Taller

Ejercicios de manejo del historial de cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios de creación y actualización de repositorios

Ejercicio 1

1. Mostrar el historial de cambios del repositorio.
2. Crear la carpeta `capitulos` y crear dentro de ella el fichero `capitulo1.txt` con el siguiente texto.

   Git es un sistema de control de versiones ideado por Linus Torvalds.

3. Añadir los cambios a la zona de intercambio temporal.
4. Hacer un commit de los cambios con el mensaje “Añadido capítulo 1.”
5. Volver a mostrar el historial de cambios del repositorio.

Ejercicio 2

1. Crear el fichero `capitulo2.txt` en la carpeta `capitulos` con el siguiente texto.

   El flujo de trabajo básico con Git consiste en: 1- Hacer cambios en el repositorio. 2- Añadir los cambios a la zona de intercambio temporal. 3- Hacer un commit de los cambios.

2. Añadir los cambios a la zona de intercambio temporal.
3. Hacer un commit de los cambios con el mensaje “Añadido capítulo 2.”
4. Mostrar las diferencias entre la última versión y dos versiones anteriores.

Ejercicio 3

1. Crear el fichero `capitulo3.txt` en la carpeta `capitulos` con el siguiente texto.

   Git permite la creaci amas lo que permite tener distintas versiones d mo proyecto y trabajar de manera simultanea en ellas.

2. Añadir los cambios a la zona de intercambio temporal.
3. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3.”
4. Mostrar las diferencias entre la primera y la última versión del repositorio.

Ejercicio 4

1. Añadir al final del fichero `indice.txt` la siguiente línea:

   Capítulo 5: Conceptos avanzados

2. Añadir los cambios a la zona de intercambio temporal.
3. Hacer un commit de los cambios con el mensaje “Añadido capítulo 5 al índice.”.
4. Mostrar quién ha hecho cambios sobre el fichero `indice.txt`.

Ejercicios de deshacer cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios

Ejercicio 1

1. Eliminar la última línea del fichero `indice.txt` y guardarlo.

2. Comprobar el estado del repositorio.
3. Deshacer los cambios realizados en el fichero `indice.txt` para volver a la versión anterior del fichero.
4. Volver a comprobar el estado del repositorio.

Ejercicio 2

1. Eliminar la última línea del fichero `indice.txt` y guardarlo.
2. Añadir los cambios a la zona de intercambio temporal.
3. Comprobar de nuevo el estado del repositorio.
4. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
5. Comprobar de nuevo el estado del repositorio.
6. Deshacer los cambios realizados en el fichero `indice.txt` para volver a la versión anterior del fichero.
7. Volver a comprobar el estado del repositorio.

Ejercicio 3

1. Eliminar la última línea del fichero `indice.txt` y guardarlo.
2. Eliminar el fichero `capitulos/capitulo3.txt`.
3. Añadir un fichero nuevo `captitulos/capitulo4.txt` vacío.
4. Añadir los cambios a la zona de intercambio temporal.
5. Comprobar de nuevo el estado del repositorio.
6. Quitar los cambios de la zona de intercambio temporal, pero mantenerlos en el directorio de trabajo.
7. Comprobar de nuevo el estado del repositorio.
8. Deshacer los cambios realizados para volver a la versión del repositorio.
9. Volver a comprobar el estado del repositorio.

Ejercicio 4

1. Eliminar la última línea del fichero `indice.txt` y guardarlo.
2. Eliminar el fichero `capitulos/capitulo3.txt`.
3. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje “Borrado accidental.”
4. Comprobar el historial del repositorio.
5. Deshacer el último commit pero mantener los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.
6. Comprobar el historial y el estado del repositorio.
7. Volver a hacer el commit con el mismo mensaje de antes.
8. Deshacer el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.
9. Comprobar de nuevo el historial y el estado del repositorio.

Ejercicios de gestión de ramas

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios

Ejercicio 1

Crear una nueva rama `bibliografia` y mostrar las ramas del repositorio.

Ejercicio 2

1. Crear el fichero `capitulos/capitulo4.txt` y añadir el texto siguiente

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto.

2. Añadir los cambios a la zona de intercambio temporal.
3. Hacer un commit con el mensaje “Añadido capítulo 4.”
4. Mostrar la historia del repositorio incluyendo todas las ramas.

Ejercicio 3

1. Cambiar a la rama `bibliografia`.
2. Crear el fichero `bibliografia.txt` y añadir la siguiente referencia

    - Chacon, S. and Straub, B. Pro Git. Apress.

3. Añadir los cambios a la zona de intercambio temporal.
4. Hacer un commit con el mensaje “Añadida primera referencia bibliográfica.”
5. Mostrar la historia del repositorio incluyendo todas las ramas.

Ejercicio 4

1. Fusionar la rama `bibliografia` con la rama `master`.
2. Mostrar la historia del repositorio incluyendo todas las ramas.
3. Eliminar la rama `bibliografia`.
4. Mostrar de nuevo la historia del repositorio incluyendo todas las ramas.

Ejercicio 5

1. Crear la rama `bibliografia`.
2. Cambiar a la rama `bibliografia`.
3. Cambiar el fichero `bibliografia.txt` para que contenga las siguientes referencias:

    - Scott Chacon and Ben Straub. Pro Git. Apress.

- Ryan Hodson, Ry's Git Tutorial, Smashwords (2014)

4. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Añadida nueva referencia bibliográfica."
5. Cambiar a la rama `master`
6. Cambiar el fichero `bibliografia.txt` para que contenga las siguientes referencias:

   - Chacon, S. and Straub, B. Pro Git. Apress.
   - Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.

7. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Añadida nueva referencia bibliográfica."
8. Fusionar la rama `bibliografia` con la rama `master`.
9. Resolver el conflicto dejando el fichero `bibliografia.txt` con las referencias:

   - Chacon, S. and Straub, B. Pro Git. Apress.
   - Loeliger, J. and McCullough, M. Version control with Git. O'Reilly.
   - Hodson, R. Ry's Git Tutorial. Smashwords (2014)

10. Añadir los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Resuelto conflicto de bibliografía."
11. Mostrar la historia del repositorio incluyendo todas las ramas.

Ejercicios de repositorios remotos

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre ramas

Ejercicio 1

1. Crear un nuevo repositorio público en GitHub con el nombre `libro-git`.
2. Añadirlo al repositorio local del libro.
3. Mostrar todos los repositorios remotos configurados.

Ejercicio 2

1. Añadir los cambios del repositorio local al repositorio remoto de GitHub.
2. Acceder a GitHub y comprobar que se han subido los cambios mostrando el historial de versiones.

Ejercicio 3

1. Colaborar en el repositorio remoto `libro-git` de otro usuario.
2. Clonar su repositorio `libro-git`.
3. Añadir el fichero `autores.txt` que contenga el nombre del usuario y su correo electrónico.
4. Añadir los cambios a la zona de intercambio temporal.
5. Hacer un commit con el mensaje “Añadido autor.”
6. Subir los cambios al repositorio remoto.

Ejercicio 4

1. Hacer una bifurcación del repositorio remoto `asalber/libro-git` en GitHub.
2. Clonar el repositorio creado en la cuenta de GitHub del usuario.
3. Crear una nueva rama `autoria` y activarla.
4. Añadir el nombre del usuario y su correo al fichero `autores.txt`.
5. Añadir los cambios a la zona de intercambio temporal.
6. Hacer un commit con el mensaje “Añadido nuevo autor.”
7. Subir los cambios de la rama `autoria` al repositorio remoto en GitHub.

8. Hacer un Pull Request de los cambios en la rama `autoria`.


```
> git
> mkdir capitulos
> cat > capitulos/capitulo1.txt
Git es un sistema de control de versiones ideado por Linus
Torvalds.
Ctrl+D
> git add .
> git commit -m
```

```
> git
```

```
> cat > capitulos/capitulo2.txt
El flujo de trabajo básico con Git consiste en:
1- Hacer cambios en el repositorio.
2- Añadir los cambios a la zona de intercambio temporal.
3- Hacer un commit de los cambios.
Ctrl+D
> git add .
> git commit -m
```

```
> git diff HEAD~2 .HEAD
```

```
> cat > capitulos/capitulo3.txt
Git permite la creación de ramas lo que permite tener distintas
versiones del mismo proyecto y trabajar de manera simultánea en
ellas.
Ctrl+D
> git add .
> git commit -m
> git
```

```
> git diff &lt;&lt;Codigo    de la primera version&gt;&gt; HEAD
```

```
>                                    >> indice.txt
> git add .
```


```
> git commit -m
> git annotate indice.txt
```

```
> nano indice.txt
```

```
> git status
> git checkout -- indice.txt
```

```
> git status
```

```
Ejercicio 2
```

```
> nano indice.txt
```

```
> git add .
> git status
> git reset indice.txt
> git status
> git checkout -- indice.txt
```

```
> git status
```

```
> nano indice.txt
```

```
> rm capitulos/capitulo3.txt
> touch capitulos/capitulo4.txt
> git add .
> git status
> git reset
> git status
> git checkout --
> git status
> git clean -f
```

```
> git status
```