

“Ciencia, tecnología y cultura al servicio del ser humano”

Proyecto Final

SISTEMAS VIRTUALES

Grupo: T57A

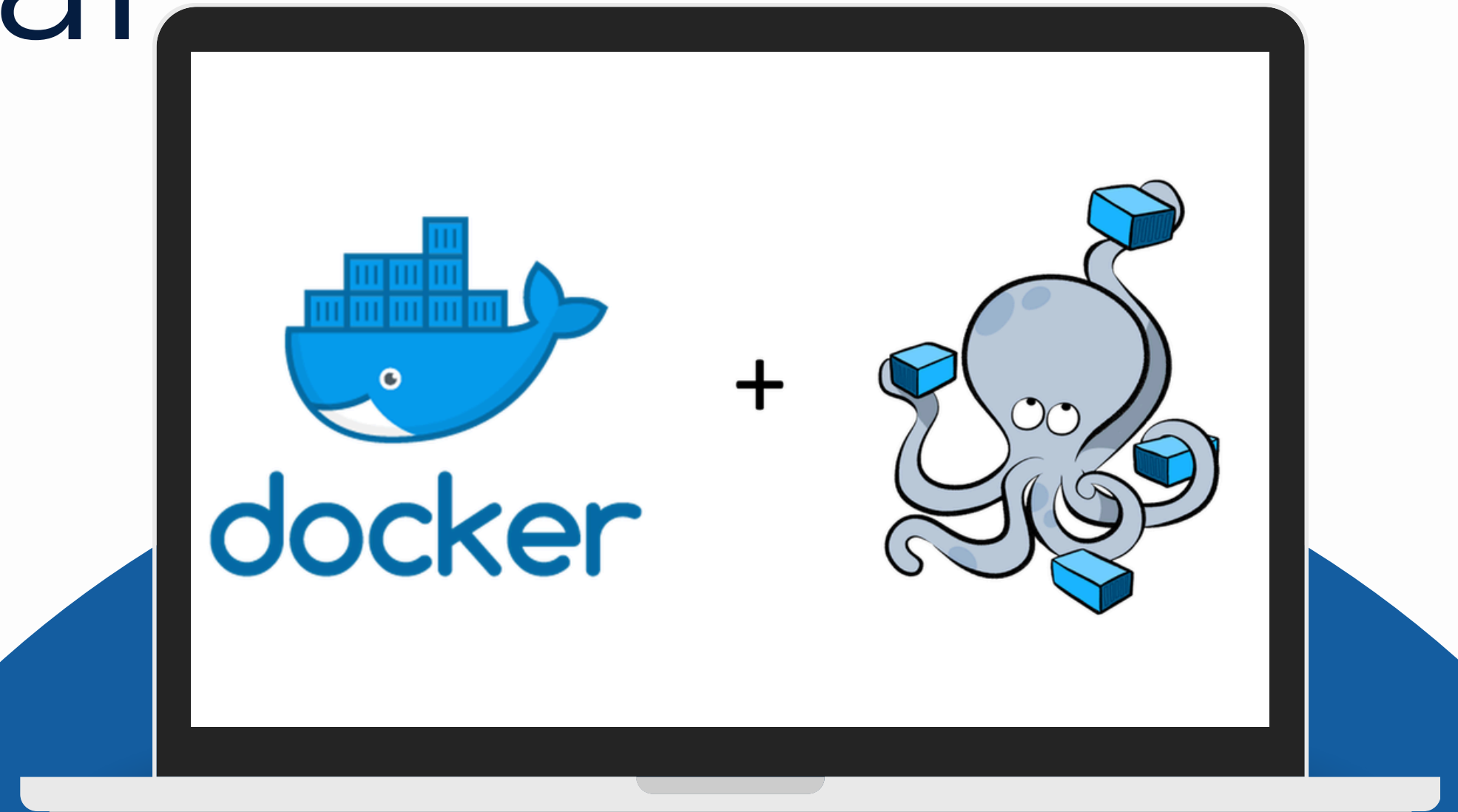
A cargo de: Mtro. Manuel Chávez

Carrera: ITI

Integrantes:

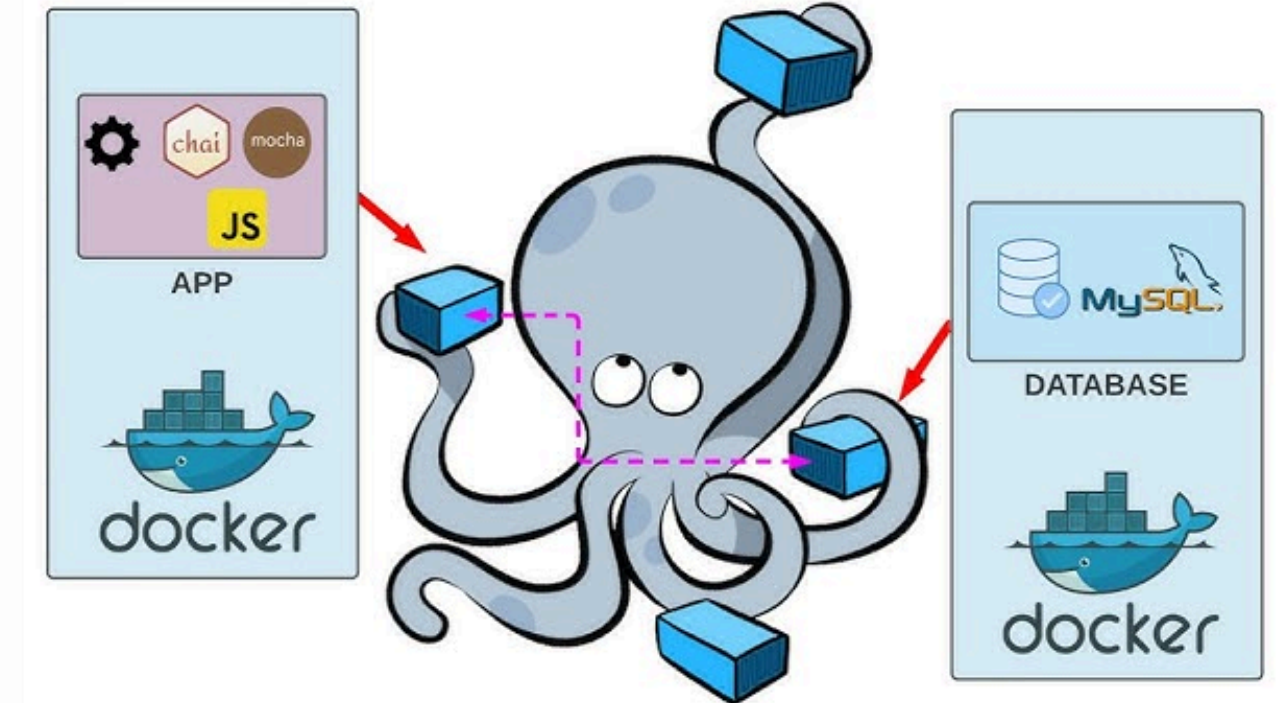
Christian Aarón Zavala Sánchez 171817

Aneth Alejandra López Cerda 173577



ÍNDICE

▶▶▶	Objetivos y Alcance	01
▶▶▶	Arquitectura Tecnológica	02
▶▶▶	Estructura del Proyecto	03
▶▶▶	Dockerfile y Compose	04
▶▶▶	Flujo de Desarrollo y Despliegue	05
▶▶▶	Experiencia de Usuario	06
▶▶▶	Conclusión	07





+



Objetivos y Alcance

Objetivo principal:

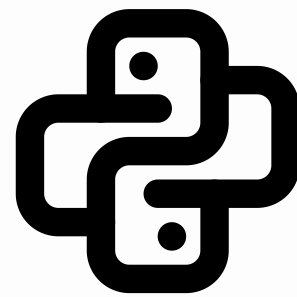
- Crear una app web que capture datos de estudiantes y los almacene en MySQL

Alcance:

- Interfaz de formulario en Flask
- Base de datos MySQL contenerizada
- Persistencia de datos ante reinicios

Arquitectura Tecnológica

- Docker Compose orquesta dos servicios y una comunicación interna por red Docker



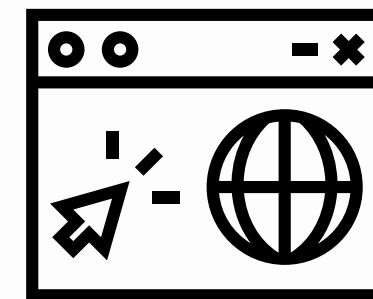
web: Flask +
Python 3.11

Micro-framework ligero que gestiona rutas, peticiones HTTP y renderiza plantillas Jinja para un desarrollo ágil y modular.



db: MySQL 5.7 con
volumen persistente

Sistema de base de datos relacional en contenedor, con un volumen dedicado que garantiza que los datos sobrevivan a reinicios.



web → db:3306

Comunicación interna sobre la red de Docker Compose, donde el servicio web resuelve al host db en el puerto 3306 para ejecutar consultas SQL.



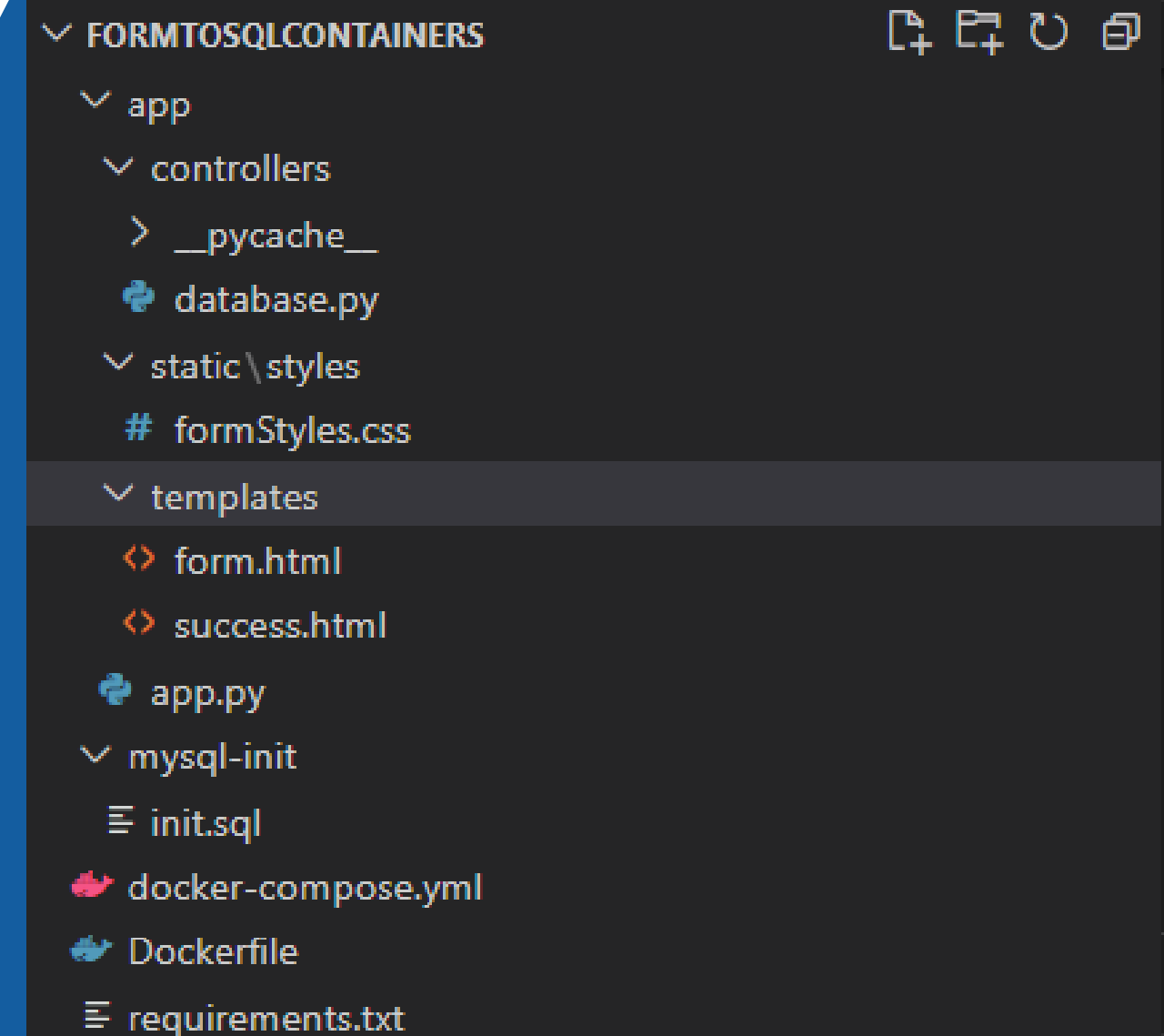
Estructura del Proyecto

Estructura del Proyecto

- app/
 - app.py → Punto de entrada de Flask
 - controllers/
 - database.py → Lógica de conexión y consultas MySQL
 - static/styles/
 - formStyles.css → Estilos del frontend
 - templates/
 - form.html → Formulario de captura
 - success.html → Página de confirmación
- mysql-init/
 - init.sql → Script de creación de BD y tabla
- Dockerfile → Imagen Python + dependencias
- docker-compose.yml → Orquestación de servicios (web + db)
- requirements.txt → Dependencias de Python

Un único comando levanta todo el entorno:

docker-compose up -d --build



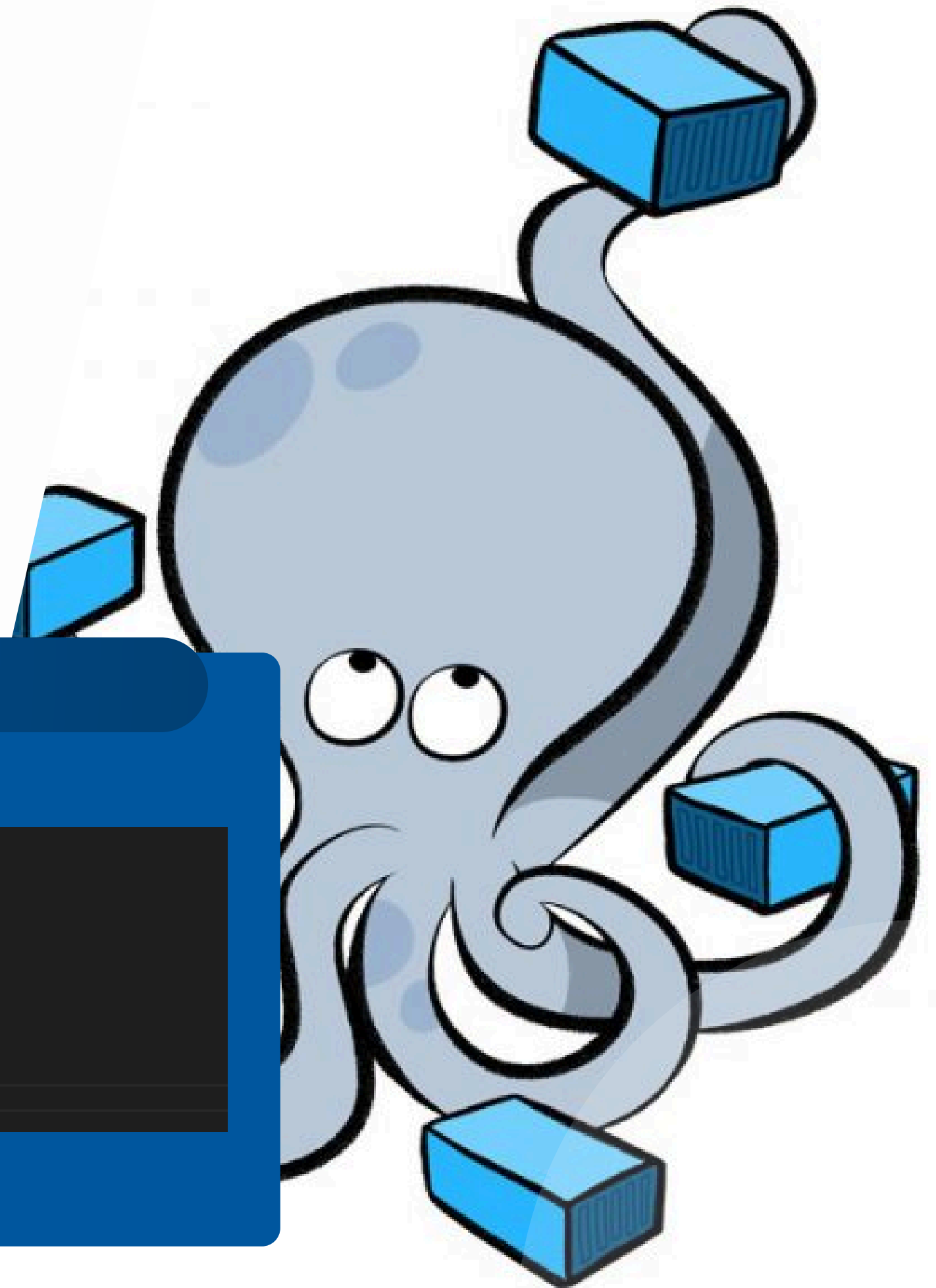
Dockerfile y Compose

Dockerfile

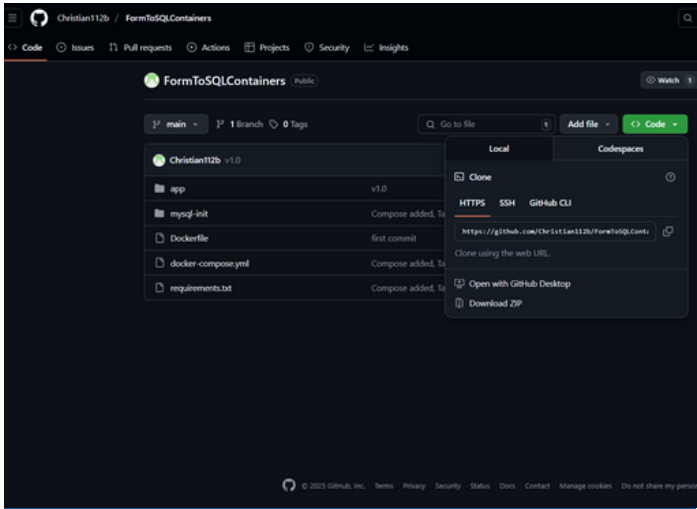
1. FROM python:3.11-slim
2. WORKDIR /app + copia app/ + dependencias
3. EXPOSE 5000 + CMD ["python","app.py"]

docker-compose.yml

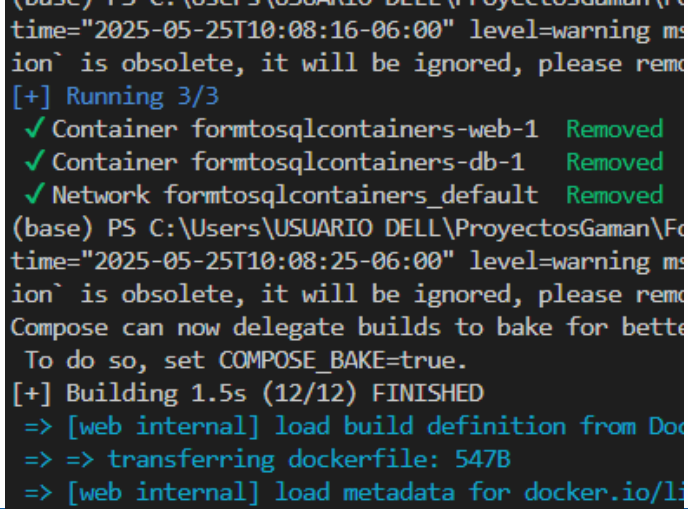
```
15 db:
16   image: mysql:5.7
17   restart: always
18   environment:
19     MYSQL_ROOT_PASSWORD: password
20     MYSQL_DATABASE: formulario
21   ports:
22     - "3307:3306"
23   volumes:
24     - ./mysql-init:/docker-entrypoint-initdb.d
```



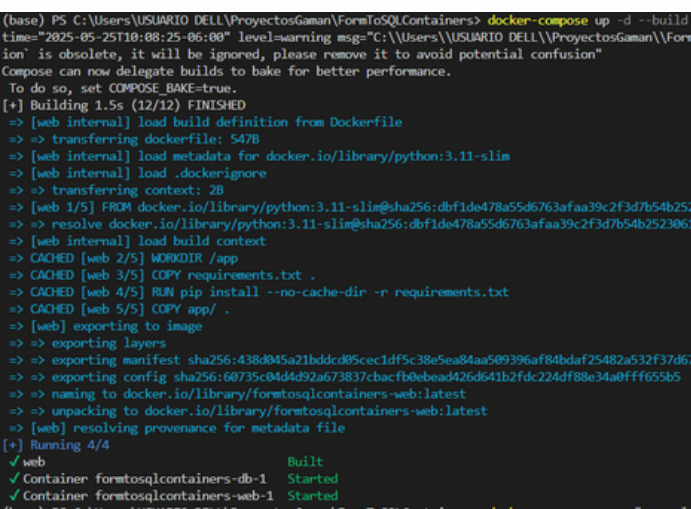
Flujo de Desarrollo y Despliegue



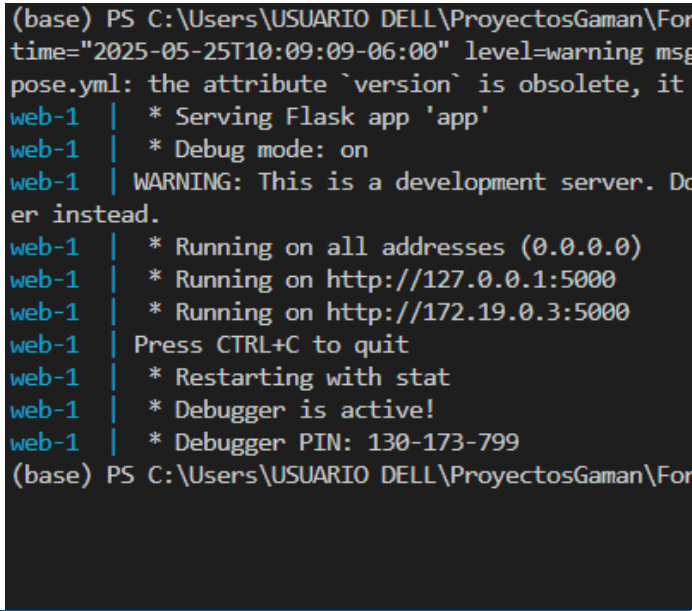
Clonar repo y ubicarse en carpeta



docker-compose down -v



docker-compose up -d --build



Verificar con docker-compose ps



Acceder a http://localhost:8000

Experiencia de Usuario

Ingrese nombre completo

Por favor ingrese un nombre válido

Edad *

Ej: 22

Sexo *

Seleccione

Carrera *

Seleccione

Semestre *

Seleccione

Matrícula *

Ej: 173577

Fecha *

dd/mm/aaaa

Fecha nacimiento *

Lugar de nacimiento *

Formulario limpio con etiquetas claras

```

seError
:tor.errors.DatabaseError: 1366 (HY000): Incorrect integer value: '' for column 'edad' at row 1

most recent call last)

@lib/python3.11/site-packages/mysqldb/connector/connection_ext.py", line 772, in cmd_query
mysql.query(

ception was the direct cause of the following exception:

@lib/python3.11/site-packages/flask/app.py", line 1536, in __call__
self.wsgi_app(environ, start_response)
.....

@lib/python3.11/site-packages/flask/app.py", line 1514, in wsgi_app
: = self.handle_exception(e)
.....

@lib/python3.11/site-packages/flask/app.py", line 1511, in wsgi_app
: = self.full_dispatch_request()
.....

@lib/python3.11/site-packages/flask/app.py", line 919, in full_dispatch_request
f.handle_user_exception(e)
.....

@lib/python3.11/site-packages/flask/app.py", line 917, in full_dispatch_request
f.dispatch_request()
.....

@lib/python3.11/site-packages/flask/app.py", line 902, in dispatch_request
elf.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]
.....

> py", line 20, in subeitt
t_estudiante(

rollers/database.py", line 36, in insert_estudiante
ute(query, (nombre, edad, sexo, carrera, semestre, matricula, fecha, fecha_nacimiento, lugar_nacimiento))

```

Validaciones básicas de tipo y formato (fechas, numérico)

Tabla de estudiantes

NOMBRE	EDAD	SEXO	CARRE...	SEMES...	MATRÍCULA	FECHA	FECHA NAC...	LUGAR NAC...
Aneth López	22	Mujer	ITI	9+	87	2025-05-...	2002-09-...	San Luis ...
Cristian Zavala	24	Hom...	ITI	9+	25	2025-05-...	2000-11-...	San Luis ...
Cesar García	23	Hom...	LAG	6	173588	2025-05-...	2001-03-...	San Luis ...

Visualización inmediata en la “Tabla de estudiantes”

Docker nos ha ofrecido un entorno totalmente aislado y reproducible, encapsulando la aplicación Flask y la base de datos MySQL en contenedores que se levantan, detienen y escalan con un solo comando gracias a Docker Compose. El uso de volúmenes garantiza la persistencia de datos tras reinicios, mientras que la red interna y los healthchecks coordinan el arranque ordenado de servicios sin depender de IPs fijas. Montar el código como volumen acelera la iteración durante el desarrollo y optimizar las capas en el Dockerfile reduce drásticamente los tiempos de build. En conjunto, esta infraestructura simplifica el despliegue, asegura consistencia en cualquier máquina y prepara la solución para futuras mejoras en producción.

THANK YOU!

