# Persistencia de Datos en Docker - Taller 6 de mayo

\*\*Nombre:\*\* [Tu nombre]

\*\*Fecha:\*\* 6 de mayo de 2025

## 1. CRUD en Docker

El sistema CRUD fue desarrollado con Python 3.13, SQLite y PyQt6. Este permite registrar, actualizar, listar y eliminar estudiantes desde una interfaz gráfica.

## 2. Dockerfile

FROM python:3.11-slim  
  
WORKDIR /app  
  
COPY requirements.txt .  
RUN pip install -r requirements.txt  
  
COPY . .  
  
CMD ["python", "main.py"]

## 3. docker-compose.yml

version: "3.8"  
  
services:  
 crud-app:  
 build: .  
 volumes:  
 - ./data:/app  
 stdin\_open: true  
 tty: true

## 4. Ejecución del contenedor

Para ejecutar el contenedor, se utiliza el siguiente comando:

docker-compose up --build

## 5. Ingreso de 3 registros

Al iniciar la aplicación, se ingresan tres estudiantes mediante la interfaz. Se observa la base de datos actualizada en la ventana de listado.

(Aquí se adjunta una captura de la interfaz con los 3 registros).

## 6. Reejecución y verificación de persistencia

Al volver a ejecutar el contenedor, se observa que los datos siguen almacenados gracias al volumen definido en el archivo docker-compose.yml.

## 7. Conclusión

El uso de Docker con volúmenes permite conservar los datos incluso si el contenedor se detiene o se elimina, asegurando la persistencia de la base de datos SQLite.