7 de Agosto, 2017

Día 1:

Profesoras: Estefanía y Belén

* Slackoverflow
* Github
* W3C

Temas de clases :

* HTML
* CSS
* BOOTSTRAP
* JS
* FRAMEWORKS JS
* (JQUERY, ANGULAR, REACT)
* JSON , AJAX

**Versionado**

\*Octocat (mascota de GitHub)

Repositorio: Carpetas que contienen el código fuente de un proyecto, y el resto de las carpetas. Cuando esta en mi maquina o en una sola maquina se llama repositorio local.

Dentro de un repositorio local: index.html / carpeta css / carpeta jpg / carpeta js / etc

Versión: conjunto de archivos a los que no quiero modificar mas. Se van a generar versiones a través de las diferentes modificaciones.

Git o bitbacket (repositorios remoto u online): Software de control de versiones. Sirve para guardar los códigos y compartirlos.

GitHub: red social de código (bueno mantenerla actualizada)

Gitclone: clonamos un repositorio online para poder trabajar de manera local.

Comandos GIT:

1. GIT pull
2. GIT status
3. GIT add
4. GIT commit
5. GIT push

8 de Agosto, 2017

Día 2:

Vinculación entre repo y nube

Descarga de cliente de git para poder hacer ese vínculo

Crear la clave o firma SCH

**Iniciación con Git**

Terminal

**git --version**

**ls (aparecen las carpetas)**

**cd carpeta para elegir**

**git clone URL (de github)**

**cd ../ (va a la carpeta anterior)**

**ls -la (muestra carpetas y archivos ocultos)**

Creamos nuevo repo en github

Nombre: prueba1

**Configuración de usuario git**

En terminal

**git config --global user.name “nombre”**

**git config --global user.email** [**nombre@mail.com**](mailto:nombre@mail.com)

**git config --list**

**Crear una carpeta y convertirla en un repo** con **git init** estando en la carpeta

Una vez que tenemos un repo podemos **vincularla con github**

**git remote add origin URL**

**git remote –v** para ver el fetch y puch o rutas

**Si agregamos un archivo cualquiera a la carpeta vinculada y apretamos git status**

El archivo aparece en rojo (no esta vinculado todavía)

Usando **git add nombre de archivo** y luego **git status** lo pasa a mostrar en verde como que **ya esta vinculado**

**También se puede subir todo por medio del git add --all**

Usando **git rm –cached nombre del archivo** nombre de archivo podemos **remover** el archivo si nos equivocamos.

**Git commit –m “comentario”** para **guardar**

**Para subir** al git hay que poner **git push origin master** y va a pedir usuario y contraseña

-## ¿Qué es y para qué sirve?

+![Git acaba con versiones finales](../assets/git/git.png)

+

+Git es un \*\*sistema de control de versiones\*\* distribuido, gratuito y de código abierto. Es decir, es un tipo de software pensando para hacer un seguimiento claro y ordenado de todos los cambios que vas haciendo en los archivos de un proyecto. Mejor todavía, te permite hacerlo en equipo, de forma colaborativa.

+

+Gracias al software de control de versiones, podés mantener tus archivos \*\*organizados, coordinados y protegidos\*\*, tanto de posibles eventos catastróficos como de los inevitables errores cometidos por tus compañeros de equipo (y bueno, también de los tuyos); si eso sucede es fácil solucionar el problema rescatando lo que haga falta de una versión anterior.

+

+Git nació en \*\*2005\*\*, después de que se venga abajo la relación entre la comunidad que desarrollaba el núcleo de Linux y la compañía responsable de BitKeeper, el sistema de control de versiones que venían usando hasta entonces. A estos muchachos (y en particular a Linus Torvalds, el creador de Linux) se les ocurrió que la mejor forma de reemplazarla era desarrollando una herramienta propia. Desde entonces Git ha logrado evolucionar manteniendo su \*\*diseño sencillo\*\* y su tremenda \*\*eficiencia\*\*, incluso para manejar proyectos muy grandes.

+

+### Lista comandos Git:

+

+ git init

+Indicamos que el directorio donde nos encontramos comenzará a ser versionado bajo Git.

+

+ git status

+Vemos el estado actual de nuestra carpeta versionada así como archivos que no están versionados, archivos modificados, archivos eliminados, etc. Los mensajes que podremos ver con este comando pueden ser los siguientes:

+Untracked files: nos mostrará todos los archivos que no han sido agregados al "Stage"

+Changes to be committed: nos indica qué archivos serán agregados a nuestro commit.

+

+ git add

+Agrega los archivos que no están bajo versionamiento (los Untracked files que vimos anteriormente).

+

+ git commit -m "mensaje"

+Confirma los cambios hechos con "git add" en una nueva versión del proyecto; con el parámetro -m le indicamos un mensaje para describir la razón del commit.

+

+ git remote add origin url\_del\_repositorio

+Hasta ahora todos los comandos visto han sido puro trabajo local, con el comando git remote add agregamos nuestro repositorio a un servidor remoto; para hacerlo debemos tener ya creado nuestro repositorio en un servidor y este nos brindará la url.

+

+ git push -u origin master

+Empujamos todo lo que tengamos en nuestros commits al repositorio remoto.

+

+ git pull

+Trae los cambios que haya trabajado otra persona en el mismo repositorio, es recomendable siempre que trabajemos un repositorio que no hayamos iniciado nosotros hacer siempre un git pull primero antes de comenzar a trabajar en los archivos.

+

+ git remote -v

+Nos muestra la url remota del repositorio en el que nos encontremos.

+

+ git clone url\_del\_repositorio nombre\_local

+Realiza una copia en nuestro directorio local de un repositorio existente.

+

+ git branch -all

+En Git se trabaja con branches (ramas en español), originalmente se crea la master por defecto, a raíz de esto nosotros podemos crear nuestras propias ramas, siendo claro una copia de la master en primera instancia. Con este comando podemos ver todas las ramas que tenga el proyecto.

+

+### Extras:

+

+Presentación hecha en Google -> [Git-Cheat Sheet](../assets/git/ppt/git-cheatSheet.pdf)

+

+### Referencias:

+

+La biblia -> [Progit](https://git-scm.com/book/es/v1)

+Link para favoritos-1 -> [Git - la guía sencilla](http://rogerdudler.github.io/git-guide/index.es.html)

+Link para favoritos-2 -> [Introducción a Github](http://www.cristalab.com/tutoriales/introduccion-a-github-en-linux-ubuntu-c106086l/)