Práctico 2: Git y GitHub

Actividades:

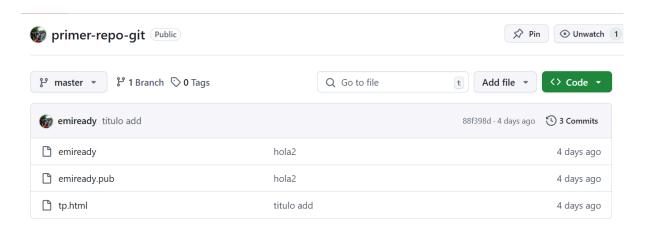
- 1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :
- ¿Qué es GitHub? Es un repositorio web para poder almacenar diferentes proyectos con git.
- ¿Cómo crear un repositorio en GitHub? Para crear un repositorio en github usamos GIT REMOTE ADD ORIGIN https://github.com/usuario/repositorio.git
- ¿Cómo crear una rama en Git? Para crear una rama en git usamos GIT BRANCH NOMBRE_RAMA
- ¿Cómo cambiar a una rama en Git? Para cambiar a una rama usamos GIT CHECKOUT RAMA_A_CAMBIAR
- ¿Cómo fusionar ramas en Git? Para fusionar ramas usamos GIT MERGE RAMA_QUEUNIMOS
- ¿Cómo crear un commit en Git? Para guardar el cambio es GIT COMMIT -M "Nombre_cambio_mensaje"
- ¿Cómo enviar un commit a GitHub? Para enviar al repositorio usamos GIT PUSH -U ORIGIN MAIN
- ¿Qué es un repositorio remoto? Es un repositorio interno de nuestro proyecto almacenada en un servidor como GITHUB, que se conecta con nuestro repositorio local para generar los cambios.
- ¿Cómo agregar un repositorio remoto a Git? GIT REMOTE ADD ORIGIN https://github.com/usuario/repositorio.git es la forma de agregar al repositorio vacio que creamos con GIT INIT
- ¿Cómo empujar cambios a un repositorio remoto? GIT PUSH es para empujar cambios.
- ¿Cómo tirar de cambios de un repositorio remoto? Con GIT PULL
- ¿Qué es un fork de repositorio? Es una copia de un repositorio creado en una cuenta diferente que permite desarrollar cambios sin afectar el original.
- ¿Cómo crear un fork de un repositorio? Usamos el boton FORK en GITHUB

¿Cómo enviar una solicitud de extracción (pull request) a un repositorio? Luego de hacer el FORK y los cambios, en la pestaña PULL REQUEST en el repositorio original, hacemos clic en NEW PULL REQUEST.

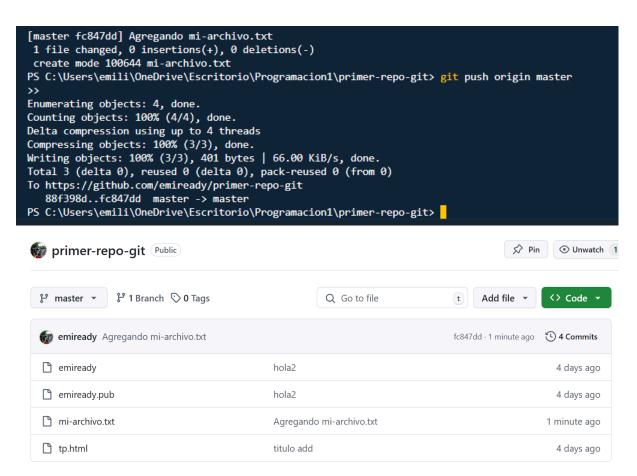
- ¿Cómo aceptar una solicitud de extracción? El propietario puede aceptar la solicitud desde su opción de PULL REQUEST, clickeando MERGE PULL REQUEST.
- ¿Qué es una etiqueta en Git? Es una referencia fija a un commit específico, para marcar versiones.
- ¿Cómo crear una etiqueta en Git? git tag v1.0
- ¿Cómo enviar una etiqueta a GitHub? Para enviar una etiqueta a GITHUB usamos GIT PUSH ORIGIN v1.0
- ¿Qué es un historial de Git? Es el registro de todos los commits realizados en un repositorio.
- ¿Cómo ver el historial de Git? Para verlo uso GIT LOG
- ¿Cómo buscar en el historial de Git? Para buscar uso GIT LOG --grep="palabra_clave"
- ¿Cómo borrar el historial de Git? Para borrar uso GIT REBASE -i
- ¿Qué es un repositorio privado en GitHub? Repositorio que solo se puede ingresar con una autorización del creador.
- ¿Cómo crear un repositorio privado en GitHub? Se elije en el sitio de GITHUB.
- ¿Cómo invitar a alguien a un repositorio privado en GitHub? En la opcion configuraciones, colaboradores agregas el usuario.
- ¿Qué es un repositorio público en GitHub? Es un repositorio libre para que ingrese cualquier persona en internet.
- ¿Cómo crear un repositorio público en GitHub? Lo mismo que privado, pero elegis publico.
- ¿Cómo compartir un repositorio público en GitHub? Se puede compartir por ejemplo compartiendo la URL del mismo.

2) Realizar la siguiente actividad:

- · Crear un repositorio.
- Crea un archivo simple, realiza los comandos git add . y git commit -m.



3) Realizar la siguiente actividad:



crear rama y subirla

```
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\primer-repo-git> git add .
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\primer-repo-git> git commit -m "Cambios en rama repo2"
On branch repo2
nothing to commit, working tree clean
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\primer-repo-git> git push origin repo2
>>
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'repo2' on GitHub by visiting:
remote: https://github.com/emiready/primer-repo-git/pull/new/repo2
remote:
To https://github.com/emiready/primer-repo-git
* [new branch] repo2 -> repo2
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\primer-repo-git>
```

Ejercicio 3.

```
en Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics)
PS C:\Users\emili\OneDrive\Escritorio\Programacion1> git clone https://github.com/emiready/-conflict-exercise
remote: Enumerating objects: 3, done. remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0) Receiving objects: 100% (3/3), done.
PS C:\Users\emili\OneDrive\Escritorio\Programacion1> cd -conflict-exercise
Set-Location : No se encuentra ningún parámetro que coincida con el nombre del parámetro 'conflict-exercise'.
En línea: 1 Carácter: 4
Cloning into 'conflic_exercise'...
remote: Enumerating objects: 3, done.
remote: Enumerating objects. 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
PS C:\Users\emili\OneDrive\Escritorio\Programacion1> cd conflic_exercise
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\conflic_exercise> git checkout -b feature-branch
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\conflic_exercise>
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\conflic_exercise>
PS C:\Users\emili\OneDrive\Escritorio\Programacion1\conflic_exercise> cd ...
PS C:\Users\emili\OneDrive\Escritorio\Programacion1> cd
PS C:\Users\emili\OneDrive\Escritorio> git clone https://github.com/emiready/conflict-exercise
remote: Counting objects: 100% (3/3), done.
PS C:\Users\emili\OneDrive\Escritorio> cd /conflict-exercise
cd : No se encuentra la ruta de acceso 'C:\conflict-exercise' porque no existe.
   + CategoryInfo : ObjectNotFound: (C:\conflict-exercise:String) [Set-Location], ItemNotFoundException + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand
PS C:\Users\emili\OneDrive\Escritorio> cd conflict-exercise
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git checkout -b feature-branch
1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git checkout master
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git branch
 main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git add README.md
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise>
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git add README.md
[main e876ed1] Added a line in main branch
1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>' hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
```

```
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
>>
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
>>
error: Merging is not possible because you have unmerged files.
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
>> git merge feature-branch
error: Merging is not possible because you have unmerged files.
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
error: Merging is not possible because you have unmerged files.
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
error: Merging is not possible because you have unmerged files.
hint: as appropriate to mark resolution and make a commit.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git add .
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git commit "resuelto el conflicto"
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
fatal: You have not concluded your merge (MERGE HEAD exists).
Please, commit your changes before you merge.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise>
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git commit -m "resuelto"
[main e58b9c2] resuelto
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge conflict-exercise
merge: conflict-exercise - not something we can merge
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git branch
 feature-branch
* main
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git merge feature-branch
Already up to date.
PS C:\Users\emili\OneDrive\Escritorio\conflict-exercise> git push origin main
Enumerating objects: 11, done. Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 761 bytes | 190.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/emiready/conflict-exercise
   a4d4411..e58b9c2 main -> main
```

