**Bonus de Modulo II Curso Git**

Si alguien quiere hacer el BONUS (+10%) debe acudir a:

<https://www.atlassian.com/git/tutorials/comparing-workflows>

Y generar un pequeño documento (1 página), explicando qué diferencias notables hay entre:

\* GitFlow (<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>)

\* ForkingFlow (<https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow>)

Y para qué usaría cada uno, con casos inventados.

(Ej: arquitectura de la aplicación, diferentes desarrolladores, equipos remotos, infraestructuras, etc) Básicamente se pide un artículo de opinión, recogiendo lo que se ha visto durante el curso.

**Diferencias entre GitFlow y ForkingFlow**

* El flujo de trabajo de Gitflow define un modelo de ramificación estricto diseñado en torno a la versión del proyecto. Esto proporciona un marco robusto para la gestión de proyectos más grandes.
* Gitflow es ideal para proyectos que tienen un ciclo de lanzamiento planificado.
* Este flujo de trabajo no agrega ningún concepto o comando nuevo más allá de lo que se requiere para el flujo de trabajo de la rama Feature.
* Se asigna funciones muy específicas a diferentes ramas y define cómo y cuándo deberían interactuar. Además de las ramas Feature, utiliza ramas individuales para preparar, mantener y registrar las publicaciones (Develop, Master, Release, HotFix).
* Por supuesto, también puede aprovechar todos los beneficios del Flujo de trabajo de la rama Feature: pull requests, experimentos aislados y colaboración más eficiente.
* Además de la idea abstracta de Flujo de trabajo de Gitflow, existe un conjunto de herramientas de flujo de git más tangible que se integra con Git para proporcionar extensiones de herramienta de línea de comando especializadas de Gitflow Git.
* El flujo de trabajo de Forking se usa comúnmente en proyectos públicos de código abierto para el cual pueden trabajar desarrolladores que quieran contribuir con el proyecto.
* La principal ventaja del flujo de trabajo de Forking es que las contribuciones se pueden integrar sin la necesidad de que todos accedan a un único repositorio central. Los desarrolladores acceden a sus propios repositorios del lado del servidor, y solo el mantenedor del proyecto puede acceder al repositorio oficial. Esto le permite al mantenedor aceptar confirmaciones de cualquier desarrollador sin darles acceso de escritura a la base de código oficial.
* El uso de Gitflow sería aconsejable para proyectos que usen metodología Agiles, es decir tengan planificadas cada una de las funcionalidades y en qué momento serán liberadas cada una de las versiones, cuenten con un equipo de trabajo grande y necesiten tener control de trabajo eficiente en el desarrollo del proyecto, pueden trabajar de forma remota ya que asignadas las funcionalidades al desarrollador se podrá con este flujo de trabajo llevar de forma organizas el control de códigos y versiones por parte del equipo de trabajo.
* En el caso de Forking es recomendables para el desarrollo de proyectos públicos donde cada desarrollador tiene su propio repositorio y a través de un Fork puede pedir contribuir con el repositorio central, donde los desarrolladores podrán solicitar la inclusión de su código en el servidor a través de un Pull Request a este repositorio central, que puede ser rechazado o no en función de su aporte y conflictos, por parte del mantenedor del proyecto o equipo a cargo responsable, por su puesto todo este trabajo se puede hacer de forma remota.

**PD:** Gracias por tu recomendación, estuvieron muy interesante los sitios estudiados y analizados para completar conocimientos con el modulo dado.

**Ariem Umpierrez Padrón**