

# Introducción al control de versiones y trabajo colaborativo con GitHub

Oscar Perpiñán Lamigueiro

Conceptos básicos

Uso de `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?

Uso de 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>



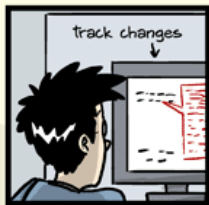
FINAL\_rev.6.COMMENTS.doc



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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

*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.<sup>1</sup>*

---

<sup>1</sup><https://git-scm.com/book/es/v1/>

# ¿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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

*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.<sup>2</sup>*

---

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



# 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?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

# ¿Qué? ¿Cuándo? ¿Quién?

Introducción al  
control de  
versiones y trabajo  
colaborativo con  
GitHub

Oscar Perpiñán  
Lamigueiro

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.

Conceptos básicos

¿Qué es el control de  
versiones?

¿Qué son Git y GitHub?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

# 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?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Uso de git y GitHub

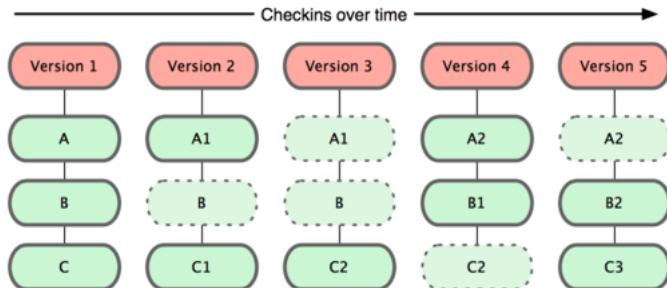
Trabajo en colaboración

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.



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?

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros de discusión (issues)

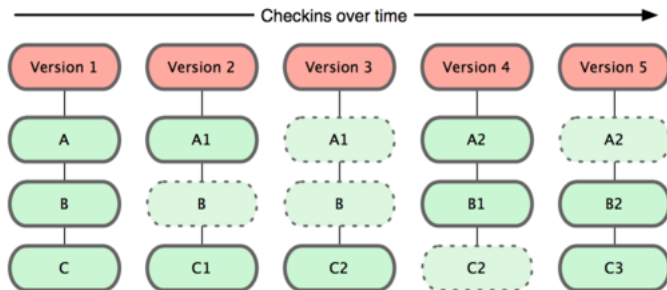
Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

# Git es 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.



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?

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros de discusión (issues)

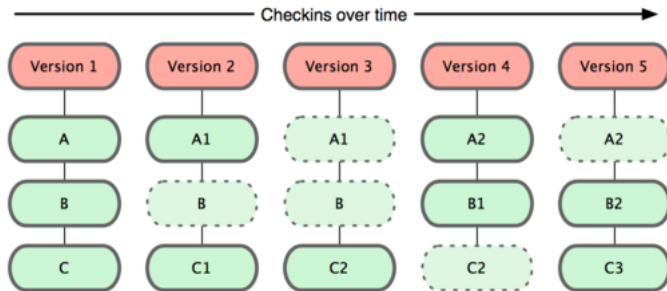
Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

# Git es un Sistema de Control de Versiones

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?

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros de discusión (issues)

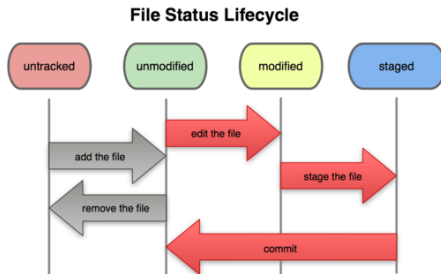
Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

# 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*)





# ¿Qué es GitHub?

- ▶ GitHub es la plataforma de alojamiento de código más importante a nivel mundial.
- ▶ Emplea el sistema de control de versiones git
- ▶ Ofrece una amplia variedad de funcionalidades
  - ▶ Alojamiento de código
  - ▶ Revisión de código
  - ▶ Trabajo colaborativo
  - ▶ Publicación de páginas web

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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

Uso de `git` y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

Conceptos básicos

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

# Creación de una cuenta en GitHub

<https://github.com/join>

Join GitHub

The best way to design, build, and ship software.

Step 1: Set up your account

Step 2: Choose your plan

Step 3: Tailor your experience

Create your personal account

Username \*

This will be your username. You can add the name of your organization later.

Email address \*

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password \*

Make sure it's more than 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more](#).

Verify account

You'll love GitHub

- Unlimited public repositories
- Unlimited private repositories
- Limitless collaboration
- Frictionless development
- Open source community

Más información en [New GitHub account](#)

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?

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

Ramas (branch)

Combinación de código (pull request y merge)

Tareas y tableros de discusión (issues)

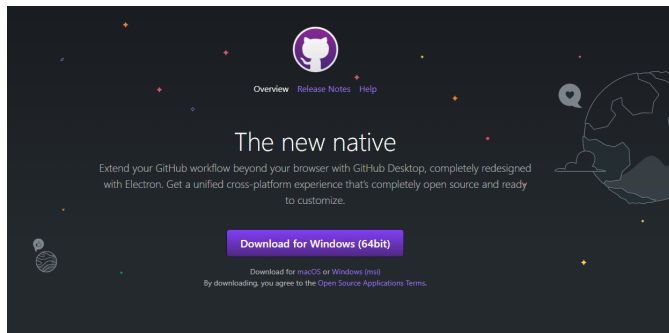
Herramientas gráficas para el análisis de un repositorio

GitHub Classroom

Publicación de páginas web en GitHub

# Instalación de GitHub Desktop

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



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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

# Conectamos Git, GitHub y GitHub Desktop

Introducción al  
control de  
versiones y trabajo  
colaborativo con  
GitHub

Oscar Perpiñán  
Lamigueiro

- Una vez instalado comienza el proceso de autenticación, usando las credenciales del paso anterior<sup>3</sup>.

*File > Options > Accounts > Sign In*

- A continuación, conectamos la información de usuario con Git<sup>4</sup>.

*File > Options > Git*

Conceptos básicos

¿Qué es el control de versiones?

¿Qué son Git y GitHub?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

---

<sup>3</sup>Más información en [Authenticating to GitHub](#).

<sup>4</sup>Más información en [Configuring Git](#).

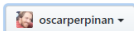
# Nuevo repositorio desde github.com

<https://github.com/new>

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name \*

Great repository names are short and memorable. Need inspiration? How about **refactored-spoon**.

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 let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## Trabajo en colaboración

Ramas (branch)

Combinación de código  
(pull request y  
merge)

Tareas y tableros de  
discusión (issues)

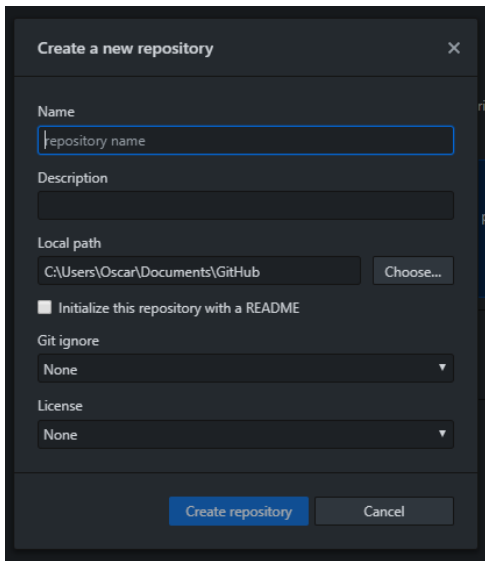
Herramientas gráficas para  
el análisis de un repositorio

## GitHub Classroom

Publicación de  
páginas web en  
GitHub

# Nuevo repositorio desde GitHub Desktop

*File > New Repository*



The screenshot shows the 'Create a new repository' dialog box in GitHub Desktop. The dialog has a dark theme and a close button (X) in the top right corner. It contains the following fields and options:

- Name:** A text input field with the placeholder text 'repository name'.
- Description:** A text input field.
- Local path:** A text input field containing 'C:\Users\Oscar\Documents\GitHub' and a 'Choose...' button to the right.
- Initialize this repository with a README:** A checkbox that is currently checked.
- Git ignore:** A dropdown menu with 'None' selected.
- License:** A dropdown menu with 'None' selected.

At the bottom of the dialog, there are two buttons: 'Create repository' (highlighted in blue) and 'Cancel'.

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?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

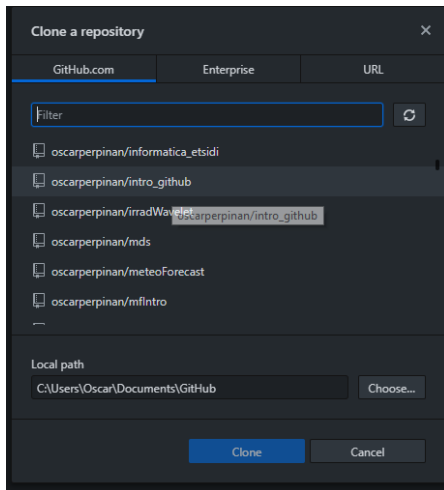
Publicación de  
páginas web en  
GitHub



# Clonar un repositorio remoto

Si hemos creado el repositorio desde github.com (*repositorio remoto*), hay que clonarlo (*copia local*).

*File > Clone Repository*



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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## Trabajo en colaboración

Ramas (branch)

Combinación de código  
(pull request y  
merge)

Tareas y tableros de  
discusión (issues)

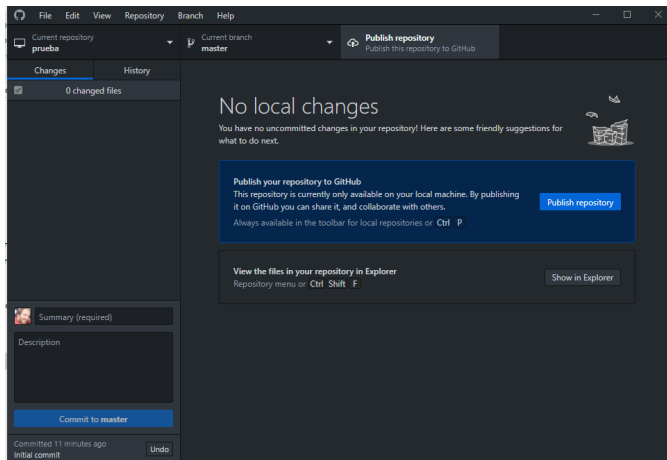
Herramientas gráficas para  
el análisis de un repositorio

## GitHub Classroom

Publicación de  
páginas web en  
GitHub

# Publicar un repositorio local

Si hemos creado el repositorio desde GitHub Desktop (*repositorio local*), hay que publicarlo en github.com (*remoto*)



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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

# Consejos básicos

- ▶ Elige bien el `.gitignore` (adecuado al proyecto). Véase <https://github.com/github/gitignore>.
- ▶ No olvides cumplimentar el `README.md`. Para el formato véase [Formatting syntax](#).
- ▶ Elige una licencia adecuada a tu proyecto y a tus intereses actuales y futuros. Véase <https://choosealicense.com>.

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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub

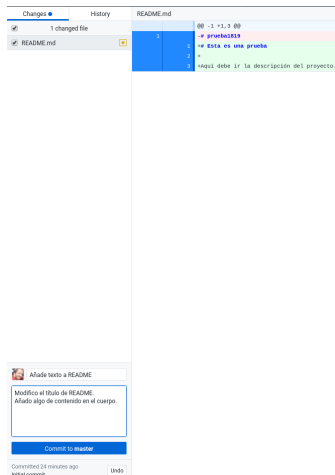
# Realizar y confirmar cambios (add y commit)

- ▶ Modifica los ficheros de la copia local.
- ▶ Añade los cambios realizados a la siguiente «instantánea» del repositorio:

```
git add
```

- ▶ Confirma los cambios (escribiendo un resumen de lo realizado):

```
git commit
```



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?

Uso de git y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

# Publicar cambios (push)

Para sincronizar los cambios realizados en la copia local con el repositorio remoto hay que publicar mediante `git push`.

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?

Uso de `git` y  
GitHub

Primeros Pasos

Flujo de Trabajo

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

# Recibir cambios de un repositorio remoto

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?

## Uso de git y GitHub

Primeros Pasos

Flujo de Trabajo

## 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

Para obtener los cambios recientes que existan en el repositorio y no en la copia local hay que emplear `git pull`, que es la combinación de la secuencia:

1. `git fetch`, obtener datos recientes del repositorio.
2. `git merge`, combinarlos con la copia local.

Conceptos básicos

Uso de `git` y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub



Conceptos básicos

Uso de 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

Uso de 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

Uso de 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

Uso de 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

Uso de `git` y GitHub

Trabajo en colaboración

**GitHub Classroom**

Publicación de páginas web en GitHub

Conceptos básicos

Uso de `git` y GitHub

Trabajo en colaboración

GitHub Classroom

Publicación de páginas web en GitHub