**UNIVERSIDAD PRIVADA DE TACNA**

****

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**ANEXO – 03**

**“PATRONES DE DISEÑO DE**

**PRESENTACION O DE MICROSERVICIO”**

Curso: Patrones de software

Docente: Mag. Patrick Cuadros Quiroga

**Integrantes:**

**Romero Roque, Angelica Beatriz (2019063327)**

**Chambilla Maquera, Aroceli Noemi (2018060897)**

**Arhuata Turpo, Alex (2019063634)**

**Lima Ccosi, Jeackory Geilly (2019063326)**

**Rojas Bedregal Brian Erik (2018060904)**

**Vilca Condori Erlang Fernando (2019064024)**

**Tacna – Perú**

***2023***

INDICE

[1. INTRODUCCION 3](#_Toc139933761)

[2. API GATEWAY: 3](#_Toc139933762)

[3. MVVM: 3](#_Toc139933763)

[4. Conclusión 4](#_Toc139933764)

[5. Referencia Bibliográfica: 4](#_Toc139933765)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | ARR BR | ARR | PCQ | 08/07/2023 11/07/2023 | Versión Original |

**PATRONES DE DISEÑO DE PRESENTACIÓN O DE MICROSERVICIOS**

# INTRODUCCION

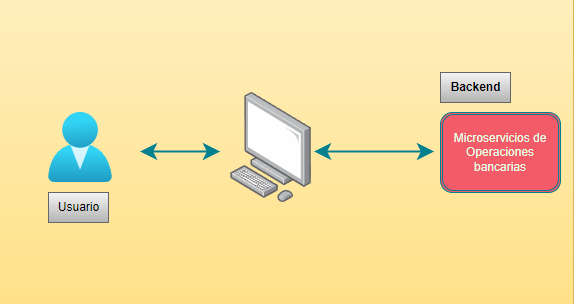
Una definición simple sería que la arquitectura de software es la guía o patrón que vamos a utilizar en el desarrollo de nuestro programa.

Muchas veces a esto no se le da excesiva importancia, pero definir al principio del desarrollo, la arquitectura con la que vamos a trabajar puede ahorrarnos muchos dolores de cabeza en el futuro.  
los patrones de diseño de microservicios se enfocan en la construcción de aplicaciones escalables y flexibles mediante la división de una aplicación monolítica en servicios independientes y autónomos.   
Estos servicios se comunican entre sí a través de interfaces bien definidas y pueden ser desarrollados, probados y desplegados de manera independiente, lo que facilita la escalabilidad y el mantenimiento del sistema.

Esto es así porque, a medida que nuestro software empiece a crecer y hacerse más complejo, el mantenimiento del código será mucho más sencillo si hemos desarrollado una buena estructura antes de empezar a programar.

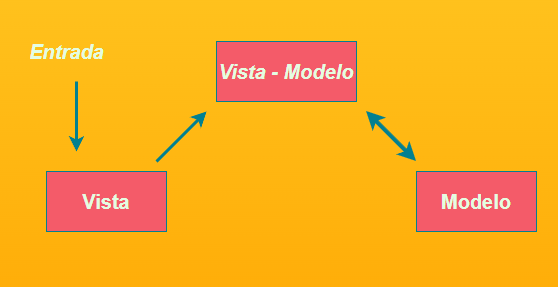
# API GATEWAY:

Una API Gateway es un gestor que acepta todas las llamadas API y luego actúa como un proxy inverso, recuperando recursos de las aplicaciones de back-end en nombre de la aplicación del cliente; maneja tareas relacionadas con los servicios de API, como autenticación de usuarios, limitación de velocidad y monitoreo.



# MVVM:

MVVM, por sus siglas en ingles Model View ViewModel, es un patrón de diseño que tiene por finalidad separar la parte de la interfaz del usuario (de ahí la V de View) de la parte de la lógica del negocio (de ahí la M de Model), logrando así que la parte visual sea totalmente independiente. El otro componente es el ViewModel que es la parte que va a interactuar como puente entre la Vista y el Modelo.



# Conclusión

En conclusión, una puerta de enlace de API es una herramienta de gestión de API que se encuentra entre el cliente y un conjunto de servicios de backend.

Funciona como un proxy inverso que acepta todas las llamadas a la interfaz de programación de la aplicación, agrega los servicios necesarios para cumplir con las solicitudes y devuelve el resultado adecuado.

Por otro lado, si queremos desarrollar un software que vaya a estar implementando cambios continuamente por diferentes personas, quizás sería más aconsejable irnos hacia el patrón MVVM, ya que al estar más desacoplado puede ser más sencillo el hacer cambios en una capa sin que afecten excesivamente al resto.

# Referencia Bibliográfica:

* ¿Cuál es la función de una puerta de enlace de API? (s/f). Redhat.com. Recuperado de https://www.redhat.com/es/topics/api/what-does-an-api-gateway-do
* Vista de Creación de una guía para la utilización de la arquitectura “clean architecture” en aplicaciones Android con lenguaje de programación Java, utilizando como modelo la implementación de un prototipo de monitoreo deportivo para un equipo de ultimate. (s/f). Utp.ac.pa. Recuperado de https://revistas.utp.ac.pa/index.php/ric/article/view/1750/html