 NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7

GA7-220501096-AA1-EV03 identifica herramientas de versionamiento


Libardo H. Bautista, Juany A. Calvache, Julián A. Ruiz y Diana Shirley Chávez

Servicio Nacional de Aprendizaje (SENA)

Grupo 2758307, Análisis y Desarrollo de Software


Ing. Fernando Bohórquez García

Abril 11 de 2024

 NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7

Contenido

Introducción	3
GA7-220501096-AA1-EV03 identifica herramientas de versionamiento	3
Tabla con las diferencias entre el sistema de control de versionamiento git local y git remoto	4
Tabla con la descripción de los comandos básicos de git remoto y git local.....	4
Conclusiones	6
Bibliografía	7

 NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7


Introducción

Git es un sistema de control de versiones distribuido que facilita el trabajo en ramas independientes, fusiones, y etiquetado de versiones. Permite a los desarrolladores trabajar localmente y sincronizar su trabajo con un repositorio remoto. Es por ello, que la elección de Git se debe a su robustez, flexibilidad y amplia adopción en la industria. Su modelo distribuido es ideal para equipos descentralizados y contribuye a una mejor gestión de conflictos y una historia de cambios más clara.

Sin embargo, debemos tener claro dos conceptos: el Git local y el Git remoto. El Git local nos permite trabajar de forma aislada. Es decir, es nuestro repositorio de Git en nuestra máquina. Aquí es donde hacemos cambios y commits. Los cambios solo afectan el repositorio local hasta que decidimos compartirlos. Mientras que el Git remoto facilita la colaboración y sirve como respaldo al código. Es una versión de nuestro proyecto alojada en la red, como GitHub. Todos los colaboradores del equipo de trabajo podemos acceder y contribuir aquí.

GA7-220501096-AA1-EV03 identifica herramientas de versionamiento

Para cumplir con esta evidencia, vamos a tener en cuenta los requisitos exigidos en la guía de trabajo # 7, que nos pide, tomar como referencia el componente formativo “Integración continua” y realizar una tabla con las diferencias entre el sistema de control de versionamiento git local y git remoto. También, se deben seguir las normas básicas de presentación de un documento

 NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7

escrito, es decir el documento debe tener como mínimo una portada, introducción, objetivo, selección de herramientas de versionamiento a utilizar; y, crear una tabla con la descripción de los comandos básicos de git remoto y git local.

Tabla con las diferencias entre el sistema de control de versionamiento git local y git remoto

Tabla N° 1

Diferencias entre Git Local Vs. Git Remoto

Ítem	Git Local	Git Remoto
Ubicación	En tu máquina local.	En la red (por ejemplo, GitHub, GitLab, Bitbucket).
Función principal	Realizar cambios y commits en el código.	Compartir el código y colaborar con otros.
Acceso	Solo tú tienes acceso a menos que compartas explícitamente.	Todos los colaboradores del proyecto pueden acceder y contribuir.
Conexión a Internet	No se necesita para trabajar.	Se necesita para empujar (push) o jalar (pull) cambios.
Integraciones	Limitado a las herramientas en tu máquina local.	Permite integraciones con otras herramientas, como CI/CD, rastreadores de problemas, etc.


Nota. Resumen hecho por los autores del trabajo con información proporcionada por el componente formativo SENA.

Tabla con la descripción de los comandos básicos de git remoto y git local.

Tabla N° 2

Descripción de los comandos básicos de git remoto y git local


Comando	Descripción	¿Es remoto o es local?
git init	Inicializa un nuevo repositorio Git en tu directorio local.	Local
git clone <url>	Clona un repositorio remoto a tu máquina local.	Ambos

 CPIENSA+ CORPORACIÓN NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7

git add <archivo>	Agrega un archivo al área de preparación para el próximo commit.	Local
git commit -m "<mensaje>"	Crea un nuevo commit con los cambios en el área de preparación.	Local
git status	Muestra el estado de tu repositorio local.	Local
git push <remoto> <rama>	Empuja los commits de tu rama local al repositorio remoto.	Remoto
git pull <remoto> <rama>	Trae los cambios del repositorio remoto a tu repositorio local.	Remoto
git remote add <nombre> <url>	Agrega un nuevo repositorio remoto con el nombre y la URL especificados.	Remoto
git remote -v	Muestra una lista de los repositorios remotos que has configurado.	Remoto
git branch <nombre>	Crea una nueva rama en tu repositorio local.	Local
git checkout <nombre>	Cambia a la rama especificada en tu repositorio local.	Local
git merge <nombre>	Fusiona la rama especificada con la rama actual en tu repositorio local.	Local
git diff <rama1> <rama2>	Muestra las diferencias entre dos ramas en tu repositorio local.	Local
git fetch <remoto>	Trae los cambios del repositorio remoto a tu repositorio local, pero no los fusiona con tu rama actual.	Remoto
git remote remove <nombre>	Elimina la referencia al repositorio remoto especificado.	Remoto
git stash	Guarda los cambios que aún no has commit en un lugar temporal para que puedas cambiar de rama sin perder tu trabajo.	Local
git stash pop	Recupera los cambios que guardaste con git stash.	Local

Nota 1. Realizado por los autores del trabajo con información proporcionada por el componente formativo SENA y la plataforma w3school.

Nota 2. Debemos reemplazar <url>, <archivo>, <mensaje>, <remoto>, <rama> y <nombre> con los valores correspondientes a tu proyecto. Recuerda que estos son solo algunos de los comandos básicos de Git, hay muchos más que podrías necesitar dependiendo de tu flujo de trabajo. También, reemplazar <nombre>, <rama1>, <rama2> y <remoto> con los valores correspondientes a tu proyecto. Que en nuestro caso es el SIFBAMBOO 1.0

 NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7


Conclusiones

Git Local es el repositorio de Git que se encuentra en tu máquina local. Aquí es donde realizas cambios en los archivos, agregas nuevos archivos, eliminas archivos y creas commits. Todos estos cambios se realizan localmente y solo afectan el repositorio de Git en tu máquina hasta que decidas compartirlos con otros. Los beneficios de tener un repositorio local incluyen:

- Puedes trabajar de manera aislada sin afectar el trabajo de los demás.
- Puedes trabajar sin conexión a internet.
- Los commits son rápidos ya que todo está en tu máquina local.

Git Remoto: Es una versión de tu proyecto que se aloja en Internet o en la red. Puede ser en GitHub, GitLab, Bitbucket, entre otros. Este es el lugar donde todos los colaboradores del proyecto pueden acceder y contribuir al código. Los beneficios de tener un repositorio remoto incluyen:

- Facilita la colaboración entre múltiples desarrolladores. Cada desarrollador puede trabajar en su propio repositorio local y luego puede empujar sus cambios al repositorio remoto.
- Sirve como un respaldo de tu código. Si algo le sucede a tu máquina local, aún tienes una copia de tu código en el repositorio remoto.
- Permite integraciones con otras herramientas, como sistemas de integración continua/despliegue continuo (CI/CD), rastreadores de problemas, etc.

 CPIENSA+ CORPORACIÓN NIT. 901.510.404-9	COMUNICACIONES EXTERNAS	Código: CCP FCE01
		Versión:01
		Página 1 de 7

En conclusión, el Git local te permite trabajar en tu código de manera aislada en tu máquina local, mientras que el Git remoto te permite compartir tu código y colaborar con otros. Ambos son componentes esenciales en el flujo de trabajo de control de versiones con Git.

Bibliografía

Atlassian. (s.f.). Qué es el control de versiones | Atlassian Git Tutorial.
<https://www.atlassian.com/es/Git/tutorials/what-is-version-control>.

Git. (2021) Git - Acerca del control de versiones. Git --Local-Branching-on-the-Cheap. <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>

Younes. (2021). Gitlab VS Github VS BitBucket. Which one deserve your time? DEV Community. <https://dev.to/yafkari/gitlab-vs-github-vs-bitbucket-which-one-deserve-your-time-2npm>