

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE  
HUAMANGA**  
**FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL**



**ARQUITECTURA DE MICROSERVICIOS PARA LA ESCALABILIDAD DE  
UNA APLICACIÓN WEB**

**AUTOR:**

**Bach. Erland Renán Huamán Borda**

**ASESOR:**

**Msc. Ing. Hubner Janampa Patilla**

**AYACUCHO - PERÚ**

**2022**

---

# Índice

## CAPÍTULO I

### INTRODUCCIÓN

## CAPÍTULO II

### JUSTIFICACIÓN

## CAPÍTULO III

### OBJETIVOS

3.1. Objetivos generales . . . . .	7
------------------------------------	---

3.2. Objetivos específicos . . . . .	7
--------------------------------------	---

## CAPÍTULO IV

### MARCO TEÓRICO

4.1. Antecedentes . . . . .	8
-----------------------------	---

4.2. Arquitectura Monolítica . . . . .	8
--	---

4.2.1. Ventajas de una Arquitectura Monolítica . . . . .	8
--	---

4.2.2. Desventajas de una Arquitectura de Monolítica . . . . .	8
--	---

4.3. Arquitectura de Microservicios . . . . .	8
4.3.1. Ventajas de una Arquitectura de Microservicios . . . . .	10
4.3.2. Desventajas de una Arquitectura de Microservicios . . . . .	10
4.4. Despliegue Independiente . . . . .	10
4.5. Dominio de Negocio . . . . .	10
4.6. Despliegue . . . . .	11
4.7. Escalabilidad . . . . .	11
4.8. Complejidad . . . . .	11
4.9. Sistema Distribuido . . . . .	11
4.10. Ley de Conway . . . . .	11

## **CAPÍTULO V**

### **METODOLOGÍA**

## **CAPÍTULO VI**

### **RESULTADOS**

## **CAPÍTULO VII**

## **CONCLUSIONES**

## **CAPÍTULO VIII**

## **REFERENCIA BIBLIOGRÁFICA**

**Referencias**

**16**

# **CAPÍTULO I**

## **INTRODUCCIÓN**

introduccion

## **CAPÍTULO II**

### **JUSTIFICACIÓN**

justificacion

## **CAPÍTULO III**

### **OBJETIVOS**

#### **3.1. Objetivos generales**

Refactorizar la arquitectura de una aplicación monolítica a una arquitectura de microservicios, mediante la investigación y uso de paradigmas de arquitectura de software, con el propósito de mejorar la escalabilidad, disminuir los tiempos de respuesta y optimizar tiempos de CPU/minimizar el impacto en la aplicación original.

#### **3.2. Objetivos específicos**

1. Mejorar la escalabilidad de la aplicación para la mejor utilización de recursos computacionales.
2. Disminuir los tiempos de respuesta de funciones/módulos limitados procesamiento para que la aplicación sea más responsiva.
3. Minimizar el impacto en la aplicación original para evitar los tiempos de inactividad de la aplicación.
4. Integración de servicios

## **CAPÍTULO IV**

### **MARCO TEÓRICO**

#### **4.1. Antecedentes**

#### **4.2. Arquitectura Monolítica**

Según (Dmitry y Manfred, 2014) una aplicación monolítica puede ser

##### **4.2.1. Ventajas de una Arquitectura Monolítica**

##### **4.2.2. Desventajas de una Arquitectura de Monolítica**

#### **4.3. Arquitectura de Microservicios**

De acuerdo con (Dmitry y Manfred, 2014), un microservicio es un servicio ligero e independiente que realiza funciones únicas y colabora con otros servicios similares utilizando una interfaz bien definida. Una arquitectura basada en microservicios es un método para desarrollar una aplicación como un conjunto de servicios pequeños e independientes. Cada uno de los servicios está siendo ejecutado bajo un proceso independiente propio. Dichos servicios pueden comunicarse mediante mecanismos



ligeros (usualmente bajo HTTP). Servicios como estos pueden ser desplegados completamente independientes los unos de los otros. Los servicios pueden estar escritos en diferentes lenguajes de programación, diferentes paradigmas, usar sus propios modelos de datos, etcétera.

Por su parte (Alshuqayran, Ali, y Evans, 2016) definen la arquitectura de microservicios como el estilo de arquitectura que pone énfasis en dividir el sistema en servicios pequeños y ligeros que están contruidos para llevar a cabo una función de negocio de manera muy cohesiva.

Mientras que (Newman, 2019) indica que para definir una arquitectura de microservicios primero tenemos que definir a los microservicios. Los microservicios son un conjunto de servicios independientemente desplegables modelados alrededor de un dominio de negocio. Se comunican entre ellos mediante redes y como una elección de arquitectura ofrecen varias opciones para resolver los problemas que un equipo u organización pueden enfrentar. Por lo anterior indicado, se concluye que una arquitectura de microservicios es una arquitectura basada en múltiples microservicios trabajando en colaboración.

En resumen una arquitectura de microservicios está constituida por múltiples unidades que pueden ser desplegadas independientemente las unas de las otras. Este conjunto de servicios han sido desarrollados para solucionar problemas de un dominio de negocio.

#### **4.3.1. Ventajas de una Arquitectura de Microservicios**

#### **4.3.2. Desventajas de una Arquitectura de Microservicios**

### **4.4. Despliegue Independiente**

De acuerdo con (Newman, 2019) el despliegue independiente es la idea de que podemos hacer un cambio a un microservicio y desplegarlo a un ambiente de producción sin tener que utilizar ningún otro servicio.

### **4.5. Dominio de Negocio**

Según (Khononov, 2021) un dominio de negocio define el principal área de actividad de la empresa. En general es el servicio que la empresa provee a sus clientes. Una empresa puede operar en múltiples dominios de negocio.

## **4.6. Despliegue**

## **4.7. Escalabilidad**

## **4.8. Complejidad**

## **4.9. Sistema Distribuido**

## **4.10. Ley de Conway**

Existen patrones en las arquitecturas utilizadas por cada organización, estas observaciones las hizo (Conway, 1968): “Cualquier organización que diseñe un sistema producirá un diseño que copia la estructura de comunicación de dicha organización”

De acuerdo a estas observaciones (Newman, 2019) asevera que la arquitectura de tres capas (MVC) surge de la forma en la cual organizaciones de TI agrupaban a las personas: se creaban grupos separados de administradores de bases de datos, desarrolladores de infraestructura y desarrolladores de interfaces gráficas. Entonces concluye que si agrupamos a las personas con respecto a sus competencias principales, estas personas organizarán el software de manera similar.

Actualmente la forma más común de agrupar personas en organizaciones de TI es en

equipos multidisciplinarios armando equipos ágiles que tienen la responsabilidad de lanzar software de calidad lo más rápido posible.

## **CAPÍTULO V**

### **METODOLOGÍA**

## **CAPÍTULO VI**

### **RESULTADOS**

## **CAPÍTULO VII**

### **CONCLUSIONES**

## CAPÍTULO VIII

### REFERENCIA BIBLIOGRÁFICA

#### Referencias

- Alshuqayran, N., Ali, N., y Evans, R. (2016). A systematic mapping study in microservice architecture. En *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA)* (pp. 44–51). doi: doi:10.1109/SOCA.2016.15
- Conway, M. E. (1968). How do committees invent. *Datamation*, 14(4), 28–31.
- Dmitry, N., y Manfred, S.-S. (2014). On micro-services architecture. *International Journal of Open Information Technologies*, 2(9), 24–27.
- Khononov, V. (2021). *Learning domain-driven design*. O'Reilly Media, Inc.
- Newman, S. (2019). *Monolith to microservices: evolutionary patterns to transform your monolith*. O'Reilly Media.