

Solución ejercicios avanzados de Git, GitHub y Markdown

Notas

- Este ejercicio es continuación del anterior por lo que tendréis que seguir trabajando en el repositorio **masteruah**.
- También tendréis que ir poniendo los **comandos** que habéis tenido que utilizar durante todos los ejercicios y las **explicaciones y capturas de pantalla** que consideréis **necesarias** al fichero README.md del citado repositorio.

Crear una rama v0.2

- Crear una rama **v0.2**.
`git branch v0.2`
- Posiciona tu carpeta de trabajo en esta rama.
`git checkout v0.2`

Añadir fichero 2.txt

- Añadir un fichero **2.txt** en la rama **v0.2**.
`touch 2.txt`
`git add .`
`git commit -m "añadido 2.txt"`

Crear rama remota v0.2

- Subir los cambios al repositorio remoto.
`git push origin v0.2`

Merge directo

- Posicionarse en la rama **master**.
`git checkout master`
- Hacer un merge de la rama **v0.2** en la rama **master**.
`git merge v0.2 -m "merge v0.2 sin conflictos"`

Merge con conflicto

- En la rama **master** poner **Hola** en el fichero **1.txt** y hacer commit.
`git checkout master`
`echo "Hola" >> 1.txt`
`git add .`
`git commit -m "hola en 1.txt"`
- Posicionarse en la rama **v0.2** y poner **Adios** en el fichero "1.txt" y hacer commit.
`git checkout v0.2`
`echo "Adios" >> 1.txt`
`git add .`

```
git commit -m "adios en 1.txt"
```

- Posicionarse de nuevo en la rama **master** y hacer un merge con la rama **v0.2**

```
git checkout master
```

```
git merge v0.2
```

```
vim 1.txt
```

```
git add .
```

```
git commit -m "arreglado merge en 1.txt"
```

Listado de ramas

- Listar las ramas con merge y las ramas sin merge.

```
git branch --merged
```

```
git branch --no-merged
```

Arreglar conflicto

- Arreglar el conflicto anterior y hacer un commit.

```
vim 1.txt
```

```
git add .
```

```
git commit -m "arreglado merge en 1.txt"
```

Borrar rama

- Crear un tag **v0.2**

```
git tag v0.2
```

- Borrar la rama **v0.2**

```
git branch -d v0.2
```

Listado de cambios

- Listar los distintos commits con sus ramas y sus tags.

```
git config --global alias.list 'log --oneline --decorate --graph --all'
```

```
git list
```

Crear una organización

- Crear una organización llamada **masteruah-tunombredeusuariodegithub**

En el menú de GitHub > Pulsar en el icono "+" > Pinchar en "New organization"

Crear equipos

- Crear 2 equipos en la organización **masteruah-tunombredeusuariodegithub**, uno llamado **administradores** con más permisos y otro **colaboradores** con menos permisos.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub> > Pinchar en "Teams" > Pulsar en "New team"

- Meter a github.com/asanzdiego y a 2 de vuestros compañeros de clase en el equipo **administradores**.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub> > Pinchar en "Teams" > Seleccionar el equipo "administradores" > Pulsar en "New member"

- Meter a github.com/asanzdiego y a otros 2 de vuestros compañeros de clase en el equipo **colaboradores**.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub> > Pinchar en "Teams" > Seleccionar el equipo "colaboradores" > Pulsar en "New member"

Crear un index.html

- Crear un index.html que se pueda ver como página web en la organización.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub> > Pulsar en "New"
Añadir un repositorio llamado masteruah-tunombredeusuariodegithub.github.io
Añadir un index.html a dicho repositorio

Crear pull requests

- Hacer 2 forks de 2 repositorios **masteruah-tunombredeusuariodegithub.github.io** de 2 organizaciones de las que no seáis ni administradores ni colaboradores.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub/masteruah-tunombredeusuariodegithub.github.io> > Pulsar en "Fork"

- Crearos una rama en cada fork.

En tu repo forkeado creas una rama.

- En cada rama modificar el fichero **index.html** añadiendo vuestro nombre.

Situada en dicha rama modificar el fichero index.html.

- Con cada rama hacer un pull request.

Ir a <http://github.com/masteruah-tunombredeusuariodegithub/masteruah-tunombredeusuariodegithub.github.io> > Pinchar en "Pull requests"
Pulsar en "New pull request"

Gestionar pull requests

- Aceptar los pull requests que lleguen a los repositorios de tu organización.

Dentro del repositorio de la organización gestionar dichos pull requests.