



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

EXPERIENCIA PROFESIONAL EN GESTIÓN DE SERVICIOS
PARA SOPORTE, DESARROLLO, DESPLIEGUE Y
AUTOMATIZACIÓN DE APLICACIONES MULESOFT

TRABAJO MONOGRÁFICO
PARA OBTENER EL GRADO DE
INGENIERA EN REDES

PRESENTA

GISELLE GUADALUPE BALAM AVENDAÑO

SUPERVISORES

M.S.I. LAURA YÉSICA DÁVALOS CASTILLA

DR. JAIME SILVERIO ORTEGÓN AGUILAR

DR. JAVIER VÁZQUEZ CASTILLO

M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

M.T.I. MELISSA BLANQUETO ESTRADA

CHETUMAL QUINTANA ROO, MÉXICO, SEPTIEMBRE DE 2022





UNIVERSIDAD AUTÓNOMA DEL ESTADO DE QUINTANA ROO

DIVISIÓN DE CIENCIAS, INGENIERÍA Y TECNOLOGÍA

TRABAJO MONOGRÁFICO TITULADO

“EXPERIENCIA PROFESIONAL EN GESTIÓN DE SERVICIOS PARA SOPORTE, DESARROLLO, DESPLIEGUE Y AUTOMATIZACIÓN DE APLICACIONES MULESOFT”

ELABORADO POR

GISELLE GUADALUPE BALAM AVENDAÑO

BAJO SUPERVISIÓN DEL COMITÉ DEL PROGRAMA DE LICENCIATURA Y APROBADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE:

INGENIERA EN REDES

COMITÉ SUPERVISOR

SUPERVISORA:

M.S.I. LAURA YÉSICA DÁVALOS CASTILLA

SUPERVISOR:

DR. JAIME SILVERIO ORTEGÓN AGUILAR

SUPERVISOR:

DR. JAVIER VÁZQUEZ CASTILLO

SUPERVISOR SUPLENTE:

M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA

SUPERVISORA SUPLENTE:

M.T.I. MELISSA BLANQUETO ESTRADA



CHETUMAL QUINTANA ROO, MÉXICO, SEPTIEMBRE DE 2022



RESUMEN

El presente trabajo monográfico, detalla cómo se desarrolló la experiencia profesional en el transcurso de dos años en AVIO Consulting, en la cual se aplicaron los conocimientos adquiridos durante la carrera, así como los adquiridos por la práctica.

Se describe el objetivo general, los objetivos principales, así como el marco contextual que ofrece información necesaria para entender los aspectos generales de la empresa como sus valores y los servicios que ofrece.

Posteriormente se explica de manera sencilla y se detallan las actividades de los equipos de Managed Services y Expert on Demand, además de abarcar el proceso de capacitación, certificaciones y el proceso de Onboarding que se realiza para cada uno de los clientes.

AGRADECIMIENTOS

Agradezco a la Universidad Autónoma del Estado de Quintana Roo que me abrió las puertas como estudiante y me permitió terminar la carrera de Ingeniería en Redes.

Gracias a la M.S.I. Laura Yésica Dávalos Castilla, Dr. Jaime Silverio Ortegón Aguilar y el Dr. Javier Vázquez Castillo por aceptar este trabajo monográfico bajo su dirección.

Gracias al M.T.I. Vladimir Veniamín Cabañas Victoria por haber sido mi tutor durante toda la carrera y apoyarme en todo lo que necesitaba. Incluso para poder terminar este trabajo monográfico.

Gracias al M.S.I. Rubén E. González Elixavide por apoyarme durante la carrera y acercarme a la empresa donde actualmente trabajo y con quién aún después de egresada me siguió apoyando. En general, a todos los maestros que tuve durante la carrera, gracias por las enseñanzas y el apoyo brindado.

A mis padres, por su arduo esfuerzo para darme un mejor futuro, este logro también es de ellos.

A mis hermanas, cuñados y sobrino, gracias por siempre apoyarme.

A mis compañeros de trabajo, que han influido en mi vida profesional, y que hemos formado un gran equipo, gracias por tenerme tanta paciencia.

A mis amigos, a los que hice durante la carrera y en el trabajo, gracias por su apoyo y palabras de aliento durante todo el proceso de este trabajo monográfico.

Y por supuesto a Paul, por su apoyo incondicional y por siempre motivarme a terminar este proyecto.

DEDICATORIA

El siguiente trabajo está dedicado a:

A mis padres, Margarita Avendaño Soto y Fernando Balam Perera, que me dieron la vida. Porque han velado por mi bienestar y me han dado una carrera para mi futuro. Por creer en mí en todo momento y jamás dudar de mis capacidades. Los amo.

A mis hermanas, Karen Abigail Balam Avendaño y Nereida Balam Avendaño, por compartir sus vidas conmigo. Por tenerme paciencia y estar a mi lado aun cuando no merecía que lo estuvieran. Las amo.

A mi sobrino, Diego Jesús Quijano Balam, espero que esto te inspire y te sirva de ejemplo de que todo se puede lograr.

A Paul Peyrefitte Reyes, mi compañero en este gran viaje, deseo poder compartir contigo este y muchos otros logros, sin duda me has apoyado en los buenos momentos y en aquellos cuando más te he necesitado. Te amo.

CONTENIDO

- Capítulo 1 Introducción 1
 - Objetivos..... 1
 - General. 1
 - Específicos..... 1
 - Alcance 2
- Capítulo 2 Marco Contextual 3
 - Razón social y giro de la empresa 3
 - Visión 3
 - Misión 3
 - Valores..... 3
 - Ubicación 4
 - Servicios 4
 - Mercado..... 5
 - Premios y certificaciones 5
- Capítulo 3 Plan de capacitación y certificación Mulesoft..... 6
 - Plan de capacitación..... 6
 - Certificación Mulesoft 9
 - Formato 10
 - Costo..... 10
 - Validez 10
- Capítulo 4 Onboarding y documentación de nuevos clientes 12
 - Proceso de Onboarding 12

AVIO Service Desk	13
Documentación	20
Capítulo 5 Managed Services	30
Monitoreo diario	30
Análisis de la causa raíz de un incidente en una aplicación.....	31
Documentación de incidencias	33
Automatización de actividades	37
Capítulo 6 Expert on Demand	40
Desarrollo de nuevas funcionalidades y correcciones	40
API/RAML Design	40
Logging	41
Error Handling	42
Dataweave	42
Pruebas Unitarias MUnit.....	43
Prioridad de Tickets	44
Crítico - Prioridad 1 (P1).....	44
Mayor - Prioridad 2 (P2).....	44
Menor - Prioridad 3 (P3).....	45
Trivial - Prioridad 4 (P4)	45
Status de los Tickets.....	46
Waiting for support	46
In Progress.....	47
Waiting for customer	47
Pending	47

Cancelled	47
Customer Review	47
In Testing	48
Resolved	48
Closed	48
Capítulo 7 Conclusiones	49
Bibliografía	50

LISTA DE FIGURAS

Figura 1 Certificado Digital MCD-1. Fuente: Elaboración propia.	11
Figura 2 LastPass Vault. Fuente: Elaboración propia.	13
Figura 3 Agregar organización. Fuente: Elaboración propia.	14
Figura 4 Agregar usuarios. Fuente: Elaboración propia.	14
Figura 5 Información de contacto. Fuente: Elaboración propia.	16
Figura 6 Herramientas y detalles de versiones. Fuente: Elaboración propia.	17
Figura 7 Ambientes y diagrama de infraestructura. Fuente: Elaboración propia. ...	18
Figura 8 Aplicaciones. Fuente: Elaboración propia.	19
Figura 9 Tickets abiertos. Fuente: Elaboración propia.	20
Figura 10 Información de conexión y repositorios de Terraform. Fuente: Elaboración propia.	21
Figura 11 Información de infraestructura con administración Ansible. Fuente: Elaboración propia.	22
Figura 12 Diagrama de aplicación Mulesoft. Fuente: Elaboración propia.	23
Figura 13 Diagrama de Infraestructura Mulesoft. Fuente: Elaboración propia.	24
Figura 14 Diagrama de secuencia. Fuente: Elaboración propia.	25
Figura 15 Sistemas y mapeado de datos. Fuente: Elaboración propia.	27
Figura 16 Diagrama de secuencia y sus pasos. Fuente: Elaboración propia.	28
Figura 17 Información de las pruebas MUnits. Fuente: Elaboración propia.	29
Figura 18 Monitoreo con la herramienta Observe. Fuente: Elaboración propia. ...	30
Figura 19 Monitoreo con la herramienta Kibana. Fuente: Elaboración propia.	31
Figura 20 Monitoreo con la herramienta Anypoint Monitoring. Fuente: Elaboración propia.	31
Figura 21 Descripción y mensajes del incidente. Fuente: Elaboración propia.	34

Figura 22 Causa raíz del incidente. Fuente: Elaboración propia.	35
Figura 23 Solución ofrecida para evitar la incidencia. Fuente: Elaboración propia.	36
Figura 24 Regla para el monitoreo diario. Fuente: Elaboración propia.	37
Figura 25 Código para actualizar la versión del agente Mule. Fuente: Elaboración propia.	38
Figura 26 Código para transformar un HTML a formato RSS. Fuente: Elaboración propia.	39
Figura 27 Configuración global. Fuente: Elaboración propia.....	42
Figura 28 Workflow del status de un ticket. Fuente: Elaboración propia.	46

CAPÍTULO 1 INTRODUCCIÓN

El presente documento contiene la descripción de las actividades realizadas en un proyecto para maximizar la eficacia y calidad en la gestión de servicios de manera remota para aplicaciones Mulesoft, cumpliendo con el contrato de servicio también conocido como SLA, por sus siglas en inglés.

Durante el proceso de soporte y desarrollo se utilizaron motores de búsqueda de texto completo con interfaz RESTful para análisis de errores a través de queries, y Mulesoft como software de servicio que permite la integración de aplicaciones con servicios terceros.

Para la fase de despliegue en la nube, se utilizó el servicio de Azure DevOps de Microsoft, así como CloudHub que se trata del servicio de plataforma de Mulesoft. Además, se utilizó Terraform para la fase de automatización, se trata de un software de infraestructura como código utilizando la sintaxis de HashiCorp.

OBJETIVOS

General.

Gestionar los servicios para realizar el soporte, desarrollo, despliegue y automatización de aplicaciones Mulesoft utilizando software especializado y servicio de nube.

Específicos.

- Analizar la situación actual de la aplicación del proyecto.
- Documentar la información de accesos (VPN, Anypoint Platform, Azure DevOps, etc.) y la conexión de la aplicación con servicios terceros (base de datos, Salesforce, etc.).
- Definir las herramientas de soporte (Observe Inc., ELK stack, Sensu Go, etc.) y las plataformas de despliegue (Azure DevOps, CloudHub).
- Implementar las mejoras y/o nuevo desarrollo en las aplicaciones utilizando el Anypoint Studio.

- Realizar las pruebas unitarias utilizando el framework de MUnit proporcionado por Mulesoft.
- Desplegar las aplicaciones en su respectiva plataforma, siguiendo los lineamientos de despliegues por ambientes (Desarrollo, Pre-producción, Producción).
- Actualizar las versiones de los frameworks utilizados en las fechas agendadas.
- Realizar un análisis de errores persistentes y ofrecer soluciones del mismo.

ALCANCE

La memoria de experiencia profesional comprende de agosto 2019 a julio 2022.

Se define como alcance la documentación de los procedimientos de integración y funcionamiento de los equipos de Managed Services y Expert on Demand.

CAPÍTULO 2 MARCO CONTEXTUAL

RAZÓN SOCIAL Y GIRO DE LA EMPRESA

La empresa en donde se realiza la experiencia profesional se denomina “AVIO Consulting”.

Tiene una antigüedad de 15 años en el mercado, ofreciendo liderazgo intelectual y la mejor entrega en su clase en la intersección del desarrollo de software moderno y la integración empresarial con la plataforma Mulesoft.

VISIÓN

“Estamos enfocados en el éxito de nuestros clientes creando y entregando soluciones para lograr sus objetivos para el futuro, hoy. Debido a que el ritmo del cambio continúa acelerándose, AVIO se especializa en ayudar a nuestros clientes a lograr sus objetivos estratégicos futuros lo antes posible a través de la evolución digital” (AVIO Consulting, 2021).

MISIÓN

“Lo que parece el mañana nunca ha sido tan incierto. Es por eso que AVIO Consulting ha construido nuestra firma en torno a la comprensión rápida y la entrega de los proyectos críticos de nuestros clientes” (AVIO Consulting, 2021).

VALORES

Haremos lo que sea necesario para garantizar que nuestros clientes tengan éxito, trabajamos incansablemente para ofrecer los mejores resultados posibles (AVIO Consulting, 2021).

- **Piensa en grande:** Nos gusta pensar en grande, pero empezamos con poco y actuamos rápido.
- **Proporcionar conocimiento:** Brindamos consultores expertos y profesionales en cada paso del camino.

- **Hazlo bien la primera vez:** Creemos en empezar con el pie derecho siempre.
- **Lo que sea necesario:** Hacemos lo que sea necesario para que nuestros clientes tengan éxito. Creamos una visión compartida y la implementamos de manera eficiente y ágil (AVIO Consulting, 2021).

UBICACIÓN

La empresa se encuentra ubicada en Addison en el condado de Dallas, estado de Texas en Estados Unidos.

SERVICIOS

AVIO Consulting ofrece diferentes servicios dentro de los cuales se encuentran los siguientes:

- **Architectural Assessment:** garantiza que sus iniciativas de plataforma Anypoint estén estructuradas para el éxito, identificando oportunidades de optimización que mejoran la capacidad de mantenimiento y el rendimiento al compararlas con nuestras prácticas recomendadas de entorno, diseño y desarrollo.
- **Accelerated Success:** paquete inicial de implementación de la plataforma Mulesoft Anypoint, diseñado para clientes que desean configurar y habilitar la plataforma Anypoint basándose en el liderazgo intelectual de AVIO, en las mejores prácticas de desarrollo de aplicaciones y alineadas con los principios estratégicos de Mulesoft.
- **Maturity Assessment:** es una herramienta de diagnóstico para evaluar la competencia de su empresa en cinco categorías del ciclo de vida del desarrollo de software.
- **Managed Services:** brinda un monitoreo proactivo confiable de Mulesoft Anypoint Platform para que tenga la libertad de dirigir recursos para crear nuevos servicios que brinden un valor comercial real.

- **Proof of Concept:** brinda una visión interna del poder de nuestra práctica galardonada de Mulesoft, aplicada a contextos específicos críticos para la evolución de su negocio.
- **Expert on Demand:** soporte de duración fija diseñado para clientes existentes de Mulesoft que requieren acceso ad-hoc a nuestros expertos certificados de Mulesoft. Expert on Demand garantiza que se mantenga al día con la guía de mejores prácticas sobre nuevos desarrollos, mejoras, problemas de diseño, seguridad y más (AVIO Consulting, 2021).

MERCADO

Su mayor mercado se enfoca en las siguientes industrias.

- Servicios Financieros.
- Cuidados de la salud.
- Manufactura.
- Medios y telecomunicaciones.
- Tecnología.

PREMIOS Y CERTIFICACIONES

AVIO Consulting pertenece al listado de partners que tiene Mulesoft. Además de contar con premiaciones:

- Top Solution Partner of the Year: North America – 2018
- Mulesoft AMER Emerging Partner of the Year –2021

CAPÍTULO 3 PLAN DE CAPACITACIÓN Y CERTIFICACIÓN MULESOFT

PLAN DE CAPACITACIÓN

El plan de capacitación que ofrece AVIO Consulting consiste en un conjunto de cursos que permiten la formación de los nuevos integrantes del equipo Managed Services, mejorando sus habilidades técnicas.

Dicho plan de capacitación abarca software de infraestructura, sistemas para la automatización de despliegues, plataformas para configurar y administrar ordenadores, entre otros, que son muy comúnmente utilizados para operaciones de TI.

Se listan las herramientas y las habilidades que espera se obtengan al finalizar con el plan de capacitación:

- **Mulesoft**
 - Comprensión de la arquitectura Mulesoft.
 - Comprender la funcionalidad proporcionada por la plataforma Anypoint con énfasis en Runtime Manager, Access manager, Monitoring y API Manager.
 - Comprender como funciona la gestión de accesos, roles, permisos y organizaciones/grupos empresariales.
 - Poder agregar, eliminar, detener e iniciar componentes de CloudHub Runtime (servidores, aplicaciones, etc.).
 - Administrar, monitorear y analizar las aplicaciones Mulesoft, utilizando Runtime, API Manager y las herramientas de monitoreo en Anypoint.
 - Capacidad para configurar tableros usando Monitoring y API Manager.
 - Analizar una aplicación Mulesoft en Anypoint Studio y determinar la causa raíz de un error en la aplicación.
 - Comprender los logs de Mulesoft y qué datos capturan.

- **Amazon Web Services**
 - Comprender los patrones de infraestructura en la nube.
 - Comprender las diferencias entre regiones, zonas de disponibilidad y ubicaciones de borde de AWS.
 - Conocer los servicios principales de AWS, incluidos el procesamiento, la red, las bases de datos y el almacenamiento.
 - Conocer las prácticas de seguridad y cumplimiento en la nube de AWS.
- **Kubernetes**
 - Entender que es un clúster de Kubernetes, cómo implementarlo y administrarlo.
 - Entender como Kubernetes interactúa con otras tecnologías importantes como Docker.
 - Conocer los principales componentes de Kubernetes que permiten implementar y administrar aplicaciones.
- **EKS**
 - Comprender Amazon Elastic Kubernetes Service (EKS), como administrarlo sin necesidad de instalar.
 - Comprender como operar el control plane de Kubernetes o worker nodes.
 - Conocer las ventajas de usar EKS para alojar cargas de trabajo de Kubernetes.
 - Configurar un clúster de EKS y ponerlo en marcha.
 - Ejecutar una aplicación y explorará las funciones de EKS.
- **Elastic Stack**
 - Comprender los conceptos básicos de Elastic Stack.
 - Crear logs y recursos de sistema mediante Beats, Elasticsearch y Kibana.
 - Visualizar y analizar logs y datos en Kibana utilizando Dashboards.

- Generar alertas utilizando el módulo de Watcher.
- Conectar el servicio de Kibana con servicios terceros como Slack o Jira para automatización de tickets.
- **Observe Inc.**
 - Conocer los conceptos básicos de Observe Inc.
 - Generar Worksheets utilizando DataSets, analizar logs utilizando funciones y visualizaciones.
 - Generar Monitors que se ejecuten cuando se detecten errores en los logs.
 - Conectar los Monitors para que envíen notificaciones a servicios de comunicación como Slack.
- **Terraform**
 - Conocer los conceptos básicos de Terraform como Scope, Initialize, Plan y Apply.
 - Administrar la infraestructura en múltiples plataformas en la nube.
 - Realizar seguimientos de los cambios de recursos a través de los deployments.
- **Jenkins**
 - Conocer y entender los conceptos básicos de Jenkins, como el archivo Jenkinsfile y los stages.
 - Administrar Jenkins y poder hacer configuraciones desde su interfaz web.
 - Administrar pipelines y como configurarlas para automatización de procesos y despliegues.
 - Administrar el archivo Jenkinsfile y poder realizar actualizaciones y mejoras.
- **Ansible**

- Comprender como funciona Ansible como herramienta para automatización de despliegues de aplicaciones, orquestar ambientes y recursos en los servidores.
- Conocer los conceptos básicos de Ansible como Playbooks, Tasks, Inventory y Roles.
- Entender cómo se estructuran los archivos YAML, la cual es la estructura de los Playbooks.
- Realizar ajustes y actualizaciones a los Playbooks acorde a la infraestructura de cada cliente.

CERTIFICACIÓN MULESOFT

Para los integrantes del equipo de Managed Services y Expert on Demand se espera que concluyan con el curso de autoestudio **Anypoint Platform Development: Fundamentals**. Esto con la finalidad de obtener el certificado de **Mulesoft Certified Developer – Level 1**.

Aquellos que logren la certificación deberán ser capaces de:

- Utilizar la plataforma Anypoint alojada en Mulesoft para llevar una API básica a través de todos los pasos de su ciclo de vida: diseñar, construir, implementar, administrar y gobernar.
- Utilizar Anypoint Studio para crear, probar y depurar integraciones básicas e implementaciones de API.
- Conectarse a una variedad de recursos que incluyen bases de datos, archivos, servicios web, aplicaciones SaaS y JMS queues.
- Realizar transformaciones de datos básicas utilizando DataWeave 2.0.
- Controlar el flujo de eventos y manejo de errores.
- Procesar registros por lotes (batch records).

Formato

El examen es en línea, protegido, a libro cerrado, contando con un total de 60 preguntas de opción múltiple. La duración máxima de este examen es de 120 minutos. El puntaje de aprobación es del 70%. El examen es completamente en inglés (Mulesoft, 2022).

Costo

El costo del examen es de \$250 USD, el cual incluye un reintento gratis con una espera de 24 horas entre cada intento. De igual manera se puede obtener dos intentos con la conclusión del curso que ofrece la plataforma (Mulesoft, 2022).

Validez

El certificado expira dos años a partir de la fecha en que se pasa el examen (ver Figura 1). Para poder extender la validez después de esta fecha, se debe de tomar el examen **Mulesoft Certified Developer – Level 1 Maintenance**. El cual consiste en 25 preguntas de opción múltiple, la duración del examen es de 45 minutos, con los mismos parámetros que con el primer examen (Mulesoft, 2022).



Figura 1 Certificado Digital MCD-1. Fuente: Elaboración propia.

CAPÍTULO 4 ONBOARDING Y DOCUMENTACIÓN DE NUEVOS CLIENTES

PROCESO DE ONBOARDING

Existe un periodo de Onboarding para clientes nuevos en el cual se recaban información sobre su infraestructura y sus aplicaciones.

Durante este se realizan las siguientes actividades:

1. Obtener todos los accesos necesarios para los recursos del cliente. Por ejemplo:
 - Anypoint Platform
 - VPN
 - Source Control
 - CI/CD system
 - Email/Slack/Teams system

Durante este proceso también se obtiene información sobre Encryption Keys, o credenciales únicas para todo el equipo. Esta información se almacena en LastPass Vault que ofrece AVIO Consulting (ver Figura 2).

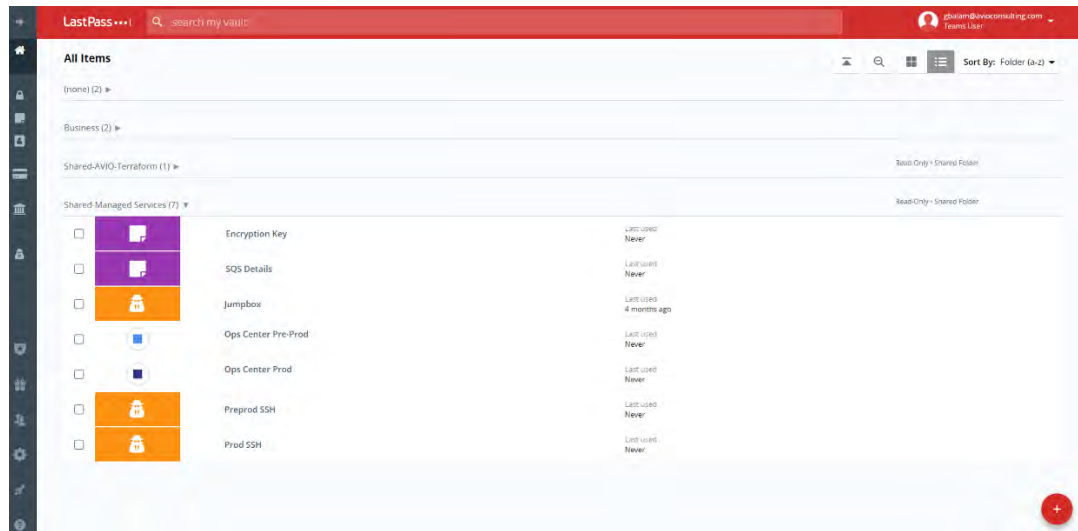


Figura 2 LastPass Vault. Fuente: Elaboración propia.

2. Establecer cuál será el medio de comunicación. Esto con la finalidad de resolver temas o dudas más rápido. La comunicación normalmente se realiza a través de Slack o Teams.
3. Revisar la documentación existente de la infraestructura Mulesoft del cliente, así como el de las aplicaciones (para cliente Expert on Demand). Algunos clientes cuentan con cierta documentación, la cual se deberá revisar durante el tiempo de Onboarding con la finalidad de resolver dudas.
4. Revisar sus procesos de desarrollo y despliegues (para clientes Expert on Demand). Una vez teniendo accesos a sus sistemas, se revisa sus procesos de desarrollo y despliegues para ofrecer alternativas o mejoras.

AVIO SERVICE DESK

Como parte del proceso de Onboarding se debe de otorgar a los nuevos clientes acceso al Service Desk. Esto con la finalidad de que los clientes puedan abrir tickets de incidencias o de consultas.

Lo primero que hay que hacer es agregar una organización en el Service Desk de AVIO Consulting (ver Figura 3).

Add organizations

Choose or create an organization

Reyes Holdings

Add Cancel

Figura 3 Agregar organización. Fuente: Elaboración propia.

Una vez agregada la organización, se pueden agregar usuarios (ver Figura 4), utilizando los correos electrónicos de las personas que se encargarán de crear tickets por parte del equipo del cliente.

Add customers

Customer emails

Enter one or more email address

Add to organizations

Reyes Holdings

Customers will receive an invitation email.
Go to customer notifications to [view or customize the invitation email](#).

Add **Cancel**

Figura 4 Agregar usuarios. Fuente: Elaboración propia.

Como segundo paso es crear una página para el cliente dentro de la wiki de AVIO Consulting. En esta página se incluirá la siguiente información:

- **Información de contacto:** Esta información deberá contener Nombre, Posición, Número de Teléfono y Correo electrónico (ver Figura 5). Además, también se incluye información de los puntos de contacto por parte del equipo de AVIO Consulting, quienes se encargarán de atender los tickets del cliente. Y la información del Service Desk del cliente en caso de contar con uno.
- **Herramientas:** Listado de herramientas que se deben de utilizar para poder trabajar con las aplicaciones del cliente, los accesos a estas herramientas se deberán obtener en el proceso de Onboarding (ver Figura 6).
- **Detalles de versiones:** Listado de las versiones con las cuales está trabajando el cliente, así como sistema operativo, detalles de su infraestructura y sus ambientes.
- **Ambientes:** listado de los ambientes activos con los que cuenta el cliente, así como su ID en el Anypoint Platform y el propósito de cada uno de estos ambientes.
- **Diagrama de la infraestructura:** Se agrega un diagrama de la infraestructura del cliente, que incluyen información de VPNs, Clusters, etc (ver Figura 7).
- **Aplicaciones:** Listado de las aplicaciones Mulesoft que se encuentran desplegadas en todos sus ambientes (ver Figura 8).
- **Tickets abiertos:** Tabla que contiene toda la información de tickets abiertos con el cliente, para mejor visibilidad de los avances en éstos (ver Figura 9).

Group: Managed Services and Expert on Demand

- [Company Information](#)
- [AVIO EOD Contacts](#)
- [Contact Information](#)
- [Client Support Desk](#)
- [Tools](#)
 - [MuleSoft Details](#)
 - [MuleSoft Environments](#)
- [Infrastructure Diagram](#)
- [Applications](#)
- [Active Requests](#)

Company Information

URL to Client website: [BUILT](#) [STRONG.](#) | [Group](#)

URL to AVIO Wiki: [Group: Expert On Demand](#)

AVIO EOD Contacts

Primary Contact: Alejandro Jimenez

Backup Contact: Michel Hernandez

Contact Information

Name	Position	Phone Number	E-Mail Address
Jason Rosfeld	Mgr, Application Development	513-XXX-XX65	jason.rosfeld@group.com
Daryl Mullins	MuleSoft Developer		Daryl.Mullins@group.com
Ron Jones	Network Team		Ron.Jones@group.com

Figura 5 Información de contacto. Fuente: Elaboración propia.

Tools

- **Log aggregation:** *Elastic Cloud (Elasticsearch)*
- **Monitoring:** *Anypoint Monitoring*
- **Support Ticketing:** *AVIO Service Desk*
- **Source Control:** *GitHub*
- **CI/CD:** *Jenkins*

MuleSoft Details

- **Version:** 4.3.0 (3 apps)
- **Runtime:** Runtime Fabric with Self Managed Kubernetes
 - **Self Hosted Infrastructure:** (VMs, Network, Heartbeat, Logstash and Ansible)
 - **Operating System:** (Windows[RDP]/Linux[SSH])
 - **Kubernetes Details:**
 - **Clusters:** 1
 - **Controller Nodes:** 3
 - **Worker Nodes:** 3
 - **Worker Size:** 1 CPU core and 1 GB of memory

Figura 6 Herramientas y detalles de versiones. Fuente: Elaboración propia.

MuleSoft Environments

Environment Name	Environment ID	Environment Purpose
DEV	4e7b2d3d-4657-4569-aeb6-52a2c72f9ccf	Unit Testing
UAT	78b9f5fe-5705-4048-86fe-b4a038ba3d0f	Integrated testing
PROD	21e94361-d3e6-42d5-b07d-98469d8432de	Production

Infrastructure Diagram

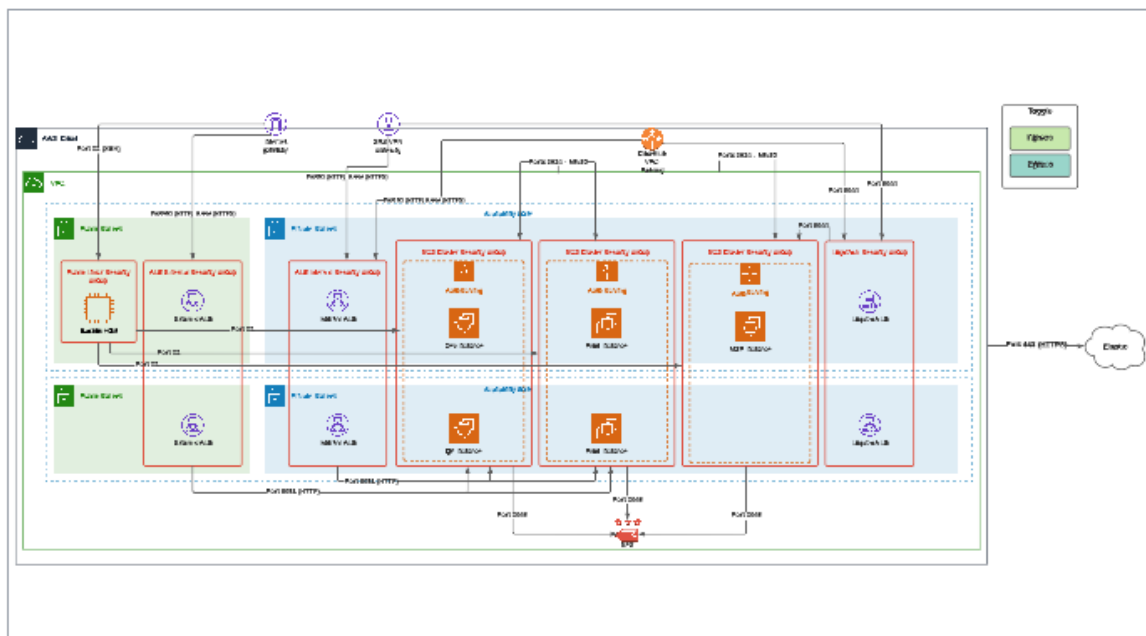


Figura 7 Ambientes y diagrama de infraestructura. Fuente: Elaboración propia.

Applications

Application Name	Description	Integrations	Links
duracell-prc-api	Called by scheduler to run batch processes to synchronize data across systems.	None	https://anypoint.mulesoft.com/exchange/a4c42b99-19f4-4db0-a3c9-0d0e7e54878b/rh-scheduler-app/minor/1.0/pages/Customer - SNFK to SFDC/
e1-sys-api	Provide real-time and batch interface to Salesforce E360 Account data.	Salesforce	https://anypoint.mulesoft.com/exchange/a4c42b99-19f4-4db0-a3c9-0d0e7e54878b/rh-scheduler-app/minor/1.0/pages/Customer AR - VIP to SFDC/
hdc-sys-api	Interface to query Customer data in VIP backend.	Snowflake	https://anypoint.mulesoft.com/exchange/a4c42b99-19f4-4db0-a3c9-0d0e7e54878b/rh-sfdc-account-sapi/

Figura 8 Aplicaciones. Fuente: Elaboración propia.

Active Requests

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
MULESUP-2307	hg-salesforce-sys-api: client connection was closed		25/Aug/22 7:48 AM	25/Aug/22 9:45 AM		Alejandro Jimenez	Jason W Rosfeld		CUSTOMER REVIEW	Unresolved
MULESUP-2303	hg-salesforce-sys-api-qa: Couldn't create partner connection because an invalid login exception has occurred		22/Aug/22 1:15 PM	Tuesday 10:31 AM		Alejandro Jimenez	Jason W Rosfeld		CUSTOMER REVIEW	Unresolved
MULESUP-2300	Logs not writing to Elasticsearch		18/Aug/22 6:54 AM	19/Aug/22 1:23 PM		Adam Mead	Jason W Rosfeld		PENDING	Unresolved
MULESUP-2297	Investigate Alert from e1-sys-api		16/Aug/22 10:43 AM	16/Aug/22 4:47 PM		Alejandro Jimenez	Daryl Mullins		CUSTOMER REVIEW	Unresolved
MULESUP-2296	hg-salesforce-sys-api: NPE in put/accounts/d/d		11/Aug/22 12:11 PM	12/Aug/22 1:54 PM		Alejandro Jimenez	Jason W Rosfeld		CUSTOMER REVIEW	Unresolved
MULESUP-2295	hdc-sys-api records are not being processed in accounts scheduler		10/Aug/22 5:53 PM	10/Aug/22 6:07 PM		Alejandro Jimenez	Jason W Rosfeld		CUSTOMER REVIEW	Unresolved
MULESUP-2286	Identify and fix causes for 'product catalog' errors in hdc-sys-api		27/Jul/22 12:15 PM	Friday 2:03 PM		Alejandro Jimenez	Jason W Rosfeld		CUSTOMER REVIEW	Unresolved
MULESUP-2269	Error when grouping items for quantity validations in shipment-prc-api		30/Jun/22 3:05 PM	08/Jul/22 1:29 PM		Alejandro Jimenez	Daryl Mullins		CUSTOMER REVIEW	Unresolved

Figura 9 Tickets abiertos. Fuente: Elaboración propia.

DOCUMENTACIÓN

Una vez finalizado el proceso de Onboarding se realiza una documentación de manera interna. Esto consiste en generar páginas en la wiki de AVIO Consulting que contengan información sobre las aplicaciones del cliente, de su infraestructura, pasos para la conexión a sus servidores o links de repositorios.

Estas son las páginas que se generan:

- **Infraestructura:** Se trata de una documentación sobre la infraestructura Mulesoft del cliente, descripción de la misma, indicando sus ambientes y como está siendo administrado (ver Figura 10 y Figura 11).

AWS Profile Config

You need to have an AWS profile configured for SRS in order to execute the Terraform. The profile name used in Terraform is `live`

Either add this to your `~/.aws/credentials` file:

```
1 [ live]
2 aws_access_key_id = YOUR ACCESS KEY ID
3 aws_secret_access_key = YOUR SECRET ACCESS KEY
```

OR add this to your `~/.aws/config` file:

```
1 [profile : live]
2 role_arn = arn:aws:iam::86061:role/OrganizationAccountAccessRole
3 source_profile = YOUR AWS PROFILE NAME
```

Terraform Code Repositories

Project Name	Link	Purpose
aws- -site	avioconsulting/aws- -site	Contains the majority of the networking infrastructure for all environments. Includes VPN, Security Groups, VPC Peering, VPN, IAM Roles, Route53 zone and certificate.
aws- -compute	avioconsulting/aws- -compute	Contains all of the compute infrastructure for environments. Contains ECS config, Route53, load balancers, bastion host, s3 buckets, sqs queues
aws-...b-site	avioconsulting/aws-...b-site	Needs review

Figura 10 Información de conexión y repositorios de Terraform. Fuente: Elaboración propia.

Infrastructure Description

The Pegasus infrastructure consists of three environments (Development, Stage, and Production) and a server that is used for monitoring and running Ansible

Development (DEV)

Hostname	.corp.local
IP Address	10.1.1.138
CPU Cores	1
Memory	16 GB
Disk	500 GB
MULE_HOME	C:/mule/mule-enterprise-standalone-4.4.0-dev01

Stage

Hostname	.corp.local
IP Address	10.1.1.137
CPU Cores	4
Memory	32 GB
Disk	500 GB
MULE_HOME	C:/mule_ee/mule-enterprise-standalone-4.4.0-stage01

Figura 11 Información de infraestructura con administración Ansible. Fuente: Elaboración propia.

- **Diagramas:** De igual forma se generan diagramas de infraestructura, de aplicaciones y los sistemas con los que están conectados, además de diagramas de secuencia (ver Figura 12, Figura 13 y Figura 14).

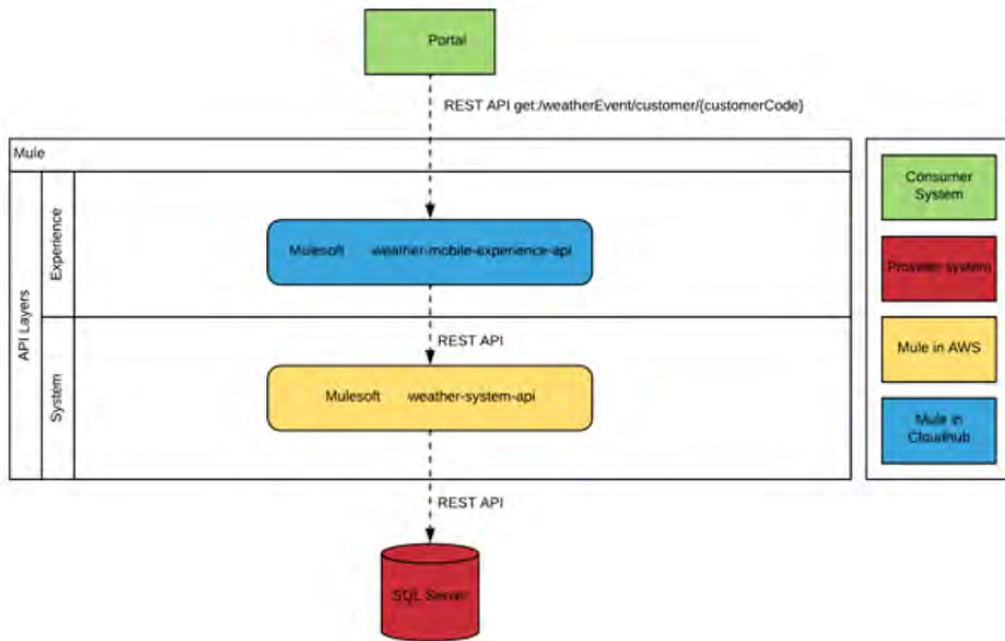


Figura 12 Diagrama de aplicación Mulesoft. Fuente: Elaboración propia.

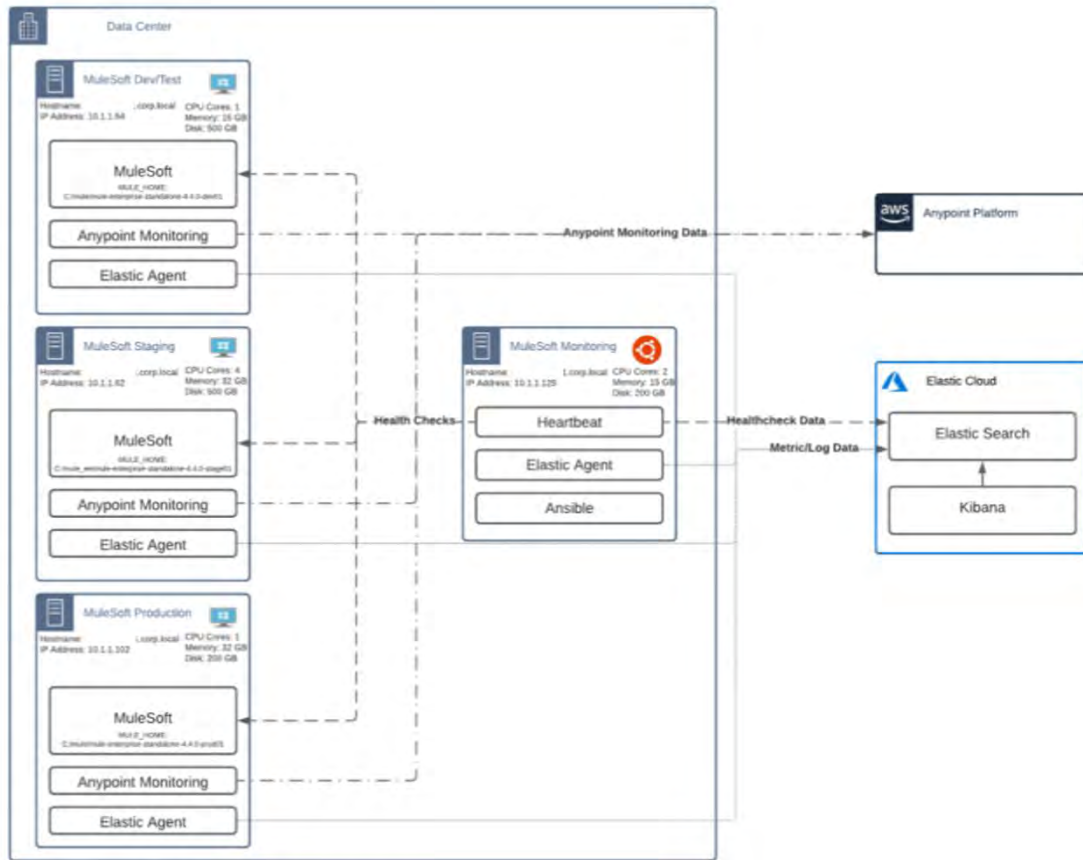


Figura 13 Diagrama de Infraestructura Mulesoft. Fuente: Elaboración propia.

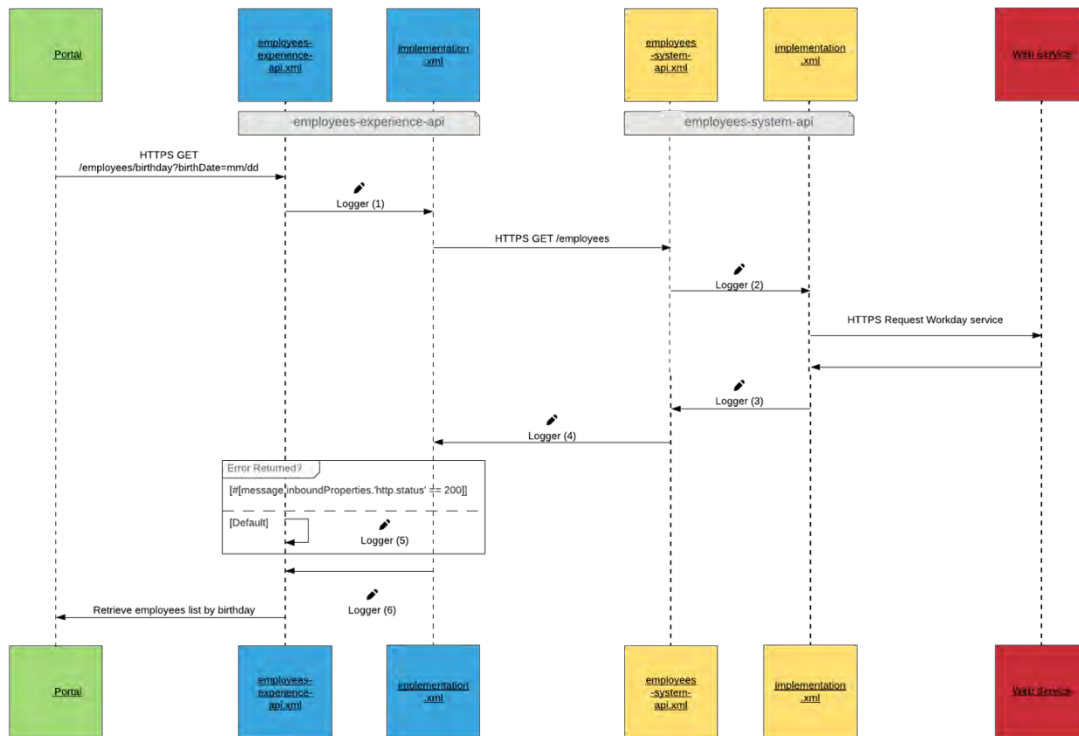


Figura 14 Diagrama de secuencia. Fuente: Elaboración propia.

- **Documentación de desarrollo:** También realizamos documentación de las aplicaciones, acorde a las buenas prácticas de AVIO Consulting. Los tipos de documentación que se realizan son los siguientes:
 - **Anypoint Exchange:** Exchange es una plataforma donde se encontrar y utilizar assets creados por el ecosistema Mulesoft, o agregar assets específicos de una empresa para colaboración y uso compartido interno. Se espera que la documentación contenga lo siguiente:
 - Documentación de alto nivel que explica lo que está haciendo la aplicación.
 - Descripciones de cada uno de los recursos(endpoints) y lo que hacen con los datos que se envían o recuperan.

- Descripciones de los datos que se devuelven.
 - Ejemplos de request exitosos y de error para que los usuarios sepan qué esperar.
 - Enlaces a documentación más detallada a la que solo se puede acceder internamente.
- **Documentación de alto nivel:** En la documentación de alto nivel se espera tener información que sea relevante para un Product Owner o un Tester (ver Figura 15 y Figura 16).
- Esta documentación solo debe ser accesible de manera interna.
 - Tener un listado de los sistemas involucrados.
 - Mapeado de datos.
 - Diagramas de secuencia y una breve descripción de los pasos.

Systems involved

- Snowflake
- Salesforce
- VIP
- Stibo

Data mappings

Customer Snowflake to Salesforce.

Snowflake Customers with type: Customers (embedded Retail Stores) and Suppliers are sent to [Salesforce](#) Account Object with Record Types: Customer, Supplier and Retail Store via a MuleSoft nightly batch.

Snowflake		Salesforce		
Field Table	Field Name	Object	Field	Transformation
D_CUSTOMER	ACCOUNT_CLASSES_DESC	Account	Account_Class_c	
D_CUSTOMER	ACCOUNT_MANAGER_ID	Account	Account_Manager_Route_c	
		Account	BillingCountry	Hardcode: United States

Figura 15 Sistemas y mapeado de datos. Fuente: Elaboración propia.

Sequence Diagrams

Customer Snowflake to Salesforce.

1. A scheduled batch is triggered via 'rh-scheduler-app' based on the job schedule in MuleSoft.
2. On trigger, MuleSoft retrieves the last successful runtime from Object Store to formulate the date-time range
3. The scheduled job then calls the Customer Master pAPI to initiate the data sync process
4. pAPI receives the date-time range and calls the Snowflake sAPI to retrieve the records created/updated in the given time range
5. If pAPI fails to retrieve the records, it logs an ERROR in the pAPI logs and the process stops
6. If pAPI is able to get the records from Snowflake, it then calls the Salesforce sAPI in async pattern using MuleSoft batch
7. Each record is then upserted in Salesforce and the response is persisted in temporary variables
8. Once all the records are done passing through the batch, pAPI then logs a report with the Record IDs and any failures with the respective request/response from Salesforce

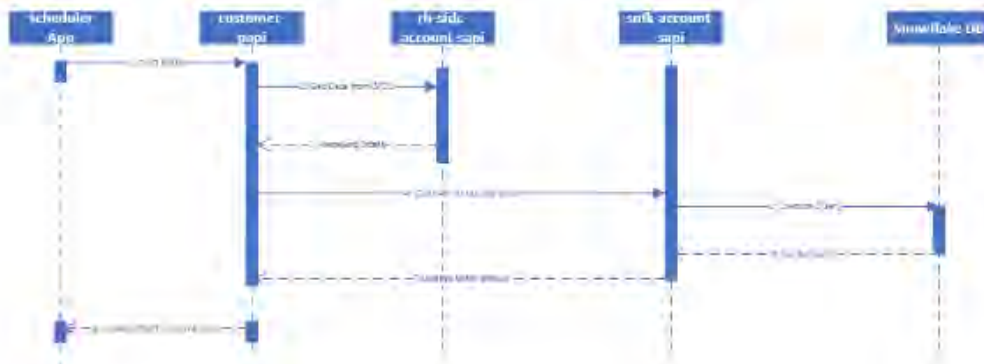


Figura 16 Diagrama de secuencia y sus pasos. Fuente: Elaboración propia.

- **Documentación de bajo nivel:** Este tipo de documentación, está más orientada hacia los desarrolladores, contiene información necesaria para poder ejecutar request, MUnits e información para poder hacer el despliegue de dichas aplicaciones. Esta documentación se espera que se encuentre dentro del archivo Readme del repositorio (ver Figura 17).

- Una imagen de los diagramas de secuencia.
- Información sobre cómo realizar pruebas, ya sea con los MUnits o a través de request utilizando los Postman Collection.
- Información sobre las ramas y en cual se debe trabajar, además de la información del pipeline que se debe utilizar para un despliegue.
- Información de los errores que ya se tiene contemplados en la aplicación, como una URL incorrecta, datos incorrectos, etc.
- Una tabla con información de links que también serán necesarios para el desarrollador.

The screenshot shows a 'Testing Details' page with a dark background. It features a table of MUnits and a code block for Properties.

Testing Details

MUnits

Name	Description
get-customer-supplier-error-handling-test.xml	
rh-snfk-customer-sapi-bad-request-test-suite.xml	MUnit to validate the Bad Request on the GET Request
rh-snfk-customer-sapi-test-suite.xml	MUnit to test GET Request and the Update

Properties

You will need to fill the next properties for the Munits.

```
vault:  
  key: "" #just a sample value  
  
snowflake:  
  url: ""  
  username: ""  
  password: ""  
  customer.attributes.filename: "customer_attributes.csv"  
  
databaseRetry:  
  maxRetry: ""  
  timesBetweenRetry: ""  
  
api:  
  autodiscovery: ""
```

Figura 17 Información de las pruebas MUnits. Fuente: Elaboración propia.

CAPÍTULO 5 MANAGED SERVICES

Uno de los servicios que ofrece AVIO Consulting es el de Managed Services, que se encarga del manejo de la infraestructura Mulesoft de los clientes.

Dentro de las actividades realizadas por el equipo se encuentran las siguientes:

Monitoreo diario

Dos veces al día, se revisan los logs de la infraestructura y aplicaciones Mulesoft en busca de errores. Se generará un ticket por cada error encontrado. La herramienta utilizada puede variar dependiendo de lo que tenga contratado cada cliente (ver Figura 18, Figura 19 y Figura 20). Los que más utilizamos son:

- Observe Inc.
- Kibana, que es parte del Elastic Stack.
- Anypoint Monitoring, herramienta de la plataforma de Mulesoft.

The screenshot shows the Observe tool interface for 'Daily Error Tracking Mulesoft'. The main view displays a table of logs with columns for timestamp, logdate, logger, thread, correlationid, and message. Below the table, a 'Definition' section shows a KQL query used for filtering the logs.

timestamp (CDT)	logdate (CDT)	logger	thread	correlationid	message
08/25/22 23:08:39.827	08/25/22 23:08:39.745	OnErrorPropagateHandler	[MuleRuntime].uber.117635...		
08/25/22 23:08:39.827	08/25/22 23:08:39.747	OnErrorPropagateHandler	[MuleRuntime].uber.117635...	23c8f879-2c63-422b-9d37-48...	
08/25/22 23:08:39.827	08/25/22 23:08:39.747	OnErrorPropagateHandler	[MuleRuntime].uber.117635...	23c8f879-2c63-422b-9d37-48...	
08/25/22 23:01:26.937	08/25/22 23:01:26.870	OnErrorPropagateHandler	[MuleRuntime].uber.115933...		
08/25/22 23:01:26.937	08/25/22 23:01:26.879	OnErrorPropagateHandler	[MuleRuntime].uber.115933...	bd719a81-8a86-4dc9-b392-b4...	
08/25/22 23:01:26.937	08/25/22 23:01:26.879	OnErrorPropagateHandler	[MuleRuntime].uber.115933...	bd719a81-8a86-4dc9-b392-b4...	
08/25/22 23:04:06.799	08/25/22 23:04:06.759	OnErrorPropagateHandler	[MuleRuntime].uber.117637...		
08/25/22 23:04:06.799	08/25/22 23:04:06.758	OnErrorPropagateHandler	[MuleRuntime].uber.117637...	84E37Eaa-1972-4208-B1F8-07...	
08/25/22 23:04:06.799	08/25/22 23:04:06.758	OnErrorPropagateHandler	[MuleRuntime].uber.117637...	84E37Eaa-1972-4208-B1F8-07...	
08/25/22 23:04:13.544	08/25/22 23:04:13.713	OnErrorPropagateHandler	http.Listener.62.SelectorRu...	3738a85f-9d84-43b6-bc84-ac...	

```

query container_name = "app"
extract_regex(log, /\{\{?P:date=Vd[-]\d{4}[-]\d{2}[-]\d{2}\}\}\s\{?P:time=Vd[-]\d{2}[-]\d{2}[-]\d{2}\}\}\s\{?P:level=(DEBUG|INFO|WARN|ERROR|FATAL|TRACE)\}\s\{?P:logger=[[:alpha:]]\}\s\{?P:thread=[-]*\}\s\{event:\s\{?P:correlationid=[-]*\}\}\s\{?P:message=[-]*\}\}
make_cal logdate: parse_iso8601(date + "T" + time + "Z")
make_cal app:string(labels.app)
filter level = "ERROR"
filter message != "Unknown error"
filter not match_regex(thread, /gizly/)
filter not match_regex(thread, /OnErrorPropagateHandler/)
//filter match_regex(log, /Error Failed to throw ADP/)
  
```

Figura 18 Monitoreo con la herramienta Observe. Fuente: Elaboración propia.

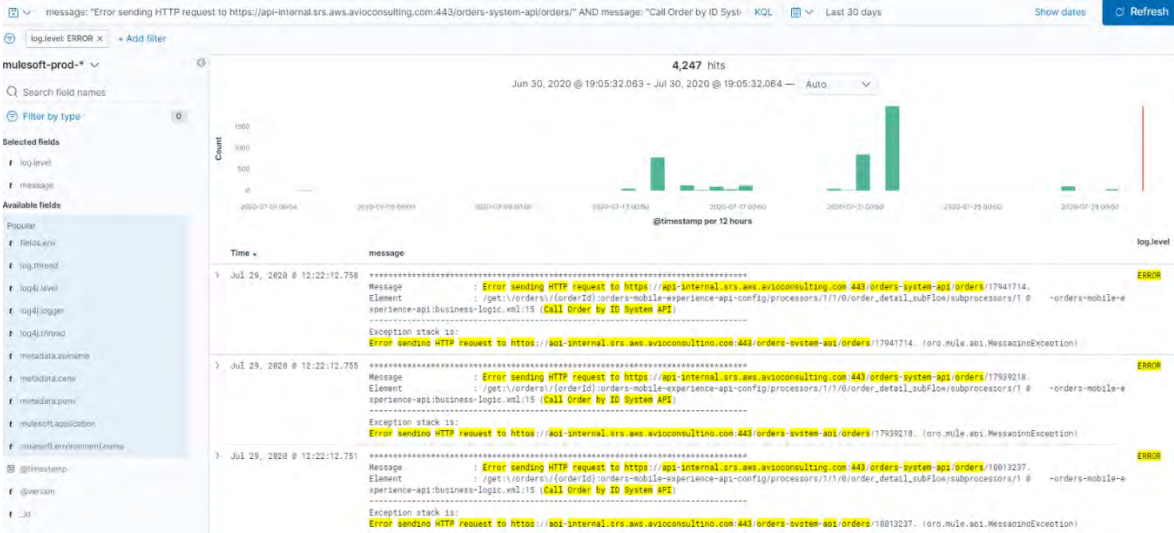


Figura 19 Monitoreo con la herramienta Kibana. Fuente: Elaboración propia.

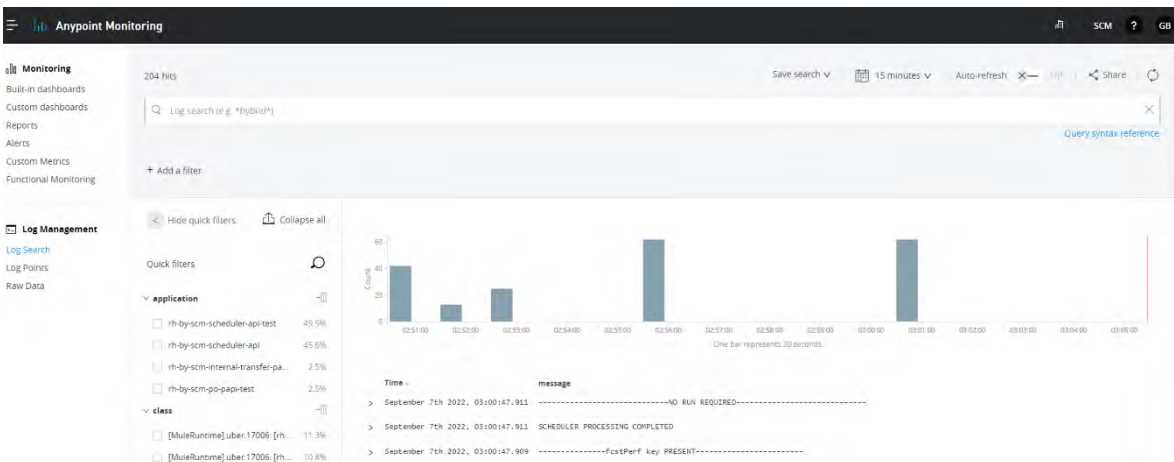


Figura 20 Monitoreo con la herramienta Anypoint Monitoring. Fuente: Elaboración propia.

Análisis de la causa raíz de un incidente en una aplicación

Para cada uno de los tickets de incidentes abiertos, ya sea por el cliente o por el equipo durante el monitoreo diario de errores, debemos determinar la causa raíz del incidente.

Para determinar la causa raíz del incidente, se debe de analizar el mensaje de error para determinar en qué aplicación y en qué procesador se generó el error. Con esa información, clonamos el código de la aplicación y se rastrea el error hasta la causa raíz utilizando el Anypoint Studio.

La resolución del ticket puede ser uno de los siguientes escenarios:

- **Problema de datos**

A veces, los datos incorrectos que se envían a la aplicación provocan errores. Muchas veces, esto se puede manejar mediante una mejor validación de datos (y se incluiría en la categoría de Cambio de código necesario). A veces, los datos pasan validaciones, pero cuando se envían a sistemas externos, causan un error. En el caso de este tipo de error, el cliente decidirá qué quiere que suceda. Puede ser que el error deba registrarse en un nivel de registro más bajo, como WARN o INFO; o podría ser que realmente sea un error y deban ser notificados cuando suceda.

- **Cambio de código necesario**

Si es necesario un cambio de código para corregir el problema, señale la parte de la aplicación que debe actualizarse y qué debe actualizarse.

- **Otro sistema necesita ser investigado**

A veces, los problemas en Mulesoft son causados por sistemas externos con los que se comunica la aplicación. Puede ser que el error haya sido un tiempo de espera en la comunicación o que haya un error devuelto desde el sistema externo. En cualquier caso, es posible que se deba investigar el otro sistema para encontrar la causa raíz del error.

También, es posible que se deba aumentar un valor de tiempo de espera en la aplicación Mulesoft para manejar el problema. En ese caso, sería necesario un cambio de código.

- **Recomendación emitida**

Si ya se ha dado una recomendación sobre el error, deberemos señalar esa recomendación en el ticket. También debemos agregar el error a la lista de errores que filtramos de nuestro monitoreo diario.

DOCUMENTACIÓN DE INCIDENCIAS

Cuando ya se ha dado la resolución del ticket se deberá generar un documento con la información recabada del análisis. Este documento se generará para cada una de las incidencias con la siguiente estructura (ver Figura 21, Figura 22 y Figura 23):

- **Descripción del incidente:** Se describe de manera breve el incidente, y las aplicaciones y/o sistemas involucrados.
- **Mensajes de error:** Se agregan todos los mensajes de errores relacionados a esta incidencia.
- **Causa raíz:** Se detalla toda la información encontrada respecto al incidente, las aplicaciones y sistemas involucrados. Indicando cual fue la causa que provocó el incidente.
- **Solución ofrecida:** Finalmente se agrega la solución ofrecida para el cliente, tomando en cuenta los criterios de resolución previamente mencionados.

Problem

- i This error message is thrown when srs-branch-mobile-experience-api makes a request to srs-branch-system-api and the response timeout is exceeded.

Related error

Affected endpoint: get:/branch/customer/{customerNumber} in -branch-mobile-experience-api

```

*****
Message           : Response code 408 mapped as failure.
Element           : /get:/brandList:branch-mobile-experience-api-
config/processors/0/brand-list-apiFlow/subprocessors/1 @ -branch-mobile-experience-
api:business-logic.xml:16 (Call Brand List System API)
-----
Exception stack is:
Response code 408 mapped as failure.
(org.mule.module.http.internal.request.ResponseValidatorException)
org.mule.module.http.internal.request.SuccessStatusCodeValidator.validate(SuccessStatusCodeV
alidator.java:37)
org.mule.module.http.internal.request.DefaultHttpRequester.validateResponse(DefaultHttpReque
ster.java:438)
org.mule.module.http.internal.request.DefaultHttpRequester.innerProcess(DefaultHttpRequester
.java:421)
(202 more...)

(set debug level logging or '-Dmule.verbose.exceptions=true' for everything)
*****

```

Figura 21 Descripción y mensajes del incidente. Fuente: Elaboración propia.

Root cause

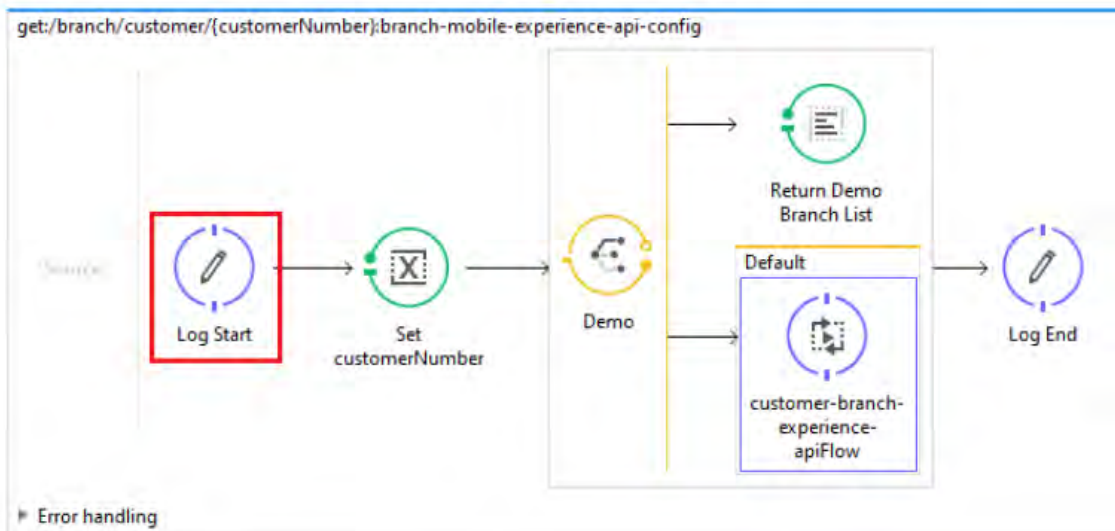
- The determination of the root cause is based on the trace between documents in Kibana and mule code.

The correlationId used in this example is 8b46a029-53eb-4582-a53b-b615c420446f from the September 12th, 2020.

Branch Mobile Experience API

- `get:/branch/customer/{customerNumber};branch-mobile-experience-api-config`

1. The request started in the main flow logged the message: **Started Processing Request Delivery Details for Customer Number RLG535 , Correlation GUID: 8b46a029-53eb-4582-a53b-b615c420446f.**



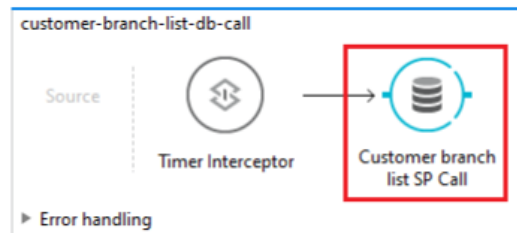
- `customer-branch-experience-apiFlow`

2. Then it navigates to this flow, and here is logged a new log message: **Calling System API for Branch List. Correlation Guid: 8b46a029-53eb-4582-a53b-b615c420446f.**

Figura 22 Causa raíz del incidente. Fuente: Elaboración propia.

- **customer-branch-list-db-call**

5. In this flow, a database does an operation, which generated a timeout due to the processing time on the database was 983060 ms (More 16 minutes), it's mean that exceeded the *Response Timeout* defined (10 minutes). See properties in point 3.



```
1 Sep 12, 2020 @ 18:30:31.348 Flow : customer-branch-list-db-call Processing time : 983060 ms
```

Solution Offered

- MS team saw that issue appeared for the first time September 12th, 2020, and that since then it has not been very frequent. However, in the case to find it again, if they require, we can ask SRS team to consider the option to increase the response time in the requester to respond in considerable time and reduce the times out.

Figura 23 Solución ofrecida para evitar la incidencia. Fuente: Elaboración propia.

AUTOMATIZACIÓN DE ACTIVIDADES

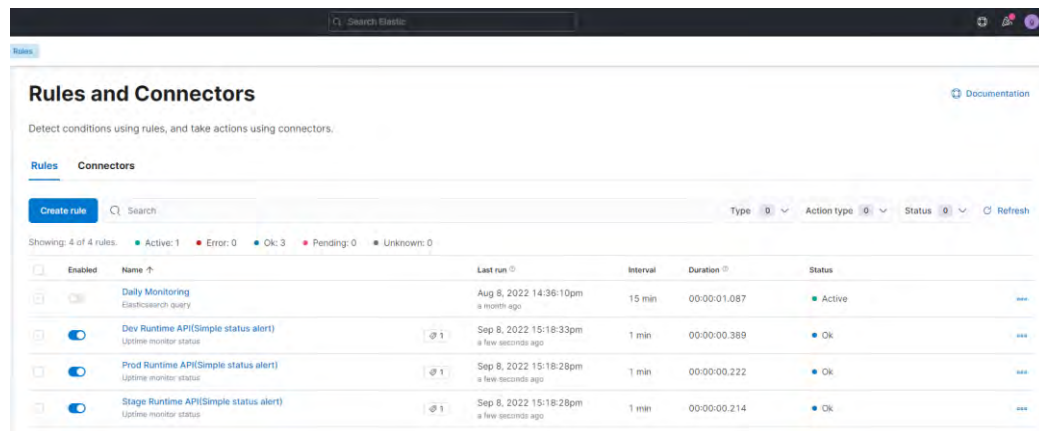
Como parte de las actividades del equipo de Managed Services, se encuentra la automatización de actividades. Esto consiste en encontrar soluciones para que toda aquella actividad que sea manual se pueda automatizar, ayudando a optimizar mejor los tiempos del equipo.

Estas automatizaciones, en su mayoría, se tratan de hacer de tal manera que nos puedan servir para trabajar con cualquier cliente y se puedan adaptar a sus herramientas.

Algunas de las automatizaciones que hemos hecho son las siguientes:

- **Generación automática de ticket en Jira:** Esta automatización es para agilizar el monitoreo diario, consiste en conectar el servicio de visualización de datos con Jira para la gestión de tickets. Kibana al ser una herramienta más completa, cuenta con conectores y reglas que permite esta automatización (ver Figura 24).

Esto se logra a través de una regla, si encuentra el mismo mensaje de error con una concurrencia de 10, en un periodo de 15 minutos, se generará un ticket nuevo.



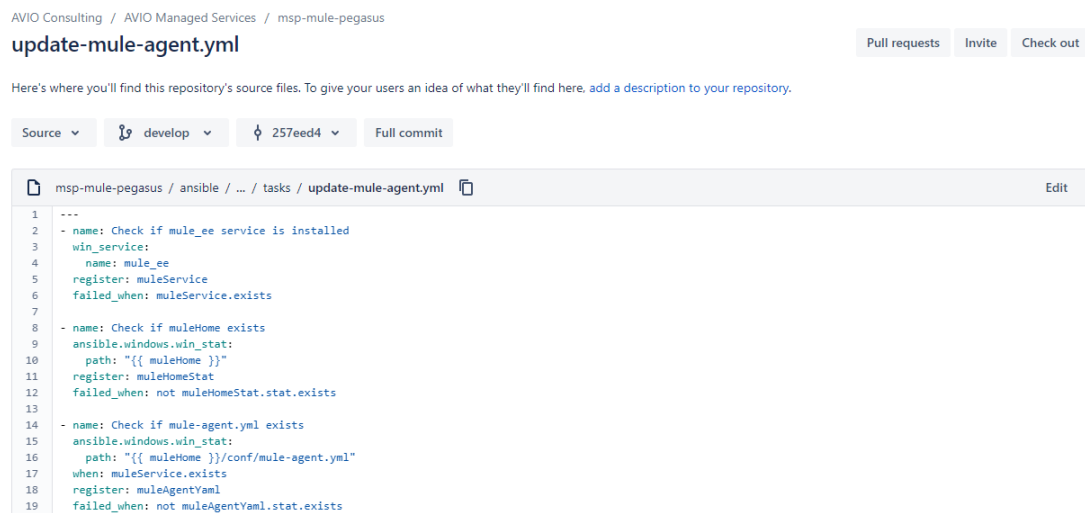
The screenshot shows the 'Rules and Connectors' page in Kibana. It displays a table of rules with columns for 'Enabled', 'Name', 'Last run', 'Interval', 'Duration', and 'Status'. There are four rules listed, all with a status of 'Ok'.

Enabled	Name	Last run	Interval	Duration	Status
<input type="checkbox"/>	Daily Monitoring Elasticsearch query	Aug 8, 2022 14:36:10pm a month ago	15 min	00:00:01.087	Active
<input checked="" type="checkbox"/>	Dev Runtime API(Simple status alert) Uptime monitor status	Sep 8, 2022 15:18:33pm a few seconds ago	1 min	00:00:00.389	Ok
<input checked="" type="checkbox"/>	Prod Runtime API(Simple status alert) Uptime monitor status	Sep 8, 2022 15:18:28pm a few seconds ago	1 min	00:00:00.222	Ok
<input checked="" type="checkbox"/>	Stage Runtime API(Simple status alert) Uptime monitor status	Sep 8, 2022 15:18:28pm a few seconds ago	1 min	00:00:00.214	Ok

Figura 24 Regla para el monitoreo diario. Fuente: Elaboración propia.

- **Instalación de software utilizando Ansible:** Cada mes, se deberá actualizar el software a sus versiones más recientes. En lugar de hacer esto

uno por uno, se generaron *tasks* en Ansible que se encargan de actualizar las versiones de manera automática. De manera inicial, este proceso estaba únicamente para servidores Windows, ahora se agregó un apartado para servidores Linux (ver Figura 25).



The screenshot shows a GitHub repository page for 'update-mule-agent.yml'. The repository is named 'msp-mule-pegasus' and is located under 'AVIO Consulting / AVIO Managed Services / msp-mule-pegasus'. The file path is 'ansible / ... / tasks / update-mule-agent.yml'. The code is displayed in a light blue editor with line numbers from 1 to 19. The code is an Ansible task that checks for the installation of the mule-agent.yml file on Windows servers. It includes a 'win_service' section to check if the mule_ee service is installed, an 'ansible.windows.win_stat' section to check if the muleHome directory exists, and another 'ansible.windows.win_stat' section to check if the mule-agent.yml file exists in the muleHome directory. The task is registered as 'muleAgentYaml' and fails if the file does not exist.

```

1 ---
2 - name: Check if mule_ee service is installed
3   win_service:
4     name: mule_ee
5     register: muleService
6     failed_when: muleService.exists
7
8 - name: Check if muleHome exists
9   ansible.windows.win_stat:
10    path: "{{ muleHome }}"
11    register: muleHomeStat
12    failed_when: not muleHomeStat.stat.exists
13
14 - name: Check if mule-agent.yml exists
15   ansible.windows.win_stat:
16    path: "{{ muleHome }}/conf/mule-agent.yml"
17    when: muleService.exists
18    register: muleAgentYaml
19    failed_when: not muleAgentYaml.stat.exists

```

Figura 25 Código para actualizar la versión del agente Mule. Fuente: Elaboración propia.

- **Notificación de los Releases de Mulesoft:** Mulesoft cuenta con un apartado de releases por mes, esto abarcan la información de actualizaciones desde el Runtime hasta conectores. La página no cuenta con algún servicio de notificación por correo o RSS para su distribución.

Se desarrolló una API utilizando Mulesoft, se encarga de transformar la información de los releases a un formato de distribución. Además, se utilizó Zapier como servicio de integración, que se encarga de mandar las notificaciones a un canal de Slack, herramienta de comunicación que utiliza AVIO Consulting (ver Figura 26).


```

%dw 2.0
output application/xml

fun returnPathIfAvailable(line) =
  if (line contains "xref")
    (line splitBy ("\\"))[1]
  else
    null

fun returnTitlesIfAvailable(line) =
  if (line contains "xref")
    (line splitBy (/[<\>\/]))[4]
  else
    null

var paths = ((payload splitBy ("\n") map (line, index) -> {
  path: returnPathIfAvailable(line)
}) filter ($.path != null)).path

var titles = ((payload splitBy ("\n") map (line, index) -> {
  title: returnTitlesIfAvailable(line)
}) filter ($.title != null)).title

---
{rss @(version: "2.0"):{
  channel: {
    title: "Release Note Summary by Month",
    link: "https://docs.mulesoft.com/release-notes/quick-refs/by-date-index",
    description: "Release Note Summary by Month",
    item: titles[0 to 9] map ((item, index) -> {
      title: item,
      link: paths[index] replace ".." with("https://docs.mulesoft.com/release-notes"),
      description: "Release Note of " ++ item
    })
  }
}}

```

Figura 26 Código para transformar un HTML a formato RSS. Fuente: Elaboración propia.

CAPÍTULO 6 EXPERT ON DEMAND

DESARROLLO DE NUEVAS FUNCIONALIDADES Y CORRECCIONES

El equipo de Expert on Demand, se encarga de dar soporte a los clientes con sus nuevos desarrollos o correcciones de lo que ya han desarrollado. AVIO Consulting cuenta con su documentación sobre buenas prácticas de desarrollo para aplicaciones Mulesoft.

API/RAML Design

Para el desarrollo del RAML se guían de los siguientes principios:

- **Satisface los requisitos comerciales:** Obviamente, la especificación RAML debe satisfacer funcionalmente los requisitos comerciales.
- **Sigue los estándares de nomenclatura acordados en todos los equipos:** EL API specification en RAML debe seguir los estándares de nomenclatura acordados por todos los equipos del proyecto.
- **Documenta minuciosamente los endpoints de la API:** Cuando la API debe ser revisada por el equipo de desarrollo en Exchange, sus endpoints de interfaz deben documentarse explícitamente (usando definiciones de tipos de datos y ejemplos).
- **Hace uso de artefactos publicados:** Reutiliza artefactos (fragmentos de código) publicados en Exchange. Este es el primer paso para producir una API más corta y legible, incorporando el uso de librerías para extraer ciertas definiciones de RAML y no saturar el API RAML.
- **Maximizar el uso de fragmentos de tipos de datos:** Los archivos RAML de tipos de datos que contienen definiciones de tipos de datos, documentan los tipos de datos que se esperan recibir y lo que debe responder, esto para todos los endpoints del API RAML.
- **El archivo principal RAML es conciso y legible:** Si se siguen los principios anteriores, entonces un archivo RAML principal será más corto y fácil de entender.

Logging

Al hacer el registro de actividad (logging) dentro de una aplicación Mulesoft, se puede proporcionar información valiosa sobre una API en tiempo de ejecución. Si se hace correctamente, permitirá la trazabilidad de una transacción, el tiempo y la duración del proceso, y registrará cualquier error que ocurra.

- **Qué registrar:** Identificadores de transacción que puedan identificar de forma única cada ejecución de la API. Nombres de la aplicación y versión; esto generalmente se define como un marcador de posición de propiedad. Registros de errores informativos y bien escritos.
- **Qué no registrar:** Información sensible con contraseñas, números de tarjetas de crédito, números de seguro social, entre otros. Tampoco se deberán registrar *payloads* muy largos; es decir, aquellos que contengan información binaria como información de imágenes.
- **Cuando utilizar estos registros:** Se puede registrar inicio y fin de la ejecución de una API, inicio y fin de un flujo, antes y después de una llamada a un servicio externo.
- **Como escribir estos registros:** Todos los registros se crean utilizando el *framework* Log4j. Nos debemos de asegurar de especificar el nivel y categoría apropiada para cada registro. Estos son los niveles válidos y una descripción de cuando se debe de utilizar cada uno:
 - **TRACE:** Eventos informativos más detallados que DEBUG.
 - **DEBUG:** Eventos informativos detallados que son más útiles para depurar una aplicación.
 - **INFO:** Eventos informativos que resaltan el progreso de la aplicación.
 - **ERROR:** Eventos de error que aún podrían permitir que la aplicación continúe ejecutándose.

Error Handling

Hay dos enfoques distintos para consolidar el manejo de errores en una aplicación Mulesoft. Ambos enfoques son considerados como buenas prácticas por AVIO Consulting.

- **Controlador de errores global:** Deberá ser declarado, para esto, se debe agregar como una propiedad de configuración de elemento global (ver Figura 27).

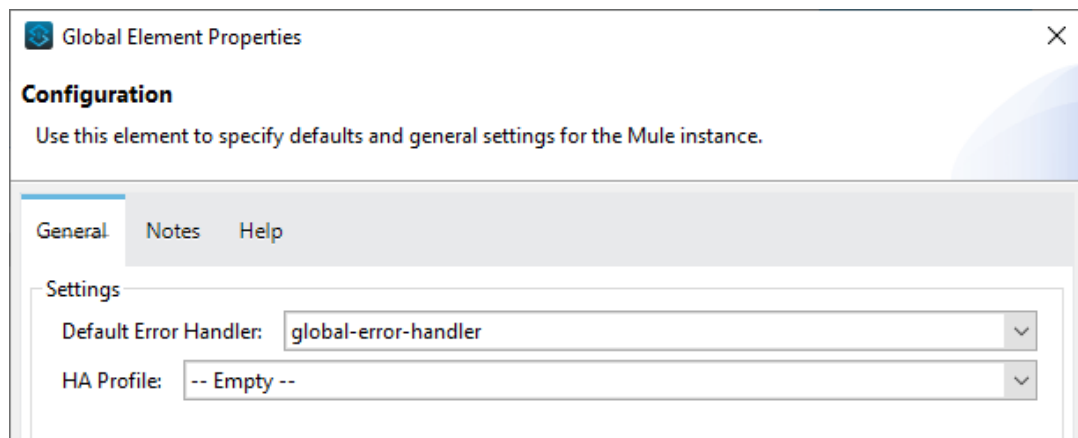


Figura 27 Configuración global. Fuente: Elaboración propia.

- **Definir el manejo de errores dentro de los flujos principales:** Los flujos principales con manejo de errores son la parada final en cualquier propagación de manejo de errores y nunca se propagarán a un controlador de errores global, incluso si uno está definido en la aplicación Mule.

Dataweave

Dataweave fue introducido por primera vez con Mule 3 (DataWeave 1.0). Provee una manera de hacer transformaciones mediante scripts que no requieren de lenguajes como Java o Groovy. Hubo una actualización significativa de DataWeave cuando se introdujo Mule 4 (DataWeave 2.0).

- **Reusable:** En lugar de copiar funciones de una transformación a otra, se pueden agregar a archivos DWL. Esto los hace más sencillo la refactorización en caso de ser necesario.

- **Organizado:** Los archivos DWL deben ser nombrados de manera apropiada, dando alguna indicación del origen y la finalidad de esa transformación.
- **Librería de funciones:** Antes de escribir una función hay que asegurarnos de que no exista alguna librería que con lo que estamos buscando.

PRUEBAS UNITARIAS MUNIT

El concepto de la prueba unitaria es validar la corrección de una unidad individual de código fuente. Una unidad individual de código fuente es la parte comprobable más pequeña de una aplicación. En el caso de las aplicaciones Mulesoft, la parte comprobable más pequeña de una aplicación es un flujo o subflujo.

El objetivo de las pruebas unitarias es garantizar que todas las funciones y características de una sola unidad de código funcionen como se especifica en el diseño. Las pruebas se realizan de forma aislada utilizando interfaces simuladas para sistemas y/o recursos externos. Los procedimientos de prueba unitaria consisten en:

- Identificación de unidades de código.
- Creación de condiciones previas y posteriores para cada unidad.
- Definir pruebas MUnit para cada unidad.
- Ejecutar las pruebas de MUnit durante el desarrollo.
- Informar y revisar los resultados de las pruebas unitarias.

Se debe desarrollar un conjunto de pruebas de MUnit para que se ejecute bajo demanda durante el desarrollo, así como también, una secuencia de pruebas de regresión de integración continua.

El desarrollador de API es responsable del desarrollo y ejecución de las pruebas de MUnit y de la creación de conjuntos de pruebas de MUnit.

Para las pruebas Munit, se consideran los siguientes puntos como buenas prácticas:

- Hacer pruebas de escenarios positivos y negativos.

- Indicar en el título de la prueba Munit, la condición de prueba o la condición de error.
- Imitar todas las llamadas a sistemas externos.
- Ejecutar las pruebas Munit desde la línea de comandos.
- Únicamente empuje las pruebas Munit que funcionen de manera exitosa, sin errores.
- Organizar los conjuntos de pruebas, ya sea por recurso o por funcionalidad.
- Asegurarse de que todas las dependencias de las pruebas Munit se encuentre en el POM file.

PRIORIDAD DE TICKETS

Los tickets de Expert on Demand pueden tener 4 prioridades. La prioridad del ticket debe verificarse antes de ser aceptado, si el cliente lo marca como una prioridad más alta que la permitida, la prioridad deberá reducirse y proporcionar una explicación al cliente.

Crítico - Prioridad 1 (P1)

Una incidencia crítica en el entorno de Producción que debe ser resuelta con urgencia. Los tickets con esta prioridad son para interrupciones de una o más aplicaciones o componentes en producción causados por un incidente en tiempo real. Ejemplo:

- Entorno de producción no disponible.

Mayor - Prioridad 2 (P2)

Un problema crítico que resulta en el uso restringido de las aplicaciones y afecta a los usuarios comerciales del cliente. El problema debe estar afectando en tiempo real al negocio. Ejemplos:

- Errores en producción que impiden la correcta ejecución de una aplicación.
- Errores en el entorno de control de calidad (QA) que bloquean el lanzamiento de producción.

Menor - Prioridad 3 (P3)

Cuando los usuarios comerciales del cliente se ven afectados, pero el negocio aún está operativo. La resolución puede programarse y/o no se anticipa un impacto comercial en un futuro cercano.

Ejemplos:

- Tareas de desarrollo.
- Errores en el entorno de control de calidad (QA) que no bloquean el lanzamiento de producción.
- Errores en el entorno de desarrollo.

Trivial - Prioridad 4 (P4)

Número único o limitado de usuarios comerciales afectados y el impacto en el negocio no se prevé en un futuro cercano.

Ejemplos:

- La mayoría de los errores de encontrados durante el monitoreo diario.
- Tareas de desarrollo.

STATUS DE LOS TICKETS

Cuando un cliente genera un nuevo ticket, este es el proceso que seguirá, y los pasos que se deberán de hacer en cada paso del proceso (ver Figura 28).

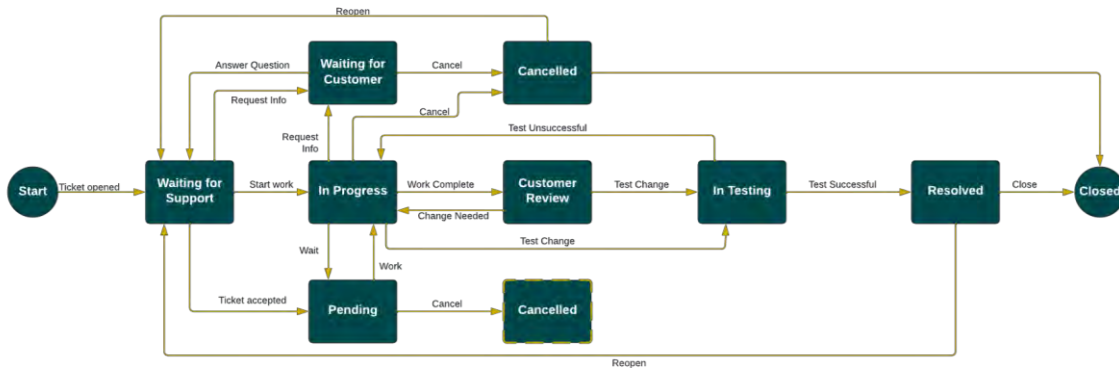


Figura 28 Workflow del status de un ticket. Fuente: Elaboración propia.

Waiting for support

Al inicio del proceso, el cliente ha abierto el ticket y pasa a estado **Waiting for support**. En este momento, estas son las acciones que deben tomarse:

- Leer la descripción del ticket.
- Determinar si hay preguntas adicionales cuyas respuestas necesitamos para comenzar el trabajo.
- El ticket se asignará a un miembro del equipo, designado por el líder del equipo.
- Calcular el tiempo que llevará completar el ticket (en horas) y agregarlo al campo **Estimate**.

Si hay más preguntas, se deberá cambiar el estado del ticket a **Waiting for customer** y agregue las preguntas al ticket como comentarios para que el cliente pueda responderlas.

Si hay suficiente descripción para comenzar a trabajar en el ticket, se moverá a **In progress** y se comenzará a trabajar en el ticket.

In Progress

Un ticket solo debe estar en el estado **In Progress** cuando se está trabajando activamente. Si se comienza a trabajar en otra tarea, el ticket se moverá al estado **Pending**.

Una vez que se haya completado el trabajo para el ticket, se moverá al estado **Customer Review** para que el cliente pueda revisar los cambios. Si la solicitud involucró cambios de código en una o más aplicaciones, se deben de incluir enlaces de los pull requests en los comentarios del ticket.

Waiting for customer

Un ticket pasará al estado **Waiting for customer** cuando se necesite información adicional del cliente para continuar o comenzar a trabajar en el ticket. Una vez que el cliente proporcione la información adicional, el ticket volverá a **Waiting for support**.

Pending

Cuando el ticket está en estado **Pending**, significa que no se está trabajando en él. El ticket se puede mover a **In progress**, una vez que el trabajo comienza o continúa en el ticket, o al estado **Cancelled** si ya no se necesita el ticket.

Cancelled

El ticket se pone en estado **Cancelled** si el trabajo ya no es necesario. Desde **Cancelled**, se puede volver al estado **Waiting for support** o a **Closed**. El ticket pasará automáticamente a **Closed** después de 7 días.

Customer Review

En el estado **Customer review**, el cliente revisa el ticket para validar que la respuesta es satisfactoria para cumplir con la solicitud. La respuesta podría ser una respuesta a una pregunta, o la entrega de cambios de correcciones o código nuevo. Si la respuesta es algo que se puede probar, el cliente puede mover el ticket al estado **In testing** mientras se realiza la prueba.

In Testing

En el estado **In Testing**, el cliente prueba la respuesta al ticket en el ambiente del cliente. Si se encuentran problemas, el ticket se puede regresar al estado **In progress**. Si la prueba es exitosa y el trabajo está completo, el ticket debe pasar al estado **Resolved**.

Resolved

En el estado **Resolved**, el trabajo en el ticket se considera completo, pero el ticket se puede volver a abrir si se encuentra un problema con la resolución. Una vez que el ticket haya estado en estado Resuelto durante una semana, pasará al estado **Closed** automáticamente.

Closed

En el estado **Closed**, todo el trabajo está completo en el ticket. Si se encuentra un problema con la resolución del ticket, se deberá abrir un nuevo ticket.

CAPÍTULO 7 CONCLUSIONES

Estar laborando con AVIO Consulting para los equipos de Managed Services y Expert on Demand durante los últimos dos años, me han brindado la oportunidad de desarrollarme profesionalmente, me ha permitido obtener mi primer triunfo en mi carrera profesional que fue certificarme como Mulesoft Certified Developer – Level 1, y posteriormente poder renovarlo por otros dos años.

Ha sido una gran oportunidad en la cual he puesto en práctica los conocimientos adquiridos en las aulas de esta universidad, de materias como Programación, Administración de sistemas operativos de red, Sistemas operativos basados en Unix y Gestión de proyectos tecnológicos.

Uno de los mayores retos durante mi carrera profesional, fue descubrir que muchas de las herramientas existentes para funciones de TI, están cerradas a un proveedor en específico. Esto en un principio, me orilló a aprender lo esencial de varias herramientas para poder administrarlas y/o utilizarlas.

Tomando esto en cuenta, el equipo de Managed Services se centró en encontrar herramientas con las cuales se pueda trabajar con más de un proveedor, para poder gestionar, administrar y automatizar aplicaciones y servicios. Además, se encargó de ofrecer capacitación de estas herramientas para los nuevos integrantes del equipo, y de esta manera se pudieran integrar y realizar las actividades de manera eficiente.

De igual manera, al ser parte del equipo de Expert on Demand, me permitió aprender de las buenas prácticas de desarrollo Mulesoft por parte de AVIO Consulting, al implementarlas se logra disminuir las situaciones retrabajo en el desarrollo de las aplicaciones.

Finalmente, el trabajar en AVIO Consulting, me permitió aprender de situaciones reales que suceden en el trabajo y como adaptarme a ellas. Me ayudó a desarrollar aptitudes como la resiliencia y la disciplina, además de actitudes como la adaptabilidad para poder aprender múltiples herramientas.

BIBLIOGRAFÍA

- AVIO Consulting. (2021). *AVIO Consulting Company*. Recuperado el 12 de Septiembre de 2022, de <https://www.avioconsulting.com/company>
- AVIO Consulting. (2021). *AVIO Consulting Expert on Demand*. Recuperado el 12 de Septiembre de 2022, de <https://www.avioconsulting.com/mulesoft-support-services>
- AVIO Consulting. (2021). *AVIO Consulting Managed Services*. Recuperado el 12 de Septiembre de 2022, de <https://www.avioconsulting.com/managed-services>
- Mas, A., Mesquida, A. L., & Amengual, E. (2009). La madurez de los servicios TI. *Revista Española de Innovación*, 77-87. Recuperado el 12 de Septiembre de 2022, de <https://www.redalyc.org/articulo.oa?id=92217153011>
- Morgan, T. (2010). *The real value of Managed Services: More than just “keeping the lights on”*. Recuperado el 12 de Septiembre de 2022, de <https://www.ciena.com/insights/articles/the-real-value-of-managed-services-more-than-just-keeping-the-lights-on.html>
- Mulesoft. (2022). *MuleSoft Certified Developer - Level 1*. Recuperado el 12 de Septiembre de 2022, de <https://training.mulesoft.com/certification/developer-mule4-level1>
- Uzialko, A. (2022). *Pros and Cons of an IT Managed Service Provider for a Small Business*. Recuperado el 12 de Septiembre de 2022, de <https://www.businessnewsdaily.com/5115-managed-services.html>