Sergio Montenegro – Comisión 17

Práctico 2: Git y GitHub

ACTIVIDAD 1

• ¿Qué es GitHub?

GitHub es una plataforma en línea que sirve principalmente para desarrollar, compartir y colaborar en proyectos de software. Es muy conocida por su capacidad de alojamiento de repositorios de código, lo que permite a los desarrolladores gestionar versiones de su código y trabajar en equipo de manera eficiente.

• ¿Cómo crear un repositorio en GitHub?

Inicia sesión en GitHub: Ve a GitHub e inicia sesión con tu cuenta. Si no tienes una cuenta, puedes registrarte gratuitamente.

Crea un nuevo repositorio:

- Haz clic en el botón verde **"New"** (Nuevo) que aparece en la esquina superior izquierda o dentro de la pestaña de repositorios de tu perfil.
- Completa los detalles del repositorio:
  - **Nombre del repositorio**: Escoge un nombre único.
  - **Descripción** (opcional): Añade una breve descripción sobre el propósito del repositorio.
  - Selecciona si será **público** (visible para todos) o **privado** (solo accesible para ti y los colaboradores que invites).

Crea el repositorio: Haz clic en el botón **"Create repository"** (Crear repositorio) y listo

• ¿Cómo crear una rama en Git?

git branch nombre-de-la-rama

• ¿Cómo cambiar a una rama en Git?

git checkout nombre-de-la-rama

• ¿Cómo fusionar ramas en Git?

git merge rama-a-fusionar

• ¿Cómo crear un commit en Git?

git commit -m "Mensaje del commit"

• ¿Cómo enviar un commit a GitHub?

git push origin rama

• ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de un repositorio alojada en un servidor o plataforma en línea, como GitHub, GitLab o Bitbucket. Este tipo de repositorio permite que varios desarrolladores colaboren y compartan cambios en un proyecto desde diferentes ubicaciones.

• ¿Cómo agregar un repositorio remoto a Git?

git remote add origin URL-del-repositorio-remoto

• ¿Cómo empujar cambios a un repositorio remoto?

git push origin rama

Sustituye rama por el nombre de la rama a la que deseas subir los cambios, como main o master.

• ¿Cómo tirar de cambios de un repositorio remoto?

git pull origin rama

Donde origin es el nombre por defecto del repositorio remoto y rama es la rama desde la cual quieres traer los cambios.

• ¿Qué es un fork de repositorio?

Un **fork** de un repositorio es una copia de un repositorio existente que se crea en tu cuenta en una plataforma como GitHub. Permite que trabajes de forma independiente en el código original sin afectar directamente al repositorio fuente. Es especialmente útil para colaborar en proyectos públicos o para experimentar con cambios antes de proponerlos al proyecto original.

• ¿Cómo crear un fork de un repositorio?

Encuentra el repositorio que deseas "forkear".

En la parte superior derecha de la página del repositorio, encontrarás un botón que dice **Fork**.

Haz clic en él

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

Una **solicitud de extracción (pull request)** en GitHub permite proponer cambios desde tu rama o fork hacia un repositorio original. Es una forma de colaborar y someter tus contribuciones para que otros las revisen e integren.

Pasos para enviar una Pull Request:

1. **Asegúrate de haber realizado los cambios en tu fork o rama**:
    - Haz todos los cambios necesarios en tu código y confirma (commitea) esos cambios en tu rama local o en el fork.
2. **Empuja tus cambios al repositorio remoto**:
    - Asegúrate de que los cambios están en el servidor de GitHub usando:
    - git push origin nombre-de-la-rama
3. **Accede al repositorio original en GitHub**:
    - Ve al repositorio original (el destino de la solicitud) en GitHub.
4. **Inicia la solicitud de extracción**:
    - Si trabajas en un fork, aparecerá un botón en la página de tu repositorio que dice **"Compare & pull request"** (Comparar y solicitar extracción). Haz clic en él.
    - Si trabajas en una rama dentro del mismo repositorio, selecciona la pestaña **"Pull Requests"** y haz clic en **"New Pull Request"**.
5. **Selecciona las ramas que deseas fusionar**:
    - Asegúrate de seleccionar tu rama o fork como la **rama de origen** (source branch) y la rama del repositorio original como **rama de destino** (base branch).
6. **Escribe un título y descripción clara**:
    - El título debe ser conciso y directo, como "Añade nueva funcionalidad X".

    - En la descripción, explica qué cambios hiciste, por qué son importantes y cualquier información relevante para que los revisores comprendan tu trabajo.
7. **Revisa y envía**:
    - Asegúrate de que los cambios a fusionar sean los correctos.
    - Haz clic en **"Create Pull Request"** (Crear solicitud de extracción).
8. **Espera revisión**:
    - Los colaboradores del proyecto revisarán tu solicitud y, si todo está correcto, la fusionarán o te pedirán realizar ajustes.

• ¿Cómo aceptar una solicitud de extracción?

Aceptar una solicitud de extracción (**pull request**) significa integrar los cambios propuestos en una rama específica de tu repositorio.

1. **Ve a la sección de Pull Requests**:
    - Accede a tu repositorio en GitHub.
    - Haz clic en la pestaña **"Pull Requests"** para ver todas las solicitudes pendientes.
2. **Revisa la solicitud de extracción**:
    - Selecciona la solicitud que deseas aceptar.
    - Revisa cuidadosamente los cambios propuestos:
        - Examina el código, las líneas modificadas y el propósito del pull request.
        - Si tienes dudas, puedes dejar comentarios o pedir cambios al autor.
3. **Prueba los cambios (opcional pero recomendable)**:
    - Clona la rama del pull request en tu máquina local para probar los cambios:
    - git fetch origin nombre-de-la-rama
    - git checkout nombre-de-la-rama
      Asegúrate de que los cambios no introducen errores ni conflictos.
4. **Resuelve conflictos (si los hay)**:
    - Si hay conflictos entre la rama principal y el pull request, GitHub te notificará. Resuelve los conflictos manualmente desde la interfaz web o en tu máquina local.
5. **Fusiona la solicitud**:
    - Una vez que estés satisfecho con los cambios, haz clic en el botón verde **"Merge pull request"** (Fusionar solicitud de extracción).
    - Opcionalmente, puedes añadir un comentario final antes de confirmar.
6. **Elimina la rama (opcional)**:
    - Tras fusionar los cambios, puedes eliminar la rama asociada al pull request desde GitHub haciendo clic en el botón **"Delete branch"**. Esto ayuda a mantener limpio el repositorio.

• ¿Qué es un etiqueta en Git?

Una **etiqueta (tag)** en Git es un marcador especial que se utiliza para señalar puntos específicos en el historial de commits, generalmente para versiones importantes de un proyecto (como lanzamientos o releases). Las etiquetas son muy útiles para identificar y referenciar commits de forma sencilla.

• ¿Cómo crear una etiqueta en Git?

- **Etiqueta ligera**:
  git tag nombre-de-la-etiqueta
- **Etiqueta anotada**:
  git tag -a nombre-de-la-etiqueta -m "Mensaje descriptivo"

• ¿Cómo enviar una etiqueta a GitHub?

- Para enviar una etiqueta específica:
  git push origin nombre-de-la-etiqueta
- Si quieres enviar todas las etiquetas de una vez:
  git push origin --tags

• ¿Qué es un historial de Git?

El **historial de Git** es el registro de todos los cambios realizados en un repositorio a lo largo del tiempo. Contiene una lista completa de los commits, que son las "instantáneas" de los cambios hechos al código. Este historial es esencial para entender cómo ha evolucionado un proyecto, quién realizó qué cambios y cuándo se hicieron.

¿Qué información incluye el historial?

1. **Commits**:
   - Cada commit incluye un identificador único (hash), un mensaje descriptivo, la fecha y hora del cambio, y el autor.
2. **Ramas**:
   - El historial puede mostrar cómo las diferentes ramas han contribuido al desarrollo.
3. **Fusiones (merges)**:
   - También se registran los eventos donde ramas han sido fusionadas.

• ¿Cómo ver el historial de Git?

- **Historial básico:**
  git log
  Muestra una lista de commits con sus detalles.
- **Formato compacto:**
  git log --oneline
  Presenta los commits en una sola línea por cada uno, ideal para visualización rápida.
- **Con grafo visual:**
  git log --oneline --graph --all
  Esto muestra las ramas y sus fusiones de manera gráfica.

• ¿Cómo buscar en el historial de Git?

1. Búsqueda por mensaje del commit:

Si sabes palabras clave del mensaje del commit, puedes usar:
git log --grep="palabra clave"
Esto mostrará solo los commits cuyo mensaje contenga las palabras que especifiques.

2. Búsqueda de cambios en archivos:

Si deseas ver qué commits afectaron un archivo específico:
git log -- nombre-del-archivo
Esto lista todos los commits que hicieron cambios en ese archivo.

3. Búsqueda por autor:

Para ver los commits realizados por un autor específico:
git log --author="nombre-del-autor"

4. Búsqueda de contenido modificado:

Si quieres encontrar un commit donde se agregó o modificó un contenido específico:

```
git log -S "texto o código específico"
```

Este comando busca el texto proporcionado en los cambios realizados.

5. Formato compacto para facilitar lectura:

Puedes combinar filtros con un formato más simple para obtener resultados rápidos:

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

6. Visualización de diferencias específicas:

Si quieres ver los cambios exactos que introdujo un commit en el contenido, puedes usar:

```
git show identificador-del-commit
```

• ¿Cómo borrar el historial de Git?

rm -rf .git

• ¿Qué es un repositorio privado en GitHub?

Un **repositorio privado** en GitHub es un tipo de repositorio al que solo tú y las personas que invites pueden acceder. A diferencia de los repositorios públicos, los privados no están visibles para el público en general, lo que los hace ideales para proyectos que contienen información confidencial, código en desarrollo o colaboraciones privadas.

Características principales de un repositorio privado:

1. **Control de acceso**:
   - Solo los usuarios que invites como colaboradores podrán ver o contribuir al repositorio.
2. **Propósito**:
   - Ideal para proyectos personales, internos de una empresa o colaboraciones privadas que no deseas compartir públicamente.
3. **Funciones completas**:
   - Un repositorio privado tiene las mismas capacidades que uno público, como el manejo de issues, pull requests, wikis, etc.
4. **Gratis para proyectos pequeños**:
   - GitHub permite repositorios privados sin costo para usuarios individuales o pequeños equipos (aunque existen límites en el número de colaboradores en cuentas gratuitas).
5. **Confidencialidad y seguridad**:
   - El contenido no será indexado ni visible públicamente, protegiendo datos sensibles o en desarrollo.

• ¿Cómo crear un repositorio privado en GitHub?

Cuando creas un nuevo repositorio en GitHub, simplemente selecciona la opción **"Private" (Privado)** durante el proceso de configuración

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

Accede a tu repositorio privado:

- Ve a GitHub e inicia sesión en tu cuenta.
- Selecciona el repositorio privado en el que deseas añadir colaboradores.

Abre la configuración del repositorio:

- Haz clic en la pestaña **"Settings"** (Configuración) ubicada en la parte superior del repositorio.

Ve a la sección de "Collaborators" (Colaboradores):

- En la barra lateral izquierda, busca y haz clic en la opción **"Collaborators"** dentro del apartado **"Access"** o similar.

Invita al colaborador:

- En el cuadro de texto que aparece, escribe el nombre de usuario de GitHub o la dirección de correo electrónico de la persona que deseas invitar.
- Haz clic en **"Add collaborator"** (Añadir colaborador).

Confirma la invitación:

- GitHub enviará una invitación al usuario. Este deberá aceptarla para poder acceder al repositorio.
- Puedes verificar el estado de la invitación dentro de la misma sección de colaboradores.

• ¿Qué es un repositorio público en GitHub?

Un **repositorio público** en GitHub es un repositorio al que cualquier persona puede acceder y visualizar. Es ideal para compartir proyectos, colaborar abiertamente y contribuir al conocimiento colectivo de la comunidad de desarrollo. Aquí tienes más detalles sobre sus características:

Características de un repositorio público:

1. **Accesibilidad**:
    - Todos pueden ver el contenido del repositorio, aunque solo los colaboradores con permiso podrán realizar cambios directos.
    - Otros usuarios pueden realizar forks del repositorio para trabajar en sus propias versiones.
2. **Colaboración abierta**:
    - Permite recibir contribuciones de cualquier persona mediante **pull requests**, promoviendo el trabajo en equipo y la mejora constante del proyecto.
3. **Uso común**:
    - Se utiliza ampliamente en proyectos de código abierto, donde los desarrolladores comparten su trabajo para que otros lo usen, modifiquen y aprendan de él.
4. **Indexación**:
    - Los repositorios públicos pueden aparecer en motores de búsqueda, lo que ayuda a que más personas descubran y contribuyan a tu trabajo.
5. **Gratuidad**:
    - GitHub permite crear repositorios públicos sin costo, lo que facilita su uso para proyectos de cualquier tamaño.

• ¿Cómo crear un repositorio público en GitHub?

1. Inicia sesión en GitHub:

- Ve a GitHub e inicia sesión con tu cuenta.

2. Crea un nuevo repositorio:

- Haz clic en el botón verde **"New"** (Nuevo) que aparece en la esquina superior izquierda o en tu perfil.

3. Configura el repositorio:

- **Nombre del repositorio**: Elige un nombre único y descriptivo.
- **Descripción** (opcional): Explica brevemente la finalidad del proyecto.
- **Visibilidad**: Selecciona **"Public"** (Público) para que sea accesible a todos.

4. Opciones iniciales (opcionales):

- Puedes agregar un archivo README.md inicial para documentar el propósito del proyecto.
- Añadir un archivo .gitignore para excluir ciertos archivos o directorios.
- Seleccionar una licencia para definir cómo otros pueden usar tu código (como MIT, GPL, etc.).

5. Crea el repositorio:

- Haz clic en **"Create repository"** (Crear repositorio).

• ¿Cómo compartir un repositorio público en GitHub?

1. Comparte la URL del repositorio:

- Ve a la página del repositorio en GitHub.
- Copia la URL que aparece en la barra de direcciones del navegador (algo como https://github.com/tu-usuario/nombre-del-repositorio).
- Comparte esta URL con quien desees, ya sea por correo electrónico, mensajería o publicándola en otras plataformas.