

Práctico 2: Git y GitHub

Actividades

1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

- **¿Qué es GitHub?**

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código. Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente: Presentar o compartir el trabajo. Seguir y administrar los cambios en el código a lo largo del tiempo.

- **¿Cómo crear un repositorio en GitHub?**

Dentro de la pagina web de GitHub utiliza el menú desplegable de la esquina superior derecha de cualquier página y selecciona Nuevo Repositorio. Escribe un nombre para el repositorio. También puedes añadir una descripción al mismo. Elige la visibilidad del repositorio: público, interno o privado.

- **¿Cómo crear una rama en Git?**

Para crear una rama en Git abrimos la consola de Git (no es la unica manera de hacerlo), Git Bash y una vez inicializado el git en la carpeta repositorio local donde nos queremos situar, escribimos el comando git branch [nombre de la rama].

- **¿Cómo cambiar a una rama en Git?**

Para cambiarnos a una rama en Git, utilizamos el comando git checkout [nombre]

- **¿Cómo fusionar ramas en Git?**

Para fusionar ramas en git debemos hacer un merge, utilizando el comando git merge [nombre de la rama a fusionar].

- **¿Cómo crear un commit en Git?**

Para crear un commit en git podemos usar el comando git commit -m "Mensaje que indique los cambios realizados".

- **¿Cómo enviar un commit a GitHub?**

Primero seguimos los cambios realizados con el comando git add .

Segundo preparamos los cambios que seran enviados a GitHub con el comando git commit -m "mensaje"

Tercero enviamos los cambios a GitHub con el comando git push origin master.

El comando git push toma dos argumentos, un nombre remoto y el nombre de la nueva rama.

- **¿Qué es un repositorio remoto?**

Un repositorio remoto es una copia de un proyecto que se almacena en un servidor remoto, en internet o en una red. Se comparte entre varios miembros de un equipo.

- **¿Cómo agregar un repositorio remoto a Git?**

Para crear un nuevo repositorio, usarás el comando git init . git init es un comando único que se usa durante la configuración inicial de un nuevo repositorio. Al ejecutar este comando, se creará un nuevo subdirectorio .git en tu directorio de trabajo actual. Esto también creará una nueva rama principal.

- **¿Cómo empujar cambios a un repositorio remoto?**

Para empujar cambios a un repositorio remoto con Git, se utiliza el comando git push.

Sintaxis

git push <nombre del repositorio> <nombre de la rama>

• ¿Cómo tirar de cambios de un repositorio remoto?

Para obtener los cambios de un repositorio remoto, puedes usar el comando git fetch. Este comando recupera el trabajo realizado por otras personas.

• ¿Qué es un fork de repositorio?

Un fork de repositorio es una copia exacta de un repositorio original que se crea para colaborar en un proyecto. En español, se le conoce como bifurcación.

• ¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio en GitHub, se puede hacer lo siguiente:

- 1 - Ir a la página del repositorio que se desea bifurcar
- 2 - Hacer clic en el botón Fork que se encuentra en la parte superior derecha de la página
- 3 - Seleccionar la cuenta donde se desea crear la copia
- 4 - Darle un nombre y una descripción al fork
- 5 - Hacer clic en Create fork

El fork es una copia exacta del repositorio original, pero con una URL diferente. Esto permite a los desarrolladores contribuir a un proyecto de forma segura.

• ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

Para enviar una solicitud de extracción (pull request) a un repositorio, puedes seguir los pasos que correspondan a la herramienta que uses.

GitHub

- 1 - Ir a la página principal del repositorio
- 2 - Seleccionar la rama que contiene los cambios
- 3 - Hacer clic en "Comparar y solicitud de incorporación de cambios"
- 4 - Seleccionar la rama base y la rama de comparación

5 - Escribir un título y una descripción

6 - Hacer clic en "Crear solicitud de incorporación de cambios"

- **¿Cómo aceptar una solicitud de extracción?**

Para aceptar una solicitud de extracción, puedes revisar los cambios propuestos, dejar comentarios y hacer clic en el botón "Aprobar".

Pasos para aprobar una solicitud de extracción

1 - Abre la solicitud de extracción.

2 - Revisa los cambios propuestos.

3 - Deja un comentario expresando tu aprobación.

4 - Si es necesario, solicita cambios o deja sugerencias.

5 - Haz clic en el botón "Aprobar".

- **¿Qué es un etiqueta en Git?**

Una etiqueta en Git es una marca que se aplica a una confirmación para identificar un punto importante en el historial de un repositorio.

- **¿Cómo crear una etiqueta en Git?**

Para crear una etiqueta en Git, puedes usar el comando git tag.

Crear una etiqueta anotada

1 - Usa la opción -a

2 - Especifica el mensaje de la etiqueta con la opción -m

3 - Escribe el nombre de la etiqueta

4 - Escribe el mensaje de la etiqueta

Crear una etiqueta ligera

1 - No uses las opciones -a, -s ni -m

2 - Escribe el nombre de la etiqueta

Ver las etiquetas

Para ver las etiquetas almacenadas en un repositorio, puedes usar el comando git tag.

- **¿Cómo enviar una etiqueta a GitHub?**

Enviar las etiquetas a GitHub

Para enviar las etiquetas creadas desde local al repositorio GitHub, puedes lanzar el push con la opción `-tags`

• ¿Qué es un historial de Git?

El historial de Git es un registro de los cambios realizados en un proyecto, que se almacena como un gráfico de confirmaciones.

Características del historial de Git

Cada confirmación muestra la fecha, el autor, el mensaje y el SHA.

Las confirmaciones pueden tener varios elementos primarios, lo que crea un historial similar a un gráfico.

El historial de Git permite volver al proyecto para ver quiénes son los autores, averiguar dónde se introdujeron los errores y revertir los cambios problemáticos.

• ¿Cómo ver el historial de Git?

Para ver el historial de Git, puedes usar el comando `git log` o la interfaz de usuario de GitLab o GitHub.

Con el comando `git log`

`git log` muestra las confirmaciones de un repositorio en orden cronológico inverso

`git log` proporciona información detallada sobre cada confirmación, incluyendo el hash, el autor, la fecha y el mensaje

`git log --oneline` muestra los primeros siete caracteres del hash SHA-1 y el mensaje de confirmación

`git log --oneline --graph` muestra el historial de confirmaciones en un gráfico ASCII

• ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes utilizar el comando `git log` con diferentes opciones.

Opciones de `git log`

`git log -L` Muestra el historial de una línea de código o función

`git log -p` Muestra las diferencias

`git log -G` Busca diferencias que coincidan con una expresión regular

`git log -S` Busca diferencias que agreguen o eliminen una cadena específica

`git log --grep=<pattern>` Busca un patrón en los mensajes de confirmación

`git log --author "nombre de autor"` Filtra el historial por autor

- ¿Cómo borrar el historial de Git?

Para borrar el historial de Git localmente, puedes borrar la carpeta `.git`. Si ya has hecho un push al repositorio remoto, el historial permanecerá ahí.

Para eliminar un archivo del historial de Git, puedes reescribir la historia del repositorio. Para ello, puedes usar `git-filter-repo`.

Borrar el historial de Git localmente

Busca y borra la carpeta `.git`, que se encuentra en la raíz de tu carpeta.

Vuelve a crear la carpeta con el mismo nombre.

- **¿Qué es un repositorio privado en GitHub?**

Un repositorio privado en GitHub es un espacio donde se puede guardar código, archivos y revisiones de forma privada. Solo los usuarios con permiso pueden ver el contenido de un repositorio privado.

- **¿Cómo crear un repositorio privado en GitHub?**

Para crear un repositorio privado en GitHub, puedes seguir estos pasos:

Ir a la esquina superior derecha de cualquier página de GitHub

Seleccionar Nuevo repositorio

Escribir un nombre para el repositorio

Agregar una descripción opcional

Elegir la visibilidad del repositorio

Seleccionar Crear repositorio

- **¿Cómo invitar a alguien a un repositorio privado en GitHub?**

Para invitar a alguien a un repositorio privado en GitHub, puedes seguir estos pasos:

Ir a la página principal del repositorio

Hacer clic en Configuración

En la barra lateral, hacer clic en Colaboradores y equipos

Hacer clic en Agregar personas

Escribir el nombre de usuario o correo electrónico de la persona a invitar

Elegir el rol de repositorio que se le dará a la persona

Hacer clic en Agregar NOMBRE al REPOSITORIO

- **¿Qué es un repositorio público en GitHub?**

Un repositorio público en GitHub es un espacio en la nube donde se puede almacenar y compartir código, archivos y revisiones de forma gratuita y accesible para todos.

- **¿Cómo crear un repositorio público en GitHub?**

Para crear un repositorio público en GitHub, puedes seguir estos pasos:

Ir a GitHub

En la esquina superior derecha, seleccionar Nuevo repositorio

Escribir un nombre para el repositorio

Agregar una descripción opcional

Elegir la visibilidad del repositorio como público

Seleccionar Inicializar este repositorio con un archivo Léame

Hacer clic en Crear repositorio

- **¿Cómo compartir un repositorio público en GitHub?**

Para compartir un repositorio público en GitHub, puedes invitar a colaboradores o compartir la URL del repositorio.

Invitar colaboradores

Ir a la página principal del repositorio

Hacer clic en Configuración

En la sección "Acceso", hacer clic en Colaboradores y equipos

Hacer clic en Agregar personas o Agregar equipos

Escribir el nombre de la persona o equipo a invitar

Seleccionar el rol de repositorio que se desea otorgar

Hacer clic en Agregar NOMBRE AL REPOSITORIO.

2) Ejercicio en Git y Github resueltos

link del repositorio creado:

https://github.com/Ozzetas/Tp2_ejercicio2.git

3) Segundo repositorio creado para resolver este ejercicio:

https://github.com/Ozzetas/Tp2_ejercicio3.git