Práctica 0 - Ejercicios de GIT.

Comparar e investigar para completar los siguientes ejercicios.

1. Creación y actualización de repositorios

Ejercicio 1

Configurar Git definiendo el nombre del usuario, el correo electrónico y activar el coloreado de la salida. Mostrar la configuración final.

`git config --global user.name gonzalo:` establece el nombre de usuario global de Git como "gonzalo".

`git config --global user.email gonzaloarielduarte2@gmail.com:` establece la dirección de correo electrónico global de Git como "gonzaloarielduarte2@gmail.com".

`git config --global color.ui auto:` habilita el color de la interfaz de usuario de Git para mostrar información de manera más clara y legible.

`git config --list:` muestra una lista de la configuración actual de Git, incluyendo tanto la configuración global como la específica del repositorio.

Ejercicio 2

Crear un repositorio nuevo con el nombre libro y mostrar su contenido.

`mkdir libro:` crea un directorio llamado "libro".

`cd libro:` entra en el directorio "libro".

`git init:` inicializa un nuevo repositorio Git en el directorio actual.

`ls -la:` lista los archivos y directorios en el directorio actual, incluyendo los archivos ocultos.

Ejercicio 3

1. Comprobar el estado del repositorio.
2. Crear un fichero indice.txt con el siguiente contenido:

   Capítulo 1: Introducción a Git.

   Capítulo 2: Flujo de trabajo básico.

   Capítulo 3: Repositorios remotos.
3. Comprobar de nuevo el estado del repositorio.
4. Añadir el fichero a la zona de intercambio temporal.
5. Volver a comprobar una vez más el estado del repositorio.

`git status:` muestra el estado actual del repositorio Git en el directorio actual. Esto puede incluir archivos modificados, archivos eliminados y archivos nuevos que aún no han sido agregados al área de preparación para el siguiente commit.

`cat > indice.txt:` crea un archivo llamado "indice.txt" y permite agregar contenido al archivo. En este caso, se agregan algunos títulos de capítulos como ejemplo.

`Capítulo 1: Introducción a Git`

`Capítulo 2: Flujo de trabajo básico`

Capítulo 3: Repositorios remotos

Ctrl+D: guarda el archivo "indice.txt" y sale del modo de edición.

git status: después de crear el archivo "indice.txt", Git muestra que hay un nuevo archivo sin seguimiento en el directorio de trabajo.

git add indice.txt: agrega el archivo "indice.txt" al área de preparación para el siguiente commit. Esto indica que se desea incluir los cambios en el archivo en el próximo commit.

git status: después de agregar el archivo al área de preparación, Git muestra que el archivo "indice.txt" está listo para ser confirmado (commit).

Ejercicio 4

Realizar un commit de los últimos cambios con el mensaje “Añadido índice del libro.” y ver el estado del repositorio.

Ejercicio 5

1. Cambiar el fichero indice.txt para que contenga lo siguiente:

   Capítulo 1: Introducción a Git.

   Capítulo 2: Flujo de trabajo básico.

   Capítulo 3: Gestión de ramas.

   Capítulo 4: Repositorios remotos.

2. Mostrar los cambios con respecto a la última versión guardada en el repositorio.
3. Hacer un commit de los cambios con el mensaje “Añadido capítulo 3 sobre gestión de ramas”.

   git commit -m "Añadido índice del libro.": confirma los cambios agregados al área de preparación en un nuevo commit. El mensaje "Añadido índice del libro." se agrega para proporcionar una descripción breve y clara de los cambios que se realizaron en este commit.

   git status: muestra el estado actual del repositorio Git en el directorio actual. Después de hacer un commit, Git muestra que no hay cambios pendientes y que el repositorio está actualizado

   cat > indice.txt: Este comando crea un archivo llamado "indice.txt" y permite agregar contenido a él. La primera parte del comando "> cat" se utiliza para redirigir la entrada estándar, en este caso desde el teclado, hacia el comando "cat", el cual se utiliza para concatenar archivos. La segunda parte del comando " > indice.txt" se utiliza para redirigir la salida del comando "cat" hacia el archivo "indice.txt".

   Capítulo 1: Introducción a Git

   Capítulo 2: Flujo de trabajo básico

   Capítulo 3: Gestión de ramas

   Capítulo 4: Repositorios remotos

   Ctrl+D: Para indicar que se ha terminado de agregar contenido al archivo.

   git diff: Este comando se utiliza para mostrar las diferencias entre los archivos del área de trabajo y el área de preparación. Como acabas de crear el archivo "indice.txt" y aún no lo has agregado al área de preparación, este comando no mostrará ninguna diferencia.

   git add indice.txt: Este comando agrega el archivo "indice.txt" al área de preparación, lo que significa que Git ahora está al tanto de los cambios realizados en el archivo.

git commit -m "Añadido capítulo 3 sobre gestión de ramas": Este comando confirma los cambios realizados en el archivo "indice.txt" y los agrega al historial del repositorio. El mensaje de confirmación "-m" se utiliza para agregar una descripción breve de los cambios realizados. En este caso, el mensaje de confirmación es "Añadido capítulo 3 sobre gestión de ramas".

Ejercicio 6

1. Mostrar los cambios de la última versión del repositorio con respecto a la anterior.
2. Cambiar el mensaje del último commit por “Añadido capítulo 3 sobre gestión de ramas al índice.”
3. Volver a mostrar los últimos cambios del repositorio.

2. Manejo del historial de cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios de creación y actualización de repositorios o bien hacer un clon del repositorio remoto:

git@github.com:iAlphaDataScience/LibroGit.git

Ejercicio 1

1. Mostrar el historial de cambios del repositorio.
2. Crear la carpeta capítulos 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.

git log: Este comando se utiliza para mostrar el historial de confirmaciones (commits) realizadas en el repositorio. Al ejecutar este comando, se mostrará la lista de commits, que incluyen la descripción de los cambios realizados, la fecha y la hora de cada confirmación, entre otros detalles.

mkdir capitulos: Este comando crea un nuevo directorio llamado "capitulos" en el directorio de trabajo actual. El comando mkdir (make directory) se utiliza para crear un nuevo directorio.

cat > capitulos/capitulo1.txt: Este comando crea un nuevo archivo llamado "capitulo1.txt" en el directorio "capitulos" y permite agregar contenido a él. El comando cat se utiliza para concatenar archivos, mientras que la redirección > capitulos/capitulo1.txt se utiliza para redirigir la salida del comando cat hacia el archivo "capitulo1.txt".

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

Ctrl+D: Para indicar que se ha terminado de agregar contenido al archivo.

git add .: Este comando agrega todos los archivos modificados en el directorio de trabajo actual al área de preparación. El punto después del comando add se utiliza para indicar que se deben agregar todos los archivos modificados en el directorio actual.

git commit -m "Añadido capítulo 1.": Este comando confirma los cambios realizados en el archivo "capitulo1.txt" y los agrega al historial del repositorio. El mensaje de confirmación "-m" se utiliza para agregar una descripción breve de los cambios realizados. En este caso, el mensaje de confirmación es "Añadido capítulo 1."

git log: Este comando se utiliza nuevamente para mostrar el historial de confirmaciones realizadas en el

repositorio, que ahora incluirá la confirmación que acabas de realizar con el mensaje "Añadido capítulo 1.".

Ejercicio 2

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

Git permite la creación de ramas lo que permite tener distintas versiones del mismo 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.

3. Deshacer cambios

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios o bien hacer un clon del repositorio remoto:

git@github.com:iAlphaDataScience/LibroGit.git

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.

nano indice.txt: Abrir el archivo indice.txt con el editor de texto nano. Si el archivo no existe, se creará uno

nuevo.

Eliminar la última línea del archivo indice.txt. Presionar la tecla Ctrl+O para guardar. Después, se puede salir del editor nano presionando Ctrl+X.

git status: Verificar el estado del repositorio.

git checkout -- indice.txt: Deshacer los cambios realizados en el archivo indice.txt. Este comando descarta los cambios realizados en el archivo y restaura su contenido al último commit.

git status: Verificar el estado del repositorio de nuevo con el comando. Se debería ver que el archivo indice.txt está en el estado que se encuentra en el último commit y que no hay cambios pendientes de confirmación.

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.

nano indice.txt: Abrir el archivo indice.txt con el editor de texto nano. Si el archivo no existe, se creará uno nuevo.

Eliminar la última línea del archivo indice.txt. Presionar la tecla Ctrl+O para guardar. Después, se puede salir del editor nano presionando Ctrl+X.

git add .: Agregar los cambios al área de preparación.

git status: Verificar el estado del repositorio.

git reset indice.txt: Este comando elimina los cambios del área de preparación y restaura el archivo a su estado anterior.

git status: Verificar el estado del repositorio.

git checkout -- indice.txt: Deshacer los cambios realizados en el archivo indice.txt. Este comando descarta los cambios realizados en el archivo y restaura su contenido al último commit.

git status: Verificar el estado del repositorio de nuevo con el comando. Se debería ver que el archivo indice.txt está en el estado que se encuentra en el último commit y que no hay cambios pendientes de confirmación.

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.

4. Gestión de ramas

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre historial de cambios o bien hacer un clon del repositorio:

git@github.com:iAlphaDataScience/LibroGit.git

Ejercicio 1

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

> git branch bibliografía: Crea una nueva rama llamada "bibliografia" a partir de la rama actual en la que se encuentra el usuario.
>
> git branch -av: Muestra una lista de todas las ramas en el repositorio, incluyendo la rama actual y su historial. La opción -a indica que se muestren tanto las ramas locales como las remotas, y la opción -v muestra información adicional, como el último commit en cada rama.

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.

> cat > capitulos/capitulo4.txt: Se crea un archivo llamado "capitulo4.txt" dentro de la carpeta "capitulos"

En este capítulo veremos cómo usar GitHub para alojar repositorios en remoto y finalizando con la combinación de teclas Ctrl+D

git commit -m "Añadido capítulo 4.": Se hace un commit con el mensaje "Añadido capítulo 4."

git log --graph --all –oneline: Para mostrar el historial de commits de forma gráfica y concisa. La opción --graph muestra el historial de commits en forma de grafo, lo que facilita la visualización de las ramas y fusiones. La opción --all muestra todos los branches en el repositorio, incluso aquellos que no están actualmente chequeados. La opción --oneline muestra cada commit en una sola línea, lo que hace que sea más fácil de leer.

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.

5. Repositorios remotos

Para hacer estos ejercicios es necesario haber hecho antes los ejercicios sobre ramas o bien hacer un clon del repositorio remoto:

git@github.com:iAlphaDataScience/LibroGit.git

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 duartegonzaloariel/libro-git2 en GitHub.
1. Clonar el repositorio creado en la cuenta de GitHub del usuario.
2. Crear una nueva rama autoria y activarla.
3. Modificado el correo al fichero autores.txt.

4. Añadir los cambios a la zona de intercambio temporal.
5. Hacer un commit con el mensaje “Modificado el correo.”
6. Subir los cambios de la rama autoria al repositorio remoto en GitHub.
7. Hacer un Pull Request de los cambios en la rama autoria.