

## Primera consigna

### 1. Quién inventó el sistema de control de versiones Git y por qué?

Git fue creado en el 2005 por Linus Torvals y fue creado para ser utilizado en el desarrollo del kernel del Linux. Durante el desarrollo del kernel de Linux, la relación entre la comunidad que desarrollaba el kernel y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux a crear su propia herramienta basada en algunas de las lecciones que aprendieron mientras usaban BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

Velocidad, diseño sencillo, gran soporte para desarrollo no lineal (miles de ramas paralelas), completamente distribuido y capaz de manejar grandes proyectos (como el kernel de Linux) eficientemente (velocidad y tamaño de los datos)

### 2. ¿A quién pertenece actualmente Github y por qué?

El actual propietario de Github es Microsoft Inc. ¿Por qué? Porque Github es la plataforma más popular por las múltiples opciones y posibilidades colaborativas para proyectos de código abierto (Open Source). La clave del éxito que llevó a la mira de Microsoft sea la universalidad de sus dos pilares fundamentales, que son permitir crear versiones diferentes de un mismo proyecto para que cada uno lo adapte a sus necesidades y que estas versiones puedan integrarse después al proyecto principal. GitHub es la plataforma preferida para el trabajo colaborativo sin importar cuál sea el proyecto. Además, es posible recibir ayuda de quienes les interese lo que estás haciendo, dando siempre valioso feedback y colaborando.

### 3. ¿Hay otra forma que no sea la terminal para trabajar con Github?

Sí, se puede descargar la interfaz de Github para trabajar en ella de forma gráfica y sin necesidad de línea de comando.

## Segunda consigna

Comandos más utilizados hasta el momento:

- ★ git init: Crear nuevo repositorio
- ★ git add: Añade cambios del directorio de trabajo en el entorno de ensayo.
- ★ git status: Muestra el estado de los archivos modificados del directorio de trabajo.
- ★ git pull: Extrae y descarga contenidos de un repositorio remoto y actualiza el repositorio local para ver ese contenido.
- ★ git clone: Se utiliza principalmente para apuntar a un repositorio existente y clonar o copiar dicho repositorio en un nuevo directorio, en otra ubicación.
- ★ git commit: Captura el estado de un proyecto en ese determinado momento.