

Capacitación LIFIA

TP Integrador Informe Github | Avola - Burgos

Qué es GIT

Git es un sistema libre de código abierto para control de versiones, es decir la gestión de cambios producidos por uno o múltiples usuarios a documentos, programas, sitios y otros grandes conjuntos de información.

Qué es GitHub

GitHub es solo una de las plataformas o sitios web basados en GIT y desarrollados para la alocaación online de repositorios: es decir, proyectos o bien el directorio en el que será mantenido el proyecto. Otros sitios similares son GitLab o Bitbucket.

Cómo operar en GitHub

Para empezar solo debe crearse una cuenta de manera gratuita con usuario, nombre y contraseña.

Operaciones que pueden hacerse en Github

Algunos comandos GIT como los que se detallan a continuación son representativos de las funcionalidades ofrecidas por GitHub:

- **status:** permite saber qué cambios se produjeron dentro del proyecto
- **add:** agrega de forma provisoria todos o algunos de los archivos que hayan sufrido cambios al repositorio local
- **commit:** guarda los cambios del proyecto en el repositorio local
- **push:** sube los "commits" a un repositorio remoto, en este caso el que se haya creado en GitHub
- **pull:** lo opuesto a push. descarga cambios del repositorio remoto a la máquina local
- **clone:** descarga la máquina local un repositorio que está ubicado en otra parte

Validaciones

Para poder elevar cambios al repositorio remoto, es decir alocado en GitHub, es preciso autenticarse ante GitHub conectando la máquina local con la cuenta personal de GitHub mediante claves SSH. Los pasos a seguir son:

- Generar una clave SSH localmente mediante el comando:
`~ssh-keygen -t rsa -b 4096 -C "mailejemplo@ejemplo.com"`
- El email provisto debe ser aquel con el que registramos la cuenta en GitHub
- En la ruta Users/usuario/.ssh se generarán ubicará el archivo que contendrá la clave
- Se generarán dos claves: testkey y testkey.pub.
- Esta última es la que debe compartirse con GitHub
- En los settings de Github, en la sección SSH and GPG keys se debe crear una nueva clave en cuya entrada debe pegarse el cuerpo de testkey.pub
- Por último debe la terminal GIT local debe estar al tanto de la nueva clave SSH

Comenzando a operar en GitHub

El repositorio puede comenzar a desarrollarse desde el propio editor de GitHub o bien puede hacerse un push del repositorio local mediante el comando: `git push origin master` donde `origin` es el comando que representa a nuestro repositorio y `master` la rama de nuestra cuenta donde será subido el repositorio. Si el repositorio ya contara con commits previos, `git push origin master` generará un nuevo commit al mismo.

Otra opción es iniciar un repositorio GIT localmente simplemente creando un nuevo directorio (manualmente, desde consola o desde un editor de código), ubicándose dentro del mismo mediante consola o editor de código y ejecutando el comando `git init`. Crear un repositorio GIT implicará que dentro del directorio escogido se ubique un directorio escondido `.git` en el que se llevará registro de los distintos cambios realizados en el directorio creado, dirección del repositorio remoto, etc. Para poder pushear el repositorio local recientemente creado (y nunca asociado a una cuenta de GitHub) podrá crearse un nuevo repositorio desde GitHub y ejecutar en consola el comando `git remote add origin git@github.com:nombreUsuario/nombreRepositorio` y `git remote -v`. Por último sí, ejecutar el comando `git push origin master`.

Branching

El concepto puede pensarse fácilmente como la posibilidad de ramificar un proyecto para que en distintas ramas del mismo (líneas de trabajo) puedan implementarse y publicarse distintos cambios los cuales son solo conocidos por la rama en la que dichos cambios (commits) fueron hechos. Este servicio es especialmente útil para poder probar desarrollos alternativos sobre una versión distinta a la existente en la rama principal (master branch).

Comandos

- `git checkout -b nombreBranch` para crear un nuevo branch (esto ejecuta y nos posiciona dentro del nuevo branch)
- `git checkout otroBranch` nos saca del branch actual para pasar a otroBranch
- `git diff otroBranch` compara el branch actual con otroBranch
- `git merge master` combina los cambios del branch actual al master branch
- `git reset` deshace un add
- `git reset HEAD~1` deshace el último commit hecho
- `git log` muestra un registro histórico de commits

Forking

Permite hacer un fork o bifurcación de un repositorio existente y ajeno. Esto permite ubicar un repositorio de un tercero bajo nuestra cuenta de modo que podamos implementar cambios al mismo.

Pull request

Solicita a quienes administran el repositorio al cual se hizo fork que levanten los cambios que produjimos en el repositorio “forkeado”.

Algunas buenas prácticas

- Crear un archivo readme que dé cuenta del propósito del proyecto
- Hacer comentarios descriptivos para cada commit hecho
- Eliminar del repositorio (tanto local como remoto) aquellos branches que queden obsoletos
- Otras?