

La entrega de esta práctica se realizará mediante la elaboración de un manual con las respuestas, explicando cada apartado y realizando capturas siempre que sea necesario para complementar la explicación de cada solución.

El documento se entregará en PDF.

Trabajando con Git y GitHub

Vamos a partir de un supuesto práctico. Supongamos que estamos en una empresa en la cual nos han mandado realizar una página web, somos el jefe de proyecto (desarrollador 1) y trabajamos junto a otro desarrollador (desarrollador 2).

El trabajo lo hemos dividido de forma que el el desarrollador 1 se encargará de realizar el HTML y el desarrollador 2 realizará el CSS, además, el desarrollador 1 se encargará de llevar a master las funcionalidades realizadas a través de PULL REQUESTS, por lo tanto:

- Prepara el entorno de ambos desarrolladores creando un nuevo repositorio en GitHub (como hicimos con el tutorial, el desarrollador 1 es quien trabaja en el equipo anfitrión y el desarrollador 2 trabaja en la máquina virtual).
 - El entorno debe ser como lo vimos en el tutorial, con la rama develop, añadiendo como colaborador al desarrollador 1, etc...
- Deberás realizar **2 funcionalidades** que serán llevadas a “master” una vez finalizadas. Cada funcionalidad deberá estar dividida en 3 subtareas de las cuales harás commit cuando las termines, ejemplo:
 - **Funcionalidad 1: creación de la página home:**
 - Desarrollador HTML:
 - Subtarea 1: definir la estructura del html
 - Subtarea 2: definir la cabecera y footer
 - Subtarea 3: definir el body
 - Desarrollador CSS:
 - Subtarea 1: definir la estructura del css
 - Subtarea 2: crear el css para la cabecera y el footer
 - Subtarea 3: crear css para el body
 - **Funcionalidad 2: creación de la página productos:**
 - Desarrollador HTML:
 - Subtarea 1: definir la estructura del html
 - Subtarea 2: definir la cabecera y footer
 - Subtarea 3: definir el body
 - Desarrollador CSS:
 - Subtarea 1: definir la estructura del css
 - Subtarea 2: crear el css para la cabecera y el footer
 - Subtarea 3: crear css para el body

- **ACLARACIONES:**

- El **tutorial** es esencial para hacer esta práctica.
- El trabajo es individual. Como hicimos en el tutorial, simularemos que somos dos trabajadores distintos.
- Las funcionalidades propuestas son un ejemplo, puedes hacerlas como tú quieras, pero siempre que se siga esa estructura mínima (2 funcionalidades mínimo y 3 subtareas mínimo por funcionalidad).
- La idea es seguir el GitFlow explicado en el tutorial de forma que el desarrollador 1 y 2 creen sus PROPIAS ramas para la primera funcionalidad, comiencen a desarrollar cada subtarea y vayan haciendo commits conforme las van terminando (1 commit por subtarea). Cuando terminan la funcionalidad, llevan los cambios a develop y cuando la funcionalidad completa está realizada (HTML + CSS) entonces se hace el PULL REQUEST para llevarla a master. Y se hace el mismo procedimiento con la funcionalidad 2.
 - **Importante:** Cada desarrollador debe desarrollar en su propia rama, es decir, no pueden estar modificando código varios desarrolladores en la misma rama.
- El contenido es indiferente, es decir, mi intención no es que desarrolléis una web, si no que trabajéis con GIT, por lo tanto, si en la subtarea 3, únicamente ponéis `<p>Este es el body</p>` es más que suficiente.

- **REQUERIMIENTOS**

- Utilizar el **GitFlow** explicado en el tutorial para el trabajo en equipo.
- **Documentar** todo lo que váis haciendo, es decir, tenéis que ir explicando con capturas lo que váis haciendo, se que es engorroso, pero necesito ver qué es lo que habéis realizado. El tutorial es un ejemplo de documentar algo y ver paso a paso qué se está haciendo, tómallo como referencia.
- **A parte de lo comentado arriba, necesito que hagáis también lo siguiente durante el desarrollo del proyecto:**
 - Resolver un **conflicto**, haz que en algún momento ambos desarrolladores modifiquen el mismo archivo y la misma línea, supongamos que uno de los desarrolladores se ha equivocado y ha modificado algo que no debe, lo cual ha provocado un conflicto al hacer merge.
 - En **.gitignore** deberás ignorar todos los archivos que contengan ".log" y todo el contenido de las carpetas que se llamen "cache" y "dev".
 - Cuando finalices, **elimina las ramas** utilizadas para cada funcionalidad, de esa forma, el proyecto quedará limpio para empezar una supuesta funcionalidad 3 (todas las ramas, las locales y las remotas), solo master y develop estarán disponibles.
 - *Investiga los comandos que necesitarás para hacer esto.*