Repositorio Remoto

Contenido

1. Introducción.

En el mundo del desarrollo de software, la colaboración eficiente y la gestión de versiones son esenciales para construir proyectos exitosos. En este documento, exploraremos la función vital que desempeñan los repositorios remotos en la facilitación de la colaboración entre equipos y cómo Git, se convierte en la herramienta fundamental para lograrlo.

Un repositorio remoto en Git es una versión de tu proyecto que se encuentra alojada en un servidor o en otro lugar fuera de tu máquina local. Este repositorio remoto puede ser accesible y compartido entre diferentes colaboradores de un proyecto. Los repositorios remotos permiten la colaboración y la sincronización de cambios entre diferentes desarrolladores.

Algunas características clave de un repositorio remoto son:

- **Colaboración**:

Varios desarrolladores pueden trabajar en un proyecto compartido al acceder y clonar el mismo repositorio remoto.

- **Sincronización de cambios**:

Los desarrolladores pueden enviar (push) sus cambios al repositorio remoto para que otros puedan verlos y descargarlos (pull). Esto facilita la colaboración y la integración de cambios.

- **Copia de seguridad**:

El repositorio remoto sirve como una copia de seguridad centralizada del proyecto. Si tu máquina local falla, puedes recuperar el código desde el repositorio remoto.

- **Facilita la distribución**:

Permite distribuir el código a través de equipos geográficamente dispersos. Los desarrolladores pueden trabajar desde diferentes ubicaciones y contribuir al mismo proyecto.

- **Seguimiento de versiones**:

Los repositorios remotos mantienen un historial de versiones del proyecto, lo que facilita la colaboración y la gestión de cambios a lo largo del tiempo.

Los repositorios remotos suelen estar alojados en plataformas como GitHub, GitLab, Bitbucket u otros servicios de alojamiento de Git. Estas plataformas ofrecen funciones adicionales, como seguimiento de problemas, solicitudes de extracción y otras herramientas de colaboración.

2. Iniciando el repositorio remoto.

En esta sección, exploraremos los pasos necesarios para crear una cuenta en **GitLab** y realizar la configuración inicial para comenzar a trabajar con repositorios remotos.

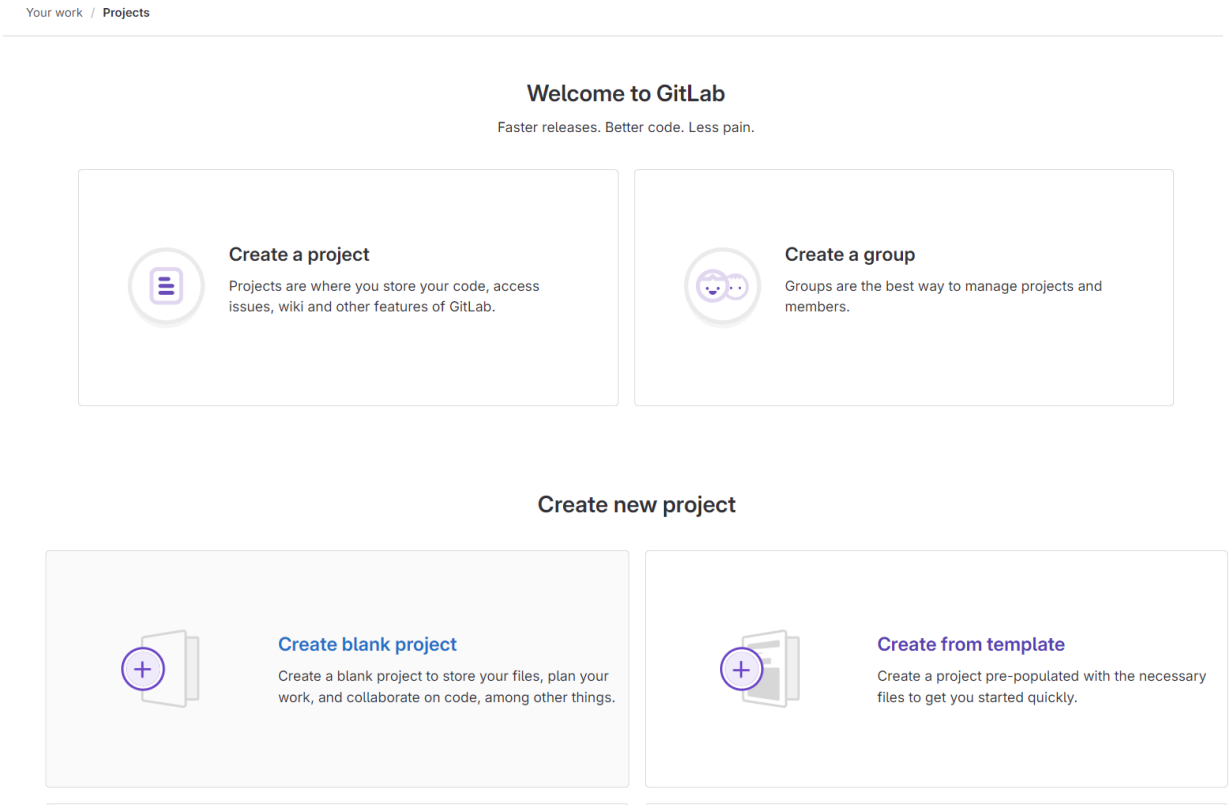
En primer lugar, debemos crear una cuenta en el [sitio web de Gitlab](). Una vez hecho esto, pasamos a configurar en nuestro equipo el nombre de usuario y la dirección de correo electrónico con la que nos vamos a identificar en el repositorio. Para ello abrimos el terminal de git utilizamos los siguientes comandos:

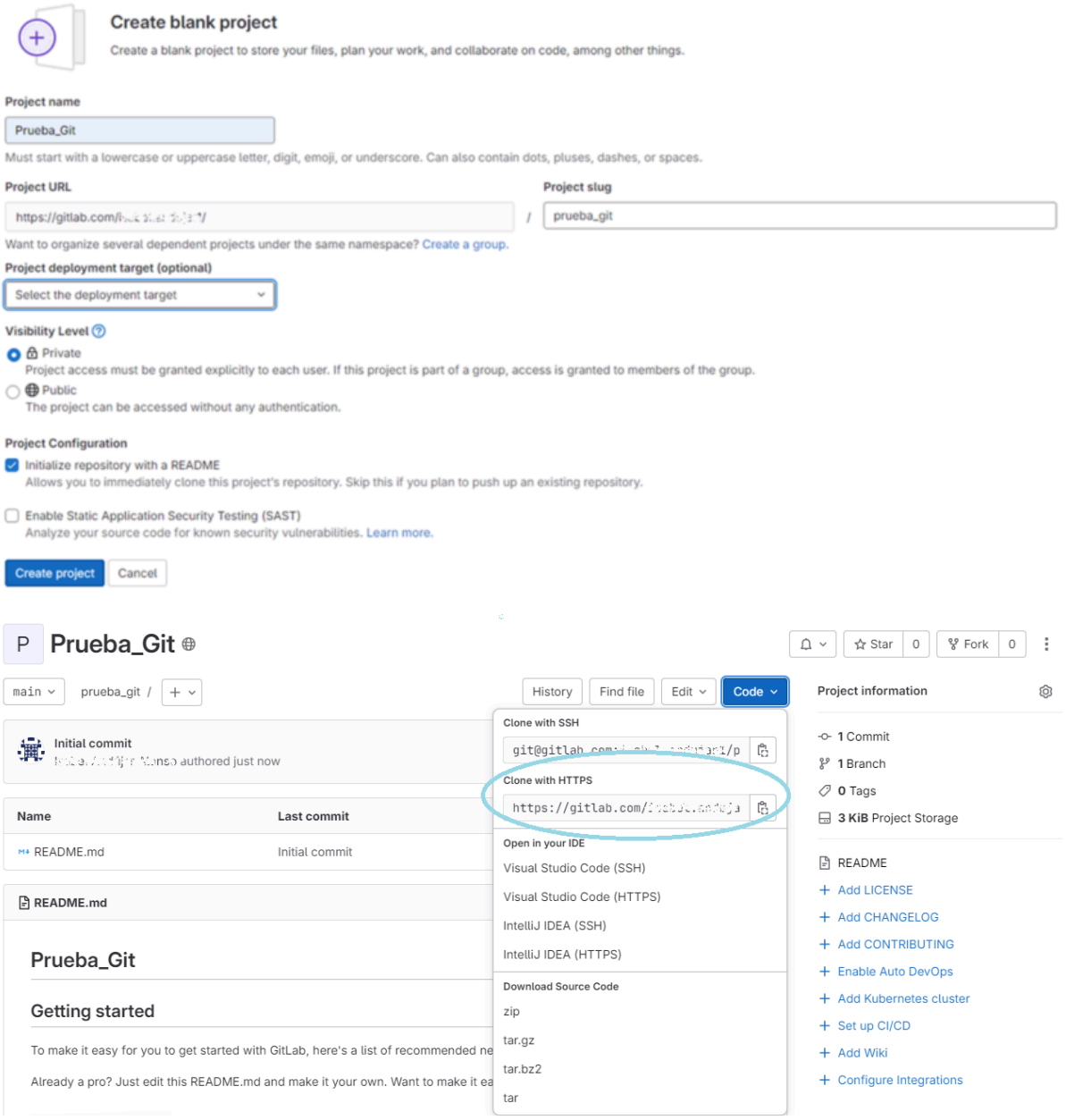
El comando ***git config*** puede ser utilizado principalmente de dos formas:

1. `'git config --global user.name "Tu Nombre"'`
2. `'git config --global user.email "tu@email.com"'`

El parámetro --global sirve para fijar esa información de forma global para todos los proyectos. En el caso de que queramos modificar esta configuración para algún proyecto en particular, simplemente tenemos que ejecutar los comandos sin ese parámetro el parámetro --global.

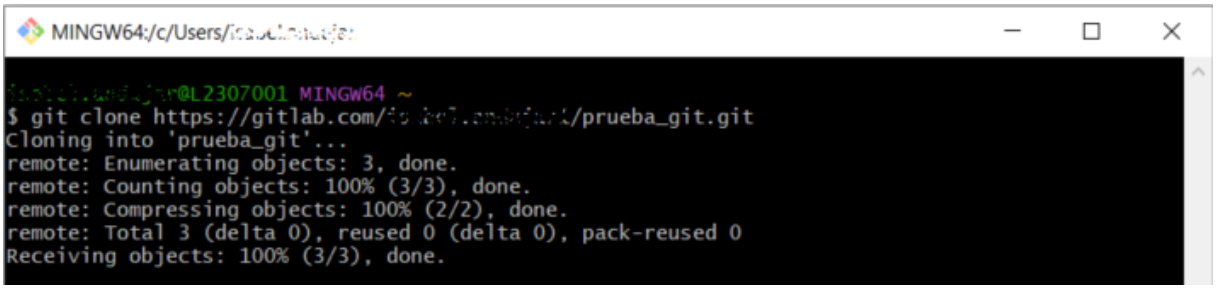
El siguiente paso será crear un proyecto dentro de nuestra sesión de GitLab y obtener su URL para clonarlo mediante https:





Disponemos de dos métodos de clonado y la principal diferencia es que HTTPS utiliza autenticación mediante nombre de usuario y contraseña y es fácil de configurar, mientras que SSH utiliza autenticación basada en claves, que ofrece mayor seguridad, pero requiere configuración adicional. Ambos son seguros para interactuar con repositorios de GitLab y la elección depende de las preferencias, requisitos de seguridad y entorno corporativo.

Una vez hecho esto, tenemos que clonar el repositorio remoto en nuestro equipo usando el comando **‘`git clone`’**.

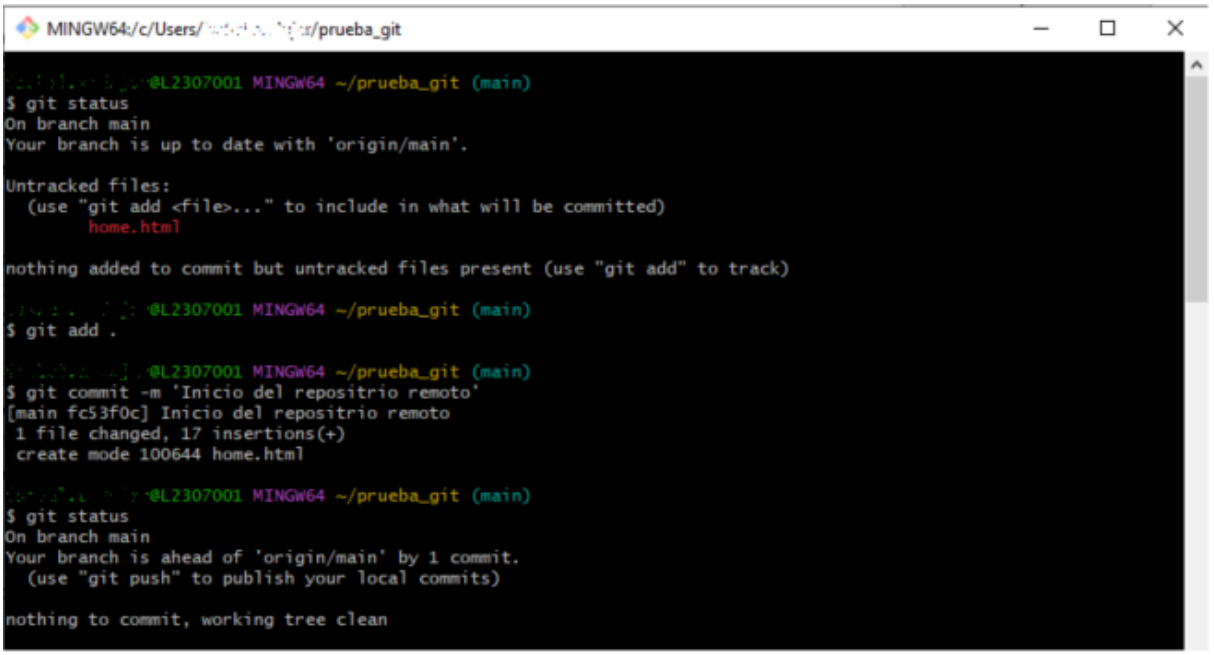


⚠ **A la hora de clonar repositorios tenemos que prestar atención al usuario que tenemos configurado, ya que si el repositorio no es público no podríamos clonarlo.**

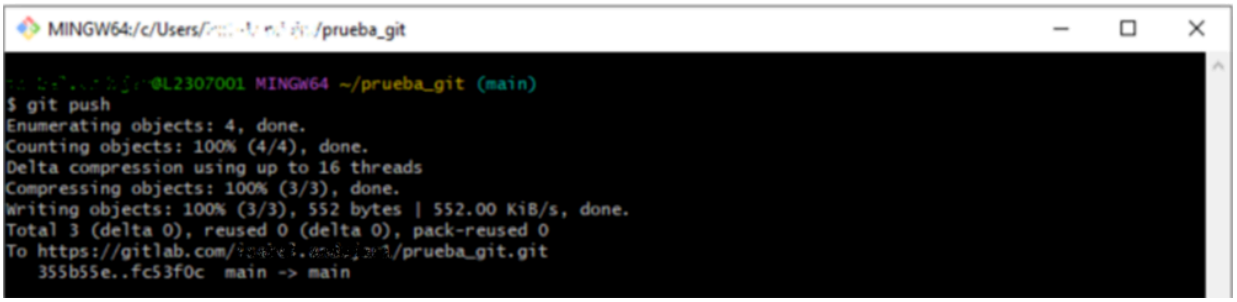
3. Trabajando con repositorio remoto.

- **Subiendo cambios al repositorio remoto**:

Para comenzar a trabajar con el repositorio remoto empezaremos añadiendo al directorio local nuestro archivo 'home.html' y hacemos nuestro primer commit:

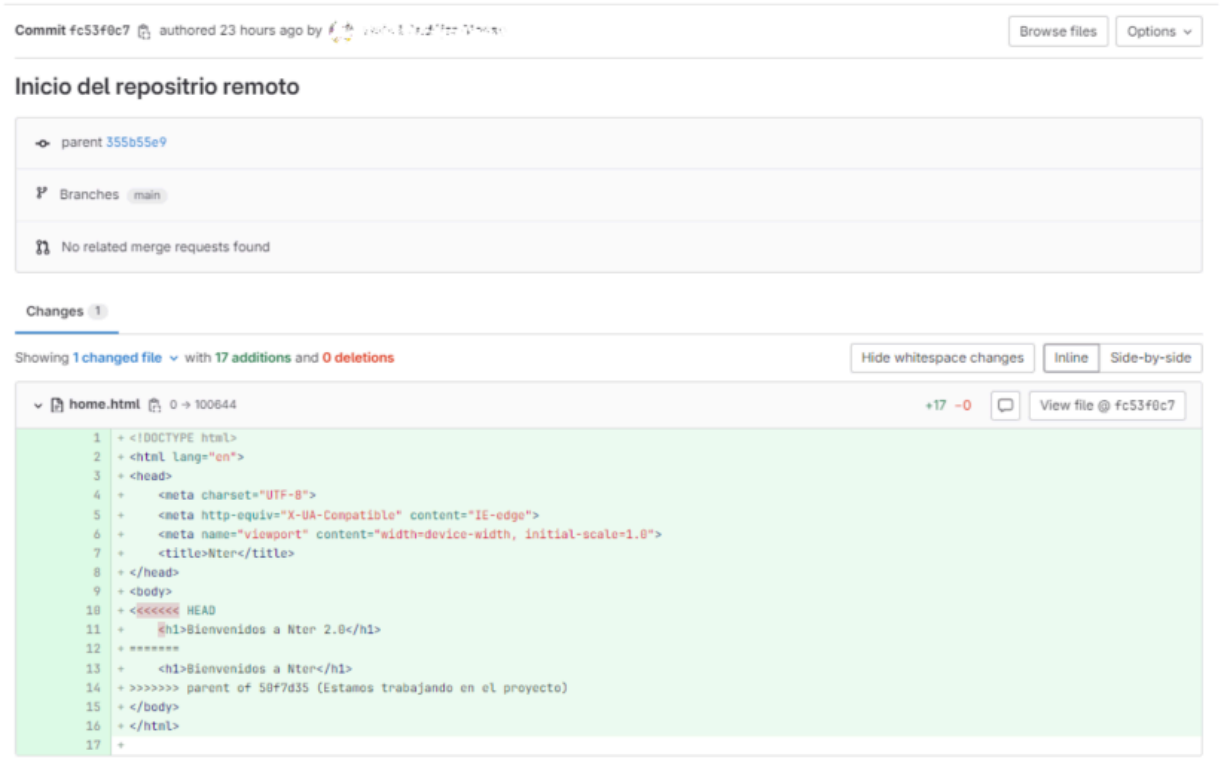


Vemos que git nos informa de que nuestro repositorio local está a un commit del repositorio remoto. Para actualizar el repositorio remoto debemos usar el comando **‘git push’**:



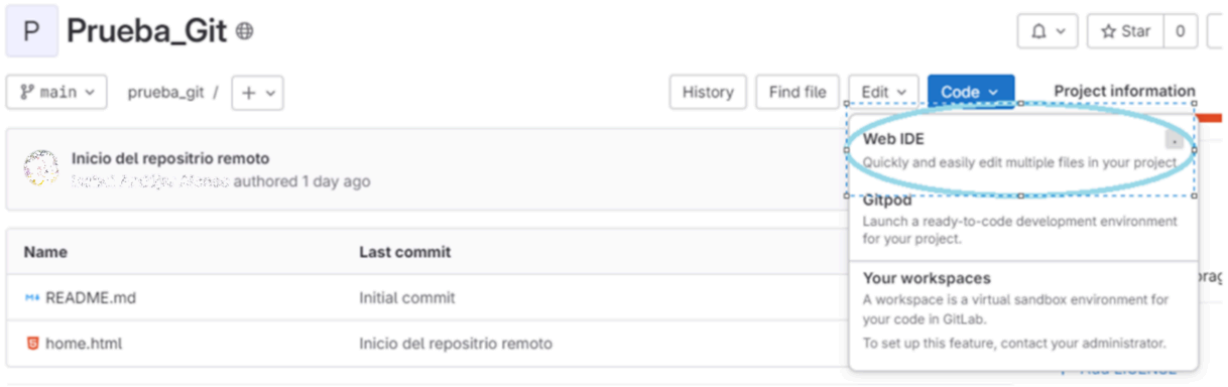
Este comando nos sirve para subir los cambios que tenemos en nuestro repositorio local hacia el repositorio remoto.

Una de las ventajas de GitLab es que tenemos la opción de ver toda la información sobre los commits realizados de una forma muy intuitiva:

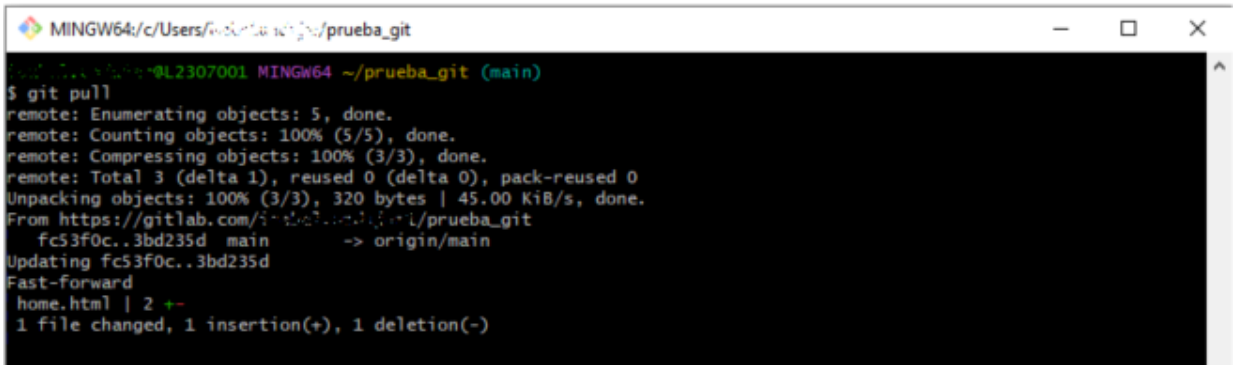


• Actualizando el repositorio local:

Por último, nos quedaría ver como actualizar nuestro repositorio local en relación con el repositorio remoto. Para comenzar necesitamos hacer cambios en el repositorio remoto y para ello utilizaremos 'Web IDE', una utilidad de GitlLab que nos permite editar los archivos de nuestro proyecto de una forma rápida y sencilla.



Para actualizar los cambios en nuestro repositorio local usamos el comando **'git pull'**:



Este comando actualiza la rama del repositorio local en la que se está trabajando y además nos da un resumen de las modificaciones que se han realizado.

¡Listo! Ya estás preparado para trabajar con repositorios remotos.