Tecnología de la información en las organizaciones: Trabajo práctico especial

Fecha de entrega: 15 de octubre de 2019

Integrantes: • Duca Bordignon, Horacio Marcelo

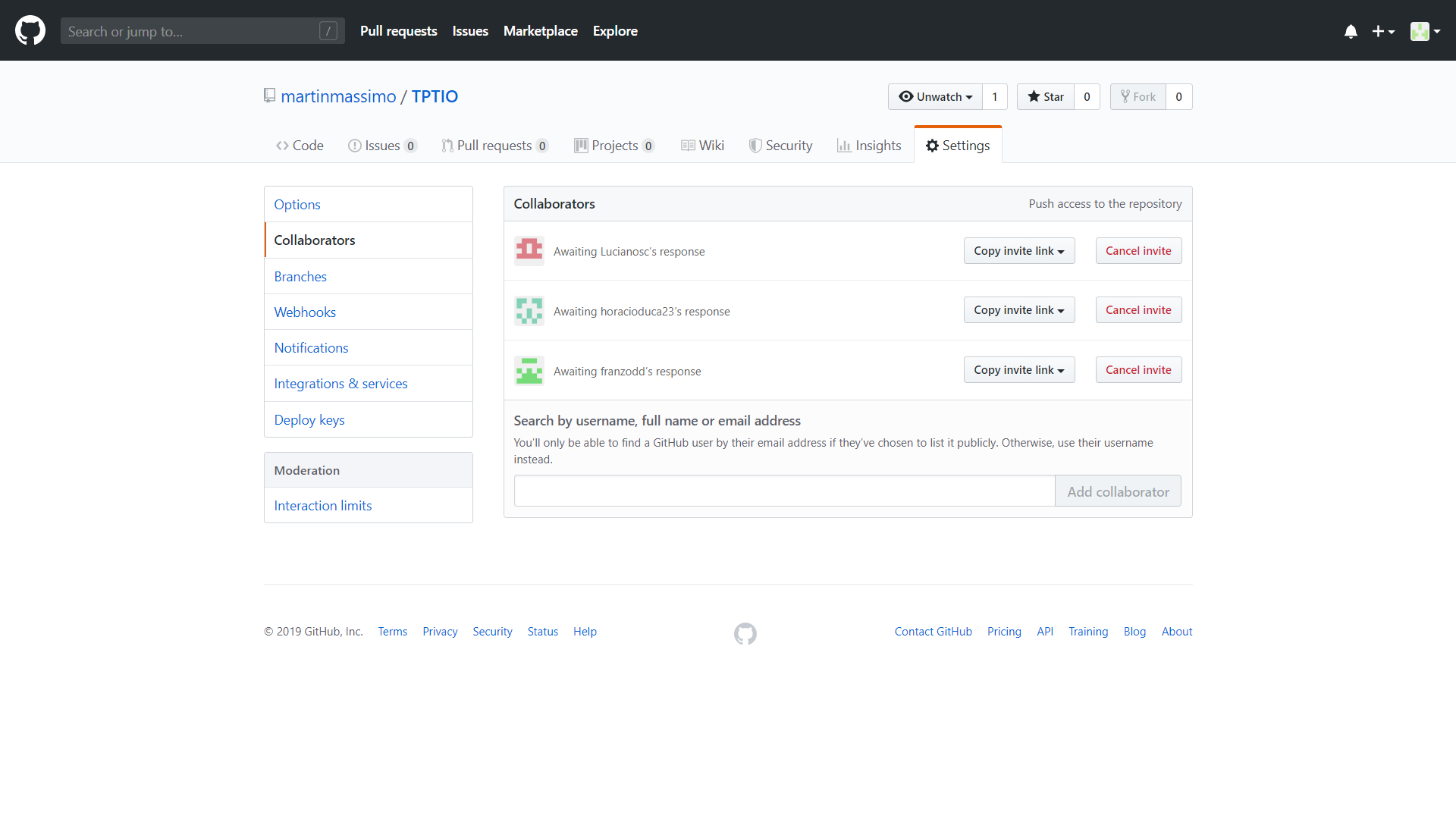
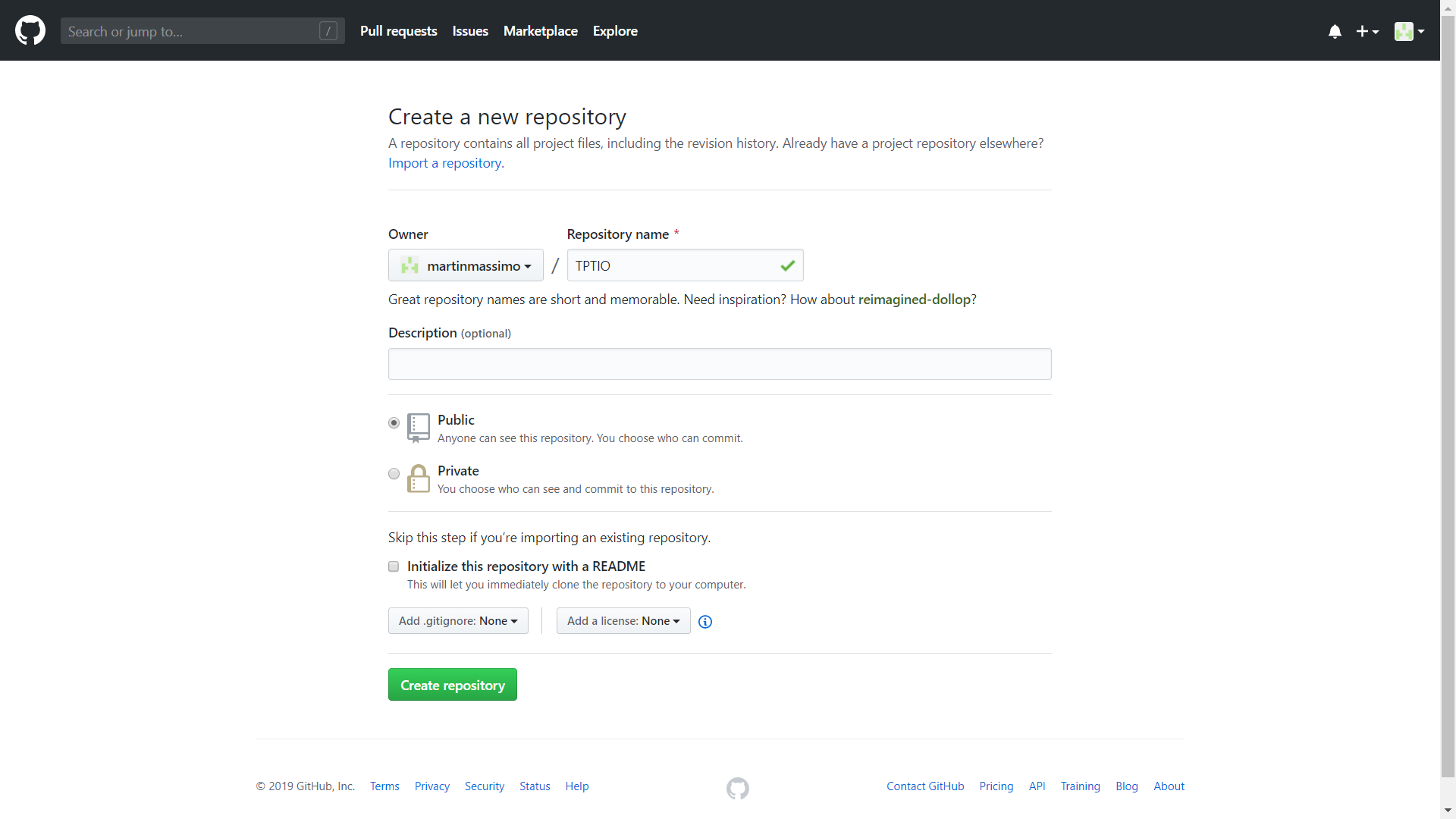
• López, Francisco Daniel

• Massimo, Martín

• Scaminaci, Luciano

Introducción

Para comenzar, lo que hicimos fue crear un repositorio público en GitHub con dominio en <https://github.com/martinmassimo/TPTIO> y luego agregar a cada uno de los integrantes del equipo a colaboradores. En el repositorio mencionado se encuentran alojados los archivos que fuimos manipulando para la realización de éste trabajo.

*Creando un nuevo repositorio público y añadiendo colaboradores al nuevo repositorio “TPTIO”.*

Decidimos crear 5 archivos con diferentes extensiones para el trabajo, en el primero README.md, se encuentran algunas anotaciones a modo de ayuda memoria para la organización.

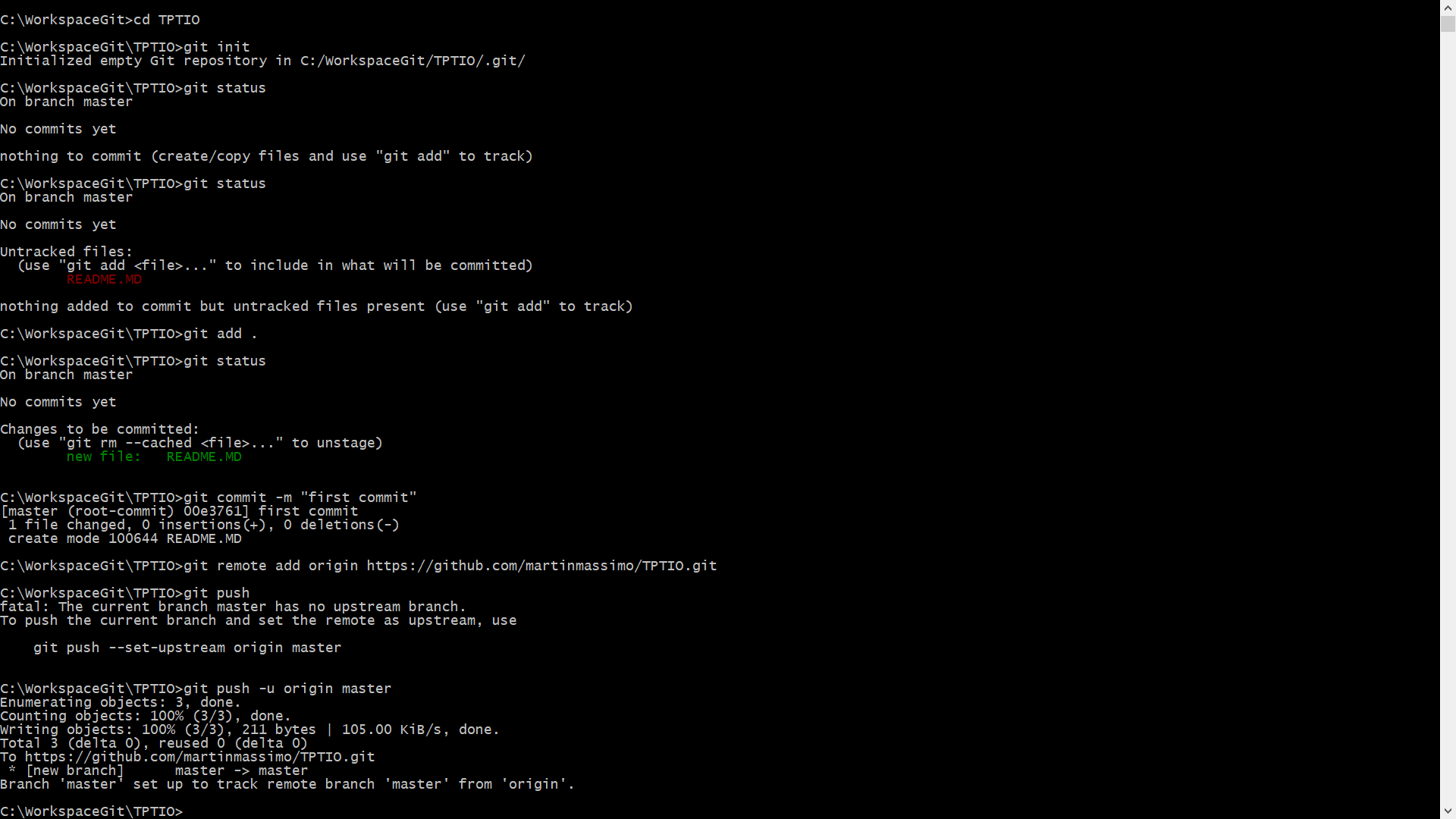
En el segundo, index.html, se encuentra el código principal para la creación de la página web que decidimos trabajarlo en la rama “estructura”.

En el tercer archivo, style.css se puede encontrar la edición y diseño de la pagina web, la cual trabajamos en la rama “estilo”.

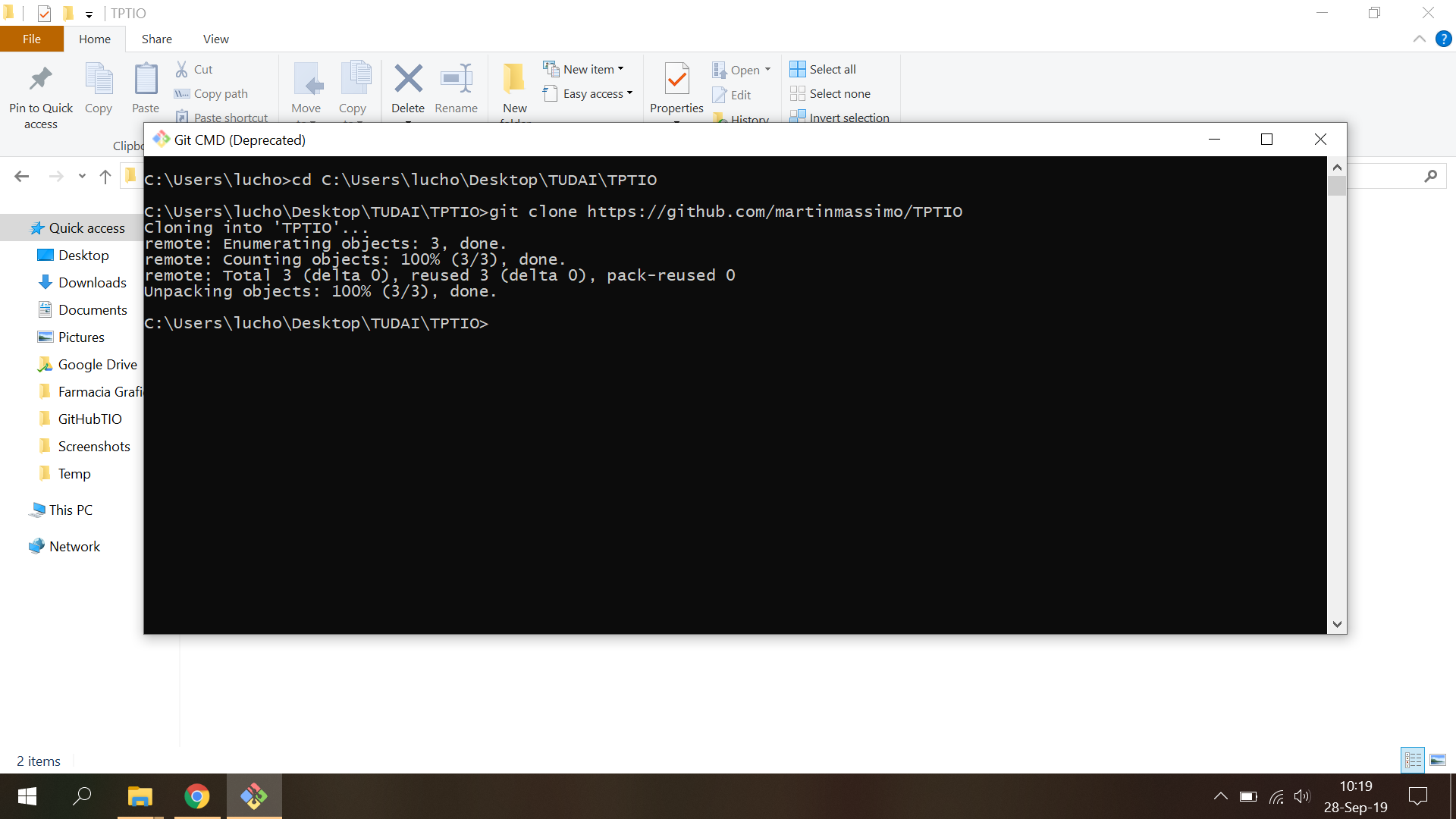
Luego, el archivo Reuniones.xlsx para llevar un cronograma de las reuniones.  
Finalmente, en el archivo word decidimos subir éste informe para tener un fácil y rápido acceso a él.

Utilizando la consola Git

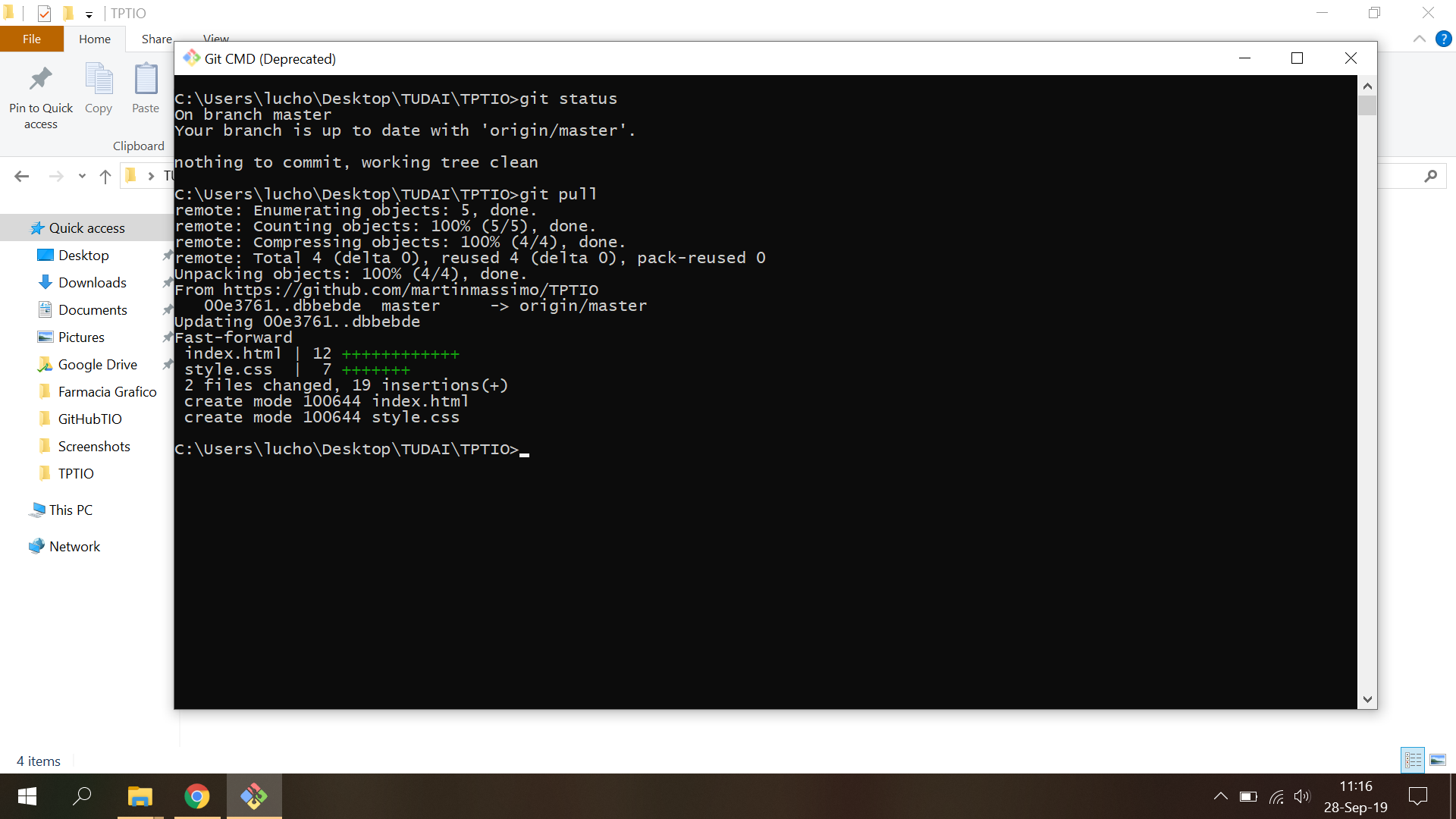
Luego de haber creado el repositorio público en GitHub y agregado a los colaboradores lo que hicimos fue usar el comando “git init” para iniciar el repositorio de forma local, luego de comprobar el estado con “git status”, añadimos el primer archivo README.MD con el comando “git add” al “stage area”. A continuación usamos el comando “git commit -m” para realizar el primer commit, después el comando “git remote add” para conectar con el repositorio remoto de GitHub. Al intentar subir los cambios con el comando “git push” nos encontramos con nuestro primer error al no definir la rama de origen para la subida de archivos, con lo cual usamos el comando “git push -u” para definir la rama de origen “Master” para subir los archivos al repositorio remoto. El resto de colaboradores utilizó el comando “git clone” y la dirección URL del repositorio en GitHub para clonar el mismo de manera sencilla.



*Utilizando git init, git status, git add, git commit -m, git remote add, git push y, finalmente, git push -u origin master.*



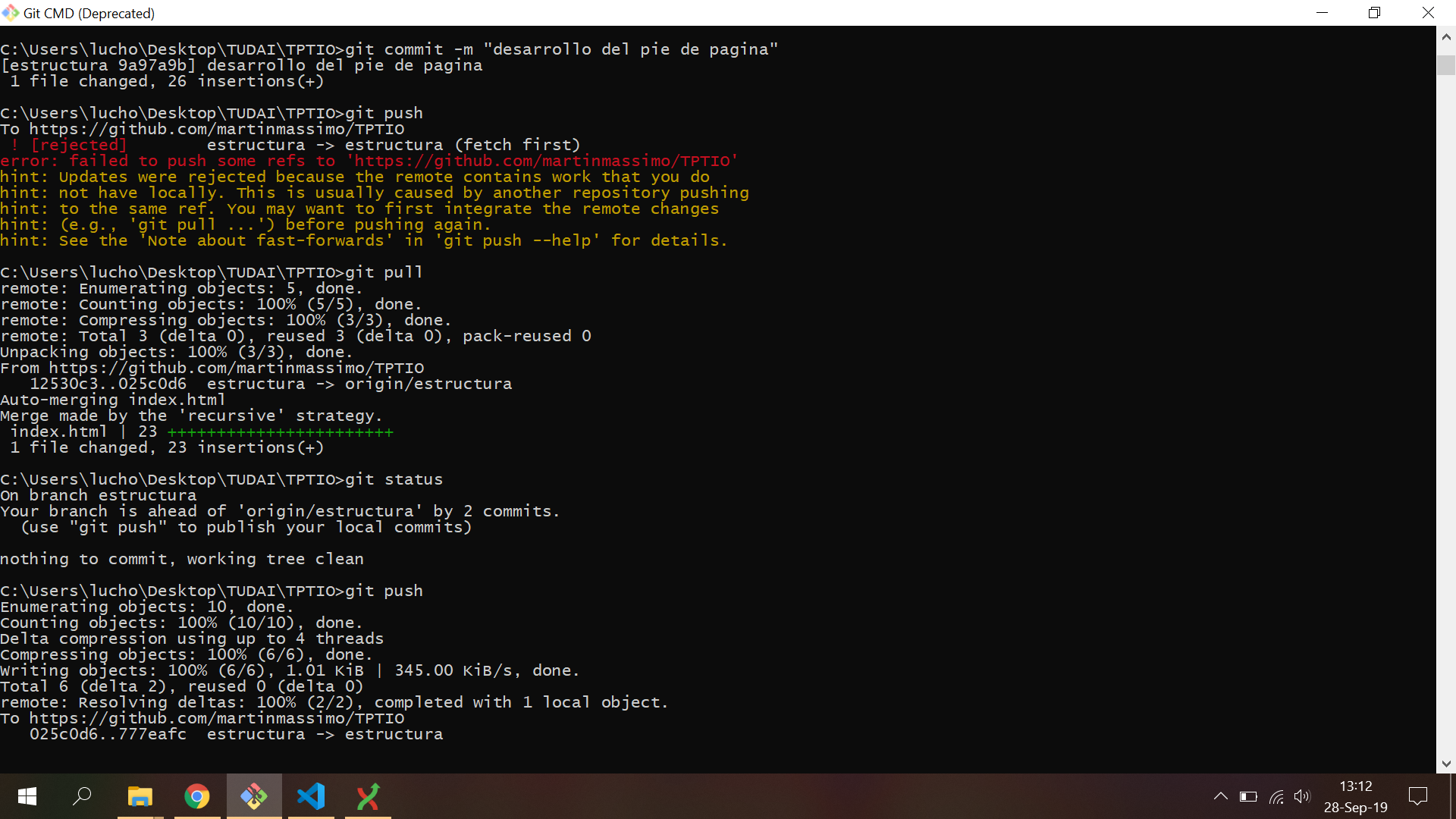
*Utilizando git clone*



*Uno de los colaboradores cargando actualizaciones a través del comando git pull.*

**

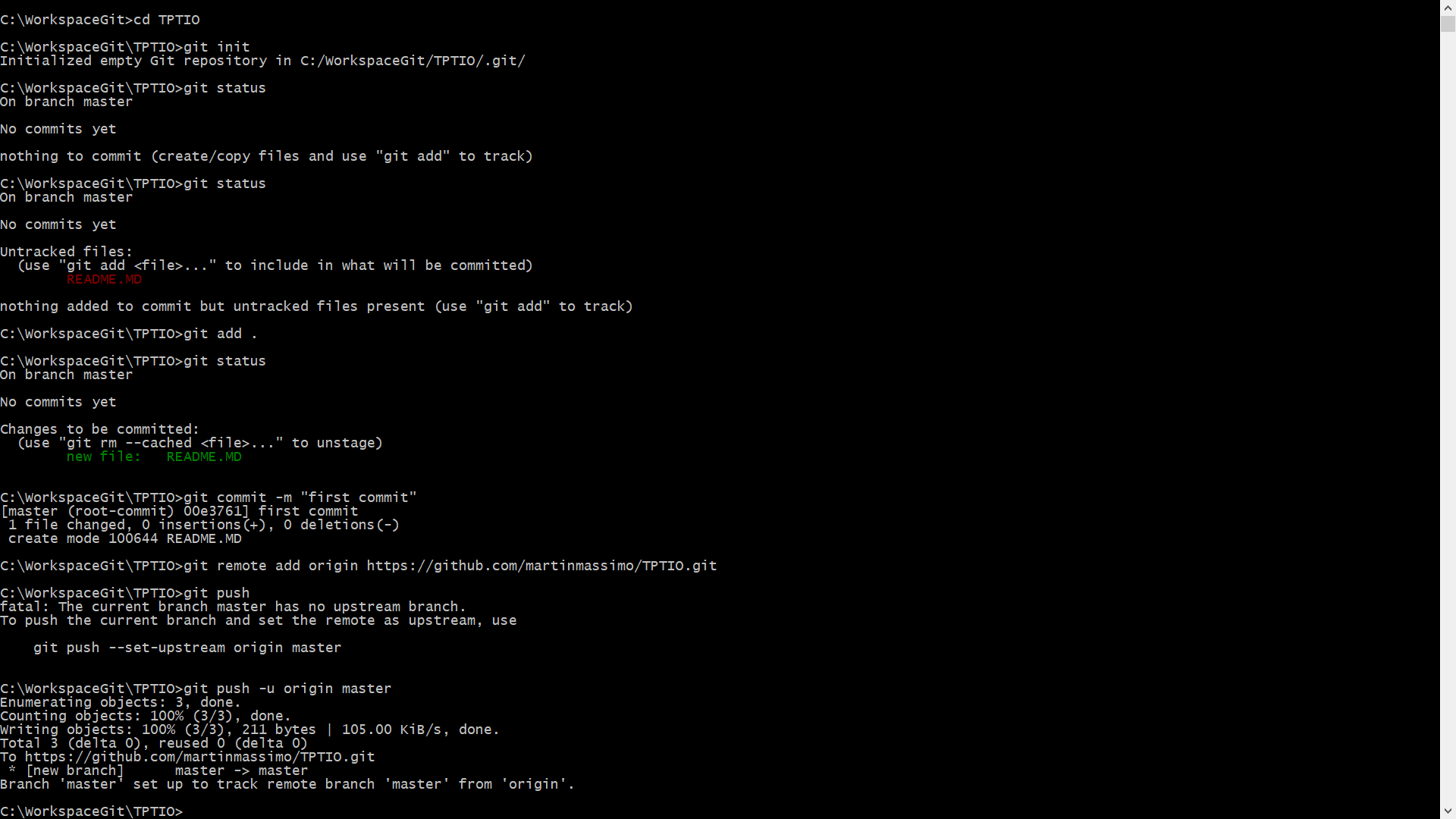
*Otro de los colaboradores utilizando el comando git pull.*

**

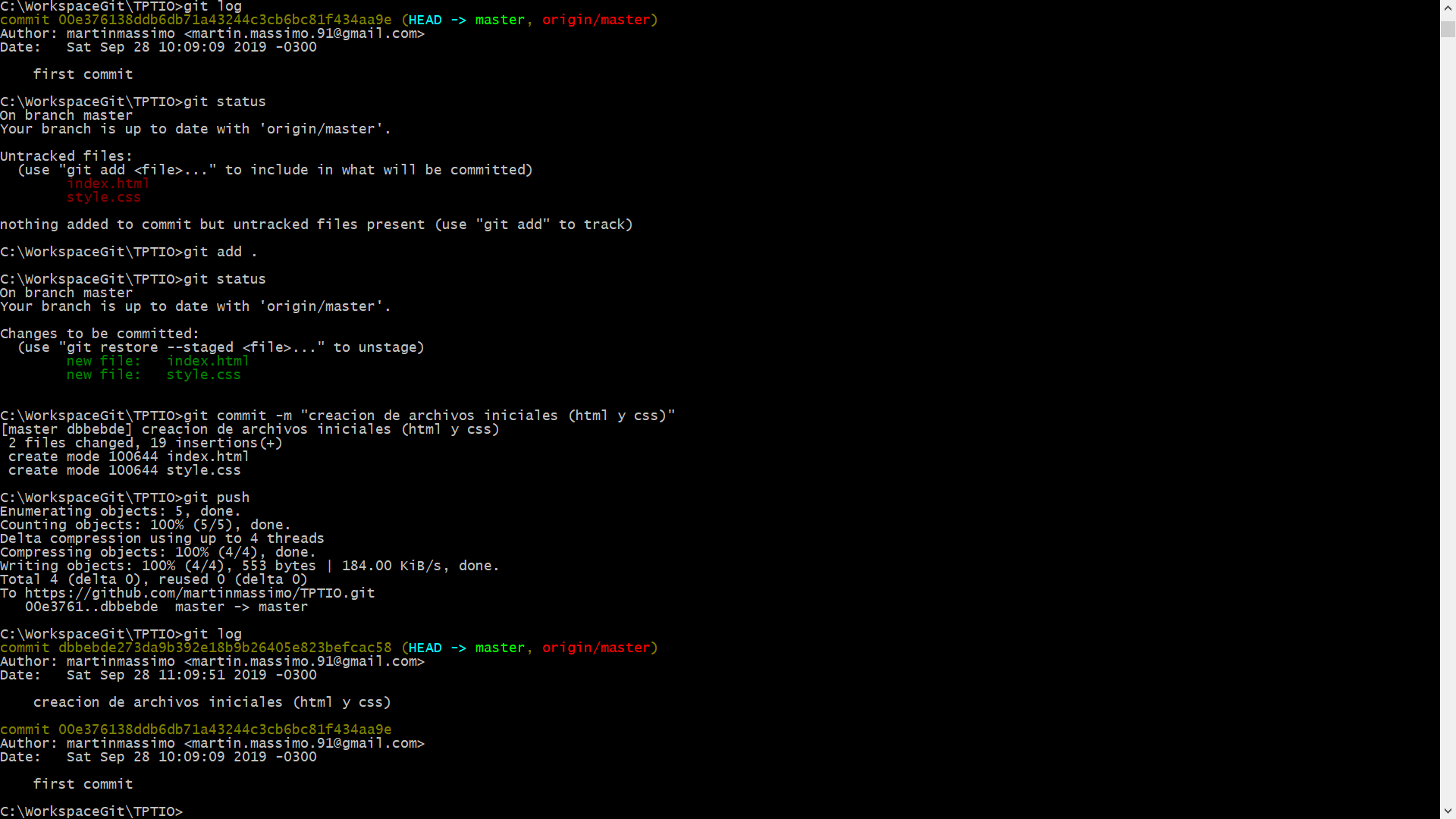
*Por último, forzamos un error con dos colaboradores que usaron “push” para subir sus cambios al repositorio sin haber actualizado antes con el comando “pull”, en esta ocasión Git resolvió con un Auto-Merge del archivo index.html.*

* Hasta aquí; hemos demostrado cómo iniciamos el repositorio con los respectivos comandos en Git, indicamos el enlace del repositorio en GitHub, nos hemos encontrado con 2 errores que luego resolvimos y finalmente hemos añadido al menos 3 comandos pull y push.

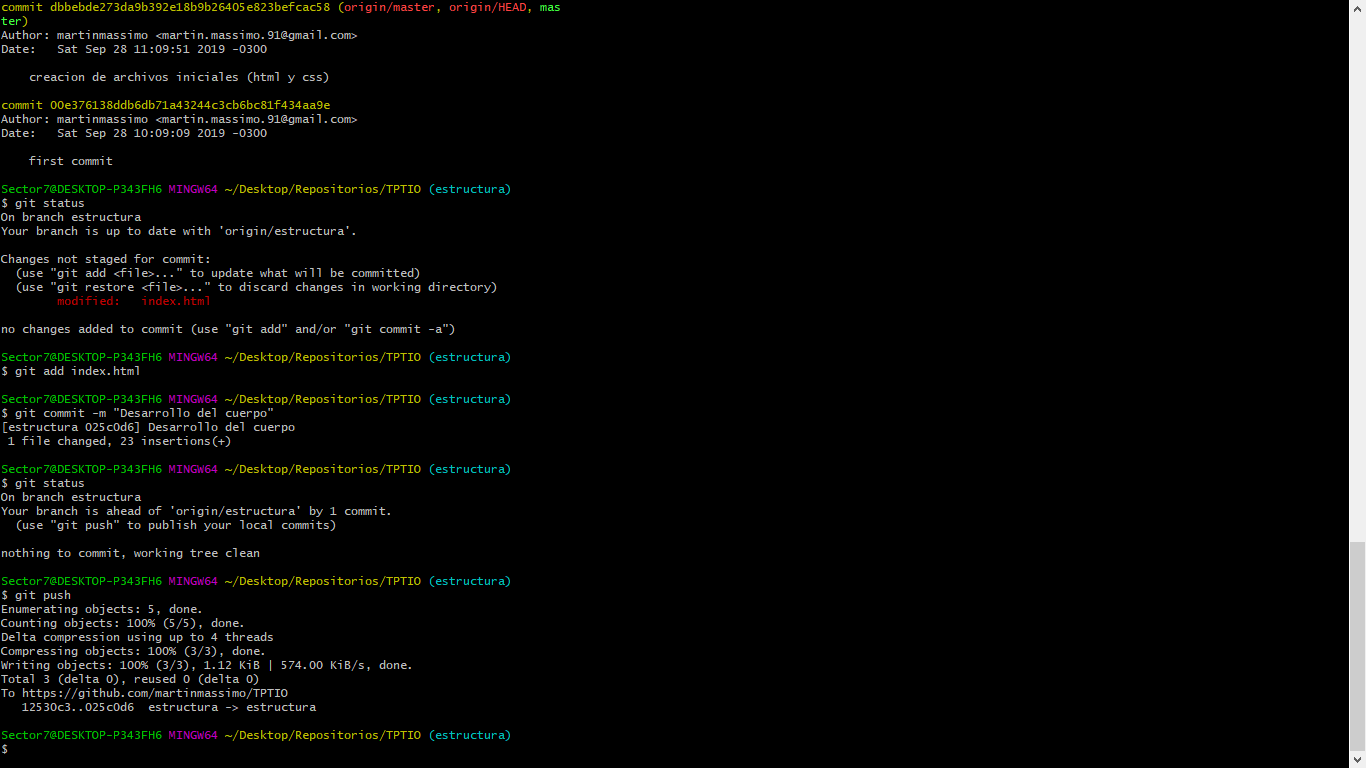
A continuación se mostraran al menos 3 secuencias “add-commit” que se registraron para la comunicación del equipo.



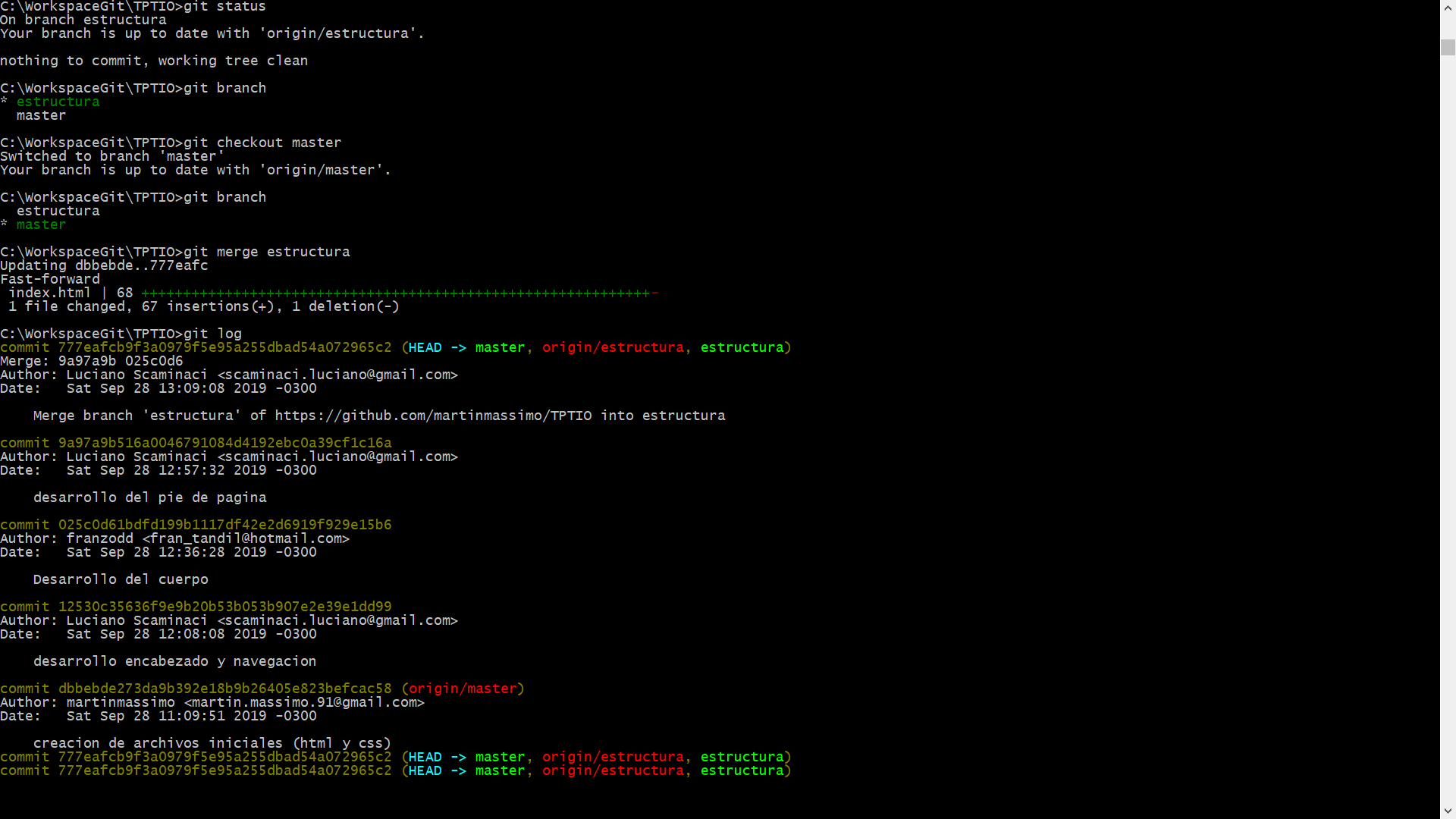
*Primer secuencia add-commit.*



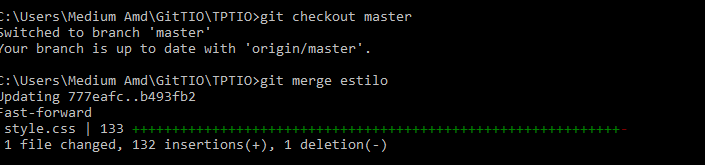
*Secuencia add-commit y push de los ficheros con extensiones .html y .css.*



Secuencia add-commit por la modificación de uno de los ficheros de la rama estructura.

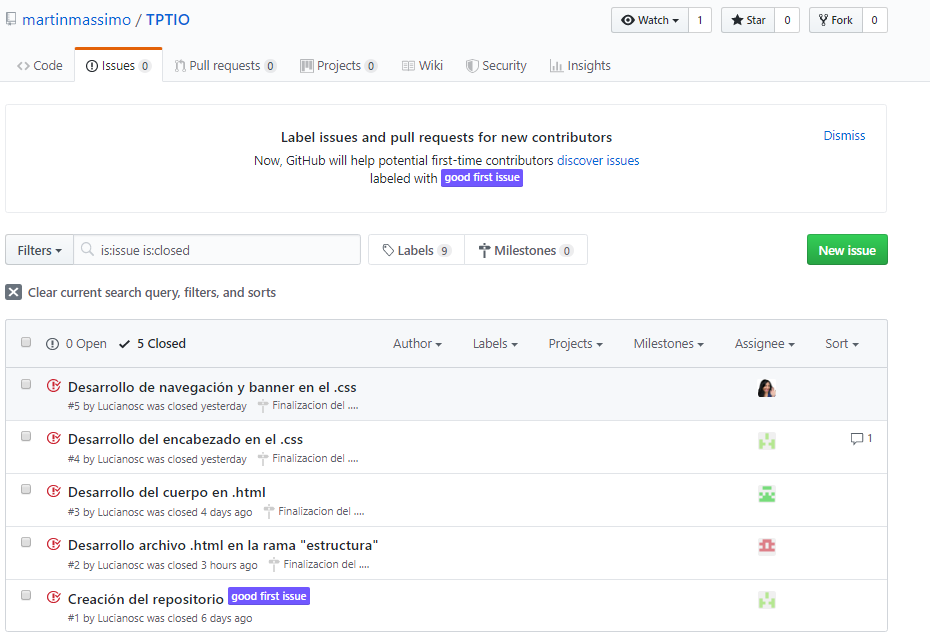


*Aquí junto con el “merge” de la rama estructura a la rama master se puede ver la secuencia de mensajes “commits” a través del comando “git log”.*

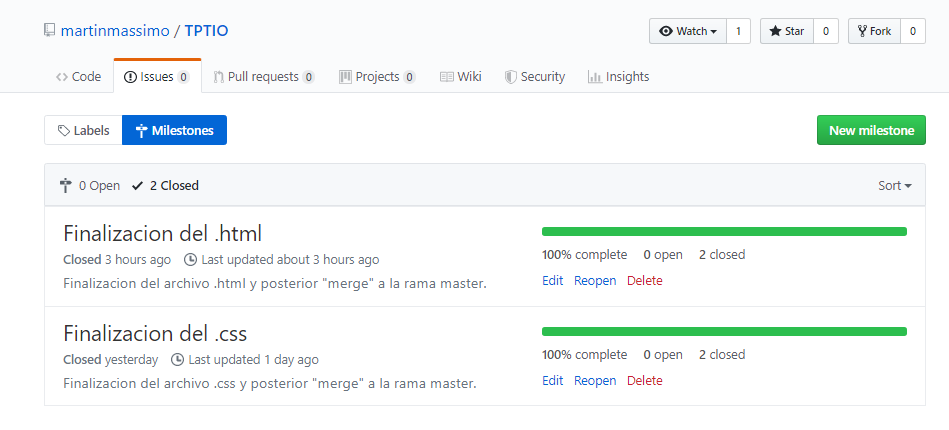


En ésta imagen se puede apreciar cómo se utiliza el comando git merge para trasladar los cambios de la rama “estilo” a la “master”.

Por último, lo que hicimos fue crear dos Milestone con distintos issues asignados a cada uno de los colaboradores según correspondiere.



*5 issue “cerrados” (cada participante tiene al menos un issue asignado).*



*Dos milestone creadas y “cerradas” con dos issues dentro cada una.*