

C1 | PRIME TRIMESTRE

Z	AA	AB	AC	AD	AE	AF
TRIMESTRE	TEMAS PUNTUALES	INSTRUCTORES SUGERIDOS	ENTREGABLES DE PROYECTO	COMPETENCIA ASOCIADA	RESULTADOS ASOCIADOS	
C#, PHP , PYTHON ) 6 HORAS	ESTRUCTURAS DE CONTROL DEL LENGUAJE, POO(METODOS , CLASES , HERENCIAS ), CRUD , INTRODUCCION ARQUITECTURA ( MVC) (Claridad Dar continuidad al Lenguaje Visto en Tercer Trimestre)	EDWY PATIÑO- JAVA , EDILFREDO PINEDA- PHP , C#, LILLIANA URIBE[php,python,java, JS, DARK], HECTOR MAYA, JUAN DAVID CARVAJAL, GUILLERMO GURGOS, JESUS PACHECO	PROYECTO AL 70% EN LENGUAJE NATIVO Y CRUD CON FRAMEWORK, VALIDACIONES JAVASCRIPT CON API, BASE DE DATOS, MATRÍCULA DE HORARIOS DE USUARIO, DICCIONARIO DE CLASES, DICCIONARIO DE DATOS ACTUALIZADO, LOGIN CON SESIONES, CASOS DE PRUEBA, APLICACIÓN DE METODOLOGÍAS: SPRING PLANNING, ROLES, UTILIZACIÓN DE HERRAMIENTAS NOTION O MIRO, TABLERO CANVAS O DASH BOARD, PROYECTO EN GIT, MVT,MVT	DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO	04 CODIFICAR EL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO.	
C#, PHP , PYTHON ) 6 HORAS	ESTRUCTURAS DE CONTROL DEL LENGUAJE, POO(METODOS , CLASES , HERENCIAS ), CRUD , INTRODUCCION ARQUITECTURA ( MVC) (Claridad Dar Nuevo Lenguaje diferente al Visto en Tercer Trimestre)	EDWY PATIÑO: JAVA - EDILFREDO PINEDA: PHP , C#, LILLIANA URIBE[php,python,java, JS, DARK], HECTOR MAYA, JUAN DAVID CARVAJAL, GUILLERMO GURGOS, JESUS PACHECO	PROYECTO AL 70% EN LENGUAJE NATIVO Y CRUD CON FRAMEWORK, VALIDACIONES JAVASCRIPT CON API, BASE DE DATOS, MATRÍCULA DE HORARIOS DE USUARIO, DICCIONARIO DE CLASES, DICCIONARIO DE DATOS ACTUALIZADO, LOGIN CON SESIONES, CASOS DE PRUEBA, APLICACIÓN DE METODOLOGÍAS: SPRING PLANNING, ROLES, UTILIZACIÓN DE HERRAMIENTAS NOTION O MIRO, TABLERO CANVAS O DASH BOARD, PROYECTO EN GIT, MVT,MVT	DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO	04 CODIFICAR EL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO.	
O WEB - DOCUMENTACION ANALISIS 16 HORAS EL INSTRUCTOR PROGRAMAR EL EDT CON RENDIMIENTO	FUNDAMENTOS DE CALIDAD: CONCEPTOS, GESTIÓN DE PROCESOS, NORMAS ISO/IEC 25000 SQUARE, ISO/IEC 15504, IEEE, PSP FUNDAMENTALS, TSP	EDWIN PATIÑO, EDILFREDO PINEDA , GUILLERMO BURGOS, JESUS PACHECO, LILLIANA URIBE, HAROL GOMEZ , HECTOR MAYA, JUAN DAVID CARVAJAL		Controlar la calidad del servicio de software de acuerdo con los estándares técnicos	02 VERIFICAR LA CALIDAD DEL SOFTWARE DE ACUERDO CON LAS PRÁCTICAS ASOCIADAS EN LOS PROCESOS DE DESARROLLO. 01 INCORPORAR ACTIVIDADES DE ASEGURAMIENTO DE LA CALIDAD DEL	AS

Diagramas de clase de proyecto oop 4 tablas de la base de datos( esto se replanteo por que los proyectos son grades)

lo que vamos hacer:

<https://github.com/LillianaU/AgendaORMRelacional/tree/main/EjemploModular>

23 noviembre

```

EjemploModular/
    ├── models/                  # Clases POO
    │   ├── __init__.py
    │   ├── database.py          # Conexión MySQL (Singleton)
    │   ├── persona.py           # Clase Persona (herencia, composición)
    │   ├── telefono.py          # Clase Teléfono (asociación)
    │   └── email.py              # Clase Email (asociación)

    ├── crud/                   # Operaciones con base de datos
    │   ├── __init__.py
    │   ├── persona_crud.py     # CRUD Persona + Login
    │   ├── telefono_crud.py    # CRUD Teléfono
    │   └── email_crud.py       # CRUD Email

    ├── tests/                  # Pruebas unitarias
    │   ├── __init__.py
    │   ├── test_modelos.py      # Tests de clases POO
    │   ├── test_cruds.py        # Tests de operaciones CRUD
    │   └── .....                 # Tests de otras operaciones CRUD clases

    ├── .env                     # Configuración BD (CREAR MANUALMENTE)
    ├── main.py                  # Aplicación principal (menú interactivo)
    └── test_runner.py          # Ejecutor de todas las pruebas

```

## meet

<https://meet.google.com/ixq-vgtb-jrb>

## base de datos

```

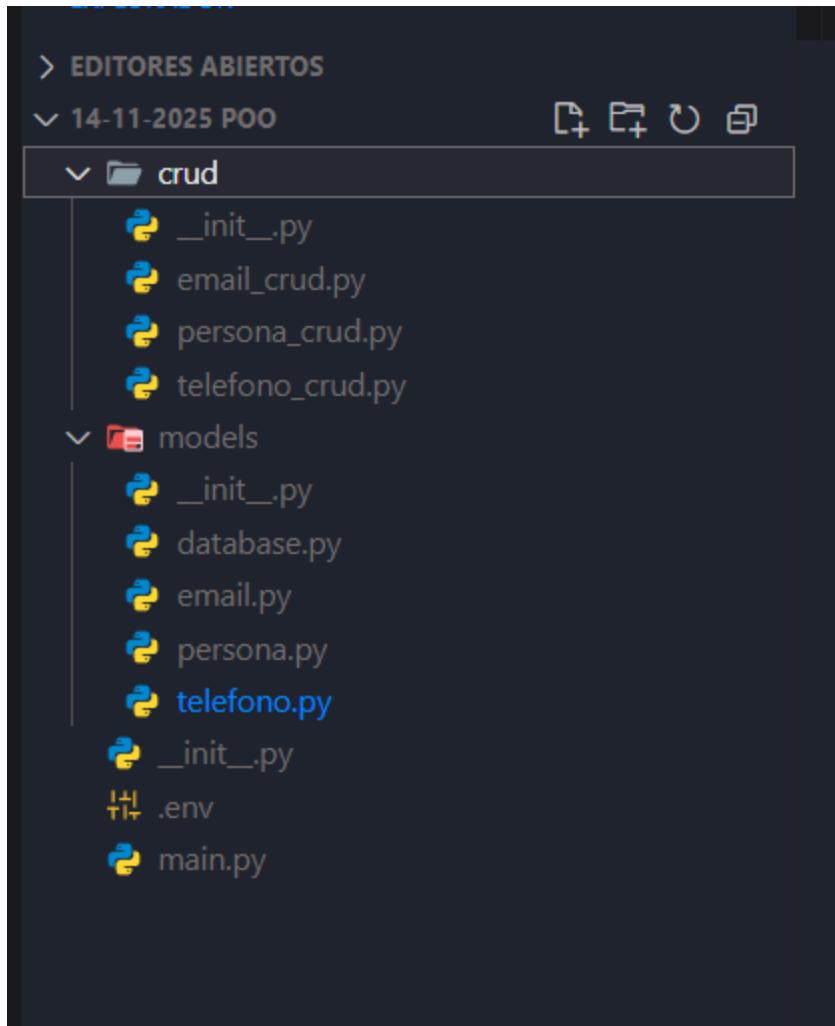
CREATE TABLE personas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellido VARCHAR(200) NOT NULL,
    username VARCHAR(50) UNIQUE NOT NULL,
    password_hash TEXT NOT NULL
);

```

```
CREATE TABLE telefonos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    persona_id INT NOT NULL,
    telefono VARCHAR(20) NOT NULL,
    FOREIGN KEY (persona_id) REFERENCES personas(id) ON DELETE CASCADE
);
```

```
CREATE TABLE emails (
    id INT AUTO_INCREMENT PRIMARY KEY,
    persona_id INT NOT NULL,
    email VARCHAR(100) NOT NULL,
    FOREIGN KEY (persona_id) REFERENCES personas(id) ON DELETE CASCADE
);
```

\_\_init\_\_.py



Bueno chicos yo les voy a explicar En qué consiste las arquitecturas de desarrollo de software como pueden observar es la Organización en carpetas del código Qué Busca las arquitecturas limpias muchachos de que cuando ustedes le van a hacer modificación al código mantenimiento o de pronto ya hay una parte del sistema que dejó de utilizarse sea fácil de mantener en el tiempo por eso cada que ustedes hacen un proyecto por ejemplo una de las arquitecturas que puede ser muy fácil de emigrar para otra arquitectura las modulares pero siempre que ustedes vean en el proceso que el proyecto puede crecer en el tiempo se busca implementar otras arquitecturas mucho más grandes como las micro servicios hexagonales que son las arquitecturas más grandes y las que más se utilizan en la industria también el modelo vista controlador y a otro que es el más popular que se trabaja mucho en el itm porque ellos se casaron con esa arquitectura es arquitectura de capas ellos son matados con eso pero después de que me gradué ya me entendí porque la enseñaban tanto y eran tan tan existente con eso que la arquitectura por capas prácticamente es una forma de aprender fácil para poderlo emigrar a unas arquitecturas más poderosas Entonces cuando hablamos de arquitecturas de desarrollo de software es cómo modelar el proyecto para que sea en el tiempo fácil de mantener actualizar o emigrar en su caso a otra criatura mucho más grandes entonces chicos bajo lo que ustedes tienen que entregar de fase que usted ya tiene por decirlo así en

este momento ya una un módulo de cuatro tablitas más o menos ustedes vamos a buscar una un proyectil un módulo que tenga integración de Cuatro Tablas ya ustedes pueden decir tenemos una estructura ya orientada objetos que perfectamente se puede emigrar para freímos de desarrollo de software ya sea djingo ya sea Pasa Api ya sea fácil o es su efecto la árabe ahorita más tarde les voy a hacer como unos diagramas para que ustedes vean Cómo es la inmigración de lo más de fácil de esto entonces ahí fue cuando yo entendí Con razón los profesores en el itm los tántate tanto azote para que aprendiéramos a programar primero modular y después por capas ya uno entiende que hay O sea que esta verraca carpa esta carpeta cambia de nombre luego hago las conexiones en el mail y ya salió para pinturas bueno suena muy fácil pero a la hora de la verdad muchachos para que coja las rutas absolutas y las rutas relativas de un camellito ahí bastante peculiar si ustedes no tienen bien organizada en el carpetado entonces en cierta forma que es una arquitectura organizada de una manera técnica en carpetas

crea el archivo requirements.txt

```
sqlalchemy==2.0.23
mysql-connector-python==8.1.0
cryptography==41.0.7
```

lo ejecuta asi

```
# como se ejecuta el requirements.txt
...
pip install -r requirements.txt
...
otra forma
...
python pip install -r requirements.txt
...
# persiste problema instala
python.exe -m pip install --upgrade pip
```

```
# como se ejecuta el requirements.txt
...
pip install -r requirements.txt
```

```
...
otra forma
...
python pip install -r requirements.txt
...

# persiste problema instala
python.exe -m pip install --upgrade pip

# verificar instalaciones

...
pip list
...

# otra forma de instalar separado
...
pip install mysql-connector-python==8.1.0
pip install cryptography==41.0.7

...
python -m pip install mysql-connector-python==8.1.0
python -m pip install cryptography==41.0.7
...
```

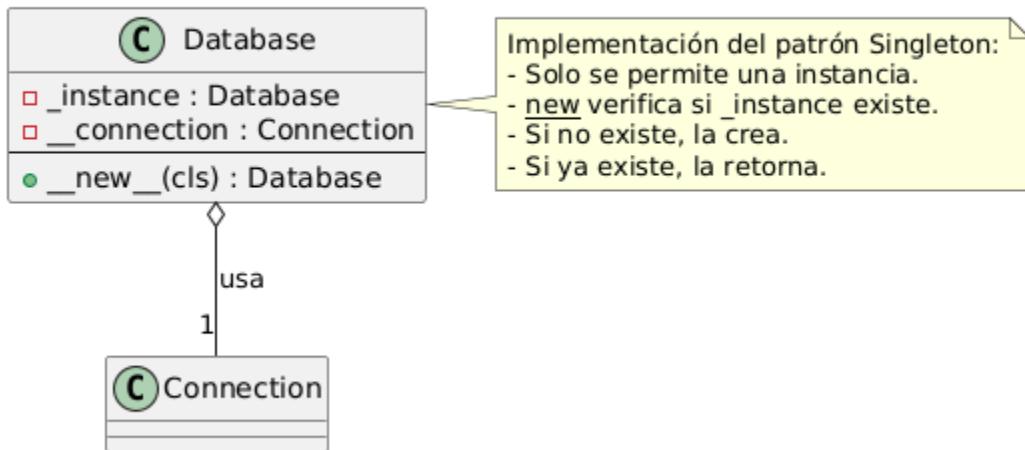
<https://refactoring.guru/es/design-patterns/singleton>

## que estamos haciendo:

1. Primero que todo qué es lo que estamos haciendo Recuerda que nosotros tenemos un diagrama de clases que les puse a ustedes como tarea les dije a ustedes que cogieron un solo modelito de su proyecto Eso significa Cuatro Tablas a esas Cuatro Tablas nosotros lo que vamos a hacerle mediante programación orientada a objetos y una arquitectura limpia que en este caso es modular simular un club con ese club estamos adquiriendo la lógica necesaria para ustedes trabajar con cualquier frame deba que ya sea laravel ya sea diyango ya sea springboot ya sea Fast ya sea con pass Api Entonces estamos en este momento que hicimos como Primavera ejecución del trabajo les mostré la arquitectura que estábamos haciendo

Entonces:

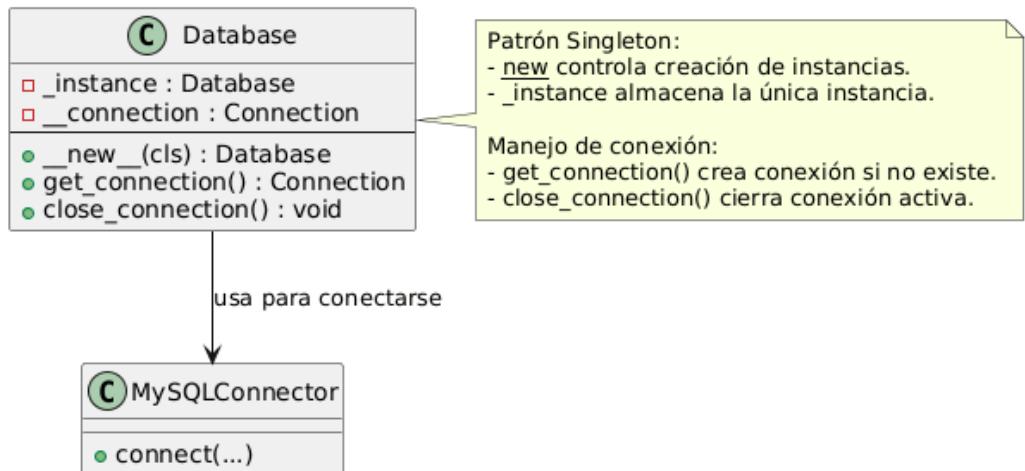
### Diagrama de la clase Database (Patrón Singleton)



<https://refactoring.guru/es/design-patterns/singleton>

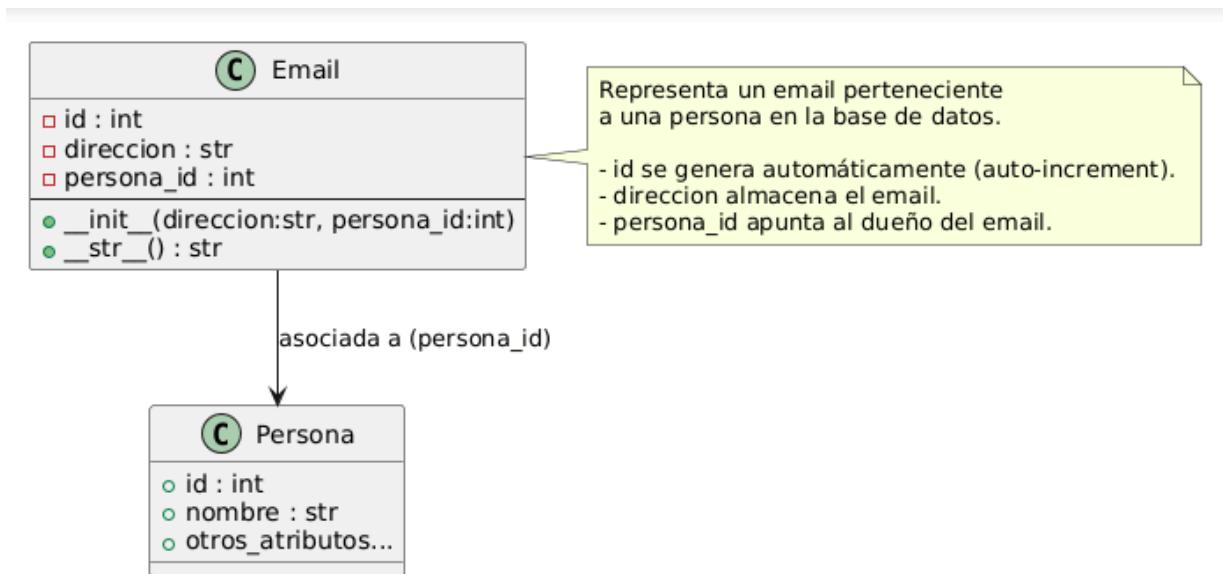
conexión lista

### Diagrama de la clase Database (Patrón Singleton + Conexión MySQL)





## Modelo email



## Modelo teléfono

