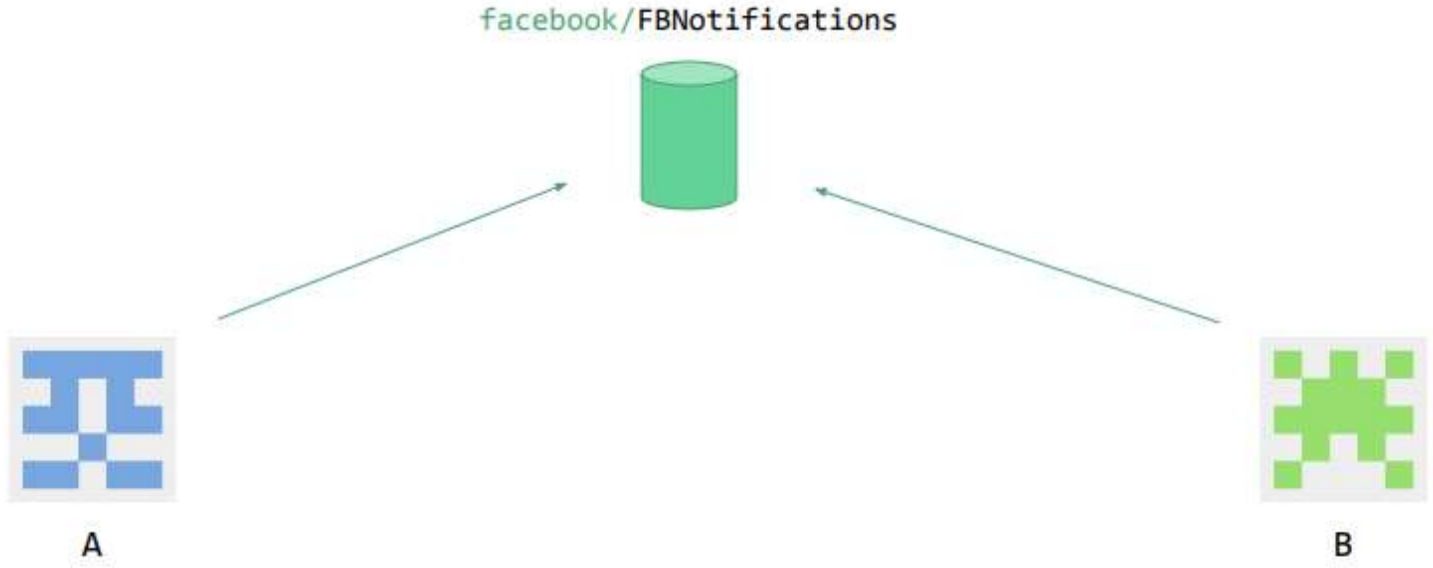
WORKFLOW EN GIT

Problemas en nuestro workflow

- Ejemplo: Dos desarrolladores (A y B) están colaborando en un proyecto open-source de Facebook llamado FBNotifications.



Problemas en nuestro workflow

- A está muy seguro de que sus últimos cambios en local son correctos por lo que procede a añadirlos a un commit y hacer un push directamente al repositorio

Problemas en nuestro workflow

- PROBLEMA: Al pasar los tests del commit del repositorio, el manager de los desarrolladores descubre que hay un bug causado por un bucle infinito en el código de A, por lo que le pide a B que lo solucione.

Problemas en nuestro workflow

- POSIBLE SOLUCIÓN:

Problemas en nuestro workflow

- SOLUCIÓN IMPUESTA POR EL MANAGER:
- A partir de ahora habrá una nueva forma de trabajar: Siempre y cuando se deba hacer cambios en el repositorio, se hará mediante **Pull Request**

Trabajando con Workflow en Git

- Pull Request:
  - Conjunto de cambios propuestos a un repositorio por un usuario y aceptado o rechazado por otro usuario colaborador de ese repositorio.
  - Cada Pull Request tiene su propio hilo de discusión
- Fork
  - Una copia personal en tu cuenta del repositorio de otro usuario
  - Permite tener dos repositorios git idénticos pero con distinta URL
  - Los forks nos permiten hacer cambios libremente de un proyecto sin afectar el original
  - Esta copia permanecerá adjunta al original permitiendo remitir los cambios a éste mediante un mecanismo llamado pull-request
  - También es posible mantener tu fork up-to-date actualizándose con los últimos cambios del original
  - Tendremos dos repositorios independientes que pueden cada uno evolucionar de forma totalmente autónoma.

Diferencia entre Fork y Clone

- CLONE
  - Cuando hacemos un clon de un repositorio, bajas una copia del mismo a tu máquina
  - Empiezas a trabajar, haces modificaciones y haces un push
  - Cuando haces el push estás modificando el repositorio que has clonado
- FORK
  - Cuando haces un fork de un repositorio, se crea un nuevo repositorio en tu cuenta de Github o Bitbucket, con una URL diferente (fork)
  - Acto seguido tienes que hacer un clon de esa copia sobre la que empiezas a trabajar de forma que cuando haces push, estás modificando TU COPIA (fork)
  - El repositorio original sigue intacto

Fork - Uso

- Permitir a los desarrolladores contribuir a un proyecto de forma segura.
- Ejemplo de cosas que podemos hacer en un proyecto:
  1. Usuario1 hace un fork de mi repositorio, para lo que sólo necesito darle permiso de lectura.
  2. Usuario1 trabaja en SU COPIA (fork) del repositorio. Como es suya, puede hacer lo que quiera, la puede borrar, corromper, reescribir la historia del proyecto…, es su copia(fork) y no nos influye
  3. Cuando Usuario1 termina de programar y testear el parche, me avisa de que ya lo tiene y me dice “En la rama parche_de_usuario1 de MI COPIA (fork), tienes el parche que corrige el Bug XXXX”
  4. Voy a su repositorio, miro lo que ha hecho y si está bien lo incorporo (merge) a mi repositorio, que es el original

Fork - Uso

- Las ventajas que tiene utilizar Fork:
  1. Usuario1 trabaja con SU COPIA. En ningún momento le tengo que dar acceso al repositorio central.
  2. El proceso de incorporación de los cambios de Usuario1 es muy sencillo. Si no hay conflictos en los ficheros puede que sea tan simple como ejecutar un par de comandos git.
  3. Usuario1 tiene muy fácil contribuir, no le cuesta esfuerzo.
  4. Puedo gestionar muy fácilmente las contribuciones de muchas personas

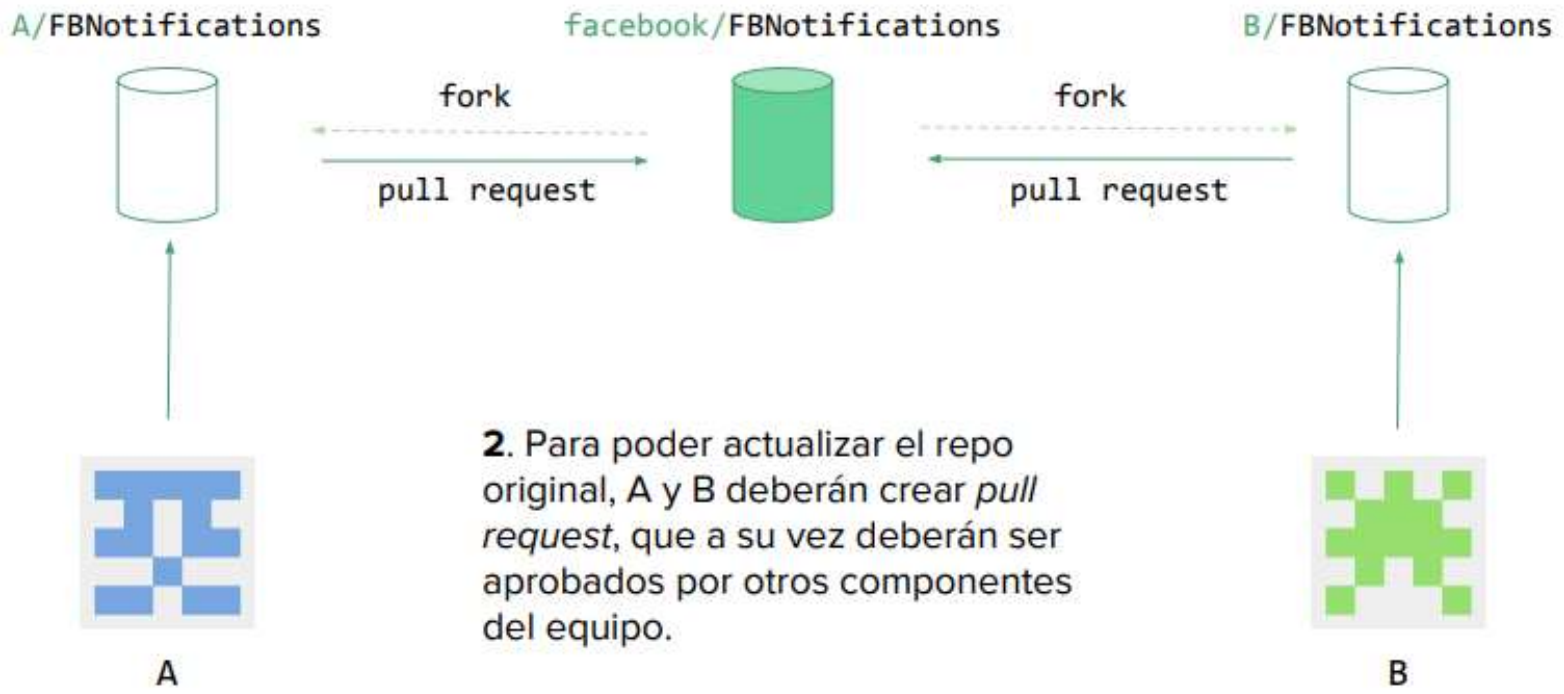
Trabajando con Workflow en Git

- Para el ejemplo anterior, el manager podría haber usado una combinación entre Fork y Pull-Request para evitar problemas de código erróneo.
- Primero, los usuarios hacen un “fork” del repositorio

1. A y B se hacen un *fork* cada unx del repositorio para hacer *commit* de sus cambios sin afectar al original.

Trabajando con Workflow en Git

- Los usuarios realizan sus modificaciones en local. Cuando han terminado crean un “pull request” y lo envían al repositorio. Sus cambios deben ser aprobados por el administrador



Trabajando con Workflow en Git

- Ejemplo: trabajando con Fork y Git Pull Request

1. Crear forks

http://aprendegit.com/fork-de-repositorios-para-que-sirve/

2. Trabajar con Pull Request

http://aprendegit.com/que-es-un-pull-request/

- Trabaja con Forks – múltiples repositorios remotos

http://aprendegit.com/mantener-tu-fork-al-dia/

Otro ejercicio sencillo:

- https://github.com/hcs/bootcamp-git/wiki/Exercise-Making-a-Pull-Request