

Manejo de repositorios en Github

Taller de desarrollo de habilidades de investigación 2023

Andrés Vallone ¹

¹Escuela de Ciencias Empresariales



UCN CIENCIAS
EMPRESARIALES



Introducción

Git y GitHub

Usando GitHub

Introducción



UCN CIENCIAS
EMPRESARIALES

Introducción


- El incremento en el desarrollo colectivo de software
- La necesidad de tener estabilidad en los proyectos de investigación
- El aumento en la transparencia en las publicaciones, “la crisis de la replicabilidad”
- El uso de repositorios nos facilita poder cumplir con estos desafíos

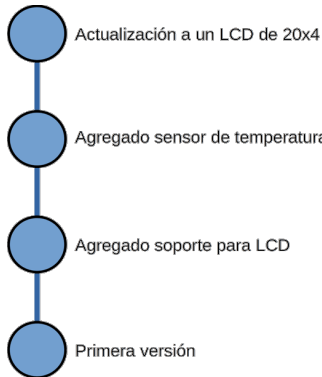
- Se espera que al final de la sesión usted sea capaz de:
 - crear un repositorio en la en GitHub
 - clonar un repositorio
 - subir y bajar información de un repositorio

Git y GitHub



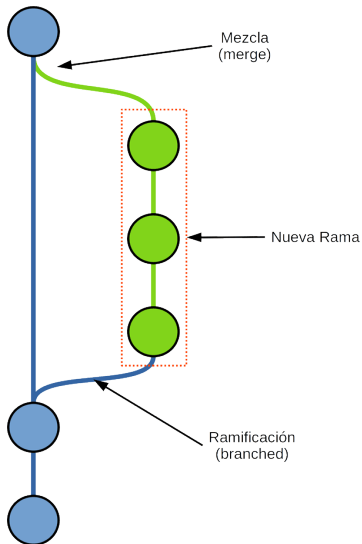
UCN CIENCIAS
EMPRESARIALES

- Git es un software de control de versiones distribuido, optimiza el trabajo en proyectos. 
- Un software de control de versiones: registra cambios realizados en un archivo o conjunto de archivos en tiempo, entonces se puede recuperar versiones específicas.
- Los proyectos se almacenan en repositorios que contiene los archivos del proyecto junto a todo el historial de cambios.
- Es distribuido, por lo que cada desarrollador tiene una clon local del repositorio principal



- Git almacena cada vez que se realiza y confirma (commit) algún cambio
- Cada uno de los nodos representa un cambio confirmado (commit).
- A una sucesión de commits se le conoce como rama (branch)
- La rama principal del proyecto es la rama master
- La master es la rama estable, por tanto se trata de no modificarla.
- Para hacer cambios sin modificar la master, es posible crear una nueva rama.

¿Qué es Git?



- Cada uno de colaboradores puede tener una rama propia y modificar el proyecto.
- Cada una de las ramas tendrá su propia historia.
- Los cambios en las ramas son independientes de la master.
- Para incorporar los cambios al master se debe solicitar un merge

- Es una plataforma de desarrollo colaborativo para alojar y gestionar proyectos utilizando el sistema de control de versiones Git.
- Se utiliza principalmente para la creación de código fuente (código simple o un programa completo)
- GitHub ofrece rasgos similares a los de una red social: Se puede seguir un desarrollador, un proyecto, un tipo de programación
- La plataforma permite que cualquier persona sea capaz visualizar el código y colaborar con su desarrollo, es decir, máxima expresión del “open source”



- Es eficiente para
 - Alojar base de datos de papers y los códigos de ejecución
 - Utilizar un sistema de control de cambio en archivos de binarios (L^AT_EX, html, Markdown, R, python,...)
 - Trabajar en simultaneo con otros investigadores en distintas partes de un código
 - Desarrollar software colaborativamente (Paquetes en R, librerías de python, dashboard,...)
- Se puede, pero no es lo ideal
 - Alojar archivos para difusión.
 - Usar como control de versión de archivos no binarios.
 - Usar como backup de archivos.



Desktop

- Git se utiliza mediante línea de comandos en el terminal o consola
- GitHub es Git, por tanto, las líneas de comando funcionan, pero se puede utilizar mediante un web browser
- GitHub Desktop es una aplicación que te habilita para interactuar con GitHub utilizando una GUI en vez de la línea de comandos o un web browser.
- Es un facilitador de uso de repositorios, es decir tenemos las bondades del Git sin usar comandos.

Usando GitHub



UCN CIENCIAS
EMPRESARIALES

1. Cree su cuenta en GitHub accediendo aquí
2. Descargue e instale Git desde aquí
3. Descargue e instale GitHub Desktop desde aquí

Existen un conjunto de acciones básicas que se deben conocer al trabajar con repositorios:

- **pull:** Obtiene los archivos del repositorio remoto y los combina con el local.
- **commit:** Crea una entrada en el índice de historiales, establece la versión.
- **push:** Envía todos los objetos modificados localmente al repositorio remoto
- **clone:** Crea una copia de repositorio GIT de una fuente externa
- **fork:** Crea una bifurcación o rama nueva.
- **merge:** Fusiona una o más ramas con otra rama activa y crea automáticamente un nuevo commit si no hay conflictos

- Crear un repositorio usando web browser
- Crear desde GitHub Desktop
- Clonar el repositorio
- Modificar un archivo en su repositorio y hacer un commit
- Hacer un push
- Modificar el README.md de un repositorio y hacer un pull