

# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Adereso . . . . .	2
1.1.1. Clientes de Adereso . . . . .	2
1.2. El Problema . . . . .	4
1.3. Proyecto . . . . .	5
1.3.1. Objetivo del proyecto . . . . .	5
1.4. Solución . . . . .	5
1.5. Alternativas . . . . .	6
<b>2. Estado del Arte</b>	<b>8</b>
2.1. Arquitecturas de Software . . . . .	8
2.1.1. Servicio monolítico . . . . .	8
2.1.2. Microservicios . . . . .	9
2.1.3. Software serverless . . . . .	9
2.1.4. Levantamiento de servidores manejados . . . . .	10
2.2. Caso Netflix . . . . .	10
2.3. Tecnologías y servicios . . . . .	12
2.3.1. RabbitMQ . . . . .	12
2.3.2. Kafka . . . . .	13
2.3.3. Celery . . . . .	13
2.3.4. Servicios de Amazon . . . . .	14

2.3.5.	Servicios de Google . . . . .	16
2.3.6.	Servicios de monitoreo . . . . .	17
2.4.	Metodología de desarrollo . . . . .	18
2.4.1.	Testing . . . . .	18
2.4.2.	Coverage . . . . .	18
2.4.3.	Pull Requests . . . . .	19
2.4.4.	Integración Continua . . . . .	19
2.4.5.	Control de calidad . . . . .	19
<b>3.</b>	<b>Problema</b>	<b>20</b>
3.1.	Situación actual: servicio legacy . . . . .	21
3.1.1.	Arquitectura del Servicio . . . . .	21
3.1.2.	Problemas de escalabilidad . . . . .	22
3.1.3.	Problemas de mantención . . . . .	22
3.2.	Situación actual: prototipo Chaski . . . . .	23
3.2.1.	Arquitectura del Servicio . . . . .	23
3.2.2.	Uso actual del Chaski . . . . .	23
3.2.3.	Monitoreo . . . . .	24
3.2.4.	Errores observados . . . . .	25
3.2.5.	Casos reales . . . . .	25
3.3.	Características faltantes . . . . .	26
3.3.1.	Asegurar la persistencia de la información . . . . .	27
3.3.2.	Robustez del servicio . . . . .	27
3.3.3.	Logs y diagnóstico de problemas . . . . .	27
3.3.4.	Monitoreo de funcionamiento . . . . .	27
3.4.	Relevancia . . . . .	28
3.5.	Objetivos . . . . .	28
3.5.1.	Objetivos específicos . . . . .	28

3.5.2. Deseables . . . . .	29
<b>4. Solución</b>	<b>30</b>
4.1. Arquitectura del servicio . . . . .	30
4.2. Estructura de la solución . . . . .	31
4.2.1. Servicios usados . . . . .	32
4.2.2. Librerías escritas . . . . .	32
4.2.3. Descripción de las lambdas . . . . .	34
4.3. Sistema de reportes . . . . .	35
4.4. Sistema de Reintentos . . . . .	36
4.5. Ciclo de vida de un mensaje . . . . .	37
4.6. Despliegue de la Solucion . . . . .	40
4.6.1. Deployment fallido . . . . .	40
4.6.2. Solución implementada: Asegurar retrocompatibilidad de las lambdas	41
<b>5. Validación</b>	<b>43</b>
5.1. Proceso de Desarrollo . . . . .	43
5.1.1. Sobre el proceso de evaluación de pares . . . . .	43
5.1.2. Sobre el proceso de QA . . . . .	44
5.1.3. Cobertura e integración continua . . . . .	44
5.2. Pruebas realizadas en ambiente QA . . . . .	44
5.2.1. Explicación de la prueba . . . . .	45
5.2.2. Resultados de la prueba . . . . .	45
5.3. Resultados vistos en producción . . . . .	48
5.3.1. Mensajes procesados . . . . .	48
5.3.2. Manejo de errores . . . . .	50
5.3.3. Seguimiento de mensajes . . . . .	52
<b>6. Conclusiones</b>	<b>56</b>

6.1. Resumen del trabajo realizado . . . . .	56
6.2. Grado de cumplimiento de los objetivos . . . . .	57
6.3. Análisis de los resultados y el impacto del proyecto . . . . .	58
6.4. Aprendizajes del proceso de desarrollo . . . . .	58
6.4.1. Planificación ideal en retrospectiva . . . . .	59
6.5. Beneficios de las arquitecturas basadas en Cloud Computing . . . . .	60
6.5.1. Bajo costo del prototipado . . . . .	60
6.5.2. La clara definición de costos . . . . .	61
6.6. Ideas para el futuro de Chaski . . . . .	61
6.6.1. Mejorando el servicio . . . . .	61
6.6.2. Nuevos usos de Chaski . . . . .	61
<b>Bibliografía</b>	<b>67</b>