

Práctico 2: Git y GitHub

Fabrizio Puccio

Actividades

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

- ¿Qué es GitHub?

GitHub es una plataforma de trabajo colaborativo basada en la nube, a los programadores les permite almacenar código y compartirlo. Además permite la gestión del código a través de la herramienta de control de versiones Git.

- ¿Cómo crear un repositorio en GitHub?

Debemos tener una cuenta en GitHub e iniciar sesión en la plataforma.

Hay mas de una forma de crear un repositorio, pero desde la primer página que vemos al iniciar nuestra cuenta en GitHub podemos hacer clic en el botón de la esquina superior derecha que tiene el símbolo +, se despliega un menú donde se encuentra la opción new repository que debemos seleccionar, nos lleva a una página con el título de Create a new repository para configurar nuestro repositorio dándole un nombre, descripción, visibilidad y para lograr la creación del repositorio debemos hacer clic en el botón Create repository.

- ¿Cómo crear una rama en Git?

Para crear una nueva rama debemos usar el siguiente comando:

```
git branch nombre-rama
```

- ¿Cómo cambiar a una rama en Git?

Debemos utilizar el siguiente comando para cambiar a una rama:

```
git checkout nombre-de-la-rama
```

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas debemos hacer un merge.

Nos posicionamos en la rama donde queremos traer los cambios con el comando git checkout nombre-de-la-rama.

Para unir las ramas y traer los cambios a donde nos encontramos utilizamos el comando:

```
git merge nombre-de-la-rama-a-unir
```

- ¿Cómo crear un commit en Git?

Para guardar cambios que realizamos en nuestro proyecto tenemos que utilizar los siguientes comandos:

```
git add .
```

Se añaden los nuevos archivos realizados o cambios a el área de preparación.

Se puede especificar el archivo que quisiéramos de la siguiente manera:

```
git add nombre-del-archivo
```

Ejemplo:

```
git add index.html
```

Luego hacemos un commit agregando un mensaje descriptivo.

```
git commit -m "Descripción del cambio o nueva función, información, etc."
```

- ¿Cómo enviar un commit a GitHub?

Para subir en nuestro repositorio remoto en GitHub los cambios del proyecto usamos:
git push origin nombre-de-la-rama

- ¿Qué es un repositorio remoto?

Un repositorio remoto es un lugar donde podemos guardar información, la plataforma de GitHub nos permite almacenar repositorios en la nube, colaborar con otros desarrolladores, mantener copias de seguridad, etcétera.

- ¿Cómo agregar un repositorio remoto a Git?

Si ya tenemos un repositorio remoto, si ya lo hemos creado en GitHub, lo primero que debemos hacer es vincularlo con el repositorio local con el comando:

```
git remote add origin URL-del-repositorio
```

De esta manera agregamos o vinculamos nuestro repositorio local con el remoto.

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios a un repositorio remoto ingresamos el siguiente comando:

```
git push -u origin nombre-rama
```

nombre-rama debe ser nuestra rama principal main o master.

- ¿Cómo tirar de cambios de un repositorio remoto?

Desde un repositorio remoto podemos traer los cambios a nuestro repositorio local y fusionarlos con nuestro código en local con el siguiente comando:

```
git pull origin nombre-de-la-rama
```

Si solo se quiere descargar los cambios sin fusionarlos usamos:

```
git fetch origin
```

Esto nos permite ver las diferencias y luego fusionarlas manualmente .

- ¿Qué es un fork de repositorio?

Un fork en GitHub es hacer una copia de un repositorio que alguien subió a la plataforma a nuestra propia cuenta de GitHub. Esta herramienta es útil para contribuir a proyectos de código abierto y sino quizás experimentar sin afectar al original.

- ¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio debemos seguir los siguientes pasos:

1. Ve al repositorio que quieres forkar en GitHub.
2. En la esquina superior derecha, haz clic en el botón **"Fork"**.
3. GitHub creará una copia del repositorio en tu cuenta.

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

Para crear un pull request en GitHub entramos en nuestro repositorio y seleccionamos la pestaña Pull request, luego hacemos el clic en New pull request.

Debemos asegurarnos que el dato de base repository sea el repositorio original, base sea la rama donde se quieren fusionar los cambios (normalmente main o develop), head repository sea nuestro repositorio forkeado y compare sea la rama con los cambios que se hicieron.

Podemos escribir un título y una descripción con los cambios que se hicieron.

Clic en Create pull request.

- ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción, desde GitHub accedemos a las solicitudes en la pestaña Pull request, se debe primero por buena practica revisar los cambios, la plataforma de GitHub nos muestra un resumen de los archivos modificados, podemos probar los cambios en nuestra maquina sin fusionar y si todo está bien desde GitHub hacemos clic en Merge Pull Request, colocamos un comentario y presionamos Confirm Merge.

Utilizar el comando git pull para actualizar en local.

- ¿Qué es una etiqueta en Git?

Una etiqueta (tag) en Git es un marcador que especifica un punto en la historia del repositorio. Se usan, por ejemplo, para señalar versiones de software y facilitar la gestión del código.

- ¿Cómo crear una etiqueta en Git?

Hay dos tipos de etiquetas, ligeras y anotadas. La última, porque permite rastrear mas detalles, es la más recomendada.

Para crearla usamos el siguiente comando:

```
git tag -a nombre-del-tag -m "Mensaje descriptivo"
```

- ¿Cómo enviar una etiqueta a GitHub?

Por defecto las etiquetas no se suben con git push a nuestro repositorio en remoto, para enviarlas manualmente:

```
git push origin nombre-del-tag
```

- ¿Qué es un historial de Git?

Un historial en Git es el registro de los cambios realizados en un repositorio, es útil para entender la evolución del proyecto, nos permite identificar errores porque podemos rastrear cambios, saber quién los hizo y cuándo.

- ¿Cómo ver el historial de Git?

```
git log
```

Comando para ver el historial completo.

- ¿Cómo buscar en el historial de Git?

Podemos buscar commits relacionados con una palabra clave:

```
git log --grep="palabra clave"
```

- ¿Cómo borrar el historial de Git?

Se puede borrar el historial de git en local sin afectar al remoto utilizando en bash el comando rm -rf .git

Si luego usamos el comando git init se creará un nuevo repositorio vacío en la carpeta actual.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio donde el código y el historial de cambios no son visibles para el público. Solo las personas con permisos específicos pueden acceder a él. Se puede modificar y hacer publico si se desea.

- ¿Cómo crear un repositorio privado en GitHub?

La clave de la creación de un repositorio privado en GitHub es seleccionar como private la visibilidad de nuestro proyecto antes de hacer clic en Create repository.

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

Si quieres que otras personas accedan al repositorio:

Ve a la pestaña "Settings" en el repositorio, en la barra lateral, selecciona "Collaborators and teams", se debe hacer clic en "Add people" y escribir el nombre del usuario de GitHub. Asigna permisos y envía la invitación.

- ¿Qué es un repositorio público en GitHub?

En GitHub, un repositorio es público cuando el código y historial de cambios son visibles para todo el mundo.

- ¿Cómo crear un repositorio público en GitHub?

Al crear un repositorio en la plataforma de GitHub, en la sección de visibilidad, se debe seleccionar la opción "Public".

- ¿Cómo compartir un repositorio público en GitHub?

La forma más fácil de compartir un repositorio público es proporcionando el enlace directo. En la barra de direcciones del navegador, copiamos la URL del repositorio y lo compartimos a quién corresponda.

2) Comparto el link de la actividad 2: <https://github.com/FabrizioPuccio/actividad2>

3) Link del repositorio en GitHub del ejercicio 3: <https://github.com/FabrizioPuccio/conflict-exercise>

