

Instalación de GitLab



1 La documentación del proceso de instalación se encuentra disponible en el enlace



Enlace web. <https://get.docker.com/>

Figura 1

Ingreso Docker

```

Actividades   Navegador web Firefox - 23 de jun 22:08
get.docker.com/ + https://get.docker.com/
#!/bin/sh
set -e
# Docker CE for Linux installation script
#
# See https://docs.docker.com/engine/install/ for the installation steps.
#
# This script is meant for quick & easy install via:
# $ curl -fsSL https://get.docker.com -o get-docker.sh
# $ sh get-docker.sh
#
# For test builds (ie. release candidates):
# $ curl -fsSL https://test.docker.com -o test-docker.sh
# $ sh test-docker.sh
#
# NOTE: Make sure to verify the contents of the script
#       you downloaded matches the contents of install.sh
#       located at https://github.com/docker/docker-install
#       before executing.
#
# Git commit from https://github.com/docker/docker-install when
# the script was uploaded (Should only be modified by upload job):
SCRIPT_COMMIT_SHA="7cae5f8b0decc17d6571f9f52eb840fbc13b2737"

# The channel to install from:
# * nightly
# * test
# * stable
# * edge (deprecated)
DEFAULT_CHANNEL_VALUE="stable"
if [ -z "$CHANNEL" ]; then
    CHANNEL=$DEFAULT_CHANNEL_VALUE
fi

DEFAULT_DOWNLOAD_URL="https://download.docker.com"
if [ -z "$DOWNLOAD_URL" ]; then
    DOWNLOAD_URL=$DEFAULT_DOWNLOAD_URL
fi

```

2 Si el primer comando no se ejecuta, muy seguramente necesitará instalar curl, lo cual se puede realizar desde una terminal con el comando:

Apt install curl

Figura 2

Instalación curl

```

root@ubuntu-VirtualBox:/home/ubuntuuser# apt install curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi
  libgstreamer-plugins-bad1.0-0 libva-wayland2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libcurl4
Se instalarán los siguientes paquetes NUEVOS:
  curl
Se actualizarán los siguientes paquetes:
  libcurl4
1 actualizados, 1 nuevos se instalarán, 0 para eliminar y 237 no actualizados.
Se necesita descargar 161 kB/395 kB de archivos.
Se utilizarán 411 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

```

- 3** Si ya lo tiene instalado puede ejecutar los dos comandos establecidos por la guía de Docker.

Figura 3

Guía Docker

```
root@ubuntu-VirtualBox: /home/ubuntuser# curl -fsSL https://get.docker.com -o get-docker.sh
root@ubuntu-VirtualBox: /home/ubuntuser# sh get-docker.sh
```

- 4** Una vez tenga instalado Docker puede descargar y ejecutar el servicio de GitLab por medio del siguiente comando:

```
sudo docker run gitlab/gitlab-ce
```

Figura 4

Descarga y ejecución del servicio de GitLab

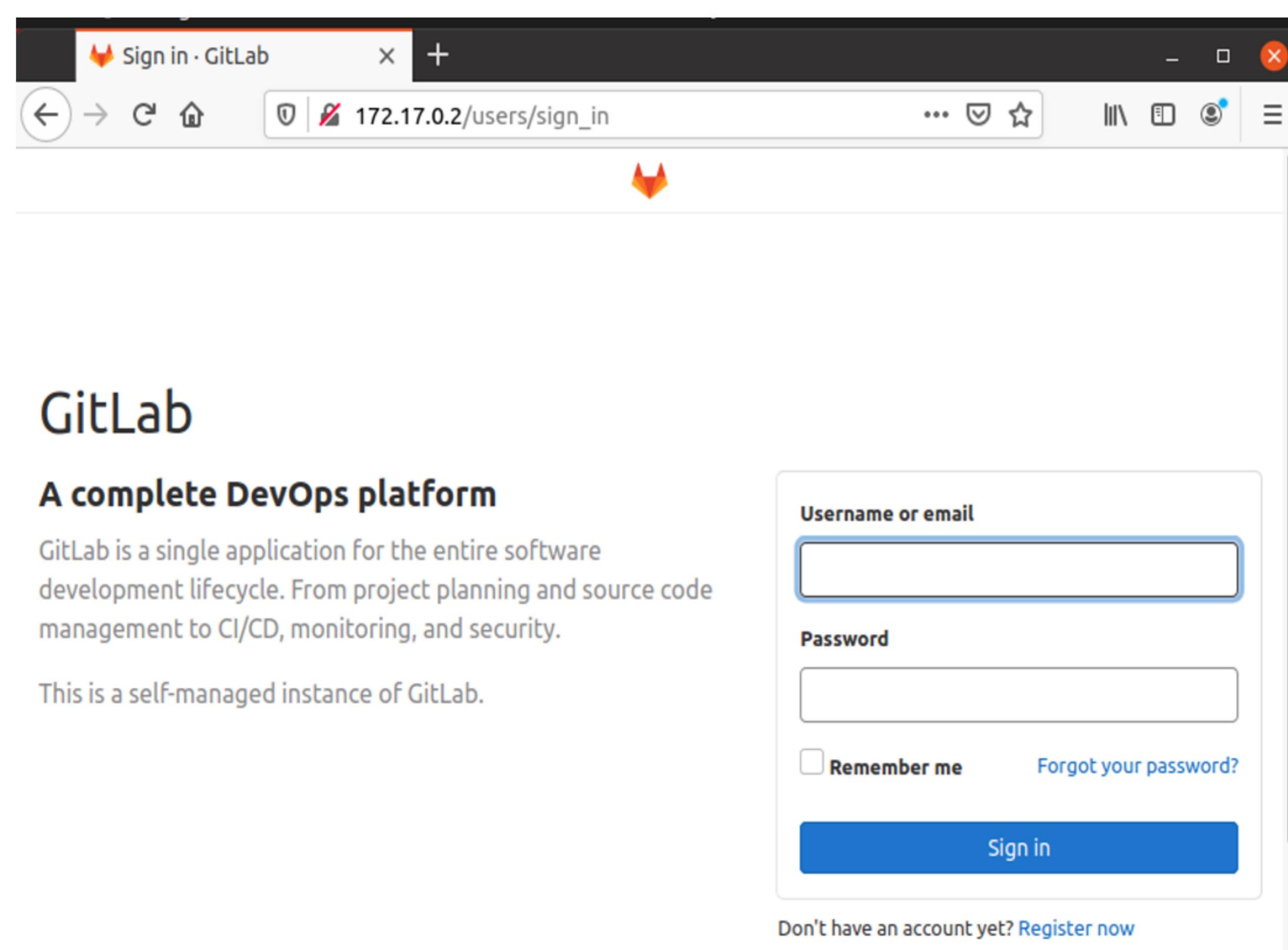
```
ubuntuser@ubuntu-VirtualBox:~$ sudo docker run gitlab/gitlab-ce
Unable to find image 'gitlab/gitlab-ce:latest' locally
latest: Pulling from gitlab/gitlab-ce
c549ccf8d472: Downloading 15.02MB/28.55MB
6d5bae66cc40: Download complete
417711cd5db5: Download complete
83276431516a: Download complete
38cdeadd2167: Download complete
39fb7b82764c: Download complete
e492df77c9e9: Download complete
2dfaafc7a986: Downloading 1.613MB/935.3MB
```

- 5** Si es la primera vez que se ejecuta el comando, la imagen de GitLab será descargada (solo la primera vez) para luego ejecutarse.

Una vez el proceso inicia su ejecución, si es que no se ha desplegado una página en el navegador, podrá acceder por la dirección en la que se despliega el servicio, en el caso de la máquina virtual de este ejemplo es en la dirección 172.17.0.2.

Figura 5

Autenticación GitLab



- 6** En algunos casos el sistema desplegará la interfaz para asignar el password a la cuenta de superadministrador root. Si es así, asigne el password requerido y omita el siguiente paso. Para el caso en el que el sistema por alguna razón no le permita registrar el password del usuario root deberá asignarlo manualmente desde el contenedor donde se está ejecutando GitLab, para esto debe consultar el identificador del contenedor en ejecución por medio del comando:

```
docker ps -all
```

- 7** Una vez obtenido el identificador deberá ingresar al contenedor usando bash por medio del comando:

```
docker exec -it IDENTIFICADOR bash
```

- 8** Estando en el contenedor deberá ingresar a la consola de producción de GitLab por medio del comando:

```
gitlab-rails console -e production
```

- 9** Por medio de la consola ya podrá ejecutar los comandos para obtener una referencia del objeto que representa el usuario root y realizar el proceso de cambio de contraseña.

```
user = User.where(id:1).first
user.password = 'Nuevo password'
user.password_confirmation = 'Nuevo password'
user.save!
exit
```

Figura 6

Cambio de contraseña

The terminal window shows the following session:

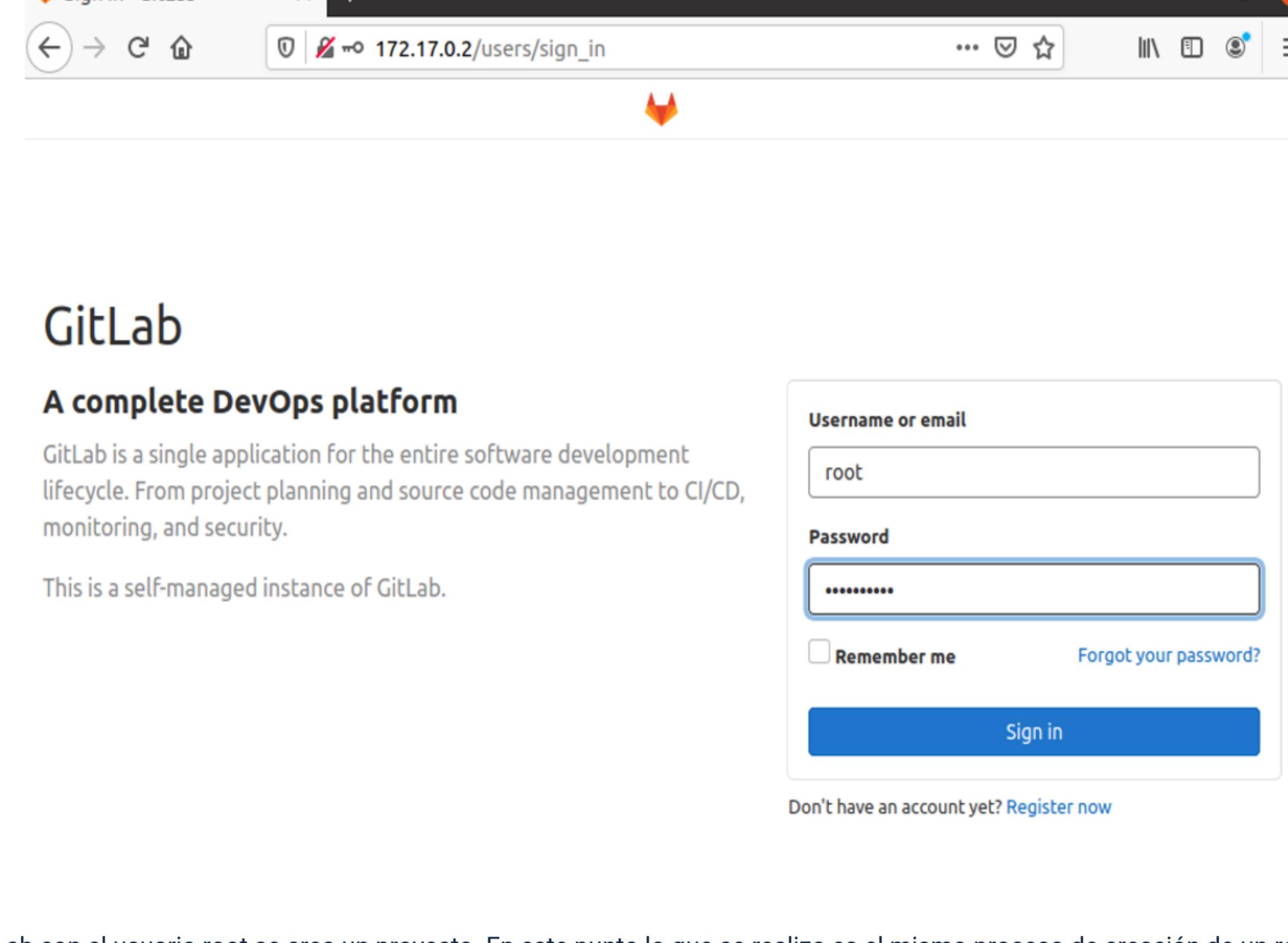
```
root@ffbc4c412962: /home/ubuntu# docker ps -all
CONTAINER ID IMAGE COMMAND CREATED STATUS
ffbc4c412962 gitlab/gitlab-ce "/assets/wrapper" 4 minutes ago Up 4 minutes (health: starting) 22/tcp, 80/tcp, 443/tcp xenodochial_khavvam
root@ubuntu-VirtualBox:/home/ubuntu# docker exec -it ffbc4c412962 bash
root@ffbc4c412962:/# gitlab-rails console -e production
-----
Ruby: ruby 2.7.2p137 (2020-10-01 revision 5445e04352) [x86_64-linux]
GitLab: 14.0.0 (f1926d2aa64) FOSS
GitLab Shell: 13.19.0
PostgreSQL: 12.6
-----
Loading production environment (Rails 6.1.3.2)
irb(main):001:0> user = User.where(id:1).first
=> #<User id:1 @root>
irb(main):002:0> user.password = "XXXXXXXXXX"
=> "XXXXXXXXXX"
irb(main):003:0> user.password_confirmation = "XXXXXXXXXX"
=> "XXXXXXXXXX"
irb(main):004:0> user.save
Enqueued ActionMailer::MailDeliveryJob (Job ID: 829026da-51d2-49a7-b5b2-a28d640ae403) to Sidekiq(mailers) with arguments: "DeviseMailer", "password_change", "deliver_now", {:args=>[#<GlobalID:0x00007f33749a75c8 @uri=<URI::GID gid://gitlab/User/1>>]}
=> true
irb(main):005:0> exit
```

The password change command is highlighted with red boxes around the assignment statements for `user.password` and `user.password_confirmation`.

- 10** Una vez se tengan las credenciales vuelva a la ventana de inicio de sesión e ingrese con el usuario **root**.

Figura 7

Credenciales del sistema



- 11** Una vez dentro de GitLab con el usuario root se crea un proyecto. En este punto lo que se realiza es el mismo proceso de creación de un repositorio vacío, que luego será clonado por los integrantes del equipo para establecer sus ambientes de trabajo y el repositorio central, como se pudo practicar en el componente formativo de integración continua.

Figura 8

Creación de nuevo proyecto – parte 1

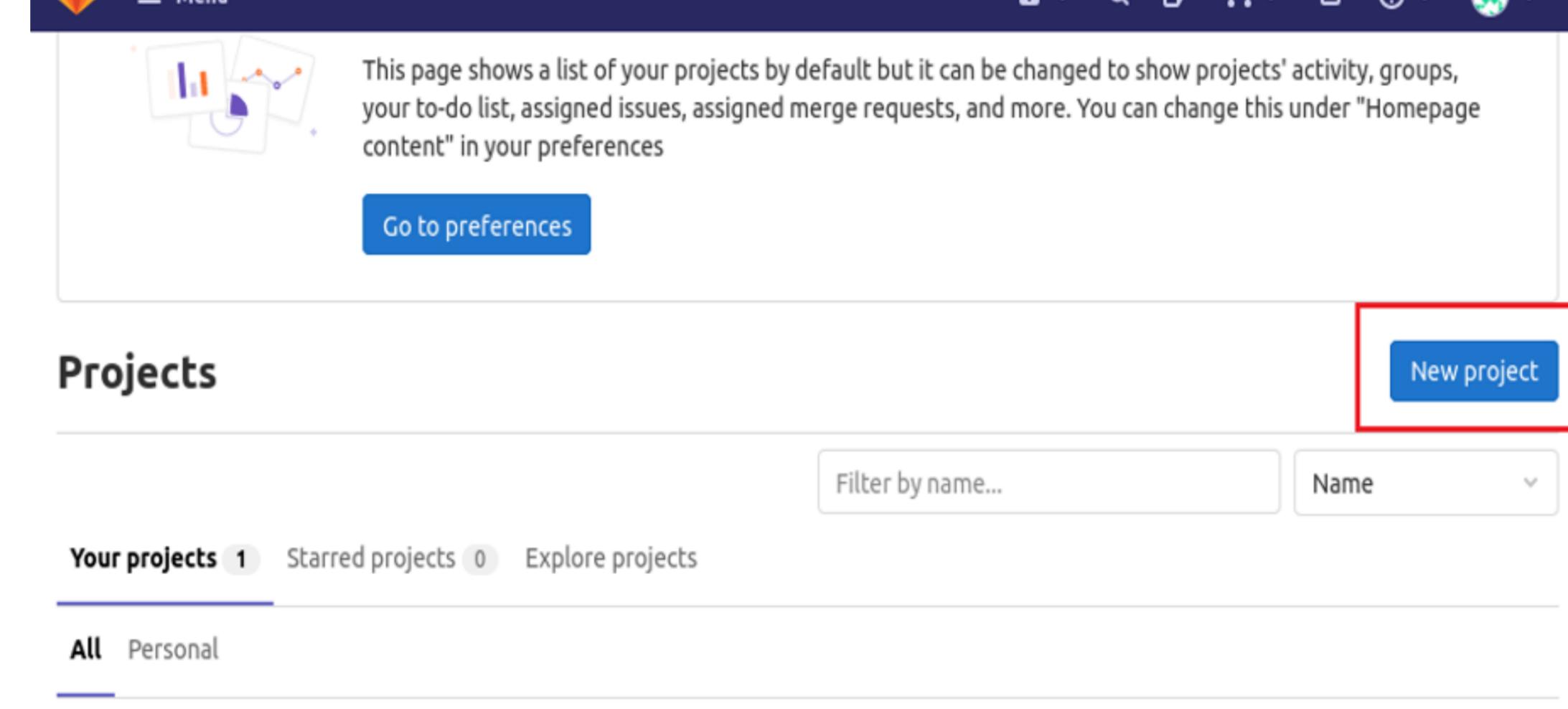


Figura 9

Creación de nuevo proyecto – parte 2

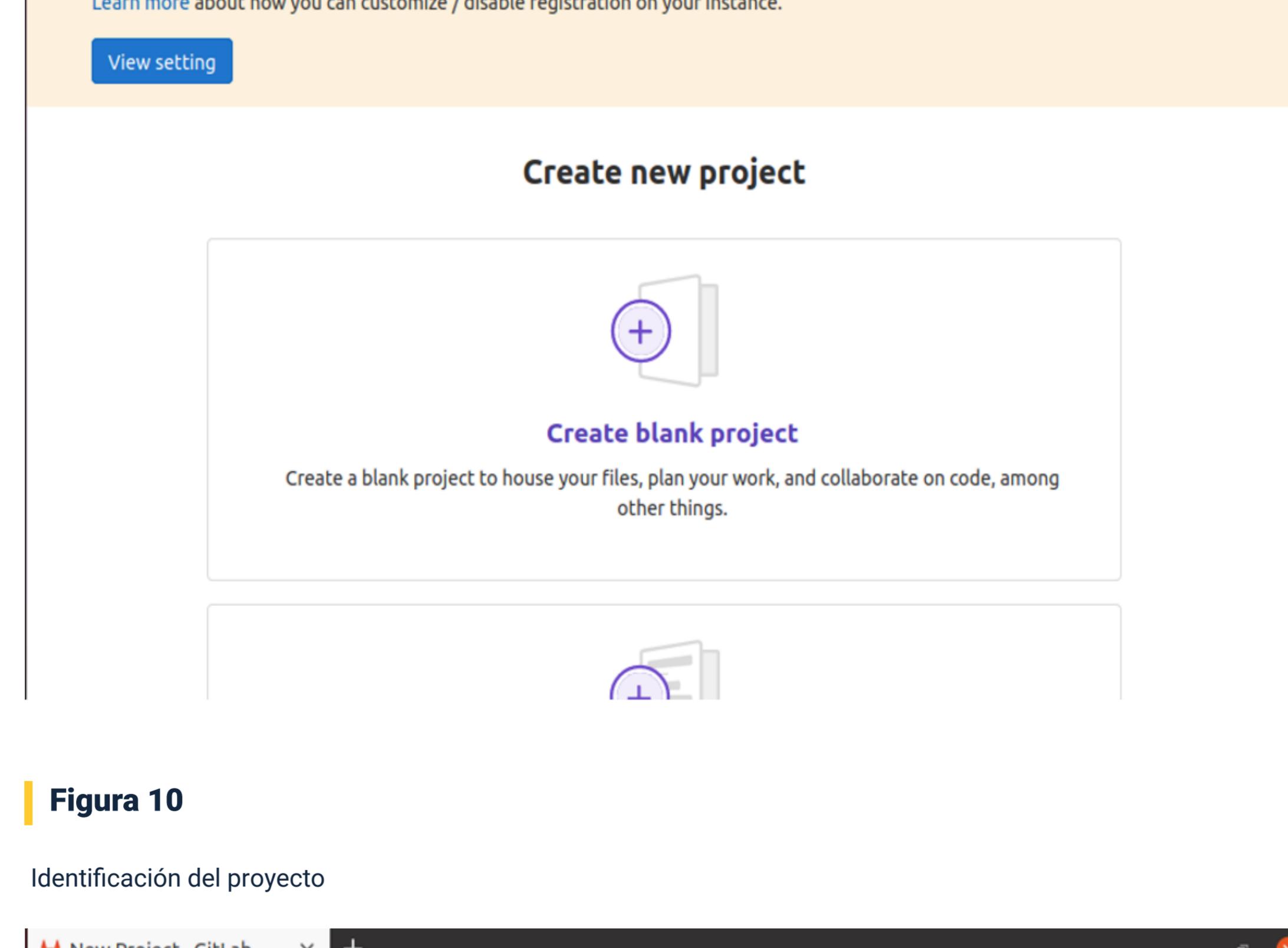
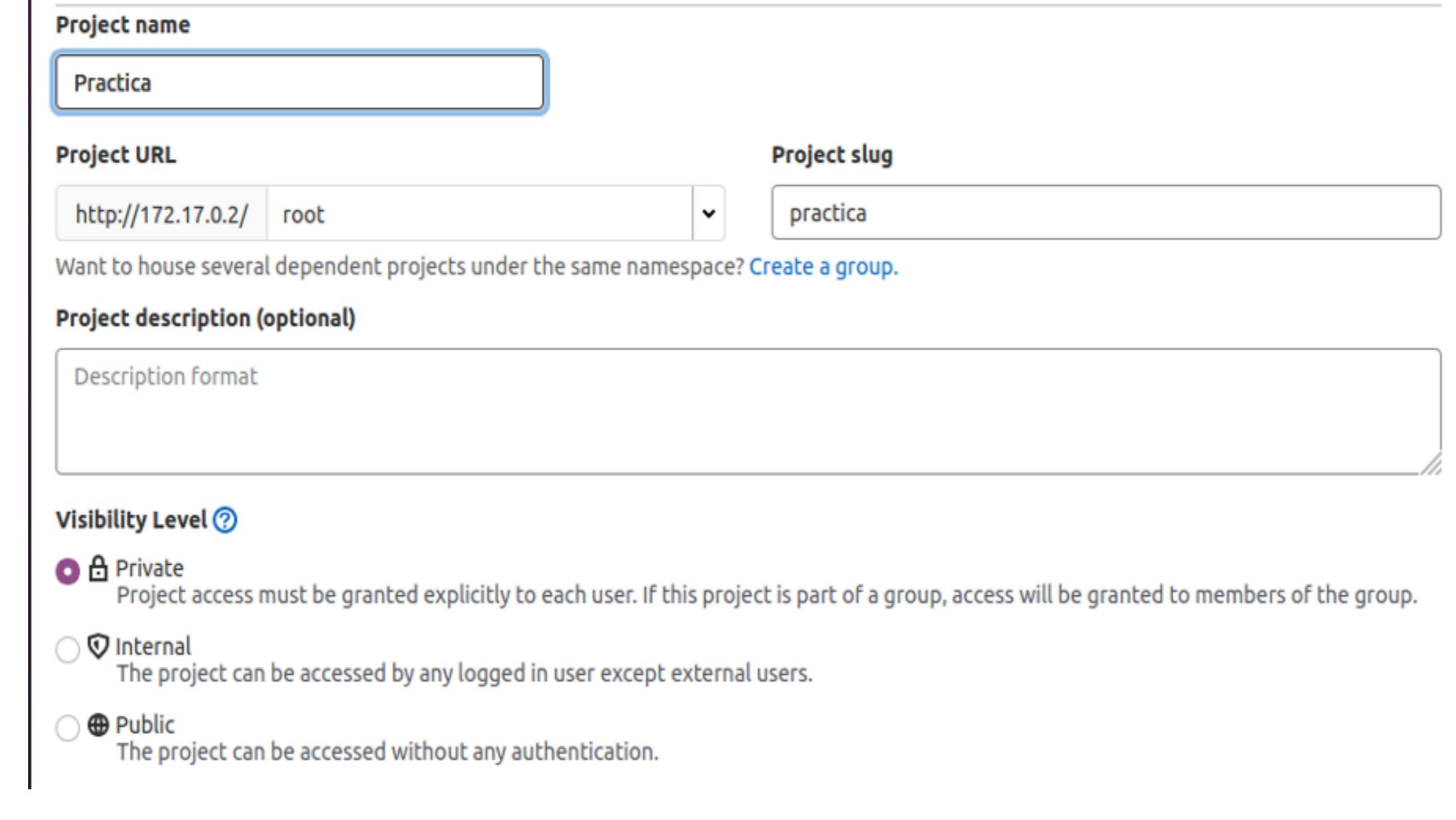


Figura 10

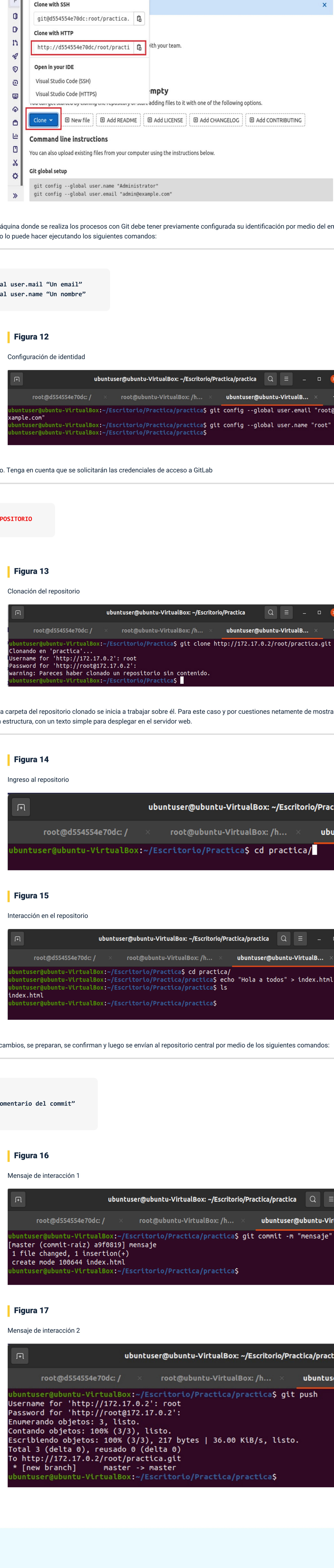
Identificación del proyecto



12 Una vez se crea el proyecto se obtiene la URL donde se ubica el repositorio para su clonación por parte del equipo de desarrollo.

Figura 11

URL del proyecto

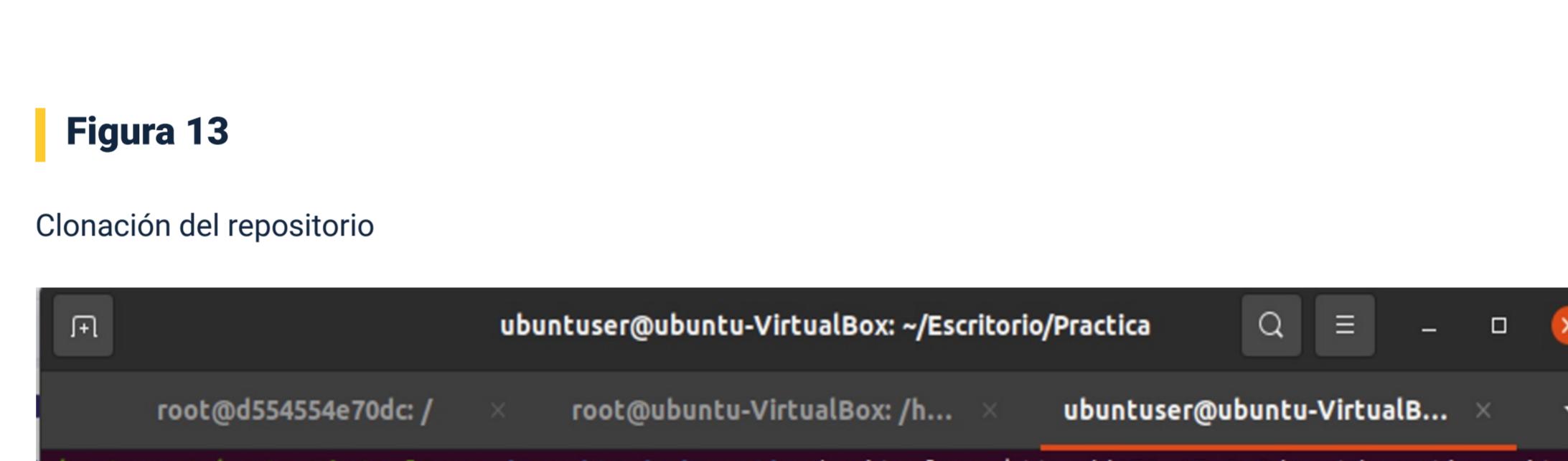


13 Recuerde que en la máquina donde se realiza los procesos con Git debe tener previamente configurada su identificación por medio del email y del usuario, si aún no ha realizado este proceso lo puede hacer ejecutando los siguientes comandos:

```
git config --global user.mail "Un email"
git config --global user.name "Un nombre"
```

Figura 12

Configuración de identidad

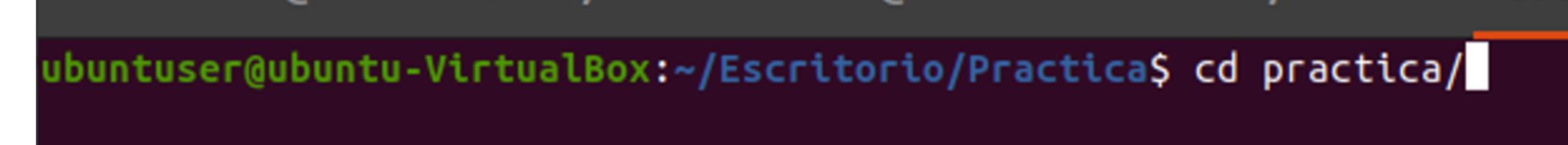


14 Se ejecuta el comando. Tenga en cuenta que se solicitarán las credenciales de acceso a GitLab

```
git clone URL_REPO
```

Figura 13

Clonación del repositorio



15 Una vez se ingresa a la carpeta del repositorio clonado se inicia a trabajar sobre él. Para este caso y por cuestiones netamente de mostrar un ejemplo sencillo se crea un único archivo html sin estructura, con un texto simple para desplegar en el servidor web.

Figura 14

Ingreso al repositorio

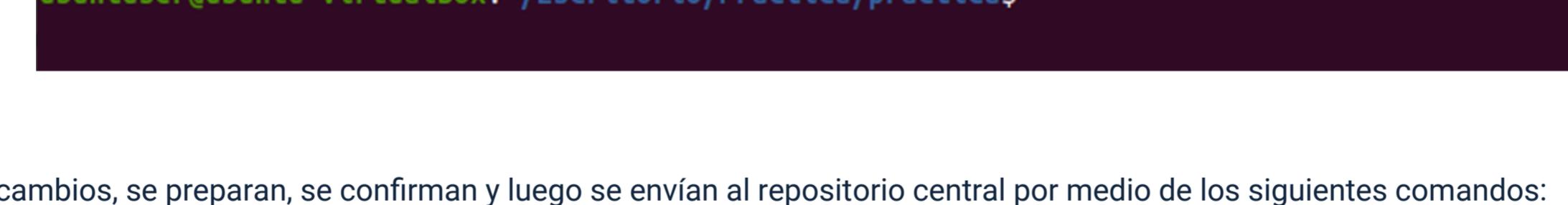
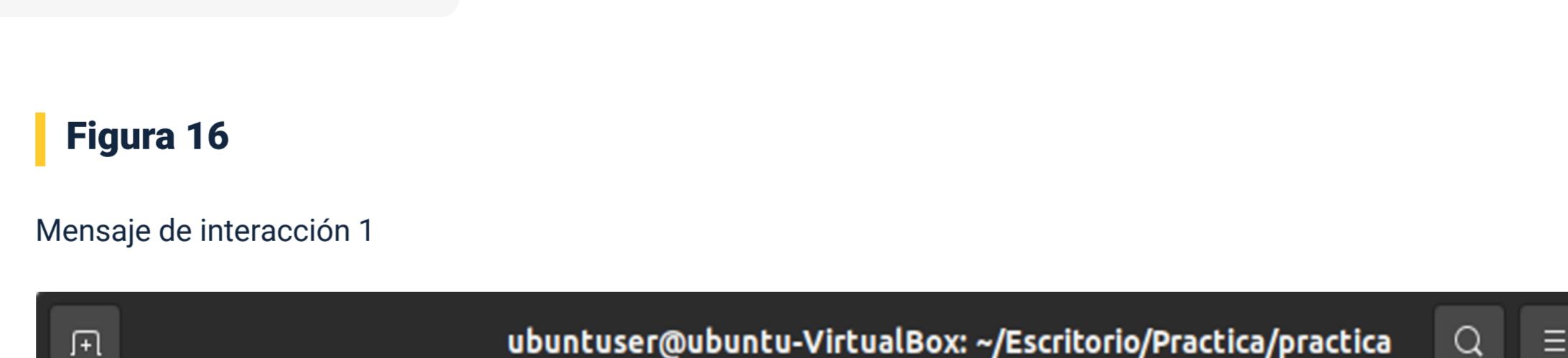


Figura 15

Interacción en el repositorio



16 Luego de realizar los cambios, se preparan, se confirman y luego se envían al repositorio central por medio de los siguientes comandos:

```
git add *
git commit -m "Comentario del commit"
git push
```

Figura 16

Mensaje de interacción 1

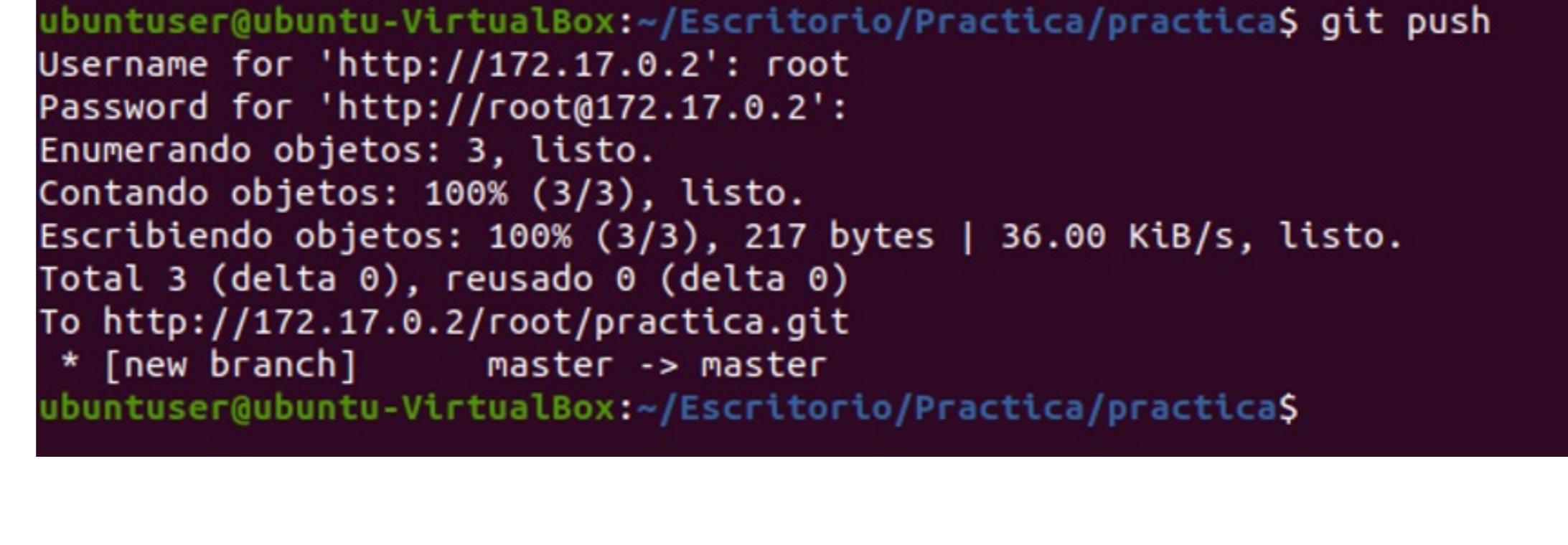
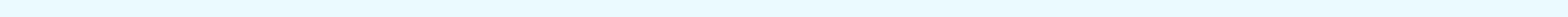


Figura 17

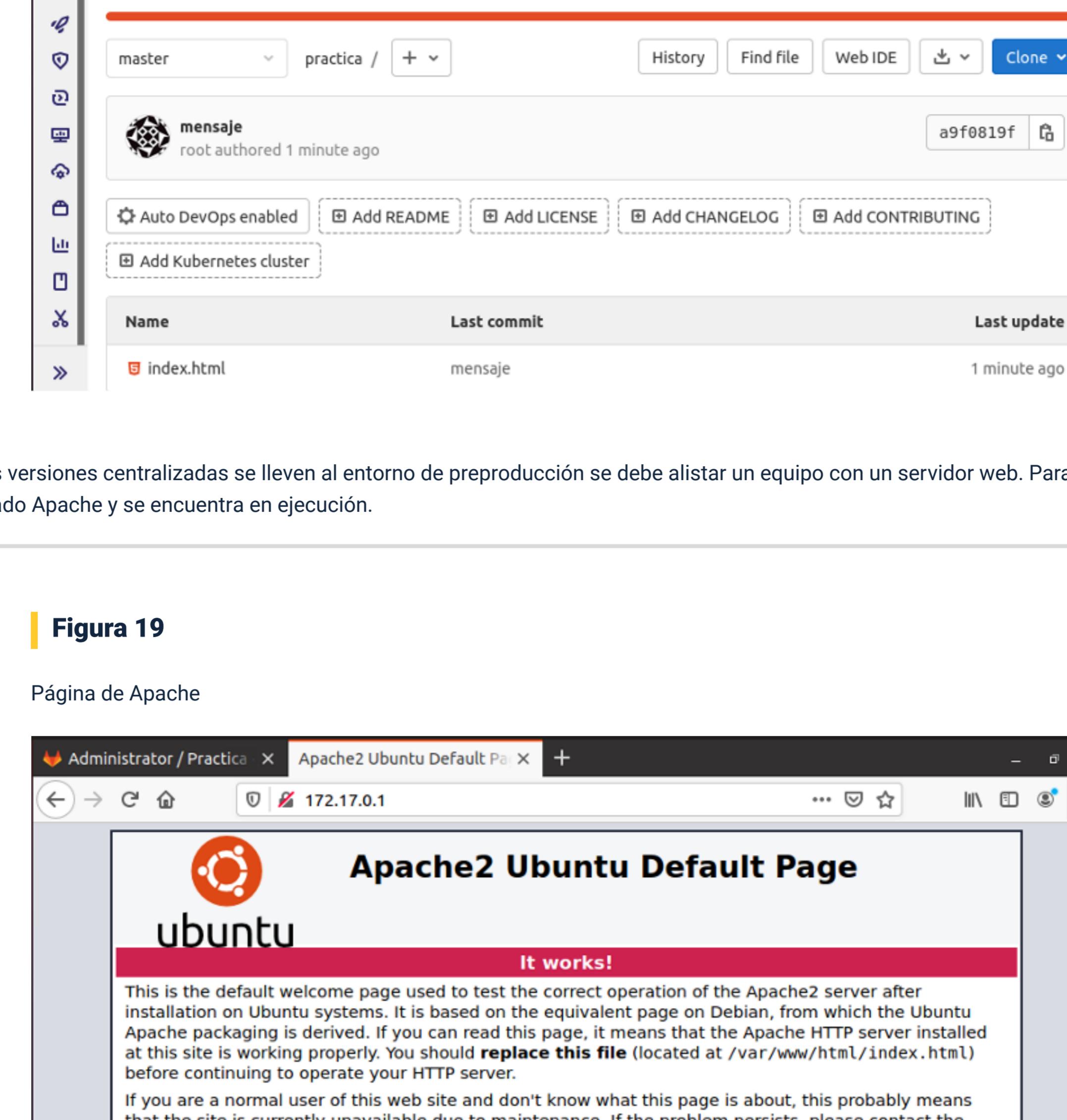
Mensaje de interacción 2



- 17** Con esto ya el equipo de trabajo podría enviar su información al servidor y trabajar de manera colaborativa, implementando la práctica de integración continua.

Figura 18

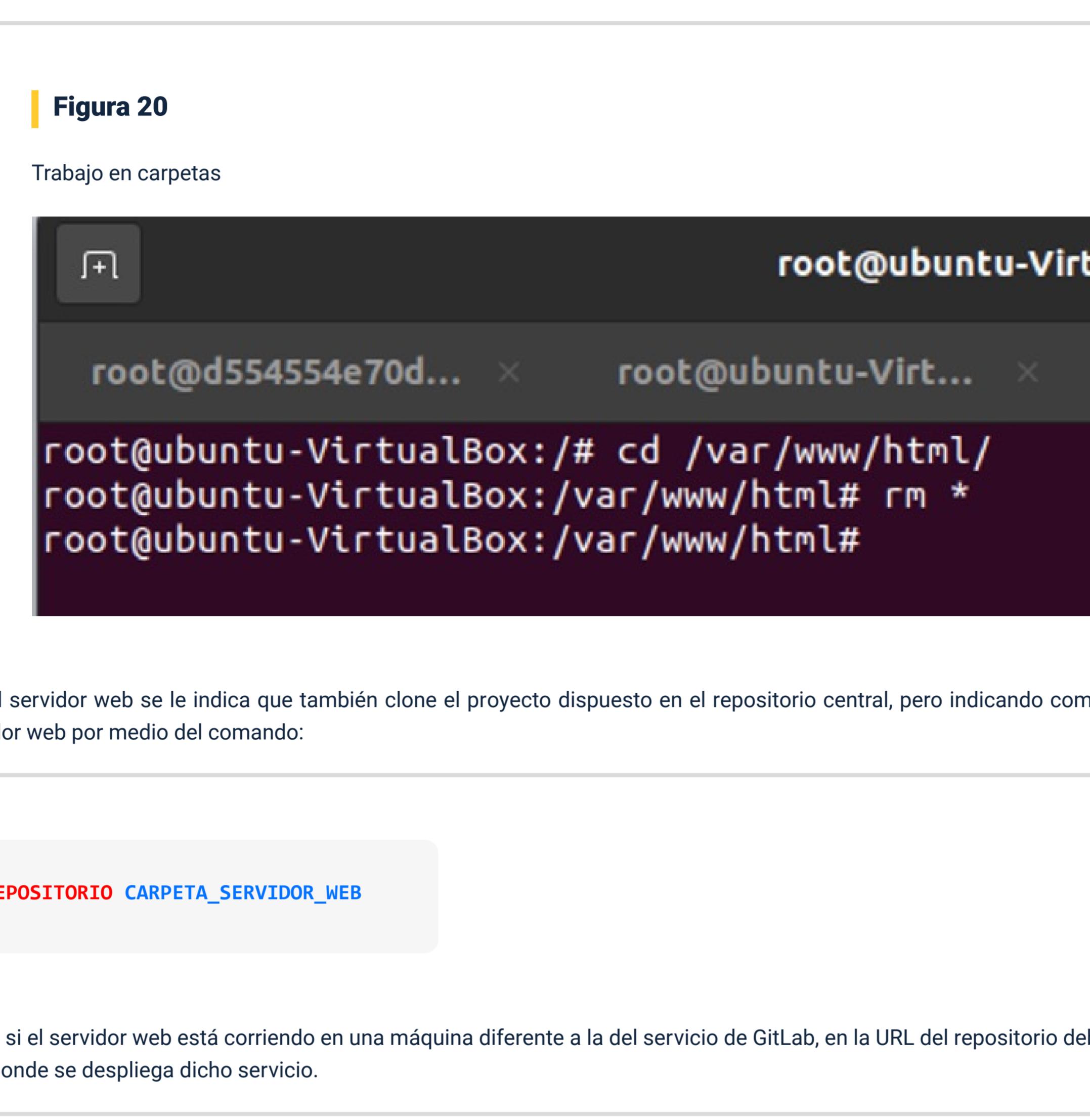
Integración continua



- 18** Para que ahora estas versiones centralizadas se lleven al entorno de preproducción se debe alistar un equipo con un servidor web. Para este ejemplo sobre la máquina virtual ya está instalado Apache y se encuentra en ejecución.

Figura 19

Página de Apache

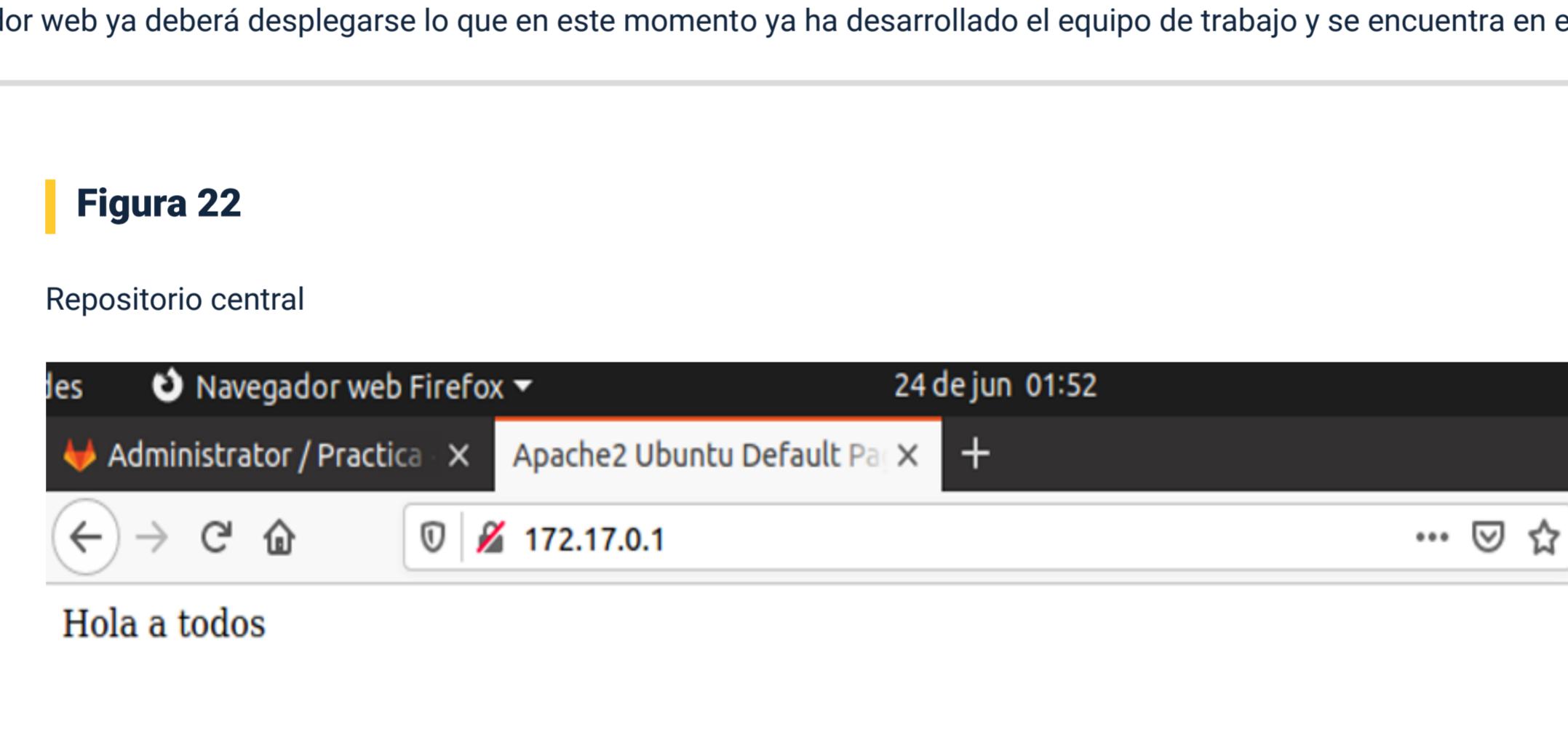


- 19** Se requiere entonces, en primer lugar, eliminar todos los archivos de la página web que se muestra por defecto con el servicio de Apache, ya que se requiere que ahora el servicio despliegue el trabajo que se ha venido centralizando en GitLab. La ubicación de estos archivos está dentro de la carpeta /var/www/html/

Cada servidor web tiene su propio directorio de despliegue y si utiliza uno diferente al de este ejemplo deberá hacer el mismo proceso en la carpeta correspondiente.

Figura 20

Trabajo en carpetas



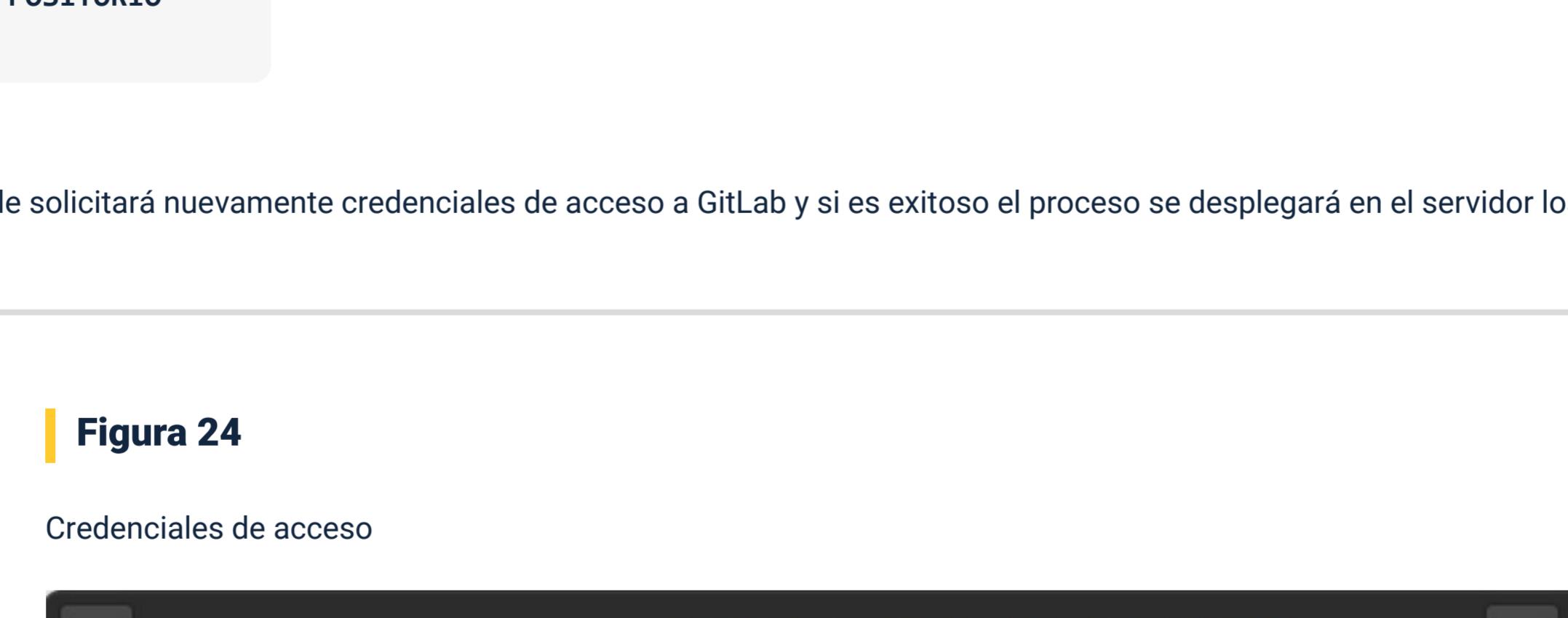
- 20** Luego, estando en el servidor web se le indica que también clone el proyecto dispuesto en el repositorio central, pero indicando como destino la carpeta de despliegue del servidor web por medio del comando:

```
git clone URL_REPOITORIO CARPETA_SERVIDOR_WEB
```

- 21** Tenga en cuenta que si el servidor web está corriendo en una máquina diferente a la del servicio de GitLab, en la URL del repositorio deberá cambiar la ubicación local por la IP del equipo donde se despliega dicho servicio.

Figura 21

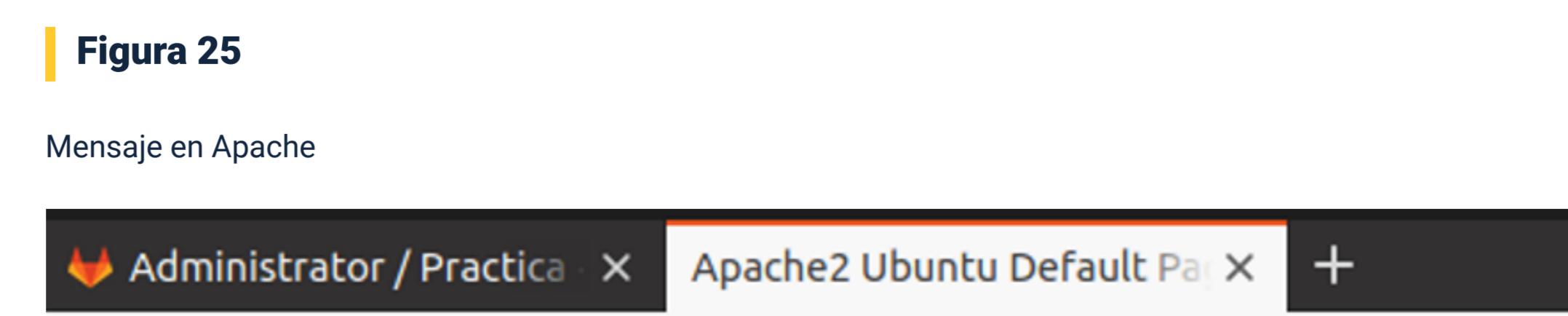
Verificación de URL



- 22** Con esto, en el servidor web ya deberá desplegarse lo que en este momento ya ha desarrollado el equipo de trabajo y se encuentra en el repositorio central.

Figura 22

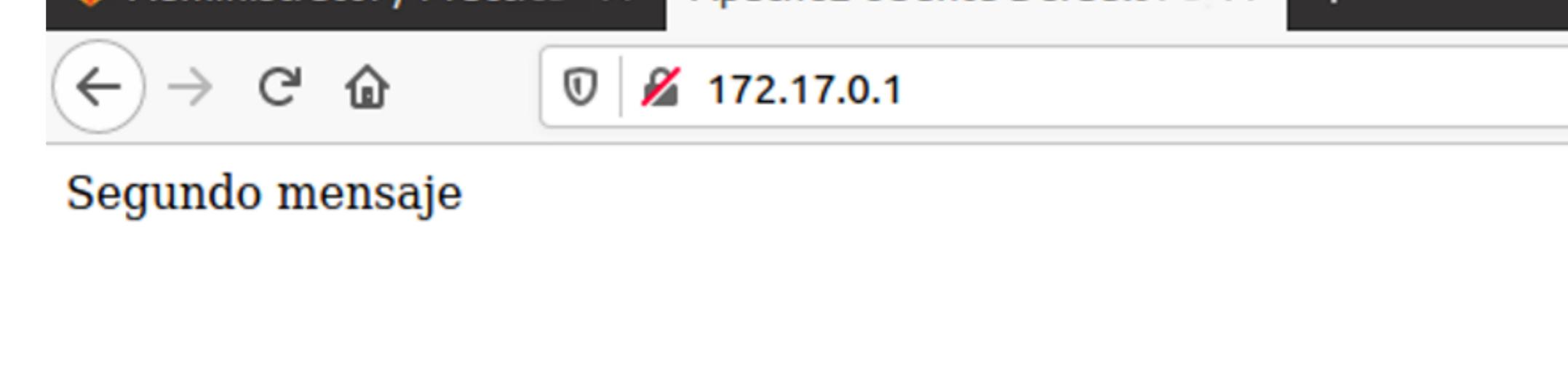
Repositorio central



- 23** Para probar que el proceso está funcionando deberá realizar cambios en el ambiente de trabajo de los desarrolladores, preparar los cambios, confirmarlos y enviarlos al repositorio central.

Figura 23

Cambios en el ambiente de trabajo



- 24** Luego, en el equipo servidor se debe hacer el proceso de actualización de la información dispuesta en el repositorio por medio del comando:

```
git pull URL_REPOITORIO
```

- 25** El comando anterior le solicitará nuevamente credenciales de acceso a GitLab y si es exitoso el proceso se desplegarán en el servidor los cambios presentes en el repositorio.

Figura 24

Credenciales de acceso



Figura 25

Mensaje en Apache



- 26** Para evitar estar ingresando las credenciales desde el servidor web cada vez que actualiza la información a desplegar, desde el repositorio se puede usar un comando pull que integre las credenciales de la siguiente forma:

```
git pull usuario_gitlab:password@URL_REPOITORIO
```

Este proceso se podría repetir cuantas veces sea necesario manualmente, con lo que conseguirá llevar desde un entorno de desarrollo a un entorno de preproducción, lo cual es una implementación básica de la práctica de entrega continua, sin embargo, este proceso se puede automatizar implementando pipelines de CI/CD que ofrece GitLab.