

TESTING

MODO AUTOMÁTICO



TENGO LOS TESTS



AVISOS

¿  VS  ?



MANUALMENTE

PROBLEMA: PODEMOS OLVIDARNOS



PROCESO DE DESARROLLO

PUNTOS CLAVES



CONTROL DE VERSIONES

COMMIT



"ARREGLO FALLO ULTIMO COMMIT"

VAMOS A INTENTAR EVITAR ESTE TIPO DE MENSAJES



CONTROL DE VERSIONES

PUSH



"HE MEZCLADO TU ÚLTIMO CÓDIGO Y PETA"

EVITEMOS MALOS TRAGOS A COMPAÑEROS



INTEGRACIÓN CONTINUA TRABAJO DE INTEGRACIÓN



***"LO ÚLTIMO QUE HABEIS SUBIDO NO
FUNCIONA"***

**EVITEMOS QUE EL CLIENTE NOS TENGA QUE AVISAR DE
ALGO QUE YA DEBERÍAMOS SABER.**





REFLEXIONEMOS

Ahora, antes de cada commit o push

¿realizamos todas las tareas que sabemos que tendríamos que hacer para comprobar que la aplicación funciona?



REPETITIVO

=

ABURRIDO



REPETITIVO

=

DESIDIA



REPETITIVO

=

FALLOS



TRASLADAR RESPONSABILIDAD



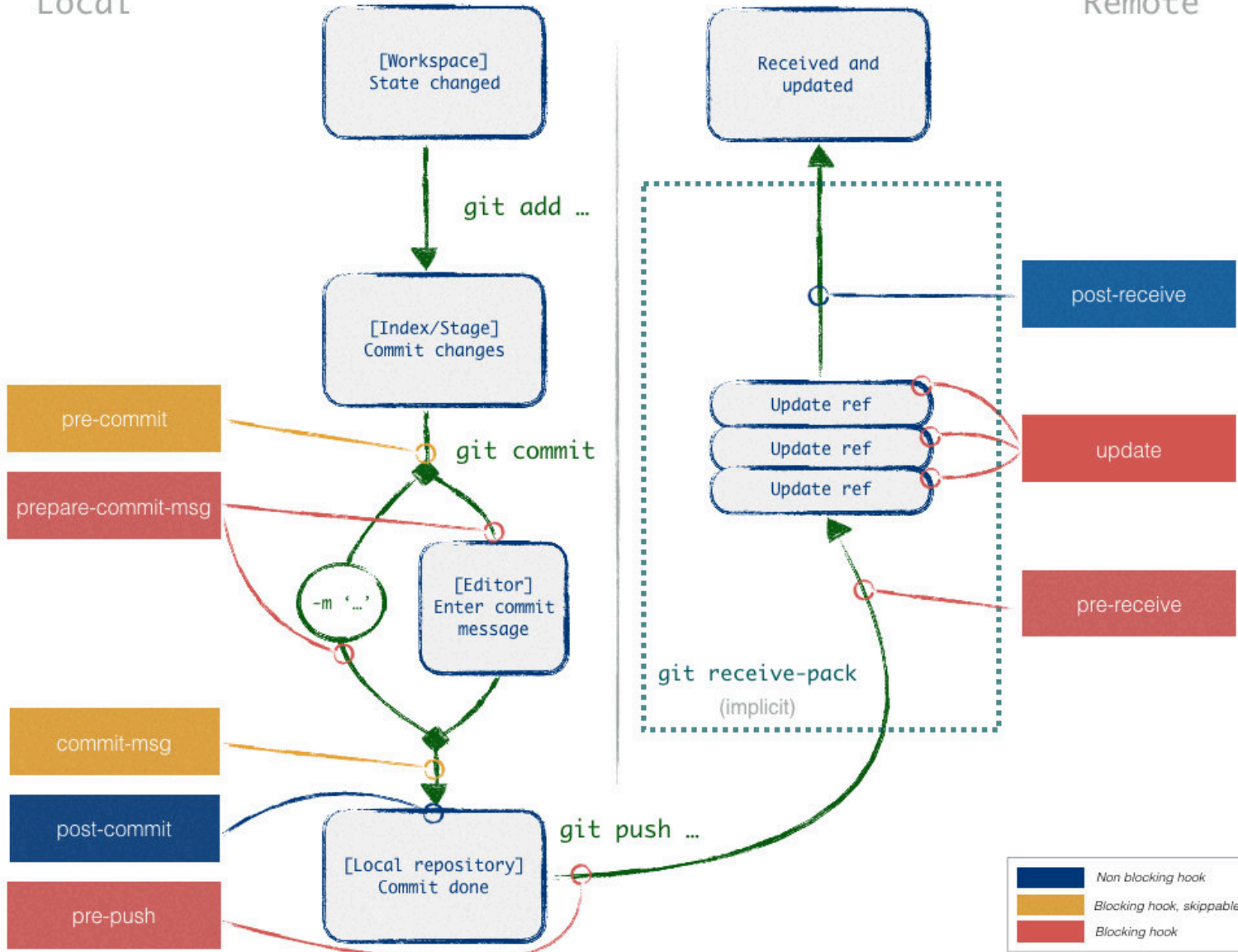
GIT HOOKS

**GIT NOS OFRECE PODER LANZAR COMANDOS ANTES Y
DESPUÉS DE MUCHOS EVENTOS**



Local

Remote



CARPETA .git

```
cd proyectoConRepositorioGit  
ls -lah .git/hooks
```



```
$ ls -la .git/hooks |tr -s ' ' |cut -d' ' -f9-
```

```
.  
..  
applypatch-msg.sample  
commit-msg.sample  
fsmonitor-watchman.sample  
post-update.sample  
pre-applypatch.sample  
pre-commit.sample  
pre-merge-commit.sample  
pre-push.sample  
pre-rebase.sample  
pre-receive.sample  
prepare-commit-msg.sample
```



PRE-COMMIT

```
# Comando de test en archivo pre-commit  
echo "mvn test" > .git/hooks/pre-commit  
# Darle permisos de ejecución  
chmod +x .git/hooks/pre-commit
```



PRE-PUSH

```
# Comando de test en archivo pre-commit  
echo "mvn test" > .git/hooks/pre-push  
# Darle permisos de ejecución  
chmod +x .git/hooks/pre-push
```



GITLAB

`.gitlab-ci.yml`




```
test-job1:  
  stage: test  
  script:  
    - mvn test  
  
build-job:  
  stage: build  
  script:  
    - mvn package  
  
deploy-prod:  
  stage: deploy  
  script:  
    - scp project.war git@servidorproduccion:/var/autodeploy/
```



For `desarrollo`

latest 6 jobs 3 minutes 43 seconds, queued for 1 seconds

Pipeline Needs Jobs 6 Tests 305 Code Quality

Group jobs by Stage Job dependencies

test	build	deploy
✔ apptests	✔ build	✔ deploy
✔ code_quality		
✔ code_quality_html		
✔ secret_detection		













For **desarrollo**

7 jobs 3 minutes 29 seconds, queued for 1 seconds

Pipeline Needs Jobs 7 Failed Jobs 1 Tests 281 Code Quality

Group jobs by **Stage** Job dependencies

test	build	deploy
 apptests 	 build	 deploy
 code_quality 		
 code_quality_html 		
 secret_detection 		



AHORRO

**NO PENSEIS EN EL TRABAJO QUE OS LLEVA
PENSAD EN EL TRABAJO QUE OS AHORRA**

