



Introducción a los Repositorios de Código Distribuido

Repositorios Remotos

Diego Madariaga

Contenidos de la clase

1. Repositorios remotos
2. Conceptos de repositorios
3. Visualización de operaciones

1.

Repositorios remotos

Repositorios remotos

- ▷ En un principio, trabajamos con repositorios locales
 - Primeros 2 laboratorios
- ▷ La existencia de repositorios remotos es una característica importante de Git, que permite trabajar de forma distribuida

git clone

- ▷ Un *clon* es una copia de un repositorio
- ▷ Contiene todos los objetos del repositorio original
 - Es un repositorio independiente y autónomo
- ▷ Permite el trabajo local y descentralizado
- ▷ Es lo que permite que proyectos grandes sean manejados de forma eficiente con Git

git clone

- ▷ Clonar un repositorio es el primer paso en el desarrollo distribuido
- ▷ Los repositorios se relacionan entre sí mediante conexiones que permiten el intercambio de datos
 - Git establece estas conexiones mediante *remotes*

remote

- ▷ Un *remote* es una referencia a otro repositorio
- ▷ Contiene una URL a un repositorio Git
- ▷ Un repositorio puede tener múltiples referencias *remote*, que permiten crear redes para el intercambio de datos

remote

- ▷ Git permite transferir datos desde un repositorio a otro mediante. Por ejemplo, con los comandos push y pull
 - **Pull:** sincronizar un clon trayendo los cambios del repositorio original
 - **Push:** Enviar los cambios realizados en un clon al repositorio original

2.

Conceptos de repositorios

Tipos de repositorios

- ▷ Un repositorio puede ser de dos tipos:
 - **Bare** (vacío)
 - **Development, *nonbare*** (de desarrollo)

Repositorios de tipo Development

- ▷ Usado para desarrollo diario, normal
- ▷ Mantiene una noción de “*rama actual*” y provee una copia de la rama actual en el directorio de trabajo

Repositorios de tipo Bare

- ▷ No tiene un directorio de trabajo
- ▷ No debe ser usado para desarrollo normal
- ▷ No tiene una noción de “*rama actual*”
- ▷ No se debieran hacer commits directamente a un repositorio bare
- ▷ No mantiene referencias *remote*

Repositorios de tipo Bare

- ▷ Puede parecer de poca utilidad
- ▷ Tiene un rol crucial como fundamento para el desarrollo colaborativo
- ▷ Los desarrolladores clonan (**clone**) y extraen (**fetch**) desde un repositorio bare, para luego enviarle actualizaciones (**push**)

Tipos de repositorios

- ▷ Por defecto `git clone` crea un repositorio de tipo `development`
- ▷ El comando `git clone` con la opción `--bare` crea un repositorio de tipo `bare`
- ▷ Como buena práctica, todo repositorio publicado debería ser de tipo `bare`

Repositorios clones

- ▷ El comando `git clone` crea un nuevo repositorio basado en uno original
- ▷ Por defecto, el repositorio clon mantiene una referencia a su repositorio padre por medio de un *remote* llamado “origin”
- ▷ Por el contrario, el repositorio original no tiene noción alguna de ningún clon

Repositorios remotos

- ▷ El repositorio en el que se realiza el desarrollo diario se llama repositorio actual o local
- ▷ Un repositorio con el cual se intercambian archivos, se llama repositorio remoto
- ▷ Un repositorio remoto no tiene que estar necesariamente en una máquina distinta o físicamente remoto (puede estar en la misma máquina)

Repositorios remotos

- ▷ Una referencia remota provee un nombre para un repositorio específico, en vez de usar la URL del repositorio
- ▷ Para realizar una conexión con una referencia remota, se utiliza el nombre de la referencia en conjunto con una rama la cual se quiere seguir

Repositorios remotos

- ▷ El comando `git remote` se utiliza para manipular las referencias remotas
- ▷ Además de `git clone`, existen otros comandos que refieren al uso de repositorios remotos

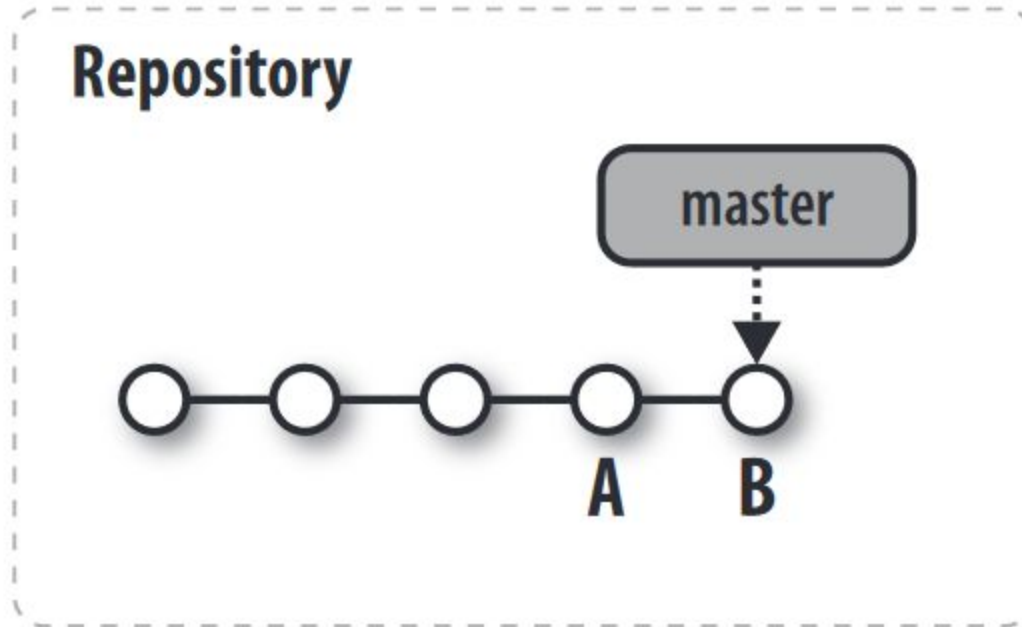
Repositorios remotos

- ▷ `git fetch` descarga contenidos de un repositorio remoto (objetos y metadatos)
- ▷ `git pull` es similar a `git fetch`, pero inmediatamente ejecuta `git merge` para crear un commit fusionando con los contenidos remotos
- ▷ `git push` transfiere objetos y metadatos hacia un repositorio remoto

3.

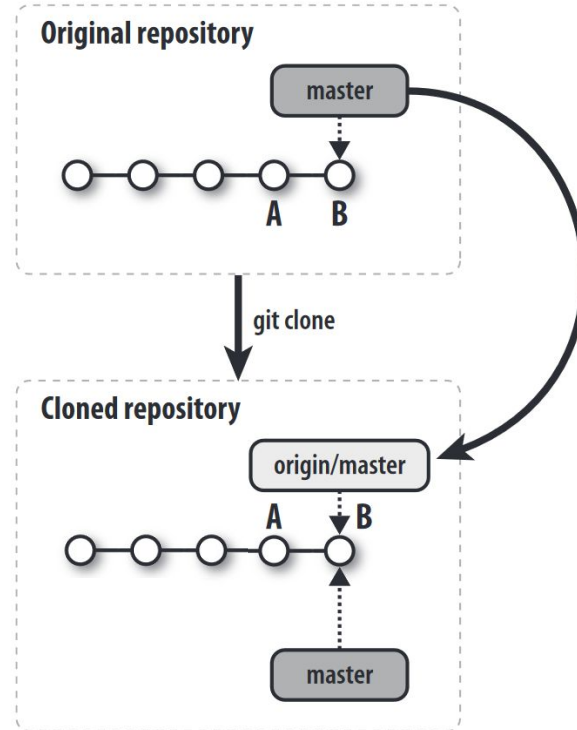
Visualización de operaciones

Ejemplo de repositorio remoto



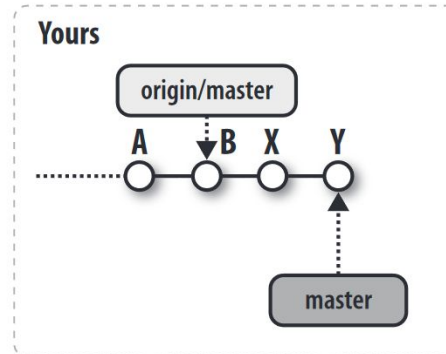
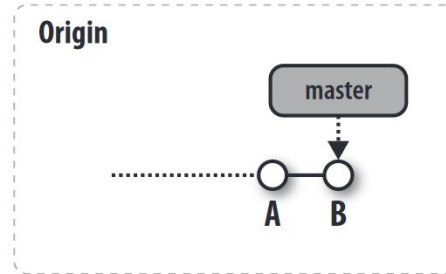
Ejemplo de repositorio remoto

git clone produce 2
repositorios
independientes



Ejemplo de repositorio remoto

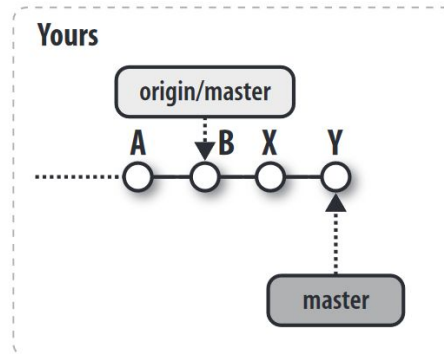
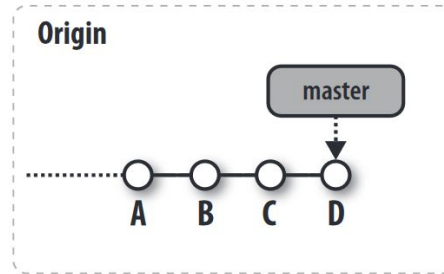
Historias alternativas



Ejemplo de repositorio remoto

Historias alternativas

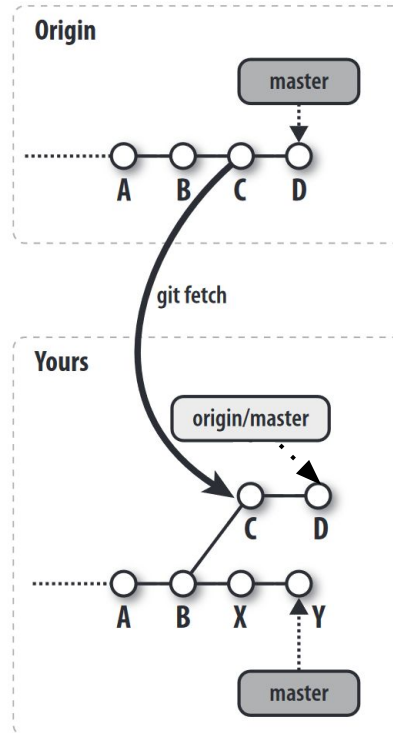
En este punto, un push al repositorio original sería rechazado



Ejemplo de repositorio remoto

Historias alternativas

git fetch

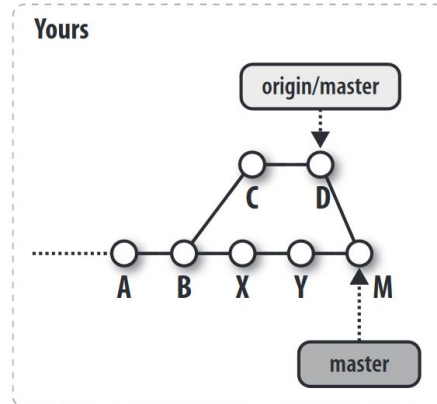
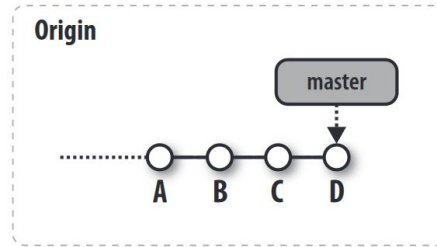


Ejemplo de repositorio remoto

Historias alternativas

git fetch

git merge origin/master



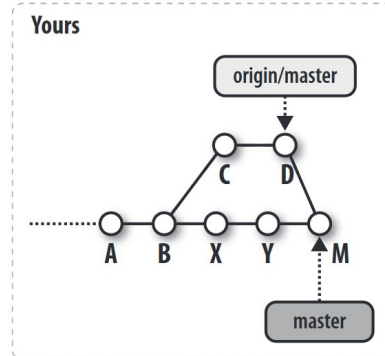
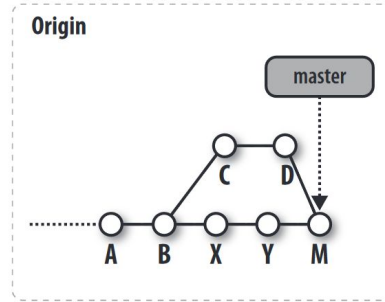
Ejemplo de repositorio remoto

Historias alternativas

git fetch

git merge origin/master

git push





Introducción a los Repositorios de Código Distribuido

Repositorios Remotos

Diego Madariaga