simplifying git

Pepe Fabra Valverde

modelo mental

modelo mental

01 git
Lorem ipsum dolor sit amet

02 enfoque
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 git

Sistema Control de Versiones

01 git

Sistema Control de Versiones

¿Por qué?

01 git

Sistema Control de Versiones

¿Por qué?

Soluciones incompletas o demasiado complejas

¿Para qué?

01 git

Sistema Control de Versiones

¿Por qué?
Soluciones incompletas o demasiado complejas

¿Para qué?
Poder versionar y compartir el kernel de Linux

02 enfoque

Desde lo conceptual

Árboles, semillas y ramas

Desde lo técnico

Deltas, LinkedLists y hashes

config

config

01 tipos
Lorem ipsum dolor sit amet

02 prioridades
Lorem ipsum dolor sit amet

03 sensitive
Lorem ipsum dolor sit amet

04 rerere
Lorem ipsum dolor sit amet

01 tipos

- **system**
  - máquina
- **global**
  - usuario
- **local**
  - repositorio

02 prioridades

1. local
2. Si no existe, system
3. Si no existe, global

03 case sensitive

git es case sensitive, tiene en cuenta mayúsculas y minúsculas

En Windows, activa:

git config [scope] core.ignorecase false

04 rerere

Una resolución para siempre

git config [scope] rerere.enabled true

working, estados y remotes

working, estados y remotes

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 remotes
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

01 working[name]

02 estados

03 remotes

commits y puntero

commits y puntero

01 commit
Lorem ipsum dolor sit amet

02 puntero
Lorem ipsum dolor sit amet

03 movimiento
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 commit

El átomo de git

02 puntero (HEAD)

03 movimiento

ramas y movimientos

ramas y movimientos

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 ramas

01,5 ramas huérfanas

02 movimiento

02 movimiento

Práctica: https://learngitbranching.js.org/

merge y rebase

merge y rebase

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 merge

02 rebase

03 rebase interactivo

cherry-pick

cherry-pick

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 cherry-pick

02 rev-parse

resolución de conflictos

resolución de conflictos

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 resolución de conflictos

pull y push

pull y push

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

00 fetch

01 pull

02 push

03 pull-request

tags

tags

01 tag
Lorem ipsum dolor sit amet

02 desde otro
Lorem ipsum dolor sit amet

03 push
Lorem ipsum dolor sit amet

04 pull
Lorem ipsum dolor sit amet

01 tag

00 tag desde otro tag

03 push tags

git push --tags [remote]

04 pull tags

git fetch --tags [remote]

git hooks

git hooks

01 lifecycle
Lorem ipsum dolor sit amet

02 hook
Lorem ipsum dolor sit amet

03 --no-verify
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

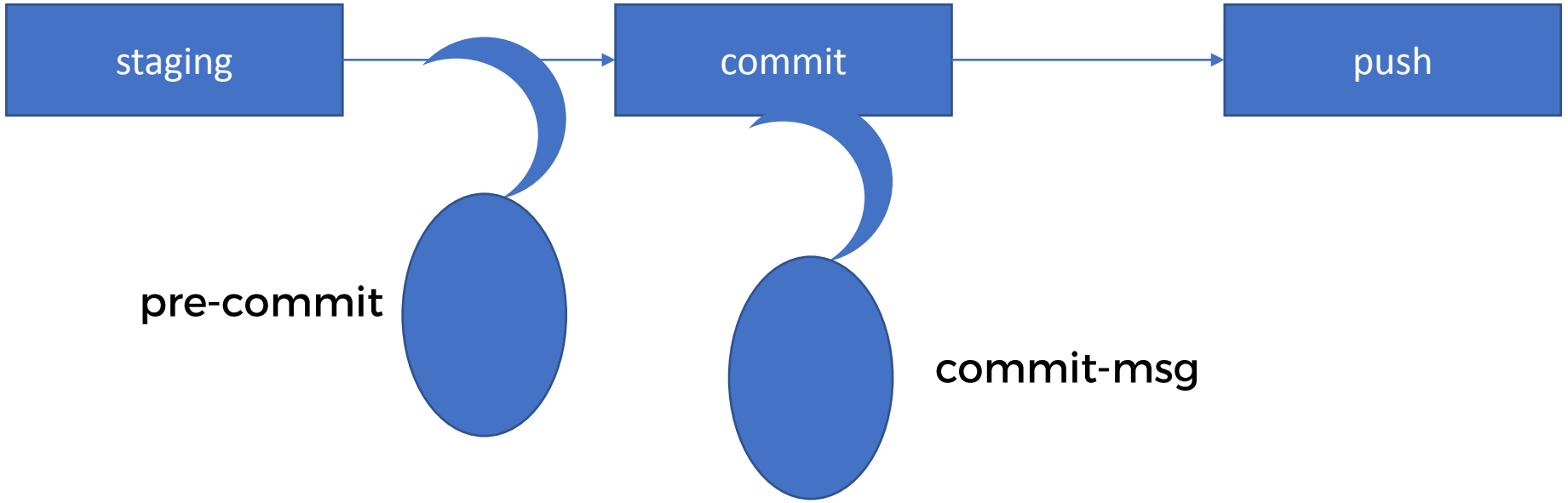
01 lifecycle

https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks

pre-commit, prepare-commit-msg, commit-msg, post-commit

01 lifecycle

02 hook



03 --no-verify

trunk based development

trunk based development

01 integración
Lorem ipsum dolor sit amet

02 feature flag
Lorem ipsum dolor sit amet

03 branches
Lorem ipsum dolor sit amet

04 despliegues
Lorem ipsum dolor sit amet

00 integración continua

00 feature flags

00 long-lived branches

00 despliegues a prod

Despliegue a prod != reléase de una feature

Es Producto quien tiene el control de sacar una feature ya desarollada

trucos

01 eliminar virtualmente

A veces hemos comitteado un archivo que no hacía falta, pero no lo queremos borrar en local.

git rm --cached [filename]

agradecimientos

slides

preguntas

título de sección

título de sección

01 apartado
Lorem ipsum dolor sit amet

02 apartado
Lorem ipsum dolor sit amet

03 apartado
Lorem ipsum dolor sit amet

04 apartado
Lorem ipsum dolor sit amet

01 apartado