



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

Lenguajes y Sistemas Informáticos

TRABAJO FIN DE GRADO/MÁSTER Nº ???

Explotación, integración y visualización de múltiples fuentes de datos mediante un Data Lakehouse

Mier Montoto, Juan Francisco

TUTORES:

D. Augusto Alonso, Cristian

D. Morán Barbón, Jesús

D. Vázquez Faes, Eduardo

FECHA: julio 2024

Índice de contenido

| | |
|----------------------------------------------------------------|-----------|
| Índice de contenido | 1 |
| Índice de figuras | 3 |
| 1. Introducción | 4 |
| 1.1. Antecedentes | 4 |
| 1.2. Motivación | 6 |
| 1.3. La empresa | 7 |
| 1.4. Objetivos | 8 |
| 2. Fundamento teórico | 9 |
| 2.1. <i>Big data</i> | 9 |
| 2.2. Paradigmas de almacenamiento de datos | 10 |
| 2.2.1. Data warehouse | 10 |
| 2.2.2. Data lake | 10 |
| 2.2.3. Data lakehouse | 11 |
| 2.3. Procesos ETL | 12 |
| 2.3.1. Funcionamiento | 13 |
| 2.3.2. Alternativas | 15 |
| 2.4. Cuadros de mandos (<i>dashboards</i>) | 16 |
| 2.5. Infraestructura como código | 17 |
| 3. Descripción general del proyecto | 18 |
| 3.1. Partes interesadas (stakeholders) | 18 |
| 3.2. Alternativas existentes | 19 |
| 3.3. Descripción del proyecto | 20 |
| 4. Planificación del proyecto | 21 |
| 4.1. Metodología | 21 |
| 4.1.1. Scrum | 22 |
| 4.1.2. Visualización de la planificación | 23 |
| 4.1.3. Comunicación | 24 |
| 4.1.4. Plataformas de planificación | 24 |
| 4.2. Planificación inicial | 25 |
| 4.3. Presupuesto | 26 |
| 5. Diseño del sistema | 27 |
| 5.1. Estudio de alternativas | 27 |
| 5.1.1. Herramientas de despliegue de infraestructura | 27 |
| 5.1.2. Ingesta de datos | 28 |
| 5.2. Arquitectura del sistema | 29 |
| 5.3. Modelo de datos | 30 |
| 6. Implementación | 31 |
| 7. Resultados | 32 |

| | |
|-----------------------------------------|-----------|
| 8. Conclusiones y trabajo futuro | 33 |
| Bibliografía | 34 |

Índice de figuras

| | |
|-----------------------------------------------------------|----|
| 2.1. Fases de un proceso ETL | 13 |
| 2.2. Ejemplo de flujo con virtualización | 15 |
| 2.3. Diagrama de flujo de un proceso <i>ELT</i> | 15 |
| 4.1. Diagrama de la metodología <i>Scrum</i> | 22 |
| 4.2. Roadmap de apartados de la memoria | 24 |

1. Introducción

El proyecto que se presenta en este documento tiene como objetivo la automatización de despliegue de la infraestructura y procesos que permitan hacer un análisis masivo de datos. Para conseguir este objetivo, se analizan las necesidades de la empresa y sus antecedentes, se valoran las tecnologías y herramientas disponibles y se plantea una solución que permita la integración, almacenamiento y análisis de grandes volúmenes de datos, provenientes de múltiples fuentes y en diferentes formatos.

1.1. Antecedentes

Hoy en día, nos encontramos en una era donde la generación y almacenamiento de datos crece exponencialmente¹, un hecho que se ve reflejado en el ámbito empresarial. La diversidad de fuentes y formatos de estos datos introduce una complejidad significativa en su manejo, conocida como *heterogeneidad*², siendo las bases de datos, archivos de registros y APIs las fuentes más habituales.

El término *big data* describe este fenómeno de acumulación masiva de datos, cuya magnitud y complejidad sobrepasan las capacidades de los métodos de procesamiento convencionales. El *big data* por tres características principales: volumen, variedad y velocidad - su adecuada gestión y análisis pueden otorgar ventajas competitivas significativas a las empresas, tales como el descubrimiento de patrones ocultos, identificación de nuevas oportunidades de mercado y optimización de procesos de toma de decisiones.

Uno de los procesos que permite la extracción de esta información es la pirámide DIKW, [1] es un modelo que describe la relación entre los datos, la información, el conocimiento y la sabiduría. Según este modelo, los datos son la materia prima de la información, que a su vez es la materia prima del conocimiento, que a su vez es la materia prima de la sabiduría. Una organización sin los procesos adecuados para la gestión y análisis de estos datos, se enfrenta a importantes desafíos, como la dificultad para identificar patrones y tendencias, la toma de decisiones incorrectas y la pérdida de oportunidades de negocio. Por otro lado, una organización que logre extraer información valiosa de sus datos, podrá mejorar su eficiencia, aumentar su competitividad y adaptarse mejor a un entorno empresarial en constante cambio.

La evolución tecnológica ha propiciado el desarrollo de innovadoras herramientas y me-

¹<https://www.statista.com/statistics/871513/worldwide-data-created/>

²<https://www.sciencedirect.com/topics/computer-science/data-heterogeneity>

tecnologías diseñadas para enfrentar estos desafíos. Entre ellas, los *data lakes* (o *lagos de información*) se destacan por su capacidad para consolidar vastos volúmenes de datos heterogéneos, facilitando su posterior análisis y aprovechamiento de manera más efectiva.

Sin embargo, a pesar de que existen herramientas de almacenamiento, el proceso de integración, visualización y análisis de estos datos es una tarea desafiante, ya que requiere de una gran cantidad de recursos y de un tiempo de desarrollo considerable del que, normalmente, no se dispone en el ámbito empresarial.

Con la ingesta masiva de datos, se presentan nuevos problemas a la hora de analizar y obtener información de ellos:

- Sin la necesaria automatización y correcta aplicación de los procesos ETL, al tratarse de un crecimiento exponencial de los datos y, por lo tanto, de la fuerza de trabajo necesaria para manejarla, los resultados del análisis pueden ser incorrectos, lo que deriva en errores y decisiones de negocio equivocadas que impactan negativamente en la empresa.
- La heterogeneidad de los datos, tanto en formato como en origen, dificulta su consolidación y análisis.
- La masificación de información impide el análisis manual de los mismos, requiriendo resúmenes estadísticos o representaciones gráficas como *dashboards* para su correcta interpretación. La visualización de datos es una técnica que permite representar la información de manera visual, para facilitar su análisis y comprensión. La visualización de datos es una parte importante del proceso de análisis de datos, ya que permite identificar patrones, tendencias y anomalías en los datos de forma más rápida y sencilla.

1.2. Motivación

Okticket, como el resto de empresas, se enfrenta a la necesidad de gestionar y analizar grandes volúmenes de datos, provenientes de múltiples fuentes y en diferentes formatos. La correcta gestión y análisis de estos datos es fundamental para la toma de decisiones y para la mejora de los procesos internos de la empresa.

En la actualidad, la empresa dispone de una gran cantidad de datos que se encuentran en diferentes formatos y en diferentes ubicaciones, lo que dificulta su análisis y explotación. Por otra parte, se depende de la consulta manual o de servicios de terceros para poder analizar estos datos, lo que supone un coste adicional.

El proyecto surge de la necesidad de la empresa de extraer información y conocimiento de las múltiples y heterogéneas fuentes de datos de las que se disponen, tanto internas (e.g. bases de datos, archivos de registros, APIs, entre otros), como externas (e.g. APIs o datos de webs de terceros, datos de fuentes públicas...).

Además del uso interno, la empresa también quiere ofrecer a sus clientes la posibilidad de consultar estos datos de forma visual y sencilla, para que puedan analizarlos y explotarlos de forma autónoma, lo que supondría un valor añadido para los mismos.

1.3. La empresa

Okticket es una startup nacida en Gijón en 2017 cuyo producto principal es un servicio software que escanea automáticamente de tickets y notas de gastos lo que permite reducir los costes y el tiempo que invierten las empresas en contabilizar y manejar los gastos de viaje de los profesionales.

La empresa tienen su sede principal en el Parque Tecnológico de Gijón, aunque cuenta con un número de sedes creciente en varios países, como Francia, Portugal o, más recientemente, México. En esta oficina principal se encuentran los departamentos de ventas y marketing, así como el equipo de desarrollo y soporte.

Okticket es una de las empresas que más crecen tanto del sector como del propio Parque Tecnológico. Debido a este rápido crecimiento, el equipo está en constante desarrollo y cambio, tanto aquí en España como en el resto de sedes. Este crecimiento se refleja en la recepción de un gran número de galardones y reconocimientos.^{3 4 5 6}

La parte principal del negocio es el núcleo del software como servicio (Software as a Service en inglés, en adelante *SaaS*), es decir, la aplicación completa tanto para administradores como para empleados. Este SaaS se oferta a empresas de cualquier tamaño, cuyo precio final varía en función del número de usuarios, las características e integraciones que requiera la empresa cliente y el soporte que se ofrezca.

Recientemente se han añadido nuevas propuestas a la cartera de servicios ofertada por Okticket, como la OKTCard - una tarjeta inteligente que gestiona automáticamente los gastos, así como la inclusión de nuevos “módulos” de gestión de gastos y viajes.

Debido a todo este crecimiento, la empresa maneja una gran cantidad de datos importantes que se encuentran en diferentes formatos y en diferentes ubicaciones, lo que dificulta su análisis y explotación. Por otra parte, se depende de la consulta manual o de servicios de terceros para poder analizar estos datos, lo que supone un coste adicional.

³ Okticket en el especial startups 2023 de Forbes (LinkedIn)

⁴ Arcelor y Okticket, premios nacional de Ingeniería Informática (EL COMERCIO)

⁵ Okticket recibe el sello Pyme Innovadora (okticket.es)

⁶ Okticket, empresa emergente certificada (okticket.es)

1.4. Objetivos

El objetivo del proyecto es la creación de un proceso que permita el despliegue automático de una infraestructura de datos que permita la integración, almacenamiento y análisis de grandes volúmenes de datos, provenientes de múltiples fuentes y en diferentes formatos. La infraestructura de datos debe ser escalable, flexible y robusta, para poder adaptarse a las necesidades cambiantes de la empresa.

La infraestructura de datos debe permitir la integración de datos de múltiples fuentes, tanto internas como externas, y en diferentes formatos, como bases de datos, archivos de registros, APIs, entre otros. La integración de datos debe ser automática y programable, para poder automatizar el proceso de ingestión de datos y reducir el tiempo y los costes asociados.

Pese a que el entregable principal de este proyecto es la creación de una infraestructura, se esperan también otros entregables en forma de herramientas de software, como *scripts*, que faciliten la integración y análisis de los datos, así como la visualización de los mismos.

2. Fundamento teórico

En este capítulo se presentan los conceptos y términos fundamentales que se utilizan en el proyecto, para proporcionar una base teórica sobre la que se desarrolla el trabajo. Se discuten los conceptos de *data lake*, *procesos ETL* y *dashboards*, que son fundamentales para el desarrollo del proyecto.

2.1. *Big data*

El término *big data* se refiere a la gestión y análisis de grandes volúmenes de datos que no pueden ser tratados de manera convencional. Estos datos se caracterizan por su volumen, velocidad y variedad, lo que hace que sea difícil procesarlos con las herramientas tradicionales de gestión de datos.

El *big data* se caracteriza por tres características principales: volumen, variedad y velocidad. Estas características se conocen como las “tres V” del *big data*:

- **Volumen:** se refiere a la cantidad de datos que se generan y almacenan en un determinado periodo de tiempo. El volumen de datos que se maneja en el *big data* es mucho mayor que el que se maneja en los sistemas tradicionales de gestión de datos.
- **Variedad:** se refiere a la diversidad de fuentes y formatos de los datos que se manejan en el *big data*. Los datos pueden provenir de diferentes fuentes, como bases de datos, archivos de registros y APIs, y pueden estar en diferentes formatos, como texto, imágenes, audio, vídeo, etc.
- **Velocidad:** se refiere a la velocidad a la que se generan y se procesan los datos. En este ámbito, los datos se generan y se procesan a una velocidad mucho mayor que en los sistemas tradicionales de gestión de datos.

El *big data* es un fenómeno que ha surgido en los últimos años debido al crecimiento exponencial de los datos que se generan y almacenan en todo el mundo. Este crecimiento ha sido impulsado por el aumento de la conectividad y el uso de dispositivos móviles, que han permitido a las empresas y a los usuarios generar y almacenar grandes cantidades de datos de manera más eficiente.

2.2. Paradigmas de almacenamiento de datos

En el ámbito del *big data*, existen diferentes paradigmas de almacenamiento de datos que se utilizan para almacenar y analizar grandes cantidades de información. Los tres paradigmas a considerar para este proyecto son los *data warehouses*, los *data lakes* y los *data lakehouses*.

2.2.1. Data warehouse

Un *data warehouse*¹, también conocido en español como almacén de datos, es una base de datos que se utiliza para almacenar y analizar grandes cantidades de datos de manera eficiente. Los almacenes de datos proporcionan acceso rápido y compatible con plataformas de consultas (como SQL) a grandes cantidades de datos, lo que permite a los analistas y a los científicos de datos realizar análisis complejos sobre los datos almacenados.

Todos los datos almacenados en un *data warehouse* se encuentran en un formato común, para lo que se aplican procesos ETL (extracción, transformación y carga) que transforman los datos de diferentes fuentes en un formato común. Esto significa que la información se encuentra en un formato o esquema optimizado y específico, lo que facilita su manipulación y análisis pero limita la flexibilidad al acceso de los datos y genera costes adicionales en el caso de tener que modificar o transferir los mismos para su uso.

2.2.2. Data lake

Los *data lakes*² son almacenes de datos que guardan grandes cantidades de datos de manera no estructurada [2]. En el ámbito de una empresa, un *data lake* contiene datos de diferentes fuentes de valor no considerado hasta su análisis, de manera que su explotación posterior y su análisis no depende de una estructuración y transformación compleja, reduciendo los costes de los procesos ETL derivados, una flujo de tareas que se aplican sobre la información para ingestarla. Esto no quiere decir que no se apliquen estos procesos a los datos, sino que se aplican de manera más flexible y básica que en otras estructuras de almacenamiento de datos con esquemas predefinidos, como los *data warehouses*. [3]

A diferencia de los *data warehouses*, los *data lakes* no tienen un esquema definido, lo que permite almacenar datos *heterogéneos*. Esto permite almacenar grandes cantidades de información sin tener que definir un esquema de antemano, lo que puede ser útil en aquellos casos en los que no se conoce la estructura de los datos que se van a almacenar.

¹<https://aws.amazon.com/es/data-warehouse/>

²<https://aws.amazon.com/es/what-is/data-lake/>

Estas características de los *data lakes* hacen que sean más atractivos en el sector empresarial de cara al análisis de información, en contraste con las estructuras planteadas normalmente en el campo de la investigación académica.

Para consultar esta gran cantidad de datos almacenados, se suelen utilizar técnicas de visualización de datos, como los *dashboards*, herramientas de visualización que permiten observar los datos de manera sencilla y eficiente.

2.2.3. Data lakehouse

Los *data lakehouses* son una combinación funcional de los dos paradigmas vistos anteriormente, los *data lakes* y los *data warehouses*. Los *data lakehouses* permiten almacenar datos tanto de manera estructurada como no estructurada, lo que facilita aprovechar la información al contar con una única estructura de bajo coste que ofrece a los usuarios que lo necesiten explorar y analizar los datos según sus necesidades.

2.3. Procesos ETL

Si anteriormente se presentaban los distintos paradigmas de almacenamiento de datos, para su creación y mantenimiento se requieren aplicar unos ciertos procesos que permitan la correcta ingesta y almacenamiento de los datos. Estos procesos se conocen como *procesos ETL*.

Formalmente se definen los procesos ETL [2] como procesos que combinan datos de múltiples fuentes en un único destino, transformando los datos en un formato común. Estos procesos se utilizan para extraer datos de diferentes fuentes, transformarlos en un formato común y cargarlos en un destino común, como puede ser un *data lake*.

Los procesos ETL, fundamentales en el ámbito de la gestión de datos, presentan atributos distintivos que facilitan la integración eficaz de información procedente de diversas fuentes:

- **Adaptabilidad:** los procesos ETL deben adaptarse a la estructura de los datos de la fuente de origen, ya que dichas fuentes pueden tener diferentes estructuras y tener tipos de datos diferentes (la característica de *heterogeneidad* de los datos que ya se ha mencionado).
- **Escalabilidad:** otra de las características clave de los procesos ETL es que sean escalables, ya que los datos que se muestran en los dashboards suelen ser datos que se generan de manera continua, y por lo tanto los procesos ETL deben ser capaces de procesar grandes cantidades de datos de manera eficiente. En ocasiones, los procesos ETL se pueden realizar en *streaming*, lo que significa que los datos se procesan en tiempo real a medida que se generan.
- **Eficiencia:** los procesos ETL deben ser eficientes, puesto que el tiempo de procesamiento de los datos es un factor vital en el ámbito del *big data*. Los procesos ETL deben ser capaces de procesar grandes cantidades de datos en un tiempo razonable para que los datos estén disponibles en el menor tiempo posible.
- **Fiabilidad:** la fiabilidad es un componente crítico de todo el flujo de datos, ya que estos se utilizan para la toma de decisiones importantes de cualquier empresa. Los procesos ETL deben ser capaces de procesar los datos de manera fiable y consistente, para que los datos que se visualicen y analicen posteriormente sean correctos y fiables.

2.3.1. Funcionamiento

Los procesos ETL se dividen en tres fases principales: (1) *Extraer*, (2) *Transformar* y (3) *Cargar*, como se muestra en el siguiente diagrama:

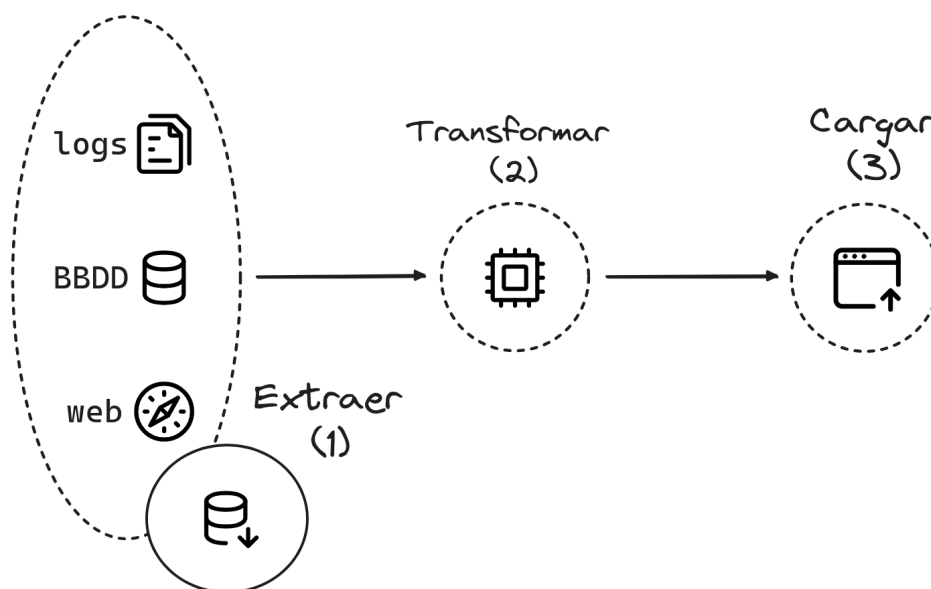


Figura 2.1: Fases de un proceso ETL

Como entrada, se tienen datos presuntamente heterogéneos que no se pueden analizar de manera eficiente. Tras aplicar todos los pasos de las fases anteriores, se obtiene como salida un conjunto de datos homogéneos y listos para ser analizados en el destino indicado (en el caso de este proyecto, un *data lake*).

Extracción (1) En este proceso se obtienen los datos de las fuentes de datos, que pueden ser bases de datos, logs, APIs, etc. En esta fase, se pueden aplicar filtros para extraer solo los datos que se necesiten, y se pueden extraer datos de múltiples fuentes *heterogéneas*.

La fase de extracción se puede realizar de dos formas: continua o incremental. Una extracción incremental se realiza de manera periódica, por ejemplo, cada hora, cada día o cada semana, y se extraen los datos que se han generado desde la última extracción. Esto es útil cuando los datos se generan de manera periódica y se necesita mantener actualizada la información. Por otro lado, en una extracción continua se extraen los datos en tiempo real según se van generando. Esto puede ser útil para procesar datos que se generan en tiempo real, como logs o datos de sensores.

Transformación (2) Durante esta fase, se transforman los datos extraídos en la fase anterior, normalmente aplicándoles un proceso de limpieza y transformación a un formato común. En este paso, se pueden aplicar diferentes operaciones a los datos, como la limpieza, la agregación, la normalización, la conversión de formatos, etc.

Uno de los tipos de transformaciones de datos más comunes es la limpieza, que consiste en la revisión y corrección de los datos extraídos, para asegurar que se almacena información correcta y consistentes. Durante esta fase se contemplan operaciones más complejas, como pueden ser la agregación de datos, la conversión de formatos, la normalización de datos, el cifrado, etc. La limpieza de datos puede ser una tarea muy sencilla, como la eliminación de caracteres delimitadores, o muy compleja, como la corrección de errores en los datos o la detección de duplicados.

Estos procesos de transformación son vitales cuando el sistema maneja una gran cantidad de datos heterogéneos de múltiples fuentes de manera simultánea, como puede ser el caso de un *data lake* o un *data warehouse*. En el caso del primero, no es necesaria la transformación de los datos a un formato común, pero si otros procesos clave como la limpieza y la normalización de los datos, entre otros.

Carga (3) En este proceso se vuelcan los datos transformados en el destino final. Frecuentemente, los datos se almacenan, dependiendo del paradigma de almacenamiento elegido, en una *data lake*, *data warehouse* o *data lakehouse* para su posterior análisis.

2.3.2. Alternativas

Aunque lo más común es el flujo anteriormente explicado de *extracción*, *transformación* y *carga*, existen algunos flujos y procesos alternativos que evitan algunos de estos pasos, normalmente en casos específicos que se beneficien del cambio:

- **Virtualización de datos:** capa virtual de abstracción que permite acceder a los datos de las fuentes sin necesidad de extraerlos. Esto permite ahorrar espacio de almacenamiento y tiempo de procesamiento, pero suele ser menos eficiente en términos de rendimiento y no es compatible con todas las arquitecturas de datos.

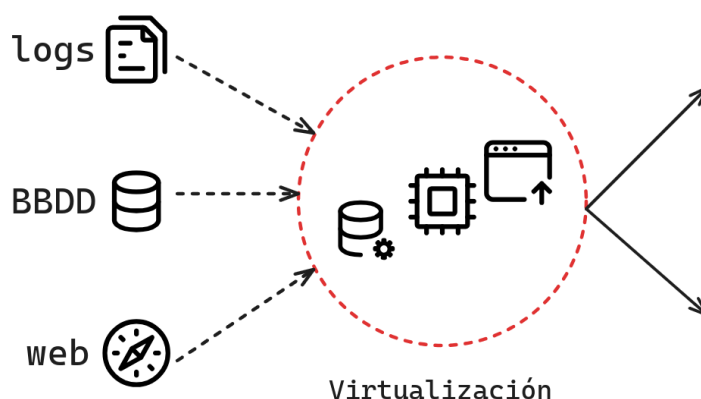


Figura 2.2: Ejemplo de flujo con virtualización

- **Proceso *ELT*³:** en lugar de transformar los datos antes de cargarlos en el destino, se cargan los datos en bruto y se transforman en el destino. Funciona bien para grandes conjuntos de datos sin estructura que requieran una carga (o recarga) continua, aunque, al igual que la virtualización, puede ser menos eficiente o incompatible con algunas arquitecturas de datos, como los *data warehouses*.

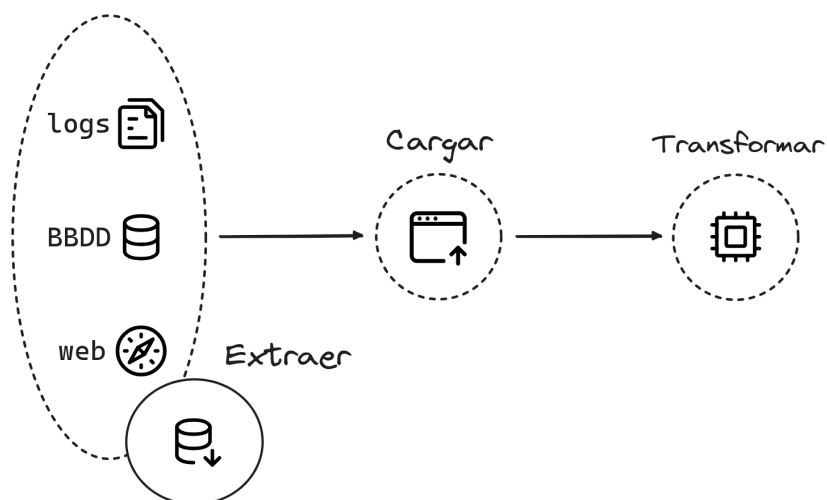


Figura 2.3: Diagrama de flujo de un proceso *ELT*

³<https://www.ibm.com/topics/elt>

2.4. Cuadros de mandos (*dashboards*)

Definición Los cuadros de mandos, en adelante *dashboards*, es un término que se utiliza para referirse a cualquier interfaz gráfica que muestre información relevante de manera visual sobre un proceso o negocio. Aunque el término se utiliza en muchos ámbitos: indicadores comerciales, de producción, de marketing, de calidad, de recursos humanos. . . , en este proyecto se utilizará en el ámbito de la monitorización de sistemas y procesos de negocio.

En el ámbito de este proyecto, los *dashboards* reflejan en tiempo real el rendimiento de actividades o procesos de negocio, y se utilizan para tomar decisiones informadas basándose en los mismos. Por ejemplo, el *dashboard* de una empresa digital puede mostrar desde el rendimiento de la arquitectura en tiempo real hasta el número de ventas conseguidas, y permitir a los directivos tomar decisiones informadas sobre el futuro de la empresa (e.g. necesidad de aumentar la capacidad de los servidores, lanzar una campaña de marketing, etc.).

Características Los *dashboards* cuentan con una serie de características que los hacen útiles para la toma de decisiones: [2]

- **Visualización de datos:** es la característica fundamental de cualquier *dashboard*, y aquella que determina su utilidad. La visualización de datos es la ciencia de presentar los datos de manera que se pueda extraer información útil y realizar decisiones informadas sobre ellos. Un buen *dashboard* cuenta con gráficas, tablas, indicadores, etc. que permiten al usuario entender la información que se está presentando con un conocimiento técnico mínimo.
- **Interactividad y personalización:** un *dashboard* debe permitir al usuario interactuar con los datos (filtrarlos, ordenarlos, profundizar en ellos...) y ajustar la información que se muestra sobre cada proceso o negocio que se esté evaluando (granularidad de la información). Esta capacidad asegura que el *dashboard* se adapte tanto a las necesidades actuales como a las evoluciones futuras de lo que se esté analizando.
- **Accesibilidad y portabilidad:** un *dashboard* debe ser accesible desde una variedad de situaciones y dispositivos, manteniendo su funcionalidad y forma. Aunque normalmente los *dashboards* se analizan en pantallas grandes, es importante que también se puedan consultar en otras circunstancias, como dispositivos móviles.

Dashboards planteados Para el sistema que se describe, se plantean dos tipos de dashboards diferentes:

- **Dashboards internos:** que reflejan el rendimiento de la plataforma en tiempo real. Estos dashboards están destinados al uso interno de la empresa, y permiten a los empleados monitorizar el rendimiento de la plataforma y tomar decisiones informadas sobre su mantenimiento y evolución.
- **Dashboards externos:** que reflejan el rendimiento de las ventas y permiten a los clientes tomar decisiones informadas sobre su negocio. Estos dashboards están enfocados a los clientes de la empresa, y permite a los mismos obtener información relevante sobre su negocio que tenga Okticket.

2.5. Infraestructura como código

La infraestructura como código (o *IaaS* por sus siglas en inglés) es una práctica que consiste en gestionar la infraestructura de un sistema de manera automática y programática, utilizando código en lugar de configuraciones manuales. La infraestructura como código permite gestionar la infraestructura de manera eficiente y escalable, y facilita la creación y el mantenimiento de entornos de desarrollo y producción.

En el ámbito de este proyecto, la infraestructura como código se utiliza para gestionar la infraestructura de los sistemas de almacenamiento y procesamiento de datos, como los *data lakes* y los *data warehouses*. La infraestructura como código permite gestionar la infraestructura de manera eficiente y escalable, y facilita la creación y el mantenimiento de entornos de desarrollo y producción.

3. Descripción general del proyecto

Esta sección describe el proyecto en términos generales, incluyendo una descripción de los problemas que se pretenden resolver, las partes interesadas en el proyecto y una valoración de las alternativas consideradas.

3.1. Partes interesadas (stakeholders)

Las partes interesadas en el proyecto son aquellas personas o entidades que tienen un interés en el mismo, ya sea porque se ven afectadas por el resultado del proyecto, o porque tienen algún tipo de interés en el mismo. Las partes interesadas en este proyecto son las siguientes:

1. **Okticket:** la empresa es la principal parte interesada en el proyecto, ya que es la que se beneficiará directamente de los resultados del mismo, así como de las oportunidades de negocio que se abren con la explotación de los datos. Dentro de la empresa, se pueden identificar dos entidades:
 - **Equipo de desarrollo de la empresa:** el equipo de desarrollo es otra parte interesada en el proyecto, ya que son los encargados de llevar a cabo la implementación del sistema y de garantizar su correcto funcionamiento, además de gestionar el soporte de servicio a nivel técnico.
 - **Equipo de soporte de la empresa:** el sistema planteado ahorraría tiempo al equipo de soporte, ya que les permitiría analizar los datos de forma más eficiente e identificar problemas antes de que tener que resolver las peticiones de los clientes afectados a nivel básico.
2. **Clientes:** los clientes de la empresa también son partes interesadas, puesto que se beneficiarán de los nuevos servicios que se ofrecen, como los dashboards de negocio que se han descrito anteriormente. Estos clientes no son necesariamente los usuarios finales, sino los administradores y gestores de las empresas que utilizan Okticket como herramienta de gestión de gastos.
3. **Investigador y desarrollador (*Mier Montoto, Juan Francisco*):** el desarrollador del proyecto tiene la oportunidad de aplicar los conocimientos adquiridos en el desarrollo de un proyecto real, y de adquirir nuevos conocimientos en el proceso.

3.2. Alternativas existentes

3.3. Descripción del proyecto

4. Planificación del proyecto

La planificación de un proyecto es fundamental para su correcto funcionamiento y desarrollo, dentro de los plazos y costes establecidos. Se presenta un primer apartado de metodología, un segundo apartado con la planificación inicial para posteriormente inferir en base a esta el presupuesto inicial

4.1. Metodología

En este capítulo se aborda la metodología adoptada para el desarrollo del proyecto, fundamentada en principios ágiles y enfocada en la entrega continua de valor. La elección de *Scrum*, una metodología que permite elaborar productos software de manera incremental, revisando el producto continuamente y adaptándolo a las necesidades del cliente, como marco de trabajo subraya nuestro compromiso con la adaptabilidad y la mejora continua del producto.

La estructura de este capítulo se organiza en torno a la descripción detallada de la metodología *Scrum*, la visualización de la planificación y las estrategias de comunicación adoptadas. A través de esta metodología, buscamos optimizar los recursos disponibles, ajustarnos a los plazos establecidos y garantizar la calidad del producto final.

La implementación de *Scrum* se complementa con herramientas de visualización y gestión de proyectos, como los tableros *Kanban*, que facilitan la organización y seguimiento de las tareas. Además, se pone especial énfasis en la comunicación efectiva dentro del equipo de desarrollo y con los stakeholders, asegurando así una alineación constante con los objetivos del proyecto. Existen otras variantes de los tableros *Kanban* que se pueden utilizar para visualizar el progreso de las tareas, pero en este proyecto se ha elegido esta alternativa para facilitar la visualización de las tareas y su estado.

Este enfoque metodológico no solo refleja la planificación y ejecución del proyecto, sino que también establece las bases para una gestión eficaz, adaptativa y orientada a resultados.

4.1.1. Scrum

Para la planificación del proyecto se ha escogido *Scrum*, una metodología “ágil” que se basa en la realización de iteraciones cortas y en la adaptación a los cambios. La metodología *Scrum* se estructura en *sprints* (iteraciones cortas de una duración fija), en las que se llevan a cabo una serie de tareas que se han planificado previamente.

El primer paso de la metodología *Scrum* es la creación de un *product backlog*, una lista ordenada de las tareas a realizar durante el desarrollo del producto, a partir de los requisitos del sistema, que a su vez son una versión refinada de los requisitos iniciales del proyecto. A partir de este *product backlog* se planifican las tareas que se llevarán a cabo en cada *sprint*, de manera que sea posible cumplir con los objetivos del proyecto en el tiempo establecido.

A diferencia de metodologías tradicionales o *en cascada*, *Scrum* permite la adaptación a los cambios y la mejora continua del producto, ya que se revisa y se adapta en cada *sprint* según las necesidades del cliente y del equipo de desarrollo. Por otro lado, *Scrum* se diferencia de otras metodologías ágiles como *XP* en que no se centra tanto en las prácticas de desarrollo, sino en la gestión del proyecto y en la entrega de valor al cliente.

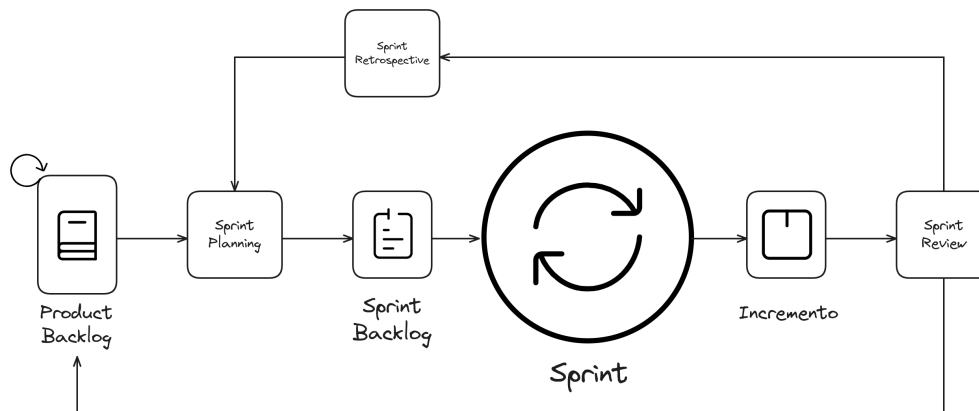


Figura 4.1: Diagrama de la metodología *Scrum*

Roles En *Scrum* se distinguen tres roles principales:

- **Product Owner:** es la persona responsable de definir los requisitos del producto y de priorizar las tareas del *product backlog*. Es el enlace entre el equipo de desarrollo y el cliente, y es el responsable de garantizar que el producto cumple con las expectativas del cliente. En el caso de este proyecto, el *Product Owner* es el director tecnológico de la empresa.
- **Scrum Master:** es la persona responsable de garantizar que el equipo de desarrollo

sigue la metodología *Scrum* y de eliminar los obstáculos que puedan surgir durante el desarrollo del proyecto. El *Scrum Master* es el encargado de organizar las reuniones diarias y de asegurar que el equipo de desarrollo cumple con los plazos y los objetivos del proyecto. En este proyecto, el *Scrum Master* son los tutores académicos del proyecto.

- **Equipo de desarrollo:** es el equipo encargado de llevar a cabo las tareas del *product backlog* y de entregar el producto final. El equipo de desarrollo es autoorganizado y multidisciplinario, y se organiza en torno a las tareas que se van a realizar en cada *sprint*. Para este proyecto, el “equipo” de desarrollo está constituido únicamente por el alumno, que se encarga de todas las tareas de desarrollo y documentación.
- **Stakeholders:** son las partes interesadas en el proyecto, como los clientes, los usuarios finales y los patrocinadores, que desconocen el proceso de desarrollo pero tienen un interés en el producto final y en su correcto funcionamiento.

4.1.2. Visualización de la planificación

Para la visualización de la planificación se ha utilizado la herramienta de gestión de proyectos de *GitHub*, que permite múltiples visualizaciones de tareas e *issues* en tableros separados.

- Se utiliza un tablero de *requisitos* al estilo *Kanban* para visualizar los requisitos del proyecto y su estado, siguiendo con la metodología *Scrum*. Un tablero *Kanban* es una herramienta visual que permite gestionar el flujo de trabajo de un proyecto por “sprints”, dividiendo las tareas en columnas y moviéndolas de una columna a otra según su estado.
- Se utiliza un *roadmap* de apartados de la memoria, donde se visualiza su estado y sus fechas límite. Este *roadmap* no está relacionado con la metodología *Scrum*, sino que se ha creado a propósito para facilitar la visualización del progreso de cada sección y de la memoria en general.



Figura 4.2: Roadmap de apartados de la memoria

4.1.3. Comunicación

La comunicación con los tutores y con el equipo de desarrollo se considera fundamental para el correcto desarrollo del proyecto. Puesto que el trabajo se desarrolla de manera presencial en la oficina de la empresa, la comunicación con el equipo de desarrollo se realiza de manera frecuente y directa, mientras que la comunicación con los tutores se realiza de manera remota pero igual de frecuente, manteniendo el contacto mediante correo electrónico y Teams para pedir revisiones e informar sobre el estado del trabajo en todo momento.

4.1.4. Plataformas de planificación

Con el objetivo de facilitar las tareas de desarrollo y cumplimentar los requisitos por parte de la empresa, se utilizan las siguientes plataformas y herramientas de desarrollo para la fabricación del proyecto:

- **GitHub:** Plataforma de desarrollo colaborativo para el desarrollo del proyecto. Se utiliza para la gestión de tareas, seguimiento de desarrollo, documentación y colaboración.
- **Atlassian suite (Jira, Bitbucket):** Suite de herramientas de gestión de proyectos y desarrollo colaborativo. Se utiliza para el desarrollo y documentación del proyecto de parte de la empresa.
- **Microsoft Teams:** Herramienta de comunicación y colaboración en tiempo real.
- **Microsoft Outlook:** Herramienta de comunicación por correo electrónico.

4.2. Planificación inicial

4.3. Presupuesto

5. Diseño del sistema

5.1. Estudio de alternativas

5.1.1. Herramientas de despliegue de infraestructura

A la hora de desplegar la infraestructura de un proyecto, se consideran varias herramientas populares que permiten automatizar este proceso. Entre ellas, se encuentran *Terraform*, *AWS CloudFormation* y *Ansible*.

A continuación, se describen brevemente estas herramientas y se comparan sus características.

5.1.1.1. Alternativas

Terraform *Terraform* es una herramienta de código abierto desarrollada por *HashiCorp* que permite definir y desplegar infraestructura de forma declarativa. *Terraform* permite definir la infraestructura en un archivo de configuración JSON, que describe los recursos que se desean crear y sus dependencias. A partir de este archivo, *Terraform* se encarga de desplegar los recursos en el proveedor de nube especificado, que en el caso de este proyecto es *AWS*.

AWS CloudFormation *AWS CloudFormation* es un servicio de *Amazon Web Services* similar a *Terraform* que permite definir y desplegar infraestructura en la nube de forma declarativa. *AWS CloudFormation* permite definir la infraestructura o bien mediante un archivo de configuración (en formato JSON o YAML), o bien gráficamente mediante diagramas.

Ansible *Ansible* es una herramienta multi-propósito de automatización de tareas entre las que se incluye el despliegue y orquestación de infraestructura. Se trata de una herramienta desarrollada por *Red Hat* que permite definir la infraestructura mediante *playbooks* escritos en YAML, que describen las tareas a realizar y los servidores en los que se deben ejecutar.

5.1.1.2. Comparación

Comenzando la comparativa por la facilidad de uso de cada herramienta, *Terraform* es la alternativa planteada más fácil de usar, ya que permite definir la infraestructura en un archivo con sintaxis sencilla y desplegarla con un solo comando. Por otro lado, *AWS CloudFormation* es un poco más complejo de usar, ya que requiere definir la infraestructura en un archivo de

configuración o en un diagrama, y luego desplegarla mediante la consola de AWS. *Ansible* es más complejo de usar, ya que requiere definir la infraestructura mediante *playbooks* y ejecutarlos en los servidores, pero es más flexible y potente que las otras dos herramientas.

Mientras que *Terraform* y *Ansible* son herramientas *multi-cloud*, lo que significa que funcionan con cualquier proveedor de nube, *AWS CloudFormation* es una herramienta específica de AWS y solo funciona con sus servicios, lo que puede suponer tanto una ventaja como una desventaja, dependiendo de las necesidades del proyecto. Puesto que este proyecto se despliega en AWS sin intención de migrar a otra plataforma, pero a la vez no se requiere el uso de servicios específicos de AWS, no se considera una ventaja significativa.

5.1.1.3. Decisión

Ninguna de las alternativas consideradas es claramente superior a las demás, ya que se tratan de herramientas con características y funcionalidades similares y una amplia acogida en la industria. Sin embargo, se ha decidido utilizar *Terraform* para el despliegue de la infraestructura de este proyecto, ya que es la herramienta más “sencilla” de utilizar, la que mejor se podría adaptar a las necesidades del proyecto y la única que ya se ha utilizado en proyectos anteriores dentro de la empresa.

5.1.2. Ingesta de datos

5.2. Arquitectura del sistema

5.3. Modelo de datos

6. Implementación

7. Resultados

8. Conclusiones y trabajo futuro

Bibliografía

- [1] Wikipedia contributors, “Dikw pyramid — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=DIKW_pyramid&oldid=1211227190, 2024. [Online; accessed 7-April-2024].
- [2] J. Mier, “Presentación de datos: dashboards y procesos ETL.” Primera entrega de teoría de la asignatura Inteligencia de Negocio, EPI Gijón, curso 23-24, 2023.
- [3] P. Khine and Z. Wang, “Data lake: a new ideology in big data era,” *ITM Web of Conferences*, vol. 17, p. 03025, 01 2018.
- [4] J. Mier, “latexTemplate.” <https://github.com/miermontoto/latexTemplate>, 2024. Plantilla de L^AT_EX personal para trabajos académicos.
- [5] J. Mier, “Minería de anomalías.” Segunda entrega de teoría de la asignatura Inteligencia de Negocio, EPI Gijón, curso 23-24, 2024.