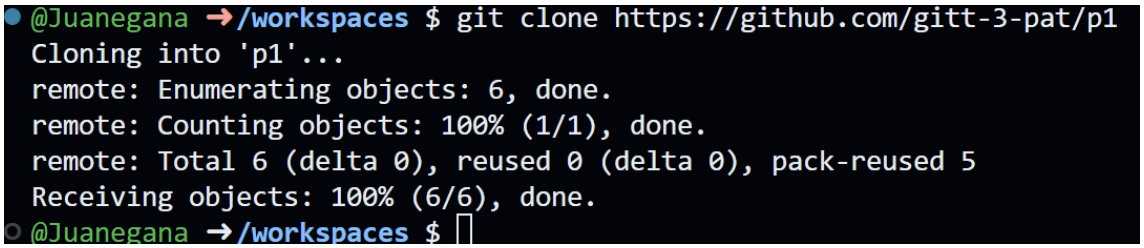
COMANDOS GIT

En esta práctica se nos pide que en el codespace de nuestro repositorio probemos a aplicar una serie de comandos en el terminal de este. Los comandos o instrucciones que nos proporciona la práctica son los siguientes:

- git clone https://github.com/gitt-3-pat/p1
- git status
- git add .
- git commit -m "TU MENSAJE"
- git push
- git checkout -b feature/1
- git checkout main

git clone https://github.com/gitt-3-pat/p1

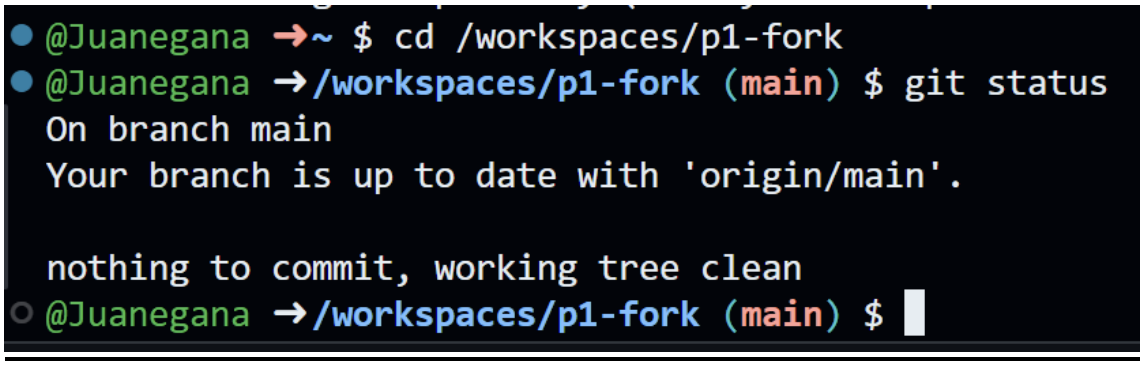
Introducimos el comando en el terminal y nos aparece lo siguiente:



Lo que hace este comando es copiar el repositorio de la dirección URL que le añadamos después. En nuestro caso https://github.com/gitt-3-pat/p1. Aparecen luego unos logs que indican que la copia ha sido exitosa.
Una vez el comando se ejecuta correctamente se creará una copia en la maquina local. Esta copia local te permite trabajar con los archivos del repositorio, hacer cambios, realizar commits, y enviar estos cambios de vuelta al repositorio remoto si tienes los permisos adecuados. Es una forma común de comenzar a trabajar en proyectos colaborativos o contribuir a proyectos de código abierto.

git status

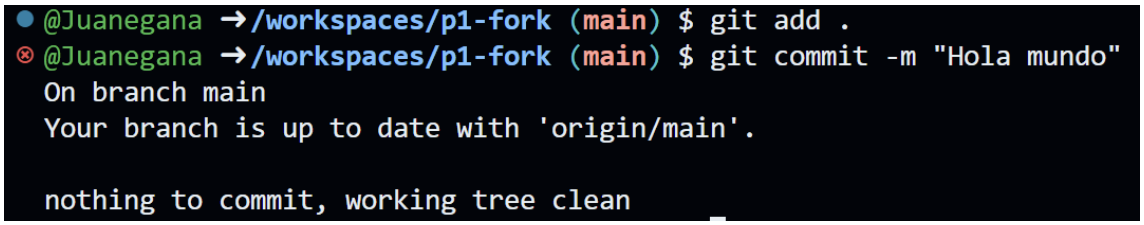
Introducimos el comando en el terminal y nos aparece lo siguiente:



El comando git status nos proporciona un resumen del estado actual del repositorio. Incluyendo los archivos modificados, los archivos en el área de preparación y los archivos no rastreados.
En nuestro caso nos esta diciendo que nos encontramos en la rama “main”, la cual es la rama predeterminada en nuestro repositorio. Además, no está diciendo que nuestra rama local está actualizada con respecto a la rama remota llamada “main” o “origin” en el repositorio remoto. Y finalmente nos indica que no hay cambios pendientes que deban ser confirmados.

git add y git commit -m “TU MENSAJE”

Introducimos el comando en el terminal y nos aparece lo siguiente:



El comando git add agrega todos los cambios en tu directorio de trabajo al área de preparación. El punto (`.`) es un comodín que indica todos los archivos y directorios en tu directorio de trabajo. Si has realizado modificaciones en varios archivos y deseas incluir todos esos cambios en el próximo commit, puedes utilizar este comando para agregarlos todos de una vez al área de preparación.

Por otro lado, el comando git commit -m "hola mundo realiza un commit con los cambios que previamente agregaste al área de preparación. El `-m` se utiliza para especificar un mensaje de commit en la misma línea de comando. En este caso, el mensaje del commit es "hola mundo".

En nuestro caso el mensaje que nos devuelve indica que la rama en la que te encuentras está actualizada con respecto a la rama remota y que no había cambios nuevos para

incluir en un nuevo commit. Esto significa que no se realizó ningún commit porque no había cambios adicionales desde el último commit realizado en la rama. Esto significa que nuestro "working tree" (árbol de trabajo) está limpio, lo que significa que no hay modificaciones sin guardar en nuestros archivos.

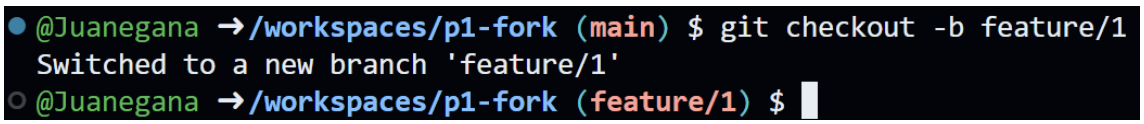
git push

Introducimos el comando en el terminal y nos aparece lo siguiente:

El comando git push sirve para enviar tus commits locales al repositorio remoto, manteniendo así sincronizados los cambios entre tu repositorio local y el remoto.
En nuestro caso, el mensaje "everything up-to-date" indica que no se necesitaba enviar ningún cambio adicional en ese momento porque tu repositorio local ya estaba al día con el remoto.

git chechkout -b feature/l

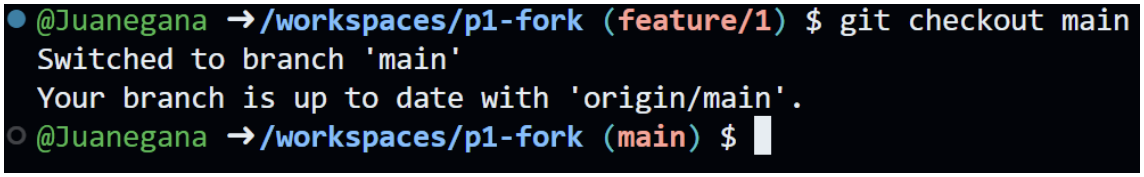
Introducimos el comando en el terminal y nos aparece lo siguiente:



El comando git checkout -b <nombre-de-la-rama> nos permite crear una nueva rama y moverte instantáneamente a ella, lo que facilita la creación de nuevas líneas de desarrollo sin afectar a la rama principal de nuestro proyecto.
En nuestro caso la rama se llama “feature/1”, y el mensaje que sale después nos indica que todo se ha ejecutado correctamente.

git checkout main

Introducimos el comando en el terminal y nos aparece lo siguiente:



Estamos haciendo lo mismo que en el comando previo, solo que en vez de crear una rama nueva nos estamos moviendo de vuelta a la rama ‘main’ además nos indica que la rama ‘main’ está actualizada con respecto con el remoto como hemos explicado antes.