



RIDUNAJ
Repositorio Institucional
Digital UNAJ



Universidad Nacional
ARTURO JAURETCHE

Tesis de Grado

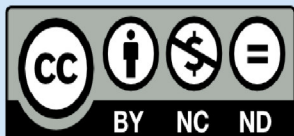
Blasi, Eduardo Ezequiel

Recolector de trazas

2023

Instituto: Ingeniería y Agronomía

Carrera: Ingeniería en Informática



Esta obra está bajo una Licencia Creative Commons.
Atribución – No comercial – Sin obra derivada 4.0
<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Documento descargado de RID - UNAJ Repositorio Institucional Digital de la Universidad Nacional Arturo Jauretche

Cita recomendada:

Blasi, E. E. (2023). *Recolector de trazas* [Práctica profesional supervisada de grado, Universidad Nacional Arturo Jauretche]. Disponible en RID - UNAJ Repositorio Institucional Digital UNAJ

<https://biblioteca.unaj.edu.ar/rid-unaj-repositorio-institucional-digital-unaj>

Universidad Nacional Arturo Jauretche

Instituto de Ingeniería y Agronomía

Ingeniería en Informática



Práctica Profesional Supervisada

Informe final

Recolector de trazas

Blasi Eduardo Ezequiel

Florencio Varela, noviembre de 2023

PRÁCTICA PROFESIONAL SUPERVISADA (PPS)

Recolector de trazas

Informe Programa de Actividades

DATOS DEL ESTUDIANTE

Apellido y Nombres: Blasi, Eduardo Ezequiel

DNI: 33982034

N.º de Legajo: 19851

Correo electrónico: profebiasi@gmail.com

Cantidad de materias aprobadas al comienzo de la PPS: 40

PPS enmarcada en el artículo 4 de la Resolución (CS) 103/16

DOCENTE SUPERVISOR

Apellido y Nombres: Dr. Ing. Gross, Patricio Martín

Correo electrónico: pgross@unaj.edu.ar

DOCENTE TUTOR DEL TALLER DE APOYO A LA PRODUCCIÓN DE TEXTOS ACADÉMICOS DE LA UNAJ

Apellido y Nombres: N/A

Correo electrónico: N/A

DATOS DE LA ORGANIZACIÓN DONDE SE REALIZA LA PPS

Nombre o Razón Social: Correo Andreani S.A.

Dirección: Av. Leandro N. Alem 639 P. 7 Dpto. 1 (1001) CABA

Teléfono: 0800-122-1112

Sector: Soluciones de trazabilidad

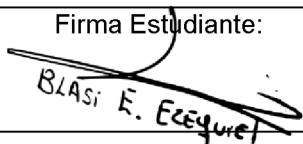
TUTOR DE LA ORGANIZACIONAL

Apellido y Nombres: Olivera Lucas

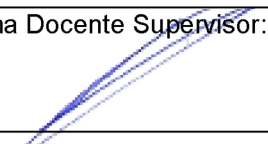
Correo electrónico: lolivera@andreani.com

FIRMA DEL COORDINADOR DE LA CARRERA

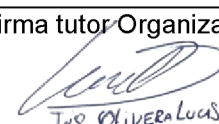
Firma Estudiante:


BLASI E. Ezequiel

Firma Docente Supervisor:



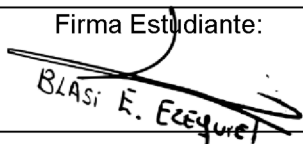
Firma tutor Organizacional:


ING. OLIVERA LUCAS

1. Resumen

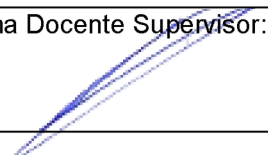
El propósito principal de este proyecto es abordar la necesidad de mejorar la eficiencia operativa y la experiencia del cliente mediante la migración de un sistema de seguimiento de envíos. La infraestructura actual se basa en tecnologías tales como IBM MQ, Redis, .NET Framework 4.5, SQL Server y Elasticsearch, lo que resulta en limitaciones en términos de escalabilidad y costos de licenciamiento. Este sistema de seguimiento no solo beneficiará a los clientes finales, sino que también a varios sectores internos de la empresa. Por lo tanto, se plantea una migración para superar estos desafíos y sentar las bases de un sistema más robusto y adaptable, que represente un paso fundamental hacia la mejora continua de la operatividad y la satisfacción del cliente. Esta migración no solo aborda las limitaciones actuales, sino que también establece las bases para un sistema más flexible, escalable y alineado con las últimas tecnologías disponibles en la empresa. El resultado final es un sistema que brinde una experiencia mejorada tanto para los clientes como para los diferentes sectores internos que dependen de la información de seguimiento de envíos.

Firma Estudiante:

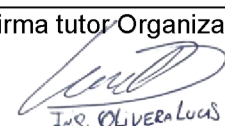


BLAS E. Echeverri

Firma Docente Supervisor:



Firma tutor Organizacional:

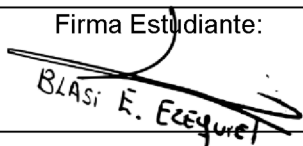


JOS. OLIVERA LUCAS

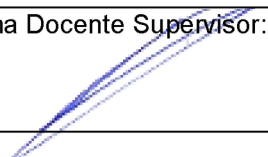
2. Abstract

The primary purpose of this project is to address the need to improve operational efficiency and customer experience through shipping tracking system migration. The current infrastructure is based on technologies such as IBM MQ, Redis, .NET Framework 4.5, SQL Server and ElasticSearch, resulting in limitations in terms of scalability and licensing costs. This tracking system will not only benefit end customers, but also various internal sectors of the company. Therefore, a migration was proposed to overcome these challenges and lay the foundations for a more robust and adaptable system, which represents a fundamental step towards continuous improvement of operation and customer satisfaction. This migration not only addresses current limitations, but also establishes the foundation for a more flexible, scalable system aligned with the latest technologies available in the company. The end result is expected to be a system that provides an improved experience for both customers and the different internal sectors that depend on shipment tracking information.

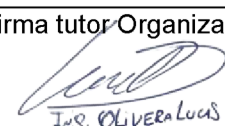
Firma Estudiante:


BLAS E. Echeverri

Firma Docente Supervisor:



Firma tutor Organizacional:


Tutor OLIVERA LUCAS

3. Dedicatorias y agradecimientos

Quisiera expresar mi profundo agradecimiento a todas las personas que contribuyeron de manera significativa a la realización de esta PPS, a mi tutor Dr. Ing. Gross Patricio por su orientación experta y dedicación incansable. Sus conocimientos y dirección han sido fundamentales para dar forma a este trabajo y para mi crecimiento académico con el desarrollo de la Práctica Profesional Supervisada (PPS) y al Coordinador de la carrera Ingeniería en Informática Dr. Ing. Martín Morales.

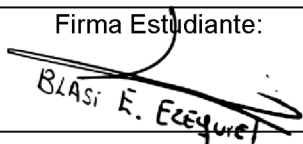
Agradezco sinceramente a los profesionales de la empresa Andreani que participaron en este proyecto. Su colaboración, información valiosa y disposición para compartir sus experiencias fueron fundamentales para la comprensión profunda del tema.

Mi gratitud se extiende a mis compañeros de clase y amigos que brindaron su apoyo moral y motivación constante. Sus ideas y perspectivas enriquecieron mi enfoque hacia este proyecto.

Finalmente, a mi familia y seres queridos, les agradezco por su paciencia, comprensión y aliento constante. Su respaldo incondicional ha sido mi mayor fortaleza.

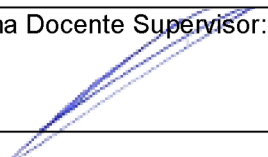
Este trabajo no habría sido posible sin el aporte de cada uno de ustedes. Gracias por ser parte de este viaje académico.

Firma Estudiante:

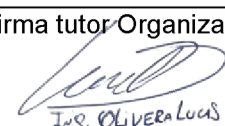


BLAS E. Eceyurel

Firma Docente Supervisor:



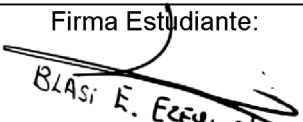
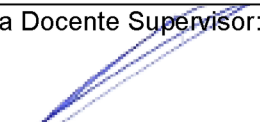
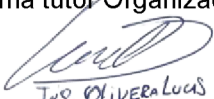
Firma tutor Organizacional:



JOS. OLIVERA LUCAS

4.ÍNDICE

Introducción.....	7
Tecnologías utilizadas.....	12
Apache Kafka	12
.NET CORE 6	16
ElasticSearch	18
Open Shift	21
Metodología.....	23
Scrum Máster	23
Arquitectura	26
Principios solid	28
SonarQube	30
Multi-Threaded	31
Funcionamiento previo del sistema de seguimiento de envíos.....	34
Funcionamiento actual del sistema de seguimiento de envíos.....	41
Circuit Breaker	42
Using Script ElasticSearch	43
Escalabilidad en despliegue	44
Orquestación con Kubernetes	44
Conclusiones.....	49
Reflexión acerca del desarrollo	49
Referencias bibliográficas.....	52

<p>Firma Estudiante:</p>  <p>BLAS E. ECEGUET</p>	<p>Firma Docente Supervisor:</p> 	<p>Firma tutor Organizacional:</p>  <p>JOS. OLIVERA LUCAS</p>
---	--	--

INTRODUCCIÓN

El presente informe detalla el proceso y los resultados de la Práctica Profesional Supervisada (PPS) realizada en la empresa Correo Andreani SA. El objetivo principal de este proyecto es abordar la necesidad de mejorar la eficiencia operativa y la experiencia del cliente mediante la migración del sistema de seguimiento de envíos. Actualmente, el sistema utiliza una arquitectura basada en IBM MQ, Redis, 2 servicio de .NET Framework 4.5, una base de datos SQL Server como almacenamiento y ElasticSearch donde se publica un documento con los eventos del envío. Sin embargo, esta infraestructura presenta limitaciones en términos de tecnología obsoleta, costos de licenciamiento y escalabilidad.

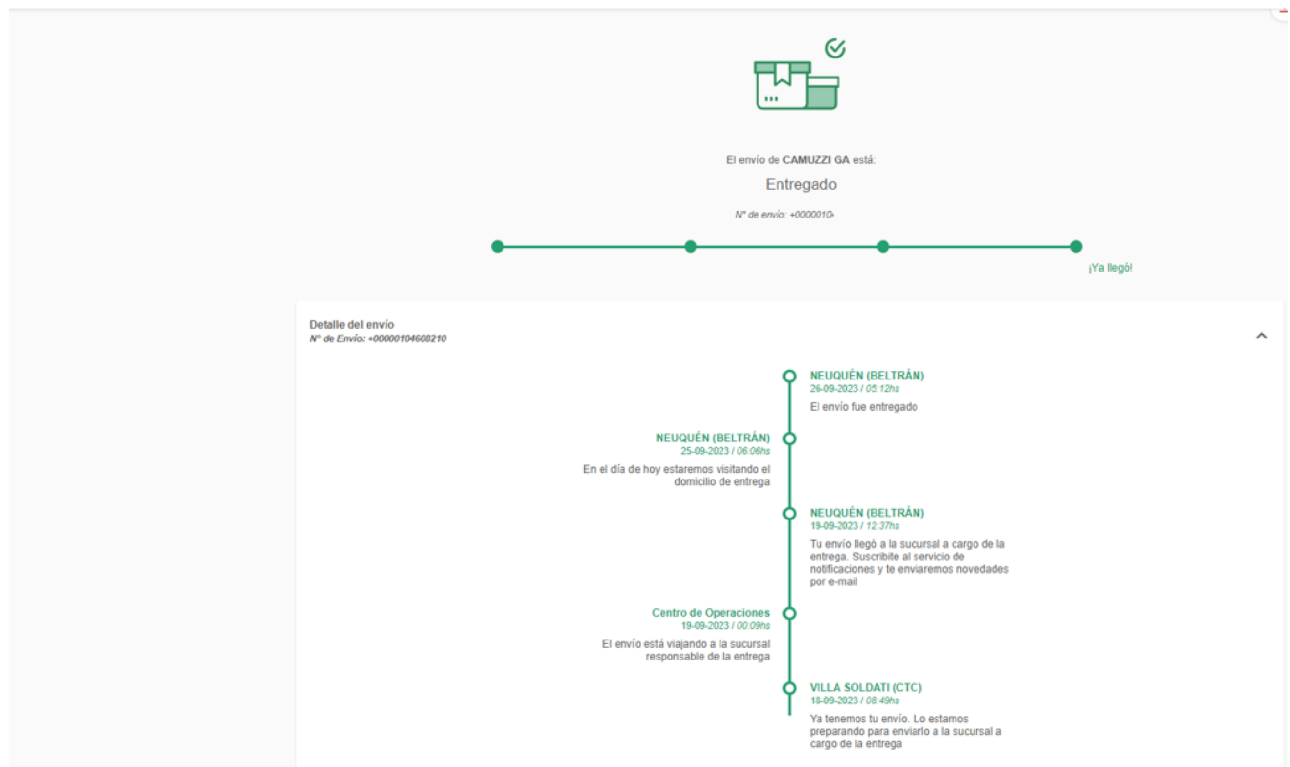


FIGURA 1 - TRAZAS DE UN ENVÍO, QUE SE REFLEJA EN LA WEB DE ANDREANI

Firma Estudiante: 	Firma Docente Supervisor: 	Firma tutor Organizacional:
-----------------------	-------------------------------	---------------------------------

A través de una migración planificada en dos fases, se busca modernizar la tecnología y alinearse con las últimas tecnologías disponibles en la empresa.

El sistema de seguimiento de envíos desempeña un papel crucial en la operatividad y la satisfacción del cliente en Andreani SA. No obstante, la infraestructura existente presentaba obstáculos significativos, como la versión obsoleta de .NET Framework 4.5, los costos elevados de licenciamiento asociados a MQ (IBM) y la falta de escalabilidad para afrontar futuras demandas.

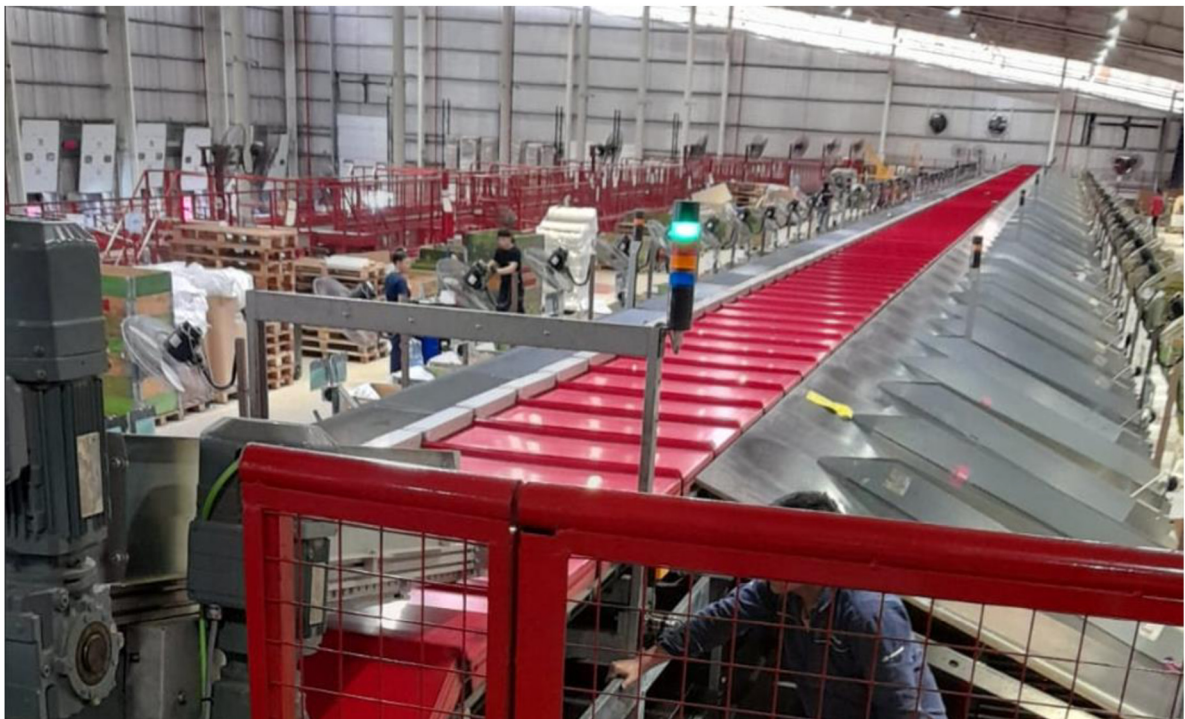
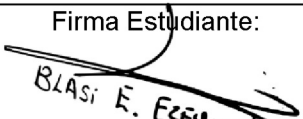
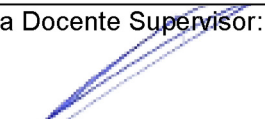
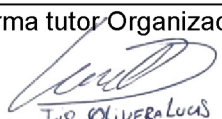


FIGURA 2 - PROCESO OPERATIVO EN EL CENTRO DE TRANSFERENCIA INTELIGENTE

Es relevante destacar que el sistema de seguimiento no solo es utilizado por los clientes finales, sino que también diversos sectores internos de la empresa se benefician de estos eventos para llevar a cabo diferentes procesos.

Por lo tanto, se planteó una migración como solución para superar estos desafíos y sentar las bases de un sistema más robusto y adaptable. La migración se estructuró en dos fases secuenciales: en la primera fase que es la que se va a exponer en este

Firma Estudiante:  BLAS E. Eceguet	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

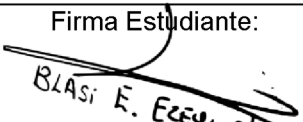
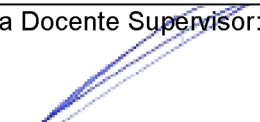
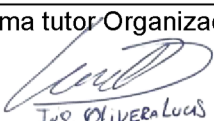
trabajo, se propuso el reemplazo del servicio de mensajería MQ (IBM) por Kafka, una plataforma de mensajería y transmisión de eventos más moderna y escalable.

Se tomó la decisión de eliminar las capas intermedias de Redis y SQL Server, lo que permitirá un procesamiento más directo y rápido de los eventos. En lugar de pasar por Redis y una base de datos SQL para el almacenamiento de espera y el procesamiento intermedio, el evento se insertará directamente en ElasticSearch, el motor de búsqueda y análisis. Este enfoque simplificado y directo agiliza aún más el procesamiento de eventos y reduce la latencia, permitiendo que la información llegue a los usuarios finales de manera más rápida y eficiente. Además, se llevará a cabo la actualización del entorno de desarrollo a .NET CORE 6, aprovechando sus ventajas en términos de rendimiento y funcionalidades avanzadas. Se planea implementar un enfoque que permita un mayor procesamiento en menos tiempo mediante el trabajo con múltiples instancias y el uso de hilos. Esta estrategia se traducirá en una mejora significativa de la eficiencia del sistema.

Esta actualización no solo mejora la agilidad del desarrollo, sino que también permite un procesamiento de eventos más eficiente y rápido.

En la segunda fase, se prevé el desarrollo de un frontend interactivo que permitirá a los usuarios consultar en tiempo real los datos almacenados en ElasticSearch.

Esta modernización del frontend es esencial para proporcionar una experiencia más ágil y accesible a los clientes y a los sectores internos de la empresa. Es importante mencionar que el frontend existente se encuentra desarrollado en PHP, y se tiene previsto realizar una actualización tecnológica hacia React, una de las tecnologías más avanzadas y versátiles en el desarrollo de interfaces de usuario. Este cambio permite una mayor interactividad, una respuesta más rápida y una mayor capacidad de personalización, mejorando significativamente la experiencia del usuario y permitiendo a los diferentes sectores internos obtener información valiosa para sus operaciones de manera más efectiva. Al unificar la infraestructura tecnológica en todas las etapas del

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

proceso, desde el backend hasta el frontend, se logra una mayor coherencia y eficiencia en todo el sistema de seguimiento de envíos de Andreani SA.

El presente trabajo aborda la implementación y evolución de un sistema de seguimiento de envíos, focalizándose en la integración de diversas tecnologías y en la mejora continua de su arquitectura y funcionalidades. Para comprender en su totalidad el alcance y la significancia de este proyecto, es esencial que el lector tenga conocimientos previos sobre los conceptos clave que sustentan esta investigación.

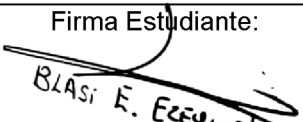
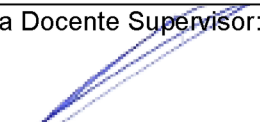
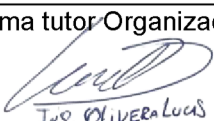
En los primeros capítulos, se proporcionará una sólida base conceptual, explicando las tecnologías fundamentales utilizadas en el desarrollo del sistema. Desde las capacidades de procesamiento de eventos en tiempo real de Apache Kafka, hasta la implementación de servicios escalables en .NET Core 6, y la indexación y búsqueda eficientes en Elasticsearch, cada tecnología desempeña un papel crucial en el funcionamiento integral del sistema.

Posteriormente, se detallará la metodología adoptada para la gestión y desarrollo del proyecto, destacando la aplicación de principios ágiles, la estructura de trabajo bajo el marco Scrum y la arquitectura basada en principios SOLID.

Además, se explorará el funcionamiento previo del sistema, identificando sus limitaciones y áreas de mejora. A continuación, se presentará el funcionamiento actualizado, enfocándose en las modificaciones tecnológicas implementadas y su impacto en la eficiencia y escalabilidad del sistema.

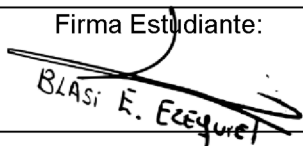
Cada capítulo se desarrollará de manera lógica, guiando al lector desde los conceptos fundamentales hasta la comprensión detallada de la arquitectura y funcionamiento del sistema de seguimiento de envíos. Al concluir, se proporcionará una reflexión crítica sobre el proceso de desarrollo y las lecciones aprendidas.

La combinación de tecnologías avanzadas, metodologías ágiles y la evolución del sistema garantiza un análisis integral y una visión completa de este proyecto. A través de estas páginas, el lector será guiado en un viaje que revelará no solo las

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

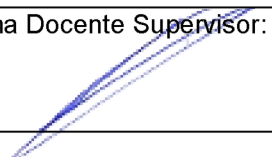
complejidades técnicas, sino también la aplicación efectiva de estrategias para mejorar la calidad del software y la eficiencia operativa.

Firma Estudiante:

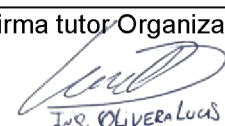


BLAS E. ECEGUET

Firma Docente Supervisor:



Firma tutor Organizacional:



JOS. OLIVERA LUCAS

TECNOLOGÍAS UTILIZADAS

APACHE KAFKA

El comienzo del ciclo es con Apache Kafka que es una plataforma de mensajería y transmisión de datos que garantiza una transmisión fluida y un procesamiento rápido y eficiente de los flujos de datos. Existe un software que facilita mucho ambas cosas: la aplicación de mensajería y transmisión de Apache Kafka. Esta solución de código abierto, desarrollada como servicio de cola de mensajes para la plataforma de LinkedIn en sus inicios, ofrece en la actualidad un completo paquete para el almacenamiento, la transmisión y el procesamiento de datos.

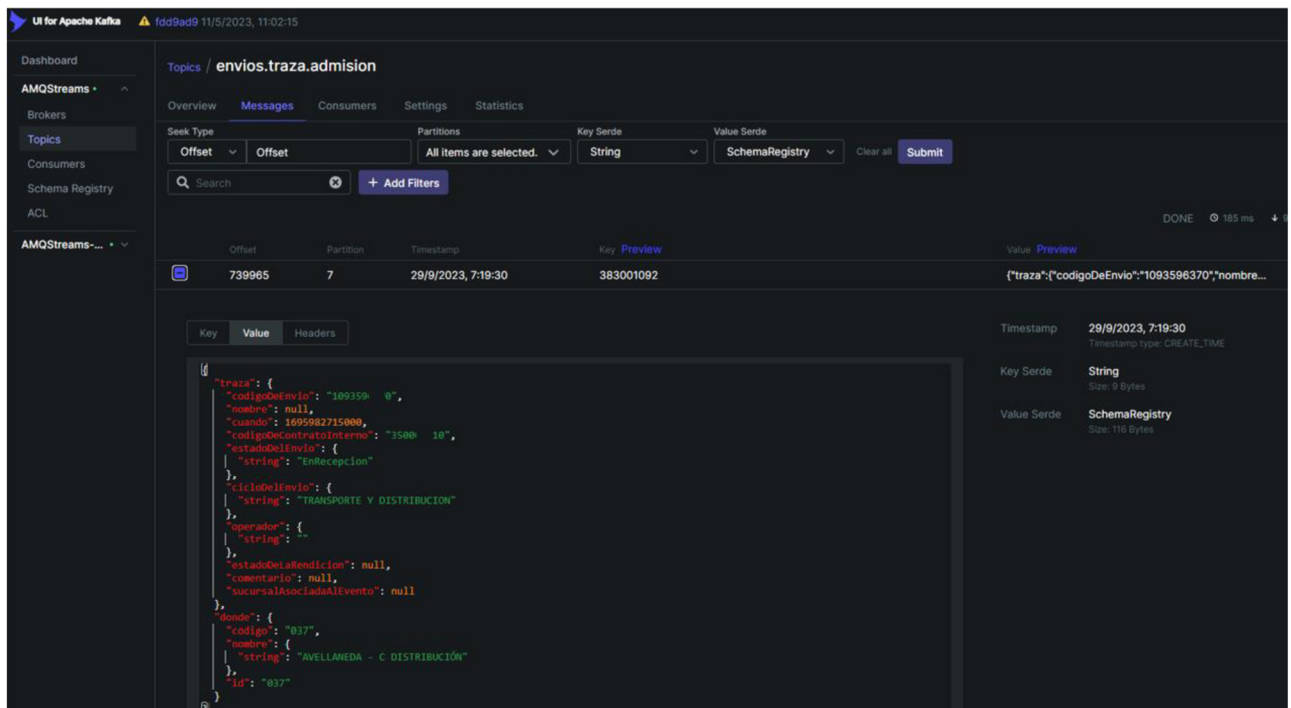
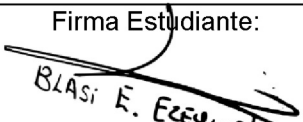
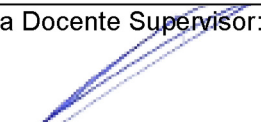
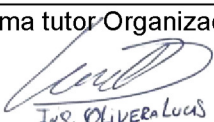


FIGURA 3 - MENSAJE DE EJEMPLO EN TÓPICO DE KAFKA

El software Apache Kafka es una aplicación multiplataforma de código abierto

Referencias [01] – Apache Kafka

Firma Estudiante: 	Firma Docente Supervisor: 	Firma tutor Organizacional: 
--	--	--

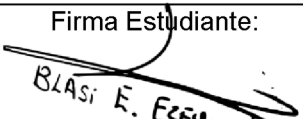
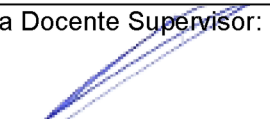
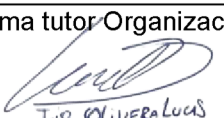
desarrollada por Apache Software Foundation y especializada en el procesamiento de flujos de datos.

Ante todo, Apache Kafka ha sido diseñado para optimizar la transmisión y el procesamiento de los flujos de datos que se intercambian entre la fuente y el receptor por conexión directa. Kafka actúa como una entidad de mensajería entre el emisor y el receptor, solventando los inconvenientes más habituales que conlleva este tipo de conexión.

Uno de los problemas más típicos que soluciona la plataforma de Apache es la imposibilidad de almacenar datos o mensajes en caché si el destinatario no está disponible (por ejemplo, si hay fallos en la red). Además, una cola de Kafka bien configurada evita que el receptor se vea saturado por el emisor, como suele pasar cuando la información enviada por conexión directa va más rápido de lo que puede recibirse y procesarse. El software de Kafka también responde bien cuando al receptor le llega el mensaje, pero se bloquea en el momento de procesarlo: con Apache Kafka, el emisor recibe un aviso de error que le informa del fallo, en lugar de asumir que se ha procesado el mensaje.

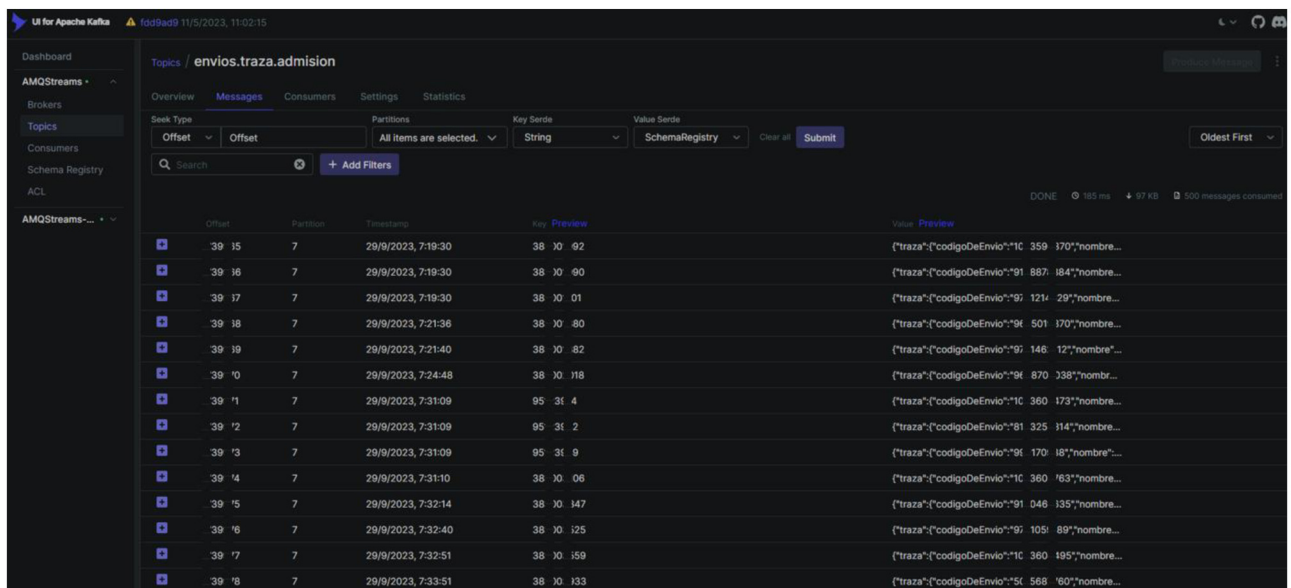
A diferencia de los meros servicios de colas de mensajes, como las bases de datos, el software de Apache Kafka tolera los errores, lo que le permite procesar los mensajes o datos de forma continua. Esta característica de Apache Kafka, en combinación con su alta escalabilidad y su capacidad para distribuir la información a un número ilimitado de sistemas (registro de transacciones distribuidas), lo convierte en una solución ideal para los servicios que deban asegurar un procesamiento y almacenamiento rápido de datos, así como una gran disponibilidad.

Apache Kafka se ejecuta como un clúster (red de ordenadores) en uno o más servidores que pueden encontrarse en centros de datos diferentes. Los nodos o puntos de intersección del clúster, denominados brókeres, almacenan los flujos de datos entrantes, clasificándolos en los llamados tópicos. Los datos se dividen en particiones y se replican y distribuyen en el clúster, recibiendo un sello de tiempo.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
---	--	--

De esta manera, la plataforma de transmisión asegura una gran disponibilidad y un acceso de lectura rápido. Apache Kafka distingue los temas entre “normal topics” y “compacted topics”. Los mensajes de normal topics pueden borrarse una vez se ha excedido el período o el límite de almacenamiento, mientras que las entradas de compacted topics no están sujetas a limitaciones de tiempo ni espacio.

Cada bróker de un clúster de Kafka puede alojar uno o varios tópicos, que a su vez pueden dividirse en distintas particiones y ponerse a disposición de un número ilimitado de consumidores.

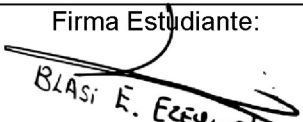
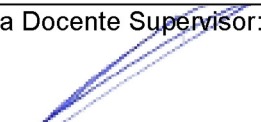



The screenshot shows the Apache Kafka UI for the topic 'envios.traza.admision'. The interface includes a sidebar with navigation options like Dashboard, Brokers, Topics, Consumers, Schema Registry, and ACL. The main area displays a list of messages with columns for Offset, Partition, Timestamp, Key, and Value. The messages are sorted by offset, showing a sequence of records with timestamps ranging from 2023-09-29 7:19:30 to 7:33:51. Each message contains a JSON payload with fields like 'codigoDeEnvio', 'nombre', and 'telefono'.

Offset	Partition	Timestamp	Key	Value
39 15	7	29/9/2023, 7:19:30	38 30 92	{\"traza\":{\"codigoDeEnvio\":\"1C 359 370\",\"nombre...\"}
39 16	7	29/9/2023, 7:19:30	38 30 90	{\"traza\":{\"codigoDeEnvio\":\"91 887 184\",\"nombre...\"}
39 17	7	29/9/2023, 7:19:30	38 30 01	{\"traza\":{\"codigoDeEnvio\":\"92 121 29\",\"nombre...\"}
39 18	7	29/9/2023, 7:21:36	38 30 80	{\"traza\":{\"codigoDeEnvio\":\"96 501 370\",\"nombre...\"}
39 19	7	29/9/2023, 7:21:40	38 30 82	{\"traza\":{\"codigoDeEnvio\":\"92 146 12\",\"nombre...\"}
39 10	7	29/9/2023, 7:24:48	38 30 118	{\"traza\":{\"codigoDeEnvio\":\"94 870 338\",\"nombr...\"}
39 11	7	29/9/2023, 7:31:09	95 31 4	{\"traza\":{\"codigoDeEnvio\":\"1C 360 173\",\"nombre...\"}
39 12	7	29/9/2023, 7:31:09	95 31 2	{\"traza\":{\"codigoDeEnvio\":\"81 325 114\",\"nombre...\"}
39 13	7	29/9/2023, 7:31:09	95 31 9	{\"traza\":{\"codigoDeEnvio\":\"96 170 18\",\"nombre...\"}
39 14	7	29/9/2023, 7:31:10	38 30 06	{\"traza\":{\"codigoDeEnvio\":\"1C 360 163\",\"nombre...\"}
39 15	7	29/9/2023, 7:32:14	38 30 147	{\"traza\":{\"codigoDeEnvio\":\"91 046 135\",\"nombre...\"}
39 16	7	29/9/2023, 7:32:40	38 30 125	{\"traza\":{\"codigoDeEnvio\":\"92 105 89\",\"nombre...\"}
39 17	7	29/9/2023, 7:32:51	38 30 159	{\"traza\":{\"codigoDeEnvio\":\"1C 360 195\",\"nombre...\"}
39 18	7	29/9/2023, 7:33:51	38 30 133	{\"traza\":{\"codigoDeEnvio\":\"5C 568 160\",\"nombre...\"}

FIGURA 4 - TÓPICO DE KAFKA CON EL FLUJO DE MENSAJES

Las aplicaciones que escriben datos en los clústeres de Kafka se denominan productores, mientras que las aplicaciones que leen los datos de un clúster se llaman consumidores. El componente central al que acceden los productores y consumidores cuando procesan secuencias de datos es una biblioteca de Java llamada Kafka Streams. La característica de escritura transaccional garantiza que los mensajes se transmitan solo una vez (sin duplicación), lo que también se conoce como “exactly-once delivery” (o “entrega una sola vez”).

<p>Firma Estudiante:</p>  <p>BLAS E. ECEGUET</p>	<p>Firma Docente Supervisor:</p> 	<p>Firma tutor Organizacional:</p>  <p>TOR. OLIVERA LUCAS</p>
---	--	--

Apache Kafka ofrece cinco interfaces básicas distintas, mediante las cuales las aplicaciones pueden acceder al software:

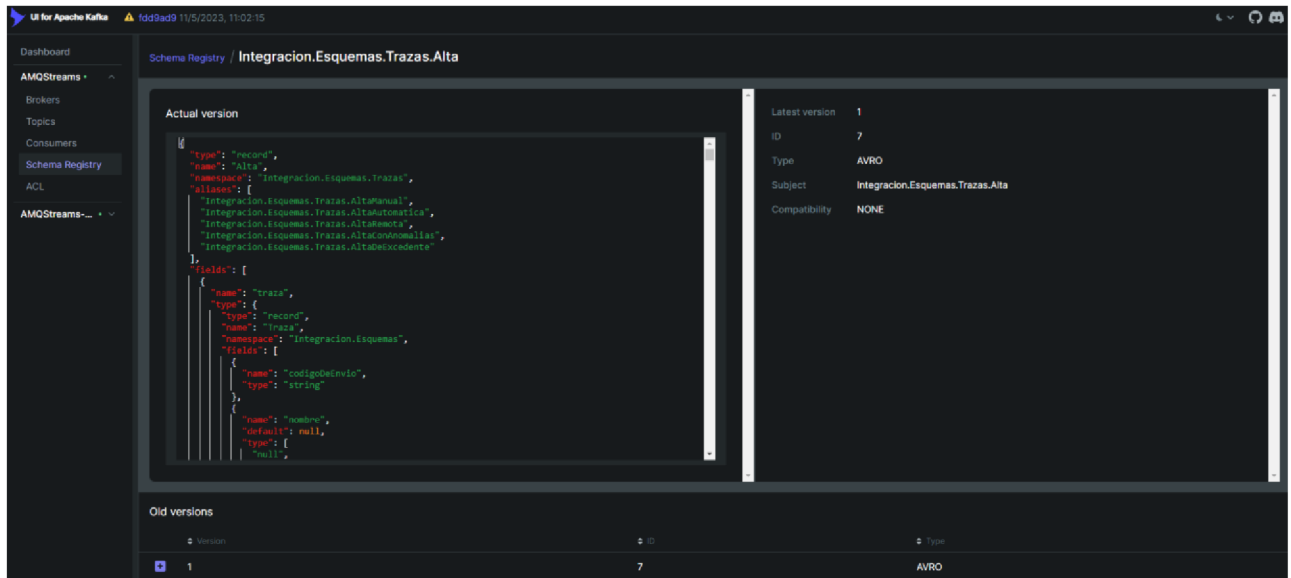
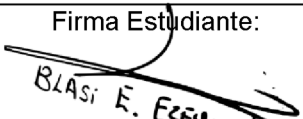
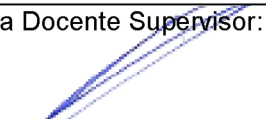
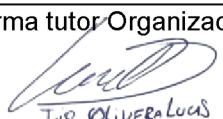


FIGURA 5 - ESTRUCTURA DE LOS EVENTOS DE TIPO ALTA

- Kafka Producer: la API Producer permite que las aplicaciones envíen flujos de datos a los brókeres de los clústeres de Apache para categorizarlos y almacenarlos (en los tópicos mencionados anteriormente).
- Kafka Consumer: la API Consumer proporciona a los consumidores de Apache Kafka acceso de lectura a los datos almacenados en los topics del clúster.
- Kafka Streams: mediante la API Streams, las aplicaciones pueden actuar como procesadores de flujos para convertir los flujos de datos entrantes en salientes.
- Kafka Connect: gracias a la API Connect, es posible configurar productores y consumidores reutilizables que comuniquen los topics de Kafka con las aplicaciones o sistemas de bases de datos existentes.
- Kafka Admin Client: la interfaz Admin Client se utiliza para facilitar la administración y supervisión de los clústeres de Kafka.

Firma Estudiante:  BLAS E. EZEQUIEL	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
--	--	--

Las aplicaciones cliente y los servidores individuales de los clústeres de Apache se comunican a través de un protocolo sencillo, de alto rendimiento y multilenguaje basado en TCP. De manera predeterminada, los desarrolladores ponen a disposición un cliente Java para Apache Kafka, aunque también hay clientes en muchos otros lenguajes, como PHP, Python, C/C ++, Ruby, Perl o Go.

.NET CORE 6

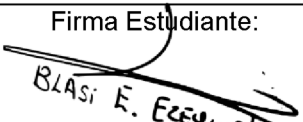
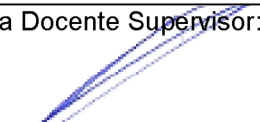
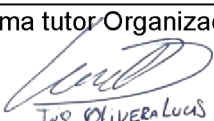
La plataforma de desarrollo se realiza con .NET Core 6 que es la plataforma de desarrollo de Microsoft más moderna, multiplataforma y de alto rendimiento para la creación de todo tipo de aplicaciones.

Fue liberada en el año 2016, y es el resultado de múltiples esfuerzos para hacer más ágil el desarrollo en .NET puesto que el .NET Framework, depende totalmente del sistema operativo Windows.

.NET Core, a diferencia del .NET Framework, no tiene este tipo de dependencia del sistema Windows y es modular, usando el sistema de paquetes Nuggets, gracias al cual, recibiremos las diversas actualizaciones de .NET Core, a diferencia de .NET Framework que se actualiza a través de Windows Update.

Esta arquitectura modular permite que podamos incluir únicamente lo necesario que requieren nuestras aplicaciones, haciendo que tengan un peso menor a la hora de ser desplegadas y que sea más sencilla su actualización a través de updates de librerías. NET Core es multiplataforma y viene con soporte para su uso con contenedores Docker, tiene alto rendimiento, se ha desarrollado desde cero y se le ha dado una alta importancia a esta característica, maneja asincronía con el uso de async/await. Se ha implementado este patrón en todas las librerías comunes para mejorar el rendimiento en las llamadas I/O es Open Source.

Referencias [02] – .NET Core 6

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

.NET Core se centra en el rendimiento y la escalabilidad. Ofrece tiempos de arranque rápidos y un alto rendimiento, lo que lo hace adecuado para aplicaciones críticas en términos de rendimiento.

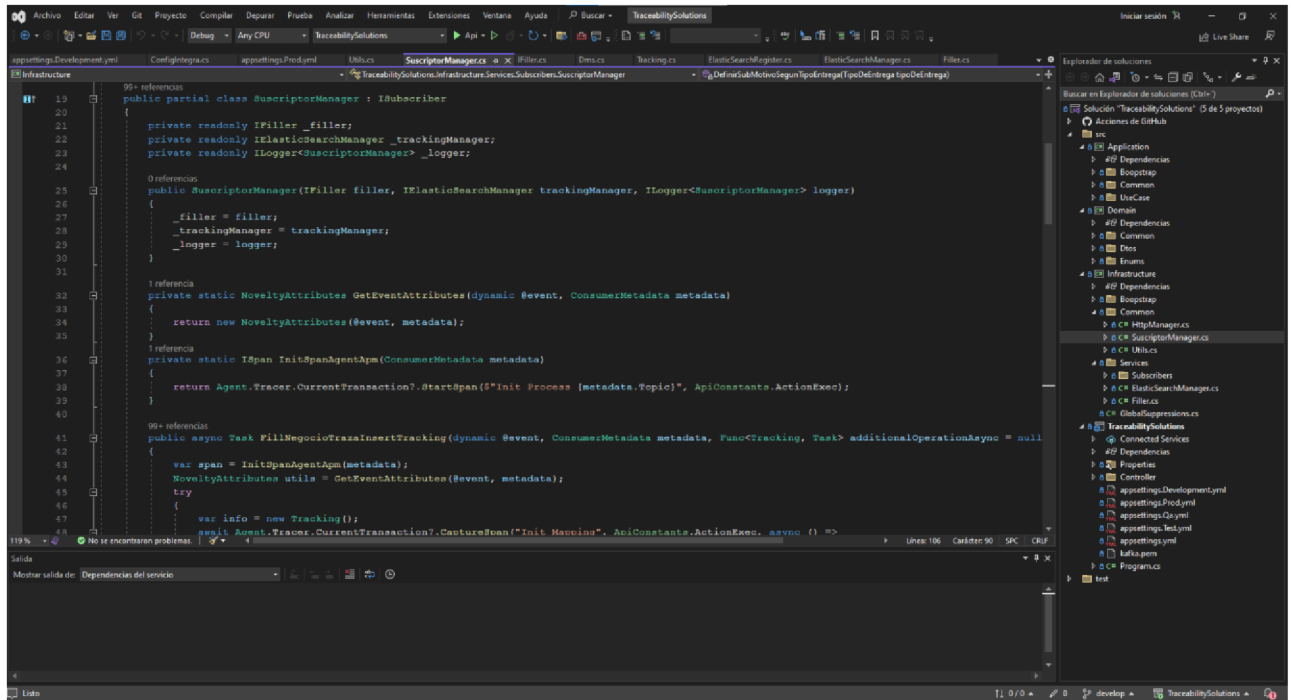
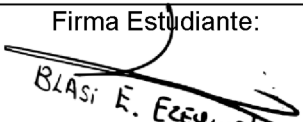
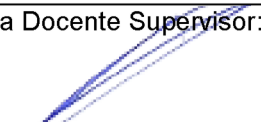
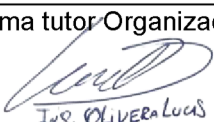


FIGURA 6 - IDE VISUAL ESTUDIO 2022 CON PARTE DEL DESARROLLO

Admite varios lenguajes de programación, como C#, F#, y Visual Basic. Esto permite a los desarrolladores elegir el lenguaje con el que se sientan más cómodos.

En el lenguaje de C# que es una programación multiparadigma y muy sofisticado que ha evolucionado en conjunto con la estrategia de .NET. Tiene nuevas características, como tipos por referencia nullables, rangos, índices, streams asíncronos... es la opción por defecto a la hora de crear aplicaciones en .NET Core y de la que más documentación vas a encontrar.

El framework utiliza una biblioteca de clases unificada que incluye una amplia gama de funcionalidades, lo que facilita el desarrollo de aplicaciones ricas y diversas. Se integra bien con contenedores, lo que lo hace ideal para el desarrollo de aplicaciones basadas

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

en microservicios. Se puede implementar fácilmente aplicaciones en contenedores Docker y gestionarlas con Kubernetes.

Incluye características de seguridad avanzadas, como autenticación y autorización integradas, protección contra ataques comunes y soporte para certificados SSL. Permite desarrollar aplicaciones web a través de ASP.NET Core, que es un marco de trabajo para la creación de aplicaciones web modernas. Ofrece soporte para API REST, aplicaciones de una sola página (SPA) y más.

Se integra con Visual Studio y Visual Studio Code, dos entornos de desarrollo ampliamente utilizados que ofrecen una variedad de características y extensiones para simplificar el desarrollo.

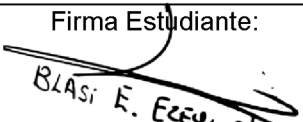
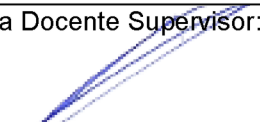
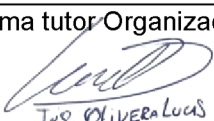
Posee una comunidad activa y un sólido soporte por parte de Microsoft, lo que garantiza actualizaciones regulares, corrección de errores y mejoras de rendimiento.

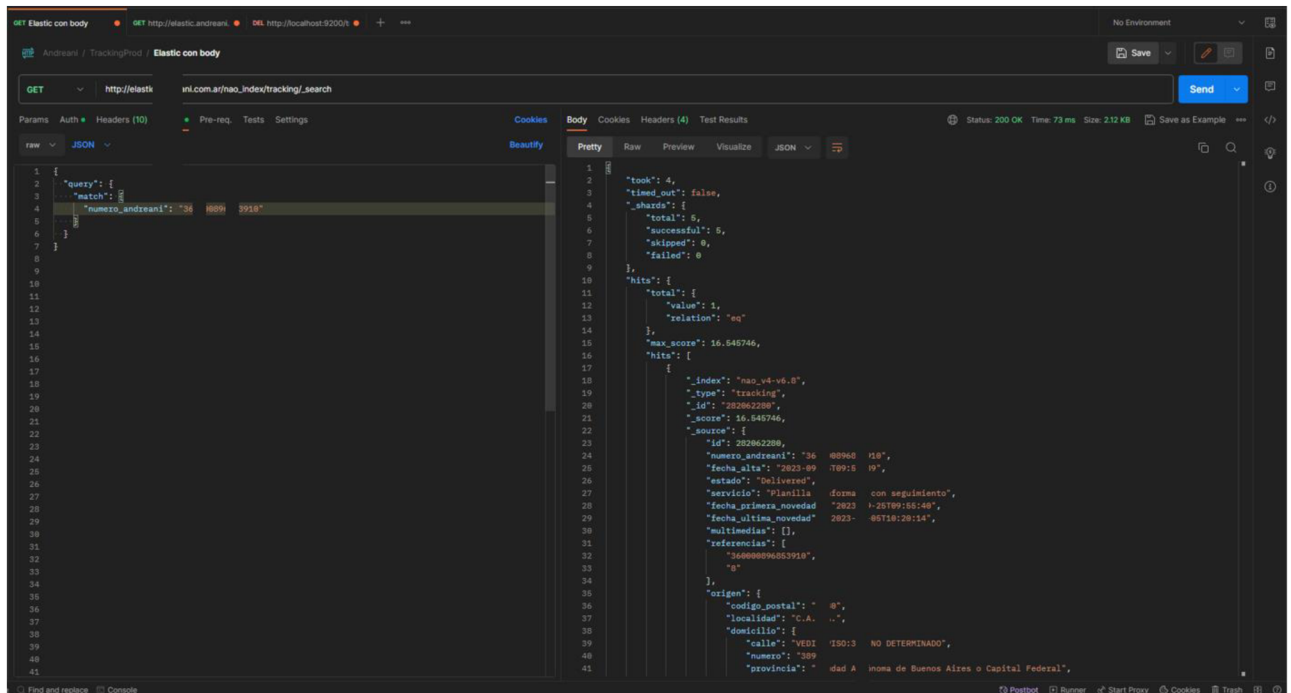
ElasticSearch

La persistencia se implementa en una base de datos no relacional de buena performance a la hora de realizar búsquedas en un caudal de datos muy grande como lo provee ElasticSearch Una tecnología que puede ayudarte en tu análisis de datos y búsquedas masivas de Big Data.

ElasticSearch es un motor de búsqueda empresarial open source, escalable, y de arquitectura distribuida. A través de una API REST, puede ser accedida y extendida para darle soporte a todas tus aplicaciones de business, intelligence, big data o business analytics.

ElasticSearch posee una variedad de conectores y plugins que permiten obtener información desde múltiples orígenes de datos: Archivos de logs, bases de datos, sistemas operativos, monitoreo de alertas, etc.

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
--	--	---



```
1 {
2   "query": {
3     "match": {
4       "numero_andreani": "36 809 3918"
5     }
6   }
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599

```

mensajes en documentos internos y los guarda en una estructura básica interna similar a un objeto JSON. Cada documento es simplemente un set de claves y valores. Las claves serán cadenas de texto, mientras que las claves pueden ser de varios tipos.

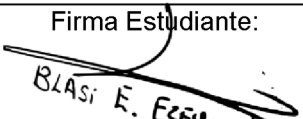
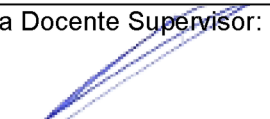
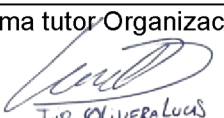
Si bien existen conectores que pueden guardar datos en forma automática, también se pueden agregar manualmente. Simplemente con requests post hacia la API, se puede transmitir nuestros datos. Las búsquedas también se realizarán mediante JSON. La API de Elasticsearch permite recuperar, enviar, verificar y validar datos directamente desde la línea de comandos y desde varios lenguajes de programación.

Debemos recalcar que Elasticsearch no es una base de datos relacional, por lo que dichos conceptos no aplican. El concepto más importante es la normalización. No permite los join o subconsultas, por lo que tener los datos desnormalizados es esencial para el desempeño correcto.

Típicamente, se guardará un documento una única vez en cada repositorio que resida. Si bien es contraria a la perspectiva de las bases de datos relacionales convencionales, es óptimo para este motor de búsqueda. Las búsquedas de texto serán extremadamente rápidas debido a la cercana proximidad entre los documentos guardados y su correspondiente meta data en los índices.

ElasticSearch puede escalar hasta miles de servidores y administrar peta bytes de datos. Es el resultado de la arquitectura completamente distribuida. Otra ventaja es que el usuario final no debe preocuparse por toda la administración ya que todo esto ocurre sin que nos enteremos.

Es una gran herramienta para empresas que requieren búsquedas de texto o análisis en tiempo real de gran cantidad de datos. Además, permite analizar logs de servidores web para detectar anomalías y enviar alertas ante posibles fallos. Incluso permite integrarse con herramientas de machine learning para predecir inconvenientes.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUIS
---	--	---

Open Shift

La infraestructura de servidor donde está desplegado el proyecto es Open Shift, una plataforma de contenedores de código abierto desarrollada por Red Hat que proporciona un entorno de desarrollo, implementación y administración de aplicaciones en contenedores, lo que facilita la creación, el despliegue y la escalabilidad de aplicaciones de manera eficiente.

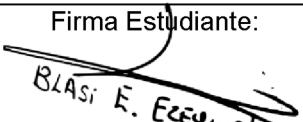
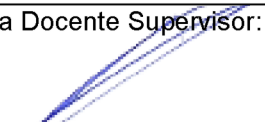
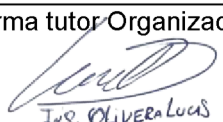
Esta infraestructura permite una orquestación de Contenedores ya que se basa en Kubernetes, una plataforma de orquestación de contenedores ampliamente utilizada, para gestionar y orquestar contenedores Docker. Kubernetes automatiza tareas como la implementación, escalado y balanceo de carga de contenedores, lo que simplifica la administración de aplicaciones.

Facilita el proceso de despliegue de aplicaciones al proporcionar herramientas y flujos de trabajo para empaquetar aplicaciones en contenedores y luego implementarlos en clústeres Kubernetes. Los desarrolladores pueden utilizar comandos simples o integraciones con CI/CD (Integración Continua/Entrega Continua) para implementar aplicaciones de manera rápida y coherente.

Permite escalar aplicaciones de manera horizontal y vertical, según las necesidades de carga de trabajo. La plataforma incluye funciones avanzadas de administración de recursos, como límites y cuotas, para garantizar un uso eficiente de los recursos del clúster.

Fomenta las prácticas de DevOps al proporcionar una plataforma unificada para desarrolladores y operadores. Las integraciones con herramientas de CI/CD permiten automatizar las pruebas y el despliegue continuos, lo que acelera el ciclo de desarrollo y mejora la calidad del software.

Incorpora características de seguridad avanzadas, como la segmentación de red, la autenticación y la autorización basadas en roles. Esto ayuda a proteger las aplicaciones y los datos en entornos de múltiples usuarios y equipos.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
---	--	--

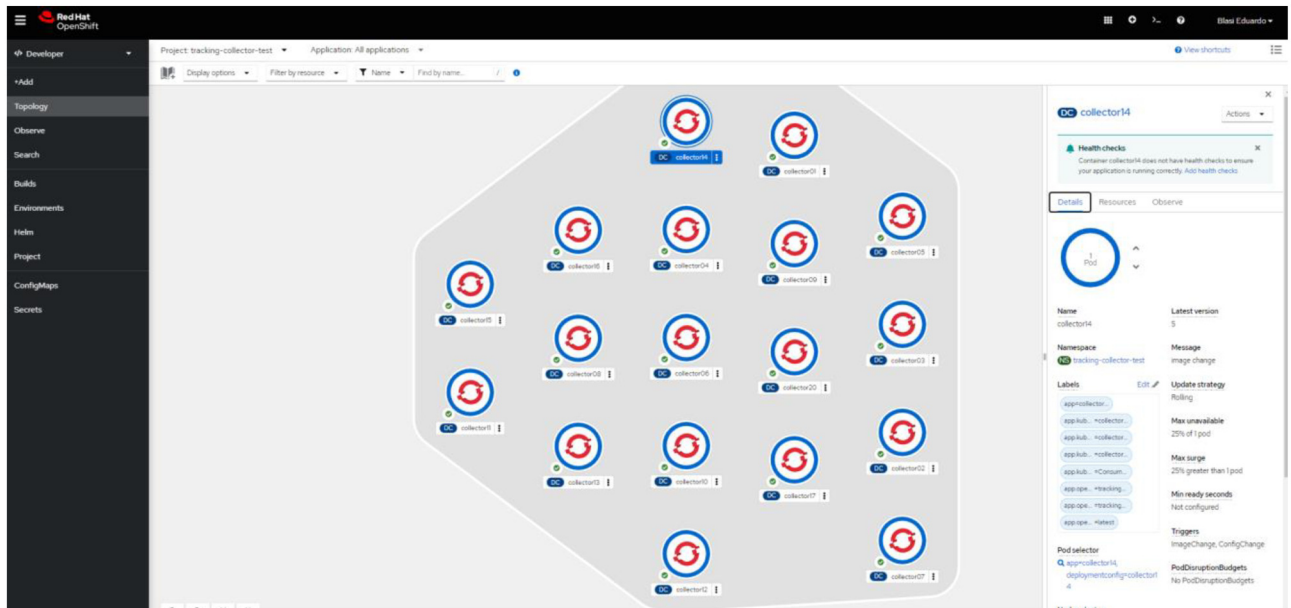


FIGURA 8 - DESPLIEGUE DE INSTANCIAS CON VARIOS POD DE LA MISMA APLICACIÓN

Open Shift es compatible con la arquitectura de microservicios y facilita la administración y el despliegue de servicios individuales. Ya que se pueden implementar y escalar microservicios de manera independiente, lo que mejora el modularidad y la flexibilidad de las aplicaciones.

Proporciona un Marketplace de complementos y extensiones que permite a los usuarios agregar nuevas funcionalidades a sus clústeres. Esto incluye servicios de bases de datos, monitoreo, registro y muchas otras opciones. Es compatible con una variedad de infraestructuras, incluyendo despliegues en la nube pública, entornos locales y otras soluciones de infraestructura, lo que brinda flexibilidad para implementar aplicaciones en diferentes entornos.

Firma Estudiante:

BLAS E. ECEYURET

Firma Docente Supervisor:

[Firma manuscrita]

Firma tutor Organizacional:

JOS. OLIVERA LUIS

Metodología

Para abordar el proyecto de migración de nuestro sistema existente a una nueva plataforma, se optó por implementar una metodología ágil. Este enfoque de desarrollo se ha vuelto cada vez más relevante y efectivo en situaciones donde los requisitos pueden evolucionar o no están completamente definidos desde el principio, lo cual es particularmente pertinente en el contexto de una migración de sistema.

Scrum

La metodología ágil se caracteriza por su enfoque en la entrega continua y la flexibilidad en la gestión de proyectos. En lugar de una planificación exhaustiva desde el inicio, el proyecto se divide en iteraciones o Sprint más cortos, generalmente de 2 a 4 semanas de duración. Esto permite responder rápidamente a cambios en los requisitos, desafíos inesperados o nuevas necesidades a medida que surgen durante el proceso de migración.

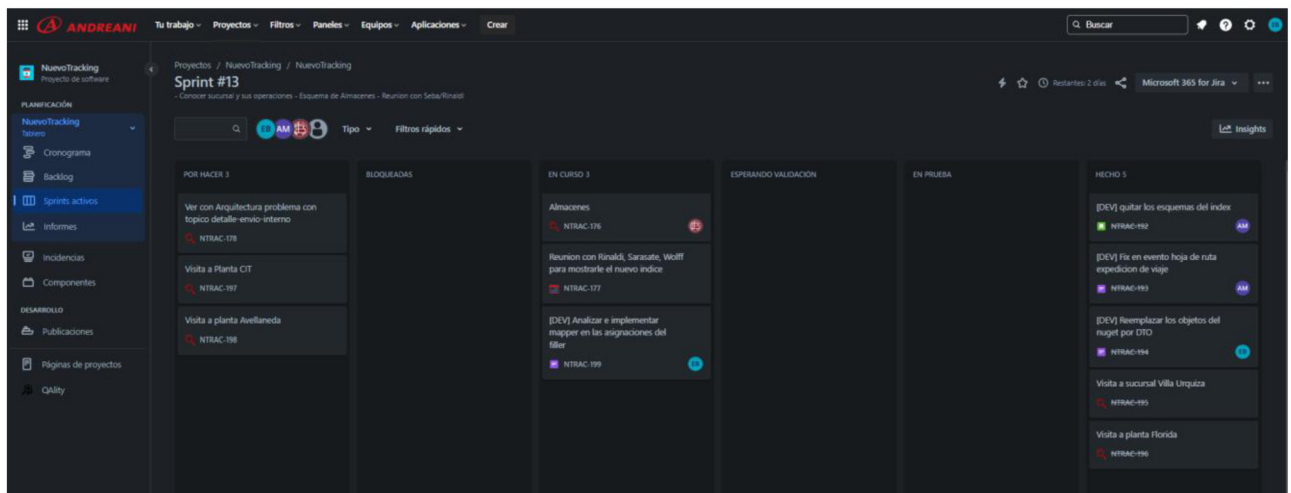
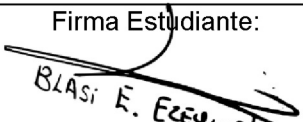
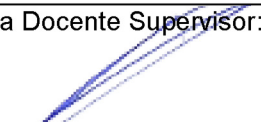
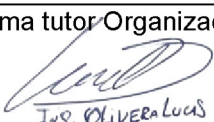


FIGURA 9 - TABLERO DEL EQUIPO EN EL SPRINT N° 13

Referencias [04] – Scrum

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

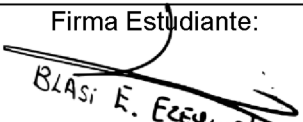
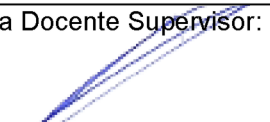
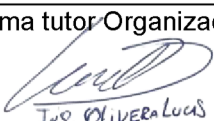
Uno de los beneficios clave de utilizar una metodología ágil en una migración de sistema es su capacidad para adaptarse a modificaciones en el camino. Dado que las migraciones a menudo involucran la comprensión y el traspaso de sistemas existentes que pueden tener lagunas de documentación o comportamientos inesperados, la flexibilidad de la metodología ágil permite abordar estos desafíos de manera efectiva.

La elección de implementar la metodología ágil en este proyecto no solo refleja la capacidad de adaptarse a las cambiantes necesidades y desafíos inherentes a una migración de sistema, sino que también refuerza la determinación de establecer una comunicación sólida y una comprensión profunda de todos los sectores involucrados en este proceso. Para garantizar el éxito de la empresa, se lleva a cabo un esfuerzo exhaustivo para interactuar y colaborar estrechamente con los equipos de operaciones en cada una de nuestras plantas y sucursales.

Esta decisión ha implicado la realización de un gran número de viajes a diversas ubicaciones para sumergirse en el funcionamiento interno de las operaciones. Se trabajó incansablemente para comprender las operaciones productivas desde adentro, escuchando y aprendiendo de los empleados y equipos que desempeñan un papel vital en el éxito de las plantas y sucursales. Estos viajes no solo han fortalecido las relaciones con los equipos locales, sino que también han proporcionado una visión invaluable sobre las particularidades de cada ubicación.

La metodología seleccionada tiene como primer paso la formación del equipo Scrum. Este equipo está compuesto por tres roles principales: el Product Owner (Propietario del Producto), el Scrum Máster, y el Equipo de Desarrollo. El Product Owner es responsable de definir las prioridades y requisitos del proyecto, el Scrum Máster actúa como facilitador y elimina obstáculos para el equipo, y el Equipo de Desarrollo es el grupo que trabaja en las tareas del proyecto.

El Product Owner crea y mantiene el "Backlog del Producto". Este es una lista priorizada de todas las características, funcionalidades y tareas que deben realizarse durante el

Firma Estudiante:  BLAS E. EZEQUEL	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

proyecto. El backlog se actualiza continuamente a medida que se obtiene más información y se ajustan las prioridades.

Un Sprint es un período de tiempo fijo, típicamente de 2 a 4 semanas, durante el cual se trabaja en un conjunto de elementos del backlog. Antes de cada Sprint, el equipo se reúne en una "Planificación de Sprint" en la que selecciona las tareas del backlog que se compromete a completar durante ese Sprint. Estas tareas se trasladan al "Backlog del Sprint".

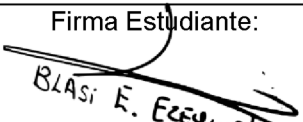
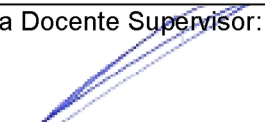

Durante el Sprint, el Equipo de Desarrollo trabaja en las tareas seleccionadas del Backlog del Sprint. Realizan reuniones diarias cortas llamadas "Reuniones Diarias de Scrum" para mantenerse actualizados sobre el progreso y resolver cualquier obstáculo.

Al final de cada Sprint, se lleva a cabo una "Revisión del Sprint". El equipo demuestra el trabajo realizado durante el Sprint, mostrando las nuevas características o funcionalidades completadas. Esto permite obtener retroalimentación temprana y ajustar la dirección del proyecto según sea necesario.

Después de la Revisión del Sprint, el equipo realiza una "Retrospectiva del Sprint". En esta reunión, el equipo analiza lo que funcionó bien y lo que podría mejorarse en términos de procesos y colaboración. Se establecen acciones para mejorar en el próximo Sprint.

El proceso se repite en ciclos sucesivos, con nuevos Sprint planificados y ejecutados. La metodología Scrum se basa en la idea de mejora continua y adaptación a medida que se avanza en el proyecto.

La implementación de Scrum se centra en la transparencia, la inspección y la adaptación. Facilita la colaboración efectiva y la entrega de valor de manera incremental y constante. Scrum es especialmente adecuado cuando se trabaja en proyectos en los que los requisitos pueden cambiar o no están completamente definidos desde el principio, lo que lo convierte en una metodología ágil altamente adaptable y flexible.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUCAS
---	--	--

Arquitectura

La arquitectura limpia es un enfoque de diseño de software que busca crear sistemas robustos, mantenibles y escalables al enfocarse en la separación de preocupaciones y la dependencia de capas. Fue popularizada por el autor y consultor Robert C. Martin, conocido como "Uncle Bob". La arquitectura limpia propone un diseño que se divide en capas y círculos concéntricos, con el objetivo de mantener un alto nivel de independencia entre estas capas.

En el centro, se encuentra el núcleo de la aplicación, que contiene las reglas de negocio y la lógica de dominio. A medida que te alejas del núcleo, encuentras capas como la de aplicación, la de interfaz de usuario y las capas de infraestructura, cada una con un nivel creciente de dependencia hacia afuera.

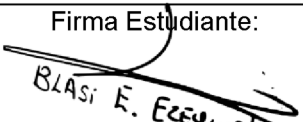
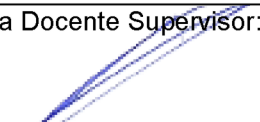
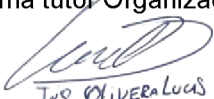
La arquitectura limpia busca minimizar la dependencia de frameworks y tecnologías específicas en las capas interiores. Esto facilita la sustitución de componentes o la adaptación a cambios tecnológicos sin afectar la lógica de negocio.

Se promueve el principio de Inversión de Dependencia que sugiere que las dependencias deben apuntar hacia abajo en la jerarquía de capas, lo que permite una mayor flexibilidad y mantenibilidad del sistema.

Esta arquitectura limpia favorece la creación de unidades de prueba para todas las capas del sistema, lo que facilita la detección de problemas y la implementación de cambios sin impactos no deseados.

Se busca minimizar el acoplamiento entre las capas, permitiendo que cada una evolucione de forma independiente. Esto hace que sea más fácil realizar mejoras o extensiones sin afectar otras partes del sistema.

La distribución por capas es un patrón arquitectónico común en el desarrollo de software que organiza el sistema en capas bien definidas. Cada capa tiene un conjunto específico de responsabilidades y se comunica con las capas adyacentes a través de

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

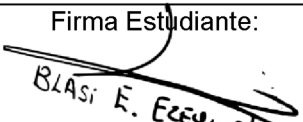
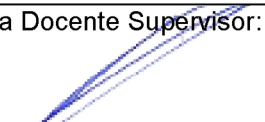
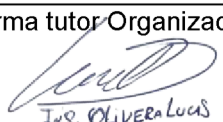
interfaces claramente definidas. Estas son algunas de las capas típicas en una arquitectura de distribución por capas:

- **Capa de Presentación o Interfaz de Usuario:** Esta capa se encarga de la interacción con los usuarios, como la presentación de datos y la captura de entrada. Puede incluir interfaces gráficas, aplicaciones web o servicios API.
- **Capa de Aplicación o Lógica de Aplicación:** Aquí reside la lógica de la aplicación que coordina las operaciones y la comunicación entre la capa de presentación y la capa de dominio. Suele gestionar flujos de trabajo, seguridad y transacciones.
- **Capa de Dominio:** Esta capa contiene la lógica de negocio central y las reglas del dominio. Es independiente de las preocupaciones de la interfaz de usuario y la infraestructura. Es la capa donde se implementan las reglas de negocio críticas para la aplicación.
- **Capa de Acceso a Datos:** Aquí se gestiona la interacción con la base de datos o el almacenamiento de datos. Se encarga de la lectura y escritura de datos, así como de la traducción entre la representación de datos en la base de datos y los objetos de dominio.

"Clean Code" se refiere a la práctica de escribir código fuente de alta calidad que sea fácil de entender, mantener y extender. La idea es reducir la complejidad, aumentar la legibilidad y seguir principios sólidos de diseño de software. Algunos aspectos clave de la implementación de Clean Code incluyen:

- **Nombres descriptivos:** Usa nombres de variables, funciones y clases que reflejen su propósito y significado en lugar de abreviaturas o nombres confusos.

Referencias [05] – Clean Code

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
--	--	--

- Funciones pequeñas: Divide funciones largas en funciones más pequeñas con nombres descriptivos. Cada función debe hacer una sola cosa y hacerla bien.
- Comentarios significativos: Utiliza comentarios para explicar el por qué detrás del código en lugar de simplemente describir lo que hace. Un buen código debe ser auto explicativo en la medida de lo posible.
- Evita la duplicación: No repitas el mismo código en múltiples lugares. Usa funciones o clases para encapsular la lógica compartida.
- Pruebas unitarias: Escribe pruebas unitarias para verificar que el código funcione como se espera y continúa funcionando a medida que evoluciona.
- Refactorización constante: Mejora el código continuamente a medida que adquieres un mejor entendimiento de los requisitos y las estructuras del sistema.

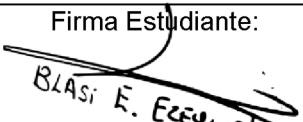
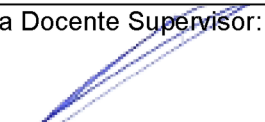
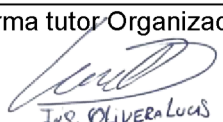
Principios solid

Los principios SOLID son un conjunto de pautas de diseño de software que promueven la creación de sistemas más mantenibles, extensibles y robustos. A continuación, se describe cómo se aplicarán estos principios en la arquitectura:

Principio de Responsabilidad Única (Single Responsibility Principle - SRP): Cada componente de la arquitectura se diseñará de manera que tenga una única razón para cambiar. Por ejemplo, los módulos de registro de trazas se encargarán exclusivamente de la gestión de trazas, separando claramente esta responsabilidad de otras funcionalidades.

Principio de Abierto/Cerrado (Open/Closed Principle - OCP): La arquitectura se diseñará de manera que esté abierta a extensiones, pero cerrada a modificaciones. Se utilizarán interfaces y abstracciones para permitir la incorporación de nuevas características sin alterar el código existente.

Principio de Sustitución de Liskov (Liskov Substitution Principle - LSP): Se garantizará que las clases derivadas puedan sustituir a sus clases base sin afectar el

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
--	--	---

SonarQube

SonarQube es una plataforma de código abierto para la inspección continua de la calidad del código a través de diferentes herramientas de análisis estático de código fuente. Proporciona métricas que ayudan a mejorar la calidad del código de un programa permitiendo a los equipos de desarrollo hacer seguimiento y detectar errores y vulnerabilidades de seguridad para mantener el código limpio.

Es una herramienta esencial para la fase de testing y auditoría de código dentro del ciclo de desarrollo de una aplicación y se considera perfecta para guiar a los equipos de desarrollo durante las revisiones de código. Soporta una etapa de inspección continua.

SonarQube está escrito en Java, pero puede analizar y administrar código 29 lenguajes de programación importantes, incluidos c / c ++, PL / SQL, Cobol, etc., a través de conjuntos de reglas incorporados que pueden ser ampliados con varios complementos.

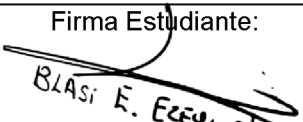
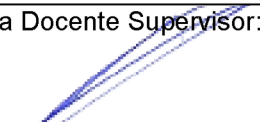
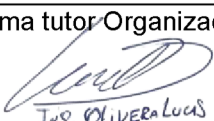
Esta herramienta al centrarse en el nuevo código ayuda a la detección de muchos problemas, facilitando su solución de manera rápida en cuanto aparece por lo que permite mantener el código limpio, sencillo y fácil de leer.

Los desarrolladores se apoyan en esta plataforma para lograr la integración e implementación continuas del código, que no solo funciona para detectar los problemas, sino también, para rastrear, controlar y verificar la calidad continua del código.

Realiza revisiones automáticas con análisis de código estático detectando problemas que afectan la calidad del código.

Facilita informes ofreciendo información objetiva de la calidad actual de los proyectos utilizando métricas y gráficos de prueba de calidad avanzados. Esto incluye datos sobre

Referencias [06] – SonarQube

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
--	--	--

código duplicado, estándares de codificación, pruebas unitarias, cobertura de código, complejidad del código, errores potenciales, comentarios, diseño y arquitectura.

Se integra con toda la cadena de herramientas de DevOps ayudando al flujo de trabajo productivo, incluidos los sistemas de compilación, los motores de CI, utilizando webhooks y su Rest API integral.

Como beneficio esta herramienta alerta de manera automática a los desarrolladores de los errores de código para corregirlos previamente a la implementación en producción.

No sólo muestra los errores, también las reglas de codificación, la cobertura de las pruebas, las duplicaciones, la complejidad y la arquitectura, plasmando todos estos datos en paneles de control detallados.

Ayuda al equipo a mejorar en sus habilidades como programadores al facilitar un seguimiento de los problemas de calidad.

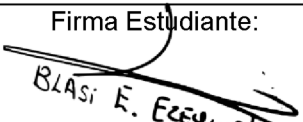
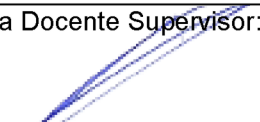
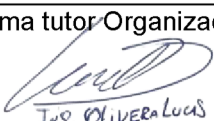
Permite la creación de paneles y filtros personalizables para centrarse en áreas clave y entregar productos de calidad a tiempo.

Favorece la productividad al reducir la complejidad del código acortando tiempos y costes adicionales al evitar cambiar el código constantemente.

Multi-Threaded

Para optimizar aún más el rendimiento del sistema, se implementará el procesamiento en paralelo mediante el uso de hilos. Esto permitirá que múltiples tareas de procesamiento se ejecuten simultáneamente, lo que es especialmente beneficioso para la gestión de trazas de envío concurrentes. Los componentes clave serán:

Pool de Hilos: Se establecerá un conjunto de hilos disponibles para manejar tareas de procesamiento de trazas de envío.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J.R. OLIVERA LUGAS
---	--	--

División de Tareas: Se diseñará el sistema de manera que las trazas de envío se puedan dividir en tareas independientes que se procesarán en paralelo.

Control de Concurrencia: Se implementarán mecanismos de control de concurrencia, como semáforos o bloqueos, para evitar condiciones de carrera y garantizar la integridad de los datos compartidos.

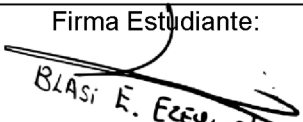
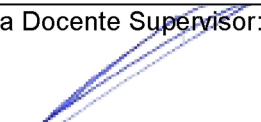
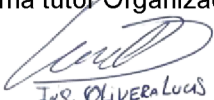
La utilización de hilos para el procesamiento en paralelo mejorará significativamente el rendimiento del sistema, permitiendo una mayor capacidad de procesamiento y una menor latencia en el procesamiento de las trazas de envío.

La combinación de estos principios, herramientas y la arquitectura limpia con la distribución por capas proporcionará una base sólida para la migración del servicio de trazas de envío, mejorando la mantenibilidad, la extensibilidad y la robustez del sistema.



FIGURA 11 - SEGUIMIENTO DE BUENAS PRÁCTICAS QUE INTENTA PROMULGAR LA EMPRESA

Para mejorar la eficiencia y la capacidad de respuesta del sistema, se implementará un mecanismo de consumo de eventos asincrónico utilizando Apache Kafka. Esto permitirá que el sistema procese eventos de trazas de envío de manera eficiente y sin bloqueos. Los componentes clave incluirán:

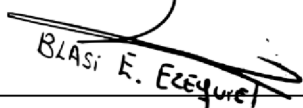
<p>Firma Estudiante:</p>  <p>BLAS E. Eceguet</p>	<p>Firma Docente Supervisor:</p> 	<p>Firma tutor Organizacional:</p>  <p>JOS. OLIVERA LUCAS</p>
---	--	--

Productor Kafka: Los sistemas encargados de enviar eventos de trazas de envío a los tópicos para ser consumidos por los diferentes consumidores.

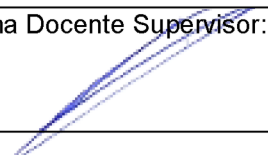
Consumidor Kafka: Responsable de suscribirse al tópico y procesar los eventos entrantes de manera asincrónica.

La arquitectura de consumo de eventos a través de Kafka permitirá una comunicación asíncrona entre diferentes partes del sistema, mejorando la escalabilidad y la capacidad de manejo de picos de carga.

Firma Estudiante:


BLAS E. ECEGUET

Firma Docente Supervisor:



Firma tutor Organizacional:


Tutor OLIVERA LUCAS

Funcionamiento previo del sistema de seguimiento de envíos

El servicio de recolección de trazas de envío tiene como objetivo capturar y gestionar la información relevante asociada a un envío desde el momento en que se generan los primeros eventos. El proceso de recolección se inicia con la publicación de novedades en un gestor de colas IBM MQ por parte de varios sistemas involucrados en el seguimiento y control de envíos.

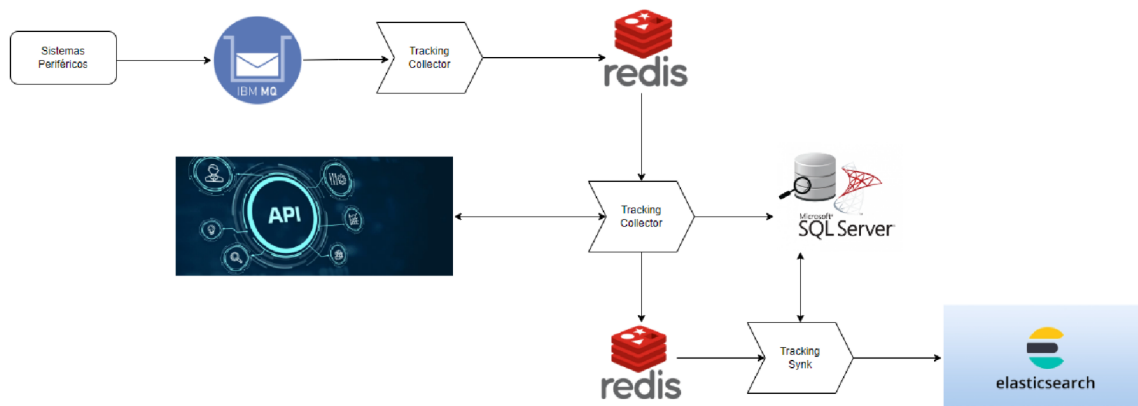
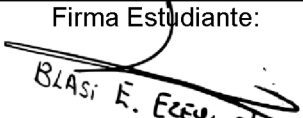
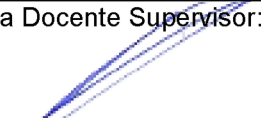
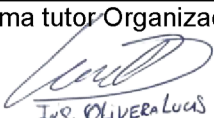


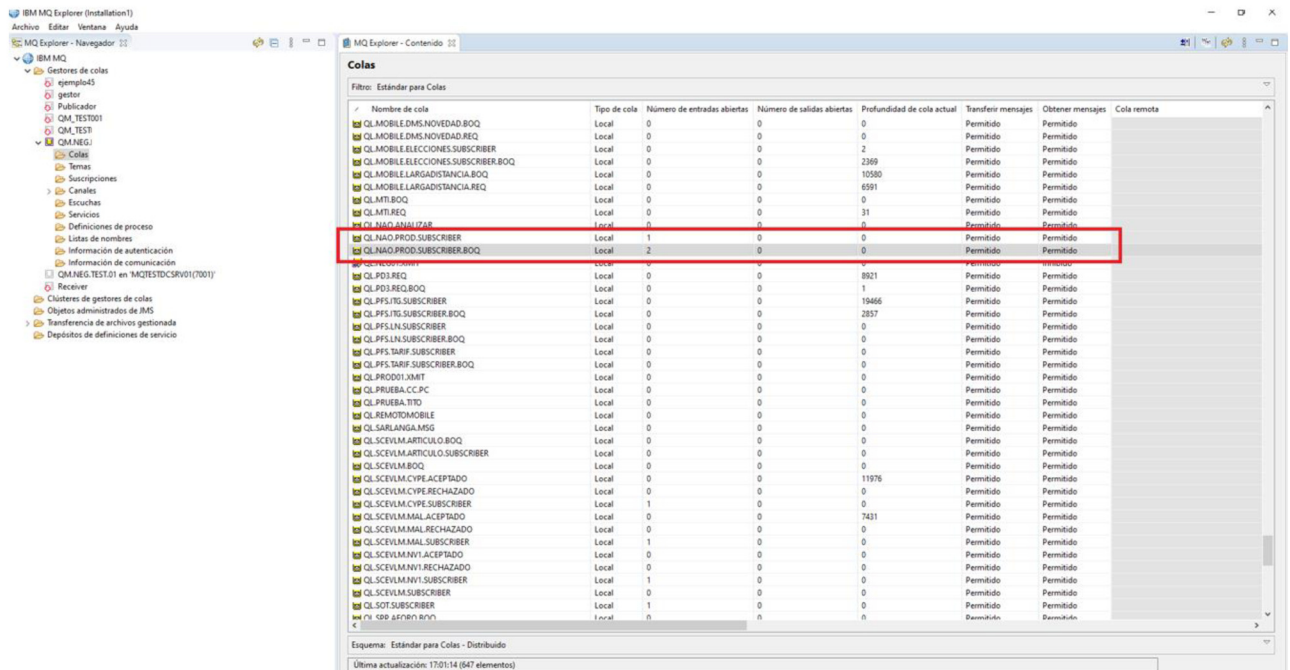
FIGURA 12 - DIAGRAMA DE FLUJO ANTERIOR VERSIÓN

En el primer paso de este proceso, diversos sistemas y aplicaciones involucrados en la logística y seguimiento de envíos generan eventos que representan novedades o cambios en el estado de un envío. Estos eventos pueden incluir información sobre la ubicación actual del paquete, actualizaciones de estado, eventos de carga y descarga, entre otros.

Una vez que se generan estos eventos, se publican de manera segura en un gestor de colas IBM MQ. Este gestor de colas actúa como un intermediario que asegura la entrega confiable y ordenada de los mensajes a los consumidores interesados, en este caso, nuestro servicio de recolección de trazas de envío.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  JOS. OLIVERA LUGAS
---	--	--

El servicio de recolección de trazas de envío, que se ejecuta de manera continua, está suscrito al gestor de colas IBM MQ. Esto significa que está a la espera de nuevos eventos publicados en la cola asociada a los envíos.



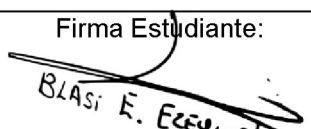
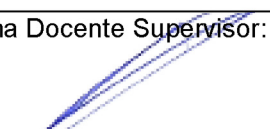
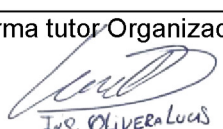
Nombre de cola	Tipo de cola	Número de entradas abiertas	Número de salidas abiertas	Profundidad de cola actual	Transferir mensajes	Obtener mensajes	Cola remota
QL.MOBILE.DMS.NOVEDAD.BQO	Local	0	0	0	Permitido	Permitido	
QL.MOBILE.DMS.NOVEDAD.REQ	Local	0	0	0	Permitido	Permitido	
QL.MOBILE.ELECCIONES.SUBSCRIBER	Local	0	0	2	Permitido	Permitido	
QL.MOBILE.ELECCIONES.SUBSCRIBER.BQO	Local	0	0	2369	Permitido	Permitido	
QL.MOBILE.LARGADISTANCIA.BQO	Local	0	0	10580	Permitido	Permitido	
QL.MOBILE.LARGADISTANCIA.REQ	Local	0	0	6591	Permitido	Permitido	
QL.MTI.BQO	Local	0	0	0	Permitido	Permitido	
QL.MTI.REQ	Local	0	0	31	Permitido	Permitido	
QM.NAQ.ANALIZAR	Local	0	0	0	Permitido	Permitido	
QM.NAQ.PROD.SUBSCRIBER	Local	1	0	0	Permitido	Permitido	
QM.NAQ.PROD.SUBSCRIBER.BQO	Local	2	0	0	Permitido	Permitido	
QL.PD31.REQ	Local	0	0	8921	Permitido	Permitido	
QL.PFS.ITG.SUBSCRIBER	Local	0	0	19466	Permitido	Permitido	
QL.PFS.ITG.SUBSCRIBER.BQO	Local	0	0	2837	Permitido	Permitido	
QL.PFS.LN.SUBSCRIBER	Local	0	0	0	Permitido	Permitido	
QL.PFS.LN.SUBSCRIBER.BQO	Local	0	0	0	Permitido	Permitido	
QL.PFS.TARIF.SUBSCRIBER	Local	0	0	0	Permitido	Permitido	
QL.PFS.TARIF.SUBSCRIBER.BQO	Local	0	0	0	Permitido	Permitido	
QL.PROD01.XMT	Local	0	0	0	Permitido	Permitido	
QL.PRUEBA.CC.PC	Local	0	0	0	Permitido	Permitido	
QL.PRUEBA.TITO	Local	0	0	0	Permitido	Permitido	
QL.REMEDIOMOBILE	Local	0	0	0	Permitido	Permitido	
QL.SARLANGLA.MSG	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.ARTICULO.BQO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.ARTICULO.SUBSCRIBER	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.BQO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.CVPE.ACCEPTADO	Local	0	0	11976	Permitido	Permitido	
QL.SCEVLM.CVPE.RECHAZADO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.CVPE.SUBSCRIBER	Local	1	0	0	Permitido	Permitido	
QL.SCEVLM.MMAL.ACCEPTADO	Local	0	0	7431	Permitido	Permitido	
QL.SCEVLM.MMAL.RECHAZADO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.MMAL.SUBSCRIBER	Local	1	0	0	Permitido	Permitido	
QL.SCEVLM.MNVI.ACCEPTADO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.MNVI.RECHAZADO	Local	0	0	0	Permitido	Permitido	
QL.SCEVLM.MNVI.SUBSCRIBER	Local	1	0	0	Permitido	Permitido	
QL.SOT.SUBSCRIBER	Local	1	0	0	Permitido	Permitido	
QM.C80.AFP01.SVN	Local	0	0	0	Desahogado	Desahogado	

FIGURA 13 - INTERFAZ DE IBM MQ DONDE ESTÁN LAS COLAS QUE RECIBEN LOS EVENTOS

Cuando el servicio de recolección de trazas de envío, que se ejecuta de manera continua, está suscrito al gestor de colas IBM MQ y almacena los mensajes en colas temporales en Redis.

Dentro del servicio de Tracking Collector, se implementa una estrategia de respaldo en Redis para mitigar la posibilidad de pérdida de mensajes en las colas de MQ debido a desbordamientos. Los mensajes recopilados se distribuyen y almacenan en 13 workeres diferentes de Redis. Esto garantiza la persistencia y disponibilidad de los mensajes incluso si las colas de MQ alcanzan su capacidad máxima.

El servicio de Tracking Collector, desarrollado en .NET Framework 4.5.2, implementa un gestor de delegados para consumir los eventos almacenados en Redis de manera

<p>Firma Estudiante:</p>  <p>BLAS E. ECEGUET</p>	<p>Firma Docente Supervisor:</p> 	<p>Firma tutor Organizacional:</p>  <p>J. S. OLIVERA LUCAS</p>
---	--	---

paralela. Cada delegado se encarga de procesar un evento y enriquecerlo con datos adicionales utilizando APIS externas para completar la información de la traza.

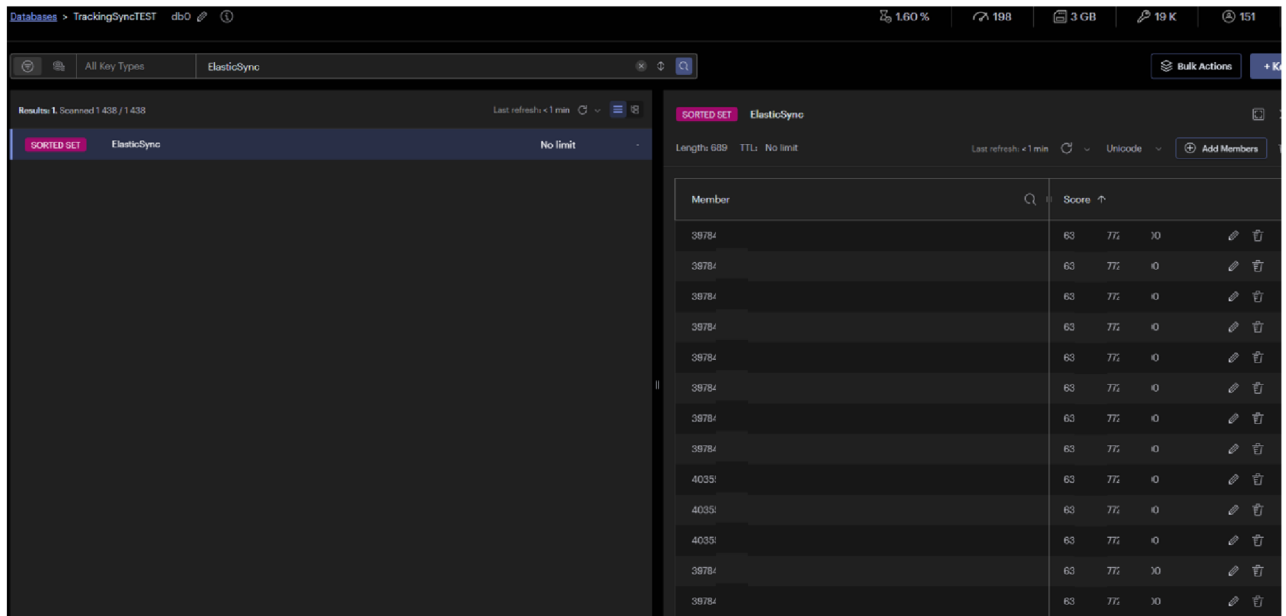
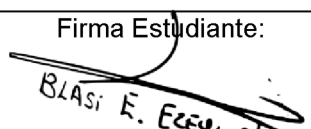
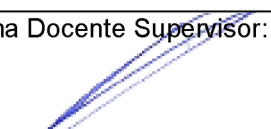
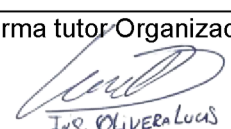


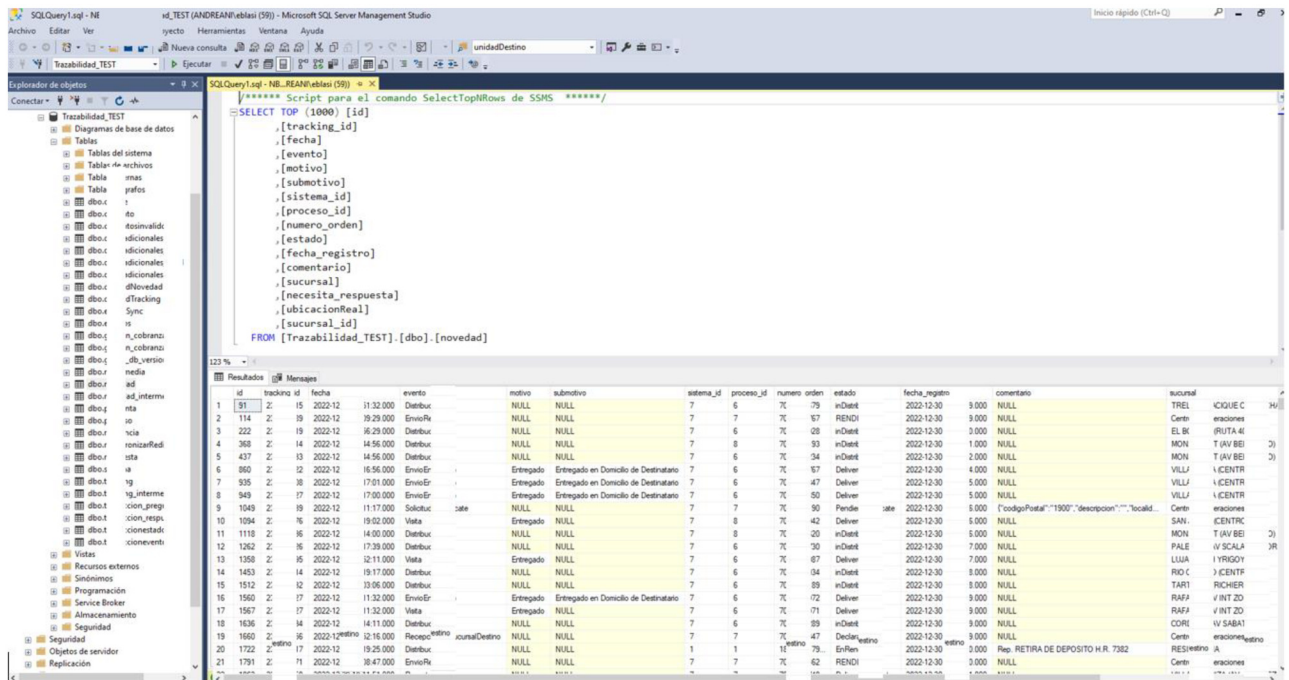
FIGURA 14 - INTERFAZ DE REDIS EN COLA ELASTICSYN Y LOS IDENTIFICADORES DE LOS ENVÍOS

Durante el procesamiento, los delegados pueden acceder a APIS externas o fuentes de datos para obtener información adicional relacionada con los eventos de envío. Esta información enriquecida se agrega a los eventos para generar un único formato de guardado que incluye todos los datos necesarios para un análisis completo de la traza de envío.

Una vez que los eventos se han enriquecido y formateado adecuadamente, se almacenan de manera persistente en una base de datos SQL. Esto crea un registro completo y estructurado de todas las trazas de envío, lo que facilita su posterior consulta y análisis.

Además de almacenar los eventos en la base de datos SQL, el servicio genera identificadores únicos para cada evento y los publica en un nuevo worker de Redis llamado "ElasticSink".

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUCAS
---	--	--



***** Script para el comando SelectTopNRows de SSMS *****

```

SELECT TOP (1000) [id]
, [tracking_id]
, [fecha]
, [evento]
, [motivo]
, [submotivo]
, [sistema_id]
, [proceso_id]
, [numero_orden]
, [estado]
, [fecha_registro]
, [comentario]
, [sucursal]
, [necesita_respuesta]
, [ubicacionReal]
, [sucursals_id]
FROM [Trazabilidad_TEST].[dbo].[novedad]

```

id	tracking_id	fecha	evento	motivo	submotivo	sistema_id	proceso_id	numero_orden	estado	fecha_registro	comentario	sucursal
1	391	15/2022-12	11:32:000	Distribuc	NULL	7	6	79	inDent	2022-12-30	9:000	NULL
2	114	19/2022-12	19:29:000	EnvioRe	NULL	7	7	87	RENDR	2022-12-30	9:000	NULL
3	222	19/2022-12	16:29:000	Distribuc	NULL	7	6	08	inDent	2022-12-30	3:000	NULL
4	368	14/2022-12	14:56:000	Distribuc	NULL	7	8	93	inDent	2022-12-30	1:000	NULL
5	437	13/2022-12	14:56:000	Distribuc	NULL	7	6	34	inDent	2022-12-30	2:000	NULL
6	860	12/2022-12	16:56:000	EnvioEr	Entregado	7	6	87	Deliver	2022-12-30	4:000	NULL
7	935	18/2022-12	17:01:000	EnvioEr	Entregado	7	6	47	Deliver	2022-12-30	5:000	NULL
8	949	17/2022-12	17:00:000	EnvioEr	Entregado	7	6	50	Deliver	2022-12-30	5:000	NULL
9	1049	19/2022-12	11:17:000	Solicit	late	7	7	90	Pendie	2022-12-30	6:000	l'codigoPostal':'1900','descripcion':'', 'localid
10	1094	16/2022-12	19:02:000	Vista	Entregado	7	8	42	Deliver	2022-12-30	6:000	NULL
11	1118	16/2022-12	14:00:000	Distribuc	NULL	7	8	20	inDent	2022-12-30	9:000	NULL
12	1262	16/2022-12	17:29:000	Distribuc	NULL	7	6	30	inDent	2022-12-30	7:000	NULL
13	1358	16/2022-12	12:11:000	Vista	Entregado	7	6	87	Deliver	2022-12-30	7:000	NULL
14	1453	14/2022-12	19:17:000	Distribuc	NULL	7	6	04	inDent	2022-12-30	8:000	NULL
15	1512	12/2022-12	13:06:000	Distribuc	NULL	7	6	89	inDent	2022-12-30	8:000	NULL
16	1560	17/2022-12	11:32:000	EnvioEr	Entregado	7	6	72	Deliver	2022-12-30	9:000	NULL
17	1567	17/2022-12	11:32:000	Vista	Entregado	7	6	71	Deliver	2022-12-30	9:000	NULL
18	1636	14/2022-12	14:11:000	Distribuc	NULL	7	6	89	inDent	2022-12-30	9:000	NULL
19	1660	16/2022-12	12:16:000	Recep	Recep	7	7	47	Decla	2022-12-30	9:000	NULL
20	1722	17/2022-12	19:25:000	EnvioRe	NULL	1	1	79	EnRen	2022-12-30	0:000	NULL
21	1791	11/2022-12	18:47:000	EnvioRe	NULL	7	7	62	RENDR	2022-12-30	0:000	NULL

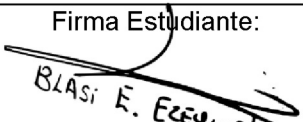
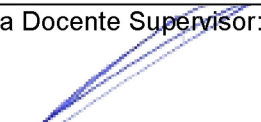
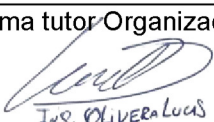
FIGURA 15 - INFORMACIÓN QUE SE PUEDE VISUALIZAR EN LA BASE DE TRAZABILIDAD

Un nuevo servicio "TrackingSync" es una parte integral de la arquitectura de seguimiento y análisis de envíos. Su función principal es recopilar información sobre los envíos y publicarla en un índice de ElasticSearch para su posterior análisis y consulta.

El servicio "TrackingSync" se conecta al worker "ElasticSync" en Redis, que contiene los identificadores de los envíos que tienen nuevas novedades. Estos identificadores individualizan a cada envío, y con ellos se puede buscar las novedades además toda información adicional que el mismo contenga.

Con los identificadores el servicio realiza consultas en la base de datos SQL para recuperar la información completa de cada envío. Esto incluye eventos, ubicaciones, estados y cualquier otro dato relevante asociado a los envíos.

Una vez que se recopila la información de los envíos, el servicio "TrackingSync" procede a armar documentos estructurados que representan cada envío y todas sus

<p>Firma Estudiante:</p>  <p>BLAS E. ECEGUET</p>	<p>Firma Docente Supervisor:</p>  <p>J.R. OLIVERA LUCAS</p>	<p>Firma tutor Organizacional:</p>  <p>J.R. OLIVERA LUCAS</p>
---	--	--

novedades. Estos documentos se preparan en un formato adecuado para su indexación en ElasticSearch.

Los documentos generados se envían a ElasticSearch en forma de operaciones de índice "bulk". Esto significa que se agrupan múltiples documentos en una sola solicitud de índice para mejorar la eficiencia y el rendimiento de la indexación en ElasticSearch.

ElasticSearch almacena los documentos indexados en un índice específico diseñado para el seguimiento de envíos. Este índice facilita la búsqueda, el análisis y la visualización de datos relacionados con los envíos en tiempo real.

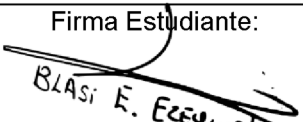
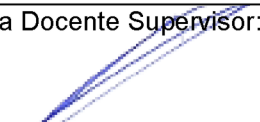
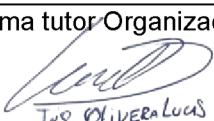
Una vez que los documentos se han indexado en ElasticSearch, la información sobre los envíos y sus novedades está disponible en tiempo real para su consulta y análisis. Los usuarios pueden realizar búsquedas, generar informes y obtener una visión completa y actualizada del estado y seguimiento de los envíos.

Causas que derivan en una migración

El servicio comenzó su primera etapa de procesamiento de mensajes con IBM MQ como su servicio de mensajería principal. Sin embargo, desde el principio, se enfrentó a dificultades notables, principalmente relacionadas con el desbordamiento de colas que resultó en la pérdida y retransmisión de eventos para recuperar la integridad de los datos. Estas dificultades se volvieron cada vez más problemáticas a lo largo del tiempo.

En consecuencia, en los últimos años, la dirección gerencial tomó la decisión de cerrar el contrato con el servicio de mensajería IBM MQ. Esta decisión fue motivada por una serie de factores clave:

- Costo en el mantenimiento y uso de IBM MQ se volvió costoso en comparación con alternativas más modernas ya que su licencia anual era de 300000 dólares, y no ofrecía las opciones de rendimiento que el negocio necesitaba en este momento.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

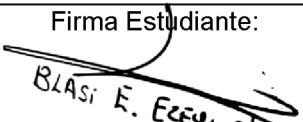
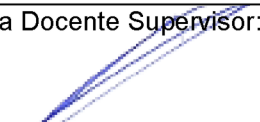
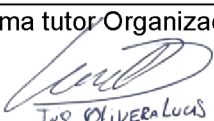
- Escalabilidad Limitada, IBM MQ demostró limitaciones en cuanto a escalabilidad, lo que dificultaba la gestión de un creciente volumen de mensajes y eventos.
- Estrategia Tecnológica, la empresa estaba evolucionando su estrategia tecnológica y requería una solución de mensajería más alineada con las tecnologías y herramientas modernas que estaban adoptando.

Esta decisión de migrar de IBM MQ a una solución alternativa refleja el compromiso de la empresa en mejorar la eficiencia, la confiabilidad y la escalabilidad de su sistema de mensajería, así como su adaptación a las tendencias tecnológicas actuales. Esta migración busca superar los desafíos experimentados previamente y establecer una base más sólida para la gestión de eventos y mensajes en el futuro.

El proceso existente involucraba una serie de etapas complejas. Primero, los mensajes se transferían desde IBM MQ a Redis, luego pasaban por un proceso de procesamiento en una plataforma heredada como .NET Framework 4.5.2 y finalmente se almacenaban en una base de datos SQL. Para luego ser procesados nuevamente por otro servicio que se comunicaba con la base de datos para recuperar toda la información y con ella armar un documento y publicarlo en forma de lote en ElasticSearch. Esta estructura multietapa añadía una capa de complejidad al sistema y aumentaba significativamente la posibilidad de fallos en cualquier punto del proceso.

Además, la falta de soporte para .NET Framework 4.5.2, una plataforma en la que se basaba parte del proceso se convirtió en un problema crítico. El área de arquitectura de la empresa ya no brindaba soporte para esta versión, y la organización estaba priorizando la adopción de .NET 6. Esto generaba desafíos significativos en términos de mantenimiento y evolución del servicio existente, ya que no estaba alineado con las tecnologías y estándares modernos.

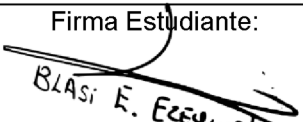
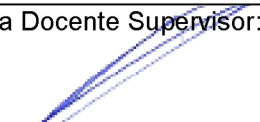
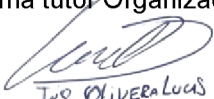
La detección y corrección de errores también se veía obstaculizada debido a la naturaleza fragmentada del proceso de mensajería y la cantidad de etapas involucradas. Esta dificultad en la detección de problemas podría tener un impacto

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUCAS
--	--	--

negativo en la confiabilidad del servicio, lo que hacía necesario revisar la arquitectura y simplificarla.

La necesidad de integrar mejoras y mantener el servicio en un entorno que no era compatible con las herramientas y tecnologías actuales, como .NET 6, generaba conflictos y retrasos en el desarrollo, lo que limitaba la capacidad de la empresa para mantenerse ágil y competitiva en un entorno empresarial en constante cambio.

Por último, la migración a Kafka se consideraba una solución que ofrecía mejoras sustanciales en términos de eficiencia y escalabilidad. Esta transición permitiría a la empresa gestionar de manera más efectiva un creciente volumen de mensajes, simplificar la arquitectura y reducir la complejidad inherente al proceso de mensajería. Esto, a su vez, respaldaría la capacidad de la empresa para adaptarse y crecer en un mercado dinámico.

Firma Estudiante:  BLAS E. Echeverri	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

Funcionamiento actual del sistema de seguimiento de envíos

El servicio ha evolucionado significativamente y se ha consolidado en una sola aplicación basada en .NET Core 6, aprovechando las últimas innovaciones del sector de arquitectura. Esta aplicación está estrechamente integrada con el servicio de Kafka, lo que nos permite consumir mensajes de manera asíncrona, mejorando la eficiencia y la capacidad de respuesta de nuestro sistema.

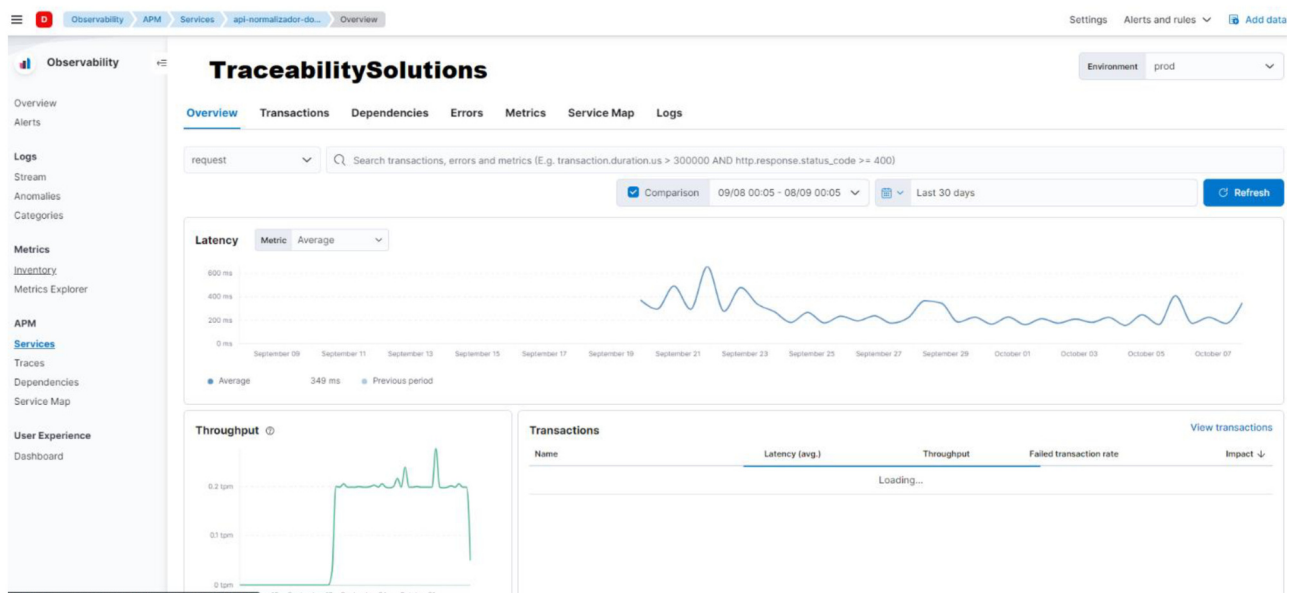
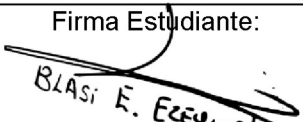
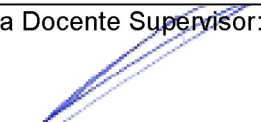
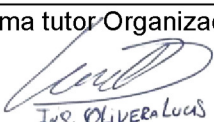


FIGURA 16 - SERVICIO DE OBSERVABILIDAD PARA DETECCIÓN DE FALLOS

Una de las características destacadas de esta nueva arquitectura es la implementación de un mecanismo de respaldo directamente en el servicio de mensajería. Esto significa que, en caso de que la ejecución de uno de los hilos que procesa un evento falle, el sistema cuenta con un mecanismo automático para respaldar la operación.

Referencias [07] – Application Performance Monitoring

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  JOS. OLIVERA LUCAS
---	--	--

Este enfoque de respaldo garantiza una mayor robustez y continuidad en el procesamiento de eventos, reduciendo al mínimo los tiempos de inactividad no planificados.

Además, se ha implementado una capacidad mejorada de detección rápida de fallos. Esto significa que, en caso de que se produzca un fallo en la ejecución de cualquier parte del proceso, el sistema es capaz de identificar el problema de manera inmediata y tomar medidas para corregirlo. Esta capacidad de detección y respuesta rápida asegura que los problemas se aborden de manera eficiente, minimizando el impacto en el servicio y reduciendo la probabilidad de interrupciones prolongadas.

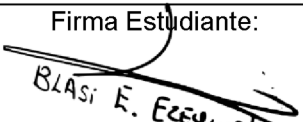
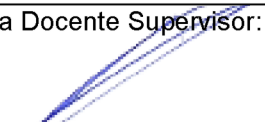
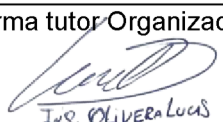
Circuit Breaker

La implementación de un "Circuit Breaker" en el consumo de eventos del sistema de envío es una estrategia fundamental para garantizar la eficiencia, la detección temprana de errores y la satisfacción del cliente. Esta técnica es particularmente importante en arquitecturas de microservicios y sistemas distribuidos donde la comunicación entre componentes puede ser propensa a errores y fallos.

Se utiliza para evitar la degradación del servicio y las demoras significativas en la respuesta del sistema. Cuando se detecta un número excesivo de fallos en la comunicación con un servicio externo, el "Circuit Breaker" se activa y corta temporalmente la comunicación con ese servicio. Esto garantiza que las solicitudes futuras no se queden esperando indefinidamente y permite que el sistema responda rápidamente, incluso si hay problemas con uno de los servicios.

Está diseñado para monitorear constantemente la salud y la disponibilidad de los servicios externos. Cuando detecta un aumento en la tasa de errores o tiempos de respuesta anormalmente largos, se activa y corta la comunicación con ese servicio.

Referencias [08] – Circuit Breaker

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
---	--	--

Esta detección temprana de errores permite que el equipo de desarrollo tome medidas inmediatas para solucionar los problemas subyacentes antes de que se propaguen y afecten negativamente a otros componentes del sistema.

Uno de los beneficios más evidentes de usar un "Circuit Breaker" es la mejora de la experiencia del cliente. Al evitar demoras prolongadas y tiempos de espera innecesarios, los usuarios experimentan un servicio más rápido y confiable. Esto contribuye a la satisfacción del cliente y puede aumentar la confianza en la plataforma.

Cuando un servicio externo experimenta problemas, como una sobrecarga, puede provocar que las solicitudes se acumulen en su cola, lo que afecta negativamente a la latencia y el rendimiento general. El "Circuit Breaker" evita que se realicen más solicitudes a un servicio que ya está saturado, evitando así una mayor sobrecarga y garantizando la disponibilidad de recursos para otras partes del sistema.

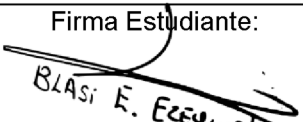
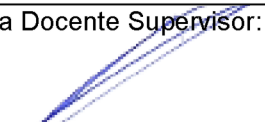
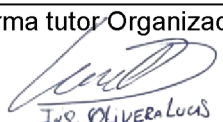
Es un componente importante en la estrategia de tolerancia a fallos de un sistema. Al aislar servicios con problemas, se evita que los errores se propaguen y afecten a otros componentes. Esto permite que el sistema continúe funcionando de manera eficaz, incluso en presencia de fallos transitorios.

El "Circuit Breaker" también es útil durante tareas de mantenimiento y actualizaciones. Puede ser configurado para redirigir el tráfico hacia un servicio en funcionamiento, lo que permite realizar actualizaciones sin interrumpir los servicios para los usuarios.

Using Script ElasticSearch

Para mejorar la performance y la capacidad de búsqueda de eventos de trazas de envío, se implementa la comunicación con ElasticSearch en una sola llamada. Esto permite la actualización o creación de envíos y la agregación de eventos de manera personalizada en ElasticSearch. Los componentes clave serán:

Cliente de ElasticSearch: Se utilizará un cliente de ElasticSearch para interactuar con el motor de búsqueda.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J.R. OLIVERA LUCAS
---	--	--

Actualización o Creación de Envíos: Se diseña una lógica que permita actualizar envíos existentes o crear nuevos envíos en ElasticSearch, todo en una sola operación atómica.

Agregación de Eventos Personalizada: Se proporciona una funcionalidad personalizada para agregar eventos específicos a los envíos en ElasticSearch. Esto podría incluir la ejecución de scripts para procesar y enriquecer los eventos antes de almacenarlos.

La comunicación eficiente con ElasticSearch en una sola llamada simplifica las operaciones relacionadas con los envíos y permitirá una búsqueda más rápida y precisa de eventos de trazas de envío.

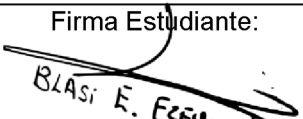
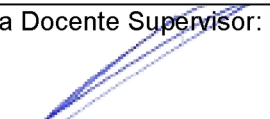
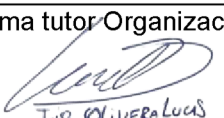
La inserción de datos en ElasticSearch se ha optimizado significativamente mediante este script inicial que carga todos los procedimientos necesarios. Esta estrategia permite realizar la inserción en una sola transacción, lo que se traduce en una comunicación más rápida y eficiente. Esta mejora es esencial para mantener la velocidad y la capacidad de respuesta del sistema.

Escalabilidad en despliegue

La infraestructura seleccionada de Open Shift desempeña un papel fundamental en la capacidad del sistema para visualizar trazas casi en tiempo real, sin demoras ni errores en el procesamiento. A través de la implementación de Kubernetes y Open Shift, se logra una escalabilidad dinámica que permite aumentar la cantidad de pods que contienen la aplicación de manera automática o bajo demanda. Los aspectos clave incluyen:

Orquestación con Kubernetes

Open Shift se basa en Kubernetes para orquestar y administrar contenedores, lo que facilita la creación, escalabilidad y administración de pods de aplicación.

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUCAS
--	--	--

Auto escalabilidad Horizontal (Horizontal Pod Autoscaling): Se configurará la auto escalabilidad horizontal para que, en momentos de alta demanda, se creen automáticamente nuevos pods de aplicación y se distribuyan las trazas de manera eficiente.

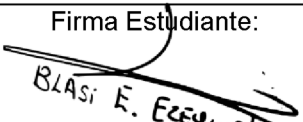
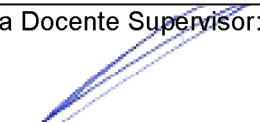
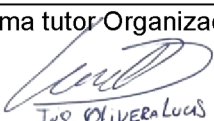
Balanceo de Carga: Open Shift proporciona funcionalidades de balanceo de carga para distribuir el tráfico de manera uniforme entre los pods de la aplicación, lo que evita la sobrecarga de un pod específico.

Monitorización y Métricas: Se utilizarán herramientas de monitorización y recopilación de métricas para supervisar el rendimiento de los pods y garantizar que se puedan realizar ajustes de escalabilidad en función de las métricas en tiempo real.

La infraestructura de OpenShift garantiza que la aplicación pueda adaptarse dinámicamente a las fluctuaciones en la carga de trabajo, lo que resulta en una visualización casi en tiempo real de las trazas de envío sin degradación del rendimiento ni demoras en el procesamiento.

El servicio ha experimentado una transformación significativa y se ha adaptado a las mejores prácticas y tecnologías actuales. La integración con Kafka, la implementación de respaldo y la capacidad de detección de fallos han fortalecido la confiabilidad y la eficiencia del servicio, garantizando una operación más fluida y robusta en un entorno empresarial en constante evolución.

Tras un análisis exhaustivo de los eventos, se identificó que la mayoría de los datos necesarios se pueden obtener directamente del cuerpo de los eventos. Esta optimización ha llevado a una notable reducción en las consultas a servicios externos, disminuyendo la latencia, evitando puntos de falla en servicios periféricos y mejorando la eficiencia general del sistema.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
---	--	---

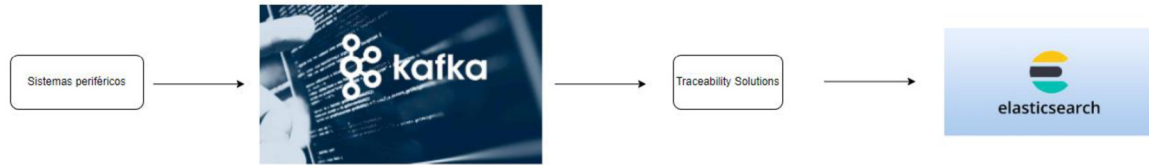


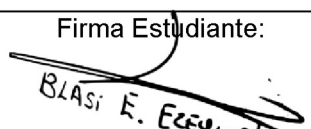
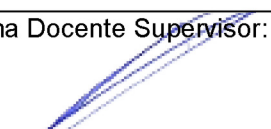
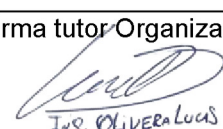
FIGURA 17 - DIAGRAMA DE FLUJO



FIGURA 18 - VISITA A SUCURSALES Y PLANTAS OPERATIVAS

También se incorporó por parte del equipo de desarrollo visitas a sectores operativos como sucursales o plantas de procesamiento con el compromiso de una mejora continua que se refleja en las observaciones críticas de las operaciones que se llevan a cabo.

Estas visitas permitieron un estudio profundo de los procesos operativos con la posibilidad de identificar posibles mejoras y abordar debilidades, entre otros aspectos. La incorporación de estas mejoras ha contribuido a una mayor eficacia en la operación general del servicio.

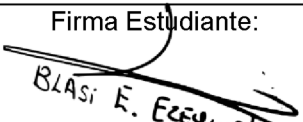
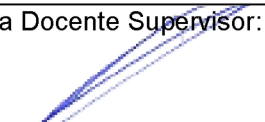

Firma Estudiante:  BLAS E. Eceguet	Firma Docente Supervisor: 	Firma tutor Organizacional:  J.R. OLIVERA LUCAS
---	--	--

Estas acciones reflejan un enfoque proactivo hacia la mejora continua y la eficiencia operativa en el servicio. La reducción de las consultas a servicios externos, las visitas a sectores operativos y la optimización en la inserción de datos son ejemplos claros de cómo el servicio se ha adaptado y fortalecido para ofrecer un rendimiento superior y una experiencia más sólida para los usuarios.

La realización de recorridos por los sectores operativos desempeña un papel fundamental en la eficiencia y el funcionamiento del negocio. Estos recorridos no solo permiten un mejor entendimiento de los procesos logísticos, sino que también han facilitado la identificación de patrones y la generalización de un esquema genérico y altamente adaptable. Este esquema ha demostrado ser de vital importancia, ya que posibilita el manejo eficiente de los 294 procesos diversos que pueden surgir en la operación diaria de la empresa para cada envío.

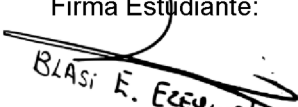
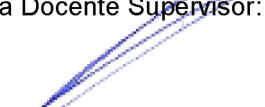

En un entorno donde se procesan y gestionan alrededor de 19 millones de eventos diariamente, la necesidad de un sistema unificado y consistente para acceder a la traza de cada envío se convierte en una prioridad. Los usuarios, ya sean clientes, empleados de la empresa o socios comerciales, deben tener la capacidad de rastrear y supervisar sus envíos de manera única y sencilla. Es en este contexto que el esquema genérico desarrollado a partir de la generalización de los recorridos operativos se erige como un elemento clave.

Este esquema genérico, al ser altamente adaptable tanto en su almacenamiento como en el procesamiento a nivel código, se convierte en la base sobre la cual se pueden gestionar los diversos procesos de envío que se presentan en la operación diaria. Al abordar de manera consistente y eficiente los 294 procesos potencialmente diferentes, se asegura la uniformidad y la coherencia en el seguimiento de los envíos. Esto, a su vez, garantiza que los usuarios puedan acceder a su traza de manera unívoca y sin complicaciones, independientemente de la complejidad de los procesos de envío involucrados.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

La capacidad de la empresa para ofrecer a sus clientes un acceso claro y directo a la información de seguimiento se convierte en un diferenciador competitivo significativo. Los clientes valoran la transparencia y la capacidad de rastreo de sus envíos, lo que a su vez contribuye a la satisfacción del cliente y a la confiabilidad del servicio.

En resumen, la identificación de patrones y la generalización de un esquema adaptable han permitido a la empresa abordar con éxito la diversidad de los procesos de envío que se encuentran en constante evolución. Esto, a su vez, garantiza que los usuarios puedan acceder en tiempo real a la traza de sus envíos en medio de la abrumadora cantidad de eventos que se procesan diariamente. Este logro no solo mejora la eficiencia operativa de la empresa, sino que también refuerza su compromiso con la excelencia en el servicio al cliente.

Firma Estudiante:  BLAS E. Eceyurel	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUCAS
--	--	---

Conclusiones

Reflexión acerca del desarrollo

Para iniciar con las conclusiones se toma como punto de partida los objetivos propuestos en la presente Práctica Profesional Supervisada y que dan cuenta que durante el avance de la práctica los mismos fueron alcanzados. La migración resultante pasó por las diversas etapas de investigación, prueba y puesta en marcha donde ya se pudo visualizar la mejora significativa en varios aspectos.

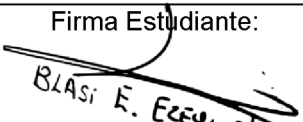
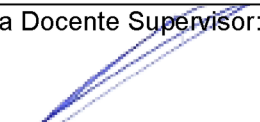
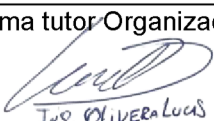
Uno de los logros más destacados de este proyecto fue la implementación exitosa de los principios SOLID en la arquitectura del sistema. Estos principios permitieron una mayor modularidad y mantenibilidad del código, lo que se tradujo en una base más sólida para futuras expansiones y mejoras en el servicio de trazas de envío.

La adopción de la arquitectura limpia y la distribución por capas también se mostraron fundamentales para la organización del sistema. La separación de preocupaciones y la inversión de dependencias contribuyeron a un diseño más claro y a una mejor gestión de las interacciones entre los componentes.

La implementación de la comunicación asíncrona a través de Kafka añadió una capa adicional de eficiencia al sistema, permitiendo el procesamiento de eventos de trazas de envío de manera más ágil y sin bloqueos. La capacidad de escalar automáticamente los pods en OpenShift fue esencial para garantizar que el sistema pudiera responder a picos de carga y visualizar las trazas casi en tiempo real.

Además, la integración con Elasticsearch en una sola llamada permitió una búsqueda más rápida y una agregación de eventos personalizada, mejorando la eficiencia en el manejo de datos y enriqueciendo la capacidad de búsqueda.

El desarrollo de código genérico y el enfoque adoptado en el desarrollo de este sistema de seguimiento de trazas no solo han mejorado significativamente la eficiencia

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  Tutor OLIVERA LUIS
---	--	--

operativa de la empresa, sino que también han establecido una base sólida para su escalabilidad horizontal. Este sistema, concebido a partir de la generalización de los recorridos operativos, está diseñado para crecer en consonancia con las necesidades de la empresa de correo y logística.

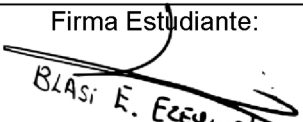
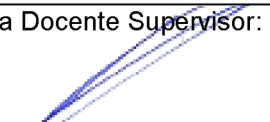
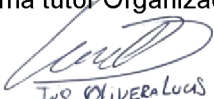
La escalabilidad horizontal que ofrece este enfoque implica que todos los sectores de la empresa pueden beneficiarse de un seguimiento de trazas consistente, independientemente de su escala y complejidad. A medida que la empresa crece y evoluciona, la infraestructura y la lógica subyacente de este sistema pueden adaptarse fácilmente para dar cabida a una mayor demanda y a la incorporación de nuevos procesos de envío.

Esta escalabilidad no solo contribuirá a una mayor eficiencia y capacidad de respuesta en toda la empresa, sino que también consolidará su posición como líder en la industria de correo y logística. Los clientes, los empleados y los socios comerciales seguirán disfrutando de un acceso unificado y sencillo a la información de seguimiento de envíos, respaldando así la satisfacción del cliente y la excelencia en el servicio.

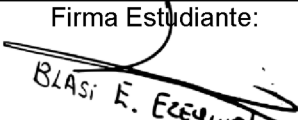
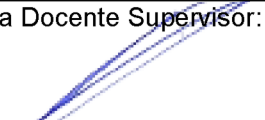

La implementación de este sistema y su escalabilidad horizontal representan un paso significativo en la transformación digital de la empresa, asegurando su capacidad para adaptarse y prosperar en un entorno empresarial en constante cambio.

La eficiencia y rendimiento de este sistema lo posicionan como un modelo a seguir para el desarrollo futuro de servicios de monitoreo de trazas. Su capacidad para adaptarse a las necesidades cambiantes de la empresa, así como su capacidad para manejar sin esfuerzo el crecimiento y la incorporación de nuevos procesos de envío, garantizan una respuesta ágil y eficaz a medida que la empresa avanza.

En resumen, esta Práctica Profesional Supervisada no solo logró cumplir con los objetivos planteados, sino que también brindó valiosas lecciones sobre cómo aplicar principios de diseño sólidos, arquitectura limpia y tecnologías modernas de manera efectiva en un entorno de migración de servicios. El conocimiento y la experiencia

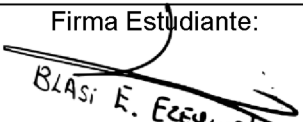
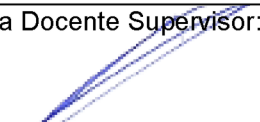
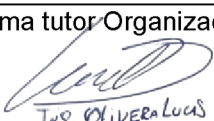
Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUCAS
---	--	--

adquiridos en este proyecto sin duda serán de gran valor en futuros desafíos profesionales.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. S. OLIVERA LUCAS
---	--	---

Referencias bibliográficas

1. Apache Kafka
<https://kafka.apache.org/documentation/>
<https://www.redhat.com/es/topics/integration/what-is-apache-kafka>
 2. .NET Core 6
<https://learn.microsoft.com/es-es/dotnet/core/compatibility/6.0>
 3. Elasticsearch
<https://www.elastic.co/es/elasticsearch>
<https://aws.amazon.com/es/what-is/elasticsearch/>
 4. Scrum Máster
<https://asana.com/es/resources/scrum-master>
<https://proyectosagiles.org/que-es-scrum/>
 5. Clean Code
<https://elpythonista.com/analisis-clean-code>
 6. SonarQube
<https://sentry.io/blog/que-es-sonarqube/>
 7. Application Performance Monitoring (APM)
<https://www.elastic.co/es/observability/application-performance-monitoring>
 8. Circuit Breaker
<https://keepcoding.io/blog/que-es-el-patron-circuit-breaker/>
- ✚ Robert C. Martins Series, (2008), "Clean code, a handbook of agile software craftsmanship"
 - ✚ Vaughn Veron, (2016), "Domain-Driven Design Distilled"
 - ✚ Pressman, Roger, (2002), "Ingeniería del software, Un enfoque práctico". McGraw Hill 2002.
 - ✚ Sommerville, Ian, (2005), "Ingeniería del software". Pearson Educación. Madrid.

Firma Estudiante:  BLAS E. ECEGUET	Firma Docente Supervisor: 	Firma tutor Organizacional:  J. OLIVERA LUIS
---	--	---