

Branching patterns

Integrantes:

- Beatriz Arbizu Ramírez - UO225211
- Juan Mencía Menéndez - UO264197
- Joaquín Hermida Giganto - UO271718

Contenido:

Existen distintos patrones que permiten a los equipos de desarrollo trabajar en ramas de forma eficiente, sin malgastar tiempo o complicándose a la hora de juntar las distintas partes.

Patrones base

Ramificación del origen: Para permitir que distintos desarrolladores puedan trabajar sobre el mismo código sin molestarse unos a otros (si uno quiere compilar y probar su parte, tiene que pedirle al resto que paren de trabajar), cada desarrollador trabaja sobre una copia del código base. A esta copia se le llama rama o branch. Un sistema de control de versiones ayuda a la hora de gestionar el trabajo realizado y fusionar distintas ramas.

Línea principal: Consiste en tener una rama a la que consideramos como el trabajo actual del equipo (master/trunk). Cuando un desarrollador quiera compartir su funcionalidad, lo que hará será actualizar esta rama master con su trabajo. Así, si alguien quiere empezar a desarrollar, puede crear una rama a partir de master; y si ya ha empezado un trabajo, puede mantenerlo actualizado con el trabajo de sus compañeros pidiendo a master los cambios.

Rama saludable: Se trata de escribir tests a medida que se desarrolla el trabajo, de forma que si este los pasa se puede considerar que no contiene errores o bugs. Se comprueban cada vez que un desarrollador actualiza una rama y, si fallan, ésta se deja congelada, sin permitir añadir contenido que no sea para solucionar los problemas.

El resto de los patrones más desarrollados se pueden clasificar en:

- **Integración.** Definen cómo organizar las diferentes ramas en las que los desarrolladores trabajan y cómo combinarlas al trabajo principal del proyecto.
- **Producción.** Son los patrones que definen cómo pasar el contenido de la rama principal de desarrollo a una que está ya en producción.

Patrones de Integración

Existen unas características que pueden afectar a cómo los desarrolladores llevan a cabo el sistema de branching:

- **Frecuencia de Integración:** Es la frecuencia con la que los desarrolladores suben los cambios a la rama principal. Una baja frecuencia puede generar varias situaciones problemáticas.
- **Fricción de Integración:** Si el proceso de integración presenta muchas pegas, podría llevar al proyecto a que tenga una menor frecuencia de Integración.
- **Modularidad del proyecto:** Tener un buen modularidad, aunque costoso, siempre es beneficioso. Al dividir bien el proyecto, los cambios que haga cada desarrollador no afectarán a una parte tan grande de la aplicación como lo haría una mala modularidad. Además, esto implicará que los desarrolladores puedan trabajar con menos riesgos de

que sus cambios choquen con los cambios de otro desarrollador del equipo a la hora de juntar sus trabajos individuales.

Cada desarrollador puede seguir distintos patrones a la hora de crear y trabajar con sus ramas:

- **Integración en línea principal:** Hay una rama principal en la que los desarrolladores mergean los cambios que hacen en sus ramas personales, y de la que obtienen los cambios que hacen el resto de los desarrolladores.
- **Ramas de funcionalidades:** Los desarrolladores, dentro de cada repositorio propio, emplean ramas secundarias para desarrollar funcionalidades específicas que luego mergean junto a su rama principal personal. De esta rama secundaria se pasa a la rama principal personal y, de esta, a la rama principal del proyecto. No se hace merge de la rama de la funcionalidad a la rama master hasta que la funcionalidad esté terminada. Este patrón tiene una muy baja frecuencia de integración.
- **Integración continua:** Es la alternativa al patrón "Ramas de funcionalidades". Consiste en que todos los desarrolladores deben subir los cambios que hagan con menos de 1 día de trabajo. Este patrón tiene como desventaja que se sube a la rama principal código sin terminar, lo que puede ocasionar errores en el código de los desarrolladores que hagan un pull de ese código incompleto.
- **Revisión pre-Integración:** Consiste en que, tras cada commit a la rama principal, hay que comprobar que el código esté bien hecho y que no haya errores que puedan perjudicar al proyecto. Esta técnica se combina bien con la Integración en línea principal, y muy bien con las Ramas de funcionalidades. Sin embargo, no funciona muy bien cuando se combina con la Integración continua.

El camino desde la línea principal hasta el lanzamiento de producción

Se basa en el patrón base de rama saludable pues es dicha rama la que se considera el producto en ejecución. Y solo se emplea en proyecto de actualización continua.

- **Rama de lanzamiento:** Se basa en la creación de versiones del producto; cada vez que se añade la funcionalidad que se desea para una versión se crea una copia de la rama principal, en la cual no se permitirá añadir nueva funcionalidad, aunque si corregir errores. Si se desea añadir más funcionalidad al producto se realizará en la rama principal para que posteriormente se crea la rama de versión correspondiente.
- **Rama de madurez:** Consiste en crear varias ramas que indiquen el estado de madurez de un producto y en ellas se indica en que versión del producto se encuentra dicho estado de madurez. Cuando una versión acaba una etapa de flujo de desarrollo se copia su contenido en la etapa posterior sin unir las ramas y se indica la versión que es.
- **Subdivisión de medio Ambiente:** Consiste en crear ramas a partir de puntos de versión seleccionados en los cuales se modifican manualmente algunos valores para permitir el empleo del código en diferentes entornos.
- **Rama de revisión:** Cuando aparece un error grave se crea una rama desde la última versión publicada, se realiza la revisión y corrección del error.
- **Tren de lanzamiento:** Se establecen periodos de entrega a partir de los cuales se crean ramas de lanzamiento que no aceptan más funcionalidades solo correcciones.
- **Línea principal lista para lanzamiento:** Mantenga la línea principal lo suficientemente saludable para que la cabeza de la línea principal siempre se pueda poner directamente en producción.