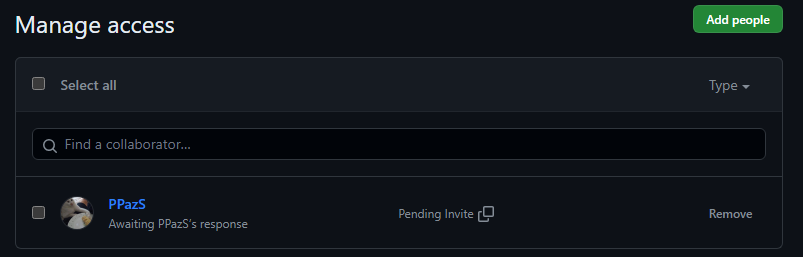
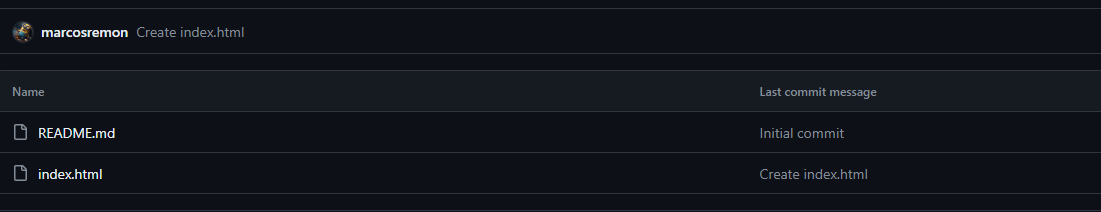
**Nombres:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Elige una pareja para realizar el siguiente ejercicio, uno haréis el papel de Ana y el otro el de Juan.**

Supongamos que dos alumnos, Ana y Juan, están colaborando en un proyecto utilizando Git y GitHub. El proyecto es un simple sitio web estático y ambos están trabajando en nuevas funcionalidades. Ana se encargará de agregar una página de contacto, mientras que Juan trabajará en la página de inicio.

1. **Crear un Repositorio en GitHub:**

* Ana crea un nuevo repositorio en GitHub llamado "ProyectoWeb".
* Dentro de este proyecto hay un archivo llamado index.html
* Ana invita a Juan como colaborador al repositorio.

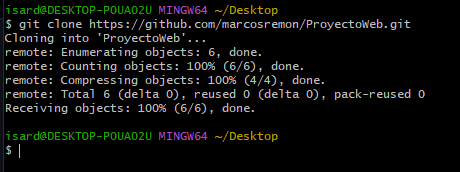


1. **Clonar el Repositorio:**

* Ana clona el repositorio en su computadora local usando el comando

***git clone <URL del repositorio>.***

* Juan también clona el repositorio en su computadora local.

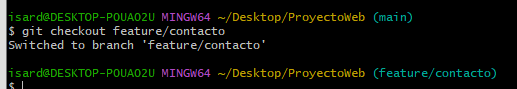


1. **Crear Ramas para Trabajar:**

* Ana crea una nueva rama llamada feature/contacto usando

***git checkout -b feature/contacto.***



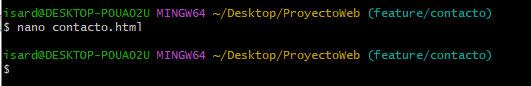


* Juan crea una nueva rama llamada feature/inicio usando

***git checkout -b feature/inicio.***

1. **Trabajar en las Funcionalidades:**

* Ana agrega la página de contacto al proyecto y hace algunos cambios en el archivo

***nano contacto.html.***

* Juan agrega contenido a la página de inicio y modifica el archivo

***nano index.html.***

1. **Commit y Push:**

* Ana hace commit de sus cambios usando

***git add .***

***git commit -m "Agregar página de contacto".***

* Luego, ella hace push a la rama remota usando

**git push origin feature/contacto.**

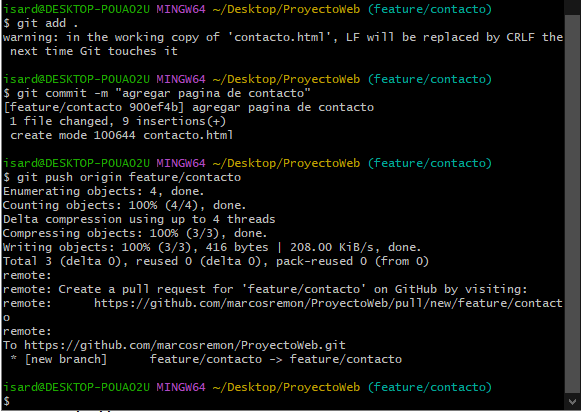
* Juan hace commit de sus cambios usando

***git add .***

***git commit -m "Agregar contenido a la página de inicio".***

* Luego, él hace push a la rama remota usando

***git push origin feature/inicio.***

******

1. **Resolución de Conflictos:**

* Mientras Ana y Juan están trabajando en sus respectivas ramas, Ana se da cuenta de que necesita hacer algunos cambios en la página de inicio. Ella hace un nuevo commit en su rama feature/contacto.
* Juan también necesita hacer algunos cambios en la página de inicio y hace un nuevo commit en su rama feature/inicio.
* Cuando Ana intenta hacer push a su rama, recibe un mensaje de error indicando que hay conflictos con los cambios de Juan en la misma línea del archivo index.html.

1. **Solución de Conflictos:**

* Ana y Juan se comunican para resolver el conflicto. Deciden cómo quieren combinar sus cambios en la página de inicio.
* Ana hace un ***git pull*** para ver los conflictos existentes, abre el archivo index.html, resuelve los conflictos manualmente, elimina las marcas de conflicto y guarda los cambios.
* Ana hace commit de los cambios resueltos usando

git add .

git commit -m "Resolver conflicto en index.html".

* Ana hace push de sus cambios resueltos a su rama remota usando

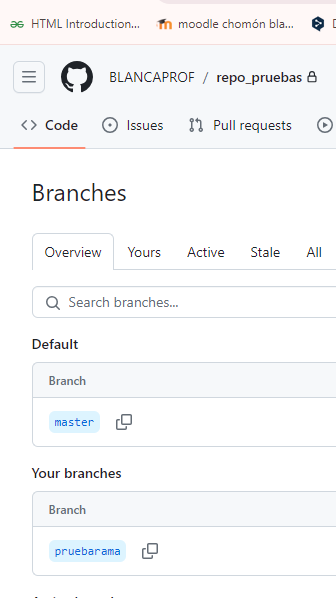
git push origin feature/contacto.

Una vez que Ana ha terminado el proceso se comunica con Juan y

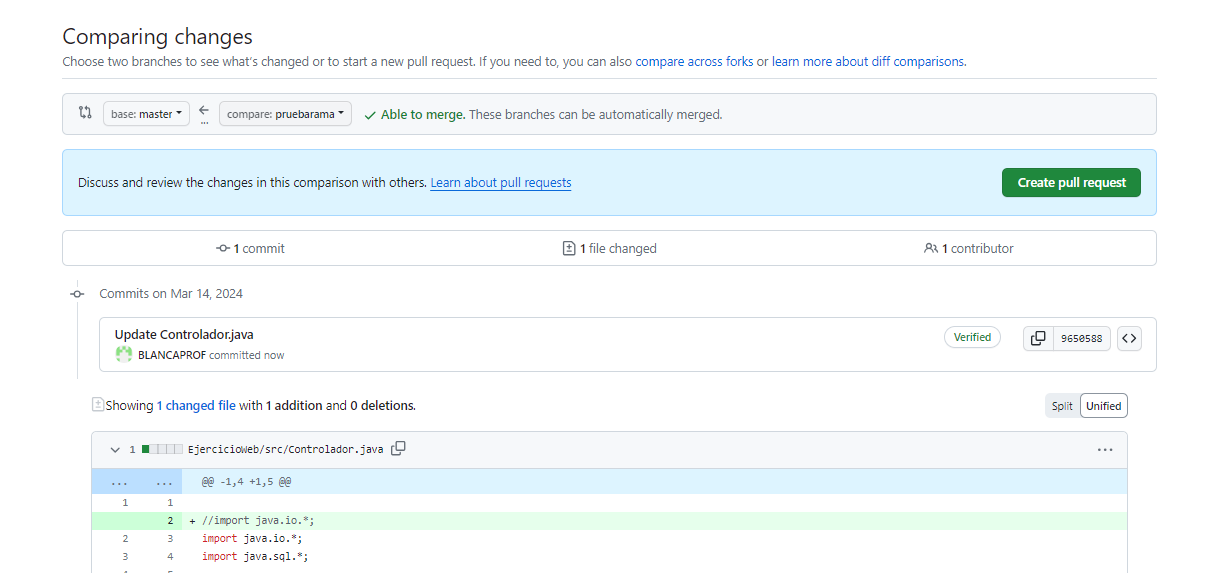
* Juan realiza el mismo procedimiento que Ana para resolver el conflicto en su rama feature/inicio.
* Si la comunicación ha ido bien al hacer el push ambos archivos coincidirán y el conflicto estará resuelto

1. **Merge de Ramas:**

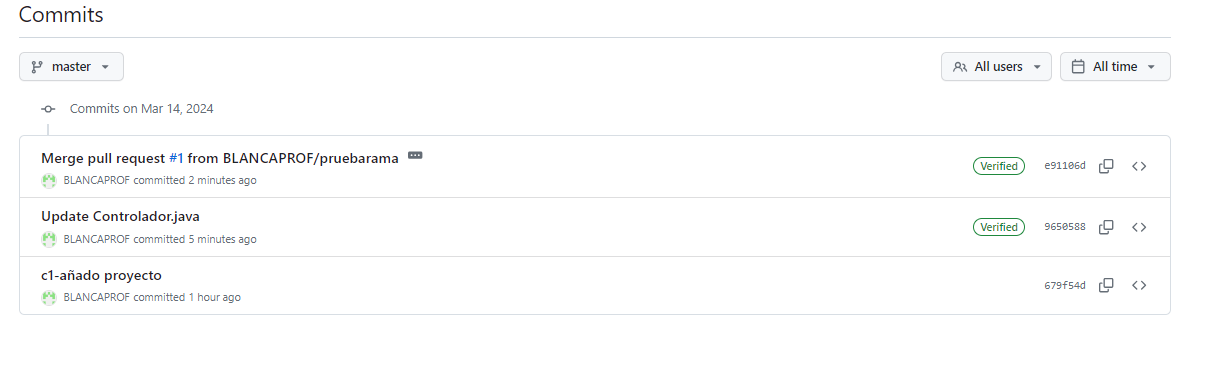
* Una vez que los conflictos se han resuelto en ambas ramas, Ana y Juan hacen ***pull request*** de sus respectivas ramas a la rama principal del repositorio en GitHub. El propietario del repositorio (puede ser Ana o Juan en este caso) revisa los cambios y, si todo está bien, realiza el merge de las ramas feature/contacto y feature/inicio en la rama principal (main o master).



1. **Abrir un Pull Request en GitHub:**
   * Ana y Juan acceden al repositorio en GitHub.
   * Ana navega a la pestaña "Pull Requests" y hace clic en el botón "New pull request".
   * Selecciona la rama **feature/contacto** como la rama base y la rama **main** (o **master**) como la rama de comparación.
   * Ana completa la descripción del pull request explicando los cambios que ha realizado en la rama feature/contacto.
   * Ana hace clic en "***Create pull request"*** para abrir el pull request.



1. **Revisión del Pull Request:**
   * El propietario del repositorio (Ana o Juan) revisa el pull request para asegurarse de que los cambios son correctos.
   * Puede agregar comentarios, sugerencias o solicitar cambios adicionales si es necesario.
   * Una vez que todo está revisado y confirmado, el propietario del repositorio hace clic en ***"Merge pull request".***
2. **Confirmación del Merge:**
   * GitHub mostrará una pantalla de confirmación del merge con la opción de eliminar la rama después del merge.
   * El propietario del repositorio confirma el merge haciendo clic en ***"Confirm merge".***



1. **Repetir el Proceso para la Otra Rama:**
   * Juan repite los pasos anteriores para abrir un pull request desde la rama **feature/inicio** hacia la rama principal (**main** o **master**).
   * El propietario del repositorio revisa y, si todo está bien, realiza el merge de esta segunda pull request.
2. **Actualización Local:**

* Ana y Juan actualizan sus repositorios locales con los cambios de la rama principal usando

***git checkout master***

***git pull origin master.***

Ana y Juan han completado exitosamente sus tareas y han resuelto cualquier conflicto que haya surgido durante el proceso. Ahora pueden continuar trabajando en nuevas funcionalidades o realizar otras tareas en el proyecto.