Universidad de Ingenieria y Tecnologia
Departamento de Ciencias de la Computación

Ingeniería de Software I

Lab VII: Pipelines & Project Structure
2024 – II

TEAM	**+Qtu**
INTEGRANTES	Kalos Lazo (Líder) Lenin Chavez Matias Castillo Gabriel Blanco Jose Tlpula Valeria valdez

Background:
Pipelines
Project Structure

CASO DE ESTUDIO (level: advanced):

Devies Inc. tiene un problema constante con sus nuevos empleados que no siguen los estándares que espera management. Se tienen constantes pérdidas de tiempo al hacer onboarding de nuevos empleados. Y también cuando se hacen integraciones de código es muy difícil combinar los diferentes estilos de código.

Problema:
Se le contrata como devops para desarrollar herramientas para resolver esta situación. Por lo que usted decida crear un CLI en donde por comandos se podrá generar el esqueleto de la aplicación estándar de la empresa.

1. **Elegir la estructura de proyecto estándar. (3-tier, event driven, etc)**
- Se considera utilizar como estructura de proyecto el **3-tier,** dado que este proyecto no tiene el objetivo de variar mucho en el tiempo. No se planea tener una interfaz gráfica, todo se maneja por consola, y al ser una herramienta que debe ser flexible y sencilla, se elige esta arquitectura.

2. **Crear una CLI que genere el esqueleto de proyecto en base a lo seleccionado en el punto 1 para un CRUD y otro para crear una API con conexión a base de datos.**
3. **La API creada debe contener un endpoint /hello que devolverá un json sample.**

Ejemplo:

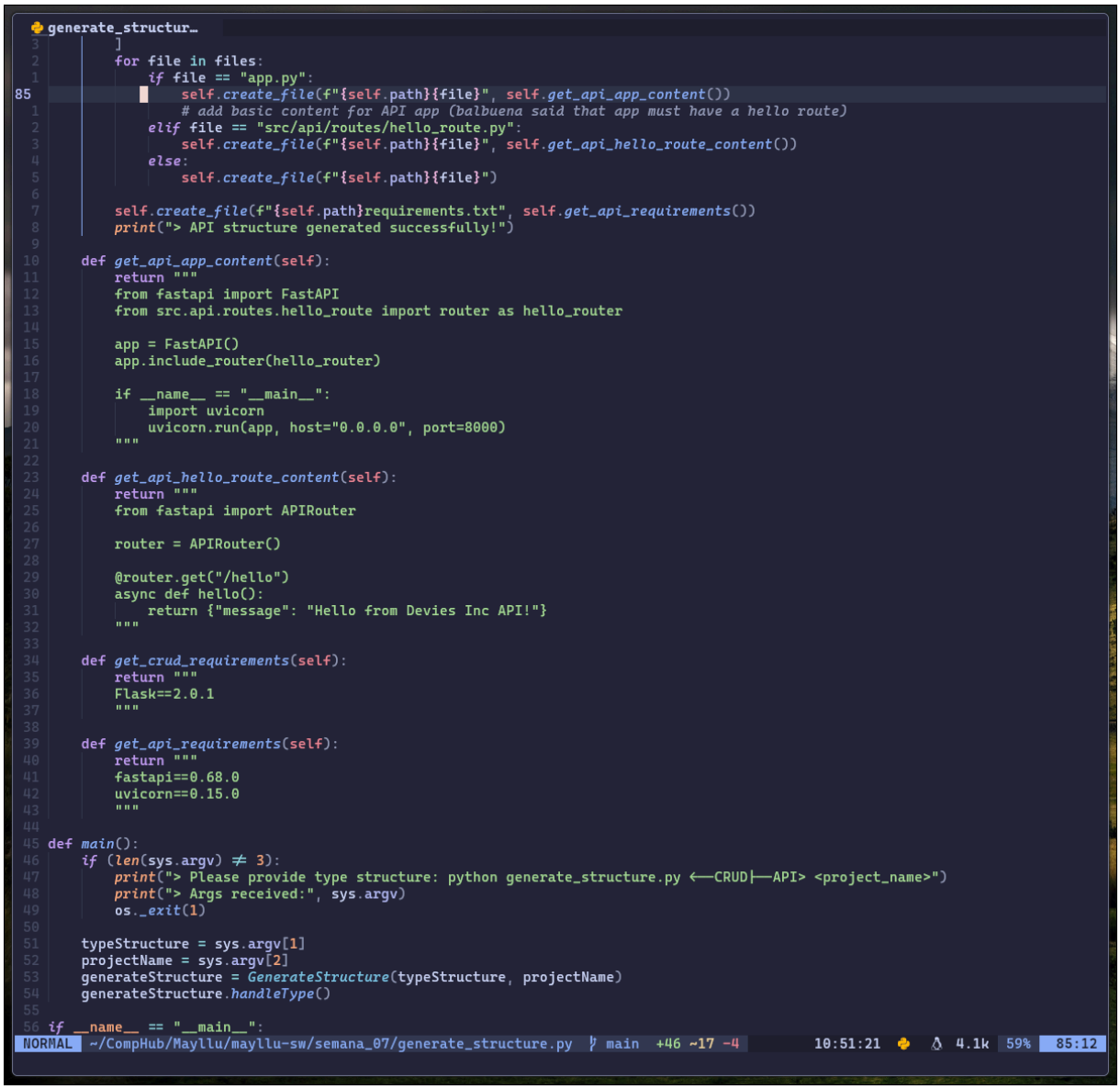
>> generate –crud newProject
..generating CRUD
..output of project structure in : C:/newProject

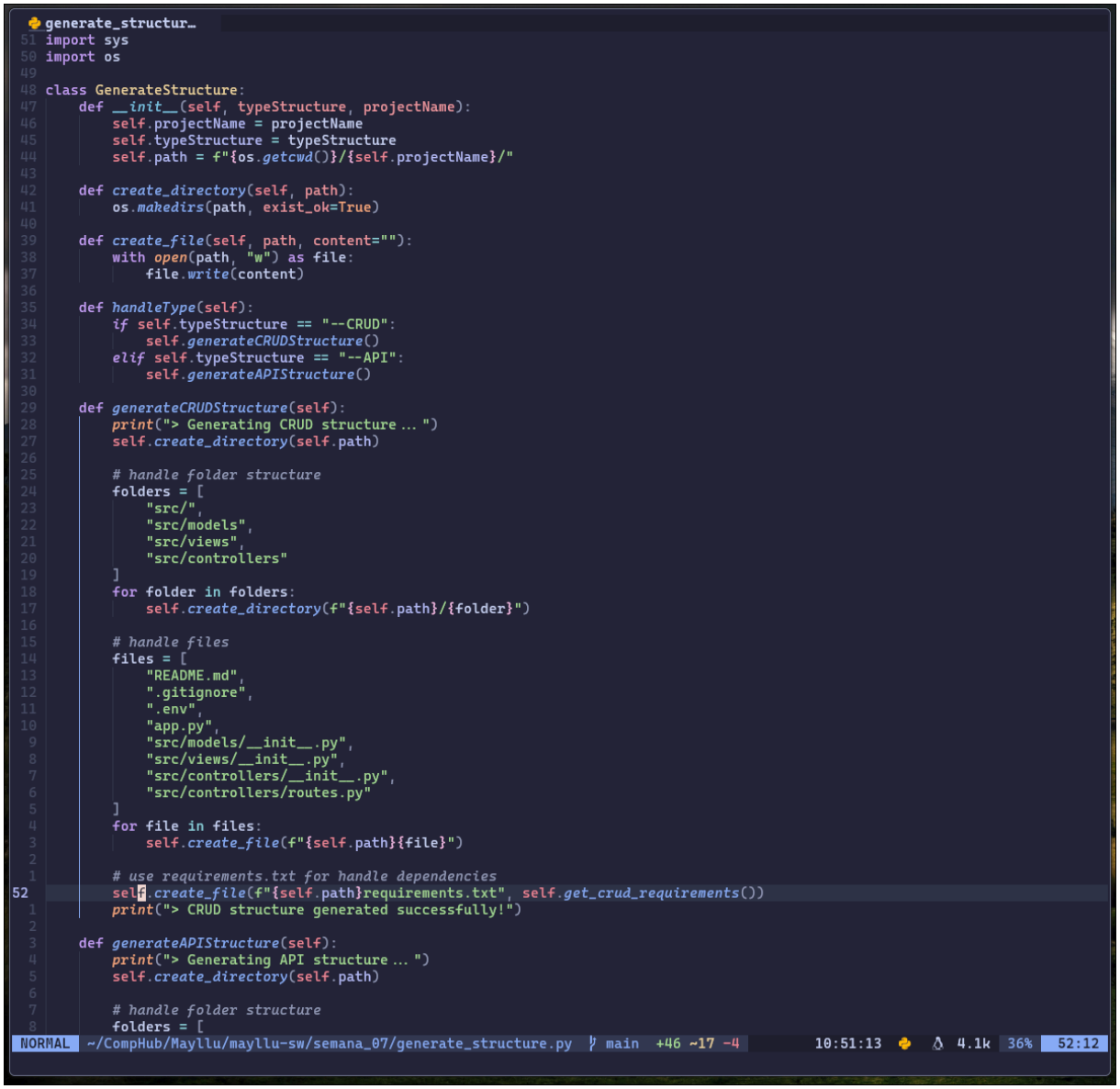
>>generate -api newAPI
..generating CRUD
..output of project structure in : C:/newAPI

Coloque la url del Repo de su CLI y screenshots del output

Se adjunta URL y además capturas tanto de código como de output con estructura creada. Se puede observar que se puede crear un archivo externo .json para guardar los paths y así mantenerlo de manera más adecuada, el generador es muy fácil de actualizarse en caso así lo requiera.

Link código:
https://github.com/utecsw20242/Mayllu/blob/main/semana_07/generate_structure.py





Project API structure (considerando hello path):


~/CompHub/Mayllu/mayllu-sw/
src
api
middlewares
routes
hello_route.py
validators
core
db
models
services
tests
.env
.gitignore
README.md
app.py
requirements.txt
hello_route.py
app.py
from fastapi import FastAPI
from src.api.routes.hello_route import router as hello_router
app = FastAPI()
app.include_router(hello_router)
if __name__ == "__main__":
import uvicorn
uvicorn.run(app, host="0.0.0.0", port=8000)