[[1]](#footnote-1)

Repositorios Remotos GIT

Lòpez, Juan.

jl091505@ingenieria.sanmartin.edu.co

Fundación Universitaria San Martin

*Resumen*— Para poder colaborar en cualquier proyecto Git, necesitara saber cómo gestionar sus repositorios remotos. Los repositorios remotos son versiones de su proyecto que se encuentran alojados en Internet o en algún punto de la red. Puede tener varios, cada uno de los cuales puede ser de sólo lectura, o de lectura/escritura, según los permisos que tenga. Colaborar con otros implica gestionar estos repositorios remotos, y mandar (push) y recibir (pull) datos de ellos cuando necesite compartir cosas.

Gestionar repositorios remotos implica conocer cómo añadir repositorios nuevos, eliminar aquellos que ya no son válidos, gestionar ramas remotas e indicar si están bajo seguimiento o no, y más cosas. En esta sección veremos todos estos conceptos.

*Índice de Términos—Push, Ficth,*

# repositorios remotos

## Diferencia entre colores RGB y Hexadecimal

## 

La descripción RGB (del inglés Red, Green, Blue; "rojo, verde, azul") de un color hace referencia a su composición de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul. Este un modelo de color basado en lo q se conoce como síntesis aditiva, con lo que es posible representar a un color por la mezcla por adición de los tres colores luz primarios.

Para indicar con qué proporción se mezcla cada color, le asignamos un valor a cada uno de los colores primarios, así, por ejemplo, el valor 0 significa que no interviene en la mezcla, y en la medida que ese valor aumenta, aportará más intensidad a la mezcla.

Lo que conocemos como píxel es en realidad un conjunto de tres puntos, uno rojo, uno verde y uno azul, cada uno de los cuales brilla con una determinada intensidad.

La gama de colores de la web consiste en 216 combinaciones de rojo, verde y azul, donde cada color puede tomar un valor entre seis diferentes (en hexadecimal): #00, #33, #66, #99, #CC o #FF, que tienen un porcentaje de intensidad de 0%, 20%, 40%, 60%, 80% y 100%, respectivamente.

La codificación hexadecimal del color nos permite expresar fácilmente un color especifico de la escala RGB, utilizando la notación hexadecimal. esta notación la encontramos, por ejemplo, en el lenguaje HTML, JavaScript, o ActionScript, y otros lenguajes

## Convertir RGB a Hexadecimal.

Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos: - Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176

Transformar el valor decimal de cada componente al sistema numérico hexadecimal. Se trata de una operación exclusivamente matemática, por lo que puedes utilizar una calculadora. En el ejemplo anterior, el valor hexadecimal de cada componente es: R = 47, G = 62, B = B0 - Para obtener el color completo en formato RGB hexadecimal, se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #. De esta forma, el color del ejemplo anterior es #4762B0 en formato RGB hexadecimal.

Siguiendo el mismo ejemplo de las secciones anteriores, el color del párrafo se indica de la siguiente forma utilizando el formato RGB hexadecimal:

p { color: #4762B0; }

El formato RGB hexadecimal es la forma más compacta de indicar un color, ya que incluso es posible comprimir sus valores cuando todas sus componentes son iguales dos a dos:

#AAA = #AAAAAA

#FFF = #FFFFFF

#A0F = #AA00FF

#369 = #336699

En el siguiente ejemplo se establece el color de fondo de la página a blanco, el color del texto a negro y el color de la letra de los titulares se define de color rojo:

body { background-color: #FFF; color: #000; }

h1, h2, h3, h4, h5, h6 { color: #C00; }

Las letras que forman parte del color en formato RGB hexadecimal se pueden escribir en mayúsculas o minúsculas indistintamente. No obstante, se recomienda escribirlas siempre en mayúsculas o siempre en minúsculas para que la hoja de estilos resultante sea más limpia y homogénea.

Framework Responsivos y mas usados

## Recibiendo de repositorios remotos

Como acabas de ver, para recuperar datos de tus repositorios remotos puedes ejecutar:

$ git fetch [remote-name]

Este comando recupera todos los datos del proyecto remoto que no tengas todavía. Después de hacer esto, deberías tener referencias a todas las ramas del repositorio remoto, que puedes unir o inspeccionar en cualquier momento. (Veremos qué son las ramas y cómo utilizarlas en más detalle en el Capítulo 3.)

Si clonas un repositorio, el comando añade automáticamente ese repositorio remoto con el nombre de "origin". Por tanto, git fetch origin recupera toda la información enviada a ese servidor desde que lo clonaste (o desde la última vez que ejecutaste fetch). Es importante tener en cuenta que el comandofetch sólo recupera la información y la pone en tu repositorio local —no la une automáticamente con tu trabajo ni modifica aquello en lo que estás trabajando. Tendrás que unir ambos manualmente a posteriori—.

puede usar el comando git pull para recuperar y unir automáticamente la rama remota con tu rama actual. Éste puede resultarte un flujo de trabajo más sencillo y más cómodo; y por defecto, el comando git clone automáticamente configura tu rama local maestra para que siga la rama remota maestra del servidor del cual clonaste (asumiendo que el repositorio remoto tiene una rama maestra). Al ejecutar git pull, por lo general se recupera la información del servidor del que clonaste, y automáticamente se intenta unir con el código con el que está trabajando actualmente.

## Enviando a repositorios remotos

Cuando tu proyecto se encuentra en un estado que quieres compartir, tienes que enviarlo a un repositorio remoto. El comando que te permite hacer esto es sencillo: git push [nombre-remoto][nombre-rama]. Si quieres enviar tu rama maestra (master) a tu servidor origen (origin), ejecutarías esto para enviar tu trabajo al servidor:

$ git push origin master

Este comando funciona únicamente si has clonado de un servidor en el que tienes permiso de escritura, y nadie ha enviado información mientras tanto. Si tú y otra persona clonais a la vez, y él envía su información y luego envías tú la tuya, tu envío será rechazado. Tendrás que bajarte primero su trabajo e incorporarlo en el tuyo para que permita hacer un envío..

## Inspeccionando un repositorio remoto

Si quiere ver más información acerca de un repositorio remoto en particular, puedes usar el comando git remote show [nombre]. Si ejecutas este comando pasándole el nombre de un repositorio, comoorigin, obtienes algo así:

$ git remote show origin

\* remote origin

URL: git://github.com/schacon/ticgit.git

Remote branch merged with 'git pull' while on branch master

master

Tracked remote branches

master

ticgit

Esto lista la URL del repositorio remoto, así como información sobre las ramas bajo seguimiento. Este comando te recuerda que si estás en la rama maestra y ejecutas git pull, automáticamente unirá los cambios a la rama maestra del remoto después de haber recuperado todas las referencias remotas. También lista todas las referencias remotas que ha recibido.

El anterior es un sencillo ejemplo que te encontrarás con frecuencia. Sin embargo, cuando uses Git de forma más avanzada, puede que git remote show muestre mucha más información:

$ git remote show origin

\* remote origin

URL: git@github.com:defunkt/github.git

Remote branch merged with 'git pull' while on branch issues

issues

Remote branch merged with 'git pull' while on branch master

master

New remote branches (next fetch will store in remotes/origin)

caching

Stale tracking branches (use 'git remote prune')

libwalker

walker2

Tracked remote branches

acl

apiv2

dashboard2

issues

master

postgres

Local branch pushed with 'git push'

master:master

Este comando muestra qué rama se envía automáticamente cuando ejecutas git push en determinadas ramas. También te muestra qué ramas remotas no tienes todavía, qué ramas remotas tienes y han sido eliminadas del servidor, y múltiples ramas que serán unidas automáticamente cuando ejecutes git pull.

# Conclusiones

Si quieres renombrar una referencia a un repositorio remoto, en versiones recientes de Git puedes ejecutar git remote rename. Por ejemplo, si quieres renombrar pb a paul, puedes hacerlo de la siguiente manera:

$ git remote rename pb paul

$ git remote

origin

paul

Conviene mencionar que esto también cambia el nombre de tus ramas remotas. Lo que antes era referenciado en pb/master ahora está en paul/master.

Si por algún motivo quieres eliminar una referencia —has movido el servidor o ya no estás usando un determinado mirror, o quizás un contribuidor ha dejado de contribuir— puedes usar el comando git remote rm:

$ git remote rm paul

$ git remote

origin

## Eliminando y renombrando repositorios remotos

Si quieres renombrar una referencia a un repositorio remoto, en versiones recientes de Git puedes ejecutar git remote rename. Por ejemplo, si quieres renombrar pb a paul, puedes hacerlo de la siguiente manera:

$ git remote rename pb paul

$ git remote

origin

paul

Conviene mencionar que esto también cambia el nombre de tus ramas remotas. Lo que antes era referenciado en pb/master ahora está en paul/master.

Si por algún motivo quieres eliminar una referencia —has movido el servidor o ya no estás usando un determinado mirror, o quizás un contribuidor ha dejado de contribuir— puedes usar el comando git remote rm:

$ git remote rm paul

$ git remote

origin

referencias

1. http://git-scm.com/book/es/Fundamentos-de-Git-Trabajando-con-repositorios-remotos

1. [↑](#footnote-ref-1)