



git

Trabajando con remotos

Cloning

Pushing and pulling

Github

Trabajando con remotos

- Si queremos compartir un repositorio a un colaborador, no podrá acceder ya que se encuentra en tu ordenador
- Una solución es hacer una copia del repositorio en el ordenador del colaborador, esto se llama **cloning**
- Con el tiempo se generarían dos historias diferentes del repositorio, con diferentes versiones y ramas



Trabajando con remotos

- De forma parecida a como se reunifican ramas en tu historial, también puedes unir diferentes copias de historiales en un mismo repositorio.
 - Subiendo los cambios de tu repositorio local (**pushing**) a un repositorio remoto
 - Bajando los cambios del repositorio remoto (**pulling**) a tu repositorio local
- Una idea sencilla y a la vez muy potente que permite trabajar fácil a equipos de diferentes tamaños y distintas filosofías de gestión de proyectos



Trabajando con remotos

- Podemos administrar los proyectos de una forma que represente la naturaleza descentralizada de Git



- Pero también podemos poner orden en el sistema estableciendo un repositorio remoto

Trabajando con remotos

Otros equipos pondrán orden en el sistema estableciendo un repositorio remoto en un servidor, como un repositorio central



Este repositorio central actúa como la versión oficial del código

Los programadores que trabajen en estos proyectos tendrán la costumbre de realizar con frecuencia subir (push) y bajar (pull) del repositorio central para asegurarse que su trabajo está sincronizado con el resto del equipo de desarrollo

Cloning

A no ser que comiences tú el proyecto, lo primero que se hace antes de trabajar con git es clonar el repositorio en tu repositorio local y guardar los repositorios remotos con los que trabajarás

Para ello utilizamos el comando `git clone`

A continuación indicamos la dirección del repositorio que queremos clonar

```
git clone https://github.com/usuario/repositorio.git
```

De esta forma creamos una copia perfecta del repositorio, con todo el historial, ramas y versiones

Git configurará automáticamente un repositorio remoto llamado origin

Cloning

Podemos añadir un repositorio remoto mediante el comando `git remote add`

Para ejecutarlo indicaremos el nombre de nuestro repositorio remoto y dónde se localiza

```
git remote add nuevoremoto https://github.com/usuario/repositorio.git
```

¿cuál será entonces la diferencia entre “remote add” y “clone”?

Pushing and pulling

pushing

Para actualizar el repositorio remoto después de realizar un commit utilizamos el comando `git push`

A este comando le pasaremos el nombre del repositorio remoto al que lo vamos a subir y el nombre de la rama que queremos subir (tenemos que haber creado antes la rama en local)

```
git push origin nueva_rama
```

Si vamos a subir ficheros en una rama que otro colaborador ha modificado, mostrará un error, por lo que tendremos que descargar los cambios antes

Pushing and pulling

pulling

Para actualizar el repositorio local con los datos del repositorio remoto utilizamos el comando `git pull`

A este comando le pasaremos el nombre repositorio remoto del que vamos a descargar en nuestra rama

```
git pull origin una_rama
```

Si se ha modificado un fichero que nosotros también hemos modificado, ocurrirá un conflicto, lo resolveremos de la misma manera

- ✓ Corregir los conflictos editando los ficheros
- ✓ Añadir los ficheros al stage area
- ✓ Realizar un commit
- ✓ Volver a realizar un push

Github

Existen muchas opciones para alojar repositorios remotos, puedes incluso crear el tuyo propio

Hay muchas opciones en la red, muchas gratuitas y además públicas (opensource)

Assembla

Bitbucket

CloudForge

Gitorious

Unfuddle

code.google

etc.

Github

Uno de los más populares servicios de repositorios remotos ahora es **GitHub**

Es utilizado por millones de desarrolladores,

Provee muchas funcionalidades adicionales para ayudar a administrar el trabajo en tus proyectos

Existen utilidades de terceros para añadir más funcionalidades como waffle.io, git-flow, etc.

Podemos explorar multitud de proyectos abiertos que están compartidos en github como por


ejemplo ruby on rails

Al buscar un proyecto tenemos distinta información.

Una breve descripción del proyecto

Estadísticas sobre el repositorio

Información sobre la rama actual, normalmente master

Es posible desde la misma página explorar el  [Code](#)

Al final de la página está la información del README donde explica las características del proyecto y cómo contribuir

Todos los proyectos incluyen un **control de errores (Issues)** a la derecha de la página  [Issues](#)

También existe la opción de peticiones (**pull requests**) para modificar código, junto a un hilo para discutir sobre esta  [Pull Requests](#) 72

Github



Watch 1,627

Star 23,479

Fork 8,950

Ruby on Rails <http://rubyonrails.org>

47,298 commits 35 branches 231 releases 2,445 contributors

branch: master rails / +

Merge pull request #17014 from grosser/grosser/fast-fixtures

tenderlove authored 2 days ago latest commit 5127857947

actionmailer	Only include the needed assertions	5 days ago
actionpack	Request#check_method no longer breaks when :en is not available locale	3 days ago
actionview	remove un-needed sort	5 days ago
activejob	tests, move `original_` var assignments outside of `begin`.	2 days ago
activemodel	Merge pull request #16409 from justinweiss/update_validation_context_...	9 days ago
activerecord	Merge pull request #17014 from grosser/grosser/fast-fixtures	2 days ago
activesupport	Merge pull request #17184 from fillman/master	5 days ago
ci	Skip isolated tests for PRs	22 days ago

Code

Issues 327

Pull Requests 431

Pulse

Graphs

HTTPS clone URL

<https://github.com/>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP

Github

En la esquina superior izquierda encontramos los siguientes botones




un cambio importante en el proyecto


Star lo añadirá a nuestra lista de proyectos favoritos


Fork nos permite clonar el proyecto en nuestra cuenta

Github

Crear un proyecto

 Owner


 Repository name

 vensiere ▾


 /

Great repository names are short and memorable. Need inspiration? How about **miniature-octo-wallhack**.

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**


You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** ▾

 |

Add a license: **None** ▾ 

Create repository

Github

Actualizar el repositorio

Quick setup — if you've done this kind of thing before

 **Set up in Desktop** or **HTTP** **SSH** `https://github.com/vensiere/prueba.git` 

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


...or create a new repository on the command line

```
touch README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/vensiere/prueba.git
git push -u origin master
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/vensiere/prueba.git
git push -u origin master
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

Webgrafia

<http://git-scm.com/>

<https://github.com>