



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

Informe Final

**Desarrollo de un Sistema de Gestión de Usuario
en el dominio Configuración**

Curso: Base de Datos II

Docente: Ing. Patrick Jose Cuadros Quiroga

Integrantes:

<i>Chambe Torres, Edgard Reynaldo</i>	<i>2019064917</i>
<i>Nina Vargas, Luigui Augusto</i>	<i>2019065166</i>
<i>Arce Bracamonte, Sebastián</i>	<i>2019062986</i>
<i>Condori Vargas, Tomas Yoel</i>	<i>2018000487</i>
<i>Casilla Maquera, Tell Ivan</i>	<i>2017057888</i>

**Tacna – Perú
2023**

ÍNDICE GENERAL

1. Resumen.....	3
2. Abstract.....	3
3. Introducción.....	3
4. Planteamiento del Problema.....	4
5. Objetivos.....	5
6. Marco Teórico.....	5
7. Desarrollo de la Solución.....	6
8. Conclusiones.....	13
9. Bibliografía.....	13

1. Resumen

Este proyecto se propone desarrollar una API eficiente para la gestión de usuarios, haciendo uso de tecnologías contemporáneas como Redis y MySQL para almacenar datos no relacionales y relacionales, respectivamente. La implementación se lleva a cabo con Flask y Python, integrando Amazon Key Management Services (KMS) para asegurar una gestión segura de claves. La API permitirá a los administradores definir roles y permisos, simplificando la administración de cuentas de usuario y optimizando la eficiencia del sistema. Se aborda la falta de una solución integral para la gestión de usuarios, y se busca ofrecer una solución completa y moderna que garantice robustez y seguridad.

2. Abstract

This project aims to develop an efficient API for user management, making use of contemporary technologies such as Redis and MySQL to store non-relational and relational data, respectively. The implementation is carried out with Flask and Python, integrating Amazon Key Management Services (KMS) to ensure secure key management. The API will allow administrators to define roles and permissions, simplifying user account management and optimizing system efficiency. The lack of a comprehensive solution for user management is addressed, and the aim is to offer a complete and modern solution that guarantees robustness and security.

3. Introducción

Este proyecto tiene como objetivo principal desarrollar una API eficiente para la gestión de usuarios, utilizando tecnologías contemporáneas como bases de datos no relacionales, en este caso, Redis, y bases de datos relacionales, específicamente MySQL, junto con el servicio de gestión de claves Amazon Key Management Services (KMS). La implementación se lleva a cabo mediante el framework Flask en conjunto con el lenguaje de programación Python.

En este contexto, la API permitirá a los administradores definir roles y permisos, simplificando la administración de cuentas de usuario. La combinación de Redis para el almacenamiento de datos no relacionales, MySQL para la gestión de datos relacionales y Amazon KMS para la gestión segura de claves contribuirá a la robustez y seguridad del sistema. Los principios de Flask facilitarán el desarrollo ágil y la creación de endpoints eficientes.

Este proyecto busca ofrecer una solución completa y moderna para la gestión de usuarios a través de una API, aprovechando la sinergia entre Redis, MySQL, Amazon KMS, Flask y Python en el proceso.

4. Planteamiento del Problema

4.1. Problema

El problema abordado en este proyecto es la falta de una solución integral puede llevar a la ineficiencia en la administración de cuentas de usuario y a la falta de control sobre los recursos y funcionalidades del sistema.

4.2. Justificación

Este proyecto surge de la necesidad de desarrollar una API robusta para la gestión de usuarios. Se utiliza el framework Flask con Python, combinando bases de datos no relacionales como Redis y bases de datos relacionales como MySQL. Además, se implementa Amazon Key Management Services (KMS) para asegurar una gestión segura de claves. La integración de estas tecnologías busca ofrecer una solución completa y moderna que optimice la eficiencia en el manejo de datos y garantice la seguridad en la gestión de claves, abordando así los requisitos esenciales del proyecto.

4.3. Alcance

El alcance de este proyecto se extiende al desarrollo de una API integral para la gestión de usuarios mediante el uso de Flask y Python. Se incluirá la implementación segura de claves a través de Amazon KMS. El sistema final, respaldado por esta API, posibilitará a los usuarios, ya sean administradores o usuarios regulares, acceder a perfiles y visualizar listas de usuarios. Además, se contemplará la capacidad de gestionar roles y permisos para los administradores, proporcionando una funcionalidad más completa y adaptada a las necesidades de la gestión de usuarios.

5. Objetivos

5.1. General

El objetivo general de este proyecto es desarrollar una API eficiente para la gestión de usuarios, utilizando estas tecnologías contemporáneas brindando una solución completa para la administración de cuentas de usuario y a la vez seguridad en la gestión de claves.

5.2. Específico

- Utilizar Redis como base de datos no relacional para el almacenamiento eficiente de datos de usuario.
- Integrar Amazon KMS para una gestión segura de claves, garantizando la seguridad de la información.
- Simplificar las tareas administrativas mediante procesos automatizados.

6. Marco Teórico

6.1. Flask Framework

Flask es un microframework web escrito en Python, reconocido por su enfoque minimalista y modularidad. Su sistema de enrutamiento simple permite asociar funciones con URLs específicas, facilitando la creación de endpoints personalizados. Utiliza el motor de plantillas Jinja2 para generar HTML dinámico, lo que simplifica la creación de vistas interactivas. Flask es altamente extensible, permitiendo la incorporación de extensiones para agregar funcionalidades específicas como autenticación y manejo de bases de datos. Su filosofía de desarrollo ágil facilita la creación rápida de prototipos y la iteración eficiente durante el desarrollo de aplicaciones web.

6.2. Redis

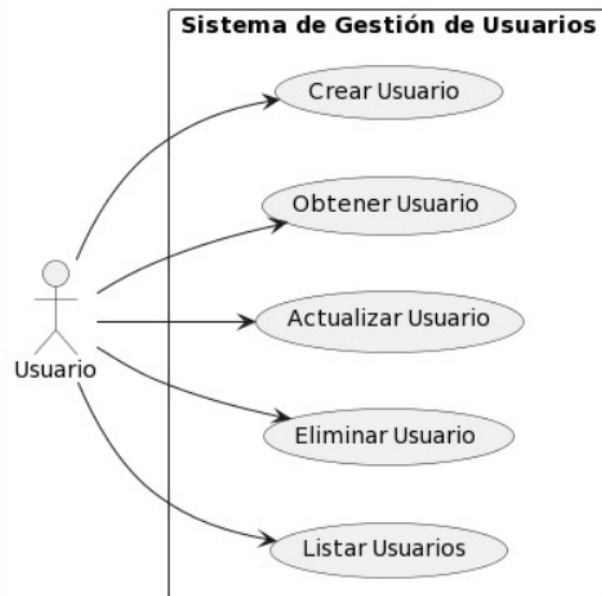
Redis es una base de datos en memoria que opera como un almacén de estructuras de datos clave-valor. Almacenando todos los datos en memoria, ofrece un acceso rápido y eficiente a la información. Admite diversas estructuras de datos como strings, hashes y listas, lo que facilita la implementación de diversas funcionalidades. Aunque es en memoria, Redis ofrece opciones de persistencia opcional para garantizar la durabilidad de los datos. Con operaciones atómicas, asegura la consistencia de los datos, y su alta escalabilidad lo hace apto tanto como base de datos principal como para almacenamiento en caché.

6.3. MySQL

MySQL es un sistema de gestión de bases de datos relacionales ampliamente utilizado. Siguiendo el modelo relacional, organiza datos en tablas relacionadas, lo que facilita la representación y manipulación de datos estructurados. Utiliza el lenguaje SQL para realizar operaciones como consultas, inserciones y actualizaciones en la base de datos. Ofrece mecanismos para garantizar la integridad referencial entre tablas, asegurando la consistencia de los datos. Adhiriéndose a los principios ACID, garantiza la fiabilidad de las transacciones. MySQL es escalable y eficiente en el manejo de grandes conjuntos de datos, proporcionando rendimiento robusto en aplicaciones con necesidades de almacenamiento de datos importantes.

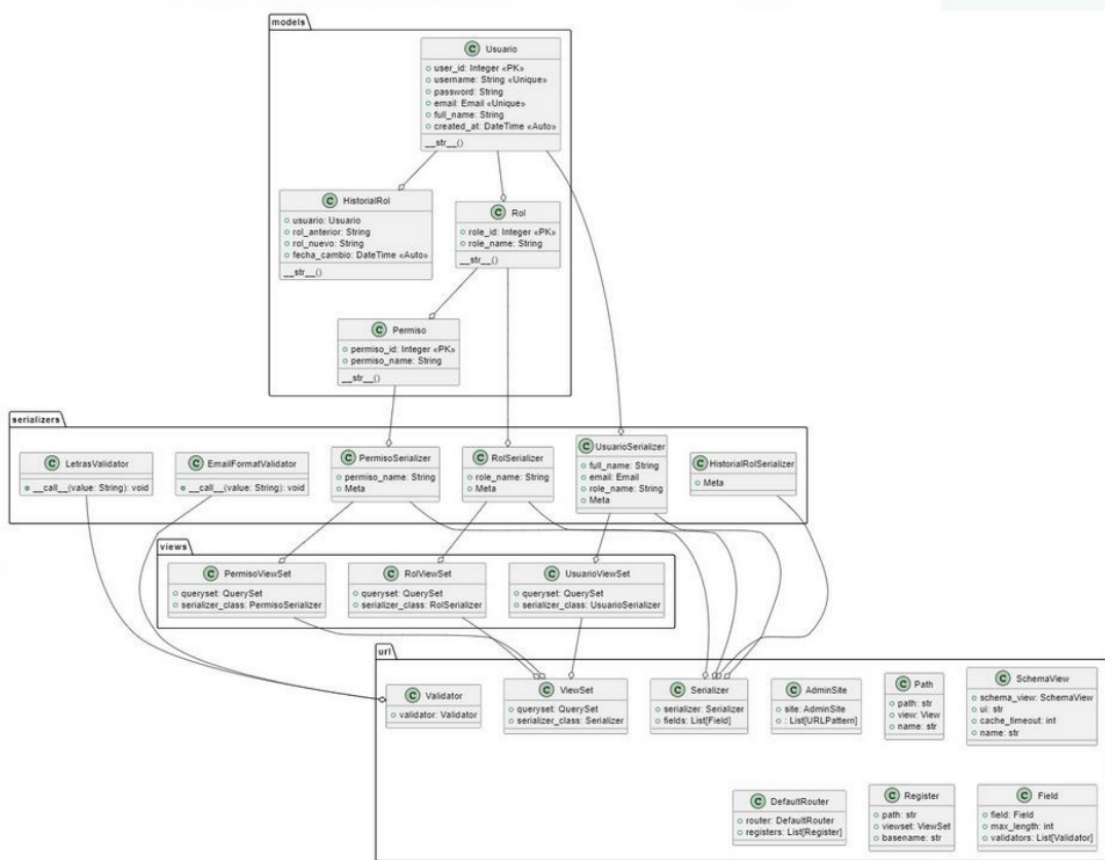
7. Desarrollo de la Solución

7.1. Casos de Uso API



7.2. Diagrama de Clases

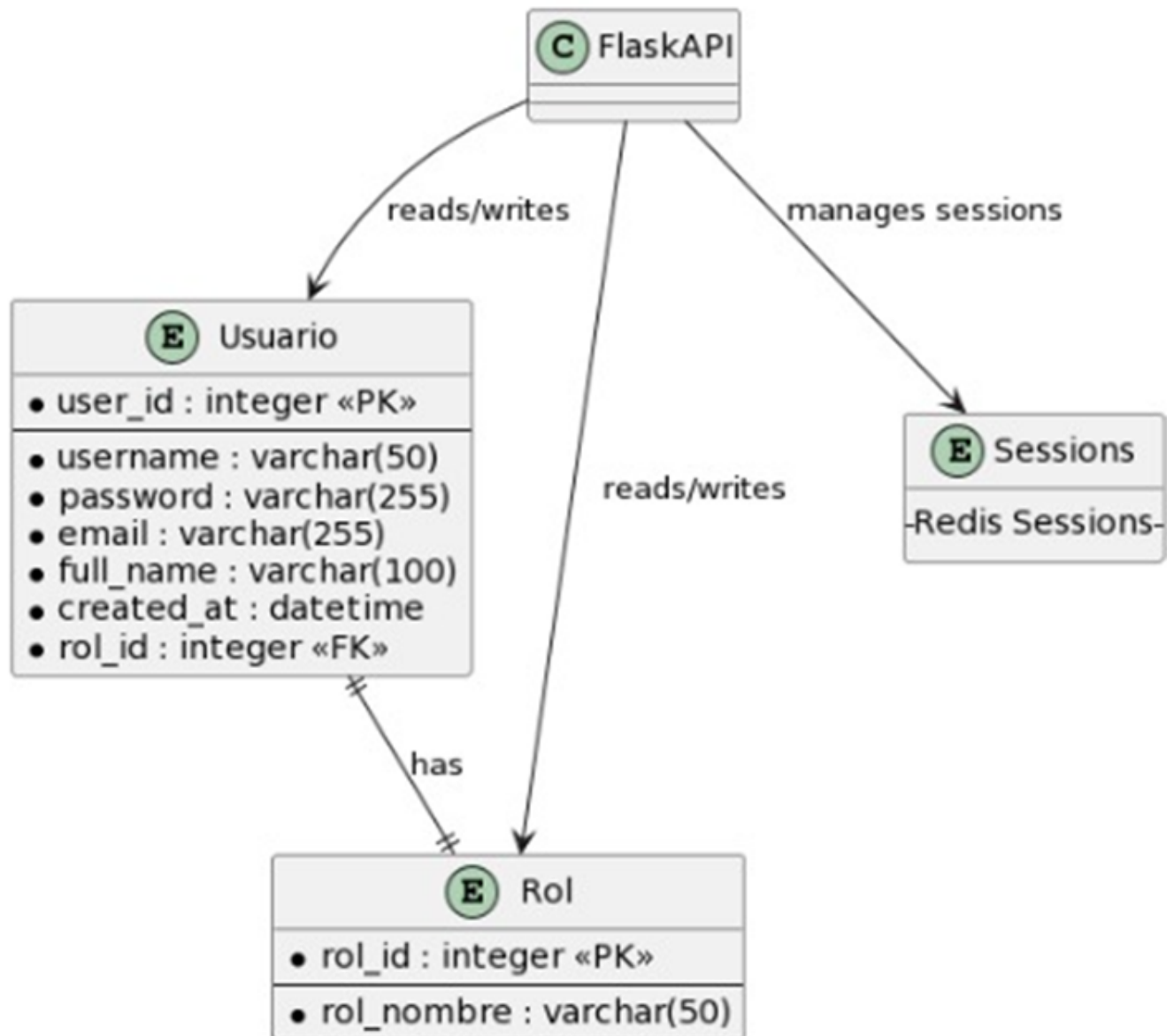
Unidad 1



Unidad2 - Redis

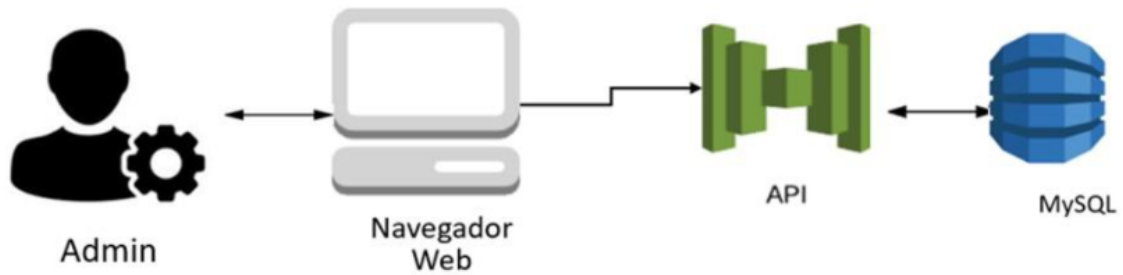


Unidad 3:

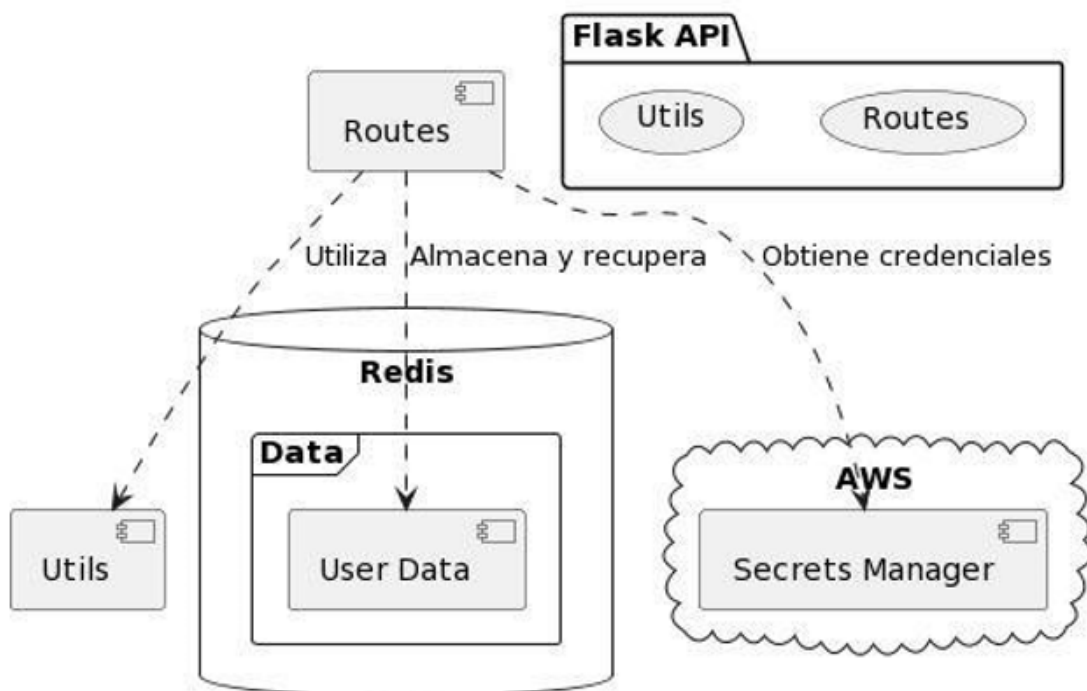


7.3. Diagrama de Arquitectura

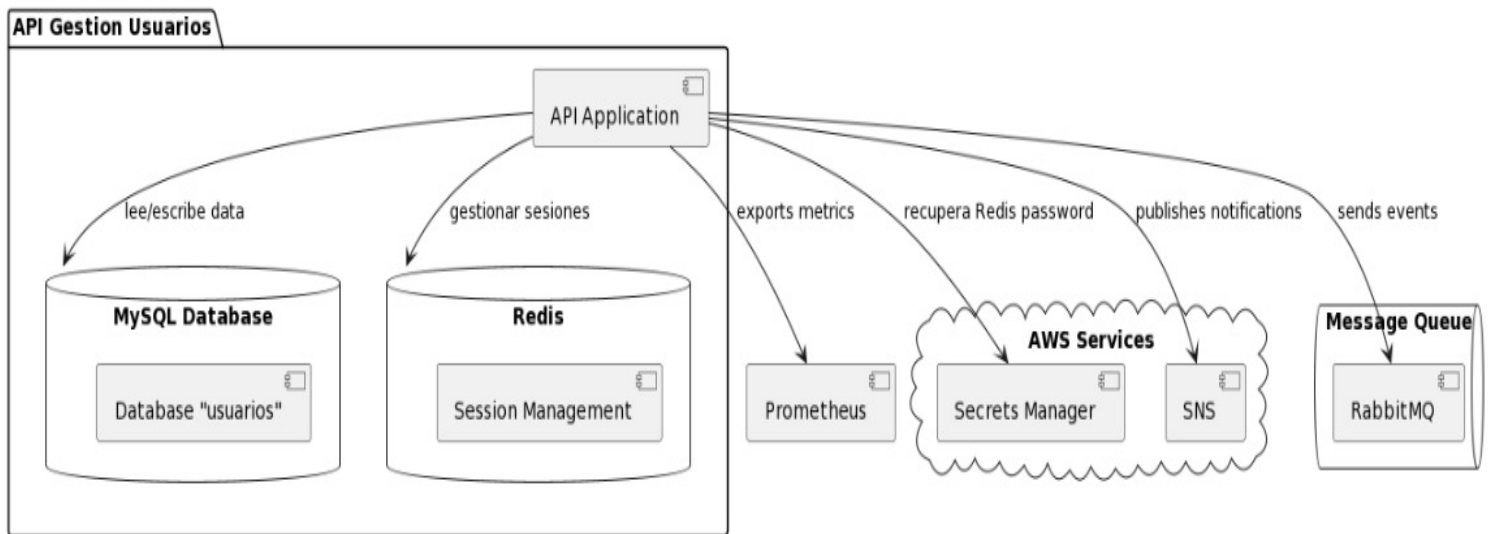
Unidad 1 - MySQL



Unidad 2 -Redis No relacional

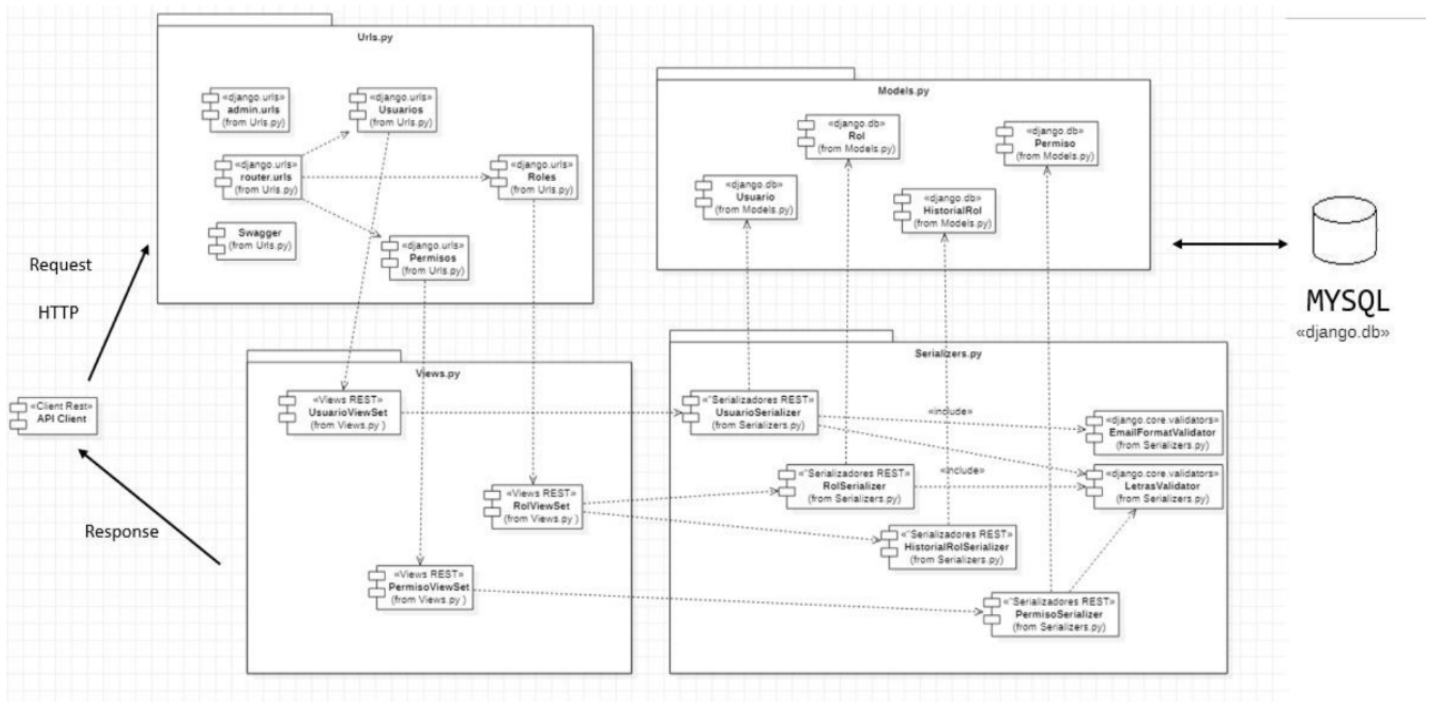


Unidad 3 - Integración :

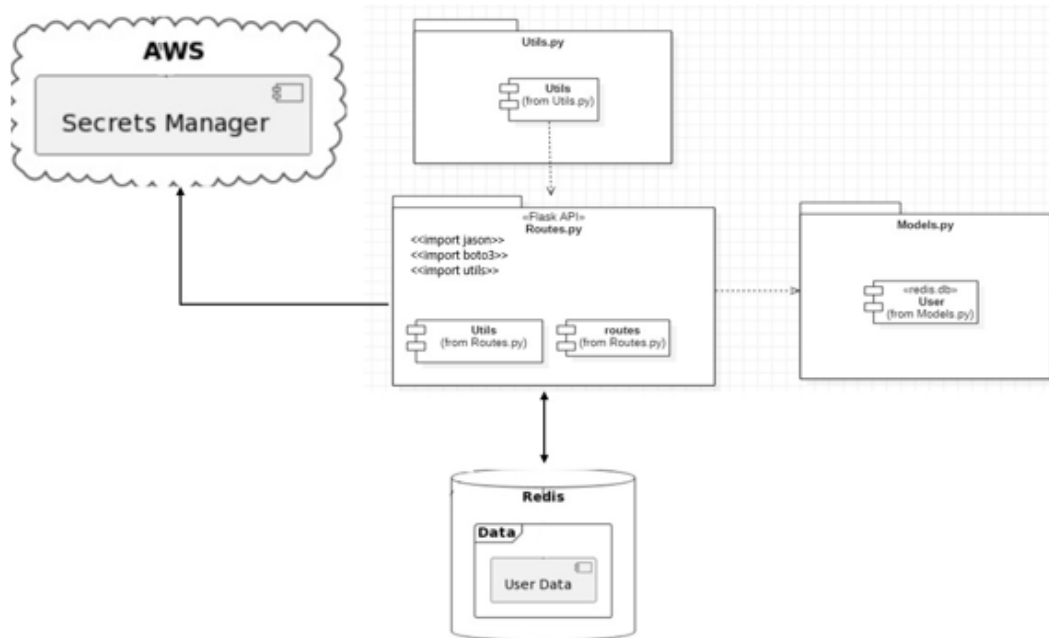


7.4. Diagrama de Componentes

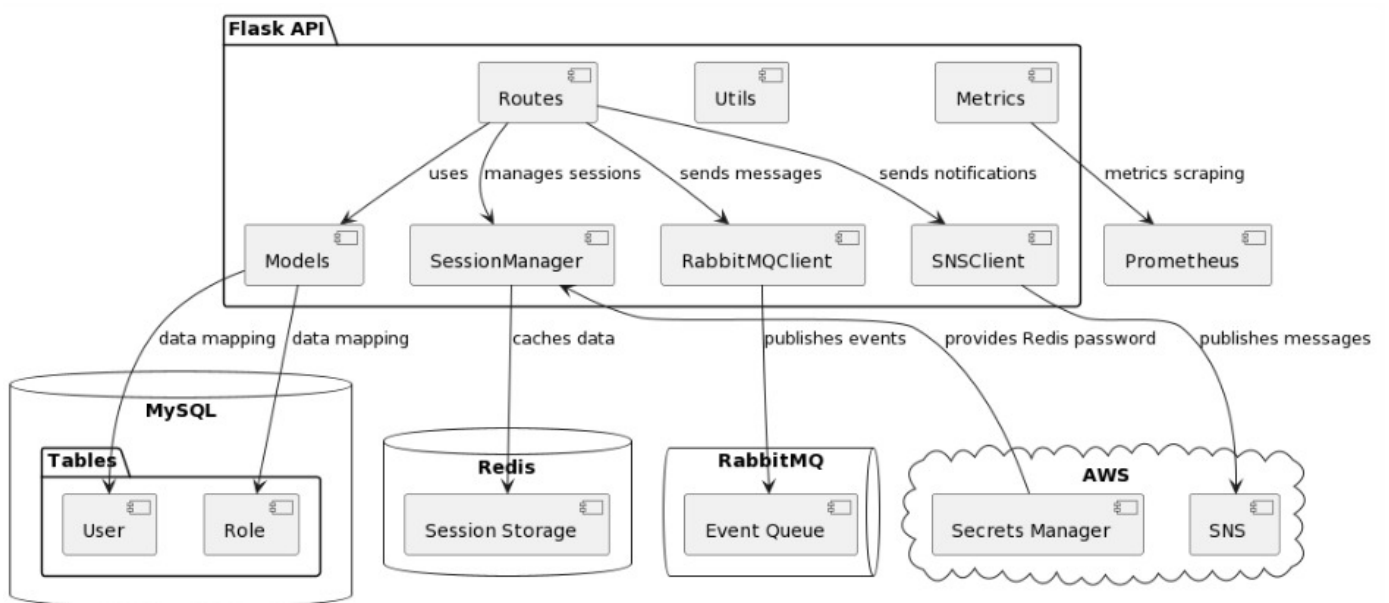
Unidad 1 - MySQL Relacional



Unidad 2 -Redis No relacional

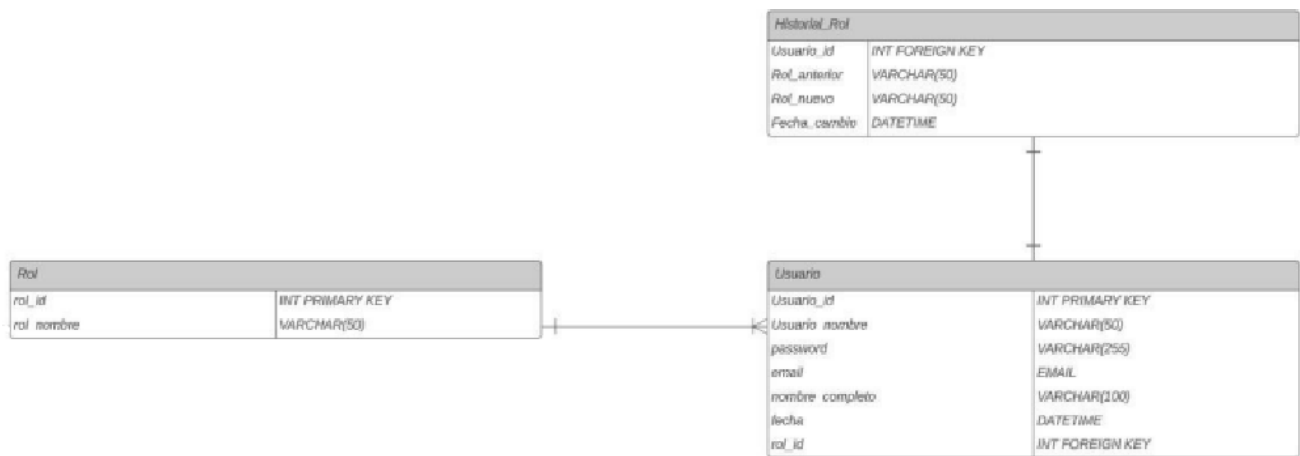


Unidad 3 - Integración

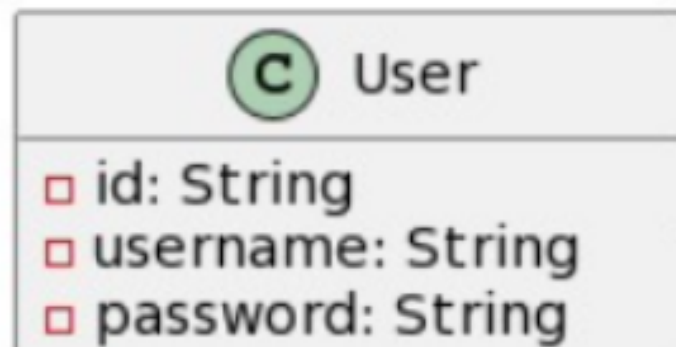


7.5. Diagrama de base de datos

7.5.1. MySQL



7.5.2. Redis



8. Conclusiones

En conclusión, la integración de Flask, Redis y MySQL en la creación de una API para la gestión de usuarios ofrece una solución equilibrada y eficiente.

Flask proporciona agilidad en el desarrollo, Redis garantiza un almacenamiento rápido y versátil para datos no relacionales, y MySQL aporta estructura para datos relacionales.

La inclusión de Amazon KMS refuerza la seguridad en la gestión de claves. En conjunto, estas tecnologías forman una API completa y moderna que cumple con los requisitos de rendimiento, seguridad y eficiencia en la gestión de usuarios.

9. Bibliografía

- <https://codigofacilito.com/articulos/orm-explicacion>
- <https://redis.io/docs/connect/clients/python/>
- <https://redis-py.readthedocs.io/en/stable/>
- <https://github.com/features/actions>
- <https://dev.to/msnmongare/title-building-a-food-api-with-django-and-mysql-5b12>
- <https://docs.docker.com/engine/api/>
- https://docs.aws.amazon.com/es_es/prescriptive-guidance/latest/patterns/run-an-asp-net-core-web-api-docker-container-on-an-amazon-ec2-linux-instance.html