

PRIMEROS PASOS EN GIT

Guillermo Julián Moreno Pedro Valero Mejía

9 de marzo de 2016

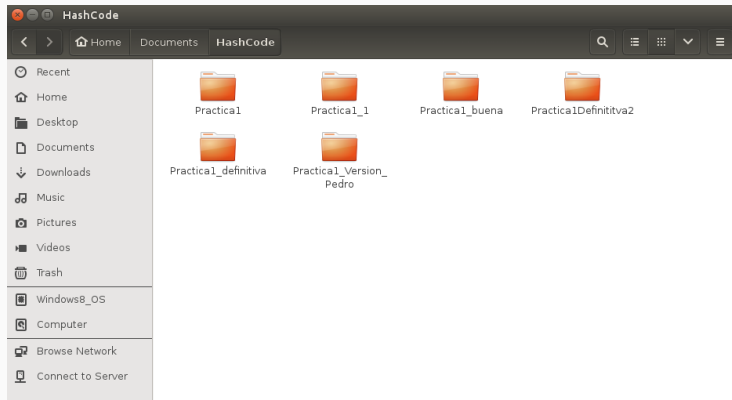
¿QUÉ ES GIT?

Es un software de control de versiones.

Básicamente permite que diversas personas trabajen de forma paralela sobre los mismos archivos evitando conflictos y pérdida de información (siempre y cuando no modifiquen exactamente la misma parte del texto)


1. Empleando dropbox, si dos compañeros nos bajamos el word del trabajo el segundo en subir sus cambios borra todo lo del primero. Git evita estos problemas.
2. Git nos permite mantener un histórico de **qué** cambios ha realizado cada persona y **cuándo**.
3. Por tanto podemos deshacer los cambios realizados en un cierto momento sin alterar el resto del proyecto.

Porque a todos nos ha pasado...





Y porque a todos nos gusta ver


Commits on Mar 6, 2016





Intento de ejercicio que acabó en fracaso
pevalme committed 3 hours ago


 2130754 

Commits on Mar 5, 2016





Añadidas nuevas entradas en el índice alfabético xxx
EduardoMiravalls committed a day ago


 d5b8c28 





Añadido algo de teoria al final a modo de resumen
pevalme committed a day ago


 a131ac9 

Commits on Mar 3, 2016






GT: Apuntes 03/03/2016
gjullianm committed 3 days ago

 77d9ac4 






AF: Apuntes 03/03/2016
gjullianm committed 3 days ago

 0144a6b 





Merge branch 'master' of github.com:VideJuan/Apuntes
AlbertoParramon committed 4 days ago


 f7c0888 





AC: al día
AlbertoParramon committed 4 days ago


 af26b50 

Commits on Mar 2, 2016






Ejercicio 6
pevalme committed 4 days ago

 176095f 






Revisados ejercicios de Hebb y casi todos los de perceptron
pevalme committed 4 days ago

 38bb419 





EDP: Algo
gjullianm committed 4 days ago

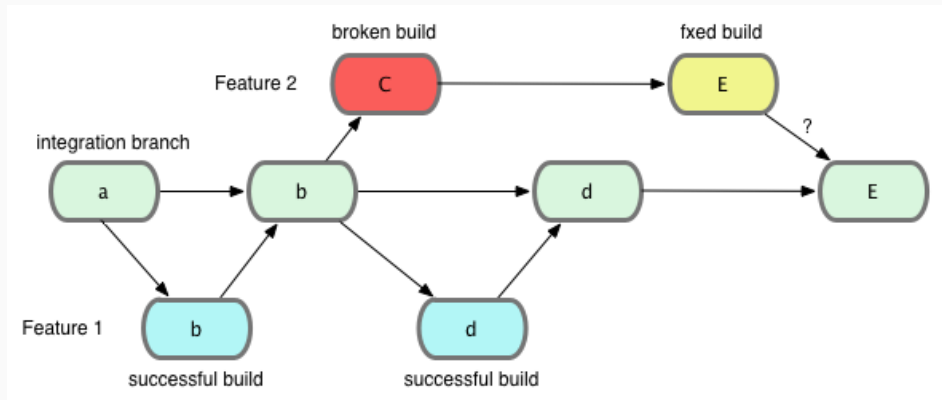
 f657ae7 



AF: Apuntes 02/03/2016
gjullianm committed 4 days ago

 85e7dc7 

¿CÓMO FUNCIONA GIT?



- Tenemos un proyecto en un servidor de Git. El proyecto se encuentra en el estado A .
- Antes de empezar a trabajar debemos “bajarnos” el estado actual del proyecto.
- Realizamos los cambios deseados hasta quedarnos satisfechos, llegando al estado B .
Llega el momento de subir los cambios.
 1. Cuando queremos actualizar los archivos del servidor, le decimos que el archivo que estaba en el estado A queremos que pase al estado B .
 2. Si el estado actual en el servidor es A , se actualiza con la versión B .
 3. Si el estado actual en el servidor es otro distinto, C , comprueba si los cambios que implicó C sobre A son conflictivos con los que implica B . En caso afirmativo te avisa. En caso negativo, actualiza C con lo que le faltase de B .

COMANDOS DE GIT

Vamos a ver los comandos más habituales en Git, aquellos que se usan *constantemente* en el día a día.

Estos cambios permiten llevar a cabo las acciones descritas en el apartado anterior así como descartar cambios realizados en local.

No obstante, como hemos dicho al principio, Git guarda el histórico de los cambios y pueden realizarse muchísimas operaciones sobre este historial. No entraremos en ello por ahora pero es realmente interesante. **Git tiene un enorme potencial más allá de estas transparencias**

1. **pull** Descarga los cambios del servidor a nuestra máquina.
2. **add** En local considera los cambios realizados en un fichero.
3. **commit** Prepara los cambios para ser subidos al repositorio. Permite además añadir comentarios.
4. **push** Sube los cambios al servidor.
5. **checkout** Descarta los cambios realizados en un fichero dejándolo en el estado en que se descargó.

Example

```
git pull
```

```
git add fichero1.c include/fichero1.h
```

```
git commit -m "Pila implementada"
```

```
git push
```

```
git checkout fichero2.c
```

CREAMOS UN REPOSITORIO
