

# **TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA**

Programación I

Manuel Williams

## **Práctico 2: Git y GitHub**

### **Actividades:**

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

#### **¿Qué es GitHub?**

GitHub es una plataforma de alojamiento de código basada en la nube que permite la colaboración en proyectos utilizando **Git**, un sistema de control de versiones distribuido. GitHub facilita la gestión del código, el seguimiento de cambios y la colaboración entre desarrolladores.

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

- 1- Inicia sesión en GitHub.
- 2- Haz clic en el botón New en la pestaña Repositories.
- 3- Ingresa un nombre para el repositorio.
- 4- También puedes agregar una descripción y seleccionar si será público o privado.

#### **¿Cómo crear una rama en Git?**

Para crear una rama nueva, se utiliza el comando: `git branch + el nombre de la rama`. por ej: `git branch nueva_rama`

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

Para cambiar a una rama ya existente, se utiliza el comando: `git checkout` o `git switch` (en versiones nuevas) + el nombre de la rama a la que te deseas cambiar. Por ej: `git checkout nueva_rama`

### **¿Cómo fusionar ramas en Git?**

Para fusionar ramas se utiliza el comando: `git merge` + el nombre de la rama que deseas fusionar.

Para hacerlo debes primero cambiar a la rama donde deseas fusionar los cambios.

Si existen conflictos entre los cambios realizados en las ramas, Git te pedirá resolverlos manualmente. Una vez resueltos los conflictos, guarda los archivos modificados y realiza un commit para confirmar la fusión.

### **¿Cómo crear un commit en Git?**

Un commit en Git es un registro de cambios en el código. Para crear un commit se utiliza el comando: `git commit`

Para hacerlo debes antes añadir los archivos modificados al área de preparación (staging area) con el comando: `git add`

También puedes agregar al commit un mensaje que contenga la descripción de los cambios con el comando: `-m` + el mensaje entre comillas. Por ej: `git commit -m "Descripción del commit"`

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

Para enviar un commit a GitHub primero debes commitear las modificaciones que hayas realizado en tu repositorio local. y si la rama ya está vinculada al repositorio remoto, se utiliza el comando: `git push` + origin (alias del repositorio remoto en GitHub) + nombre\_rama. Por ej: `git push origin main`

Si la rama aún no existe en el repositorio remoto entonces debes agregar el comando: `-u` luego de `git push`. Por ej: `git push -u origin main`

### **¿Qué es un repositorio remoto?**

Un repositorio remoto es una versión alojada en la nube de un repositorio Git. Es accesible a través de internet y permite a varios desarrolladores colaborar en un mismo proyecto.

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

Para agregar un repositorio remoto a git se utiliza el comando: `git remote add` + origin (alias del repositorio remoto en GitHub) + URL del repositorio de GitHub. Por ej: `git remote add origin https://github.com/usuario/nombre-repositorio.git`

Para hacerlo debes primero crear el repositorio en GitHub y copiar la URL que necesitas para vincularlo a tu repositorio local.

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

Para empujar cambios a un repositorio remoto si la rama ya está vinculada al repositorio remoto, se utiliza el comando: `git push + origin (alias del repositorio remoto en GitHub) + nombre_rama`. Por ej: `git push origin main`

Si la rama aún no existe en el repositorio remoto entonces debes agregar el comando: `-u` luego de `git push`. Por ej: `git push -u origin main`

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

Para traer cambios desde un repositorio remoto si la rama ya está vinculada al repositorio remoto, se utiliza el comando: `git pull + origin (alias del repositorio remoto en GitHub) + nombre_rama`. Por ej: `git pull origin main`

Si la rama no está vinculada al repositorio remoto deberás vincularlas previamente.

Si existen conflictos entre los cambios locales y los remotos, Git te pedirá resolverlos manualmente. Una vez resueltos los conflictos, deberás guardar los archivos modificados y realizar un commit para confirmar la fusión.

### **¿Qué es un fork de repositorio?**

Un fork de un repositorio es una copia del repositorio donde puedes escribir, haciendo públicos tus propios cambios, como una manera abierta de participación.

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

Para crear un fork de un repositorio en GitHub debes:

- 1- Iniciar sesión en tu cuenta de GitHub.
- 2- Ingresar al repositorio que deseas bifurcar (forkear).
- 3- Hacer clic en el botón "Fork" (ubicado en la esquina superior derecha).
- 4- Seleccionar la cuenta donde quieres alojar el fork.
- 5- GitHub creará una copia del repositorio en tu cuenta.

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

Para enviar una solicitud de extracción a un repositorio deberás antes realizar un fork del repositorio original en GitHub, Clonar tu fork en tu máquina local y Agregar el repositorio original como remoto para poder actualizar tu fork más adelante

luego una vez que hayas commiteado cambios en la rama deberás enviar los cambios a tu repositorio de GitHub con el comando `git push`.

Ahora podrás enviar una solicitud de extracción al repositorio.

Para hacerlo:

- 1- Ve a tu fork en GitHub.
- 2- Verás un mensaje que dice: "Compare & pull request" (Comparar y solicitar extracción). Haz clic ahí.
- 3- Verifica que el repositorio base sea el original y la rama en la que hiciste cambios esté seleccionada.
- 4- Agrega un título y una descripción sobre los cambios realizados.
- 5- Haz clic en "Create pull request"

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

Para aceptar una solicitud de extracción (pull request) en GitHub, sigue estos pasos:

- 1- Accede al repositorio en GitHub y ve a la pestaña "Pull requests".
- 2- Selecciona el pull request que deseas revisar.
- 3- Verifica los cambios revisando los archivos modificados y asegurándote de que todo esté correcto.
- 4- Si es necesario, deja comentarios o solicita cambios antes de aceptar el PR.
- 5- Si todo está bien, haz clic en "Merge pull request" y luego en "Confirm merge" para fusionar los cambios.
- 6- Opcionalmente, puedes eliminar la rama si ya no es necesaria haciendo clic en "Delete branch".

### **¿Qué es una etiqueta en Git?**

Una etiqueta (tag) en Git es un marcador que se utiliza para señalar versiones específicas de un repositorio, generalmente versiones importantes como lanzamientos de software.

### **¿Cómo crear una etiqueta en Git?**

Para crear una etiqueta en Git se utiliza el comando: `git tag + nombre_etiqueta`

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

Si deseas compartir una etiqueta en GitHub debes usar el comando: `git push + origin + nombre_etiqueta`, o si deseas subir todas las etiquetas creadas debes usar el comando: `git push + --tags`

### **¿Qué es un historial de Git?**

El historial de Git es el registro de todos los cambios realizados en un repositorio, incluyendo commits, fusiones (merges), ramas y etiquetas. Permite ver quién hizo cada cambio y cuándo lo hizo.

### **¿Cómo ver el historial de Git?**

Para ver el historial de Git se utiliza en comando: `git log`

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

Para buscar en el historial de Git, puedes utilizar varios comandos que te permiten encontrar commits específicos según palabras clave, autor, fecha o contenido modificado.

1. Si recuerdas una palabra clave en el mensaje del commit puedes buscar commits por mensaje usando el comando: `git log --grep="palabra clave"`
2. Si deseas buscar los commits hechos por una persona en particular, puedes buscar commits por autor usando el comando: `git log --author="Nombre del autor"`
3. Si quieres buscar los commits hechos en un período específico, puedes buscar commits dentro de un rango de fechas específicas usando el comando: `git log --since="2024-01-01" --until="2024-03-01"`
4. Si necesitas ver los commits donde un archivo fue modificado puedes buscar un commit con un archivo específico con el comando: `git log -- nombre-del-archivo`
5. Si deseas encontrar un commit donde una línea específica de código fue agregada o modificada, puedes buscar un cambio en el contenido del código con el comando: `git log -S "fragmento de código"`

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

Para borrar el historial de Git, puedes utilizar varios métodos dependiendo de lo que desees lograr. Aquí te dejo algunas opciones:

- 1- Para borrar el último commit pero mantener los cambios en tus archivos (como modificaciones no confirmadas), puedes usar el comando: `git reset --soft HEAD~1`
- 2- Para borrar el último commit y eliminar los cambios realizados en él, puedes usar el comando: `git reset --hard HEAD~1`

3- Si necesitas borrar varios commits, puedes retroceder varios pasos. Por ejemplo, para eliminar los últimos tres commits, usa el comando: `git reset --hard HEAD~3`

4- Si deseas reescribir el historial de commits y eliminar o modificar varios commits, puedes hacerlo de manera interactiva con el comando: `git rebase -i HEAD~n` (n es el número de commits hacia atrás)

5- Si necesitas eliminar el historial en un repositorio remoto y reemplazarlo con tu historial local, puedes usar el comando: `git reset --hard HEAD~n`

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

Un repositorio privado en GitHub es un repositorio al que solo pueden ingresar usuarios autorizados. A diferencia de los repositorios públicos, donde cualquier usuario puede ver y clonar el código, en un repositorio privado solo los colaboradores que tú autorices pueden acceder, ver, modificar o contribuir al código

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

- 1- Inicia sesión en GitHub.
- 2- Haz clic en el botón New en la pestaña Repositories.
- 3- Ingresa un nombre para el repositorio.
- 4- En "Repository visibility", selecciona "Private" para que el repositorio sea privado.

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

Para invitar colaboradores a tu repositorio privado, ve a "Settings" > "Manage access" y haz clic en "Invite a collaborator". Luego, ingresa el nombre de usuario de GitHub de la persona que quieres invitar.

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

Un repositorio público en GitHub es un repositorio cuyo código y archivos son accesibles para cualquier persona en Internet. Cualquier usuario puede ver, clonar y hacer contribuciones al repositorio (si tiene permisos). Estos repositorios son ideales para proyectos de código abierto, ya que fomentan la colaboración y permiten que cualquier persona contribuya.

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

- 1- Inicia sesión en GitHub.
- 2- Haz clic en el botón New en la pestaña Repositories.
- 3- Ingresa un nombre para el repositorio.

4- En la sección "Repository visibility", selecciona la opción "Public". Esto hará que tu repositorio sea accesible para cualquier persona en Internet.

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

Para compartir un repositorio público en GitHub, debes ingresar al repositorio que deseas compartir en GitHub y en la página principal del repositorio, en la parte superior, encontrarás la URL del repositorio en la barra de direcciones del navegador, para compartirlo simplemente copias la URL y la compartes.