**¿Qué es una computadora?**

Seguramente alguna vez nos hemos preguntado ¿cómo funciona la computadora? Nos detuvimos a pensar, al menos por un momento, qué componentes hay dentro de ella y cómo se relacionan para que esta funcione.

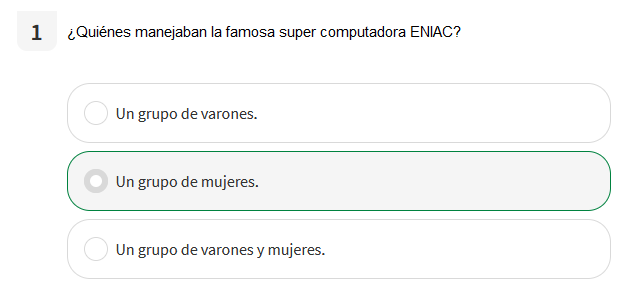
Como ya sabemos, una computadora es un dispositivo electrónico capaz de recibir instrucciones y ejecutarlas. Estas instrucciones son dadas por un usuario, por medio de una interfaz que presenta el sistema operativo lo cual hace más sencilla la comunicación entre el usuario y la máquina. El sistema operativo interpreta y ejecuta con los recursos que dispone, a estos recursos se les conoce como hardware y todo lo que interpreta y ejecuta los deseos del usuario es el software.

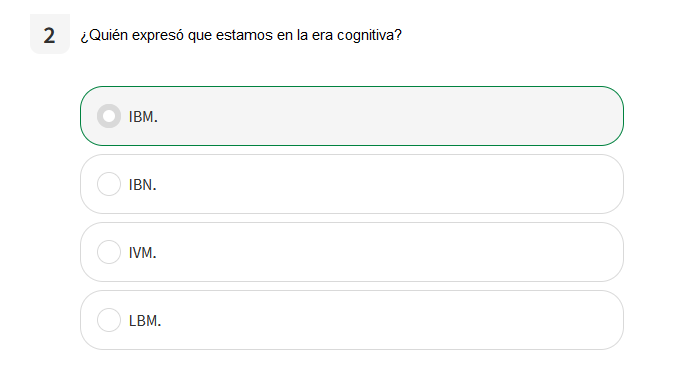
Pero, ¿esto fue siempre así? A lo largo de la historia, esto ha ido evolucionando de manera continua, gracias a la aparición de nuevos componentes y diferentes personas que idearon una forma de maximizar la utilización de estos avances tecnológicos.

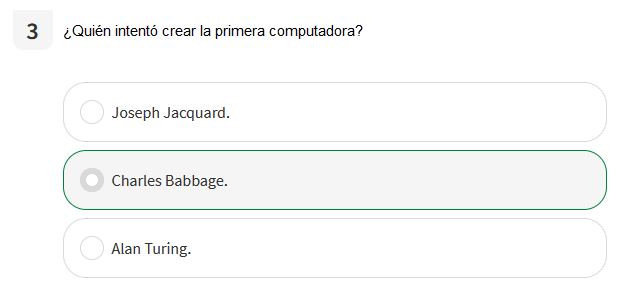
Pues bien, ha llegado el momento de adentrarnos en el estudio de esta maravillosa herramienta. También tendremos la oportunidad de conocer explorar y dar órdenes a nuestro sistema informático mediante texto, ¡qué loco suena! ¿no? Solemos dar instrucciones por medio de una interfaz que nos permite comunicarnos con la computadora, ahora conoceremos otros medios para poder comunicarnos ¡Genial!

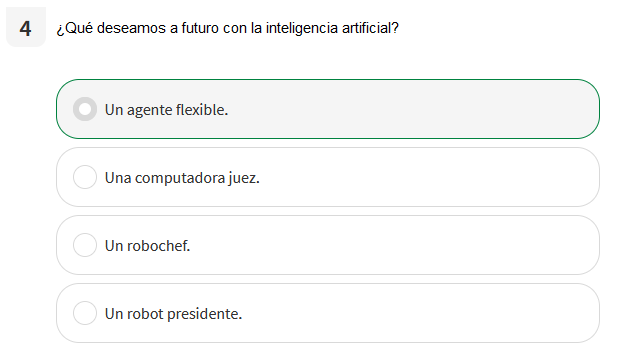
Además, conoceremos otra gran herramienta de la informática para el control de versiones, ¿qué es el control de versiones? Eso lo descubriremos adentrándonos en la unidad y sabremos bien qué herramientas utilizar y cómo llevar sus procesos a cabo.

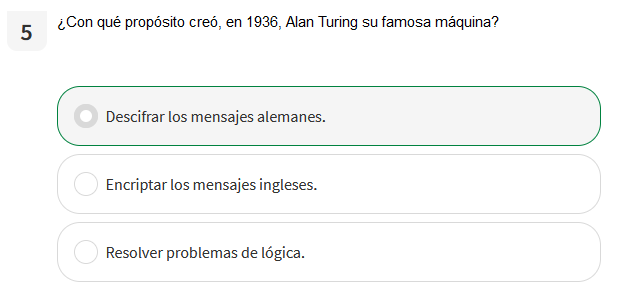
Al finalizar este módulo, contaremos con los conocimientos y capacidades suficientes para comprender el lenguaje técnico en cuanto a componentes y unidades de medidas utilizadas en la informática, como así también, conocerás un poco de la historia y cómo fue que la computadora llegó a ser lo que es hoy en día











# Introducción a la terminal

Cuando hablamos de terminal nos referimos a un software o programa contenedor que ejecuta un shell. Hace décadas, este era un dispositivo físico que consistía en poco más que un monitor y un teclado. Pero, ¿qué es un shell? Es la interfaz de línea de comandos con la que vamos a interactuar, se encarga de procesar datos y devolver resultados.  
  
Entonces, una terminal es una interfaz que nos sirve para comunicarnos con una computadora, un teclado para entrada de datos y una pantalla para mostrar únicamente caracteres alfanuméricos (sin gráficos).  
  
Y cuando hablamos de consola, ¿a qué nos referimos? Una consola es un *tipo especial de terminal*. Para empezar a escribir en la consola es necesario hacerlo a través de líneas de comandos, es decir, una instrucción u orden que nosotros (usuarios) le proporcionamos a un sistema informático. Es una instrucción específica dada a una aplicación informática para realizar algún tipo de tarea o función. La línea de órdenes permiten a los usuarios dar instrucciones por medio de textos sencillos como mostrar archivos, crear archivos, mostrar datos, llamar procesos, entre otros.  
  
Ahora bien, cada sistema operativo incorpora un determinado número de comandos básicos, que permiten ejecutar las tareas más simples con órdenes directas. Esos comandos son propios y generalmente varían según el sistema operativo.  
  
Veamos en mayor profundidad estos conceptos y los diversos comandos que podemos utilizar.

|  |  |
| --- | --- |
| **Mostrar** | |
| **$ ls** | Lista archivos en el directorio o carpeta |
| **$ ls -a** | Lista todos los archivos, incluyendo los archivos ocultos |
| **$ ls -l** | Muestra toda la información de una carpeta: usuario, grupo, permisos, tamaño, fecha y hora de creación. |
| **$ ls -R** | Muestra las carpetas y los archivos contenidos en ellos de manera recursiva |
| **$ pwd** | Muestra la carpeta en la que se está trabajando actualmente |
| **more [Nombre del archivo]** | Muestra el contenido de un archivo |

|  |  |
| --- | --- |
| **Crear** | |
| **$ mkdir [Carpeta]** | Crea una nueva directorio o carpeta |
| **$ touch [Nombre del archivo]** | Crea un nuevo archivo |

|  |  |
| --- | --- |
| **Eliminar** | |
| **$ rm [Nombre del archivo]** | Elimina un archivo |
| **$ rmdir [Nombre de la carpeta]** | Elimina una carpeta vacía |
| **$ rm -r [Nombre de la carpeta]** | Elimina una carpeta y su contenido |

|  |  |
| --- | --- |
| **Copiar/Mover/Renombrar** | |
| **$ mv [ruta/archivo1] [ruta/archivo2]** | Renombra archivos (archivo2 no debe existir o será sobreescrito) |
| **$ mv [ruta/carpeta1] [ruta/carpeta2]** | Renombra la carpeta1 como carpeta2 (carpeta 2 no debe existir) |
| **$ mv [ruta/carpeta1] [ruta/carpeta2]** | Mueve contenido de carpeta1 a carpeta2 (carpeta 2 debe existir) |
| **$ cp [ruta/archivo1] [ruta/archivo2]**  **$ cp [ruta/archivo1] [ruta/archivo2]** | Copia un archivo o carpeta |
| **opción: -r** | Indica que copie recursivamente el contenido de las subcarpetas |

|  |  |
| --- | --- |
| **Navegación entre carpetas** | |
| **$ cd ..** | Sube un nivel de carpeta |
| **$ cd** | Cambia de carpeta |
| **$ cd/chosen/ directory** | Cambia a una carpeta específica |

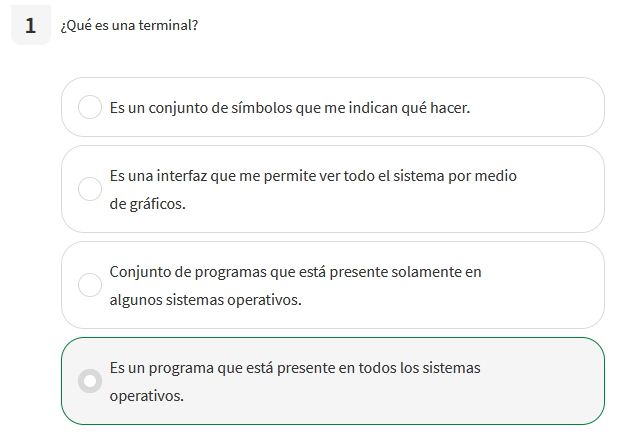
|  |  |
| --- | --- |
| **Otros comandos** | |
| **$ clear** | Limpia la pantalla de la terminal |
| **$ comando --help** | Muestra ayuda del comando |

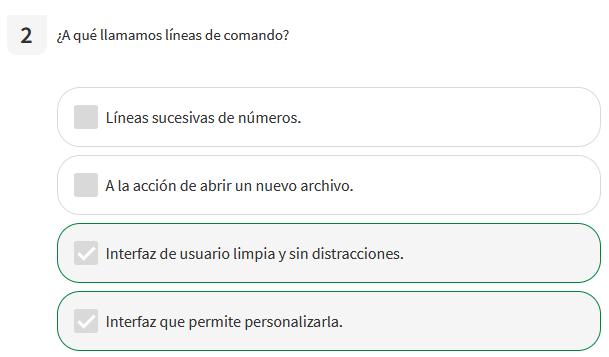
|  |  |
| --- | --- |
| **Atajos de teclado** | |
| **$ ctrl + c** | Finaliza un proceso vigente que está corriendo en la terminal |
| **$ ctrl + l** | Limpia la pantalla de la termina |

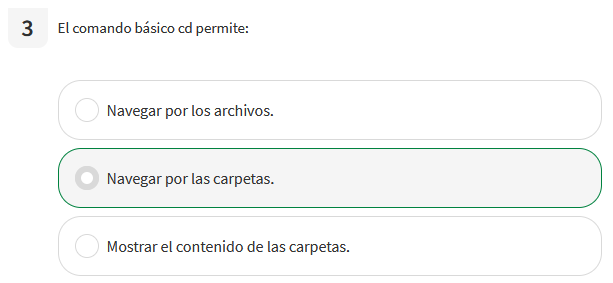
|  |  |
| --- | --- |
| **Caracteres especiales** | |
| **“” (comillas)** | Nos permiten utilizar términos que consistan en más de una palabra |
| **. (el punto)** | Permite hacer referencia al directorio donde estamos ubicados actualmente |

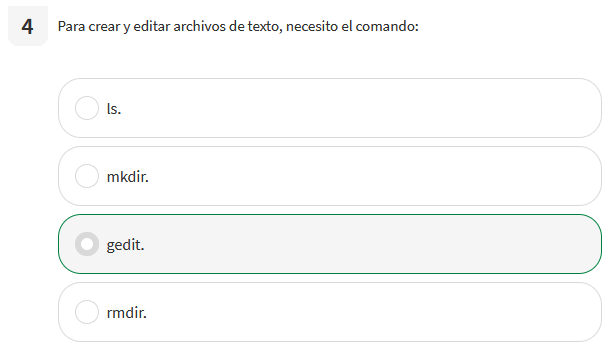


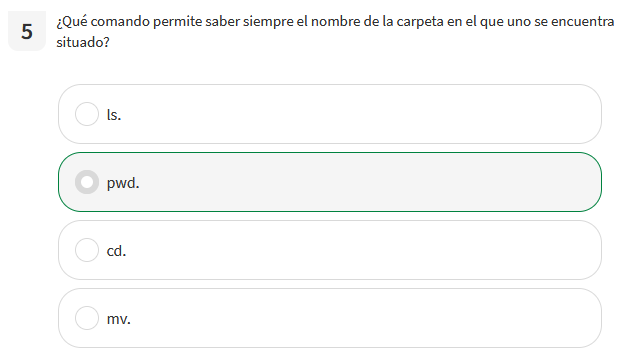






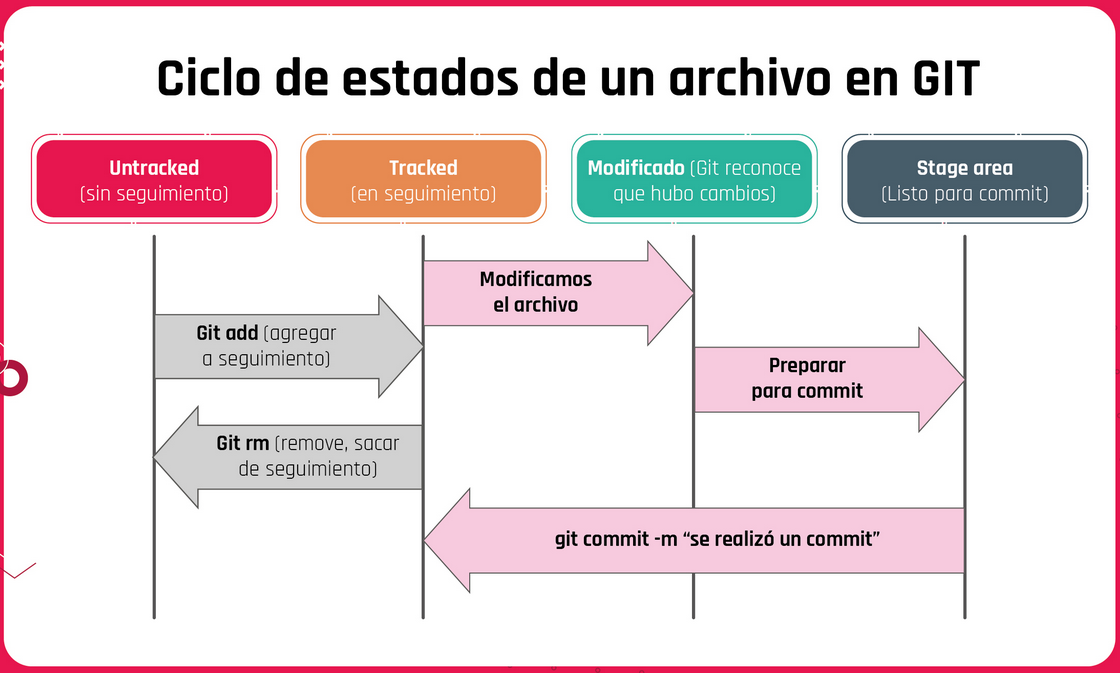








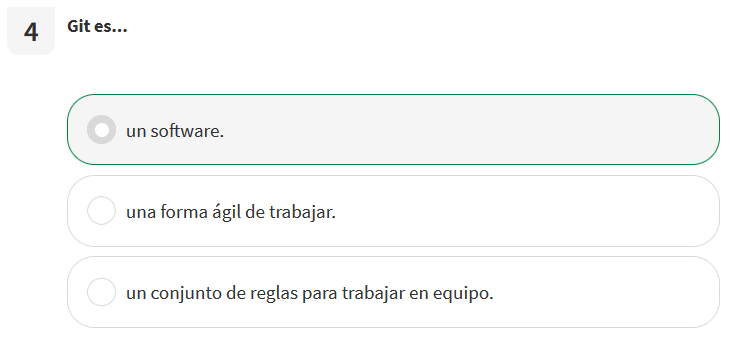
# GIT

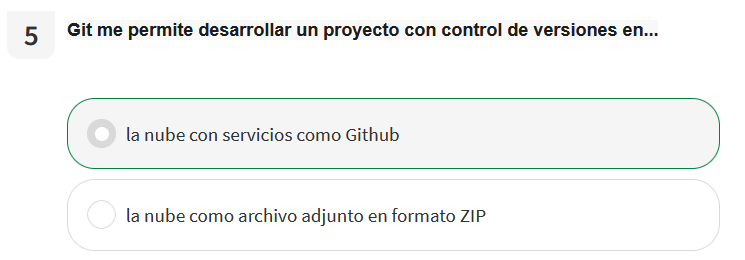


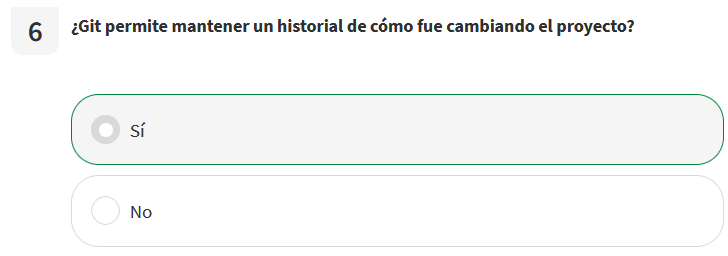
## En Git el ciclo de vida del archivo es:

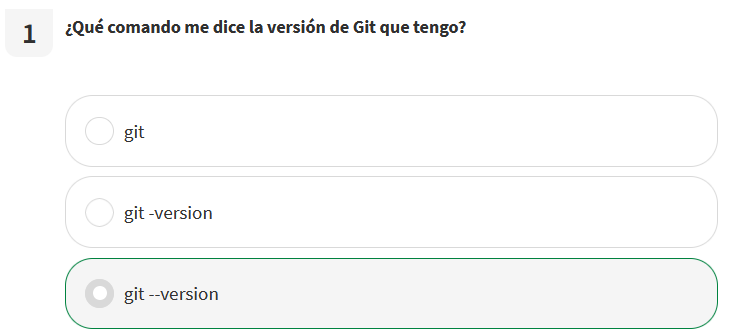
1. Creación del archivo, por ejemplo index.html (aquí el archivo está en estado untracked).
2. Cambiamos el estado del archivo a seguimiento a través del comando git add index.html (especificando nombre del archivo index.html para este caso). Si el archivo ya no será modificado, pasar al paso 6.
3. Si el archivo es modificado, deja de estar en seguimiento y pasa a un estado de modificado, ya que Git detecta que hubo cambios.
4. Guardamos el archivo con ctrl+g (o ctrl + s) y luego escribimos git add index.html (especificando nombre del archivo index.html para este caso), el archivo pasa al stage area.
5. El archivo está listo para realizar el commit, que es cuando se genera el punto histórico, a través del comando git commit.

Para especificar que git no realice más seguimiento a un archivo, usamos el comando git rm --cached donde si bien el archivo como tal no se ha eliminado, Git procede a ignorarlo y pasa a un estado de untracked.









## Palabras claves

A modo de resumen, repasemos las palabras claves de este video:

* Repositorio local: es el que tiene todos los archivos (que hayas guardado en él) en nuestra computadora.
* commits: son los paquetes que nos van a permitir ir haciendo un seguimiento de los cambios que vamos realizando, dado que cada uno de ellos tiene una timestamp, o fecha de creación, y un autor.

Los commits van a ser nuestro historial de cambios que se fueron haciendo en el proyecto.

### Comandos paso a paso

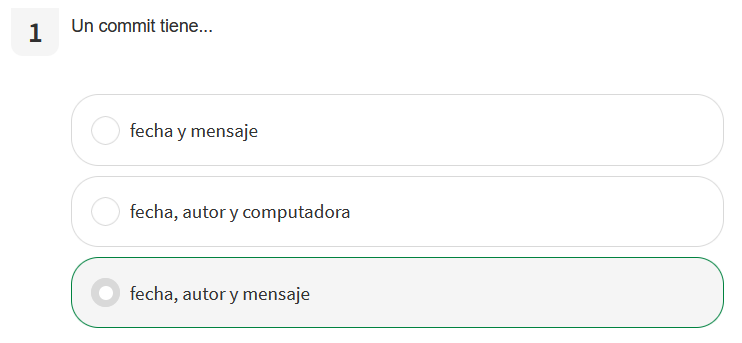
Vamos a ver paso a paso lo que necesitamos hacer para crear nuestro primer repositorio local.

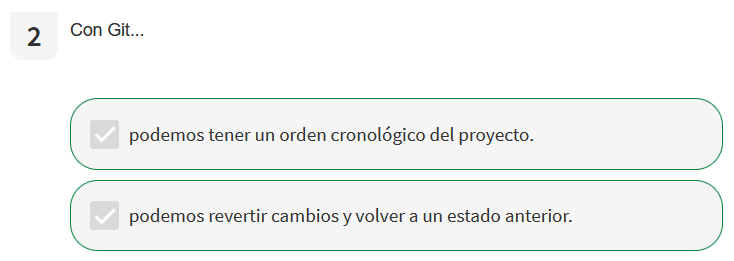


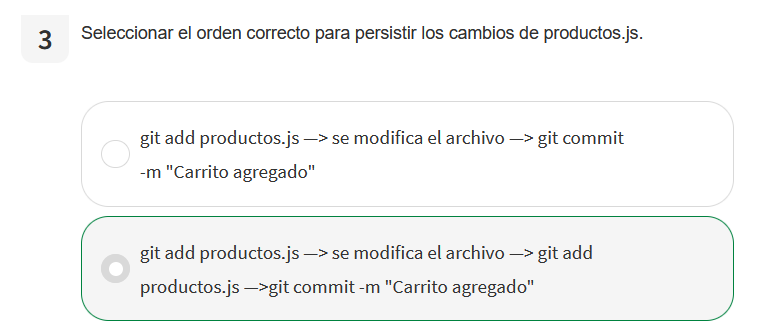
## Confirmando archivos

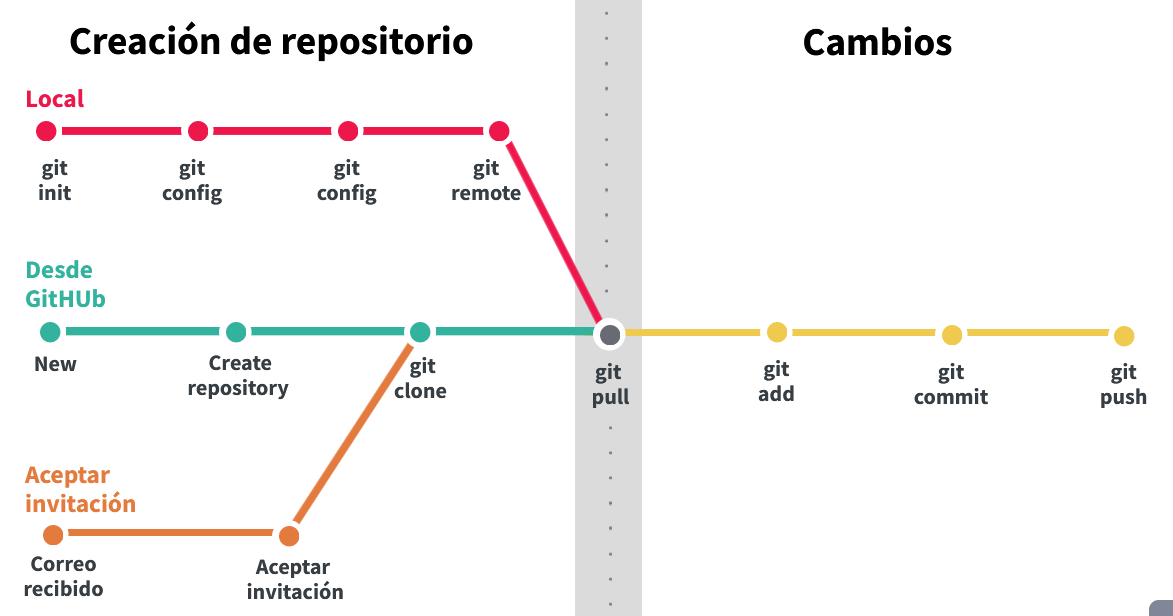
Cuando trabajamos con archivos, estamos acostumbrados a que estos se guarden en forma automática o, a pedirle al programa que lo guarde (el famoso Ctrl+s). Confirmar las modificaciones en Git es de suma importancia ya que nos permite establecer un punto de control.

Comandos: confirmando archivos









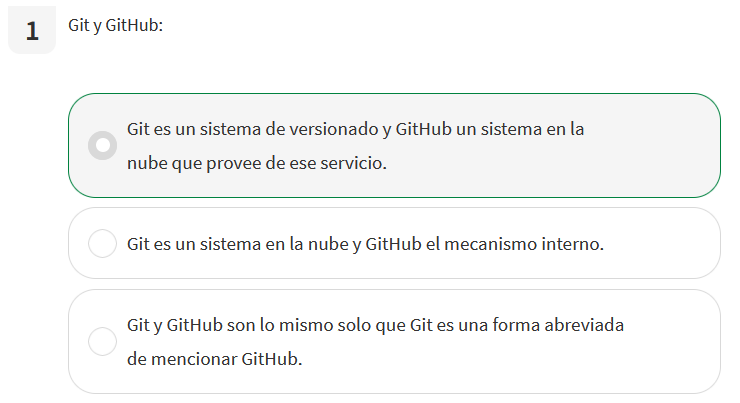
# GitHUB

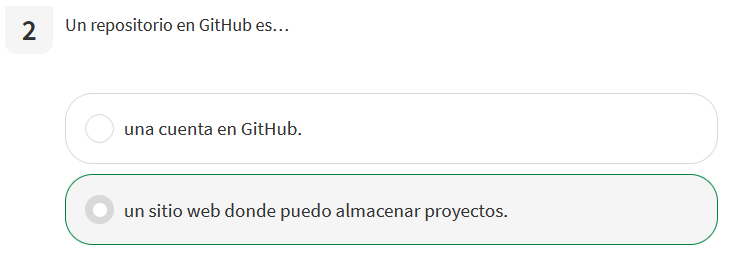
Es una plataforma colaborativa que nos va a permitir llevar un control de versión  
sobre nuestro código.

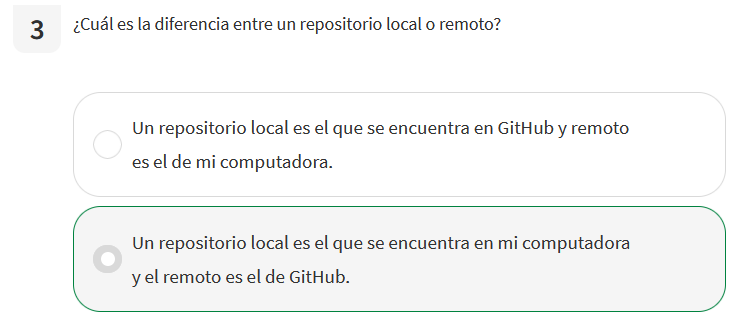
## Palabras clave

A modo de resumen, repasamos las palabras y conceptos clave del video sobre GitHub:

* GitHub es un lugar en la nube.
* Repositorio es el lugar en donde se irán almacenando los archivos de nuestro proyecto y a través del cual podremos hacer seguimiento de los mismos.
* Repositorios remotos: viven en la nube, es decir, en GitHub.
* Repositorios locales: viven en nuestra computadora.







## Subiendo archivos

