**Integración Continua**

**Proyecto de software basado en herramientas de integración continua**

**Andrés Felipe Rojas Parra - Código:100234216  
   
Daniel Ricardo Gómez Riveros - Código 1520010641   
   
Nelson Lainelec Santisteban Toloza - Código: 1721026128**

**Yohanny Murillo Urrutia - Código: 1721025369  
   
Cesar - Código: 1411024392**

**Tutor:**

Natalia Martinez Rojas

Politécnico Gran Colombiano

Ingeniería Diseño e Innovación

Integración continua

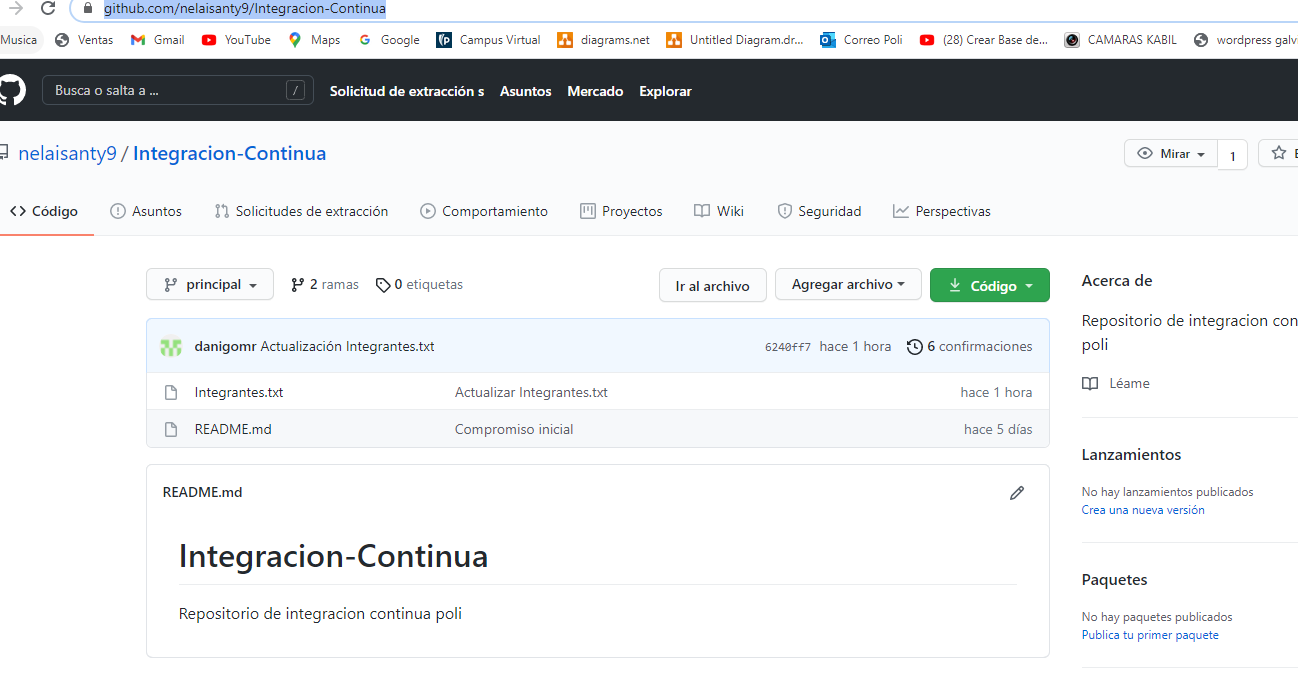
2021

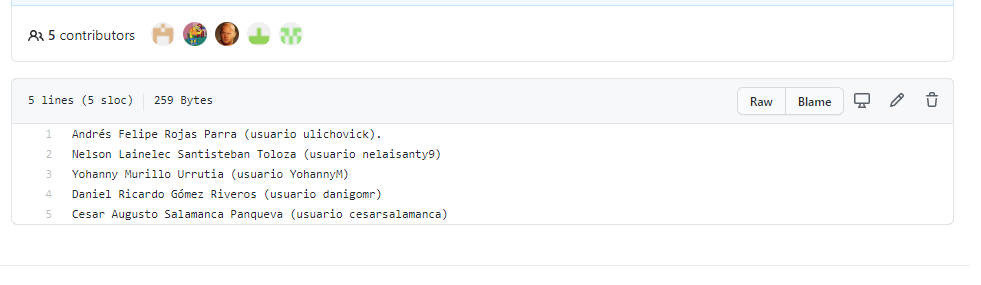
**Entrega I Semana 3**

1. Creación de Repositorio en Github

De acuerdo a lo planteado por la docente, adjuntamos el link correspondiente a nuestro repositorio y su respecta captura:

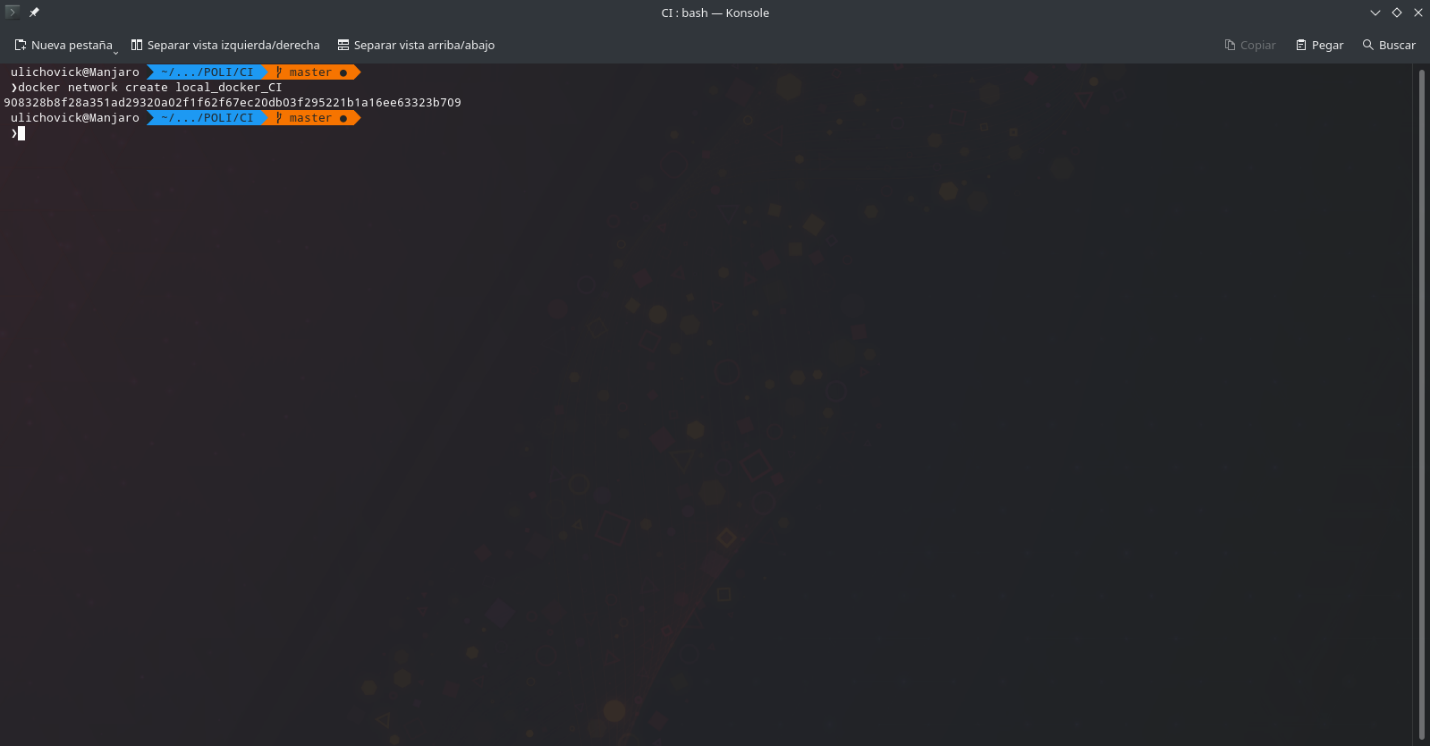
<https://github.com/nelaisanty9/Integracion-Continua>

* Creación de repositorio en Github:
* Actualización del archivo .txt con los nombres de los integrantes del grupo:



1. Creación de los dos Contenedores

# Paso 1. crear red local en docker

docker network create local\_docker\_CI

*Paso 1. Crear Red*

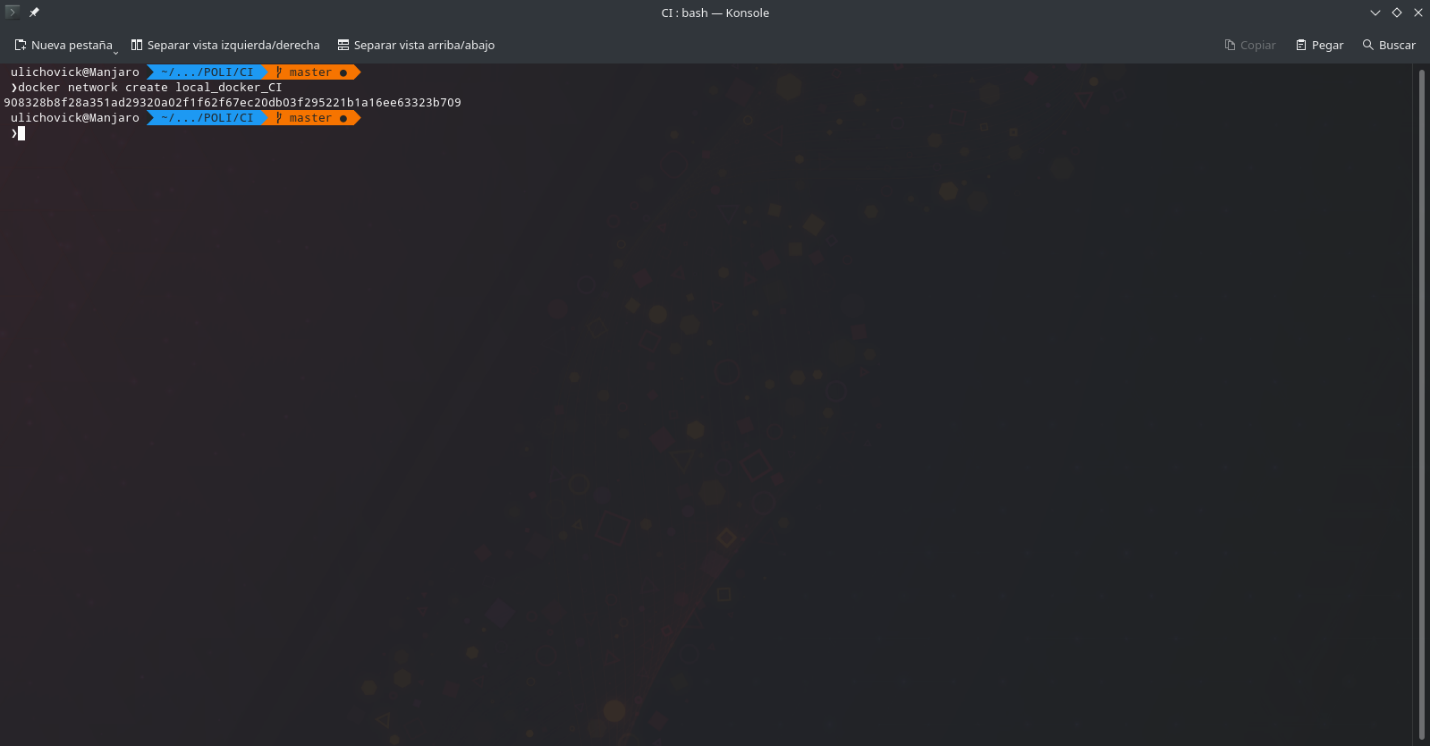
#Paso 2. Construir imágenes:

crear imágen de python en docker

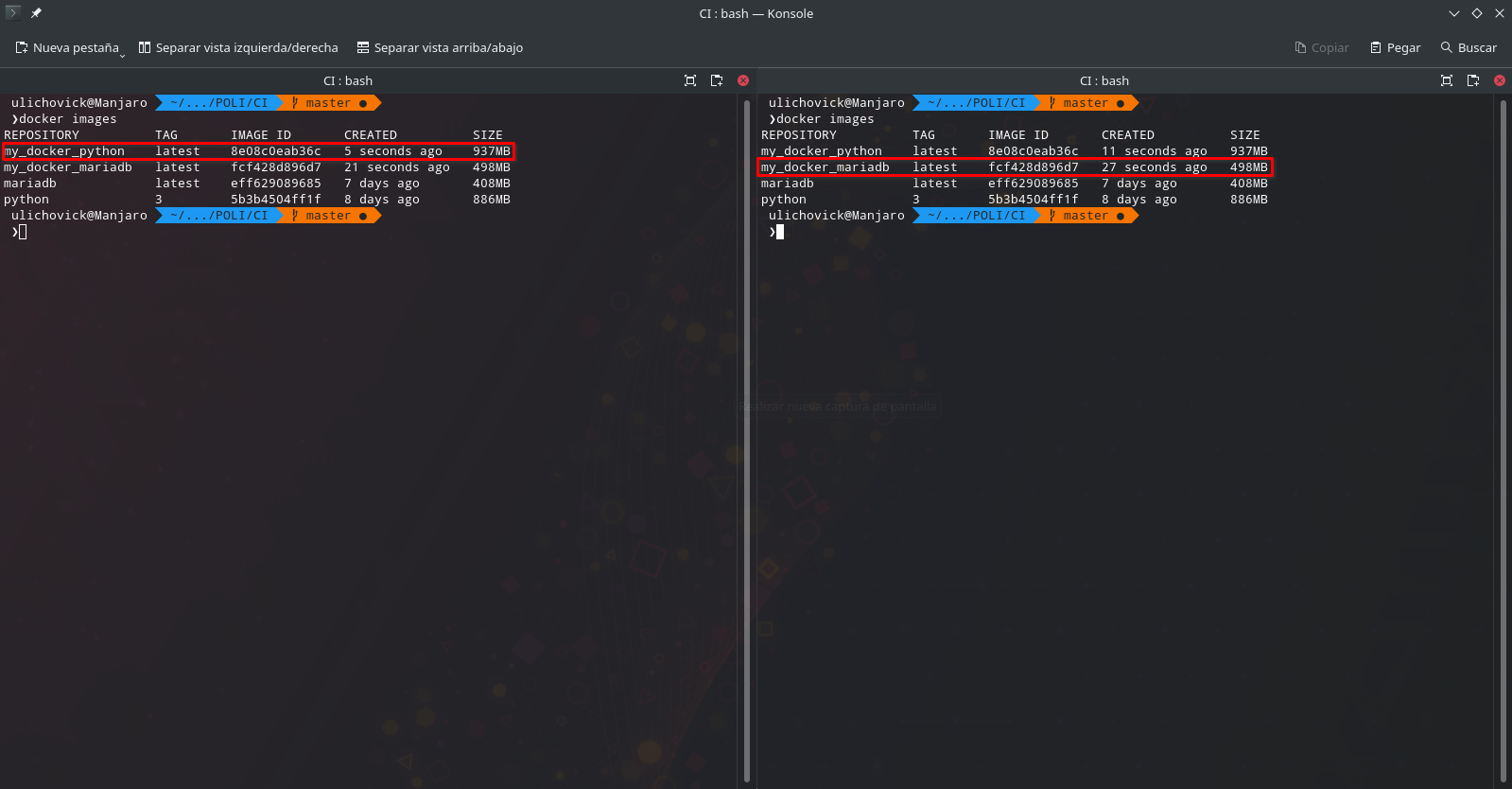
docker build -t my\_docker\_python my\_docker\_python/

crear imágen de mariadb en docker

docker build -t my\_docker\_mariadb my\_docker\_mariadb/



*Paso 2. Construir Imagenes*

#Paso 3. Verificar imágenes:

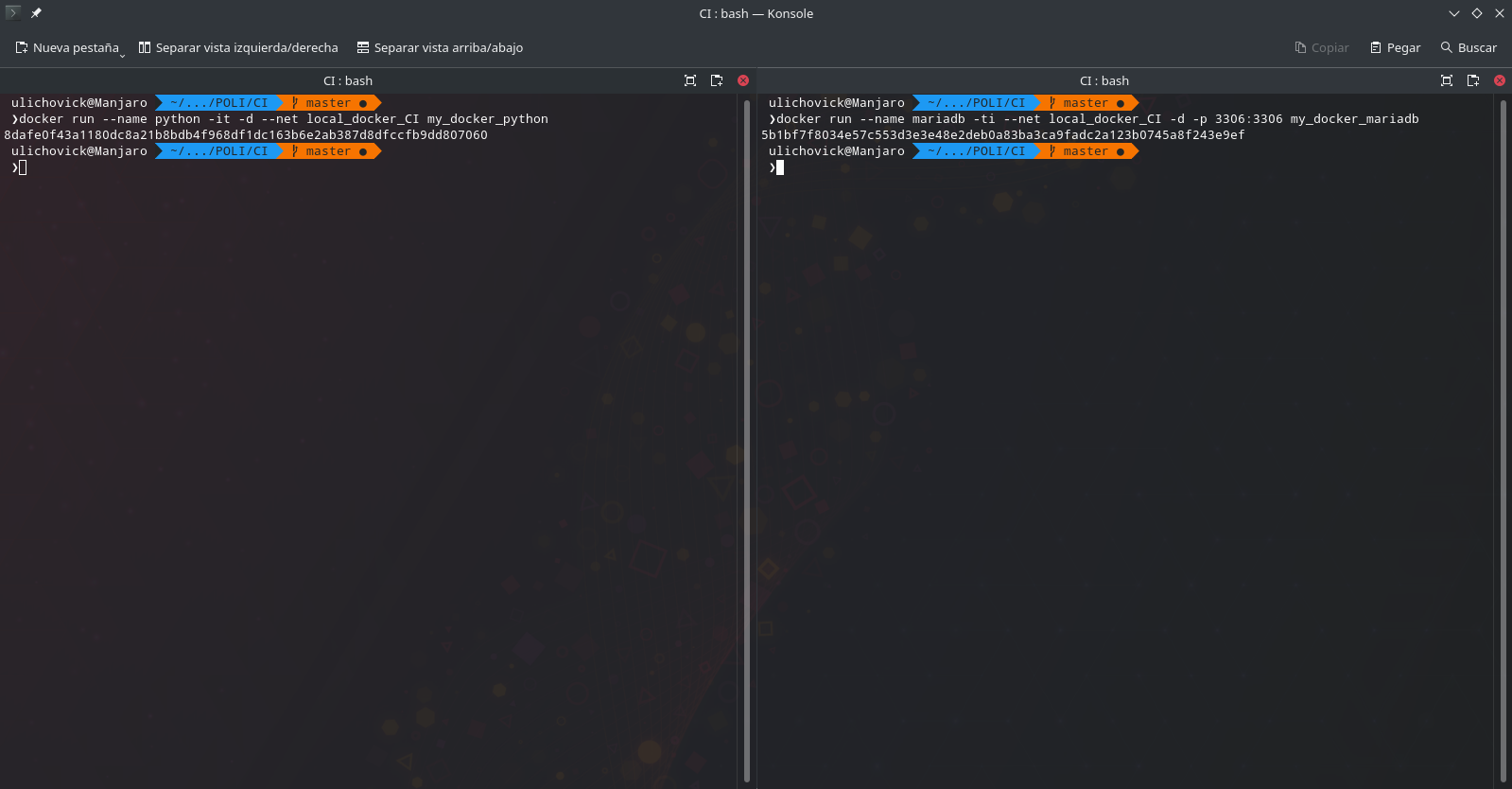
*Paso 3. Verificar Imagenes*

#Paso 4. Construir Contenedores:

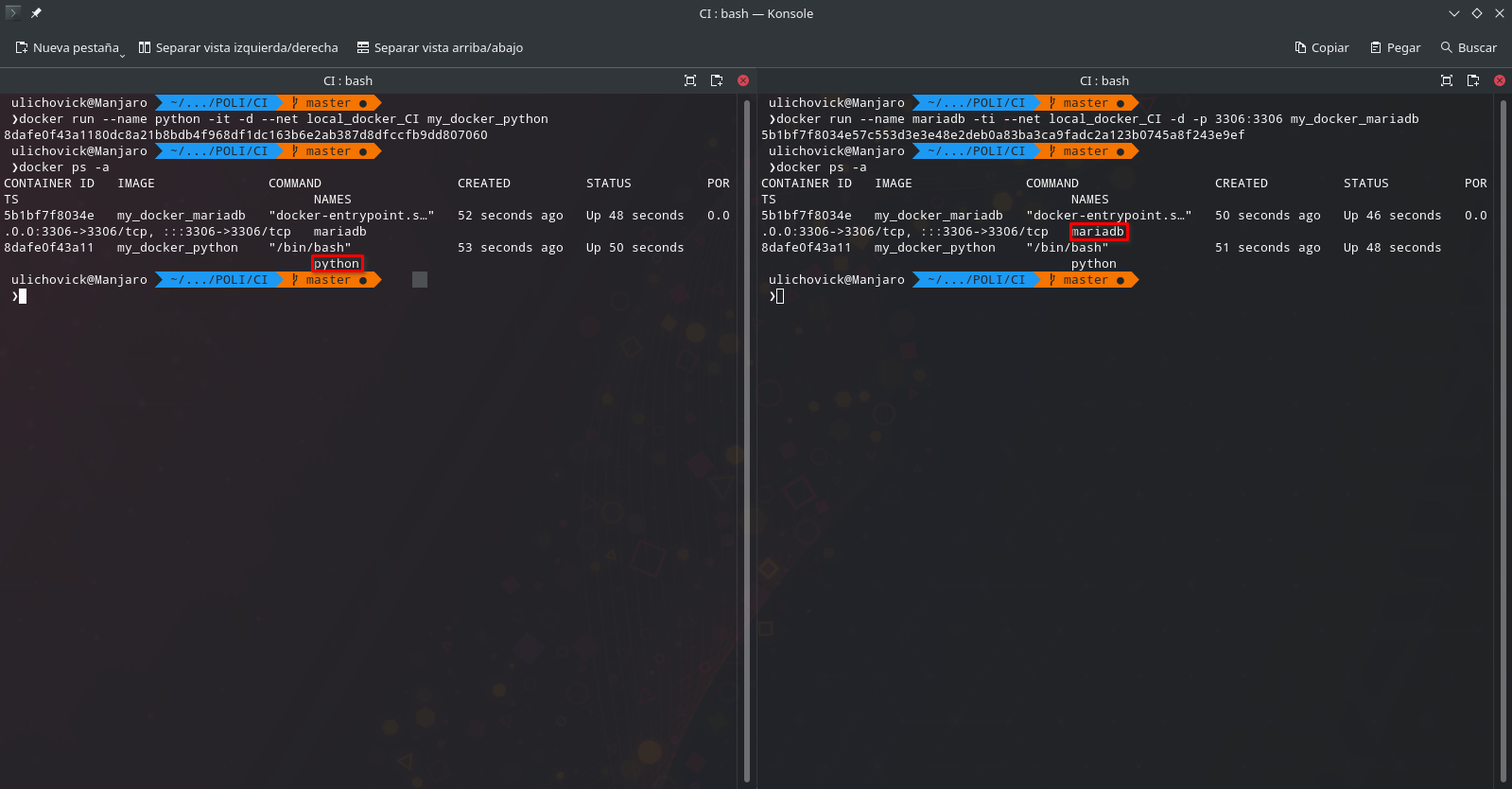
crear contenedor de docker en modo interactivo, que corra en segundo plano y que esté anclado a la red previamente creada

docker run --name python -it -d --net local\_docker\_CI my\_docker\_python

docker run --name mariadb -ti --net local\_docker\_CI -d -p 3306:3306 my\_docker\_mariadb

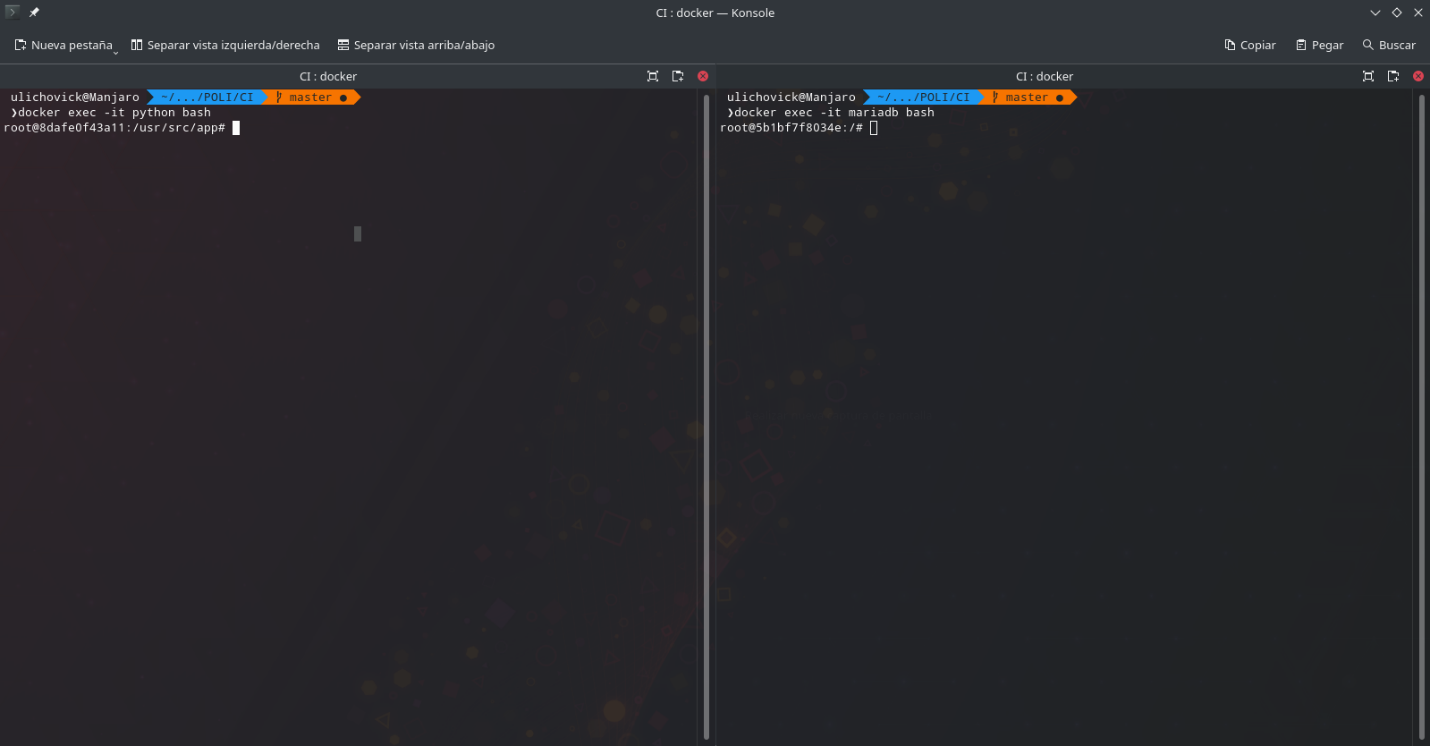


*Paso 4. Construir Contenedores*

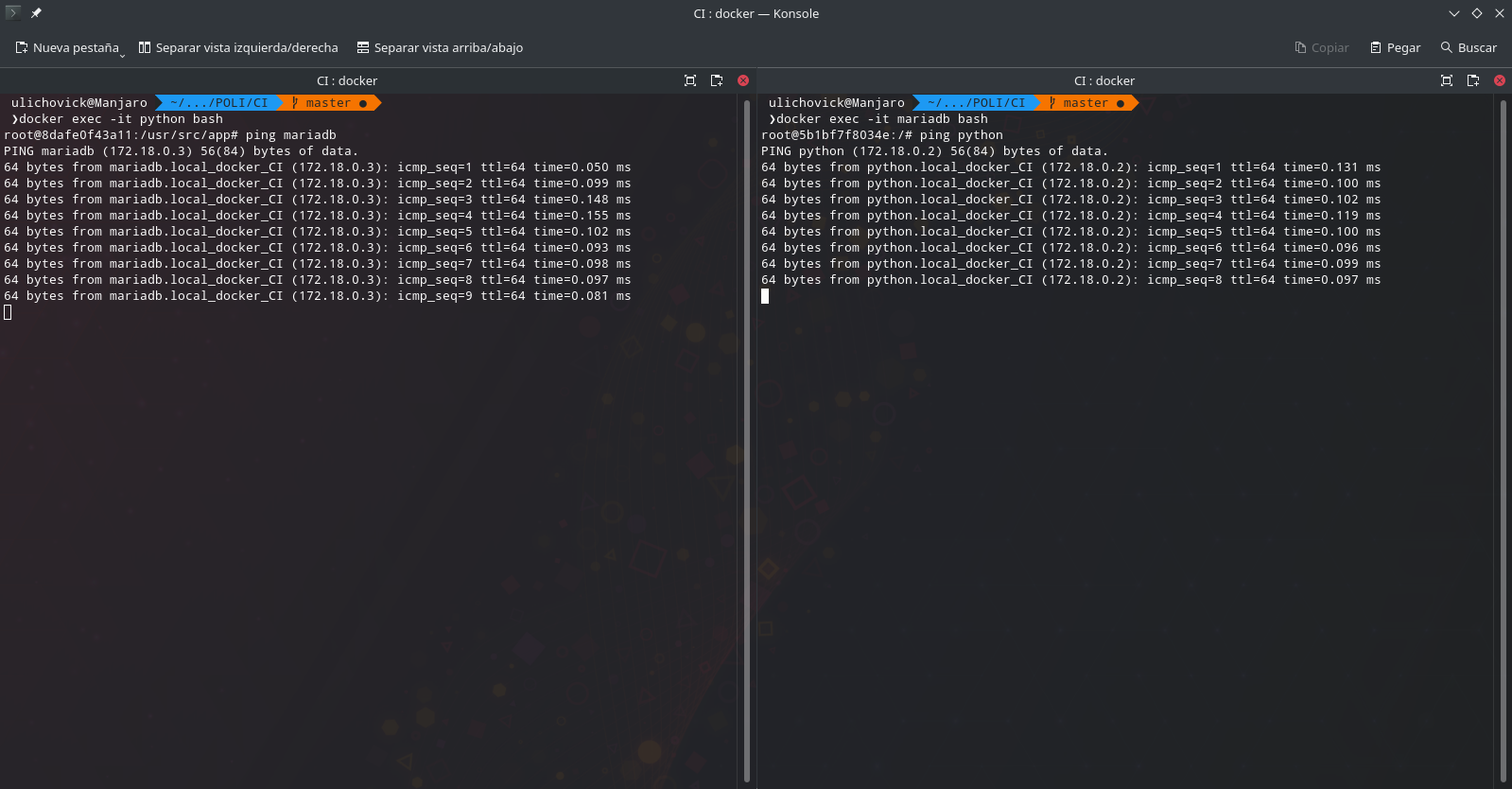
#Paso 5. Verificar Contenedores:

*Paso 5. Verificar Contenedores*

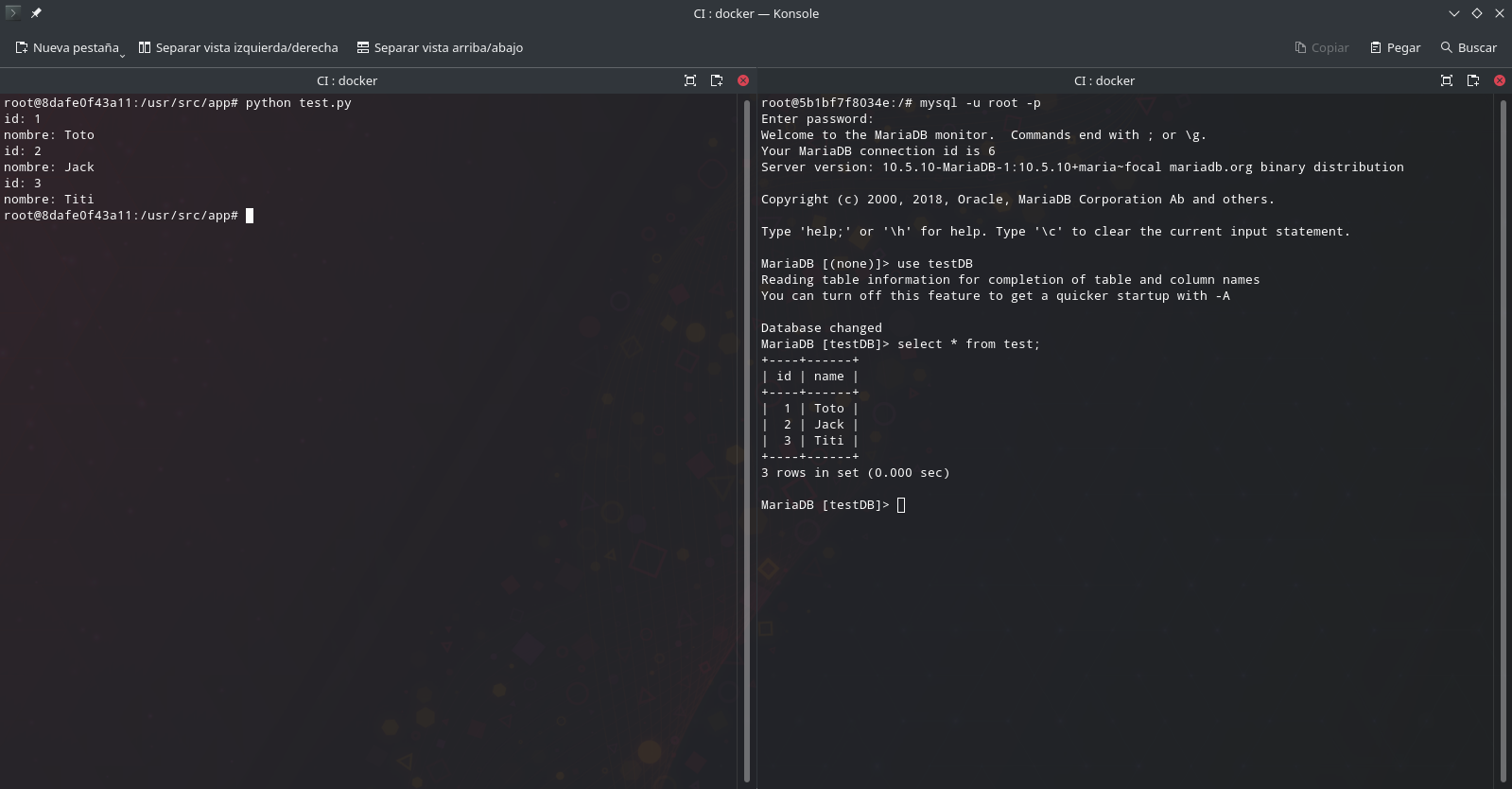
#Paso 6. correr contenedor de manera interactiva

docker exec -it mariadb bash

*Paso 6. Correr Contenedores*

#Paso 7. Ping entre Contenedores

*Paso 7. Ping entre Contenedores*

#Paso 8. Datos de mariadb a py

Paso 8. Datos de mariadb a py

1. Conclusiones

* Trabajamos con la herramienta GitHub y lo primero que reconocemos es la gran comunidad de desarrolladores de todo el mundo que utilizan esta plataforma web gratis, rápida y eficiente para repositorios y proyectos, para utilizarla lo único que necesitamos es instalar Git en nuestro equipo de cómputo, crear tu cuenta en GitHub y luego de realizar algunas configuraciones sencillas y ejecutar unos comandos sencillos podemos enviar documentos o archivos con algún tipo de código y luego invitar a tus colegas o grupo de trabajo para que tengan acceso a esa información y poderla modificar por parte de todos los invitados a través de una clonación, modificación, adición y los famosos commits que quedan grabados en la misma plataforma.
* GitHub es una excelente herramienta para el trabajo en equipo y facilita mucho la creación y desarrollo de proyectos, el equipo de trabajo puede trabajar al mismo tiempo desde diferentes lugares, diferentes ciudades incluso diferentes países del mundo. Además, el no tener copias de seguridad y la garantía de poder volver al estado inicial por si se cometen errores hace que sea muy versátil trabajar y muy seguro.
* También trabajamos con Docker que es una plataforma que empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute. Los contenedores virtualizan el sistema operativo de un servidor. Docker se instala en cada servidor y con comandos sencillos permite crear, iniciar o detener contenedores.
* Utilizamos imágenes de Docker las cuales son plantillas de solo lectura que definen al contenedor, estas imágenes contienen un código que se va a ejecutar y así esta ejecución es la que da como resultado un contenedor.
* Nos dimos cuenta desde el inicio que Docker utiliza ciertos aspectos de Linux por lo que si no teníamos este sistema operativo era mejor trabajar en una máquina virtual Linux. Con respecto a la comunicación de los contenedores se prefirió utilizar una red virtual personalizada y no la propia red de Docker para ahorrarnos los pasos de buscar las IP de los contenedores y en vez de eso conectarlos con su alias.