



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

## Gestión de Golden Record

Autor/es

Ibón Cañete Bielsa

Director

Daniel Domínguez Guillen

Ponente

Francisco Javier López Pellicer

Escuela de Ingeniería y Arquitectura  
2023

# Agradecimientos

Me gustaría expresar mi sincero agradecimiento a Hiberus por su invaluable contribución en la realización de este trabajo. Su apoyo y colaboración han sido esenciales para hacer posible la culminación exitosa de este proyecto y para enriquecer mi experiencia en el ámbito profesional. Este Trabajo de Fin de Grado no solo representa un logro personal, sino también un paso significativo hacia mi futuro profesional, y esto no habría sido posible sin la confianza y el respaldo de Hiberus.

Asimismo, quiero extender mi gratitud al profesor Francisco Javier López Pellicer por su dedicación y orientación durante todo el proceso. Su seguimiento cercano y sus valiosos consejos han sido fundamentales para dar forma y dirección a este trabajo. La guía que me ha proporcionado no solo ha enriquecido mi comprensión de los conceptos involucrados, sino que también ha contribuido a mi crecimiento académico y personal.

Sin la colaboración y el apoyo de Hiberus y la guía del profesor Francisco Javier López Pellicer, este logro no habría sido posible. Espero que este trabajo no solo sea un reflejo de mis esfuerzos, sino también una muestra de mi agradecimiento por la ayuda inestimable que he recibido en este trayecto.

# Resumen

En este trabajo, se aborda el desafío que enfrenta Hiberus al gestionar información diversa en su entorno organizativo. Para solucionar este problema, se propone la implementación de un sistema basado en el concepto de “Golden Record”, que permitirá unificar datos de múltiples fuentes y proporcionar una visión precisa y completa de la información.

Un “Golden Record” (Registro Único) es una representación maestra de datos que se considera la fuente de verdad de una entidad o entidad de datos en un sistema. Contiene información precisa y actualizada, evitando duplicaciones y errores, y se utiliza para garantizar la consistencia y calidad de los datos en una organización. Este registro único actúa como la referencia central para la toma de decisiones y el análisis de datos, ayudando a mantener la integridad de la información en entornos empresariales.

Se lleva a cabo una evaluación exhaustiva de los posibles riesgos y obstáculos que podrían surgir durante la implementación de este sistema. Además, se realiza un análisis detallado de diferentes arquitecturas con el objetivo de seleccionar la más adecuada para el desarrollo del sistema de “Golden Record”. Se consideran criterios como el control centralizado, la coherencia de datos, la escalabilidad y la eficiencia en la integración y consulta de datos.

En términos de tecnologías, se exploran diversas opciones para la creación y consumo del “Golden Record”. Se identifican las dificultades que surgieron durante el proceso de implementación, y se describen las medidas y estrategias utilizadas para mitigar estos desafíos.

Finalmente, se establece una visión a futuro del trabajo, indicando posibles direcciones y áreas de mejora en relación con el sistema de “Golden Record”. Esto implica la continua optimización de la arquitectura y tecnologías empleadas, así como la adaptación a las cambiantes necesidades y desafíos de la organización.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Golden Record y su Relevancia Empresarial . . . . .	2
1.3. Objetivo del trabajo . . . . .	3
1.4. Obstáculos y riesgos . . . . .	3
1.5. Metodología . . . . .	4
1.6. Esfuerzo y Tiempo dedicado . . . . .	5
<b>2. Análisis del Sistema</b>	<b>7</b>
2.1. Arquitecturas para un Golden Record . . . . .	7
2.2. Proceso de elección de la arquitectura . . . . .	9
2.3. Arquitectura escogida . . . . .	9
2.4. Creación y Gestión de un Golden Record . . . . .	11
2.5. Diseño del modelo común . . . . .	12
2.6. Extracción de datos . . . . .	12
2.6.1. Basado en Eventos . . . . .	13
2.6.2. Polling . . . . .	14
2.6.3. Elección . . . . .	14
2.7. Tratamiento de los datos extraídos . . . . .	15
2.7.1. Procesos de detección y resolución de conflictos . . . . .	15
2.7.2. Proceso de carga . . . . .	16
2.8. Solución como plantilla . . . . .	16
2.9. Conclusiones . . . . .	17
<b>3. Diseño del Sistema</b>	<b>18</b>
3.1. Arquitectura del Sistema . . . . .	18
3.2. Solución como plantilla . . . . .	21
3.2.1. Fuente de Datos . . . . .	21
3.2.2. Conectores . . . . .	22
3.2.3. Topics . . . . .	22
3.3. Front-End . . . . .	23
<b>4. Implementación y pruebas</b>	<b>25</b>
4.1. Componentes . . . . .	25
4.1.1. Canonizador . . . . .	25
4.1.2. Unificador . . . . .	26
4.1.3. Calidad y Canonización . . . . .	26
4.1.4. Golden Record CRUD . . . . .	26
4.2. Kafka . . . . .	27

4.3. Conectores . . . . .	27
4.4. Monitorización . . . . .	28
4.5. Seguridad . . . . .	28
4.5.1. Autenticación . . . . .	28
4.5.2. Copia de Seguridad . . . . .	29
<b>5. Prueba de concepto</b>	<b>30</b>
5.1. Definición de las fuentes de datos . . . . .	30
5.2. Golden Record y definición de las reglas de canonización . . . . .	31
5.3. Creación de los conectores . . . . .	33
5.4. Carga de datos y cómo se comporta el Golden Record . . . . .	33
5.4.1. Comportamiento del Golden Record . . . . .	34
5.5. Ejemplo de uso del Frontal y Monitorización . . . . .	34
<b>6. Conclusiones y Trabajo Futuro</b>	<b>35</b>
6.1. Conclusiones . . . . .	35
6.2. Trabajo a Futuro . . . . .	36
<b>Anexos</b>	<b>41</b>
<b>A. Ejemplo configuración de un Source Connector</b>	<b>41</b>
<b>B. Reglas de Canonización del proyecto de ejemplo</b>	<b>43</b>
<b>C. Comportamiento</b>	<b>44</b>
<b>D. Requisitos</b>	<b>46</b>
D.1. Requisitos Funcionales . . . . .	46
D.2. Requisitos No Funcionales . . . . .	47
<b>E. Tableros Kanban</b>	<b>48</b>
<b>F. Ejemplo de Uso del Frontal</b>	<b>50</b>
F.1. Monitorización . . . . .	52
<b>G. Tiempo dedicado</b>	<b>54</b>

# 1 Introducción

Este Trabajo de Fin de Grado (TFG) se enfocará en comprender y analizar la gestión de un “Golden Record”, explorando diversas arquitecturas para su implementación. Se abordará la importancia de la integración de datos, la resolución de conflictos y la actualización continua, con el objetivo de proporcionar recomendaciones concisas para mejorar su comercialización y maximizar su impacto en diversas áreas empresariales.

## 1.1. Contexto

En el mundo actual, las organizaciones se enfrentan a un reto importante: manejar información de diversas fuentes. En una empresa los datos provienen de diferentes lugares, como bases de datos y plataformas en línea. Esto puede generar el problema de cómo reunir todos esos datos en un solo lugar para obtener una vista completa y precisa. Aquí es donde entra en juego el concepto de “Golden Record”. En castellano se diría “Registro de oro”, pero ese término no se utiliza comúnmente en el ámbito técnico. En lugar de ello, nos referimos a este concepto como “Golden Record” en inglés, una expresión que ha ganado aceptación global en el contexto de la gestión de datos y la integración de información proveniente de diversas fuentes.

Un “Golden Record” es una única y bien definida versión de todas las entidades de datos en un ecosistema organizativo. En este contexto, a veces se le llama al “Golden Record” la “única versión de la verdad”, donde “verdad” se entiende como la referencia a la cual los usuarios de datos pueden recurrir cuando desean asegurarse de que tienen la versión correcta de una pieza de información [1].

Cada fuente de datos tiene su propio formato y estructura. Esta variedad de formatos hace que sea complicado comparar y unificar la información. Además, es probable que algunos datos estén desactualizados o incorrectos, lo que hace que tomar decisiones basadas en ellos sea arriesgado.

Dentro del marco de colaboración en mi TFG con la empresa Hiberus, se aborda la implementación de un sistema de Golden Record. Esta iniciativa se alinea con un enfoque similar al concepto de “solución white-label” [2]. Aunque no se trata de una aplicación o plataforma en sí, el propósito es desarrollar una estructura versátil y adaptable. Esta estructura tiene como objetivo capacitar a Hiberus para crear soluciones de Golden Record personalizadas para diversas empresas. En lugar de ser una aplicación cerrada, estamos construyendo un marco robusto que puede ser modelado y ajustado según las necesidades específicas de cada cliente. Esta flexibilidad y adaptabilidad nos permitirán ofrecer soluciones de Golden Record eficientes y a medida para una variedad de empresas y entornos, proporcionando así una respuesta precisa a los desafíos y requisitos únicos de cada cliente.

Siguiendo la misma filosofía, esta estructura se convertirá en una “plantilla” que, una vez personalizada, podrá implementarse en diferentes contextos empresariales. Este enfoque tiene como objetivo agilizar la creación de soluciones de Golden Record a medida, reduciendo tiempos y costos

de desarrollo, al tiempo que brinda a Hiberus la capacidad de ofrecer un producto sólido y eficiente a sus clientes.

## 1.2. Golden Record y su Relevancia Empresarial

En el entorno empresarial, donde los datos fluyen desde diversas fuentes, es común que los mismos registros se encuentren dispersos en diferentes lugares. Este fenómeno da lugar a datos duplicados y desactualizados, lo que puede causar confusión y errores. Sin embargo, la implementación de un Golden Record ofrece una solución efectiva. El Golden Record tiene la capacidad de identificar y unificar estos registros repetidos, creando una única fuente confiable donde la información está consolidada y actualizada.

A continuación, exploraremos ejemplos de su aplicación en diferentes contextos. Estos ejemplos ilustran cómo esta herramienta se convierte en un elemento fundamental para optimizar procesos y promover la consistencia en toda la organización.

- La información que proviene de diferentes fuentes a menudo puede ser divergente o contradictoria, dando lugar a inconsistencias en los datos que dificultan la obtención de una imagen precisa. Un Golden Record entra en acción para resolver este problema. Al fusionar los datos de diversas fuentes, el Golden Record es capaz de presentar una versión coherente y confiable de la realidad, eliminando contradicciones y generando una fuente de verdad unificada.
- La gestión de datos dispersos en diferentes sistemas puede llevar a la falta de coherencia y sincronización. La falta de alineación entre los datos puede resultar en información desactualizada y errores en la toma de decisiones. La solución radica en la implementación de un sistema Golden Record, que reúne todos los datos en un solo lugar. Al consolidar los datos en un esquema, el Golden Record garantiza que la información esté actualizada y coherente, lo que mejora la precisión de las decisiones.
- Cuando una empresa interactúa con sus clientes en múltiples puntos de contacto, como ventas y servicio al cliente, puede ser complicado obtener una vista completa de cada cliente. Aquí es donde el sistema Golden Record desempeña un papel crucial. Al unificar los datos de diferentes interacciones, el Golden Record brinda una visión completa y unificada del cliente en todos los aspectos. Esta visión 360° permite a la empresa entender mejor las necesidades del cliente y personalizar su experiencia.
- En entornos donde se requiere un control riguroso sobre los datos personales y el cumplimiento normativo, como en el caso de las leyes de protección de datos personales (GDPR), un Golden Record puede ser una herramienta esencial. Al mantener los datos personales en una sola fuente de confianza, el Golden Record ayuda a asegurar que se sigan las regulaciones y que los datos se manejen de manera adecuada y precisa.

La duplicación de datos y la falta de coherencia pueden ralentizar los procesos internos y llevar a la ineficiencia operativa. Aquí es donde la metodología Golden Record entra en acción. Al eliminar los datos duplicados y consolidar la información en un solo lugar, el Golden Record agiliza los procesos y reduce la pérdida de tiempo en la búsqueda y reconciliación de información.

### 1.3. Objetivo del trabajo

El objetivo principal de este trabajo de investigación es abordar de manera integral la implementación exitosa de un sistema de Golden Record, enfocándose en garantizar la calidad de los datos, la seguridad y la eficiencia en su administración.

Para lograr este propósito, se establecen los siguientes objetivos específicos:

- **Diseño de la Estructura del Golden Record:** Definir y desarrollar una estructura de datos coherente y eficiente para el Golden Record. Esta estructura deberá ser capaz de albergar información diversa y heterogénea, y a la vez, permitir su consulta y análisis eficaz.
- **Desarrollo de Procesos de Integración:** Diseñar y ejecutar procesos de integración de datos que abarquen la extracción, transformación y carga (ETL) desde las diversas fuentes hasta el Golden Record. Estos procesos deben ser automatizados y optimizados para garantizar la precisión y actualización de los datos.
- **Integración en Tiempo Real:** Implementar un sistema de integración en tiempo real que permita la actualización constante del Golden Record a medida que los datos cambian en las fuentes subyacentes. Esto asegurará que la información sea siempre actual y relevante.
- **Seguridad contra la Pérdida de Datos:** Establecer medidas de seguridad para prevenir la pérdida de datos críticos en el proceso de integración y almacenamiento del Golden Record. Se implementarán procedimientos de respaldo y recuperación de datos para mitigar los riesgos de pérdida.
- **Monitorización del Sistema:** Desarrollar una infraestructura de monitorización para supervisar el rendimiento, la integridad y la disponibilidad del sistema de Golden Record. Esto permitirá identificar y abordar problemas en tiempo real, asegurando un funcionamiento óptimo.
- **Auditoría de Accesos y Cambios:** Establecer un sistema de auditoría que permita revisar de manera retrospectiva todos los cambios y accesos efectuados en el Golden Record. Esto proporcionará un historial completo y verificable de las acciones realizadas en el sistema.
- **Creación de Roles para la Seguridad:** Diseñar y definir roles de usuario que otorguen diferentes niveles de acceso y autorización en función de las responsabilidades y funciones de los usuarios. Esto garantizará que solo personal autorizado pueda acceder y manipular los datos.

Al cumplir con estos objetivos, se busca no solo implementar un sistema de Golden Record funcional, sino también asegurar que sea seguro, eficiente y capaz de mantener la calidad y la integridad de los datos a lo largo del tiempo. Estos aspectos son esenciales para que el sistema sea una herramienta valiosa para la toma de decisiones informadas y estratégicas en la organización.

### 1.4. Obstáculos y riesgos

Las tablas 1.1 y 1.2 reúnen los riesgos u obstáculos más importantes que podrían surgir durante la ejecución del proyecto. Para cada uno de ellos, se ha estimado la probabilidad de que realmente ocurran y se han propuesto medidas para reducir su impacto. La tabla 1.1 destaca la anticipación a posibles desafíos en la implementación y esalta la importancia de un análisis detallado de las fuentes de datos y la automatización de procesos para reducir la complejidad de integración. Además,

subraya estrategias proactivas, como la gestión del cambio y medidas de seguridad robustas, para abordar riesgos potenciales en aspectos tecnológicos y de privacidad. En la tabla 1.2, se presenta una evaluación retrospectiva de los riesgos enfrentados. Se destaca la aplicación exitosa de mitigaciones en áreas críticas como la complejidad de integración y los cambios en los requisitos. Aunque se reconoce una decisión estratégica de no aplicar reglas exhaustivas en proyectos grandes, se resalta la importancia de ajustar el enfoque según las circunstancias específicas para garantizar el éxito continuo del sistema de Golden Record.

<b>Riesgo</b>	<b>Probabilidad de Ocurrencia</b>	<b>Mitigación Inicial Prevista</b>
Complejidad de Integración	Moderada	Realizar un análisis exhaustivo de las fuentes de datos y sus formatos antes de comenzar el proceso de integración. Desarrollar transformaciones y limpiezas de datos escalables y automatizadas para minimizar el esfuerzo manual.
Incompatibilidades Tecnológicas	Alta	Realizar un inventario detallado de las tecnologías y los sistemas por integrar, identificando posibles conflictos de compatibilidad.
Gestión del Cambio	Moderada	Diseñar e implementar un plan de capacitación y comunicación para el personal afectado por el cambio. Identificar y abordar las preocupaciones de los empleados de manera proactiva, demostrando los beneficios y ventajas del nuevo sistema.
Seguridad y Privacidad	Alta	Implementar medidas de seguridad robustas, como cifrado de datos, autenticación de usuarios y control de acceso. Realizar auditorías y pruebas de penetración de forma regular para identificar y corregir posibles vulnerabilidades.
Cambios en los Requerimientos	Moderada	Diseñar el sistema de Golden Record con flexibilidad y modularidad para permitir ajustes y adaptaciones según los cambios en los requerimientos. Mantener una comunicación constante con los usuarios y las partes interesadas para entender y abordar los cambios en tiempo real.
Alcance (Proyecto Grande)	Muy Alta	Identificar los aspectos críticos y prioritarios del proyecto. Realizar una evaluación continua del alcance para detectar posibles desviaciones. En caso de sobrepasar el alcance, considerar la posibilidad de recortar en áreas no esenciales.

Tabla 1.1: Riesgos y Mitigaciones para el Sistema de Golden Record.

## 1.5. Metodología

En nuestro enfoque metodológico de desarrollo, comenzamos por aprovechar los requisitos proporcionados por la empresa como punto de partida. Estos requisitos sirvieron como base para nuestro análisis, donde evaluamos su adecuación y adaptación a la idea y alcance de nuestro TFG. Durante este proceso, se generaron requisitos adicionales para garantizar una alineación precisa con los objetivos del proyecto (ver anexo D).

Riesgo	¿Ha Ocurrido?	Mitigación Aplicada
Complejidad de Integración	Sí	Se realizó un análisis exhaustivo de los datos y se automatizaron procesos mediante <i>scripts</i> para la transformación y limpieza de datos.
Cambios en los Requerimientos	Sí	Se realizó una reevaluación de los requerimientos y se realizaron ajustes en la planificación y el diseño del sistema.
Alcance (Proyecto Grande)	Sí	Se optó por no aplicar reglas de calidad exhaustivas, solo se implementará la regla básica de comprobar la clave primaria antes de realizar actualizaciones.

Tabla 1.2: Ocurrencia de Riesgos y Mitigaciones Aplicadas.

Una vez establecidos y refinados los requisitos, implementamos una metodología de desarrollo que maximizara la eficiencia y el seguimiento de las tareas. Para ello, adoptamos un enfoque Kanban, un marco de trabajo visual que facilita la planificación, la priorización y el monitoreo de las actividades. Para la gestión del backend y el frontend, creamos dos tableros Kanban separados, cada uno enfocado en sus respectivas áreas de desarrollo. Estos tableros sirvieron como paneles de control centralizados, donde categorizamos las funcionalidades en diferentes columnas, como "Pendientes", "En Curso", "Completadas" y "Bloqueadas". Esta representación visual permitió un seguimiento instantáneo de la evolución de cada tarea y su estado (ver anexo E).

## 1.6. Esfuerzo y Tiempo dedicado

Para obtener una visión cuantitativa del esfuerzo invertido en cada fase, se presenta la Tabla 1.3, que resume las horas totales dedicadas a cada etapa del TFG. Este resumen proporciona una visión general del proyecto, desde sus fases iniciales de definición hasta las iteraciones finales que condujeron al resultado definitivo. Se han empleado un total de 370 horas, en las que el mayor impacto ha tenido ha sido el desarrollo del backend debido a su mayor complejidad, invirtiendo 120 horas.

Fase	Horas Totales
Análisis y Diseño	50
Implementación - Backend	120
Implementación - Frontend	50
Pruebas y Proyecto de Ejemplo	50
Correcciones	40
Redacción de la Memoria	60
<b>Total</b>	<b>370</b>

Tabla 1.3: Horas Totales por Fase

El desarrollo del TFG se llevó a cabo en dos fases fundamentales, cada una reflejada en los diagramas de Gantt de las Figuras G.1 y G.2. En el primer mes, se dedicó tiempo exclusivo al análisis y diseño del "Golden Record", estableciendo así los cimientos y requisitos del proyecto. La implementación del "Backend" comenzó posteriormente, respaldada por pruebas continuas y un proyecto de ejemplo para garantizar la correcta funcionalidad.

La segunda fase, abarcando de septiembre a diciembre, se desarrolló simultáneamente el “Backend”, el “Frontend” y la redacción de la memoria. Este enfoque dual permitió avanzar de manera eficiente tanto en la implementación como en la documentación, asegurando coherencia y progresión paralela. Las correcciones se llevaron a cabo de forma constante, basadas en pruebas y retroalimentaciones, culminando en la versión final del proyecto.

## 2 Análisis del Sistema

En el capítulo de Análisis, vamos a profundizar en cómo se crea y utiliza un Golden Record, analizaremos las diferentes arquitecturas disponibles para su implementación, evaluando cuál es la mejor opción. Veremos cómo elegir la tecnología adecuada y analizar cómo aplicarla en el proyecto.

### 2.1. Arquitecturas para un Golden Record

La arquitectura de un sistema de Golden Record puede variar según las necesidades y los requisitos específicos de la organización. A continuación, se detallarán diferentes arquitecturas, las cuales nos permitirán evaluar cuidadosamente las opciones disponibles y seleccionar la arquitectura más adecuada para nuestro sistema. Existen las siguientes arquitecturas entre otras:

**Arquitectura de Almacenamiento Centralizado(Data Warehouse [3]):** En esta arquitectura, todos los datos provenientes de diversas fuentes se extraen, transforman y cargan (ETL) en un almacén centralizado. El sistema de Golden Record accede a este almacén para crear y gestionar el Golden Record. Esta arquitectura permite un control centralizado sobre los datos y facilita la integración y consulta eficiente. Puede ser útil cuando se requiere un alto grado de coherencia y uniformidad en los datos.



Figura 2.1: Data Warehouse

**Arquitectura de Microservicios [4]:** En un enfoque basado en microservicios, cada fuente de datos puede estar asociada con un microservicio que se encarga de la integración y la transformación de los datos. Se tendría el microservicio de canonizador para recibir los datos de las fuentes de datos y hacer los cambios de nombre de campos, el servicio de canonización para crear unas reglas de canonización

que establezcan los campos que tiene o no el golden record y que hará uso el canonizador. Luego, un servicio para la calidad de los mensajes y el unificador para mandar al golden record el resultado final. Finalmente, un servicio centralizado se encarga de consolidar y administrar el Golden Record. Esta arquitectura es escalable y flexible, lo que la hace adecuada para organizaciones con múltiples fuentes de datos y flujos de trabajo complejos.

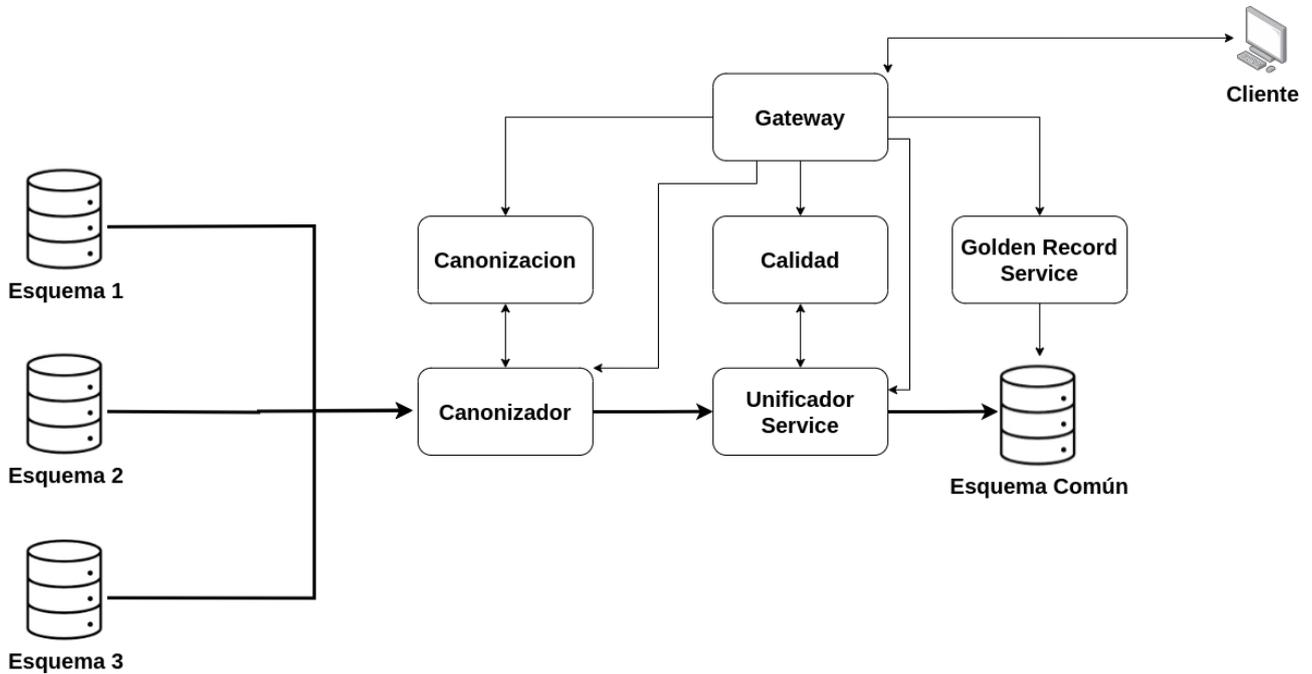


Figura 2.2: Microservicios

**Arquitectura de Almacenamiento Distribuido y Big Data [5]:** Se basa en un enfoque escalable y resiliente. Los datos del “Golden Record”, que representan una fuente única de verdad para la información, se almacenan de manera distribuida en un sistema altamente disponible, como un clúster Hadoop. La capa de procesamiento y análisis de *big data* trabaja en conjunto con este almacenamiento distribuido, permitiendo la extracción eficiente de información valiosa y la detección de patrones significativos. Mediante técnicas de procesamiento paralelo y distribuido, los datos se pueden procesar en lotes o en tiempo real, según las necesidades.

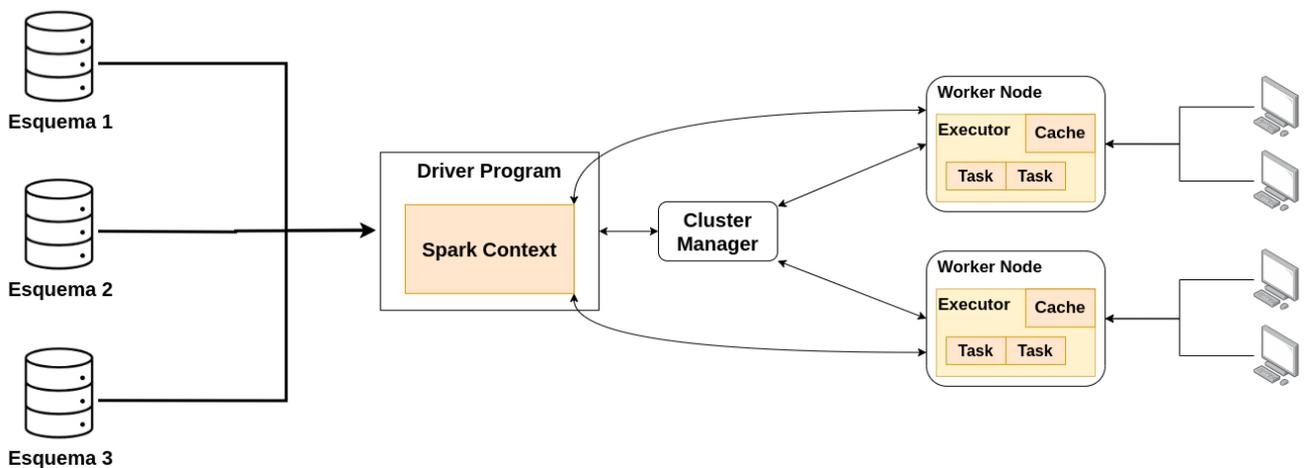


Figura 2.3: Almacenamiento Distribuido y Big Data

## 2.2. Proceso de elección de la arquitectura

Dado el contexto proporcionado, puntuemos las diferentes arquitecturas en función de los criterios relevantes para la implementación de un sistema de Golden Record:

Arquitectura	Control centralizado	Coherencia y uniformidad	Flexibilidad y escalabilidad	Eficiencia en integración y consulta
Almacenamiento Centralizado	Sí	Alta	Baja	Alta
Microservicios	Sí	Moderada	Alta	Moderada
Almacenamiento Distribuido y Big Data	Bajo	Variable	Alta	Moderada

Tabla 2.1: Puntuación de Arquitecturas para Implementar el Sistema de Golden Record

Dado que el objetivo es desarrollar una estructura versátil y adaptable para la creación de soluciones de Golden Record a medida, que sea eficiente en tiempo y costos, y que permita a Hiberus ofrecer un producto sólido, la Arquitectura de Microservicios parece ser una opción prometedora. Esta arquitectura ofrece flexibilidad y escalabilidad, lo que se alinea con el enfoque de crear soluciones personalizadas para diferentes empresas. Además, permite una administración descentralizada de fuentes de datos mientras centraliza la gestión del Golden Record, lo que facilita el desarrollo ágil y la adaptabilidad. Dado que la iniciativa busca un enfoque similar a una “solución white-label”, donde se busca personalizar la solución para cada cliente, la arquitectura de microservicios resulta aún más idónea para esta tarea. La naturaleza modular y desacoplada de los microservicios permitiría una personalización más eficiente y específica para satisfacer las necesidades únicas de cada cliente, sin comprometer la eficacia del sistema general. Si bien no ofrece el mismo grado de coherencia centralizada que la Arquitectura de Almacenamiento Centralizado, esta desventaja es contrarrestada por su capacidad para manejar múltiples fuentes de datos y flujos de trabajo complejos, en perfecta sincronía con los objetivos y la filosofía de desarrollo ágil de la empresa Hiberus.

## 2.3. Arquitectura escogida

Dado que el objetivo es desarrollar una estructura versátil y adaptable para la creación de soluciones de Golden Record a medida, que sea eficiente en tiempo y costos, y que permita a Hiberus ofrecer un producto sólido, la Arquitectura de Microservicios parece ser una opción prometedora. Aquí hay una justificación detallada:

**Enfoque del proyecto:** Dado que el objetivo es desarrollar una estructura versátil y adaptable para la creación de soluciones de Golden Record a medida, que sea eficiente en tiempo y costos, y que permita a Hiberus ofrecer un producto sólido, la Arquitectura de Microservicios parece ser una opción prometedora. Esta arquitectura ofrece flexibilidad y escalabilidad, lo que se alinea con el enfoque de crear soluciones personalizadas para diferentes empresas. Además, permite una administración descentralizada de fuentes de datos mientras centraliza la gestión del Golden Record, lo que facilita el desarrollo ágil y la adaptabilidad. Dado que la iniciativa busca un enfoque similar a una “solución white-label”, donde se busca personalizar la solución para cada cliente, la arquitectura

de microservicios resulta aún más idónea para esta tarea. La naturaleza modular y desacoplada de los microservicios permitiría una personalización más eficiente y específica para satisfacer las necesidades únicas de cada cliente, sin comprometer la eficacia del sistema general. Si bien no ofrece el mismo grado de coherencia centralizada que la Arquitectura de Almacenamiento Centralizado, esta desventaja es contrarrestada por su capacidad para manejar múltiples fuentes de datos y flujos de trabajo complejos, en perfecta sincronía con los objetivos y la filosofía de desarrollo ágil de la empresa Hiberus.

**Modularidad y Desacoplamiento:** La arquitectura de microservicios permite descomponer la funcionalidad en servicios independientes y autónomos, cada uno enfocado en una tarea específica. En el caso de un sistema de Golden Record, donde hay varios aspectos que deben gestionarse de manera separada (captura de cambios, unificación de información, gestión del propio Golden Record, seguridad, entre otros), los microservicios ofrecen la ventaja de modularidad. Esto significa que cada microservicio puede ser diseñado, desarrollado y mantenido de manera individual, lo que facilita la gestión y la evolución de cada componente.

**Flexibilidad en el Desarrollo y Escalabilidad:** La arquitectura de microservicios permite el desarrollo y la implementación de cada servicio de manera independiente. Esto es especialmente beneficioso para el sistema de Golden Record, donde los requisitos y las funcionalidades pueden variar entre los diferentes aspectos (captura, unificación, gestión, seguridad, etc.). Cada microservicio puede ser desarrollado utilizando la tecnología más adecuada para su tarea específica, lo que permite un alto grado de adaptabilidad y eficiencia.

**Escalabilidad Horizontal:** Los microservicios pueden escalarse individualmente según la demanda. En el caso de un sistema de Golden Record, donde ciertos componentes pueden experimentar cargas variables (por ejemplo, la captura de cambios), la escalabilidad horizontal puede ser implementada en los microservicios relevantes. Esto garantiza un rendimiento óptimo incluso en momentos de alta carga.

**Gestión Simplificada de Cambios:** La gestión de cambios se simplifica con microservicios. Si se necesita realizar una actualización o mejora en un componente específico (por ejemplo, la seguridad del sistema), esto se puede hacer sin afectar el funcionamiento de los demás microservicios. Esto reduce los riesgos y facilita la implementación de mejoras de manera ágil.

**Adaptación a Cambios Futuros:** La arquitectura de microservicios es flexible y adaptable a cambios futuros en los requerimientos. A medida que la organización evoluciona y las necesidades cambian, es más fácil incorporar nuevos microservicios o ajustar los existentes para abordar nuevos desafíos.

**Facilita la Innovación y Experimentación:** La estructura de microservicios permite experimentar y adoptar nuevas tecnologías o enfoques en componentes individuales sin afectar el funcionamiento global del sistema.

## 2.4. Creación y Gestión de un Golden Record

Entender cómo se crea y se consume un Golden Record es crucial para analizar las arquitecturas de implementación, ya que proporciona la base para diseñar un sistema que garantice la calidad, consistencia y accesibilidad de los datos. Comprender el proceso de generación del Golden Record permite establecer reglas claras para la recopilación, limpieza y unificación de datos, asegurando que la fuente de verdad sea confiable. Por otro lado, conocer cómo se consume el Golden Record es esencial para determinar cómo se integrará en los sistemas existentes, cómo se compartirá con diferentes equipos y cómo se mantendrá actualizado a lo largo del tiempo. Esta comprensión profunda de la generación y el uso del Golden Record es fundamental para evaluar las arquitecturas adecuadas y tomar decisiones informadas en su implementación, garantizando así una gestión eficaz y beneficiosa de los datos en toda la organización [6].

En cuanto a la creación del golden record se tienen en cuenta siempre estos cinco pasos:

- **Identificación de fuentes de datos:** Se recopilan datos de diversas fuentes internas y externas, como sistemas de gestión de relaciones con clientes (CRM), bases de datos, redes sociales y más (véase en la parte izquierda de la figura 2.4). Estas fuentes pueden contener información duplicada, desactualizada o contradictoria.
- **Creación del esquema canónico:** Para facilitar la comunicación con las fuentes de datos, se establece un esquema canónico. Este esquema define de manera precisa cómo se recopilan, estructuran y representan los datos. Este paso es esencial para lograr consistencia y coherencia en la gestión de datos maestros, permitiendo una integración efectiva y eficiente de información heterogénea.
- **Limpieza de datos:** Los datos recopilados son sometidos a un proceso de limpieza, donde se eliminan duplicados, se estandarizan formatos (como nombres y direcciones) y se corrigen errores tipográficos u otras inconsistencias.
- **Detección y resolución de conflictos:** Cuando hay discrepancias entre los datos de diferentes fuentes, se implementa un proceso para resolver esos conflictos, ya sea a través de reglas predefinidas o intervención humana.
- **Generación del Golden Record:** Utilizando los datos limpios, enriquecidos y las decisiones de resolución de conflictos, se crea el Golden Record, que es la versión unificada y precisa de los datos de la entidad.

Para consumir el Golden Record se hace de la siguiente manera:

- **Integración con sistemas:** El Golden Record se integra en los sistemas relevantes de la organización, como sistemas CRM, sistemas de análisis, plataformas de marketing, etc.
- **Acceso y consulta:** Los equipos y empleados autorizados pueden acceder al Golden Record para obtener una vista única y confiable de la entidad en cuestión. Esto es fundamental para tomar decisiones informadas y brindar un servicio personalizado. Esto se puede apreciar en la parte derecha de la figura 2.4
- **Actualización y sincronización:** A medida que se recopilan nuevos datos, se someten al mismo proceso de limpieza y enriquecimiento antes de incorporarlos al Golden Record. Esto garantiza que la fuente de verdad se mantenga actualizada y precisa.
- **Gestión de cambios:** Cualquier cambio importante en el Golden Record, como una actualización de información crítica, se gestiona de manera controlada para asegurar la coherencia y la calidad de los datos.

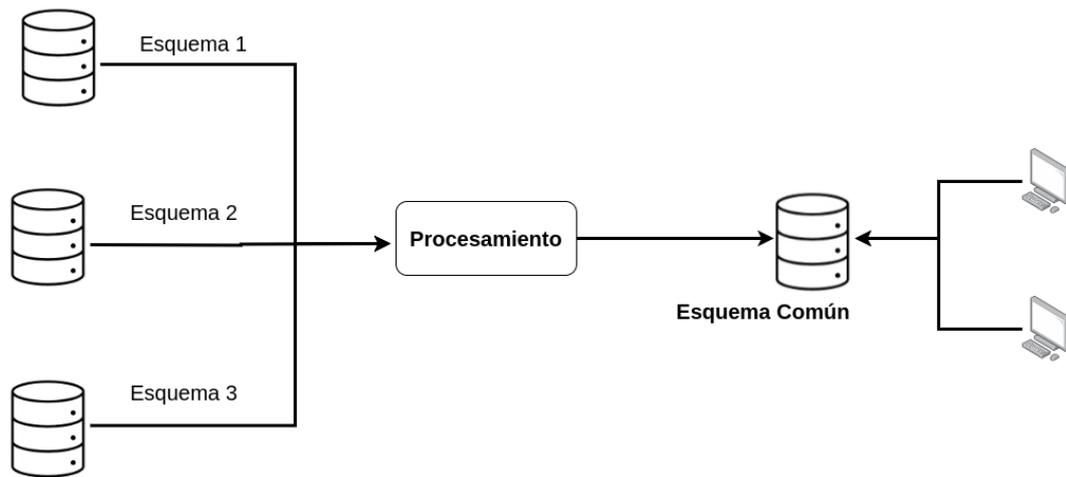


Figura 2.4: Creación y consumo del Golden Record

## 2.5. Diseño del modelo común

El elemento clave para esta integración eficaz es la implementación de una clave unificada que se comparte entre todas las fuentes de datos.

### Clave Unificada

Identificamos una clave principal única que actúa como el vínculo entre cada fuente de datos. Esta clave, presente de manera consistente en todas las fuentes, facilita la integración sinérgica de la información, permitiendo la creación de un Golden Record coherente y completo.

### Normalización de Datos

Previo a la integración, aplicamos un proceso de normalización para estandarizar formatos y estructuras de datos. Esta etapa garantiza la coherencia y comparabilidad de la información, esencial para el éxito del modelo común.

### Vinculación de Registros

Utilizando la clave unificada, realizamos la vinculación de registros entre las diferentes fuentes. Este paso es fundamental para consolidar la información relativa a una entidad específica, creando así un registro maestro que abarca todas las fuentes de datos.

### Resultados Esperados

La implementación de este modelo común del Golden Record no solo asegura la cohesión de datos, sino que también simplifica la gestión y análisis de información proveniente de múltiples fuentes. La clave unificada sirve como el hilo conductor que teje la trama completa de la información, ofreciendo una visión holística y precisa de las entidades analizadas.

## 2.6. Extracción de datos

La extracción de datos se realiza a través de un proceso fundamental que define cómo capturamos y gestionamos la información para nuestro sistema. En nuestro enfoque, evaluamos dos alternativas

principales: el método de polling y el enfoque basado en eventos. Finalmente se tomará una decisión de elección entre las dos.

### 2.6.1. Basado en Eventos

Este enfoque se centra en la sincronización y la captura de datos cuando ocurren eventos relevantes, permitiendo la integración de información proveniente de diversas fuentes. Esta metodología optimiza la eficiencia al proporcionar datos actualizados y contextualmente significativos, ya que, en lugar de un proceso continuo, la extracción basada en eventos se activa cuando se producen cambios o interacciones clave en las fuentes de datos.

Optamos por utilizar Debezium [7], un sistema basado en eventos, para la extracción de datos, y Kafka Connect [8] para integrar de manera eficiente Debezium con nuestra infraestructura. Este enfoque nos permite capturar cambios en las bases de datos en tiempo real y transmitir eventos asociados a esos cambios mediante Kafka. La estructura de mensajes Debezium proporciona detalles esenciales sobre las operaciones realizadas, los metadatos de origen y marcas de tiempo. La integración con Kafka Connect facilita la canalización fluida de estos eventos hacia los diversos destinos deseados, permitiéndonos gestionar la información de manera eficaz y escalable en nuestro entorno de procesamiento de datos.

Los datos de las diferentes fuentes de datos se reciben mediante eventos en una cómoda estructura dada por Debezium. Consta de un mensaje en formato que cuenta con dos partes, el *schema* y el *payload*. El *schema* se encarga de describir cómo es, qué tipos tiene y demás información sobre el *payload*. El *payload* siempre cuenta cinco atributos:

- **op:** indica que el tipo de operación es (Create, update, read, delete...).
- **source:** contiene una estructura que describe los metadatos de origen del evento. Dependiendo del conector pueden diferir los campos que contiene. La estructura puede incluir los siguientes campos: La versión de Debezium, el nombre del conector y nombre de usuario que hizo el cambio entre otras.
- **ts\_ms:** campo que muestra la hora a la que el conector procesó el evento. La hora se basa en el reloj del sistema en la JVM que ejecuta la tarea Kafka Connect.
- **before:** en caso de que la operación sea un update indica cómo era el objeto antes de cambiarse.
- **after:** indica cómo es el objeto después de los cambios (en caso de ser operación update) o cómo es el nuevo objeto en caso de ser operación Create.

En conclusión, este enfoque se destaca por su eficiencia al sincronizar y capturar datos mediante eventos, facilitando la integración de información diversa. La elección de Debezium como sistema de extracción, con su capacidad para capturar cambios en tiempo real, ha sido fundamental. La estructura de mensajes Debezium proporciona detalles esenciales sobre operaciones y metadatos, facilitando la recepción de datos de diferentes fuentes en un formato organizado. Este método, basado en eventos, no solo optimiza la eficiencia al ofrecer datos actualizados, sino que también garantiza la relevancia contextual de la información recibida, consolidando su efectividad en la construcción del sistema de Golden Record.

## 2.6.2. Polling

La segunda alternativa consiste en utilizar el enfoque de Polling para la extracción de datos directamente desde las bases de datos. Este método involucra la ejecución de consultas SQL específicas para obtener información actualizada periódicamente. La estructura de los datos extraídos dependerá de las consultas SQL definidas para cada fuente de datos.

### Ejecución de Consultas SQL

Se realizan consultas SQL directamente a las bases de datos para extraer datos actualizados.

### Control Directo

Proporciona un control más directo sobre cuándo y qué datos se extraen, ya que las consultas se ejecutan en intervalos definidos.

### Flexibilidad

Permite adaptarse a diferentes esquemas y estructuras de bases de datos, brindando flexibilidad en la selección de datos y la frecuencia de extracción.

### Implementación

Un breve ejemplo de este tipo de extracción de datos tendría lo siguiente:

- **Definir una tabla de control:** lo primero que hay que hacer es crear una tabla en cada una de las fuentes de datos que actuará como un registro de control. Podrá contener información sobre la última vez que se realizó una consulta o cualquier otro metadato relevante
- **Consultas periódicas:** se desarrollaría un *script* o aplicación que se ejecute periódicamente para realizar consultas a las fuentes de datos. También se encargará de verificar si hay cambios desde la última vez que se ejecutó.
- **Manejar resultados:** se trata de procesar los resultados de la consulta y realizar las acciones necesarias (enviar notificaciones...).

Es importante ajustar la frecuencia de las consultas periódicas, ya que un tiempo de espera demasiado corto puede aumentar la carga en la base de datos, mientras que uno demasiado largo puede causar retrasos en la detección de cambios [9].

## 2.6.3. Elección

Finalmente se ha optado por utilizar un sistema basado en eventos en lugar de Polling, ya que es una solución más eficiente y reactiva. En un entorno donde la inmediatez en la detección de cambios son críticas, los eventos permiten una notificación instantánea, eliminando la necesidad de consultas periódicas.

Dentro de del sistema basado en eventos, utilizaremos Debezium como la solución principal, ya que ofrece una estructura eficiente y bien definida para gestionar datos provenientes de diversas fuentes. Su enfoque basado en eventos, donde los cambios en las bases de datos son capturados y transmitidos en tiempo real, proporciona una forma precisa y oportuna de mantener actualizado nuestro Golden Record.

La estructura de mensajes Debezium, que consta de un *schema* y un *payload*, nos brinda la capacidad de comprender la naturaleza y el contexto de cada cambio en los datos. La inclusión de información

como la operación realizada, los metadatos de origen y marcas de tiempo, facilita la detección y resolución de conflictos de manera efectiva.

Además, la capacidad de Debezium para integrarse con diversas bases de datos y sistemas hace que sea una elección versátil y adaptable a nuestras necesidades. La decisión de utilizar Debezium se alinea con nuestra búsqueda de una solución robusta y eficiente para garantizar la calidad y coherencia de nuestro Golden Record a medida que evolucionamos y expandimos nuestro sistema.

## **2.7. Tratamiento de los datos extraídos**

En la gestión del sistema de Golden Record, los procesos de detección y resolución de conflictos desempeñan un papel crucial al recibir datos y aplicar reglas específicas de canonización según la operación realizada. Este enfoque se extiende a situaciones de eliminación, actualización, inserción y otras operaciones, destacando la atención especial a la gestión de errores para mantener la integridad del sistema. Además, el proceso de carga se presenta como una fase esencial, donde el envío de datos sigue un esquema preciso con parámetros. Estos procesos forman la base para un sistema de Golden Record eficiente y preciso en la consolidación y gestión de datos provenientes de diversas fuentes.

### **2.7.1. Procesos de detección y resolución de conflictos**

Los datos se reciben dado el esquema especificado y dependiendo del tipo de operación se hace una cosa u otra. Una vez recibido el mensaje y dadas unas reglas de canonización (ser recibirá un listado en el formato como se muestra en el ejemplo 2.2) del servicio al que pertenece el evento se tratará de la siguiente manera. Una de las fuentes de datos será la que tenga mayor privilegio para el caso de borrados.

#### **Eliminación (Delete)**

En esta situación, el sistema verifica si el mensaje es nulo, lo que denota una operación de eliminación. Si la fuente de datos es la que tiene mayor privilegio, significará que ese dato se va a eliminar por completo en el Golden Record por lo que se enviará un mensaje nulo. En caso de ser una de las fuentes de datos con menos privilegios, se enviará el mensaje con los campos que solo tiene esta fuente de datos con valor a nulo, para que así solo se borren esos campos. Una vez generado un mensaje canonizado y lo envía para su procesamiento.

#### **Actualización (Update)**

Se analiza el mensaje para determinar si es una actualización y, en caso afirmativo, se calculan las diferencias entre los datos antes y después de la operación. Posteriormente, se aplican reglas de canonización específicas para los campos de la diferencia que se ha generado. El mensaje resultante se envía al sistema de procesamiento.

#### **Inserción y Otros (Resto)**

Para mensajes que no representan eliminación ni actualización, es decir, inserciones y otras operaciones, el código convierte el mensaje en una representación canónica de acuerdo con las reglas específicas del servicio. Luego, este mensaje transformado se envía al sistema de procesamiento.

#### **Manejo de Errores**

El código se encuentra equipado para gestionar situaciones inesperadas durante el procesamiento de mensajes. Cuando se produce un error en cualquiera de las operaciones mencionadas anteriormente,

se captura la excepción. Si el número de intentos de reenvío del mensaje es menor o igual al máximo permitido, el código programa un reintento de procesamiento utilizando los mismos argumentos. Sin embargo, si se alcanza el límite de reintentos, se registra un mensaje de error que informa que se han superado los intentos máximos de procesamiento del mensaje y se incluye información sobre el error que ocurrió. Esto asegura que el sistema intente corregir los errores temporalmente antes de registrar una falla definitiva.

Servicio	Origen	Destino
facturacion	mail	eMail

Tabla 2.2: Ejemplo de Regla de Canonización

## 2.7.2. Proceso de carga

Los datos se envían siguiendo un esquema específico. Se trata de 3 parámetros:

- **object:** almacena el cuerpo de los diferentes mensajes canonizados. Este “cuerpo” representa la información contenida en el mensaje original que ha sido transformada y adaptada de acuerdo con las reglas de canonización específicas para el servicio. Puede contener datos relevantes que describen algún cambio, actualización, inserción o eliminación en la base de datos u otra fuente de información. Es esencial para el procesamiento posterior de los datos.
- **op:** es una representación de la operación que se ha realizado en los datos contenidos en object. Puede tomar valores como “update” para indicar una actualización, “create” para una inserción, “delete” para una eliminación, o cualquier otro tipo de operación que caracterice el cambio realizado en los datos. Esta información es fundamental para que el sistema comprenda cómo interpretar y procesar los datos de manera adecuada.
- **service:** almacena el nombre del servicio o componente que ha generado el evento. Este dato es esencial para encaminar y dirigir el mensaje canonizado hacia el destino correcto en el sistema. Indica la fuente del evento, lo que ayuda a garantizar que los datos se envíen a la ubicación adecuada para su posterior procesamiento o almacenamiento.

En resumen, el proceso de envío de datos consta de tres parámetros clave: *object* para la información canonizada, “op” para la operación realizada y *service* para identificar el origen del evento. Estos elementos son esenciales para el procesamiento efectivo en el sistema de Golden Record.

## 2.8. Solución como plantilla

Para poder terminar el análisis hay que aclarar los puntos que forman parte de la plantilla y cuáles los tendría que hacer o completar el cliente. En el punto 5 se podrá ver un ejemplo de implementación de estos apartados.

- **Fuente de Datos:** las fuentes de datos serán parte del cliente.
- **Conectores:** ya que se utilizará Debezium para la captura de datos, dependiendo del tipo de Base de Datos cambian los conectores. Además, habrá que personalizar cada conector con la base de datos que se quiera capturar.
- **Topics:** ya que los conectores cambian, los topics que escucha el canonizador también cambian. Habrá que especificar los nombres de los topics en los que publicarán los mensajes los diferentes conectores.

- **Reglas de Canonización:** se tendrá que especificar y personalizar las reglas de canonización que se quiera, aunque esto ya forma parte del uso del propio golden record.

En conclusión, la solución propuesta opera como una plantilla adaptable, requiriendo la participación del cliente en aspectos específicos. Las fuentes de datos, reglas de canonización, topics y conectores son responsabilidad del cliente, mientras que el resto del sistema no hará falta cambiar nada. Para una implementación práctica, se puede consultar el ejemplo detallado en el capítulo 5.

## 2.9. Conclusiones

En este capítulo se examinaron detalladamente las distintas arquitecturas para la creación de un sistema Golden Record, considerando *Data Warehouse*, Microservicios y Almacenamiento Distribuido y Big Data. Tras evaluar criterios relevantes como control centralizado, coherencia, flexibilidad, escalabilidad, eficiencia en integración y consulta, se concluyó que la arquitectura de Microservicios es la opción más prometedora.

Las razones principales para esta elección incluyen la flexibilidad y escalabilidad de los Microservicios, su capacidad para gestionar diversas fuentes de datos y flujos de trabajo complejos, y su naturaleza modular que facilita la personalización eficiente. Aunque no ofrece la misma coherencia centralizada que el Almacenamiento Centralizado, estas desventajas se compensan con la capacidad de adaptarse a las necesidades específicas de cada cliente.

El capítulo detalla el proceso de selección de arquitectura, proporcionando una tabla de puntuación comparativa. Además, se profundiza en la creación y gestión del Golden Record, abordando aspectos clave como la identificación de fuentes de datos, esquema canónico, limpieza de datos, detección y resolución de conflictos, y generación del Golden Record. Se destaca el enfoque basado en eventos con Debezium para el diseño del modelo común y la extracción de datos.

La sección final del capítulo se centra en el tratamiento de los datos extraídos, abordando procesos de detección y resolución de conflictos, así como la carga de datos. Se subraya la importancia de manejar errores y garantizar la integridad del sistema. La propuesta actúa como una plantilla adaptable, requiriendo la participación del cliente en aspectos específicos como fuentes de datos, reglas de canonización y conectores, permitiendo una implementación personalizada para satisfacer las necesidades únicas de cada cliente.

## 3 Diseño del Sistema

En este capítulo de diseño, se describirán las funciones específicas de cada microservicio en el sistema del “Golden Record”. También se explicará cómo estos microservicios se comunican entre sí mediante conectores personalizados. Además, se abordará la manera en que se aborda la parte visual del sistema, conocida como front-end, para asegurar que sea fácil de usar y se ajuste a las necesidades de los usuarios.

### 3.1. Arquitectura del Sistema

En la figura 3.1 se puede apreciar como los diferentes conectores *source* Debezium se encargan de capturar cambios en las fuentes de datos, enviándolos mediante “topics” hacia el canonizador.

Una vez en el canonizador, las reglas de canonización entran en juego. Estas reglas definen cómo estandarizar y unificar los datos, asegurando coherencia en toda la información procesada. Después de aplicar estas reglas, los datos canonizados se envían al Golden Record mediante otro conector *sink* de Debezium.

También se ha integrado el Gateway y el Eureka Server para mejorar la comunicación entre microservicios. Eureka Server, desarrollado por Netflix, actúa como un catálogo centralizado para el registro y descubrimiento dinámico de microservicios, facilitando la escalabilidad y resiliencia [10]. Por otro lado, Gateway, basado en el ecosistema de Spring, ofrece una puerta de enlace que permite el enrutamiento dinámico y el manejo de solicitudes, mejorando la flexibilidad y modularidad en el diseño de la arquitectura [11]. Estos elementos no solo facilitan la conexión y colaboración entre diferentes partes del sistema, sino que también refuerzan la seguridad en la integración, asegurando un intercambio de datos eficiente y protegido. Este enfoque integral garantiza no solo la coherencia y calidad de los datos, sino también una arquitectura sólida y segura.

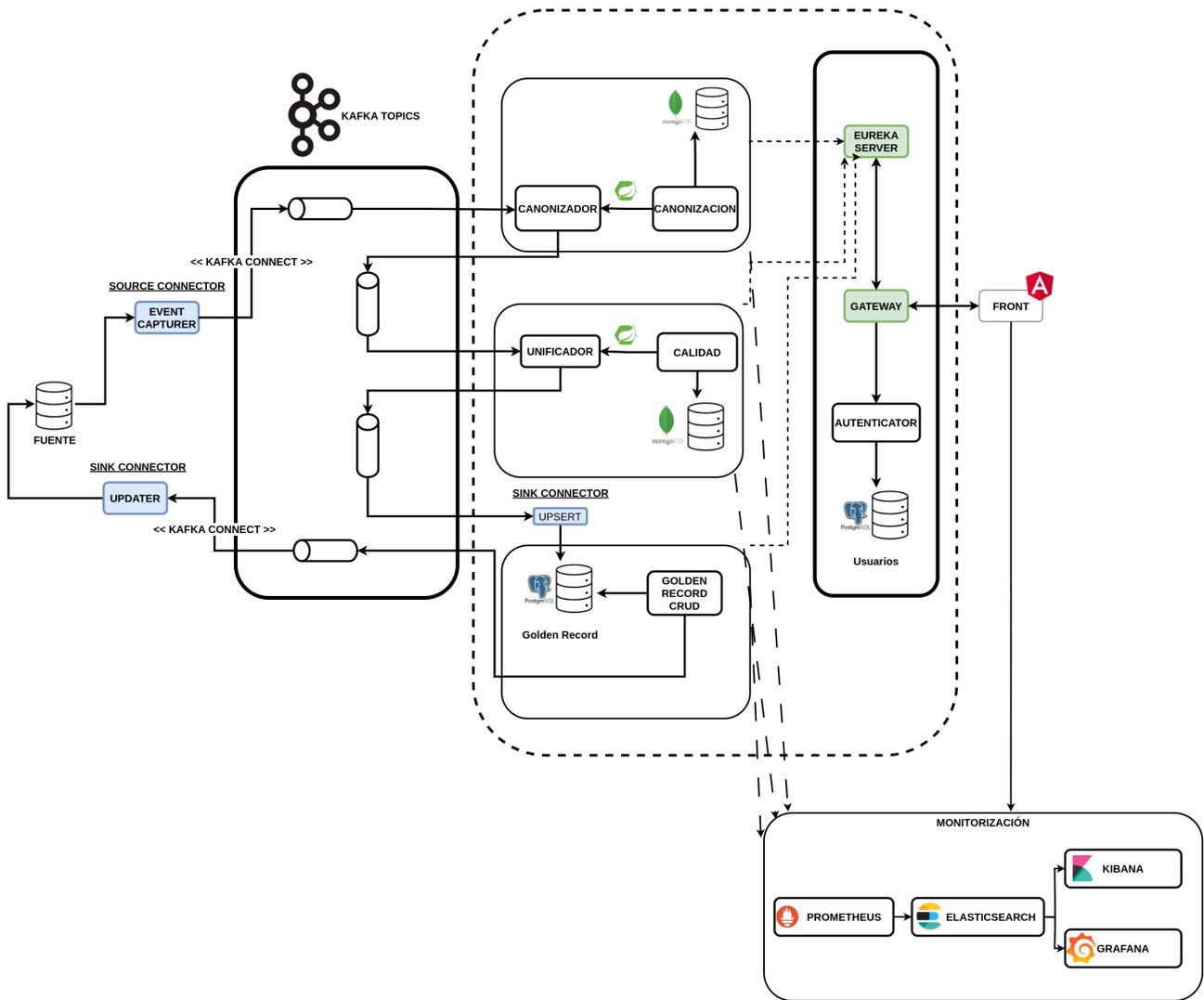


Figura 3.1: Diagrama de Arquitectura

### Microservicios que componen el Golden Record

- Canonización:** El microservicio de canonización desempeña un papel fundamental en el sistema de Golden Record al encargarse de crear, leer, modificar y borrar reglas de canonización a través de una API REST. Estas reglas juegan un papel crucial en el procesamiento de mensajes y en el tratamiento de atributos duplicados en el sistema. La canonización es el proceso de estandarizar, normalizar y unificar datos provenientes de diversas fuentes en un formato coherente y uniforme. A través de la definición de reglas de canonización, es posible establecer criterios específicos para identificar atributos duplicados, así como para decidir si deben ser descartados o renombrados. Almacena los resultados de estas gestiones en una base de datos NoSQL, lo que permite una rápida y eficiente recuperación y gestión de estas, facilitando la adaptación dinámica a las necesidades cambiantes de las diferentes empresas y escenarios.
- Canonizador:** El canonizador asume un rol esencial en el sistema de Golden Record al actuar como intermediario para la integración y unificación de datos provenientes de distintas fuentes. Este componente recibe las actualizaciones de las fuentes de datos a través de Kafka Connect, y luego se comunica con el microservicio de canonización para obtener las reglas específicas que se aplicarán a cada fuente. Utilizando estas reglas, el canonizador procesa los cambios de las fuentes, aplica las transformaciones necesarias para lograr una unificación coherente y, una vez finalizada la canonización, envía los resultados a través de un topic de Kafka al unificador.

Esta comunicación asincrónica garantiza un flujo eficiente de datos en tiempo real, cerrando el ciclo de transformación y preparando los datos para el siguiente paso en el proceso del sistema de Golden Record.

- **Unificador:** El unificador desempeña un papel esencial en el sistema de Golden Record al actuar como el último paso en el proceso de consolidación de datos. Este componente recibe los datos canonizados a través de un topic de kafka, provenientes del canonizador. Una vez recibidos, el unificador establece comunicación con el microservicio de calidad, el cual proporciona las reglas de calidad específicas para los datos canonizados. Estas reglas determinan los criterios y estándares que deben cumplir los datos antes de ser incluidos en el Golden Record. Una vez que los datos cumplen con las reglas de calidad, el unificador los almacena en el Golden Record mediante Kafka Connect utilizando un conector Sink. Este proceso asegura que solo los datos de alta calidad y coherencia sean incorporados al Golden Record, garantizando la integridad y precisión de la información almacenada. En resumen, el unificador cierra el ciclo del sistema de Golden Record al recibir, evaluar y finalmente integrar los datos canonizados y de calidad en el Golden Record, asegurando la creación de una fuente única y confiable de información.
- **Calidad:** A través de una API REST, este componente permite la creación, modificación, lectura y eliminación de reglas de calidad, definiendo criterios como validación de formatos, detección de valores anómalos y cumplimiento de estándares predefinidos. Garantiza que solo los datos coherentes y confiables sean considerados aptos para integrarse en el Golden Record, asegurando la integridad y precisión de la información almacenada. Estas reglas de calidad son gestionadas y almacenadas en una base de datos NoSQL, lo que permite una rápida y eficiente administración de las reglas y su adaptación a los requisitos cambiantes del sistema y las empresas.
- **Golden Record CRUD:** El componente de Gestión del Golden Record opera como un sistema CRUD (Crear, Leer, Actualizar y Borrar), permitiendo un acceso directo a los registros almacenados en la base de datos del Golden Record. Esto otorga la capacidad de crear nuevos registros, modificar aquellos ya existentes, visualizar la información completa de los registros y también eliminar registros si es necesario. En situaciones de eliminación, se utilizan conectores *sink* configurados para operar de manera automática, garantizando la propagación eficiente del mensaje de borrado a las fuentes de datos de origen relacionadas. Esta capacidad bidireccional no solo preserva la integridad del Golden Record, sino que también mantiene la consistencia global del sistema, asegurando que los registros eliminados se reflejen de manera precisa en todas las fuentes de datos conectadas. En consecuencia, este enfoque integral respaldado por Kafka Connect garantiza una gestión completa y efectiva del ciclo de vida de los datos en el entorno del Golden Record.

## Infraestructura

- **Autenticador:** Permite gestionar tanto el registro seguro de nuevos usuarios como la autenticación de usuarios existentes. Se hará uso de tokens JWT [12], que son estructuras compactas que contienen información verificable y confiable, empleadas para la autenticación y transmisión segura de datos entre sistemas mediante una firma digital. A través de la generación de estos tokens, la asignación de roles, la validación de credenciales y la implementación de medidas de seguridad, el Autenticador garantiza un acceso controlado y protegido, asegurando la integridad y confidencialidad de la información en nuestro sistema.
- **Eureka Server:** Permite que los microservicios se registren y encuentren entre sí dinámicamente. Cada microservicio se registra en el servidor Eureka con su nombre y dirección, lo que facilita la localización de servicios disponibles y la carga equilibrada de las solicitudes entre

múltiples instancias de un mismo servicio. El servidor Eureka juega un papel crucial en la escalabilidad y la tolerancia a fallos de nuestra arquitectura, permitiendo que nuevos microservicios se agreguen sin interrupciones y que los servicios se adapten a cambios en la infraestructura.

- **Gateway:** Actúa como entrada principal en nuestra arquitectura de microservicios, dirigiendo solicitudes a los microservicios adecuados según rutas definidas. Importante aún, verifica la validez de los tokens JWT en las solicitudes usando el microservicio “Autenticador”, asegurando que los usuarios estén autenticados y tengan los permisos correctos. Esto asegura el acceso seguro y controlado a los recursos y servicios subyacentes, fortaleciendo la seguridad y facilitando la distribución eficiente de solicitudes.
- **Monitorización:** los microservicios del sistema envían métricas a *Prometheus* [13] a través de *exporters* [14], que a su vez son almacenadas y gestionadas en “Elasticsearch” [15]. Utilizamos “Kibana” [16] para visualizar y explorar estos datos almacenados de manera eficiente. Además, hemos integrado “Grafana” [17] para crear *dashboards* interactivos basados en las métricas recopiladas por “Prometheus”. Para facilitar el acceso y la monitorización, se establecen enlaces directos desde el front a los *dashboards* específicos en “Grafana” y “Kibana”, permitiendo a los usuarios analizar fácilmente el rendimiento y la salud de nuestros microservicios de manera intuitiva y en tiempo real. Esto mejora significativamente la experiencia del usuario al proporcionar un acceso directo y eficaz a las herramientas de visualización y análisis de nuestro sistema.

## 3.2. Solución como plantilla

Abordamos la solución como una plantilla para nuestro sistema Golden Record. Destacamos la importancia del diseño adecuado de las fuentes de datos, enfocándonos en dos aspectos esenciales: el uso uniforme de claves de identificación entre todas las fuentes y la compatibilidad con Debezium para capturar cambios. Exploramos la estrategia de construcción de conectores, adaptando conectores *source* según el tipo de fuente y utilizando conectores *sink* genéricos “JDBCSinkConnector” en el extremo del Golden Record (explicado más en detalle en 3.2.2). Además, resaltamos la gestión y configuración de *topics* en Kafka, esenciales para la correcta captura y procesamiento de datos. En resumen, esta plantilla asegura una integración eficiente y coherente, manteniendo flexibilidad sin comprometer la uniformidad del proceso.

### 3.2.1. Fuente de Datos

Para que nuestro Golden Record funcione de manera eficiente, el diseño adecuado de las fuentes de datos es esencial. La clave reside en la uniformidad y coherencia de la información que fluye a través del sistema. Aquí hay dos aspectos cruciales que debemos considerar.

#### Uso de la Misma Clave en las Fuentes de Datos:

Un requisito fundamental es que todas las fuentes de datos utilicen la misma clave de identificación. Esto simplifica enormemente el proceso de integración, ya que las actualizaciones y cambios pueden correlacionarse de manera directa y sin ambigüedades entre las distintas fuentes. En el contexto de mensajes Kafka, esta clave se utiliza como el “key” del mensaje. Mantener esta clave uniforme en todas las fuentes asegura la coherencia y la correlación directa de los datos, simplificando la tarea del sistema al construir y mantener un Golden Record consolidado.

#### Compatibilidad con Debezium

Dado que utilizamos Debezium para capturar cambios en las fuentes de datos, es esencial asegurarse de que estas sean compatibles con esta herramienta. Afortunadamente, Debezium ofrece una amplia variedad de posibilidades y conectores para adaptarse a diversas tecnologías de bases de datos y sistemas. Sin embargo, es necesario realizar una evaluación previa para confirmar la compatibilidad específica de las fuentes de datos con Debezium.

### 3.2.2. Conectores

#### Conectores Source Específicos para el Tipo de Fuente de Datos

Debezium ofrece una amplia gama de conectores source específicos para diferentes tipos de bases de datos, como PostgreSQL, Oracle, MySQL, SQL Server, entre otros. La elección del conector source dependerá del tipo de fuente de datos que estemos utilizando. Cada conector source está diseñado para capturar cambios específicos y relevantes para su respectiva tecnología subyacente. Esta personalización específica garantiza una captura precisa de los cambios en las fuentes de datos, lo que es fundamental para mantener la integridad y coherencia en el proceso de integración.

**Conectores Sink JDBC Sink Connector:** En el extremo del Golden Record, todos los conectores sink se diseñarán utilizando el tipo genérico JDBC Sink Connector. Este tipo de conector sink ofrece versatilidad al trabajar con diferentes bases de datos relacionales a través de JDBC (Java Database Connectivity). La elección de un conector sink genérico simplifica la configuración y gestión del flujo de datos hacia el Golden Record, permitiendo una mayor flexibilidad en la elección de la base de datos de destino sin comprometer la uniformidad en el proceso.

En resumen, la estrategia de conectores Debezium se basa en la adaptación específica de conectores source según el tipo de fuente de datos, asegurando una captura precisa de cambios. Por otro lado, la uniformidad y la flexibilidad se mantienen en el extremo del Golden Record mediante el uso de conectores sink genéricos del tipo JDBC Sink Connector. Esta combinación permite una integración eficiente y coherente de datos, desde diversas fuentes hasta el punto central del Golden Record, sin sacrificar la especificidad necesaria en la captura de cambios [18].

### 3.2.3. Topics

Un aspecto crítico en la implementación de nuestro sistema es la gestión y configuración de los conectores, ya que estos desempeñan un papel esencial en la captura y transferencia de datos entre el canonizador y el Golden Record. Aquí, destacamos la necesidad de identificar y especificar claramente los topics asociados a cada conector, así como la importancia de ajustar estas configuraciones en una instancia dada.

#### Identificación y Especificación de Topics

Cada conector Debezium está asociado con uno o varios topics en Kafka, que actúan como canales de comunicación para los cambios en las fuentes de datos. Identificar y especificar claramente los topics correspondientes a cada conector es crucial. Esta información se utiliza tanto en el canonizador como en el Golden Record para asegurar la correcta captura y procesamiento de datos. La claridad en la designación de topics facilita la comprensión y mantenimiento de la arquitectura.

#### Modificaciones en el Canonizador y Golden Record

Dado que los conectores forman parte de lo que se debe modificar en una instancia específica, es esencial adaptar las configuraciones de los conectores en el canonizador y el Golden Record según las necesidades particulares de cada implementación. Esto puede incluir ajustes en la especificación

de los topics que los conectores escuchan en el canonizador y escriben en el Golden Record. Esta flexibilidad permite una adaptación precisa a los requisitos específicos de cada instancia, asegurando una integración efectiva y eficiente.

En resumen, la identificación clara de los topics asociados a cada conector y la capacidad de realizar modificaciones específicas en el canonizador y el Golden Record son pasos cruciales en la configuración de conectores. Estas acciones garantizan una implementación precisa y adaptada a los requisitos particulares de cada instancia, permitiendo la captura y transferencia efectiva de datos a lo largo de todo el sistema.

### **3.3. Front-End**

El front-end del proyecto consta de varios componentes funcionales que contribuyen a la interfaz y la experiencia del usuario. Comenzando con la autenticación en el proceso de inicio de sesión, esta etapa asegura que solo los usuarios autorizados puedan acceder al sistema, proporcionando una capa de seguridad esencial.

Una vez autenticados, los usuarios son dirigidos a la página principal, que actúa como el centro de control para acceder a las diferentes funcionalidades. Desde aquí, los usuarios tienen la opción de navegar a diferentes secciones clave: el Golden Record, la canonización, la calidad y las métricas.

En la sección del Golden Record, los usuarios pueden interactuar con la base de datos principal, lo que implica ver, editar, crear y eliminar registros. Esto facilita la gestión de los datos esenciales que conforman el Golden Record y permite que los usuarios realicen cambios según sea necesario.

La sección de canonización ofrece a los usuarios la posibilidad de aplicar reglas de canonización a los datos, asegurando que sigan un formato y estructura uniformes. Además, en la sección de calidad, los usuarios pueden evaluar y mejorar la calidad de los datos, lo que puede implicar la detección y corrección de errores.

Finalmente, la sección de métricas proporciona una visión general de las estadísticas y el rendimiento del sistema, ofreciendo información valiosa para evaluar su funcionamiento y eficacia.

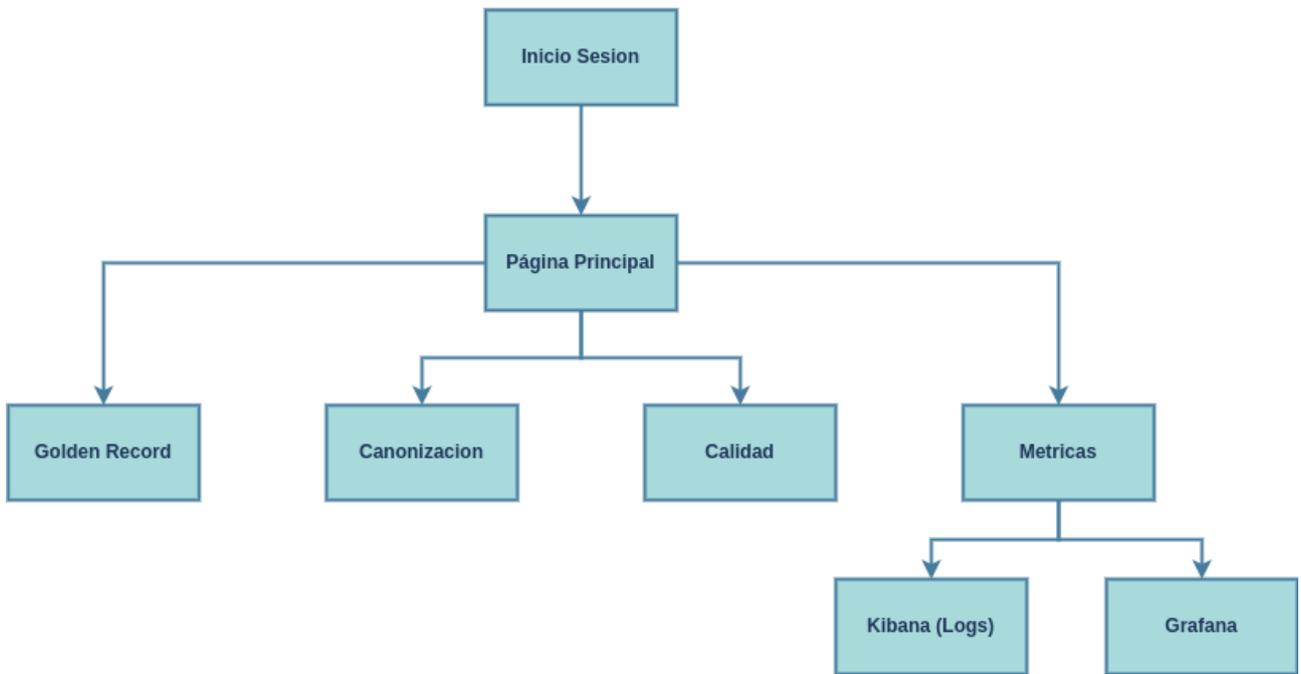


Figura 3.2: Diagrama de navegación de alto nivel

## 4 Implementación y pruebas

En el capítulo de Implementación y Pruebas se detalla la implementación de la mensajería asíncrona con Kafka. Se examina la seguridad a través del microservicio de Autenticador, destacando medidas de protección. Además, se profundiza en cada microservicio, desde sus roles y comunicación hasta sus características únicas, así como la monitorización del sistema.

### 4.1. Componentes

En el núcleo de la arquitectura se encuentra el Canonizador, un microservicio esencial que procesa y enruta datos desde diversas fuentes hacia servicios específicos, asegurando la integridad en toda la arquitectura. El Unificador desempeña un papel central al recibir datos canonizados y gestionar la creación o actualización de registros en el Golden Record. Los microservicios de Calidad y Canonización comparten una estructura similar, permitiendo operaciones CRUD eficientes, mientras que el Golden Record CRUD facilita la gestión completa de registros, destacando su enfoque único de eliminación para garantizar coherencia entre el Golden Record y las fuentes originales. En conjunto, estos componentes forman una arquitectura robusta y coherente para la gestión de datos en el sistema Golden Record.

#### 4.1.1. Canonizador

Uno de los microservicios cruciales es el Canonizador, que desempeña un papel esencial en el flujo de datos dentro de la arquitectura. Aquí se explica cómo opera este microservicio en varios pasos:

- El Canonizador recibe información de diferentes fuentes de datos a través de distintos topics. Cada servicio clave tiene su propio topic dedicado para transmitir los cambios correspondientes a esa área específica, lo que permite una segmentación y gestión eficientes de los datos.
- Los mensajes recibidos están en un formato específico, ya que provienen de Debezium. Para procesar estos mensajes, primero se deserializan utilizando una plantilla predeterminada. Esta etapa de deserialización es crucial para extraer los detalles esenciales del mensaje.
- Tras completar la deserialización, el Canonizador utiliza un servicio de “OpenFeign” [19] para establecer una comunicación con el microservicio Canonización. El objetivo aquí es obtener las reglas de canonización necesarias para el mensaje específico proveniente del servicio indicado.
- Mediante el uso del servicio “processor”, proporcionándole una lista de reglas de canonización junto con el payload del mensaje Debezium nos devolverá como resultado el payload canonizado con las reglas de canonización proporcionadas.

- Con el resultado del “processor” en mano, se crea un JSON que incluye la operación realizada, el servicio al que pertenece y el propio objeto canonizado. Este JSON encapsula la información esencial sobre el cambio o la acción que se produjo en los datos.
- Finalmente, el Canonizador envía el JSON creado a través del topic de Kafka. Esto garantiza que el mensaje transformado y enriquecido llegue al destino adecuado y pueda ser procesado por los componentes pertinentes.

En conjunto, estos pasos demuestran cómo el microservicio Canonizador cumple una función esencial al procesar, enriquecer y enrutar los datos provenientes de distintas fuentes hacia los servicios y topics correspondientes, manteniendo la integridad y la coherencia de la información en toda la arquitectura.

#### **4.1.2. Unificador**

El componente unificador desempeña un papel central al recibir los datos canonizados a través del topic. Una vez que estos datos son recibidos, el unificador inicia un proceso crucial en el que invoca el servicio OpenFeign para obtener las reglas de calidad correspondientes (véase punto de Trabajo Futuro). Posteriormente, el unificador genera un mensaje que incluye el esquema del dato (necesario para la correcta ejecución de los conectores Sink) y el propio contenido del mensaje. Este mensaje es entonces enviado a través del topic del conector Sink, cuya función primordial es administrar la creación de nuevos registros en el Golden Record o la actualización de los registros existentes, en línea con las directrices establecidas por las reglas de calidad. Este proceso de unificación y actualización de datos asegura que el Golden Record refleje la información más precisa y actualizada a partir de los datos canonizados, manteniendo la integridad y la coherencia del sistema en su conjunto.

#### **4.1.3. Calidad y Canonización**

Tanto el microservicio de Calidad como el de Canonización se han diseñado con una estructura similar y funcionalidad equivalente. Ambos componentes permiten llevar a cabo operaciones de creación, modificación, visualización y eliminación de datos utilizando los métodos HTTP correspondientes: POST, PUT, GET y DELETE, respectivamente. Estos servicios ofrecen una interfaz uniforme y coherente para interactuar con las reglas de calidad y canonización de manera eficiente y cohesiva. Cada microservicio proporciona una forma de gestionar las reglas de manera intuitiva y accesible, lo que agiliza la administración y el mantenimiento de estas reglas esenciales para la coherencia y la calidad de los datos en el sistema Golden Record.

#### **4.1.4. Golden Record CRUD**

El componente de operaciones CRUD en el sistema Golden Record se asemeja en su funcionamiento al de Calidad y Canonización, al facilitar la creación, modificación, visualización y eliminación de registros mediante los métodos HTTP correspondientes. Sin embargo, el Golden Record CRUD introduce un aspecto distintivo: lleva a cabo un análisis detallado de los cambios y accesos a la base de datos del Golden Record mediante la función de Envers, como se describe en el anexo adjunto. Además de los métodos HTTP previamente mencionados, este componente añade una funcionalidad adicional al proporcionar un método extra para recuperar todos los datos registrados en Envers.

En cuanto al proceso de eliminación de registros, el Golden Record CRUD adopta un enfoque particular. Al eliminar un registro, el sistema envía un mensaje con contenido nulo a través de cada topic de los conectores Sink asociados a las fuentes de datos originales. Esta acción es interpretada

como una eliminación en la fuente de datos de origen, lo que garantiza la coherencia entre el Golden Record y las fuentes de datos subyacentes. Este enfoque asegura que los cambios se propaguen de manera efectiva a través de todas las fuentes de datos relevantes, manteniendo la integridad y consistencia de la información en todo el sistema.

## 4.2. Kafka

Para habilitar el uso de Kafka en mi proyecto, fue necesario llevar a cabo un despliegue en Docker Compose de varios componentes esenciales.

- **kafka y kafka-connect:** Se implementaron estos contenedores para la gestión de eventos y conectores respectivamente. “debezium/kafka” proporciona la infraestructura necesaria para gestionar flujos de eventos en tiempo real, mientras que “debezium/connect” permite la conexión y configuración de conectores que facilitan la transferencia de datos hacia y desde Kafka.
- **zookeeper:** Este contenedor desempeña un papel esencial como servicio de coordinación centralizado. Se encarga de administrar el estado y la configuración de los clústeres de Kafka. Su función crítica asegura la coherencia y disponibilidad en el sistema al gestionar la información sobre la ubicación de líderes de particiones y la configuración de brokers [20].
- **schema registry:** Este componente proporciona una ubicación centralizada para el almacenamiento y la gestión de los esquemas utilizados en la serialización y deserialización de mensajes. Asegura que los datos en los mensajes sigan un formato uniforme y coherente, lo que es esencial para la interoperabilidad y la evolución de los datos a lo largo del tiempo.
- **kowl:** Para facilitar la supervisión y la visualización de los temas y consumidores de Kafka, se implementó el contenedor “kowl”. Este componente proporciona una interfaz de usuario amigable que permite a los usuarios monitorear la actividad de Kafka, lo que contribuye a una gestión efectiva y una comprensión más profunda del flujo de eventos [21].

## 4.3. Conectores

Para crear un conector en el entorno de Debezium, se sigue un proceso que implica la creación de un archivo JSON con las diferentes propiedades del conector deseado. Este archivo JSON contiene la configuración específica (véase anexo A) que define cómo el conector se conectará a las fuentes de datos y cómo gestionará los eventos.

Una vez que el archivo JSON de configuración está listo, se realiza una solicitud HTTP POST al contenedor de Debezium. En mi caso, este contenedor se encuentra en el puerto 8083. Al realizar la solicitud POST con el archivo JSON de configuración, se inicia el proceso de creación del conector. Debezium utiliza la información proporcionada en el archivo JSON para establecer la conexión con la fuente de datos, capturar eventos y transferirlos a Kafka para su posterior procesamiento. En mi caso cuento con un script que crea todos los conectores de una vez.

## 4.4. Monitorización

Para la monitorización del sistema se ha recurrido al uso de contenedores Docker. A través de esta tecnología, se ha logrado implementar soluciones de monitorización eficientes. En particular, se ha aprovechado Grafana para la creación de *dashboards* que ofrecen una visión consolidada del rendimiento y la salud del sistema. Por otro lado, Kibana ha sido empleado para la lectura y análisis de los registros (logs) generados por cada microservicio, permitiendo realizar trazas y diagnósticos detallados. Es importante destacar que la habilitación de estas capacidades en Grafana y Kibana implicó la adición de dependencias específicas en los proyectos, así como la configuración de archivos pertinentes para comunicar a ambas herramientas cuáles microservicios requerían de estas funcionalidades.

## 4.5. Seguridad

En el ámbito de la seguridad, se ha implementado un robusto sistema de autenticación respaldado por un Gateway. Todas las solicitudes desde el frontend atraviesan este Gateway, que valida la autenticación mediante un token. Este token se obtiene durante el proceso de inicio de sesión y se valida a través del microservicio Autenticador. Posteriormente, el Gateway dirige la solicitud al microservicio correspondiente, incluyendo el rol del usuario para autorización. Los roles determinan las acciones permitidas, asegurando un acceso controlado a los recursos. Además, se ha establecido un sistema de copias de seguridad automatizadas cada 12 horas para resguardar la integridad y disponibilidad de los datos en el Golden Record. Este enfoque equilibrado garantiza la continuidad y confiabilidad del sistema, ofreciendo una capa adicional de seguridad ante situaciones imprevistas.

### 4.5.1. Autenticación

Para asegurar la integridad del sistema, se ha implementado la seguridad a través del uso de un Gateway. Como se puede apreciar en la figura 4.1, cada solicitud originada desde el frontend es canalizada a través del gateway proporcionado por el registro de cada microservicio en el Eureka Server. El proceso de seguridad implica que todas las solicitudes entrantes en el gateway requieren un token de autenticación (excepto el proceso de inicio de sesión, que obtiene el token y el rol del usuario). El gateway se encarga de validar este token mediante una llamada al microservicio de Autenticador.

Una vez validado el token, el gateway dirige la solicitud al microservicio correspondiente, incluyendo el rol del usuario en la cabecera. Cada microservicio interpreta el rol del usuario para autorizar o denegar el acceso a los recursos. La asignación de roles determina las acciones que un usuario puede llevar a cabo. Por ejemplo, los usuarios con roles de “usuario” solo pueden realizar solicitudes para visualizar datos. Aquellos con roles de “moderador” pueden crear y modificar valores además de visualizarlos, mientras que los roles de “administrador” tienen el permiso adicional de eliminar valores.

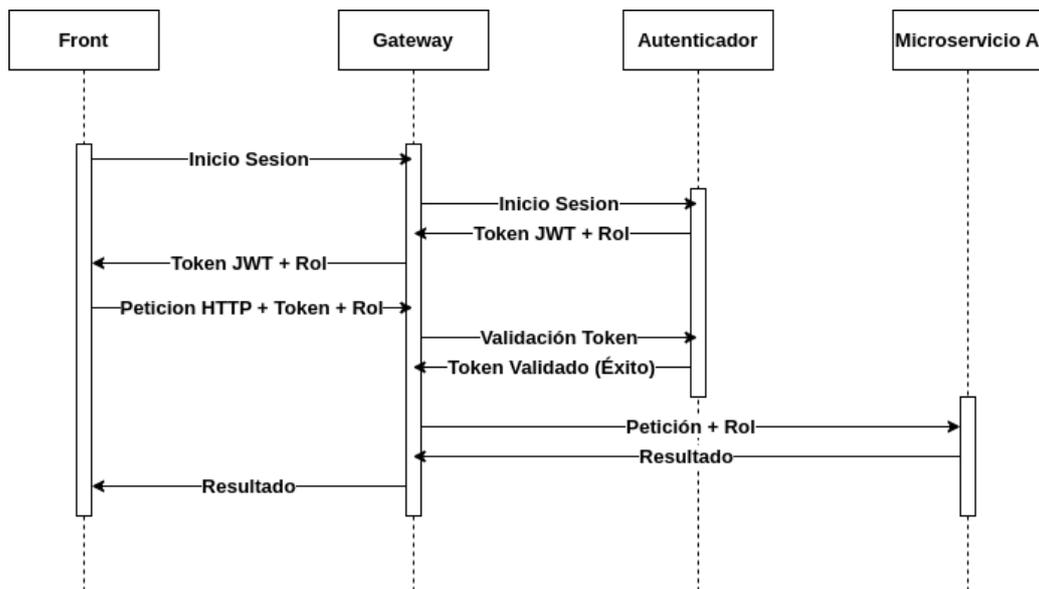


Figura 4.1: Diagrama Secuencia Autenticación

#### 4.5.2. Copia de Seguridad

Para salvaguardar la integridad y disponibilidad de los datos almacenados en el Golden Record, se implementó un enfoque de respaldo automatizado. Se diseñó un *script* que ejecuta una copia de seguridad de la base de datos del Golden Record en intervalos regulares, en este caso, cada 12 horas. Esta solución automatizada garantiza que los datos críticos almacenados en el Golden Record sean respaldados de manera periódica y confiable.

La selección del intervalo de 12 horas se basa en la búsqueda de un equilibrio entre la frecuencia de respaldo y el consumo de recursos. Al programar la ejecución automática del *script* a intervalos regulares, se reduce la necesidad de intervención manual y se asegura que el proceso de respaldo se realice de manera consistente. Esto, a su vez, disminuye el riesgo potencial de pérdida de datos en caso de un fallo inesperado o eventos que puedan afectar la base de datos.

La incorporación de este enfoque de respaldo automatizado añade un nivel de seguridad adicional a los datos almacenados en el Golden Record. En situaciones inesperadas, como un fallo del sistema o errores de datos, la copia de seguridad realizada cada 12 horas permitirá recuperar la información hasta la última instancia respaldada. Esta práctica es esencial para garantizar la continuidad y confiabilidad del sistema, asegurando que los datos cruciales estén protegidos y disponibles en todo momento.

# 5 Prueba de concepto

Para asegurar la eficacia y versatilidad de nuestra solución “white-label”, hemos creado un proyecto de ejemplo integral. Este proyecto no solo sirve como demostración práctica de implementación, sino que también actúa como un modelo que puede adaptarse fácilmente a diversos contextos y necesidades específicas de marca. La solución “white-label” está diseñada para ser personalizable y escalable, permitiendo que empresas de diferentes sectores y dimensiones integren nuestra tecnología de manera fluida. Este proyecto de ejemplo ofrece una visión concreta de cómo nuestra solución puede implementarse de manera exitosa, facilitando a los usuarios la comprensión de su funcionamiento y adaptación a sus requerimientos particulares.

## 5.1. Definición de las fuentes de datos

Se ha explorado exhaustivamente la duplicación de campos, utilizando combinaciones de mayúsculas y minúsculas para evaluar la respuesta del proyecto ante estas variaciones. Además, cada fuente de datos incorpora un campo único que no comparte con las demás, permitiendo observar el comportamiento específico ante la eliminación, especialmente en casos donde el servicio es crítico. Además, cada fuente de datos, que proviene de sistemas distintos, se encuentra en un entorno diferente: una en PostgreSQL, otra en MySQL y la última en Oracle. Esta diversidad de plataformas permite profundizar en las diferencias notables al configurar los conectores Debezium, ofreciendo una visión completa de cómo el proyecto maneja la replicación de datos en entornos variados.

### Centro de Contactos:

Esta fuente abarca todas las interacciones entre nuestros clientes y los agentes de servicio al cliente.

**Base de Datos:** PostgreSQL

<b>contactos</b>
nombre
primerapellido
segundoapellido
direccion
telefono
dni
residencia
email
intereses

## Facturación:

La fuente de datos de facturación se enfoca en la gestión financiera asociada con los clientes. Incluye información relevante para la facturación de servicios o productos.

**Base de Datos:** MySQL

<b>facturacion</b>
nombre
primerApellido
segundoApellido
direccionFacturacion
telefono
dni
mail

## Servicio a Domicilio:

Esta fuente concentra información logística relacionada con los servicios entregados a domicilio.

**Base de Datos:** Oracle

<b>domicilios</b>
NOMBRE
APELLIDOS
DIRECCIONENVIO
NUMEROTLF
DNI
CORREO
PAIS

En resumen, contamos con duplicados, duplicados con nombres distintos y datos exclusivos, proporcionando así un ejemplo integral y representativo de los escenarios que queremos abordar.

## 5.2. Golden Record y definición de las reglas de canonización

En las figuras 5.1, 5.2 y 5.3 se puede apreciar los campos que queremos de cada uno de las fuentes de datos. En cuanto a las reglas de canonización son numerosas, ya que hay que indicar de cada fuente de datos todos los campos (véase Anexo B.1 )

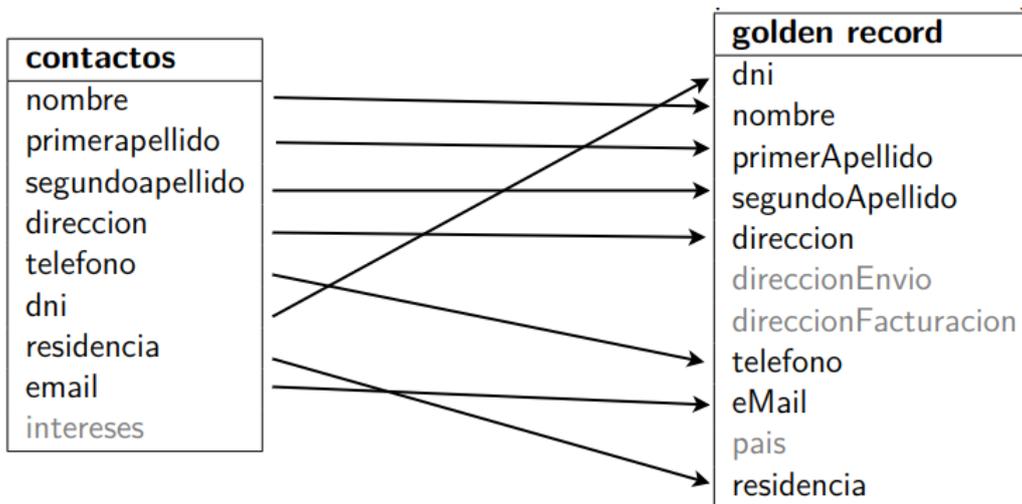


Figura 5.1: Mapping entre Fuente de Datos contactos y Golden Record

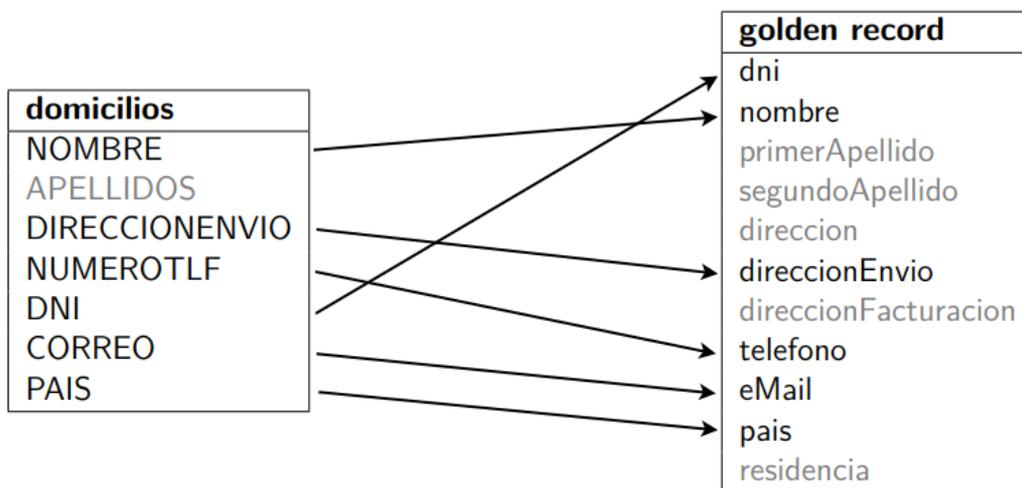


Figura 5.2: Mapping entre Fuente de Datos domicilios y Golden Record

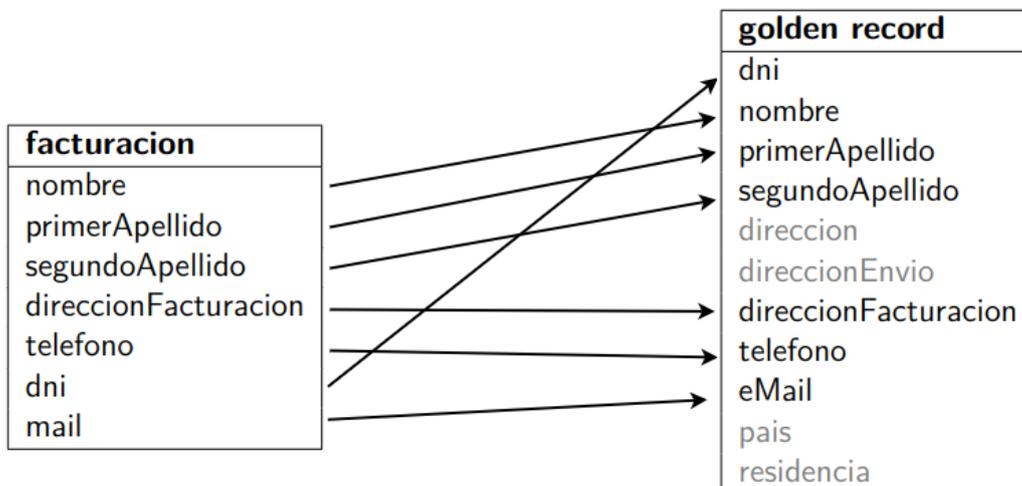


Figura 5.3: Mapping entre Fuente de Datos de facturación y Golden Record

### 5.3. Creación de los conectores

Para la creación de los conectores hay que tener en cuenta que todos los que hay que configurar. Se desarrollarán tres conectores Source y cuatro Sink.

**Source:** Los conectores Source serán MySQL [22], PostgreSQL [23] y OracleDatabase [24], [25]. Estos proporcionan los cambios de cada una de las fuentes de datos al canonizador mediante diferentes Topics de Kafka.

**Sink:** En cuanto a conectores Sink [26], [27], el primero de ellos transfiere la información del Unificador a la BBDD Golden Record. Los tres restantes se encargan de propagar los borrados desde el Golden Record CRUD a las fuentes de datos de origen.

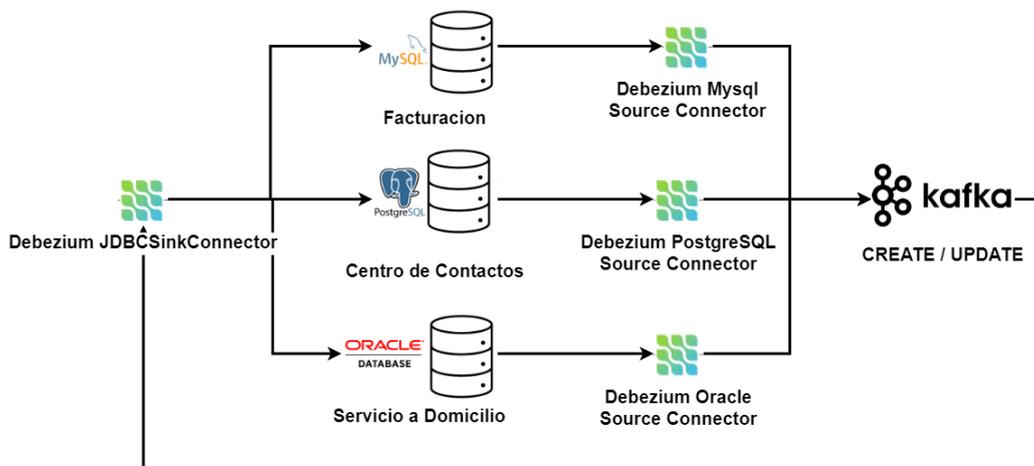


Figura 5.4: Conectores Sink y Source de las Fuentes de Datos

### 5.4. Carga de datos y cómo se comporta el Golden Record

Creamos un mismo cliente para todas las fuentes de datos, usando nombres ficticios. Una vez hecho eso se comenzará a hacer variaciones de campos, eliminación del usuario en diferentes fuentes de datos que no sean críticas y finalmente se hará el borrado del cliente en la fuente de datos crítica que provocará el borrado del cliente en el golden record.

campos	valores
nombre	Elena
primerApellido	Garcia
segundoApellido	Garcia
direccion	C/ Primavera 123
direccionEnvio	C/ Primavera 123
direccionFacturacion	C/ Primavera 123
dni	12345678A
eMail	elena.garcia88@gmail.com
telefono	612 345 678
pais	Colombia
residencia	Antioquia

### 5.4.1. Comportamiento del Golden Record

Tendremos como fuente de datos crítica a Centro de Contactos, por lo que en caso de borrado se tendrá en cuenta.

Cambio de la dirección de Facturación en la fuente de datos de facturación. No supone ningún cambio crítico, el Golden Record solo se vería afectado por ese cambio (véase resultado en Anexo C.1) Una vez hecho eso cambiaremos el número de teléfono en el Centro de Contactos (véase resultado en Anexo C.2). Tampoco supone nada, pero si actualizamos en Servicio a Domicilio también el teléfono, el teléfono que se quede en el Golden Record será este puesto que es el último que ha cambiado (véase resultado en Anexo C.3).

Probamos a borrar el usuario de Elena en Servicio a Domicilio y vemos que la cosa empieza a cambiar, por lo que nos hemos quedado sin los valores que son únicos en la fuente de datos de Servicio a Domicilio (véase resultado en Anexo C.4).

Finalmente borramos el usuario en Centro de Contactos, por lo que siendo la fuente de datos crítica nos quedaremos sin el usuario Elena en el Golden Record.

## 5.5. Ejemplo de uso del Frontal y Monitorización

En cuanto a la navegación se ha detallado anteriormente en el diagrama 3.3. Aquí se pondrá con ejemplos. En cuanto al inicio de sesión es bastante sencillo (F.1), ya que en lo que se profundiza es el contenido, al igual que en la pantalla de inicio (F.2). En cuanto a las reglas de canonización y calidad se puede apreciar lo similares que son, permiten crear, editar, eliminar como se había descrito anteriormente (véase anexo F.3, F.4). En cuanto al Golden Record se puede apreciar en el F.5 y F.6 como se puede usar esta pantalla, permitiendo listar todos los datos del Golden Record, crear, editar y eliminar.

En cuanto a la monitorización accederemos a través de la pantalla de métricas (véase anexo F.7). Se podrá observar Grafana (véase anexo F.8) y Kibana (véase anexo F.9), permitiendo editar o añadir cualquier tipo de *dashboard*.

## 6 Conclusiones y Trabajo Futuro

En este apartado, se consolidan las conclusiones derivadas del proyecto, destacando los logros obtenidos y los aprendizajes adquiridos a lo largo de la investigación. A su vez, se proyecta una mirada hacia el futuro, delineando el trabajo pendiente por abordar.

### 6.1. Conclusiones

En el transcurso de este proyecto, se ha explorado a fondo la problemática que implica la gestión de información diversa en las organizaciones y cómo esta problemática es abordada mediante la implementación de un sistema de “Golden Record”. Se ha logrado una comprensión profunda de la esencia de este concepto, que no solo busca unificar datos provenientes de diversas fuentes, sino también ofrecer una visión completa y precisa de la información. A medida que se ha avanzado en el estudio, se ha llegado a una apreciación más nítida de por qué un enfoque de “Golden Record” es crucial para superar los desafíos que plantea el manejo de datos heterogéneos en las estructuras organizativas.

En respuesta a esta necesidad, se ha colaborado estrechamente con la empresa Hiberus para desarrollar una solución concreta y adaptable a distintos entornos empresariales. Este proceso ha involucrado la evaluación y análisis detallado de los riesgos, así como la exploración de diversas arquitecturas y tecnologías disponibles para implementar el sistema de “Golden Record”. Cada etapa ha sido meticulosamente planificada y ejecutada, desde la identificación de objetivos específicos hasta la selección de la arquitectura de microservicios como la más idónea para lograr soluciones personalizadas y eficientes en la integración del sistema.

El diseño del sistema en sí mismo ha sido un logro destacado en este recorrido. Se ha delineado con claridad qué función desempeñará cada microservicio dentro del sistema, cómo se interconectarán a través de conectores específicos y cómo se gestionará la comunicación entre ellos. Además, se ha abordado el enfoque del front-end, esbozando cómo los usuarios finales interactuarán con el sistema y cómo se presentará la información consolidada.

Aunque la implementación ha sido exitosa en gran medida, es importante reconocer que por la envergadura del proyecto, algunos requisitos aún quedan por completar. Sin embargo, esta situación ha sido anticipada y se ha contemplado en forma de trabajo futuro. La construcción de este sistema ha demostrado ser un proceso complejo pero esclarecedor, revelando la importancia de la planificación minuciosa y la adaptabilidad en el desarrollo de soluciones tecnológicas de esta magnitud. En última instancia, este trabajo sienta una base sólida para la culminación exitosa de un sistema de “Golden Record” y proporciona valiosas lecciones para proyectos similares en el futuro.

## 6.2. Trabajo a Futuro

Es importante destacar que el microservicio encargado de las Reglas de Calidad se ha implementado en el proyecto, y existe una comunicación funcional entre el componente unificador y el servicio de calidad. Sin embargo, la funcionalidad de aplicar efectivamente estas reglas de calidad dentro del unificador no se ha implementado debido a las limitaciones de alcance previamente identificadas en el proyecto, que se discuten en la sección de riesgos. Se ha establecido que el uso de estas reglas de calidad, a pesar de no haber sido implementado, es de gran relevancia para garantizar la integridad y calidad de los datos en el Golden Record. En la actualidad, el Conector Sink juega un papel crucial al asegurarse de que la clave primaria (en este caso, el DNI de los clientes) coincida durante el proceso de actualización y creación de registros. Esta implementación es la única regla de calidad aplicada hasta el momento. Se reconoce que, en un futuro, la implementación completa de estas reglas de calidad será esencial para fortalecer aún más la integridad y coherencia de los datos, y se contempla su consideración en proyectos posteriores.

# Lista de Tablas

1.1. Riesgos y Mitigaciones para el Sistema de Golden Record. . . . .	4
1.2. Ocurrencia de Riesgos y Mitigaciones Aplicadas. . . . .	5
1.3. Horas Totales por Fase . . . . .	5
2.1. Puntuación de Arquitecturas para Implementar el Sistema de Golden Record . . . . .	9
2.2. Ejemplo de Regla de Canonización . . . . .	16
B.1. Reglas de Canonización . . . . .	43
C.1. Actualización de la dirección de facturación . . . . .	44
C.2. Primera actualización del teléfono . . . . .	44
C.3. Segunda actualización del telefono . . . . .	45
C.4. Borrado Servicio a Domicilio . . . . .	45

# Lista de Figuras

2.1. Data Warehouse . . . . .	7
2.2. Microservicios . . . . .	8
2.3. Almacenamiento Distribuido y Big Data . . . . .	8
2.4. Creación y consumo del Golden Record . . . . .	12
3.1. Diagrama de Arquitectura . . . . .	19
3.2. Diagrama de navegación de alto nivel . . . . .	24
4.1. Diagrama Secuencia Autenticación . . . . .	29
5.1. Mapping entre Fuente de Datos contactos y Golden Record . . . . .	32
5.2. Mapping entre Fuente de Datos domicilios y Golden Record . . . . .	32
5.3. Mapping entre Fuente de Datos de facturación y Golden Record . . . . .	32
5.4. Conectores Sink y Source de las Fuentes de Datos . . . . .	33
E.1. Tablero Kanban del front-end . . . . .	48
E.2. Tablero Kanban del back-end . . . . .	49
F.1. Pantalla login . . . . .	50
F.2. Pantalla Home . . . . .	50
F.3. Pantalla Calidad . . . . .	51
F.4. Pantalla Canonización . . . . .	51
F.5. Pantalla Golden Record . . . . .	52
F.6. Pantalla Edición Golden Record . . . . .	52
F.7. Pantalla Métricas . . . . .	52
F.8. Pantalla Grafana . . . . .	53
F.9. Pantalla Kibana . . . . .	53
G.1. Diagrama de Gantt del Trabajo de Fin de Grado (TFG) - mayo a agosto. . . . .	55
G.2. Diagrama de Gantt del Trabajo de Fin de Grado (TFG) - septiembre a diciembre. . . . .	55

# Bibliografía

- [1] «Golden Record,» 2011. dirección: <https://www.grcdi.nl/dqglossary/golden%20record.html>.
- [2] SYDLE. «E-commerce White-Label: ¿qué es? ¿Cómo se usa y qué ventajas tiene?» (), dirección: <https://www.sydle.com/es/blog/e-commerce-white-label-60a435455d8d6378b32ae27d>. (accessed: 02/11/2023).
- [3] «Building Data Civilizer Pipelines with an Advanced Workflow Engine,» dirección: <https://raulcastrofernandez.com/papers/icde2018-demo-civilizer.pdf>.
- [4] I. Muñoz-Reja, A. I. Caballero Gómez, C. Carretero y G. F., *Calidad De Datos*. Madrid: RAMA Editorial, 2018, <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5758915>.
- [5] D. Sundaram y M. Vidhya, «DATA LAKES-A NEW DATA REPOSITORY FOR BIG DATA ANALYTICS WORKLOADS,» *International Journal of Advanced Computer Research*, vol. 7, oct. de 2016.
- [6] «What is a Golden Customer Record in Master Data Management,» dirección: <https://www.stibosystems.com/blog/benefits-of-creating-golden-customer-records>.
- [7] «What is Debezium?» (2023), dirección: [https://debezium.io/documentation/faq/#what\\_is\\_debezium](https://debezium.io/documentation/faq/#what_is_debezium).
- [8] «Kafka Connect.» (2014 - 2023), dirección: <https://docs.confluent.io/platform/current/connect/index.html#:~:text=Kafka%20Connect%20is%20a%20tool,in%20and%20out%20of%20Kafka..>
- [9] «Polling in sql server.» (2021), dirección: <https://learn.microsoft.com/en-us/biztalk/adapters-and-accelerators/adapters/sql/polling-in-sql-server-using-the-sql-adapter>.
- [10] «Spring Cloud Netflix.» (2023), dirección: <https://cloud.spring.io/spring-cloud-netflix/reference/html/>.
- [11] «Spring Cloud Gateway.» (2023), dirección: <https://cloud.spring.io/spring-cloud-gateway/reference/html/>.
- [12] «JSON Web Tokens.» (2023), dirección: <https://auth0.com/docs/secure/tokens/json-web-tokens>.
- [13] «Prometheus data source.» (2023), dirección: <https://grafana.com/docs/grafana/latest/datasources/prometheus/>.
- [14] «Exporters.» (2023), dirección: <https://prometheus.io/docs/instrumenting/exporters/>.
- [15] «What is Elasticsearch?» (2022), dirección: <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>.
- [16] «Data Analysis with Kibana.» (2023), dirección: <https://www.elastic.co/es/training/data-analysis-with-kibana>.
- [17] «About Grafana.» (2023), dirección: <https://grafana.com/docs/grafana/latest/introduction/>.

- [18] «Connectors.» (2023), dirección: <https://debezium.io/documentation/reference/stable/connectors/index.html>.
- [19] «Spring Cloud OpenFeign.» (2023), dirección: <https://spring.io/projects/spring-cloud-openfeign>.
- [20] «What is ZooKeeper?» (2020), dirección: <https://zookeeper.apache.org/>.
- [21] «Kowl - A Web UI for Apache Kafka.» (2022), dirección: <https://github.com/theurichde/kowl#kowl---a-web-ui-for-apache-kafka>.
- [22] «MySQL Connector - Debezium Documentation.» (2023), dirección: <https://debezium.io/documentation/reference/stable/connectors/mysql.html>.
- [23] «PostgreSQL Connector - Debezium Documentation.» (2023), dirección: <https://debezium.io/documentation/reference/stable/connectors/postgresql.html>.
- [24] C. Cranford. «Debezium for Oracle - Part 1: Installation and Setup.» Publicado el 30 de septiembre de 2022. Disponible en: <https://debezium.io/blog/2022/09/30/debezium-oracle-series-part-1/>. (sep. de 2022), dirección: <https://debezium.io/blog/2022/09/30/debezium-oracle-series-part-1/>.
- [25] C. Cranford. «Debezium for Oracle - Part 2: Running the connector.» Publicado el 6 de octubre de 2022. Disponible en: <https://debezium.io/blog/2022/10/06/debezium-oracle-series-part-2/>. (oct. de 2022), dirección: <https://debezium.io/blog/2022/10/06/debezium-oracle-series-part-2/>.
- [26] «JDBC Connector - Debezium Documentation.» (2023), dirección: <https://debezium.io/documentation/reference/stable/connectors/jdbc.html>.
- [27] «JDBC Sink Connector - Oracle Event Hub Cloud Service Documentation.» (2022), dirección: <https://docs.oracle.com/en/cloud/paas/event-hub-cloud/admin-guide/jdbc-sink-connector.html>.

# A Ejemplo configuración de un Source Connector

Se ha incluido el siguiente fragmento proporcionado para ofrecer detalles específicos sobre un ejemplo de configuración de un conector Source en el contexto de la implementación de Kafka Connect con Debezium. El código detalla la configuración del conector Debezium para PostgreSQL, especificando parámetros clave como la dirección y el puerto de la base de datos, las credenciales de autenticación, la tabla a incluir en el flujo de eventos, el prefijo para los nombres de los temas en Kafka, y las configuraciones de los convertidores para serializar mensajes clave y de valor como JSON.

```
{
  "name": "contactos-source",
  "config": {
    "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
    "plugin.name": "pgoutput",
    "database.hostname": "postgres",
    "database.port": "5432",
    "database.user": "admin",
    "database.password": "****",
    "database.dbname": "contactos-db",
    "database.server.name": "postgres",
    "table.include.list": "public.contactos",
    "topic.prefix": "server1",
    "key.converter": "org.apache.kafka.connect.json.JsonConverter",
    "value.converter": "org.apache.kafka.connect.json.JsonConverter",
    "key.converter.schemas.enable": "false",
    "value.converter.schemas.enable": "false"
  }
}
```

- **“name”**: Define el nombre del conector. En este caso, se le ha dado el nombre “contactos-source”.
- **“connector.class”**: Especifica la clase del conector que se utilizará. En este ejemplo, se está utilizando el conector `io.debezium.connector.postgresql.PostgresConnector` para conectarse a una base de datos PostgreSQL.
- **“database.hostname”** y **“database.port”**: Indican la dirección y el puerto de la base de datos PostgreSQL a la que el conector se conectará.
- **“database.user”** y **“database.password”**: Representan las credenciales de autenticación del usuario que se utilizará para acceder a la base de datos.

- **“database.dbname”**: Indica el nombre de la base de datos específica (“contactos-db” en este caso) a la que se conectará el conector.
- **“table.include.list”**: Esta propiedad permite especificar las tablas de la base de datos que se deben incluir en el flujo de eventos. En este ejemplo, se incluirá la tabla `public.contactos`.
- **“topic.prefix”**: Define un prefijo que se utilizará al crear los nombres de los temas en Kafka donde se enviarán los eventos. En este caso, se usaría “server1” como prefijo.
- **“key.converter”** y **“value.converter”**: Indican los convertidores utilizados para serializar los mensajes clave y de valor que se envían a Kafka. En este ejemplo, se están utilizando convertidores JSON.
- **“key.converter.schemas.enable”** y **“value.converter.schemas.enable”**: Estas propiedades deshabilitan el uso de esquemas para los mensajes clave y de valor. Al configurarlas en “false”, se evita incluir información de esquema en los mensajes, lo que puede resultar útil para ciertos casos.

## B Reglas de Canonización del proyecto de ejemplo

En la tabla B.1 se pueden observar todas las reglas de canonización del proyecto de ejemplo. Aunque pueda parecer que hay muchas reglas con los campos origen y destino iguales, hay que hacerlo de esta forma, puesto que, si no se crea, se ignorará ese campo y no se transmitirá al Golden Record. Cabe destacar que las reglas de canonización son sensibles a la distinción entre mayúsculas y minúsculas. Por ejemplo, una regla que especifica la canonización de “email” a “eMail” tendrá un efecto diferente a una regla que canoniza “eMail” a “email”. Esto se debe a que el sistema de canonización considerará las letras en mayúsculas y minúsculas como entidades distintas, asegurando la precisión en la transformación de datos.

<b>Servicio</b>	<b>Origen</b>	<b>Destino</b>
facturacion	mail	eMail
facturacion	primerApellido	primerApellido
facturacion	segundoApellido	segundoApellido
facturacion	nombre	nombre
facturacion	direccionFacturacion	direccionFacturacion
facturacion	telefono	telefono
facturacion	dni	dni
servicioDomicilio	CORREO	eMail
servicioDomicilio	NUMEROTLF	telefono
servicioDomicilio	DNI	dni
servicioDomicilio	NOMBRE	nombre
servicioDomicilio	DIRECCIONENVIO	direccionEnvio
servicioDomicilio	PAIS	pais
contacto	nombre	nombre
contacto	primerapellido	primerApellido
contacto	segundoapellido	segundoApellido
contacto	email	eMail
contacto	direccion	direccion
contacto	telefono	telefono
contacto	dni	dni
contacto	residencia	residencia
contacto	intereses	intereses

Tabla B.1: Reglas de Canonización

## C Comportamiento

La fuente de datos crucial para nuestro sistema es el Centro de Contactos, y cualquier eliminación de datos se considerará de manera significativa.

Cuando cambiamos la dirección de facturación en la fuente de datos correspondiente, este ajuste no tiene un impacto crítico en el Golden Record, como se puede ver en la tabla C.1.

campos	valores
nombre	Elena
primerApellido	Garcia
direccion	Garcia
direccion	C/ Primavera 123
direccionEnvio	C/ Primavera 123
direccionFacturacion	<b>C/ Invierno 123</b>
dni	12345678A
eMail	elena.garcia88@gmail.com
telefono	612 345 678
pais	Colombia
residencia	Antioquia

Tabla C.1: Actualización de la dirección de facturación

Posteriormente, modificamos el número de teléfono en el Centro de Contactos (tabla C.2). Aunque esto no tiene un impacto directo en el Golden Record, si actualizamos el teléfono en el Servicio a Domicilio después, el número que permanecerá en el Golden Record será el último modificado (tabla C.3).

campos	valores
nombre	Elena
primerApellido	Garcia
segundoApellido	Garcia
direccion	C/ Primavera 123
direccionEnvio	C/ Primavera 123
direccionFacturacion	<b>C/ Invierno 123</b>
dni	12345678A
eMail	elena.garcia88@gmail.com
telefono	<b>666 666 666</b>
pais	Colombia
residencia	Antioquia

Tabla C.2: Primera actualización del teléfono

<b>campos</b>	<b>valores</b>
nombre	Elena
primerApellido	Garcia
direccion	Garcia
segundoApellido	C/ Primavera 123
direccionEnvio	C/ Primavera 123
direccionFacturacion	<b>C/ Invierno 123</b>
dni	12345678A
eMail	elena.garcia88@gmail.com
telefono	<b>111 111 111</b>
pais	Colombia
residencia	Antioquia

Tabla C.3: Segunda actualización del telefono

Al intentar eliminar el usuario de Elena en el Servicio a Domicilio, observamos un cambio sustancial, ya que perdemos valores únicos de la fuente de datos de dicho servicio (tabla C.4).

<b>campos</b>	<b>valores</b>
nombre	Elena
primerApellido	Garcia
direccion	Garcia
segundoApellido	C/ Primavera 123
direccionEnvio	
direccionFacturacion	<b>C/ Invierno 123</b>
dni	12345678A
eMail	elena.garcia88@gmail.com
telefono	<b>111 111 111</b>
pais	
residencia	Antioquia

Tabla C.4: Borrado Servicio a Domicilio

# D Requisitos

En el desarrollo de sistemas de información, la definición clara y detallada de los requisitos es esencial para garantizar el éxito del proyecto. Los requisitos pueden clasificarse en dos categorías principales: funcionales y no funcionales. Los requisitos funcionales describen las funciones específicas que el sistema debe realizar, mientras que los no funcionales abordan características más amplias relacionadas con el rendimiento, la integración y otros aspectos que afectan la operación del sistema.

## D.1. Requisitos Funcionales

- RF1.** El sistema permitirá canonizar y unificar la información de todas las fuentes de datos en un único Golden Record.
- RF2.** El sistema permitirá dar de alta a clientes en el Golden Record
- RF3.** El sistema permitirá dar de baja clientes en el Golden Record.
- RF4.** El sistema permitirá modificar clientes en el Golden Record.
- RF5.** El sistema permitirá obtener la información de un listado de clientes del Golden Record.
- RF6.** El sistema permitirá visualizar información sobre el Golden Record en un cuadro de mandos.
- RF7.** El sistema permitirá dar de alta reglas de canonización.
- RF8.** El sistema permitirá dar de baja reglas de canonización.
- RF9.** El sistema permitirá modificar reglas de canonización.
- RF10.** El sistema permitirá obtener las reglas de canonización.
- RF11.** El sistema permitirá dar de alta reglas de calidad.
- RF12.** El sistema permitirá dar de baja reglas de calidad.
- RF13.** El sistema permitirá modificar reglas de calidad.
- RF14.** El sistema permitirá obtener las reglas de calidad.
- RF15.** El sistema permitirá crear usuarios con rol.
- RF16.** El sistema permitirá iniciar sesión.

## D.2. Requisitos No Funcionales

- RNF1.** El sistema permitirá canonizar y unificar la información de todas las fuentes de datos en un único Golden Record.
- RNF2.** El sistema debe garantizar una buena trazabilidad de todas las operaciones que se lleven a cabo.
- RNF3.** El sistema debe garantizar una buena trazabilidad de la información manejada.
- RNF4.** El sistema debe garantizar un rendimiento óptimo poder sincronizarlos cuando se requiera.
- RNF5.** El sistema debe integrarse con otros servicios a través de API REST o tópicos de Kafka.
- RNF6.** El sistema debe ser flexible a la hora de añadir nuevas reglas de canonización o actualizar las ya existentes.
- RNF7.** El sistema recoge todos los cambios/accesos en la base de datos relacionados con la información personal.
- RNF8.** El sistema utilizará roles para impedir modificaciones/borrados no deseadas
- RNF9.** El sistema realizará copias de seguridad del Golden Record

## E Tableros Kanban

Los tableros Kanban, herramientas visuales para gestionar proyectos, han sido esenciales tanto en el desarrollo frontend (tabla E.1) como en el backend (tabla E.2). En el tablero del frontend, se coordinan tareas de diseño e interfaz, mientras que el del backend se centra en la lógica de negocio y gestión de datos.

La visualización clara promueve la colaboración, mejora la transparencia y ayuda en la identificación de tareas prioritarias. La flexibilidad permite adaptarse rápidamente a cambios, y la colaboración mejorada facilita la resolución eficiente de problemas. En resumen, los tableros Kanban optimizan la organización y eficiencia en el desarrollo, manteniendo un enfoque claro en la entrega de funcionalidades de calidad.

Al revisar el progreso, observamos que el frontend ha abordado un total de 7 tareas significativas. Es notable se ha enfocado especialmente en asignar una tarea por pantalla, a excepción de aquellas que presentaban similitudes notables, donde se ha optado por reciclar la pantalla correspondiente para optimizar el proceso de desarrollo.

Por otro lado, el backend ha enfrentado un conjunto más amplio de desafíos, con un total de 16 tareas. Estas tareas abarcan la creación de microservicios, configuraciones esenciales y la compatibilidad con diversas versiones de Spring Boot, entre otras consideraciones cruciales para el correcto funcionamiento del sistema. Este enfoque detallado en el backend refleja el compromiso con la robustez y la eficiencia en la arquitectura del proyecto, asegurando un rendimiento óptimo y una gestión eficaz de los recursos.

Con ambas áreas de desarrollo avanzando de manera coordinada, se ha logrado un sistema integral y exitoso que cumple con las expectativas y requisitos del proyecto.

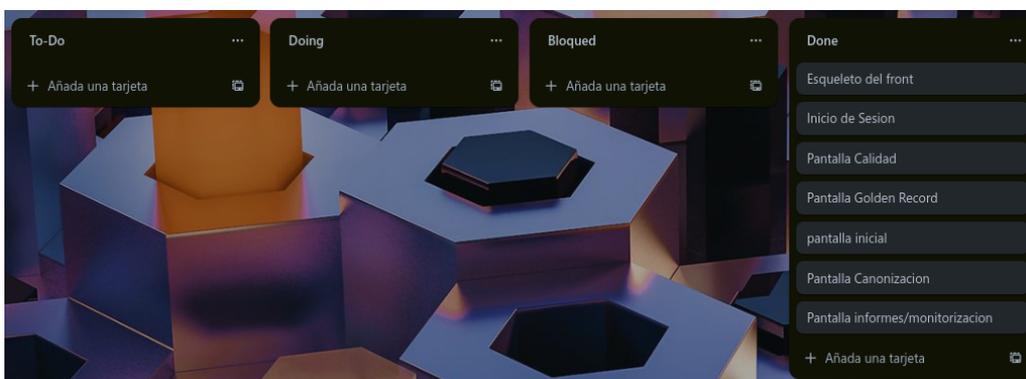


Figura E.1: Tablero Kanban del front-end

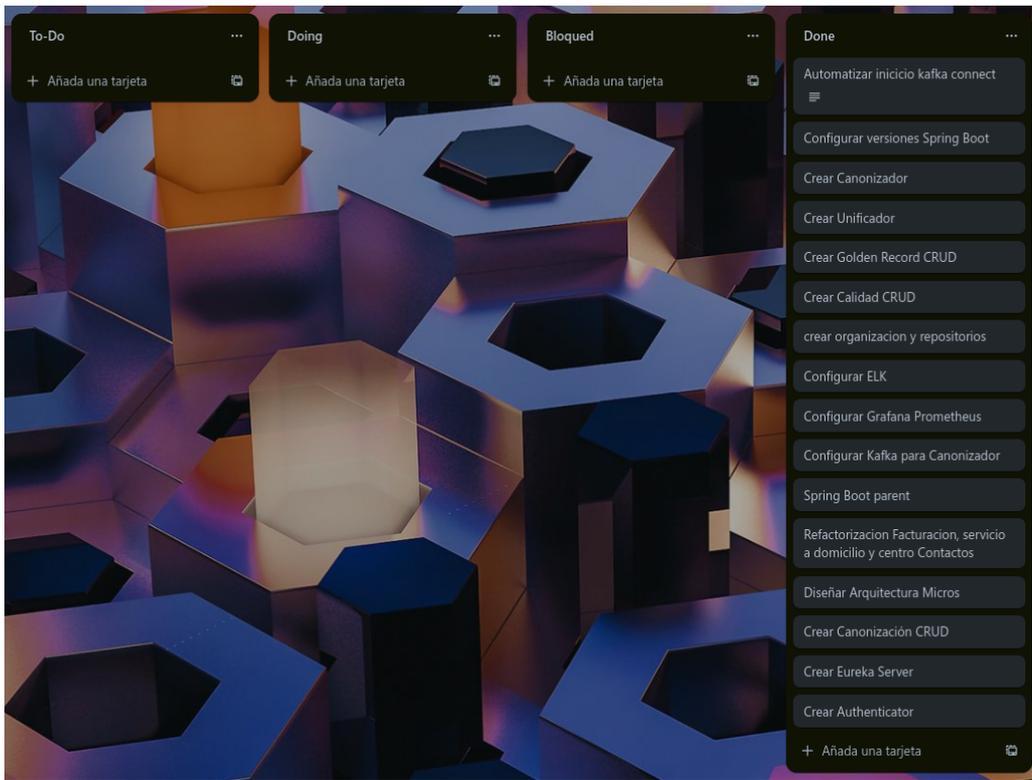


Figura E.2: Tablero Kanban del back-end

## F Ejemplo de Uso del Frontal

La pantalla F.1 representa la interfaz de inicio de sesión del sistema. Los elementos principales incluyen campos para ingresar el nombre de usuario y la contraseña.

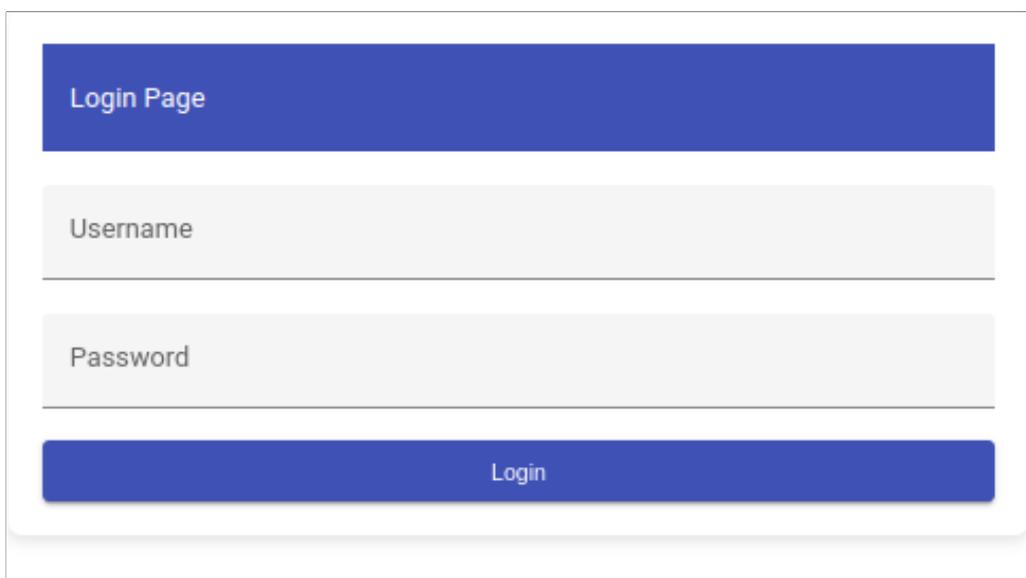


Figura F.1: Pantalla login

La pantalla F.2 después de iniciar sesión en el sistema. Muestra elementos clave, como ir a calidad, canonización, golden record o métricas



Figura F.2: Pantalla Home

La pantalla F.3 se relaciona con la calidad en el contexto de la aplicación o sistema. Ofrece información específica sobre aspectos de calidad y permite a los usuarios realizar acciones relacionadas con la gestión de la calidad.

### Reglas de Calidad

#### Crear nueva regla de calidad

Nombre:

Descripción:

Categoría:

Prioridad:

Condiciones:

Acción Correctiva:

Referencias:

Activa:

[Crear Regla de Calidad](#)

### Reglas de Calidad



Figura F.3: Pantalla Calidad

La pantalla F.4 permite a los usuarios generar reglas de canonización para los diferentes servicios, así como visualizar todas las reglas existentes.

#### Crear nueva regla

Servicio:

Origen:

Destino:

[Crear Regla](#)

#### Reglas de Canonización



Servicio	Origen	Destino	Eliminar Regla
facturacion	mail	eMail	 
facturacion	primerApellido	primerApellido	 
facturacion	segundoApellido	segundoApellido	 
facturacion	nombre	nombre	 
facturacion	direccionFacturacion	direccionFacturacion	 
facturacion	telefono	telefono	 
facturacion	dni	dni	 
servicioDomicilio	CORREG	eMail	 
servicioDomicilio	NUMEROTLF	telefono	 
servicioDomicilio	DNI	dni	 
servicioDomicilio	NOMBRE	nombre	 
servicioDomicilio	DIRECCIONENVI	direccionEnvio	 
servicioDomicilio	PAIS	pais	 
contacto	nombre	nombre	 
contacto	primerApellido	primerApellido	 
contacto	segundoApellido	segundoApellido	 

Figura F.4: Pantalla Canonización

La pantalla F.5 permite a los usuarios ver todos los datos existentes en el Golden Record, crear, editar (en específico F.6) y eliminar.

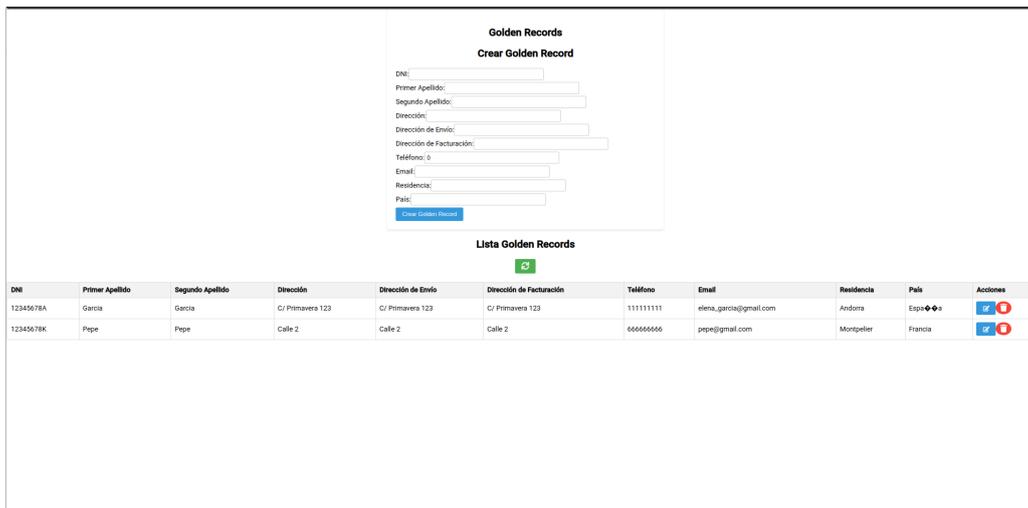


Figura F.5: Pantalla Golden Record

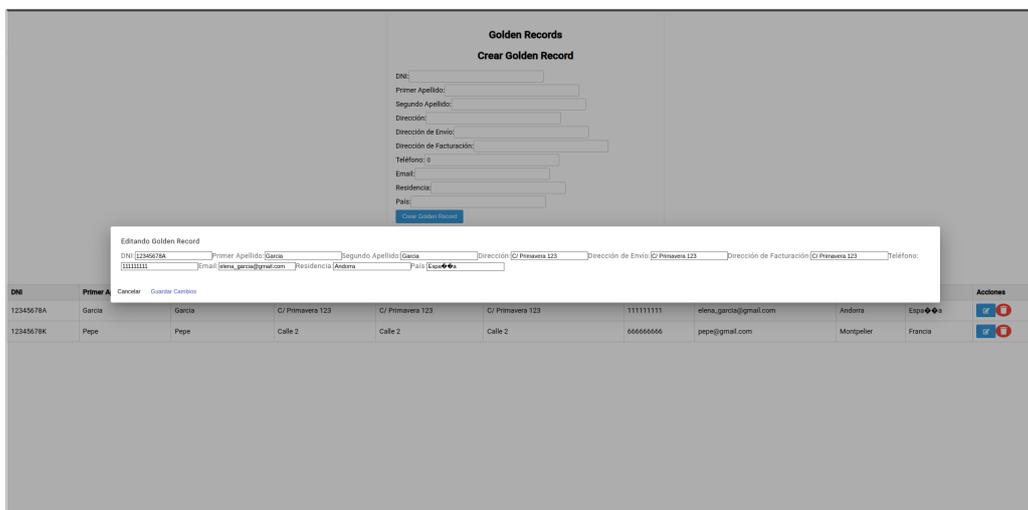


Figura F.6: Pantalla Edición Golden Record

La pantalla F.7 contiene accesos directos a kibana y grafana.



Figura F.7: Pantalla Métricas

## F.1. Monitorización

La pantalla F.8 contiene el dashboard por defecto de Grafana, se podrá personalizar a gusto de cada uno.

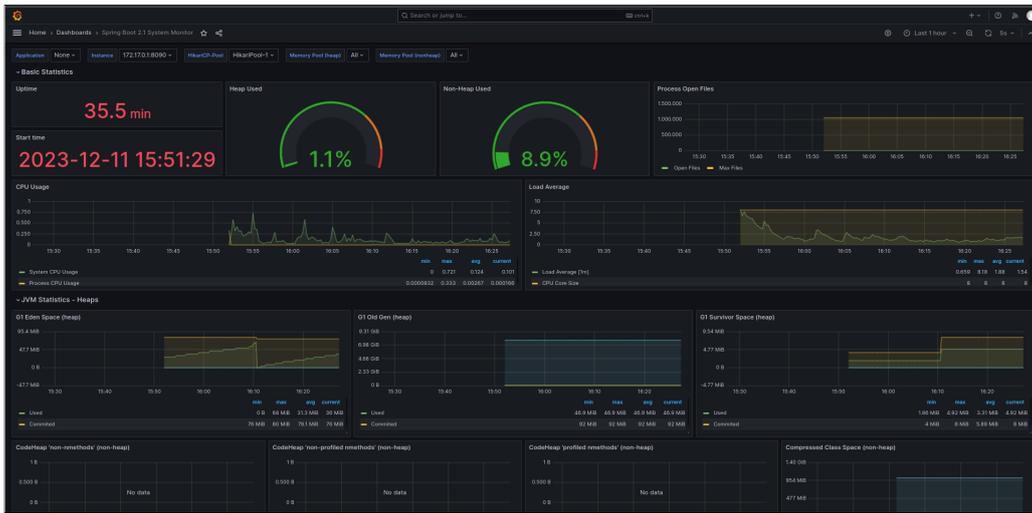


Figura F.8: Pantalla Grafana

La pantalla F.9 contiene el dashboard por defecto de Kibana, se podrá personalizar a gusto de cada uno.

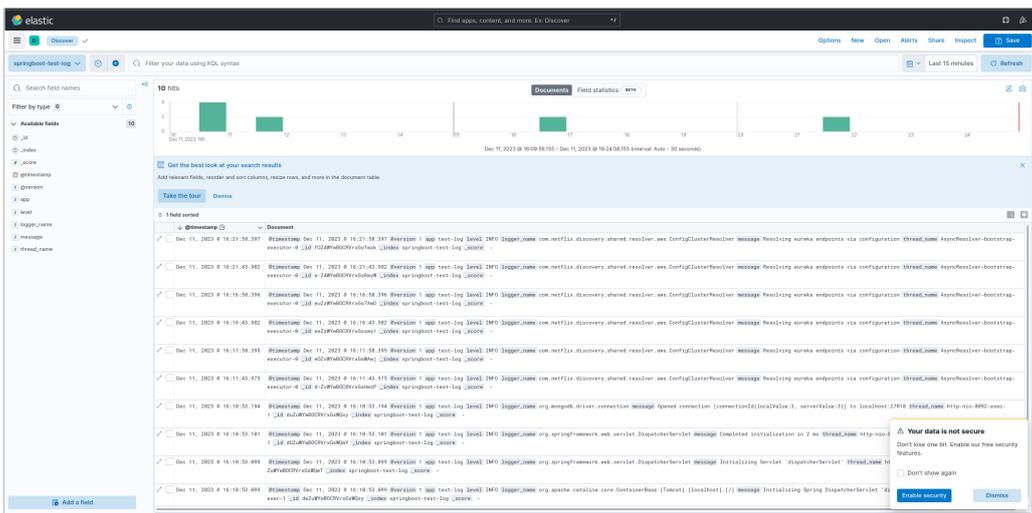


Figura F.9: Pantalla Kibana

## G Tiempo dedicado

El primer diagrama de Gantt, representado en la Figura G.1, abarca el periodo desde mayo hasta agosto y se enfoca en las fases iniciales del Trabajo de Fin de Grado (TFG). Durante el primer mes, se dedicó por completo al análisis y diseño del "Golden Record", una fase crítica para establecer los fundamentos y requisitos del proyecto. Este periodo inicial fue esencial para definir claramente los objetivos y asegurar una comprensión completa de los requisitos.

Posteriormente, se inició a la implementación del "Backend", ya que esta fase requería una cantidad significativa de trabajo y constituía una parte central del proyecto. Para garantizar una implementación precisa, se iniciaron las pruebas y se creó un proyecto de ejemplo. Esta estrategia permitió verificar que cada componente funcionara según lo requerido antes de avanzar en el desarrollo completo.

El segundo diagrama de Gantt, ilustrado en la Figura G.2, abarca el periodo de septiembre a diciembre y se centra en las fases avanzadas del TFG. Mientras se continuaba con el desarrollo del "Frontend" y "Backend", se comenzó simultáneamente a redactar la memoria del proyecto. Esta dualidad permitió avanzar en ambas áreas de manera paralela y garantizar la coherencia entre la implementación y la documentación.

A lo largo de las etapas finales, se realizaron correcciones continuas. Cada corrección se basó en pruebas, retroalimentación y análisis, lo que condujo gradualmente a la versión final del proyecto. La dedicación constante a las correcciones aseguró que el trabajo cumpliera con los estándares y requisitos establecidos.

Estas representaciones visuales y tabulares brindan una visión comprensiva del desarrollo del proyecto, desde la fase inicial de análisis y diseño hasta la iteración y correcciones finales que llevaron a la versión definitiva.

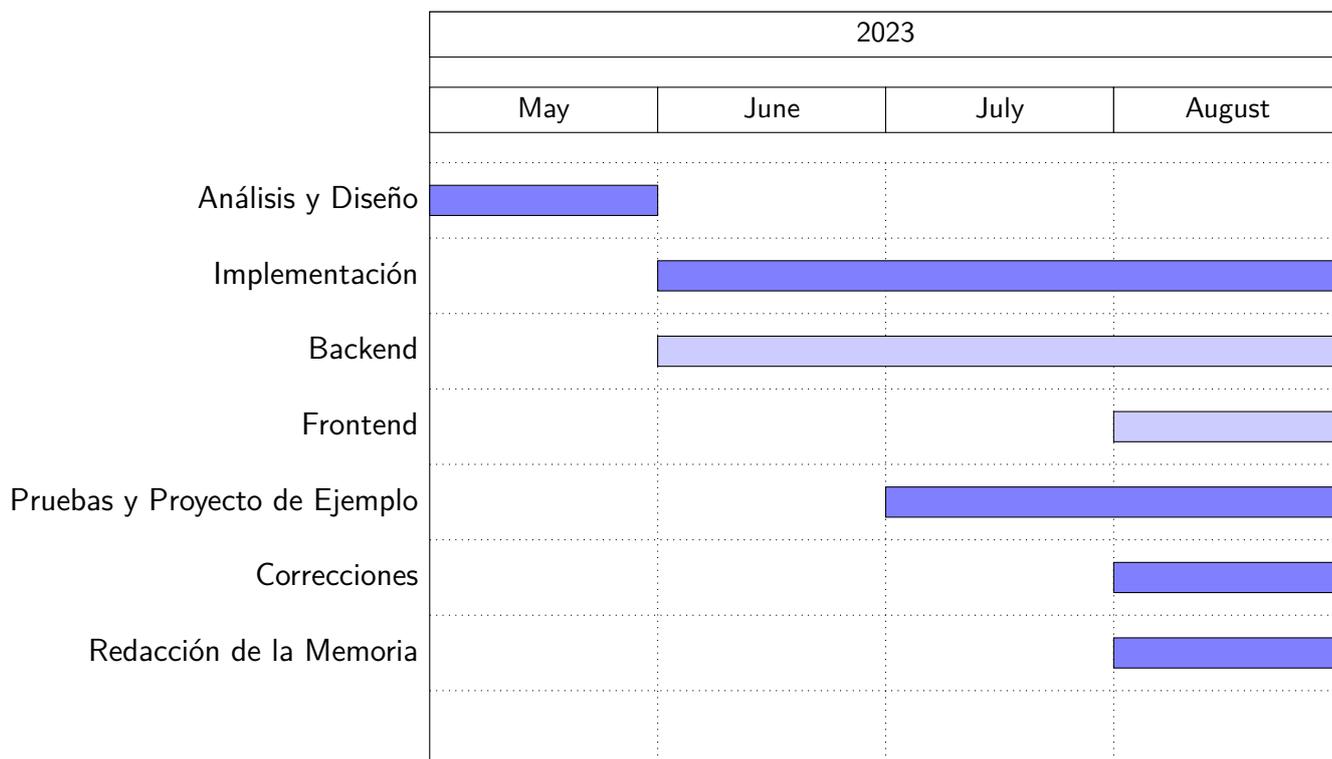


Figura G.1: Diagrama de Gantt del Trabajo de Fin de Grado (TFG) - mayo a agosto.

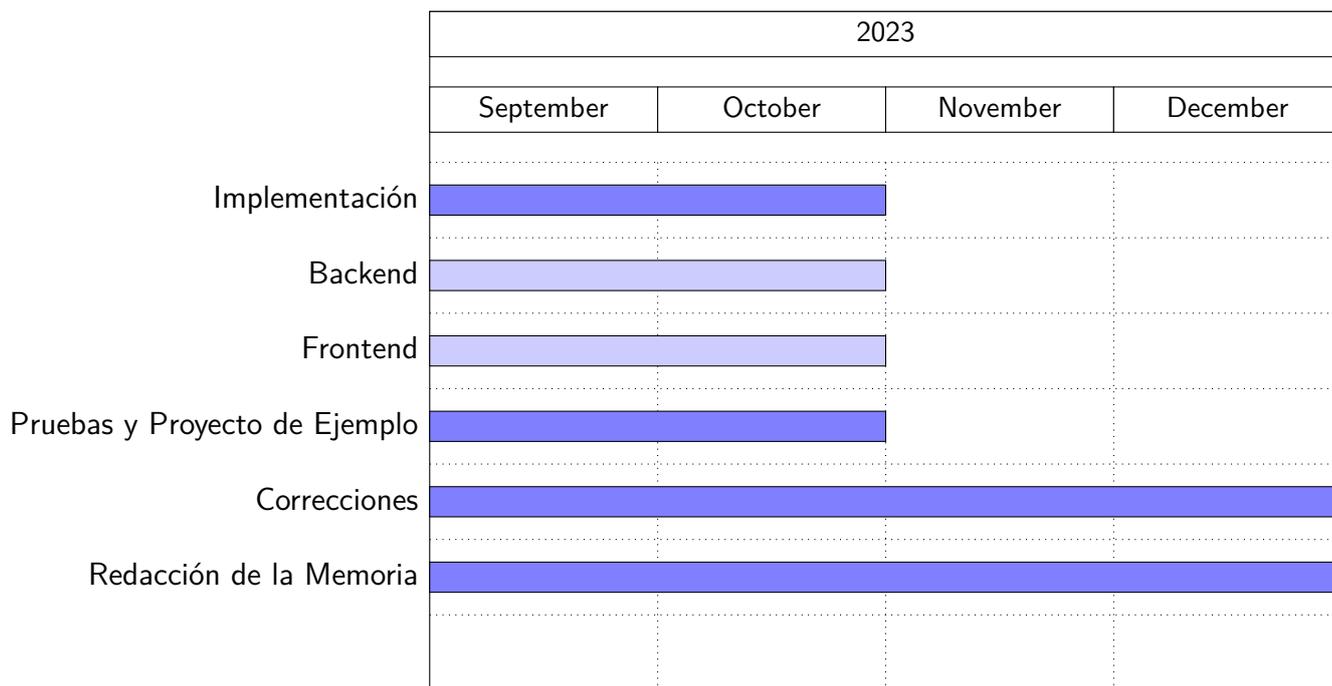


Figura G.2: Diagrama de Gantt del Trabajo de Fin de Grado (TFG) - septiembre a diciembre.