**Versionamiento**

Portocarrero Rojas Dillan Smith

Rojas Garzón Farith Santiago

Servicio Nacional de Aprendizaje SENA

Centro de Electricidad, Electrónica y Telecomunicaciones

2202766: Inducción

Tovar Rúgeles Paola Tatiana

16 de febrero del 2021



Mapa mental de conceptos sobre el Versionamiento

<https://mm.tt/1809965980?t=WRG1pL6M2u>

1. ¿Qué es un repositorio y mencione 2 ejemplos de los mismos (proveedores online)?

Los repositorios y archivos digitales de acceso abierto son espacios virtuales, son sistemas de información que preservan y organizan con soporte de base de datos, en los que se puede depositar documentación científica de todo tipo y en todos los formatos posibles (materiales docentes, vídeos, fondo antiguo, colecciones patrimoniales, colección de investigación, etc.).

* Registro de repositorios abiertos OAI.
* Repositorios institucionales internacionales.

2. ¿Cuáles son los principales componentes de un versionamiento en la herramienta GIT?

Commit: Esta operación indica que los cambios se han almacenado, aunque de momento, no estarán disponibles en el servidor para el resto de los miembros del equipo (Solís, 2019).

Push: pública los cambios en el servidor y hace que estos estén disponibles para el resto del equipo (Solís, 2019).

Pull: provoca la descarga de los cambios enviados al servidor por el otro miembro del equipo (Solís, 2019).

Clone: descargar una copia completa del código desde el servidor a su máquina local (Solís, 2019).

Pull request: lo que haces es *solicitar* que otro desarrollador (por ejemplo, el mantenedor del proyecto) *incorpore* (o haga un pull) una rama de tu repositorio al suyo.

Stash: almacena temporalmente (o guarda en un *stash*) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa

RT: es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

* Short and sweet. Si estás comenzando y quieres entender cómo funciona [Git](http://git-scm.com/) o cuales son los primeros comandos para empezar con él.
* Longer, More In Depth. Se nos explica una introducción sobre cómo diseñar una arquitectura para nuestro proyecto que utilice [Git](http://git-scm.com/), una explicación sobre cómo funciona internamente [Git](http://git-scm.com/) y guías más amplias si queremos aprender [Git](http://git-scm.com/) en profundidad.

3.

¿Mencione con sus propias palabras las ventajas que tiene GIT frente a otros proveedores de repositorios?

RT:

Las ventajas que tiene gif cuenta con una potente comunidad de desarrolladores en todo el mundo, permite utilizar flujos de trabajo más flexibles que te ayudan a trabajar mejor y con menos estrés, en general si no se hace un sistema de control de versiona miento GIT ayuda a manejar un código y mejorar el flujo de trabajo y será más óptimo y productivo en tiempo de trabajo.

**4.** Mencione por lo menos 5 ejemplos de los comandos básicos que se usan en GIT.

RT:

* Git help : Muestra una lista con los comandos más utilizados en GIT.
* Git init :Podemos ejecutar ese comando para crear localmente un repositorio con GIT y así utilizar todo el funcionamiento que GIT ofrece.
* Git add + path: Agrega al repositorio los archivos que indiquemos.
* Git add -A: Agregar al repositorio TODOS los archivos y carpetas que estén en nuestro proyecto, los cuales GIT no está siguiendo.
* Git commit -m "mensaje" + archivos: Hace commit a los archivos que indiquemos, de esta manera quedan guardados nuestras modificaciones.

**5.** ¿Qué son y cuáles son las funciones de los branch?

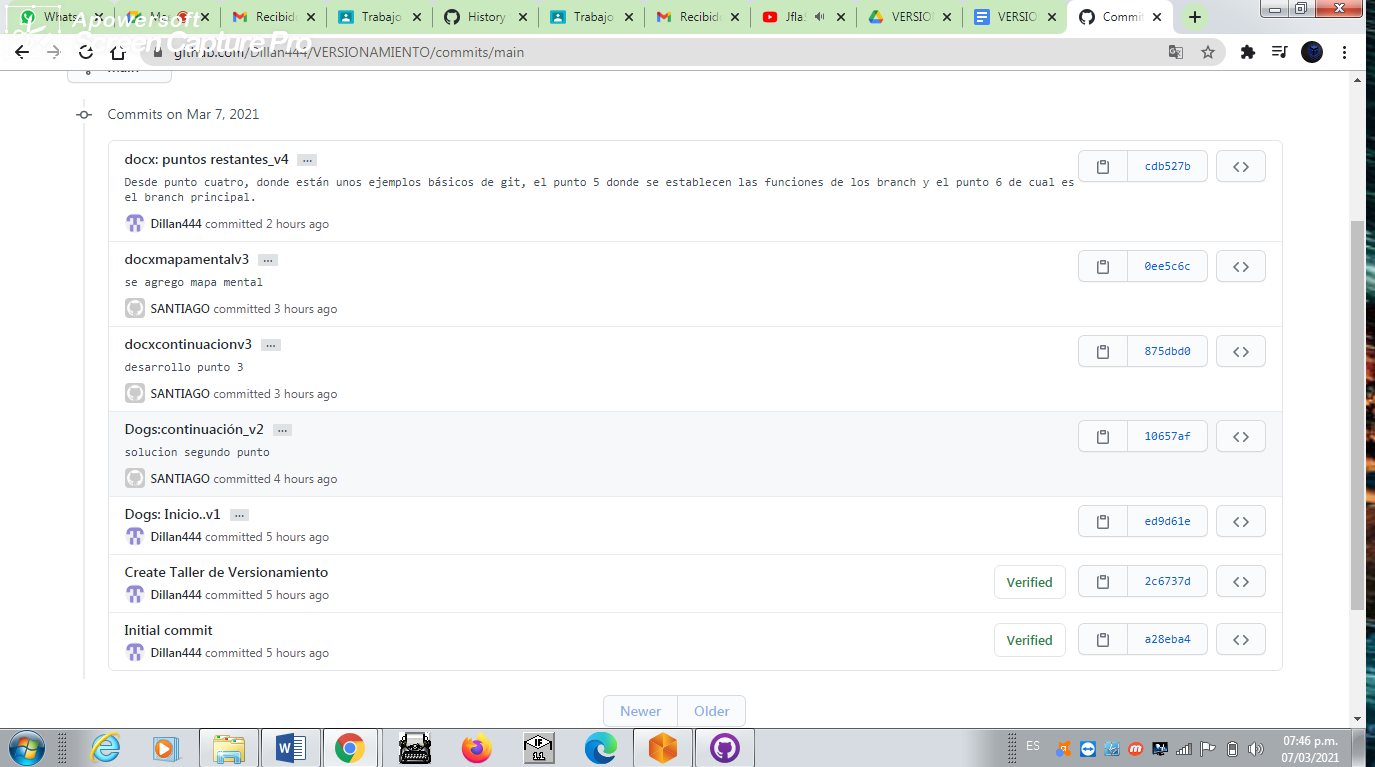
RT:  Las funciones más importantes de Git es el control de las ramas de desarrollo, conocidas como Branch, en general, una rama de desarrollo (“Git Branch”) es una bifurcación del estado del código que crea un nuevo camino para la evolución del mismo. Puede ir en paralelo a otras Git Branch que se pueden generar.

cuáles son las funciones:

* Para desarrollar una nueva funcionalidad sin afectar al máster mientras lo hacemos.
* Para hacer un hotfix en una versión que ya ha salido a producción.
* Para hacer un branch de producción, otro de pre, otro de testing, y así ir promoviendo los cambios de uno a otro.
* Para gestionar distintas versiones de un mismo producto: podríamos tener un branch por cada cliente donde está instalado el producto (la verdad es que esto no lo recomiendo porque en cuento tengas más de un cliente la gestión se vuelve un infierno, es mejor tener un proyecto único y módulos de personalización; aquí lo comento simplemente porque he visto a gente trabajar así en la oscuridad cerca de la puerta de  
  Tannhäuser).
* Rama master: la rama principal del proyecto. Sobre esta rama no se hace trabajo, solamente se utiliza como copia de las versiones del software que se despliegan. Esta rama estará configurada con los valores de configuración para el entorno del servidor de producción (Solís, 2019).
* Rama develop: la rama de trabajo diario del proyecto. La mayor parte de los cambios que se realizan sobre el código del proyecto se realizan en esta rama. Esta rama estará configurada con los valores de configuración para los entornos de desarrollo (Solís, 2019).
* Ramas feature: cuando se necesita desarrollar una funcionalidad aislada en un proyecto se abre una rama de este tipo y se realiza el desarrollo completo de dicha funcionalidad en esta rama. Una vez completado el desarrollo, se vuelca la rama dentro de la rama develop para incluir la nueva funcionalidad en el código del proyecto (Solís, 2019).
* Ramas release: cuando se va a realizar un despliegue de una versión del código en el servidor de producción se abre una rama de este tipo. Una vez abierta se hacen los cambios de configuración relativos al entorno de producción y se sube el código del proyecto al servidor. Cuando el despliegue se completa de forma satisfactoria, la rama reléase se vuelca a la rama master y se añade una etiqueta con el número de versión correspondiente (Solís, 2019).
* Rama hotfix: cuando se quiere solventar un error pequeño en el código que está desplegado en el servidor, se abre una rama de este tipo desde la rama master y se realizan los cambios pertinentes en ella. Una vez completados los cambios se vuelcan tanto en la rama develop como en la rama master, que se etiqueta con la nueva versión. En ese momento, el arreglo estará disponible en las dos ramas fundamentales del repositorio del proyecto. (Solís, 2019)

**6.** ¿Cuál es el nombre del branch principal?

RT: Cuando un proyecto no tiene ramas específicas se considera que el desarrollo se está produciendo en la rama principal, también conocida como "línea primaria" o "trunk".

**Repositorio de versionamiento**

(Fuente propia, 2021)

**Bibliografías**

Solís, J. (2019). *neozink*. Obtenido de Desarrollo Web en Equipo: Como usar GIT y no morir en el intento: https://neozink.com/blog/como-usar-git/#:~:text=Un%20repositorio%20GIT%20se%20compone,las%20funcionalidades%20o%20mejoras%20realizadas.