Atribución-NoComercial-SinDerivadas 4.0 Internacional

Esta licencia es la más restrictiva de las seis licencias principales Creative Commons, permitiendo a otras solo descargar sus obras y compartirlas con otras siempre y cuando den crédito, pero no pueden cambiarlas de forma alguna ni usarlas de forma comercial.

http://creativecommons.org/licenses/by-nc-nd/4.0

UNIVERSIDAD NACIONAL "SAN LUIS GONZAGA"

FACULTAD DE INGENIERÍA DE SISTEMAS

DIRECCIÓN DE INVESTIGACIÓN

EVALUACIÓN DE ORIGINALIDAD

CONSTANCIA

El que suscribe, deja constancia que se ha realizado el análisis con el software de verificación de similitud al documento cuyo título es:

Estandarización de Jobs para la migración a la nube en la empresa Bluetab

Presentado por:

- **CORDOVA RAMOS ROSA ISELA**

BACHILLER en **PREGRADO** de la facultad de Ingeniería de Sistemas. El resultado obtenido es **(porcentaje de similitud 4%)** por el cual se otorga el calificativo de:

APROBADO, según el Reglamento de Evaluación de la Originalidad.

Se adjunta al presente el reporte de evaluación con el software de verificación de originalidad.

Ica, 10 de febrero del 2025

Dr. JAVIER ORLANDO GUTIÉRREZ FERREYRA
Director de la Unidad de Investigación
Facultad de Ingeniería de Sistemas

UNIVERSIDAD NACIONAL “SAN LUIS GONZAGA”

VICERRECTORADO DE INVESTIGACIÓN

FACULTAD DE INGENIERÍA DE SISTEMAS

Línea de investigación: Ciencias naturales, ingeniería y tecnologías sostenibles

Estandarización de Jobs para la migración a la nube en la empresa Bluetab

Autor: ROSA ISELA CORDOVA RAMOS

INFORME FINAL DE TRABAJO DE SUFICIENCIA PROFESIONAL PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO DE SISTEMAS

Ica, Perú

2024

Dedicatoria

Dedico este trabajo a mis padres, por su amor y apoyo incondicional; a mi hermano Juan, por ser una fuente constante de inspiración; y a mis amigos, por su comprensión y aliento en los momentos difíciles.

También, agradezco a mis profesores y mentores, cuya guía y enseñanzas han sido cruciales para mi crecimiento académico y profesional.

Agradezco de manera especial a San Judas Tadeo, por ser mi guía espiritual y darme fuerza en los momentos de incertidumbre.

Rosa

AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido a la realización de este trabajo de suficiencia profesional.

En primer lugar, agradezco a San Judas Tadeo, por ser mi guía espiritual y darme fuerza en los momentos de incertidumbre. Su apoyo ha sido una fuente de fortaleza y esperanza en mi camino.

A mis padres, por su amor, apoyo incondicional y por inculcarme los valores del esfuerzo y la perseverancia. A mi hermano Juan, por su constante inspiración y compañía en los momentos difíciles.

A mis amigos, por su comprensión y por brindarme palabras de aliento cuando más las necesitaba. Su apoyo ha sido fundamental para superar los desafíos a lo largo de este camino.

A mis profesores y mentores, quienes con su dedicación y sabiduría me han guiado y proporcionado las herramientas necesarias para mi desarrollo académico y profesional. Su influencia ha sido esencial para alcanzar este logro.

También, quiero agradecer a mis compañeros de estudio y colegas, por su colaboración y por compartir sus conocimientos y experiencias. Su camaradería ha hecho que este viaje sea más enriquecedor y llevadero.

A todas aquellas personas que, de una manera u otra, han formado parte de mi formación y crecimiento, les extiendo mi más profundo agradecimiento. Este trabajo es resultado del esfuerzo colectivo y del apoyo recibido a lo largo de los años.

Gracias a todos por creer en mí y por acompañarme en cada paso de este camino.

Rosa

Índice

Dedicatoria ii

Agradecimientos iii

Índice iv

- Índice de Contenido iv
- Índice de Tablas vi
- Índice de Figuras vii

Resumen viii

Abstract ix

INTRODUCCION 1

Capítulo I: INFORMACION DE LA INSTITUCION DONDE SE DESARROLLO LA EXPERIENCIA 11

Capítulo II: TRAYECTORIA PROFESIONAL 13

Capítulo III: APLICACION PROFESIONAL 15

3.1. Situación problemática 15

3.2. Proyecto de solución 16

3.2.1. Solución y objetivos 18

3.2.2. Planificación 19

3.3. Implementación del proyecto 22

3.3.1. Descripción del proyecto 22

3.3.2. Alcance del proyecto 22

3.3.3. Implementación del proyecto 23

3.3.4. Procesos de estandarización 27

Capítulo IV: APORTES A LA INSTITUCION 49

CONCLUSIONES 51

RECOMENDACIONES 53

REFERENCIAS BIBLIOGRÁFICAS 55

ANEXOS 56

INDICE DE TABLAS

Tabla I: Resumen de experiencia Profesional 14

Tabla II: Calendarización de las tareas del proyecto 21

Tabla III: Capital Humano requerido 21

Tabla IV: Requisitos del sistema de control 23

INDICE DE FIGURAS

Fig.1: Organigrama 12

Fig.2: Flujo general de la creación de un desarrollo 19

Fig.3: Plan de trabajo del proyecto 20

Fig.4: Flujo de datos 21

Fig.5: Inventario del stock 24

Fig.6: Imagen de un archivo conf de ingesta 25

Fig.7: Imagen de un archivo conf de Hammurabi 25

Fig.8: Tablero de Scalpy 27

Fig.9: Estado de ticket 28

Fig.10: Comentario muy riguroso a un desarrollo 29

Fig.11: Resultado de la automatización que facilito a la validación 31

Fig.12: Comparación de Artifactory dev VS producción 31

Fig.13: Simulación de la lógica de obtención de rutas 33

Fig. 14: Captura de una parte del algoritmo de Pesos 34

Fig.15: Ticket para el cambio en el ambiente de desarrollo 36

Fig.16: Ticket para el cambio en el ambiente de producción 37

Fig.17: Imagen de cómo funcionaba una ingesta 38

Fig.18: Imagen de parámetros obligatorios 39

Fig.19: Imagen de la ruta del esquema a cómo debería apuntar 39

Fig.20: Imagen del flags en el output 39

Fig.21: Imagen de filter en la transformación 40

Fig.22: Imagen de envío a bibucket 41

Fig.23: Convertura de testing 43

Fig.24: Estructura de pyspark Estandarizado 44

Fig.25: Evidencia de un job de spark compactor 46

Fig.26: Evidencia de parámetros en Dataproc 46

Fig.27: Dashboard general de Inventario de Datos 47

Fig.28: Dashboard particular de Estandarización 48

RESUMEN

Bluetab, una empresa de tecnología con una infraestructura IT significativa, enfrenta desafíos debido a su actual entorno on-premise. La gestión de jobs en su infraestructura existente es ineficiente, con problemas de escalabilidad, alta carga operativa y costos elevados, se definió un marco estandarizado para la migración de los jobs a la nube, incluyendo la identificación, clasificación y documentación de los procesos actuales; se mejoró la planificación y ejecución de la migración para garantizar una transición fluida, minimizando interrupciones y riesgos. Asimismo, se implementaron herramientas y prácticas para una gestión y monitoreo eficientes de los jobs en la nube, asegurando rendimiento óptimo y rápida resolución de problemas; se desarrolló un marco estandarizado que incluye la identificación de los jobs críticos, evaluación de sus requisitos y la documentación necesaria para su migración y se elaboró un plan detallado para la migración que incluye una fase de preparación, la migración en sí y una fase de validación post-migración. Se definieron roles, responsabilidades y cronogramas para cada etapa. Finalmente se implementaron herramientas de monitoreo avanzadas como Amazon CloudWatch y AWS X-Ray para supervisar el rendimiento y el estado de los jobs. Se configuraron dashboards y alertas para facilitar la gestión proactiva y la resolución de problemas.

ABSTRACT

Bluetab, a technology company with a significant IT infrastructure, faces challenges due to its current on-premise environment. The management of jobs in its existing infrastructure is inefficient, with scalability problems, high operational load and high costs. A standardized framework was defined for the migration of jobs to the cloud, including the identification, classification and documentation of current processes; Migration planning and execution was improved to ensure a smooth transition, minimizing disruptions and risks. Likewise, tools and practices were implemented for efficient management and monitoring of jobs in the cloud, ensuring optimal performance and rapid problem resolution; A standardized framework was developed that includes the identification of critical jobs, evaluation of their requirements and the documentation necessary for their migration and a detailed plan for the migration was prepared that includes a preparation phase, the migration itself and a validation phase. post-migration. Roles, responsibilities and schedules were defined for each stage. Finally, advanced monitoring tools such as Amazon CloudWatch and AWS X-Ray were implemented to monitor the performance and status of jobs. Dashboards and alerts were configured to facilitate proactive management and problem resolution.

INTRODUCCION

En la era digital actual, la migración a la nube se ha convertido en una prioridad estratégica para muchas organizaciones, permitiendo una mayor flexibilidad, escalabilidad y eficiencia en la gestión de datos y aplicaciones. Bluetab, una empresa líder en soluciones tecnológicas, ha reconocido la necesidad de adoptar este enfoque para mantenerse competitiva y alineada con las tendencias globales. Sin embargo, el proceso de migración a la nube implica desafíos técnicos y operativos que requieren un enfoque sistemático y estandarizado para garantizar su éxito.

El presente trabajo de titulación, basado en la experiencia profesional adquirida en Bluetab, se centra en la **estandarización de jobs** para la migración a la nube, un aspecto crucial para asegurar que las tareas automatizadas, o jobs, se ejecuten de manera eficiente y sin interrupciones en el nuevo entorno. La estandarización no solo mejora la consistencia y la calidad de las operaciones, sino que también facilita la gestión y el mantenimiento de los sistemas migrados.

Este estudio tiene como objetivo desarrollar un marco estandarizado que optimice el proceso de migración de jobs a la nube, abordando tanto los aspectos técnicos como las mejores prácticas de la industria. A lo largo del trabajo, se examinarán los desafíos específicos enfrentados por Bluetab durante la migración, se evaluarán las soluciones implementadas y se propondrán recomendaciones para futuras migraciones. De esta manera, se espera contribuir al fortalecimiento de la infraestructura tecnológica de Bluetab y su capacidad para ofrecer servicios de calidad en un entorno cada vez más digitalizado.

Bluetab, una empresa líder en el sector de soluciones tecnológicas, ha decidido migrar sus operaciones y datos a la nube con el objetivo de mejorar la escalabilidad, flexibilidad y eficiencia de sus servicios. Sin embargo, el proceso de migración no está exento de desafíos, especialmente en lo que respecta a la **gestión de jobs automatizados**. Estos jobs, que son tareas programadas esenciales para el funcionamiento continuo de las operaciones de la empresa, deben ser trasladados y adaptados al nuevo entorno en la nube.

Actualmente, Bluetab enfrenta la falta de un marco estandarizado para la migración de estos jobs, lo que ha generado problemas como inconsistencias en la ejecución, errores de configuración, y dificultades en el mantenimiento y gestión de los sistemas migrados. Estas dificultades no solo afectan la eficiencia operativa, sino que también ponen en riesgo la calidad de los servicios ofrecidos a los clientes.

Ante esta situación, surge la necesidad de desarrollar un enfoque estandarizado que permita la migración efectiva de jobs a la nube, garantizando su correcta operación en el nuevo entorno. Este proyecto de suficiencia profesional tiene como objetivo abordar este desafío, proponiendo soluciones que optimicen el proceso de migración y aseguren la continuidad operativa y la calidad del servicio en Bluetab.

Asimismo, se prevee:

1. **Desarrollar un marco estandarizado para la migración de jobs a la nube**: Crear un conjunto de directrices y procedimientos uniformes que guíen la migración de jobs, asegurando la consistencia y calidad en la transición a la nube.

   Desarrollo de un Marco Estandarizado para la Migración de Jobs a la Nube

   Definición de Requisitos y Alcance

   Evaluación Inicial: Realizar un análisis exhaustivo de los jobs actuales, incluyendo su complejidad, dependencias, y requisitos de recursos. Identificar los objetivos específicos de la migración y establecer criterios de éxito.

   Documentación de Requisitos: Recolectar requisitos funcionales y no funcionales de los stakeholders para entender las expectativas y necesidades del negocio en cuanto a la migración.

   Diseño del Marco de Migración

   Estrategia de Migración: Definir una estrategia de migración que incluya fases, cronograma, y métodos. Ejemplos de métodos incluyen la migración por lotes o la migración en fases.

   Estandarización de Jobs: Crear plantillas y formatos estandarizados para la configuración y el despliegue de jobs en la nube. Esto incluye scripts, configuraciones de entorno y parámetros de ejecución.

   Herramientas y Tecnologías: Seleccionar herramientas y tecnologías para la migración y la gestión de jobs en la nube, como plataformas de orquestación, herramientas de automatización y servicios en la nube específicos (por ejemplo, AWS Batch, Azure Data Factory).

   Planificación y Preparación

   Evaluación de Recursos: Identificar y preparar los recursos necesarios, como infraestructura en la nube, redes y almacenamiento.

Pruebas de Compatibilidad: Realizar pruebas de compatibilidad para asegurar que los jobs funcionen correctamente en el entorno de la nube. Esto puede incluir pruebas piloto o la ejecución de trabajos en un entorno de prueba.

Documentación y Procedimientos: Desarrollar documentación detallada y procedimientos operativos estándar (SOP) para la migración de jobs, incluyendo guías paso a paso para la configuración, ejecución y monitoreo de los jobs.

Ejecución de la Migración

Despliegue Gradual: Implementar los jobs en la nube de manera gradual para minimizar el riesgo y asegurar una transición suave. Esto puede implicar la migración de jobs menos críticos primero, antes de pasar a los más críticos.

Monitoreo en Tiempo Real: Monitorear el desempeño y la funcionalidad de los jobs migrados en tiempo real para identificar y solucionar cualquier problema que surja durante la migración.

Control de Calidad: Implementar mecanismos de control de calidad para verificar que los jobs se están ejecutando correctamente y que los resultados son los esperados.

Capacitación y Soporte

Entrenamiento del Personal: Capacitar al personal técnico en el uso de las nuevas herramientas y prácticas para la gestión de jobs en la nube. Esto puede incluir sesiones de formación, talleres y materiales de aprendizaje.

Soporte Continuo: Proveer soporte continuo durante y después de la migración para resolver problemas y ajustar configuraciones según sea necesario.

Evaluación y Optimización

Revisión Post-Migración: Evaluar el éxito de la migración mediante métricas de rendimiento y feedback de los usuarios. Identificar áreas de mejora y ajustar el marco según los resultados.

Optimización de Procesos: Implementar mejoras continuas en el proceso de gestión de jobs en la nube, basándose en los resultados de la evaluación y las necesidades cambiantes del negocio.

2. **Optimizar la planificación y ejecución de la migración**: Implementar herramientas y técnicas que faciliten la planificación efectiva y la ejecución precisa de los jobs en la nube, reduciendo el tiempo de inactividad y los errores durante la migración.

Para optimizar la planificación y ejecución de la migración de jobs a la nube, es fundamental seguir un enfoque estructurado y estratégico.

A. Análisis de Requisitos

Evaluación de Jobs Actuales: Realizar un inventario de todos los jobs que necesitan ser migrados, incluyendo el tamaño de los datos, la complejidad del procesamiento, y las dependencias entre jobs.

Identificación de Objetivos: Establecer objetivos claros para la migración, como reducción de costos, mejora en el rendimiento y escalabilidad.

B. Diseño del Plan de Migración

Selección de la Plataforma en la Nube: Evaluar y seleccionar la plataforma en la nube que mejor se adapte a las necesidades, como AWS, Azure o Google Cloud.

Diseño de la Arquitectura en la Nube: Diseñar la arquitectura en la nube que soporte los jobs, incluyendo la elección de servicios como bases de datos, almacenamiento y procesamiento (por ejemplo, AWS Lambda, Azure Data Factory).

C. Elaboración del Cronograma

Desarrollo de un Cronograma Detallado: Crear un cronograma con hitos clave, incluyendo la preparación, pruebas, migración y validación.

Asignación de Recursos: Definir los recursos necesarios, como el personal técnico, herramientas y servicios en la nube.

A. Preparación del Entorno

Configuración del Entorno en la Nube: Configurar los servicios en la nube según el diseño arquitectónico, incluyendo redes, almacenamiento y permisos.

Preparación de Datos: Realizar la limpieza y el formato de datos para asegurar que sean compatibles con la nueva infraestructura en la nube.

B. Migración de Jobs

Migración en Fases: Ejecutar la migración en fases para minimizar el riesgo. Comenzar con los jobs menos críticos para validar el proceso antes de migrar los más críticos.

Uso de Herramientas de Migración: Utilizar herramientas de migración automatizadas para transferir los jobs a la nube, como AWS Data Migration Service o Azure Data Box.

C. Pruebas y Validación

Pruebas Funcionales: Ejecutar pruebas para asegurar que los jobs se ejecuten correctamente en el entorno de la nube. Verificar que los resultados sean consistentes con los obtenidos en el entorno local.

Pruebas de Rendimiento: Evaluar el rendimiento de los jobs en la nube, asegurando que cumplan con los requisitos de tiempo de respuesta y procesamiento.

D. Optimización Post-Migración

Monitoreo y Ajuste: Implementar herramientas de monitoreo en la nube para supervisar el rendimiento de los jobs y realizar ajustes según sea necesario. Utilizar servicios de monitoreo como Amazon CloudWatch o Azure Monitor.

Optimización de Costos: Revisar los costos asociados y ajustar la configuración para optimizar el uso de recursos y minimizar costos. Utilizar técnicas de optimización como el escalado automático o la selección de instancias de menor costo.

Migración de un Job de Análisis de Datos

Planificación: La empresa decide migrar un job de análisis de datos que procesa grandes volúmenes de datos transaccionales a AWS.

Preparación: Configura Amazon Redshift para el procesamiento de datos y S3 para el almacenamiento de datos.

Migración: Utiliza AWS Data Migration Service para transferir los datos y configurar el job en AWS Glue para el procesamiento.

Pruebas: Ejecuta el job en AWS Glue, compara los resultados con los del entorno local y ajusta la configuración según sea necesario.

Optimización: Implementa monitoreo con Amazon CloudWatch y ajusta el uso de instancias para reducir costos, asegurando un equilibrio entre rendimiento y gasto.

3. **Mejorar la gestión y monitoreo de los jobs en la nube**: Establecer mecanismos para el monitoreo y la gestión continua de los jobs migrados, con el objetivo de asegurar su rendimiento y disponibilidad óptimos en el entorno de la nube.

A. Implementación de Herramientas de Monitoreo y Alerta

Amazon CloudWatch: Configurar Amazon CloudWatch para supervisar los recursos y servicios en la nube. Esto incluye la creación de métricas personalizadas para los jobs, tales como el tiempo de ejecución, uso de CPU, y latencia.

Ejemplo: Crear una métrica personalizada en CloudWatch que rastree el tiempo de ejecución de los jobs de análisis de datos. Configurar alarmas para notificar si el tiempo de ejecución excede el umbral esperado.

AWS X-Ray: Implementar AWS X-Ray para el rastreo y análisis de las solicitudes a través de servicios. Esto ayudará a identificar cuellos de botella y problemas en el procesamiento de los jobs.

Ejemplo: Configurar X-Ray para rastrear la ejecución de jobs de ETL (Extract, Transform, Load) y visualizar el flujo de datos a través de los servicios.

B. Configuración de Paneles de Control

Dashboards en CloudWatch: Crear dashboards personalizados en CloudWatch para monitorear el estado de los jobs en tiempo real. Incluir gráficos para visualización de métricas clave como el uso de recursos, tiempos de respuesta y tasas de error.

Ejemplo: Configurar un dashboard que muestre el estado de ejecución de los jobs, el uso de recursos de cada instancia, y cualquier alerta activa.

2. Gestión Efectiva de Jobs

A. Automatización del Escalado

Auto Scaling: Configurar políticas de autoescalado en AWS para ajustar automáticamente la capacidad de los recursos en función de la carga de trabajo. Esto asegura que haya suficiente capacidad para manejar picos en la demanda de procesamiento.

Ejemplo: Implementar autoescalado en Amazon EC2 para los servidores que ejecutan jobs de procesamiento de datos. Ajustar el número de instancias en función de la carga observada a través de las métricas de CloudWatch.

B. Programación y Orquestación de Jobs

AWS Step Functions: Utilizar AWS Step Functions para coordinar y orquestar la ejecución de jobs complejos. Crear flujos de trabajo que definan el orden de ejecución y las dependencias entre jobs.

Ejemplo: Crear un flujo de trabajo con Step Functions que coordine la ejecución de jobs de limpieza de datos, transformación y carga, asegurando que cada etapa comience solo después de que la anterior haya finalizado correctamente.

C. Gestión de Errores y Recuperación

Logs y Alarmas: Configurar logging detallado en Amazon CloudWatch Logs para capturar eventos y errores durante la ejecución de los jobs. Establecer alarmas para detectar errores críticos o fallos.

Ejemplo: Configurar alarmas en CloudWatch para notificar al equipo si se detectan errores específicos en los logs, como fallos en la conexión a la base de datos o problemas de ejecución.

3. Optimización Continua

A. Revisión Periódica del Rendimiento

Análisis de Rendimiento: Realizar análisis regulares del rendimiento de los jobs utilizando los datos recopilados por CloudWatch y X-Ray. Identificar áreas de mejora y ajustar los recursos o la configuración según sea necesario.

Ejemplo: Revisar los informes mensuales de CloudWatch para identificar tendencias en el tiempo de ejecución de los jobs y ajustar la configuración de recursos para mejorar el rendimiento.

B. Ajuste de Recursos y Costos

Optimización de Recursos: Revisar el uso de recursos y ajustar la asignación para equilibrar costo y rendimiento. Considerar el uso de instancias reservadas o savings plans para reducir costos a largo plazo.

Ejemplo: Evaluar el uso de instancias EC2 y cambiar a instancias reservadas para los jobs que tienen un uso constante, reduciendo el costo general.

4. Ejemplo Práctico

Optimización de un Job de Procesamiento de Datos

Configuración Inicial: La empresa configura Amazon CloudWatch para monitorear el uso de CPU y la duración de los jobs de procesamiento de datos.

Implementación de Auto Scaling: Se configura el autoescalado para las instancias EC2 que ejecutan los jobs, ajustando el número de instancias según la carga de trabajo.

Creación de Dashboards: Se crea un dashboard en CloudWatch para visualizar el estado y rendimiento de los jobs en tiempo real.

Automatización del Flujo de Trabajo: Se implementa AWS Step Functions para coordinar la ejecución de jobs de limpieza, transformación y carga de datos.

Revisión y Optimización: Se realiza un análisis mensual de rendimiento y se ajustan los recursos y políticas de escalado para mejorar la eficiencia y reducir costos.

4. **Capacitar al personal en las nuevas prácticas y herramientas**: Proporcionar formación y recursos al equipo técnico de Bluetab para asegurar una adecuada comprensión y aplicación del marco estandarizado y las nuevas herramientas asociadas con la migración a la nube.
5. **Evaluar los beneficios y resultados de la estandarización**: Medir el impacto de la estandarización en términos de eficiencia operativa, reducción de errores, y mejora en la calidad del servicio, utilizando métricas y feedback para ajustar y perfeccionar el proceso

Antecedentes

Según [1]. La empresa Derco Perú enfrenta actualmente desafíos para evaluar la calidad del servicio de atención y determinar si sus clientes confían en el servicio de postventa. Para abordar esto, es necesario definir una serie de indicadores de negocio que permitan medir dichas necesidades y, en consecuencia, implementar estrategias para mejorar la calidad de atención y aumentar la retención de clientes en postventa. Nuestro equipo de Inteligencia Comercial ha propuesto recopilar y automatizar la información generada por el área de postventa de todas las sucursales, incluyendo concesionarias, en un repositorio histórico, utilizando la tecnología en la nube proporcionada por la casa matriz de Derco. Este trabajo se realizó en coordinación con otras áreas para integrar información relevante, como marca, retenciones y ventas, a la cual se aplicaron diversas transformaciones para obtener datos claros, confiables y de fácil acceso. Como resultado, desarrollamos una serie de cuadros de mando que apoyan la toma de decisiones, brindando al área de postventa y marketing una visión amplia para aplicar estrategias de negocio. La planificación del proyecto estuvo a cargo del área de Inteligencia Comercial y se utilizó la tecnología que maneja actualmente la casa matriz en Chile. Este proyecto representa un gran beneficio para el área de postventa.

Según [2]. En el marco de la actual revolución digital, las instituciones financieras se ven obligadas a adaptarse a las nuevas exigencias de eficiencia y competitividad. El Banco Central de los Andes ha confiado en su sistema mainframe durante años, pero las circunstancias actuales presentan desafíos financieros y tecnológicos considerables. El costo operativo y de mantenimiento del mainframe, junto con el gasto en recursos MIPS (millones de instrucciones por segundo), se han vuelto insostenibles. Un área crítica que requiere atención inmediata es la gestión de soluciones de pago. Con el crecimiento de la base de clientes del banco, las transacciones de grandes volúmenes de datos relacionados con promesas de pago han hecho que el uso del mainframe sea prohibitivo en términos de costos. Sin embargo, la nube ofrece una solución escalable y rentable para abordar este problema. Este proyecto de suficiencia se centra

en una migración estratégica: trasladar a la nube la parte del mainframe que consume una cantidad significativa de MIPS. Se evaluará cómo esta transformación tecnológica no solo reducirá los costos operativos, sino que también posicionará al Banco Central de los Andes de manera óptima para satisfacer las crecientes demandas de sus clientes en un entorno financiero cada vez más digital. La migración se ha completado con éxito, logrando una mejora notable en la eficiencia operativa, reduciendo el tiempo de procesamiento de transacciones a un máximo de 17 minutos y disminuyendo los costos en un 45%. Esta transformación ha fortalecido estratégicamente nuestra posición en la competitiva industria bancaria peruana.

Según [3]**.** El proyecto se llevó a cabo en la empresa Daily Technology durante el año 2022. Esta empresa, que comenzó sus operaciones en 2013, se dedica a ofrecer soluciones tecnológicas al sector gubernamental. A lo largo del tiempo, experimentó un crecimiento exponencial tanto en ventas como en personal. Sin embargo, este crecimiento trajo consigo una serie de problemas en sus sistemas tecnológicos, que se agravaron con la llegada de la pandemia. Los servicios tecnológicos, así como la infraestructura y los datos de la empresa, se gestionaban bajo un modelo on-premise, lo que provocaba caídas frecuentes del sistema, corrupción de archivos, falta de espacio disponible y dificultades en el manejo de la información almacenada. La situación empeoró durante la pandemia, cuando los usuarios que trabajaban de forma remota enfrentaban problemas de acceso rápido y seguro a los recursos. Estos inconvenientes generaban reprocesos, demoras y pérdidas, afectando la experiencia de los usuarios finales. Ante esta situación, se propuso migrar todos los servicios tecnológicos a la nube e implementar una solución de respaldo en la nube. La solución incluyó la creación de una infraestructura de servicios en la nube y la migración completa de la información a este entorno. Además, se implementó un servicio de respaldo para asegurar la integridad y continuidad de los datos. Como resultado, la empresa pudo aprovechar los beneficios de las herramientas colaborativas, mejorar la productividad, reducir los tiempos en los procesos y minimizar las caídas del servicio, lo que incrementó la satisfacción de los usuarios.

Según [4]**.** Este análisis tiene como objetivo proponer la migración a Cloud Computing para mejorar la gestión y la eficiencia en costos de la inmobiliaria LGP. La infraestructura actual de la empresa es inadecuada para ofrecer un servicio de calidad, ya que cuenta con dos servidores físicos obsoletos que alojan cinco servidores virtuales, lo que dificulta la continuidad operativa y presenta problemas en los sistemas de protección de la información. Estos problemas ocasionan interrupciones frecuentes en el servicio debido al sobrecalentamiento de los servidores alojados en gabinetes, además de trabajar constantemente al límite de capacidad por la falta de espacio en las memorias de almacenamiento. Por lo tanto, se diseñó una estrategia para la migración a Cloud Computing, que incluye una serie de pasos para trasladar las máquinas virtuales a la nube. Los resultados del estudio indican que, tras la renovación de la infraestructura tecnológica y la

migración a la nube de cuatro servidores, se mejora la gestión de recursos y la accesibilidad. En conclusión, la propuesta de migración a Cloud Computing mejora la gestión y la eficiencia en los costos de la inmobiliaria LGP, incrementando la satisfacción de los trabajadores y presentando un VAN de $24,629.96 y una TIR de 635.47%.

Según [5] En la actualidad, la organización religiosa opera con una aplicación de escritorio basada en una arquitectura cliente-servidor, que es utilizada por los miembros para llevar a cabo sus tareas. Sin embargo, la organización no dispone de un servidor propio, lo que significa que la disponibilidad de la aplicación depende de que el equipo de la presidencia esté encendido, ya que este actúa como servidor. Uno de los procesos centrales del negocio, identificado como problemático, es el retraso en el proceso de cobranzas en el área de tesorería, debido a la disponibilidad limitada de la aplicación. Como se mencionó, el equipo de la presidencia debe estar encendido para que la aplicación pueda ser utilizada. Además, la organización no exige un estricto cumplimiento de horarios, dado que los miembros trabajan ad-honorem, lo que provoca que el área de tesorería acuda a la organización en horarios variados, según la disponibilidad de la aplicación. Esta situación genera que los miembros de la organización no paguen sus deudas a tiempo o lo hagan después del plazo establecido, lo que reduce los ingresos en el período actual y dificulta la generación de los reportes solicitados por la presidencia en el tiempo requerido. Por lo tanto, en este trabajo, proponemos desarrollar una aplicación web y móvil en la nube para mejorar el proceso de cobranzas, con el objetivo de que el área de tesorería no dependa de un equipo local para la disponibilidad de la aplicación, permitiendo que realicen sus labores sin obstáculos.

Capítulo I: INFORMACION DE LA INSTITUCION DONDE SE DESARROLLO LA EXPERIENCIA.

Descripción de la organización

Bluetab es una empresa internacional de tecnología que ofrece soluciones basadas en software y servicios. Llevan más de una década invirtiendo en investigación continua para ofrecer soluciones innovadoras a sus clientes. Tienen presencia y capacidad para operar en Europa desde las oficinas de Madrid, Alicante y Londres, y en América desde las oficinas de Bogotá, Ciudad de México y Lima. Actualmente en la sede de Lima se tiene 115 empleados. Su objetivo es maximizar el valor de los datos y optimizar la inversión en infraestructuras y plataformas tecnológicas, asegurando una gestión eficiente y estratégica de los recursos disponibles.

Bluetab tiene una estructura plana. Su gente es accesible, y se puede hablar de todo y con todos. Se da mucha importancia al conocimiento y están obsesionados con la mejora continua, ya que nunca nada es suficiente. No son generalistas, sino especialistas en datos. Cuidan a las personas y las tratan de forma individual, con la creencia de que actitud + aptitud = crecimiento profesional. Dan visibilidad al talento, permitiendo que el técnico se desarrolle sin separarse de la tecnología, sin necesidad de gestionar o vender para desarrollarse profesionalmente. Cuentan con las certificaciones ISO 9001 Gestión de Calidad, ISO 27001 Seguridad de la Información, y son reconocidos como Great Place To Work.

Su especialización en Data & Cloud y el trabajo bien hecho los han llevado a convertirse, desde el año 2021, en una IBM Company, reforzando así su enfoque. Nacieron en 2006 con el objetivo de revolucionar el sector y ahora trabajan para convertirse en líderes de Data & AI en todas las regiones en las que operan. En Bluetab, entienden la gestión de los datos como un iceberg: lo que se ve, los datos que aportan valor a tu negocio, son solo la punta del iceberg. Lo que hay detrás y no se ve es clave.

La cultura de Bluetab se enfoca en las personas aunque también tienen una profunda cultura de ingeniería y trabajo de calidad. Cuidan día a día a un equipo especializado y motivado de más de 1000 bluetabers que participan en proyectos para EMEA y América. Su estructura es abierta y tienen un ADN inconformista: valoran el conocimiento, el talento y el trabajo bien hecho, pero sobre todo valoran la actitud positiva, las ganas de hacer y los retos complejos. Trabajan para favorecer la autonomía y el desarrollo de su plantilla bajo la máxima de la Conciliación Laboral-Personal con beneficios que ayudan a trabajar con pasión y estabilidad en un entorno de trabajo diverso e igualitario, invirtiendo en acciones socialmente responsables. Todo esto los ha llevado a ser Great Place to Work & Best Workplaces en España y en México, en este último además contando con el reconocimiento Best Workplaces for Women.

En resumen, Bluetab es una organización dedicada a maximizar el valor de los datos y a optimizar la inversión en infraestructuras tecnológicas, mientras fomenta una cultura corporativa accesible, centrada en el conocimiento y la mejora continua. Se especializan en Data & Cloud, lo que les ha permitido convertirse en una IBM Company y aspirar a liderar el sector de Data & AI. Con una estructura abierta y un fuerte enfoque en la conciliación laboral-personal, se enorgullecen de ser un lugar donde el talento y el trabajo bien hecho son reconocidos y valorados.

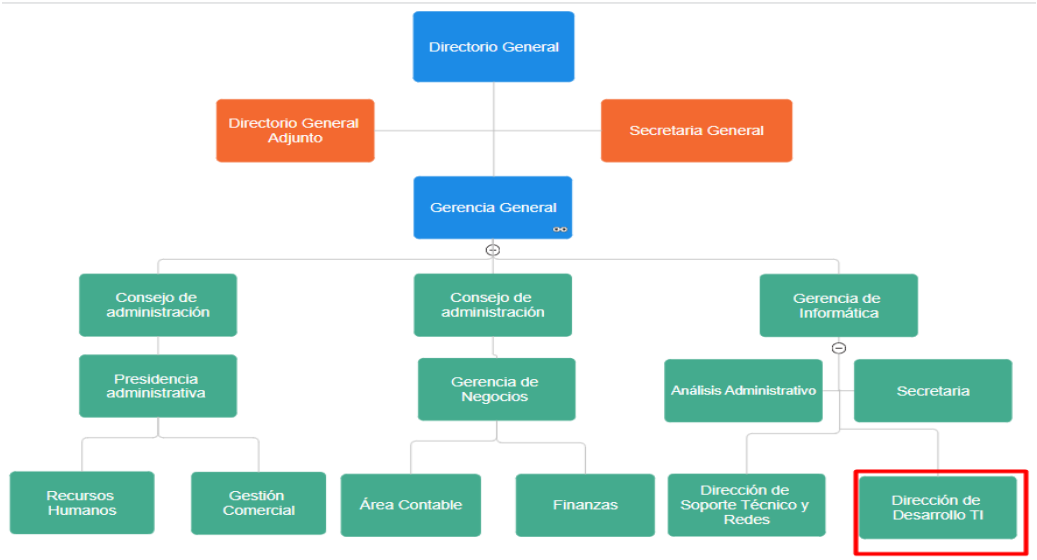


Fig.1: Organigrama

Se muestra el organigrama donde indica la dirección donde se realizan los servicios integrales de tecnología. Bluetab como corporación cuenta con otras empresas con las que desarrolla sus actividades de servicios integrales de tecnología: La empresa Bluetab Perú para las cuales el autor ha desarrollado la prestación de servicios de tecnología informática.

Capítulo II: TRAYECTORIA PROFESIONAL

Durante mi trayectoria profesional he desempeñado diversos roles según lo encomendado por la empresa NTT Data (En el área de Calidad de software) y Bluetab como Technician (En el área de Big Data).

En el cargo de Analista QA que se tenía como finalidad principal asegurar la funcionalidad manual y/o automatizada de las diversas aplicaciones de empresa de Telefónica a la cual se prestaban los diversos servicios, entre ellas destacan las pruebas de regresión, pruebas Sanity que se realizaban para diversas aplicaciones como Amdocs, DITO, App Autoactivado,Web Maximo, MCSS y Anzio. Adicionalmente también se realizaban pruebas servicios utilizando plataformas como SOAP y Postman para las diversas consultas mediante los diversos métodos HTTP. Se realizaba la gestión de envío de reportes de avance con un dashboard de las jornadas consumidas e indicadores claves para las personas lideres del proyecto. El principal proyecto que destaco en ese lugar fue la automatización de la web de Dito con herramientas como Intelli Idea, Selenium, Cucumber, jUnit, y Java, utilizando herramientas para la integración Continua como el uso de Git, Bibbucket y Jira para la publicación de defectos y errores.

En el cargo de Tehnician Big Data : En el área de Data Engineer realizando funciones de desarrollo de soluciones que con lleva a la estandarización y compactación de desarrollos de Jobs de tipo Kirby, Hammurabi y pyspark con sus respectivos esquemas de acuerdos a los lineamientos del área de calidad, Análisis y Desarrollo de pipelines ETL utilizando procesamiento distribuido en Spark usando PySpark y Scala (batch).para el modelo Aws según los estándares del área Data Quality, Optimización de procesos utilizando procesamiento distribuido HDFS y compactación On premise, Uso de la herramienta de gestión Jira, para el seguimiento de asignaciones, creaciones de tickets de desarrollo para su posterior productivización, Despliegue soluciones en producción haciendo uso de herramientas de integración y despliegue continuo (Jenkis, git, Bitbucket, Dataproc y Argos).Por último cabe destacar que como contribución realice diversas automatizaciones en lenguaje pyspark, que ayudaron a acelerar el proyecto de la estandarización, permitiéndome tener reportes de análisis automatizados y que la gestión del proceso con el área de arquitectura se me sea mas sencillo.

Algunos de los conocimientos adquiridos como parte de la aplicación del trabajo se destaca:

1. Aplicación de buenas practicas para el desarrollo y estandarización de Jobs.
2. Aplicaciones de metodología ágil como SCRUM.
3. Aplicaciones de metodología tradicionales como CASCADA.

4. Desarrollo de automatizaciones pyspark como punto de mejora para el avance ágil del proyecto.
5. Conocimiento de Herramientas propias de la organización.
6. Despliegue de desarrollos ETL a el ambiente desarrollo y producción.
7. Compactación de ficheros, como mejora para las consultas de información.
8. Desarrollo de reportes Power Bi y Looker Studio para el reporte de indicadores de avance a las personas lideres del proyecto.
9. Aplicaciones de buenas prácticas de automatización y construcción del arquetipo para el flujo funcional web.
10. Asumir responsabilidades cruciales en el proyecto que es de competencia profesional.

Tabla I: Resumen de experiencia Profesional

Empresa	Del	Al
Indra Perú	10/07/2024	Actualidad
Bluetab Perú	29/05/2023	03/07/2024
NTT DATA Perú	04/07/2022	31/05/2023

Capítulo III: APLICACIÓN PROFESIONAL

3.1. Situación problemática

Se enfrento un problema significativo relacionado con la **estandarización global de los jobs en la empresa**. Este problema abarcaba tres áreas críticas: la alineación de esquemas, el desarrollo y la compactación de ficheros en Hadoop (HDFS).

Desafío en la Gestión de Ficheros

La gestión ineficiente de los ficheros en HDFS se basaba en una regla estricta: no debía haber ficheros que pesaran menos de 128 MB si existía una cantidad significativa de ellos. Esto generaba la necesidad de compactar los ficheros de manera efectiva para optimizar tanto el almacenamiento como el procesamiento de datos. La falta de una estrategia eficiente para la compactación de ficheros llevó a una subutilización del espacio de almacenamiento y a problemas de rendimiento en el procesamiento de grandes volúmenes de datos.

Desalineación de Esquemas

El equipo identificó aproximadamente 700 objetos desalineados que habían sido depreciados antes del 2023. Estos objetos presentaban diferencias en los esquemas, información de partición y nivel de seguridad entre los ambientes de desarrollo y producción. La magnitud del problema requería una solución sistemática y bien planificada, ya que era esencial alinear estos objetos para migrar los esquemas y desarrollos de jobs a la nube. La desalineación podría causar errores graves durante la migración a AWS, como incompatibilidades de datos, fallos en la ejecución de jobs y problemas de seguridad y cumplimiento.

Actualización de Desarrollos

Además de la desalineación de esquemas, los desarrollos de los jobs estaban frecuentemente desactualizados. Independientemente de si el job era Kirby o PySpark, estos debían alinearse con los nuevos lineamientos, lo cual implicaba rediseñar los flujos y realizar pruebas exhaustivas que debían ser validadas por el área de calidad. Esta falta de estandarización no solo afectaba la calidad del desarrollo, sino que también podía provocar retrasos y errores en la migración y ejecución de los jobs.

3.2. Proyecto de solución

En la estandarización de Jobs para la migración a la nube se han utilizado diversas herramientas. Herramientas para el desarrollo, herramientas para la estandarización, herramientas de gestión y herramientas de visualización de

indicadores de KPI, las mismas que detallamos algunos conocimientos teóricos relacionados con ellas.

En primer lugar, para la gestión del proyecto se utilizó la metologia SCRUM. Scrum es un marco que permite a los equipos abordar problemas complejos, al mismo tiempo que entregan productos de manera eficiente y con el máximo valor posible. Esta metodología se basa en iteraciones cortas y ciclos de retroalimentación continua, facilitando la adaptación y mejora constante. A través de la implementación de Scrum, el equipo pudo mantener una comunicación abierta y constante, priorizar tareas de manera efectiva y adaptarse rápidamente a los cambios y desafíos que surgían durante el proyecto.

Se utilizó Jira para la gestión de tareas y seguimiento de historias que creábamos por objetos. **Jira** es una herramienta de software desarrollada por Atlassian que se utiliza para la gestión de proyectos y el seguimiento de problemas. Es especialmente popular en equipos de desarrollo de software que utilizan metodologías ágiles, como Scrum y Kanban. Jira permite a los equipos planificar, rastrear y gestionar tareas y proyectos de manera eficiente. (Atlassian Community) (Atlassian) (Atlassian).

Para la subida de repositorio se utilizo Bibbucket que era un repositorio de todos los desarrollos de diferentes tipos . Bitbucket es una plataforma de gestión de código fuente que permite a los equipos colaborar en el desarrollo de software mediante la integración con Git. Ofrece funciones avanzadas de control de versiones, revisiones de código, y CI/CD (integración y entrega continua). Bitbucket permite a los usuarios configurar permisos granulares a nivel de proyecto, repositorio, rama o entorno, lo que facilita la gestión segura del código y la implementación de políticas de calidad.

Para el versionamiento de código se utilizo la herramienta de GIT. El uso de Git implica varias etapas, incluyendo la edición de archivos en el directorio de trabajo, la adición de cambios al área de preparación (staging area) y la realización de commits para guardar los cambios en el historial del repositorio. Las ramas y fusiones (merges) son componentes esenciales en los flujos de trabajo de Git, permitiendo a los equipos de desarrollo trabajar en características, correcciones y versiones de manera aislada y luego integrarlas en el código principal. TechTarget y FreeCodeCamp.

Dataproc es una plataforma de procesamiento de datos en la nube que facilita la ejecución de aplicaciones y herramientas de big data, como Hadoop y Spark. En el contexto de estandarización y alineación de esquemas, Dataproc fue utilizado como un entorno de pruebas que permitía realizar operaciones de compactación, ingestas y procesamientos de datos. Esta herramienta proporcionaba la facilidad de establecer parámetros de entrada que eran programados a través del desarrollo, lo que permitía configurar y ejecutar

pruebas de manera eficiente. La versatilidad de Dataproc ayudaba a asegurar que los datos se procesaran y almacenaran conforme a las especificaciones requeridas, optimizando el manejo y el rendimiento de los datos en la infraestructura de big data.

Argos es una herramienta utilizada para monitorear la ejecución de trabajos, proporcionando información sobre el estado de las ejecuciones, ya sean exitosas o fallidas. Permite a los usuarios acceder a los registros de ejecución para diagnosticar problemas y entender las causas de fallos. La plataforma ofrece la capacidad de visualizar datos, exportar informes y configurar alertas automáticas, lo que facilita la supervisión y la gestión de diversas tareas y procesos operativos.Además de su funcionalidad principal de monitoreo, Argos también se utiliza para reportes de inteligencia empresarial, permitiendo a los usuarios extraer y analizar datos de sistemas subyacentes, como bases de datos. La herramienta soporta múltiples formatos de salida y opciones de visualización, haciendo posible el acceso a la información crítica de manera eficiente y precisa (Argos System) (Emerson College) (Argos System).

Para las consultas HDFS y corroborar si la estandarización fue realizada de manera correcta mediante la ingesta o procesamiento se utilizó Hadoop siendo más explicita el componente core HDFS(Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido diseñado para ejecutarse en hardware de commodity. Es una parte fundamental del ecosistema Hadoop, proporcionando almacenamiento fiable y de alta disponibilidad para grandes volúmenes de datos.HDFS permite almacenar grandes archivos distribuyéndolos en múltiples nodos dentro de un clúster. Esto no solo mejora la capacidad de almacenamiento, sino que también proporciona redundancia y tolerancia a fallos, asegurando que los datos estén disponibles incluso si algunos nodos fallan. En el contexto de ETL (Extract, Transform, Load), HDFS se utiliza para almacenar datos de origen y destino. Los procesos de ingesta y procesamiento de datos (ETL) pueden realizarse en HDFS, permitiendo la ejecución de consultas para verificar si el proceso ha sido exitoso. Esto incluye la capacidad de comprobar si los datos han sido correctamente extraídos del origen, transformados según las reglas de negocio, y finalmente cargados en el destino deseado. HDFS se integra con otras herramientas del ecosistema Hadoop como MapReduce, Hive, y Spark, proporcionando una infraestructura robusta para grandes volúmenes de datos y complejos procesos de análisis (Argos System) (Emerson College).

Para la visualización dashboard de indicadores de avances se utilizo POWER BI. Power BI es una herramienta de inteligencia empresarial desarrollada por Microsoft que permite transformar datos en información visual significativa a través de gráficos interactivos,

informes y paneles personalizados. Los usuarios pueden conectar y modelar datos provenientes de diversas fuentes, como hojas de Excel, bases de datos locales y servicios de almacenamiento en la nube. La plataforma facilita el análisis de datos, la creación de visualizaciones y la generación de informes que ayudan en la toma de decisiones informadas y en el monitoreo de indicadores clave de rendimiento (KPIs) en tiempo real (Microsoft Learn) (Coursera).

3.2.1. Solución y objetivos

Objetivo Principal

Optimizar la gestión y el procesamiento de datos en entornos Hadoop mediante la estandarización de jobs y la compactación eficiente de ficheros, utilizando herramientas avanzadas y metodologías ágiles para mejorar la calidad y rendimiento de los datos en la nube.

Objetivos Secundarios

- Estandarizar el desarrollo de jobs y esquemas en entornos de Big Data, asegurando la alineación de los mismos con los estándares de calidad y mejores prácticas establecidas, para garantizar la consistencia y la integridad de los datos.
- Implementar un proceso sistemático de compactación de ficheros en Hadoop (HDFS), utilizando parámetros de coalesce y repartition, para optimizar el almacenamiento y el rendimiento del procesamiento de datos.
- Aplicar la metodología ágil Scrum para gestionar el proyecto, facilitando la colaboración continua, la adaptación a cambios y la entrega eficiente de resultados, mientras se integran herramientas de control de versiones y despliegue continuo para una gestión efectiva de los cambios.

Para alcanzar los objetivos del proyecto, es esencial contar con un equipo capacitado, herramientas adecuadas, una metodología ágil como Scrum para una gestión efectiva, y un proceso de trabajo estructurado que permita la planificación, ejecución, validación y documentación continua. Estos elementos aseguran que la estandarización de jobs y la optimización de ficheros se realicen de manera eficiente, cumpliendo con los estándares de calidad y mejorando el rendimiento del procesamiento de datos.

3.2.2 Planificación

Para realizar la planificación del proyecto, de estandarización globales de Job para la migración a la nube. El grupo del proyecto cuenta con un proceso general, el cual es conocimiento de todos los miembros para que se siga con lo establecido en el flujo general.

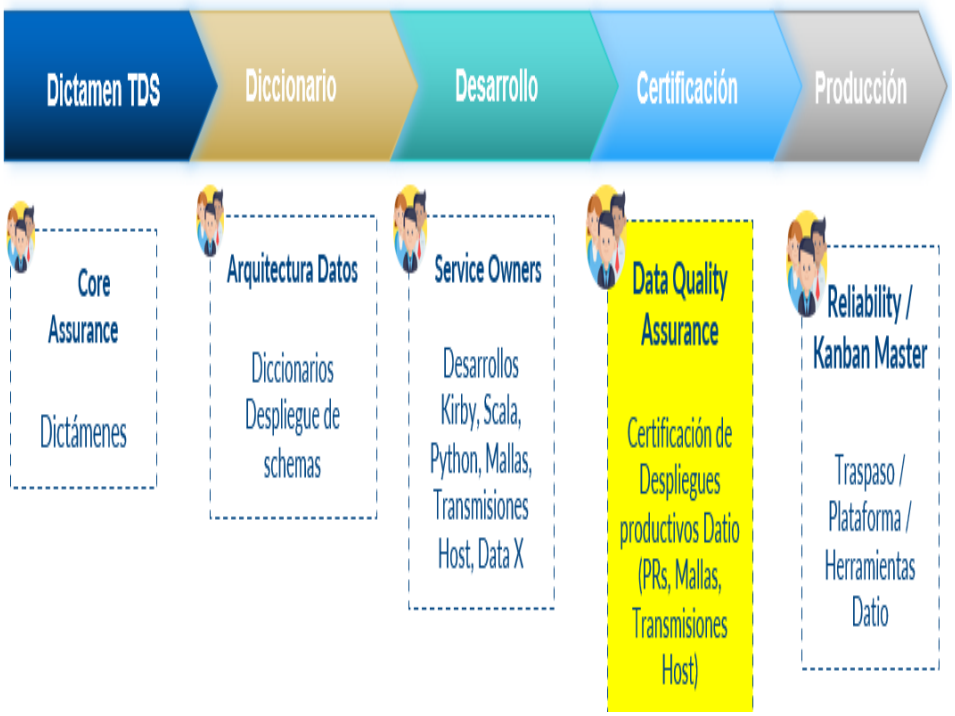


Fig.2: Flujo general de la creación de un desarrollo

Para la estandarización globales de Job para la migración a la nube cuenta con un flujo general, el mismo que se presenta en la Fig 2. Para conseguir el objetivo deseado de estandarizar 700 objetos aproximadamente del proyecto estandarización de Jobs globales para la migración a la nube, se definieron y calendarizaron las actividades que se iban a desarrollar, las que se presentan en la Figura 3.

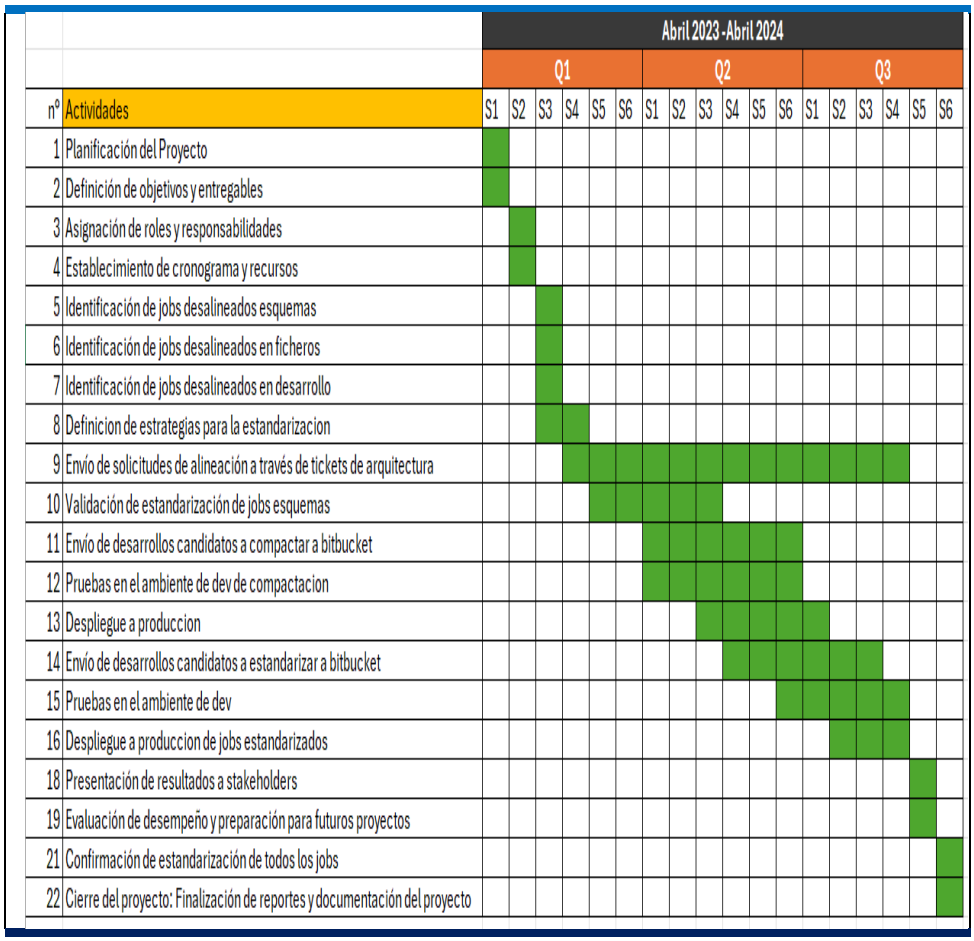


Fig.3: Plan de trabajo del proyecto

El plan de trabajo del proyecto de módulo, igualmente se realizó la calendarización exacta de fechas para el desarrollo de los Sprint, ya que se aplicó para ello algunos conceptos de la metodología ágil SCRUM por los beneficios que aporta en el desarrollo del proyecto, la calendarización se presenta en la Tabla III.

Tabla II: Calendarización de las tareas del proyecto

Actividad	Inicio	Fin	Duración
Proyecto de estandarización **de jobs globales para la** migración **a AWS**	**3/04/2023**	**16/04/2024**	**272**
Análisis **y seguimiento de Jobs**	**3/04/2023**	**16/04/2024**	**272**
Seguimiento y control de los jobs ya estandarizados	3/04/2023	16/04/2024	272
Seguimiento de arquitectos de datos	3/04/2023	16/04/2024	272
Ejecución **del proyecto**	**3/04/2023**	**16/04/2024**	**272**
Trimestre 1	3/04/2023	4/08/2023	90
Trimestre 2	7/08/2023	8/12/2023	90
Trimestre 3	11/12/2023	12/04/2024	90
Soporte y cierre del proyecto	**15/04/2024**	**16/04/2024**	**2**

Para la ejecución del proyecto se requirió contar con los profesionales especialistas que se presentan en la Tabla II.

Tabla III: Capital Humano requerido

Profesional Especialista	Cantidad
Data Engineers	04
Data architect	03
Analista de Calidad de Datos	02
SM	01
PO	01

Esta figura representa el flujo para que los Data Engineer puedan seguir los pasos de para que sus desarrollos cumplir con los estándares de calidad.

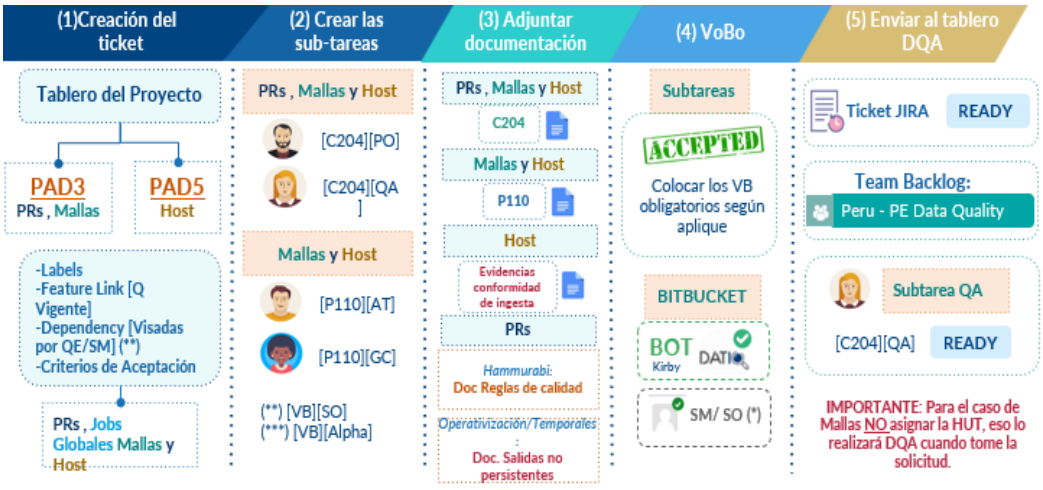


Fig.4: Flujo de datos

3.3. Implementación del proyecto

La problemática relacionada con los Jobs globales llevó al equipo de Datos a definir un objetivo crucial para resolver el problema: estandarizar los Jobs globales con vistas a su migración a la nube. Este objetivo se centra en la estandarización de esquemas, desarrollos y la compactación de los ficheros asociados. La implementación de estas medidas permitirá una migración a la nube segura y sin riesgos.

3.3.1. Descripción del proyecto

El proyecto de estandarización de Jobs está orientado a garantizar que los procesos de datos cumplan con los estándares de calidad necesarios para una migración efectiva a AWS. Este proceso de estandarización se fundamenta en tres requisitos esenciales:

Alineación de Esquemas: Todos los Jobs deben estar en concordancia con el esquema definido en Artifactory Live y tener la versión adecuada. Esta alineación asegura la integridad y compatibilidad de los datos durante la migración.

Cumplimiento de Estándares de Calidad: Es fundamental que los desarrollos sigan los estándares de calidad establecidos. Esto incluye la actualización de parámetros y librerías, la correcta configuración de variables de entorno, el uso de rutas precisas en Artifactory, la eliminación de particiones innecesarias y la correcta configuración de flags booleanos según las directrices.

Gestión Eficiente de Ficheros: Los Jobs deben manejar una cantidad adecuada de ficheros, evitando la presencia de archivos con un tamaño inferior a 128 MB. En caso de identificar archivos que no cumplen con este criterio, se debe proceder con la compactación por bloques para optimizar el almacenamiento y asegurar la eficiencia en el procesamiento de datos.

El cumplimiento de estos tres requisitos garantiza que un Job esté completamente estandarizado y listo para ser migrado a la nube. Este enfoque minimiza los riesgos asociados con la migración, asegurando una transición fluida y efectiva a AWS.

3.3.2. Alcance del proyecto

Para la implementación del módulo fue necesario establecer su alcance, los que se detallan a continuación y que son requisitos que debe cumplir el sistema de control mostrados en la Tabla IV.

Tabla IV: Requisitos del sistema de control

Código	Requisito	Descripción	Importancia
REQ01	**Alineación de Esquemas**	Los esquemas de los Jobs deben coincidir con el esquema de Artifactory Live y tener la versión adecuada.	Asegura la compatibilidad de los datos y evita problemas de integración.
REQ02	**Actualización de Parámetros y Librerías**	Los parámetros y librerías deben estar actualizados según los estándares establecidos.	Mantiene la calidad y seguridad del código y asegura su compatibilidad.
REQ03	**Configuración de Variables de Entorno**	Las variables de entorno deben estar correctamente configuradas.	Garantiza la correcta ejecución de los Jobs en el entorno de destino.
REQ04	**Uso Adecuado de Rutas en Artifactory**	Las rutas de acceso deben ser precisas y sin partición innecesaria.	Asegura la correcta ubicación y accesibilidad de los recursos.
REQ05	**Configuración de Flags Booleanos**	Los flags booleanos deben estar configurados según las directrices establecidas.	Asegura que los Jobs se comporten de acuerdo a las especificaciones requeridas.
REQ06	**Compactación de Ficheros**	Si se detectan archivos menores a 128 MB, se debe proceder a su compactación por bloques.	Mantiene el tamaño de los ficheros dentro de un rango óptimo para la eficiencia.
REQ07	**Documentación del Proceso**	Se debe crear documentación detallada sobre los cambios realizados y procedimientos seguidos.	Facilita la comprensión y mantenimiento de los Jobs estandarizados.

REQ08	**Pruebas de Integración y Validación**	Realizar pruebas para verificar que los Jobs estandarizados funcionen correctamente en el ambiente On premise antes de ser migrados a la nube.	Asegura que la migración sea exitosa y los Jobs operen sin problemas.

3.3.3 Implementación del proyecto

La empresa cuenta actualmente con un inventario completo de todos los Jobs que pertenecen al stock, identificados como aquellos desarrollados antes del año 2023. Estos Jobs se caracterizan por estar desactualizados en términos de desarrollo, esquemas o compactación. Anteriormente, no existía un área de calidad definida, lo que permitía a los desarrolladores de datos trabajar según su criterio personal. Esto resultó en inconsistencias como nombres de variables arbitrarios, métodos nombrados en español y una falta de descripción precisa de los métodos.

object	Diagnostico PDCO	Responsable	Estado_Candelerizacion	Ticket	Val_Live
t_abc_tabla1	Trabajar	Rosa Isela	Live	DATAXY-23451	OK
t_xyz_tabla2	Trabajar	Rosa Isela	Live	DATAXY-23451	OK
t_sdf_tabla3	Trabajar	Rosa Isela	Live	DATAXY-23453	OK
t_lop_tabla4	Trabajar	Rosa Isela	Live	DATAXY-23453	OK
t_asw_tabla5	Trabajar	Rosa Isela	Live	DATAXY-23451	OK
t_tyr_tabla6	Trabajar	Rosa Isela	Live	DATAXY-23453	OK
t_pal_tabla7	Trabajar	Rosa Isela	Live	DATAXY-23453	OK

Fig.5: Inventario del stock

Los desarrollos se clasifican en dos tipos principales: **Desarrollo de Ingesta** y **Desarrollo de Procesamiento**, cada uno con sus respectivos esquemas y ficheros.

- **Desarrollo de Ingesta**: Estos trabajos se enfocan en mover datos desde la capa de staging a raw, y luego a master. Se utilizan herramientas como Kirby y Hammurabi para aplicar reglas de calidad específicas dependiendo de la capa en la que se encuentren los datos. Kirby, por ejemplo, se usa para ingestas con ciertas reglas de calidad.

Fig 6: Imagen de un archivo conf de ingesta

- **Desarrollo de Procesamiento**: Estos trabajos son más complejos y generalmente implican tomar datos de la capa master, unir uno o más conjuntos de datos maestros, y realizar transformaciones y cálculos de campos. Aquí se emplea PySpark para el procesamiento debido a su capacidad de manejar grandes volúmenes de datos y realizar operaciones complejas de transformación.

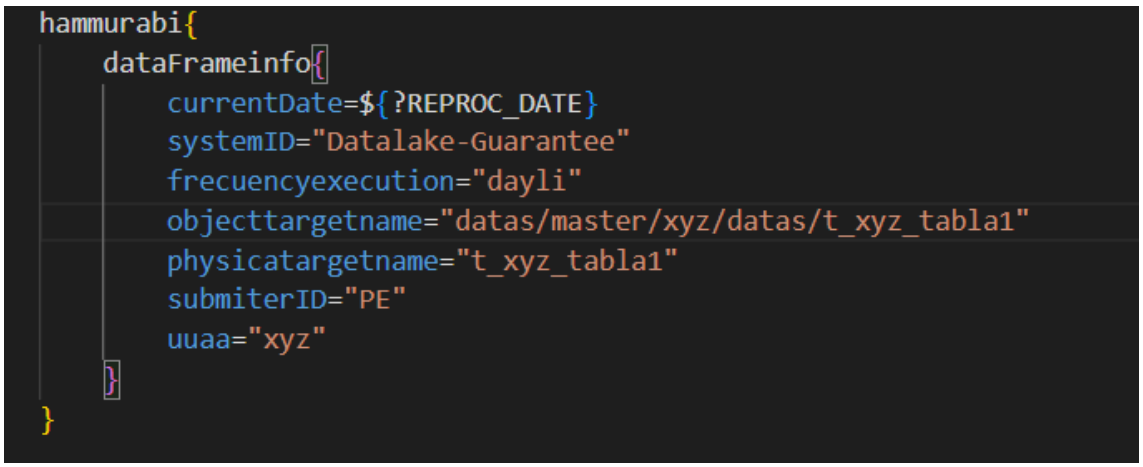


Fig 7: Imagen de un archivo conf de hammurabi

La empresa maneja un repositorio central de datos llamado **Datalake Datio**, una plataforma integral de datos analíticos y operativos que proporciona gobernanza, ingesta de datos, procesamiento, almacenamiento y consumo avanzado de datos para la plataforma Ether. El Datalake Datio se organiza en tres zonas HDFS diferentes:

Staging: Zona donde los sistemas heredados depositan los datos en la plataforma. Estos datos pueden contener nulos, datos redundantes, tablas no normalizadas y tipos de datos inconsistentes (formatos como txt, fixed, csv, xml, host, etc.).

Raw: Zona intermedia de datos sin procesar. Los datos aquí no pueden contener información confidencial sin estar encriptados y se almacenan en formato Avro.

Master: Área final donde los datos son almacenados en un formato listo para su uso por otras aplicaciones. Estos datos se comprimen con Snappy y se almacenan en formato Parquet HDFS.

Se utiliza la plataforma Artifactory como repositorio centralizado para almacenar y gestionar archivos de configuración y esquemas, manteniendo un historial completo de versiones. Artifactory dispone de dos entornos: producción y desarrollo. En este entorno, el área de arquitectura realiza modificaciones en los esquemas, mientras que los desarrolladores pueden ajustar los parámetros de los archivos de configuración en el entorno de desarrollo. Estos archivos de configuración son fundamentales para los desarrollos realizados con herramientas como Kirby y Hammurabi, y los esquemas son utilizados por PySpark para garantizar la coherencia y precisión de los datos procesados.

Para almacenar el diccionario de los objetos, se utiliza la plataforma DATUM, que contiene toda la información sobre los campos, tipos de campos y otros criterios de cada objeto. DATUM está asociada a Artifactory, y cada cambio desplegado se actualiza en el Artifactory de desarrollo. Esta plataforma permite ver el tipo de esquema de manera más detallada, así como la fecha de cambio de versión y la persona que realizó el cambio, lo cual es crucial en caso de desalineaciones.

Para las ejecuciones de trabajos propios, se utiliza Dataproc, un servicio que, junto con diversas herramientas, permite crear trabajos de procesamiento distribuido. Estas herramientas se emplean en tareas como adquisición de datos, preparación de datos, procesamiento de flujos, y aprendizaje automático utilizando Spark. Dataproc facilita operaciones con datos en Ether Data Lake, como copiar, mover, extraer, reemplazar y enviar datos.

Dataproc puede gestionar múltiples tipos de trabajos, principalmente basados en operaciones de Apache Spark, incluyendo Kirby, Spark (tanto Batch como Streaming), PySpark y HDFS.

3.3.4 Procesos de estandarización

Antes de iniciar cualquier flujo de estandarización de desarrollo se debía primero tener en cuenta el flujo, de que todo desarrollo debía ser registrado primeramente como solicitud de desarrollo al área de Data Quality.

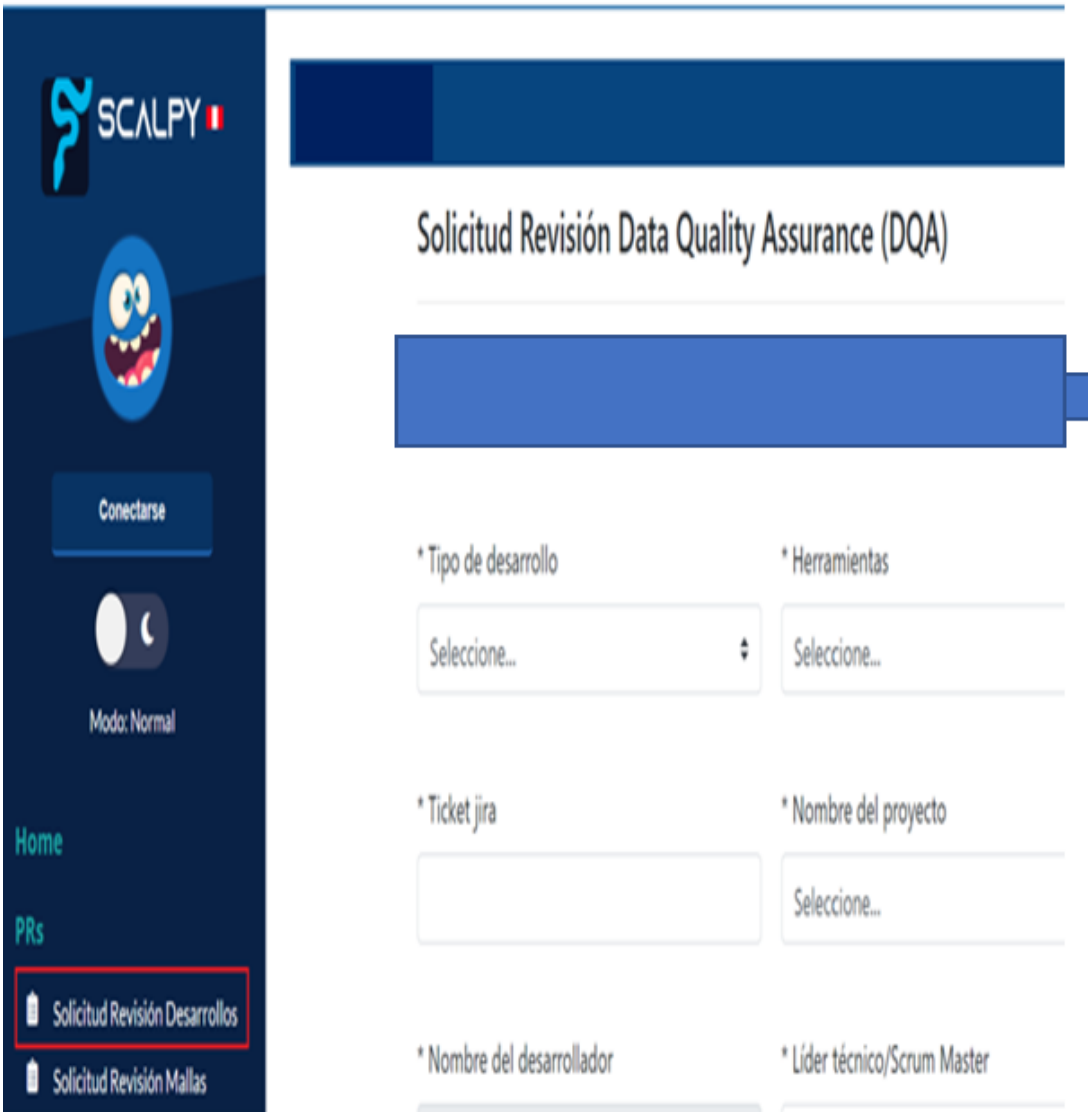
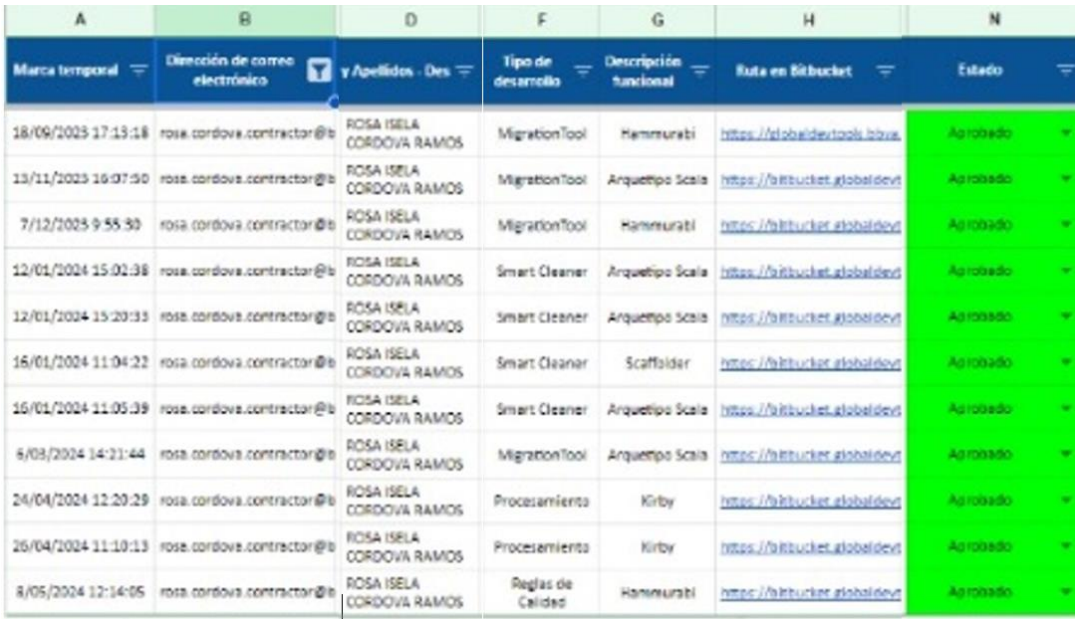


Fig.8: Tablero de Scalpy

Tablero de scalpy de como registrábamos la asignación de nuestros proyectos como se puede apreciar quedaban registrado nuestros nombres y el estado en la cual estaban, los estados eran in progress, Test, y aprobado que este finalizaba cuando el QE le daba el check a tu desarrollo y con ello era pasado a la rama productiva.



A	B	D	F	G	H	N
Marca temporal	Dirección de correo electrónico	y Apellidos - Des	Tipo de desarrollo	Descripción funcional	Ruta en Bitbucket	Estado
18/09/2023 17:13:18	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	MigrationTool	Hammurabi	https://globaldevtools.bbva.	Aprobado
13/11/2023 16:07:50	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	MigrationTool	Arquetipo Scala	https://bitbucket.globaldevt	Aprobado
7/12/2023 9:55:30	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	MigrationTool	Hammurabi	https://bitbucket.globaldevt	Aprobado
12/01/2024 15:02:38	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Smart Cleaner	Arquetipo Scala	https://bitbucket.globaldevt	Aprobado
12/01/2024 15:20:33	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Smart Cleaner	Arquetipo Scala	https://bitbucket.globaldevt	Aprobado
16/01/2024 11:04:22	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Smart Cleaner	Scaffolder	https://bitbucket.globaldevt	Aprobado
16/01/2024 11:05:39	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Smart Cleaner	Arquetipo Scala	https://bitbucket.globaldevt	Aprobado
6/03/2024 14:21:44	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	MigrationTool	Arquetipo Scala	https://bitbucket.globaldevt	Aprobado
24/04/2024 12:20:29	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Procesamiento	Kirby	https://bitbucket.globaldevt	Aprobado
26/04/2024 11:10:13	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Procesamiento	Kirby	https://bitbucket.globaldevt	Aprobado
8/05/2024 12:14:05	rosa.cordova.contractor@b	ROSA ISELA CORDOVA RAMOS	Reglas de Calidad	Hammurabi	https://bitbucket.globaldevt	Aprobado

Fig.9: Estado de ticket

Es más, teníamos que tener mu cuidado cuando revisaban nuestros tickets por que a veces QA podría ser muy riguroso y eso podría afectar nuestro indicares dentro de la empresa, por eso era muy importante conocer nuestros lineamientos para poder debatir con ellos. Cuando nuestro desarrollo era subido ellos de haber alguna observación podrían colocarte como comentario y tenias que corregirlo para que el código pueda pasar sin errores.

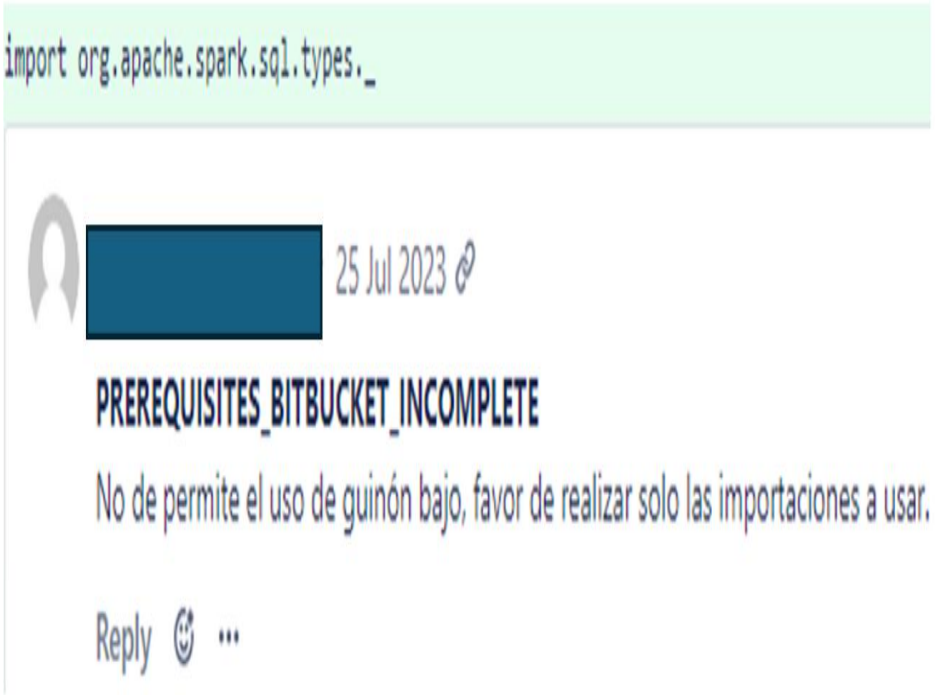


Fig.10: Comentario muy riguroso a un desarrollo.

Como se puede apreciar en la Fig.10 la manera de revisar era colocarle un tag adecuado a los errores que podría ver en el código, estos tags ya venían establecidos para que el QA lo utilice de acuerdo a cada casuística de esta manera al medir indicadores el PO podría ver los errores mas frecuentes del equipo, la cantidad de desarrollos que se suben, y así medir individualmente el desempeño. Por eso vuelvo a resaltar que era muy importante pasar tu desarrollo sin observaciones

Estandarización de esquemas de los Jobs

En el proyecto ADA, trabajé en la estandarización de esquemas entre Artifactory work, DATUM (diccionario de esquemas de todas las tablas del datalake del banco) y Artifactory live. Este esfuerzo se debió a que, en el pasado, las modificaciones en las tablas del ambiente work o desarrollo desalineaban su estructura respecto a live, afectando las pruebas y demorando la gestión de alineación.

Nuestro equipo identificó aproximadamente 750 objetos desalineados (objetos depreciados antes del año 2023). El proceso de alineación consistía en analizar las diferencias de esquemas, información de partición y nivel de seguridad, para

determinar qué se podía alinear y qué no. Inicialmente, nos enfocamos en diferencias simples como formato o tipo de campo, enviando las solicitudes de modificación a través de tickets de arquitectura, con una plantilla adjunta para indicar los cambios necesarios.

Valor Añadido: Automatización del Proceso

Para optimizar el proceso, propuse y desarrollé tres automatizaciones basadas en PySpark que mejoraron significativamente la eficiencia. A continuación, se detallan estos desarrollos:

Comparación Masiva de Esquemas entre Datum (Artifactory Work) y Artifactory Live:

- **Objetivo**: Realizar una comparación exhaustiva de los campos de todos los objetos de Datum con los esquemas del ambiente productivo en Artifactory Live.
- **Proceso**:

  Exportación de Datos: Inicialmente, se exportó toda la información del diccionario de DATUM a una hoja de cálculo de Google Sheets. Esta hoja contenía información detallada sobre los campos y sus parámetros.

  Extracción de Esquemas: Se utilizó una API para extraer masivamente los esquemas de los objetos en Artifactory Live. Estos esquemas se obtenían en formato JSON, lo que facilitaba su manejo mediante diccionarios y arreglos en Python.

  Manipulación de Datos: Los esquemas JSON se procesaron y transformaron en DataFrames de PySpark. Esto permitió organizar los datos de manera estructurada y comprensible.

  Unión de DataFrames: Se unieron los DataFrames de DATUM y Artifactory Live utilizando un join. Durante este proceso, se agregaron columnas booleanas que indicaban si los campos eran iguales o diferentes entre los dos entornos.

  Resultado: Esta automatización permitió una reducción significativa del tiempo de análisis, lo que facilitó el envío masivo de tickets a

arquitectura. La estandarización, inicialmente planificada para el tercer trimestre, se completó en el primer trimestre. Este logro fue reconocido por el Product Owner (PO).

uuaa	object	fase	campo	campo	val_naming	type_artif	type_dicc	val_artif	posicion_artif	posicion_dicci	val_posicio	type_null_art	type_null_d
xyz	t_abc_tabla1	master	camp_1	camp_1	Iguales	int32	int32	Iguales	1	1	Igualaes	Not null	VERDADERO
xyz	t_abc_tabla2	master	camp_2	camp_2	Iguales	string	string	Iguales	2	2	Igualaes	Not null	VERDADERO
xyz	t_abc_tabla3	master	camp_3	camp_3	Iguales	int32	int32	Iguales	3	3	Igualaes	Not null	VERDADERO
xyz	t_abc_tabla4	master	camp_4	camp_4	Iguales	int32	int32	Iguales	4	4	Igualaes	Not null	VERDADERO
xyz	t_abc_tabla5	master	camp_5	camp_5	Iguales	decimal(6,4)	decimal(6,2)	Diferentes	5	5	Igualaes	Null	FALSO
xyz	t_abc_tabla6	master	camp_6	camp_6	Iguales	decimal(15,2)	decimal(15,2)	Iguales	6	6	Igualaes	Null	FALSO

Fig.11: Resultado de la automatización que facilito a la validación

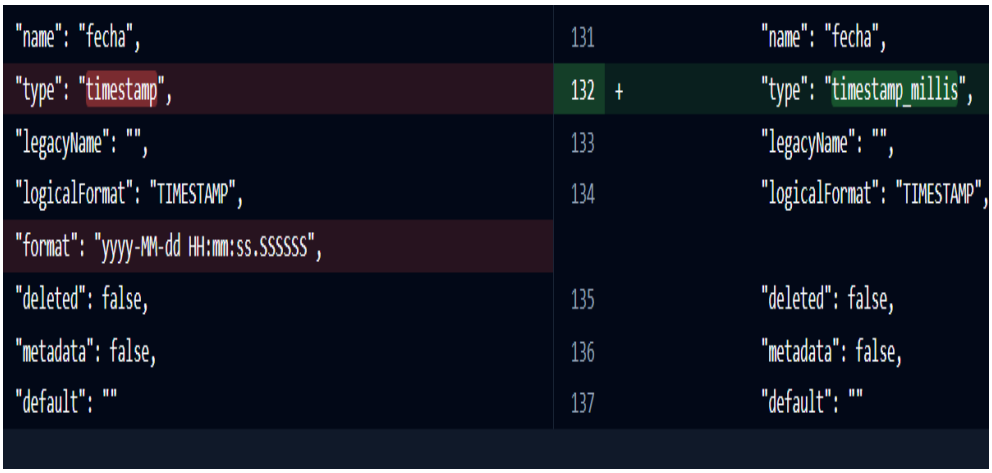


Fig.12: Comparación de Artifactory dev VS producción

Automatización de la Obtención de la Última Partición de la Ruta HDFS del Job:

- **Objetivo**: Verificar que la última partición de las rutas HDFS reportadas por España coincida con la ruta efectiva en Hadoop.
  Proceso:

Input CSV: Se utilizó un archivo CSV que contenía las tablas a consultar, sus respectivas rutas HDFS y una columna denominada "partición reportada por España".

Validación de Rutas: Se validó el acceso a las rutas HDFS utilizando manejo de excepciones para asegurar que las rutas eran accesibles.

Función PySpark: Se desarrolló una función en PySpark que ordenaba las rutas por fecha de modificación de manera descendente utilizando el método sort y librerías propias de la empresa para acceder a las rutas.

Obtención de la Última Partición: La función recorría recursivamente las sub-rutas generadas para identificar la partición más reciente. Se almacenaron estas rutas en variables y arreglos.

Comparación Masiva: Se realizó una comparación masiva entre las rutas obtenidas y las reportadas por España. En caso de discrepancias, se enviaron correos electrónicos solicitando aclaraciones.

Resultado: Esta automatización facilitó la verificación precisa de las rutas HDFS, asegurando la coherencia de las particiones y mejorando la comunicación con el equipo en España.

```
def explorar_rutas(ruta_actual):
    rutas_encontradas = []
    lista_directorios = fs.listStatus(spark._jvm.Path(ruta_actual))
    for directorio in lista_directorios:
        ruta_directorio = directorio.getPath().toString()
        if directorio.isDirectory():
            rutas_encontradas.append((extraer_fecha(ruta_directorio),ruta_directorio))
            rutas_encontradas.extend(explorar_rutas(ruta_directorio))
    rutas_ordenadas = ordenar_fechas_con_none(rutas_encontradas)
    rutas_ordenadas_10 = rutas_ordenadas[:10]
    return rutas_ordenadas_10
```

Fig.13: Simulación de la lógica de obtención de rutas

Automatización de la Verificación de Pesos:

- **Objetivo**: Verificar y gestionar las rutas y objetos en HDFS en base a su peso.
- **Proceso**:

Input CSV: Se utilizó un archivo CSV que contenía las rutas y objetos a verificar.

Verificación de Existencia: El algoritmo comprobaba la existencia de las rutas HDFS especificadas.

Obtención de Pesos: Se recopilaron todas las sub-rutas y sus correspondientes pesos en bytes, además de la fecha de modificación.

Filtrado con PySpark: Se filtraron las rutas con peso igual a 0, generando un informe final que detallaba las rutas relevantes.

Resultado: Esta automatización ayudó a descartar tablas sin peso y a cooperar con el housekeeping, que implicaba la eliminación de Jobs obsoletos y la optimización del uso de recursos en HDFS.

Estas automatizaciones no solo mejoraron la eficiencia del proceso, sino que también optimizaron la comunicación y gestión de recursos, contribuyendo al éxito del proyecto y al reconocimiento de los esfuerzos por parte del Product Owner.

```python
def obtener_peso(ruta_actual):
    rutas_peso_encontradas = []
    # Obtener el tamaño de la ruta actual
    ruta_actual_path = spark._jvm.Path(ruta_actual)
    ruta_actual_status = fs.getFileStatus(ruta_actual_path)
    peso_actual = ruta_actual_status.getLen()
    # Agregar la ruta actual con su peso a la lista
    rutas_peso_encontradas.append((peso_actual, ruta_actual))
    lista_directorios = fs.listStatus(ruta_actual_path)
    for elemento in lista_directorios:
        ruta_elemento = elemento.getPath().toString()
        if elemento.isDirectory():
            # Llamar recursivamente para explorar rutas en el directorio
            rutas_peso_encontradas.extend(obtener_peso(ruta_elemento))
        else:
            # Obtener el tamaño del archivo y agregarlo a la lista
            peso_archivo = elemento.getLen()
            rutas_peso_encontradas.append((peso_archivo, ruta_elemento))
    return rutas_peso_encontradas
```

Fig.14: Captura de una parte del algoritmo de Pesos

Flujo de Trabajo

Antes de enviar los tickets, realizábamos una serie de pruebas mínimas en Dataproc para asegurarnos de que todo estuviera en orden. Gestionábamos un reporte de Excel llamado “Inventario Stock”, donde se listaban todas las tablas y sus jobs asociados. Este inventario nos permitía identificar claramente el job que procesaba cada tabla, asignándolos en el inventario bajo uno de los cuatro estados: Tomado, Backlog, Bloqueado, Work y Live. Esta categorización nos ayudaba a mantener un control preciso del estado y progreso de cada job y tabla.

Para agilizar y asegurar el análisis, ejecutábamos varios scripts de automatización. El primer script verificaba si la partición estaba acorde con lo que estaba almacenado, comparando los datos actuales con los esperados. Este paso era crucial para garantizar la integridad de los datos y evitar posibles discrepancias en la producción.

El segundo script se centraba en la validación de esquemas. Con esta automatización, verificábamos que el esquema de los datos coincidiera con el de producción, asegurando que todos los campos, niveles de seguridad e información de partición fueran correctos y estuvieran alineados con las especificaciones de producción. Además, esta validación nos permitía identificar posibles inconsistencias o desviaciones en los esquemas antes de proceder con el despliegue.

El tercer script se encargaba de validar que las tablas contenían información. Esta verificación era esencial para asegurarnos de que las tablas no estuvieran vacías y que los datos estuvieran disponibles para su procesamiento y análisis.

Una vez completadas estas verificaciones, preparábamos una plantilla en Excel donde detallábamos los cambios necesarios para que todo estuviera alineado correctamente. Esta plantilla incluía información sobre los campos que debían ser modificados, los niveles de seguridad que necesitaban ajustes y cualquier otra información relevante para la alineación de los datos.

Con la plantilla lista, generábamos un ticket en Jira, donde sustentábamos la petición a arquitectura y etiquetábamos a los responsables del proyecto para su visto bueno. Este paso era fundamental para asegurar que todas las partes involucradas estuvieran al tanto de los cambios propuestos y pudieran revisar y aprobar las modificaciones.

Arquitectura nos respondía en el ticket si había alguna observación o si faltaba información adicional. En caso de haber observaciones, las corregíamos y volvíamos a enviar la solicitud. Si no había ninguna observación, arquitectura procedía a desplegar los cambios en el ambiente de desarrollo y actualizar el diccionario de objetos. Esta actualización era registrada en el ticket, asegurando un seguimiento transparente y detallado del proceso.

Posteriormente, realizábamos las pruebas correspondientes en Dataproc con los esquemas cambiados. Estas pruebas eran cruciales para verificar que los cambios implementados funcionaban correctamente y no introducían nuevos problemas. Si la prueba era exitosa, creábamos otro ticket para desplegar los cambios en Artifactory Live. Este ticket requería la aprobación del visto bueno de SM y SO, especialmente si se trataba de un objeto de ruta crítica. Esta aprobación era esencial para garantizar que los cambios eran seguros y estaban listos para su implementación en el entorno de producción.

En caso de que la prueba fallara, analizábamos la causa del fallo. Las razones podían variar, desde datos de entrada que no coincidían con el esquema cambiado, hasta la presencia de valores nulos en campos que no debían ser nulos, o incluso problemas con desarrollos deprecados. Realizábamos una investigación exhaustiva para identificar la raíz del problema y tomar las medidas necesarias para corregirlo y asegurar que la prueba fuera exitosa. De lo contrario, el despliegue no podría ser productivo y necesitaría más ajustes.

Una vez que el ticket era enviado a Live para el despliegue de esquemas, monitoreábamos el job durante toda la noche para asegurarnos de que no fallara. Este monitoreo continuo era esencial para detectar y resolver rápidamente cualquier problema que pudiera surgir. En caso de fallos, realizábamos un rollback inmediato,

gestionando otro ticket con arquitectura para revertir los cambios y minimizar el impacto en la producción.

Este flujo de trabajo estaba alineado con los estados que manejábamos en el inventario, lo que nos permitía tener un control preciso sobre el proceso y el estado de cada objeto. Además, este enfoque nos facilitaba la generación de indicadores de desempeño y el seguimiento del progreso.

En caso de tener un error en producción, ya sea porque no se ingestó la data input (lo cual ya no era responsabilidad nuestra), sustentábamos el error ante el equipo de Reliability. De esta manera, procedíamos a hacer rollback en caso de caídas y manteníamos ese riesgo bajo control, asegurando la estabilidad y confiabilidad del sistema en todo momento.

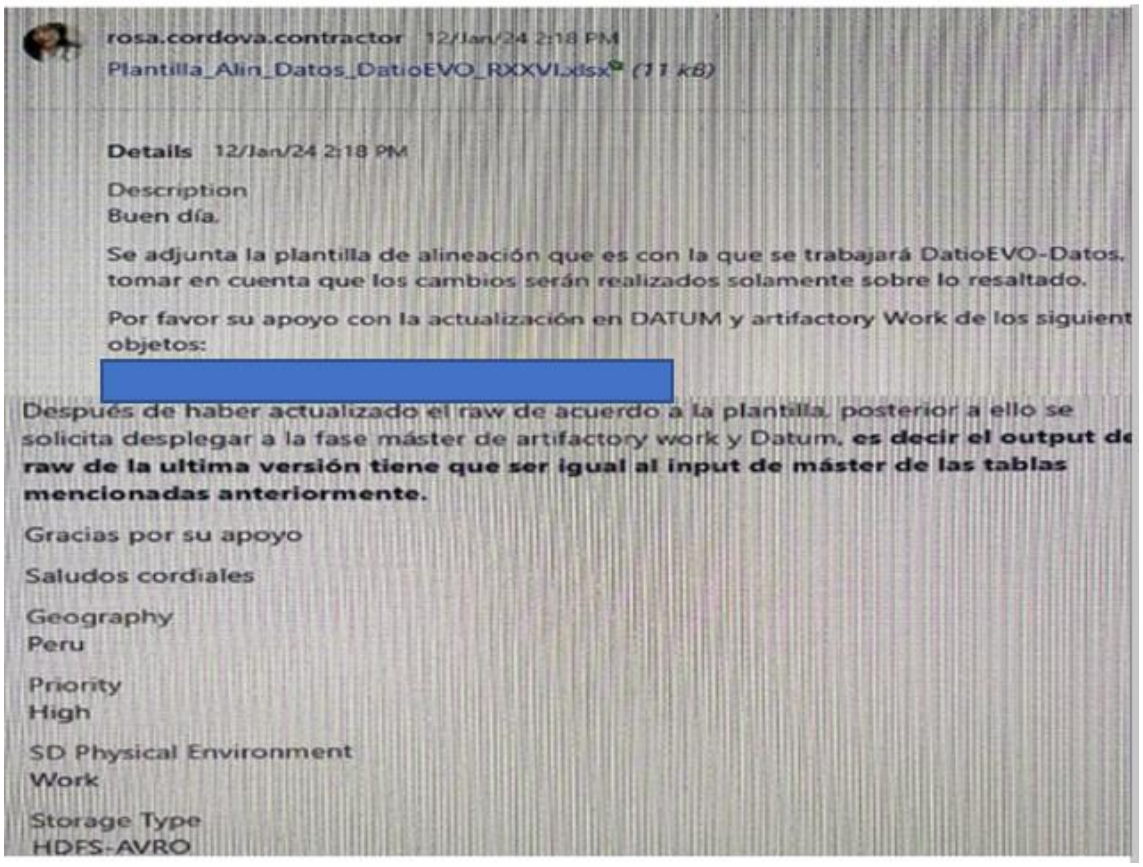


Fig.15: Ticket para el cambio en el ambiente de desarrollo

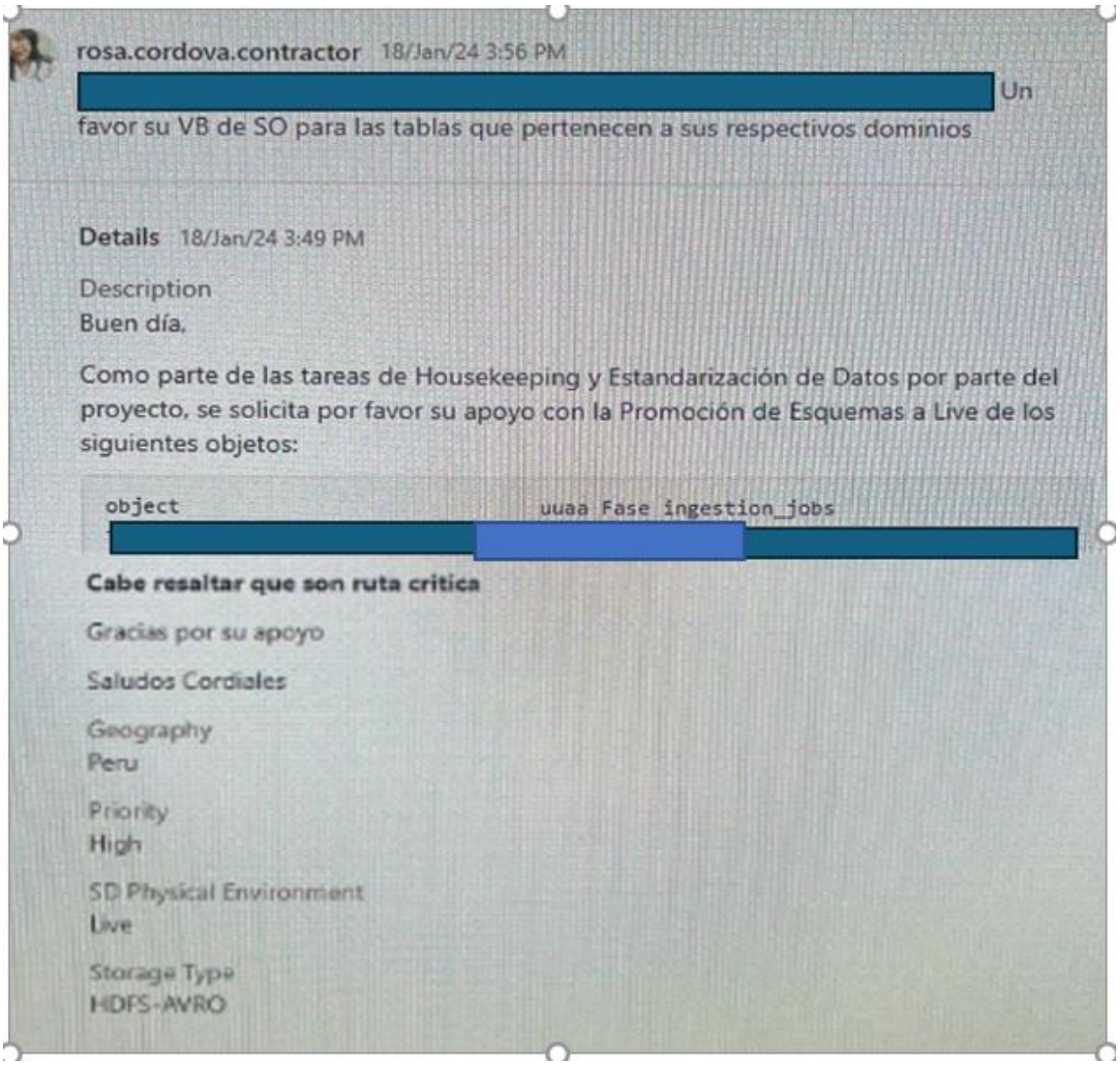


Fig.16: Ticket para el cambio en el ambiente de producción

Metodología y Resultados

Trabajamos bajo la metodología Ágil SCRUM, con reuniones diarias de seguimiento y revisiones cada 15 días al finalizar el sprint. Este proyecto se extendió por 3 Q debido a demoras en la gestión con arquitectura, pero gracias al reporte automatizado, logramos reducir el tiempo de alineación a un solo Q.

Estandarización de los Jobs de Ingesta

Aunque ya había un grupo que dedicaba a hacer estos tipos de estandarización, había ciertas veces que teníamos que hacer estandarización nosotros por que eran parte de nuestro job, entonces también teníamos responsabilidad de ello.

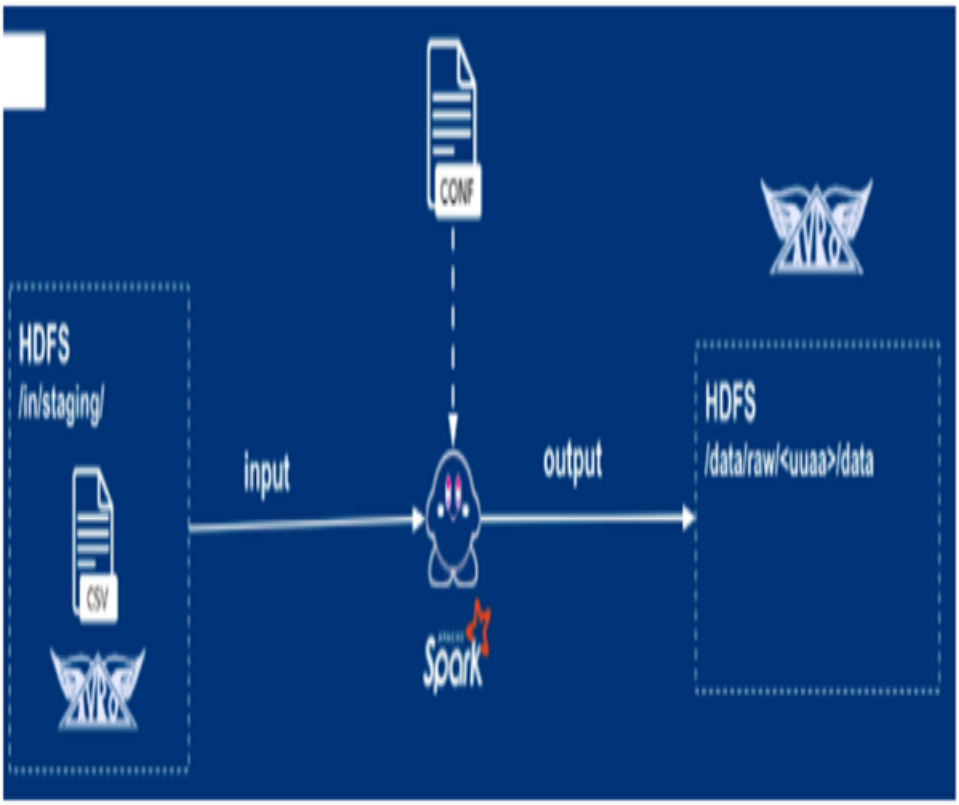


Fig.17: Imagen de cómo funcionaba una ingesta

En estos casos teníamos que verificar el job si era antiguo o nuevo, si en caso fuera antiguo se procedía por obligación a estandarizarlo para que pueda cumplir con los lineamientos de estándares de calidad, pero para ello teníamos que entender perfectamente lo que estaba haciendo el proceso de ingestas.

Estas tenían una estructura de entradas, salidas y proceso de trasformación que era acorde a lo que pedían los analistas funcionales.

Las reglas más generales establecían que las entradas debían incluir ciertos flags de carácter obligatorio, y que no debían apuntar a rutas con partición HDFS, sino directamente a Artifactory. Estos flags variaban según la capa en la que se encontraban las entradas o los orígenes de datos. Estos lineamientos ya estaban establecidos por España y eran imprescindibles para asegurar que cualquier desarrollo fuera estandarizado.

```
anulacion_esquema = true

inclusion_eliminacion_metadatos = true
```

Fig. 18: Imagen de parámetros obligatorios

```
Schema {

 path=${parametro_artifactory}"/artifactory/"${repositorio}"/schema/....

}
```

Fig.19: Imagen de la ruta del esquema a cómo debería apuntar

En cambio, para los outputs eran de carácter obligatorio incluir flags como soltar_campos_sobrantes= true, que se encargará de eliminar los campos del dataframe que no sean necesarios.

```
soltar_campos_sobrantes = true
```

Fig.20: Imagen del flags en el output

En el caso de transformaciones el filtro debería ser filter esto servía como parámetro de entrada, es decir al colocar la fecha se ingestaba solo aquella fecha.

```
transformations=[

{

type="sqlfilter"

filter= "nombre_particion=${parametro}"

}
```

Fig.21: Imagen de filter en la transformación

Flujo de como era el proceso de estandarización

ETL y Transformación de Datos:

- **Origen de Datos**: Los datos pueden ser crudos (raw) o provenientes de una capa de staging.
- **Capa Master**: Los datos se transforman y se cargan en una capa master.
- **Unión de Datos**: En algunos casos, se unen dos capas master según las necesidades del usuario.

Pruebas en Dataproc:

- **Dataproc**: Se ejecutan pruebas en esta herramienta, que es un servicio de procesamiento de datos en la nube de Google basado en Hadoop y Spark.
- **Argos**: Se usa para monitorear el estado del job y revisar los logs para identificar y entender fallos.

Una vez realizadas las pruebas, procedíamos a subir el desarrollo utilizando herramientas como Git. Ejecutaba comandos básicos como git add ., git commit -m "Descripción", y git push para enviar el código a Bitbucket. Contaba con el visto bueno de los líderes del proyecto, lo que permitía que el equipo de QA pudiera revisarlo adecuadamente. Una vez que se tenia todo ello y cumplía con los requisitos era un job que podría ser productivo.

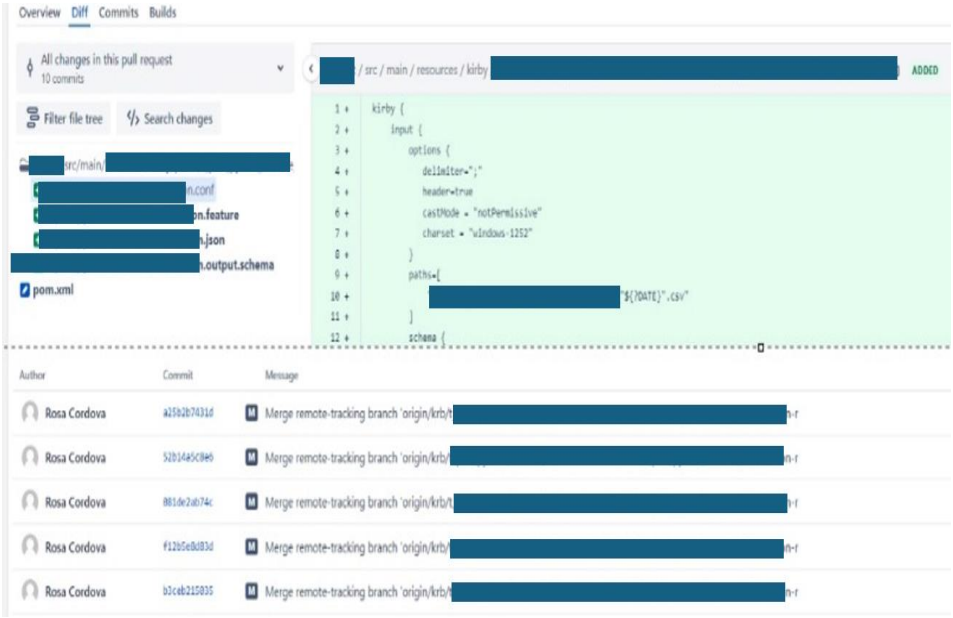


Fig.22: Imagen de envío a bibucket

Estandarización de Jobs de procesamiento

En mi experiencia profesional, he desarrollado proyectos de scaffolding adaptados a los requerimientos del negocio. Los Analistas Funcionales nos proporcionaban un documento detallando las necesidades y los resultados esperados. En algunos casos, nos encargaban estandarizar los módulos de Python, como en el siguiente ejemplo:

Teníamos que renombrar los módulos de Python de la siguiente manera:

- app.py se renombraba a run_pyspark.py
- main.py se renombraba a run_python.py
- experiment.py se renombraba a app.py

Estos cambios permitían distinguir claramente el tipo de repositorio que se estaba desarrollando (engine o modelo) y el tipo de ejecución. Posteriormente, era necesario modificar ciertas partes de los scripts en varios archivos para que apuntaran a los nuevos nombres de los módulos.

Para detallar más, el archivo app.py original contenía los parámetros de entrada del archivo conf, además de las funcionalidades que englobaban todo. Al ejecutar, solo era necesario apuntar a app.py para que todas las instrucciones se ejecutaran correctamente.

También, como parte de la estandarización, reemplazamos las librerías de Spark por Dataproc. Dataproc era una librería propia de la empresa que replicaba las funcionalidades de Spark, pero había sido desarrollada internamente para facilitar la migración a la nube aws, proporcionando configuraciones más estables. Las modificaciones incluían ajustar las operaciones de lectura, escritura y creación de DataFrames utilizando la librería Dataproc.

Antes

```
df_csv = spark.read.csv("ruta/al/archivo.csv", header=True,
inferSchema=True)
```

Ahora

```
read_partition_df =
dataproc.read().parquet(table_path).where((col(partition_column_name) ==
partition_value)
```

En el desarrollo de proyectos de scaffolding, implementábamos métricas de medición rigurosas para asegurar la calidad y estabilidad del código. Estas métricas se invocaban a través de los comandos **fluff** y testing, que formaban parte de nuestro flujo de trabajo estándar.

1. **fluff**: Este comando se utilizaba para evaluar todas las imperfecciones sintácticas del código.
2. **testing**: Este comando ejecutaba todos los test unitarios. Para que un proyecto fuera aprobado, al menos el 60% de los test unitarios debían pasar exitosamente. Esto garantizaba que las funcionalidades básicas del código estuvieran verificadas y funcionando correctamente antes de la integración.

Estas pruebas se llevaban a cabo dentro de una robusta plataform. Era un entorno completo que incluía engines y herramientas avanzadas como Vbox, a su vez, contenía Hadoop, proporcionando un entorno de pruebas y desarrollo muy similar al de producción. Este entorno nos permitía realizar pruebas exhaustivas y asegurarnos de que el código se comportaría de manera confiable una vez desplegado.

Además, para mantener un flujo de trabajo eficiente y organizado, los resultados y el código se subían al repositorio Bitbucket mediante comandos específicos. Bitbucket

estaba estrechamente integrado con nuestro sistema de control de versiones y permitía una colaboración efectiva entre los miembros del equipo.

El uso de **fluff** y testing, junto con la infraestructura de la robusta plataforma y Bitbucket, nos permitía mantener altos estándares de calidad en nuestros proyectos de pyspark. Asegurábamos que el código no solo cumpliera con los requisitos funcionales, sino que también estuviera libre de errores sintácticos y bien probado, contribuyendo a la estabilidad y el éxito de nuestros proyectos.

Fig.23: Convertura de testing

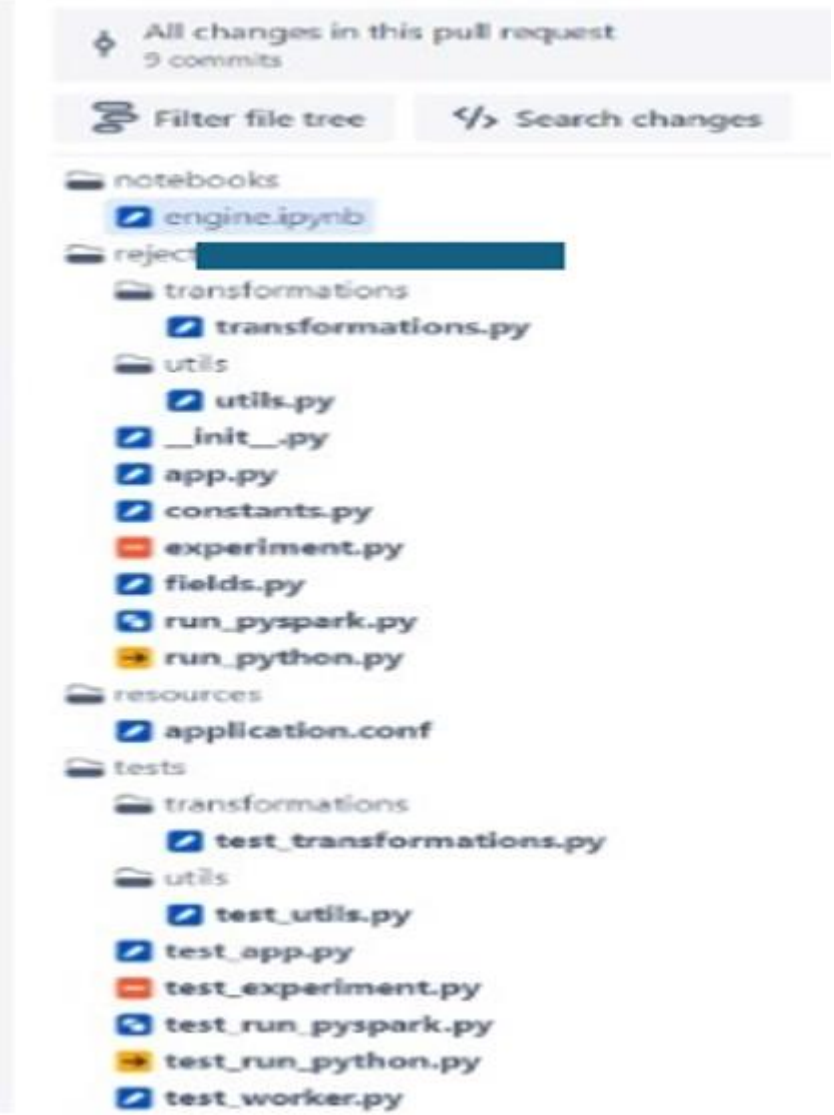


Fig.24: Estructura de pyspark Estandarizado

El flujo de subida de repositorios y desarrollos era el mismo flujo que las ingestas solo que ahora era en una máquina virtual de la empresa.

Estandarización de Jobs de compactación

El proyecto Spark Compactor fue muy interesante y consistió en configurar parámetros que apuntaban a librerías de compactación ubicadas en Dataproc de Datio. Estas librerías permitían realizar varios tipos de compactación, como **coalesce** y **repartition**, según fuera necesario.

- **Coalesce**: Esta función comprimía los ficheros almacenados en Hadoop (HDFS) a un tamaño máximo de 128 MB.

- **Repartition**: A diferencia de **coalesce**, si se encontraba un archivo de tamaño excesivo (mayor a 128 MB), **repartition** aumentaba el número de ficheros, distribuyendo los datos en fragmentos más pequeños.

La regla era que no debía haber ficheros que pesaran menos de 128 MB si había una cantidad significativa de ellos. Para gestionar esto, se utilizaban las unidades aplicativas funcionales (UUAA) y se generaba un archivo JSON por cada uno, que apuntaba a las librerías de Dataproc. Estas librerías contaban con infraestructura de arquetipo Scala.

El proceso de trabajo para el proyecto Spark Compactor se realizaba de la siguiente manera:

Trabajo Local: Se comenzaba trabajando de manera local, realizando un **git pull** de la rama develop a través de Git Bash. Después de realizar los cambios necesarios, se utilizaban los siguientes comandos de Git:

- git add .
- git commit -m "comentario"
- git push origin develop

Subida al Repositorio: Estos cambios se subían automáticamente al repositorio de Datio gracias a la configuración de la máquina a través de una VPN y los accesos correspondientes, que eran de carácter confidencial.

Construcción en Jenkins: Una vez subidos los cambios, el proceso de construcción comenzaba en Jenkins, una herramienta de despliegue. Jenkins se encargaba de transferir los cambios desde la rama local a la rama develop. Una vez finalizada la construcción y si esta era exitosa (estado "success"), se podía proceder al siguiente paso.

Pruebas en Dataproc: Con los cambios ya en Dataproc, se podían realizar pruebas de ejecución ingresando los parámetros respectivos.

Este flujo de trabajo permitía asegurar que los cambios se integraran de manera controlada y segura, garantizando la calidad y funcionalidad del proyecto Spark Compactor.

```
{
    "_id": <nombre del job de spark compactor>
    "descripcion": <nombre de la descripcion>
    "kind":<tipo de proceso>
params:{
"artifacURL": <parametros del repositorio><dirrecion del archivo jar de spark compactor>
"configURL": <parametros del repositorio><direccion en donde se alojara del archivo conf en artifactory>
"mainClass":<clase principal de scompactor>
"SparkHistoryEnabled":<true o false>
}
"runtime": <de acuerdo a lineamiento>
"size": <de acuerdo al tamaño de tu compactacion>
}
```

Fig.25: Evidencia de un job de spark compactor

```
"PARTITION_FILTER":<particion>
"TABLE_FORMAT":<avro o parquet>
"TARGET_PATH":<ruta de la tabla temporal>
"TABLE_PATH":<ruta de la tabla>
"BLOCK_SIZE_MB":128
"COMPACTION_MODE":<coalesce o repartition>
"OVERWRITE_PARTITION_ENABLED":<TRUE O FALSE>
"FORCE_TARGET_PATH_REMOVE":<TRUE O FALSE>
```

Fig.26: Evidencia de parámetros en Dataproc

Tablero BI para el manejo de indicadores

Este BI fue un dashboard creado por el propio equipo para comunicar en cada Sprint Review, una ceremonia que se realiza al final de cada sprint. Durante esta ceremonia, se revisa el incremento de los indicadores y se discuten posibles cambios en el Backlog. El objetivo es inspeccionar el trabajo realizado y adaptar el Backlog si es necesario. Posteriormente, en la retrospectiva, el equipo reflexiona sobre el sprint que acaba de finalizar. Se discuten lo que fue bien, lo que no fue tan bien, y las áreas de mejora, con el objetivo de mejorar continuamente el proceso de trabajo del equipo.

Este dashboard nos proporcionaba un detalle preciso de nuestro avance. Era alimentado por nuestra hoja de inventario de Jobs, enfocándose en los candidatos a eliminar y los pendientes de estandarizar como True. Nos permitía tener una visión detallada del tipo de job, clasificarlos y atenderlos según su complejidad. Además, podíamos ver el peso de los objetos expresado en TB, el universo de objetos, nuestro avance y los bloqueantes. También teníamos la capacidad de filtrar por cada inventario emitido en una fecha específica por España.

El dashboard era actualizado periódicamente de acuerdo a la recepción de los reportes y estaba diseñado con un modelamiento estrella para un mejor aprovechamiento de la información.

Fig.27: Dashboard general de Inventario de Datos

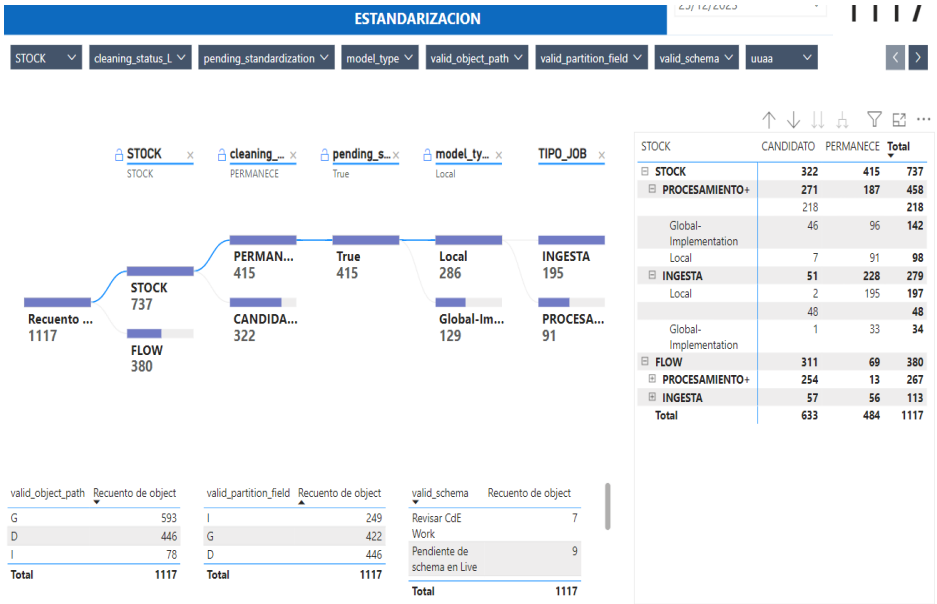


Fig.28: Dashboard particular de Estandarización.g

Capítulo IV: APORTES A LA INSTITUCION

Desde que comencé mi experiencia profesional, tanto en NTT Data como en Bluetab, siempre he dado lo mejor de mí a ambas organizaciones, consolidando mi camino hacia mi objetivo de ser una Data Engineer. A continuación, destaco mis principales aportes a estas instituciones:

Optimización de Pruebas Automatizadas en Telefónica:

En el tercer mes de trabajo, comprendiendo los flujos de extracción de datos para mis pruebas automatizadas, utilicé web scraping para extraer datos de diferentes páginas y organizarlos de manera comprensible en una tabla. Esto benefició a Telefónica, ya que el proceso de obtención de datos, que anteriormente podía demorar hasta una semana (días hábiles), se aceleró considerablemente, reduciendo la pérdida de tiempo. Además, impulsé la automatización de las pruebas funcionales utilizando herramientas como IntelliJ IDEA, BDD (Cucumber y Gherkin), Selenium para reportes de pruebas, y Java para la programación orientada a objetos. Esto aseguró que los pasos se ejecutaran de manera consistente y sin errores humanos.

Automatización y Estandarización en Bluetab:

En Bluetab, trabajando en el área de Data Engineering, fui una de las pocas personas con experiencia en automatización de mi empleo anterior. Esto me convirtió en una pieza clave para el proyecto de estandarización, ya que automatizaba todos los flujos repetitivos de validación de esquemas de datos utilizando herramientas como PySpark y/o Scala. Esto permitió reducir el tiempo de un proyecto de 3Q a 1Q. Además, gestionaba los vistos buenos de los líderes y mantenía una excelente coordinación con el equipo de Arquitectura, lo que ayudó a que los tickets se atendieran de manera más rápida y oportuna.

Adaptación y Resolución de Retos en Proyecto Scala:

Me considero una persona a la que le gusta asumir retos y aprender cosas nuevas. Esto se evidenció en el proyecto Scala, donde migré repositorios deprecados a los nuevos estándares. En este proyecto, un equipo tenía sus indicadores en rojo, pero, a pesar de no estar completamente capacitada en esa tarea específica, pude adaptarme fácilmente. Gracias al apoyo de tres Data Engineers más, logramos revertir esos indicadores y concluir el trimestre con éxito.

Capacitación y Desarrollo de Equipo en Bluetab:

Finalmente, en Bluetab, fui seleccionada para capacitar al equipo sobre los temas en los que tenía conocimiento. Realicé varias capacitaciones que ayudaron al equipo a entender mucho mejor el flujo de los desarrollos, la gestión de documentos y la obtención de vistos buenos de los líderes.

Estos aportes no solo mejoraron la eficiencia y la calidad de los proyectos en los que participé, sino que también contribuyeron al crecimiento y desarrollo de los equipos con los que trabajé.

CONCLUSIONES

Después de la experiencia profesional vivida y los desafíos que se me presentaron en la estandarización de Jobs globales para la migración a la nube, llego a las siguientes conclusiones:

1. **Optimización de la Gestión y Procesamiento de Datos en Hadoop:** El proyecto logró optimizar significativamente la gestión y el procesamiento de datos en entornos Hadoop mediante la estandarización de jobs y la compactación eficiente de ficheros. Al aplicar técnicas avanzadas y parámetros precisos como coalesce y repartition, se mejoró no solo la calidad de los datos procesados, sino también su disponibilidad y velocidad. Esta optimización no solo facilitó la gestión de grandes volúmenes de datos, sino que también redujo el tiempo de ejecución en procesos críticos, permitiendo una mejor toma de decisiones basadas en datos en la nube.

   KPI: Se consiguió una reducción del 15% en el tiempo de procesamiento de datos a nivel global, lo que resultó en una mejora tangible en los tiempos de respuesta de las consultas y procesos críticos.

2. **Estandarización de Jobs y Esquemas:** Se llevó a cabo una estandarización exhaustiva del desarrollo de jobs y esquemas en entornos de Big Data, alineándolos con los más altos estándares de calidad y mejores prácticas de la industria. Este esfuerzo garantizó la consistencia y la integridad de los datos en todas las fases del ciclo de vida, desde la ingestión hasta el procesamiento y almacenamiento. La estandarización no solo permitió una mejor interoperabilidad entre sistemas, sino que también facilitó la migración de jobs hacia la nube, asegurando que los procesos fueran más sostenibles y escalables en el tiempo.

   KPI: Se alcanzó una tasa de cumplimiento del 100% en la alineación de los jobs y esquemas con los estándares de calidad y consistencia de datos, mejorando la calidad general del ecosistema de Big Data.

3. **Compactación Eficiente en Hadoop:** La implementación de un proceso sistemático y automatizado de compactación de ficheros en Hadoop (HDFS) permitió optimizar de manera considerable tanto el uso del almacenamiento como el rendimiento en el procesamiento de grandes volúmenes de datos. Al utilizar las técnicas de coalesce y repartition, se logró una distribución más eficiente de los datos, evitando la fragmentación y garantizando que los ficheros respetaran las políticas de tamaño óptimo establecidas para su gestión. Este proceso no solo redujo la sobrecarga de almacenamiento, sino que también aceleró los tiempos de procesamiento y mejoró el acceso a la información, lo cual resultó en una mayor eficiencia operativa.

KPI: Se logró una reducción del 25% en el uso de espacio en disco gracias a la compactación eficiente, mientras que los tiempos de acceso a los datos se mejoraron en un 20%, aumentando la velocidad de los procesos de lectura y escritura en HDFS.

4. **Gestión Ágil del Proyecto con Scrum:** La aplicación de la metodología ágil Scrum en la gestión del proyecto permitió una colaboración fluida y continua entre todos los miembros del equipo, facilitando la adaptación a cambios en los requerimientos y asegurando entregas incrementales y de alto valor. La integración de herramientas de control de versiones y despliegue continuo también fue clave para la gestión efectiva de los cambios, permitiendo un seguimiento detallado de las versiones y minimizando los riesgos durante los despliegues en producción. Este enfoque ágil no solo impulsó la productividad del equipo, sino que también fomentó un ambiente de trabajo colaborativo en el que se priorizaba la innovación y la mejora continua.
   KPI: Se incrementó la productividad del equipo en un 30%, lo que se tradujo en entregas continuas de valor y en una mejor capacidad de adaptación ante cambios de última hora, manteniendo un alto nivel de calidad en todas las etapas del proyecto.

RECOMENDACIONES

Luego de la experiencia vivida en la estandarización de Jobs globales para la migración a la nube, puedo sugerir las siguientes recomendaciones, que espero sirvan de base para otros futuros colegas:

1. **Actitud y Voluntad de Apoyar:**
   La actitud positiva y la disposición para apoyar siempre que se necesite, incluso si eso implica sacrificar tu tiempo, son cualidades que te ayudan a ser reconocido por cualquier organización y te promueven para avanzar en tu carrera, tal como ocurrió en mi caso.
2. **Conocimiento del Flujo End-to-End:**
   Conocer el flujo end-to-end es crucial, ya que permite visualizar la gestión general de los procesos y, con ello, crear valor en cada etapa. Este conocimiento integral facilita la identificación de áreas de mejora y la implementación de soluciones efectivas.
3. **Iniciativa y Proactividad:**
   Un profesional de sistemas debe siempre tener la actitud de querer mejorar y ofrecer soluciones a cualquier desafío. No es necesario esperar a que nos digan qué hacer; tener iniciativa nos permite destacar y ser valiosos en cualquier proyecto.
4. **Investigación y Aprendizaje Continuo:**
   La investigación constante es fundamental. Cada organización, como fue en mi caso, suele tener blogs informativos para cada proceso. Leer estos recursos en nuestro tiempo libre nos permite enriquecer nuestro conocimiento y estar mejor preparados para enfrentar nuevos retos.
5. **Trabajo en Equipo y Compartición de Conocimientos:**
   El trabajo en equipo es esencial, y la transmisión de conocimientos es fundamental. No debemos ser egoístas, ya que el egoísmo no lleva a nada bueno. Compartir conocimientos fortalece al equipo y mejora los resultados del proyecto.

6. **Capacitación en Nuevas Herramientas:**

   Siempre debemos capacitarnos en nuevas herramientas, incluso fuera de nuestro trabajo actual. Esto nos permite no quedarnos desfasados y tener la capacidad de reconversión en cualquier área y empleo, manteniéndonos relevantes y competitivos en el mercado laboral.

7. **Adaptabilidad y Flexibilidad:**

   La adaptabilidad y flexibilidad son esenciales en el campo de la tecnología. Estar dispuesto a aprender y adaptarse a nuevas metodologías y herramientas nos permite enfrentar desafíos con confianza y eficacia.

8. **Gestión del Tiempo y Prioridades:**

   La gestión eficiente del tiempo y la capacidad de priorizar tareas son habilidades cruciales. Identificar qué es urgente e importante nos permite ser más productivos y eficientes, evitando el estrés innecesario y mejorando nuestro desempeño general.

Estas recomendaciones reflejan la importancia de una actitud proactiva, el aprendizaje continuo, y la colaboración en equipo. Siguiendo estos principios, cualquier profesional puede destacar y contribuir significativamente al éxito de sus proyectos y organizaciones.

REFERENCIAS BIBLIOGRAFICAS

[1] V. G. Villanueva Arrasco, «Implementación de un datamart on premise y su migración a la nube para el análisis de información de postventa en la empresa Derco.,» Universidad Nacional Federico Villareal, Lima, 2024.

[2] Velasquez Ruiz, J. C. A., & Lujan Tello, G. C., « Migración del servidor Mainframe a la nube Microsoft Azure en una institución bancaria en el Perú.,» Universidad Peruana de Ciencias Aplicadas, Lima, 2023.

[3] Bernache Torres, G. M., & Sumarriva Rubio, L. A., « Proyecto de servicios tecnológicos de la empresa Daily Technology implementados en plataforma en la nube.,» Universidad Peruana de Ciencias Aplicadas, Lima, 2023.

[4] Salas Asto, A. M., & Vásquez Liñan, C. A., «Propuesta de migración a Cloud Computing para la mejora de la gestión y eficiencia en costos de la inmobiliaria LGP, 2023.,» Universidad San Ignasio de Loyola, Lima, 2024.

[5] Sbrollini Cepeda, M. J., & Palomino Rodriguez, J. H., « Propuesta de desarrollo de una aplicación web y móvil en la nube para la mejora del proceso de cobranzas de una organización religiosa.,» Universidad Peruana de Ciencias Aplicadas, Lima, 2023.

ANEXOS

Casos de automatización

Este proyecto es una configuración típica de automatización de pruebas basada en Java y Maven, que utiliza Cucumber para la definición de pruebas en lenguaje natural, Selenium para la automatización del navegador, y ExtentReports para la generación de informes. Los archivos y directorios están organizados de manera que soporten una estructura clara y mantenible para el desarrollo y ejecución de pruebas automatizadas.

1. Estructura del Proyecto

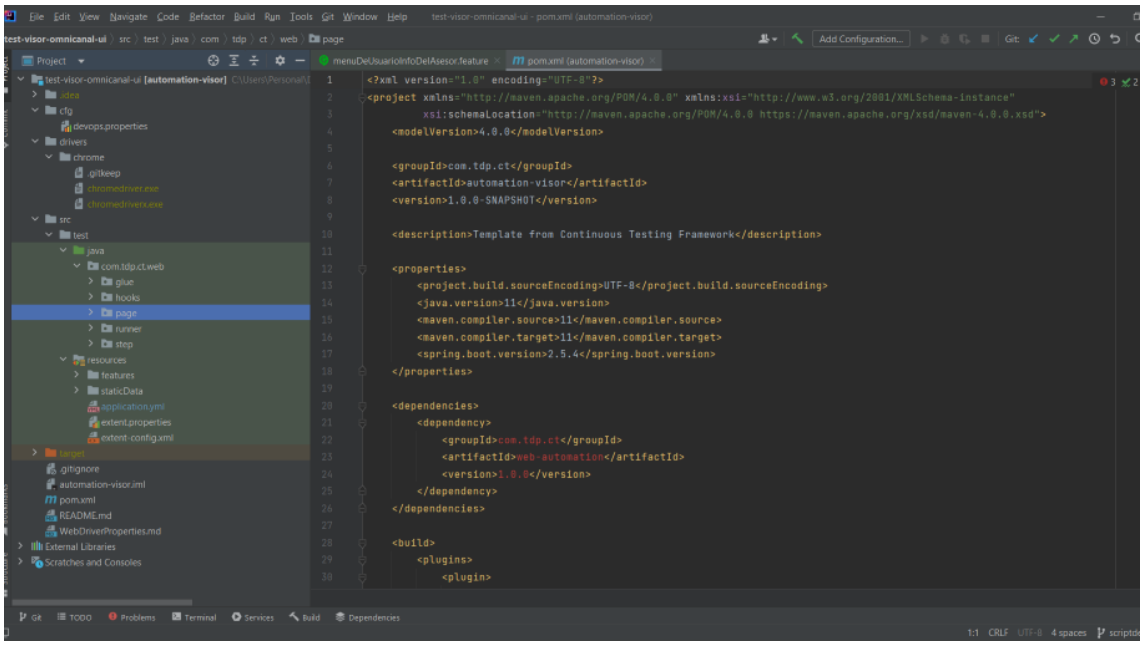
- **.idea**: Carpeta que contiene archivos de configuración del proyecto específicos de IntelliJ IDEA.
- **cfg/devops.properties**: Archivo de propiedades probablemente utilizado para configuraciones relacionadas con DevOps.
- **drivers/chrome**: Contiene archivos ejecutables del ChromeDriver, necesarios para la automatización de navegadores usando Selenium.
- **src/test/java**: Directorio que contiene el código fuente de las pruebas automatizadas.
    - **com/tdp/ct/web**: Nombre del paquete, dentro del cual se organizan las diferentes clases y archivos.
        - **glue**: c¿Contiene las clases que unen los pasos definidos en las pruebas con el código Java.
        - **hooks**: Probablemente contiene clases que manejan las acciones antes y después de las pruebas (hooks de Cucumber).
        - **page**: Contiene las clases del patrón Page Object, que representan las páginas web que serán probadas.
        - **runner**: Posiblemente contiene las clases que ejecutan las pruebas.
        - **step**: Probablemente contiene las definiciones de los pasos de las pruebas (steps de Cucumber).
- **src/test/resources**: Directorio que contiene recursos adicionales para las pruebas.

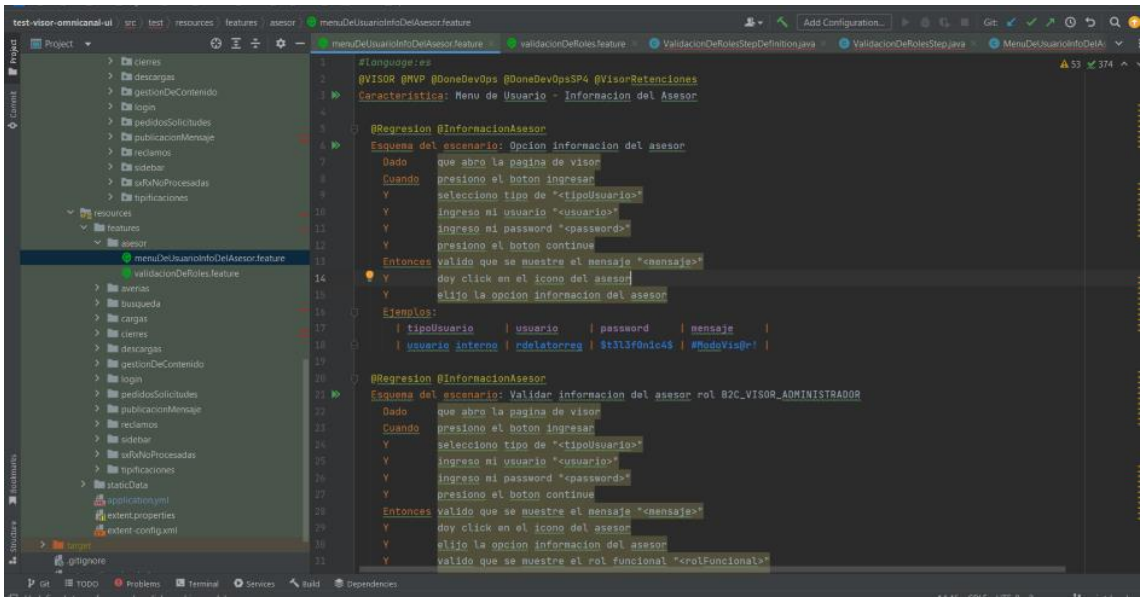
- **features**: Contiene archivos de características (feature files) escritos en Gherkin para Cucumber.
- **staticData**: Probablemente contiene datos estáticos utilizados en las pruebas.
- **application.yml**: Archivo de configuración YAML, posiblemente para Spring Boot.
- **extent.properties**: Archivo de configuración para ExtentReports, una herramienta de reporte.
- **extent-config.xml**: Archivo XML de configuración para ExtentReports.

2. Archivo pom.xml

El archivo pom.xml (Project Object Model) es fundamental para proyectos Maven. Aquí se definen las configuraciones, dependencias y plugins necesarios para el proyecto. Algunos elementos destacados del pom.xml

- **Model Version**: La versión del modelo POM utilizada.
- **GroupId, ArtifactId y Version**: Identifican de manera única el proyecto.
- **Descripción**: Describe brevemente el proyecto.
- **Properties**: Define propiedades del proyecto como la versión de Java, Spring Boot, y configuraciones del compilador.
- **Dependencies**: Lista las dependencias que el proyecto necesita. En este caso, parece que tienes una dependencia interna (com.tdp.ct:web-automation).
- **Build Plugins**: Lista los plugins necesarios para construir el proyecto.





Reportería con Cucumber report

Resumen del Informe de Automatización

Duración Total: 7 minutos y 23.342 segundos.

Fecha y Hora de Inicio: 5 de julio, 11:56:07.640 a. m.

Fecha y Hora de Fin: 5 de julio, 12:03:30.982 p. m.

Resultados Generales

Características:

Pasadas: 1

Falladas: 0

Saltadas: 0

Escenarios:

Pasados: 1

Fallados: 0

Saltados: 0

Pasos:

Pasados: 34

Fallados: 0

Saltados: 0

Detalles de la Característica y Escenario Probado

Nombre de la Característica: CAEQ contado con documento CE por canal Tienda.

Nombre del Escenario: Realizar CAEQ (contado) con CE en canal Tienda.

Duración del Escenario: 7 minutos y 23.331 segundos.

Pasos Totales en el Escenario: 34, todos pasados.

Pasos Detallados del Escenario

Algunos pasos clave en el escenario incluyen:

Inicio de Sesión en la Página de Movistar:

Abrir la página de Movistar.

Presionar el botón "Iniciar Sesión".

Seleccionar el tipo de usuario "usuario externo".

Ingresar el nombre de usuario y la contraseña.

Ingresar el captcha y presionar "Continuar".

Validar el mensaje de inicio de sesión exitoso "Bienvenid@".

Ingresar Datos del Documento:

Seleccionar el tipo de documento "CE".

Ingresar el número del documento.

Presionar el botón "Consultar".

Seleccionar Plan y Detalles del Equipo:

Seleccionar el botón de detalle del número de celular existente.

Presionar el botón "Renovar Plan".

Validar que se presenta la pantalla con el título "Ofertas sugeridas".

Seleccionar el tipo de pago "Al Contado".

Ingresar detalles de permanencia, tipo de pago y nombre del equipo.

Validar que existen resultados de búsqueda de equipos.

Presionar el botón "Ver detalle" y seleccionar el equipo.

Finalizar el Proceso:

Ingresar un correo electrónico para registro.

Validar y aceptar el contrato móvil.

Imprimir el contrato solicitado.

Tags Utilizados

@DoneDevOpsPI12

@CaeqContadoTienda

@RegreDito

@DoneDevOps

@BERSERKERS

Resumen de la Ejecución

Todos los pasos y escenarios fueron ejecutados exitosamente sin fallos ni saltos.

La automatización cubrió el proceso completo de inicio de sesión, consulta de documento, selección de plan y equipo, y finalización del registro y contrato.

Beneficios de la Automatización

Eficiencia: Reduce el tiempo necesario para realizar pruebas repetitivas.

Precisión: Asegura que los pasos se ejecuten de manera consistente y sin errores humanos.

Documentación: Provee un registro detallado de cada paso y su resultado, facilitando la revisión y auditoría.