# Respuestas sobre Git y GitHub

## Conceptos básicos

### ¿Qué es GitHub?

GitHub es una plataforma de alojamiento de código basada en la nube que utiliza el sistema de control de versiones Git. Permite a los desarrolladores colaborar en proyectos, mantener un historial de cambios, gestionar ramas de desarrollo y compartir código. GitHub ofrece características adicionales como seguimiento de issues, wikis, GitHub Actions para CI/CD, y un sistema para revisión de código mediante pull requests.

### ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu proyecto que está alojada en un servidor (como GitHub, GitLab o Bitbucket) en lugar de estar en tu máquina local. Permite colaboración entre múltiples desarrolladores y sirve como punto centralizado para sincronizar cambios.

## Creación y configuración

### ¿Cómo crear un repositorio en GitHub?

1. Inicia sesión en tu cuenta de GitHub

2. Haz clic en el botón "+" en la esquina superior derecha y selecciona "New repository"

3. Ingresa un nombre para tu repositorio

4. (Opcional) Agrega una descripción

5. Elige si será público o privado

6. Puedes inicializarlo con un README (recomendado para nuevos usuarios)

7. Haz clic en "Create repository"

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

```bash

git remote add origin https://github.com/usuario/repositorio.git

```

Esto asocia tu repositorio local con el remoto, usando "origin" como alias.

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

Sigue los mismos pasos para crear un repositorio, pero selecciona "Private" en lugar de "Public" en la opción de visibilidad.

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

Sigue los pasos para crear un repositorio y selecciona "Public" en la opción de visibilidad.

## Ramas (branches)

### ¿Cómo crear una rama en Git?

```bash

git branch nombre-de-la-rama # Crea la rama

git checkout nombre-de-la-rama # Cambia a la rama

```

O en un solo comando:

```bash

git checkout -b nombre-de-la-rama

```

### ¿Cómo cambiar a una rama en Git?

```bash

git checkout nombre-de-la-rama

```

O en Git versión 2.23+:

```bash

git switch nombre-de-la-rama

```

### ¿Cómo fusionar ramas en Git?

1. Primero cambia a la rama que recibirá los cambios (normalmente main/master):

```bash

git checkout main

```

2. Luego fusiona la otra rama:

```bash

git merge nombre-de-la-rama

```

## Commits y cambios

### ¿Cómo crear un commit en Git?

1. Primero agrega los cambios al área de staging:

```bash

git add . # Agrega todos los cambios

# o

git add archivo.txt # Agrega un archivo específico

```

2. Luego crea el commit:

```bash

git commit -m "Mensaje descriptivo del cambio"

```

### ¿Cómo enviar un commit a GitHub?

```bash

git push origin nombre-de-la-rama

```

Esto envía los commits de la rama local a la rama correspondiente en el repositorio remoto.

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

```bash

git push origin nombre-de-la-rama

```

Si es la primera vez que empujas esa rama:

```bash

git push -u origin nombre-de-la-rama

```

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

```bash

git pull origin nombre-de-la-rama

```

Esto es equivalente a hacer `git fetch` seguido de `git merge`.

## Forks y colaboración

### ¿Qué es un fork de repositorio?

Un fork es una copia personal de un repositorio de otro usuario que te permite hacer cambios sin afectar el proyecto original. Los forks son comunes en GitHub para contribuir a proyectos open source.

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

1. Navega al repositorio en GitHub que quieres bifurcar

2. Haz clic en el botón "Fork" en la esquina superior derecha

3. Selecciona tu cuenta como destino (si tienes múltiples organizaciones)

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

1. Haz cambios en tu fork y envíalos a GitHub

2. Navega a tu fork en GitHub

3. Haz clic en "Pull request" (aparece cuando GitHub detecta cambios recientes)

4. Selecciona las ramas correctas (tus cambios vs el original)

5. Agrega una descripción clara de tus cambios

6. Haz clic en "Create pull request"

### ¿Cómo aceptar una solicitud de extracción?

1. Navega al repositorio en GitHub

2. Ve a la pestaña "Pull requests"

3. Selecciona la solicitud que quieres revisar

4. Revisa los cambios y los comentarios

5. Si todo está bien, haz clic en "Merge pull request"

6. (Opcional) Borra la rama de origen

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

1. Ve a la página principal del repositorio

2. Haz clic en "Settings" > "Manage access"

3. Haz clic en "Invite a collaborator"

4. Ingresa el nombre de usuario, nombre completo o email

5. Selecciona el nivel de permiso adecuado

6. Haz clic en "Add [usuario] to this repository"

## Etiquetas (tags)

### ¿Qué es una etiqueta en Git?

Una etiqueta (tag) es una referencia estática a un punto específico en el historial de Git, comúnmente usada para marcar versiones de lanzamiento (v1.0, v2.0, etc.).

### ¿Cómo crear una etiqueta en Git?

Etiqueta ligera:

```bash

git tag nombre-de-la-etiqueta

```

Etiqueta anotada (recomendada):

```bash

git tag -a v1.0 -m "Versión 1.0 estable"

```

### ¿Cómo enviar una etiqueta a GitHub?

Para una etiqueta específica:

```bash

git push origin nombre-de-la-etiqueta

```

Para todas las etiquetas:

```bash

git push origin --tags

```

## Historial

### ¿Qué es un historial de Git?

El historial de Git es un registro completo de todos los commits realizados en un repositorio, mostrando quién hizo qué cambios y cuándo. Es una de las características principales de Git que permite rastrear y revertir cambios.

### ¿Cómo ver el historial de Git?

Formato básico:

```bash

git log

```

Formato compacto:

```bash

git log --oneline

```

Con gráfico de ramas:

```bash

git log --graph --oneline --decorate --all

```

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

Por mensaje de commit:

```bash

git log --grep="texto a buscar"

```

Por contenido en los cambios:

```bash

git log -p -S"texto a buscar"

```

Por autor:

```bash

git log --author="nombre del autor"

```

### ¿Cómo borrar el historial de Git?

Git está diseñado para mantener el historial, pero puedes:

1. Crear un nuevo commit que elimine los archivos no deseados

2. Usar `git filter-branch` o `git filter-repo` para reescribir el historial (cuidado: esto afecta a todos los colaboradores)

3. Para un nuevo inicio, puedes eliminar el directorio .git y reinicializar:

```bash

rm -rf .git

git init

```

## Compartir repositorios

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

Los repositorios públicos son visibles para cualquiera en internet. Puedes compartir:

1. El enlace directo: https://github.com/usuario/repositorio

2. Usar el botón "Share" en la página del repositorio

3. Incrustar en sitios web usando widgets de GitHub

4. Compartir en redes sociales usando los botones integrados