

DAM

Contornos de
desenvolvimento

UD2 **INSTALACIÓN Y
USO DE ENTORNOS DE
DESARROLLO:
GIT EN VISUAL STUDIO CODE**

PROFESORA | CRISTINA PUGA FDEZ

UNIDADE | UD2

TRIMESTRE | 1

MÓDULO | CD

Tabla de contenido

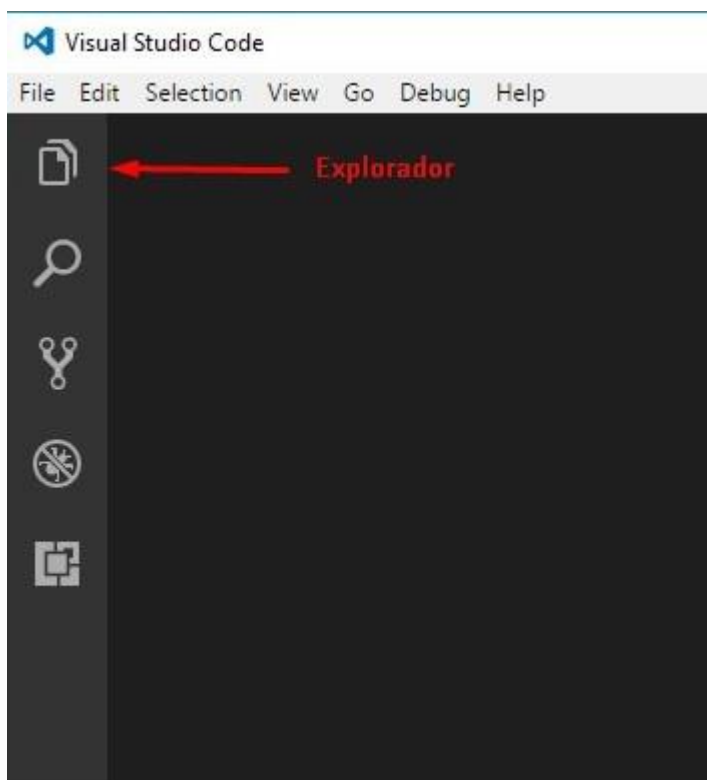
1.	VISUAL STUDIO CODE	3
1.1	Herramientas importantes	3
1.2	Ver iconos	5
1.3	Temas.....	7
1.4	Instalar extensiones.....	8
1.5	Buscar y reemplazar	10
1.6	Comentar código	12
1.7	Grupos.....	13
1.8	Proyecto java	14
1.9	Depuración.....	15
1.10	GIT	16
1.10.1	Clonar repositorio	16
1.10.2	Commit	17
1.10.3	Push	18
1.10.4	Pull	19
1.10.5	Resolver conflictos.....	19
1.10.6	Creando ramas.....	21
1.10.7	Cambiar de rama	21
1.10.8	Descartar cambios	22
1.10.9	Merge con master.....	23
	ANEXO I: Materiales	23

1. VISUAL STUDIO CODE

Primero debemos bajarnos los ejemplos de:

<https://github.com/johnpapa/pluralsight-vscode-samples>

1.1 Herramientas importantes



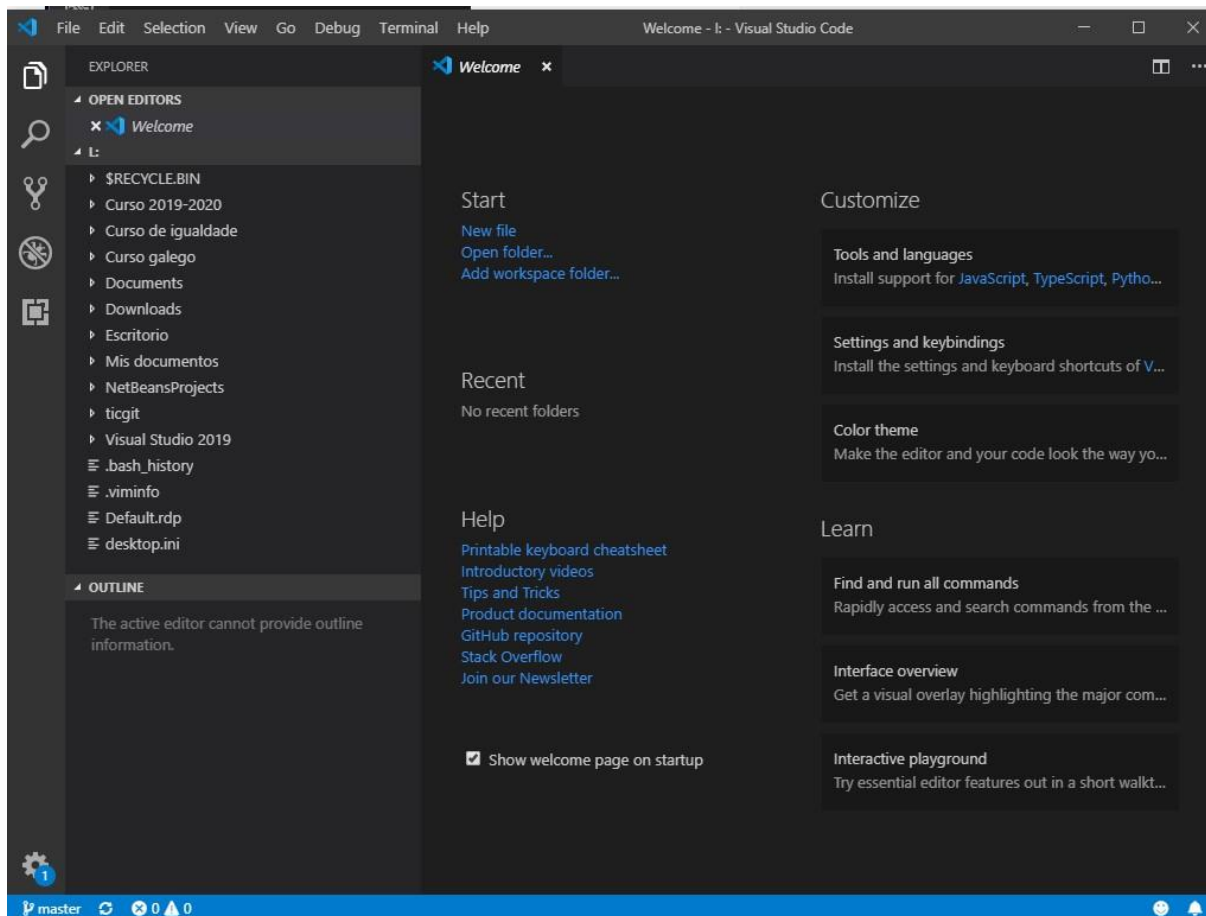
Es importante saber que Visual Studio Code trabaja con carpetas. Si abrimos code desde el terminal:

```
C:\> Command Prompt
Microsoft Windows [Versión 10.0.18362.239]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

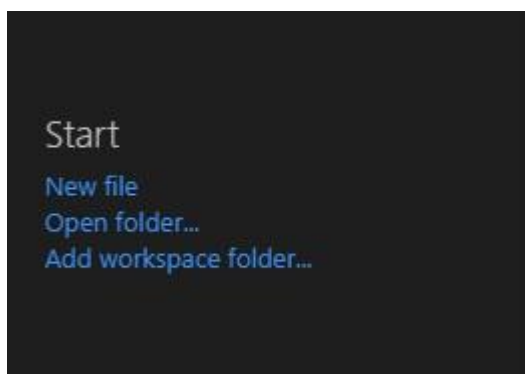
L:\>code .

L:\>
```

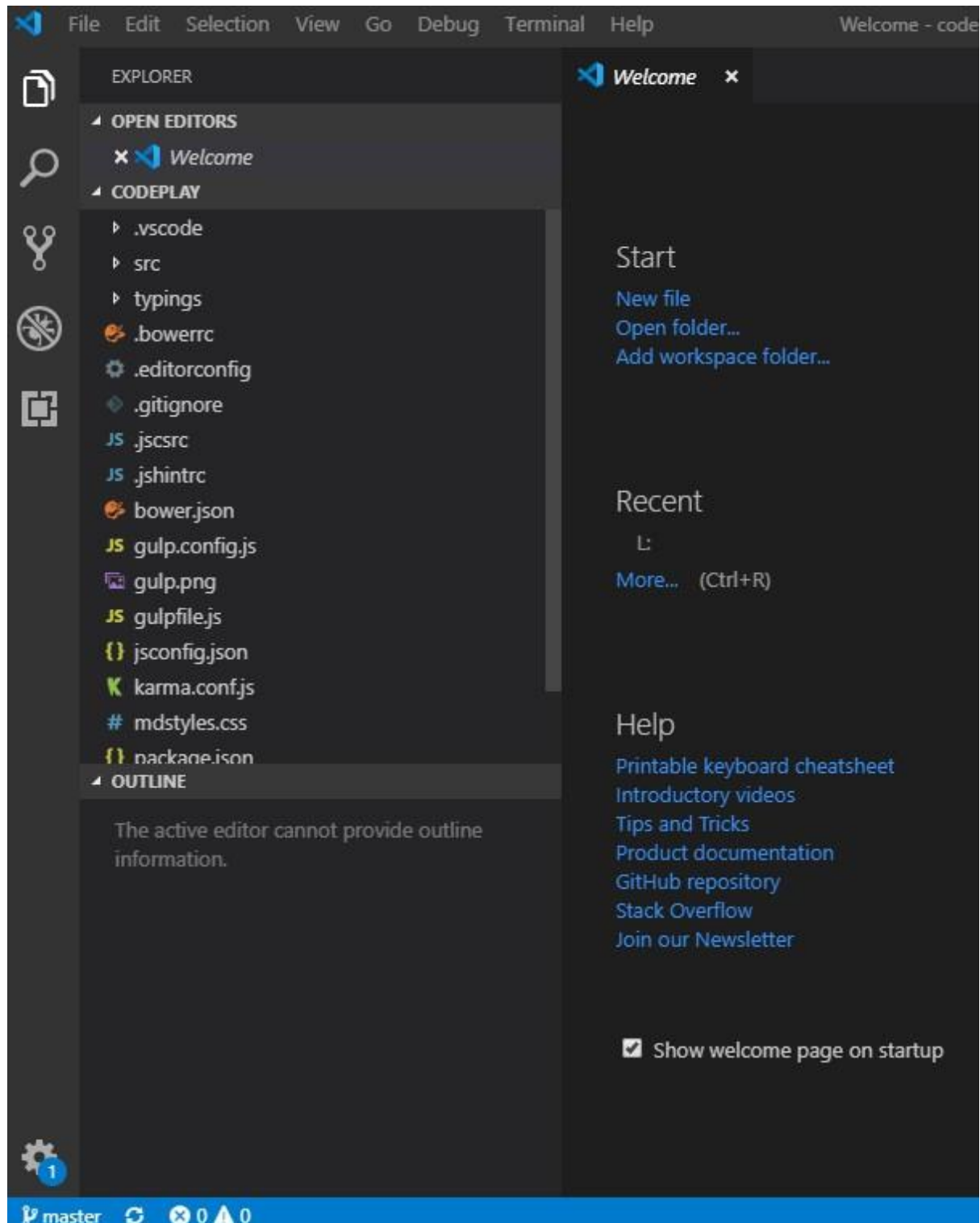
En la parte superior izquierda aparecen los archivos utilizados recientemente.



Vamos a abrir el proyecto codeplay. Para ello vamos a la opción Open folder y seleccionamos la carpeta:



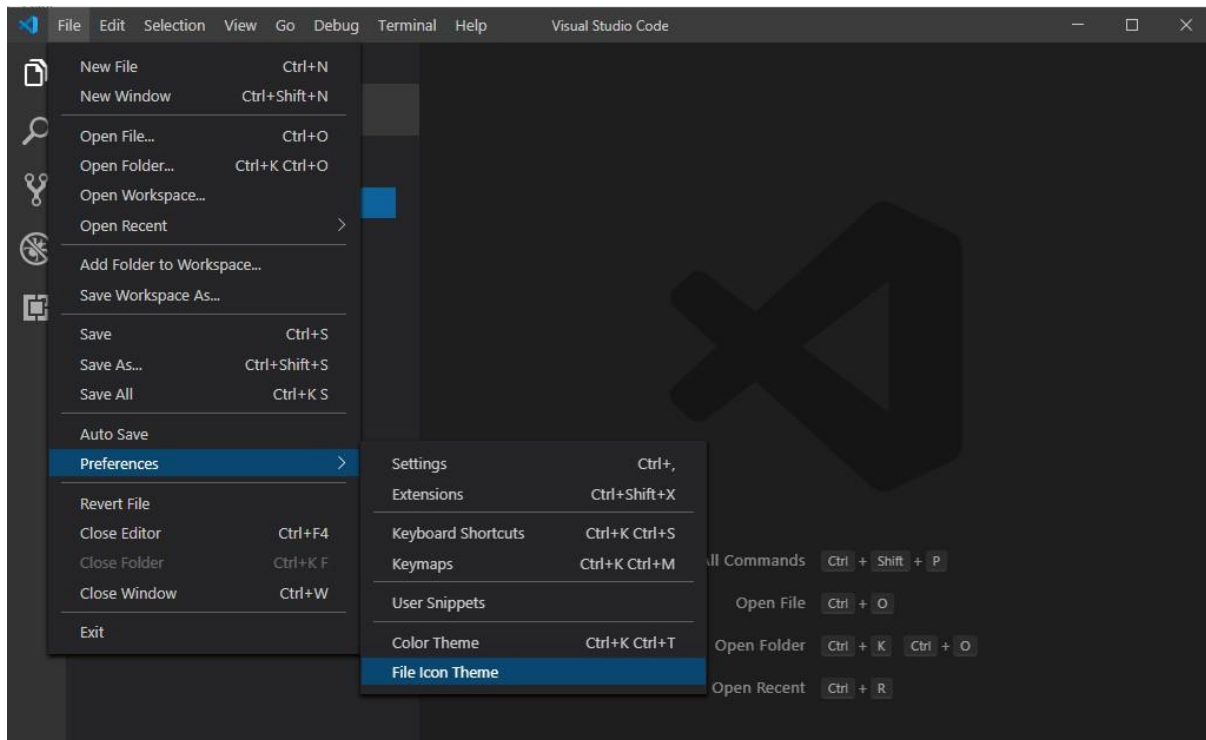
Vemos que se nos despliega una carpeta en el explorer:



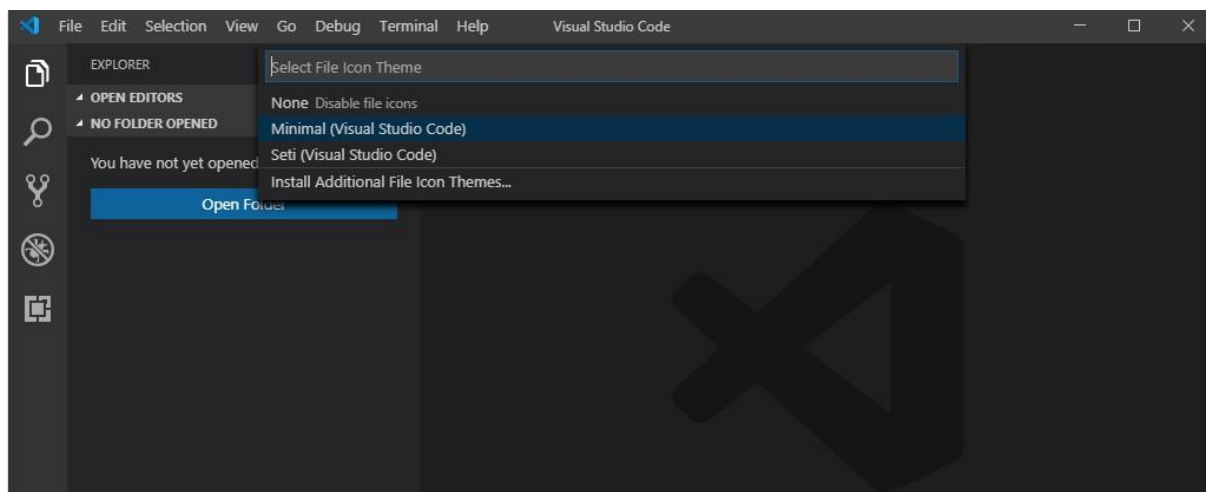
1.2 Ver iconos

Al agregar la extensión, Visual Studio Code reconoce automáticamente el icono según la extensión, esto no viene con Visual Studio Code, hay que descargar un tema de iconos. Para activar

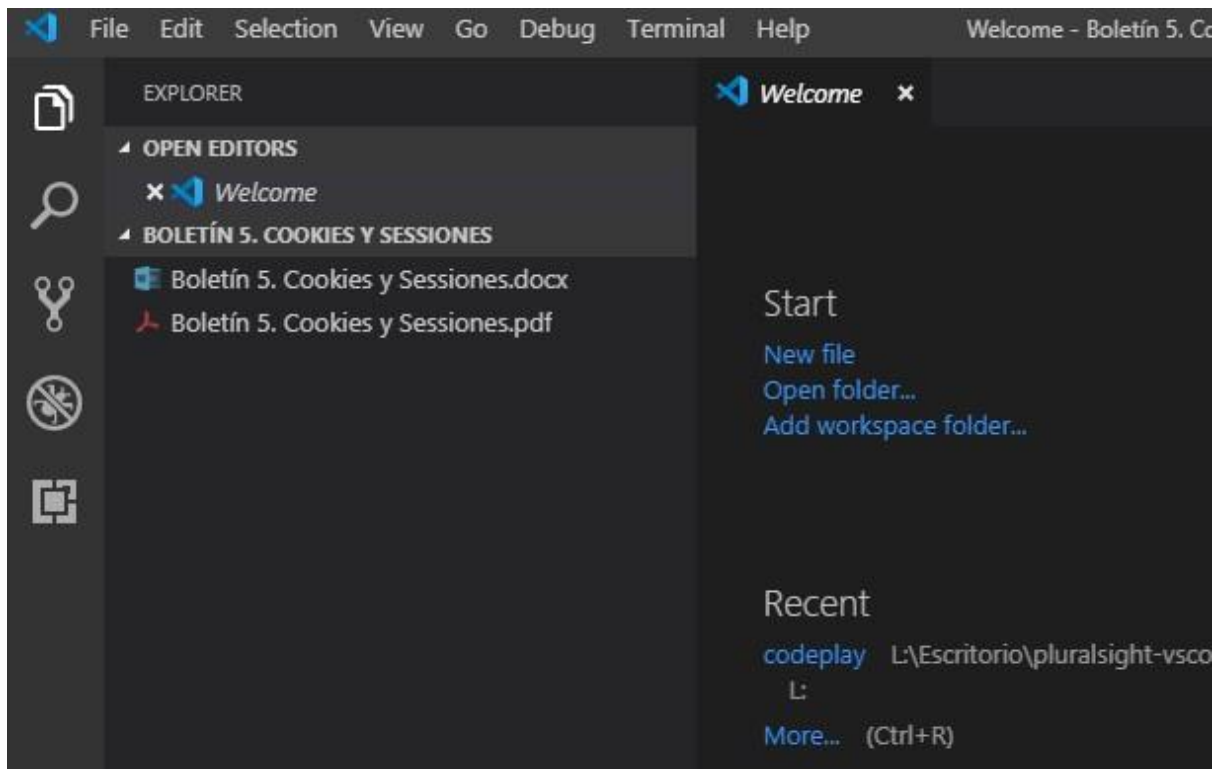
los iconos a través del menú (Code), (Preferences),
(File Icon Theme) puedo seleccionar diferentes iconos



Nos ofrece diferentes opciones:

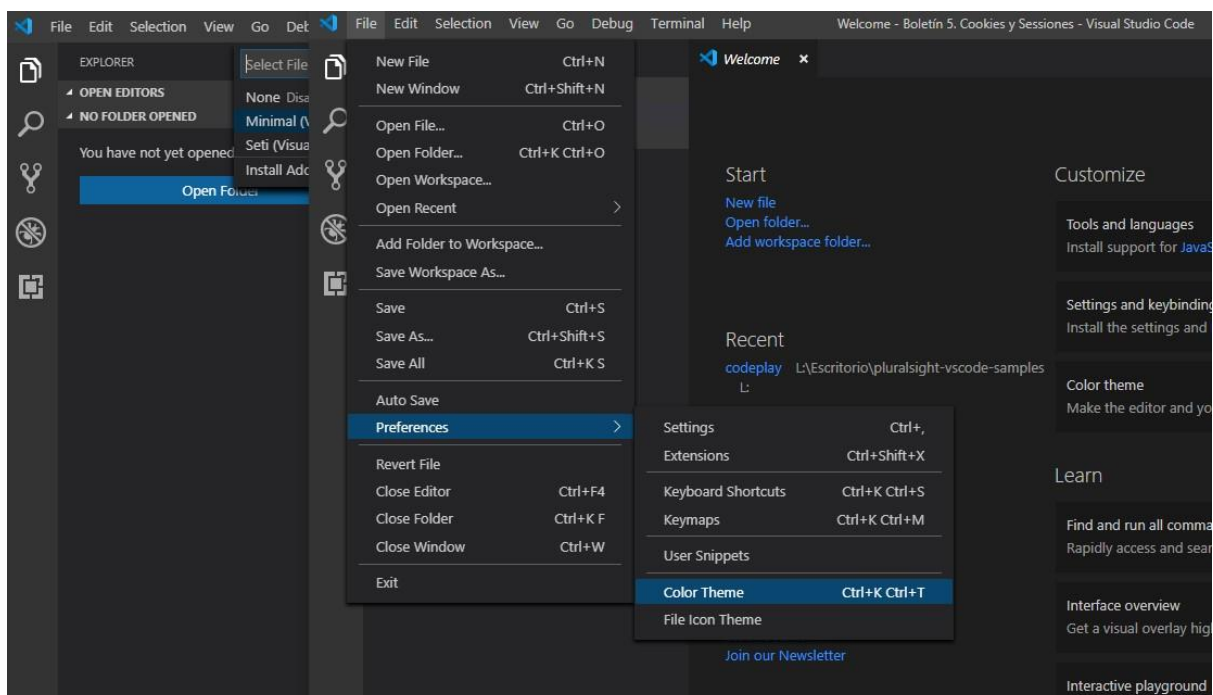


Al seleccionar la opción Seti vemos que nos muestra los iconos de los documentos que tengamos en nuestra carpeta:

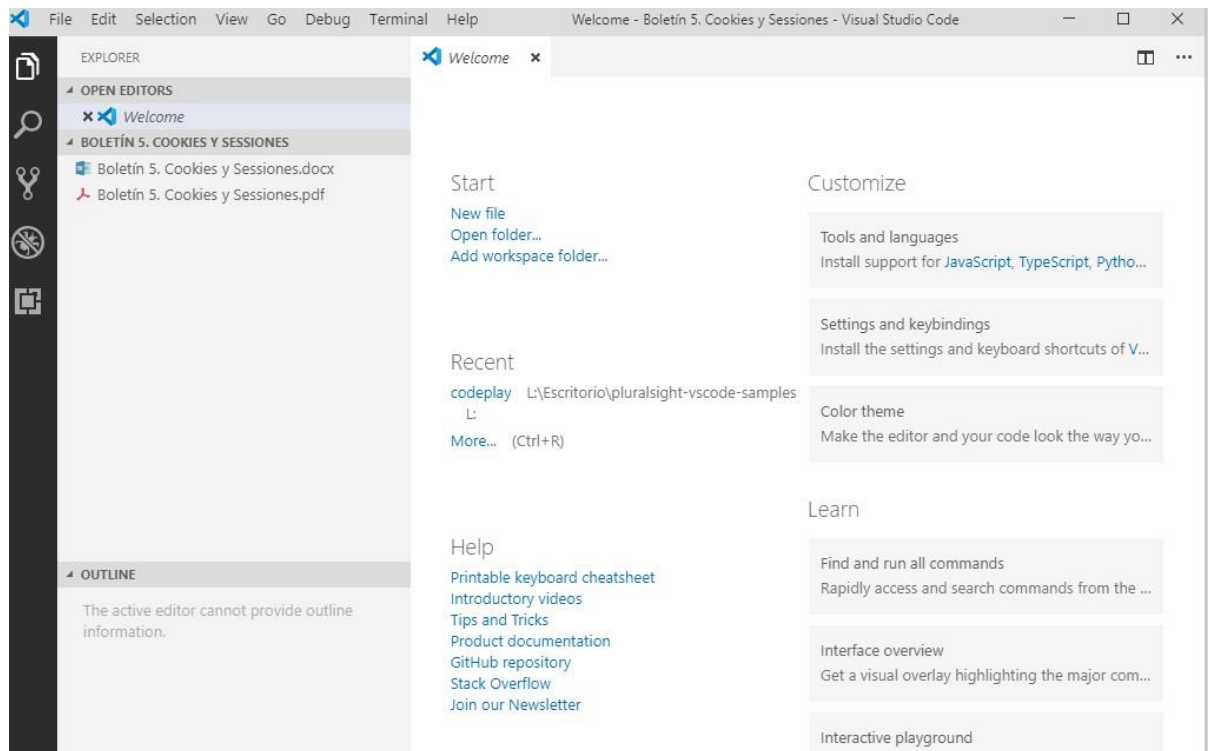


1.3 Temas


En la parte derecha está el editor, donde se puede abrir cualquier archivo (Visual Studio Code soporta una gran cantidad de formatos). También se puede cambiar el tema de visual Studio Code, en (Prefecences), (Color Theme):



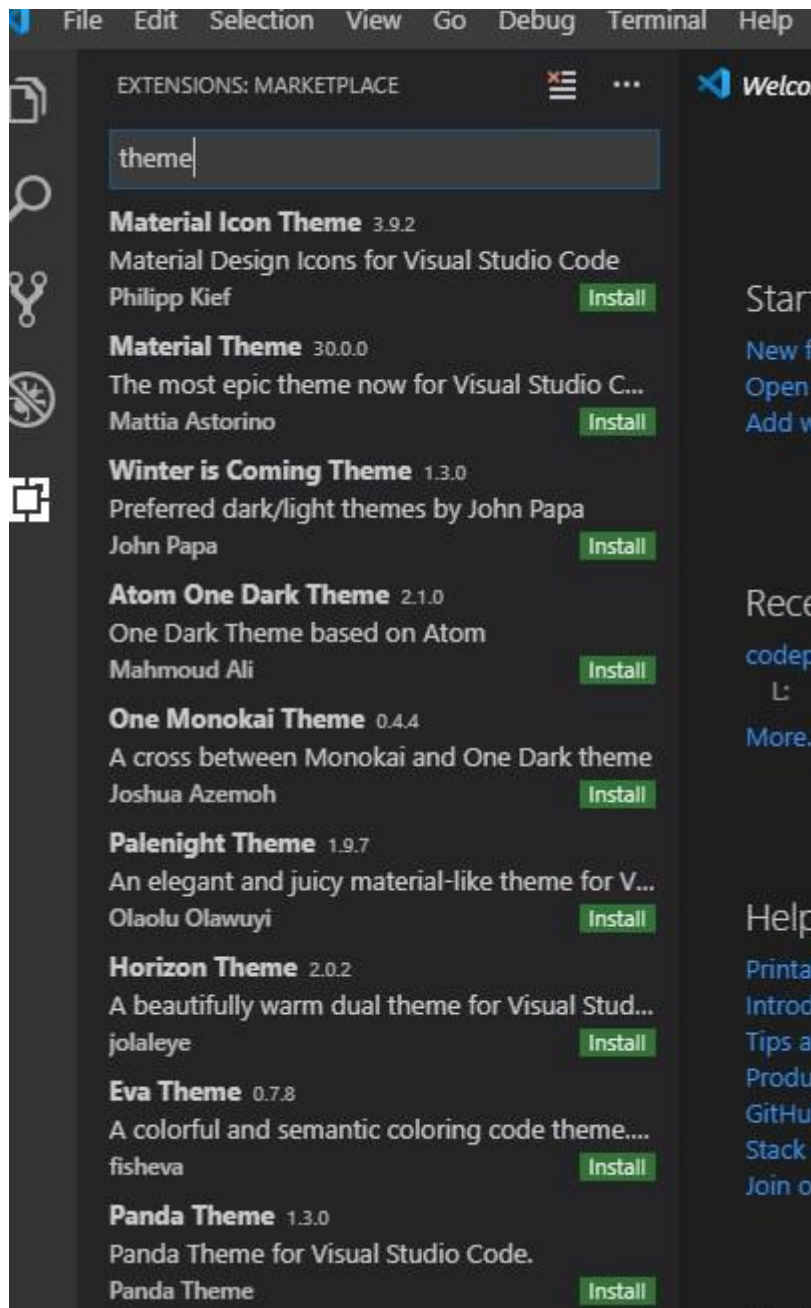
Si elegimos uno, vemos que la apariencia del visual studio code cambia:



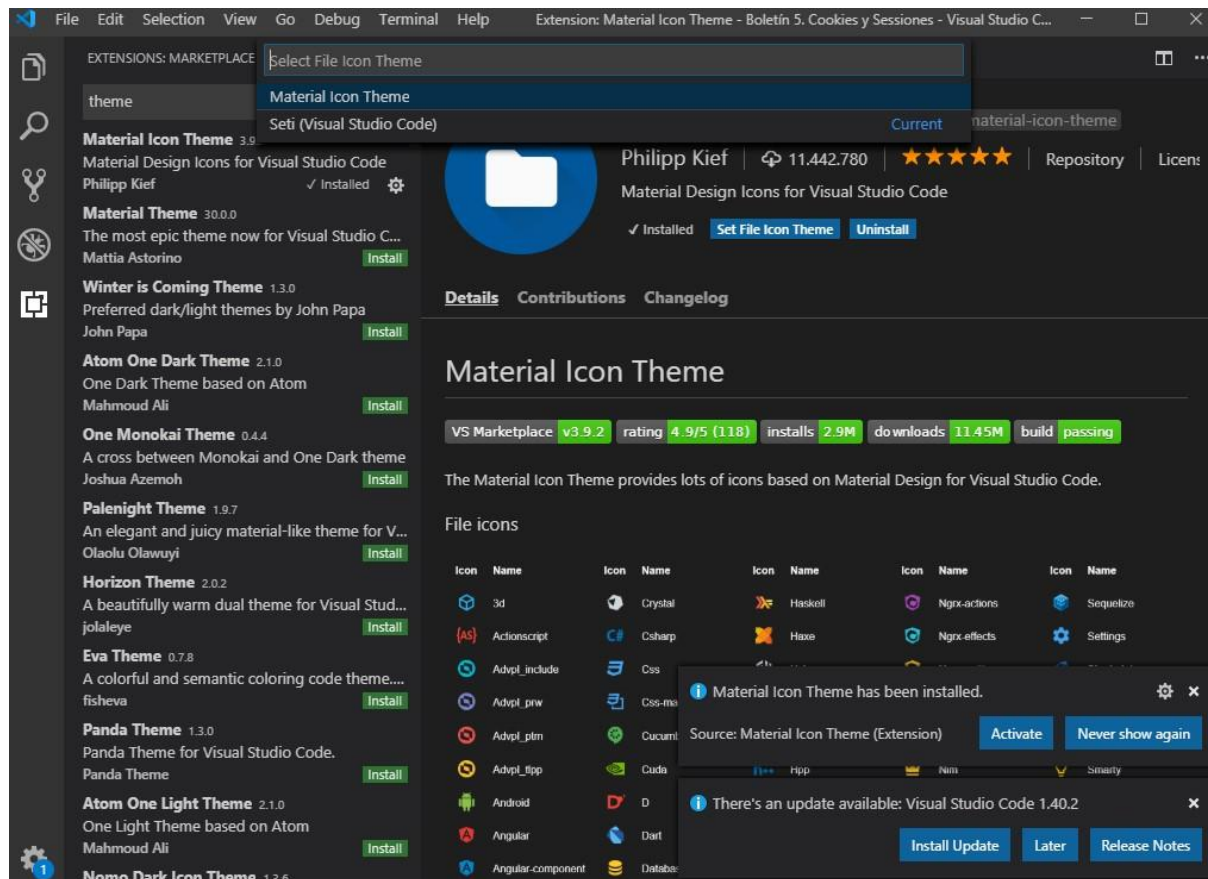
1.4 Instalar extensiones

Podemos descargar más temas. En esta última opción del menú izquierdo  podemos encontrar las diferentes extensiones que podemos instalar.

Si buscamos por theme encontramos multitud de temas a instalar:



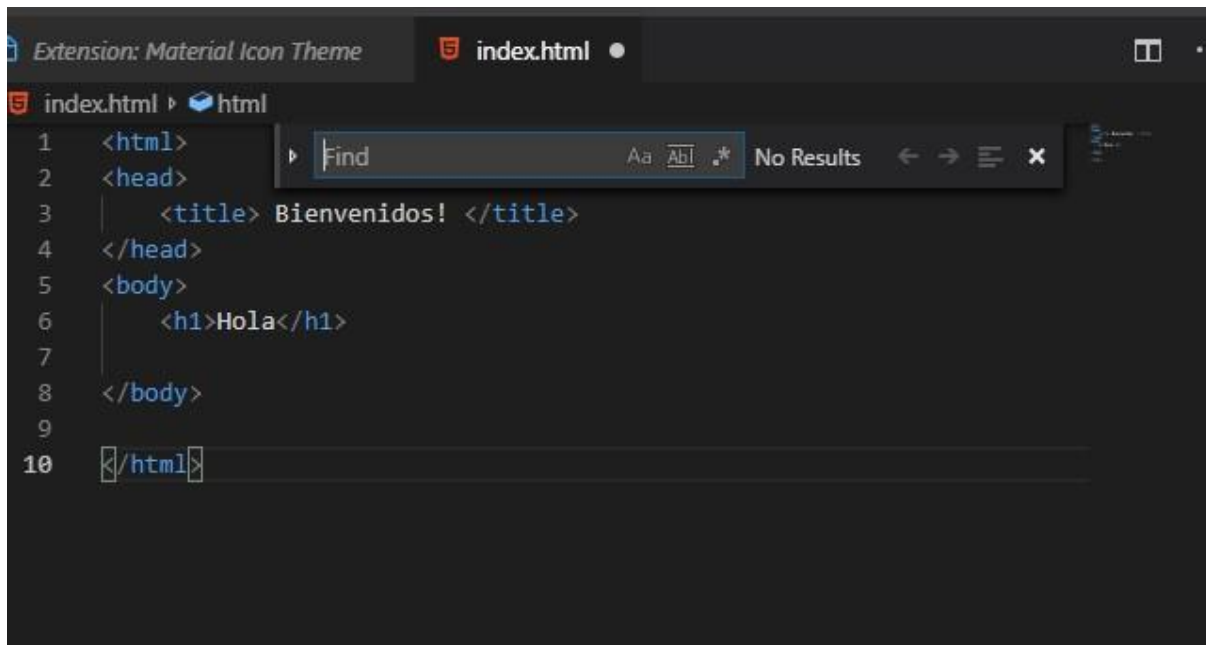
Para instalarla simplemente pulsamos el botón install.



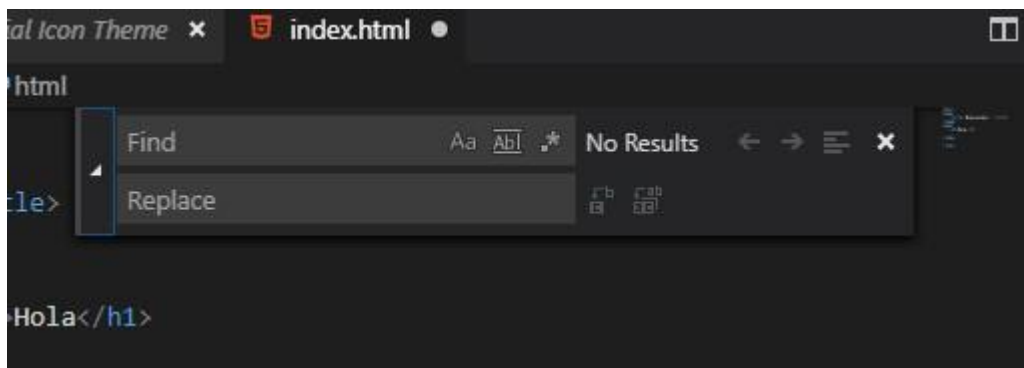
Ya estará disponible para su activación.

1.5 Buscar y reemplazar

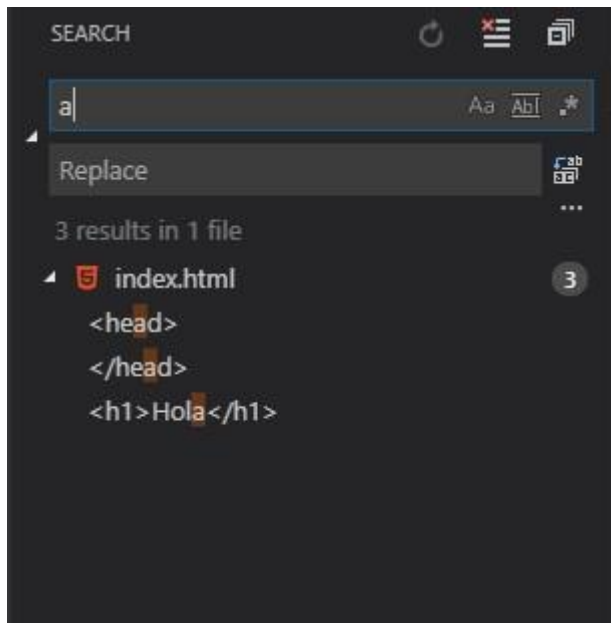
Para buscar podemos utilizar el atajo CTRL+F y nos aparecerá algo como esto:



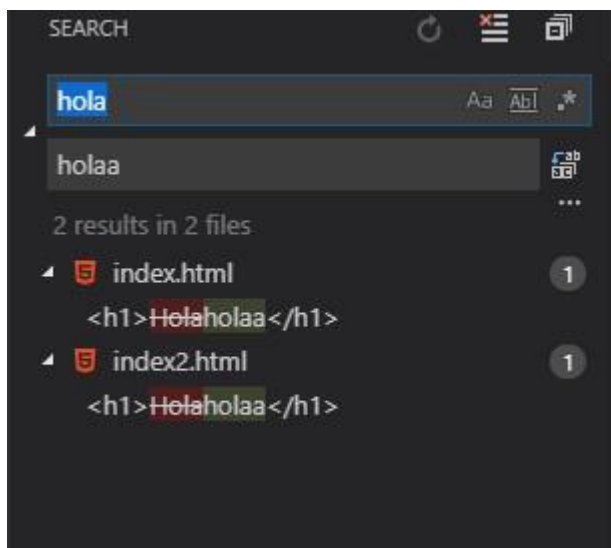
Si desplegamos nos da la opción de reemplazar:



También lo podemos hacer en la parte de los archivos con las opciones: control + shift + f :



Lo reemplazaríamos en todos los ficheros.

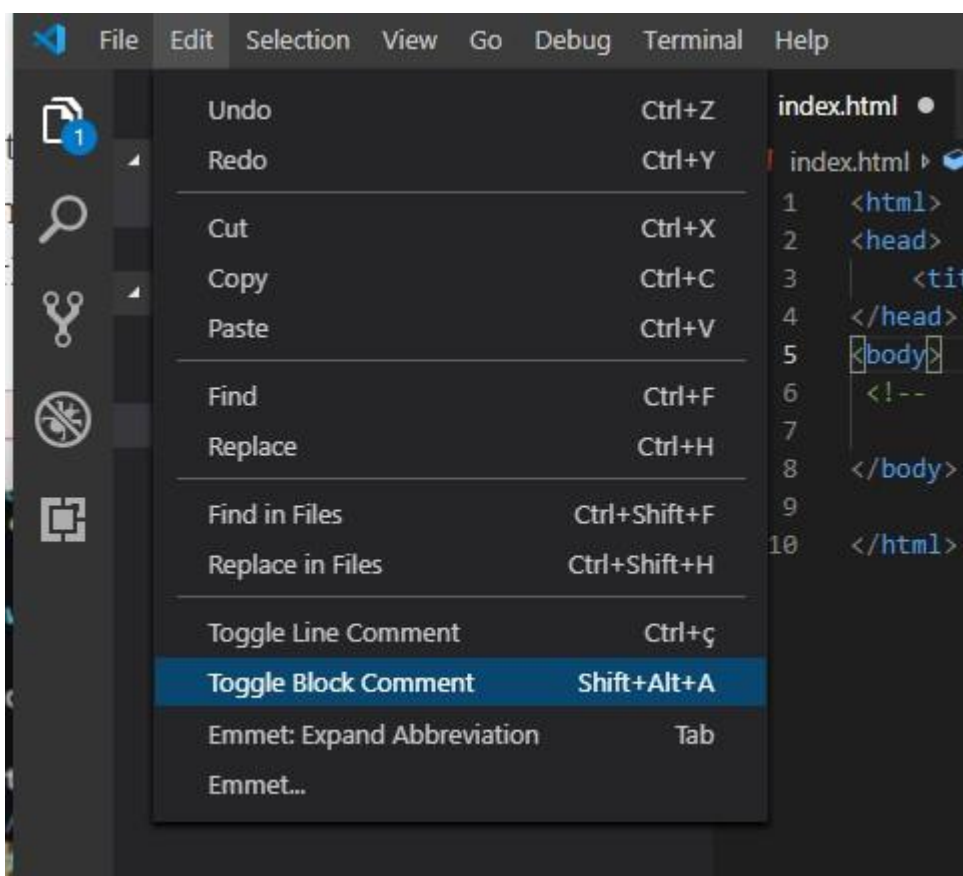


1.6 Comentar código

Podemos utilizar el atajo Shift+Alt+A para comentar las líneas:

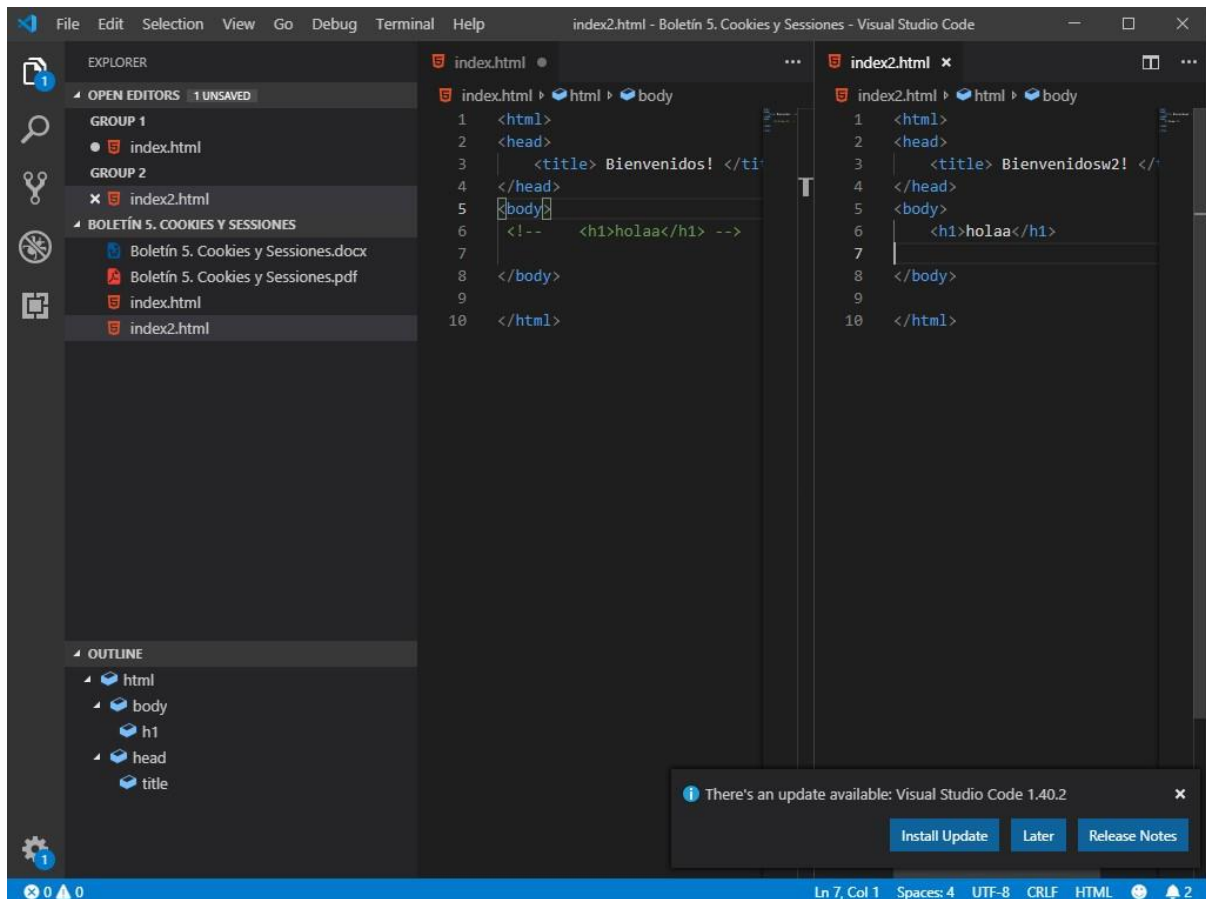
```
index.html ▶ html ▶ body
1  <html>
2  <head>
3    <title> Bienvenidos! </title>
4  </head>
5  <body>
6    <!-- <h1>holaa</h1> -->
7
8  </body>
9
10 </html>
```

O venir al menú de edición:



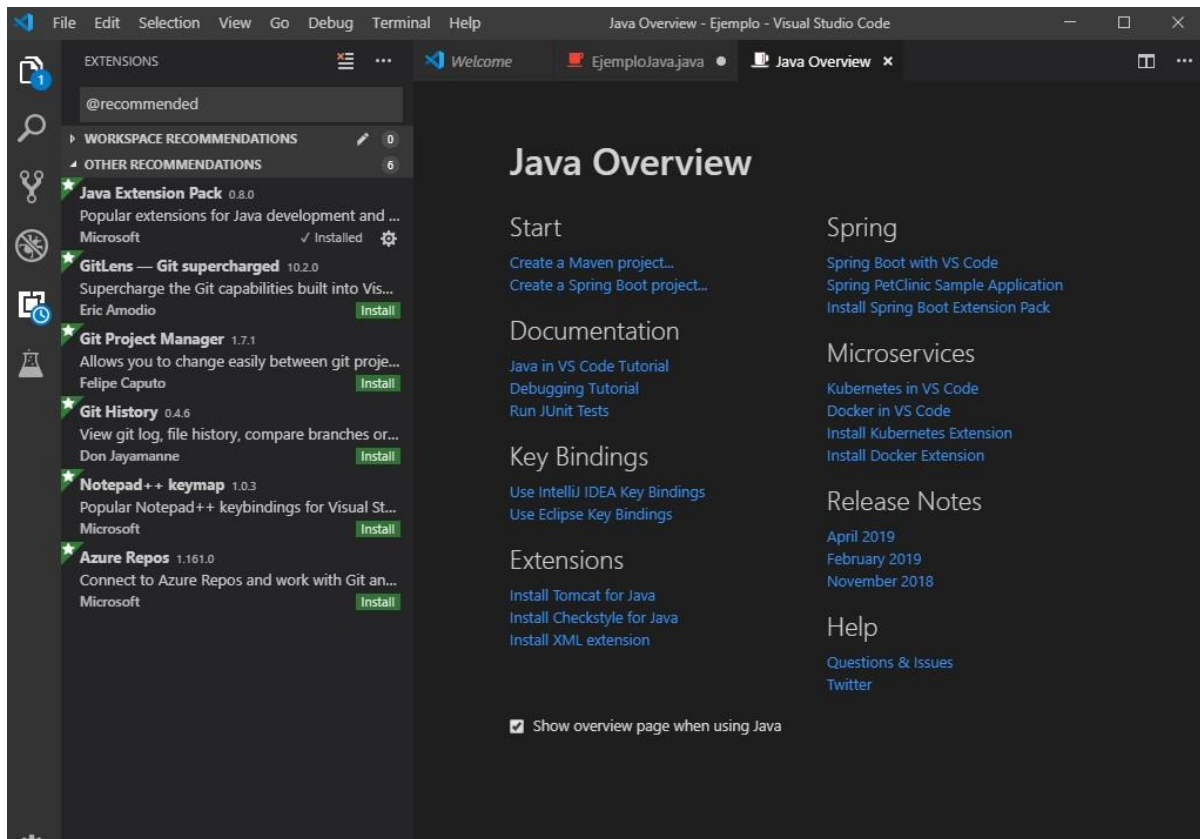
1.7 Grupos

Los grupos es cuando partimos nuestra ventana en dos o hasta en tres ventanas al mismo tiempo para trabajar en varios archivos.

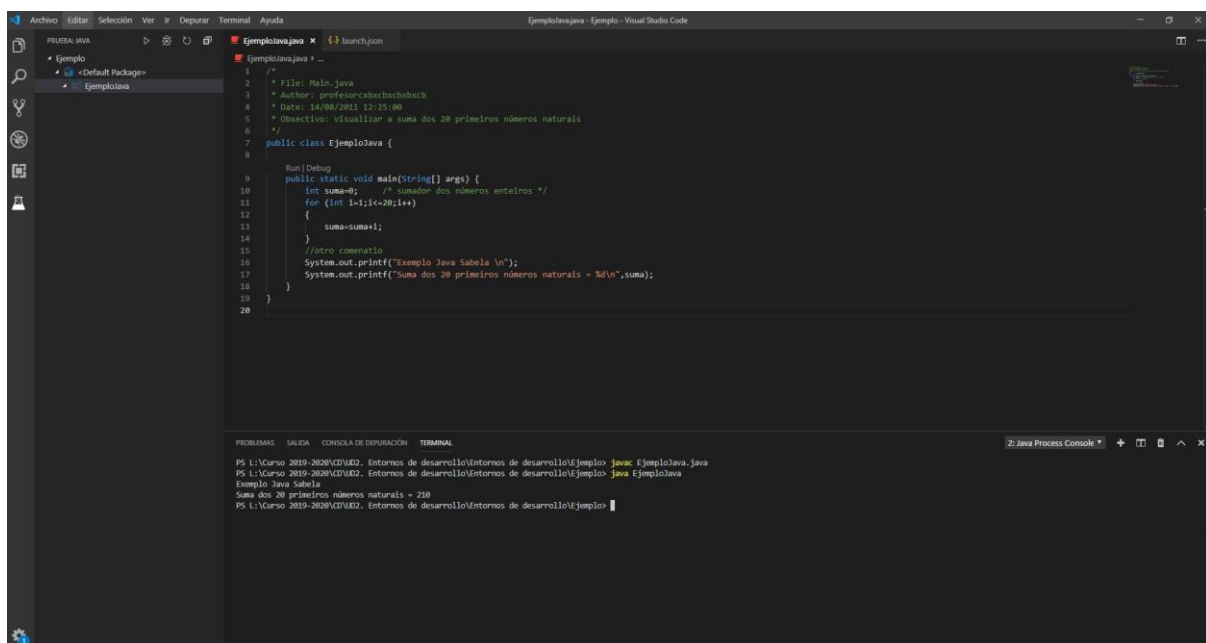


1.8 Proyecto java

Instalamos la extensión Java Extension Pack:

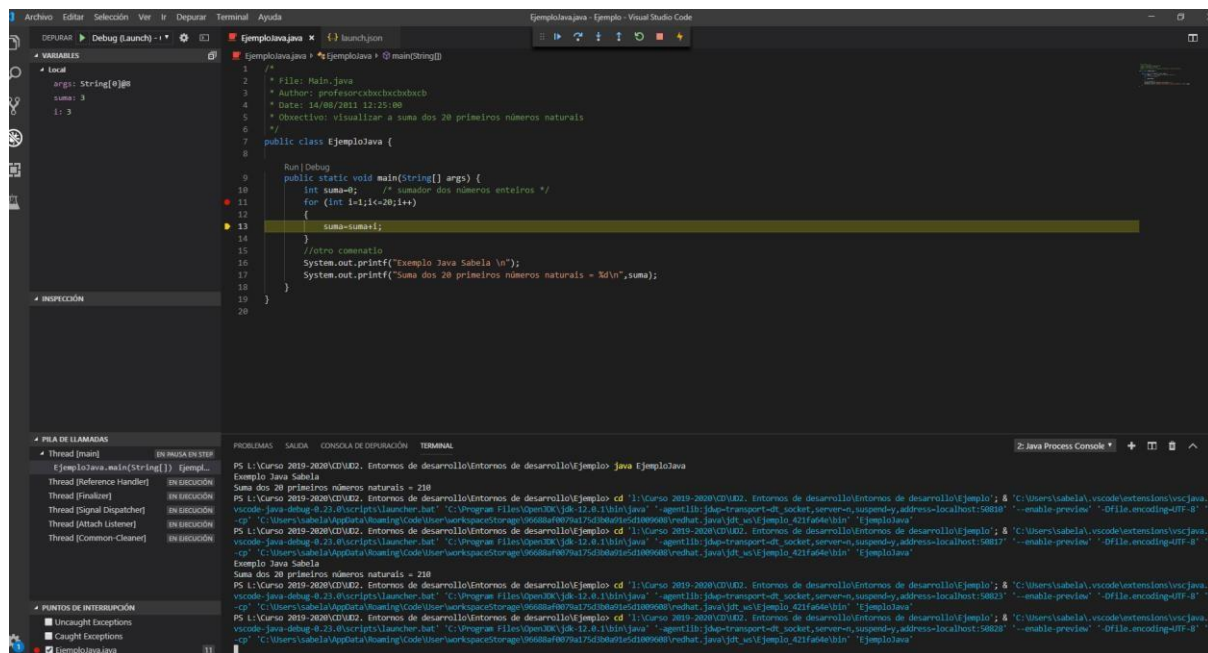


Crear y ejecutar un archivo .java



1.9 Depuración

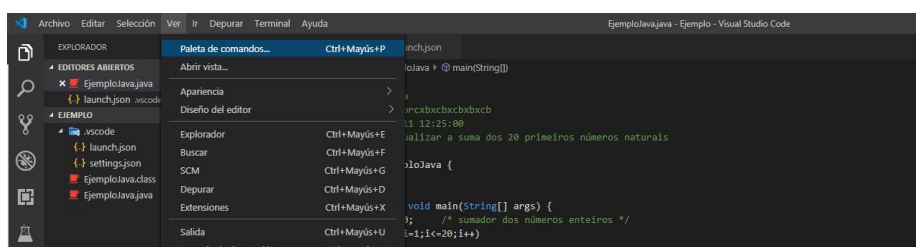
Podemos ir viendo cada uno de los puntos



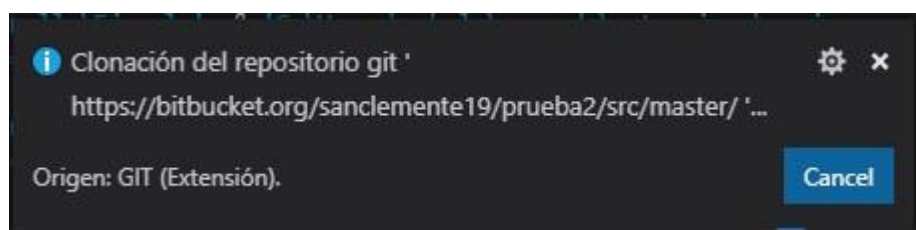
1.10 GIT

1.10.1 Clonar repositorio

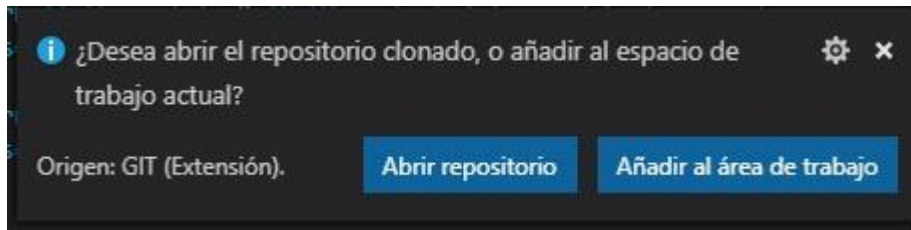
Abrimos la paleta de comandos. Para mostrar la paleta de comandos: podemos pulsamos la combinación de teclas **CTRL+SHIFT+P** o desde el menú view seleccionamos "command palette".



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

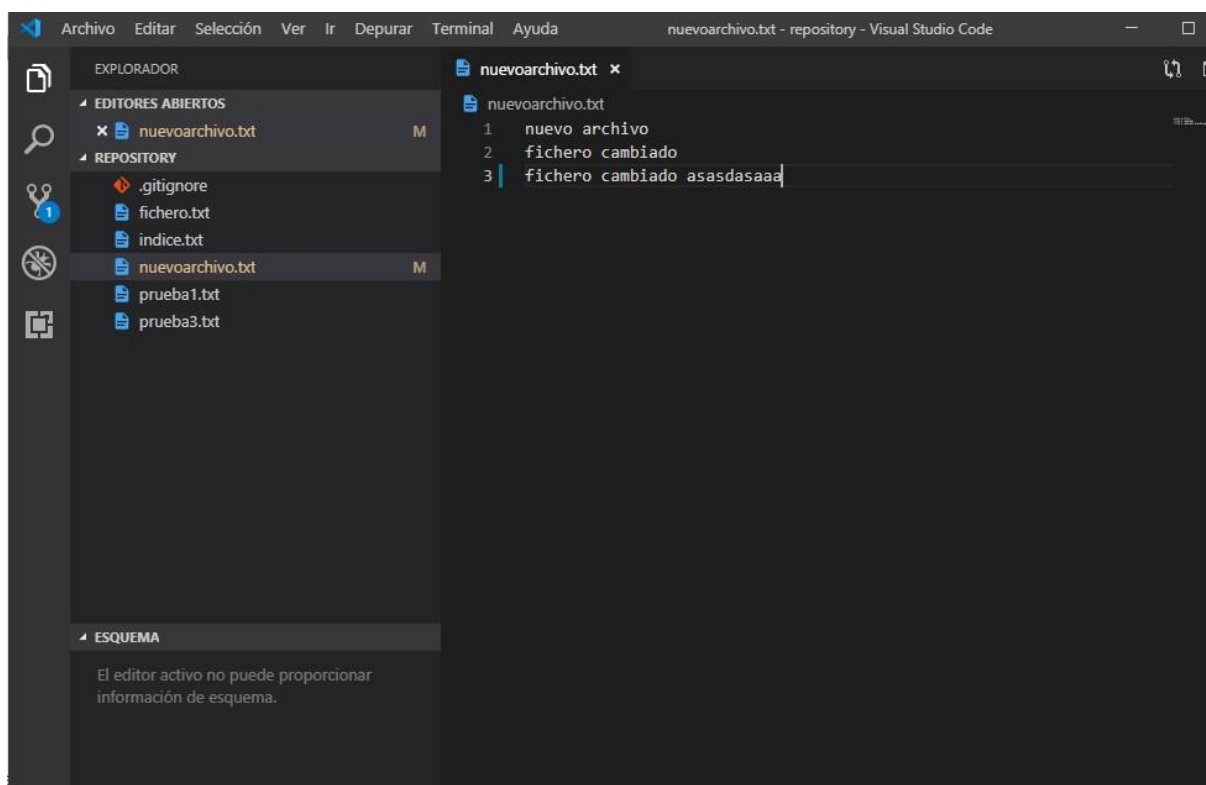


Una vez termine le damos a abrir repositorio:

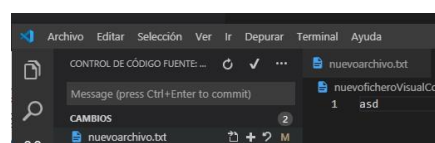


1.10.2 Commit

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

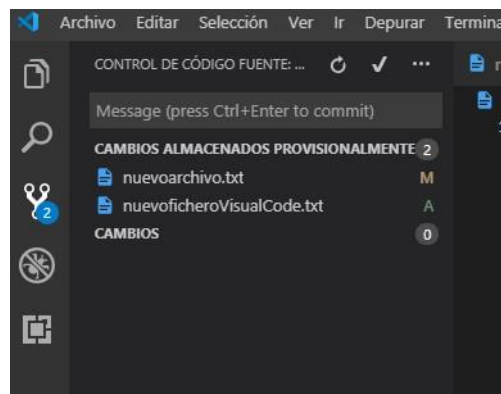


Cuando nos ponemos encima podemos ver diferentes opciones:



Cuando queramos subir cambios nos vamos a la pestaña de git y vemos qué ficheros están

modificados. Para preparar los ficheros (add) le damos al botón + y pasan a la pestaña cambios almacenados provisionalmente.



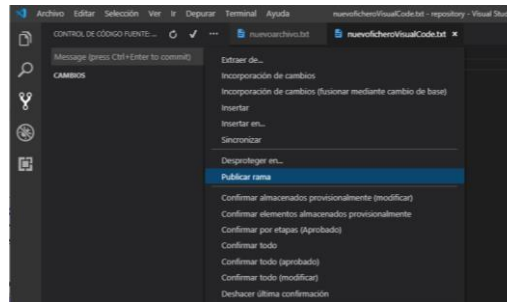
Una vez que estamos listos le damos al botón de commit (esto sólo guarda en local, tenemos que hacer un push para llevar los cambios a servidor). Introducimos un mensaje para el commit y presionamos Ctrl + Enter.



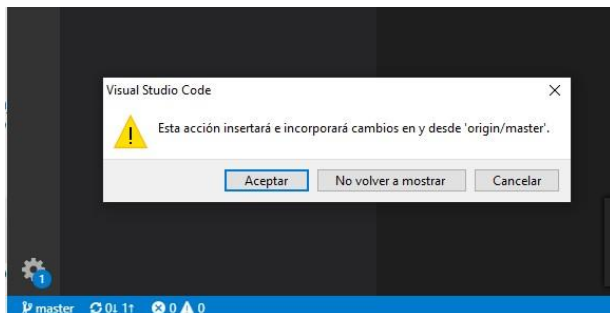
1.10.3 Push

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*.

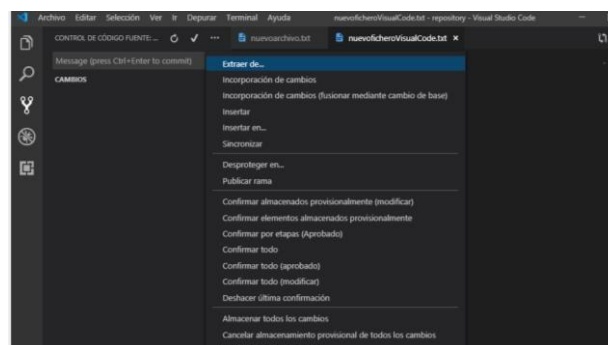


- O bien en la parte inferior pulsando sobre sincronizar (ojo esto haría push y pull).



1.10.4 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 *extraer de*. También podríamos hacer como en el paso anterior (pulsando en los iconos de la parte inferior, sincronizar).

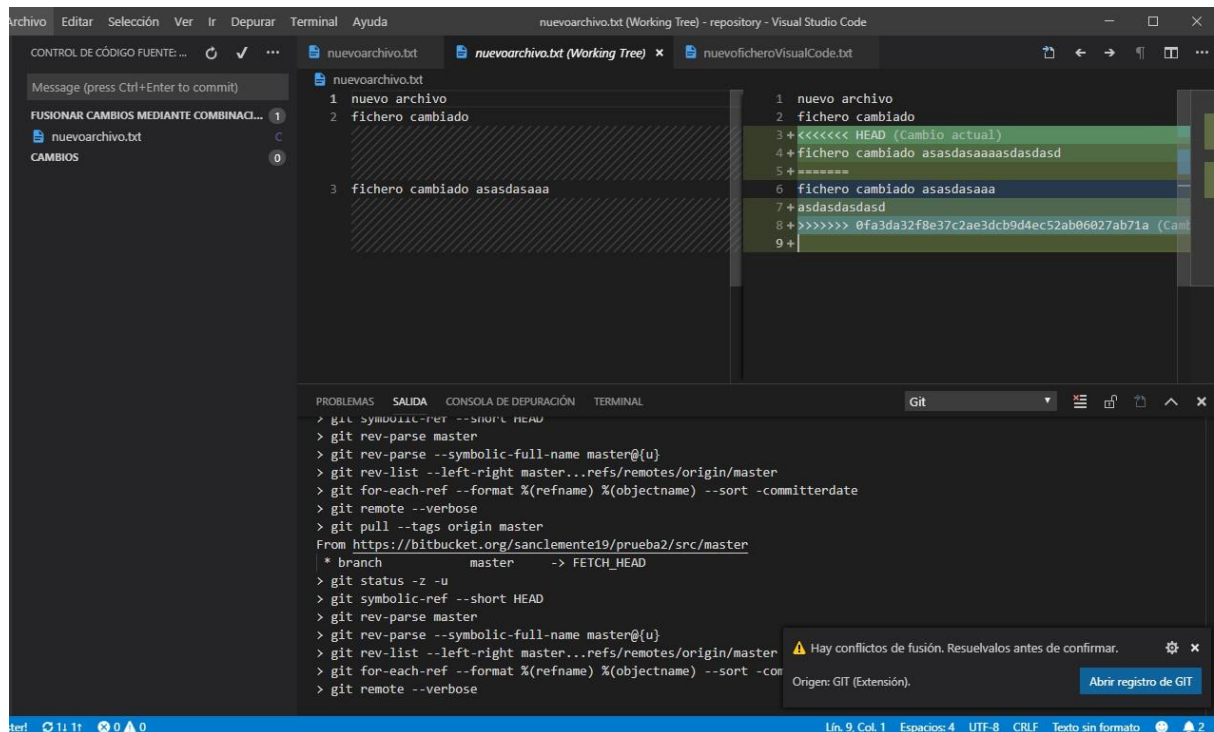


1.10.5 Resolver conflictos

¿ 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 nuevoarchivo.txt, 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 están los conflictos.
- Ahora podemos elegir que parte es la que queremos tomar como buena (la nuestra o la de servidor) y la aceptamos:



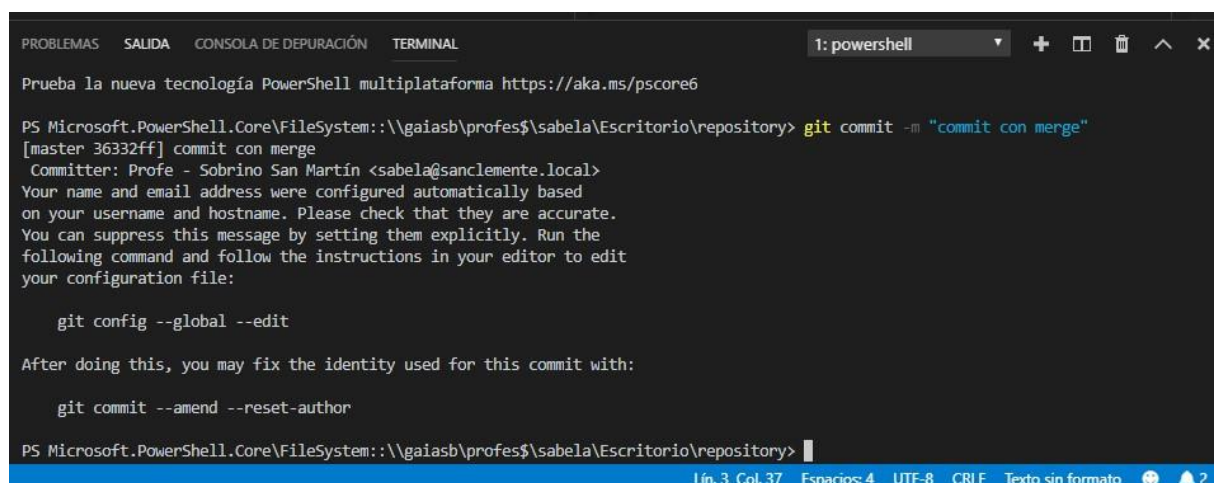
```

nuevoarchivo.txt (Working Tree)
1 nuevo archivo
2 fichero cambiado
3 +<<<<<< HEAD (Cambio actual)
4 +fichero cambiado asdasdasdasdasd
5 +
6 fichero cambiado asdasdasaa
7 +asdasdasdasd
8 +>>>>>> 0fa3da32f8e37c2ae3dcb9d4ec52ab06027ab71a (Cambio de servidor)
9 +

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL
> git symbolic-ref --short HEAD
> git rev-parse master
> git rev-parse --symbolic-full-name master@{u}
> git rev-list --left-right master...refs/remotes/origin/master
> git for-each-ref --format %(refname) %(objectname) --sort -committerdate
> git remote --verbose
> git pull --tags origin master
From https://bitbucket.org/sanclemente19/prueba2/src/master
* branch      master      -> FETCH_HEAD
> git status -z -u
> git symbolic-ref --short HEAD
> git rev-parse master
> git rev-parse --symbolic-full-name master@{u}
> git rev-list --left-right master...refs/remotes/origin/master
> git for-each-ref --format %(refname) %(objectname) --sort -committerdate
> git remote --verbose

Hay conflictos de fusión. Resuélvalos antes de confirmar.
Origen: GIT (Extensión).
Abrir registro de GIT
  
```

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



```

1: powershell
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS Microsoft.PowerShell.Core\FileSystem::\\galias\profes\sabela\Escritorio\repository> git commit -m "commit con merge"
[master 36332ff] commit con merge
Committer: Profe - Sobrino San Martín <sabela@sanclemente.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

PS Microsoft.PowerShell.Core\FileSystem::\\galias\profes\sabela\Escritorio\repository>
  
```

- Por último, hacemos el push para subir los cambios al repositorio.

1.10.6 Creando ramas

Podemos crear una rama mostrando la paleta de comandos (CTRL+SHIFT+P) y eligiendo el comando *git branch*, después 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.

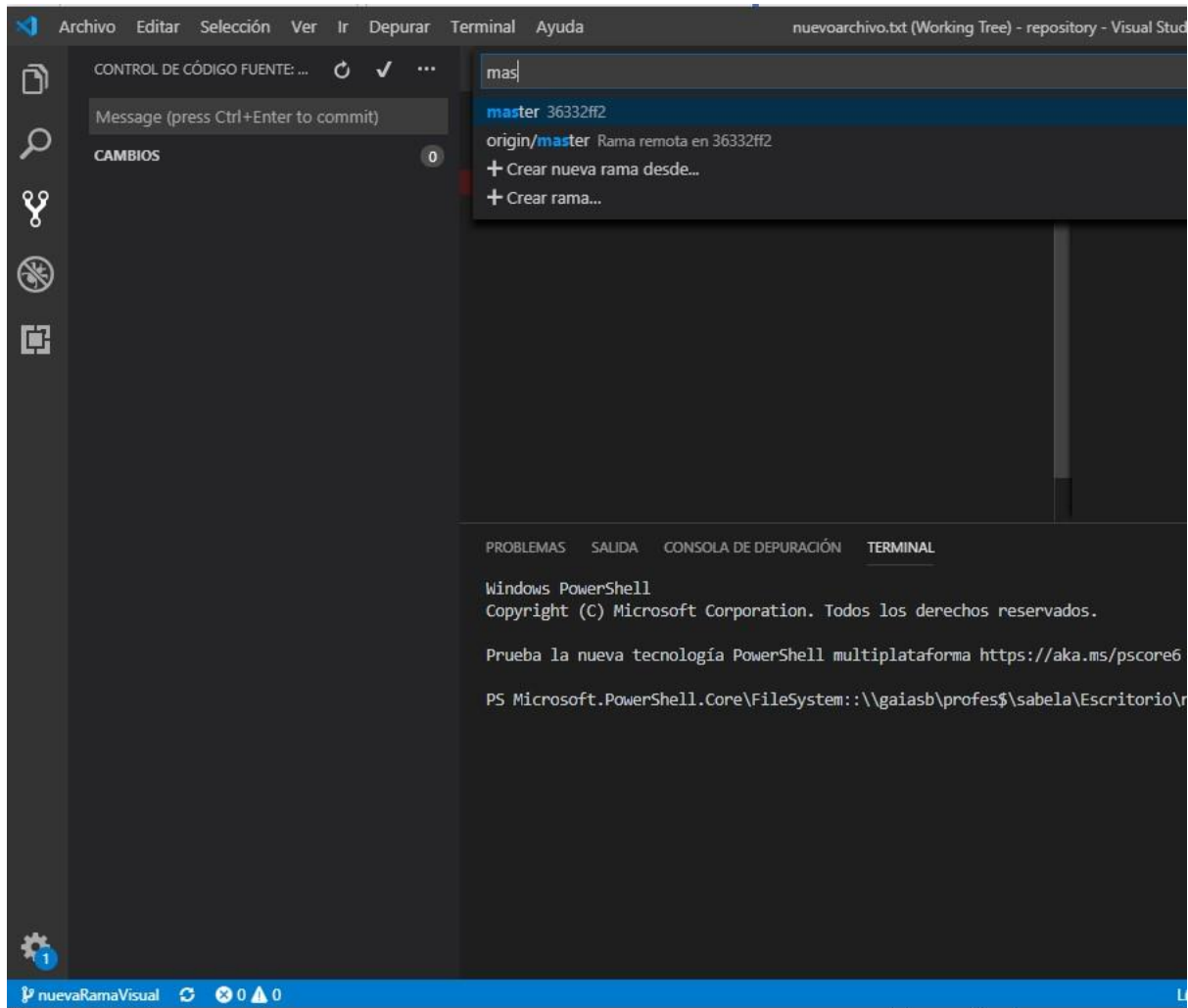


Para subir la rama:



1.10.7 Cambiar de rama

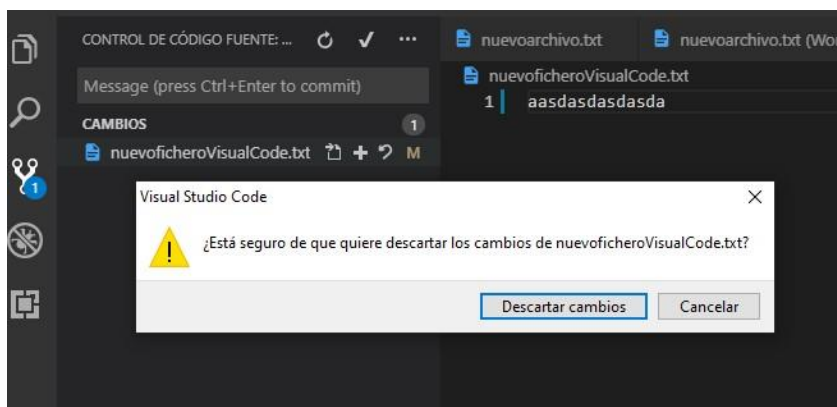
Para cambiar de rama, pinchamos sobre la parte inferior en la que aparece el nombre de la rama actual, y nos aparece un diálogo en el que podemos teclear el nombre de la rama al que queremos navegar.



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

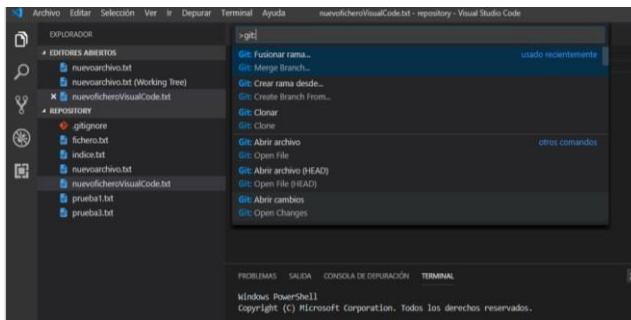
1.10.8 Descartar cambios

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*.



1.10.9 Merge con master

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.



ANEXO I: Materiales

I. Textos de apoyo o de referencia

- https://manuais.iessanclemente.net/index.php/Visual_Studio_Code

II. Recursos didácticos

- Apuntes en el aula virtual.
- Ordenador personal, con navegador web y conexión a internet.
- Software para elaboración de documentos de texto.