

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddy Manuel	1	Programación-Richardo	20/05/25

**Title:** Resumen de los comandos Git

<b>Keyword</b>	<b>Topic:</b> Comando Commit
-m: Permite escribir el mensaje del commit directamente en la línea de comandos.	<b>Notes:</b>  Antes de que tus cambios se guarden permanentemente en el historial de tu proyecto, necesitas "commit" en ellos. Esto crea un nuevo punto en la línea de tiempo de tu proyecto. Es muy importante escribir mensajes de commit claros y concisos que expliquen qué cambios hiciste y por qué.
<b>Questions</b>	<b>Ejemplo:</b>
¿Qué pasa si no guardas tus cambios con un Commit?	1. Primero, asegurate de haber añadido los archivos modificados al área de staging con <code>git add &lt;archivo&gt;</code> o <code>git add</code> para añadir todos los cambios.  2. Realiza el commit: <code>git commit -m "añade la función de sesión de usuario"</code>

**Summary:** Un commit guarda los cambios en tu proyecto como una "foto" y crea un nuevo punto en su historial, con mensajes claros que expliquen qué y por qué cambiaste.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddy Manuel	2	Programación-Richardo	20/05/25

**Title:** Resumen de los comandos Git

<b>Keyword</b>	<b>Topic:</b> Comando Clone
máquina local: Es la computadora desde donde trabajas.	<b>Notes:</b> Este comando crea una copia local de un repositorio remoto. El git clone descarga todo el historial del repositorio, todas las ramas y todos los archivos a tu máquina local. Esto te da un punto de partida completo para contribuir al proyecto.
	Ejemplo: <code>git clone https://github.com/micos01/repositorio-EP.git</code>
<b>Questions</b>	Con ese comando clonas el repositorio
¿Existe alguna otra forma de clonar un repositorio?	
Si, con <code>rmkdir</code>	

**Summary:** Git clone descarga una copia completa de un repositorio remoto a tu máquina para comenzar a trabajar en él.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddymanuel	3	Programación-Richardo	20/05/25

**Title:** Resumen de los comandos Git

<b>Keyword</b>	<b>Topic:</b> Comando Branch
<p>-d: Se usa para eliminar una rama.</p> <p>-D: Se usa para forzar a eliminar una rama.</p>	<p><b>Notes:</b> Este comando gestiona las ramas en tu repositorio. Las ramas te permiten trabajar en nuevas funcionalidades o corregir errores de forma aislada sin afectar la rama principal (normalmente llamada main).</p> <p>Puedes crear nuevas ramas, listar las existentes, cambiar entre ramas y eliminar ramas con este comando.</p> <p><b>Ejemplo:</b></p> <ol style="list-style-type: none"> <li>1.- Listar las ramas (git branch)</li> <li>2.- Crear una nueva rama (git branch nueva-rama)</li> <li>3.- Cambiar a una rama existente</li> <li>4.- Crear y cambiar a una nueva rama</li> <li>5.- Eliminar una rama (git branch -d rama).</li> </ol>
<b>Questions</b>	
¿Cómo ayuda git branch a organizar el trabajo en un proyecto?	

**Summary:** El comando git branch se usa para gestionar las ramas en tu repositorio, permitiéndote crear, listar, cambiar de ramas y eliminar ramas.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddy manuel	4	Programación-Richardo	20/05/25

**Title:** Resumen de los comandos git

<b>Keyword</b>	<b>Topic:</b> Comando tag
-a: Indica que se creará una etiqueta anotada.	<b>Notes:</b> Este comando asigna etiqueta (tag) a punto específico en la historia de tu repositorio. Esto se usa comúnmente para marcar lanzamiento de versiones.
-m: Permite añadir un mensaje a la etiqueta.	Las etiquetas son como snapshots permanentes de tu proyecto en un momento dado.
<b>Questions</b>	<b>Ejemplos:</b>
¿Cómo facilitan las etiquetas (git tag)	1.- Listar las etiquetas: <code>git tag</code>
	2.- Crear una etiqueta ligera: <code>git tag v1.1</code>
	3.- Crear una etiqueta anotada: <code>git tag -a v1.2</code> -m "Lanzamiento de la versión 1.2".
	4.- Ver información de una etiqueta: <code>git show v1.2</code>

**Summary:** git tag crea marcadores permanentes en momentos específicos del historial del repositorio, como versiones o lanzamientos.



NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddy Manuel	5	Programación-Richardo	20/05/25

**Title:** Resumen de los comandos git

### Keyword

-v: se usa para establecer el seguimiento de una nueva rama

### Topic: Comando Push

**Notes:** Este comando envía tus commits a un repositorio remoto. Esto es como compartir tus cambios con otros colaboradores y actualizar el repositorio central. Necesitas especificar a qué repositorio remoto y a qué rama quieres enviar tus commits.

### Questions

¿Cómo asegura git push que tus cambios se compartan con el equipo?

Ejemplos:

1.- Enviar la rama actual al repositorio remoto origen.

`git push origin develop`

2.- La primera vez que envías una nueva rama, es posible que necesites usar -v para (upstream).

`git push -v origin nueva-rama`

### Summary:

git push envía tus commits locales a un repositorio remoto, permitiendo compartir tus commits y actualizar el repositorio central.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Eddymanuel	6	Programación-Richardo	20/05/25

Title: Resumen de los comandos git

<b>Keyword</b>	<b>Topic:</b> Comando Pull
main-Rama principal del proyecto	<p><b>Notes:</b> Este comando descarga los cambios desde un repositorio remoto y los integra en tu rama local actual.</p> <p>Esto es como te mantienes actualizando con los cambios realizados por otros colaboradores en el repositorio remoto. git pull es esencialmente un git fetch seguido de un git merge.</p>
<b>Questions</b>	<p>Ejemplo:</p> <p>¿Cuándo debería usar git pull en lugar de git fetch?</p> <ol style="list-style-type: none"> <li>1.- Descargar y fusionar los cambios de la rama main del repositorio remoto origin en tu rama actual: git pull origin main</li> <li>2.- Si tu rama local no está configurada para rastrear una rama remota (lo cual suele suceder después de un git clone o git push), simplemente puedes ejecutar: git pull.</li> </ol>
<b>Summary:</b>	Git pull descarga e integra los cambios del repositorio remoto en tu rama local, manteniéndote actualizado con las modificaciones de otros colaboradores.