

# pontIA

## Ejercicio Práctico - Módulo 1: Versionado de Código y Colaboración

### Introducción

El control de versiones es una habilidad esencial en el desarrollo de software moderno, y git es una de las herramientas más utilizadas para gestionar código de forma eficiente, segura y colaborativa. En esta práctica, tendrás la oportunidad de afianzar los conocimientos adquiridos durante el módulo a través de un ejercicio práctico que te llevará a realizar diversas operaciones fundamentales en git y GitHub: creación y configuración de repositorios, trabajo con ramas, gestión de conflictos, y automatización de tareas con GitHub Actions.

Este ejercicio está diseñado para simular un flujo de trabajo real en un entorno de desarrollo colaborativo, por lo que también se te pedirá configurar protecciones en la rama principal, trabajar con Pull Requests y utilizar acciones automatizadas.

### Forma de evaluación

La evaluación se realizará en función de los siguientes criterios:

- Correcta creación y configuración del repositorio git y su publicación en GitHub.
- Correcto uso del archivo .gitignore.
- Uso adecuado de ramas, commits descriptivos y mensajes de Pull Requests.
- Configuración del repositorio remoto para proteger la rama principal.
- Resolución de conflictos de manera adecuada.
- Implementación funcional de la acción de GitHub solicitada.
- Entrega puntual del ejercicio y cumplimiento de todas las instrucciones.

### Ejercicio práctico

A continuación se detallan los pasos que debes seguir para completar el ejercicio:

1. Crear el repositorio.

- Crea un repositorio en el que en el primer commit añades un archivo *README.md* con tu nombre completo. Publica el repositorio en GitHub con ese primer commit.
2. Configurar el repositorio remoto.  
Protege la rama main del repositorio remoto, de forma que no se puedan hacer commits directamente en ella y que todos los cambios sean a través de Pull Requests.  
Además, añádeme como colaborador del repositorio, de forma que pueda ver los cambios. Para hacerlo, en la página del repositorio navega por *Settings > Collaborators > Add people* y añádeme cómo colaborador con el email *ivangmasir@gmail.com*.
  3. Añadir un *.gitignore*.  
Crea una nueva rama en la que añadir un archivo *.gitignore* que ignore todos los archivos que tengan extensión *.log* del repositorio (no solo de la raíz del repositorio, sino dentro de cualquier directorio).  
Sube la rama a GitHub y abre un Pull Request a main con estos cambios. Explica brevemente en la descripción por qué esta configuración del *.gitignore* es correcta y cuando esté todo listo mergea la rama a main. Actualiza tu rama main en local con los nuevos cambios.
  4. Ramas de desarrollo.  
Asegúrate de tener la rama main en local actualizada y en ese punto crea dos ramas: **feature/suma** y **feature/resta**. En cada rama, crea un archivo *Operations.py* que contenga la definición de la función suma (*def sum*) en la rama **feature/suma** y la definición de la función resta (*def sub*) en la rama **feature/resta**. Sube ambas ramas a GitHub y abre un Pull Request de cada una a main.
  5. Simular y resolver un conflicto.  
Acepta el Pull Request de **feature/suma** a main y haz el merge y comprueba cómo en el Pull Request de **feature/resta** aparece un conflicto para el merge. Resuelve el conflicto en la rama **feature/resta** en local. Para ello, recuerda actualizar tu rama main en local y hacer un merge de la rama main en la rama **feature/resta** para actualizarla con los nuevos cambios que se han mergeado anteriormente.  
En ese momento tendrás que resolver el conflicto en el fichero *Operations.py* para incluir las dos definiciones. Haz un commit y sube los cambios a la rama **feature/resta**.  
Una vez resuelto, completa el merge de **feature/resta** a main.
  6. GitHub Actions.  
Actualiza la rama main con los nuevos cambios y crea una nueva rama **feature/ci** desde ese punto.  
En esta rama, añade un archivo *demo.yml* en el directorio *.github/workflows* en el que implementes un GitHub Action que, al ejecutarse manualmente (como *workflow\_dispatch*), imprima tu nombre completo ejecutando el comando *echo "<Tu Nombre Completo>"*.  
Sube tus cambios al repositorio cuando estén listos y crea un Pull Request de la rama **feature/ci** a main. Mergea cuando esté listo.  
Finalmente, comprueba que puedes ejecutar manualmente el GitHub Action desde la pestaña de Actions del repositorio, y que la salida es la esperada.
  7. Entrega.  
Puedes añadir cualquier comentario que quieras hacerme sobre la entrega en el archivo *README.md*.

Recuerda haberme añadido como colaborador del repositorio, de forma que pueda acceder a él.

Completa la entrega añadiendo el enlace al repositorio de GitHub en la entrega habilitada en el campus.

## Instrucciones de entrega

Envíame el link con el repositorio de GitHub a través de la entrega habilitada en el campus, tienes hasta el día 19 de octubre. Asegúrate de haberme añadido como colaborador del repositorio GitHub (a mi email [ivangmasir@gmail.com](mailto:ivangmasir@gmail.com) ), de forma que pueda acceder correctamente a él.