

Control de versiones con Git

¡Hola!

Iván López González
jlopez@suse.de



David Díaz González
dgonzalez@suse.de

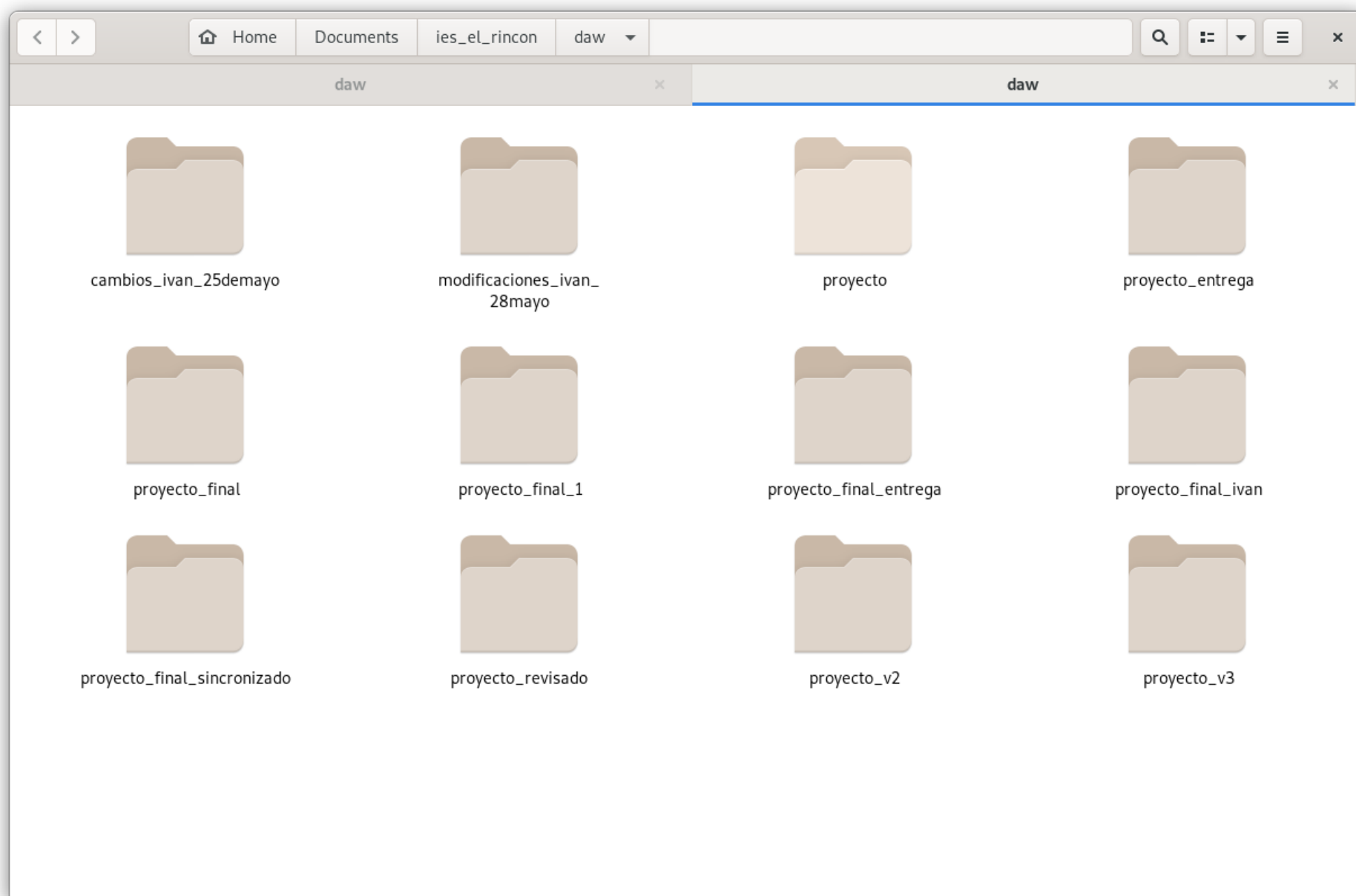
YaST

the installation and configuration tool for openSUSE and the SUSE Linux Enterprise

<http://yast.opensuse.org>

Nuestro primer SCV: copias manuales

Periódicamente y/o antes de introducir nuevos cambios.
(localmente, en un dispositivo externo, Dropbox, Google Drive, etc)



¿Problemas?

¿Problemas?

- Ineficiente.
- Propenso a errores.
- Duplicidad innecesaria.
- No se registran los cambios o diferencias.
- Tampoco quién ni cuándo.
- Complica (muchísimo) el trabajo en equipo.
- ...

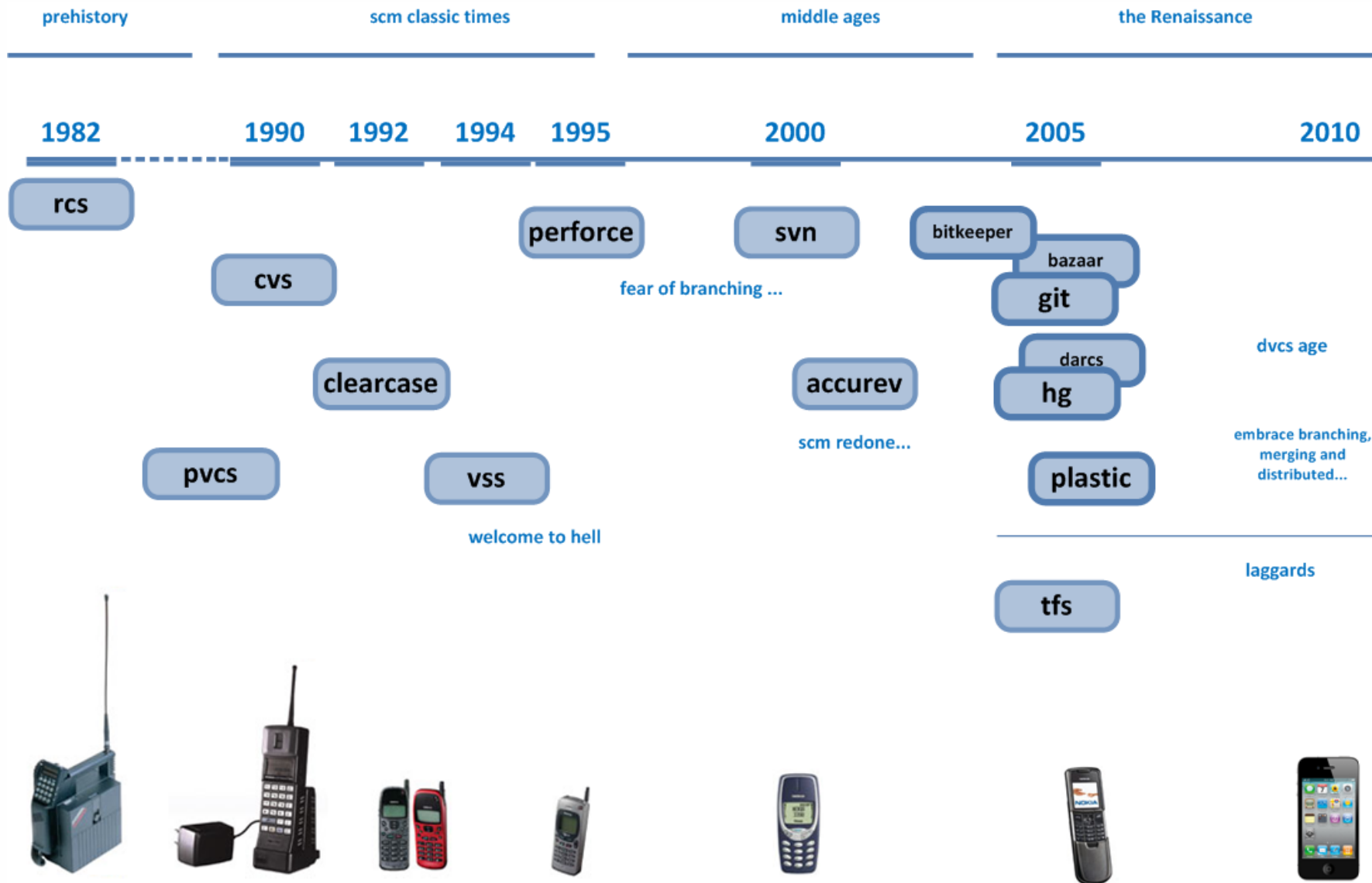
Sistemas de control de versiones

*[...] **registra los cambios realizados** en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.*

[Git Book - 1.1 Acerca del control de versiones](#)

Sistemas de control de versiones

- Diseñados para la gestión y el desarrollo de proyectos.
- Mantienen un histórico completo del proyecto.
- Permiten recuperar estados anteriores.
- Ver diferencias entre estados/versiones.
- Facilitan el trabajo en equipo.



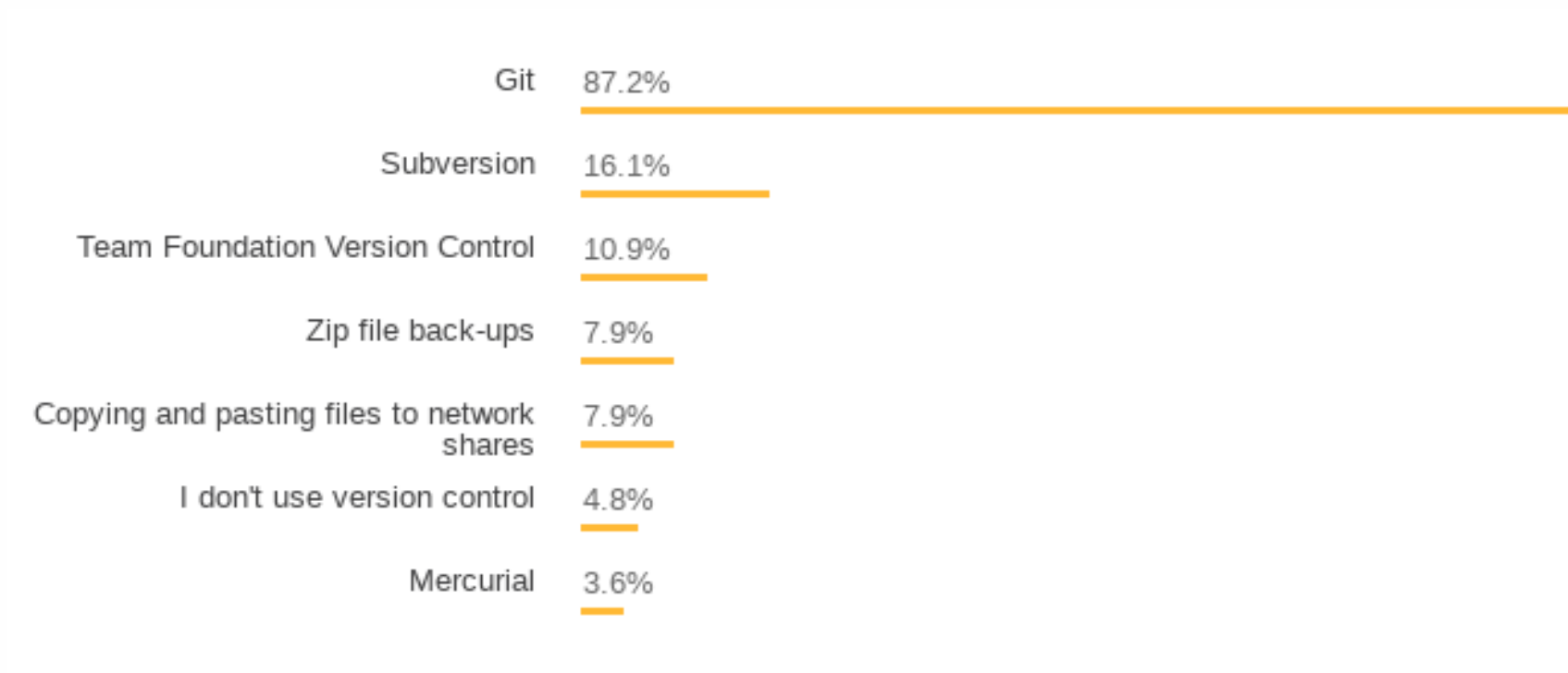
<http://blog.plasticscm.com/2010/11/version-control-timeline.html>



git

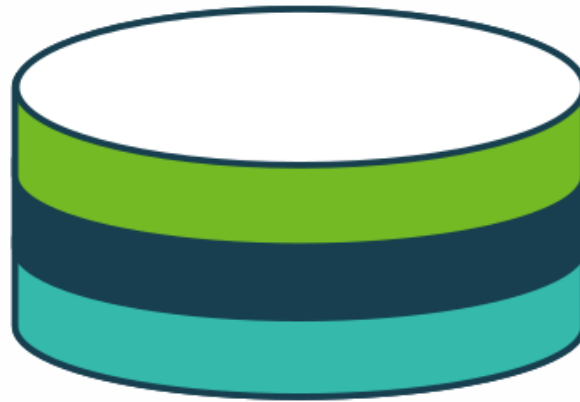
¿Por qué Git?

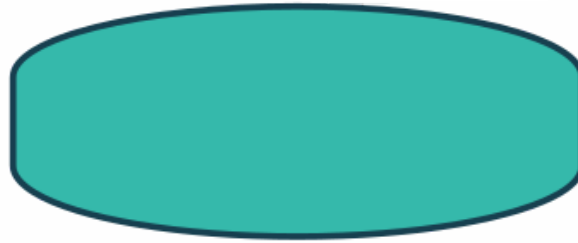
- Es *open source*.
- Rápido y eficiente.
- El más usado (según encuesta de StackOVerflow).
- El que nosotros usamos.



<https://insights.stackoverflow.com/survey/2018#work--version-control>

Repositorio









preparar





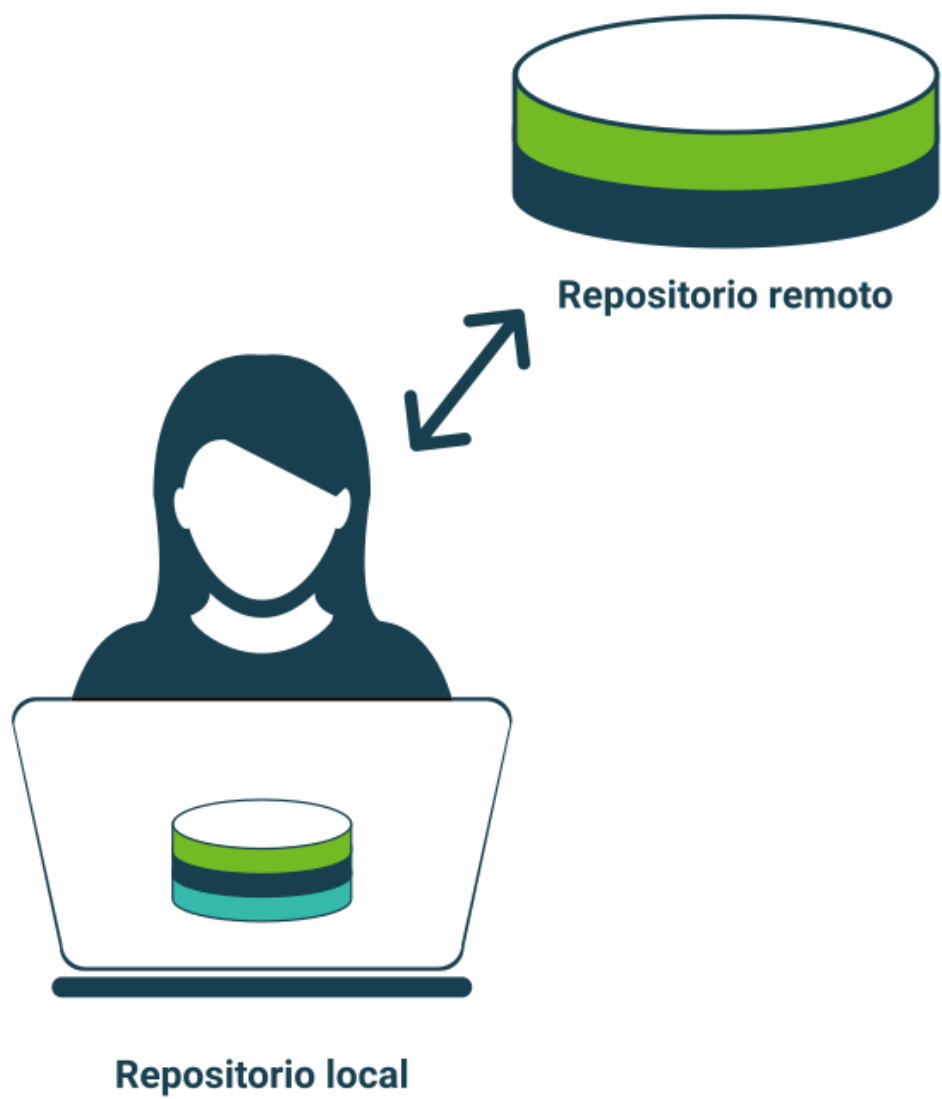


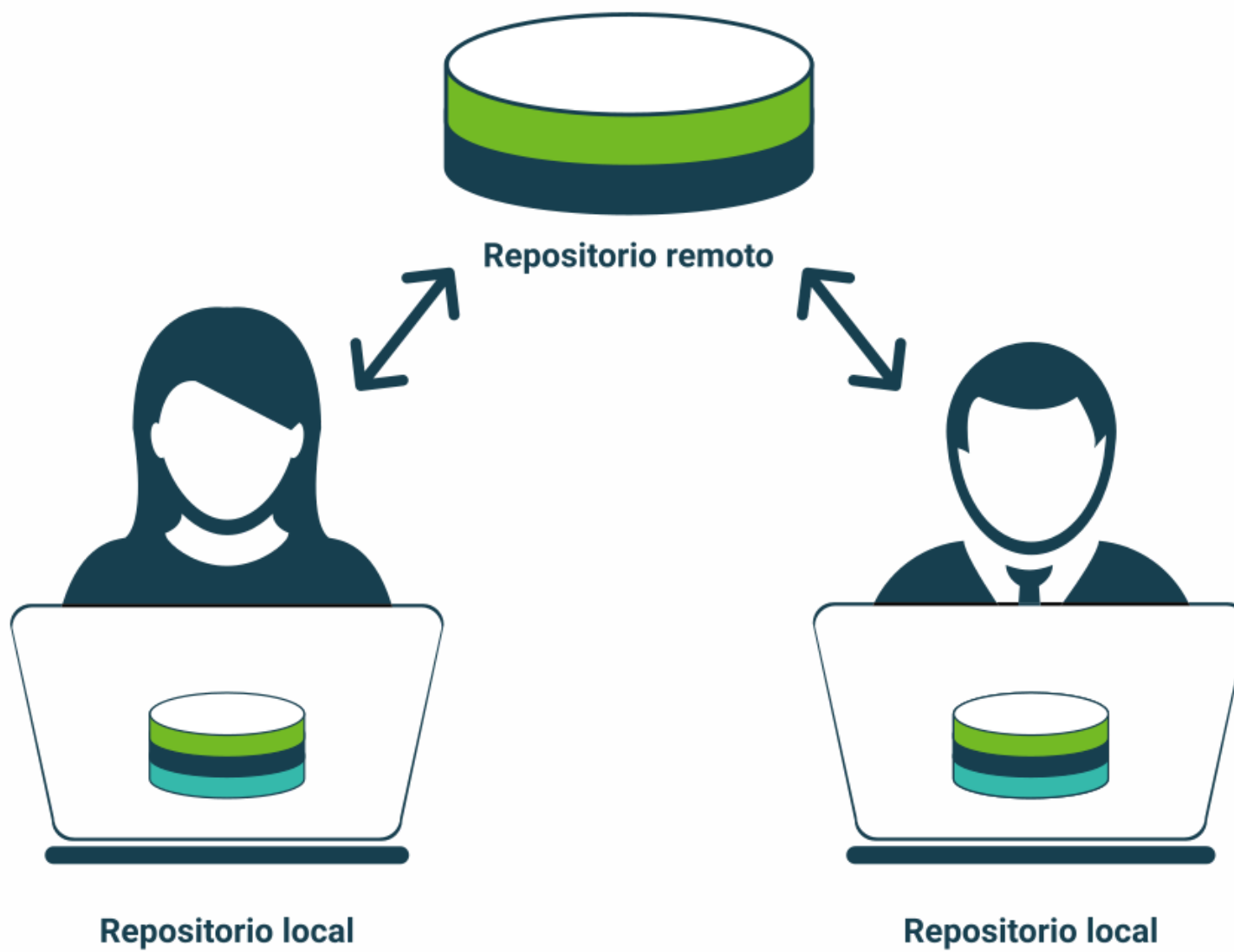
Git es *distribuido*

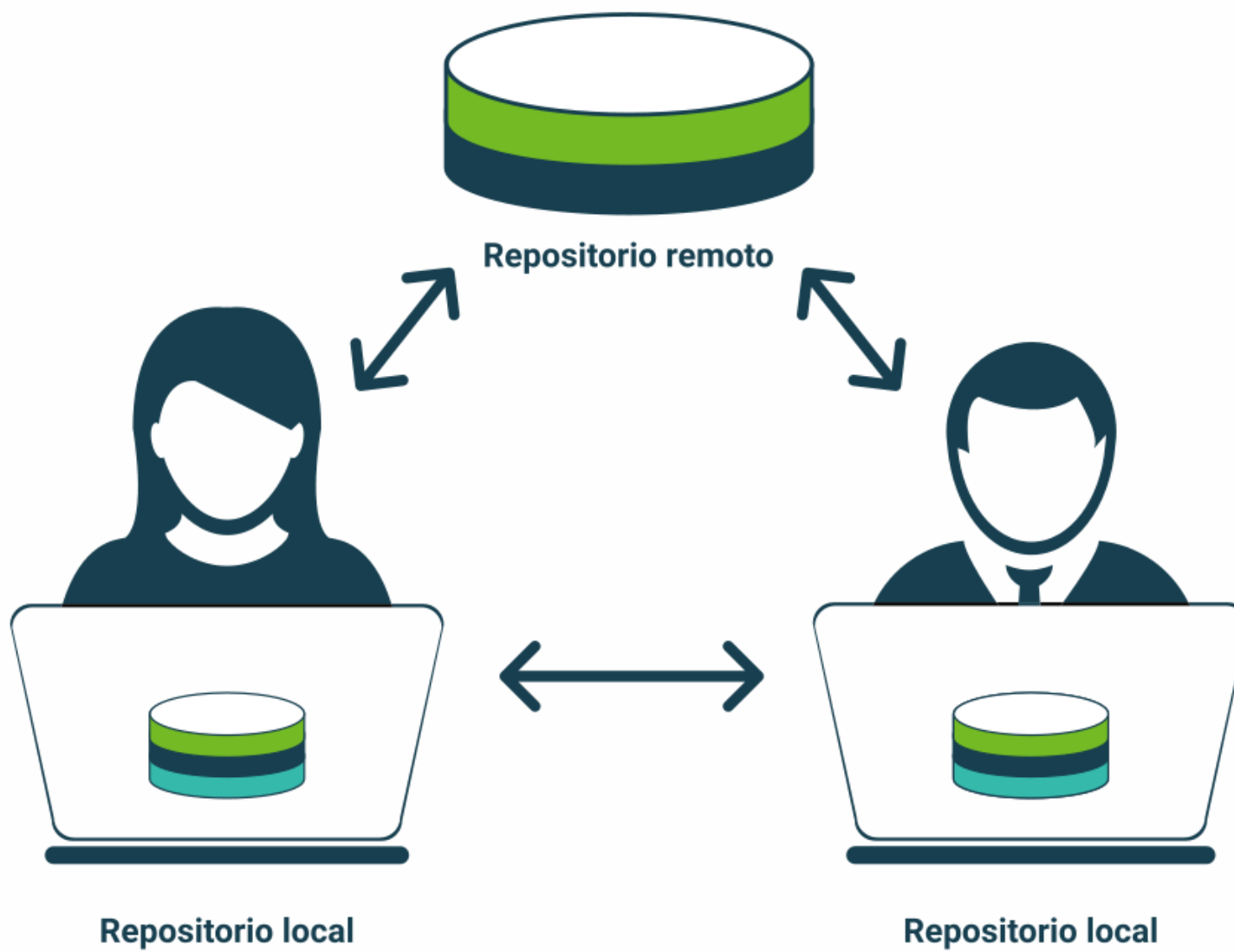
- Cada colaborador tiene una copia completa del repositorio y su histórico.
- La mayor parte de las operaciones habituales se pueden hacer sobre la copia local.
- Sólo se necesita conexión a Internet (o una red local, según el caso) para "enviar" o "recibir" los últimos cambios a/desde cualquier copia accesible.



Repositorio local







Plataformas de desarrollo colaborativo



GitLab

<https://about.gitlab.com/>



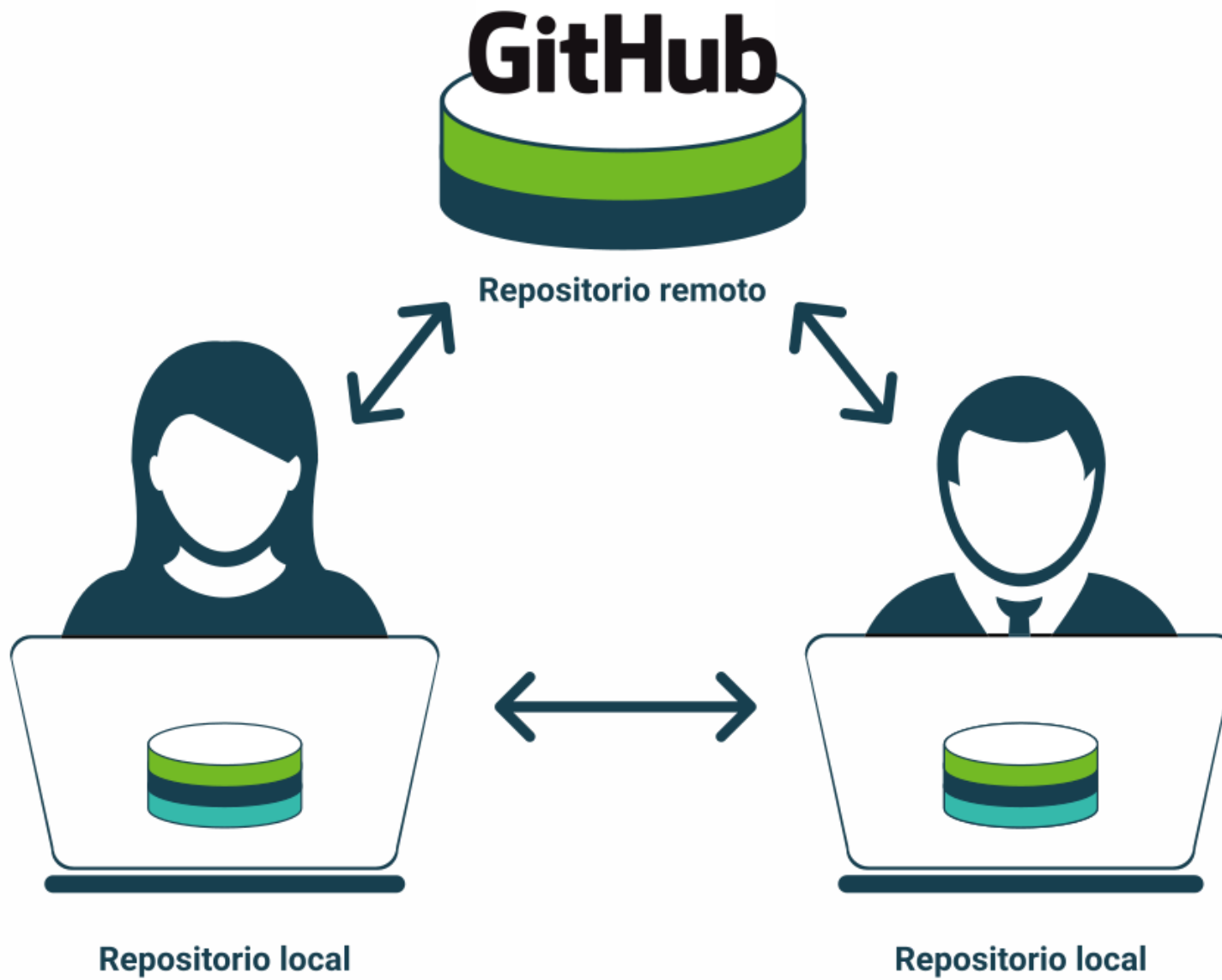
GitHub

<https://github.com/>



Bitbucket

<https://bitbucket.org>



Manos a la obra

Instalación

 ¡Ya incluido en **openSUSE!** 

Para la instalación en otras distribuciones de GNU/Linux u otros sistemas,
[visitar la documentación oficial](#)

Configuración básica

- `git config --global user.name 'Nombre'`
- `git config --global user.email 'email@ejemplo.com'`

Comandos básicos (para trabajar en local)

- `git init` Inicializa un repositorio.
- `git status` Muestra el estado actual del repositorio.
- `git add <fichero>` Prepara fichero(s) modificados para ser confirmados.
- `git commit` Confirma los cambios generando, una entrada en el histórico.
- `git log` Muestra todos los *commits* que se han realizado.
- `git diff` Permite visualizar las diferencias entre el directorio de trabajo y el *área de preparación*.
- `git diff --cached` Permite visualizar las diferencias entre el *área de preparación* y la última confirmación.


Comandos básicos (para sincronizar con una copia remota)

- `git pull` Actualiza la copia local con los cambios *recibidos* desde la copia remota.
- `git push` Actualiza la copia remota *enviando* los cambios disponibles en la copia local.

Recursos

Afortunadamente, hay una gran cantidad de recursos disponibles (la mayoría gratuitos).

Algunas recomendaciones para empezar,

- La documentación de Git - <https://git-scm.com/doc>
Especialmente [Pro Git book](#), también [disponible en español](#)
- [git - the simple guide](#), también [disponible en español](#)
- Los tutoriales de
 - Atlassian Bitbucket - <https://www.atlassian.com/git/tutorials>
Bastante detallados e ilustrados.
 - GitHub - <https://guides.github.com>
[Git Handbook](#) y [Hello World](#) pueden ser buenos puntos de partida.
- [Domina Git desde cero](#) - una serie de completos artículos en español
-  [Git tutorial](#)
- ...

¡Gracias!

