P5. Acerca de GitHub

GitHub es una plataforma donde puedes almacenar, compartir y trabajar junto con otros usuarios el código.

Al almacenar el código en un "repositorio" en GitHub te permite:

- **Presentar o compartir** el trabajo.
- **Seguir y administrar** los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
- **Colaborar** en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, es una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

Acerca de Git

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas que están haciendo cambios en los mismos archivos al mismo tiempo.

Normalmente, para hacerlo en un flujo de trabajo basado en Git, harías lo siguiente:

- **Crear una rama** a partir de la copia principal de archivos en los que tú (y tus colaboradores) estáis trabajando.
- **Realizar modificaciones** en los archivos de forma independiente y segura en tu propia rama personal.
- Dejar que Git **fusione mediante combinación** y de forma inteligente los cambios específicos en la copia principal de archivos, de modo que los cambios no afecten a las actualizaciones de otras personas.
- Dejar que Git **realice un seguimiento** de tus cambios y los de otras personas, por lo que todos siguen trabajando en la versión más actualizada del proyecto.

¿Cómo funcionan Git y GitHub juntos?

Al cargar archivos en GitHub, los almacenas en un "repositorio de Git". Esto significa que al realizar cambios en los archivos de GitHub, Git se iniciará automáticamente para realizar un seguimiento y administrar los cambios.

Hay muchas acciones relacionadas con Git que puedes completar en GitHub directamente en el navegador, como crear un repositorio de Git, crear ramas y cargar y editar archivos.

Sin embargo, la mayoría de las personas trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando empieces a trabajar con otros en el mismo repositorio al mismo tiempo, se puede hacer:

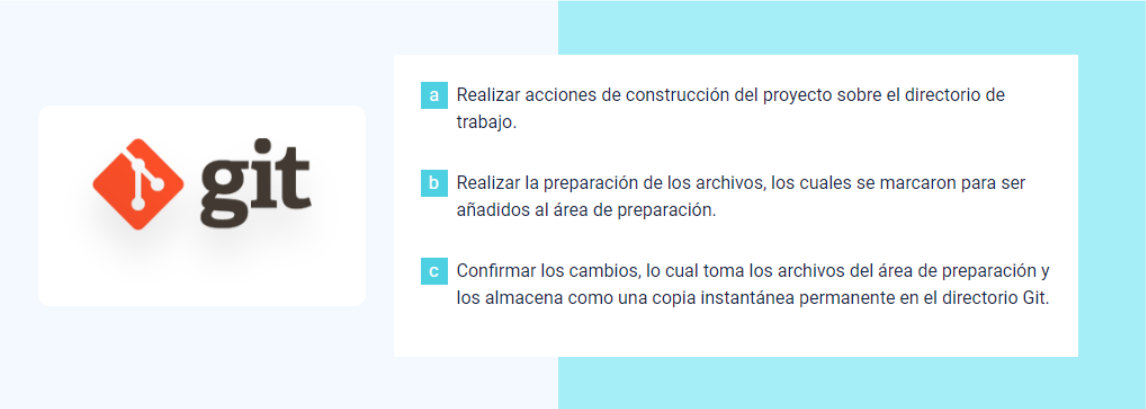
- **Extraer** todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub.
- **Transferir de vuelta** los cambios al mismo repositorio remoto en GitHub.

Git determina cómo combinar inteligentemente este flujo de cambios y GitHub te ayuda a administrar el flujo a través de características como "solicitudes de cambios".

¿Dónde empiezo?

Las tareas que puedes completar directamente en el explorador en GitHub, te ayudarán a:

- **Crear una cuenta** en GitHub.
- **Aprender el "flujo GitHub"** y los principios clave del trabajo colaborativo (ramas, compromisos, solicitudes de cambios, fusiones por combinación).
- **Personaliza tu perfil** para compartir tus intereses y aptitudes.
- **Explora GitHub** para encontrar inspiración para tus propios proyectos y conectarte con otros usuarios.
- Obtén información sobre cómo **descargar** código interesante para tu propio uso.
- Obtén información sobre cómo **cargar** algo en lo que estás trabajando en un repositorio GitHub.



P6. "Plataformas para implementar integración continua – GitLab".

El texto introductorio indica que existen múltiples plataformas en la nube que permiten integración continua, con funciones como:

- Creación de repositorios públicos y privados.
- Integración con sistemas de control de versiones.
- Registro de grupos de trabajo y colaboración.

Luego se presentan tres plataformas principales:

1. GitHub
    - Portal para alojar código de aplicaciones.

    - Facilita repositorios públicos y privados.
    - URL: https://github.com.
2. GitLab
    - Servicio web que controla versiones y permite desarrollo colaborativo basado en Git.
    - URL: https://about.gitlab.com.
3. Bitbucket
    - Servicio de alojamiento basado en web para proyectos que usan Mercurial y Git.
    - URL: https://bitbucket.org.

📖 Explicación del tema

Estas tres plataformas son ampliamente utilizadas en la industria para la integración continua y control de versiones:

- GitHub → muy popular en proyectos open source y empresas. Se integra con GitHub Actions para CI/CD.
- GitLab → integra de forma nativa CI/CD en la misma plataforma (pipelines).
- Bitbucket → ofrece integración con Atlassian (Jira, Confluence) y Bitbucket Pipelines.

✅ En conclusión:
La imagen resume que, además de GitLab, existen otras plataformas clave como GitHub y Bitbucket que cumplen el mismo propósito de gestionar repositorios y habilitar la integración continua en proyectos de software.

P12. Comandos de GIT básicos

Aquí hay algunos comandos básicos de GIT que debes conocer:

- **git init** creará un nuevo repositorio local GIT. El siguiente comando de Git creará un repositorio en el directorio actual:

git init

- Como alternativa, puedes crear un repositorio dentro de un nuevo directorio especificando el nombre del proyecto:

git init [nombre del proyecto]

- **git clone** se usa para copiar un repositorio. Si el repositorio está en un servidor remoto, usa:

git clone nombredeusuario@host:/path/to/repository

- A la inversa, ejecuta el siguiente comando básico para copiar un repositorio local:

git clone /path/to/repository

- **git add** se usa para agregar archivos al área de preparación. Por ejemplo, el siguiente comando de Git básico indexará el archivo temp.txt:

git add <temp.txt>

- **git commit** creará una instantánea de los cambios y la guardará en el directorio git.

git commit –m “El mensaje que acompaña al commit va aquí”

- **git status** muestra la lista de los archivos que se han cambiado junto con los archivos que están por ser preparados o confirmados.

git status

- **git remote** te permite ver todos los repositorios remotos. El siguiente comando listará todas las conexiones junto con sus URLs:

git remote -v

- Para conectar el repositorio local a un servidor remoto, usa este comando:

git remote add origin <host-or-remoteURL>

- Por otro lado, el siguiente comando borrará una conexión a un repositorio remoto especificado:

git remote <nombre-del-repositorio>

Presentacion:

1. Descarga **Git**
2. En **cmd** ejecuta **git versión**
3. Creamos una carpeta de Windows, luego abrimos **visual code studio** la pegamos en ella.
4. En visual creamos archivos **index.html** y colocamos una estructura básica de html con **!** para que salga la información de html.
5. Luego creamos **nosotros.html** y una estructura básica de html con el símbolo **!** y guardamos.
6. Dentro de visual abrimos la terminal.
7. Registramos nuestro usuario en Git: **git config --global user.name "mi nombre"** juanma
8. Si ejecuto el comando de nuevo me va a mostrar el usuario creado: **git config --global user.name**
9. Con **cls** limpiamos la terminal

10. Ahora asociamos el correo con el **comando git config --global user.email "myemail@example.com"**
11. Luego verificamos si quedo el correo registrado con el comando **git config --global user.email**
12. Con el comando **git init** se inicializa un proyecto con git apenas lo ejecutemos verifica que los dos archivos index y nosotros.html se colocan en color verde y aparece una letra U al lado, esto significa que ya estamos trabajando con git.
13. En el explorador de Windows nos aperece una carpeta oculta de .git, si no aparece vamos al menú de vista y activamos la casilla de elementos ocultos.
14. Ahora vamos a guardar los archivos de visual en una memoria virtual de git con el comando **git add .** y va a tomar todos los archivos al repositorio de git.
15. Ahora generamos nuestro primer comic, esto es tomar una foto de los cambios que se realizaron y se guarda con un nombre, cada vez que se haga un cambio hay que repetir este proceso para identificar que cambios se realizaron. Git commit -m “mi primer commit”
16. Después de ejecutar el comando los archivos de visual desaparece el color y esto indica que se creo el primer repositorio y quedo en el repositorio local.
17. Si deseo visualizar todos los commiy que se hayan creado se ejecuta con el comando. **Git log –oneline**
18. En este momento solo se a trabajado local en git, que sucede si hacemos unos cambio en los archivos, ejemplo cambiar el idioma y colocar un texto con h1 lorem y guardamos los cambios en visual los archivos aparecen una M en el archivo modificado y eso aparece por que cada cambio se ve reflejado en el repositorio de git, cuando se desee actulizarlo en el repositorio se ejecuta el comando **git add .** y hace un barrido en todos los archivos y los vuelve a subir a la parte amarilla como se ve en la imagen.
19. Ahora se guarda con un mensaje para llevar el historial de cambios y eso se hace con commit, **git commit -m “cambie mi pagina al español”.**
20. Ahora si desea revisar los commit generados se ejecuata **git log --oneline** y te sale los mensajes generados y cambio hechos.
21. Ahora vamos a subir los archivos a GitHub, importante crear un usuario en la plataforma, cuando lo creas te pide un usuario, una contraseña y correo electrónico, yo lo realice con una de Gmail, al correo que registro le envia un código de 6 numeros y ese código hay que registrarlo en la pagina de github, si les preguna que plan va a manejar escoja el gratuito de 500 mb y ya ingresamos a la pagina.
22. Ya estando dentro de la pagina vamos a crear un repositorio, en la parte supeior derecha aparece un mas (+), hay escojemos un nuevo repositorio, le asignamos un nombre para mi caso dejo mi primer repo, lo dejamos public y crear repositorio.
23. Ahora nos recomienda algunos comando para subir el repositorio local a la nube. Yo escojo el segundo es mas recomendado.

…or push an existing repository from the command line

```
git remote add origin https://github.com/jmaldanaz/Mi-primer-Repo.git
git branch -M main
git push -u origin main
```

24. El proceso lo hace sobre HTTPS o también se puede hacer por SSH si lo desea, dejamos por default que es HTTPS, seleccionamos el primer comando **git remote add origin** **https://github.com/jmaldanaz/Mi-primer-Repo.git** y lo copiamos en la ventana de

cmd de visual. Tomamos la segunda línea la copiamos y hacemos el mismo paso **git branch -M main** y hacemos con el ultimo comando la línea tres **git push -u origin main**

25. Cuando ejecutemos la ultima línea nos abre una ventana de autorización, solicitando una credencial, dale clip al enlace y te sale el permiso de autorización.
26. Cuando sale la autorización en la ventana de CMD de visual nos muestra los archivos subidos en el repositorio de la pagina web https://github.com/jmaldanaz/Mi-primer-Repo.git ahora nos desplazamos a la pagina donde estaban los comando , la actualizamos y hay nos aparece los archivos subidos.
27. En la pagina aparece un icono de reloj que son los commit realizados so le das clip los puede ver los dos que creamos.
28. Ahora que paso si el día de mañana actualizamos el otro archivo del proyecto, en este caso el index.html lo cambiamos a español **en a es, a**gradamos un texto H1 Lorem3 que es un texto de 3 palabras, guardamos los cambios y en visual studio aparece la M y el color en los archivos modificados y se repite el proceso en la terminal de Visual.
29. Se digita los comandos 1. **git add .** luego el mensaje con commit para registrar el cambio 2. **Git commit -m "cambien mi index a español"**, luego revisamos que se cargo el commit con el comando3**. git log –oneline** y por ultimo lo subimos a github con el comando 4. **Git push** , por que ya se había hecho la vinculación antes, queda pendiente actualizar la pagina web para ver los cambios.
30. Si desea abre el index.html mira los cambio y adicional los commit aumetaron a 3, antes había 2.
31. Por ultimo si desean compartir con otras personas solo debe compartir el enlace y estar este repositorio publico.

Descarga Git

https://git-scm.com/downloads/win

Pagina Github

https://github.com/

¿Cómo subir mi primer repositorio a Github? [Tutorial desde cero]

https://www.youtube.com/watch?v=L_lWQZNhN7w

- **Como crear un Repositorio y Subir Proyecto a 👉GITHUB👈 Paso a Paso**

https://www.youtube.com/watch?v=eQMcIGVc8N0

Flujo de trabajo de GIT

