

Práctica GitHub: Trabajando de forma colaborativa en GitHub

En esta práctica trataremos de simular una situación real de trabajo con repositorios remotos (GitHub). Para ello vamos a trabajar en equipos de **3 personas**, es decir vamos a hacer que estamos trabajando en un equipo de 3 desarrolladores y todos estamos trabajando sobre un mismo repositorio remoto alojado en GitHub.

Para desarrollar esta práctica hay que hacer lo siguiente:

Parte 1:

1) Formar los equipos de 3 desarrolladores (si hay algún alumno/a descolgado se podrá trabajar de cuatro de forma excepcional).

2) Cada equipo tendrá que **decidir** un **proyecto** sobre el que trabajar:

No tiene que ser un proyecto complicado, de hecho pueden elegir algo que ya hayan trabajado. Eso sí, se debe partir desde cero para ir viendo los distintos commits de cada desarrollador.

3) Una vez se haya decidido el proyecto, cada equipo de desarrollo tendrá que **dividirse** el **trabajo** y **asignar** una parte del mismo a cada desarrollador.

4) Llegados a este punto, **un miembro** del equipo tendrá que **crear** un **repositorio** en **GitHub**, añadir los primeros ficheros para el desarrollo del proyecto y añadir a sus compañeros como **colaboradores** en el repositorio. De entre estos ficheros, se deberá incluir un fichero **README.md** cuyo contenido será una explicación de en qué consiste el proyecto.

Parte 2:

Una vez hayamos completado estos 4 puntos, los equipos podrán empezar a desarrollar su proyecto. Para ello:

- Cada desarrollador tiene que clonar el repositorio en local y comenzar a desarrollar su parte.
- Los desarrolladores deberán trabajar sobre el mismo archivo (para generar conflictos futuros).
- Cada desarrollador irá completando su parte de proyecto y confirmando (commit) sus cambios tanto en local, como al repositorio remoto (es decir hacer push de cuando en cuando).

Lo que se pide en esta práctica es que se trabaje de forma **colaborativa** y se resuelvan como equipo los **conflictos** que vayan ir surgiendo.

Se debe ir documentando los pasos que se han seguido (los conflictos, como se han resuelto, etc.) en una **memoria** (con las capturas que crean convenientes, etc.). El **formato de la memoria** se explicará al final de este documento, sección: “**Estructura de la memoria y contenido**”).

Se da libertad a cada equipo para que trabaje como quiera, pero se deben **forzar al menos** las dos siguientes situaciones:

Situación 1:

Trabajar directamente desde la rama principal (**master**). Es decir, que cada miembro del equipo haga un clone del repositorio y comience a trabajar sobre la copia de la master.

Cada miembro del equipo tiene que trabajar en su parte del proyecto, ir haciendo commits en local y hacer algún push al repositorio remoto. En este punto tienen que

forzar algún conflicto y ver como o solucionan (al estar trabajando 3 desarrolladores sobre el mismo repo y la misma rama no es raro que se den conflictos).

(En la memoria tienen que documentar el conflicto que ha ocurrido y la solución que han dado).

Situación 2:

Una vez resueltos los conflictos y haber actualizado la rama master de la situación 1. Se pide **crear** en el repositorio **remoto una rama por cada miembro del equipo**, como si fuese una rama feature (se debe incluir en el nombre de la rama, el nombre del desarrollador para tener claro cuál es la de cada uno).

La idea de esta situación es que cada desarrollador/a sincronice su rama remota con su repo local (lo hemos visto en la **práctica anterior, ver apartado 5.5** si no te acuerdas) y que trabajen de forma paralela cada uno en su rama. Cada desarrollador deberán ir haciendo push a su rama remota.

Cuando ya tengan algo de trabajo hecho y actualizada su branch remota, cada desarrollador deberá hacer un **pull request** (desde github) para fusionar su rama a la rama master. Cuando lleguen a este punto de las pullrequest es conveniente que los miembros del equipo se pongan de acuerdo para que sea **uno el que haga los merges y acepte la pullrequest**.

(En la memorias tienen que documentar el conflicto que ha ocurrido y la solución que han dado).

Después de haber forzado estas dos situaciones, el equipo deber seguir y **terminar** el **proyecto** con la **estrategia de branching** (git flow) que crean más conveniente.

Parte 3:

La **semana siguiente**, cada equipo deberá **presentar** su memoria de prácticas al resto de la clase (aproximadamente **15 minutos** de presentación) donde se muestre la experiencia del trabajo colaborativo con GitHub.

La idea es que comenten su experiencia, los conflictos que tuvieron, como lo resolvieron, su estrategia de branching (git flow), etc. Y que enseñen el repositorio remoto al resto de compañeros (las ramas, los comitts, pullrequests... en definitiva lo que han hecho).

Estructura de la memoria y contenido:

En la **memoria** se tiene que incluir:

- Nombre y apellidos de los miembros del equipo.
- Proyecto en el que han decidido trabajar.
- Descripción del proyecto (en qué consiste, lenguaje en que está implementado, etc)
- Asignación de tareas/partes del proyecto a cada miembro del equipo (que quede bien definido quién hace qué parte/s).
- URL del repositorio en GitHub donde han trabajado.
- Documentar el proceso de trabajo que han seguido (donde se incluya las al menos las **2 situaciones** que han descrito, otras situaciones que se hayan dado, estrategia de branching que han seguido, etc. Con capturas de pantalla que crean convenientes para que se siga la memoria).

IMPORTANTE:

Para la entrega de esta práctica se tiene que subir **un solo fichero PDF** que contenga:

- **Memoria** de la prácticas con el contenido que se especifica en este documento. La memoria se ha de subir en formato **.PDF**.

El **nombre** de la memoria debe seguir el siguiente formato:

UT4_Practica_7_NombresMiembrosEquipos.pdf

(donde NombresMiembrosEquipos debe ser el nombre de los integrantes del grupo, no hace falta apellido para que no tenga nombre muy largo el documento)