



INSTITUTO TECNOLÓGICO DE COSTA RICA

CE-5508 SOA4ID: Arquitectura Orientada a
Servicios Aplicada a Sistemas Emergentes

Grupo 1

Investigación 1

Git

Estudiante:
LUIS CARLOS MORA FONSECA- 2017197955

Profesora:
Alejandra Bolaños Murillo

I SEMETRES 2022

¿Qué son las "Branching strategies"?

Las branching strategies se le llama a las diferentes maneras de gestionar un proyecto mediante el uso de ramas en un manejador de versiones como lo es git, donde cada rama va a tener un rol bien definido, esto también dependiendo de que estrategia se esté usando, de modo que, a la hora de desarrollar nuevas funcionalidades, reparar errores y demás, estas no interfieran con el trabajo que se está realizando en otras ramas o con la versión principal y más estable, hasta que dichas nuevas funcionalidades, reparación de errores y demás, estén completamente desarrollados y estables de manera que se pueda fusionar con la rama principal o con otras ramas dependiendo de la estrategia.

¿Cuál es la diferencia entre un Mono repo y un poly-repo?

Cuando se desarrolla un proyecto, dependiendo del tamaño del mismo, este se puede separar en diferentes funcionalidades o componentes, a la hora de manejar este mediante un controlador de versiones se puede realizar principalmente de dos maneras, Mono-repo significa que cada funcionalidad o componente se separará a lo sumo en diferentes directorios, pero siempre en el mismo repositorio, por otro lado, en Poly-repo estos se pueden manejar cada uno en su propio repositorio, lo cual puede ser bastante útil si el mismo componente o servicio se desea utilizar en varios proyectos diferentes. En realidad, cada uno tiene sus ventajas y desventajas, dependerá del criterio del desarrollador determinar cuál utilizar dependiendo del proyecto.

¿Qué son git submodules?

Los git submodules no son más que la manera de incluir otros repositorios a nuestro proyecto, de modo que podemos utilizar los componentes o servicios que nos ofrecen estos otros proyectos dentro del repositorio donde se creó el submódulo.

¿Qué es el Git flow?

Git Flow se podría decir que es uno de los branching strategies más populares, sino el más popular, de manera resumida básicamente consiste en una rama Main donde se manejará la versión principal y más estable del proyecto, que también sería la que se entrega al cliente o al mercado, a partir de esta se crea una rama Develop que va a servir como rama de integración, cada nueva funcionalidad debe tener su propia rama, a estas ramas se les llama Feature y se crean a partir de la rama Develop, una vez terminada la funcionalidad la rama Feature de esta se debe fusionar con la rama Develop, luego, cuando se desee o se deba realizar una nueva publicación, se debe bifurcar una nueva rama llamada Release a partir de Develop la cual una vez verificada se debe integrar a la rama Main, por último se tiene la rama Hotfix la cual se crea a partir de Main para darle mantenimiento, una vez este completado se debe fusionar tanto con Main como con Develop.

¿Qué es el upstream y el remote para un repositorio local de git?

Los repositorios remotos son versiones del proyecto que no se encuentran ubicados de manera local, sino que en internet o alguna otra red. Ahora para entender el upstream se debe hacer en el contexto de los forks, que son la copia de un repositorio en nuestra propia cuenta, luego dicha copia se clona en un repositorio local y de manera predeterminada se tendrá el remoto llamado *origin*, luego para mantener nuestro proyecto actualizado con los cambios realizados en el proyecto original se realiza un remoto comúnmente llamado *upstream* de la siguiente manera:

```
git remote add upstream git://github.com/<aUser>/<aRepo.git>
```

De esta forma se pueden traer los cambios a nuestro repositorio.

¿Qué es un Pull request?

Un Pull request no es más que una solicitud para fusionar dos ramas diferentes, de modo que una vez las aportaciones estén completas en la rama secundaria, se crea un pull request para que dichas aportaciones sean verificadas y aceptadas o rechazadas por el administrador del proyecto.

¿Cuál es la ventaja de las soluciones de manejo de código como: Github, Gitlab, Bitbucket?

Entre sus múltiples ventajas se encuentran, la posibilidad de trabajar en equipo sobre un mismo proyecto compartido, tener un mayor control sobre las modificaciones que se realizan en el proyecto, quien las realiza, así como un historial de estas, mayor autonomía, ya que cada desarrollador tiene su copia de manera local, esto mismo permite que la falta de conexión (no de manera permanente) no impida seguir trabajando en el proyecto.