

Tópicos Especiales II
Facultad de Ingeniería de Sistemas Computacionales
Grupo VII-342
2do Semestre 2018

Practica #5 – Git y GitHub

Git es un sistema de control de versiones distribuida que se origina a partir del desarrollo del kernel de Linux y es usado por muchos proyectos populares Open Source. Como vimos en clase, la principal diferencia entre Git y cualquier otro sistema de control de versiones es cómo Git modela sus datos. Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que confirmas un cambio, o guardas el estado de tu proyecto en Git, él básicamente hace una foto del estado de todos tus archivos en ese momento, y guarda una referencia a esa instantánea. Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo sino un enlace al archivo anterior idéntico que ya tiene almacenado.

1. Introducción a Git

En el caso que no tengamos acceso super usuario o no tengamos una computadora podemos inicialmente trabajar con el tutorial de introduccion a github en 25 pasos que se encuentra en esta dirección: <https://try.github.io/levels/1/challenges/1>

2. Creacion de un repositorio desde su propia cuenta de GitHub

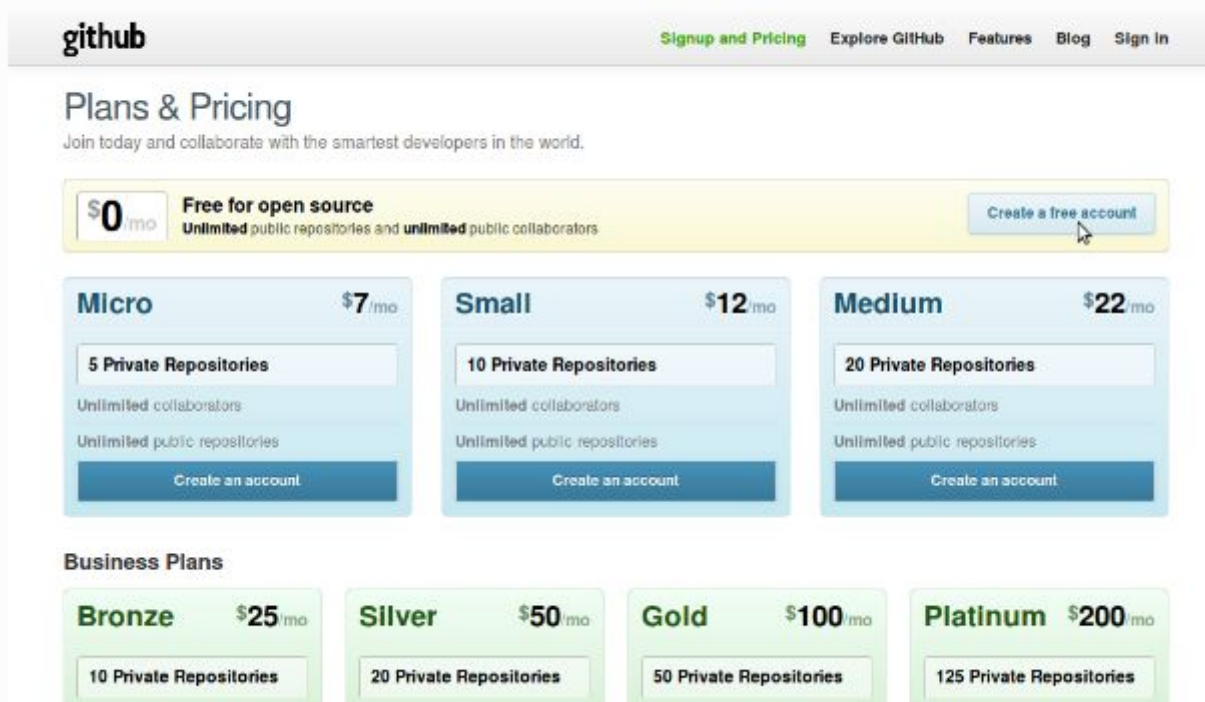
Para la creacion de un repositorio vamos a seguir los pasos que nos indican en este manual de acceso publico. <https://conociendogithub.readthedocs.io/en/latest/data/dinamica-de-uso/>



2. Hacer clic en *Signup and Pricing*, de la barra de herramientas de la página.



3. Ahora en *Create a free account*, para una cuenta gratuita.



4. Finalmente, solo habrá que llenar un pequeño formulario con los siguientes datos:

- Nombre de usuario.
- Dirección email.
- Contraseña.
- Confirmar contraseña.

Create your free personal account

Username

Email Address

We promise we won't share your email with anyone.

Password

Must contain one lowercase letter, one number, and be at least 7 characters long.

Confirm Password

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

Manejo de repositorios

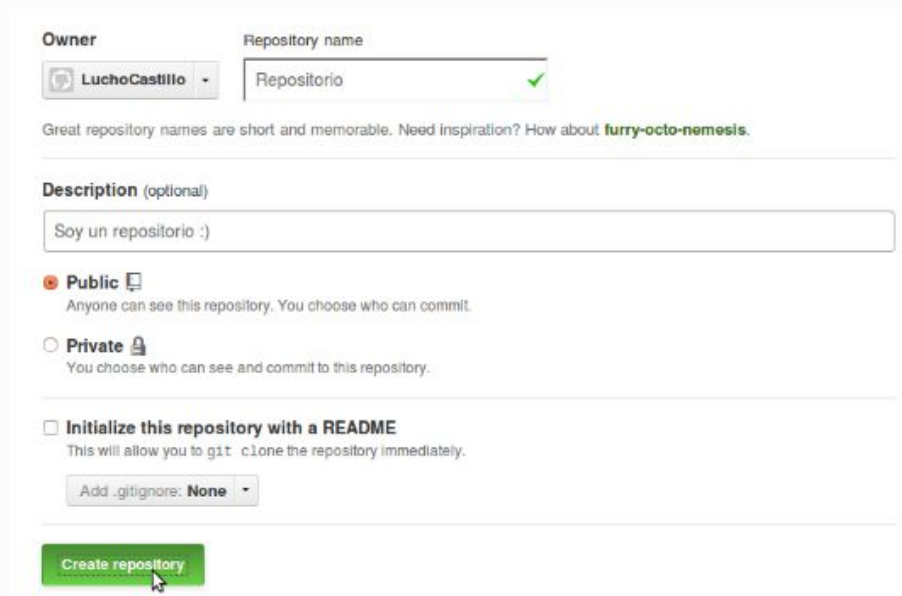
¿Cómo crear un repositorio?

Para crear un repositorio en GitHub, solo hay que seleccionar el botón "Create a New Repo", de la barra de herramientas, habiendo entrado a [GitHub](#) con tu cuenta:

The screenshot shows the GitHub web interface. At the top, there's a navigation bar with links for 'Explore', 'Gist', 'Blog', and 'Help'. The user's profile 'LuchoCastillo' is visible, and a dropdown menu is open showing the 'Create a New Repo' button. Below the navigation bar, there are tabs for 'Pull Requests', 'Issues', and 'Stars'. The main content area features a 'Getting started' section with three cards: 'Create A Repository' (with a box icon), 'Fork a Repository' (with two cat icons), and 'Be social' (with a group of cat icons). At the bottom, there's a 'Broadcast' section with a green card titled 'GitHub for Mac: Username Autocompletion'.

Ahora habrá que llenar dos datos:


1. Nombre del repositorio
2. Descripción del repositorio (opcional)




Owner LuchoCastillo **Repository name** Repositorio ✓

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

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.

Add .gitignore: **None**

Create repository

¡Listo! Repositorio creado, ahora lo vas a poder ver en tu perfil.

Crear un proyecto

Al crear el repositorio, inmediatamente nos va a llevar a él. Como nuestro proyecto no tiene nada en su interior, no nos va a mostrar más que una ayuda para subir archivos y proyectos.

Para crear un proyecto desde cero, habrá que comenzar creando los archivos del mismo y luego subiéndolos a la página.

En el primer recuadro de la ayuda, verás una serie de comandos para el terminal.

```
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
```

Subir proyecto

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

En este caso, los archivos junto con su contenido, ya están hechos. Solo hay que subirlos y editarlos si surge algún inconveniente.

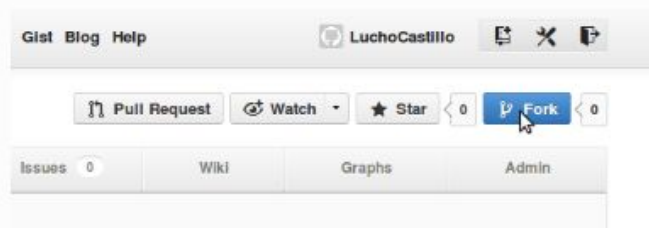
Para subir un archivo hay que ubicarse en la carpeta del repositorio y seguir estos pasos:

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

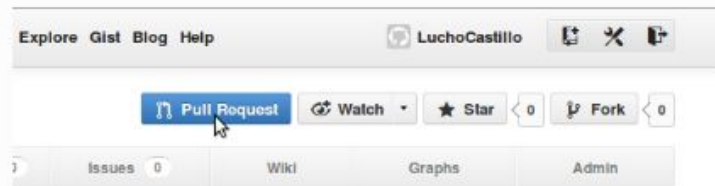
Es importante realizar los 3, ya que si no se ingresa un comentario, no se realiza el cambio.

Colaborar en un proyecto ajeno

Para colaborar en un proyecto ajeno simplemente basta con buscarlo dentro de los repositorios, y luego presionar el botón *fork*. Esto genera automáticamente una copia del mismo en tu perfil.



Al terminar tus modificaciones podrás presionar *Pull Request* para enviárselo al creador del mismo.



Verifique y haga forks a su propio GitHub de algunos de los proyectos con mas actividad¹ como lo son:

- Libreria NodeJS (<https://github.com/nodejs/node>)
- Libreria D3 (<https://github.com/d3/d3>)

3. Instalacion de Git

De todos modos la manera mas completa de trabajar con Git seria bajarlo de la direccion <https://git-scm.com/> e instalarlo en sus computadoras.

Podemos seguir los pasos del tutorial que se encuentra en esta dirección <https://rogerdudler.github.io/git-guide/index.es.html>

¹<https://www.computerworld.com/article/2473298/application-development/120001-GitHub-s-top-10-rock-star-projects.html>

1. Inicie su repositorio con:
git init
2. Cree algunos archivos para su repositorio local y añadalos a su repositorio con:
echo "Algun Comentario" >> README.md
git add README.md
3. Realice un commit
git commit -m "comentario en mi repositorio"
4. Utilizando la cuenta de GitHub que creo usted puede:

Enviar su repositorio local a un repositorio publico (propio)
git push -u origin master

Mande sus cambios locales al repositorio remoto (propio)
git remote add origin <direccion> (por ejemplo <https://github.com/try-git/try-git.git>)

Usted podria tambien trabajar con un repositorio existen como se vio en el punto 2 con los siguientes comando

1. Crea una copia local del repositorio 1IL251 (Dispuesto para este laboratorio) ejecutando:
git clone <https://github.com/jsgalan/1IL251.git>
2. Para actualizar tu repositorio local al commit más nuevo, ejecuta
git pull
3. Puede hacer que los cambios en su repositorio local suban a la rama principal
git push
4. Podira funcionar con alguna rama activa o el branch principal con”
git merge <branch>

Referencias

Git - Manual de usuario (Version 1)
Cooperativa de Trabajo Fiquis LTDA
<http://blog.fiquis.webfactional.com/wp-content/uploads/2014/05/git-manual.pdf>

Aprendiendo a usar GitHub
<https://conociendogithub.readthedocs.io/en/latest/data/dinamica-de-uso/>

Git - la guía sencilla
<https://rogerdudler.github.io/git-guide/index.es.html>