



Boletín de ejercicios Tema 2 - RESUELTO

Ejercicios de creación y actualización de repositorios

1. Ejercicio 1

Configura Git definiendo tu nombre del usuario y tu correo electrónico. Muestra la configuración final

```
> git config --global user.name "Alumno Montecastelo"
> git config --global user.email "alumno@ciclosmontecastelo.com"
> git config --list
```

2. Ejercicio 2

Crea un repositorio nuevo con el nombre **Montecastelo** y muestra su contenido.

```
> mkdir Montecastelo
> cd Montecastelo
> git init
> ls -la
```

3. Ejercicio 3

Realiza las siguientes acciones:

- Comprueba el estado del repositorio.
- Crea un fichero **objetivosDAW.txt** con el siguiente contenido:
 - Trabajo con servidores
 - Conocimiento de GIT
 - Trabajo con Docker
- Comprueba de nuevo el estado del repositorio.



- Añade el fichero a la zona de intercambio temporal.
- Vuelve a comprobar una vez más el estado del repositorio

```
> git status
> notepad objetivosDAW.txt
Trabajo con servidores
Conocimiento de GIT
Trabajo con Docker
> git status
> git add objetivosDAW.txt
> git status
```

4. Ejercicio 4

Realiza un commit de los últimos cambios con el mensaje “Añadidos objetivos de la asignatura.” y comprueba el estado del repositorio.

```
> git commit -m "Añadidos objetivos de la asignatura"
> git status
```

5. Ejercicio 5

- Cambia el fichero **objetivosDAW.txt** para que contenga lo siguiente:
 - Trabajo con servidores
 - Conocimiento de GIT
 - Trabajo con Docker
 - Kubernetes
- Muestra los cambios con respecto a la última versión guardada en el repositorio.
- Haz un commit de los cambios con el mensaje “Añadido Kubernetes”.



```
> notepad objetivosDAW.txt
Trabajo con servidores
Conocimiento de GIT
Trabajo con Docker
Kubernetes
> git diff
> git add objetivosDAW.txt
> git commit -m "Añadido Kubernetes"
```

6. Ejercicio 6

- Muestra los cambios de la última versión del repositorio con respecto a la anterior.
- Cambia el mensaje del último commit por "Añadida sección de Kubernetes."
- Vuelve a mostrar los últimos cambios del repositorio.

```
> git show
> git commit --amend -m "Añadida sección de Kubernetes."
> git show
```

Ejercicios del manejo del historial de cambios

1. Ejercicio 1

- Muestra el historial de cambios del repositorio.
- Crea la carpeta profesores y crea dentro de ella el fichero **profesores.txt** con el siguiente texto: "El profesor de DAW es más simpático que el de DWES"
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit de los cambios con el mensaje "Añadido profesor DAW"
- Vuelve a mostrar el historial de cambios del repositorio.



```
> git log
> mkdir profesores
> notepad profesores/profesores.txt
El profesor de DAW es más simpático que el de DWES
Ctrl+D
> git add .
> git commit -m "Añadido profesor DAW"
> git log
```

2. Ejercicio 2

- Crea el fichero **profesores2.txt** en la carpeta profesores con el siguiente texto: "Ramón es un crack"
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit de los cambios con el mensaje "Añadido profe Ramón"
- Muestra las diferencias entre la última versión y dos versiones anteriores.

```
> notepad profesores/profesores2.txt
Ramón es un crack
Ctrl+D
> git add .
> git commit -m "Añadido profe Ramón"
> git diff HEAD~2..HEAD
```

3. Ejercicio 3

- Crea el fichero **profesores3.txt** en la carpeta profesores con el siguiente texto: "Enrique es el mejor en su campo"
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit de los cambios con el mensaje "Añadido profe Enrique"
- Muestra las diferencias entre la última versión y dos versiones anteriores.



```
> notepad profesores/profesores3.txt
Enrique es el mejor en su campo
Ctrl+D
> git add .
> git commit -m "Añadido profe Enrique"
> git log
> git diff
```

4. Ejercicio 4

- Añade al final del fichero **objetivosDAW.txt** la siguiente línea: Trabajo práctico en empresas
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit de los cambios con el mensaje "Añadido trabajo en empresas".
- Muestra quién ha hecho cambios sobre el fichero objetivosDAW.txt.

```
> echo "Trabajo práctico en empresas" >> objetivosDAW.txt
> git add .
> git commit -m "Añadido trabajo práctico en empresas"
> git annotate indice.txt
```

Ejercicios de deshacer cambios

1. Ejercicio 1

- Elimina la última línea del fichero **objetivosDAW.txt** y guárdalo.
- Comprueba el estado del repositorio.
- Deshaz los cambios realizados en el fichero objetivosDAW.txt para volver a la versión anterior del fichero.
- Vuelve a comprobar el estado del repositorio.



```
> notepad objetivosDAW.txt
# Eliminar la última línea y guardar el fichero.
> git status
> git checkout -- objetivosDAW.txt
> git status
```

2. Ejercicio 2

- Elimina la última línea del fichero **objetivosDAW.txt** y guárdalo.
- Añade los cambios a la zona de intercambio temporal.
- Comprueba de nuevo el estado del repositorio.
- Quita los cambios de la zona de intercambio temporal, pero mantenlos en el directorio de trabajo.
- Comprueba de nuevo el estado del repositorio.
- Deshaz los cambios realizados en el fichero **objetivos.txt** para volver a la versión anterior del fichero.
- Vuelve a comprobar el estado del repositorio.

```
> notepad objetivosDAW.txt
# Eliminar la última línea y guardar el fichero.
> git add .
> git status
> git reset objetivosDAW.txt
> git status
> git checkout -- objetivosDAW.txt
> git status
```

3. Ejercicio 3

- Elimina la última línea del fichero **objetivosDAW.txt** y guárdalo.
- Elimina el fichero **profesores/profesores3.txt**.
- Añade un fichero nuevo **profesores/profesores4.txt** vacío.
- Añade los cambios a la zona de intercambio temporal.



- Comprueba de nuevo el estado del repositorio.
- Quita los cambios de la zona de intercambio temporal, pero mantenlos en el directorio de trabajo.
- Comprueba de nuevo el estado del repositorio.
- Deshacer los cambios realizados para volver a la versión del repositorio.
- Volver a comprobar el estado del repositorio.

```
> notepad objetivosDAW.txt
# Eliminar la última línea y guardar el fichero.
> rm profesores/profesores3.txt
> notepad profesores/profesores4.txt
> git add .
> git status
> git reset
> git status
> git checkout -- .
> git status
> git clean -f
> git status
```

4. Ejercicio 4

- Elimina la última línea del fichero **objetivosDAW.txt** y guardarlo.
- Elimina el fichero **profesores/profesores3.txt**.
- Añade los cambios a la zona de intercambio temporal y haz un commit con el mensaje "Borrado accidental."
- Comprueba el historial del repositorio.
- Deshaz el último commit, pero mantén los cambios anteriores en el directorio de trabajo y la zona de intercambio temporal.
- Comprueba el historial y el estado del repositorio.
- Vuelve a hacer el commit con el mismo mensaje de antes.



- Deshaz el último commit y los cambios anteriores del directorio de trabajo volviendo a la versión anterior del repositorio.
- Comprueba de nuevo el historial y el estado del repositorio.

```
> notepad indice.txt
# Eliminar la última línea y guardar el fichero.
> rm profesores/profesores3.txt
> git commit -a "Borrado accidental."
> git status
> git log
> git reset --soft HEAD~1
> git status
> git commit -m "Borrado accidental."
> git status
> git log
> git reset --hard HEAD~1
> git log
> git status
```

Ejercicios de gestión de ramas

1. Ejercicio 1

Crea una nueva rama **Alumnado** y muestra las ramas del repositorio.

```
> git branch Alumnado
> git branch -av
```

2. Ejercicio 2

- Crea el fichero **profesores/profesores4.txt** y añade el texto siguiente:
"Fran es el que faltaba por nombrar. Todo un fenómeno"
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit con el mensaje "Añadido profe Fran"
- Muestra la historia del repositorio incluyendo todas las ramas.



```
> notepad profesores/profesores4.txt
Fran es el que faltaba por nombrar. Todo un fenómeno
> git add .
> git commit -m "Añadido profe Fran"
> git log --graph --all --oneline
```

3. Ejercicio 3

- Cambia a la rama **Alumnado**.
- Crea el fichero **alumnos.txt** y añadir la siguiente referencia: "Antonio Serpa"
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit con el mensaje "Añadido primer alumno."
- Muestra la historia del repositorio incluyendo todas las ramas.

```
> git checkout Alumnado
> notepad alumnos.txt
Antonio Serpa
> git add .
> git commit -m "Añadido primer alumno"
> git log --graph --all --oneline
```

4. Ejercicio 4

- Fusiona la rama **Alumnado** con la rama master.
- Muestra la historia del repositorio incluyendo todas las ramas.
- Elimina la rama **Alumnado**.
- Muestra de nuevo la historia del repositorio incluyendo todas las ramas.

```
> git checkout master
> git merge Alumnado
> git log --graph --all --oneline
> git branch -d Alumnado
> git log --graph --all --oneline
```



5. Ejercicio 5

- Crea la rama **Alumnado**.
- Cambia a la rama **Alumnado**.
- Cambia el fichero **alumnos.txt** para que contenga las siguientes referencias:
 - "Miguel Aguiar"
 - "Fran Espiña"
- Añade los cambios a la zona de intercambio temporal y haz un commit con el mensaje "Añadidos nuevos alumnos"
- Cambia a la rama master.
- Cambia el fichero **alumnos.txt** para que contenga las siguientes referencias:
 - "José Carlos Vaqueiro"
 - "Fran Espiña"
 - "Fernando Tornillo"
- Añade los cambios a la zona de intercambio temporal y hacer un commit con el mensaje "Añadidos más alumnos"
- Fusiona la rama **Alumnado** con la rama master.
- Resuelve el conflicto dejando el fichero alumnos.txt con las referencias:
 - "José Carlos Vaqueiro"
 - "Fran Espiña"
 - "Fernando Tornillo"
- Añade los cambios a la zona de intercambio temporal y haz un commit con el mensaje "Resuelto conflicto de alumnado."
- Muestra la historia del repositorio incluyendo todas las ramas



```
> git branch Alumnado
> git checkout Alumnado
> notepad alumnos.txt
- Miguel Aguiar
- Fran Espiña
> git commit -a -m "Añadidos nuevos alumnos"
> git checkout master
> notepad alumnos.txt
- "José Carlos Vaqueiro"
- "Fran Espiña"
- "Fernando Tornillo"
> git commit -a -m "Añadidos más alumnos"
> git merge Alumnado
# Hacer los cambios indicados en el fichero
> git commit -a -m "Solucionado conflicto alumnado."
> git log --graph --all --oneline
```

Ejercicios de repositorios remotos

1. Ejercicio 1

- Crea un nuevo repositorio público en GitHub con el nombre **montecastelo-git**.
- Añádalo al repositorio local de **montecastelo**.
- Muestra todos los repositorios remotos configurados.

```
# Crear el repositorio en GitHub y copiar su url con protocolo https.
> git remote add montecastelo-git.git
> git remote -v
```

2. Ejercicio 2

- Añade los cambios del repositorio local al repositorio remoto de GitHub.
- Accede a GitHub y comprueba que se han subido los cambios mostrando el historial de versiones.



```
> git push github master
```

3. Ejercicio 3

- Colabora en el repositorio remoto **montecastelo-git** de otro usuario.
- Clona su repositorio **montecastelo-git**
- Añade el fichero **autores.txt** que contenga el nombre del usuario y su correo electrónico.
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit con el mensaje "Añadido autor."
- Sube los cambios al repositorio remoto.

```
# Entrar en GitHub en el proyecto montecastelo-git del que seamos colaboradores y copiar la url.  
> git clone url  
> notepad autores.txt  
# Escribir el nombre del autor y su correo.  
> git add .  
> git commit -m "Añadido autor."  
> git push origin master.
```

4. Ejercicio 4

- Haz una bifurcación del repositorio remoto **otroprofe/montecastelo-git** en GitHub.
- Clona el repositorio creado en la cuenta de GitHub del usuario.
- Crea una nueva rama **MiRepo** y actívala.
- Añade el nombre del usuario y su correo al fichero **autores.txt**.
- Añade los cambios a la zona de intercambio temporal.
- Haz un commit con el mensaje "Añadido nuevo autor."
- Sube los cambios de la rama **MiRepo** al repositorio remoto en GitHub.
- Haz un Pull Request de los cambios en la rama **MiRepo**.



```
# Hacer el fork del repositorio otroprofe/montecastelo-git en GitHub
# y copiar la url del repositorio creado en la cuenta de GitHub
> git clone url
> git checkout -b MiRepo
# Editar el fichero autores.txt y añadir el nombre
# y el correo electrónico del usuario en una nueva línea.
> git commit -am "Añadido nuevo autor."
> git push origin MiRepo
# Ir al repositorio remoto en GitHub y hacer Pull Request
```