Proceso de Desarrollo de Software

Andrés Mauricio Martínez Hincapié

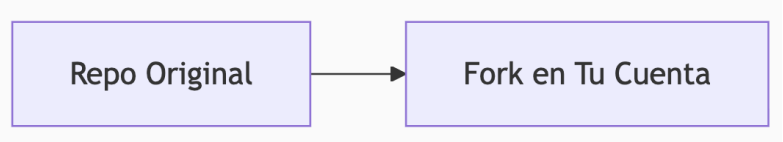
Taller Práctico: "Colaboración en Git (Fork, PRs y Convenciones)"

Objetivo: Aprender a contribuir a proyectos open-source mediante forks, pull requests y convenciones profesionales.

Paso 2: Flujo de Trabajo (Estudiantes)

A. Hacer Fork y Clonar

1. Cada estudiante hace **fork** del repo (botón "Fork" en GitHub).
   Paso 1: Hacer Fork
   a. Navega al repo original (https://github.com/amartinezh/colab-git-ucp).
   b. Haz clic en "Fork" (arriba a la derecha).
   c. Elige tu cuenta como destino.



2. Clona su fork localmente:

```
git clone https://github.com/amartinezh/colab-git-ucp

cd colab-git-ucp
```

*Tu fork **no se actualiza automáticamente** cuando el original cambia (debes sincronizarlo manualmente).*

B. Crear Rama y Editar

1. Rama con convención (pon tu nombre seg):

```
git checkout -b feat/ana-lopez
```

2. Editar `README.md`:

```
- [x] Ana López  # Reemplazar "[ ]" por "[x]"
```

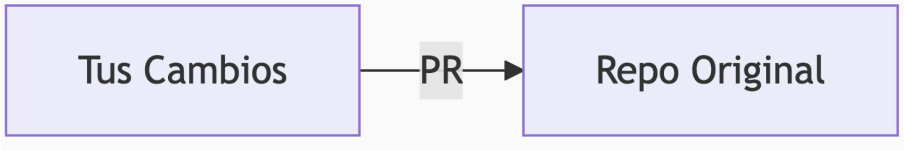
C. Commit y Push

```
git add README.md
git commit -m "feat: Add Ana López to participants"
git push origin feat/ana-lopez
```

D. Abrir Pull Request (PR)

1. En GitHub, ir a "Pull requests" → "New PR".
2. **Título**: `feat: Add Ana López to participants`.
3. **Descripción**:

```
- [x] Completed task: Add my name.
- **Checklist**:
  - [ ] Followed branch naming.
  - [ ] Used conventional commit.
```



Paso 3: Revisión entre Pares

- **Cada PR debe ser revisado por 2 compañeros**:
  1. Verificar que:
     - El nombre esté correctamente añadido.
     - El título del PR siga las convenciones.
  2. Aprobar con ✅ o pedir cambios con comentarios.

Paso 4: Merge y Sincronización

1. **Maintainer (profesor)**: Mergea PRs aprobados.
2. **Estudiantes**: Actualizan su fork:

```
git remote add upstream https://github.com/amartinezh/colab-git-ucp

git pull upstream main
```

El comando git remote add upstream seguido de git pull upstream main se usa para sincronizar tu fork local con el repositorio original (actualizado por otros colaboradores).

`git remote add upstream [URL]`

- **Qué hace**:
  - Añade una referencia al repositorio **original** (que hiciste fork) bajo el nombre `upstream`.
  - Esto te permite traer cambios nuevos del repo original a tu copia local.
- Sin esto, solo podrías acceder a tu fork (`origin`), no a las actualizaciones del proyecto principal.

`git pull upstream main`

- **Qué hace**:
  - Descarga los cambios más recientes de la rama `main` del repo original (`upstream`) y los fusiona con tu rama local actual.
- **¿Por qué usarlo?**
  - Evita que tu fork se quede obsoleto respecto al proyecto original.
  - Previene conflictos al hacer nuevos PRs.

Consejos para Éxito

- **Commits atómicos**: Un cambio por commit.
- **Títulos claros**: Ej: `fix: Correct typo in README`.
- **Sincroniza frecuentemente**: `git pull upstream main`.

Diagrama del Flujo

```
Repo Original (Profesor)
  ├── forks/estudiante1 → PR → original/main
  ├── forks/estudiante2 → PR → original/main
  └── ... (14 forks)
```

⚠️ Solución de Problemas

- **Error 403 al hacer push**:

  ```
  git remote set-url origin
  https://[token]@github.com/[usuario]/colab-git-ucp.git
  ```

- **Conflicto en README.md**:

  ```
  git pull upstream main
  # Resolver conflictos manualmente
  git add README.md
  ```

- `git commit -m "fix: Resolve merge conflict"`