**INFORME TRABAJO ESPECIAL**

**Experiencia personal del manejo de Git y Trello en equipo**

**Integrantes:** Carrizo, Noelia.

Gherbi, Andrea.

**Link al repositorio:** <https://github.com/noelia2016/tpe_web2> Este repo es privado por el momento ya que lo utilizamos para otra materia por ende los agregaremos como colaboradores.

**Link al tablero de Trello:** <https://trello.com/b/lNuEd2W3/sistema-hotelero>

Contaremos la experiencia en el uso de estas herramientas para la realización del trabajo en equipo para el práctico especial de Web II.

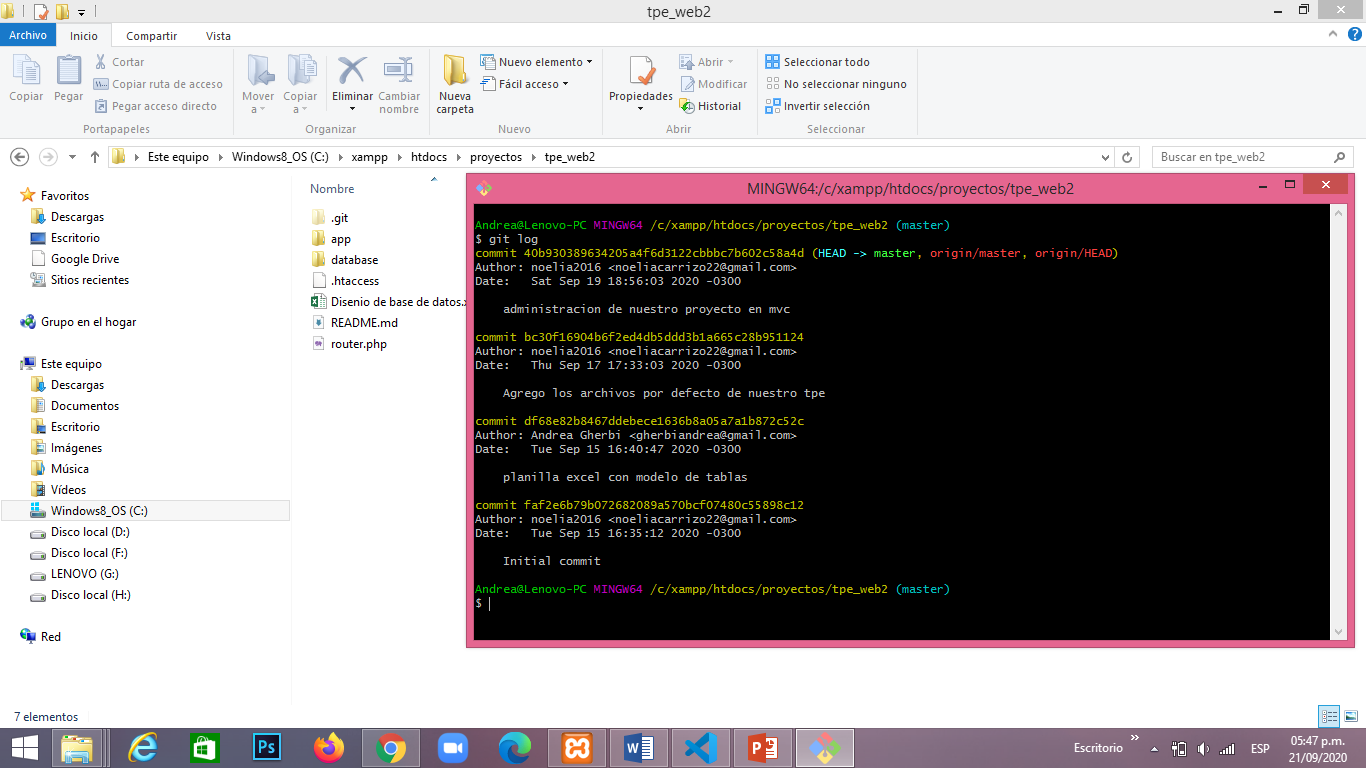
**Desarrollo**

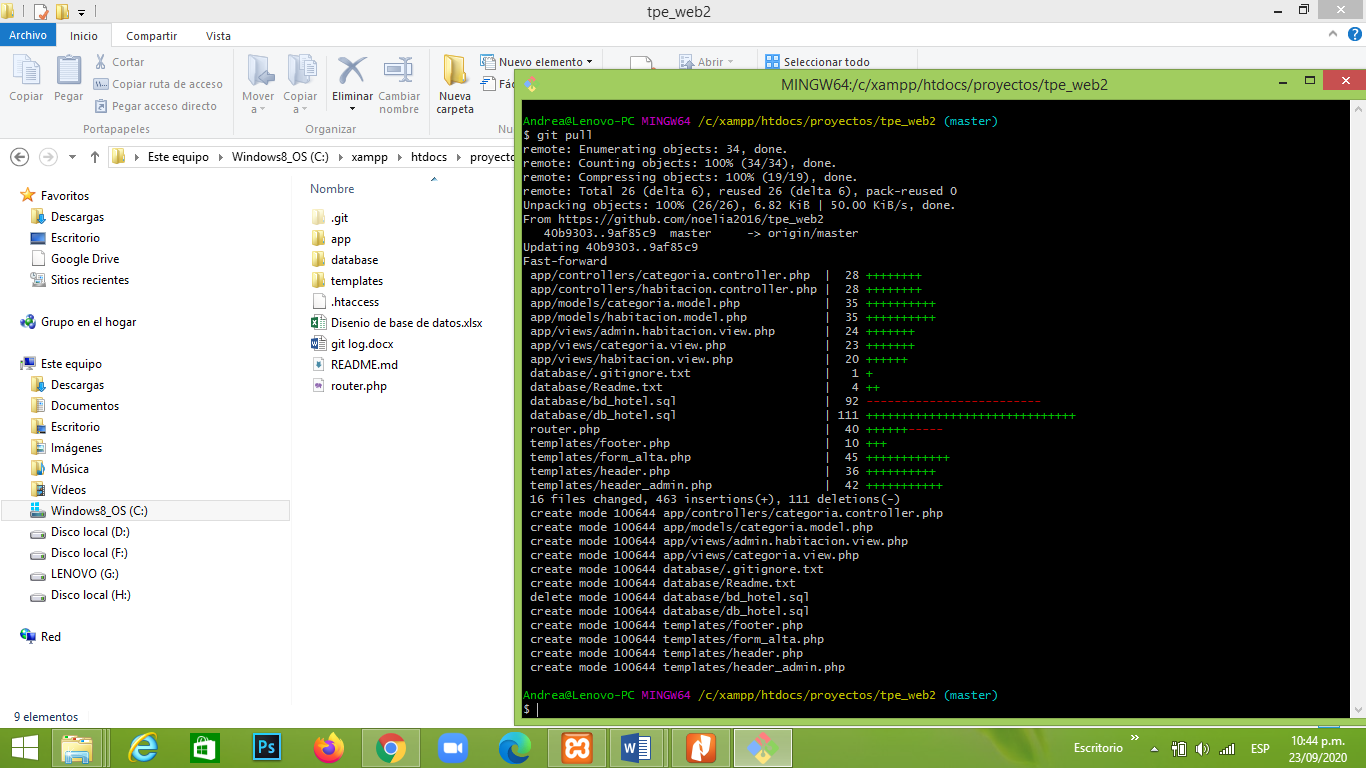
En un principio comenzamos a desarrollar el proyecto de Web II con la creación del repositorio realizando las siguientes acciones:

1. Nos registramos cada una de las integrantes en GitHub, instalamos el git en muestras Pcs y realizamos las configuraciones necesarias.
2. A continuación Noelia creó el repositorio en el remoto y se compartió el link para clonar el repositorio en cada computadora de forma local para así poder trabajar en conjunto y compartir más fácilmente.

Para clonar cada una en su pc PC realizamos lo siguiente en la consola del git:

*Git clone* [*https://github.com/noelia2016/tpe\_web2.git*](https://github.com/noelia2016/tpe_web2.git)



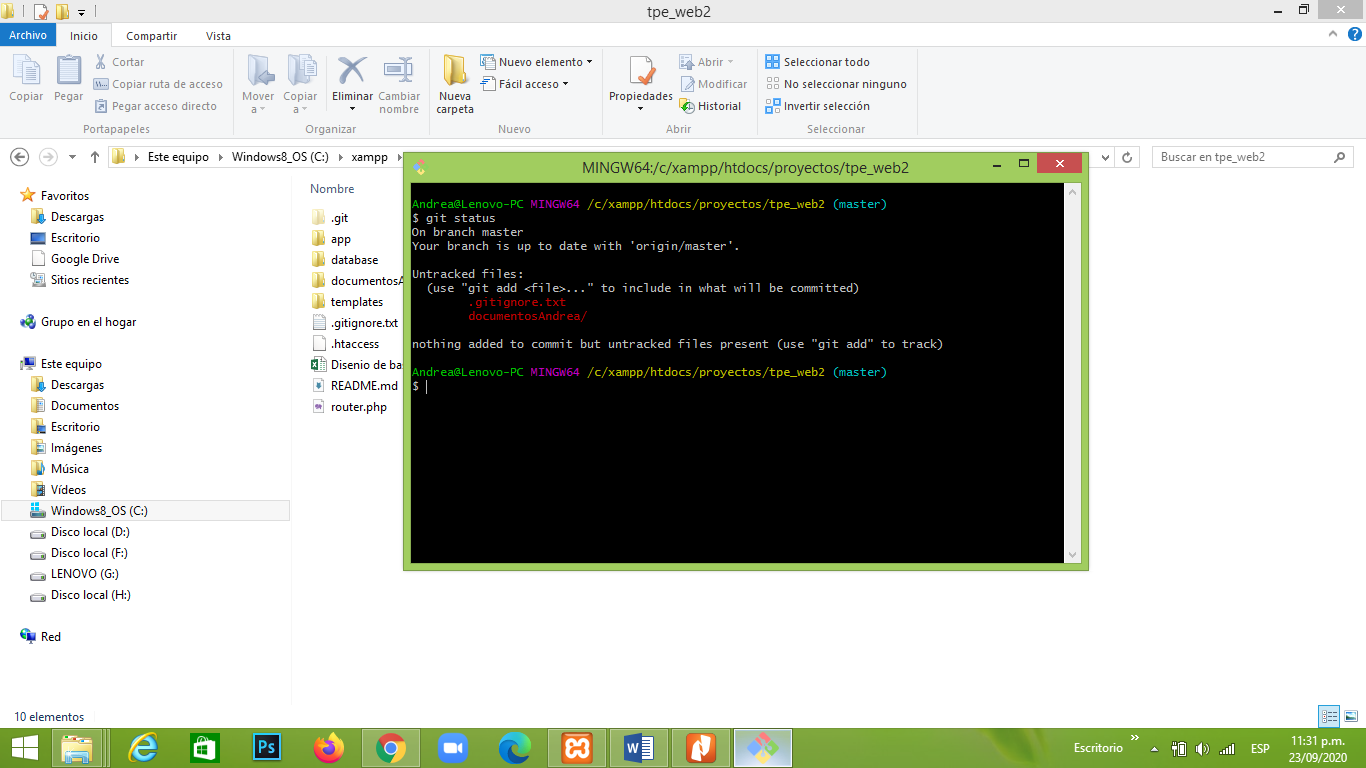


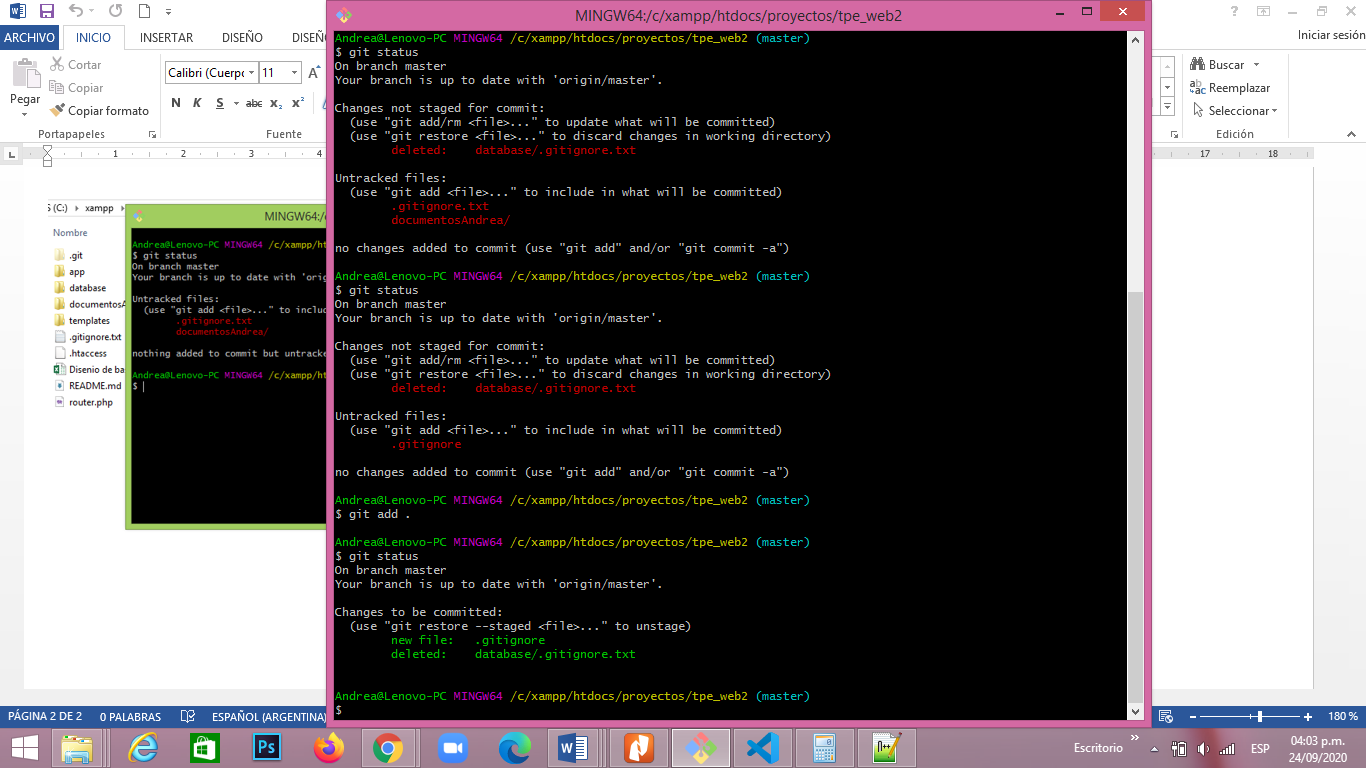
1. Después fuimos organizando como trabajar, por lo tanto una vez que ya teníamos más o menos la forma, Andrea creo el tablero en Trello para organizar las tareas y ponernos de acuerdo quien se encargaba de cada una. En ella configuramos las tareas más importantes que consideramos que le teníamos que hacer el seguimiento y algunas les pusimos fecha de vencimiento ya que necesitábamos tenerlo listo para avanzar en el proyecto con otras tareas relevantes.

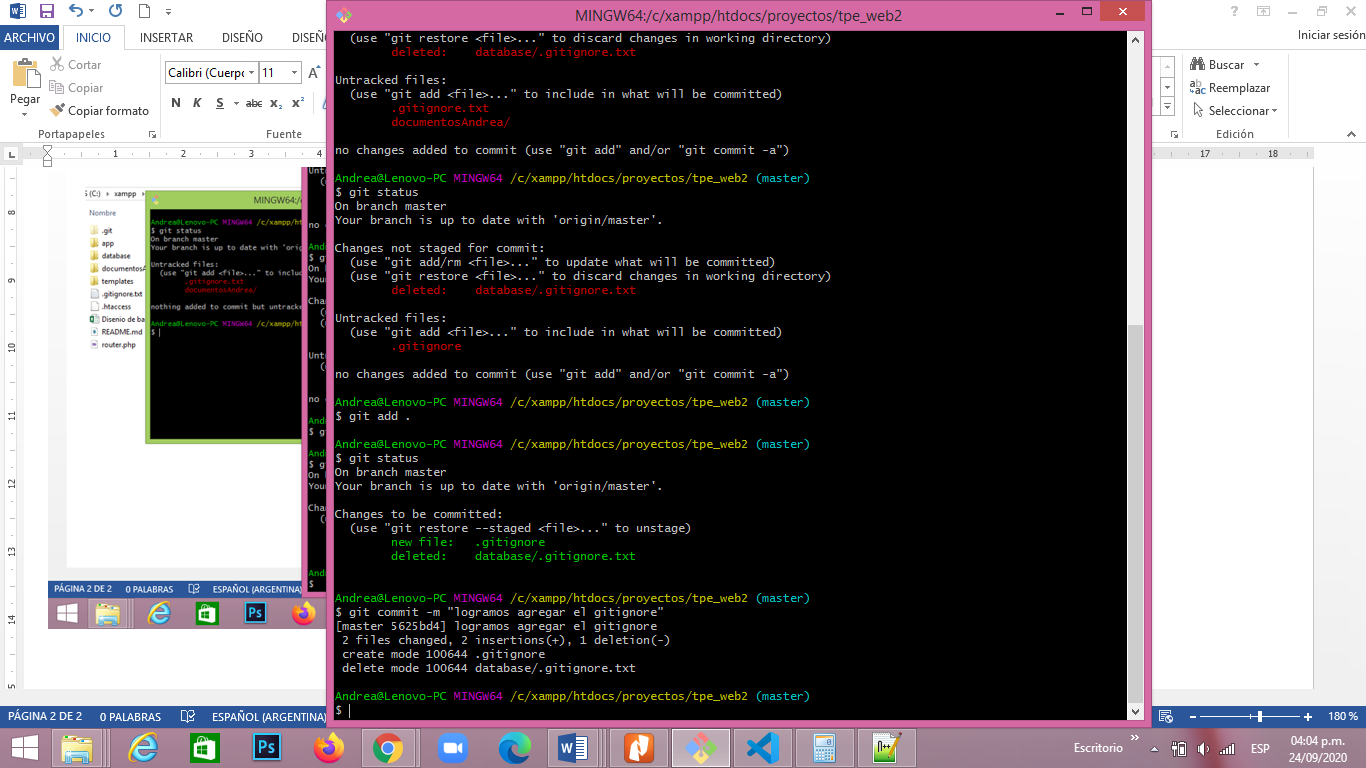
Las etiquetas elegidas representan las diferentes tecnologías y lenguajes que íbamos a utilizar para desarrollar el proyecto.

Las listas para la organización del tablero dejamos las que venían por defecto (Cosas que hacer, En proceso y Hecho) y agregamos una más para las tareas que estaban finalizadas pendientes de probar (Prueba).

1. Una vez que nos pusimos de acuerdo con la división de tareas del proyecto, creamos una rama local cada una para hacer lo correspondiente. Se comenzó a trabajar en la funcionalidad y utilizando el repositorio git para compartir los avances. A continuación contamos la experiencia que tuvimos con Git.
2. Acá empieza la experiencia de indicar a Git que ignore el seguimiento de un documento o una carpeta. Andrea en principio pensó que “.gitignore” era un comando de Git. Noelia lo había creado en una subcarpeta para ignorar un archivo puntual.





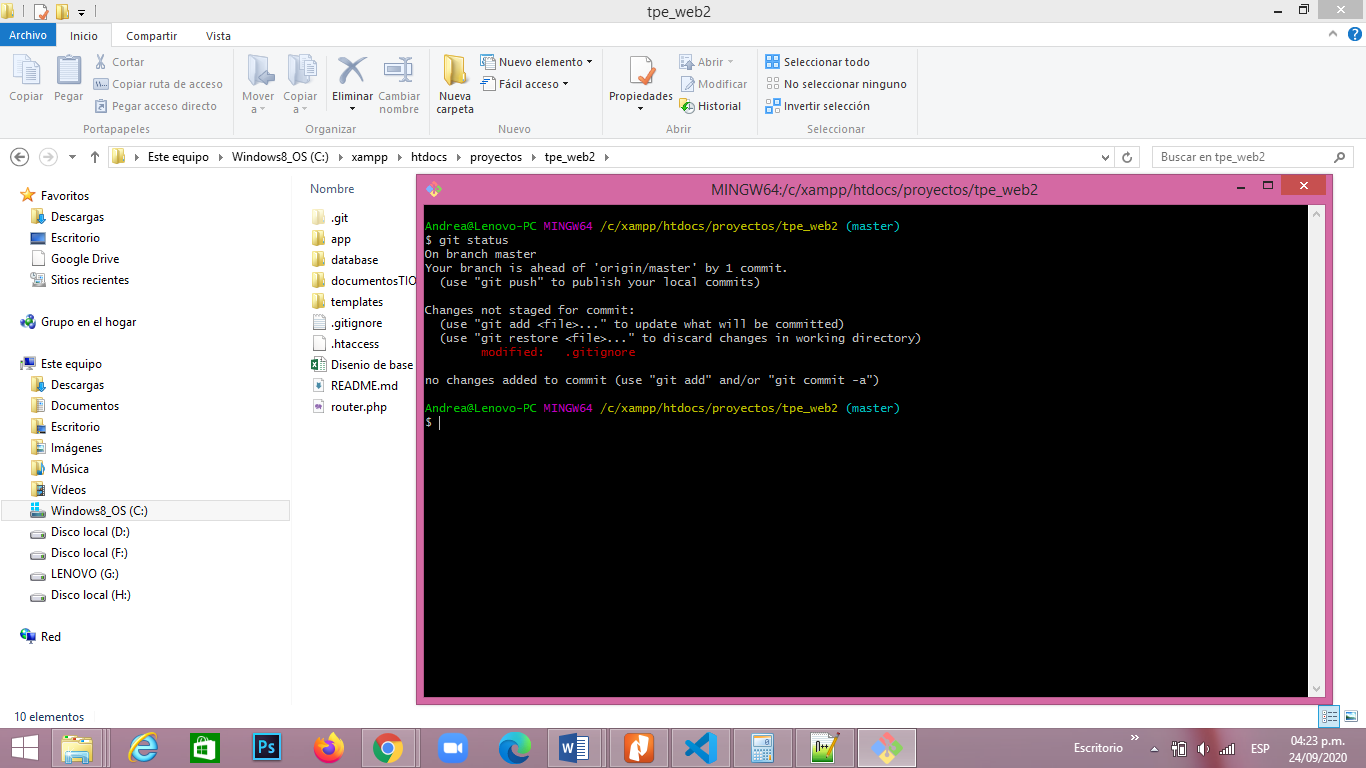


A Noelia le quedó mal el nombre dado que tenía extensión txt.

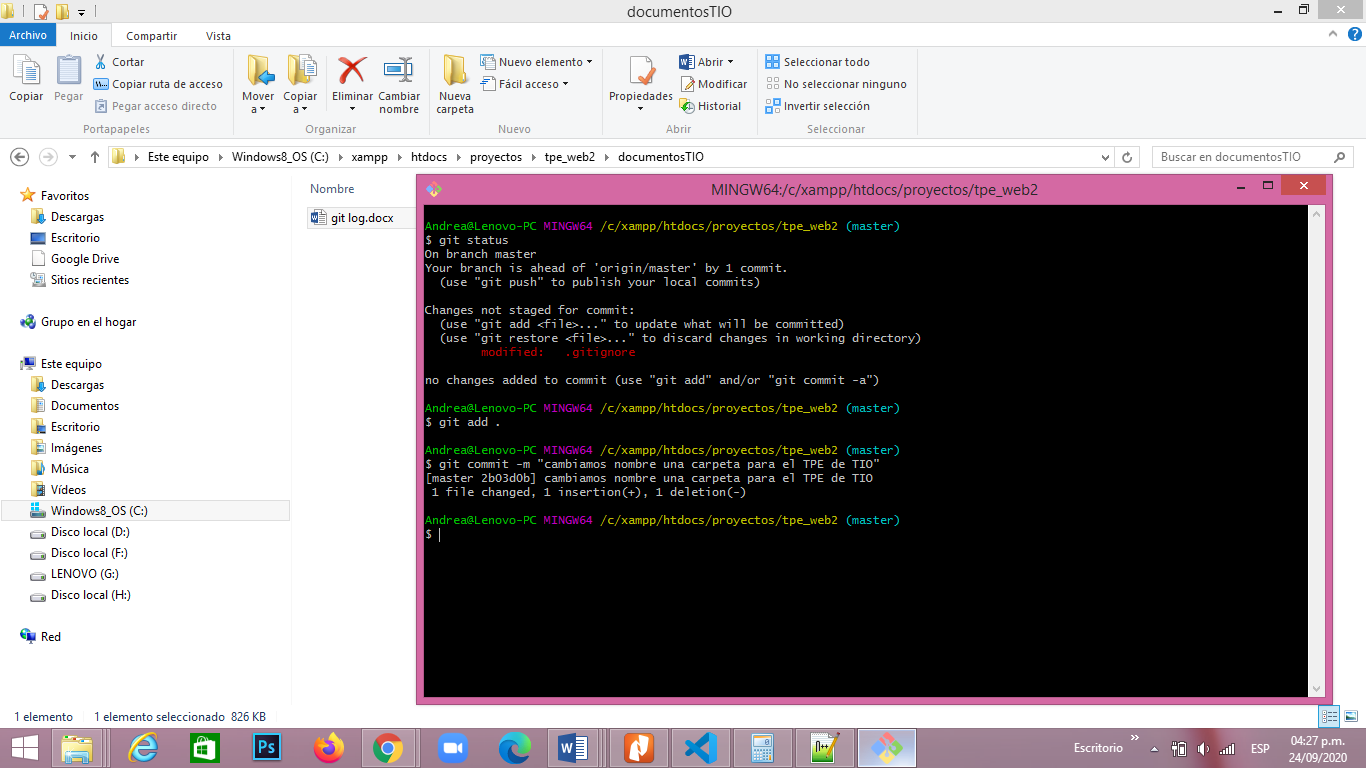
Decidimos crearlo nuevamente sin extensión y moverlo a la carpeta raíz del proyecto.

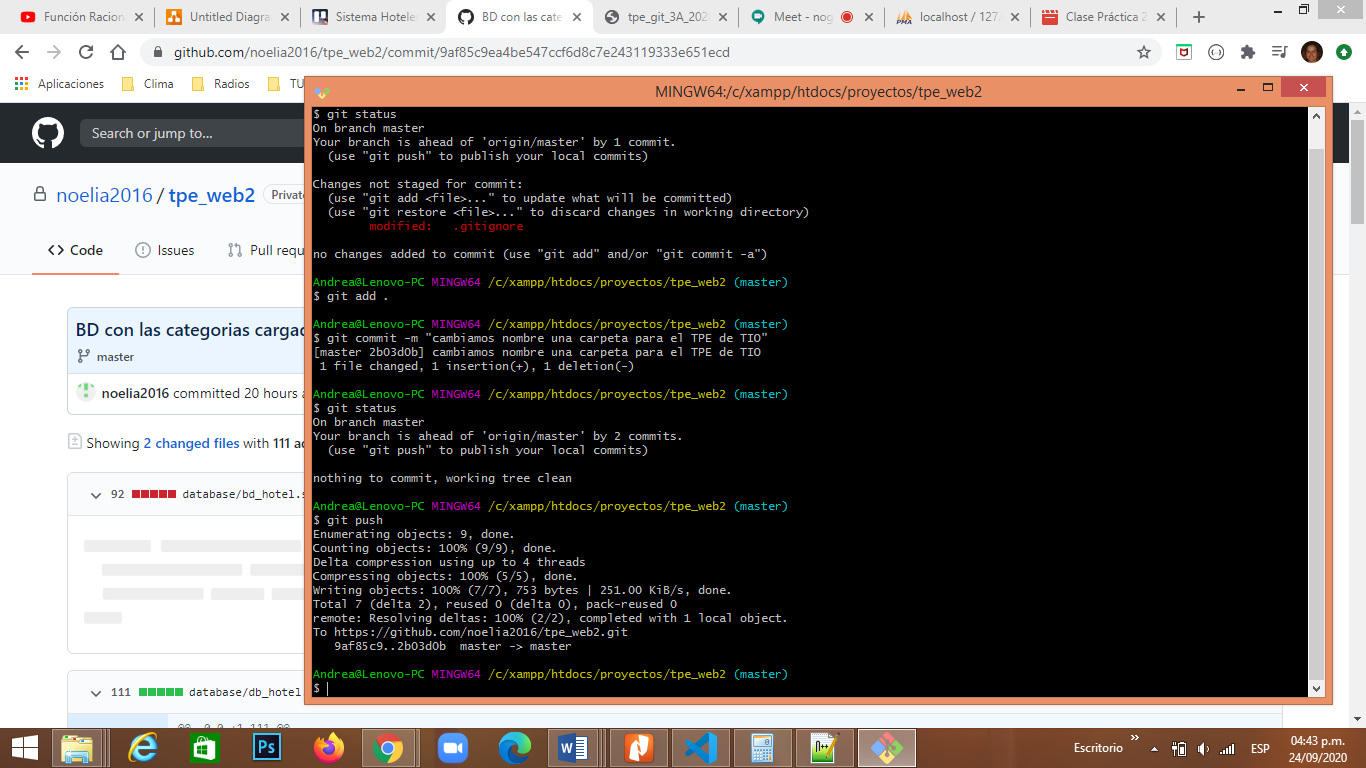
Cambiamos el nombre de la carpeta de documentos que antes habíamos incluido en el archivo “.gitignore”

Ejecutamos el comando “git status” y no nos avisa del cambio de nombre o borrado y agregado de la carpeta pero si nos indica que el archivo “.gitignore” se había modificado, dado que ya habíamos actualizado el nombre de la carpeta a ignorar en el mismo.



Agregamos el “.gitignore” a la lista de preparación, posteriormente hacemos el *commit* y *git* *push* para que quede actualizado el repositorio remoto con los últimos cambios.

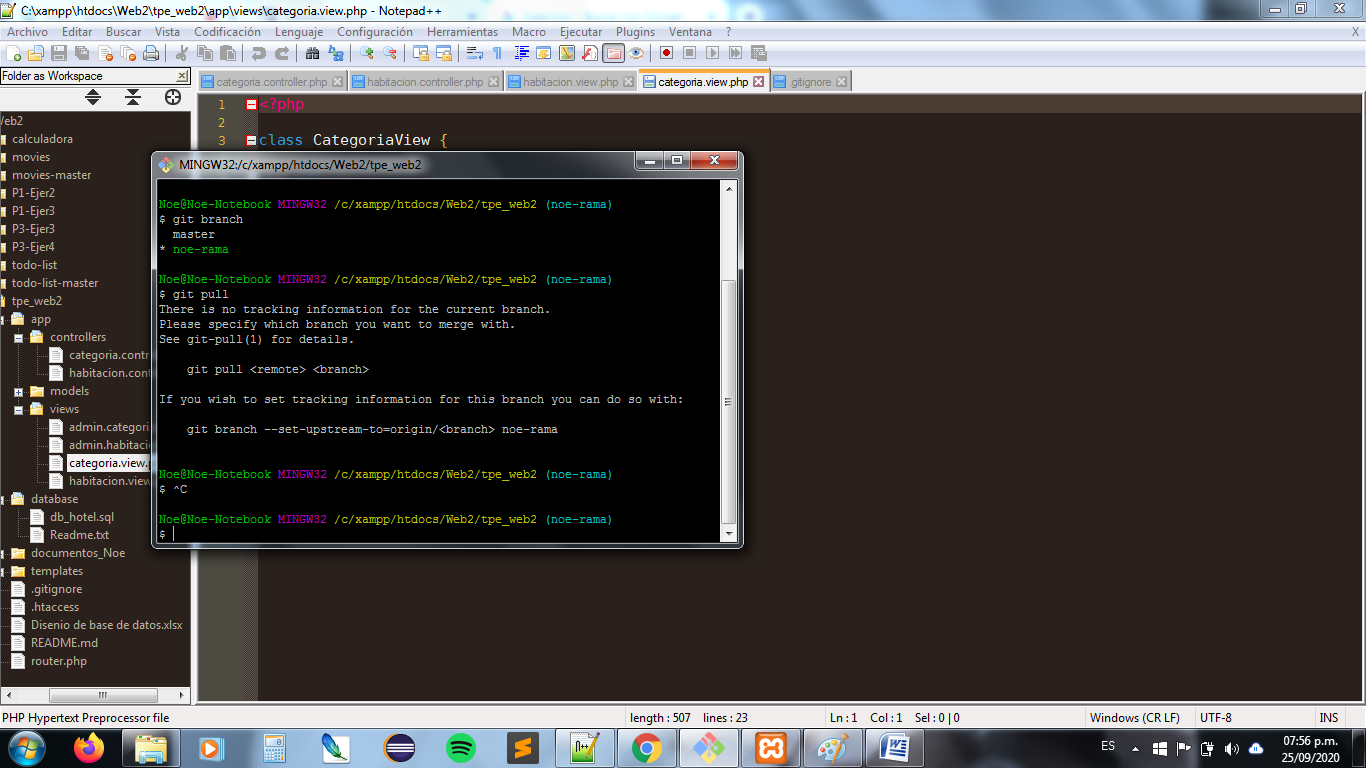




**En la creación de la rama Noelia le surgieron unos inconvenientes.**

Primero creó la rama **noe-rama** para trabajar con la funcionalidad del visitante de nuestro sitio con el siguiente comando: *git branch noe-rama*

Luego de crearla se acordó que Andrea había modificado un par de cosas que no tenía actualizado en el repositorio, por ende ejecutó *git pull* en donde estaba posicionada que era dentro de la rama donde tiene unos inconvenientes que solucionar.

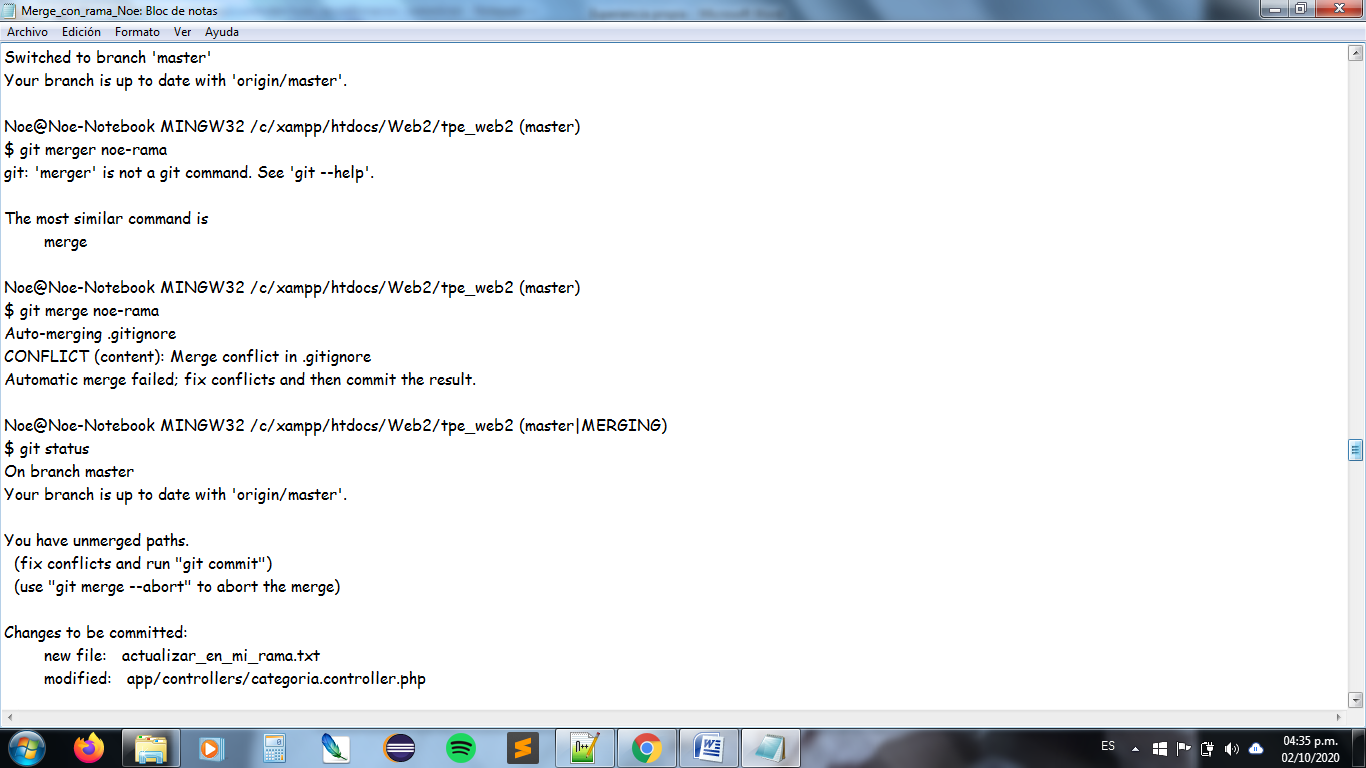


Le indicó como sugerencia en la consola lo siguiente:   
No hay información de seguimiento para la rama actual. Especifique con qué rama desea fusionarse y le recomendó que lo haga en la rama que estaba, con el siguiente comando:

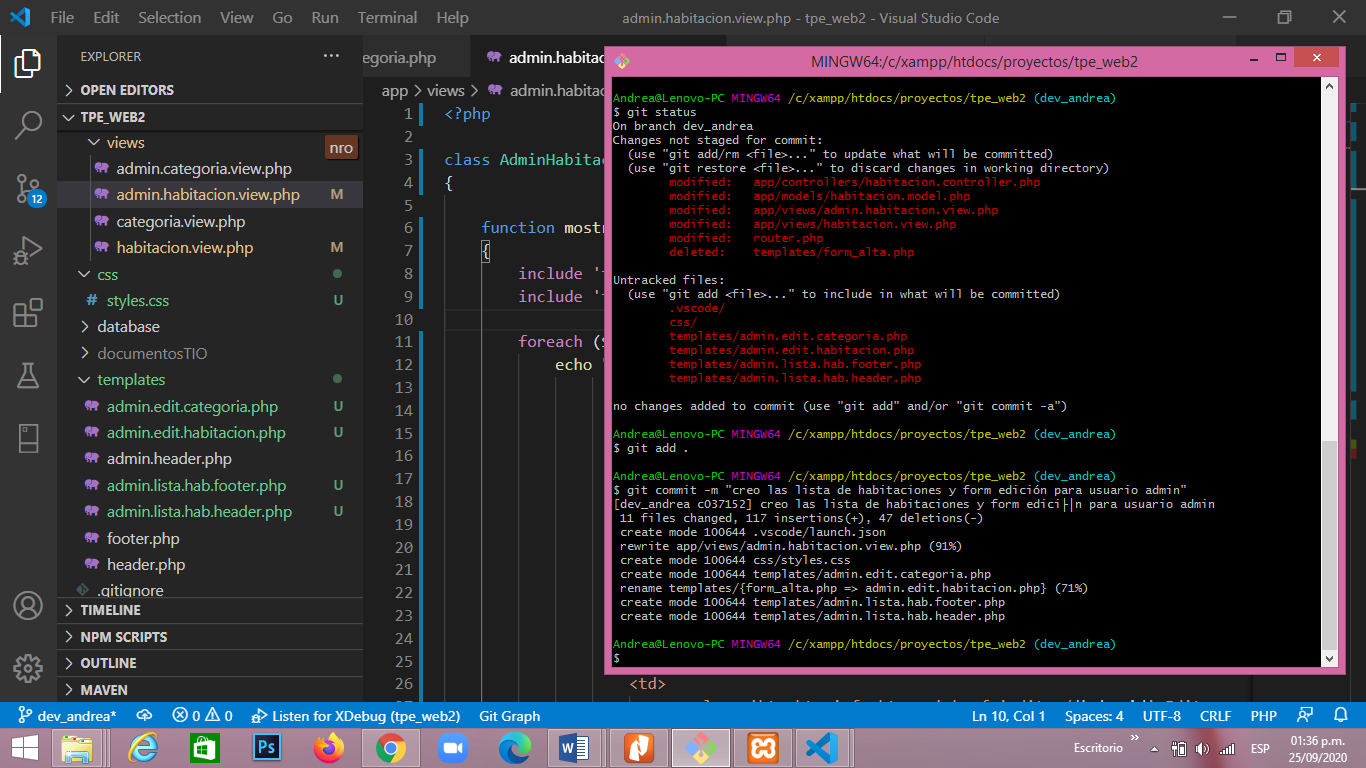
***git branch --set-upstream-to=origin/noe-rama***

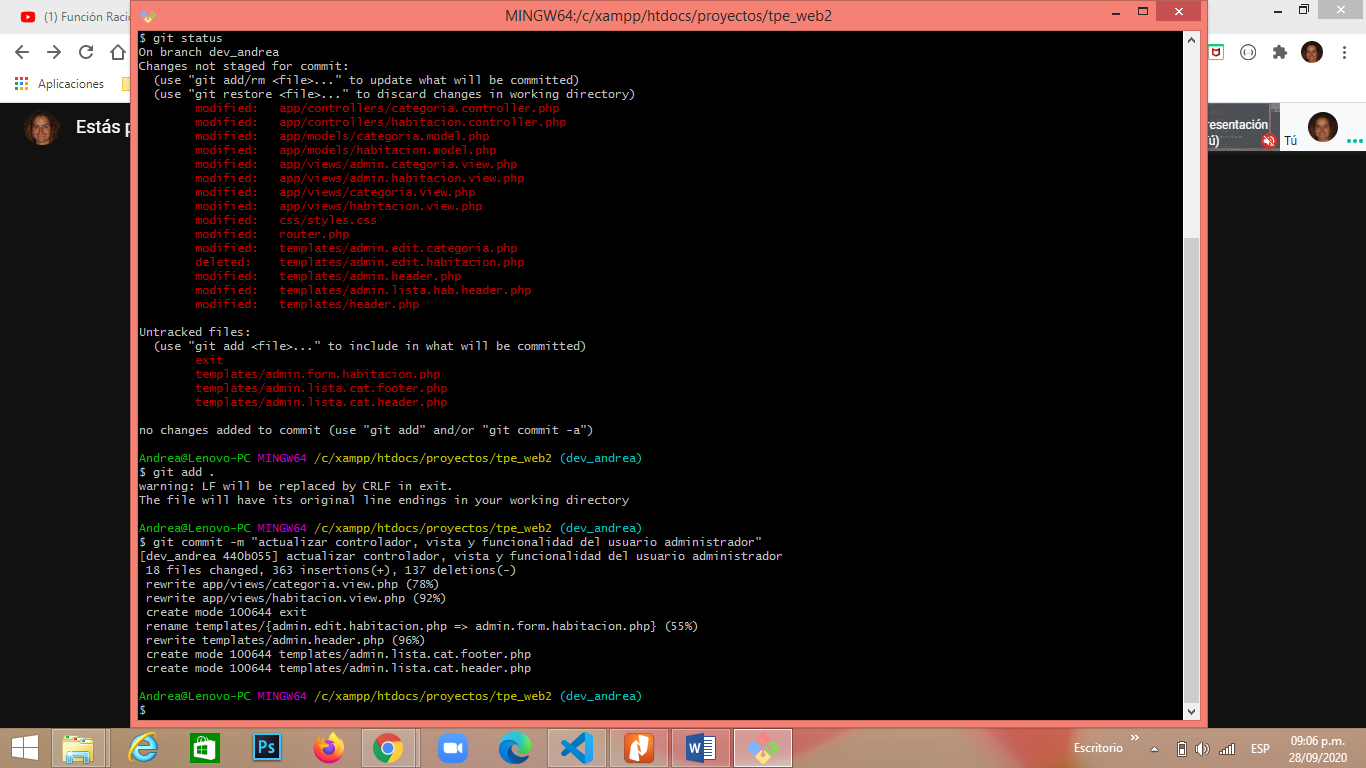
Siguiendo su consejo ejecutó ese comando y actualizó la rama, no contenta con dicha acción hizo el *git add, git commit y git push* estando posicionada en dicha rama por ello se le creó de forma remota en el repositorio la rama que solo quería local. De ahí aprendió a aprestar un poco de más atención!!!!

Noelia siguió trabajando con la rama actualizando la funcionalidad que necesitaba y finalmente hizo el *merge* para que Andrea viera lo que pasaba. Se generaron dos conflictos que se solucionaron fácilmente y pude salir de estado de merging.

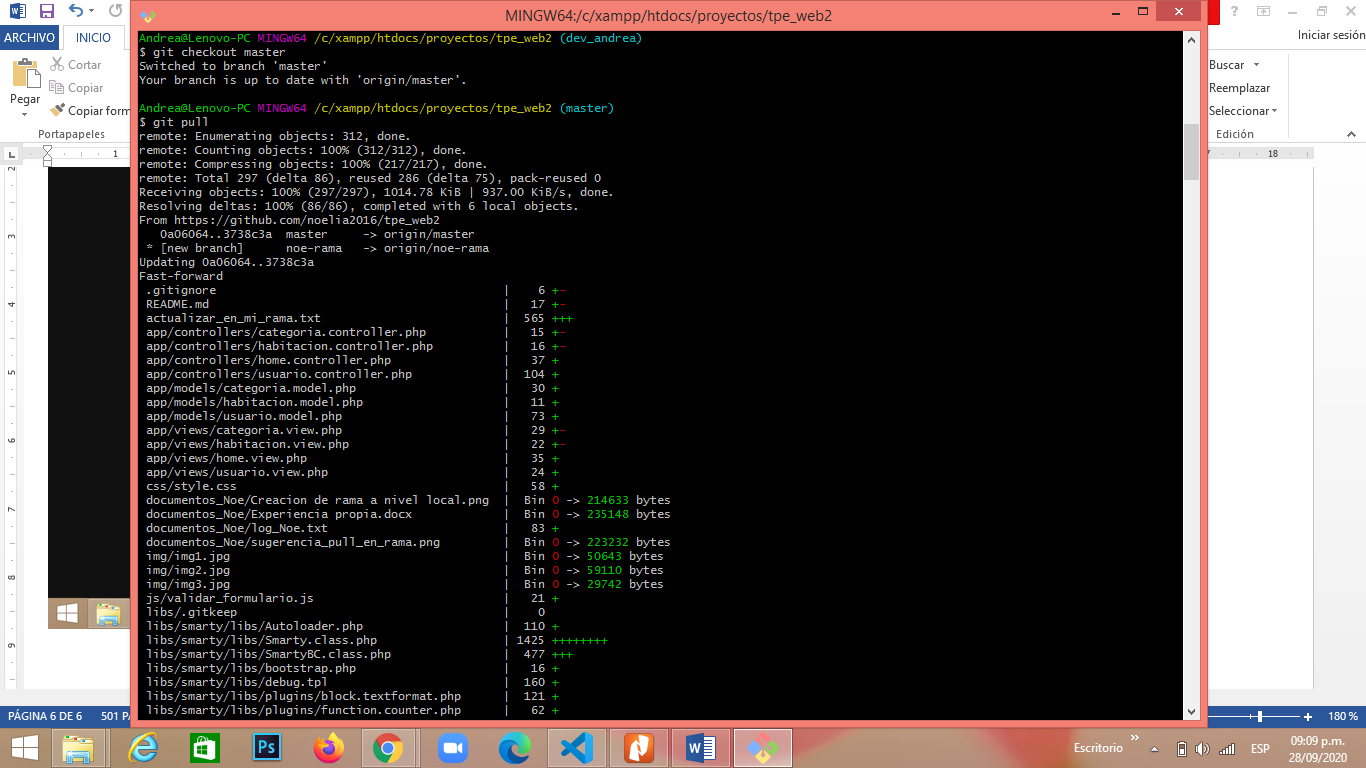


Andrea crea una rama local para desarrollo. Implementa una lista de habitaciones y formulario de edición para vista de usuario Admin y hace el commit de en esa rama.





Andrea hace el pull del repositorio remoto a su repositorio local

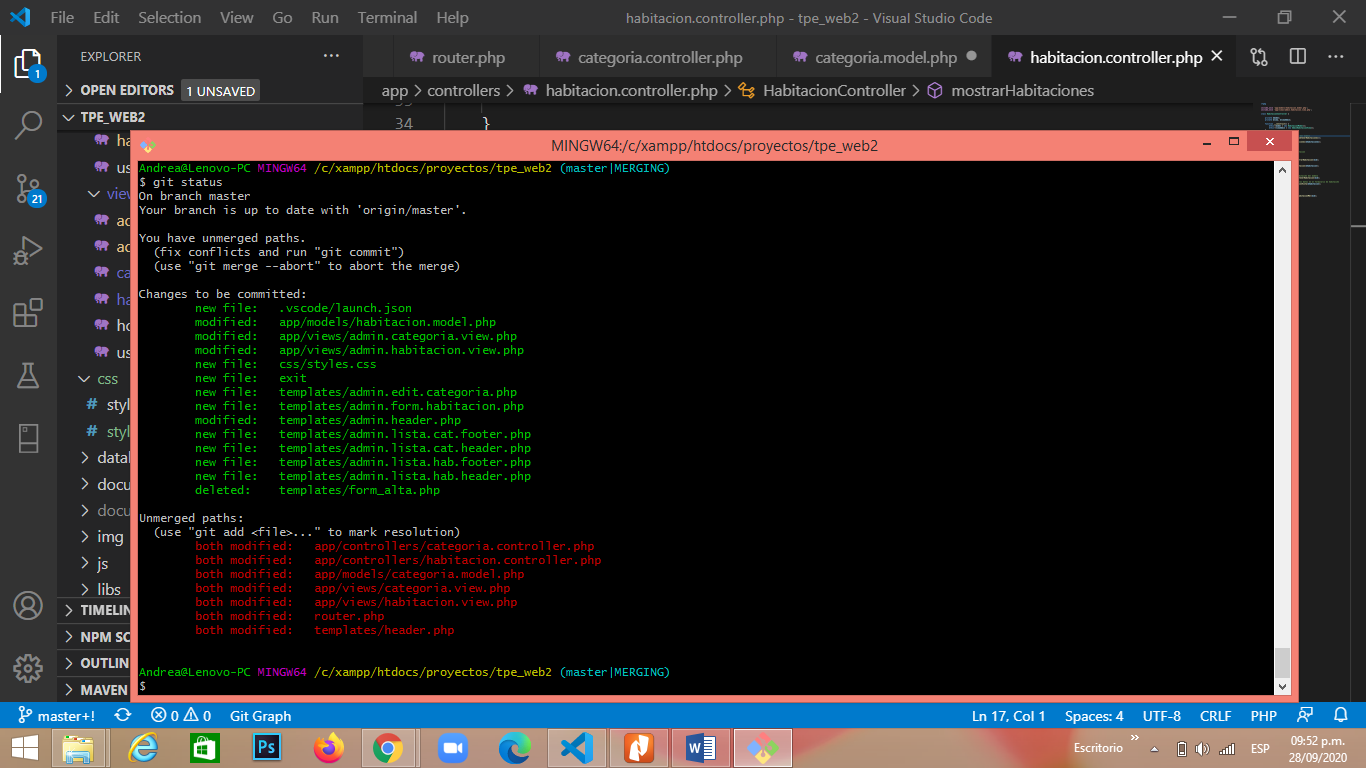


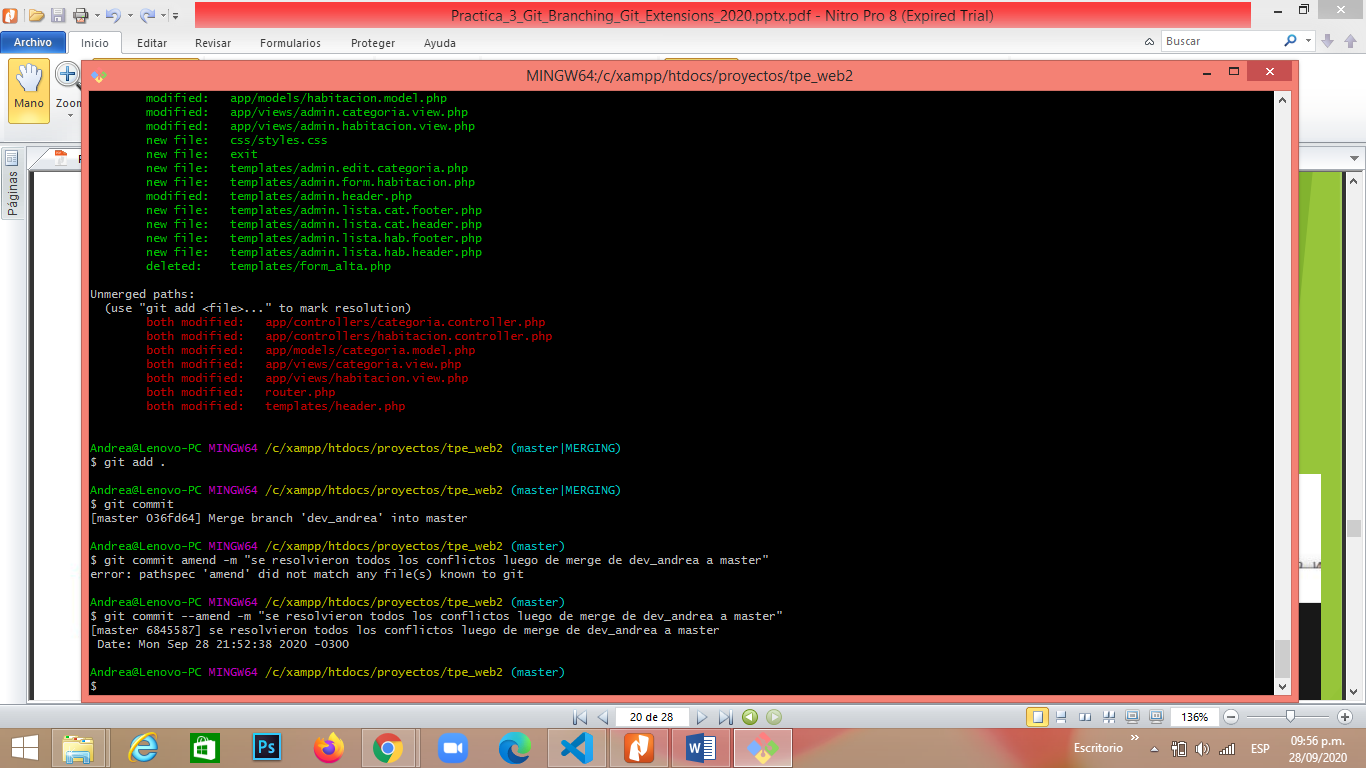


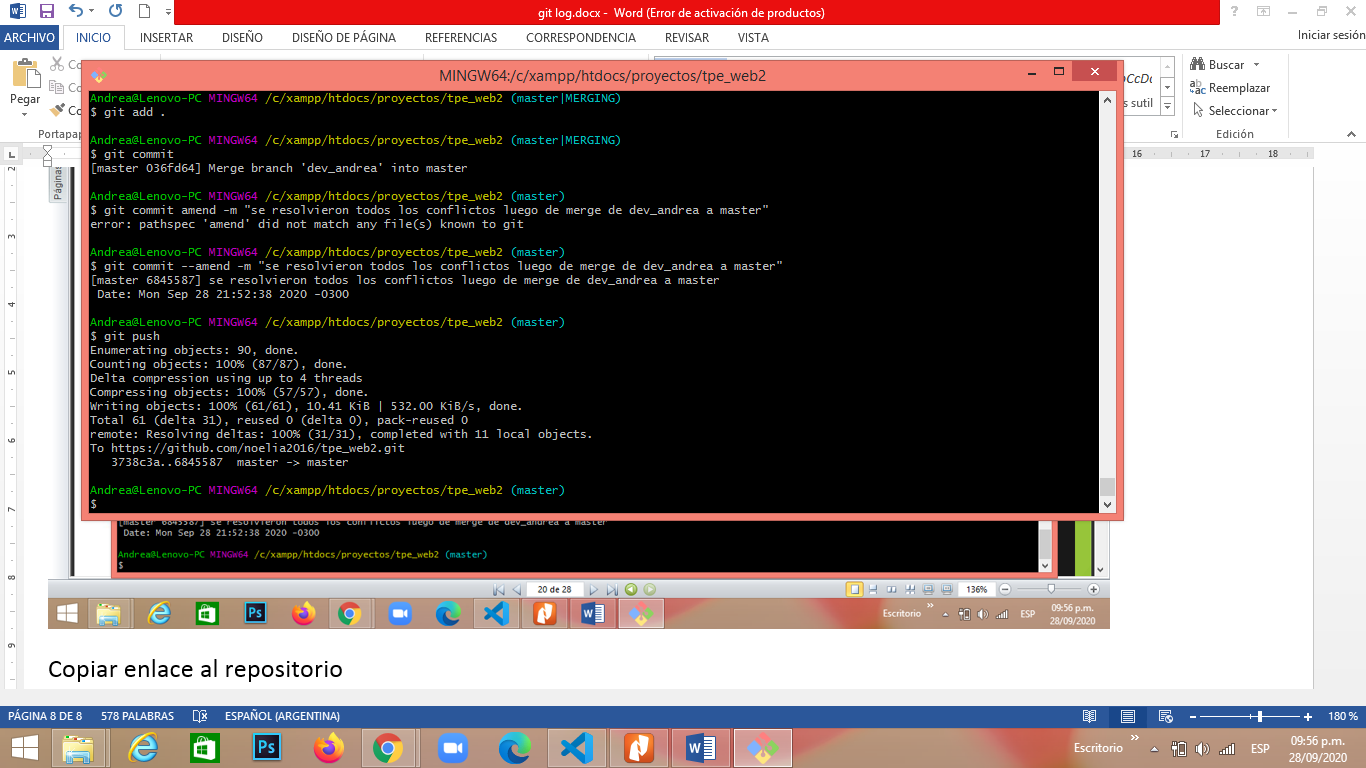
Lo que no fue para nada fácil cuando Andrea unificó las ramas de ella con master ya que Noelia había cambiado muchos archivos y habíamos usado los mismos 5 archivos ambas dándole funcionalidad diferente al sitio, ya que estábamos trabajando en ramas distintas locales y no queríamos hacer *commit* hasta que no termináramos una funcionalidad completa.

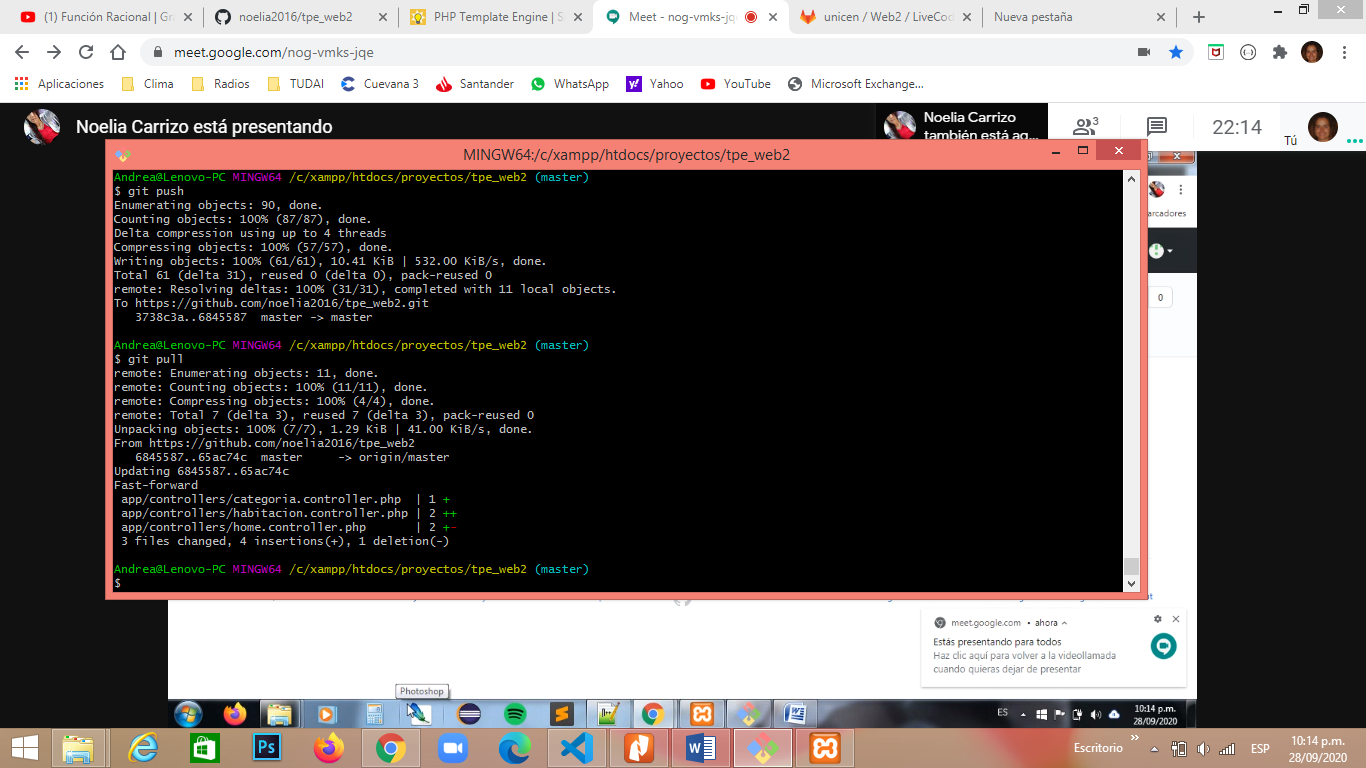
**Se detectaron varias modificaciones**

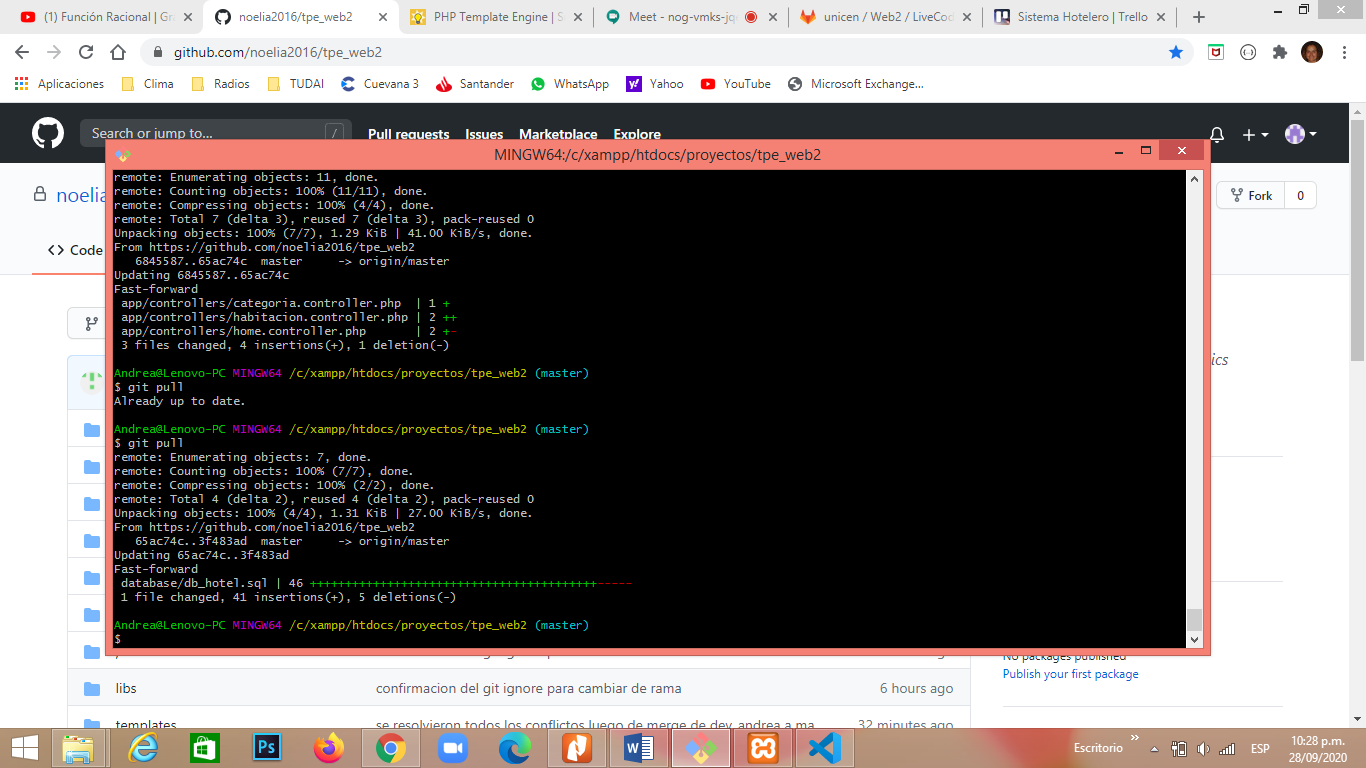
Hubo varios conflictos que resolver al hacer el *merge* de rama *master a dev\_andrea*. El más numeroso fue el router.php para lo cual se corrigió utilizando la opción “*Accept both changes”*





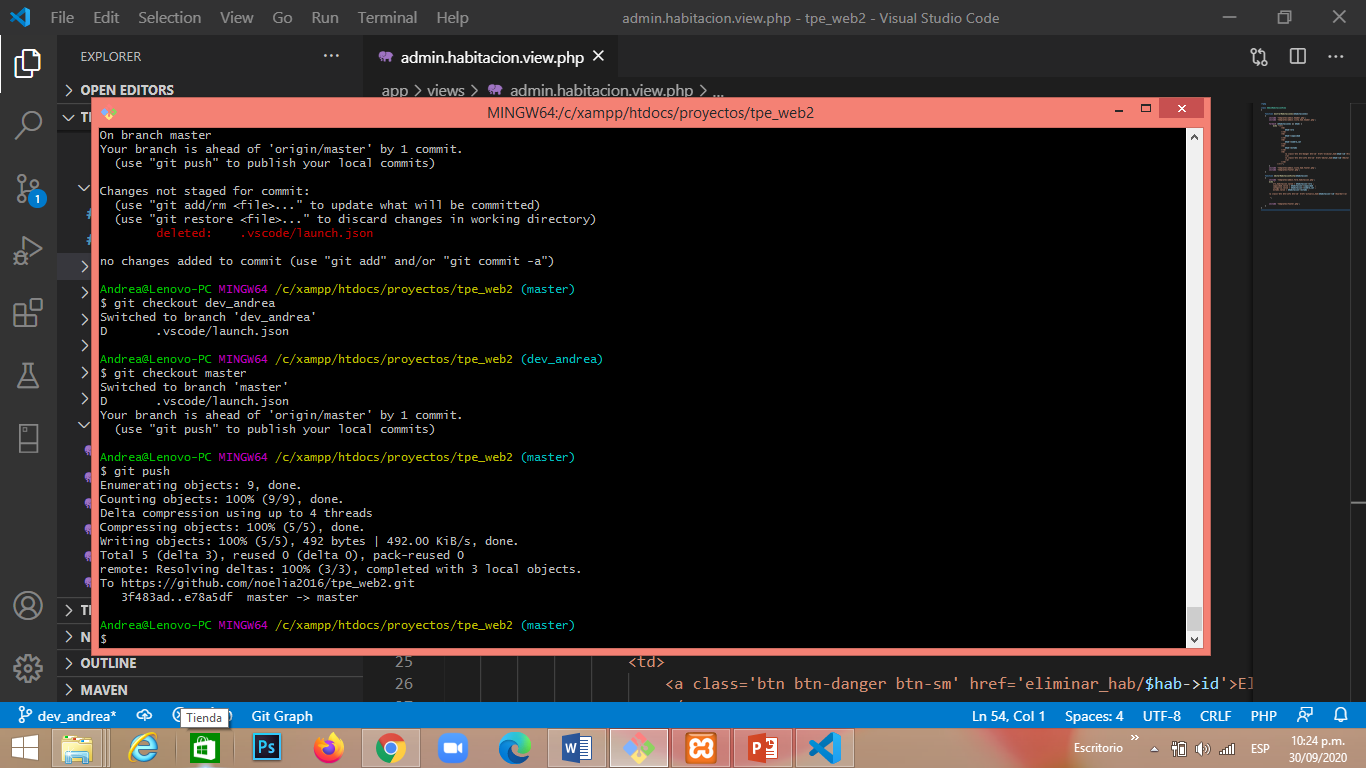




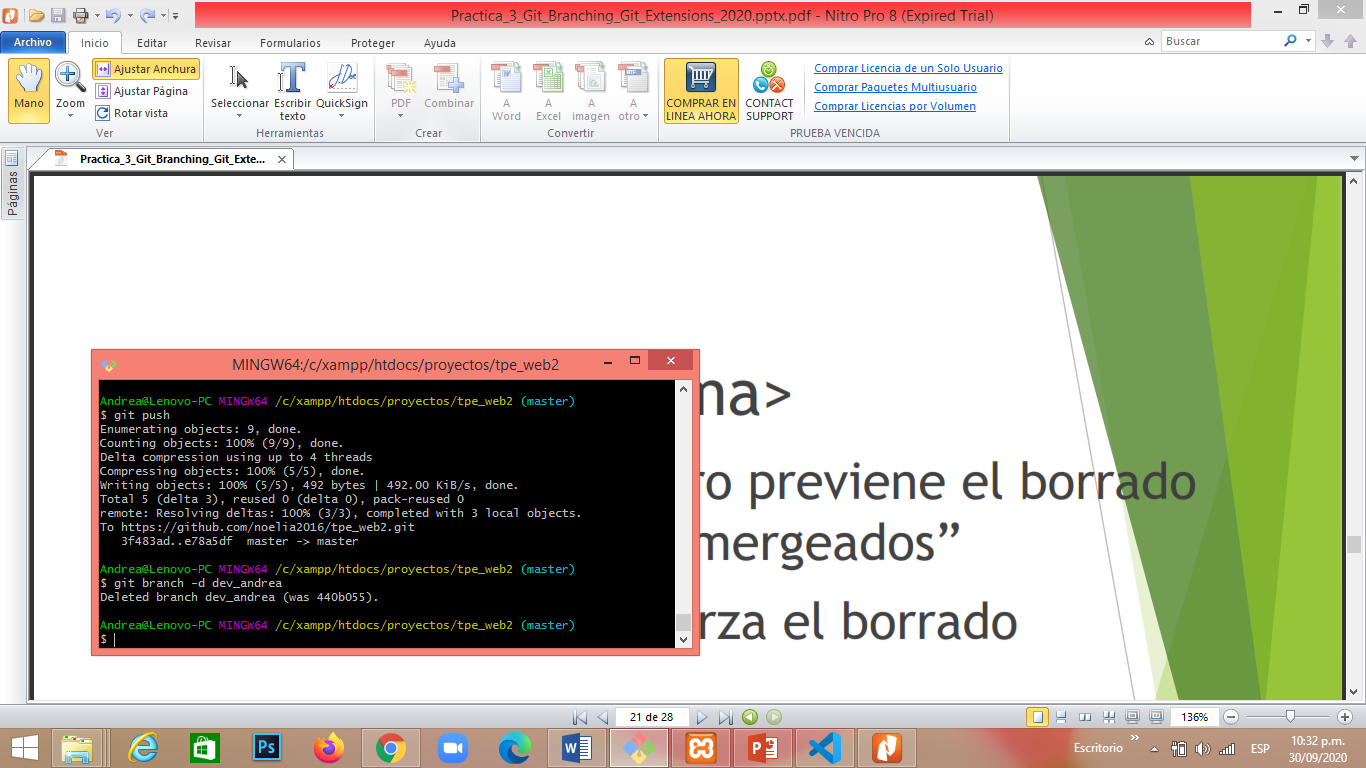


A partir de esta experiencia decidimos realizar actualizaciones más periódicas del repositorio remoto y acotadas a pequeñas nuevas funcionalidades.

Andrea hace *push* al repositorio remoto solo porque borró un archivo erróneo de VSCode



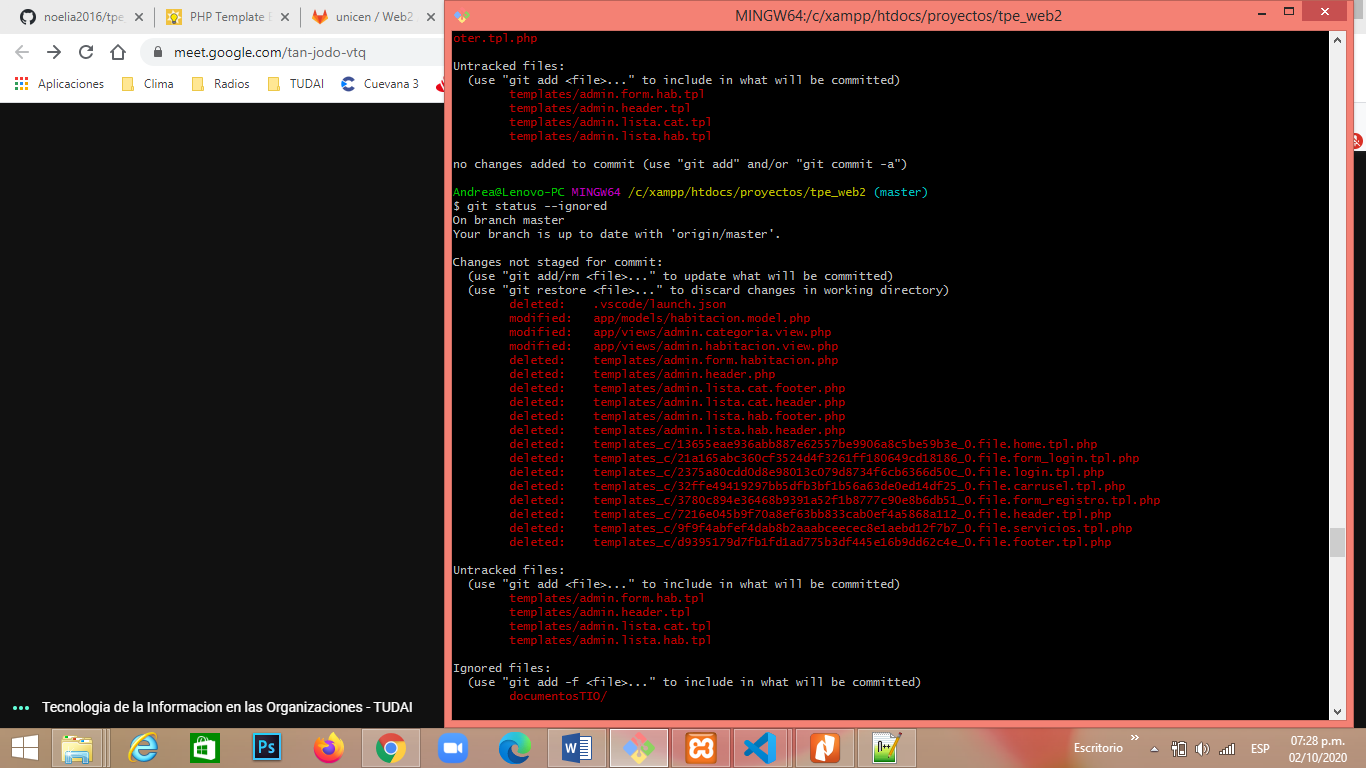
Andrea borra la rama de desarrollo local dado que había actualizado el repositorio remoto luego de finalizar la etapa de desarrollo comprometida para la misma.

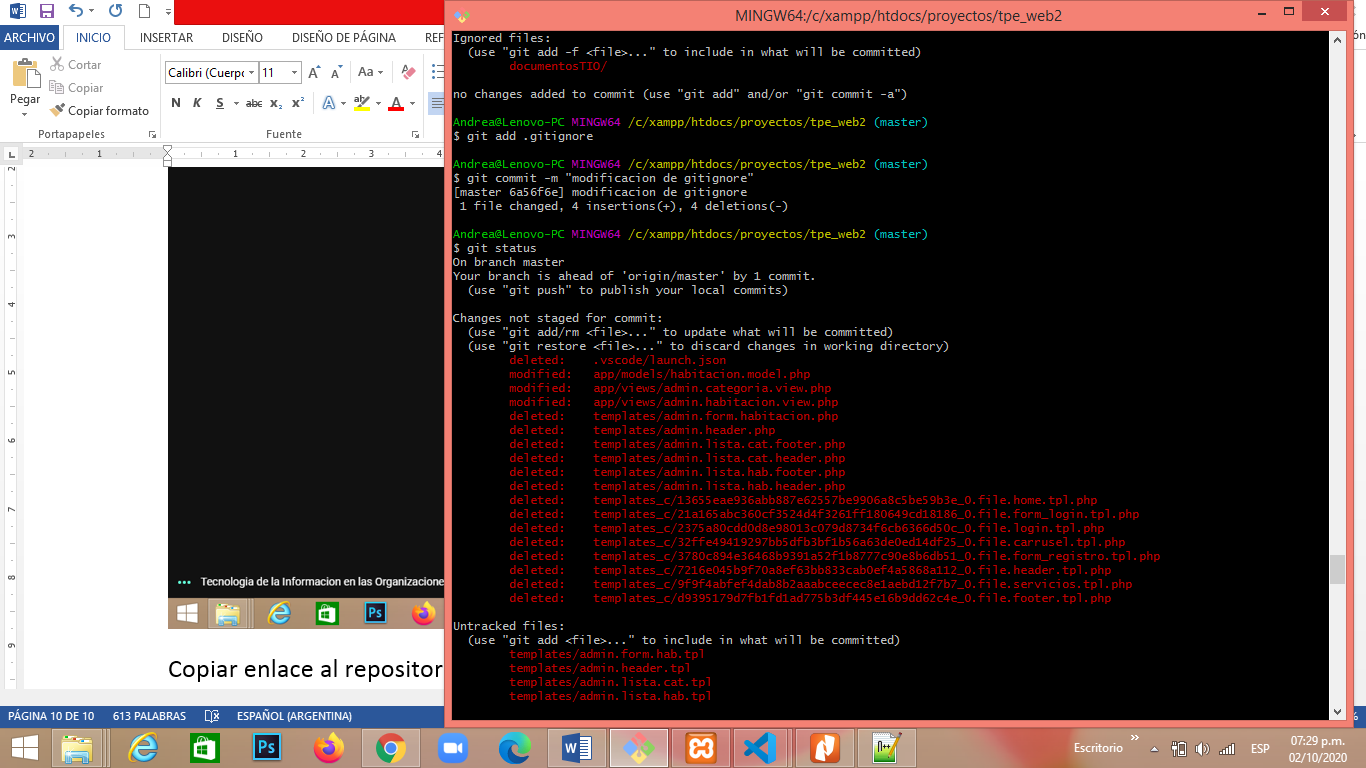


Luego Noelia creó una nueva rama local para la funcionalidad del abm de usuarios y la dejó local no hizo el *push* que es lo que había generado unos cuantos inconvenientes en los días previos y nos generó dudas respecto a que era lo que teníamos realmente en el repositorio. De errores se aprende dicen no!!! Ahora Noelia sigue trabajando en ella.

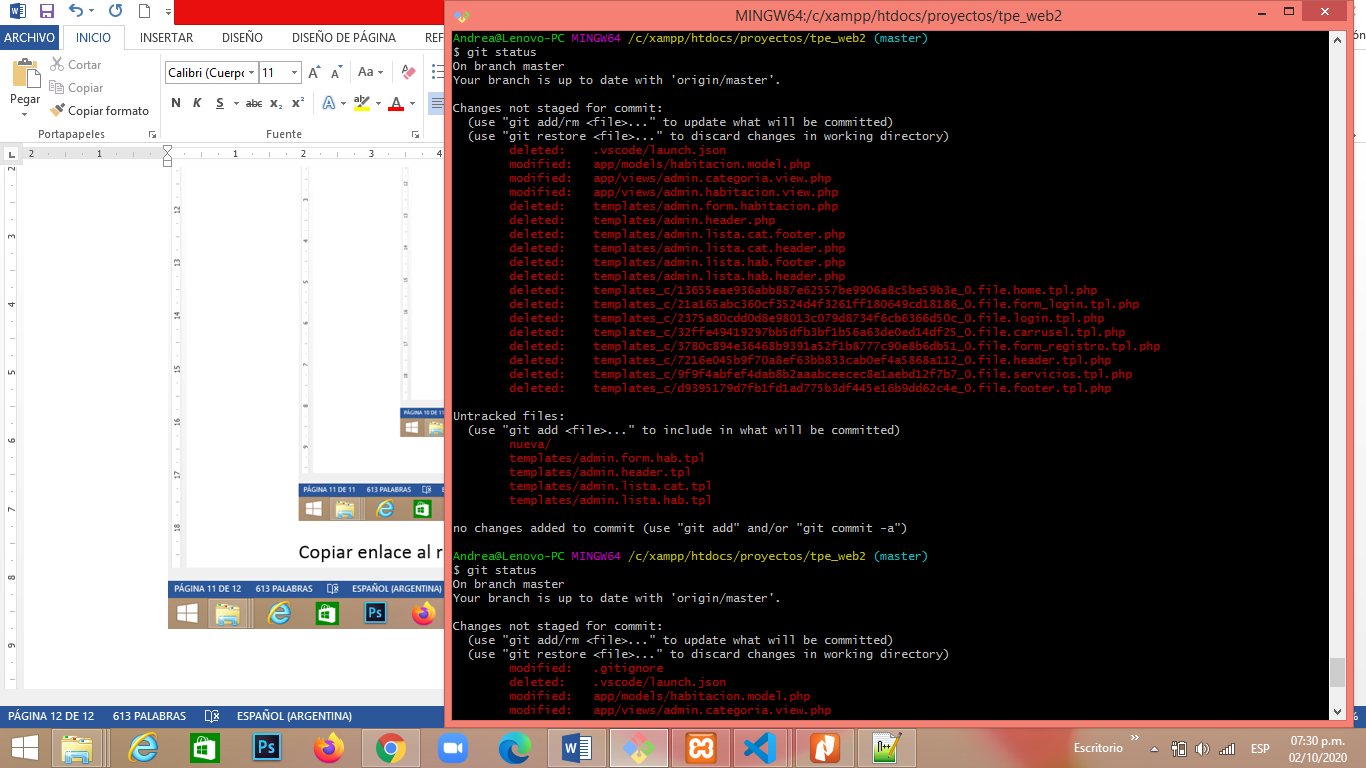
En la clase del 02/10/20 expusimos nuestras dudas sobre *.gitignore* por estar mal generado, creiamos que no estaba ignorando carpetas y probamos compartiendo pantalla. Concluimos que estabamos siguiendo archivos que habiamos pedido ignorar pero que seguramente los habiamos empezado a seguir con anterioridad.

Utilizando el comando *git status –ignored* verificamos que era solo una carpeta la que se estabamos ignorando.

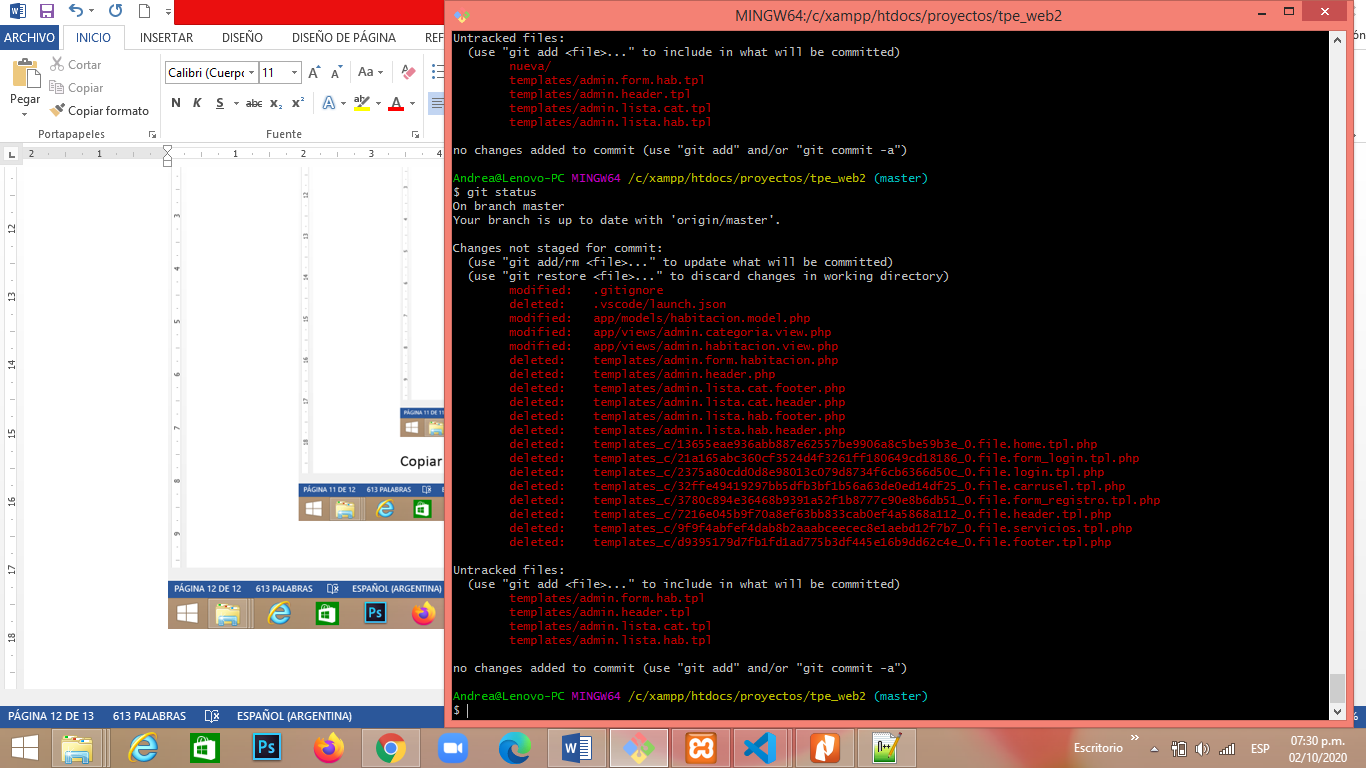




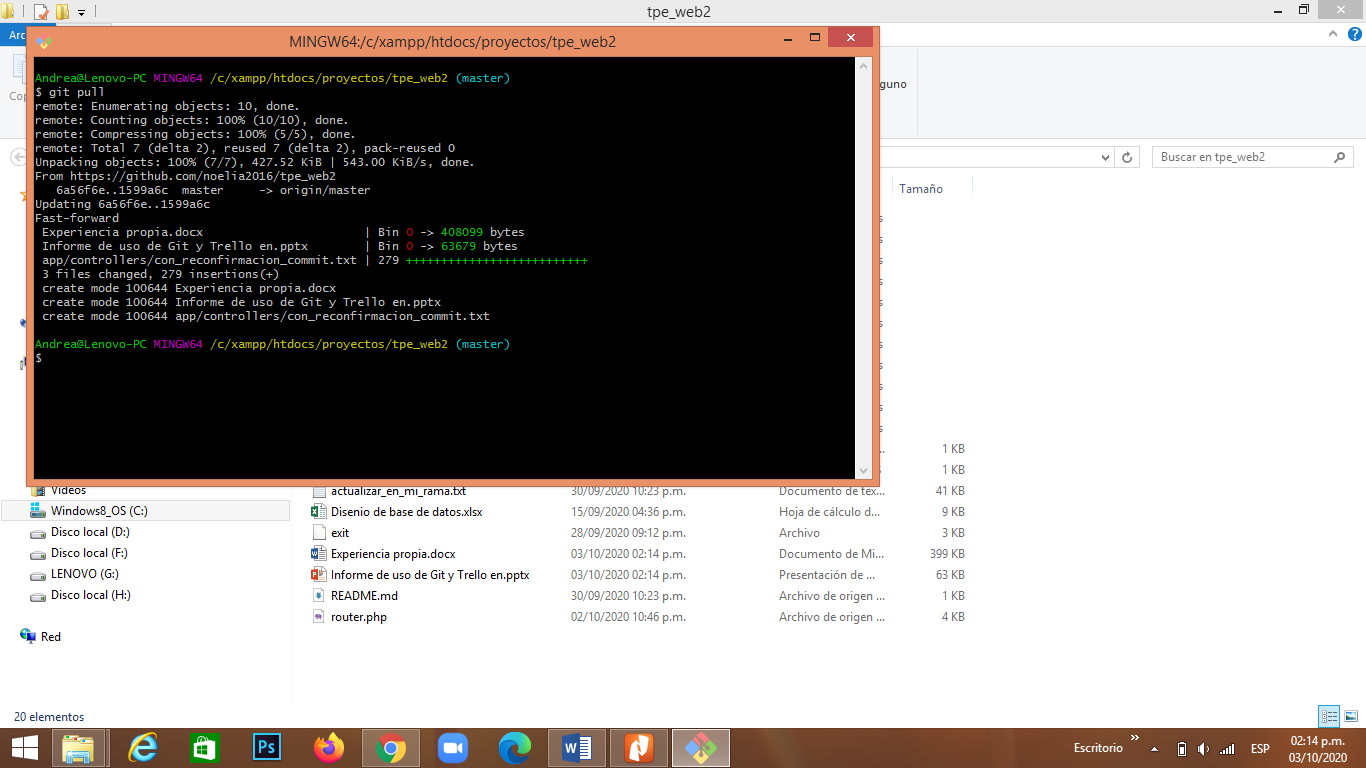
Creamos una “nueva” carpeta en el repositorio local y la agregamos a *.gitignore*



Primero el *git status* nos indica que esa carpeta está sin seguimiento. subimos el archivo *.gitignore* al repositorio remoto habiendo agregado la carpeta y posteriormente ejecutamos *git status* para comprobar que ya no nos indica que no está bajo seguimiento.



Nos reunimos para confeccionar el informe y la presentación para el TPE de TIO, con lo cual Noelia subió los archivos y Andrea hizo un *git pull* para descargarlos y trabajar con ellos.



Eliminamos archivos txt que estaban por error en el repositorio utilizando el comando *git rm*

