

Introducción al control de versiones y trabajo colaborativo con GitHub

Oscar Perpiñán Lamigueiro

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con `git` y GitHub

Realizar y confirmar cambios (`add` y `commit`)

Publicar cambios (`push`)

Recibir cambios de un repositorio remoto y combinar con una copia local (`fetch`, `merge` y `pull`)

Trabajo en colaboración

Ramas (`branch`)

Combinación de código (`pull request` y `merge`)

Tareas y tableros 1 / 30

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc

<http://phdcomics.com/comics/archive.php?comiciid=1531>



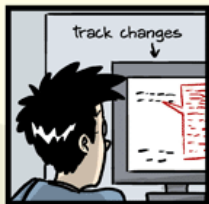
FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



<http://phdcomics.com/comics/archive.php?comiciid=1531>



FINAL_rev.18.comments7.
corrections9.MORE.30.doc

FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

<http://phdcomics.com/comics/archive.php?comid=1531>

¿Qué es el control de versiones y por qué debería importarte?

Introducción al
control de
versiones y trabajo
colaborativo con
GitHub

Oscar Perpiñán
Lamigueiro

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros 7 / 30

*El control de versiones es un sistema que **registra los cambios** realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se puedan **recuperar** versiones específicas más adelante.¹*

¹<https://git-scm.com/book/es/v1/>

¿Qué es el control de versiones y por qué debería importarte?

*El control de versiones es el cuaderno de laboratorio en el mundo digital. Es lo que los profesionales usan para realizar un **seguimiento** de lo que han hecho y para **colaborar** con otras personas. Cada gran proyecto de desarrollo de software se basa en ello, y la mayoría de los programadores lo utilizan para sus trabajos. Y **no sirve sólo para software**: libros, documentos, pequeños conjuntos de datos y cualquier cosa que cambie con el tiempo o que deba compartirse puede y debe almacenarse en un sistema de control de versiones.²*

²<https://swcarpentry.github.io/git-novice/>

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros 8 / 30

Viajar en el tiempo

- ▶ Nada que haya sido sometido a un control de versiones se pierde jamás (*salvo que realmente quieras eliminarlo...*)
- ▶ **Todas** las versiones antiguas de un fichero se almacenan: un fichero se puede revertir a un estado anterior sin límites.

Introducción al
control de
versiones y trabajo
colaborativo con
GitHub

Oscar Perpiñán
Lamigueiro

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (`add` y `commit`)

Publicar cambios (`push`)

Recibir cambios de un repositorio remoto y combinar con una copia local (`fetch`, `merge` y `pull`)

Trabajo en colaboración

Ramas (`branch`)

Combinación de código (`pull request` y `merge`)

Tareas y tableros 9 / 30

¿Qué? ¿Cuándo? ¿Quién?

Un sistema de control de versiones registra:

- ▶ El detalle de los cambios realizados.
- ▶ La fecha y hora en la que fueron realizados.
- ▶ La persona que los realizó.

Introducción al
control de
versiones y trabajo
colaborativo con
GitHub

Oscar Perpiñán
Lamigueiro

Conceptos básicos

¿Qué es el control de
versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en
GitHub

Configuración del
ordenador para el uso de la
cuenta

Mi primer repositorio: crear
y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar
cambios (`add` y `commit`)

Publicar cambios (`push`)

Recibir cambios de un
repositorio remoto y
combinar con una copia
local (`fetch`, `merge` y
`pull`)

Trabajo en colaboración

Ramas (`branch`)

Combinación de código
(`pull request` y
`merge`)

Tareas y tableros

Trabajo Colaborativo

Introducción al
control de
versiones y trabajo
colaborativo con
GitHub

Oscar Perpiñán
Lamigueiro

- ▶ Cuando un equipo de personas trabaja conjuntamente en un proyecto, es posible que se produzcan cambios incompatibles en un mismo fichero.
- ▶ El sistema de control de versiones **impide** cambios simultáneos en un fichero. A cambio, permite la **resolución de conflictos** y los documenta.

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

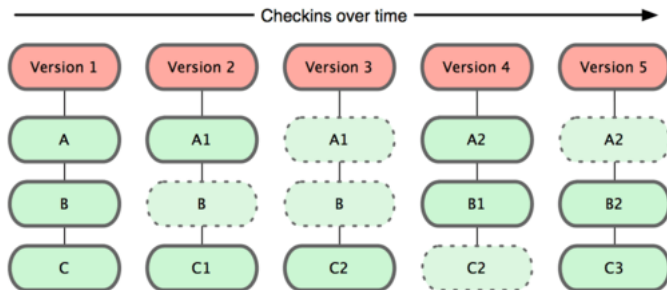
GitHub Classroom

Publicación de páginas web en GitHub

Git es un Sistema de Control de Versiones

Git es una herramienta software (accesible mediante línea de comandos con git) que implementa un Sistema de Control de Versiones.

Cada vez que se ejecuta un cambio en una estructura de ficheros controlada con Git, realiza una «foto» del estado de los archivos en ese momento, y guarda una referencia a esa instantánea. Por eficiencia, Git no almacena los archivos sin modificaciones sino un enlace al archivo anterior idéntico que ya está almacenado



Introducción al control de versiones y trabajo colaborativo con GitHub

Oscar Perpiñán
Lamigueiro

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

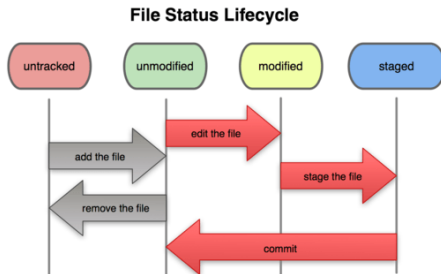
Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros

Los estados de Git

- ▶ El desarrollador incorpora uno o varios ficheros al control de versiones. (*tracked*)
- ▶ Realiza modificaciones en los ficheros (*modified*).
- ▶ Incorpora esos ficheros modificados al área de preparación (*staged*).
- ▶ Finalmente, confirma todos los cambios del área de preparación: se realiza la instantánea de los ficheros. (*committed*)



Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros

¿Qué es GitHub?

GitHub es la plataforma de alojamiento de código más importante a nivel mundial. Emplea el sistema de control de versiones `git`, y ofrece una amplia variedad de funcionalidades para el alojamiento y revisión del código, el trabajo colaborativo, y la publicación de páginas web asociadas al repositorio de código.

Introducción al
control de
versiones y trabajo
colaborativo con
GitHub

Oscar Perpiñán
Lamigueiro

Conceptos básicos

¿Qué es el control de
versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en
GitHub

Configuración del
ordenador para el uso de la
cuenta

Mi primer repositorio: crear
y clonar

Flujo de trabajo con `git` y GitHub

Realizar y confirmar
cambios (`add` y `commit`)

Publicar cambios (`push`)

Recibir cambios de un
repositorio remoto y
combinar con una copia
local (`fetch`, `merge` y
`pull`)

Trabajo en colaboración

Ramas (`branch`)

Combinación de código
(`pull request` y
`merge`)

Tareas y tableros

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

<https://help.github.com/desktop/guides/>

Conceptos básicos

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Primeros pasos

Creación de una cuenta en GitHub

Configuración del ordenador para el uso de la cuenta

Mi primer repositorio: crear y clonar

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (`add` y `commit`)

Publicar cambios (`push`)

Recibir cambios de un repositorio remoto y combinar con una copia local (`fetch`, `merge` y `pull`)

Trabajo en colaboración

Ramas (`branch`)

Combinación de código (`pull request` y `merge`)

Tareas y tableros

Conceptos básicos

Primeros pasos

- Creación de una cuenta en GitHub

- Configuración del ordenador para el uso de la cuenta

- Mi primer repositorio: crear y clonar

Flujo de trabajo con `git` y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

- Creación de una cuenta en GitHub

- Configuración del ordenador para el uso de la cuenta

- Mi primer repositorio: crear y clonar**

Flujo de trabajo con `git` y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

- Realizar y confirmar cambios (add y commit)

- Publicar cambios (push)

- Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Realizar y confirmar cambios (add y commit)

Publicar cambios (push)

Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

- Realizar y confirmar cambios (add y commit)

- Publicar cambios (push)

- Recibir cambios de un repositorio remoto y combinar con una copia local (fetch, merge y pull)

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

- Ramas (branch)

- Combinación de código (pull request y merge)

- Tareas y tableros de discusión (issues)

- Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

- Ramas (branch)

- Combinación de código (pull request y merge)

- Tareas y tableros de discusión (issues)

- Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

- Ramas (branch)

- Combinación de código (pull request y merge)

- Tareas y tableros de discusión (issues)**

- Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

- Ramas (branch)

- Combinación de código (pull request y merge)

- Tareas y tableros de discusión (issues)

- Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Primeros pasos

Flujo de trabajo con git y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub