Guía de Aprendizaje: Uso de Git y GitHub en el Desarrollo de Aplicaciones

# 🎯 Objetivos de la clase

- Comprender los conceptos básicos de control de versiones.  
- Configurar Git en el equipo de desarrollo.  
- Crear repositorios y gestionar cambios en un proyecto.  
- Trabajar de forma colaborativa usando GitHub.

# 1. Introducción a Git y GitHub

1. ¿Qué es Git?

- Sistema de control de versiones distribuido.  
- Permite llevar historial de cambios, regresar a versiones anteriores y trabajar en paralelo.

2. ¿Qué es GitHub?

- Plataforma en la nube para alojar repositorios Git.  
- Facilita la colaboración, revisión de código y despliegue de proyectos.

# 2. Configuración inicial de Git

1. Descargar e instalar Git: https://git-scm.com  
2. Configurar usuario y correo:  
 git config --global user.name "Tu Nombre"  
 git config --global user.email "tuemail@ejemplo.com"  
3. Verificar instalación:  
 git --version

# 3. Flujo de trabajo básico con Git

1. Crear repositorio local:  
 git init  
  
2. Agregar archivo:  
 git add archivo.txt  
  
3. Guardar cambios (commit):  
 git commit -m "Primer commit"  
  
4. Ver historial:  
 git log --oneline

# 4. Conexión con GitHub

1. Crear cuenta en https://github.com  
2. Crear un nuevo repositorio.  
3. Conectar repositorio local con GitHub:  
 git remote add origin https://github.com/usuario/proyecto.git  
 git branch -M main  
 git push -u origin main

# 5. Trabajo colaborativo

- Clonar repositorio:  
 git clone https://github.com/usuario/proyecto.git  
  
- Crear nueva rama (branch):  
 git checkout -b nueva-funcionalidad  
  
- Fusionar cambios (merge):  
 git checkout main  
 git merge nueva-funcionalidad  
  
- Resolver conflictos: editar archivos, luego:  
 git add .  
 git commit -m "Conflicto resuelto"

# 6. Ejercicio aplicado al proyecto del curso

Cada grupo de estudiantes debe:  
1. Crear un repositorio en GitHub con el nombre de su proyecto.  
2. Subir la versión inicial del proyecto (con los trabajos de las calases anteriores).  
3. Crear una rama llamada 'desarrollo'.  
4. Hacer cambios en la rama 'desarrollo' y luego fusionarlos con 'main'.  
5. Invitar a los compañeros del grupo como colaboradores en GitHub.

# 7. Rúbrica de evaluación

✅ Repositorio inicial creado y configurado (20%).  
✅ Uso de commits claros y descriptivos (20%).  
✅ Aplicación de ramas y fusiones (30%).  
✅ Colaboración en GitHub (20%).  
✅ Presentación final del flujo de trabajo (10%).

# 8. Recursos adicionales

- Documentación oficial Git: https://git-scm.com/doc  
- Documentación GitHub: https://docs.github.com  
- Tutorial práctico: https://learngitbranching.js.org