

SUBIR CAMBIOS AL REPOSITORIO DE CÓDIGO INSTITUCIONAL GITLAB

ROL	NOMBRE	CARGO
Elaborado por:	Iván Galván Vidalón	Coordinador de Desarrollo Tecnológico
Revisado por:	Kelly Elizabeth Paz Orellana	Jefa de la Unidad de Modernización
Aprobado por:	José Antonio Callirgos Paz	Jefe de la Oficina de Tecnología de la Información

CONTROL DE CAMBIOS

N°	Ítems (Sección del documento)	Descripción del cambio
-	-	Versión inicial del documento

ÍNDICE

1. OBJETIVO.....	4
2. ALCANCE	4
3. SIGLAS / ACRONIMOS	4
4. DEFINICIONES	4
5. DESARROLLO DEL INSTRUCTIVO	4
5.1 REQUISITOS PREVIOS.....	4
5.2 CLONAR EL REPOSITORIO.....	5
5.3 ACTUALIZAR Y PREPARAR LOS CAMBIOS.....	6
5.4 REVISAR LOS CAMBIOS	7
5.5 SUBIR LOS CAMBIOS AL REPOSITORIO DE GITLAB	7
5.6 VERIFICAR ESTADO EN GITLAB	7

1. OBJETIVO

Establecer las instrucciones para subir los cambios de código de las aplicaciones de la Sunass al repositorio GitLab, donde se encuentran alojados los aplicativos que administra la Oficina de Tecnologías de Información, garantizando la correcta gestión y control de versiones de los desarrollos realizados.

2. ALCANCE

El presente documento es de aplicación obligatoria para el equipo de desarrollo de la Oficina de Tecnologías de Información de la Sunass.

3. SIGLAS / ACRONIMOS

OTI : Oficina de Tecnologías de Información

TI : Tecnologías de la Información

4. DEFINICIONES

4.1 Repositorio: Es un almacenamiento centralizado que contiene todos los archivos, códigos fuente y la historia completa del desarrollo de un proyecto de software.

4.2 GitLab: Es una plataforma de DevOps que proporciona un entorno completo para la gestión del ciclo de vida del desarrollo de software. Es una aplicación basada en la web que combina varias herramientas y funcionalidades para el desarrollo, control de versiones, integración continua, entrega continua (CI/CD), gestión de proyectos, y colaboración entre equipos.

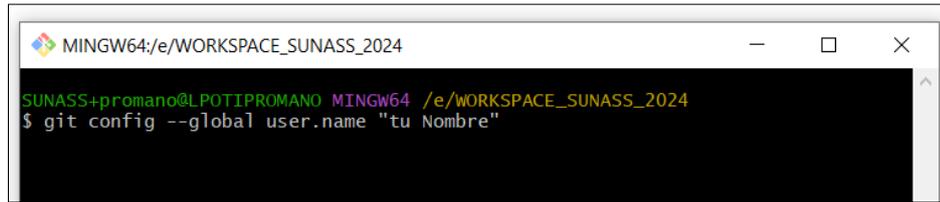
4.3 Rama: Es una versión alternativa del código fuente de un proyecto que permite trabajar de manera aislada en cambios específicos, características nuevas, corrección de errores u otros desarrollos sin afectar la rama principal

4.4 Git: Es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar simultáneamente en un proyecto sin interferencias. Creado por Linus Torvalds en 2005, GIT rastrea los cambios en los archivos de un proyecto, facilitando la colaboración y gestión del código fuente.

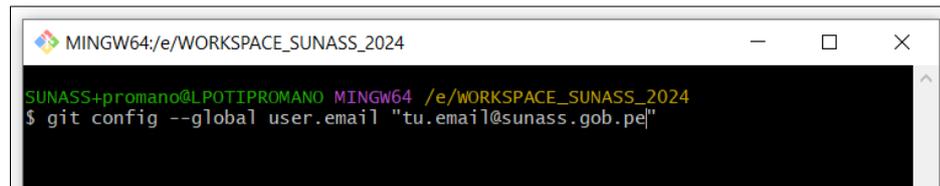
5. DESARROLLO DEL INSTRUCTIVO

5.1 REQUISITOS PREVIOS

- a) Verificar que el Git este instalado en el equipo informático. De no tenerlo, la descarga se puede realizar desde <https://git-scm.com/>.
- b) Contar con una cuenta en GitLab y acceso al repositorio donde se subirán los cambios.
- c) Configurar el nombre de usuario y dirección de correo electrónico en Git, ejecutando los siguientes comandos en la terminal, reemplazando "Tu Nombre" y "email-institucional@sunass.gob.pe." con la información real:
 - ✓ git config --global user.name "Tu Nombre"
 - ✓ git config --global user.email tu.email@sunass.gob.pe



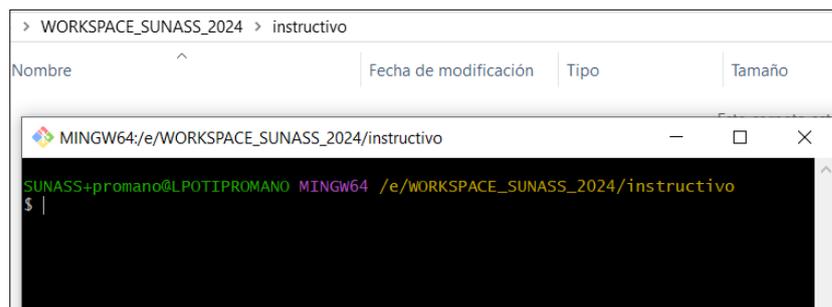
```
MINGW64:/e/WORKSPACE_SUNASS_2024
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024
$ git config --global user.name "tu Nombre"
```



```
MINGW64:/e/WORKSPACE_SUNASS_2024
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024
$ git config --global user.email "tu.email@sunass.gob.pe"
```

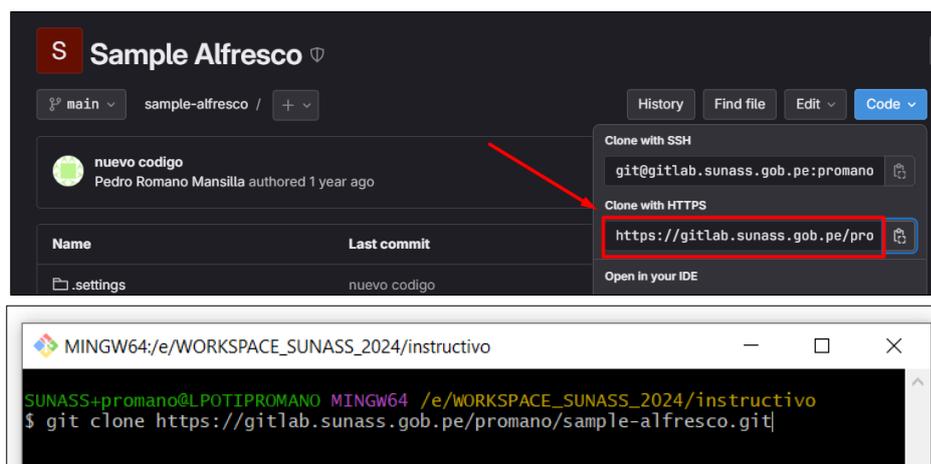
5.2 CLONAR EL REPOSITORIO

- a) Abrir el terminal y navegar al directorio en el que se va a clonar el repositorio:



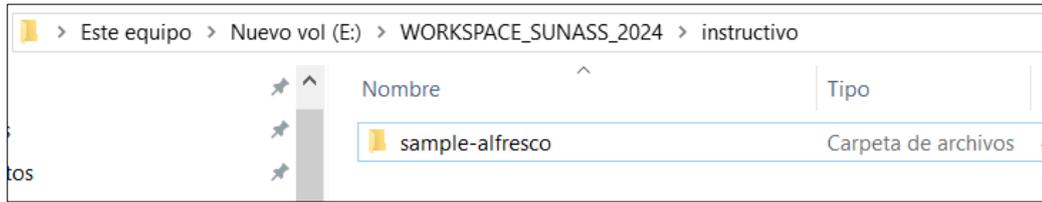
```
> WORKSPACE_SUNASS_2024 > instructivo
Nombre          Fecha de modificación  Tipo  Tamaño
MINGW64:/e/WORKSPACE_SUNASS_2024/instructivo
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo
$ |
```

- b) Ejecutar el siguiente comando "git clone". Para ello, necesitas la URL del repositorio. Por ejemplo: [git clone https://gitlab.sunass.gob.pe/tu-usuario/tu-proyecto.git](https://gitlab.sunass.gob.pe/tu-usuario/tu-proyecto.git)



The image shows a screenshot of a GitLab repository page for 'Sample Alfresco'. The repository is 'nuevo codigo' by Pedro Romano Mansilla. The 'Clone with HTTPS' option is highlighted with a red box and a red arrow. Below this, a terminal window shows the command: `git clone https://gitlab.sunass.gob.pe/promano/sample-alfresco.git`

- c) Verificar que el repositorio se haya clonado correctamente en el equipo:



5.3 ACTUALIZAR Y PREPARAR LOS CAMBIOS

- a) Navegar al directorio del repositorio clonado usando el comando **nombre de la carpeta del repositorio**, como el ejemplo de la imagen.

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo
$ cd sample-alfresco/
```

- b) Verificar que está ubicado en la rama correcta usando el comando “git Branch”. Si se requiere cambiar de rama, usar git checkout -b nombre-de-la-rama:

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (main)
$ git branch
* main

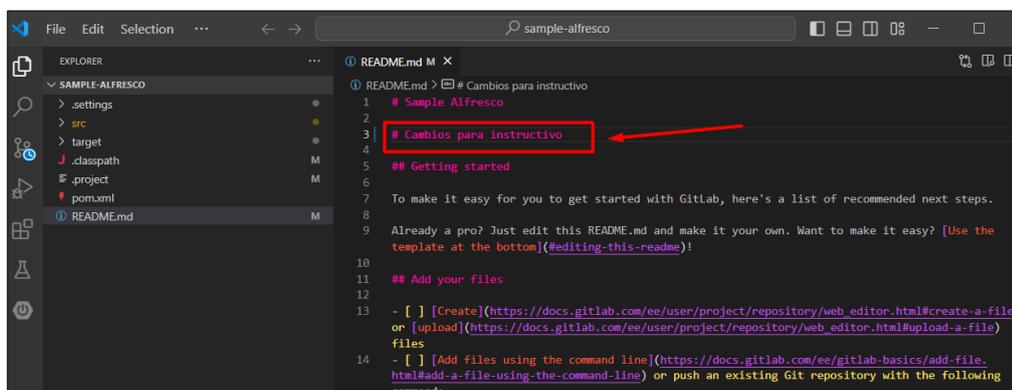
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (main)
$ git checkout -b cambios-romano
Switched to a new branch 'cambios-romano'

SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$
```

- c) Actualizar la copia local con los últimos cambios desde el servidor usando el comando “git pull”:

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$ git pull
```

- d) Realizar los cambios necesarios en el código utilizando el editor de texto o IDE preferido:



- e) Agregar los archivos modificados o nuevos al área de preparación usando el comando “git add” :

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$ git add .
```

- f) Confirmar los cambios preparados usando el comando “git commit”. Se debe escribir un mensaje de confirmación descriptivo que explique los cambios realizados: git commit -m "Descripción corta de los cambios realizados”

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$ git commit -m "nuevos cambios en el proyecto"
[cambios-romano d29d07f] nuevos cambios en el proyecto
8 files changed, 27 insertions(+), 32 deletions(-)
delete mode 100644 target/classes/META-INF/MANIFEST.MF
delete mode 100644 target/classes/META-INF/maven/com/Alfresco/pom.properties
delete mode 100644 target/classes/META-INF/maven/com/Alfresco/pom.xml
```

5.4 REVISAR LOS CAMBIOS

Utilizar el comando “git status” para ver los archivos modificados y/o agregados:

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$ git status
On branch cambios-romano
Changes to be committed:
  (use "git restore --staged <file>.." to unstage)
        modified:   README.md
```

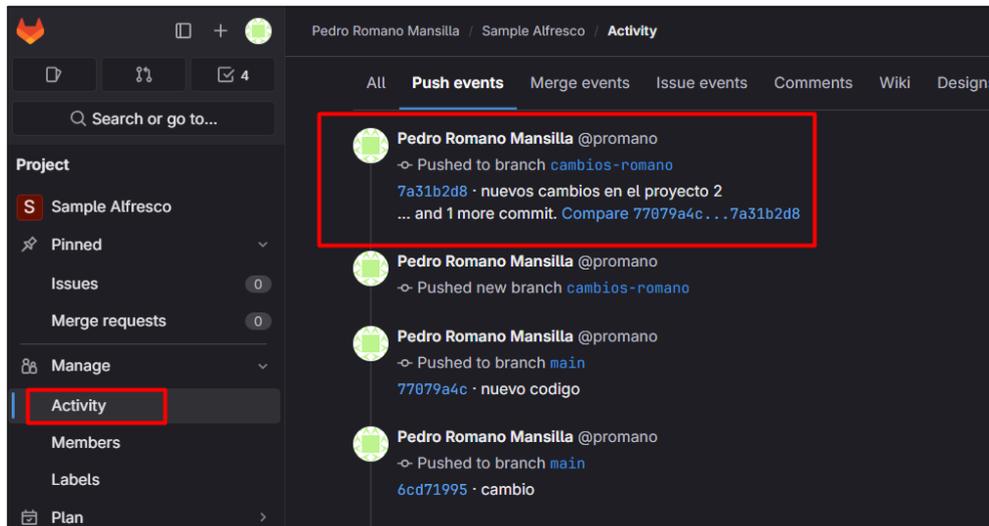
5.5 SUBIR LOS CAMBIOS AL REPOSITORIO DE GITLAB

Una vez que se haya confirmado los cambios localmente, se procede a subir al servidor de GitLab usando el comando “git push”. Esta acción envía los cambios realizados a la rama correspondiente en GitLab: git push origin nombre-de-la-rama

```
SUNASS+promano@LPOTIPROMANO MINGW64 /e/WORKSPACE_SUNASS_2024/instructivo/sample-alfresco (cambios-romano)
$ git push origin cambios-romano
warning: ----- SECURITY WARNING -----
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
warning: ----- SECURITY WARNING -----
warning: | TLS certificate verification has been disabled! |
warning: -----
warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify for more information.
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (13/13), 2.13 KiB | 1.06 MiB/s, done.
Total 13 (delta 8), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for cambios-romano, visit:
remote:   https://gitlab.sunass.gob.pe/promano/sample-alfresco/-/merge_requests/new?merge_request%5Bsource_branch%5D=cambios-romano
remote:
To https://gitlab.sunass.gob.pe/promano/sample-alfresco.git
 77079a4..7a31b2d cambios-romano -> cambios-romano
```

5.6 VERIFICAR ESTADO EN GITLAB

- a) Ingresar a la interfaz web de GitLab y verificar que los cambios se hayan subido correctamente:



- b) Revisar los commits recientes, crear solicitudes de fusión (merge requests), entre otras acciones dependiendo de los permisos asignados y del flujo de trabajo del equipo:

