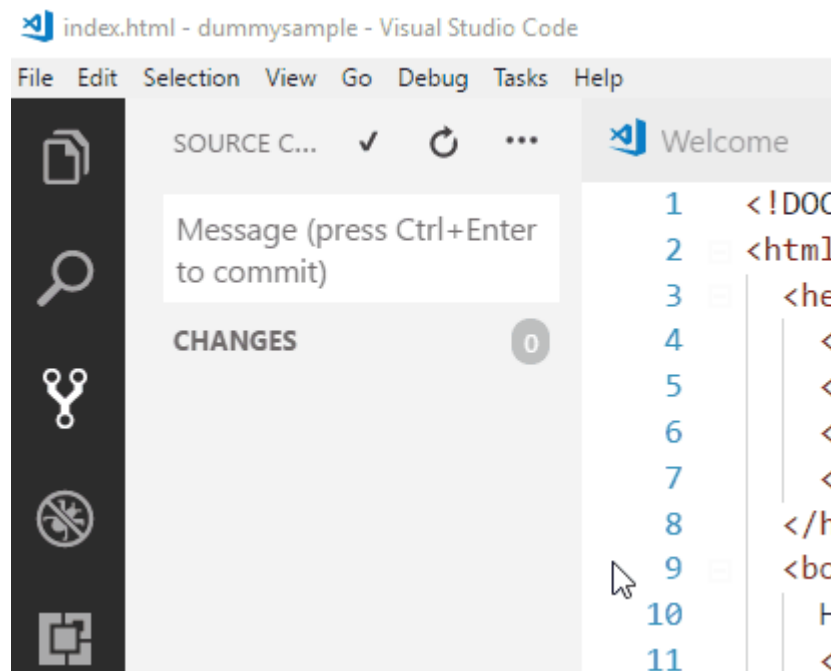




Sincronizar

Como hacer un pull

Para traernos los cambios que haya en servidor de la rama en la que estamos trabajando, podemos pulsar sobre los tres puntos (...) que hay en la parte superior derecha y pinchar sobre la opción *pull*.



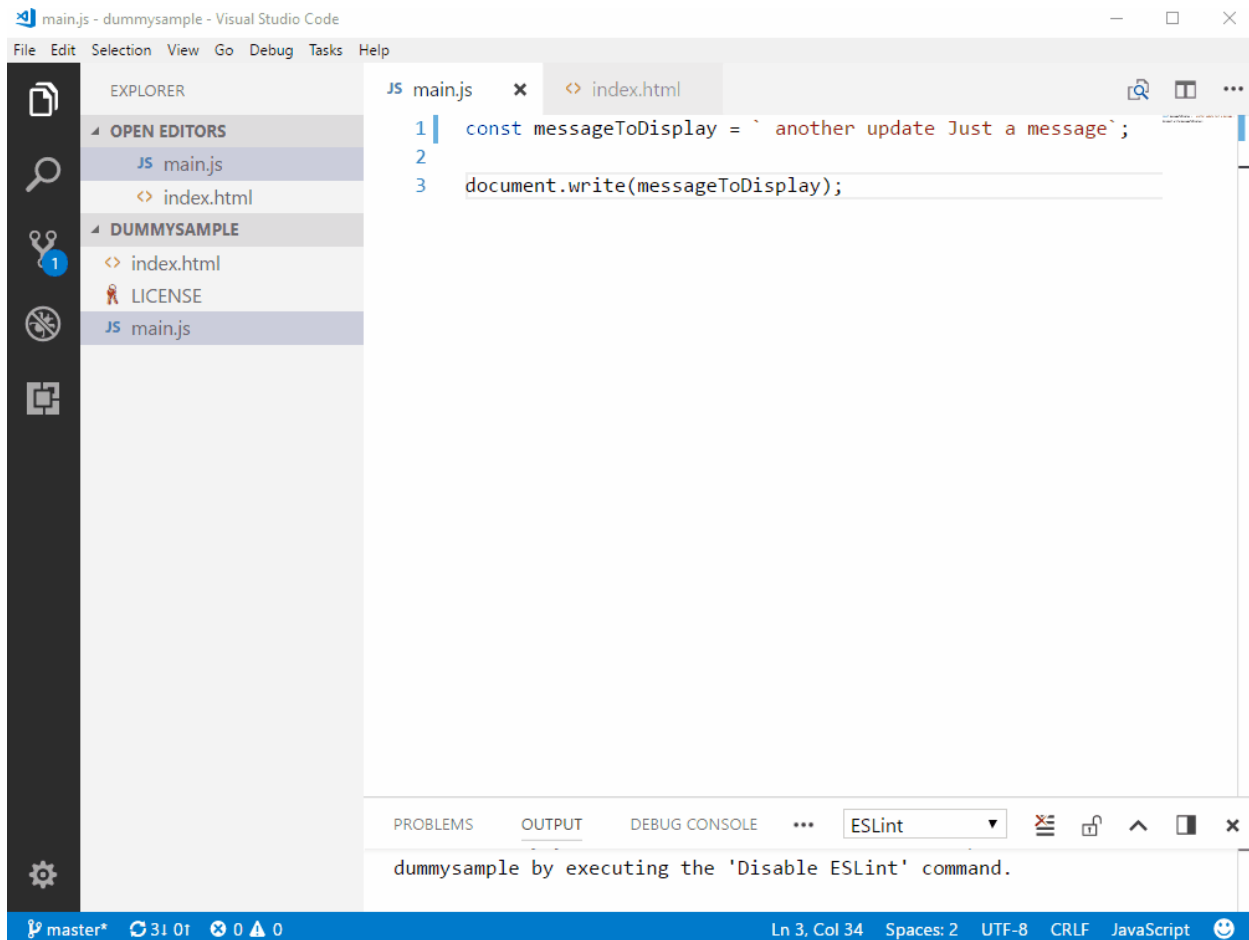
Pull

También podríamos hacer como en el paso anterior (puslando en los iconos de la parte inferior, sincronizar). Está opción la suelo usar mucho, es cómoda.

Sincronizar: Push y pull

Si queremos traernos los cambios últimos que se hayan realizado en nuestra rama y subir los nuestros podemos sincronizar el repo, pulsando sobre el icono inferior de sincronización que están en la parte de abajo de la ventana (si trabajamos con Branch per feature normalmente sólo haremos operaciones de push o de pull aunque le demos a sincronizar).

En el ejemplo que vamos a ver tenemos varios cambios en servidor y un fichero que hemos actualizado en local (por eso nos salen tres commits por bajar y un cambio por subir, cuando pinchamos en el icono baja el cambio pertinente y sube el nuestro).



Sincronizar

Resolver conflictos

¿ Qué pasa si quiero hacer pull de mi rama pero trae cambios y conflictos de servidor?
¿Cómo soluciona esto Visual Studio Code?

En este caso hemos puesto un conflicto en el fichero main.js, tenemos que decidir con que cambio nos quedamos. Para ello:

- Hacemos pull y nos sale el fichero con el conflicto, pinchamos sobre el fichero y vemos donde estan los conflictos.
- Ahora podemos elegir que parte es la que queremos tomar como buena (la nuestra o la de servidor) y la aceptamos:

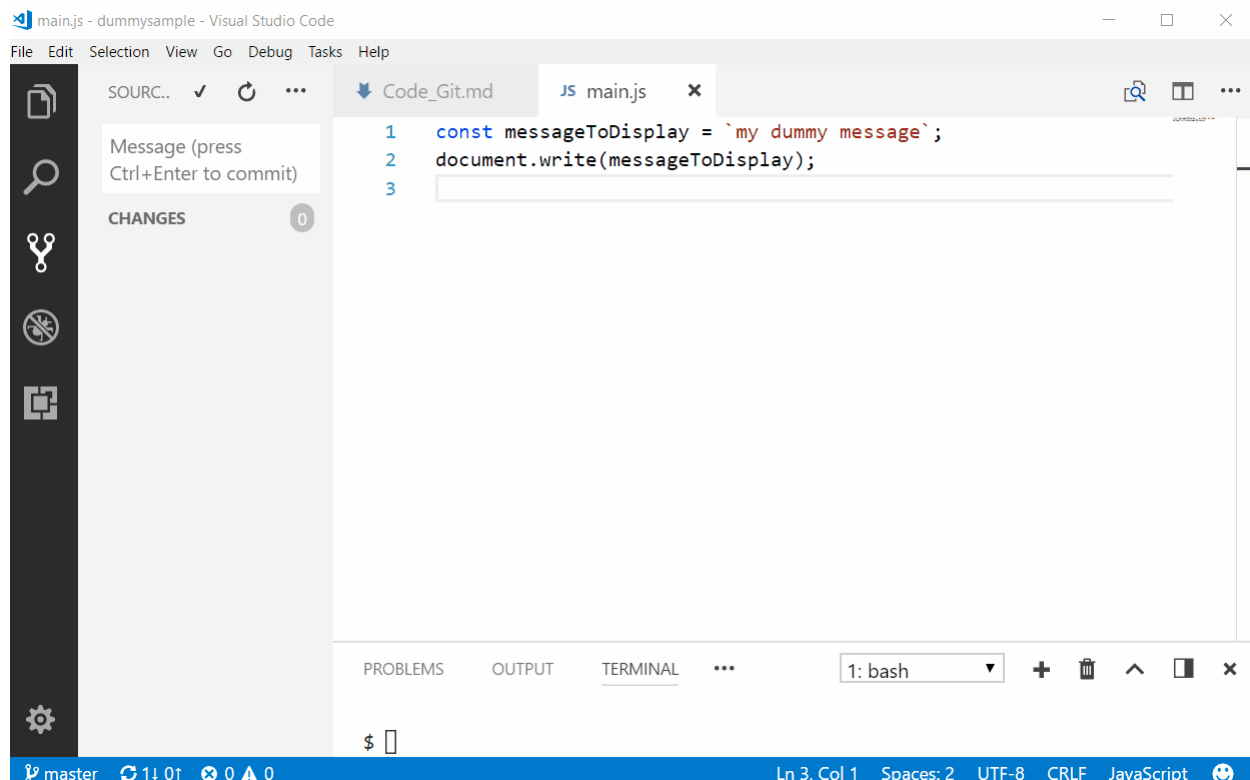
También podemos escribir en una de las partes y aceptar el cambios, o aceptar ambos cambios como buenos:

- Una vez que hemos terminado, tenemos que hacer un commit con el merge, para ellos nos vamos al terminal y ejecutamos la siguiente orden.

```
git commit -m "fixing merge conflicts"
```

- Ya estamos listos para poder hacer push del merge.

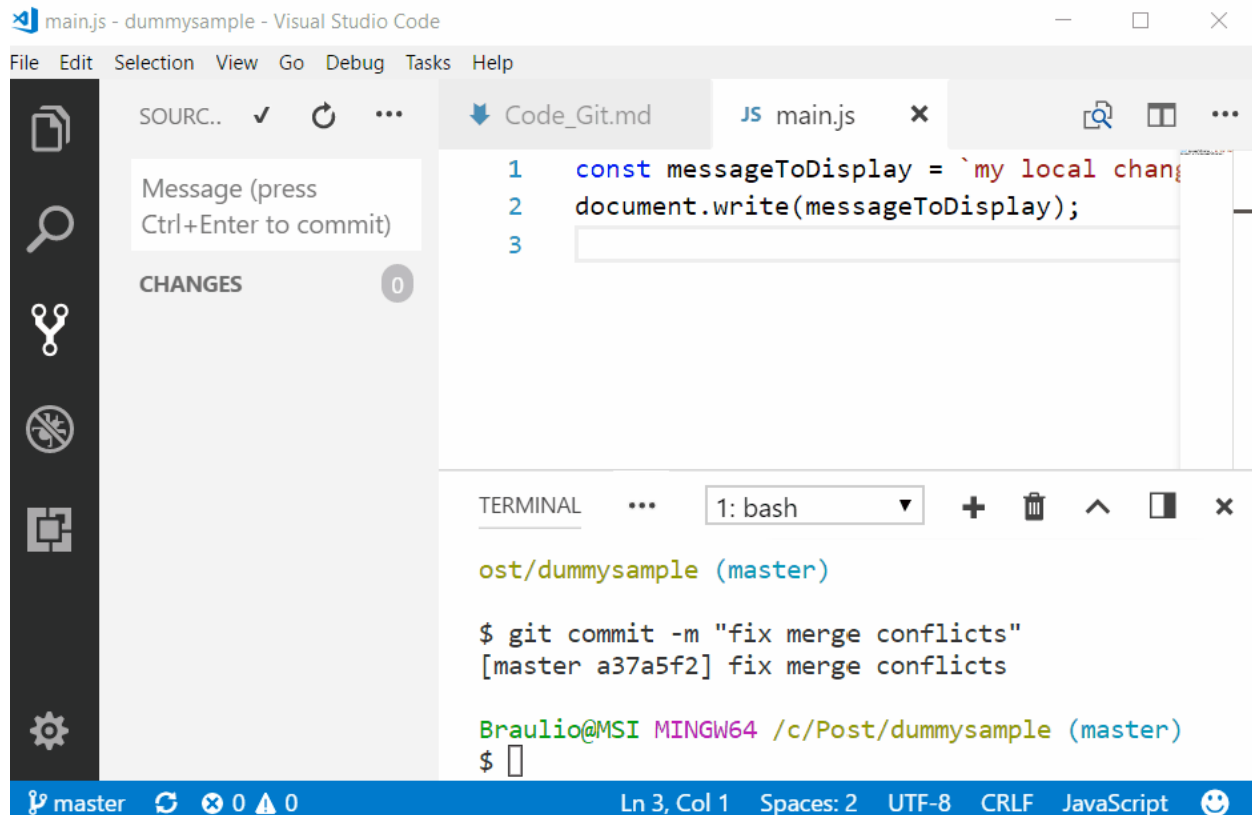
Veamos todos los pasos en acción en el siguiente gif animado:



Resolver conflictos

Creando ramas

Podemos crear una rama mostrando la paleta de comandos (CTRL+SHIFT+P) y eligiendo el comando *git branch*, despues nos preguntará por el nombre de rama. Una vez creada en local, podemos subirla al servidor pulsando sobre el indicador de la nube que aparece en la parte inferior de la ventana.

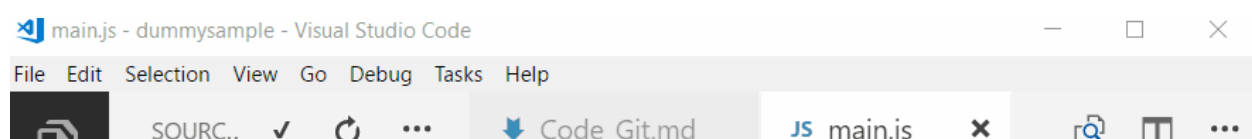


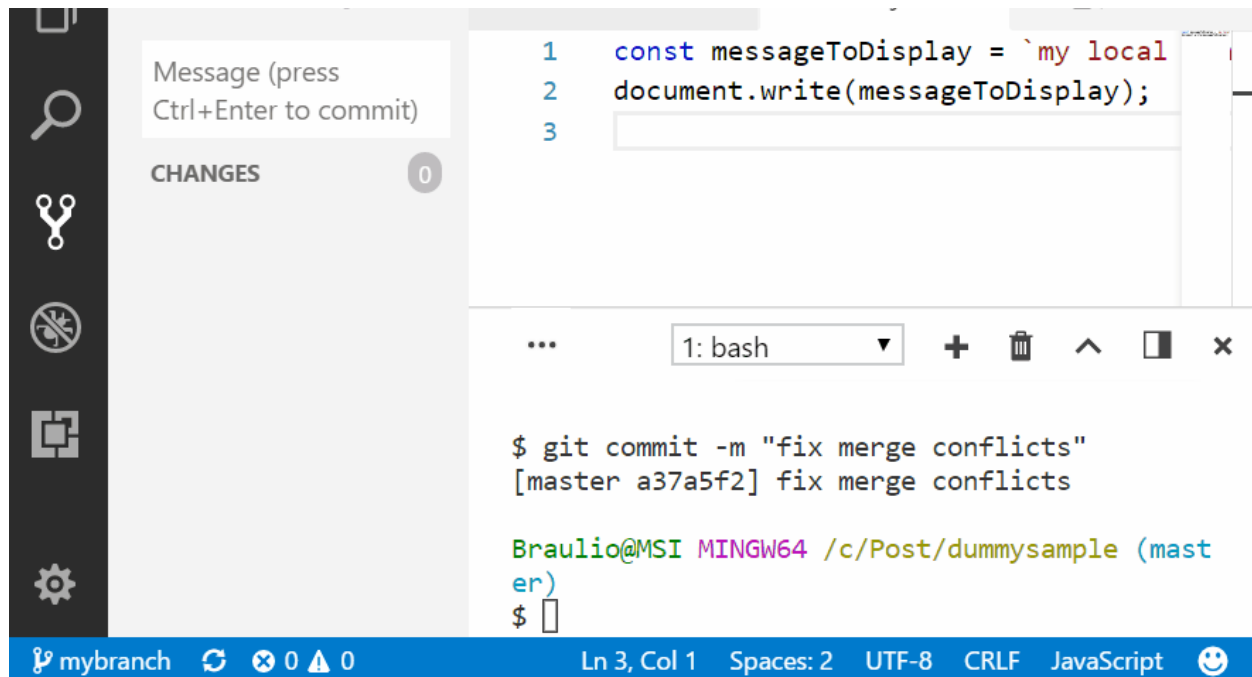
Creando ramas

Cambiar de rama

Para cambiar de rama, pinchamos sobre la parte inferior en la que aparece el nombre de la rama actual, y nos aparece un diálogo en el que podemos teclear el nombre de la rama al que queremos navegar.

Importante para poder cambiar de ramas no debemos de tener ficheros locales con cambios.





Cambiar de rama

¿Qué pasa si la rama a la que quiero hacer checkout no aparece? (no se ha sincronizado con el servidor), puedo teclear desde el terminal:

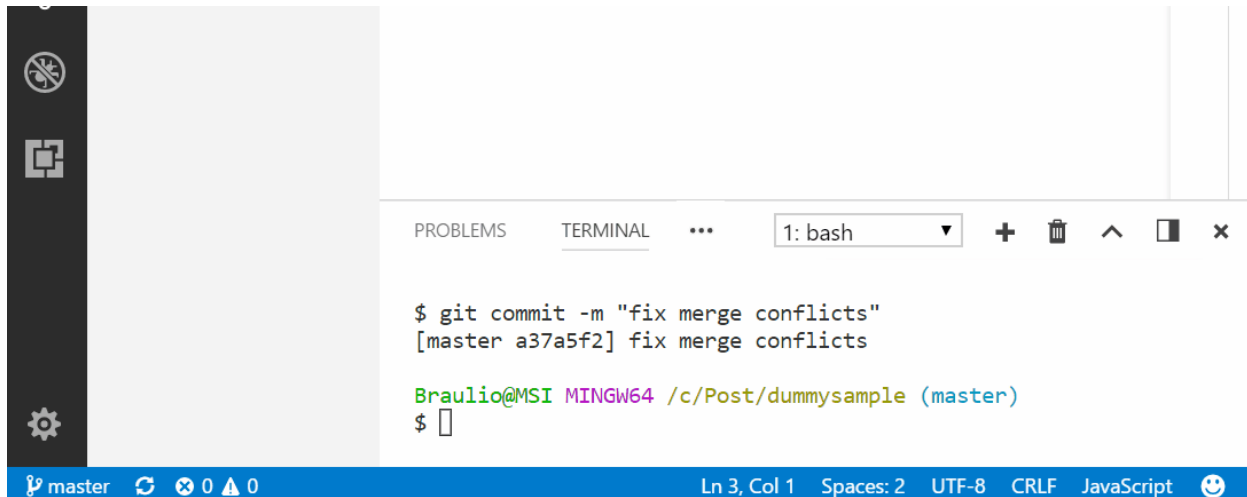
```
git fetch
```

Y ahora si le volvemos a dar a la opción de cambiar de rama, podremos ver la lista actualizada.

Descartar cambios

Para descartar cambios sobre un fichero, desde la ventana de git podemos pinchar con el botón derecho sobre el fichero que queramos descartar y elegimos la opción *Discard*.





A screenshot of the Visual Studio Code interface. The terminal window is open, showing a bash prompt. The user has entered the command `$ git commit -m "fix merge conflicts"`, and the output shows `[master a37a5f2] fix merge conflicts`. The status bar at the bottom indicates the current branch is `master` and the file is `main.js`.

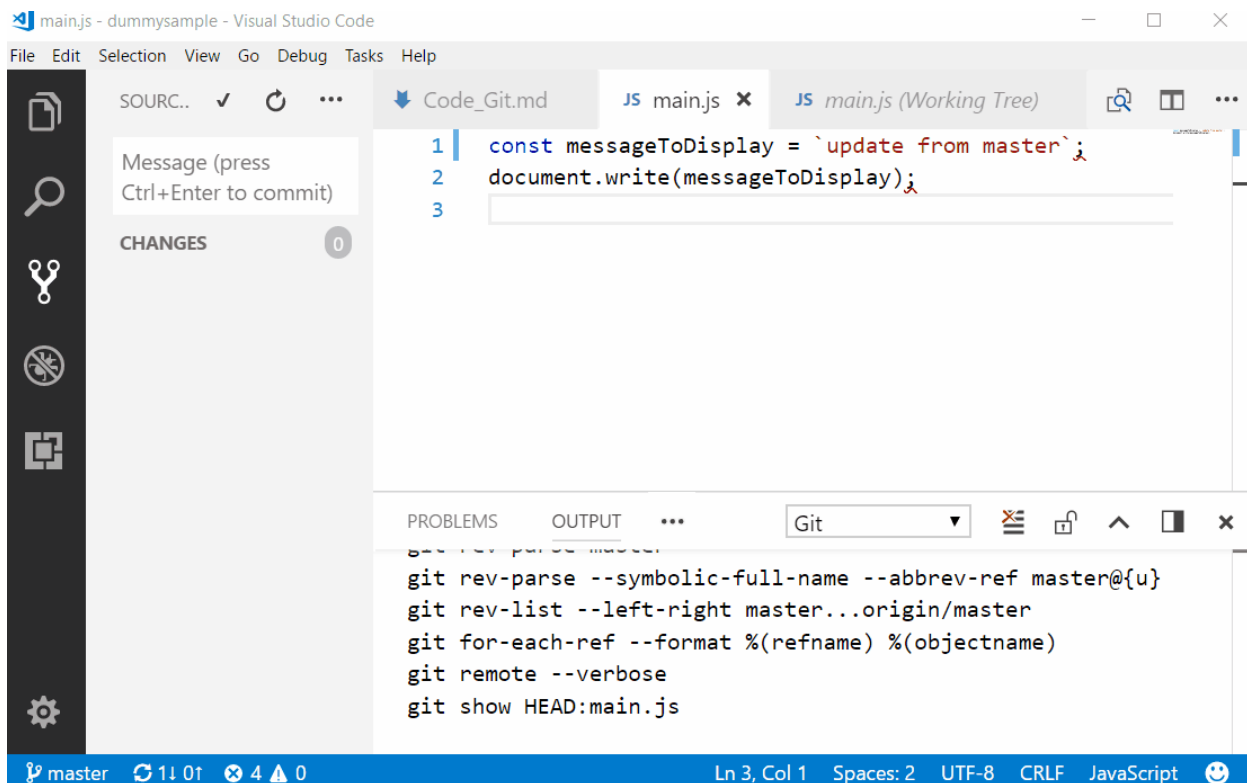
```
$ git commit -m "fix merge conflicts"
[master a37a5f2] fix merge conflicts

Braulio@MSI MINGW64 /c/Post/dummysample (master)
$
```

Descartar cambios

Mezclar máster a mi rama

Supongamos el caso en que mi rama se ha quedado desactualizada con respecto a máster y me quiero traer los cambios (tengo suerte y no hay conflictos), para hacer esto abro la paleta de comandos y tecleo `git merge` con ese comando me pregunta que rama quiero mezclar a la actual, tecleo su nombre y realizo el merge.



A screenshot of the Visual Studio Code interface. The editor shows a file named `main.js` with the following content:

```
1 | const messageToDisplay = `update from master`;
2 | document.write(messageToDisplay);
3 |
```

The terminal window is open, showing the output of the `git merge` command. The output indicates that the merge was successful and the working tree is now up-to-date with the master branch.

```
git merge master
git rev-parse --symbolic-full-name --abbrev-ref master@{u}
git rev-list --left-right master...origin/master
git for-each-ref --format %(refname) %(objectname)
git remote --verbose
git show HEAD:main.js
```

Mezclar master a mi rama actual

Si tuvieramos conflictos haríamos como en el caso anterior (arreglar conflictos en misma rama).

Conclusiones...

VSCode trae un cliente de Git ligero y prácticos, a caballo entre una herramienta de interfaz de usuario y tirar de línea de comandos.

Espero que te haya gustado esta guía y que vuelvas de vez en cuando a revisar ese paso del que no te acordabas :-).

¿Con ganas de aprender desarrollo Front End?

Si tienes ganas de ponerte al día en el mundo del Front End impartimos un máster online con clases en vivo, más información: <http://lemoncode.net/master-frontend>

↩ Share

Comments (17)

Newest First

[Preview](#)[POST COMMENT...](#)

Cesar A month ago · 0 Likes

Muy bien explicado, gracias



Emanuel A year ago · 0 Likes

Gracias.



Joselo A year ago · 0 Likes

la mejor explicacion que encuentre! gracias gracias



Jose Luis Diaz A year ago · 0 Likes

Bien. Claro y conciso. Genera mas articulos como este!!!



George 2 years ago · 0 Likes

Muchas gracias por tan excelente articulo muy preciso y completo felicidades, expones en unas pocas lineas y con ejemplo gráfico lo que se utiliza en el día a día



Cam 2 years ago · 0 Likes

¡Buenísimo!, Me ayudó mucho, ¡Gracias! :)



Arturo 2 years ago · 0 Likes

Que buen tutorial, de crack ...



Rubén Alejandro Ramirez 3 years ago · 0 Likes

Buenas, si llegaron aquí con el mismo problema que tenía yo, de que Visual Studio Code no tenía la bendita integración con GIT, les cuento que hay que instalar la extensión oficial como indica el sitio de git <https://git-scm.com/book/en/v2/Appendix-A%3A-Git-in-Other-Environments-Git-in-Visual-Studio-Code>

Luego de instalarla, esta genial y detallada publicación que leyeron más arriba empieza a tener sentido, saludos

<https://marketplace.visualstudio.com/items?itemName=GitHub.vscode-pull-request-github>



Kevin 3 years ago · 0 Likes

Muchas gracias por el artículo, muy completo!



José Luis 4 years ago · 0 Likes

Gracias, me sirvió bastante



Braulio Diez 4 years ago · 0 Likes

Muchas gracias, sólo añadir que mucho cuidado al deshacer cambios, en el caso de que los tengamos en local (commit) y no los hayamos subido a servidor (push), no hay problema. Si ya los hemos subido a servidor, si hay que meterse en consola de comandos, y si encima hay otros usuarios que han partido de ese cambio subido, hay que tener mucho cuidado.



Cristian 2 years ago · 0 Likes

Disculpa, yo tenia unos cambios por subir con un add y commit. Pasa que por accidente le doy a "Descartar todos los cambios" y se me fue todo al carajo. Qué puedo hacer?



Braulio A year ago · 0 Likes

Uff aquí solo creo que confiar en el CTRL+Z del editor :(, aquí tienes por qué y que cosas puedes hacer con un editor:
<https://stackoverflow.com/questions/43541167/how-do-you-undo-discard-all-changes-in-vs-code-git#:~:text=After%20executing%20the%20command%20git,the%20version%20>



Locknar 4 years ago · 0 Likes

Me autocontesto.

Para deshacer un commit hay que pulsar en los puntos suspensivos en la parte superior a la derecha y escoger la opción: "Deshacer última confirmación". Esto permite deshacer el último commit que se haya hecho en caso de que detectéis algún error después de haberlo hecho.

Saludos y gracias de nuevo.

Por cierto, el artículo está inmejorablemente explicado. Enhorabuena.



Braulio A year ago · 0 Likes

Perdona no vi el comentario anterior, he puesto una respuesta, siempre que no hagas push se pueden controlar, una vez que has hecho push tienes que tener cuidado



Locknar 4 years ago · 0 Likes

Una pregunta.

Una vez pulsado el botón de commit y guardados los cambios en local. Si detectas un error, ¿es posible deshacer ese commit o hay que hacer nuevos cambios en

los ficheros afectados y volver a pulsar commit?

Gracias.

Un saludo.

**Braulio**

A year ago · 0 Likes

Si no has hecho push, puedes hacer dos cosas, rebobinar la cabeza y que apunte al commit anterior, o hacer un commit de revert, todas estas cosas las tienes que hacer con cuidado, buena idea para otro post :)

Si ya has hecho push pero nadie ha hecho push sobre esa rama, puedes tirar para atras ese commit o hacer revert, pero si alguien ya ha hecho push sobre esa rama, es un tema delicado.

[Newer](#) / [Older](#)

[HOME](#)[MASTER](#)[BOOTCAMP DEVOPS](#)[BOOTCAMP JS](#)[WEBINARS GRATUITOS](#)[EMPRESAS](#)[BLOG](#)[SERVICIOS](#)[CONTACTO](#)[CONDICIONES GENERALES](#)