



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Desarrollo de software basado en microservicios: un caso de estudio para evaluar sus ventajas e inconvenientes**

**TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

*Autor:* Víctor Alberto Iranzo Jiménez

*Tutor:* Patricio Orlando Letelier Torres

Curso 2017-2018



# Resum

???

**Paraules clau:** Microservices, Arquitecturas de software, ????????????????

---

# Resumen

???

**Palabras clave:** Microservicios, Arquitecturas de software, ????????????????

---

# Abstract

???

**Key words:** Microservices, Software Architecture, ????????????????

---



# Índice general

---

Índice general	V
Índice de figuras	VII
Índice de tablas	VII

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	1
1.2	Objetivos . . . . .	1
1.3	Estructura de la memoria . . . . .	2
<b>2</b>	<b>Estado del arte</b>	<b>3</b>
2.1	¿Qué son los microservicios? . . . . .	3
2.2	Ventajas del desarrollo de software basado en microservicios	4
2.3	Inconvenientes del desarrollo de software basado en microservicios . . . .	4
2.3.1	Los microservicios en la fase de análisis de requisitos . . . . .	4
2.3.2	Los microservicios en la fase de diseño . . . . .	4
2.3.3	Los microservicios en la fase de implementación . . . . .	4
2.3.4	Los microservicios en la fase de pruebas . . . . .	4
2.3.5	Los microservicios en la fase de despliegue . . . . .	4
2.3.6	Los microservicios en la fase de mantenimiento . . . . .	4
<b>3</b>	<b>Conclusiones</b>	<b>5</b>
	<b>Bibliografía</b>	<b>7</b>

---

Apéndice		
<b>A</b>	<b>Configuración del sistema</b>	<b>9</b>
A.1	Fase de inicialización . . . . .	9



## Índice de figuras

---

## Índice de tablas

---





---

---

# CAPÍTULO 1

## Introducción

---

### 1.1 Motivación

---

En la actualidad, no es necesario un alto grado de conocimientos en ingeniería del software para desarrollar una aplicación o sistema. Personas que no tienen estudios relacionados con la informática pueden producir código que, sin ser limpio y elegante, funciona. Desarrollar sistemas de calidad requiere de grandes conocimientos, pero minimiza los costes y aumenta la productividad de una organización. Se debe poner el foco en emplear una arquitectura de software que se adapte a nuestras necesidades. De lo contrario el futuro mantenimiento será más costoso y repercutirá en la confianza de los clientes y en la moral del equipo.[2]

Las arquitecturas basadas en microservicios son una tendencia actual que emerge asociada a conceptos clave como la integración continua, el desarrollo centrado en el dominio del problema o el despliegue en contenedores. En estas arquitecturas diferentes funcionalidades se encapsulan en servicios pequeños y autónomos que cooperan entre ellos. En términos de diseño, principios como el de Responsabilidad Única son más fáciles de conseguir y los desafíos de organización del código pueden abordarse de formas más diversas por la baja granularidad de la arquitectura.

### 1.2 Objetivos

---

El objetivo de este proyecto es validar con un caso de estudio las ventajas e inconvenientes de una arquitectura basada en microservicios frente a una arquitectura monolítica. Concretamente, los objetivos específicos son:

- Desarrollar una misma aplicación para la venta de productos y la gestión de pedidos siguiendo dos arquitecturas diferentes: una basada en microservicios y otra monolítica.
- Comparar el proceso de desarrollo de ambos sistemas a lo largo del ciclo de vida del software.
- Evaluar cómo se pueden llevar a cabo diferentes modificaciones durante el mantenimiento de ambas aplicaciones una vez se ha finalizado su implementación.
- Verificar que una arquitectura basada en microservicios facilita alcanzar los requisitos no funcionales de escalabilidad y tolerancia a fallos frente a una arquitectura monolítica.

## 1.3 Estructura de la memoria

---

---

## CAPÍTULO 2

# Estado del arte

---

### 2.1 ¿Qué son los microservicios?

---

Según Newman, los microservicios son servicios pequeños y autónomos que trabajan conjuntamente.[3] Podemos desglosar esta definición así:

- Un **servicio** es un conjunto de funcionalidades que se expone a los clientes para que las empleen con diferentes propósitos. La programación orientada a servicios es un paradigma que se aplica en las arquitecturas orientadas a servicios (SOA). El objetivo principal de SOA es desarrollar servicios que aporten valor al negocio y se adapten a los cambios en las necesidades de los clientes, de forma ágil y con el menor coste posible. Las arquitecturas orientadas a servicios no están asociadas a ninguna tecnología específica. En líneas generales, dividen un sistema en componentes que cambian por el mismo motivo y promueven la flexibilidad y los servicios compartidos frente a implementaciones específicas y óptimas. Son muchos los beneficios de estas arquitecturas, sin embargo existe una falta de consenso sobre cómo debe llevarse a cabo este tipo de arquitecturas en aspectos como los protocolos de comunicación a emplear o la granularidad de los servicios.[1] Los microservicios pueden entenderse como una aproximación específica de las arquitecturas SOA.
- Diseñar microservicios con el menor **tamaño** posible no debe ser el foco principal. En todo momento han de cumplirse los principios de cohesión y coherencia: el código relacionado debe agruparse conjuntamente porque será modificado por el mismo motivo. Una regla que puede ser aplicada según el autor Jon Eaves es establecer un tamaño para un microservicio tal que pueda ser completamente reescrito en 2 semanas.
- Los servicios han de ser lo menos acoplados posibles para garantizar la **autonomía** de cada uno. Cada microservicio es una entidad separada: cambian de forma independiente al resto y al hacerlo sus consumidores no necesitan ser modificados, a menos que se modifique el contrato entre ambas partes. Para lograrlo, lo más habitual es que cada servicio exponga una interfaz (API) y todas las comunicaciones se realicen mediante llamadas a través de la red.

## **2.2 Ventajas del desarrollo de software basado en microservicios**

---

## **2.3 Inconvenientes del desarrollo de software basado en microservicios**

---

**2.3.1. Los microservicios en la fase de análisis de requisitos**

**2.3.2. Los microservicios en la fase de diseño**

**2.3.3. Los microservicios en la fase de implementación**

**2.3.4. Los microservicios en la fase de pruebas**

**2.3.5. Los microservicios en la fase de despliegue**

**2.3.6. Los microservicios en la fase de mantenimiento**

---

---

## CAPÍTULO 3

# Conclusiones

---

????? ?????????????? ?????????????? ?????????????? ?????????????? ??????????????



# Bibliografía

---

- [1] Ali Arsanjani, Grady Booch, Toufic Boubez, Paul C. Brown, David Chappell, John DeVadoss, Thomas Erl, Nicolai Josuttis, Dirk Krafzig, Mark Little, Brian Loesgen, Anne Thomas Manes, Joe McKendrick, Steve Ross-Talbot, Stefan Tilkov, Clemens Utschig-Utschig, and Herbjörn Wilhelmsen. SOA Manifesto. *SOAManifesto*, 2009.
- [2] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 2017.
- [3] Sam Newman. *Building Microservices*. 2015.





---

## APÉNDICE A

# Configuración del sistema

---

???? ????????????? ????????????? ????????????? ????????????? ?????????????

### A.1 Fase de inicialización

---

???? ????????????? ????????????? ????????????? ????????????? ?????????????