

GIT

Sistema de control de versiones

Contenido

1. Contexto
2. Definición
3. Casos de uso
4. Conceptos de funcionamiento
5. Instalación
6. Uso local
7. Conflictos
8. Proveedores remotos
9. Uso remoto
10. Buenas prácticas
11. Colaboración
12. Preguntas, licencia y agradecimientos

Contexto

GIT no es GitHub

Contexto

GIT no es una mera copia de seguridad

Contexto

GIT no es una moda

Definición

- DVCS, o sistema de control de versiones distribuido.
- Diseñado para el desarrollo del kernel de Linux.
- Desde abril del 2005. Su última versión es de marzo del 2015.
- Código abierto bajo licencia GNU 2.

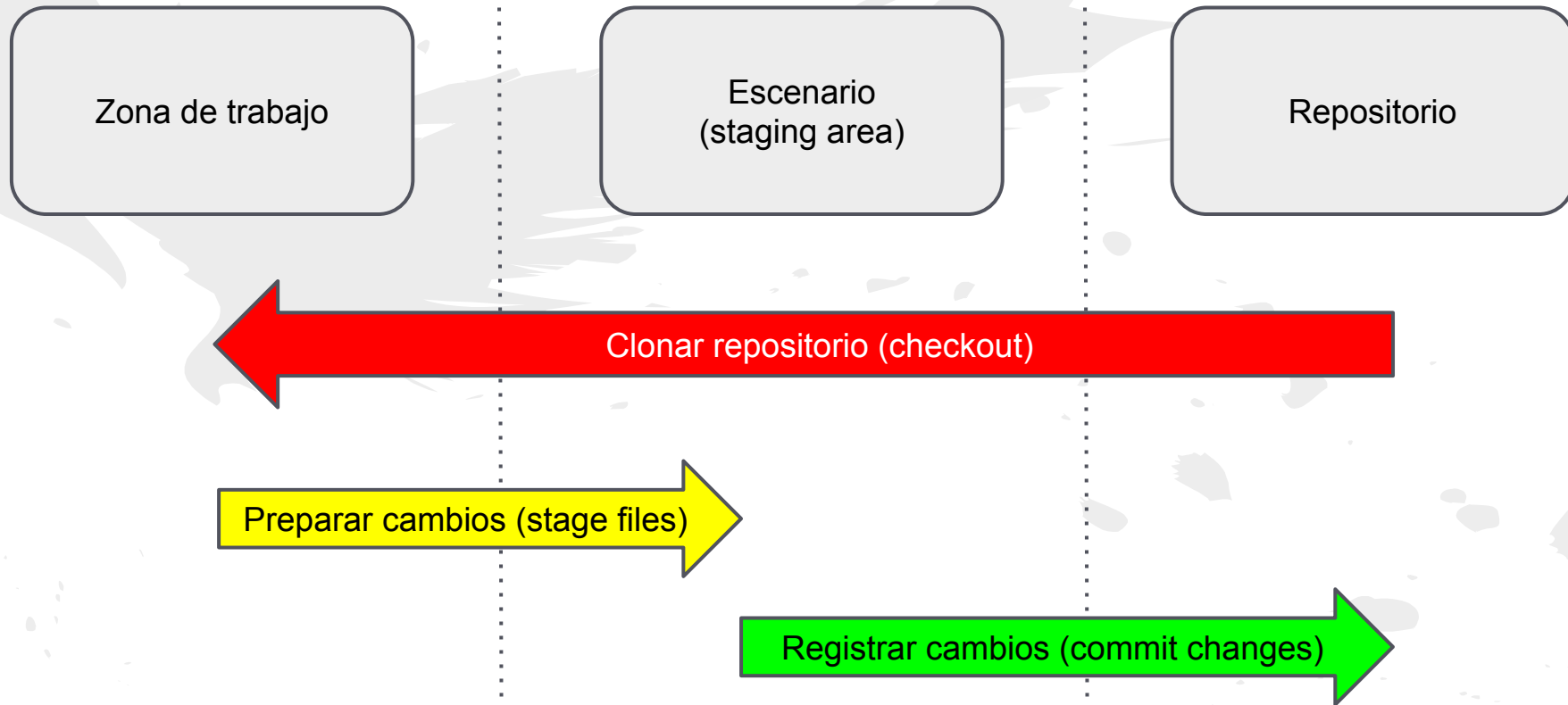
Casos de uso

Para grandes bases de código.

Para proyectos de software libre.

Para proyectos personales o académicos.

Conceptos de funcionamiento



Instalación

Multiplataforma

Distribuido en la mayoría de los sistemas

<http://git-scm.com>

Use local

```
$ git config <--global> user.name  
"First Name"
```

```
$ git config <--global> user.email  
"my@email.com"
```

Uso local

```
$ git init
```

Uso local

```
$ git status
```

Uso local

```
$ git log
```

Uso local

`.gitignore`

Use local

```
.idea/tab.xml  
bower_components/  
*.pyc
```

Use local

```
$ git add <--all|.|css/>
```


Uso local

```
$ git commit <-a|-m "Mensaje...">
```

Use local

```
$ git reset <--soft|--mixed|--hard>  
HEAD<^|^|...> <css/style.css>
```

Uso local

```
$ git revert <hash>
```

Uso local

```
$ git branch
```

Uso local

```
$ git checkout -b <branch_name>
```

Uso local

```
$ git checkout <branch_name>
```

Uso local

```
$ git branch -d <branch_name>
```

Uso local

```
$ git tag -a <tag_name> -m  
    <tag_message>
```

```
$ git tag -l
```

```
$ git show <tag_name>
```


Conflictos

```
$ git merge <--no-ff> <branch_name>
```

Conflictos

El número de planetas son

<<<<<<< HEAD

nueve

=====

ocho

>>>>>>> <branch_name>

en el sistema solar.

Conflictos

```
$ git checkout --ours
```

```
$ git checkout --theirs
```

Proveedores remotos

- GitHub
- BitBucket
- ~~Google Code~~

Proveedores remotos

GitHub

- Rep. públicos ilimitados.
- Rep. privados de pago.
- Usuarios ilimitados.
- 1GB hard limit.
- Referente para software abierto.

BitBucket

- Rep. públicos ilimitados.
- Rep. privados ilimitados.
- Hasta 5 usuarios.
- 2GB hard limit.
- Referente para proyectos personales o académicos.

Proveedores remotos

- GitLab
- Gogs
- Stash
- ~~Gitorious~~
- GitHub for Enterprise

Uso remoto

```
$ git remote add <remote> <uri>
```

Uso remoto

```
$ git fetch <--prune> <remote>
```

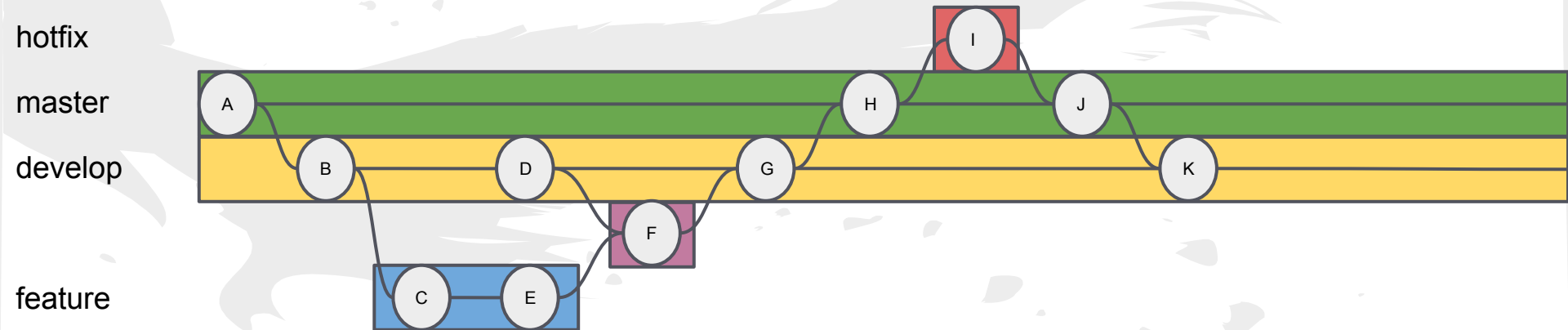

Use remote

```
$ git pull <remote> <local_branch>
```

Uso remoto

```
$ git push <remote> <local_branch>  
      <--force>
```

Buenas prácticas



Buenas prácticas

- Longitud de los mensajes, no superior a 50 columnas.
- Tiempo verbal de los mensajes, presente.
- Idioma de los mensajes, inglés.
- Evitar subir contenido sensible.
- Evitar desarrollar varias personas directamente sobre la misma rama. Usar ramas alternativas.
- No cambiar la historia después de subir cambios al remoto.

Colaboración

- Incidencias (issues).
- Documentación (wikis).
- Copias (forks).
- Petición de integración (pull requests).
- Integración continua (CI).

Preguntas, licencia y agradecimientos

Autor:

Javier Provecho Fernández

@javierprovecho



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).



Escuela Técnica Superior de Ingeniería Informática
de Valladolid



Grupo Universitario de Informática
de Valladolid