How to git

Pasos a seguir para interactuar con el web app de Github..

Todos los commando fueron inyectados atraves del Terminal de Visual Studio Code.

Para iniciar el proceso de cambio crearemos un fork. Este fork nos permitira hacer cambios aislados del baseline, la manera de hacerlo es la siguiente.

* En la pagina del repositorio (<https://github.com/Vitols-code/CalTracker>) presionamos el boton de “fork” ilustrada en la siguiente imagen.

A screenshot of a computer

Description automatically generated

Nos aseguramos de que nuestro usuario aparezca en la seccion de “owner”. Ademas verificamos que el nombre del repositorio sea del repositorio correspondiente. En Nuestro caso nuestro repositorio se llama “CalTracker”

* Una vez verificada la informacion dejamos el checkbox de “copy the main branch only” y presionamos el boton de de “Create fork” ilustrado en la siguiente foto.

A screenshot of a computer

Description automatically generated

Verificamos que este nuevo “Forked repository” haya sido creado desde Nuestro baseline “CalTracker”.

* Presionamos el boton de code verde para expandir el sub-menu donde veremos el URL del “Forked repository”.
* Copiamos el URL del repository como es ilustrado en la siguiente imagen.

A screenshot of a computer

Description automatically generated

**!!!!!NOTE!!!!!**

**El URL DE EL “FORKED REPOSITORY” TENDRA EL NOMBRE DEL USUARIO QUE CREO DICHO REPOSITORIO. NO ULITIZAR EL URL MOSTRADO EN LA IMAGEN PUES ESTE ES SOLO UN EJEMPLO.**

1. **Clone GitHub repository** - Clonar el repositorio en la maquina local para editar el codigo a su conveniencia.
   * -Create destination folder - Esto sera un folder vacio donde clonaremos el git repository
   * -git init - initialize empty git repository (Dentro del Destination folder que creamos, inicializamos el git repository como empty con este commando para luego clonar uno existente)
   * -git clone <repository URL> (i.e. https://github.com/mark3bx/CalTracker.git)
2. Crear un branch en GitHub - Separar los cambios hechos por el usuario de el baseline en GitHub. Para luego hacer un push and pull de este branch hacia el main.
   * cd <nombre del directorio donde esta el repositorio> (Para poder crear un branch, tenemos que estar dentro del directorio donde se creo el clone del repositorio. i.e (C:\Users\bob\OneDrive\Desktop\Goon bob\CalTracker-main) (Los contenidos de este Directorio deberian ser los mismos que el “Forked repository” creado en el Github Website.)
   * -git branch <nombre del branch> (El nombre del branch deberia ser indicativo de los cambios a hacer)
   * -git checkout <nombre del branch creado> (Se hace un checkout al branch creado para comenzar a hacer cambios al software)

**!!!NOTE!!!**

**LUEGO DE COMPLETAR LOS CAMBIOS DESEADOS POR EL CREADOR DEL BRANCH, PARA SUBIRLOS A GITHUB HACEMOS LOS PASOS 3 Y 4**

1. **Commit to repository** - Grabar el codigo modificado en el local git repository que clonamos al principio.
   * git add . (Hacemos stage a los cambios para prepararlos para un commit al repositorio)
   * git commit -m "<Comentario del commit>" (El comentario debe estar entre comillas para que el commando de git commit lo acepte. El comentario deberia ser corto y descriptivo de los cambios hechos en el branch.
2. **Push to repository** - Upload local Commits de tu local repository al remote repository (https://github.com/Vitols-code/CalTracker.git)

* git push <repository URL> <branch name> (El repository URL deberia ser reemplazado con el URL obtenido en el paso de git clone. El branch name con el nombre del branch en el que hicimos cambios, Tambien definido en el paso de git branch)

1. **Pull request on GitHub** - En el momento que se hace el push del branch modificado queda registrado en el repositorio en GitHub y nos permite hacer un pull request. El pull request se hace para "merge" el branch modificado al main branch/baseline.

* + Para hacer esto nos vamos a la pagina web de git hub fork y sometemos el commit a un pull request.
  + Este pull request va a pedirle al autor del repositorio, que le haga un review (Para verificar que todo compile y este en orden) y luego merge al main branch/baseline