

Prueba práctica git

Nombre y apellidos: Estefanía García Sánchez

Criterios de evaluación:

- Es capaz de crear repositorios locales
- Es capaz de realizar cambios entre el working área y staged área.
- Es capaz de realizar cambios entre el staged área y el commit área.
- Es capaz de gestionar el envío de información de repositorios locales a repositorios remotos
- Es capaz de gestionar el envío de información de repositorios remotos a repositorios locales
- Es capaz de gestionar ramas.
- Es capaz de gestionar conflictos.

Debes realizar una captura o introducir las órdenes correspondientes (asegúrate que copias bien la instrucción) para realizar las acciones que se solicitan en este fichero.

Nota: Cuando sea necesario hacer un commit debes poner un mensaje que explique claramente los cambios que has realizado en ese commit.


Ejercicio 1 – Debes tener preparado las variables de entorno básicas de git.


1. Crea un repositorio nuevo local llamado recuGit
 - mkdir recuGit
 - cd recuGit
 - git init
2. Extrae en este nuevo repositorio los ficheros que hay en el fichero comprimido que hay en la plataforma moodle en esta prueba
 - extraer los ficheros y copiarlos en la carpeta
3. Crea un repositorio privado en github llamado “recuGit”. Asígnale acceso a mi usuario (joadelvia) e indica aquí la url del repositorio.
 - <https://github.com/egarsan176/recuGit.git>


Manage access

Invite a collaborator

☐ Select all Type ▾

☐  **Joaquín Delhom Viana**
Awaiting joadelvia's response

Pending Invite 



< Previous Next >

4. Enlaza tu repositorio local con el repositorio remoto creado anteriormente.
 - git remote add origin <https://github.com/egarsan176/recuGit.git>

5. Realiza las funciones necesarias para que todos los ficheros que hemos creado al descomprimir, estén disponibles en el repositorio remoto.

- git add .
- git commit -m "añadidos los ficheros por primera vez"
- git branch -M main
- git push origin main

6. Crea un directorio llamado repositorios en tu directorio home (/home/estudiante) y dentro haz dos copias del repositorio remoto anterior de forma que queden totalmente configuradas para trabajar con el repositorio remoto llamadas "trabajoEnClase" y "trabajoEnCasa".

- cd ..
- mkdir repositorios
- git clone <https://github.com/egarsan176/recuGit.git> trabajoEnClase
- git clone <https://github.com/egarsan176/recuGit.git> trabajoEnCasa

Al final de este ejercicio deberás tener dos repositorios distintos pero con los mismo ficheros.

Ejercicio 2. Cambiate al directorio trabajoEnClase y realiza los siguientes ejercicios en él.

1. Crea una carpeta llamada “nuevosMuebles”
 - cd trabajoEnClase
 - mkdir nuevosMuebles
2. Realiza las acciones que creas necesarias para subir esta nueva carpeta al repositorio remoto que has creado.

```
estudiante@Estefana:~/repositorios/trabajoEnClase$ mkdir nuevosMuebles
estudiante@Estefana:~/repositorios/trabajoEnClase$ git status
On branch main
Your branch is up-to-date with 'origin/main'.

nothing to commit, working tree clean
estudiante@Estefana:~/repositorios/trabajoEnClase$
```

Como git no sigue carpetas vacías tenemos que crearle un fichero .keep para poder añadirla.

- touch nuevosMuebles/.keep
- git add nuevosMuebles.keep
- git commit -m “añadida la carpeta nuevosMuebles con fichero .keep”
- git push origin main

3. Crea un fichero llamado pendiente.txt con el contenido que quieras y haz las acciones que creas necesarias para que git no haga el seguimiento de este fichero.
 - nano pendiente.txt
 - mkdir .gitignore
 - mv pendiente.txt .gitignore

4. Renombra el fichero README.md por WEB.md, utiliza 'mv origen destino' o desde el entorno gráfico.
 - mv README.md WEB.md

5. Comprueba el estado del repositorio analizando los mensajes.

- git status

```
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    WEB.md

no changes added to commit (use "git add" and/or "git commit -a")
estudiante@Estefana:~/repositorios/trabajoEnClase$
```

Nos muestra que el cambio se ha producido en el fichero que se encuentra en el directorio de trabajo. Que para confirmarlo hagamos add y commit o para deshacer el cambio git restore.

6. Deshaz el cambio.

- git restore README.md
- como el archivo WEB sigue creado utilizo un rm WEB.md para borrarlo

7. Renombra el fichero README.md por WEB.md, utiliza 'git mv origen destino' o desde el entorno gráfico.

- git mv README.md WEB.md

8. Comprueba el estado del repositorio analizando los mensajes.

- git status

```
estudiante@Estefana:~/repositorios/trabajoEnClase$ git mv README.md WEB.md
estudiante@Estefana:~/repositorios/trabajoEnClase$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:    README.md -> WEB.md

estudiante@Estefana:~/repositorios/trabajoEnClase$
```

Nos muestra que el archivo eliminado se encuentra en el stage área en vez de en el directorio de trabajo como ocurría usando solo el comando mv.

9. ¿Qué diferencia hay entre las dos órdenes?

La diferencia es que con el comando mv los cambios se quedan en el directorio de trabajo mientras que si usamos git mv los cambios permanecen en el área de preparación.

10. Sube estos cambios al repositorio remoto

- git add .
- git commit -m "renombrado el archivo README.md por WEB.md"
- git push origin main

11. ## Ejercicio 3. Cambiate al repositorio trabajoEnCasa y **NO ACTUALICES EL REPOSITORIO**

- cd ..
- cd trabajoEnCasa

1. Introduce tu nombre y apellido en el fichero README.md y añade los cambios al staged area.

- nano README.md
- git add README.md

2. Deshaz los cambios

- git restore - -staged README.md
- git restore README.md

3. Introduce tu nombre y apellido en el fichero README.md y haz un commit con los cambios.

- nano README.md
- git commit -am "modificado el archivo README.md"

4. Deshaz los cambios

- git reset - -hard HEAD~1

5. Introduce tu nombre y apellido en el fichero README.md y haz un commit con los cambios.

- nano README.md
- git commit -am "modificado de nuevo el archivo README.md"

6. Deshaz los cambios dejando los cambios en el working area.

- git reset - -soft HEAD~1

7. Deshaz los cambios.

- git restore - -staged README.md
- git restore README.md

8. Añade una línea con tu nombre al fichero README.md y realiza las acciones que creas necesarias para subir estos cambios al repositorio remoto.

- nano README.md
- git commit -am "modificado otra vez el fichero README.md"
- git push origin main

```
estudiante@Estefana:~/repositorios/trabajoEnCasa$ git push origin main
Username for 'https://github.com': egarsan176
Password for 'https://egarsan176@github.com':
To https://github.com/egarsan176/recuGit.git
 ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'https://github.com/egarsan176/recuGit.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
estudiante@Estefana:~/repositorios/trabajoEnCasa$
```

Te avisa de que antes debes hacer un pull para poder hacer el push

- git pull origin main

```
estudiante@Estefana:~/repositorios/trabajoEnCasa$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 3 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:
  new file:   .gitignore/pendiente.txt
  renamed:    README.md -> WEB.md
  new file:   nuevosMuebles/.keep
```

- tenía cambios sin añadir por lo que he tenido que hacer un commit para luego poder hacer el push
- git push origin main

9. Olvidaste añadir el nombre de tu compañero, por lo que modifica el fichero README.md.

- creo que he tenido que resolver mal algún paso anterior porque el archivo README.md no se encuentra y solo tengo el WEB por lo que voy a modificar el archivo WEB.md

- nano WEB.md

10. Comprueba el estados del repositorio y las diferencias del fichero.

- git status

```
estudiante@Estefana:~/repositorios/trabajoEnCasa$ git status
On branch main
Your branch is up-to-date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   WEB.md

no changes added to commit (use "git add" and/or "git commit -a")
estudiante@Estefana:~/repositorios/trabajoEnCasa$
```

- git diff

```

estudiante@Estefana:~/repositorios/trabajoEnCasa$ git diff
diff --git a/WEB.md b/WEB.md
index 09dd6f3..aceb159 100644
--- a/WEB.md
+++ b/WEB.md
@@ -1,3 +1,4 @@
 # ProgramacionWeb
 Recuerden: Antes de hacer sus modificaciones, descargense las modificaciones de
 los demas para evitar perdida de contenido y cualquier otro conflicto.
 Estefanía García Sánchez
+María López Revilla
estudiante@Estefana:~/repositorios/trabajoEnCasa$

```

11. Añade los cambios al área de preparación y añade los cambios al commit anterior modificando el mensaje para que refleje también los nuevos cambios.

- git add .
- git commit - -amend -m "modificado el archivo WEB.md"

12. Muestra los commits realizados en formato de una línea y pega aquí el resultado.

- git log - -oneline

```

estudiante@Estefana:~/repositorios/trabajoEnCasa$ git log --oneline
eda7314 (HEAD -> main) modificado el archivo WEB.md
37893d8 modificado otra vez el fichero README.md
f5af3b8 renombrado el archivo README.md por WEB.md
169acbb intento fallido de usar gitignore
eac2a90 añadida la carpeta nuevosMuebles con fichero .keep
d56f208 añadidos los ficheros por primera vez
estudiante@Estefana:~/repositorios/trabajoEnCasa$

```

13. Envía los cambios al repositorio remoto

- git push origin main
- me dice que antes tengo que hacer un pull
- git pull origin main
- repito el push → git push origin main

Ejercicio 4. Vamos a trabajar en el repositorio de trabajoEnClase.

1. Actualiza el repositorio y analiza los mensajes

- git pull origin main


```

estudiante@Estefana:~/repositorios/trabajoEnClase$ git pull origin main
Username for 'https://github.com': egarsan176
Password for 'https://egarsan176@github.com':
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 6), reused 6 (delta 3), pack-reused 0
Unpacking objects: 100% (9/9), 998 bytes | 199.00 KiB/s, done.
From https://github.com/egarsan176/recuGit
 * branch                main                -> FETCH_HEAD
   f5af3b8..fbfe3ab      main                -> origin/main
Updating f5af3b8..fbfe3ab
Fast-forward
 WEB.md | 2 ++
 1 file changed, 2 insertions(+)
estudiante@Estefana:~/repositorios/trabajoEnClase$

```

Me muestra que un archivo ha cambiado (se muestra el archivo WEB porque el README ya no está”

2. Crea una rama llamada cambiandoCarpetas.

- git branch cambiandoCarpetas

3. Cámbiate a dicha rama.

- git checkout cambiandoCarpetas

4. Borra el fichero index.html, mira los cambios producidos en el repositorio.

- rm index.html

- git status

```

estudiante@Estefana:~/repositorios/trabajoEnClase$ rm index.html
estudiante@Estefana:~/repositorios/trabajoEnClase$ git status
On branch cambiandoCarpetas
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    index.html

no changes added to commit (use "git add" and/or "git commit -a")
estudiante@Estefana:~/repositorios/trabajoEnClase$

```

Nos dice que archivo borrado se encuentra en el directorio de trabajo. Para eliminarlo del todo hay que hacer add y commit.

5. Deshaz los cambios

- git restore index.html

6. Borra el fichero index.html con la orden “git rm index.html”.

- git rm index.html

- git status


```

estudiante@Estefana:~/repositorios/trabajoEnClase$ git rm index.html
rm 'index.html'
estudiante@Estefana:~/repositorios/trabajoEnClase$ git status
On branch cambiandoCarpetas
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    index.html

estudiante@Estefana:~/repositorios/trabajoEnClase$

```

Nos dice que el archivo eliminado está en el área de preparación y con un commit se confirma.

7. Mira los cambios producidos en el repositorio y explica las diferencias entre esta opción y la opción anterior.

Las diferencias son que el rm te deja el cambio en el working área por lo que para eliminarlos tienes q hacer un add y un commit, mientras que el comando git rm te deja el archivo en el stage área y para eliminarlo solo hay que hacer un commit.

8. Deshaz los cambios

- git restore - --staged index.html
- git restore index.html

9. Renombra la carpeta extranet a web

- mv extranet web

10. Entra en el fichero index.html y reemplaza la ocurrencia de extranet por web.

- nano index.html

11. Cámbiate a la rama main

- git checkout main

12. Crea un fichero llamado aboutus.html con la información que quieras sobre la empresa.

- nano aboutus.html
- git add .
- git commit -m "movido archivos y creado fichero aboutus.html"

13. Incorpora los cambios de la rama master a la rama cambiandoCarpetas

- git checkout cambiandoCarpetas
- git merge main

14. Muestra los cambios que se han ido produciendo en el repositorio de manera que se muestren las ramas gráficamente.

- git log - --oneline - --graph

```

estudiante@Estefana:~/repositorios/trabajoEnClase$ git log --oneline --graph
* 7809a25 (HEAD -> cambiandoCarpetas, main) movido archivos y creado fichero abo
utus.html
* fbfe3ab (origin/main, origin/HEAD) Merge branch 'main' of https://github.com
/egarsan176/recuGit into main
|\
| * 30cc131 añadidos los cambios del merge
| |\
| * | \ eda7314 modificado el archivo WEB.md
| |\ \
| | / \
| | / /
| | / /
| | / /
| | / /
| * f5af3b8 renombrado el archivo README.md por WEB.md
| * 169acbb intento fallido de usar gitignore
| * eac2a90 añadida la carpeta nuevosMuebles con fichero .keep
* | 37893d8 modificado otra vez el fichero README.md
|/
* d56f208 añadidos los ficheros por primera vez
estudiante@Estefana:~/repositorios/trabajoEnClase$

```

15. Añade algún contenido al fichero aboutus.html y sube los cambios al repositorio.

- nano aboutus.html
- git commit -am "añadido cambios en aboutus.html"

16. Incorpora los cambios de la rama cambiandoCarpetas a la rama main

- git checkout main
- git merge cambiandoCarpetas

17. Elimina la rama cambiandoCarpetas

- git branch -d cambiandoCarpetas

18. Sube los cambios al repositorio remoto.

- git push origin main

```

commit 69b55a52a6cc6f59f2a8d5ee17f11b8ff820a6b8 /clojure-koans/src/koans/10_lazy_sequences
Author: Steven Lu <stevenlu443@gmail.com>
Date: Sun Apr 7 04:48:20 2013 -0400

    working on the breakage

diff --git a/ply.js b/ply.js
index e22ca60..b0b2627 100644
--- a/ply.js
+++ b/ply.js
@@ -36,9 +36,9 @@ var PLY = (function ($data) {
    "use strict";

-   var datenow = DEBUG.datenow;
-   // var escapeHtml = DEBUG.escapeHtml;
-   var serialize = DEBUG.serialize;
+   // var datenow = DEBUG.datenow;
+   // var serialize = DEBUG.serialize;
+   // var isInDOM = DEBUG.isInDOM;

    // various parts of state of the library
@@ -46,7 +46,7 @@ var PLY = (function ($data) {
    var exposed = {
        // used by touchmove event to run code only when necessary
        // TODO: why is this public?
-       tmTime: datenow(),
+       tmTime: 100,

        // converges on the time it takes to run touchmove
        tmProfile: 3,
@@ -169,7 +169,7 @@ var PLY = (function ($data) {
        // for it is sent out!
        if (ep[x].hasOwnProperty('ni')) {
            assert($data(ep[x].e, 'ply'), "exists: data of element in pointer_state in
dexed "+x);
-            assert($data(ep[x].e, 'ply').t[x] === ep[x], "pointer_state["+x+"] is exa
ctly equal to the data of its e property: "+serialize(ep[x])+"; "+serialize($data(ep[x].e
, 'ply')));
+            assert($data(ep[x].e, 'ply').t[x] === ep[x], "pointer_state["+x+"] is exa
ctly equal to the data of its e property: "+(DEBUG?DEBUG.serialize(ep[x]):"")+"; "+(DEB
UG?DEBUG.serialize($data(ep[x].e, 'ply')):""));
            assert(ep[x].ni === $data(ep[x].e, 'ply').node_id, "node id check "+ep[x]
.ni+", "+$data(ep[x].e, 'ply').node_id);
            assert(en[ep[x].ni] === ep[x].e, "check element with id");
        }
    }
}

```

A partir de esta imagen, ¿Qué cambios puedes decir que se han producido en el repositorio?

```

ndbroadbent@ndb-desktop ~/src/ubuntu_config [master±|1.9.2p290]$ gs
# On branch: master | +2 | [*] => $e*
#
> Changes not staged for commit
#
#       deleted: [1] _shared.sh
#       modified: [2] assets/git_breeze/config.example.sh
#       modified: [3] assets/git_breeze/config.sh
#       modified: [4] assets/git_breeze/git_breeze.sh
#       modified: [5] assets/git_breeze/lib/aliases_and_bindings.sh
#       deleted: [6] assets/git_breeze/lib/git_file_shortcuts.sh
#       modified: [7] assets/git_breeze/lib/git_repo_management.sh
#       deleted: [8] assets/git_breeze/test/git_file_shortcuts_test.sh
#       modified: [9] assets/git_breeze/test/git_repo_management_test.sh
#       modified: [10] assets/git_breeze/test/support/test_helper
#
> Untracked files
#
#       untracked: [11] assets/git_breeze/install.sh
#       untracked: [12] assets/git_breeze/lib/fallback/
#       untracked: [13] assets/git_breeze/lib/git_status_shortcuts.rb
#       untracked: [14] assets/git_breeze/lib/git_status_shortcuts.sh
#       untracked: [15] assets/git_breeze/test/git_status_shortcuts_test.sh
#
ndbroadbent@ndb-desktop ~/src/ubuntu_config [master±|1.9.2p290]$ |

```

A partir de esta imagen, ¿Qué cambios puedes decir que se han producido en el repositorio?

Que existen varios archivos sin rastrear.

Que han eliminado y modificado algunos archivos no se encuentran todavía en el stage para hacer commit sino que están en el working área.