

# Práctica Formativa Obligatoria N° 1

En esta práctica vamos a centrarnos en el flujo de trabajo colaborativo del control de versiones generando ramas(branching), mezclandolas(merging) y generando y resolviendo conflictos del flujo de trabajo.

## Objetivos de la práctica

- Entender cómo se crean ramas de Git y se realiza un switch entre las mismas
- Practicar el desarrollo en paralelo entre ramas separadas
- Identificar y resolver conflictos en el merge
- Entender la importancia de los mensajes claros en los commit
- Familiarizarse con el flujo de trabajo y los pull request

Para ello vamos a generar un escenario de trabajo donde varios desarrolladores trabajan sobre un mismo proyecto para desarrollar una nueva característica de un sistema.

## Pasos de realización

- 1) Crear y configurar un repositorio
  - a) Crear un nuevo repositorio en github
  - b) Inicializar el repositorio con un archivo index.html con un contenido básico a elección y un archivo Readme.md con el título Practica Formativa y los apellidos de los participantes
  - c) Realizar un commit de este contenido en la rama principal
- 2) Crear una rama de la nueva característica
  - a) Un participante va a crear la rama feature/perfil-nuevo-ui
  - b) El otro participante va a generar la rama feature/perfil-nuevo-backend
- 3) Desarrollo independiente de ramas
  - a) Participante(UI)
    - i) Modificar el archivo index.html, agregando un form para el ingreso de los datos personales de un usuario incluyendo nombre de usuario y contraseña
    - ii) Realizar una llamada ajax o similar para enviar los datos al backend
    - iii) Documentar la llamada con los parámetros que se le envía
    - iv) Comitear los cambios con el mensaje "Agregado de datos personales"
  - b) Participante B (backend)
    - i) Crear un archivo php donde se reciban los datos enviados por frontend
    - ii) Realizar una verificación de backend de los datos recibidos
    - iii) Devolver una respuesta al frontend (ok o error)
    - iv) Comitear los cambios realizados

- 4) Simulación de conflicto
  - a) Participante(UI)
    - i) Modifica el archivo index.html agregando algún placeholder y modificando el nombre de alguno de los elementos
    - ii) Commitear los cambios realizados
  - b) Participante B (backend)
    - i) Modifica el archivo index.html agregando algún placeholder y modificando el nombre de alguno de los elementos
    - ii) Commitear los cambios realizados
- 5) Merging y resolución de conflictos
  - a) Participante(UI)
    - i) Intenta hacer un merge a main de su rama
  - b) Participante B (backend)
    - i) Intenta hacer un merge de su rama
  - c) En este caso git deberá informar de un conflicto
  - d) Ambos deben generar un issue simulando el informe del conflicto
  - e) Ambos deberán acordar las modificaciones elegidas de las dos ramas respondiendo a los issues generados
  - f) Remover los conflictos y realizar un commit 0
- 6) Push y pull request
  - a) Ambos deben pushear cambios al repositorio remoto
  - b) Ambos deben crear pull request para mergear sus ramas a main
  - c) Se deben aprobar las pull request (simulando como si un equipo en una reunión las aprobará)
  - d) **Nota:** los pull request se realizan directamente desde github al igual que los issues
- 7) Limpieza (Clean up)
  - a) Una vez hechos los merge a main se deben eliminar las dos ramas del repositorio.

## Entrega de la Práctica

- ☐ Se debe entregar un documento con los comandos ejecutados de git y los reportes de conflicto.
- ☐ En el documento se deben agregar las experiencias que se obtuvieron al realizarla.
- ☐ *Se debe incluir en el documento los problemas que puedan haber surgido al realizar la práctica ya que forman parte de la misma.*
- ☐ Se deben agregar al documento los issues generados y sus respuestas
- ☐ Se me deberá invitar al repositorio remoto(github) en el que se trabaje.
- ☐ No se analiza la parte de código tanto de backend como de frontend ya que no es el objetivo de la práctica

