

Herramientas de Desarrollo

Semana 06

Configuración de sistema de control de versiones en la nube y flujo básico de trabajo



**Universidad
Tecnológica
del Perú**

Inicio

¿Tienen alguna consulta o duda sobre la clase anterior?



Logro de la Unidad

Al finalizar la unidad, el estudiante gestiona los sistemas de control de versiones para el desarrollo de una solución de software.



Utilidad

- ¿Porqué es importante saber de configuraciones?

contenido

1 Introducción

2 Pasos para Configurar un Repositorio en la Nube

3 Flujo Básico de Trabajo en Git con Repositorios Remotos

4 Enlace

Transformación

1 Introducción

Introducción

La **configuración de un sistema de control de versiones en la nube** implica el uso de plataformas como **GitHub**, **GitLab**, **Bitbucket**, entre otras, que permiten gestionar y almacenar el código fuente de manera remota.

El uso de estas plataformas facilita la colaboración, el seguimiento de cambios y la gestión de versiones de un proyecto de software, permitiendo que varios desarrolladores trabajen simultáneamente

2 Pasos para Configurar un Repositorio en la Nube

Pasos para Configurar un Repositorio en la Nube



Crear una cuenta en una plataforma de control de versiones (GitHub, GitLab, Bitbucket, etc.):

Dirígete al sitio web de la plataforma de tu elección (por ejemplo, GitHub) y crea una cuenta si aún no tienes una.

Crear un nuevo repositorio en la nube:

En plataformas como **GitHub**, una vez que inicies sesión, puedes crear un nuevo repositorio al hacer clic en el botón “**New Repository**”.

Luego, debes asignar un nombre y una descripción a tu repositorio, decidir si será público o privado y finalmente crear el repositorio.

Pasos para Configurar un Repositorio en la Nube



Configurar tu repositorio local con el repositorio remoto:

En tu máquina local, inicializa un repositorio Git (si no lo has hecho ya):

```
git init
```

Conecta el repositorio local con el remoto usando el comando git remote add:

```
git remote add origin https://github.com/usuario/nombre-del-repositorio.git
```

Subir el código al repositorio remoto:

Realiza tus primeros commits locales, luego usa el comando git push para enviar esos commits al repositorio remoto:

```
git add .
```

```
git commit -m "Primer commit"
```

```
git push -u origin main
```

Esto subirá tu código al repositorio remoto en la rama principal (main o master).

3 Flujo Básico de Trabajo en Git con Repositorios Remotos

Flujo Básico de Trabajo en Git con Repositorios Remotos



Clonación del repositorio remoto:

Si otra persona ya ha creado un repositorio y deseas trabajar en él, puedes clonarlo:

```
git clone https://github.com/usuario/repositorio.git
```

Trabajo en una rama local:

Crea una nueva rama para trabajar en una característica o corrección específica:

```
git checkout -b feature/nueva-caracteristica
```

Realización de cambios y commits:

Realiza cambios en los archivos del proyecto, agrega esos cambios al área de preparación y luego haz un commit:

```
git add .
```

```
git commit -m "Descripción de los cambios"
```

Flujo Básico de Trabajo en Git con Repositorios Remotos



Sincronización con el repositorio remoto:

Antes de hacer un push, siempre es buena idea hacer un pull para traer los cambios más recientes del repositorio remoto:

```
git pull origin main
```

Subir tus cambios al repositorio remoto:

Después de resolver cualquier posible conflicto, sube los cambios:

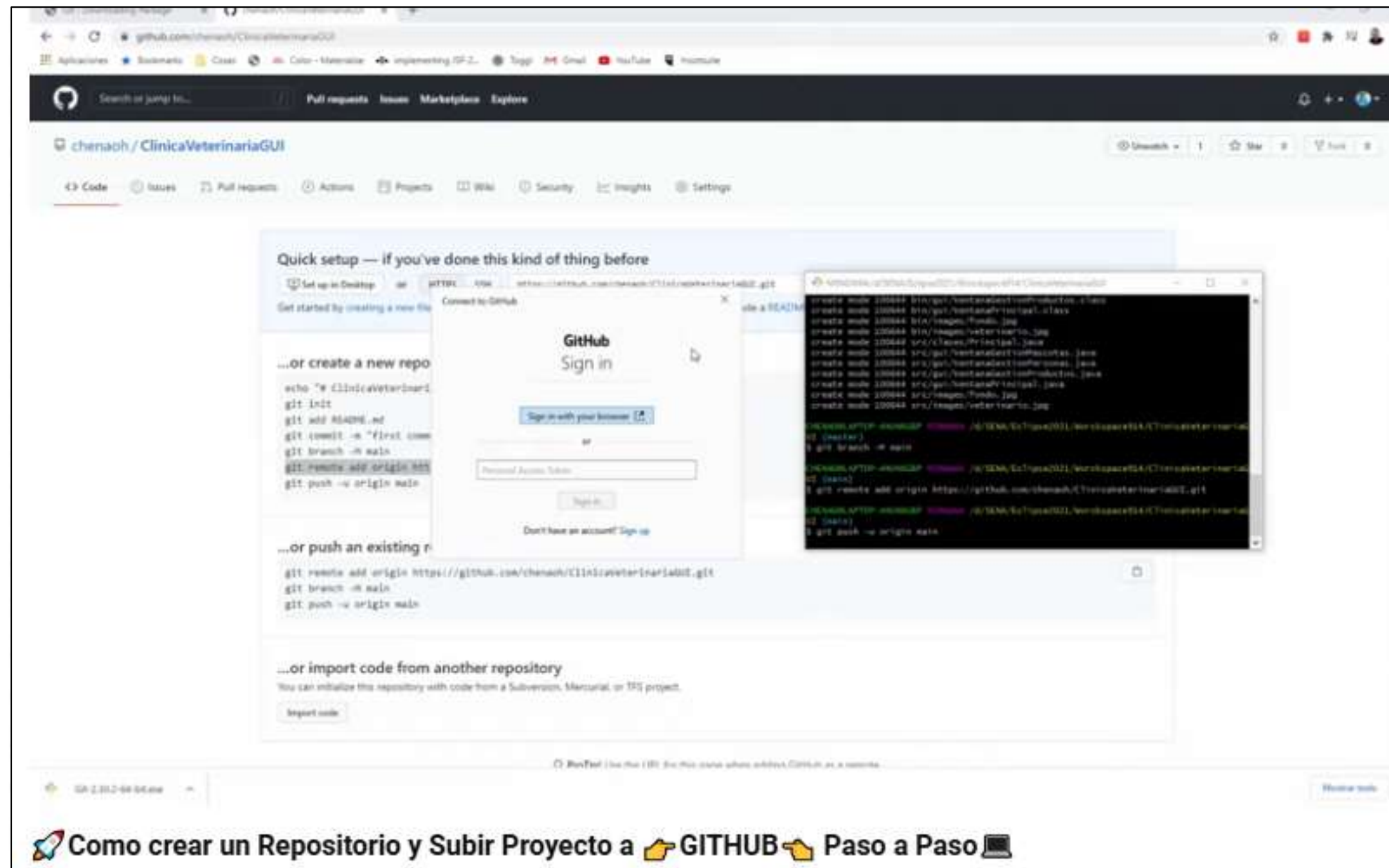
```
git push origin feature/nueva-caracteristica
```

Crear un pull request (PR) o merge request (MR):

Una vez que tu rama está lista, puedes crear un PR (en GitHub) o MR (en GitLab) para que otros revisen y fusionen tus cambios con la rama principal (main o master).

4 Enlace

Como crear un Repositorio y Subir Proyecto a GITHUB Paso a Paso



<https://www.youtube.com/watch?v=eQMclGVc8N0>

Práctica

Clonar un repositorio remoto a tu máquina local

Agregar un repositorio remoto a tu proyecto local

Actualiza tu rama local con los últimos cambios del repositorio remoto antes de hacer un push

Agregar todos los archivos modificados y hacer un commit con un mensaje

Cierre

1. ¿Qué es un repositorio remoto en el contexto de Git?
2. ¿Por qué es importante hacer git pull antes de git push?
3. ¿Qué significa la opción -u en el comando git push -u origin main?
4. ¿Qué hace el comando git remote add origin?



Bibliografía

Hernández Bejarno, Miguel. *Ciclo de vida de desarrollo ágil de software seguro*. Fundación Universitaria Los Libertadores. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36016>

⁸ Guillamón Morales, Alicia. (). *Manual desarrollo de elementos software para gestión de sistemas*.

Editorial CEP, S.L. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=34982>

Chacon, S., & Straub, B. (2014). *Pro Git* (2nd ed.). Apress.
Enlace: <https://git-scm.com/book/es/v2>

Poulton, N. (2017). *Docker Deep Dive*. Independently published.
Enlace: <https://www.nigelpoulton.com/dvd/>



**Universidad
Tecnológica
del Perú**