

1. Vamos a crear el espacio de trabajo del bootcamp donde meteremos todas nuestras prácticas del curso de ahora en adelante. Para ello, forkead y clonad el repositorio en una nueva carpeta de su workspace (renombrar la carpeta actual que ya tienes a otro nombre antes de hacer el clone):

<https://github.com/JoseHervas/fullstack-bootcamp>

Una vez clonado para tenerlo en vuestro local. Haced “cd fullstack-bootcamp” para entrar en la carpeta y luego haced un “git status” para comprobar que todo está “up to date”.

2. Dentro de la carpeta “Contents” encontraréis una estructura de carpetas preparada con todas las secciones que daremos en el curso. De ahora en adelante, las prácticas que hagamos las iréis metiendo en estas carpetas.
3. En la terminal, haced “cd” a la carpeta Contents y echadle un ojo. Esa será la carpeta donde organizaremos todos los proyectos del bootcamp. Hay una estructura de subcarpetas.
4. Nos falta una carpeta para Git, así que crearéis (mkdir) una carpeta llamada “00-Git” y haréis cd a esa carpeta.
5. Dentro de esa carpeta pueden crear otra llamada “00-Git-Basics”, para que quede en ese formato: “fullstack-bootcamp/Contents/00-Git/00-Basic-Git”
6. Id a la carpeta “fullstack-bootcamp/Utilities/Gitignore” y copiar el archivo “front-end-example.txt”. Aquí tenéis un ejemplo de un gitignore típico de un proyecto de front-end. Pegadlo en la carpeta que acabáis de crear y cambiadle el nombre por “.gitignore”.
7. En la carpeta “fullstack-bootcamp/Utilities/Licenses” tenéis un ejemplo de una licencia GNU. Copiad ese archivo y pegadlo en vuestra nueva carpeta (fullstack-bootcamp/Contents/00-Git). Haced un git status para ver los cambios que acabáis de hacer. Luego guardad los cambios en el staging/index area.
8. Commitear los cambios con un mensaje como “first commit”. Luego pushearlos a vuestro repo remoto. Id a GitHub y comprobad que todo ha ido bien (vuestra nueva carpeta debería estar en vuestro repo remoto).
9. ¡Felicidades! Ya tenemos un gitignore y una licencia, sólo nos falta un README. Cread un archivo README.md y escribid algo. Volved a repetir el proceso anterior:
 - a. guardar los cambios en el staging/index area
 - b. Commitear los cambios con un mensaje cómo “added README”
 - c. pushear los cambios al repo remoto

10. Cread un archivo index.html muy sencillo con un texto que diga “Con GitHub nunca volveré a perder mi trabajo!”. Volved a repetir el proceso anterior:

- a. guardar los cambios en el staging/index area
- b. Commitear los cambios con un mensaje como “creada estructura HTML”
- c. pushear los cambios al repo remoto

11. Cread un archivo index.css y poned una animación al texto del html que haga cambiar el color de rojo a verde. Volved a repetir el proceso anterior:

- a. guardar los cambios en el staging/index area
- b. Commitear los cambios con un mensaje como “añadidos estilos básicos”
- c. pushear los cambios al repo remoto

12. Sí, ya sé que estamos repitiendo una y otra vez el mismo proceso, pero tiene un por qué: commitear no cuesta nada, y cuando más pequeños y concisos sean los commits, más fácil será mantener el orden en el historial de los cambios. Así que, por regla general, commitear cada pocos cambios en lugar de commitear pocas veces con muchos cambios es una buena práctica.

13. Leed este artículo sobre cómo hacer buenos READMEs para vuestros proyectos:

<https://medium.com/@meakaakka/a-beginners-guide-to-writing-a-kickass-readme-7ac01da88ab3>

Es posible que aún no entendáis algunos conceptos de los que hablan (sobre todo los más técnicos) pero no os detengáis en esos detalles y fijaros en la idea general del README. Con el tiempo los detalles técnicos os irán sonando menos a chino.

14. Ahora leed el manifiesto del 100DaysOfCode, del cual vamos a formar parte nosotros:

<https://medium.freecodecamp.org/join-the-100daysofcode-556ddb4579e4>

15. Ahora cambiamos de proyecto! Id al repo del 100DaysOfCode:

<https://github.com/Kallaway/100-days-of-code>

16. Clonadlo para tener el repo local, y luego crear el repo remoto y conectarlos.

17. Tomaros un ratito para escribir vuestra primera entrada editando el archivo “log.md”

18. Repetimos el proceso de siempre:

- a. guardar los cambios en el staging/index area
- b. Commitear los cambios con un mensaje como “añadidos estilos básicos”
- c. pushear los cambios al repo remoto