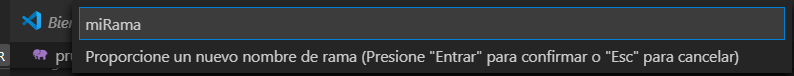
Trabajando con ramas

***Creando ramas***

Podemos crear una rama mostrando la paleta de comandos (CTRL+SHIFT+P) y eligiendo el comando **Create Branch**, después nos preguntará por el nombre de rama.

En nuestro caso vamos a crear una rama que se llame **miRama.**

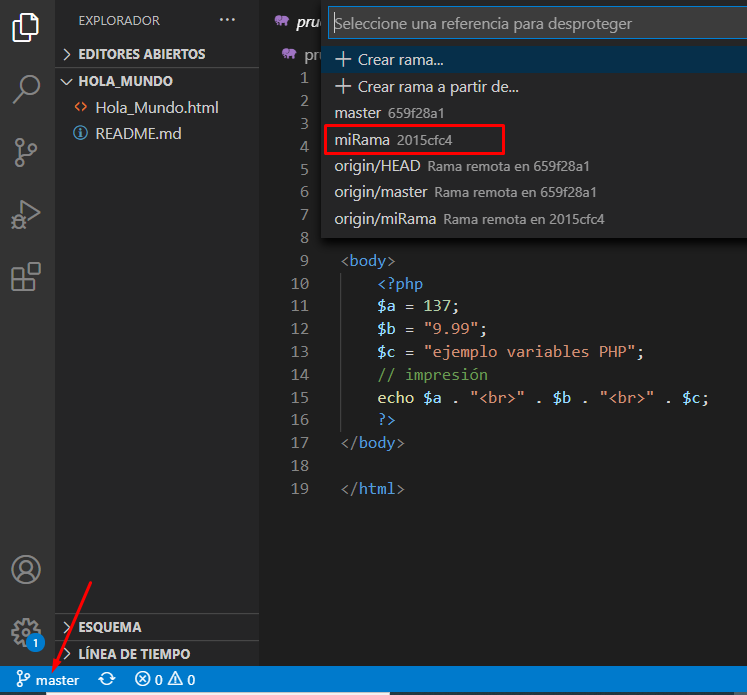


Veréis que en la parte inferior ya ha aparecido el nombre de la rama creada.

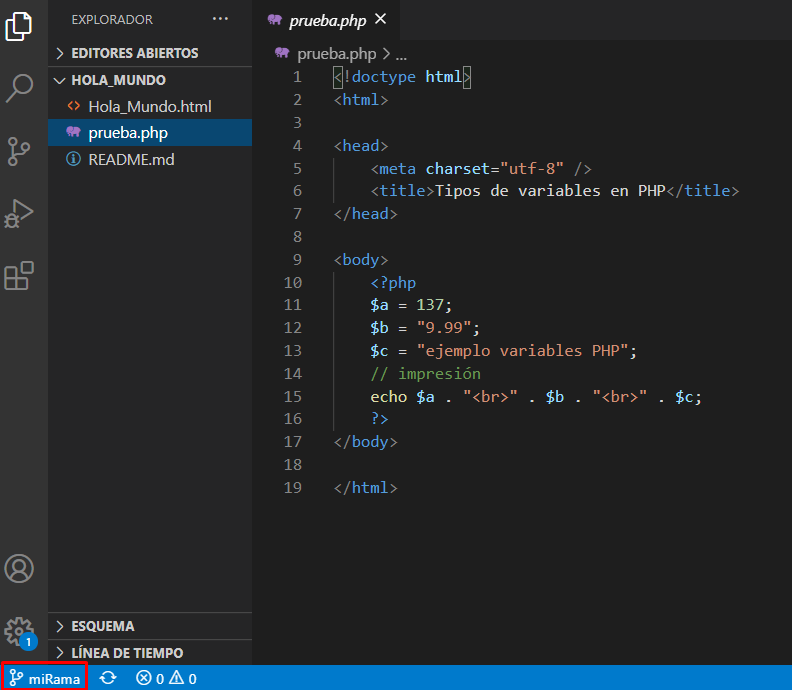


***Cambiar de ramas***

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.



Si os fijais en el proyecto HolaMundo, he añadido un archivo **prueba.php** en la rama que hemos creado anteriormente con el código que aparece en pantalla.

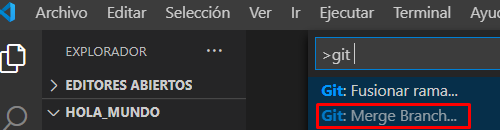


Una vez creada en local, podemos subirla al servidor pulsando sobre el indicador de la nube que aparece en la parte inferior de la ventana.



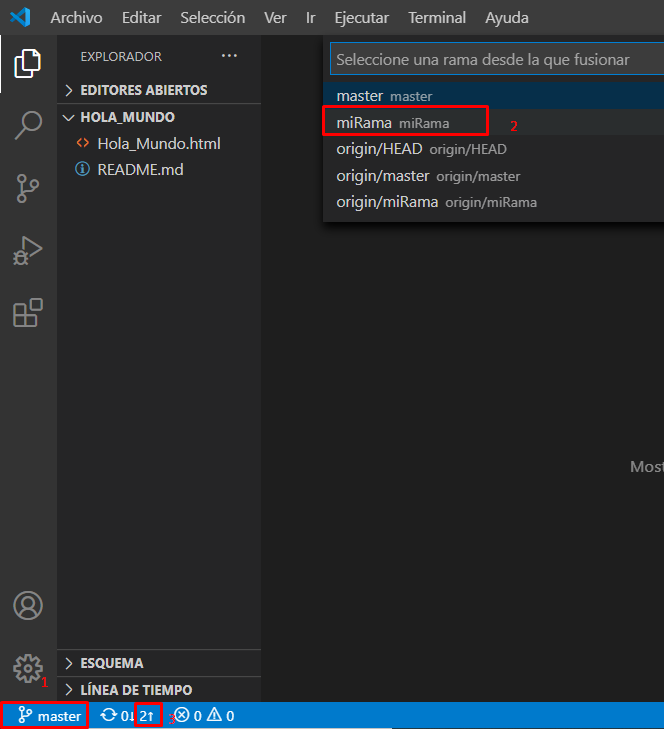
***Mezclar ramas***

Supongamos el caso en que mi rama se ha quedado desactualizada con repecto 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



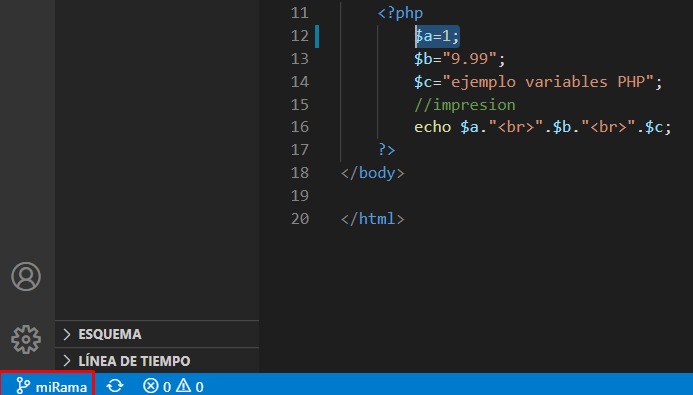
con ese comando me pregunta que rama quiero mezclar a la actual, tecleo su nombre y realizo el **merge**.

Ten en cuenta que la **mezcla de ramas** (git merge ) se hace en **local**, por tanto habrá que subir todos esos cambios al repositorio de GitHub.(paso 3)



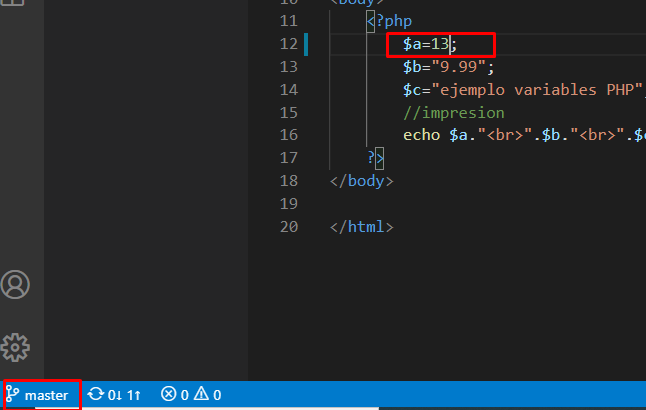
***Como resolver conflictos de fusión-merge conflicts***

Imaginarpos que vamos a modificar tanto en la rama master como el la rama miRama el fichero prueba.php y la modificación que vamos a hacer es la siguiente en ambas ramas y en el mismo fichero de ambas ramas.



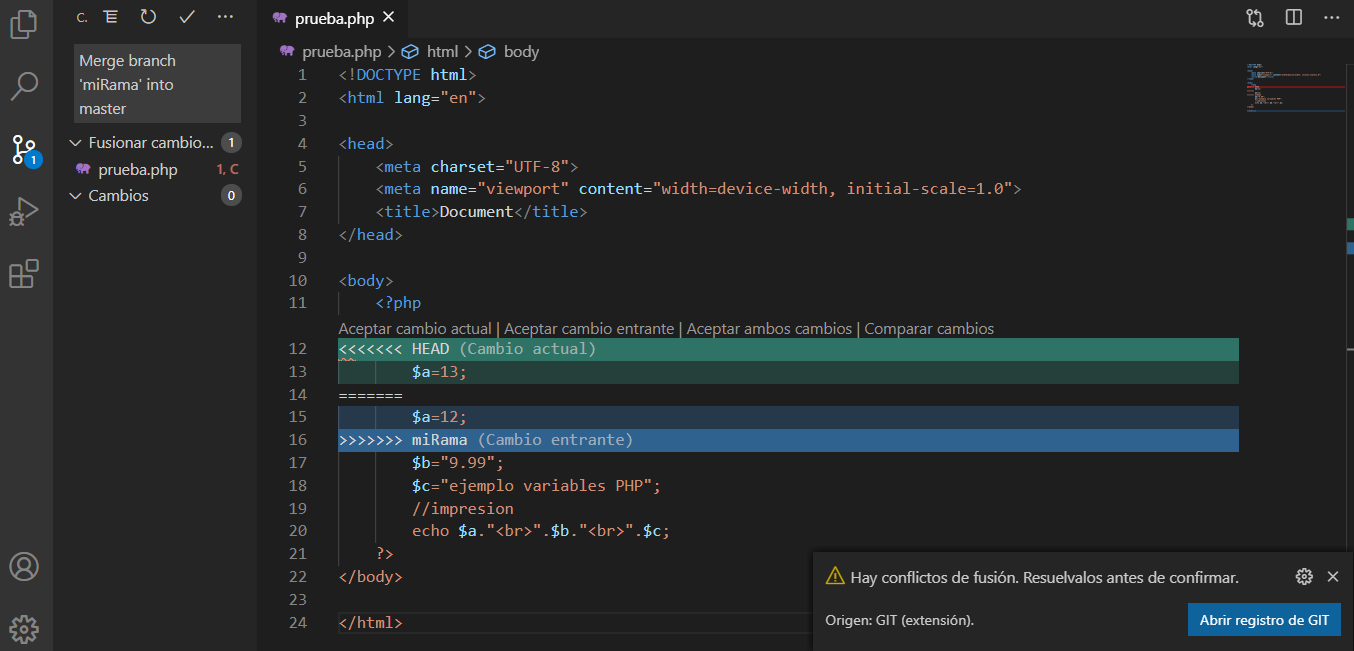
Hago un commit de la modificación de este cambio en local

Ahora me posiciono en la rama master y cambio el valor de la misma variable

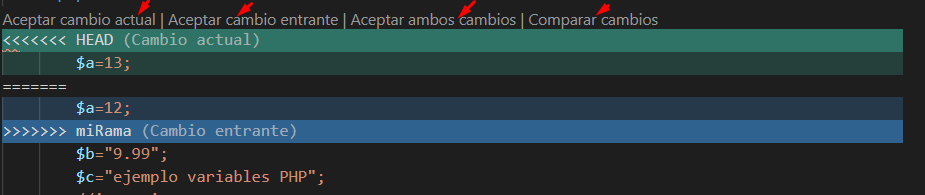


Hago el commit correspondiente en local.

Ahora voy a hacer sincronizar ambas ramas con **git merge** como hemos hecho antes, fijaros que os saldrá una pantalla como esta



Lo que ha pasado es que, al mezclar ambas ramas, el programa ha encontrado que se ha modificado la misma línea del mismo fichero y no sabe con cual de ella quedarse, y nos da la **posibilidad de quedarnos con uno de ellos**.



### PRACTICA 1.- Lista de la compra para la cena del viernes

**Hacer una lista de la compra.**

Vamos a trabajar con uno de los repositorios remotos vistos en este caso: **GitHub**

Objetivos

Practicar las funciones básicas de Git trabajando sobre un repositorio local y remoto.

Tareas

1. Crear Nuevo repositorio en el alojamiento elegido para el proyecto
2. Repositorio inicial con varios archivos:
   * **supermercado**.txt: lista compra supermercado
   * **carniceria**.txt: lista compra carnicería
   * **pescaderia**.txt: lista compra pescadería.

Inserta **varios productos** en cada una de las listas de la compra.

1. Configurar datos, para poder trabajar con VSCode en ese repositorio.
2. Crear una **rama adicional** "**cena\_viernes**". Crear una lista en esa rama (**asistentes\_cena\_viernes.txt**). Inserta algunos asistentes a la cena.
3. **Unir** (**mezclar**) los cambios de la rama cena\_viernes en master en el **repositorio local**.
4. Subir ambas ramas al **repositorio remoto** utilizado**.**
5. En el **repositorio local**, registrar **cambios adicionales** en las ramas master y cena\_viernes (en ambas)en la lista asistentes\_cena\_viernes.txt
6. Incorporar (merge) los nuevos cambios en la rama cena\_viernes a la rama master. Resolver conflictos si aparecen. ¿Cómo lo has realizado?
7. Subirlo al repositorio remoto los cambios efectuados.