

# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

## PROGRAMACIÓN 1



Profesor Coordinador: Alberto Cortez

### Comisión 1



Profesor: Prof. Cinthia Rigoni



Tutor: Prof. Martín A. García

## Práctico N° 2: Git y GitHub

Estudiante: Hernán E. Bula

### Actividades

#### 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

##### •¿Qué es *GitHub*?

Es un repositorio para alojar proyectos de código fuente de programas, utilizando el sistema de control de versiones Git.

##### •¿Cómo crear un repositorio en *GitHub*?

Comando de GIT -> `git init`

GitHub -> Mediante la interfaz de usuario web, ir a la esquina superior derecha

Nuevo repositorio -> completar nombre del repositorio, descripción, visibilidad -> Click “Crear repositorio”.

##### •¿Cómo crear una rama en *Git*?

`git branch <nombre_rama>`

##### •¿Cómo cambiar a una rama en *Git*?

`git checkout <nombre_rama>`

##### •¿Cómo fusionar ramas en *Git*?

`git merge <nombre_rama>`

##### •¿Cómo crear un commit en *Git*?

`git commit -m “Mensaje descriptivo”`

##### •¿Cómo enviar un commit a *GitHub*?

```
git init
git add .
git commit "Mensaje del commit"
git push origin master
```

• **¿Qué es un repositorio remoto?**

Un repositorio remoto es una copia de un proyecto que se encuentra alojada en un servidor remoto o internet. Esto permite compartir entre varios miembros de un equipo o con la comunidad.

• **¿Cómo agregar un repositorio remoto a Git?**

```
git remote add origin URL
```

• **¿Cómo empujar cambios a un repositorio remoto?**

```
git push origin master
```

• **¿Cómo tirar de cambios de un repositorio remoto?**

```
git pull origin master
```

• **¿Qué es un fork de repositorio?**

Fork es una copia de un repositorio que utilizo para cambiar, colaborar o mejorar ese proyecto sin modificar el original. Después puedo remitirle al propietario original mis cambios y mejoras y si le gustan o parecen buenas, puede introducirlas en su código original.

• **¿Cómo crear un fork de un repositorio?**

Para realizar un fork un proyecto en GitHub, hay que visitar la URL del proyecto y pulsar el botón "Fork" del lado superior derecho de la página. Eso abrirá una página nueva de proyecto, en tu cuenta y con tu propia copia del código fuente.

• **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**

Un pull request (PR) en GitHub es una solicitud para revisar y aprobar cambios en un repositorio antes de fusionarlos. Es una de las formas más habituales de colaborar en proyectos de código abierto.

Para hacer un PR, primero debes:

1. Forkear el repositorio
2. Clonar el repositorio en tu computadora
3. Crear una rama nueva
4. Realizar cambios y confirmarlos
5. Enviar los cambios a GitHub

Después de esto ya estás en condiciones de realizar un PR:

1. Ir a la página principal del repositorio
2. Seleccionar la rama que contiene tus confirmaciones
3. Hacer clic en "Comparar y solicitud de incorporación de cambios"
4. Seleccionar la rama en la que quieres combinar los cambios
5. Escribir un título y una descripción
6. Hacer clic en "Crear solicitud de incorporación de cambios"
7. Una vez creado el PR, los revisores podrán examinar los cambios. Si se solicitan cambios, puedes volver a la rama local y hacer las modificaciones necesarias.
8. Si tu PR es aceptado, recibirás un correo electrónico.

Fuente:

<https://www.freecodecamp.org/espanol/news/como-hacer-tu-primer-pull-request-en-github/>

### •¿Cómo aceptar una solicitud de extracción?

En GitHub, puedes aprobar una solicitud de extracción revisando los cambios propuestos y escribiendo un comentario que resuma tu opinión. Finalmente, haz clic en Aprobar.

1. Haz clic en Revisar cambios sobre el código modificado.
2. Escribe un comentario que resuma tu opinión sobre los cambios propuestos.
3. Haz clic en Aprobar para aprobar la fusión de los cambios propuestos.
4. Haz clic en Enviar revisión.

Fuente:

<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/approving-a-pull-request-with-required-reviews>

### •¿Qué es un etiqueta en Git?

En Git, una etiqueta es una marca que se aplica a una confirmación para identificarla y darle información relevante. Las etiquetas permiten marcar un punto específico en el historial de un repositorio, por ejemplo, para definir números de versión. Puedes acompañar los tags con un mensaje.

Usos de las etiquetas:

- Identificar la versión beta de un proyecto
- Crear instantáneas de un repositorio
- Etiquetar el estado del repositorio completo, por ejemplo, cuando sale una nueva versión del software

Diferencias entre etiquetas y ramas

- Las etiquetas permiten referenciar un commit en concreto, mientras que las ramas son colecciones de commits.
- Las etiquetas son fijas e inmutables, mientras que las ramas permiten fusionarse con otras.

Fuente: <https://git-scm.com/book/es/v2/Fundamentos-de-Git-Etiquetado>

### •¿Cómo crear una etiqueta en Git?

El comando **git tag** es el principal para crear, modificar y eliminar etiquetas.

Para crear una etiqueta anotada, se añade el flag "-a".

### •¿Cómo enviar una etiqueta a GitHub?

Para enviar etiquetas a GitHub, primero debes crear las etiquetas localmente y después puedes usar el comando **git push** con la opción **--tags**.

También puedes crear etiquetas directamente en la interfaz web de GitHub.

*Página principal del repositorio -> Incidencias o Solicitudes de incorporación de cambios -> Labels -> Nueva etiqueta -> escribe su nombre, su descripción y color (opcional) -> click en Create label.*

### •¿Qué es un historial de Git?

El historial de Git es un registro de los cambios realizados en un repositorio de código fuente. Se almacena como un gráfico de instantáneas, llamadas confirmaciones, en lugar de una línea recta. Cada confirmación tiene un puntero a una o varias confirmaciones anteriores. El historial de Git permite volver al pasado para ver quiénes son los autores, dónde se introdujeron los errores y revertir los cambios problemáticos. El historial de Git es similar a un gráfico en lugar de una línea recta.

Fuente:

<https://git-scm.com/book/es/v2/Fundamentos-de-Git-Ver-el-Historial-de-Confirmaciones>

### •¿Cómo ver el historial de Git?

El comando **git log** muestra todas las confirmaciones en el historial del repositorio y los cambios de un proyecto en orden cronológico. Incluye información básica del commit, como el id, autor, fecha y mensaje.

**git log --graph** muestra un gráfico ASCII con la historia de ramificaciones y uniones.

Fuente:

<https://git-scm.com/book/es/v2/Fundamentos-de-Git-Ver-el-Historial-de-Confirmaciones>

#### • ¿Cómo buscar en el historial de Git?

**git log -p** muestra las diferencias introducidas en cada confirmación.

**git log -L** muestra el historial de una función o línea de código.

**git log --author "nombre de autor"** para filtrar el historial por autor.

**git log -p -S <string>** para buscar confirmaciones que agreguen o eliminen una cadena específica.

Fuente:

<https://git-scm.com/book/es/v2/Fundamentos-de-Git-Ver-el-Historial-de-Confirmaciones>

#### • ¿Cómo borrar el historial de Git?

**git rebase** permite modificar el historial.

#### • ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un espacio cifrado donde puedes guardar código, archivos y revisiones de forma que solo tú o las personas a las que invites (colaboradores) puedan verlos.

#### • ¿Cómo crear un repositorio privado en GitHub?

Ir a la URL de tu usuario en GitHub -> repositories -> new (<https://github.com/new>)

Eso te lleva a una página (Create a new repository) donde hay que completar los datos (propietario, nombre del repositorio, descripción y en tipo de visibilidad hay que poner "Private") y darle al botón "Create repository".

#### • ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Se puede invitar a colaboradores que tengan cuenta en GitHub:

ir al repositorio -> settings -> access -> Collaborators and teams -> add people (por username, nombre, o email).

#### • ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un espacio virtual donde se puede almacenar, compartir y administrar código, archivos y revisiones de forma que sea accesible para cualquier persona en internet. Los repositorios públicos de GitHub se suelen usar para compartir software de código abierto, permiten que otros usuarios revisen el código y hagan sugerencias para mejorarlo y facilitan la colaboración en proyectos compartidos.

#### • ¿Cómo crear un repositorio público en GitHub?

Ir a la URL de tu usuario en GitHub -> repositories -> new (<https://github.com/new>)

Eso te lleva a una página (Create a new repository) donde hay que completar los datos (propietario, nombre del repositorio, descripción y en tipo de visibilidad hay que poner "Public") y darle al botón "Create repository".

#### • ¿Cómo compartir un repositorio público en GitHub?

El repositorio debe estar público.

Se puede compartir la URL del repositorio.

O se puede invitar a colaboradores:

ir al repositorio -> settings -> access -> Collaborators and teams -> add people (por username, nombre, o email).

**2) URL del repositorio de la actividad 2.**

[https://github.com/hernanbula/Programacion\\_1](https://github.com/hernanbula/Programacion_1)

**3) URL del repositorio de la actividad 3.**

<https://github.com/hernanbula/conflict-exercise>