

Formato de commits

Format file message commit

```
type: subject  
  
body  
  
footer
```

[Ver mas](#)

Procesos en diferentes escenarios

Inicio de sprint

Crear nueva rama con el formato **sprint_xx** (Donde **xx** es el número de sprint).

```
git checkout -b sprint_XX
```

Commits dentro de rama sprint

Cuando se termine el trabajo del día y debe guardar los cambios se tiene que seguir el formato previamente definido de commit.

```
vim commit-message.txt  
> type: subject  
> body  
> footer  
  
vim .gitignore  
> commit-message.txt  
  
git add .  
git commit --file commit-message.txt  
git push origin sprint_XX
```

Pase a QA

El pase sera un pull en la maquina de QA. Dev y QA funcionan los 2 en una sola rama siendo QA un commit en el tiempo de la vida del GIT.

```
git pull origin sprint_XX  
git checkout sprint_XX
```

Solucionar Issues en QA

Se dara el caso que la versión de tu versión en QA genere algun Issues y como en paralelo habras avazado puntos de otra tarjeta se regresara al commit que utiliza QA y creara una rama "fix_XXXX".

```
git checkout bcdbf2b
git branch -b fix_XXXX
git checkout fix_XXXX
```

cuando es solucionado, en la maquina QA :

```
git pull origin fix_XXXX
git checkout fix_XXXX
```

Así las pruebas se haran tomando la rama fix_XXXX hasta que se valide que no hay Issues. Cuando este validada se unira a la rama sprint_XX cuando la rama sprint_XX tenga otro commit para pasar a QA.

Maquina DEV

```
git checkout sprint_XX
git merge fix_XXXX
## Solucionar si hay conflictos

vim commit-message.txt
> merge: subject
> body
> footer

git add .
git commit --file commit-message.txt
git push origin sprint_XX
```

Maquina QA

```
git pull origin fix_XXXX
git pull origin sprint_XX
git checkout sprint_XX
```

Pase a producción

Cuando se realizen todas las pruebas y al final del sprint se procedera a enviar todo los avances a la rama Master(producción). Maquina DEV

```
git checkout master
git merge sprint_xx
#Solucionar si hay conflictos

vim commit-message.txt
> merge: subject
> body
> footer

git add .
git commit --file commit-message.txt
git push origin master
```

Maquina Producción

```
git pull origin master
```

Error en producción

Si fuera a darse que hay un error cuando este en producción, se procede a crear una rama `fix_XXX` a partir de la `master`.

```
git checkout -b fix_XXXX
```

Sprint terminado

Cuando se pasa a producción o se soluciona un `fix` desde la rama `master`.

```
git tag -m "....." -a vX.Y.Z #v1.0.0 bcdbf2b
#"....." Ingrese un mensaje puntual.
# X : Revisión importante (nueva interfaz de usuario, muchas características nuevas,
cambio conceptual, etc.)
# Y : Revisión menor (tal vez un cambio en el cuadro de búsqueda, 1 característica
añadida, colección de correcciones de errores)
# Z : Corrección de errores
```

[Ver mas](#)

Note:

- Informar al team cuando se hará un commits.
- Merges con conflictos serán solucionados por el que hizo el merge.
- Agregar tags cuando se tenga una versión estable (Posiblemente después de un sprint) #####