

ACÀMICA

TEMA DEL DÍA

GIT

Hoy trabajaremos en el proyecto 03. Verás cómo guardar tus avances en un repositorio en GitHub.



Agenda

Daily

Explicación: GitHub

Break

Workshop Proyecto 03

Cierre



Daily



Daily



Sincronizando...

Bitácora



¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

Challenge



¿Cómo te ha ido?
¿Obstáculos?
¿Cómo seguimos?

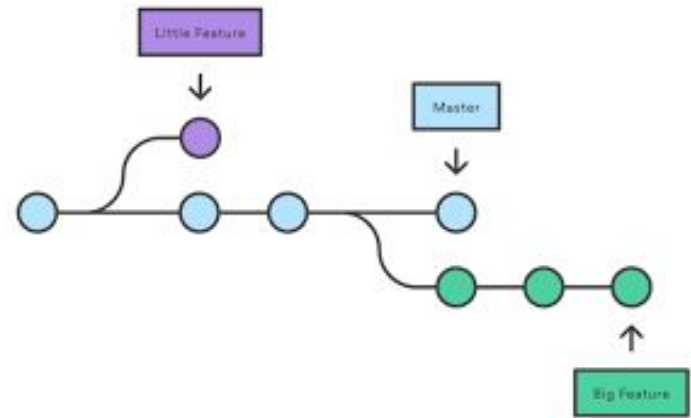
Repaso de la bitácora



Git

Es un sistema de control de versiones que te permite registrar, en una línea de tiempo, el estado de la estructura de directorios y archivos en un momento determinado.

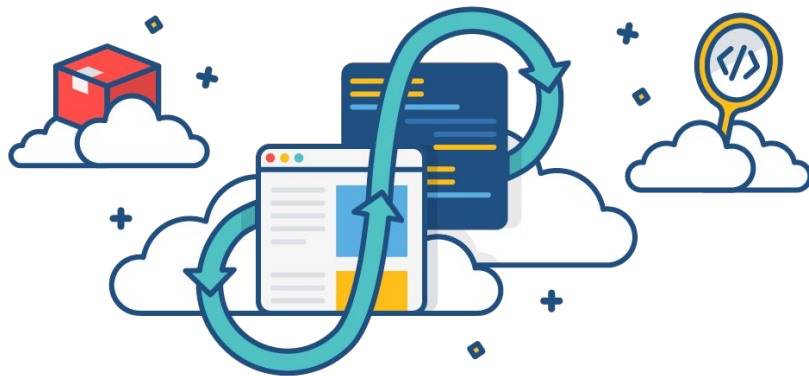
Es una herramienta muy útil para poder “viajar en el tiempo” si algo sale mal.



Github

Es el sistema web que nos permite mantener una copia remota, o repositorio, de nuestra estructura de directorios y archivos.

De esta forma, podemos compartir el código con las distintas personas que están involucradas en nuestro desarrollo.



Github





¿Qué es **GitHub**?

Una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de **control de versiones Git**. Se utiliza principalmente para la creación de **código fuente** de programas de ordenador.



¿Qué es Github?



El software que opera GitHub fue escrito en **Ruby on Rails**. Desde enero de **2010**, GitHub opera bajo el nombre de *GitHub, Inc.* Anteriormente era conocida como *Logical Awesome LLC*. El código de los proyectos alojados en GitHub se almacena típicamente de **forma pública**, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados. El 4 de junio de 2018, **Microsoft** compró GitHub por la cantidad de 7.500 millones de dólares.



¿Has instalado Git?



Si aún no lo hiciste:

1. Crea una cuenta en GitHub.
2. Instala Git (busca cómo hacerlo en tu sistema operativo).

Tu proyecto en Git/GitHub



Hay dos formas de trabajar en Git/Github:

1. A partir de un proyecto/repositorio ya existente.
 - a. Fork
 - b. Clone
2. Arrancando desde cero nuestro propio proyecto/repositorio.

Hay dos formas de trabajar en Git/Github:

1. A partir de un proyecto/repositorio ya existente.
 - a. Fork
 - b. Clone
2. Arrancando desde cero nuestro propio proyecto/repositorio.



Empezaremos así

(¡pero hay muchos pasos en común entre las dos formas!)

Para empezar desde cero





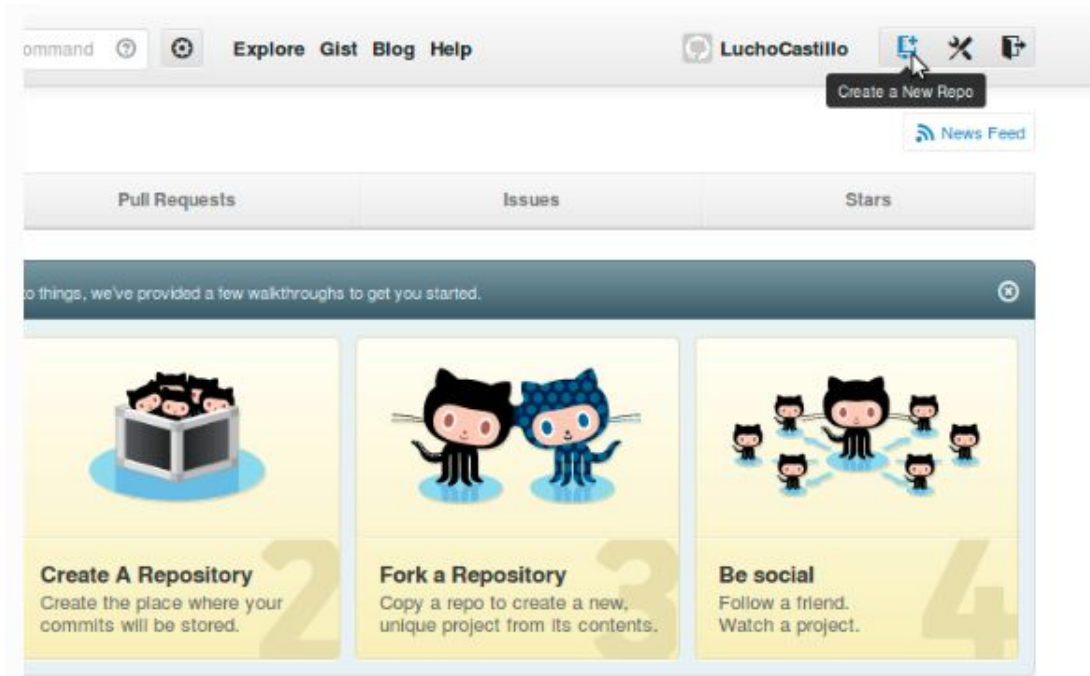
1. Crea el repositorio en GitHub.
2. Vincula el repositorio con la carpeta - en tu computadora - donde tengas el proyecto.

Crear un repositorio



Crear un repositorio

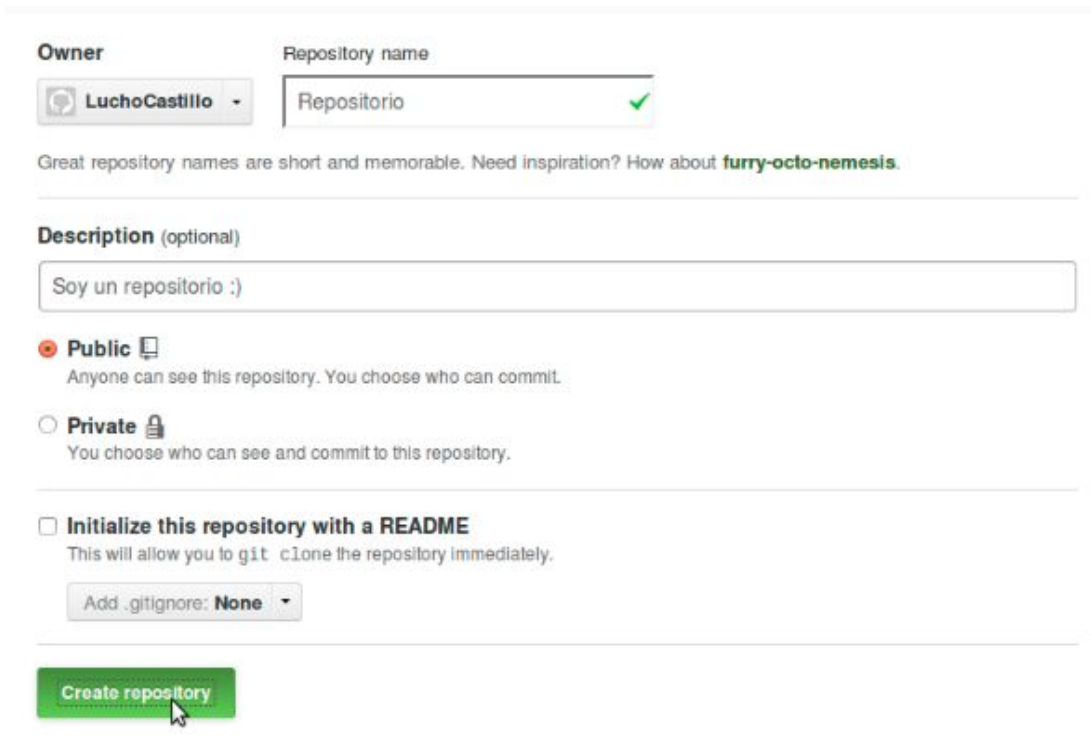
1. Entra a GitHub con tu cuenta y selecciona el botón "Create a New Repo", de la barra de herramientas, habiendo entrado a GitHub con tu cuenta.



Crear un repositorio

2. Llena dos datos:

- Nombre del repositorio
- Descripción del repositorio (opcional)




The screenshot shows the GitHub 'Create repository' form. At the top, there are two fields: 'Owner' and 'Repository name'. The 'Owner' field is a dropdown menu with 'LuchoCastillo' selected. The 'Repository name' field is a text input with 'Repositorio' entered and a green checkmark on the right. Below these fields is a hint: 'Great repository names are short and memorable. Need inspiration? How about [furry-octo-nemesis](#).' The next section is 'Description (optional)', which has a text input field containing 'Soy un repositorio :)'. Below the description field are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option has a lock icon and the text 'Anyone can see this repository. You choose who can commit.' The 'Private' option has a lock icon and the text 'You choose who can see and commit to this repository.' Below these options is a checkbox labeled 'Initialize this repository with a README', with the text 'This will allow you to `git clone` the repository immediately.' Below the checkbox is a dropdown menu for 'Add .gitignore: None'. At the bottom of the form is a green button labeled 'Create repository' with a mouse cursor pointing at it.


Owner: LuchoCastillo

Repository name: Repositorio

Great repository names are short and memorable. Need inspiration? How about [furry-octo-nemesis](#).

Description (optional): Soy un repositorio :)

☒ **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.

Add .gitignore: None

Create repository

Crear un repositorio

2. Llena dos datos:

- Nombre del repositorio
- Descripción del repositorio (opcional)

¡Listo!
Repositorio
creado, ahora
podrás verlo
en tu perfil.

Owner: **LuchoCastillo** Repository name: **Repositorio** ✓

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

Description (optional)

Soy un repositorio :)

☒ **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.

Add .gitignore: **None**

Create repository

Crear un proyecto



Crear un proyecto

El repositorio recién creado te llevará inmediatamente al proyecto. Como tu proyecto aún no tiene nada en su interior, no verás más que una ayuda para subir archivos y proyectos.

1. Para crear un proyecto desde cero, crea los archivos y luego súbelos a la página.
2. En el primer recuadro de la ayuda, verás una serie de comandos para el terminal. Antes de seguirlos, tendrás que instalar git:

```
touch README.md
git init
git add README.md
git commit -m "comentario"
git remote add origin https://github.com/LuchoCastillo/Repositorio.git
git push -u origin master
```

En linux:

```
sudo apt-get install git
```


Crear un proyecto

El repositorio recién creado te llevará inmediatamente al proyecto. Como tu proyecto aún no tiene nada en su interior, no verás más que una ayuda para subir archivos y proyectos.

1. Para crear un proyecto desde cero, crea los archivos y luego súbelos a la página.
2. En el primer recuadro de la ayuda, verás una serie de comandos para el terminal. Antes de seguirlos, tendrás que instalar git

```
touch README.md
git init
git add README.md
git commit -m "comentario"
git remote add origin https://github.com/LuchoCastillo/Repositorio.git
git push -u origin master
```

```
sudo apt-get install git
```

**¡Entra
nuevamente al
repositorio en
GitHub y fíjate si
cambió algo!**

Subir un proyecto



Subir un proyecto

Para subir un proyecto ya realizado a GitHub, sigue exactamente los mismos pasos que para iniciar un proyecto nuevo: sube los archivos creados y luego modifícalos en el editor de la página.

Para subir un archivo hay que ubicarse en la carpeta del repositorio y seguir estos pasos:

```
git add archivo  
git commit -m "comentario"  
git push
```

- **archivo:** nombre del archivo que quieres subir
- **comentario (commit):** reemplazar por una descripción del archivo.
- **El push:** puedes hacerlo luego de haber subido varios archivos.

Subir un proyecto

Para subir un proyecto ya realizado a GitHub, sigue exactamente los mismos pasos que para iniciar un proyecto nuevo: sube los archivos creados y luego modifícalos en el editor de la página.

Para subir un archivo hay que ubicarse en la carpeta del repositorio y seguir estos pasos:

```
git add archivo  
git commit -m "comentario"  
git push
```

- **archivo:** nombre del archivo que quieres subir
- **comentario (commit):** reemplazar por una descripción del archivo
- **El push:** puedes hacerlo luego de haber subido varios archivos.

Si no se realiza un comentario, ¡no se agrega el cambio!

¿Cómo seguir?



¿Cómo seguir?

- Cada vez que modifiques un/os archivo/s, y quieras guardar los cambios en el repositorio, deberás seguir la secuencia *add* → *commit* → *push*. Intenta que el commit sea breve pero descriptivo del cambio. Para ello, practica *commitear* luego de cada cambio significativo en tu código.
- Cuando uno trabaja con un equipo, es común arrancar la jornada de trabajo con un *pull*. De esta forma, nuestra versión del repositorio (rama en la que estemos) se actualizará a la más reciente.

Puede ser útil hacer la siguiente configuración

```
git config --global user.name "TU_NOMBRE_EN_GITHUB"  
git config --global user.email "MAIL_EN_GITHUB"  
git config --global -l
```

Algunos comentarios sobre Git



- **¿Todo esto te parece sumamente confuso?** ¡Seguirá pareciéndolo por un buen tiempo! A la larga, uno/a se acostumbra (aunque no estamos seguros/as de haber entendido...)
- **¿Vas a tener que aplicar todo esto para trabajar en el proyecto?** ¡No! En general, cuando uno/a trabaja solo/a - sin un equipo - con GitHub, simplemente sirve para tener un historial de versiones y un lugar donde guardar tu progreso. Eventualmente, deshacer algún error que hayamos cometido. Esto simplifica mucho el uso de esta herramienta.
- **¿Vas a tener que aplicar todo esto en tu carrera profesional?** Depende de dónde y cómo trabajes, cuál sea la filosofía del lugar de trabajo, etc. Lo más probable es que tengas que usar algo o más de lo que vimos.
- **Existen otros lugares donde guardar tu repositorio, aparte de GitHub.** Por ejemplo, [Bitbucket](#). GitHub es solamente el más conocido (al punto de que muchas personas confunden Git con GitHub).

¡Advertencias!



- **Es común encontrar tutoriales que pueden parecer contradictorios.**

Esto es porque no hay una única forma de hacer las cosas en Git.
¡Elige uno y síguelo!

- **Git y Notebooks de Jupyter tienen algunos choques no del todo resueltos.**

No te sorprendas si Git cree que modificaste un notebook, por más que no lo hayas hecho. A veces basta abrir un notebook en distintos ambientes para que git se confunda. Esto no suele ser un problema trabajando solo/a.

Workshop Proyecto 03



El Proyecto 3 de tu carrera tiene como objetivo que apliques las herramientas aprendidas dentro de tres dominios específicos sobre los que aprenderás en el Bloque 3:

- Sistemas de Recomendación
- Procesamiento de lenguaje natural
- Series de tiempo

- Si todavía no lo hiciste, elige una de las tres propuestas. Arma un grupo (tres personas) con compañeros/as que están en la misma situación. Vean las consignas, exploren los notebooks y datasets. Pueden hacer uno cada uno/a y hacer una puesta en común.
- Si ya elegiste una aplicación, júntate con compañeros/as que hayan elegido la misma aplicación (máximo tres en total) y **¡comiencen a trabajar!**
- **Crea un repositorio en GitHub y sube los avances de tu proyecto.**

Recursos



Recursos

- [GitHub](#) - Pequeño tutorial en español sobre el uso de GitHub.
- [Why Git And How To Use Git As A Data Scientist](#) - Artículo sobre Git y su uso en Data Science
- <https://www.atlassian.com/git/tutorials> - Colección sumamente completa de tutoriales de Git. Recomendamos tener a mano.



Para la próxima

- Avanza con el Proyecto
- Lee la bitácora 44 y carga las dudas que tengas al Trello.

En el encuentro que viene uno/a de ustedes será seleccionado/a para mostrar cómo resolvió el challenge de la bitácora. De esta manera, ¡aprendemos todos/as de (y con) todas/as, así que vengan preparados/as.

ACÀMICA