

RECURSO DIDÁCTICO GIT-MASTERCLASS

Argentina Programa

Preparación Entorno Git:

1. Instalación de git -Acceder al sitio oficial <https://git-scm.com/>
2. Descargamos el instalador , y tiene que estar seleccionado hitbash que es una consola de git
3. También que este seleccionado la consola vin
4. Hay que decirle que use opciones de git con Unix.
5. Dejamos preseleccionadas las librerías de OpenSSL.
6. Para saber cual es el directorio en el cual estamos parados se usa pwd.
7. Ls para listar el contenido.

Intervención de Git a través de Master Class:

1. Crear una nueva carpeta en el escritorio.
2. Abrir code agregar esa carpeta , crear un archivo prueba1.html
3. Ir al escritorio posicionarse en la carpeta , boton derecho ejecutar bashgit.
4. Ls :listar los archivos,
5. pwd carpeta actual .

-----Incializacion de GIT-----

6. **Inicializar git con el comando `git init`** , (crea un proyecto git que pertenece a esa carpeta). Se inicializa el repositorio de git con respecto a ese directorio.

-----Vinculación de Cuenta -----

6. **Vincular a una cuenta y nombre de un desarrollador en particular**, para que en los commit aparezcan estos datos que son necesarios.
7. Con el **comando `git config --global user.email`** “maestriaev@gmail.com”

(configuramos la cuenta).

8. Con el comando **git config --global user.name "David alderete"**.

Entonces tenemos enlazado el repositorio con esta cuenta de mail y nombre de desarrollador .

-----Estado del proyecto-----

9. Vemos en que **estados** están los proyectos dentro del directorio, con el comando **status** verificamos el estado en el que están los proyectos en ese directorio. **git status**.

10. Y vemos que **existe un proyecto que no ha sido comiteado** y nos dice que deberíamos utilizar el comando **git add**.

-----Agregamos un archivo y commit-----

11. Ahora agregamos un archivo desde **nuestro repositorio local** que acabamos de crear. **git add prueba1.html**.

12. Aplicamos **git add prueba1.html** y nos muestra que aun no aplicamos **commit** (ver que cambia de color)

14. Aplicamos **nuestro primer punto d partida con el comando git commit** (primera foto que sirve para el historial) y nos abre una nueva consola y escribimos un mensaje. **para escribir el mensaje presionamos la tecla i** para insertar y escribe primer commit y presionamos escape.

15. **: wq** para volver a la consola de git y me informa de los cambios realizados.

16. Comando **git log** puedo ver los cambios que ocurrieron y nos da un **numero de commit para identificar el commit** .

-----Vincular Repositorio-----

17. (La idea es crear una rama asignarle el archivo prueba1.html y actualizar los cambios)

18. Ahora creamos un **repositorio remoto en github** y creamos un nuevo repositorio con nombre prueba y tiene que ser publico create repository.

19. **Traemos el proyecto al repositorio remoto** copiando el comando que tiene texto de remote y copiamos y relaciona el proyecto al repositorio remoto **git push -u origin master** y copia los archivos al repositorio remoto.

-----Creación de rama en hithub-----

20. Para empezar a trabajar en github creamos una rama y le colocamos el nombre developer **clic en create branch** y esta rama es una copia exacta de lo que tiene master.
21. Luego en la solapa code modifico el archivo prueba 1.html agregando un h1 commit changes.
22. Podemos hacer un **merge pull request** para agregar los nuevos cambios en el repositorio master. (del archivo prueba1.html Tildar donde dice Compare y pull request de github).
23. Además github nos muestra un mensaje de que puedes hacer una fusion entre una rama esta que actualizamos (developer) y la rama base (master).
24. Luego vamos a code, elegimos master y podemos visualizar que posee el archivo prueba1.html con los cambios.
-

-----Nuevo Desarrollador Clonar Repositorio-----

ME SUMO A UN GRUPO DE DESARROLLADORES Y ME COMPARTEN EL REPOSITORIO REMOTO PARA QUE PUEDA TRABAJAR LOCALMENTE

25. Vamos a proponernos como un **nuevo desarrollador** y arrancar de cero como si estuviéramos en otra ubicación geográfica. para eso **cerramos code y eliminamos el proyecto del escritorio porque clonaremos desde el repositorio remoto**
26. Vuelvo a hit hub voy a la rama master cambio a la rama de developer y **a clonar** a través de esta rama. En donde dice **clonar copio la dirección voy a la consola a la carpeta creada en el escritorio que esta vacía**.
27. Usar el comando **git clone** con la **dirección que he copiado**

-----Gestion de Ramas -----

28. Para ver que ramas tengo uso el comando **git branch** y me muestra que ramas tengo dentro de **mi repositorio local**.

29. Creo una rama nueva **git branch develop**.

30. Luego con **git branch** veo las ramas y la que cree no esta activa con **git checkout "develop"** cambio a la rama que quiero trabajar. verifico con **git branch**

-----Modificacion de Archivos-----

31. Entro al archivo Prueba1.html y le realizo cambio en studio code , luego voy a la consola **git status** y me muestra que se han realizado cambios en prueba1.html.

32. Agregamos al repositorio local con **git add .**

33. Luego ejecutamos **git commit -m "mensaje se vincule a nuestro commit"**

34.git status

35.git log

-----Union de Ramas-----

36. Luego hacer una union de la rama desarrollar a master

37. primero cambio a la rama master con **git checkout master**

38. **git merge develop** (cambios de desarrollados fueron fusionados a a rama master)

39. Todas estas modificaciones las subimos al github

40.git checkout master

41. **git push -u origin master** y ahi realiza los cambios en el repositorio remoto