

Mantenimiento del repositorio

Para la gestión del repositorio se decidió utilizar el One Piece Flow(OPF) como flujo de trabajo principal, donde cada equipo de trabajo se enfoca en una tarea específica apartada en una rama característica, enviando pieza por pieza a la siguiente etapa del proceso de ingeniería.

De esta manera, se saca provecho de las ventajas que conlleva dicho workflow tales como reducir los tiempos de entrega, evitar encontrar bugs de forma tardía y contribuye a realizar un análisis más detallado del proceso de ingeniería, entre otras.

- **Versionado:** Se ha utilizado Github como herramienta para el control de versiones. Como estrategia de desarrollo se usa Trunk Based Development que consiste en trabajar en una rama principal y realizar integraciones frecuentes en lugar de trabajar en ramas separadas durante mucho tiempo. En la siguiente sección se entra en mayor detalle al respecto.
- **Ramas:**
 - La rama principal que se despliega a producción se denomina “main” y siempre conlleva versiones estables
 - La rama principal en el desarrollo es develop, se utiliza como intermediaria entre la rama principal main (trunk) y las demás ramas previo al merge con main con el fin de mantener el código estable. Se trata de que siempre quede pusheada una version estable.
 - Las ramas de características son las que se utilizan para nuevas funcionalidades o mejoras. Son las ramas temporales y específicas que se mergean a develop cuando se hayan testeado y estén completas. Se denominan de la siguiente manera: “feature/<nombreDeLaFeature>”.
 - Las ramas respectivas a correcciones de bugs se nombran “bugfix/<nombreDelBug>”. También son temporales y se mergean a develop.
 - Las ramas relacionadas a los realeases se denominan “release/<nombreDeLaVersion>” y se usan cuando hay una nueva versión para lanzar a producción. Estas ramas se crean a partir de main.
 - Las ramas sobre mejoras en el código o convenciones se denominan “refactor/<nombreDelRefactor>”

Para poder seguir buenas prácticas, se abre una rama particular por cada issue sea un bug o feature en la que se trabajará. Al final del día de trabajo se debe mergear dicha rama a develop para cumplir con el desarrollo basado en Trunk.

- **Merge:** Acorde a Trunk Based Development, los merges de código se realizan de la siguiente manera:
 - Todas las ramas más pequeñas, es decir las relacionadas a funcionalidades, mejoras, bugs se fusionan con develop.
 - Develop se fusiona con main únicamente
 - Main se mergea con release.

- Los merges se realizan mediante un pull request, el cual para ser aceptado debe cumplir con los estándares especificados en la documentación, pasar la cobertura de código correspondiente y luego ser aprobado por un integrante del equipo diferente a quien abrió la pull request. De esta manera podemos hacer una buena revisión entre pares y chequear que el código se mantenga estable.
- **Elementos:** Se han utilizado herramientas de Github que contribuyen a la integración continua para automatizar la construcción, pruebas y distribución entre el proceso.
 - Github Board: Con el fin de organizar el trabajo y poder seguir el progreso del proyecto se usa el board. Cada columna es una etapa del flujo de trabajo desde que entra como un requerimiento hasta que se completa.
 - Github actions: Sirven para automatizar el trabajo en el correr del flujo. Se configura para ejecutarse de manera automática en respuesta a eventos específicos.
 - Github issues: Se utilizan para registrar las mejoras o bugs que se detectan en la versión más estable del proyecto. Se hace uso de tags con el fin de categorizar los issues, priorizar y poder tener un mejor seguimiento de los mismos.