



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Software

**Planificación y ejecución de actualización de base de
datos para mantener el soporte de la plataforma como
servicio RDS de una aplicación legacy de e-learning**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Software

AUTOR

Diego Alejandro PASTOR GUERRERO

ASESOR

Víctor Hugo BUSTAMANTE OLIVERA

Lima, Perú

2025



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Pastor, D. (2025). *Planificación y ejecución de actualización de base de datos para mantener el soporte de la plataforma como servicio RDS de una aplicación legacy de e-learning*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Software]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Diego Alejandro Pastor Guerrero
Tipo de documento de identidad	DNI
Número de documento de identidad	72488849
URL de ORCID	https://orcid.org/0009-0005-8702-1844
Datos de asesor	
Nombres y apellidos	Víctor Hugo Bustamante Olivera
Tipo de documento de identidad	DNI
Número de documento de identidad	25655590
URL de ORCID	https://orcid.org/0000-0001-8805-9629
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cesar Alberto Molina Neyra
Tipo de documento	DNI
Número de documento de identidad	40553679
Miembro del jurado 1	
Nombres y apellidos	John Ledgard Trujillo Trejo
Tipo de documento	DNI
Número de documento de identidad	06187585
Datos de investigación	
Línea de investigación	No aplica
Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento

Ubicación geográfica de la investigación	País: Perú Departamento: Lima
	Provincia: Lima Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2024
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Software

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 19:00 horas del día 13 de enero del año 2025, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Molina Neyra Cesar Alberto (Presidente), Lic. Trujillo Trejo John Ledgard (Miembro) y el Lic. Bustamante Olivera Víctor Hugo (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/jtg-ystx-bsz>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“PLANIFICACIÓN Y EJECUCIÓN DE ACTUALIZACIÓN DE BASE DE DATOS PARA MANTENER EL SOPORTE DE LA PLATAFORMA COMO SERVICIO RDS DE UNA APLICACIÓN LEGACY DE E-LEARNING”**, por el Bachiller **Pastor Guerrero Diego Alejandro**; para obtener el Título Profesional de Ingeniero de Software.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 (DIECIOCHO)**.

A continuación el Presidente de Jurados el Mg. Molina Neyra Cesar Alberto, declara al Bachiller **Ingeniero de Software**.

Siendo las 19:44 horas, se levantó la sesión.

Presidente

Mg. Molina Neyra Cesar Alberto

Miembro

Lic. Trujillo Trejo John Ledgard

Miembro Asesor

Lic. Bustamante Olivera Víctor Hugo



CERTIFICADO DE SIMILITUD

Yo, BUSTAMANTE OLIVERA VICTOR HUGO en mi condición de asesor acreditado con la Resolución Decanal N° 001059-2024-D-FISI/UNMSM y Oficio N° 006-PTTSP-FISI-2024 del Trabajo de Suficiencia Profesional cuyo título es Planificación y ejecución de actualización de base de datos para mantener el soporte de la plataforma como servicio RDS de una aplicación legacy de e-learning, presentado por el bachiller PASTOR GUERRERO DIEGO ALEJANDRO para optar el título de Ingeniero de Software CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 9% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional**.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título.

Lima, 8 de enero de 2025

Firma del Asesor

Nombres y Apellidos del asesor: BUSTAMANTE OLIVERA VICTOR HUGO

DNI 25655590

DEDICATORIA

Dedico este trabajo a las personas que más han influido en mi desarrollo personal y profesional.

A mi abuela Irma, quien siempre me apoyó durante mi época de universidad y a quien siempre he admirado.

A mi abuelo Everardo, por ser una persona íntegra en todo aspecto, un gran padre y un ejemplo a seguir.

A mi madre Olga, quien ha sido mi consejera, amiga y soporte durante toda mi vida y sin quien no estaría donde estoy.

A mi padre José, por quien decidí dedicarme a esta hermosa carrera.

Y finalmente a mis hermanos Renato, Andrea y Gabriel, quienes han sido parte de mi inspiración para los logros que he obtenido a lo largo de mi carrera profesional.

AGRADECIMIENTOS

A dios, quien me ha dado la salud y fortaleza para culminar este trabajo.

A mi familia, quienes me han apoyado incondicionalmente para lograr este objetivo.

A mis compañeros de trabajo, sin quienes no habría sido posible este informe de investigación.

A mi asesor Víctor Bustamante, por sus enseñanzas y consejos en la redacción de este informe.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

**Planificación y ejecución de actualización de base de datos para
mantener el soporte de la plataforma como servicio RDS de una
aplicación legacy de e-learning**

Autor: Pastor Guerrero, Diego Alejandro

Asesor: Bustamante Olivera, Víctor Hugo

Título: Trabajo de Suficiencia Profesional para optar el Título
Profesional de Ingeniero de Software

Fecha: Diciembre 2024

RESUMEN

Este proyecto tuvo como objetivo la planificación y ejecución de una actualización de la base de datos MySQL 5.7.42 a MySQL 8.0.34, junto con modificaciones a los sistemas dependientes de la misma para una plataforma legacy de e-learning. Esta actualización se realizó debido a un cambio en el soporte de Amazon RDS para la mencionada base de datos, lo que desembocó en la creación de este proyecto con el objetivo de mantener las operaciones de la compañía. Para su realización, se utilizó la metodología Kanban junto con algunas prácticas ágiles como el planning poker y la programación en pares para una iteración constante dada la incertidumbre de los cambios necesarios en la plataforma legacy para dicha actualización; así como el uso de pruebas *end-to-end*, de estrés y mediciones constantes; todo aquello con el objetivo de que el tiempo muerto durante la actualización sea menor a quince minutos.

Palabras claves: Kanban, Base de Datos, Sistemas Legacy, Pruebas, Agile

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF SYSTEMS AND INFORMATICS ENGINEERING

PROFESSIONAL SCHOOL OF SOFTWARE ENGINEERING

**Planning and execution of database upgrade to maintain RDS platform
as a service support for a legacy e-learning application**

Author: Pastor Guerrero, Diego Alejandro

Advisor: Bustamante Olivera, Víctor Hugo

Title: Professional Sufficiency Work to opt for the Professional Title of
Software Engineer

Date: December 2024

ABSTRACT

The objective of this project was the planning and execution of an upgrade of the MySQL 5.7.42 database to MySQL 8.0.34, along with modifications to the systems dependent on it for a legacy e-learning platform. This upgrade was performed due to a change in Amazon RDS support for the mentioned database, which led to the creation of this project with the objective of maintaining the company's operations. For its implementation, the Kanban methodology was used along with some agile practices such as planning poker and pair programming for a constant iteration given the uncertainty of the changes needed in the legacy platform for this update; as well as the use of end-to-end testing, stress testing and constant measurements; all with the objective that the downtime during the update is less than fifteen minutes.

Key words: Kanban, Databases, Legacy Systems, Testing, Agile

TABLA DE CONTENIDOS

RESUMEN	v
ABSTRACT	vi
TABLA DE CONTENIDOS	vii
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS.....	xi
INTRODUCCIÓN.....	1
CAPÍTULO I: TRAYECTORIA PROFESIONAL.....	3
CAPÍTULO II CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA	6
2.1 EMPRESA – ACTIVIDAD QUE REALIZA	6
2.2 VISIÓN.....	7
2.3 MISIÓN.....	7
2.4 ORGANIZACIÓN DE LA EMPRESA	7
2.5 ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS	9
2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN.....	9
CAPITULO III ACTIVIDADES DESARROLLADAS	11
3.1 SITUACIÓN PROBLEMÁTICA.....	11
3.1.1 DEFINICIÓN DEL PROBLEMA	12
3.2 SOLUCIÓN	16
3.2.1 OBJETIVOS.....	16
3.2.2 ALCANCE.....	16
3.2.3 ETAPAS Y METODOLOGÍA	17
3.2.4 FUNDAMENTOS UTILIZADOS	20
3.2.5 IMPLEMENTACIÓN DE LAS ÁREAS, PROCESOS Y BUENAS PRÁCTICAS.....	32
3.3 EVALUACIÓN	91
3.3.3 EVALUACIÓN ECONÓMICA / EVALUACIÓN COSTO-BENEFICIO	91
CAPITULO IV REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	94
CAPITULO V CONCLUSIONES Y RECOMENDACIONES	96
5.1 CONCLUSIONES.....	96
5.2 RECOMENDACIONES	97
FUENTES DE INFORMACIÓN	98
GLOSARIO	101

ANEXOS	103
--------------	-----

ÍNDICE DE FIGURAS

Figura 1 Organigrama de la empresa.....	7
Figura 2 Línea de tiempo para el soporte de versiones MySQL	11
Figura 3 Topología del clúster de base de datos.....	13
Figura 4 Ambiente Local.....	14
Figura 5 Ambientes de Desarrollo/Pruebas	15
Figura 6 Ejemplo de un mapa de tareas.....	18
Figura 7 Ejemplo de un tablero Kanban.....	20
Figura 8 Métodos de estimación de historias de usuario.....	22
Figura 9 Mapa de tareas de PLAY-34	34
Figura 10 Actualización manual de una base de datos	35
Figura 11 Creación de un ambiente de pruebas.....	40
Figura 12 Mapa de tareas de PLAY-73	42
Figura 13 Documentación MySQL	43
Figura 14 Mapa de tareas de PLAY-74.....	44
Figura 15 Consultas realizadas	45
Figura 16 Función urlencode.....	46
Figura 17 Error en la definición de SoftDeletePromoSeats	47
Figura 18 Mapa de tareas de PLAY-75	48
Figura 19 Errores de corrupción.....	49
Figura 20 Operaciones para eliminar esquemas no usados	50
Figura 21 Eliminación de vistas no usadas en el catálogo	52
Figura 22 Mapa de tareas de PLAY-76	53
Figura 23 Sintaxis de creación de índices	54
Figura 24 Consulta sobre objetos en la base de datos.	55
Figura 25 Mapa de tareas PLAY-72.....	56
Figura 26 Comparativa de parametros entre versiones	57
Figura 27 Creación de los parámetros para MySQL 8	58
Figura 28 Mapa de tareas de PLAY-82.....	59
Figura 29 Error de conexión a la base de datos.....	60
Figura 30 Mapa de tareas para PLAY-110.....	62
Figura 31 Pruebas ejecutadas	63
Figura 32 Errores en pruebas de estrés.....	63
Figura 33 Mapa de tareas de PLAY-155	66
Figura 34 blue/green en el ambiente de desarrollo.....	67
Figura 35 lazy loading en despliegues blue/green.....	68
Figura 36 Mapa de tareas PLAY-187.....	71
Figura 37 Parametrización del scrubber	72
Figura 38 Cambios en el repositorio de infraestructura	73
Figura 39 Mapa de tareas de PLAY-203	74
Figura 40 Script de instalación de MySQL	75
Figura 41 Archivo de configuración MySQL.....	76
Figura 42 Scripts de Chef para usar el nuevo instalador	76
Figura 43 Mapa de tareas de PLAY-213	78
Figura 44 Correo de Amazon sobre la actualización de la base de datos.....	79

Figura 45	Mapa de tareas de PLAY-220	81
Figura 46	Error al ejecutar MySQL 8.0.34 en Ubuntu 18.04	82
Figura 47	Constructor de MySQL 8.0.34	83
Figura 48	Mapa de tareas de PLAY-226	85
Figura 49	Blue/Green en producción.....	86
Figura 50	Lag de replicación tras creación del blue/green	87
Figura 51	Comparación de operaciones entre las bases de datos	87
Figura 52	Bases de datos de pruebas en producción	88
Figura 53	Lag de replicación de las bases de datos de prueba	89
Figura 54	Lag de replicación de la ejecución final del plan	89

ÍNDICE DE TABLAS

Tabla 1 Formación Académica.....	3
Tabla 2 Experiencia Profesional.....	3
Tabla 3 Formación Complementaria	5
Tabla 4 Ejemplo de historia de usuario	22
Tabla 5 Épica PLAY-1	32
Tabla 6 Historia de Usuario PLAY-34	33
Tabla 7 Reporte de compatibilidad previo a la actualización.....	36
Tabla 8 Resumen de alternativas para actualizar la base de datos	38
Tabla 9 Historia de Usuario PLAY-73	41
Tabla 10 Tiempo de ejecución para actualizar la codificación a utf8mb4	43
Tabla 11 Historia de Usuario PLAY-74	44
Tabla 12 Historia de usuario PLAY-75	47
Tabla 13 Vistas en el catálogo a eliminar.....	51
Tabla 14 Historia de Usuario PLAY-76	53
Tabla 15 Historia de Usuario PLAY-72	56
Tabla 16 Historia de Usuario PLAY-82	59
Tabla 17 Historia de Usuario PLAY-110	61
Tabla 18 Historia de Usuario PLAY-155	65
Tabla 19 Tiempo aproximado al realizar la actualización.....	68
Tabla 20 Historia de Usuario PLAY-187	71
Tabla 21 Historia de Usuario PLAY-203	73
Tabla 22 Historia de Usuario PLAY-213	77
Tabla 23 Historia de Usuario PLAY-220	80
Tabla 24 Historia de Usuario PLAY-226	84
Tabla 25 Costos asociados con el proyecto en dólares.....	91
Tabla 26 Coste de soporte extendido de AWS en dólares.....	92

INTRODUCCIÓN

Hasan et al. (2023) define a un sistema legacy como un sistema crítico que soporta un proceso de negocio principal de la organización con la restricción de utilizar software, hardware y recursos obsoletos que requieren de mantenimiento.

La organización a tratar en el presente trabajo es una dedicada al e-learning, con un alcance global que no solo permite el consumo directo de cursos para los clientes, sino también cumple el rol de un *hub* de cursos que son republicados por otras plataformas de e-learning. Esta es la razón principal por la cual disrupciones en el servicio o pérdida de información puede devenir en consecuencias críticas para los acuerdos de servicio (SLAs por sus siglas en inglés) a los cuales la empresa está sujeta.

El sistema legacy que funge de base para las operaciones de la organización tenía en sus componentes una base de datos MySQL 5.7 la cual se utiliza en producción y, sobre la cual, se generan varias copias para los ambientes de desarrollo y pruebas. Dicha base de datos se despliega en el servicio RDS de Amazon Web Services (AWS), este servicio informó a la organización que la versión de la base de datos sería deprecada, lo cual derivó en el presente proyecto.

En este contexto, el proyecto tuvo tres objetivos principales. Primero, que la actualización de la versión de la base de datos se generase con una total integridad de los datos. Segundo, la actualización debía cumplir con el tiempo límite que se puede tener para el mantenimiento del sistema y debía lograrse antes de la fecha de deprecación. Finalmente, asegurar la capacidad para mantener las operaciones del sistema tras la actualización.

Para lograr estos objetivos, se conformó un sub-equipo conformado por dos ingenieros de software y un contratista especializado en bases de datos, y se aplicaron metodologías ágiles como Kanban para la organización de las tareas y prácticas como el uso de sub-equipos, programación en pares, mapas de tareas y las reuniones constantes con los *Product Manager*. El uso de dichas técnicas fue fundamental, dadas las incertidumbres que envolvían a este proceso dados los cuantiosos componentes posiblemente afectados.

Asimismo, el uso constante de pruebas *end-to-end* y de estrés ayudaron a dar confianza al equipo sobre la viabilidad de los cambios que se realizaron. Y el uso de scripts para monitorear los cambios necesarios en la base de datos sirvieron para la toma de decisiones al momento de la elaboración del plan de actualización final.

En este proyecto participé como uno de los dos ingenieros de software asignados, siendo uno de los responsables de la elaboración y ejecución del plan final, así como la documentación respectiva.

A continuación, se describen los capítulos abordados en el presente trabajo.

- En el Capítulo I, se da un resumen de la trayectoria profesional del autor
- En el Capítulo II, se detalla el contexto sobre el cual se desarrolló la experiencia profesional, indicando algunas características técnicas sobre la empresa y sus requisitos de disponibilidad.
- En el Capítulo III, se formula el problema, objetivos, alcance, las etapas y la ejecución, así como el valor del proyecto
- En el Capítulo IV, se indica la reflexión crítica de la experiencia.
- En el Capítulo V, se explican las conclusiones y las recomendaciones, algunas de las cuales fueron implementadas posteriormente.

CAPÍTULO I: TRAYECTORIA PROFESIONAL

En este capítulo se revisará la experiencia profesional y la formación académica del autor.

El autor tiene un grado de bachiller en la carrera de Ingeniería de Software de la Universidad Nacional Mayor de San Marcos. Actualmente posee el cargo de Ingeniero de Software y cuenta con más de 4 años de experiencia profesional en el desarrollo de software y la refactorización de procesos. El autor posee una pasión por la calidad de código y las prácticas ágiles, así como la capacidad para trabajar en diversas tecnologías y plataformas.

Tabla 1

Formación Académica

2015 – 2019

Bachiller en Ingeniería de Software

Universidad Nacional Mayor de San Marcos

Facultad de Ingeniería de Sistemas e Informática, EAP Software

Nota: Elaboración propia

Tabla 2

Experiencia Profesional

Tekton Labs	
Rol	Ingeniero de Software
Periodo	Diciembre 2021, Presente
Funciones	• Mantenimiento de sistemas legacy hechos en Drupal 7 • Migración de un sistema de la plataforma OpsWorks a ECS en Amazon Web Services. • Diseño e implementación de un plan de actualización de base de datos. • Desarrollo de lambdas con C#, Python y JavaScript. • Creación de pipelines de CI/CD en Github, GitLab y CircleCI. • Monitoreo y resolución de incidentes.

	• Creación de suites de pruebas unitarias y <i>end to end</i> • Diseño y arquitectura de nuevos componentes de software. • Creación de integraciones con sistemas externos como RingCentral. • Rediseño de procesos legacy utilizando diseño orientado a eventos. • Creación de scripts de migraciones de base de datos utilizando Liquidbase. • Creación de una librería de componentes de UI utilizando React.
--	---

Oechsle

Rol	Analista de Sistemas
Periodo	Abril 2021 a noviembre 2021
Funciones	• Diseño y refactorización de procesos legacy usando la plataforma GCP junto con Pub/Sub para mayor resiliencia • Desarrollo de procesos de integración para un feed de Google y Facebook utilizando la información de los clientes de la compañía. • Creación de programas de tipo batch para proporcionar información a un DataMart.

Manantial Tecnológico E.I.R.L

Rol	Desarrollador Backend
Periodo	Febrero 2020 a marzo 2021
Funciones	• Desarrollo de procesos background para consumir información de APIs externas e integrar a la base de datos de la compañía. • Creación de APIs con SpringBoot, Kotlin y NestJS • Creación de una web interna para el mantenimiento de tablas de configuración usando VueJS. • Migración y rediseño de aplicaciones de Net Framework a .Net core • Integración con la plataforma Vtex para gestiones de stock • Creación de APIs REST para un sistema de servicios. • Rediseño del proceso de facturación de e-commerce del cliente

Nota: Elaboración propia

Tabla 3

Formación Complementaria

AWS Certified Developer – Associate	2020 – 2023
AWS Certified Cloud Practitioner	2024 – 2027
Microsoft Certified: Azure Fundamentals	2024
IELTS Academic	2024 – 2026
IELTS General Training	2023 – 2025

Nota: Elaboración propia

CAPÍTULO II CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1 EMPRESA – ACTIVIDAD QUE REALIZA

Tekton Labs es una empresa de estrategia, diseño y tecnología. La empresa provee a organizaciones servicios end-to-end, desde investigación y diseño, hasta el desarrollo y escalamiento de productos. Tekton también provee de servicios de aumento de personal para compañías de Estados Unidos.

Es en una de estas compañías donde se ha realizado el proyecto. La mencionada empresa es una de e-learning. Dedicada a la distribución de contenido educativo a través de una plataforma web. A grandes rasgos la empresa posee dos formas de disponer dicho contenido.

De manera directa, mediante el acceso a los cursos directamente sobre la plataforma permitiendo diversos medios de autenticación para los distintos tipos de usuario.

Y de manera indirecta, esto es, la empresa funge como distribuidor, o intermedio, entre creadores de contenido y otras compañías que desean acceder a cursos creados por diversos proveedores, en este caso la empresa provee a las compañías de un amplio catálogo de contenido que puede ser utilizado en sus sistemas.

Los cursos creados pueden seguir distintos estándares, como por ejemplo SCORM, AICC, TIN/CAN y el sistema se encarga de traducir estos formatos para que una empresa que solo pueda consumir cursos de tipo AICC también pueda consumir un curso que internamente fue creado siguiendo el estándar SCORM.

Adicionalmente, también posee un servicio que facilita la creación de contenido a clientes mediante un software web, contenido que puede ser publicado en el catálogo general de la compañía.

2.2 VISIÓN

Imaginamos un futuro en el que las personas de todo el mundo tengan un acceso sin restricciones a las habilidades y conocimientos esenciales para alcanzar sus aspiraciones, fomentando una comunidad global en la que cualquiera pueda perseguir sus objetivos con claridad, determinación y los recursos que necesita para tener un impacto duradero.

2.3 MISIÓN

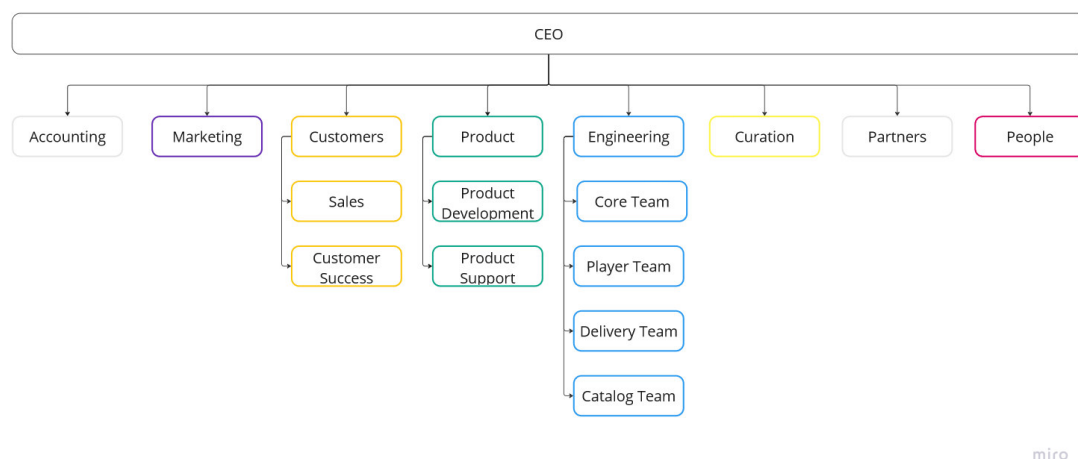
Nuestro propósito es dotar a las organizaciones de las herramientas necesarias para formar equipos altamente eficaces y ampliamente respetados. Al ofrecer soluciones de aprendizaje específicas, capacitamos a las empresas para cultivar plantillas que destaquen, inspiren y contribuyan positivamente.

2.4 ORGANIZACIÓN DE LA EMPRESA

La empresa está organizada en varias áreas o equipos cada uno encargado de un área específica del negocio que deben tratar. Algunas de estas áreas poseen divisiones menores cuando el dominio sobre el que se trabaja es más amplio. Esto se puede apreciar en el siguiente organigrama.

Figura 1

Organigrama de la empresa



Nota: Elaboración propia.

La descripción de cada área.

- **Accounting:** El equipo de contabilidad es responsable de las nóminas, la elaboración de presupuestos, las cuentas a pagar (facturas de proveedores) y las cuentas a cobrar (facturas y pagos de clientes), e informes financieros
- **Marketing:** El equipo de marketing es responsable de las campañas de prospección para la generación de oportunidades de ventas, la comunicación de marketing con los clientes, la publicidad digital, los eventos del sector, el marketing de socios, el material colateral de productos (como hojas de ventas, información sobre la competencia e historias de éxito de clientes) y, por último, las páginas de marketing del sitio web de la empresa.
- **Customers:**
 - **Sales:** Equipo responsable de permitir a nuevas empresas y clientes obtener el contenido vendido por la empresa.
 - **Customer Success:** Equipo encargado de asesorar a nuevos clientes sobre qué tipo de contenido necesitan para cumplir con sus objetivos. El equipo es responsable de garantizar el éxito del despliegue, la adopción y el crecimiento de los clientes actuales.
- **Product:**
 - **Product Development:** Equipo responsable de definir la hoja de ruta de los productos desarrollados por el equipo de ingeniería. También tiene por responsabilidad la comunicación y entrenamiento sobre los nuevos despliegues generados por ingeniería.
 - **Product Support:** El equipo encargado de brindar apoyo ante cualquier tipo de incidente a los clientes de la compañía.
- **Engineering:**
 - **Core:** Equipo enfocado en componentes relacionados a los diferentes tipos de autenticación en la plataforma.
 - **Player:** Equipo encargado del mantenimiento del “reproductor”, siendo este el software que muestra el contenido de los cursos.
 - **Delivery:** Equipo dedicado a las integraciones con los clientes que utilizan la empresa como intermediario.
 - **Catalog:** Equipo responsable del mantenimiento del “catalogo”, siendo este el software de ventas y administración del contenido.

- **Curation:** Equipo responsable de proveer a los clientes con listas de cursos que se ajusten a sus competencias organizacionales. Gestiona las relaciones con los editores y mantiene el catálogo actualizado a los cambios en el mundo.
- **Partners:** Equipo encargado de gestionar las relaciones con empresas asociadas. Pueden incluir proveedores de LMS (Learning Management System) y LXP (Learning Experience Platform) entre otros.
- **People:** El equipo de personal es responsable de la contratación, la formación y la incorporación, la administración de la oficina, los eventos internos y las operaciones de personal.

2.5 ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

El autor viene ejerciendo el cargo de Ingeniero de Software desde diciembre del 2021 en Tekton. Sin embargo, en la compañía descrita anteriormente, cumple con el mismo rol desde el 07 de noviembre del 2022.

Durante este periodo de tiempo, el autor ha estado asignado a los equipos de *core* y reproductor. Las funciones designadas fueron las siguientes:

- Mantenimiento de los sistemas de la compañía en sus diferentes tecnologías.
- Gestión de incidentes y soporte técnico.
- Diseño e implementación de soluciones de software.
- Ejecución de pruebas.
- Elaboración de documentación
- Participar en el diseño de pipelines para el despliegue de aplicaciones.

2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

El autor ha sido parte de numerosos proyectos y tareas en la compañía asignada. Algunas de estas son descritas a continuación:

- Diseño e implementación de un nuevo flujo de autenticación para clientes usando OKTA.

- Rediseño de un proceso de integración para generar actualizaciones en los LMS de clientes acerca de eventos durante la reproducción de los cursos.
- Automatizaciones de procesos mediante el uso de scripts.
- La actualización de la base de datos de la compañía.
- Detección y resolución de bugs en las soluciones de software de la compañía.
- Corrección de información errónea en las bases de datos causada por bugs en los sistemas.
- Integración de fuentes de datos entre AWS y el DATA WAREHOUSE de la compañía.

Una gran parte de las tareas realizadas tienen que ver con correcciones y mantenimiento de los sistemas legacy de la compañía. Perteneciendo a este grupo los sistemas del reproductor y el catálogo.

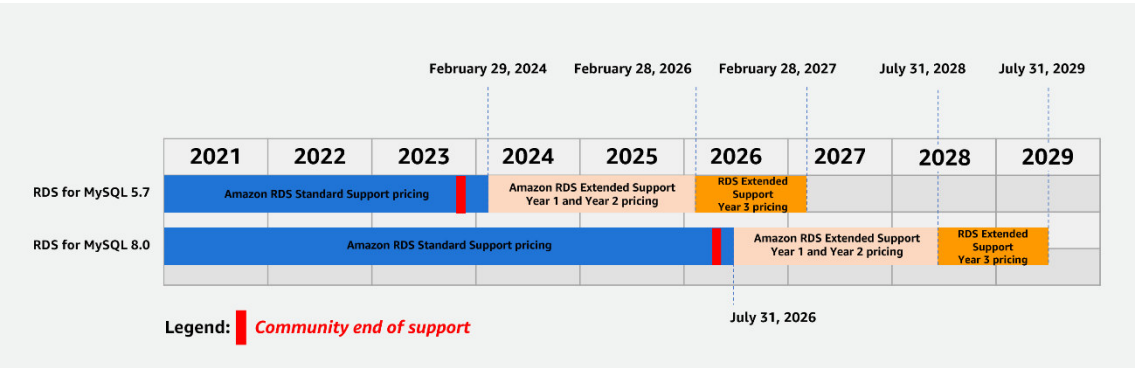
CAPITULO III ACTIVIDADES DESARROLLADAS

3.1 SITUACIÓN PROBLEMÁTICA

La organización posee un sistema legacy como base de sus operaciones, este sistema tiene una base de datos MySQL 5.7 como su repositorio central de información. Esta base de datos se despliega sobre el servicio RDS en Amazon Web Services, como se puede apreciar en la imagen, esta versión en específico estuvo pronta a ser deprecada. Esta deprecación implica que no existirán nuevos parches de seguridad, actualizaciones y/o correcciones de errores en la plataforma a menos que se opte por el plan de extensión de AWS. (Karumaji, 2023)

Para prevenir problemas de seguridad y disponibilidad a futuro, dado que no se desea el coste adicional por el programa de extensión de AWS, se necesita la creación de un plan, y su posterior ejecución, para lograr la actualización de esta base de datos manteniendo el tiempo establecido por los SLA de la organización. Adicionalmente, se debe lograr que los ambientes que dependen de esta base de datos, como los de desarrollo y pruebas, puedan continuar sus operaciones sin mayor interrupción.

Figura 2
Línea de tiempo para el soporte de versiones MySQL



Nota: Adaptado de “Introducing Amazon RDS Extended Support for MySQL databases on Amazon Aurora and Amazon RDS”, por Karumaji, 2023, AWS Database Blog.

3.1.1 DEFINICIÓN DEL PROBLEMA

La producción de sistemas legacy es una práctica inevitable en empresas que requieren de software para la ejecución de sus negocios; dichas empresas usualmente no consideran los riesgos de mantener dichos sistemas, riesgos como procesos lentos, mantenimientos complicados, problemas de seguridad, falta de soporte de sistema entre otros. (Jomhari et al, 2024) Aquel último riesgo, la falta de soporte, es la causa principal del problema en el presente proyecto.

Amazon provee de un programa de soporte extendido, este soporte extendido se habilitó tras haber iniciado el proyecto. Dicho plan acarrea costos adicionales que son dados a discreción de la plataforma AWS. Según (AWS, s.f-a) se incrementará el costo de 0 \$ a 0.1 \$ cada hora por cada VCPU usado por los primeros dos años del soporte extendido; en el tercero este valor se extenderá a 0.2 \$.

Al momento de realización del proyecto, la compañía utilizaba 152 VCPUs de entre los ambientes de producción y desarrollo. Un simple cálculo definiendo cuatro semanas por cada mes genera 122 572.8 \$ anuales adicionales por los primeros años, y el doble por el tercero. La organización consideró este coste adicional como excesivo.

Adicionalmente al costo adicional causado por el plan de soporte extendido. Se requieren evaluar los diferentes componentes que se integran con la base de datos. Para definir la escala del problema, se procederá a describir los componentes principales involucrados en la plataforma y luego se procederá a describir los diferentes ambientes involucrados. Los sistemas principales que componen la plataforma son los siguientes.

Primero, el **reproductor**. Este sistema se desarrolló en .NET Framework 4.8 y se desplegó en la plataforma de software IIS. Entre sus funciones tenemos:

1. Entregar el contenido de los cursos a la web.
2. Registrar información relacionada a los lanzamientos, las acciones de los usuarios en los cursos y el rastreo de las actividades.
3. Gestionar el acceso a los cursos mediante el uso de licencias.
4. Integración con sistemas de terceros acerca de los cursos tomados por los usuarios en la plataforma.

El segundo componente es el **catálogo**. Dicho sistema fue desarrollado sobre Drupal 7, un gestor de contenido (CSM por sus siglas en inglés) que usa PHP como lenguaje de programación. Entre las funciones del sistema tenemos:

1. Fungir como UI principal de la compañía siendo el contenedor de una aplicación web hecha en Angular doce.
2. Gestionar el proceso de compra de los cursos.
3. Ser la página de administración para los colaboradores de la empresa.
4. Encargarse de la renovación de los diversos tipos de suscripciones que oferta la compañía.

Ambos sistemas dependen del mismo servidor de base de datos generando una dependencia fuerte hacia la misma.

Incidentalmente, la compañía tiene numerosas réplicas de la misma base de datos, esto puede ser ejemplificado en la siguiente imagen.

Figura 3

Topología del clúster de base de datos

	DB identifier ▲	Status ▼	Role ▼	Engine ▼
☐	☒ legacy-prod-1	✔ Available	Primary	MySQL Co...
☐	☐ legacy-prod-1-fivetran	✔ Available	Replica	MySQL Co...
☐	☐ legacy-prod-1-noproduct	✔ Available	Replica	MySQL Co...
☐	☐ legacy-prod-1-replica	✔ Available	Replica	MySQL Co...

Nota: Adaptado de la cuenta de la empresa

En lo que respecta a los ambientes, existen tres tipos los cuales dependen de la base de datos, o de copias de esta. Entre estos tenemos:

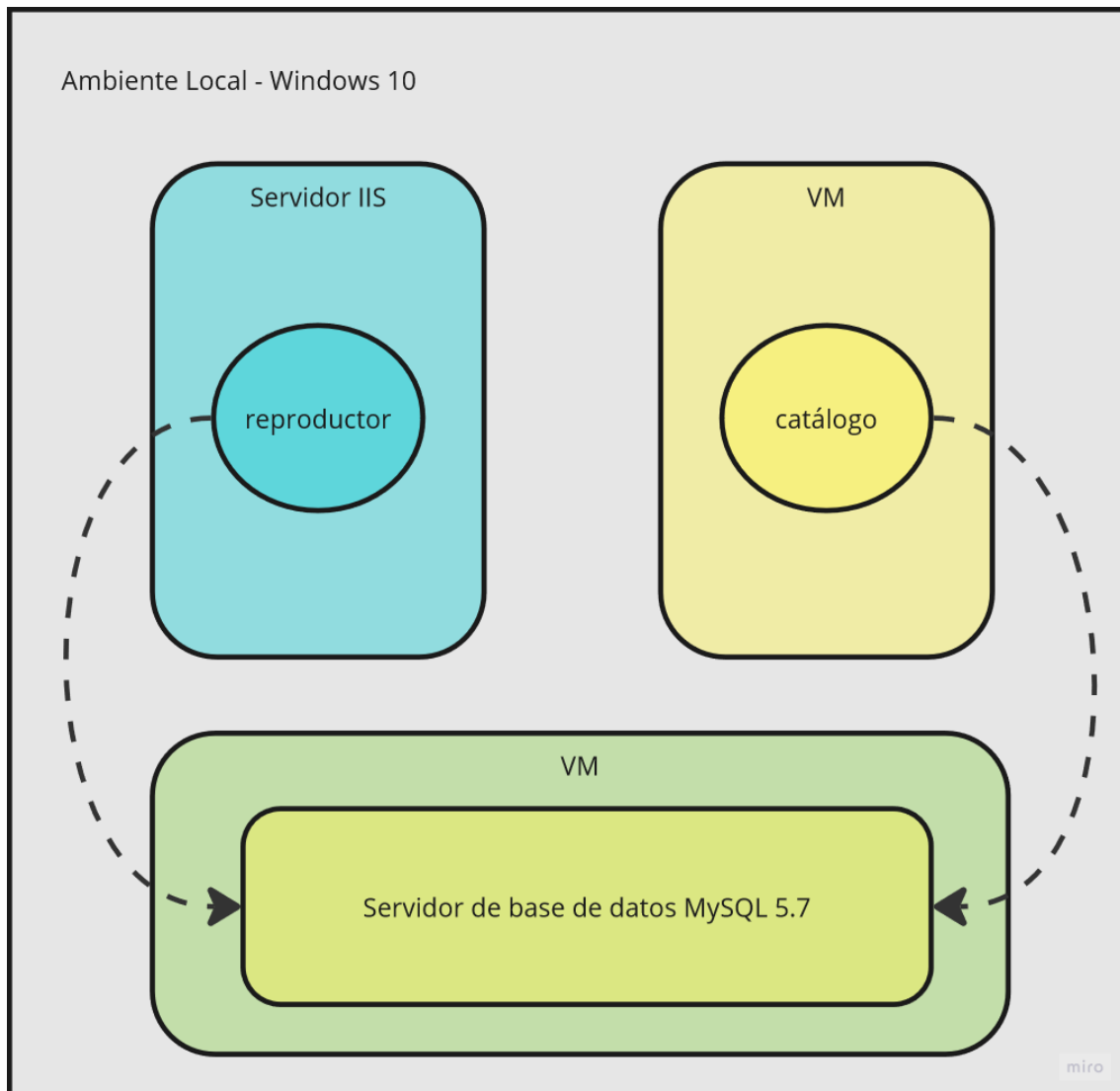
El ambiente local, el cual es usado por los desarrolladores en sus labores diarias, este ambiente consiste en los siguientes componentes:

1. Un servidor IIS donde se despliega el reproductor.

2. Una máquina virtual (VM por sus siglas en inglés) donde existe una copia del catálogo.
3. Una VM donde se despliega una copia anonimizada de la base de datos.

Figura 4

Ambiente Local



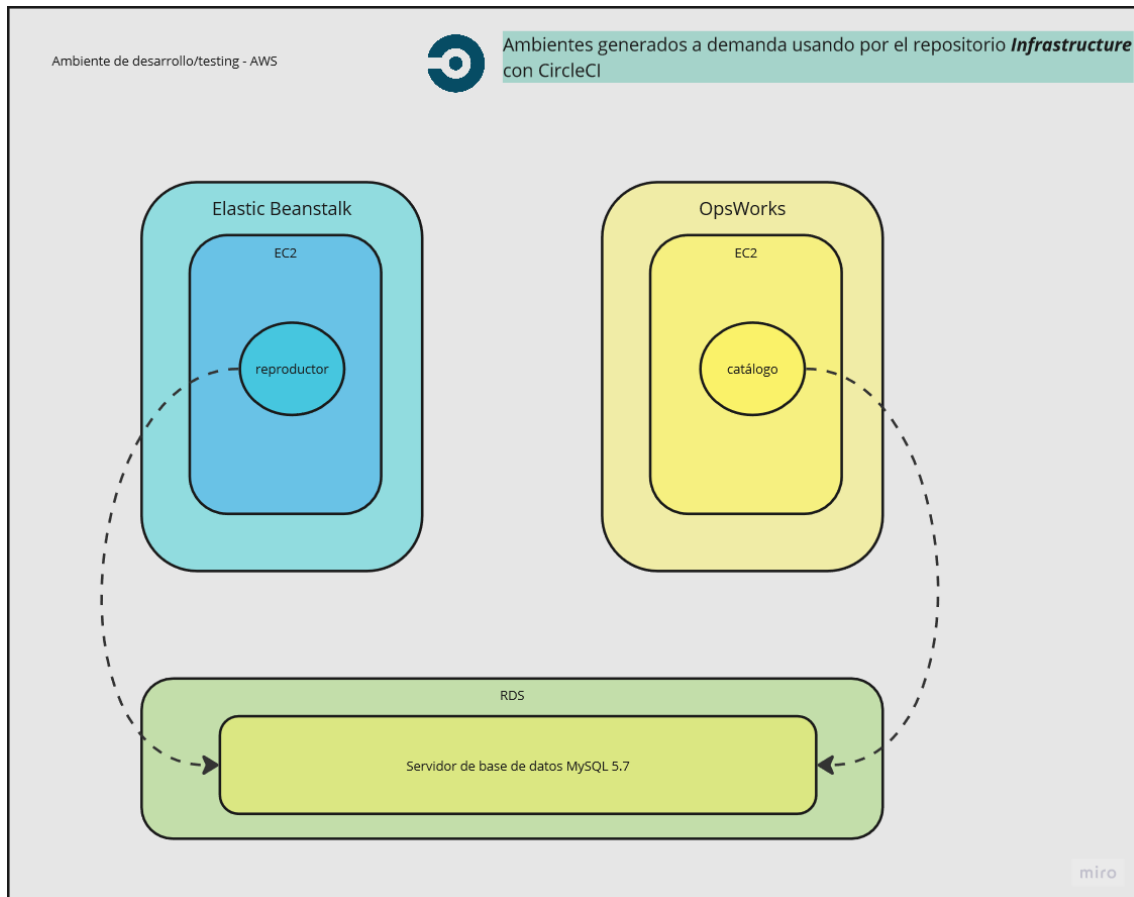
Nota: Elaboración propia

El segundo tipo de ambiente son los de pruebas y desarrollo. Estos sistemas se diferencian del local en cuanto son desplegados sobre AWS utilizando infraestructura como código con *terraform* y como motor de CI/CD a *CircleCI*. Los componentes en este escenario

son similares a los anteriores, siendo diferenciados solo por la plataforma sobre los que se despliegan

Figura 5

Ambientes de Desarrollo/Pruebas



Nota: Elaboración propia

Finalmente, el ambiente de producción es idéntico a los de desarrollo, con las siguientes diferencias:

1. El volumen de recursos dedicados es mayor. Por ejemplo, el tipo de instancia EC2 es distinto entre los ambientes.
2. No es gestionado por el repositorio de **Infrastructure**, su configuración es en gran parte manual y específica para producción.
3. La base de datos de producción genera las copias anonimizadas para desarrollo.

Es tras analizar lo anteriormente expuesto, que la complejidad de la tarea de actualizar la base de datos se da a notar, ya que es un componente usado en diversos ambientes a lo largo de la compañía. De esta forma, el problema no era solo el hecho de necesitar realizar la actualización en el ambiente de producción cumpliendo con los SLA de la compañía. Si no, también, realizar los cambios necesarios para que cada ambiente pudiese consumir la nueva versión de la base de datos y continuar con las operaciones empresa.

3.2 SOLUCIÓN

3.2.1 OBJETIVOS

Objetivo General

Crear y ejecutar un plan de actualización de base de datos a una versión MySQL 8.0 que permita la realización de este proceso en un tiempo menor a quince minutos, siendo este tiempo menor al tiempo de mantenimiento permitido por los SLA de la compañía y manteniendo la capacidad operacional de la empresa tras la actualización.

Objetivos Específicos

1. Crear un plan de actualización para la base de datos que pueda ser reutilizado para posteriores actualizaciones.
2. Crear un plan de *rollback* en caso de fallo en los procesos.
3. Identificar los componentes afectados en los diferentes ambientes utilizados en la compañía (local, pruebas, desarrollo y producción).
4. Realizar los cambios respectivos sobre las aplicaciones para soportar la nueva versión de MySQL.
5. Asegurar un umbral de tiempo de actualización de 15 minutos a lo máximo.
6. Mantener operativo el sistema en producción después de la actualización.
7. Asegurar que todos los ambientes afectados utilicen la misma versión de MySQL.

3.2.2 ALCANCE

Alcance Funcional

El presente trabajo definirá las siguientes actividades para el alcance funcional:

1. Creación de un plan reutilizable de actualización de base de datos, así como un plan de rollback sobre el mismo.
2. Documentar las diferentes dependencias entre los ambientes. Identificar los componentes en los ambientes locales, de desarrollo/pruebas y producción.
3. Realizar cambios sobre el catálogo, reproductor, repositorios de infraestructura y el proceso de creación de base de datos de desarrollo que sean necesarios para soportar la versión de MySQL 8.0.
4. Realizar cambios sobre los procesos de CI/CD en los ambientes afectados para asegurar la compatibilidad con la nueva versión de la base de datos.
5. Ejecutar pruebas unitarias, funcionales, e2e y de estrés para validar cualquier cambio necesario en el punto tres.
6. Asegurar que el tiempo de actualización en producción no exceda los quince minutos. Esta restricción no necesariamente aplicará a los ambientes de desarrollo.
7. Verificar que todos los ambientes afectados usen la misma versión de MySQL tras la actualización.

Alcance Organizacional

El alcance organizacional está orientado para los equipos del reproductor y catálogo. Equipos que dependen directamente de la base de datos para sus operaciones diarias.

3.2.3 ETAPAS Y METODOLOGÍA

Metodología

La metodología usada en la compañía para este proyecto estuvo basada en Kanban para la gestión de las tareas. Se buscó iterar constantemente en base a nueva información obtenida durante el desarrollo del proyecto.

Para ello los proyectos se definen en **épicas**, estas agrupan **historias de usuario** las cuales a su vez se disgregan en **tareas**.

Las historias de usuario son estimadas y especificadas usando la técnica del *planning poker* lo cual ayuda al equipo a estar alineado en los requerimientos para completar las

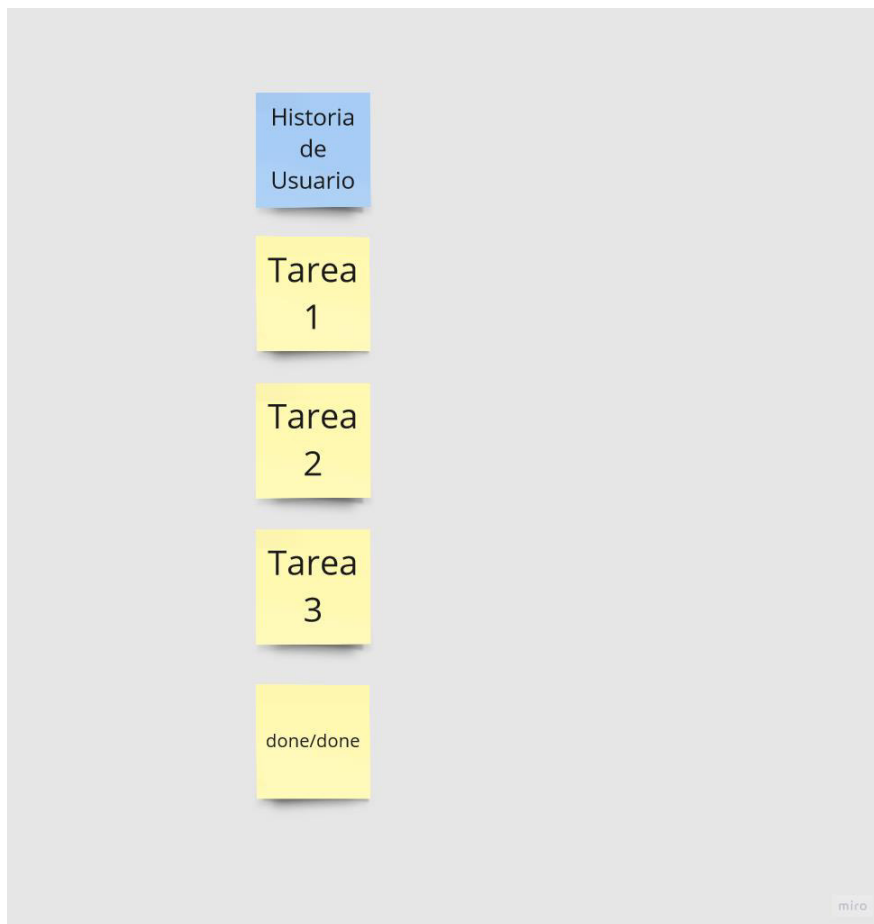
tareas. El desarrollo de estas tareas se podía realizarse utilizando programación en pares o de manera individual dependiendo de las decisiones tomadas por el equipo.

Finalmente, el uso de **mapas de tareas** fue utilizado para poder enumerar posibles tareas que se requieran por cada historia de usuario. Este mapeo de tareas se realizaba para beneficio del sub-equipo y sirvió para determinar tareas que pudiesen ser paralelizadas. Una tarea que se añade al final de cada historia es la llamada *done/done* donde los interesados se reúnen para validar que las tareas realizadas y los entregables son los adecuados para considerar la historia hecha.

Bajo esta metodología, los miembros del equipo tenían potestad de crear nuevas historias de usuario acordes a los resultados de otras historias de usuario las cuales debían ser planificadas y estimadas posteriormente.

Figura 6

Ejemplo de un mapa de tareas



Nota: Elaboración propia

Etapas

Bajo la metodología de la empresa, un proyecto tiende a ser iniciado por una historia de usuario de investigación la cual deriva en posteriores fases derivadas de los resultados obtenidos.

Por ello, para este proyecto se pudieron detectar las siguientes etapas:

Primero, una fase de **Investigación inicial**. Esta fase tenía como objetivo definir los lineamientos generales del proyecto. Ejemplos de tareas realizadas en esta etapa fueron:

- Pruebas de actualización con data de producción.
- Ejecutar una suite de pruebas.
- Evaluar opciones para realizar la actualización en producción.
- Listar elementos a corregir en etapas posteriores.

La siguiente etapa fue de **Inspección de la base de datos**. Problemas y advertencias de compatibilidad entre versiones mayores de MySQL se obtuvieron de la primera fase de investigación. El objetivo principal de esta fase fue el corregir los impedimentos a actualizar la base de datos a MySQL 8 y documentar las advertencias obtenidas de la primera fase.

Finalizado ese trabajo, continuó la etapa de **Inspección de aplicaciones**. Similar a la anterior etapa, durante la fase inicial de investigación se detectaron errores al ejecutar pruebas causadas por problemas de compatibilidad entre los sistemas legacy y la nueva versión de la base de datos. Como tal esta fase se encargaría de realizar los cambios en las aplicaciones para que la actualización se pudiese dar con normalidad.

Tras las tres etapas anteriores, siguió la **creación del plan de actualización**. Tras las correcciones tanto a nivel de base de datos como aplicaciones, se pudieron realizar pruebas utilizando las técnicas de blue/green de AWS. Se buscó obtener métricas de tiempo de actualización con información de producción, así como la creación de los planes específicos para actualizar la base de datos en producción, así como un plan de rollback en caso de errores.

Habiendo concluido la creación del plan, se procedió a una etapa de **modificación en la infraestructura de ambientes de desarrollo**. En esta fase la versión de MySQL y la

técnica para actualizar estaba definida, por lo que el siguiente paso era alinear los ambientes de desarrollo y locales para que utilicen la nueva base de datos.

Finalmente, la última etapa fue la de **ejecución y monitoreo**. Como el nombre lo indica, esta etapa se encargó de la ejecución del plan ya establecido y su posterior monitoreo.

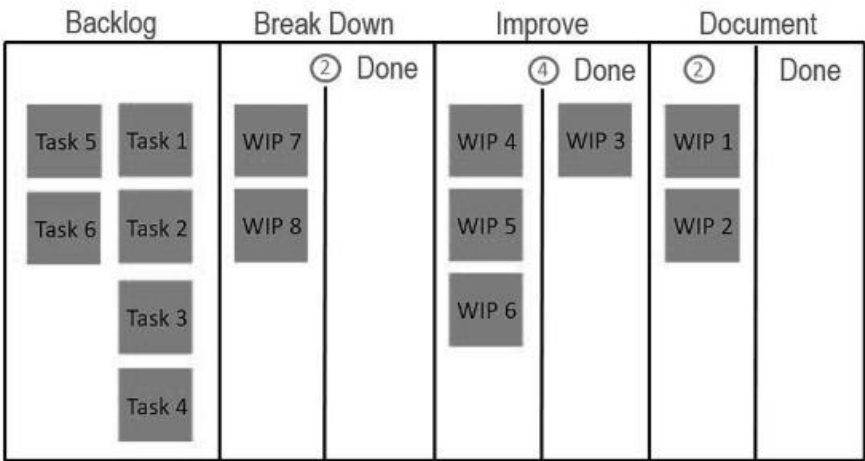
3.2.4 FUNDAMENTOS UTILIZADOS

Kanban

Weflen, MacKenzie y Rivero (2021) mencionan que Kanban es una de las metodologías nacidas de la evolución de las herramientas de gestión de proyectos iterativas y los principios ágiles. Las tareas, que generalmente son llamadas historias de usuario, alternan entre los diferentes estados de las tareas de izquierda a derecha, Kanban, permite a los equipos identificar este ciclo de trabajo y adaptarlo a sus necesidades.

Weflen et al. (2021) indican que la técnica más utilizada para la estimación de un proyecto que utilice Kanban o Scrum es una llamada *planning poker*. Técnica la cual asigna puntos a cada tarea, o historia de usuario lo cual representa el esfuerzo relativo del trabajo necesario para completar la tarea o historia de usuario.

Figura 7
Ejemplo de un tablero Kanban



Nota: Adaptado de “An influence diagram approach to automating lead time estimation in Agile Kanban project management” (p. 2), por Weflen et al, (2021). Expert Systems With Applications.

Ahmad, Dennehy, Conboy y Oivo (2017) mencionan que Kanban sigue los siguientes principios. Visualizar el flujo de trabajo, Kanban fomenta la visualización del trabajo mientras avanza sobre las distintas etapas mediante tableros. Limitar el trabajo en progreso, se deben de establecer límites a la cantidad de trabajo en progreso en cada etapa del proceso. Medir y gestionar el flujo, utilizando técnicas como los tableros Kanban o mapas de flujo de valor. Hacer explícitas las políticas de procesos, mediante la definición de condiciones de entrada y salida para cada fase. Y utilizar modelos para reconocer las oportunidades de mejora

Épicas

Rehkopf (s. f.-a) define a una épica como un conjunto de trabajo que puede desglosarse en tareas específicas, llamadas historias de usuario, basadas en los requerimientos de los clientes o usuarios finales. Las épicas, indica, son una práctica muy importante para los equipos ágiles y DevOps.

Las epopeyas son una forma útil de organizar el trabajo y crear una jerarquía. Se busca dividir el trabajo en partes que se puedan entregar, de esta forma proyectos pueden ejecutarse y se puede enviar valor a los clientes con regularidad. (Rehkopf, s. f.-a)

Historias de Usuario

Rehkopf (s. f.-b) define una historia de usuario como una explicación general e informal de una funcionalidad de software, escrita desde la perspectiva final del cliente. Su propósito es articular como una pieza de trabajo entregará un valor particular al cliente, se debe de tener en cuenta que estos clientes no necesariamente son clientes externos, estos pueden ser clientes internos o colegas de la organización que dependen de algún equipo.

Las historias encajan en marcos ágiles como SCRUM o Kanban. En el caso de SCRUM, estas historias se añaden a sprints y se van completando a lo largo de dicho sprint. En el caso de Kanban, los equipos incorporan historias de usuario a su backlog y se ejecutan a través de su flujo de trabajo. (Rehkopf, s. f.-b)

Tabla 4

Ejemplo de historia de usuario

Identificador	PLAY-1
Título	El título de la historia
Estimación	(1,2,3,5,8)
Descripción	Descripción de la historia
Criterios de Aceptación	Definición de completado para la historia

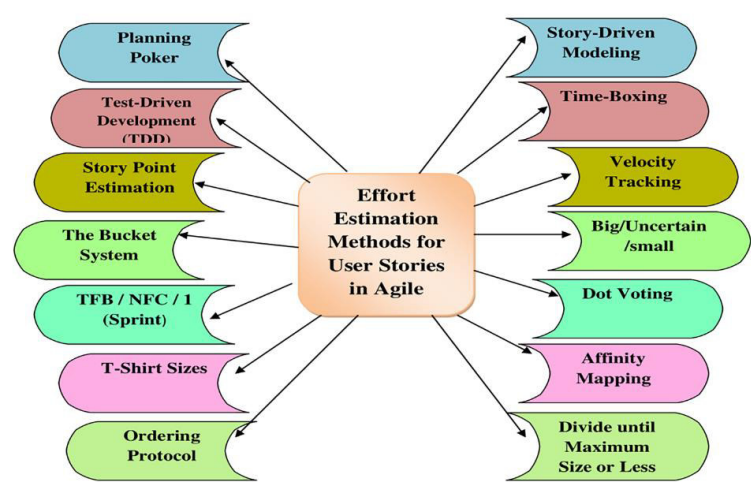
Nota: Elaboración propia.

Planning Poker

Iqbal et al. (2024) menciona en su estudio de los métodos de estimación de esfuerzo para historias de usuario que el planning poker es un método de estimación que involucra colaboración. Indicaron que es una técnica utilizada frecuentemente, y que esta popularidad se debe a que fomenta la integración de los equipos y ayuda establecer consensos.

Figura 8

Métodos de estimación de historias de usuario



Nota: Adaptado de “Exploring issues of story-based effort estimation in Agile Software Development (ASD)” (p. 4), por Iqbal et al, (2024). Science Of Computer Programming.

Según Zhang (2017) al utilizar planning poker las opiniones de los expertos, analogías y la desagregación se combinan en un enfoque agradable para estimar las historias de usuarios siendo beneficioso para obtener resultados de estimación más eficaces. Para realizar planning poker se pueden seguir los siguientes pasos:

1. El equipo discute el trabajo involucrado en la historia de usuario, generando las preguntas necesarias al product owner.
2. Cuando la historia ha sido discutida, cada miembro del equipo escribe su estimación en una nota.
3. Tras ello, las cartas son reveladas a todo el equipo en simultáneo.
4. Si las estimaciones difieren en demasía, los miembros del equipo discuten los porqués de sus estimaciones. Tras aquello se estima nuevamente.
5. Este proceso se repite hasta obtener un consenso.

Sistemas Legacy

Hasan et al. (2023) mencionan que un sistema legacy es aquel que soporta un proceso crítico del negocio con la restricción de utilizar software, hardware y recursos obsoletos y complicados de mantener. Dichos sistemas tienden a seguir una arquitectura monolítica que no considera la modularidad como parte fundamental de su diseño. Estos sistemas tienen a tener una gran complejidad tras varios años de uso derivando en que las desventajas de este tipo de arquitectura sobrepasen a sus beneficios.

Irani et al. (2022) indican que la definición de un sistema legacy puede ser dividido en dos perspectivas. Si se utiliza la perspectiva de los sistemas de información, se define a un sistema legacy como aquel que se resiste a cualquier tipo de cambios y modificaciones. En cambio, si se utiliza la perspectiva de las ciencias de la comunicación, la definición de un sistema legacy va ligada a la gobernanza de una organización, bajo este precepto, un sistema puede utilizar tecnologías novedosas, pero puede seguir siendo considerado legacy si no mejora los procesos de negocio del sistema original.

La mayoría de las disciplinas definen a estos sistemas legacy como sistemas obsoletos que siguen en utilización porque su data no puede ser cambiada a otro formato o sus aplicaciones no pueden ser actualizadas.

.Net Framework

Microsoft (s.f) define a .NET Framework como un framework de desarrollo de software que permite la creación y ejecución de aplicaciones en Windows. .NET Framework es la implementación original de .NET. Soporta la ejecución de sitios web, servicios y aplicaciones de escritorio en Windows.

Los dos componentes mayores de .NET Framework son la librería de clases (CL por sus siglas en inglés) y el motor de ejecución de lengua común (CLR por sus siglas en inglés). Este motor permite a aplicaciones escritas en C#, F# y Visual Basic ejecutarse en el mismo entorno. (Microsoft, s.f)

Microsoft (s.f) recomienda utilizar .NET en vez de .NET Framework, dado que .NET soporta más tipos de aplicaciones, plataformas y entrega un nivel alto de rendimiento.

IIS

SolarWinds (s.f) indica que Internet Information Services (IIS) es un servidor web de Microsoft que se ejecuta en el sistema operativo Windows y es utilizado para el intercambio de contenido web estático y dinámico con usuarios de internet. IIS puede ser utilizado para desplegar y gestionar aplicaciones web utilizando tecnologías como ASP.NET y PHP.

Algunos beneficios de utilizar IIS son. Seguridad robusta, creación de pools de aplicaciones e infraestructura web escalable y fiable. (SolarWinds, s.f)

Drupal

Drupal (s.f) indica que su software homónimo Drupal es uno de gestión de contenidos, utilizado para crear numerosos sitios web y aplicaciones utilizadas a diario. Entre sus características tiene; la fácil creación de contenidos, un rendimiento fiable y una excelente seguridad. Tiene como uno de sus principios básicos la modularidad y posee herramientas que ayudan a crear contenido versátil y estructurado que es requerido por las experiencias web modernas.

El proyecto Drupal es un software de código abierto, descargable por cualquiera para ser utilizado, trabajado y compartido. Este software es distribuido bajo los términos de la licencia publica general de GNU (GPL), bajo esta modalidad Drupal siempre será gratuito. (Drupal, s.f)

Terraform

Hashicorp (s.f) define a Terraform como una herramienta de infraestructura como código que permite definir recursos cloud y on-premise en archivos de configuración que pueden ser versionadas, reutilizadas y compartidas. Se puede utilizar para gestionar toda la infraestructura de manera consistente, desde componentes de nivel bajo (como almacenamiento y recursos de red) hasta elementos de nivel alto (como entradas DNS).

Terraform consiste en tres etapas (Hashicorp, s.f)

- Write. Donde se definen los recursos en sus archivos de configuración.
- Plan. Donde se genera un plan de ejecución de la infraestructura a modificar.
- Apply. Donde, tras ser aprobado el plan, se generarán los cambios establecidos.

Bases de Datos

Jatana (2012) menciona que una base de datos es una colección de grandes volúmenes de datos. Un sistema gestor de bases de datos es una forma organizada de gestionar, obtener y almacenar información. Cuando los ítems de data son organizados con la ayuda de tablas entonces la base de datos es considerada de tipo relacional. En cambio, cuando estas tablas tradicionales no son utilizadas entonces la base de datos será no relacional.

Jatana (2012) enumera los siguientes beneficios de las bases de datos relacionales:

- La mayoría de la información es almacenada en la base de datos y no en la aplicación, de esta forma la base de datos se documenta a si misma.
- Agregar, actualizar o eliminar data es sencillo.
- Brinda beneficios para resumir y obtener data, a su vez que ayuda a la generación de reportes.
- Cambios en el esquema de la base de datos son simples.

Jatana (2012) también enumera algunas de las desventajas de este tipo de base de datos:

- No soportan alta escalabilidad, se puede utilizar mejoras de hardware hasta cierto punto, luego se necesitará distribuir la base de datos.
- El uso de tablas puede dar lugar a modelos complejos en caso de que la data no pueda ser encapsulada eficientemente en esta estructura de datos.
- Muchas de las funcionalidades de las bases de datos no son utilizadas completamente. Por lo que se añaden complejidad y costo a estas herramientas.
- Las bases de datos relacionales usan SQL el cual es un lenguaje adaptado para trabajar con data estructurada. Sin embargo, si la data no es estructurada, puede generar complejidades adicionales.
- Cuando la cantidad de datos es gigantesca, la base de datos necesitará ser particionada, esto es un problema dado que es complejo hacer cruces entre tablas entre distintos servidores no es una tarea trivial.

MySQL

MySQL (s.f-a) indica que MySQL es uno de los sistemas de gestión de bases de datos Open Source más populares. Esta herramienta es apoyada por Oracle Corporation. Algunas características son las siguientes:

- MySQL es un sistema de gestión de base de datos. Una base de datos es una colección de data estructurada. Para añadir, acceder y procesar esta data se necesita un sistema de gestión de base de datos como MySQL.
- MySQL es una base de datos relacional. Una base de dato relacional almacena información en tablas separadas.
- MySQL es OpenSource. Cualquiera puede utilizar y modificar el software.
- MySQL es rápida, fiable, escalable y fácil de usar.

Amazon Web Services (AWS)

AWS (s.f-b) indica que Amazon Web Services es la nube más complete y adoptada del mundo, con un catálogo de mas de 200 servicios completos provistos desde centros de datos en todo el mundo. Poseyendo millones de clientes entre los que se incluyen las

startups de mayor crecimiento, las empresas más grandes y las principales agencias gubernamentales, dichos clientes utilizan AWS para reducir costes, ser más ágiles e innovar más rápido.

Según AWS (s.f-b) al ser el proveedor cloud líder, poseen las siguientes características.

- Mayor cantidad de funcionalidades/servicios.
- La comunidad más grande de clientes y socios.
- La más segura.
- Posee la innovación más rápida.
- Experiencia operativa más probada

Plataforma como servicios (PaaS)

Mušić et al. (2024) menciona los que las empresas proveedoras de servicios en la nube ofrecen a sus clientes la gestión de la infraestructura, plataforma o software como servicios, basándose en esta división, Existen tres categorías de entre las cuales está plataforma como servicio.

Infraestructura como servicio (IaaS por sus siglas en inglés) ofrece la gestión de una infraestructura virtualizada que comprende desde la computación hasta el almacenamiento y la gestión de las redes. La ventaja de este modelo es su flexibilidad, sin embargo, los usuarios deben de poseer conocimiento técnico para gestionar estos servicios de manera eficiente. (Mušić et al., 2024)

Plataforma como Servicios (PaaS por sus siglas en inglés) garantiza que la infraestructura necesaria sea aprovisionada en función de los requisitos del software y aplicaciones a desplegar. Servicios como el monitoreo y la gestión de los logs generalmente están incluidos en este tipo de servicio. El coste de un PaaS tiende a ser más elevado que el de un IaaS dado que la gestión de la infraestructura es abstraída. (Mušić et al., 2024)

Finalmente, Software como servicio (SaaS por sus siglas en inglés) es un modelo en el que el software es proporcionado a los usuarios finales listo para su uso. La infraestructura y la plataforma son gestionadas por el proveedor del servicio. (Mušić et al., 2024)

Amazon RDS

AWS (s.f-c) define a Amazon Relational Database Service (Amazon RDS) como un servicio web que facilita la configuración, el funcionamiento y el escalado de una base de datos relacional en la nube de AWS. Proporciona capacidad rentable y re dimensionable para una base de datos relacional estándar del sector y administra tareas comunes de administración de bases de datos.

Según AWS (s.f-c) algunas de las ventajas de Amazon RDS es que, al ser un servicio de base de datos administrado, se encarga de gestionar la mayoría de las tareas relacionadas a la infraestructura. Con respecto a bases de datos no gestionadas completamente estas son las ventajas:

- Se pueden utilizar motores de base de datos como IBM Db2, MariaDB, Microsoft SQL, MySQL, Oracle Database y PostgreSQL.
- Amazon RDS gestiona las copias de seguridad, parches de software y detección automática de fallos y su recuperación.
- Se puede habilitar las copias de seguridad (llamadas también **snapshots**) automáticas, o se pueden crear copias manuales. Se pueden utilizar estas copias de seguridad para restaurar bases de datos.
- Se puede obtener alta disponibilidad con una instancia de base de datos primaria y una instancia de base de datos secundaria síncrona a la que se puede conectar en caso de errores en el nodo principal.
- Se puede controlar el acceso a la base de datos usando el servicio de Amazon IAM para definir usuarios y permisos.

Grupos de parámetros.

Según AWS (s.f-d) un grupo de parámetros actúa como un contenedor para la configuración del motor de la base de datos que puede aplicar a más de una instancia de base de datos. Al crear una instancia de base de datos, esta usará un grupo de parámetros por defecto. Cada uno de estos grupos de parámetros contiene la configuración por defecto del motor elegido para la instancia de la base de datos. En caso la configuración necesite ser modificada, se requiere crear un nuevo grupo de parámetros ya que el grupo de parámetro por defecto es inmutable.

Despliegues blue/green en RDS

AWS (s.f-e) indica que al utilizar despliegues Amazon RDS Blue/Green se pueden crear y probar cambios en las bases de datos antes de implementar dichos cambios en los ambientes de producción. Este tipo de despliegues crea un entorno de ensayo el cual copia la topología del ambiente de producción elegido. El ambiente Blue es el ambiente de producción, mientras que el ambiente Green es el entorno de ensayo

Según AWS (s.f-e) se pueden realizar cambios en las instancias RDS del ambiente Green sin afectar producción. Ejemplos de este tipo de operaciones pueden ser actualizaciones menores o mayores de las versiones de las bases de datos, cambios de configuración en los grupos de parámetros, entre otros. Tras la realización de las pruebas necesarias, se realiza el proceso de **switchover** en el cual las bases de datos en el ambiente Green se trasladan al ambiente Blue con un tiempo muerte menor a 1 minuto y sin pérdidas de información ni necesidad de cambios en las aplicaciones que consuman la base de datos.

AWS (s.f-e) menciona los siguientes beneficios al utilizar este tipo de despliegues:

- Permite la fácil creación de un entorno de ensayo listo para producción.
- Cambios en el entorno de producción son replicados automáticamente al entorno de ensayo.
- Permite probar los cambios en las bases de datos en los ambientes de ensayo sin afectar producción.
- Mantiene al día los parches de la base de datos y las actualizaciones de los sistemas.
- Permite realizar el proceso de switchover sin afectar las aplicaciones que consumen la base de datos.
- Elimina la pérdida de datos durante el switchover.
- Permite realizar el switchover en un minuto dependiendo de la carga de trabajo en la base de datos.

Replicación de base de datos

Kelly (2024) define a la replicación de base de datos como el proceso de creación de copias de base de datos y su almacenamiento en varios destinos. El proceso de replicación se ejecuta 24/7, de esta forma, al cambiar la fuente de datos original, los cambios son propagados a las distintas copias, asegurando que los usuarios tengan acceso a la información más precisa y actualizada. Existen dos tipos de replicación:

- Active/Active. En la cual ambas bases de datos intercambian copias de manera síncrona. Ideal para el equilibrio de carga y la alta disponibilidad.
- Read Only. En la cual la base de datos primaria genera los cambios, y las copias solo se sincronizan. Este enfoque es útil para reportería.

AWS (s.f-g) menciona que AWS RDS para MySQL usa replicación síncrona. Lo que implica que bajo algunas circunstancias la replica no pueden mantenerse al día con la fuente original de datos. El resultado de este proceso es la aparición de **lag de replicación**. El valor de este lag puede ser evaluado con la columna `SECONDS_BEHIND_MASTER` obtenida del comando `SHOW SLAVE STATUS`.

Tipos de pruebas en software

Atlassian (s.f) distingue dos grandes grupos de tipos de prueba; manual y automática. En el caso de las pruebas manuales el proceso es hecho por las personas, donde se interactúa manualmente con la aplicación mediante una interfaz gráfica o mediante APIs. En el caso de las pruebas automáticas, estas son ejecutadas por máquinas que ejecutan scripts de pruebas creados con anterioridad. Este tipo de pruebas varían en complejidad, desde validar un método simple hasta la automatización de interacciones en la interfaz de usuario.

Atlassian (s.f) enumera los tipos de pruebas como los siguientes:

- Pruebas unitarias. Estas pruebas son de nivel bajo y bastante cercano al código fuente del aplicativo. Consisten en probar métodos individuales de clases, procesos o módulos. Estas pruebas son las más rápidas de ejecutar.
- Pruebas de integración. En este tipo de pruebas se validan que diferentes módulos o servicios usados por la aplicación trabajan de manera correcta entre ellos.

- Pruebas funcionales. En el caso de las pruebas funcionales, se enfocan más en probar los requerimientos funcionales del negocio, generalmente prueban los resultados mas no el cómo se llegaron a ellos.
- Pruebas end to end. End to End replica el comportamiento del usuario final mediante el uso de herramientas que permitan realizar este trabajo. Este tipo de pruebas son bastante complejos de ejecutar.
- Pruebas de aceptación. Son pruebas formales que verifican si el software satisface las necesidades del negocio.
- Pruebas de rendimiento. Evalúan al sistema bajo una carga de trabajo específica, permiten la validación de la escalabilidad, velocidad y fiabilidad de los sistemas.
- Pruebas de humo. Son un sub conjunto de pruebas que validan funcionalidad básica del sistema, la idea es que sean rápidas de ejecutar y que permitan validar la estabilidad del sistema.

Siapno (2023) añade que las pruebas de estrés son un subconjunto importante de las pruebas de rendimiento. Durante estas pruebas el sistema se lleva más allá de sus condiciones normales de funcionamiento para evaluar la estabilidad y su tolerancia a fallos. Esta clase de pruebas permite asegurar que el sistema puede soportar las condiciones más exigentes en los ambientes de producción.

3.2.5 IMPLEMENTACIÓN DE LAS ÁREAS, PROCESOS Y BUENAS PRÁCTICAS

La definición del proyecto a implementar estuvo dada por la siguiente épica:

Tabla 5

Épica PLAY-1

Identificador	PLAY-1
Tipo	Épica
Título	Actualizar la base de datos del reproductor/catálogo de MySQL 5.7 a MySQL 8.0
Descripción	El fin de vida de la versión comunitaria de MySQL 5.7 es octubre 2023. Amazon indica que es necesario realizar actualizar la plataforma para prevenir posibles pérdidas de soporte sobre la base de datos.
	La cantidad de trabajo al definir esta épica aún es incierta. Algunas líneas generales son:
Criterios de Aceptación	• Actualizar una base de datos en el ambiente de desarrollo. • Completar un ciclo de pruebas para asegurar que la actualización no genera errores en los sistemas. • Hay que asegurar que el catálogo y reproductor continúan trabajando tras la actualización. • Se debe de asegurar que la replicación entre las bases de datos continúe. • Se debe de asegurar que los desarrollos en los ambientes locales y de desarrollo continúen con normalidad. • Idealmente, se debería realizar esta actualización sin ningún tiempo muerto. De no ser esto posible, la actualización se debe realizar en la ventana de mantenimiento y con el menor tiempo posible,

Nota: Elaboración propia

3.2.5.1 Etapa de Investigación Inicial

Para determinar las líneas iniciales del trabajo se creó una historia de usuario para definir la técnica a utilizar para actualizar la base de datos en los ambientes y generar las historias de usuario iniciales.

Tras el juego de planeación, esta fue la definición obtenida:

Tabla 6

Historia de Usuario PLAY-34

Identificador	PLAY-34
Título	Investigar y planificar la épica de la actualización de la base de datos
Estimación	8
Descripción	Se requiere una investigación inicial sobre los cambios necesarios para actualizar la base de datos MySQL 5.7.42 a 8.0. Se ha hecho una actualización manual en ambientes de desarrollo sin mayores inconvenientes, pero esta actualización no se ha hecho con el volumen de información de producción.
Aspectos que tener en consideración:	1. Se necesita que la base de datos se actualice en los ambientes de desarrollo y que la suite de pruebas se ejecute contra la nueva base de datos. 2. Investigar la actualización de las bases de datos en las máquinas virtuales utilizadas en el desarrollo local. 3. Investigar el tiempo muerto que se requerirá para la actualización.
Criterios de Aceptación	• Ejecutar una actualización con data de producción y documentar resultados. • Documentar opciones para realizar la actualización y la versión de MySQL a la que actualizar. • Ejecutar una actualización manual en el ambiente dev y ejecutar una suite de pruebas contra él. Historias han sido

creadas para arreglar errores obtenidos por las ejecuciones anteriores.

Nota: Elaboración propia

Figura 9

Mapa de tareas de PLAY-34



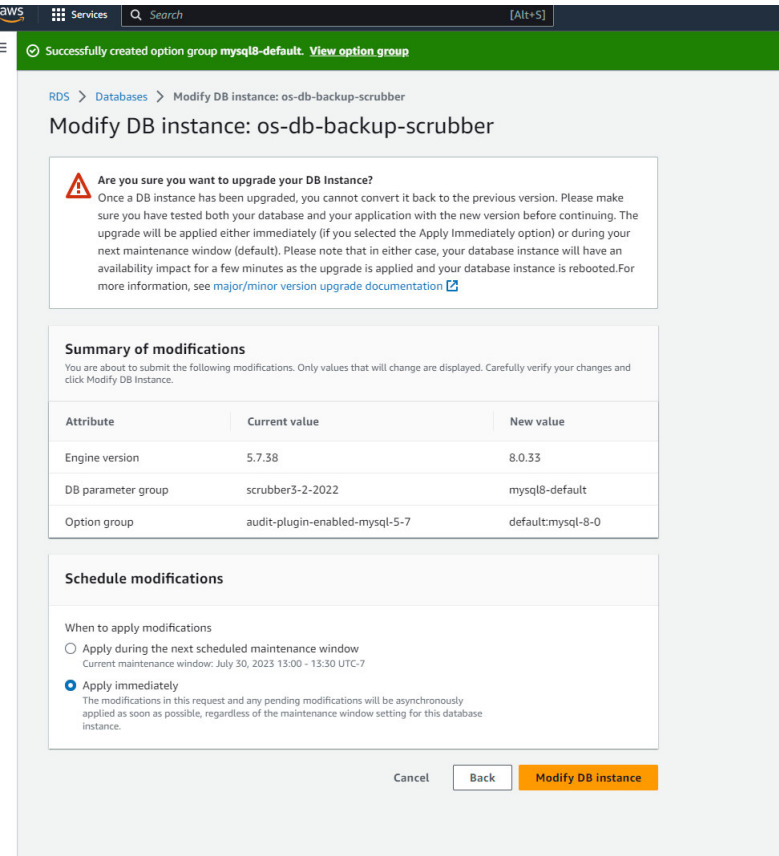
Nota: Mapa de tareas elaborado tras el juego de planeación

Primero, uno de los ingenieros había realizado previamente una actualización manual de la base de datos en un ambiente de desarrollo. Esta base de datos contiene un conjunto de datos menor al de producción y solo posee los esquemas del catálogo y reproductor, mientras que en producción se tienen esquemas adicionales y mucha más información. Por lo que se requirió realizar una de estas actualizaciones con el volumen y estructura idéntica a producción.

Para ello intentó realizar una actualización sobre una base de datos nueva con la información restaurada de un snapshot de producción. El proceso se puede apreciar en la siguiente imagen.

Figura 10

Actualización manual de una base de datos



Nota: Ejemplo obtenido del cliente.

La ejecución de este proceso falló dado a problemas de compatibilidad para la actualización entre la versión 5.7 y 8.0.33, la versión de MySQL 8 elegida en este momento fue aleatoria.

AWS genera un reporte en estos casos, en dicho reporte se toman 26 puntos, de estos 26 puntos solo se mostrarán los errores:

Tabla 7

Reporte de compatibilidad previo a la actualización

Elementos evaluados	Mensajes
1. Uso del tipo `temporal`	No se encontraron problemas.
2. Uso de objetos con nombres que poseen palabras reservadas.	Los siguientes objetos tienen nombres que entran en conflicto con las nuevas palabras clave reservadas. Asegúrese de que las consultas enviadas por sus aplicaciones utilicen `comillas` al referirse a ellos o producirán errores. • catalog.system - Table name • catalog.simpletest.function - Column name • catalog.votingapi_cache.function - Column name
3. Uso de la codificación utf8mb3	Los siguientes objetos utilizan el juego de caracteres utf8mb3. Se recomienda convertirlos a utf8mb4 para mejorar la compatibilidad con Unicode. ... una lista enorme de objetos siguió a continuación, esto es porque utf8mb3 es la codificación por defecto de la base de datos original
8. Utilización de flags sql_mode obsoletos	Aviso: Los siguientes objetos de base de datos tienen opciones obsoletas para sql_mode, que se eliminarán durante la actualización a 8.0. • catalog.SoftDeletePromoSeats • catalog.urlencode • catalog.WaitAndUpdateAmendmentExpiration

18. Problemas notificados Aquí hubo numerosos errores durante la ejecución del por el comando 'check table x proceso, algunos ejemplos son:
for upgrade '

catalog.Chartio_courses_first_published - The user specified as a definer ('production'@'<CIDR>') does not exist

catalog.Chartio_courses_first_published - Corrupt

24. La creación de índices de Tiene una o más tablas con formato de fila redundante.
más de 767 bytes en tablas La creación de un índice de más de 767 bytes puede
con formato de fila provocar la inaccesibilidad de la tabla.
redundante puede provocar la
inaccesibilidad de las tablas. ... prosiguió una lista de tablas

Nota: La información descrita es un resumen, el reporte original poseía 26 puntos de los cuales solo se han expuesto aquellos que retornaron problemas.

Los problemas anteriormente descritos se convertirían en nuevas historias. Adicionalmente intentar realizar la actualización, se tuvo que generar un nuevo grupo de parámetros para la MySQL 8, esta se generó con valores por efecto, pero se necesitaba crear uno que fuera equivalente al que se usaba actualmente en producción. Estas fueron las nuevas historias generadas:

- PLAY-72: Crear un grupo de parámetros para MySQL 8 que preserve las configuraciones del grupo actual para la base de datos 5.7.42.
- PLAY-73: Convertir o ignorar la advertencia del utf8mb3 charset.
- PLAY-74: Investigar los objetos con flags SQL obsoletos y los que usan palabras reservadas.
- PLAY-75: Investigar y corregir problemas de corrupción obtenidos por la herramienta de actualización
- PLAY-76: Investigar el problema de índices con un tamaño mayor a 767 bytes que podrían causar que las tablas se volviesen inaccesibles.

En paralelo, se realizaron las investigaciones para detectar posibles opciones para realizar la actualización. En general existieron tres opciones documentadas en la siguiente tabla:

Tabla 8

Resumen de alternativas para actualizar la base de datos

Método	Discusión
Realizar la actualización de manera manual en línea.	A favor: • La más barata al poder ser ejecutada con un simple comando y no necesitar nuevos servidores. En contra: • El tiempo de ejecución es largo, la base de datos con data de pruebas demoró 30 minutos en actualizar. Lo cual lleva a pensar que producción podría demorar mucho más. • No es posible realizar un rollback. • Se deben de actualizar todas las bases de datos de manera manual. • El ruteo de DNS que se usa en la base de datos se debe de reconfigurar manualmente.
Crear una nueva réplica de la base de datos, actualizar y promover a máster.	A favor: • El equipo tiene experiencia creando nuevas réplicas de base de datos. En contra: • La promoción de la base de datos a máster genera un tiempo de desconexión donde la información se puede perder.

	• Se debe recrear la topología del cluster de manera manual. • El ruteo de DNS que se usa en la base de datos se debe de reconfigurar manualmente.
Utilizar el servicio blue/green de AWS:	A favor: • Este método crea una topología idéntica de manera automática. • Ambos clústeres son sincronizados sin necesidad de mayor configuración. • Durante el switch over, ambas bases de datos se ponen en modo read y no hay posibilidad de pérdida de información. En contra: • Al crear una réplica exacta del clúster de base de datos, es con diferencia la solución más costosa. • No existe mecanismo de rollback directamente soportado por AWS. De necesitarse el rollback debía ser manual.

Nota: Elaboración propia.

Se consultó la información anterior con los interesados y se decidió ir con la opción del servicio blue/green de AWS. Las razones fueron las siguientes:

1. Era la opción con la menor cantidad de trabajo manual posible.
2. La actualización podría ser casi inmediata dado que ambos clústeres viven en simultaneidad hasta que se decide hacer el switchover.
3. Se puede usar el nuevo clúster para operaciones de lectura a modo de prueba.

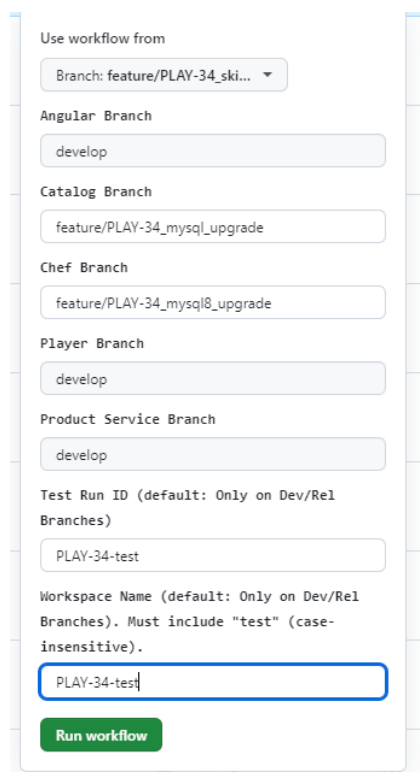
4. La sincronización previa al switchover mantiene la integridad de los datos.

Adicionalmente, para la versión de MySQL a utilizar, el contratista especializado en base de datos indicó que en su experiencia. Era más seguro utilizar la versión más cercana a la que se iba a actualizar. Por ende, se eligió MySQL 8.0.32.

Para el punto final en la historia actual, se requirió crear un ambiente de pruebas con la nueva base de datos y validar el resultado. Esto se logró mediante un pipeline que se usa para generar los ambientes de desarrollo:

Figura 11

Creación de un ambiente de pruebas



The image shows a configuration form for a workflow. It includes several sections with labels and input fields:

- Use workflow from:** A dropdown menu showing "Branch: feature/PLAY-34_ski...".
- Angular Branch:** A text input field containing "develop".
- Catalog Branch:** A text input field containing "feature/PLAY-34_mysql_upgrade".
- Chef Branch:** A text input field containing "feature/PLAY-34_mysql8_upgrade".
- Player Branch:** A text input field containing "develop".
- Product Service Branch:** A text input field containing "develop".
- Test Run ID (default: Only on Dev/Rel Branches):** A text input field containing "PLAY-34-test".
- Workspace Name (default: Only on Dev/Rel Branches). Must include "test" (case-insensitive):** A text input field containing "PLAY-34-test", which is highlighted with a blue border.
- Run workflow:** A green button at the bottom.

Nota: Elaboración propia

La ejecución de las pruebas falló por lo que se generó una nueva historia para realizar la investigación a detalle en la etapa de inspección de aplicaciones:

- PLAY-82: Investigar fallo en el comando `os-create-course` tras la actualización de la base de datos en el ambiente de desarrollo.

Con esto la primera etapa culminó con las siguientes conclusiones.

Primero, existen problemas de compatibilidad de la data de producción al intentar actualizar la base de datos. Se necesita realizar las investigaciones y correcciones respectivas.

Segundo, el fallo en la ejecución del comando indica que hay errores a nivel de aplicaciones que deben de ser corregidos previo intento de actualización.

Finalmente, la técnica propuesta a utilizar será la blue/green de AWS y la versión objetivo MySQL 8.0.32.

3.2.5.2 Etapa de Inspección de la base de datos

Esta etapa estuvo caracterizada por un análisis de los problemas obtenidos durante la etapa anterior obtenidos del reporte de actualización.

Tabla 9

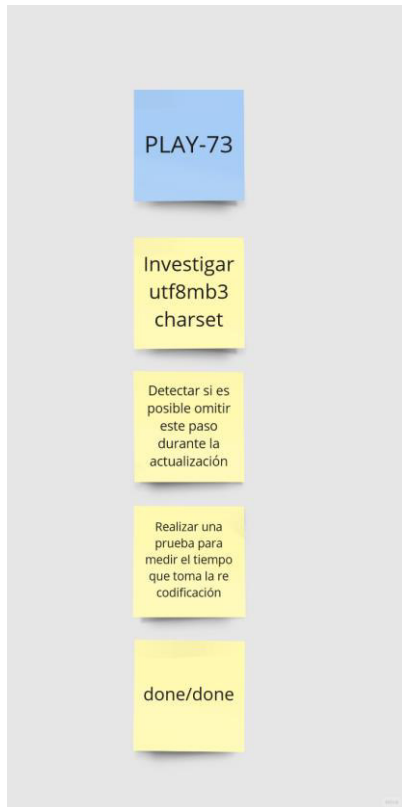
Historia de Usuario PLAY-73

Identificador	PLAY-73
Título	Convertir o ignorar la advertencia del utf8mb3 charset.
Estimación	2
Descripción	Tras el reporte de fallos en la actualización con datos de producción. Se requiere comprender las implicaciones de esta advertencia que indica la recomendación del uso de utf8mb4 y de ser posible definir si es necesario realizar estos cambios.
Criterios de Aceptación	• Se posee una mejor comprensión de la advertencia anteriormente mencionada.

Nota: Elaboración propia.

Figura 12

Mapa de tareas de PLAY-73



Nota: Elaboración propia.

MySQL (s.f-b) indica que utf8 ha sido usado como un alias para la codificación utf8mb3, y su uso está deprecado. Siendo preferible el uso de utf8mb4.

En el caso de la base de datos a tratar, esta codificada enteramente en utf8, lo cual generó las advertencias por la herramienta de actualización. Se realizaron las consultas con el experto en base de datos y nos indicó que el proceso de actualizar la codificación de la base de datos podía ser complicado y tomar bastante tiempo dado el volumen de datos.

La documentación también indica que, en casos de replicación, ambos la fuente y replica deben de tener la misma codificación para evitar posibles efectos secundarios. (MySQL, s.f-d)

Figura 13

Documentación MySQL

The following items summarize other potential incompatibilities:

- `SET NAMES 'utf8mb4'` causes use of the 4-byte character set for connection character sets. As long as no 4-byte characters are sent from the server, there should be no problems. Otherwise, applications that expect to receive a maximum of three bytes per character may have problems. Conversely, applications that expect to send 4-byte characters must ensure that the server understands them.
- For replication, if character sets that support supplementary characters are to be used on the source, all replicas must understand them as well.

Also, keep in mind the general principle that if a table has different definitions on the source and replica, this can lead to unexpected results. For example, the differences in maximum index key length make it risky to use `utf8mb3` on the source and `utf8mb4` on the replica.

If you have converted to `utf8mb4`, `utf16`, `utf16le`, or `utf32`, and then decide to convert back to `utf8mb3` or `ucs2` (for example, to downgrade to an older version of MySQL), these considerations apply:

Nota: Adaptado de “Converting Between 3-Byte and 4-Byte Unicode Character Sets”

MySQL, s.f-d

MySQL (s.f-c) también nos indica en su documentación que si bien `utf8mb3` esta deprecado, su uso se mantendrá con soporte por todas las versiones 8.0 de MySQL.

A pesar de ello se decidió hacer una pequeña prueba con información de producción para validar el tiempo que le tomaría a cada tabla la actualización, los resultados fueron los siguientes:

Tabla 10

Tiempo de ejecución para actualizar la codificación a utf8mb4

Tabla	Tiempo
field_revision_launch_specs	112629s
field_data_launch_specs	56129s
field_revision_package_id	43693s
License	38391s
field_revision_package_friendly_name	29027s

Nota: Se muestran las 5 tablas con mayor tiempo de ejecución, el listado original posee 62 tablas.

Tras la ejecución, la tabla que toma más tiempo para ejecutar la actualización es **field_revision_launch_specs** con 31.2 horas

Acorde a la documentación, necesitamos realizar esta misma actualización en cada réplica. Es por estas razones que se determinó que estas actualizaciones sean omitidas y tocadas en otra épica.

La siguiente historia que tratar fue PLAY-74.

Tabla 11

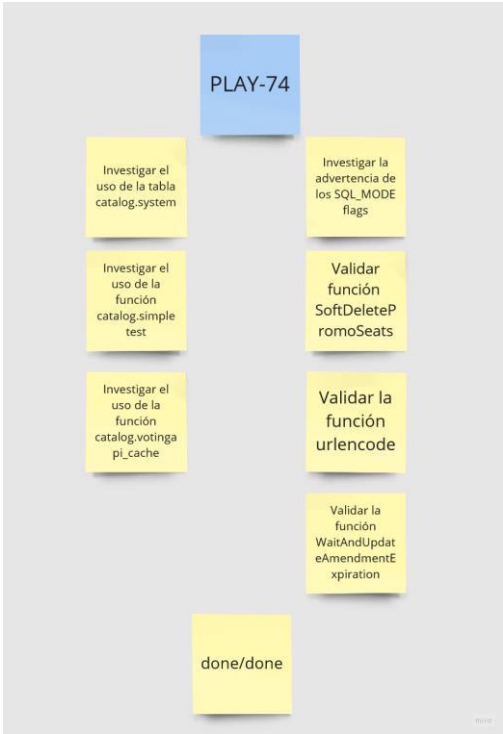
Historia de Usuario PLAY-74

Identificador	PLAY-74
Título	Investigar los objetos con flags SQL obsoletos y los que usan palabras reservadas
Estimación	3
Descripción	Tras el reporte de fallos en la actualización con datos de producción. Se requiere comprender las implicaciones de esta advertencia que indica la recomendación del uso de utf8mb4 y de ser posible definir si es necesario realizar estos cambios.
Criterios de Aceptación	Se posee una mejor comprensión de la advertencia anteriormente mencionada.

Nota: Elaboración propia.

Figura 14

Mapa de tareas de PLAY-74



Nota: Elaboración propia

Lo primero a investigar fueron los objetos con palabras reservadas. El reporte indicó que mientras dichos objetos se referencien usando el símbolo `` entonces no existe inconveniente. Los objetos problemáticos son los siguientes:

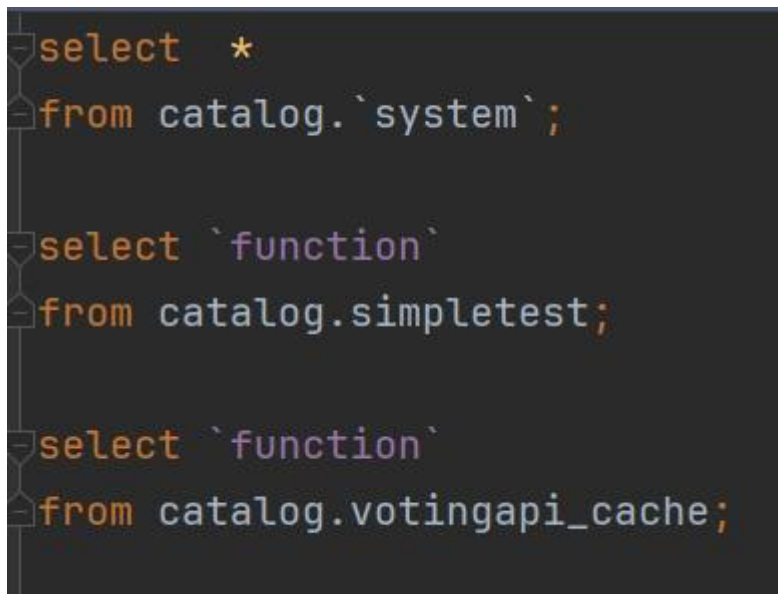
- `catalog.system` usa la palabra reservada *system* en la definición de su tabla.
- `catalog.simpletest` es una tabla con un campo llamado *function*, este nombre es reservado en MySQL 8.
- `catalog.votingapi\_cache` también es una tabla con un campo llamado *function*.

Se realizaron dos validaciones por cada uno de estos casos.

Primero, se confirmó la validez de lo que mencionó el reporte, siendo que el uso del símbolo `` prevenía errores al ejecutar consultas en la base de datos. Para ello se realizaron un conjunto de consultas en la base de datos actualizada y se pudo confirmar que todas se ejecutaron de manera correcta. También validamos que efectivamente las consultas fallaban si no se usaba el símbolo indicado.

Figura 15

Consultas realizadas



```
select *  
from catalog.`system`;  
  
select `function`  
from catalog.simpletest;  
  
select `function`  
from catalog.votingapi_cache;
```

Nota: Elaboración propia.

Lo segundo a evaluar fueron las referencias en el código. Los repositorios están almacenados en GitHub. Por ende, utilizando su motor de búsqueda pudimos encontrar referencias a estos objetos. Lo encontrado fue lo siguiente:

1. Los tres objetos son accedidos siempre mediante el uso del símbolo ‘`’ de manera interna por Drupal.
2. La única referencia directa en código es a la tabla `catalog.system`. Esta referencia se da en un script y el mismo usa ‘`’.
3. No existe ninguna otra referencia a estos objetos fuera del repositorio del catálogo.

Tras aquello se pudo determinar que no había necesidad de renombrar estos objetos de la base de datos. En especial la tabla `catalog.system` es accedida por Drupal para realizar el control de sus módulos, por lo que un cambio de nombre podría generar problemas en el sistema.

En lo que respecta a la advertencia de los SQL_MODE flags. La documentación indica que el uso de dicha variable es la forma en la cual se puede indicar a MySQL la manera en la que va a operar, esta configuración se puede dar a nivel global o a nivel de sesión. (MySQL, s.f-e).

En el caso específico de los tres objetos a analizar. Todos fueron creados utilizando una sesión que añadió el valor `NO\_AUTO\_CREATE\_USER`. Estos valores se mantienen en la definición del objeto creado como se puede apreciar en la siguiente imagen:

Figura 16

Función urlencode

```

1
2 ✓ show create function catalog.urlencode;
3

ces
Output Result 2 x
Function: urlencode, sql_mode: REAL_AS_FLOAT,PIPES_AS_CONCAT,ANSI_QUOTES,I... CREATE DEFINER="

```

Nota: Elaboración propia.

En el pasado, este valor se usaba para prevenir la creación de nuevos usuarios usando la operación GRANT, esto era necesario para prevenir problemas durante la replicación.

(MySQL, s.f-e) En MySQL 8.0 dicho problema ya no existe y por ende esta opción fue deprecada. (MySQL, s.f-e).

Se realizaron pruebas y efectivamente tras una actualización de MySQL 5.7 a 8 se validó que los esos valores fueron removidos de las funciones y que el removerlos no generó mayores inconvenientes.

Durante el proceso de validación notamos otro error que no había sido detectado inicialmente. Uno de las tres funciones afectadas por SQL_MODE, específicamente `catalog.SoftDeletePromoSeats` poseía en su definición un alias con la palabra *buffer* la cual también es reservada.

Figura 17

Error en la definición de SoftDeletePromoSeats

```
UPDATE UserSeat us
JOIN seat_buffer as buffer on buffer.seat_id = us.seat_id
WHERE buffer.id is not null;
```

Nota: La imagen es una representación del error original.

Para solucionar esto se realizó un cambio en la base de datos simplemente modificando la palabra *buffer* por un alias genérico que no poseía palabras reservadas. Este cambio se envió al repositorio general y fue desplegado en producción.

Tabla 12

Historia de usuario PLAY-75

Identificador	PLAY-75
Título	Investigar y corregir problemas de corrupción obtenidos por la herramienta de actualización
Estimación	3
Descripción	La herramienta de pre-actualización detectó numerosos errores en recursos de la base de datos los cuales no pudo analizar. Entre ellos están los esquemas: • metrics • reporting

- customer_service_dashboard

Y vistas como:

- catalog.Chartio_courses_first_published
- catalog.Chartio_line_item_detail
- catalog.Chartio_line_item_detail_escrow
- Otras vistas

Criterios de	• Se corrige los errores de corrupción.
Aceptación	• Se elimina objetos de la base de datos no utilizados.

Nota: Elaboración propia. La cantidad de vistas en la definición anterior es reducida.

Figura 18

Mapa de tareas de PLAY-75



Nota: Elaboración propia

La herramienta de actualización generó numerosos errores similares a los que se ven en la siguiente imagen:

Figura 19

Errores de corrupción

```
{
  "level": "Notice",
  "dbObject": "catalog.chartio_categories_without_description",
  "description": "Access denied for user 'root'@'%' (using password: YES)"
},
{
  "level": "Error",
  "dbObject": "catalog.chartio_categories_without_description",
  "description": "Corrupt"
},
```

Nota: Elaboración propia.

Una investigación de los objetos afectados detectó que dichos objetos fueron generados por un usuario de la base de datos que había sido eliminado y no eran accesibles por ningún otro usuario, incluyendo el usuario root. Tras una conversación con ingenieros que trabajaban en la compañía se indicó que estos objetos probablemente eran residuales de antiguos sistemas que se usaron en la organización.

Por ende, se decidió investigar estos elementos y decidir si debían ser eliminados.

En el caso de los tres esquemas; *customer_service_dashboard*, *metrics* y *reporting*. Nos indicaron que estos no eran elementos reconocidos y que se evalué si era posible removerlos.

De manera similar a PLAY-74, se escaneo todos los repositorios y no se encontraron referencias en código a estos objetos. Sin embargo, por motivos de seguridad se decidió realizar esta eliminación en dos fases:

1. Primero se renombrarían los objetos a nuevos esquemas.
2. Se esperaría por dos días, de no encontrar ningún inconveniente con otros sistemas se eliminarían los objetos.

La única excepción fue el esquema `reporting` dado que se conocía que este esquema no era utilizado en absoluto.

Figura 20

Operaciones para eliminar esquemas no usados

```
create schema tbd_metrics character set latin1 collate latin1_swedish_ci;
create schema tbd_customer_service_dashboard character set latin1 collate latin1_swedish_ci;

RENAME TABLE customer_service_dashboard.helpscout TO tbd_customer_service_dashboard.helpscout;
RENAME TABLE customer_service_dashboard.olark TO tbd_customer_service_dashboard.olark;
RENAME TABLE metrics.accounts TO tbd_metrics.accounts;
RENAME TABLE metrics.average_rating TO tbd_metrics.average_rating;
RENAME TABLE metrics.completions TO tbd_metrics.completions;
RENAME TABLE metrics.courses_count TO tbd_metrics.courses_count;
RENAME TABLE metrics.courses_rated_count TO tbd_metrics.courses_rated_count;
RENAME TABLE metrics.courses_sale_count TO tbd_metrics.courses_sale_count;
RENAME TABLE metrics.data_helpscout_conversation_tags TO tbd_metrics.data_helpscout_conversation_tags;
RENAME TABLE metrics.data_helpscout_conversations TO tbd_metrics.data_helpscout_conversations;
RENAME TABLE metrics.data_helpscout_people TO tbd_metrics.data_helpscout_people;
RENAME TABLE metrics.data_helpscout_thread_attachments TO tbd_metrics.data_helpscout_thread_attachments;
RENAME TABLE metrics.data_helpscout_threads TO tbd_metrics.data_helpscout_threads;
RENAME TABLE metrics.data_olark_chats TO tbd_metrics.data_olark_chats;
RENAME TABLE metrics.data_olark_items TO tbd_metrics.data_olark_items;
RENAME TABLE metrics.data_olark_operators TO tbd_metrics.data_olark_operators;
RENAME TABLE metrics.data_olark_visitors TO tbd_metrics.data_olark_visitors;
RENAME TABLE metrics.db_table_size TO tbd_metrics.db_table_size;
RENAME TABLE metrics.line_items TO tbd_metrics.line_items;
RENAME TABLE metrics.metrics_daily_snapshot TO tbd_metrics.metrics_daily_snapshot;
RENAME TABLE metrics.metrics_helpscout_conversations TO tbd_metrics.metrics_helpscout_conversations;
RENAME TABLE metrics.metrics_olark_chats TO tbd_metrics.metrics_olark_chats;
RENAME TABLE metrics.olark_report TO tbd_metrics.olark_report;
RENAME TABLE metrics.opportunities TO tbd_metrics.opportunities;
RENAME TABLE metrics.orders TO tbd_metrics.orders;
RENAME TABLE metrics.pipeline_values TO tbd_metrics.pipeline_values;
RENAME TABLE metrics.plus_launches TO tbd_metrics.plus_launches;
RENAME TABLE metrics.plus_launches_rollup TO tbd_metrics.plus_launches_rollup;
RENAME TABLE metrics.plus_revenue_churned TO tbd_metrics.plus_revenue_churned;
RENAME TABLE metrics.plus_revenue_churned_revision TO tbd_metrics.plus_revenue_churned_revision;
RENAME TABLE metrics.plus_user_seats TO tbd_metrics.plus_user_seats;
RENAME TABLE metrics.plus_user_seats_churned TO tbd_metrics.plus_user_seats_churned;
RENAME TABLE metrics.plus_user_seats_churned_revision TO tbd_metrics.plus_user_seats_churned_revision;
RENAME TABLE metrics.plus_user_seats_revision TO tbd_metrics.plus_user_seats_revision;
RENAME TABLE metrics.pluslaunches TO tbd_metrics.pluslaunches;
RENAME TABLE metrics.plususerseats TO tbd_metrics.plususerseats;
RENAME TABLE metrics.seller_count TO tbd_metrics.seller_count;
RENAME TABLE metrics.seller_sale_count TO tbd_metrics.seller_sale_count;
RENAME TABLE metrics.users TO tbd_metrics.users;

drop view reporting.max_seats;
drop view reporting.actual_users;
drop view reporting.recommender_courses_used;
drop view reporting.recommender_ratings;

drop schema metrics;
drop schema reporting;
drop schema customer_service_dashboard;
```

Nota: Elaboración propia.

Para las vistas del catálogo se realizó un proceso similar pero dado que el esquema del catálogo es usado en el día a día se decidió evaluar una por una las vistas afectadas para validar si se podían eliminar. Adicionalmente se decidió realizar una pequeña auditoria y también evaluar el uso de vistas que no estaban en el reporte de compatibilidad.

Esto se ve representado en la siguiente tabla. Por cada elemento se buscó si existen referencias en código y si se podía acceder a la vista usando un usuario de tipo administrador.

Tabla 13

Vistas en el catálogo a eliminar

Nombre	Es Referenciado	Problemas de acceso
12_12_license_amendments	NO	SI
Chartio_courses_first_published	SI	SI
Chartio_line_item_detail	SI	SI
Chartio_line_item_detail_edit	SI	SI
Chartio_line_item_detail_escrow	SI	SI
Chartio_line_item_detail_escrow_new	NO	SI
Chartio_marketplace_sales_by_day	SI	SI
Chartio_orders	SI	SI
Chartio_orders_by_day	SI	SI
active_license_amendments	NO	SI
active_user_seats	NO	SI
chartio_categories_without_description	SI	SI
chartio_courses_without_description	SI	SI
chartio_courses_without_os_description	SI	SI
chartio_list_of_top_level	SI	SI
chartio_os_plus_all_launches	SI	NO
chartio_os_plus_base	SI	NO
chartio_os_plus_unique_launches_by_month	SI	NO
chartio_os_plus_unique_registrations	SI	NO
chartio_top_level_comment_mapping	SI	SI
chartio_top_level_commerce_mapping	SI	SI
chartio_top_level_subject_course_mapping	SI	SI
chartio_top_level_subject_mapping	SI	SI
chartio_users	SI	NO
dev_integration	NO	SI
dev_sync	NO	SI
dev_sync_courses	NO	SI
dev_tenant	NO	SI
dev_tenant_users	NO	SI
metrics_line_item_detail	NO	NO
os_plus_license_amendments	NO	SI
rpt_max_seats_non_ppu	NO	NO
rpt_max_seats_ppu	NO	NO
seller_buyer_dates	NO	SI
seller_buyer_totals_12_month	NO	SI
seller_buyer_totals_6_month	NO	SI
site_license_amendments	NO	SI

Nota: Elaboración propia, esta lista contiene los elementos a eliminar. No se muestran las vistas que se mantuvieron.

Las vistas en la tabla anterior fueron eliminadas dado que no poseían referencias y algunas de ellas, las que no se podían acceder, impedían la ejecución de la herramienta de actualización de base de datos.

Figura 21

Eliminación de vistas no usadas en el catálogo

```
drop view catalog.12_12_license_amendments
drop view catalog.Chartio_courses_first_published
drop view catalog.Chartio_line_item_detail
drop view catalog.Chartio_line_item_detail_edit
drop view catalog.Chartio_line_item_detail_escrow
drop view catalog.Chartio_line_item_detail_escrow_new
drop view catalog.Chartio_marketplace_sales_by_day
drop view catalog.Chartio_orders
drop view catalog.Chartio_orders_by_day
drop view catalog.active_license_amendments
drop view catalog.active_user_seats
drop view catalog.chartio_categories_without_description
drop view catalog.chartio_courses_without_description
drop view catalog.chartio_courses_without_os_description
drop view catalog.chartio_os_plus_all_launches
drop view catalog.chartio_os_plus_base
drop view catalog.chartio_os_plus_unique_launches_by_month
drop view catalog.chartio_os_plus_unique_registrations
drop view catalog.chartio_top_level_comment_mapping
drop view catalog.chartio_top_level_commerce_mapping
drop view catalog.chartio_top_level_subject_course_mapping
drop view catalog.chartio_top_level_subject_mapping
drop view catalog.chartio_users
drop view catalog.dev_integration
drop view catalog.dev_sync
drop view catalog.dev_sync_courses
drop view catalog.dev_tenant
drop view catalog.dev_tenant_users
drop view catalog.metrics_line_item_detail
drop view catalog.os_plus_license_amendments
drop view catalog.rpt_max_seats_non_ppu
drop view catalog.rpt_max_seats_ppu
drop view catalog.seller_buyer_dates
drop view catalog.seller_buyer_totals_12_month
drop view catalog.seller_buyer_totals_6_month
drop view catalog.site_license_amendments
```

Nota: Elaboración propia.

Tabla 14

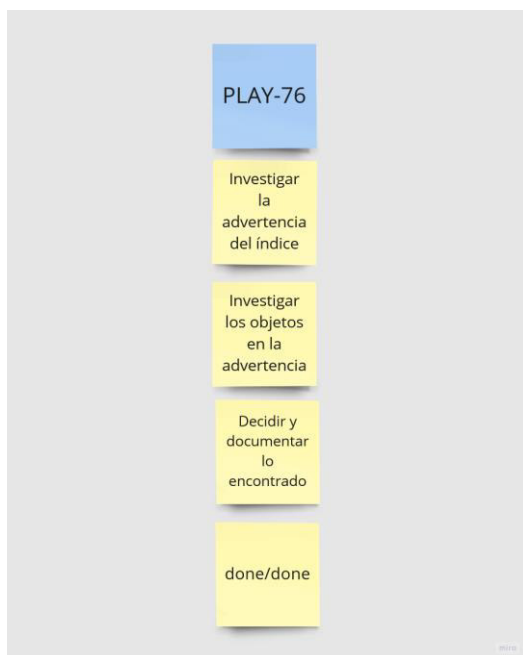
Historia de Usuario PLAY-76

Identificador	PLAY-76
Título	Investigar el problema de índices con un tamaño mayor a 767 bytes que podrían causar que las tablas se volvieran inaccesibles.
Estimación	3
Descripción	La herramienta de pre-actualización detectó informo sobre un posible problema con algunos objetos de la base de datos. Informando de que es posible que las tablas queden inaccesibles.
	Se referencia el siguiente incidente: https://bugs.mysql.com/bug.php?id=99791
Criterios de Aceptación	• Comprender la advertencia. • Decidir si es necesario realizar cambios sobre el esquema

Nota: Elaboración propia.

Figura 22

Mapa de tareas de PLAY-76



Nota: Elaboración propia.

El error es descrito por Reilly en reporte de un bug en los foros de MySQL. Tablas creadas en versiones previas a MySQL 8 que usen el formato de fila COMPRESSED o REDUNDANT y que este valor no es explícitamente indicado durante la creación, pueden crear índices sin un límite de tamaño, lo que puede generar valores mayores a 767 bytes, el cual es el límite para dichos índices; esto puede causar que la tabla se vuelva inaccesible. (Reilly, 2020)

Reilly (2020) indica que para que la tabla se vuelva inaccesible se necesita añadir un índice tras actualizar a MySQL 8, este índice sería creado sin un límite de tamaño lo cual causaría el bloqueo de la tabla.

Realizando una indagación sobre la sintaxis de creación de índices. MySQL (s.f-f) indica que para columnas de tipo cadena de caracteres se puede usar un prefijo con una longitud predeterminada.

Figura 23

Sintaxis de creación de índices

```
CREATE [UNIQUE | FULLTEXT | SPATIAL] INDEX index_name
[index_type]
ON tbl_name (key_part,...)
[index_option]
[algorithm_option | lock_option] ...

key_part: {col_name [(length)] | (expr)} [ASC | DESC]

index_option: {
    KEY_BLOCK_SIZE [=] value
  | index_type
  | WITH PARSER parser_name
  | COMMENT 'string'
  | {VISIBLE | INVISIBLE}
  | ENGINE_ATTRIBUTE [=] 'string'
  | SECONDARY_ENGINE_ATTRIBUTE [=] 'string'
}

index_type:
    USING {BTREE | HASH}

algorithm_option:
    ALGORITHM [=] {DEFAULT | INPLACE | COPY}

lock_option:
    LOCK [=] {DEFAULT | NONE | SHARED | EXCLUSIVE}
```

Nota: Adaptado de “Creating Index Statement”. MySQL, s.f-f.

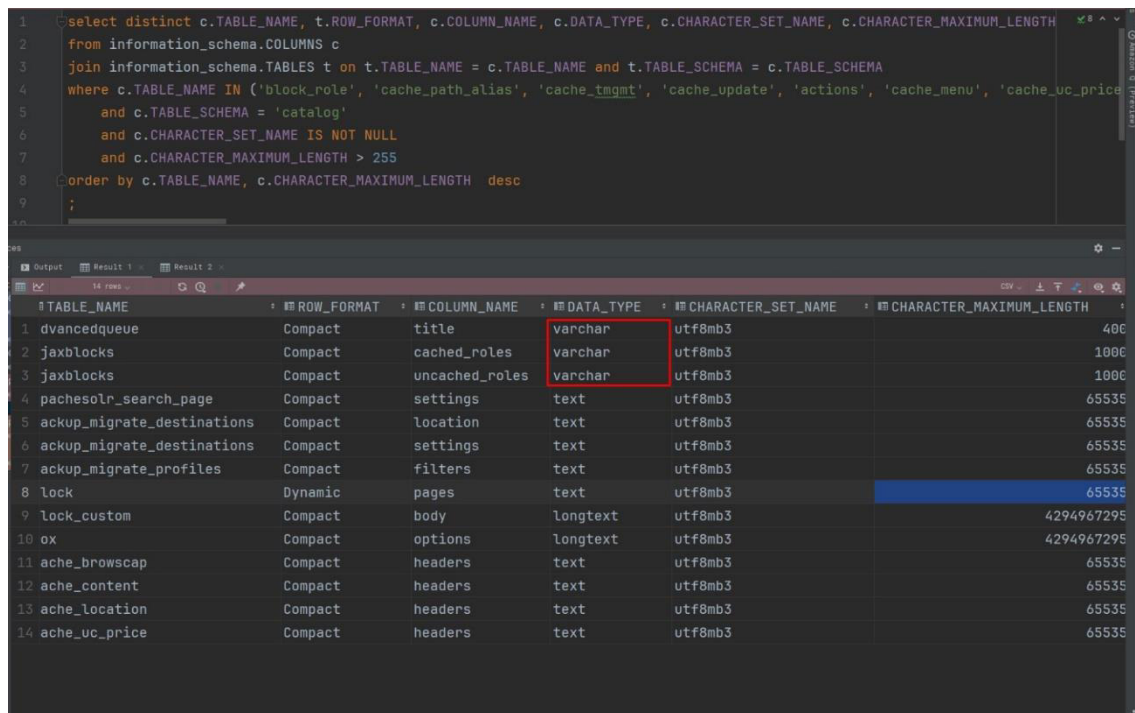
Esta sintaxis es la que previene la creación de índices mayores a 767 bytes. Si este valor no es especificado el índice tomara el valor total de la columna. En el ejemplo de Reilly uso columnas con la codificación utf8mb4 con un tamaño de 255 caracteres. Esto generaría un índice con tamaño de 1020 bytes excediendo el valor máximo del índice.

Con este conocimiento se realizó una consulta sobre la base de datos con los objetos afectados. El esquema INFORMATION_SCHEMA es útil para recolectar metadata sobre los objetos en la base de datos.

En la consulta se obtuvieron la longitud máxima de la columna, se filtró los valores menores a 255. Esto es porque la base de datos usa una codificación de utf8mb3, al dividir 767 entre 3 obtenemos aproximadamente 255, siendo este el límite de caracteres antes de que se genere el error.

Figura 24

Consulta sobre objetos en la base de datos.



```

1 select distinct c.TABLE_NAME, t.ROW_FORMAT, c.COLUMN_NAME, c.DATA_TYPE, c.CHARACTER_SET_NAME, c.CHARACTER_MAXIMUM_LENGTH
2 from information_schema.COLUMNS c
3 join information_schema.TABLES t on t.TABLE_NAME = c.TABLE_NAME and t.TABLE_SCHEMA = c.TABLE_SCHEMA
4 where c.TABLE_NAME IN ('block_role', 'cache_path_alias', 'cache_tngmt', 'cache_update', 'actions', 'cache_menu', 'cache_uc_price')
5 and c.TABLE_SCHEMA = 'catalog'
6 and c.CHARACTER_SET_NAME IS NOT NULL
7 and c.CHARACTER_MAXIMUM_LENGTH > 255
8 order by c.TABLE_NAME, c.CHARACTER_MAXIMUM_LENGTH desc
9 ;

```

TABLE_NAME	ROW_FORMAT	COLUMN_NAME	DATA_TYPE	CHARACTER_SET_NAME	CHARACTER_MAXIMUM_LENGTH
dvancedqueue	Compact	title	varchar	utf8mb3	400
jaxblocks	Compact	cached_roles	varchar	utf8mb3	1000
jaxblocks	Compact	uncached_roles	varchar	utf8mb3	1000
pachesolr_search_page	Compact	settings	text	utf8mb3	65535
ackup_migrate_destinations	Compact	location	text	utf8mb3	65535
ackup_migrate_destinations	Compact	settings	text	utf8mb3	65535
ackup_migrate_profiles	Compact	filters	text	utf8mb3	65535
lock	Dynamic	pages	text	utf8mb3	65535
lock_custom	Compact	body	longtext	utf8mb3	4294967295
ox	Compact	options	longtext	utf8mb3	4294967295
ache_browscap	Compact	headers	text	utf8mb3	65535
ache_content	Compact	headers	text	utf8mb3	65535
ache_location	Compact	headers	text	utf8mb3	65535
ache_uc_price	Compact	headers	text	utf8mb3	65535

Nota: Elaboración propia.

Tomando solo los objetos de tipo VARCHAR, terminamos con dos tablas.

- advancedqueue
- jaxblocks

La primera tabla es usada en un proceso interno del catálogo y la búsqueda se realiza sobre un identificador. La segunda tabla es una interna usada por Drupal.

Ninguna de las dos requiere la creación de índices por lo que se decidió ignorar esta advertencia dado que requeriría un cambio sobre elementos usados constantemente en el sistema.

Tabla 15

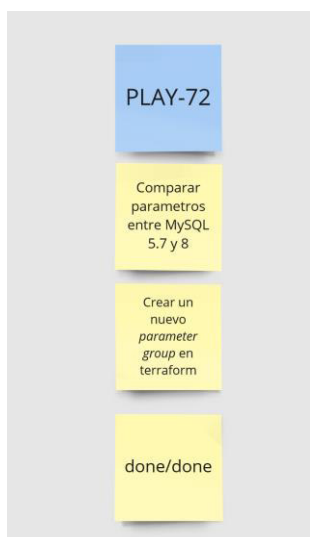
Historia de Usuario PLAY-72

Identificador	PLAY-72
Título	Crear un grupo de parámetros (<i>PG</i>) para MySQL 8 que preserve las configuraciones del grupo actual para la base de datos 5.7.42
Estimación	3
Descripción	En RDS se necesita utilizar un grupo de parámetros el cual contiene las configuraciones de la base de dato. Para las pruebas de desarrollo se generó una con valores por defecto, sin embargo, se necesita crear una que sea equivalente a la que se usa actualmente en producción y replicarlo para MySQL 8.
Criterios de Aceptación	Un nuevo PG se ha generado para las versiones 8 de MySQL.

Nota: Elaboración propia

Figura 25

Mapa de tareas PLAY-72



Nota: Elaboración propia

Para realizar la comparación, se utilizó las herramientas de AWS para obtener el PG de MySQL 5.7 y se comparó con el creado con valores por defecto. Esto genera un objeto de tipo JSON que luego fue comparado valor por valor.

El reporte final tuvo la siguiente forma:

Figura 26

Comparativa de parametros entre versiones

<pre>{ "Parameters": [{</pre>	<pre>= { "Parameters": [{</pre>
	<pre> "ParameterName": "activate_all_rol » es_on_login", "ParameterValue": "0", "Description": "Automatically set » all granted roles as active after the user h » as authenticated successfully.", "Source": "engine-default", "ApplyType": "dynamic", "DataType": "boolean", "AllowedValues": "0,1", "IsModifiable": true, "ApplyMethod": "pending-reboot" }, {</pre>
<pre> "ParameterName": "allow-suspicious » -udfs", "Description": "Controls whether u » ser-defined functions that have only an xxx » symbol for the main function can be loaded", "Source": "engine-default", "ApplyType": "static", "DataType": "boolean", "AllowedValues": "0,1", "IsModifiable": false, "ApplyMethod": "pending-reboot" },], }</pre>	<pre>= "ParameterName": "allow-suspicious » -udfs", "Description": "Controls whether u » ser-defined functions that have only an xxx » symbol for the main function can be loaded", "Source": "engine-default", "ApplyType": "static", "DataType": "boolean", "AllowedValues": "0,1", "IsModifiable": false, "ApplyMethod": "pending-reboot" },], }</pre>

Nota: La imagen contiene solo un ejemplo de la comparativa, la lista completa de valores comparados es mucho mayor.

De dicha comparación se encontraron 40 parámetros con valores no especificados o con valores distintos a los generados por defecto. Ejemplo de estos son:

- binlog_format
- character_set_connection
- default_password_lifetime
- collation_server
- read_buffer_size
- innodb_buffer_pool_size
- sql_mode
- max_allowed_packet
- slow_query_log
- sync_binlog

- otros

Tras obtener estos valores se añadió genero el nuevo PG usando infraestructura como código. El cambio fue bastante sencillo a implementar usando *terraform*, se definió el recurso y simplemente se proveyeron los valores requeridos de los parámetros mediante una variable.

Figura 27

Creación de los parámetros para MySQL 8

```
resource "aws_db_parameter_group" "mysql8-parameters" {
  description = "terraformed rds parameter group for mysql 8"
  family      = "mysql8.0"
  name        = "mysql8-parameters"

  dynamic "parameter" {
    for_each = var.parameters
    content {
      apply_method = parameter.value.apply_method
      name         = parameter.value.name
      value        = parameter.value.value
    }
  }
}
```

Nota: Elaboración propia

La etapa de inspección de la base de datos culminó con las correcciones sobre los objetos de la base de datos que prevenían la actualización. Así como la creación de un grupo de parámetros equivalente al que se usaba en el MySQL 5.7.

A su vez se ignoraron advertencias que no resultasen críticas para la realización de la actualización de la base de datos. Se documentaron estas decisiones y estos cambios serán evaluados en posteriores proyectos.

3.2.5.3 Etapa de Inspección de las aplicaciones

Tabla 16

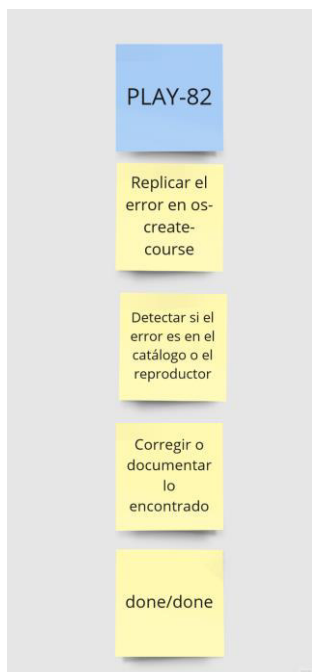
Historia de Usuario PLAY-82

Identificador	PLAY-82
Título	Investigar fallo en el comando `os-create-course` tras la actualización de la base de datos en el ambiente de desarrollo.
Estimación	3
Descripción	La ejecución de la suite de pruebas fracasó al intentar crear cursos para los diferentes escenarios de prueba tras actualizar una base de datos de pruebas a MySQL 8.0.32. Se necesita identificar porque este proceso falla.
Criterios de Aceptación	• Comprender el error • Solucionar de ser una corrección simple, de no ser este el caso crear una o más historias de usuario para realizar el trabajo.

Nota: Elaboración propia.

Figura 28

Mapa de tareas de PLAY-82



Nota: Elaboración propia.

Las pruebas se ejecutan en el reproductor y en el catálogo. En el caso de la función OS-CREATE-COURSE es un comando ejecutado directamente en el catálogo que se conecta directamente con el reproductor para crear un curso de determinado.

Replicar el error fue tan simple como ejecutar el comando manualmente. Tras aquello el equipo pudo notar que el error residía en el reproductor del cual se extrajo el siguiente log.

Figura 29

Error de conexión a la base de datos

```
"exception": {
  "error_id": "99394cd3",
  "type": "System.Data.EntityException",
  "message": "The underlying provider failed on open.",
  "stack_trace": "at System.Data.EntityClient.EntityConnection.OpenStoreConnectionIf(Boolean openCondition, DbConnection storeConnectionToOpen, DbConnection originalConnection, String exceptionCode, String attemptedOperation, Boolean& closeStoreConnectionOnFailure)\r\n at System.Data.EntityClient.EntityConnection.Open()\r\n at System.Data.Objects.ObjectContext.EnsureConnection()\r\n at System.Data.Objects.ObjectQuery`1.GetResults(Nullable`1 forMergeOption)\r\n at System.Data.Objects.ObjectQuery`1.System.Collections.Generic.IEnumerable`1.GetEnumerator()\r\n at System.Linq.Enumerable.FirstOrDefault[TSource](IEnumerable`1 source)\r\n at System.Data.Objects.Elinq.ObjectQueryProvider.<>c__11'1.<GetElementFunction>b__11_1(IEnumerable`1 sequence)\r\n at System.Data.Objects.Elinq.ObjectQueryProvider.ExecuteSingle[TResult](IEnumerable`1 query, Expression queryRoot)\r\n at System.Data.Objects.Elinq.ObjectQueryProvider.System.Linq.IQueryProvider.Execute[S](Expression expression)\r\n at System.Linq.Queryable.FirstOrDefault[TSource](IQueryable`1 source, Expression`1 predicate)\r\n at Entities.ScormPackage.Load(Guid courseGuid, Nullable`1 version) in C:\\Users\\circlecil\\project\\ScormPlayer\\Entities\\Custom\\ScormPackage.cs:line 33\r\n at Player.EngineDomain.EntityManager.LoadScormPackage(Guid courseGuid, Nullable`1 versionId, Boolean required) in C:\\Users\\circlecil\\project\\ScormPlayer\\Player\\EngineDomain\\EntityManager.cs:line 184\r\n at Player.ScormIntegration.ScormPackageHandler.ImportPackage(ICoursePackage package) in C:\\Users\\circlecil\\project\\ScormPlayer\\Player\\ScormIntegration\\ScormPackageHandler.cs:line 61\r\n at Player.Frontend.ImportVideoCourseConsumer.Process(ImportVideoCourse msg) in C:\\Users\\circlecil\\project\\ScormPlayer\\Player\\Frontend\\ImportVideoCourseConsumer.cs:line 19\r\n at ScormPlayer.Aws.Sqs.MessageBus.<>c__DisplayClass24_0'2.<Subscribe>b__0(TMessage message) in C:\\Users\\circlecil\\project\\ScormPlayer\\ScormPlayer.Aws\\Sqs\\MessageBus.cs:line 108\r\n at ScormPlayer.Aws.Sqs.MessageBus._ExecuteAction[T](String queue, IMessage`1 msg, Action`1 action, IMessageReceiver publisher) in C:\\Users\\circlecil\\project\\ScormPlayer\\ScormPlayer.Aws\\Sqs\\MessageBus.cs:line 274",
  "inner_exception": {
    "error_id": "af01554c",
    "type": "System.Security.Authentication.AuthenticationException",
    "message": "A call to SSPI failed, see inner exception.",
    "stack_trace": "at System.Net.Security.SslState.StartSendAuthResetSignal(ProtocolToken message, AsyncProtocolRequest asyncRequest, Exception exception)\r\n at System.Net.Security.SslState.StartSendBlob(Byte[] incoming, Int32 count, AsyncProtocolRequest asyncRequest, Boolean renegotiation)\r\n at System.Net.Security.SslState.ProcessReceivedBlob(Byte[] buffer, Int32 count, AsyncProtocolRequest asyncRequest)\r\n at System.Net.Security.SslState.StartReadFrame(Byte[] buffer, Int32 readBytes, AsyncProtocolRequest asyncRequest)\r\n at System.Net.Security.SslState.StartReceiveBlob(Byte[] buffer, AsyncProtocolRequest asyncRequest)\r\n at System.Net.Security.SslState.StartSendBlob(Byte[] incoming, Int32 count, AsyncProtocolRequest asyncRequest, Boolean renegotiation)\r\n at System.Net.Security.SslState.ForceAuthentication(Boolean receiveFirst, Byte[] buffer, AsyncProtocolRequest asyncRequest, Boolean renegotiation)\r\n at System.Net.Security.SslState.ProcessAuthentication(LazyAsyncResult lazyResult)\r\n at System.Net.Security.SslStream.AuthenticateAsClient(String targetHost, X509CertificateCollection clientCertificates, SslProtocols enabledSslProtocols, Boolean checkCertificateRevocation)\r\n at MySql.Data.MySqlClient.NativeDriver.StartSsl()\r\n at MySql.Data.MySqlClient.NativeDriver.Open()\r\n at MySql.Data.MySqlClient.Driver.Open()\r\n at MySql.Data.MySqlClient.Driver.Create(MySqlConnectionStringBuilder settings)\r\n at MySql.Data.MySqlClient.MySqlPool.GetPooledConnection()\r\n at MySql.Data.MySqlClient.MySqlPool.TryToGetDriver()\r\n at MySql.Data.MySqlClient.MySqlPool.GetConnection()\r\n at MySql.Data.MySqlClient.MySqlConnection.Open()\r\n at System.Data.EntityClient.EntityConnection.OpenStoreConnectionIf(Boolean openCondition, DbConnection storeConnectionToOpen, DbConnection originalConnection, String exceptionCode, String attemptedOperation, Boolean& closeStoreConnectionOnFailure)",
    "inner_exception": {
      "error_id": "6f972967",
      "type": "System.ComponentModel.Win32Exception",
      "message": "The function requested is not supported."
    }
  }
}
```

Nota: Elaboración propia.

Tras pruebas adicionales nos dimos cuenta de que el problema residía en el conector de MySQL que se estaba utilizando. Actualizar el conector de la versión 6.7 a 8.0.32 corregía el error.

Sin embargo, dicha versión generaba problemas de compatibilidad con la versión de Entity Framework usada en la aplicación. Actualizar dicha dependencia no era algo trivial por lo que se decidió crear una historia aparte para realizar este trabajo.

- PLAY-110: Actualizar versión de conector MySQL y Entity Framework

Tabla 17

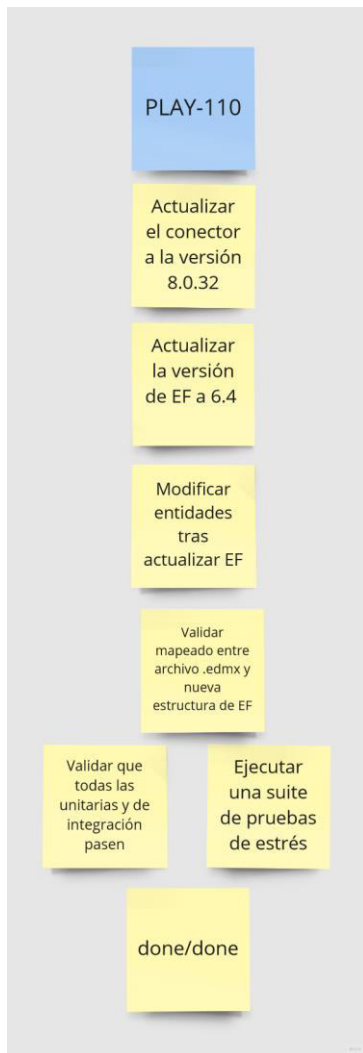
Historia de Usuario PLAY-110

Identificador	PLAY-110
Título	Actualizar el conector MySQL y la librería Entity Framework (EF) en el reproductor
Estimación	5
Descripción	Tras la resolución de PLAY-82, se detectó que el error estaba en el conector a MySQL usado en el reproductor. Se requiere actualizar el conector, sin embargo, este cambio también implica una actualización forzosa de la versión de Entity Framework
Criterios de Aceptación	• Actualizar EF a EF 6.4 • Actualizar MySQL Connector a la versión 8.0.32 • Actualizar MySQL Data a la versión 8.0.32 • La suite de pruebas debe de ejecutarse.

Nota: Elaboración propia

Figura 30

Mapa de tareas para PLAY-110



Nota: Elaboración propia

Realizar los cambios necesarios para actualizar el conector de MySQL requirió varias pruebas y muchas modificaciones al código.

Esto es porque al actualizar el conector, se tuvo que actualizar la versión de EF. Esto acarreó cambios en la forma en la cual se accedía a la base de datos. Los cambios realizados modificaron las siguientes áreas del código:

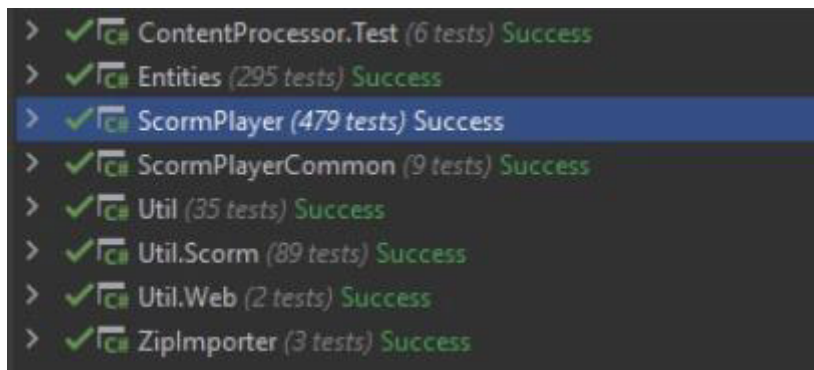
- MySQL data fue actualizado a 8.0.32
- EF fue actualizado a 6.4
- Todas las entidades de la aplicación fueron modificadas.
- Se cambió la forma de acceder a las entidades.

- Se agregaron métodos de tipo factoría a cada entidad.
- La herramienta de migraciones de base de datos fue modificada.

Tras todo aquello se ejecutaron las pruebas unitarias y de integración pertinentes dada la cantidad de cambios en el reproductor.

Figura 31

Pruebas ejecutadas



Nota: Elaboración propia

Tras ello se ejecutaron las pruebas de estrés para validar el comportamiento con tráfico más cercano a lo que se puede observar en producción. Aquí encontramos que varias de las pruebas fallaban tras la actualización, mientras que con la versión antigua del conector y EF dichos errores no ocurrían.

Figura 32

Errores en pruebas de estrés

✗	Record Results	default	POST	500	19	45 464ms	59 217ms	6 943ms
✓	Record Results	default	POST	200	78 620	549ms	10 245ms	8 737ms
✗	Record Results	default	POST	504	4	119 060ms	119 681ms	257ms
✗	Record Results Completion	default	POST	500	566	1 458ms	9 678ms	7 247ms

Nota: Elaboración propia

Se realizó una investigación y se detectó que el culpable era la forma en la que se accedía a una entidad en la base de datos tras la actualización.

Existió un bloqueo cuando este acceso se daba en simultáneo en la aplicación, por lo que la solución fue cambiar dicho método de acceso por otro. Una vez se realizó ese cambio, las pruebas de estrés finalizaron con las mismas características que su versión previa a la actualización.

Tras aquello no se logró detectar mayores inconvenientes con otras aplicaciones. El código del catálogo con Drupal 7 no tuvo ningún inconveniente en acceder a la nueva versión de la base de datos. Solo el código del reproductor tuvo que ser alterado para realizar ejecutar la actualización.

Con ello la etapa de Investigación de las aplicaciones concluyó, tras los cambios realizados, la base de datos y las aplicaciones estuvieron listas para actualizar, por lo que se procedió a la creación del plan en la siguiente etapa.

3.2.5.4 Etapa de creación del plan de actualización

Esta etapa se encargó de definir los detalles de la actualización en producción. Para ello se requirió realizar una copia de la base de datos de producción con las modificaciones hechas en las etapas anteriores con el objetivo de realizar métricas.

La definición de esto estuvo dada por la siguiente historia.

Tabla 18

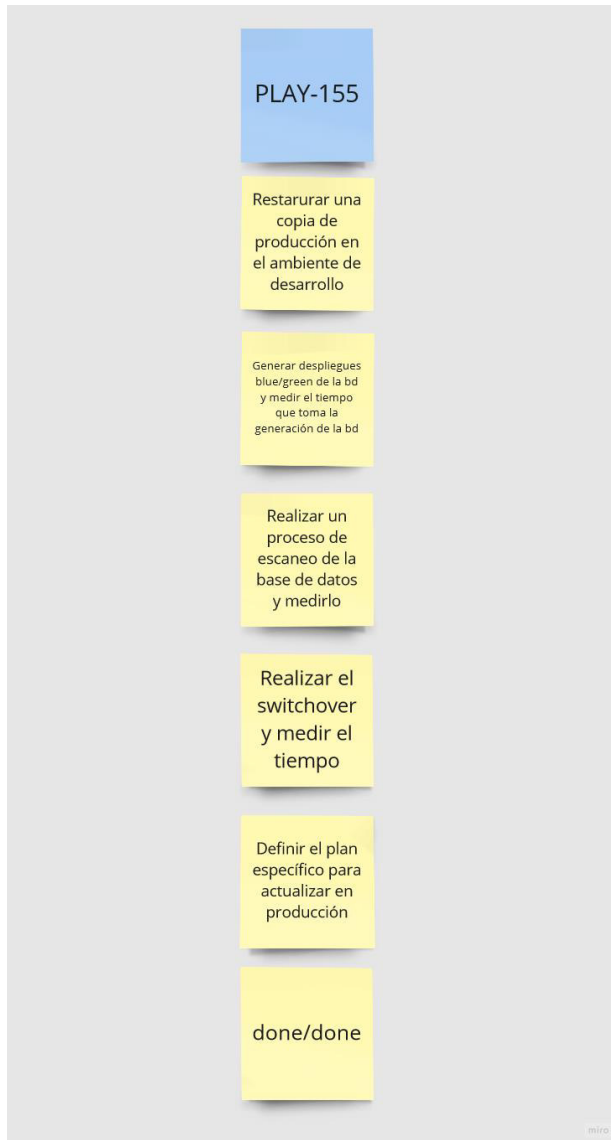
Historia de Usuario PLAY-155

Identificador	PLAY-155
Título	Ejecución de la actualización de la base de datos con prueba de producción para obtener mediciones y generar el plan de ejecución final.
Estimación	5
Descripción	Tras la información recabada y las modificaciones hechas sobre los esquemas de producción. Se necesita crear una prueba final que permita estimar el tiempo que tomará la ejecución del despliegue blue/green y definir el plan de actualización final.
Criterios de Aceptación	• Se ejecuta un despliegue blue/green y se realizan mediciones de tiempo. • El plan de actualización final es generado y discutido con los miembros del equipo.

Nota: Elaboración propia.

Figura 33

Mapa de tareas de PLAY-155



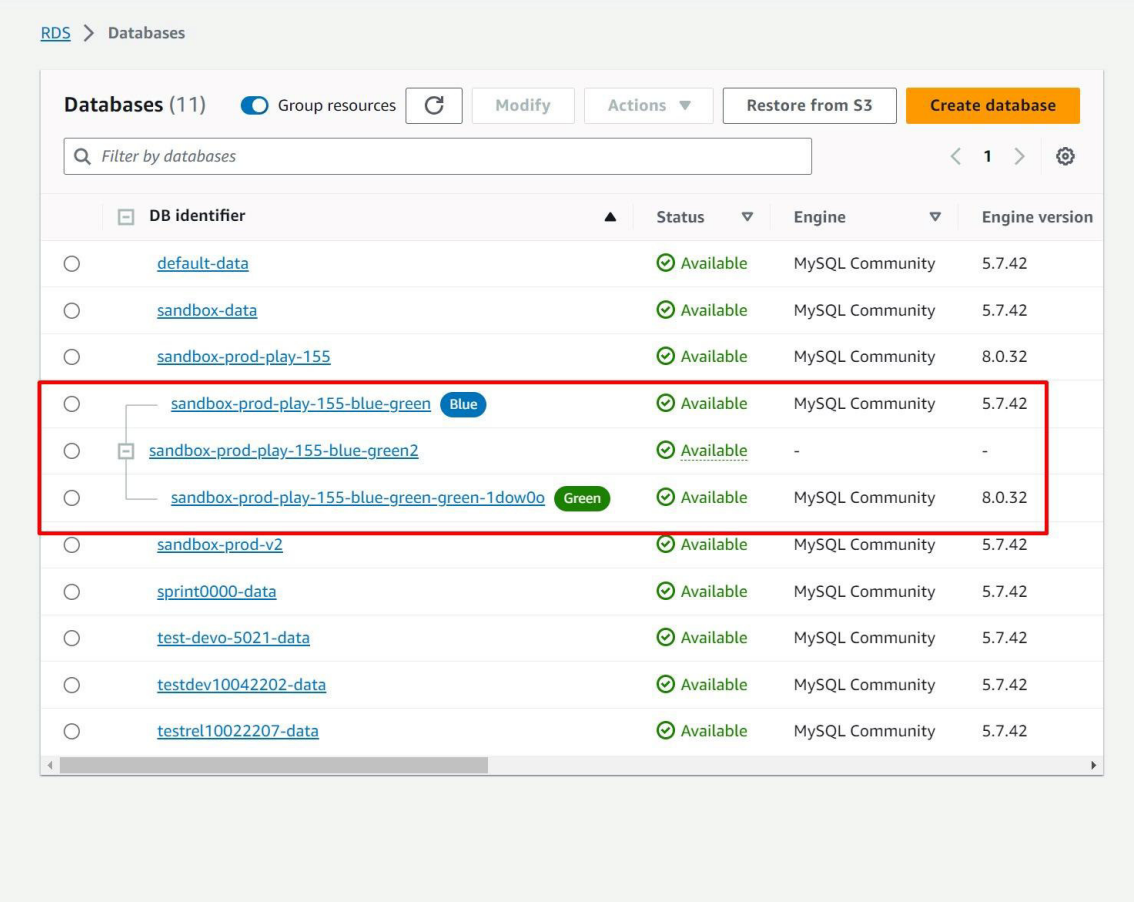
Nota: Elaboración propia

Tras los cambios realizados en la plataforma y las pruebas ejecutadas el equipo se sintió cómodo explorando un despliegue blue/green similar al que se realizaría en producción para obtener métricas sobre los diversos pasos a efectuar.

Esto es importante dado que la compañía posee acuerdos de servicio bastante estrictos, los cuales solo les permiten estar en modo mantenimiento por un periodo de máximo de tres horas. La empresa deseaba usar el menor tiempo posible para realizar la actualización y el resto para monitoreo.

Figura 34

blue/green en el ambiente de desarrollo



The screenshot shows the AWS RDS Databases console. At the top, there's a breadcrumb 'RDS > Databases'. Below it, a header bar contains 'Databases (11)', a 'Group resources' toggle, and buttons for 'Modify', 'Actions', 'Restore from S3', and 'Create database'. A search bar 'Filter by databases' is also present. The main table lists databases with columns: DB identifier, Status, Engine, and Engine version. A red box highlights three rows:

DB identifier	Status	Engine	Engine version
sandbox-prod-play-155-blue-green Blue	Available	MySQL Community	5.7.42
sandbox-prod-play-155-blue-green2	Available	-	-
sandbox-prod-play-155-blue-green-green-1dow0o Green	Available	MySQL Community	8.0.32

Nota: Elaboración propia.

Lo primero a evaluar era si la creación de los ambientes blue/green generaban algún tipo de desconexión sobre los ambientes originales durante la creación. Para ello se generó un script que se conectaba constantemente a la base de datos y realizaba consultas. Se concluyó que, efectivamente, la creación del despliegue blue/green no genera ningún tipo de interrupción en la base de datos original. Tras la restauración, dicha nueva base de datos se actualiza a la versión indicada.

Posterior a ello, se realizó un escaneo general de los objetos de la base de datos. AWS (s.f-f) indica que, al momento de restaurar la base de datos, la información no ha sido cargada completamente; por ello es recomendado realizar un `SELECT *` sobre todos los objetos que requieran acceso rápido.

Figura 35

lazy loading en despliegues blue/green

changing the DB instance class within the same request.

Handling lazy loading when you create a blue/green deployment

When you create a blue/green deployment, Amazon RDS creates the primary DB instance in the green environment by restoring from a DB snapshot. After it is created, the green DB instance continues to load data in the background, which is known as *lazy loading*. If the DB instance has read replicas, these are also created from DB snapshots and are subject to lazy loading.

If you access data that hasn't been loaded yet, the DB instance immediately downloads the requested data from Amazon S3, and then continues loading the rest of the data in the background. For more information, see [Amazon EBS snapshots](#).

To help mitigate the effects of lazy loading on tables to which you require quick access, you can perform operations that involve full-table scans, such as `SELECT *`. This operation allows Amazon RDS to download all of the backed-up table data from S3.

If an application attempts to access data that isn't loaded, the application can encounter higher latency than normal while the data is loaded. This higher latency due to lazy loading could lead to poor performance for latency-sensitive workloads.

 **Important**
If you switch over a blue/green deployment before data loading is complete, your application could experience performance issues due to high latency.

Nota: Adaptado “Creating a blue/green deployment”. AWS. S.f

Finalmente, se ejecutó el proceso de switchover. Este proceso se ejecutó tres veces obteniendo resultados similares en cada iteración. La tabla a continuación da los tiempos aproximados.

Tabla 19

Tiempo aproximado al realizar la actualización

Creación de la base de datos	Escaneo	Switchover
3.1 horas	6.1 horas	1 minuto
3.2 horas	6.2 horas	1.1 minutos
3.1 horas	6 horas	1 minuto

Nota: Elaboración propia. Se realizaron solo tres iteraciones dado el costo asociado con la creación y mantenimiento de las bases de datos.

Con la información obtenida anteriormente se tuvo una reunión con los interesados y se definió el plan de actualización y de rollback.

El plan al que se llegó fue el siguiente:

1. Coordinar una semana en la que podamos realizar la migración con los equipos de legado. Necesitamos que durante esa semana no se realicen cambios DDL contra producción por motivos de seguridad. Las escrituras/lecturas pueden seguir realizándose con regularidad.
2. Informar a los clientes de que habrá un día de inactividad.
3. El jueves anterior a la semana de la actualización se creará el despliegue blue/green apuntando a la versión 8.0.32.
4. Se realizará el escaneo de todas las tablas en el periodo de tiempo para prevenir problemas de latencia causados por la restauración de los datos.
5. Durante los días previos a la actualización, se realizarán pruebas de acceso a los datos del nuevo clúster actualizado a la versión 8.0.32.
6. En el lunes cuando se ejecute la actualización, se cambiará la configuración del CDN para poner el sitio web en modo mantenimiento.
7. Se pausará la replicación de información al DATA WAREHOUSE.
8. Se ejecutará el switch over.
9. Una vez realizado esto se realizarán directamente sobre el reproductor.
10. De ser positivas, se procederá a activar el sitio de administración usando el CDN y se procederá a lanzar cursos y se verificará su correcto funcionamiento.
11. De no encontrarse mayores errores, se activará el sitio web para que funcione con normalidad.
12. Reiniciar la replicación de información al DATA WAREHOUSE.
13. Se procederá a monitorear la aplicación por el resto de los días.
14. Finalmente se eliminará el clúster antiguo de no encontrarse errores tras una semana.

En lo que respecta al plan de rollback, se utilizó la última prueba para generar un plan de contingencia en caso de fallo. AWS no provee una forma de realizar un rollback tras el switchover. Por ende, se decidió usar los registros de DNS en Route53 para lograr redirigir el tráfico al clúster antiguo de la base de datos.

Los pasos fueron los siguientes:

1. Poner el sitio web en modo mantenimiento utilizando el mismo método que en el plan de actualización.

2. Detener la replicación al DATA WAREHOUSE.
3. Obtener las URL de conexión a las instancias de la base de datos antiguas. Estas se pueden obtener usando la consola de AWS.
4. Utilizar el servicio Route53, encontrar todas las entradas que referencian la base de datos de producción y corregir los valores con los obtenidos en el paso 1.
5. Reiniciar la aplicación del reproductor. El catálogo no necesita ser reiniciado.
6. El resto de los pasos son idénticos al del plan de actualización.

La decisión fue que este proceso solo se ejecutaría en caso de errores masivos en las aplicaciones afectadas. Esto era improbable dada la cantidad de pruebas que se realizaron, sin embargo, se decidió realizar este plan de rollback como contingencia y se probó en el ambiente de desarrollo.

Finalmente, tras haber definido la estrategia para producción, se finalizó enfocándose en los elementos de los ambientes de desarrollo y locales que faltaban tocar.

3.2.5.5 Etapa de modificaciones en la infraestructura de los ambientes de desarrollo

Tabla 20

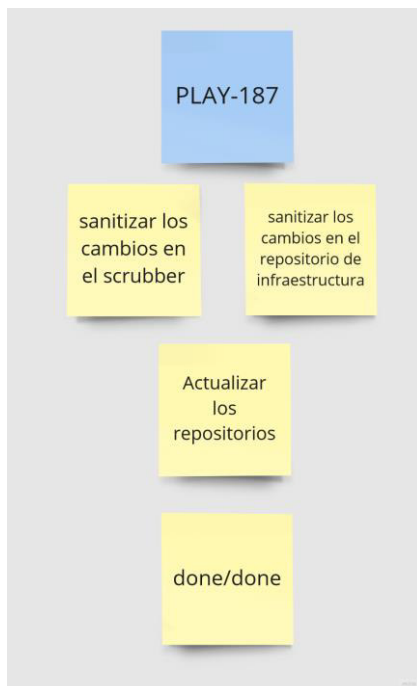
Historia de Usuario PLAY-187

Identificador	PLAY-187
Título	Actualizar scrubber y referencias en el repositorio de infraestructura de MySQL 5.7.42 a 8.0.32
Estimación	3
Descripción	Se requiere la actualización de la base de datos para los ambientes de desarrollo a MySQL 8.0.32, para ello, se requiere realizar un cambio en el scrubber y en las referencias en el repositorio de infraestructura.
Criterios de Aceptación	• Los ambientes de prueba y desarrollo utilizan MySQL 8.0.32

Nota: Elaboración propia.

Figura 36

Mapa de tareas PLAY-187



Nota: Elaboración propia.

Esta historia se encargaría de consolidar los cambios en los repositorios del scrubber y de infraestructura los cuales ya habían sido trabajados durante la fase de pruebas de todas las historias anteriores ya que algunos de estos cambios eran necesarios para generar ambientes de prueba con la nueva base de datos.

El scrubber está encargado de generar copias de seguridad anonimizadas basadas en la información de producción, para ello se restaura una copia de seguridad de producción en una nueva base de datos, que luego es anonimizada y sobre la cual se genera una copia de seguridad compartida con los ambientes de prueba. Para permitir el soporte sobre MySQL 8, simplemente se parametrizó la versión objetivo de estos back ups.

Figura 37

Parametrización del scrubber

```

12 12 SANITIZED_DB_USER = '%<@sanitized_db_user %>'
13 13 SANITIZED_DB_PASSWORD = '%<@sanitized_db_password %>'
14 14 SANITIZED_DB_SNAPSHOT_KMS_KEY_ID = '%<@sanitized_db_snapshot_kms_key_id %>'
15 15 SANITIZED_DB_ENGINE_VERSION = '%<@sanitized_db_engine_version %>'
16 16 SANITIZED_DB_SNAPSHOT_PARAMETER_GROUP_NAME = '%<@sanitized_db_snapshot_parameter_group_name %>'
17 17 SANITIZED_DB_SNAPSHOT_SUBNET_GROUP_NAME = '%<@sanitized_db_snapshot_subnet_group_name %>'
18 18 SANITIZED_DB_SNAPSHOT_SECURITY_GROUP_ID = '%<@sanitized_db_snapshot_security_group_id %>'

@@ -546,7 +547,7 @@ namespace :scrub do
546 547   delete_scrubber_db_instance RDS_SCRUBBER_SANITIZED_DB_NAME
547 548   end
548 549   end
549 549   - create_db_cmd = "aws rds create-db-instance --master-username #{SANITIZED_DB_USER} --master-user-password '#{SANITIZED_DB_PASSWORD}' --db-parameter-group-name #
(SANITIZED_DB_SNAPSHOT_PARAMETER_GROUP_NAME) --storage-encrypted --kms-key-id #{SANITIZED_DB_SNAPSHOT_KMS_KEY_ID} --engine-version %7 --backup-retention-period 0 --allocated-
storage 4500 --db-subnet-group-name #{SANITIZED_DB_SNAPSHOT_SUBNET_GROUP_NAME} --vpc-security-group-ids #{SANITIZED_DB_SNAPSHOT_SECURITY_GROUP_ID} --db-instance-identifier #
(RDS_SCRUBBER_SANITIZED_DB_NAME) --engine mysql --storage-type gp2 --db-instance-class db.m4.xlarge --no-auto-minor-version-upgrade --no-enable-iam-database-authentication --region
#{AWS_REGION} --tags Key=BackupPlanARN,Value=ScrubberBackupsBypassed"
550 550   + create_db_cmd = "aws rds create-db-instance --master-username #{SANITIZED_DB_USER} --master-user-password '#{SANITIZED_DB_PASSWORD}' --db-parameter-group-name #
(SANITIZED_DB_SNAPSHOT_PARAMETER_GROUP_NAME) --storage-encrypted --kms-key-id #{SANITIZED_DB_SNAPSHOT_KMS_KEY_ID} --engine-version #{SANITIZED_DB_ENGINE_VERSION} --backup-
retention-period 0 --allocated-storage 4500 --db-subnet-group-name #{SANITIZED_DB_SNAPSHOT_SUBNET_GROUP_NAME} --vpc-security-group-ids #{SANITIZED_DB_SNAPSHOT_SECURITY_GROUP_ID} --
db-instance-identifier #{RDS_SCRUBBER_SANITIZED_DB_NAME} --engine mysql --storage-type gp2 --db-instance-class db.m4.xlarge --no-auto-minor-version-upgrade --no-enable-iam-
database-authentication --region #{AWS_REGION} --tags Key=BackupPlanARN,Value=ScrubberBackupsBypassed"
551 551   log_info "Creating #{RDS_SCRUBBER_SANITIZED_DB_NAME} ..."
552 552   start_time2 = Time.now
553 553   exec_command create_db_cmd

```

Nota: Elaboración propia

En el caso del repositorio de infraestructura, este último es el encargado de generar los ambientes de prueba. Con respecto a la base de datos, los únicos cambios necesarios eran en los grupos de parámetros y grupo de opciones. La versión de la base de datos se toma de la copia de seguridad restaurada por lo que siempre y cuando se apunte a una copia de seguridad con la versión correcta de MySQL, esta sería restaurada sin mayores complicaciones.

Figura 38

Cambios en el repositorio de infraestructura

```
modules/database/inputs.tf
@@ -80,7 +80,7 @@ variable "auto_minor_version_upgrade" {
80 80
81 81 variable "parameter_group_family" {
82 82   description = "Family of DB parameter group"
83 83   - default   = "mysql5.7"
84 84   + default   = "mysql8.0"
85 85 }
86 86 variable "snapshot_type" {

modules/database/rds.tf
@@ -302,11 +302,11 @@ resource "aws_db_option_group" "db_parameter_group" {
302 302 resource "aws_db_option_group" "db_option_group" {
303 303   name           = "${var.database_identifier}-options"
304 304   - option_group_description = "Default options group for MySQL 5.7 with Audit Plugin Enabled"
305 305   + option_group_description = "Default options group for MySQL 8.0 with Audit Plugin Enabled"
306 306   tags           = var.default_tags
307 307   engine_name    = "mysql"
308 308   - major_engine_version = "5.7"
309 309   + major_engine_version = "8.0"
310 310   option {
311 311     option_name = "MARIADB_AUDIT_PLUGIN"
```

Nota: Elaboración propia

Tabla 21

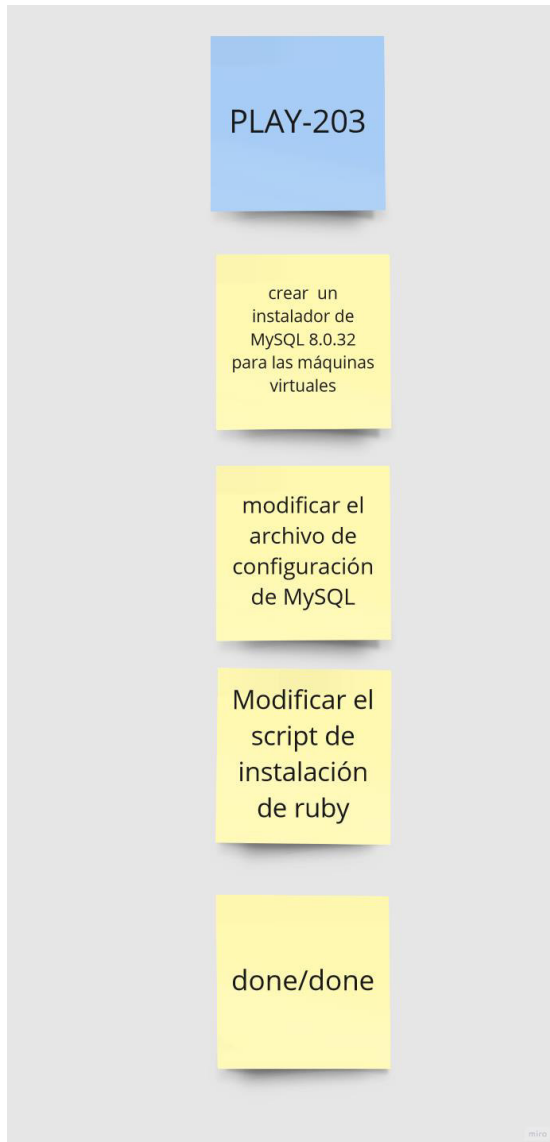
Historia de Usuario PLAY-203

Identificador	PLAY-203
Título	Instalar MySQL 8.0.32 en máquinas virtuales para el desarrollo local
Estimación	2
Descripción	Se requiere que los ambientes de desarrollo locales utilicen MySQL 8.0.32.
Criterios de Aceptación	La máquina virtual usada en los ambientes locales utiliza MySQL 8.0.32 como base de datos

Nota: Elaboración propia.

Figura 39

Mapa de tareas de PLAY-203



Nota: Elaboración propia

Los ambientes locales usan máquinas virtuales para emular las instancias de EC2 usadas en los ambientes de desarrollo y producción. Para ello se genera una máquina virtual con el catálogo y otra con la base de datos.

Para la base de datos, se utiliza código Ruby en la herramienta Chef para realizar la instalación. Por lo que se decidió cambiar este código para generar utilizar la base de datos MySQL 8.0.32.

La primera tarea fue crear un instalador que nos permitiese instalar la versión requerida en las máquinas virtuales. Se tomó como base las instrucciones en la documentación de MySQL, se creó un script bash en el cual se tomaba la versión necesaria y se realizaba la instalación.

Figura 40

Script de instalación de MySQL

```
1  #!/bin/sh
2
3  ## Docs:
4  ## * https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/secure-deployment-install.html
5  ## * https://dev.mysql.com/doc/mysql-secure-deployment-guide/8.0/en/secure-deployment-post-install.html
6
7  ## install deps
8  apt-get update -y
9  apt-get install wget lsb-release gnupg systemd libaio1 libnuma-dev -y
10
11 ## download the file
12 wget -c https://downloads.mysql.com/archives/get/p/23/file/mysql-8.0.32-linux-glibc2.12-x86_64.tar.xz -O /tmp/mysql.tar.xz
13
14 ## variables
15 MYSQL_HOME=/usr/local/mysql
16
17 ## extract and link
18 tar xvf /tmp/mysql.tar.xz --directory=/usr/local
19 ln -s /usr/local/mysql-/ $MYSQL_HOME
20 rm /tmp/mysql.tar.xz
21
22 ## create data folder
23 mkdir -p $MYSQL_HOME/data
24
25 ## create users/groups
26 groupadd -g 27 -o -r mysql
27 useradd -M -N -g mysql -o -r -d $MYSQL_HOME/data -s /bin/false -c "MySQL Server" -u 27 mysql
28
29 ## add env variables
30 export PATH=$MYSQL_HOME/bin:$PATH
31
32 ## creating safe directory
33 mkdir $MYSQL_HOME/mysql-files
34 chown mysql:mysql $MYSQL_HOME/mysql-files
35 chmod 750 $MYSQL_HOME/mysql-files
36
37 ## create sock containers
38 mkdir -p /var/run/mysqld/
39 chown mysql:mysql /var/run/mysqld
40 chmod 750 /var/run/mysqld
41
42 ## create conf.d folder
43 mkdir -p /etc/mysql/conf.d
44 chown mysql:mysql /etc/mysql/conf.d
45 chmod 750 /etc/mysql/conf.d
46
47 ## create mysql.conf.d folder
48 mkdir -p /etc/mysql/mysql.conf.d
49 chown mysql:mysql /etc/mysql/mysql.conf.d
50 chmod 750 /etc/mysql/mysql.conf.d
51
52 ## default server options
53 chown root:root /etc/mysql/my.cnf
54 chmod 644 /etc/mysql/my.cnf
55
```

Nota: Elaboración propia.

Tras aquello se realizaron los cambios en el archivo de configuración de MySQL y se realizaron los cambios en los archivos de la herramienta Chef necesarios.

Figura 41

Archivo de configuración MySQL

```
1 [client]
2 socket          = /var/run/mysql/mysql.sock
3
4 [mysqld_safe]
5 socket          = /var/run/mysql/mysql.sock
6 nice            = 0
7
8 [mysqld]
9 sql_mode       = ""
10 user           = mysql
11 pid-file       = /var/run/mysql/mysql.pid
12 socket         = /var/run/mysql/mysql.sock
13 port          = 3306
14 basedir       = /usr/local/mysql
15 datadir       = /usr/local/mysql/data
16 tmpdir        = /usr/local/mysql/data
17 secure_file_priv = /usr/local/mysql/mysql-files
18 log-error     = /usr/local/mysql/data/localhost.localdomain.err
19 skip-external-locking
20 character-set-server = utf8
21 collation-server = utf8_general_ci
22 local_infile   = OFF
23
24 bind-address   = <% @bind_address %>
25
26 key_buffer_size = 16M
27 max_allowed_packet = 16
28 thread_stack   = 192K
29 thread_cache_size = 8
30
31 myisam-recover-options = BACKUP
32 max_connections      = 250
33
34 expire_logs_days    = 10
35 max_binlog_size     = 100M
36
37 innodb_buffer_pool_size = 512M
38 default_time_zone   = '+00:00'
```

Nota: Elaboración propia.

Figura 42

Scripts de Chef para usar el nuevo instalador

```
execute 'apt-get update'

node.set[:os_solar][:mysql][:bind_address] = '192.169.0.101' unless node[:ec2]
node.default[:os_solar][:mysql][:bind_address] = '0.0.0.0'

execute 'mkdir -p /etc/mysql/mysql.conf.d'

template '/etc/mysql/mysql.conf.d/mysql.cnf' do
  source 'mysql/mysql.cnf.erb'
  variables(
    bind_address: node[:os_solar][:mysql][:bind_address]
  )
end

cookbook_file '/etc/mysql/my.cnf' do
  source 'mysql/my.cnf'
  manage_symlink_source true
end

cookbook_file '/tmp/mysql8-installer.sh' do
  source 'mysql/mysql8-installer.sh'
  mode 777
end

execute 'Installing Mysql 8.0.32' do
  command 'sh /tmp/mysql8-installer.sh'
end
```

Nota: Elaboración propia

Tabla 22

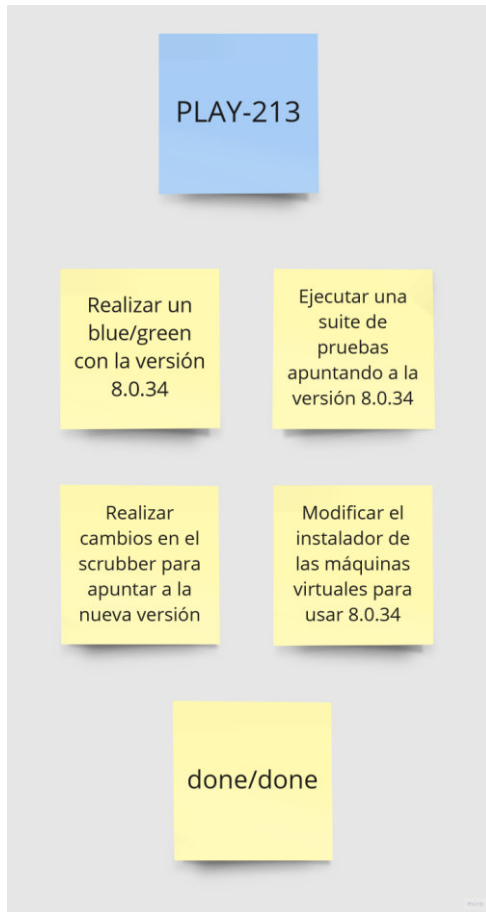
Historia de Usuario PLAY-213

Identificador	PLAY-213
Título	Probar actualización a MySQL 8.0.34 en vez de 8.0.32
Estimación	3
Descripción	Un correo de AWS ha avisado a la compañía la necesidad de actualizar MySQL a una versión 8.0.34 en vez de 8.0.32. Esto es una recomendación de AWS para un mejor soporte y mejoras en las funcionalidades.
Criterios de Aceptación	• Realizar un blue/green en desarrollo apuntando a 8.0.34 • Ejecutar una suite de pruebas apuntando a 8.0.34 • Cambiar el db scrubber para apuntar a la versión 8.0.34 • Verificar el instalador local de base de datos para las máquinas virtuales.

Nota: Elaboración propia.

Figura 43

Mapa de tareas de PLAY-213



Nota: Elaboración propia

Tras casi finalizar las tareas anteriormente descritas en otras historias, llegó un correo de Amazon informando sobre la posibilidad de un soporte extendido. En el momento que llegó el correo el proyecto estaba a punto de finalizar por lo que se decidió proceder con la finalización de este.

Sin embargo, en el mismo correo, Amazon indica que la base de datos se debería de actualizar a la versión 8.0.34 como mínimo, esto se aprecia en el correo a continuación.

Figura 44

Correo de Amazon sobre la actualización de la base de datos

Hello,

You are receiving this message because you have one or more Amazon RDS for MySQL DB instances running minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 or 5.7.42 that require your attention. MySQL community plans to deprecate MySQL major version 5.7 in October, 2023 [1] and MySQL 5.7 minor versions will no longer be supported by the community after October 31, 2023. Amazon RDS will end standard support of MySQL minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 on January 16, 2024.

We recommend that you take action and upgrade your MySQL databases (DB) running minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 to 8.0.34 or higher before RDS end of standard support in January, 2024. MySQL 8.0 offers new query functionality, enhancements for better performance, reliability, security and manageability. To learn more about the benefits of MySQL 8.0, refer to 'Amazon RDS for MySQL 8.0 Announcement' [2].

We strongly recommend that you upgrade your RDS for MySQL DB instances running minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 as soon as possible, before January 16, 2024. You may consider using Amazon Blue/Green Deployments to minimize risk and downtime during your MySQL 5.7 to MySQL 8.0 upgrade. Your Amazon RDS for MySQL instances running version 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 are listed in the "Affected Resources" tab of the AWS Health Dashboard.

If you do not upgrade your instances running minor versions listed above to MySQL 5.7.44 or major version 8.0, the timeline for your DB instance will be as follows: Starting December 16, 2023 00:00:01 AM UTC, you will not be able to create new RDS instances with MySQL minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 or 5.7.42 from either the AWS Console or the CLI. We recommend you to upgrade your databases before January 16, 2024. RDS will upgrade your MySQL databases running minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 as well as any instances restored from the snapshots of these versions to MySQL major version 8.0 during a scheduled maintenance window between January 16, 2024 00:00:01 UTC and February 16, 2024 00:00:01 UTC. On February 16, 2024 00:00:01 AM UTC, any MySQL databases running minor versions 5.7.37, 5.7.38, 5.7.39, 5.7.40, 5.7.41 and 5.7.42 that remain will be upgraded to major version 8.0 regardless of instances' scheduled maintenance window.

[RDS Extended Support for MySQL 5.7.44]

On September 1, 2023, Amazon RDS announced Extended Support for MySQL 5.7 major version. MySQL version 5.7.44 is the only minor version eligible for Extended Support. If you are unable to complete your upgrade to major version 8.0 before February 29, 2024, and would like to continue running RDS for MySQL DB instances on major version 5.7, you can opt in to RDS Extended Support and upgrade your instance to version 5.7.44 anytime before the end of standard support date on February 29, 2024. You can also opt in to Automatic Minor Version Upgrade [3] so RDS can auto upgrade your minor version to the Extended Support eligible version, 5.7.44. To learn more about RDS Extended Support, supported versions and associated charges, refer to the Amazon RDS for MySQL pricing page [4]. If RDS Extended Support is the right option for your business, you can opt in to RDS Extended Support starting in December, 2023. We will publish a notification on AWS Health Dashboard when it's time to opt-in.

If you have opted in to RDS Extended Support on your RDS for MySQL 5.7 DB instance, RDS will NOT auto-upgrade that DB instance to MySQL 8.0, but your instance will be upgraded to MySQL 5.7.44, if it's not already running this version. Such DB instances will be covered under RDS Extended Support and charged for Extended Support as per the prices on the Amazon RDS pricing page [4].

If you have any questions or concerns, the AWS Support Team is available on AWS re:Post [5] and via Premium Support [6].

[1] <https://www.mysql.com/support/>
[2] <https://aws.amazon.com/about-aws/whats-new/2018/10/amazon-rds-now-supports-mysql-8/>
[3] https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_UpgradeDBInstance_Upgrade.html#USER_UpgradeDBInstance_Upgrade_AutoMinorVersionUpgrades
[4] <https://aws.amazon.com/rds/pricing/>
[5] <https://repost.aws>
[6] <https://aws.amazon.com/support>

Nota: Elaboración propia.

Dada esta nueva información se decidió realizar las mismas pruebas descritas en todo este proceso para validar que no existiese mayores complicaciones entre las versiones. Se ejecutó lo siguiente:

- Un blue/green en desarrollo.
- Una suite de pruebas.
- Cambios en el scrubber.
- Y una prueba del instalador para las máquinas virtuales locales.

De las tareas anteriormente descritas, solo la cuarta terminó en un error, este fue causado por una dependencia que no existía en el sistema operativo de la máquina virtual, Ubuntu 18.04, sobre la cual dependía la versión de MySQL 8.0.34 a instalar.

Tras una discusión interna, se decidió generar compilar manualmente MySQL 8.0.34 desde el código fuente.

Tabla 23

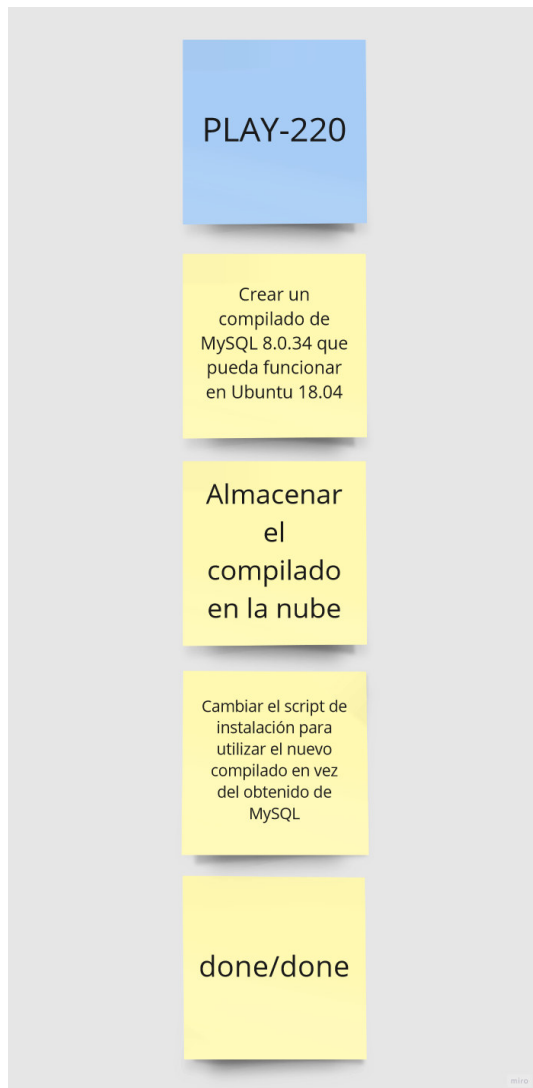
Historia de Usuario PLAY-220

Identificador	PLAY-220
Título	Generar una compilación personalizada de MySQL 8.0.34 para Ubuntu 18.04 dado a que la plataforma no soporta esta versión de manera nativa
Estimación	3
Descripción	Dados los resultados de PLAY-213, se ha detectado un problema de compatibilidad entre MySQL 8.0.34 y Ubuntu 18.04 causado por una versión de GLIBC (se necesita la versión 2.28 y Ubuntu posee 2.27). Necesitamos generar un instalador personalizado para poder ejecutar una instancia de MySQL 8.0.34 en las máquinas virtuales.
Criterios de Aceptación	• MySQL 8.0.34 se puede ejecutar en las máquinas virtuales. • El pipeline de creación de máquinas virtuales se ejecuta de manera exitosa

Nota: Elaboración propia.

Figura 45

Mapa de tareas de PLAY-220



Nota: Elaboración propia.

El error conseguido en las pruebas de PLAY-213 se puede apreciar en la siguiente imagen.

Figura 46

Error al ejecutar MySQL 8.0.34 en Ubuntu 18.04

```
bin etc games include lib man mysql mysql-8.0.34-linux-glibc2.28-x86_64 sbin share src
root@232f21734988:/usr/local# ls -l
total 36
drwxr-xr-x 2 root root 4096 May 30 02:03 bin
drwxr-xr-x 2 root root 4096 May 30 02:03 etc
drwxr-xr-x 2 root root 4096 May 30 02:03 games
drwxr-xr-x 2 root root 4096 May 30 02:03 include
drwxr-xr-x 1 root root 4096 Oct 24 19:58 lib
drwxrwxrwx 1 root root 9 May 30 02:03 man -> share/man
drwxrwxrwx 1 root root 47 Oct 24 20:05 mysql -> /usr/local/mysql-8.0.34-linux-glibc2.28-x86_64/
drwxr-xr-x 9 root root 4096 Oct 24 20:05 mysql-8.0.34-linux-glibc2.28-x86_64
drwxr-xr-x 2 root root 4096 May 30 02:03 sbin
drwxr-xr-x 1 root root 4096 Oct 24 19:58 share
drwxr-xr-x 2 root root 4096 May 30 02:03 src
root@232f21734988:/usr/local# mkdir -p $MYSQL_HOME/data
root@232f21734988:/usr/local# groupadd -g 27 -o -r mysql
root@232f21734988:/usr/local# useradd -M -N -g mysql -o -r -d $MYSQL_HOME/data -s /bin/false -c "MySQL Server" -u 27 mysql
root@232f21734988:/usr/local# export PATH=$MYSQL_HOME/bin:$PATH
root@232f21734988:/usr/local# mkdir $MYSQL_HOME/mysql-files
root@232f21734988:/usr/local# chown mysql:mysql $MYSQL_HOME/mysql-files
root@232f21734988:/usr/local# chmod 750 $MYSQL_HOME/mysql-files
root@232f21734988:/usr/local# mkdir -p /etc/mysql/conf.d
root@232f21734988:/usr/local# chown mysql:mysql /etc/mysql/conf.d
root@232f21734988:/usr/local# mkdir -p /etc/mysql/mysql.conf.d
root@232f21734988:/usr/local# chown mysql:mysql /etc/mysql/mysql.conf.d
root@232f21734988:/usr/local# chmod 750 /etc/mysql/mysql.conf.d
root@232f21734988:/usr/local# chown root:root /etc/mysql/my.cnf
root@232f21734988:/usr/local# chmod 644 /etc/mysql/my.cnf
root@232f21734988:/usr/local# mkdir -p $MYSQL_HOME/data
root@232f21734988:/usr/local# chown 750 $MYSQL_HOME/data
root@232f21734988:/usr/local# chown mysql:mysql $MYSQL_HOME/data
root@232f21734988:/usr/local# $MYSQL_HOME/bin/mysqld --defaults-file=/etc/mysql/my.cnf --initialize-insecure
/usr/local/mysql/bin/mysqld: /lib/x86_64-linux-gnu/libc.so.6: version 'GLIBC_2.28' not found (required by /usr/local/mysql/bin/mysqld)
/usr/local/mysql/bin/mysqld: /lib/x86_64-linux-gnu/libc.so.6: version 'GLIBC_2.28' not found (required by /usr/local/mysql-8.0.34-linux-glibc2.28-x86_64/bin/./libprivate/libprotobuf-lite.so.3.19.4)
root@232f21734988:/usr/local# ldd --version
ldd (Ubuntu GLIBC 2.27-3ubuntu1.6) 2.27
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
root@232f21734988:/usr/local#
```

Nota: Elaboración propia.

Para instalar MySQL se utilizó el programa pre compilado que se puede descargar desde MySQL. Para la versión 8.0.34, esta aplicación fue compilada utilizando la librería GLIBC 2.28, Ubuntu 18.04 posee una versión anterior de esta librería.

Tras una discusión interna se decidió intentar crear una versión compilada de MySQL 8.0.34. Esto es posible dado que el código fuente de MySQL es de acceso libre, por lo que decidimos crear un script que compile esta versión de MySQL en un ambiente con GLIBC 2.27.

Parte de este script puede ser visto en la imagen a continuación, se realizó buscando ser lo más agnóstico posible a la versión de MySQL a instalar dado que se esperaba que en un futuro se utilice un script similar para instalar versiones más modernas de MySQL.

Figura 47

Constructor de MySQL 8.0.34

```
BOOST_HOME=/usr/local/boost
BOOST_VERSION="1.77.0"
BOOST_DIR_NAME="boost_$(echo $BOOST_VERSION | tr '.' '_')"
wget https://sourceforge.net/projects/boost/files/boost/${BOOST_VERSION}/${BOOST_DIR_NAME}.tar.gz -O /tmp/${BOOST_DIR_NAME}.tar.xz
tar -xvf /tmp/${BOOST_DIR_NAME}.tar.xz --directory=/usr/local
mv /usr/local/boost_* $BOOST_HOME
rm /tmp/${BOOST_DIR_NAME}.tar.xz

mkdir /usr/local/mysql-8.0.34-custom -p

# Clone the MySQL source code
git config --global --unset https.proxy
git clone https://github.com/mysql/mysql-server.git /usr/local/mysql-8.0.34-custom

# Change into the MySQL server directory
cd /usr/local/mysql-8.0.34-custom

# Checkout the desired version
git checkout mysql-8.0.34

# Prepare the build
mkdir build
cd build

# For source boost
cmake .. -DWITH_BOOST=${BOOST_HOME}

# for apt based libboost
# cmake ..

# Build MySQL (This will take some time!)
make

# If you want to install it after building
# sudo make install

echo "MySQL 8.0.34 build complete!"
```

Nota: Elaboración propia.

La creación de este compilado tomó varias horas, para prevenir incrementar el tiempo de construcción de las máquinas virtuales locales. Por ende, se tomó el resultado y se almacenó en la nube.

Tras aquello, se realizó una modificación al script de instalación para apuntar al compilado generado anteriormente en vez del compilado obtenido directamente de MySQL.

3.2.5.6 Etapa de ejecución y monitoreo de la actualización

Tabla 24

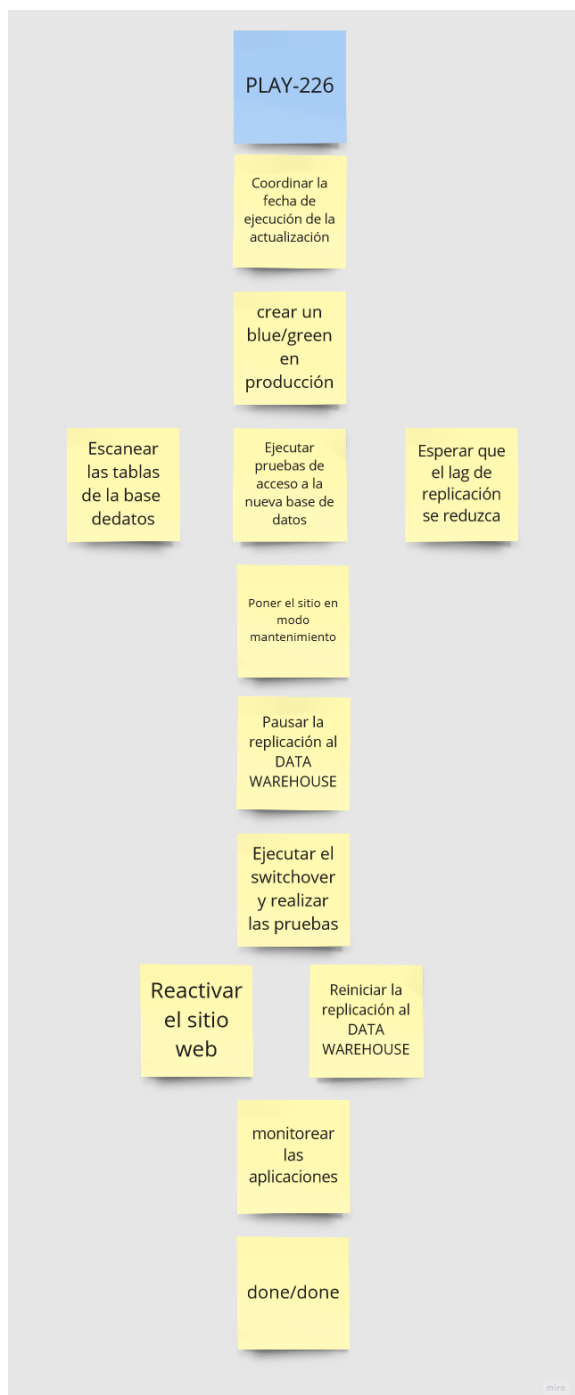
Historia de Usuario PLAY-226

Identificador	PLAY-226
Título	Ejecutar la actualización de la base de datos MySQL 5.7.42 a 8.0.34 de producción
Estimación	3
Descripción	Se requiere poner en ejecución el plan de actualización de la base de datos tras los cambios realizados en los ambientes de desarrollo, pruebas y locales.
Criterios de Aceptación	• Base de datos es actualizada con un downtime menor a 15 minutos.

Nota: Elaboración propia.

Figura 48

Mapa de tareas de *PLAY-226*



Nota: Elaboración propia.

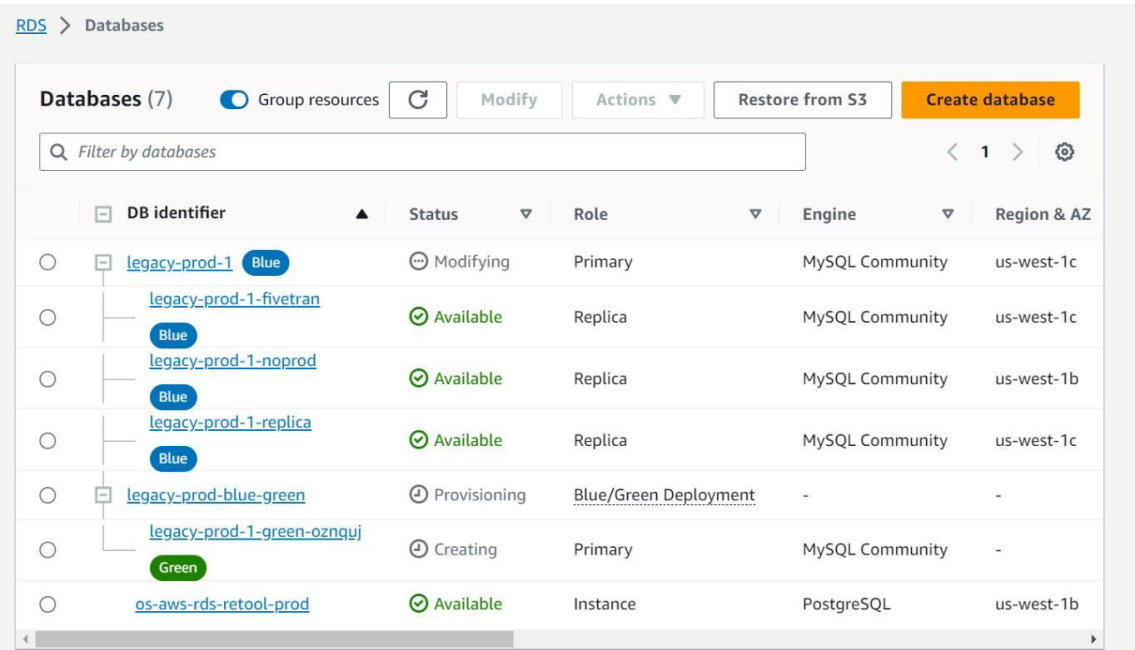
Habiendo culminado con todas las historias anteriores, se procedió a iniciar el proceso de actualización. Se puede notar que el mapa de tareas de esta historia de usuario es idéntico al plan de actualización de una forma resumida.

Dadas las características de los acuerdos de servicio de la compañía, los mantenimientos solo podían realizarse los lunes previa coordinación de quince días con los clientes.

Con ello lo primero en realizarse fue la creación del blue/green en producción, esto se puede apreciar en la siguiente imagen.

Figura 49

Blue/Green en producción



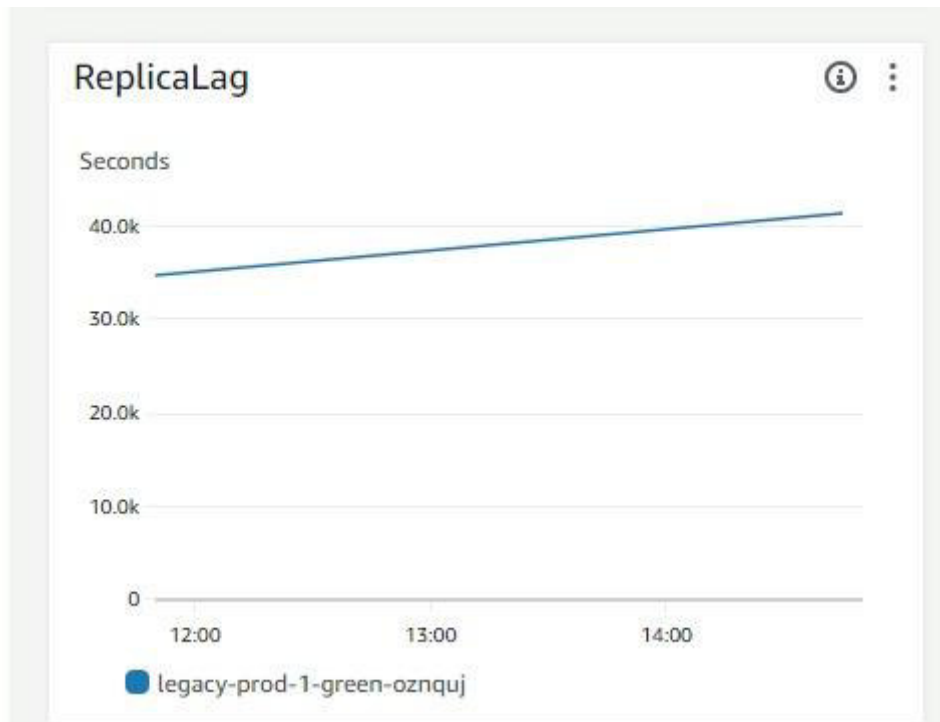
Nota: Elaboración propia.

Tras finalizar la creación, se procedió al escaneo y a la ejecución de pruebas sobre las nuevas instancias de la base de datos. Aquí encontramos un problema que no fue detectado en los ambientes de desarrollo.

El lag de replicación no parecía bajar aun tras esperar varias horas, llegando a tener más de 40 000 segundos de retraso con respecto al nodo principal. Este comportamiento era curioso dado que las réplicas originales a la base de datos no tenían este comportamiento y el aumento de este tiempo parecía seguir un ritmo lineal. Esto se puede apreciar en la siguiente imagen.

Figura 50

Lag de replicación tras creación del blue/green

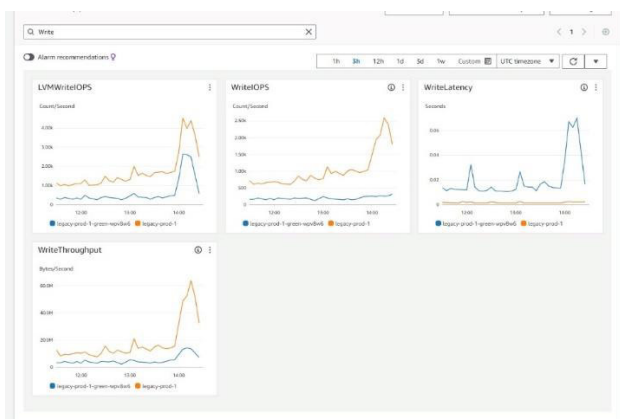


Nota: Elaboración propia.

Se realizó una investigación y se solicitó el soporte de Amazon. Pudimos notar en las métricas de la consola que las ratios de operaciones de la nueva base de datos eran menores a la anterior a pesar de poseer las mismas características técnicas. Amazon no supo darnos una explicación al porqué de este comportamiento.

Figura 51

Comparación de operaciones entre las bases de datos

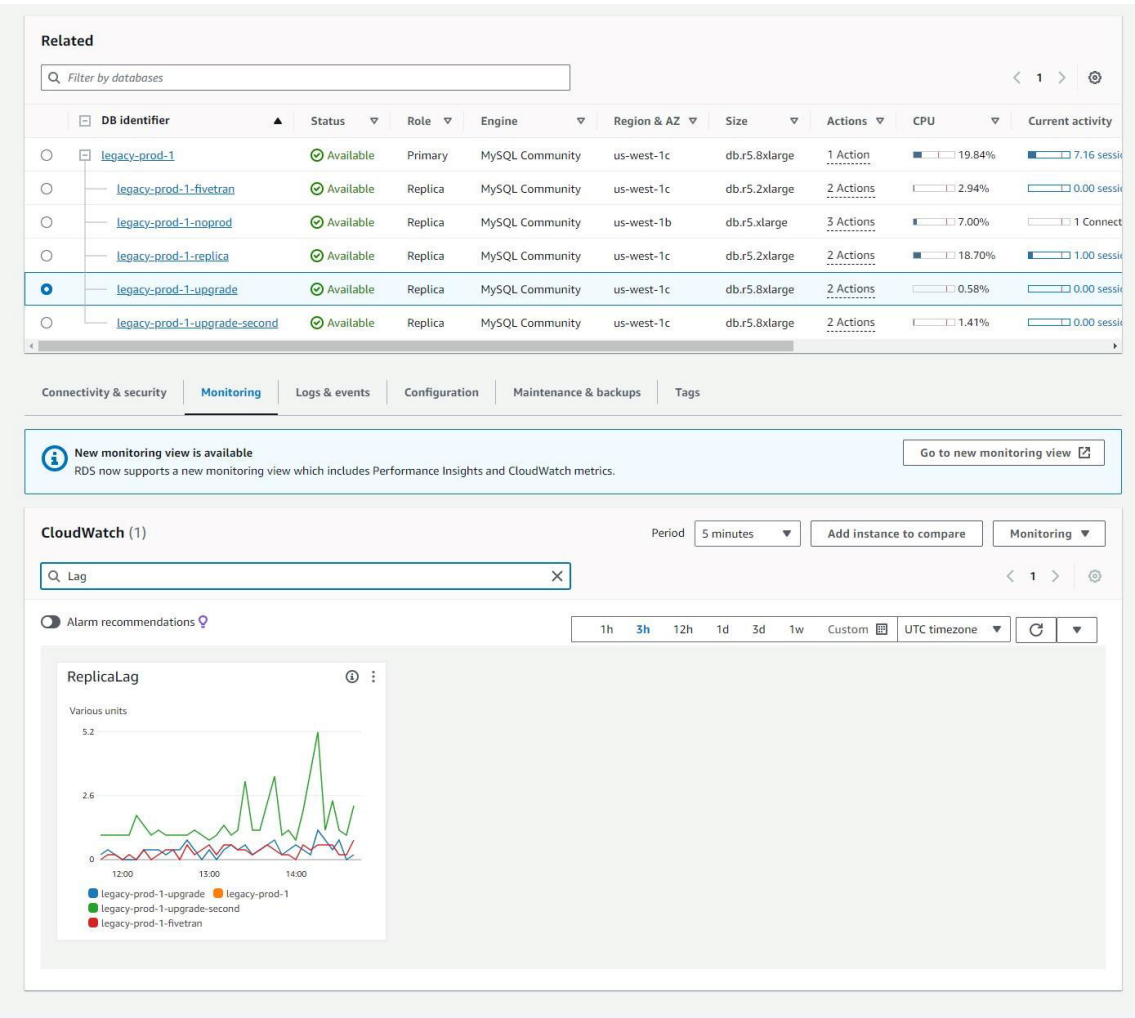


Nota: Elaboración propia.

Tras aquello se decidió eliminar el blue/green por el momento y generar bases de datos individuales para tratar de solucionar el problema.

Figura 52

Bases de datos de pruebas en producción

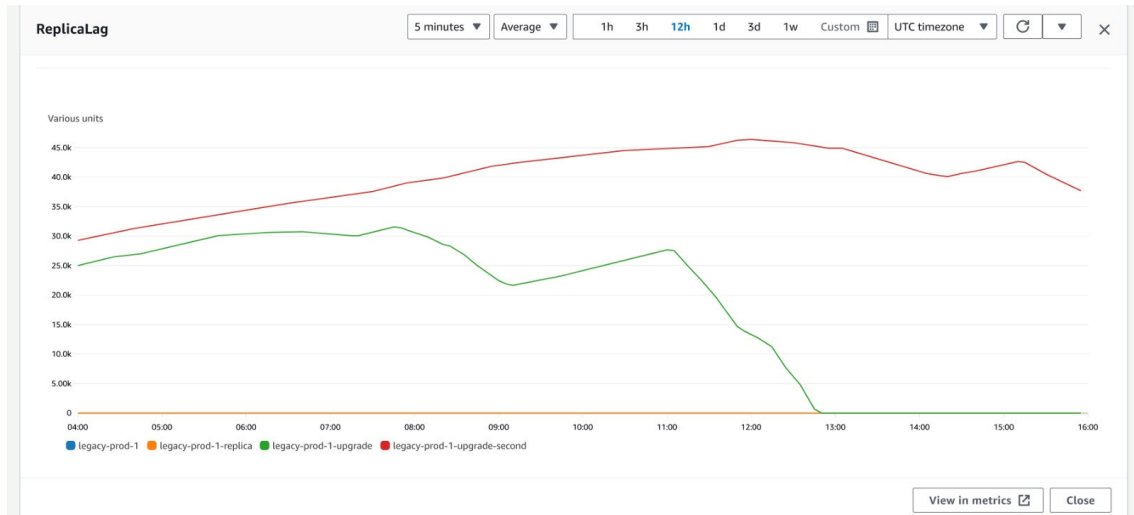


Nota: Elaboración propia.

Se probaron cambiar algunos parámetros de configuración de la base de datos, así como intentar obtener un mayor soporte de Amazon con respecto a este problema. Tras algunos días trabajando en este problema, nos dimos cuenta de que el lag empezó a decrecer lentamente. Esto se puede apreciar en la siguiente imagen, donde tras cierto momento, el lag de replicación va en picada relativamente rápido.

Figura 53

Lag de replicación de las bases de datos de prueba



Nota: Elaboración propia.

Con esto nos dimos cuenta de que era necesario tener más tiempo para que el lag decremente de manera natural. No se logró encontrar forma alguna de reducir este tiempo por lo que se modificó el plan de actualización.

En vez de iniciar el jueves de la semana anterior a la actualización, se iniciaría el lunes, dándole más tiempo al proceso para poder reducir el lag. El tiempo total que le tomó a la nueva base de datos para alcanzar a la original fue de casi cuatro días.

Figura 54

Lag de replicación de la ejecución final del plan



Nota: Elaboración propia.

Con ello se coordinó una nueva fecha de mantenimiento, y esta vez el plan fue exitoso, lográndose la actualización en menos de 5 minutos, que fue lo que le tomó a la plataforma lograr la sincronización y ejecutar el switchover.

Tras aquello, se activaron los sistemas de la compañía en el orden establecido por el plan. Nuestras herramientas de monitoreo no detectaron errores tras el mismo. El equipo se mantuvo en línea por una hora extra en caso errores empezasen a aparecer, esto no sucedió por lo que la actualización fue considerada un éxito.

Finalmente procedimos a monitorear la estabilidad de la aplicación por el resto del mes, no se encontraron errores críticos causados por la actualización, por lo que se concluyó con la eliminación del clúster original de base de datos.

3.3 EVALUACIÓN

3.3.3 EVALUACIÓN ECONÓMICA / EVALUACIÓN COSTO-BENEFICIO

El proyecto fue realizado en la búsqueda de prevenir la pérdida del soporte de AWS. Por lo que el proyecto no tuvo el objetivo de generar ganancia sino más bien la prevención de posibles pérdidas económicas.

En este contexto originalmente se pensaba que estas pérdidas podrían ser la totalidad de la facturación de la empresa, dado que la base de datos actualizada funge como su repositorio de datos central.

Sin embargo, con el soporte extendido de AWS podemos tener un cálculo del ahorro generado por la ejecución de este proyecto.

A su vez, utilizando herramientas como la calculadora de costos de AWS, podemos aproximar el costo total que requirió la ejecución de la actualización de la base de datos. El desglose de estos costos se puede ver en la siguiente tabla.

Tabla 25

Costos asociados con el proyecto en dólares

Recursos	Costo por Mes (\$)	Tiempo (meses)	Total (\$)
Base de datos para el ambiente de pruebas	763.35	3	2 290.05
Instancias Ec2 generadas para el ambiente de pruebas	366.46	3	1 099.38
Instancias EC2 usadas para generar la data de pruebas	474.59	3	1 423.77
Base de datos copiada de producción	4 145.20	1	4145.2
Base de datos para pruebas de estrés	763.35	1	763.35
Instancias EC2 para pruebas de estrés	1 099.39	1	1 099.39

Blue/Green en producción	17 084.24	1	17 084.24
Instancias EC2 para escanear las bases de datos	474.59	0.5	237.295
Sueldos del equipo de desarrollo	9 000	4	36 000
Pago al contratista de base de datos	5 000	2	10 000
Total (\$)			74 142.675

Nota: Elaboración propia.

Los costos anteriormente descritos son aproximados, ya que, durante el proceso de desarrollo del proyecto, se estuvieron creando y destruyendo múltiples ambientes de prueba, por lo que varios de los recursos anteriormente descritos no estuvieron activos durante el tiempo total del proyecto.

En lo que respecta a los montos ahorrados, se tiene una mejor visión sobre los mismos. Como ya se explicó anteriormente, la empresa usa aproximadamente 152 VCPUs entre sus ambientes de producción y de desarrollo constantemente, con ello se puede calcular el coste anual para realizar.

Tabla 26

Coste de soporte extendido de AWS en dólares

Periodo	Coste adicional por hora (\$)	VCPUs	Tiempo (horas)	Total (\$)
2024	0.1	152	8064	122 572.8
2025	0.1	152	8064	122 572.8
2026	0.2	152	8064	245 145.6
Total (\$)				490 291.2

Nota: Elaboración propia.

Con esta información se puede determinar que la realización del proyecto generó un ahorro de 416,148.53 \$ en un periodo de tres años. Esto es así dado que aún si la compañía hubiese aceptado el soporte extendido, los cambios se tendrían que hacer inevitablemente, dado que dicho soporte terminaría en el 2026.

CAPITULO IV REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

Las principales reflexiones obtenidas tras la ejecución del proyecto anteriormente expuesto fueron las siguientes:

Primero, se debe recalcar la importancia de tener un enfoque iterativo en proyectos con un alto nivel de incertidumbre es esencial. El proyecto inició con una única historia enfocada en investigar lo necesario para realizar la actualización, y culminó con numerosas áreas de código y elementos modificados. Planificar todos los cambios desde un inicio no hubiera sido viable dada la falta de conocimiento de las herramientas y los elementos a modificar.

Segundo, fue fundamental contar con la capacidad de ejecutar ambientes de prueba mediante el uso de pipelines CI/CD. De no haber tenido la capacidad de crear estos ambientes, las pruebas y experimentos realizados durante el transcurso del proyecto hubiesen sido mucho más complicados. Para esto, tener infraestructura como código fue esencial para generar estos ambientes de prueba paralelos sin afectar a otros miembros del equipo que no estuviesen trabajando en este proyecto.

Tercero, la comunicación entre los miembros del proyecto fue fundamental. La empresa posee una política de trabajo remoto completo, sin embargo, el utilizar medios de comunicación como Slack o Zoom ayudó a mantener la cohesión del equipo y a alinearnos en los objetivos del proyecto. De la misma forma, el emparejamiento en el desarrollo ayudó a mejorar la calidad de los scripts y los cambios en código realizados. Adicional a todo esto, la compañía posee un enfoque bastante horizontal en lo que respecta a las interacciones entre sus empleados, permitiendo la comunicación con líderes de equipo e inclusive con el jefe del departamento de ingeniería. Esto fue de gran ayuda para la coordinación y para solicitar ayuda al modificar áreas del sistema no tan conocidas por los miembros del proyecto.

Cuarto, los sistemas legacy generaron que este proyecto fuera más complicado de lo necesario. El tener tanta deuda tecnológica en estos sistemas generó que hubiera un alto nivel de incertidumbre sobre si las tecnologías utilizadas podrían utilizar la nueva base de datos. Como se mostró en el desarrollo del proyecto, uno de esos sistemas no era compatible con la nueva versión, teniéndose que realizar numerosos cambios en el código para poder obtener esta compatibilidad deseada. Esto generó que se tuviese que ser mucho

más cuidadoso con respecto a las pruebas, dado que no se sabía si los cambios afectarían a la funcionalidad. Cosa que sucedió y fue detectado con las pruebas de integración.

Quinto, relacionado al punto anterior y al segundo, el tener una suite de pruebas bastante comprensiva a todos los niveles ayudó a dar seguridad de que los cambios no afectarían la funcionalidad. No solo es necesario tener pruebas unitarias, sino también son fundamentales pruebas de integración y end to end.

Sexto, si bien logramos utilizar las pruebas para generar confianza en el proceso, lo cierto es que no consideramos la problemática del lag de replicación. Esto se podría haber evitado fácilmente si hubiésemos creado un blue/green en producción para verificar su comportamiento.

Finalmente, un problema que se tuvo durante todo este proceso fue el ciclo de retroalimentación, cada ejecución de pruebas y restauración de base de datos tomaba varias horas en ejecutarse. Esto implicó que el proyecto demorase más de lo que era estrictamente necesario, cambios como paralelización de pruebas o la utilización de un subgrupo de pruebas críticas podría haber acelerado el tiempo del proyecto, derivando en un menor costo.

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

1. Se generó un plan de actualización para la base de datos que pudiese ser reutilizado en posteriores actualizaciones que requieran una actualización mayor. Se sabe que se tendrá que realizar otra actualización de este tipo al pasar de MySQL 8 a MySQL 9 en un futuro.
2. Se obtuvo un plan de rollback para el fallo del plan de actualización. Este plan no solo fue probado en desarrollo, si no también ayudó a detectar.
3. El proyecto mejoró el conocimiento de los ambientes y tecnologías utilizadas por los diferentes sistemas de la organización.
4. Se actualizaron dependencias en las aplicaciones legacy para permitir el uso de la nueva base de datos manteniendo la operatividad de los sistemas.
5. Se aseguró que el tiempo muerto del mantenimiento fue menor a 15 minutos utilizando el plan de actualización descrito en el proyecto.
6. Se mantuvo la operatividad de los sistemas post actualización. No se encontraron errores ni hubo fallos de servicio relacionados con el trabajo realizado en este proyecto.
7. Se logró mantener la misma versión de MySQL para todos los ambientes de la compañía, permitiendo que las áreas locales, de prueba y de producción compartan una infraestructura similar a nivel tecnológico.

5.2 RECOMENDACIONES

1. Se recomienda establecer procedimientos estandarizados de actualizaciones periódicas. Dado que ya existe el plan de actualización, se recomienda generar un proceso que permita actualizar la plataforma de manera periódica para reducir riesgos y alinearse con las nuevas tecnologías.
2. Se recomienda la estandarización de los distintos ambientes en la nube de la compañía. Los ambientes de producción son gestionados de manera manual, por lo que utilizar herramientas como terraform podrían ayudar a alinear aún más los ambientes de desarrollo y producción.
3. Se recomienda la separación de las bases de datos del reproductor y el catálogo. Dado que ya se tiene un mayor conocimiento sobre la tecnología y plataforma, sería recomendable separar ambos esquemas dado que al estar en un solo servidor, esto resulta en un único punto de fallo de los sistemas en la compañía. Adicionalmente al separar ambos esquemas, el volumen de datos a modificar para los cambios de codificación sería menor.
4. Se recomienda crear un nuevo proyecto encargado de realizar las actualizaciones de la codificación de caracteres descrita durante el proceso de este proyecto para evitar posibles problemas de compatibilidad en futuras actualizaciones mayores de la base de datos.
5. De tener los medios económicos necesarios, es recomendable crear un blue/green o una nueva réplica en producción. Esto es para validar el comportamiento del lag en producción.
6. Se recomienda optimizar la ejecución de las pruebas para obtener ciclos de desarrollo más cortos, podría utilizarse técnicas de paralelización de pruebas o reducir la cantidad de pruebas end to end a favor de pruebas unitarias y de integración, las cuales son más veloces en su ejecución.

FUENTES DE INFORMACIÓN

- Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2017). Kanban in software engineering: A systematic mapping study. *Journal Of Systems And Software*, 137, 96-113. <https://doi.org/10.1016/j.jss.2017.11.045>
- Atlassian. (s. f.). The different types of testing in software | Atlassian. <https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>
- AWS (s.f-a). Extended support costs. https://aws.amazon.com/rds/mysql/pricing/#Amazon_RDS_Extended_Support_
- AWS. (s.f-b). What is AWS. <https://aws.amazon.com/what-is-aws/>
- AWS. (s.f-c). What is Amazon Relation Database Service (Amazon RDS). <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>
- AWS. (s.f-d). Overview of parameter groups. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/parameter-groups-overview.html>
- AWS. (s.f-e). Overview of Amazon RDS Blue/Green Deployments. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-overview.html>
- AWS. (s.f-f). Creating a blue/green deployment. <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/blue-green-deployments-creating.html#blue-green-deployments-creating-lazy-loading> costs
- AWS. (s.f-g). How can I troubleshoot high replica lag with Amazon RDS for MySQL? <https://repost.aws/knowledge-center/rds-mysql-high-replica-lag>
- Drupal. (s.f). About. <https://www.drupal.org/about>
- Hashicorp. (s.f). What is Terraform? <https://developer.hashicorp.com/terraform/intro>
- Hasan, M. H., Osman, M. H., Admodisastro, N. I., & Muhammad, M. S. (2023). Legacy systems to cloud migration: A review from the architectural perspective. *Journal Of Systems And Software*, 202, 111702. <https://doi.org/10.1016/j.jss.2023.111702>

- Iqbal, M., Ijaz, M., Mazhar, T., Shahzad, T., Abbas, Q., Ghadi, Y., Ahmad, W., & Hamam, H. (2024). Exploring issues of story-based effort estimation in Agile Software Development (ASD). *Science Of Computer Programming*, 236, 103114. <https://doi.org/10.1016/j.scico.2024.103114>
- Irani, Z., Abril, R. M., Weerakkody, V., Omar, A., & Sivarajah, U. (2022). The impact of legacy systems on digital transformation in European public administration: Lesson learned from a multi case analysis. *Government Information Quarterly*, 40(1), 101784. <https://doi.org/10.1016/j.giq.2022.101784>
- Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (2012). A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology*, 1(6), 1-5.
- Jomhari, N., Alias, N. A. A., Ellah, A. A. A., Magableh, A. A., & Ghazali, E. M. (2024). A Multi-Criteria Decision-Making for Legacy System Modernization with FUCOM-WSM Approach. *IEEE Access*, 12, 48608-48619. <https://doi.org/10.1109/access.2024.3383917>
- Karumajji, Vijay (2023). Introducing Amazon RDS Extended Support for MySQL databases on Amazon Aurora and Amazon RDS. *AWS Database Blog*. <https://aws.amazon.com/blogs/database/introducing-amazon-rds-extended-support-for-mysql-databases-on-amazon-aurora-and-amazon-rds/>
- Kelly, G. (2024, 23 septiembere). Database replication: Definition, types and setup. *Blog | Fivetran*. <https://www.fivetran.com/learn/database-replication>
- Microsoft. (s. f.). What is .NET Framework? A software development framework. <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet-framework>
- Mušić, D., Hribar, J., & Fortuna, C. (2024). Digital transformation with a lightweight on-premise PaaS. *Future Generation Computer Systems*, 160, 619-629. <https://doi.org/10.1016/j.future.2024.06.026>
- MySQL. (s.f-a). What is MySQL? <https://dev.mysql.com/doc/refman/8.4/en/what-is-mysql.html>
- MySQL. (s.f-b). The utf8 Character Set (Deprecated alias for utf8mb3). <https://dev.mysql.com/doc/refman/8.0/en/charset-unicode-utf8.html>

- MySQL. (s.f-c). The utf8mb3 Character Set (3-Byte UTF-8 Unicode Encoding).
<https://dev.mysql.com/doc/refman/8.0/en/charset-unicode-utf8mb3.html>
- MySQL. (s.f-d). Converting Between 3-Byte and 4-Byte Unicode Character Sets.
<https://dev.mysql.com/doc/refman/8.0/en/charset-unicode-conversion.html>
- MySQL. (s.f-e). Server SQL Modes. https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode_no_auto_create_user
- MySQL. (s.f-f). Creating Index Statement.
<https://dev.mysql.com/doc/refman/8.4/en/create-index.html>
- Siapno, N. (2023, 24 de julio). Performance testing vs. load testing vs. stress testing.
Blog Postman. <https://blog.postman.com/performance-testing-vs-load-testing-vs-stress-testing/>
- Rehkopf, B. M. (s. f.-a). Epics | Atlassian. Atlassian.
<https://www.atlassian.com/agile/project-management/epics>
- Rehkopf, B. M. (s. f.-b). User Stories | Examples and Template | Atlassian. Atlassian.
<https://www.atlassian.com/agile/project-management/user-stories>
- Reilly, M. (2020). MySQL 8 orphaned table due to unchecked non-existent row format check. <https://bugs.mysql.com/bug.php?id=99791>
- SolarWinds. (s.f). What is IIS Server? <https://www.solarwinds.com/resources/it-glossary/iis-server>
- Weflen, E., MacKenzie, C. A., & Rivero, I. V. (2021). An influence diagram approach to automating lead time estimation in Agile Kanban project management. *Expert Systems With Applications*, 187, 115866.
<https://doi.org/10.1016/j.eswa.2021.115866>
- Zhang, Z. (2017). The Benefits and Challenges of Planning Poker in Software Development: Comparison Between Theory and Practice.
<https://openrepository.aut.ac.nz/items/6435dc97-cade-4b0d-89b5-d8f34b642026>

GLOSARIO

- Kanban: Metodología ágil basada en el movimiento de tareas a través de etapas.
- Épica: Conjunto de trabajo que puede desglosarse en tareas.
- Historia de usuario: Unidad de trabajo que representa un requisito.
- Planning poker: Técnica de estimación de historia de usuario.
- SLA: Acuerdos de nivel de servicio en sus siglas en inglés.
- Sistemas Legacy: Sistemas con un alto grado de deuda técnico críticos para la empresa.
- IIS: Servidor web utilizado en Windows.
- Drupal: Un software de gestión de contenidos.
- Terraform: Herramienta de infraestructura como código.
- .NET Framework: Framework de desarrollo para aplicaciones web.
- C#: Lenguaje multi propósito.
- Base de datos: Colección de datos con una estructura determinada. Puede ser relacional o no relacional.
- MySQL: Una de las bases de datos relacionales más utilizadas en el mercado.
- AWS: El proveedor de servicios en la nube más completos en el mundo.
- PaaS: Modelo de servicio en la nube donde se ofertan plataformas, como bases de datos, pero donde se abstraen las capas de infraestructura.
- RDS: Es el servicio PaaS de bases de datos ofertado por AWS.
- Blue/Green: Un tipo de despliegue utilizado en AWS donde se copia la topología de la base de datos original y se crea una réplica exacta, sobre la que se pueden ejecutar pruebas.
- Swithover: Proceso mediante el cual la copia en el ambiente Green se promueve a producción, remplazando el ambiente Blue original.
- VM: Máquinas virtuales por sus siglas en inglés.
- Replicación: Proceso mediante el cual una base de datos de lectura copia información de una base de datos fuente continuamente.
- Lag de replicación: Consecuencia de la replicación entre bases de datos cuando la réplica no puede mantenerse al día con la base de datos fuente.

- Scrubber: Un sistema encargado de anonimizar la base de datos siguiendo unas reglas preconfiguradas.

ANEXOS

1. ANEXO 1 – Diagrama de flujo para la actualización de la base de datos

ANEXO 1 - Diagrama de flujo para la actualización de la base de datos

