

# GitHub

Para ver estado de documentos:

```
sudo git status
```

Para subir cambios:

```
sudo git add *  
sudo git commit -m "mensaje..."  
sudo git push origin master
```

Para crear nuevo branch:

```
sudo git checkout -b [nombre]
```

Para cambiar de branch:

```
sudo git checkout [nombre]
```

Para ver todos los branches:

```
sudo git branch
```

Para que mi branch local se refleje en github:

```
#entrar al branch develop  
sudo git add *  
sudo git commit -m "mensaje..."  
sudo git push origin develop
```

Para unir la rama a master:

```
#cambiar a master  
sudo git checkout master  
#unir la rama develop a master  
sudo git merge develop  
sudo git push origin master
```

Para crear rama personal:

```
#entrar al branch develop  
sudo git checkout -b jp-pino/[nombre del avance]
```

Cuando termino mi avance:

```
#opción 1 - Integración continua (modelo Driessen)  
sudo git checkout develop  
sudo git pull  
sudo git merge jp-pino/[nombre del avance]
```

```
sudo git push -u
#si hay conflictos al subir mi avance:
#abrir el archivo y resolver el conflicto
sudo git commit -m "mensaje..."
sudo git push -u
```

```
#opción 2
#dentro de mi rama
sudo git add *
sudo git push -u
#desde GitHub crear un Pull Request
```

Manejo y asignación de tareas:

- Opción 1: Las tareas deberían durar el tiempo que pueda trabajar. Si no partir tarea.
- Opción 2: Feature toggles - si no terminé mi código meto todo el código en desarrollo dentro de un if con una variable booleana apagada.

Diariamente hay que hacer un **pull** para actualizar develop y empezar una nueva rama para trabajar.

El que haga merge primero no tiene problemas.