



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Migración de un sistema on-premise de administración
de recursos humanos hacia una plataforma en la nube
utilizando la metodología de desarrollo Devops para el
sector de las pymes**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

José Anthony VILLASANTE MORENO

ASESOR

Jorge Santiago Pantoja Collantes

Lima, Perú

2022



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

Villasante, J. (2022). *Migración de un sistema on-premise de administración de recursos humanos hacia una plataforma en la nube utilizando la metodología de desarrollo Devops para el sector de las pymes*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	JOSÉ ANTHONY VILLASANTE MORENO
Tipo de documento de identidad	DNI
Número de documento de identidad	46085860
URL de ORCID	https://orcid.org/0000-0002-4037-6200
Datos de asesor	
Nombres y apellidos	Jorge Santiago Pantoja Collantes
Tipo de documento de identidad	DNI
Número de documento de identidad	06254022
URL de ORCID	https://orcid.org/0000-0002-7172-1206
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cayo Victor León Fernández
Tipo de documento	DNI
Número de documento de identidad	07001405
Miembro del jurado 1	
Nombres y apellidos	José César Piedra Isusqui
Tipo de documento	DNI
Número de documento de identidad	25628915
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	2022
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 Sistemas

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 14:00 horas del día 23 de agosto del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Piedra Isusqui José César (Presidente), Dr. León Fernández Cayo Víctor (Miembro) y el Lic. Pantoja Collantes Jorge Santiago (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/hqu-jkha-bft>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“MIGRACIÓN DE UN SISTEMA ON-PREMISE DE ADMINISTRACIÓN DE RECURSOS HUMANOS HACIA UNA PLATAFORMA EN LA NUBE UTILIZANDO LA METODOLOGÍA DE DESARROLLO DEVOPS PARA EL SECTOR DE LAS PYMES”**, por el Bachiller **Villasante Moreno José Anthony**; para obtener el Título Profesional de Ingeniero de Sistemas.

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 **17 (DIECISIETE)**.

A continuación el Presidente de Jurados el Mg. Piedra Isusqui José César, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 15.00 horas, se levantó la sesión.

Presidente

Mg. Piedra Isusqui José César

Miembro

Dr. León Fernández Cayo Víctor

Miembro Asesor

Lic. Pantoja Collantes Jorge Santiago



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

INFORME DE EVALUACIÓN DE ORIGINALIDAD

1. Facultad de Ingeniería de Sistemas e Informática
 2. Escuela Profesional de Ingeniería de Sistemas
 3. Autoridad académica que emite el informe de originalidad
Directora (e) de la EPIS
 4. Apellidos y Nombres de la autoridad académica
Dra. Luzmila Elisa Pró Concepción
 5. Operador del programa informático de similitudes
Dra. Luzmila Elisa Pró Concepción
 6. Documento evaluado
Título de pregrado: "Migración de un sistema On-Premise de administración de recursos humanos hacia una plataforma en la nube utilizando la metodología de desarrollo Devops para el sector de las PYMES"
 7. Autor del documento
Bach. Villasante Moreno, José Anthony
 8. Fecha de recepción del documento **23/09/2022**
 9. Fecha de aplicación del programa informático de similitudes **23/09/2022**
 10. Software utilizado
 - Turnitin
 11. Configuración del programa detector de similitudes
 - Excluye textos entrecomillados
 - Excluye bibliografía
 - Excluye cadenas menores a 40 palabras
 12. Porcentaje de similitudes según programa detector de similitudes **4 (cuatro) %**
 13. Fuentes originales de las similitudes encontradas
Se adjunta en el anexo 1
 14. Observaciones
-
15. Calificación de originalidad
 - Documento cumple criterios de originalidad, sin observaciones
 - Documento cumple criterios de originalidad, con observaciones
 - Documento no cumple criterios de originalidad
 16. Fecha de informe **24/09/2022**



Firmado digitalmente por PRO
CONOCEPCION Luzmila Elisa FAU
20148092282 soft
Motivo: Soy el autor del documento
Fecha: 14.11.2022 23:15:51 -05:00

Firma de evaluador
Dra. Luzmila E. Pró Concepción
Directora (e) de la EPIS



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

ANEXO 1

Fuentes originales de las similitudes encontradas

1. hdl.handle.net: 4%



UNMSM

Firmado digitalmente por PRO
CONCEPCION Luzmila Elisa FAU
20148092282 soft
Motivo: Soy el autor del documento
Fecha: 14.11.2022 23:15:23 -05:00

Firma de evaluador

Dra. Luzmila E. Pró Concepción

Director (e) de la EPIS

DEDICATORIA

El siguiente trabajo se lo dedico a mi padre Jose y a mi madre Elizabeth porque sacaron adelante a mis hermanos y a mí, sacrificando su tiempo y su juventud para que podamos salir adelante en la vida.

AGRADECIMIENTOS

Agradezco a las personas y las empresas que me dieron la oportunidad de crecer y trabajar, pues gracias a la experiencia obtenida pude poner en práctica mis conocimientos obtenidos.

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Migración de un Sistema On-Premise de administración de recursos humanos
hacia una plataforma en la nube utilizando la metodología de desarrollo
DevOps para el sector de las PYMES**

Autor: Villasante Moreno, José Anthony
Asesor: Pantoja Collantes, Jorge Santiago
Título: Trabajo de Suficiencia Profesional para optar el Título
Profesional de Ingeniero de Sistemas
Fecha: Julio 2022

RESUMEN

El actual trabajo de suficiencia profesional describe el proceso de migración de un Sistema de Administración de Recursos Humanos y la adaptación de esta compañía de Desarrollo de Software especializado en Recursos Humanos hacia una plataforma basada totalmente en la nube.

El objetivo de este proyecto ha sido reducir los tiempos de la creación de los entornos de cada nuevo cliente y realizar una transformación digital en los procesos utilizando nuevas tecnologías de desarrollo, además se ha generado una nueva interfaz al usuario para que pueda ayudar a mejorar la experiencia con este sistema.

Para cumplir los objetivos mencionados se ha visto necesario utilizar el marco de trabajo de Scrum y la metodología DevOps con la nube Azure de Microsoft, así pues, combinando ambas metodologías se ha dado como resultado un sistema con procesos automatizados y centralizados en la nube.

Finalmente, se concluye con la entrega del sistema migrado a la nube y con procesos automatizados.

Palabras claves: Nube, Scrum, DevOps, Azure, Automatización.

**MAJOR NATIONAL UNIVERSITY OF SAN MARCOS
FACULTY OF SYSTEMS AND COMPUTER ENGINEERING
PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING**

Migration of an On-Premise human resources management system to a cloud platform using the DevOps development methodology for the PYMES sector

Author: Villasante Moreno, José Anthony

Advisor: Pantoja Collantes, Jorge Santiago

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

Date: July 2022

ABSTRACT

The current work of professional sufficiency describes the migration process of a Human Resources Administration System and the adaptation of this company of Software Development specialized in Human Resources to a platform based entirely in the cloud.

The objective of this project has been to reduce the time of the creation of the environments of each new client and make a digital transformation in the processes using new development technologies, in addition a new user interface has been generated so that it can help to improve the experience with this system.

To meet the above objectives, it has been seen necessary to use the Scrum framework and DevOps methodology with Microsoft Azure cloud, thus, combining both methodologies has resulted in a system with automated and centralized processes in the cloud.

Finally, it is concluded with the delivery of the system migrated to the cloud and with automated processes.

Key words: Cloud, Scrum, DevOps, Azure, Automation.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xi
INTRODUCCIÓN.....	1
CAPÍTULO I - TRAYECTORIA PROFESIONAL	2
1.1 Presentación profesional.....	2
1.2 Experiencia profesional	2
CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	6
2.1 Empresa – actividad que realiza.....	6
2.2 Visión.....	6
2.3 Misión.....	6
2.4 Organización de la empresa	6
2.5 Área, cargo y funciones desempeñadas.....	7
2.6 Experiencia profesional realizada en la organización.....	7
CAPÍTULO III – ACTIVIDADES DESARROLLADAS	8
3.1 Situación Problemática	8
3.1.1 <i>Definición del problema</i>	8
3.2 Solución	9
3.2.1 <i>Objetivos</i>	9
3.2.1.1 Objetivo General.....	9
3.2.1.2 Objetivos Específicos	9
3.2.2 <i>Alcance</i>	9
3.2.3 <i>Etapas y Metodología</i>	10
3.2.3.1 Etapas del proceso de migración del SARH.....	10
3.2.3.2 Metodología SCRUM.....	10
3.2.3.3 Metodología DevOps.....	11
3.2.4 <i>Fundamentos utilizados</i>	13

3.2.4.1 Metodología Scrum	13
3.2.4.2 Principios de la Metodología Scrum	14
3.2.4.3 Artefactos de la Metodología Scrum	14
3.2.4.3.1 Product Backlog	14
3.2.4.3.2 Sprint Backlog	15
3.2.4.3.3 Sprint Goal.....	15
3.2.4.4 Ceremonias de la Metodología Scrum.....	15
3.2.4.4.1 Organización del Backlog	15
3.2.4.4.2 Planificación del Sprint	16
3.2.4.4.3 Sprint	16
3.2.4.4.4 Daily Scrum.....	17
3.2.4.4.5 Sprint Review	17
3.2.4.4.6 Sprint Retrospective	17
3.2.4.5 Tipos de Cloud Computing.....	18
3.2.4.5.1 Cloud Pública	18
3.2.4.5.2 Cloud Privada	18
3.2.4.5.3 Cloud Híbrida	18
3.2.4.6 Servicios de Cloud Computing.....	19
3.2.4.6.1 Infraestructura como servicio (IaaS)	19
3.2.4.6.2 Plataforma como servicio (PaaS)	20
3.2.4.6.3 Software como servicio (SaaS).....	21
3.2.4.7 Flujo de trabajo con DevOps.....	22
3.2.4.8 Ciclo de vida de DevOps.....	25
3.2.4.8.1 Fase de Planificación.....	25
3.2.4.8.2 Fase de Desarrollo	26
3.2.4.8.3 Fase de Entrega.....	26
3.2.4.8.4 Fase de Operación	26
3.2.4.9 Servicios de Azure DevOps.....	27
3.2.4.9.1 Azure Boards	27
3.2.4.9.2 Azure Pipelines.....	28
3.2.4.9.3 Azure Repos	29
3.2.4.9.4 Azure Test Plans.....	30
3.2.4.9.5 Azure Artifacts	31
3.2.4.10 Practicas recomendadas para DevOps	32
3.2.4.10.1 Gestión ágil de proyectos	32
3.2.4.10.2 Enfoque Shift Left con CI/ CD.....	33
3.2.4.10.3 Implementa la automatización.....	33

3.2.4.10.4 Supervisa la canalización y las aplicaciones DevOps	33
3.2.4.10.5 Recopila feedback continuo.....	34
3.2.4.10.6 Cambia la cultura.....	34
3.2.4.11 DevOps y Metodología Ágil.....	35
3.2.5 Implementación de las áreas de proceso y sus buenas prácticas	35
3.2.5.1 Diseño de la Arquitectura AS-IS	35
3.2.5.2 Diseño de la Arquitectura TO-BE	36
3.2.5.3 Planificación de actividades del equipo DevOps.	38
3.2.5.4 Planificación de las APIs para la migración.....	40
3.2.5.5 Épicas, Historias de Usuario y Sprints del Proyecto	43
3.2.5.6 Creación de App Services para FrontEnd.....	44
3.2.5.7 Implementación de repositorios con Azure Repos	46
3.2.5.8 Pipelines y despliegues de BackEnd y FrontEnd con CI.....	48
3.2.5.9 Pruebas Manuales y Automatizadas	54
3.3 Evaluación.....	57
3.3.1 Evaluación Económica / Evaluación Costo - Beneficio	57
3.3.2 Interpretación del VAN y del TIR	58
CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	59
CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES.....	60
5.1 Conclusiones	60
5.2 Recomendaciones	61
5.3 Fuentes de Información	62
5.4 Glosario	64
ANEXOS	65
Anexo 1. Políticas de Branch.....	65
Anexo 2. Políticas de Branch de Build Validation	65
Anexo 3. Release de FrontEnd de Desarrollo.....	66
Anexo 4. Pipeline FrontEnd con Integración Continua.....	74

ÍNDICE DE FIGURAS

Figura 1	<i>Organigrama de la compañía</i>	7
Figura 2	<i>Diagrama del ciclo de vida de Scrum</i>	11
Figura 3	<i>Organizaciones que llevan el nombre DevOps en sus equipos</i>	12
Figura 4	<i>Organizaciones que están iniciando con DevOps</i>	12
Figura 5	<i>Herramientas más usadas en las organizaciones</i>	13
Figura 6	<i>Tipos de Cloud Computing</i>	19
Figura 7	<i>Componentes que se administran en un modelo de IaaS</i>	20
Figura 8	<i>Componentes que se administran en un modelo de PaaS</i>	21
Figura 9	<i>Componentes que se administran en un modelo de SaaS</i>	22
Figura 10	<i>Gráfica de WorkFlow</i>	24
Figura 11	<i>Ciclo de vida de DevOps</i>	25
Figura 12	<i>Seguimiento de casos de usuario, errores, características y epopeyas</i>	28
Figura 13	<i>Conceptos principales en un Pipeline</i>	29
Figura 14	<i>Lista de Pull Request generados desde Azure Repos</i>	30
Figura 15	<i>Lista de Planes de Prueba generados desde Azure Test Plans</i>	31
Figura 16	<i>Lista de Paquetes que ofrece Azure Artifacts en la ejecución de sus tareas</i>	32
Figura 17	<i>Diagrama de la Topología Cliente Servidor</i>	35
Figura 18	<i>Diagrama de la Topología de infraestructura de la migración</i>	37
Figura 19	<i>Listado de Épicas del Proyecto</i>	43
Figura 20	<i>Creación de Historia de Usuario con Azure DevOps Boards</i>	44
Figura 21	<i>Creación de AppService para FrontEnd</i>	45
Figura 22	<i>Creación del Repositorio de BackEnd para los Microservicios</i>	46
Figura 23	<i>Creación del Repositorio de FrontEnd para los MicroFrontEnd</i>	47
Figura 24	<i>Creación del Repositorio de QA para las pruebas con Cypress</i>	47
Figura 25	<i>Listado de tareas de pipeline de BackEnd</i>	48
Figura 26	<i>Releases generados por pipeline de BackEnd</i>	48
Figura 27	<i>Integración continua de artefacto de pipeline de BackEnd</i>	49
Figura 28	<i>Stage de Desarrollo luego de generada una release en BackEnd</i>	49
Figura 29	<i>Stage de Testing para generar release en BackEnd</i>	50
Figura 30	<i>Stage de Producción para generar release en BackEnd</i>	50
Figura 31	<i>Listado de tareas de Pipeline de FrontEnd</i>	51
Figura 32	<i>Releases generados por pipeline de FrontEnd</i>	51
Figura 33	<i>Integración continua de artefacto de pipeline de FrontEnd</i>	52
Figura 34	<i>Stage de Desarrollo luego de generada una release en FrontEnd</i>	52
Figura 35	<i>Stage de Testing para generar release en FrontEnd</i>	53
Figura 36	<i>Stage de Producción para generar release en FrontEnd</i>	53
Figura 37	<i>Prueba manual de listado de opciones de menú de FrontEnd</i>	54
Figura 38	<i>Prueba manual de listado de Tenants de MicroFrontEnd</i>	55
Figura 39	<i>Lista de tareas de pipeline de Cypress para pruebas automatizadas</i>	55
Figura 40	<i>Resultados de ejecución de pipeline de pruebas automatizadas</i>	56
Figura 41	<i>Cálculo de VAN y TIR del proyecto</i>	58

ÍNDICE DE TABLAS

Tabla 1 <i>Experiencia Laboral en VISMA LATAM</i>	2
Tabla 2 <i>Experiencia laboral en INDRA</i>	2
Tabla 3 <i>Experiencia laboral en VASS LATAM</i>	3
Tabla 4 <i>Experiencia laboral en BEYOND SAC</i>	3
Tabla 5 <i>Experiencia laboral en INETUM</i>	3
Tabla 6 <i>Experiencia laboral en INNOVARIT</i>	3
Tabla 7 <i>Experiencia laboral en GEPAE</i>	4
Tabla 8 <i>Experiencia laboral en Clínica Vesalio</i>	4
Tabla 9 <i>Experiencia laboral en GEPAE</i>	4
Tabla 10 <i>Experiencia laboral en BEYOND SAC</i>	4
Tabla 11 <i>Formación Académica</i>	4
Tabla 12 <i>Certificaciones</i>	5
Tabla 13 <i>Cursos y programas</i>	5
Tabla 14 <i>Desarrollo de actividades del equipo de DevOps</i>	38
Tabla 15 <i>Desarrollo de actividades del equipo de BackEnd</i>	38
Tabla 16 <i>Desarrollo de actividades del equipo de FrontEnd</i>	39
Tabla 17 <i>Listado de APIs que se crearon en el proyecto de migracion</i>	40
Tabla 18 <i>Costos estimados de profesionales en el proyecto de migración</i>	57

INTRODUCCIÓN

El actual trabajo de experiencia profesional describe la implementación del Sistema de Administración de Recursos Humanos (SARH) de una empresa dedicada al desarrollo de Software para su comercialización. Con ello se busca reducir el tiempo de entrega del sistema de administración hacia sus clientes para la posterior gestión de sus empleados.

El camino de la migración del producto SARH On-Premise hacia la tecnología de Azure Cloud se realizó utilizando la metodología Scrum y Kanban para ir alineados a la transformación digital que se viene llevando a cabo en la propia compañía y también para dar respuesta a los clientes que vienen trabajando con los productos de On-Premise y no sean afectados por los cambios en el software.

La implementación del SARH nace por la necesidad de crear una solución eficiente, moderna y más amigable para el usuario final. Además, el desarrollo se llevó a cabo usando la metodología DevOps para optimizar y agilizar los entregables del producto y cumplir con los objetivos propuestos.

El actual trabajo de experiencia profesional se desarrolló con los siguientes capítulos:

En el CAPITULO I se menciona todo el recorrido profesional del autor, los periodos en los que estuvo trabajando, clientes para los que trabajo, los cargos que mantuvo, los logros obtenidos, además de ver un recorrido de la formación académica y los centros de estudio por los que realizo su formación educativa.

En el CAPITULO II se narra la misión, visión y se describe la estructura de la empresa donde se implementó el proyecto en mención.

En el CAPITULO III se detallan los fundamentos bajos los que se realizó el actual trabajo, las actividades que se determinaron para el proyecto de Migración del SARH, las herramientas usadas, etapas y procesos involucrados para el desarrollo de este software.

En el CAPITULO IV se describe la experiencia lograda por la participación del autor para este proyecto.

En el CAPITULO V se mencionan todas las conclusiones y las recomendaciones realizadas por el autor para el proyecto realizado.

CAPÍTULO I - TRAYECTORIA PROFESIONAL

1.1 Presentación profesional

El autor del presente trabajo de suficiencia profesional es un profesional egresado de la Facultad de Ingeniería de Sistemas e Informática en la Universidad Nacional Mayor de San Marcos con experiencia en el diseño, construcción e implementación de Sistemas de Información, además de contar con solidos valores personales y habilidades blandas que le han permitido llevar un fuerte trabajo en equipo.

Actualmente se encuentra desempeñándose como Desarrollador FrontEnd en la empresa de VISMA LATAM participando activamente en las soluciones de sistemas y realizando buenas prácticas del Desarrollo de Software.

1.2 Experiencia profesional

Tabla 1

Experiencia Laboral en VISMA LATAM

VISMA LATAM	
Desde octubre 2021 hasta la actualidad	
Cargo	Desarrollador FrontEnd
Funciones	Desarrollo MicroFrontEnd con Angular 13 y la librería SingleSPA. Seguimiento automatización con Azure DevOps. Participación en ceremonias de SCRUM.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 2

Experiencia laboral en INDRA

INDRA	
Desde abril 2021 hasta octubre 2021	
Cliente	Telefónica del Perú
Cargo	Desarrollador FrontEnd
Funciones	Desarrollo y Mantenimiento de Sistema BackOffice de Telefónica del Perú sobre el estado de Averías de Usuario con Framework Angular. Participación en ceremonias de SCRUM.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 3*Experiencia laboral en VASS LATAM*

VASS LATAM	
Desde abril 2020 hasta marzo 2021	
Cliente	Telefónica del Perú
Cargo	Desarrollador FrontEnd
Funciones	Desarrollo y Mantenimiento de aplicativo Mi Movistar utilizando Framework Móvil IONIC. Participación de ceremonias de SCRUM.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 4*Experiencia laboral en BEYOND SAC*

BEYOND SAC	
Desde enero 2019 hasta marzo 2020	
Cliente	JP Planning y Lima Golf Club
Cargo	Desarrollador Full Stack
Funciones	Desarrollo de Pagina Web de JP Planning. Desarrollo Móvil de Aplicativo de Encuestas de satisfacción para los asociados de Lima Golf Club.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 5*Experiencia laboral en INETUM*

INETUM	
Desde junio 2018 hasta diciembre 2018	
Cliente	Compartamos Financiera
Cargo	Desarrollador Full Stack
Funciones	Análisis, Desarrollo y Mantenimiento de Sistema Core de Prestamos Financieros Grupales. Participación en ceremonias de SCRUM.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 6*Experiencia laboral en INNOVARIT*

INNOVARIT	
Desde agosto 2017 hasta febrero 2018	
Cliente	INTERSEGURO
Cargo	Analista de Sistemas
Funciones	Desarrollo .NET de Migración Pólizas de SURA hacia Interseguro. Implementación y mejoras de cálculo de reservas a la SBS. Participación de ceremonias de SCRUM.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 7*Experiencia laboral en GEPAE*

GEPAE	
Desde octubre 2016 hasta julio 2017	
Cliente	Instituto de Ingenieros de Minas del Perú
Cargo	Analista Programador de Sistemas
Funciones	Desarrollo PHP de Sistema Multi Eventos de E-Commerce de Stands para Convención Minera PERUMIN.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 8*Experiencia laboral en Clínica Vesalio*

Clínica Vesalio	
Desde agosto 2015 hasta julio 2016	
Cargo	Analista y Programador de Sistemas
Funciones	Análisis, Desarrollo .NET e Implementación de Sistema de Gestión documental para Historias Clínicas. Mantenimiento de Sistema de generación de Tramas hacia SUSALUD.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 9*Experiencia laboral en GEPAE*

GEPAE	
Desde mayo 2014 hasta abril 2015	
Cliente	Instituto de Ingenieros de Minas del Perú
Cargo	Analista Programador de Sistemas
Funciones	Desarrollo PHP de Sistema de Reserva de Stands para Convención Minera PERUMIN.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 10*Experiencia laboral en BEYOND SAC*

BEYOND SAC	
Desde febrero 2011 hasta abril 2014	
Clientes	GS1 Perú y Universidad San Juan Bautista
Cargo	Programador, Analista Programador de Sistemas
Funciones	Desarrollo .NET de Sistema Contable para GS1 Perú. Desarrollo .NET de Sistema Educativo para la Universidad San Juan Bautista.

Nota. Fuente: Elaborado por el autor (2022).

Tabla 11*Formación Académica*

Formación	Institución	Grado
Ingeniería de Sistemas e Informática	Universidad Nacional Mayor de San Marcos	Bachiller en Ingeniería de Sistemas

Nota. Fuente: Elaborado por el autor (2022).

Tabla 12*Certificaciones*

Certificado	Institución	Fecha
Fundamentos de SCRUM	SCRUM Study	Mayo 2021
Cloud Computing	Google Activate	Mayo 2021
Oratoria, Comunicación Efectiva y Exposición	Pontificie Universidad Católica del Perú	Noviembre 2017

Nota. Fuente: Elaborado por el autor (2022).

Tabla 13*Cursos y programas*

Centro de Estudios	Formación
Diplomado en Herramientas de Innovación	ISIL
Redux en Angular con NGRX	UDEMY
React de cero a experto	UDEMY
Angular Avanzado: MEAN	UDEMY
Angular de cero a experto	UDEMY
Diplomado de Diseño Grafico	CIBERTEC
Business Intelligence	CIBERTEC
Android Básico	CIBERTEC
Android Avanzado	Sistemas UNI
SQL Server Implementación	Sistemas UNI
Android Básico	Sistemas UNI
Java Aplicaciones Web	Sistemas UNI
PHP Nivel Avanzado	Sistemas UNI
Java Cliente Servidor	Sistemas UNI
Android Avanzado	Sistemas UNI
SQL Server Implementación	Sistemas UNI
Android Básico	Sistemas UNI

Nota. Fuente: Elaborado por el autor (2022).

CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1 Empresa – actividad que realiza

Empresa privada del rubro tecnológico ubicada en Latinoamérica que se encarga de desarrollar software y está especializada en software para Recursos Humanos, además se encarga de maximizar el potencial humano a través de la tecnología.

Actualmente las competencias de esta compañía es mejorar la comunicación con todos los colaboradores y generar la empatía con ellos mismos, esto también significa aceptar la nueva realidad tras la coyuntura por la pandemia de COVID 19 para fomentar el trabajo remoto y la transformación digital a la que deben de apuntar las empresas hoy en día.

2.2 Visión

Crear una cultura digital con el uso de herramientas y plataformas de colaboración además de crear valor con un bajo costo y que genere un gran impacto, aumentar la confianza en los trabajadores para que puedan comprender y autorregularse con las normas de cada país.

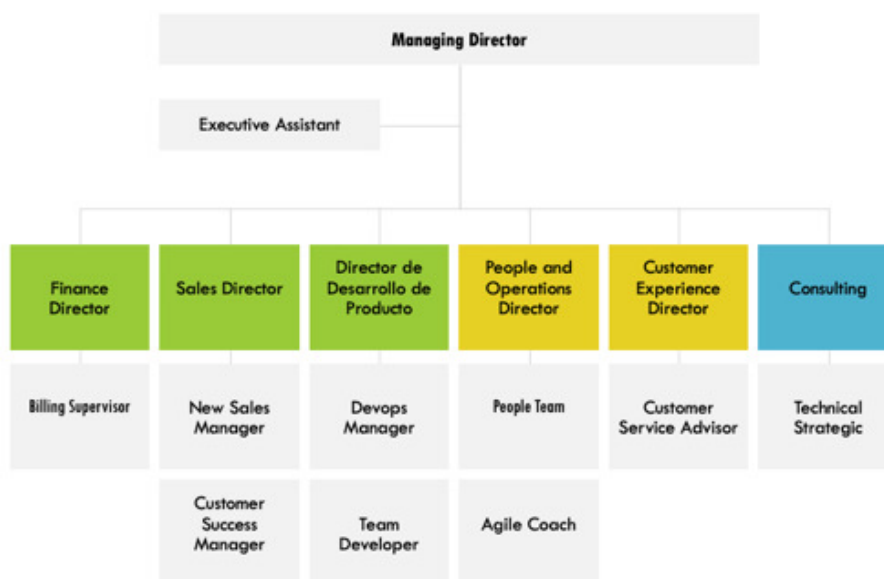
2.3 Misión

Automatizar procesos complejos y repetitivos de las personas empezando por el cambio cultural en los trabajadores y líderes de esta compañía.

2.4 Organización de la empresa

En la Figura 1 se presentan las áreas que componen el organigrama de la compañía para el año 2022.

Figura 1
Organigrama de la compañía.



Nota. Fuente: Elaborado por el autor (2022).

2.5 Área, cargo y funciones desempeñadas

El autor del actual trabajo de suficiencia profesional ejerció el rol de Desarrollador de FrontEnd, cuyas funciones desempeñadas fueron las siguientes:

- Análisis técnico y estimación de las Historias de Usuario que se desarrollaron.
- Participación en todas las ceremonias de Scrum.
- Desarrollo e Implementación de mejoras prácticas para MicroFrontEnd usando la librería SingleSPA.
- Desarrollo de otras propuestas tecnológicas de innovación como la migración de Angular a su versión más reciente.
- Participación de las reuniones con las áreas de UX/ UI para definir el alcance y los mejores insumos para el FrontEnd del diseño final.

2.6 Experiencia profesional realizada en la organización

El autor de este trabajo puso en ejecución toda la experiencia ganada a lo largo de más de 10 años en la industria de Software de Sistemas Informáticos, esto contribuyó a realizar las mejores prácticas del Desarrollo de Software y dar iniciativa con aportes para la mejora de la experiencia del usuario y así cumplir los objetivos trazados en sus proyectos y poder entregar un producto con un gran valor para los clientes.

CAPÍTULO III – ACTIVIDADES DESARROLLADAS

3.1 Situación Problemática

El SARH en On-Premise es un sistema para usuarios internos de operaciones y tiene toda su infraestructura realizada con tecnologías de Microsoft, el desarrollo de su sistema y lenguaje de programación es ASP .NET en donde los cambios se realizaban de forma local y la publicación de las modificaciones se realizaban a través de FILEZILLA u otra herramienta para la conexión de Cliente a Servidor y utilizando una conexión con IPLAN las nuevas modificaciones reemplazaban a las fuentes de producción en el servidor, siendo así un proceso 100% manual.

En una posterior actualización a esta versión de On-Premise se utilizó la misma infraestructura en Microsoft, pero ahora utilizando la nube de Azure. Con este cambio se generaron los pipelines que creaban a nuevos artefactos, pero estos tenían como objetivo la creación de una fuente comprimida en formato ZIP y las subidas de estos cambios a producción seguirían realizándose de forma manual.

Para cumplir con el objetivo de tener un proceso totalmente automatizado es que se realiza la migración de todos los sistemas hacia la nube de Azure, utilizando todas las herramientas que Azure nos brinda y migrando los códigos fuentes hacia nuevas tecnologías especializadas. Aprovechando así de crear una nueva interfaz para los usuarios y optimizando procesos que se volvían insostenibles y costosos.

3.1.1 Definición del problema

De lo expuesto en la situación problemática anterior, se define el problema como el extenso trabajo manual y repetitivo, que está sujeto a un alto nivel de error humano además de volver deficiente el control y mantenimiento de todas las versiones que se van generando de un sistema.

3.2 Solución

3.2.1 Objetivos

3.2.1.1 Objetivo General.

El objetivo principal es cumplir con la planificación de la migración del SARH haciendo uso de la Metodología DevOps tomando las herramientas de Azure DevOps y apoyándose del Framework Ágil de SCRUM para la gestión de este proyecto.

3.2.1.2 Objetivos Específicos

Se presentan los objetivos específicos:

- Realizar la planificación de la migración del SARH hacia la nube de Azure.
- Aplicar el marco de trabajo Scrum para la gestión del presente proyecto llevando a cabo las ceremonias y lineamientos de las metodologías ágiles.
- Aplicar las mejoras prácticas de DevOps con el servicio de Microsoft Azure.
- Desarrollar las pruebas automatizadas del SARH y verificar el correcto funcionamiento y desarrollo operativo de los clientes y usuarios finales.

3.2.2 Alcance

Se presentan los alcances del actual trabajo de suficiencia:

- Llevar a cabo los lineamientos de la Metodología de Scrum realizando las ceremonias de Sprint Planning, Retrospectiva y Daily Meeting utilizando las herramientas que nos brinda Azure DevOps.
- La definición de los equipos de trabajo especializados en BackEnd, FrontEnd, DevOps y Testing. El equipo de BackEnd debe de seguir los lineamientos de desarrollo utilizando MicroServicios, el equipo de FrontEnd debe usar la tecnología de MicroFrontEnd, el equipo de Testing debe de apoyar con las pruebas automatizadas de End2End y el equipo DevOps debe de apoyar con la generación de los ambientes y herramientas para los desarrolladores.
- Desarrollo de un nuevo Framework FrontEnd con Angular ya que debido al Sistema On-Premise con lenguaje ASP NET se tuvo una arquitectura centralizada y no existía un proyecto de FrontEnd ni BackEnd independiente que pueda brindar una mejor experiencia para el usuario.
- Desarrollo de los Pipeline utilizando a la Integración Continua (CI).

3.2.3 Etapas y Metodología

3.2.3.1 Etapas del proceso de migración del SARH

Se presentan las etapas del proyecto de migración:

- Planificación de las actividades de Migración del SARH.
- Desarrollo FrontEnd y BackEnd.
- Implementación de Pipelines para la automatización de software.
- Desarrollo de Pruebas Automatizadas E2E.

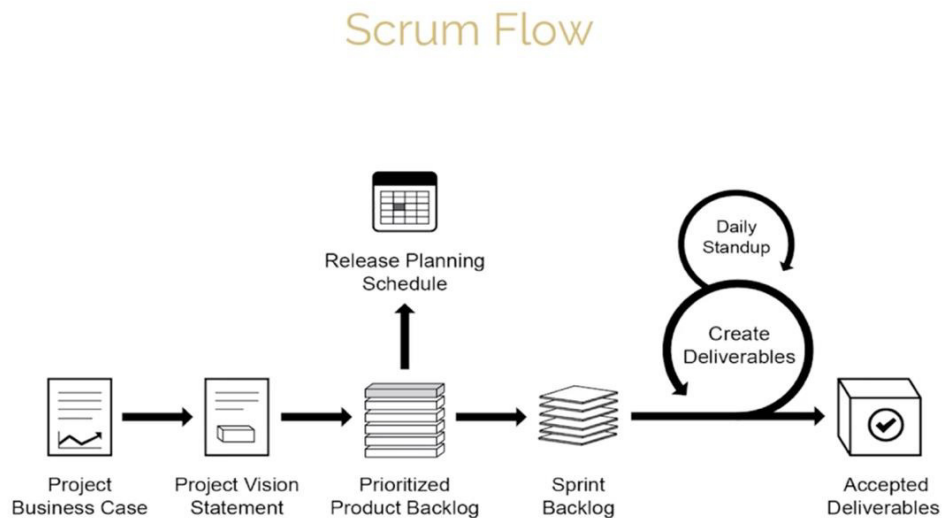
3.2.3.2 Metodología SCRUM

En el presente trabajo se llevó a cabo la metodología ágil para la gestión de proyectos de Scrum, esto implicó que se realicen las actividades de elaboración del Product Backlog y los Sprint Backlog correspondientes al proyecto, también se llevaron a cabo las ceremonias relacionadas con el trabajo del equipo de desarrolladores como son la Sprint Planning, las reuniones diarias o también llamadas Daily Scrum, además de la Sprint Review y la Sprint Retrospective, todo esto con el objetivo de poder cumplir los lineamientos de este marco de trabajo que significa la metodología Scrum y conllevar a aportar agilidad al equipo.

Es importante señalar que en el desarrollo del proyecto de migración se logró implementar y adaptar a este marco de trabajo con todos los equipos, lo que implicó que se volvieran más auto gestionables y flexibles.

A continuación, se presenta en la Figura 2, el flujo y ciclo de vida de Scrum presentado por la institución Scrum Study (SCRUM Study, 2022) que sirvió de guía para la implementación de Scrum en el actual trabajo.

Figura 2
Diagrama del ciclo de vida de Scrum



Nota. Fuente: Tomada de Scrum Study, 2022. (SCRUM Study, 2022)

3.2.3.3 Metodología DevOps

DevOps refiere a un término que implica una metodología de pasos que se deben de cumplir para entregar valor a los usuarios finales pero que lleva como mayor valor la integración de equipos.

Según lo definido por la compañía Amazon (AMAZON, 2022) se considera lo DevOps como lo siguiente:

“DevOps es una metodología que combina herramientas, mejores prácticas y una filosofía cultural que crea automatización e integra los procesos entre los equipos de desarrollo y operaciones. Se resalta con mayor importancia de la comunicación, el empoderamiento y colaboración entre los equipos y la automatización de la tecnología. El término DevOps, es una combinación de las palabras desarrollo y operaciones y refleja el proceso de integración de estas 2 áreas en un proceso común y continuo.”

En la Figura 3 que se muestra a continuación, se indica una estadística realizada por la compañía Atlassian (ATLASSIAN, 2020) donde muestra el comportamiento de los equipos que llevan como nombre de DevOps en sus títulos.

Figura 3

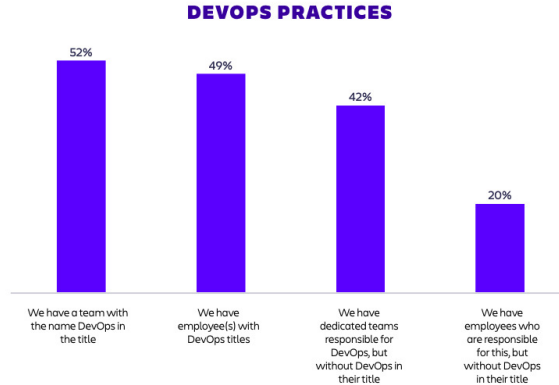
Organizaciones que llevan el nombre DevOps en sus equipos

DevOps titles are the norm...

Organizations use DevOps in titles – **69% have a team with DevOps in the name and/or employees with DevOps in their titles.**

- ▶ Larger organizations with 500+ employees are especially likely to have a team with DevOps in the title (57%).
- ▶ Organizations that have been using DevOps for 3+ years are also more likely to have a team with DevOps in the name (57%).

? Which of the following best describes your organization's management of DevOps? Please select all that apply.



Nota. Fuente: Según lo indicado por Atlassian (ATLASSIAN, 2020).

En la Figura 4, se indica una estadística realizada por la compañía Atlassian (ATLASSIAN, 2020) donde se muestra el tiempo que las empresas llevan trabajando con la metodología DevOps.

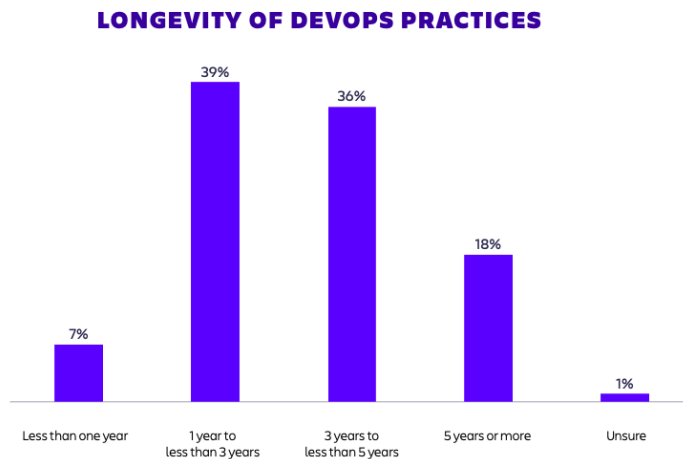
Figura 4

Organizaciones que están iniciando con DevOps

...but almost half of organizations are just starting with DevOps, practicing it for less than 3 years.

- ▶ 46% of organizations are relatively new to DevOps with under 3 years of doing it.
- ▶ 54% of organizations have been practicing DevOps for 3 years or more.

? Approximately how long has your organization used DevOps practices?



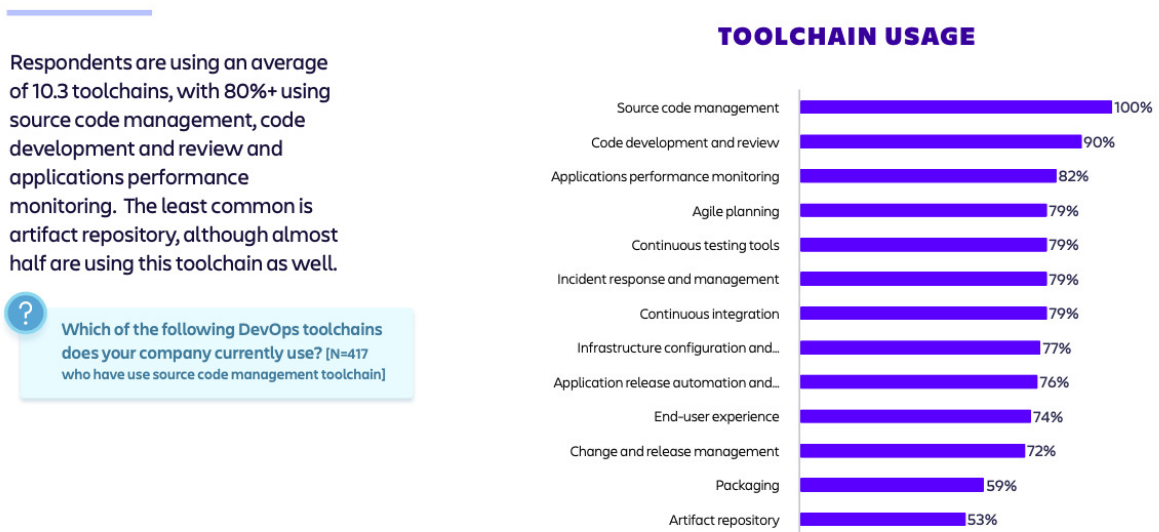
Nota. Fuente: Tomado por la compañía Atlassian (ATLASSIAN, 2020).

En la Figura 5, se indica una estadística realizada por la compañía Atlassian (ATLASSIAN, 2020) donde se muestra las herramientas más usadas por las empresas que implementan DevOps.

Figura 5

Herramientas más usadas en las organizaciones

Source code management and code development are the most widespread toolchains used



Nota. Fuente: Según lo indicado por Atlassian (ATLASSIAN, 2020).

3.2.4 Fundamentos utilizados

3.2.4.1 Metodología Scrum

Según lo definido por Scrum Study (SCRUM Study, 2022) se considera lo siguiente:

“Scrum es una metodología ágil muy conocida y es mayormente utilizada por las compañías de tecnología, pero también es de uso para compañías que no son de tecnología. Sus principales características son ser: adaptativa, iterativa, rápida, flexible y eficaz, y tiene como objetivo principal para cualquier proyecto ofrecer una entrega de gran valor en un corto tiempo. Esta metodología también garantiza comunicación inmediata y crea un entorno de colectivo y progreso continuo. Esta metodología ha sido está creada para que se puede desarrollar en equipos con cualquier producto o servicio y sin importar en tamaño y la complejidad del equipo. Uno de los puntos de mayor relevancia de Scrum es que se elabora con diversos equipos de trabajo en donde las

actividades se elaboran de forma más corta con objetivos en común para entregar valor y esto lo define un Sprint.”

3.2.4.2 Principios de la Metodología Scrum

A continuación, se presentan los 6 principios de esta metodología:

- Control de proceso empírico: Basado en los conceptos de Transparencia, Inspección y Adaptación.
- Autoorganización: Diseñado para los trabajadores de hoy, ya que al autoorganizarse aportan más que solo su experiencia técnica y esto genera mayor valor.
- Colaboración: Se debe incluir el trabajo de los Stakeholders con los miembros del equipo para generar grandes entregas de valor.
- Priorización basada en valor: Los requerimientos y las tareas respectivas son priorizadas por el mayor valor para el cliente.
- Time Boxing: El tiempo es el factor crucial en los Proyectos de Scrum, en las reuniones y los periodos de trabajo.
- Desarrollo iterativo: Permite que creen etapas en el proyecto para que todas las personas involucradas obtengan una mejor comprensión de lo que se debe entregar como parte del proyecto.

3.2.4.3 Artefactos de la Metodología Scrum

3.2.4.3.1 Product Backlog

Según lo señalado por Drumond Claire en un artículo para la compañía Atlassian (Drumond, 2022), se define el Product Backlog como lo siguiente:

“Es la principal lista de trabajo que debe desarrollarse y que es responsable el dueño del producto. Esta lista incluye a nuevas características, requisitos de los usuarios, mejoras de los sistemas o procesos y correcciones de errores y esto sirve como entrada para el backlog del Sprint. Es, esencialmente, la lista de tareas del equipo. El dueño del producto revisa, prioriza y mantiene constantemente el backlog del producto, los elementos que componen el backlog pueden dejar de ser relevantes o los inconvenientes pueden resolverse por otros caminos.”

3.2.4.3.2 *Sprint Backlog*

Según lo señalado por Drumond Claire en un artículo para la compañía Atlassian (Drumond, 2022), se define al Sprint Backlog como lo siguiente:

“Es la lista de elementos o historias de usuario, seleccionados por el propio equipo de desarrollo para su desarrollo en un ciclo de Sprint. En la reunión de planificación del Sprint, el equipo elige los elementos en los que trabajará durante el Sprint a partir del backlog del producto. Un sprint backlog debe ser flexible y puede cambiar durante un Sprint. Sin embargo, el objetivo principal del Sprint no puede asumir riesgos.”

3.2.4.3.3 *Sprint Goal*

La estrategia de Marketing Drumond Claire en un artículo para la compañía Atlassian (Drumond, 2022), define al Sprint Goal como lo siguiente:

“Es el producto final que se obtiene de un Sprint. A la definición de Hecho también se le conoce como Incremento. Sólo depende de cómo sus equipos definen Hecho y cómo definen sus objetivos de Sprint. Tomemos como ejemplo un proyecto que envía información de suscripciones a una página web y que sólo puede ser enviado a los usuarios finales después de tres meses. Es posible elegir dos semanas por cada Sprint, y tu definición de Hecho puede ser culminar una parte de una versión más grande que planificas enviar todos juntos. Sin embargo, se debe de tomar en cuenta que cuanto más tiempo se tarda en lanzar el producto final, mayor es el riesgo de que el software no llegue a integrarse correctamente.”

3.2.4.4 *Ceremonias de la Metodología Scrum*

3.2.4.4.1 *Organización del Backlog*

La estrategia de Marketing Drumond Claire en un artículo para la compañía Atlassian (Drumond, 2022), define a la Organización del Backlog como lo siguiente:

“Este evento es llevado a cabo por el dueño del producto y es conocido también como backlog grooming. Las principales actividades que se realizan en este evento son encaminar correctamente el producto y tener una dirección constante con el negocio y el usuario final. Entonces, el dueño del producto mantiene actualizada esta lista utilizando los comentarios de los usuarios y también toma en cuenta los comentarios del equipo de desarrollo para apoyar a priorizar la lista y trabajar en cualquier momento.”

3.2.4.4.2 Planificación del Sprint

La autora Drumond Claire en un artículo para la compañía Atlassian (Drumond, 2022), define a la Planificación del Sprint como lo siguiente:

“Durante este evento, todo el equipo de desarrollo organiza el trabajo a realizar durante el Sprint que va a iniciar. Esta sesión la lleva a cabo el Scrum Master y en esta se trata como punto principal poder determinar el objetivo del Sprint. En seguida, se seleccionan historias de usuario o requerimientos para el Sprint a partir del backlog del producto. Estas historias siempre se alinean con el objetivo y también son acordadas por el equipo de Scrum para que sean factibles de implementar durante el Sprint. Cuando se finaliza la reunión de planificación, todos los miembros del Scrum deben tener claro que se va a entregar en el Sprint y cómo se puede entregar el Incremento.”

3.2.4.4.3 Sprint

Según la autora Drumond Claire en un artículo para Atlassian (Drumond, 2022), define el Sprint como lo siguiente:

“Es el período de tiempo determinado por el equipo Scrum en donde se trabaja conjuntamente para cumplir con el objetivo de Sprint o cumplir con el Incremento. Dos semanas es un tamaño muy común para un Sprint, aunque en algunas compañías encuentran una semana para ser más fácil de alcance y entregar más rápido los resultados y en otros lugares eligen un mes para tener más tiempo de entregar un incremento valioso. Durante la duración del Sprint, el alcance puede renegociarse entre el dueño del producto y el equipo de desarrollo de ser necesario. Esto constituye la naturaleza empírica de Scrum. Todas las ceremonias mencionadas para Scrum se realizan durante un Sprint, el tiempo del Sprint lo debe de determinar el equipo y la sugerencia de esta Metodología es llevarlo por 2 semanas. Mantener el mismo tiempo de duración del Sprint ayuda a todo el equipo a poder comparar las experiencias de cada Sprint y mejorar para los siguientes ciclos de Sprint.”

3.2.4.4.4 Daily Scrum

Según la autora Drumond Claire en un artículo para Atlassian (Drumond, 2022), define a la Daily Scrum como lo siguiente:

“La Daily Scrum es una reunión muy breve que debe darse en un tiempo sugerido de 15 minutos para poder alinear a todo el equipo con el trabajo que va realizando cada integrante. La idea de llevar esta reunión también es para programar el trabajo durante las siguientes 24 horas. La dinámica de esta reunión tiene como objetivo que cada integrante responda a 3 preguntas: ¿Que hice ayer? ¿Qué tengo planificado para hoy? y existe algún obstáculo que dificulte mi trabajo?”

3.2.4.4.5 Sprint Review

Según la autora Drumond Claire en un artículo para Atlassian (Drumond, 2022), define a la Sprint Review como lo siguiente:

“Esta reunión de revisión de Sprint se lleva a cabo usualmente al finalizar o el último día del Sprint y sirve para poder demostrar brevemente las historias que se desarrollaron durante el Sprint, en este momento de la revisión de Sprint es el propietario del producto quien puede decidir si lanzar o no lo demostrado porque el equipo (normalmente si se decide lanzar).”

3.2.4.4.6 Sprint Retrospective

Según la autora Drumond Claire en un artículo para Atlassian (Drumond, 2022), define a la Sprint Review como lo siguiente:

“La reunión de retrospectiva es una reunión donde se enfoca en evaluar los resultados de lo que fue bien y mal, y sobre los planes de acción que se pueden tomar para mejorar aquello que no fue bien y mantener las prácticas de lo que fue bien. La idea de esta reunión es que pueda quedar documentado todas las ideas positivas, mejoras y cosas negativas que requiera el equipo. Es posible que se converse acerca de otros integrantes del equipo, de aspectos externos, herramientas y prácticas que se vienen llevando a cabo.

3.2.4.5 Tipos de Cloud Computing

3.2.4.5.1 Cloud Pública

Según el autor Zettler Kev en un artículo para la compañía Atlassian (Zettler, 2022) considera a una Cloud Pública como lo siguiente:

“Una nube pública es un modelo de Tecnología de la Información (TI) en donde se ofrecen sus recursos informáticos (como servidores y almacenamiento), a través de Internet y donde la administración la realiza el propio proveedor. Las organizaciones pagan los modelos de nube pública en base a su consumo, lo que convierte a este modelo en una solución ideal para empresas más pequeñas que buscan un ahorro económico. Microsoft Azure es un ejemplo de nube pública.”

3.2.4.5.2 Cloud Privada

Según el autor Zettler Kev en un artículo para la compañía Atlassian (Zettler, 2022) considera a una Cloud Privada como lo siguiente:

“Una nube privada es un modelo de TI que hace referencia a una infraestructura en la nube que lo utiliza exclusivamente una empresa. Algunas empresas pagan a proveedores externos para que hospeden su nube privada. Aunque este modelo es costoso, también es muy adecuado para grandes organizaciones que se centran en la seguridad y la personalización de sus sistemas.”

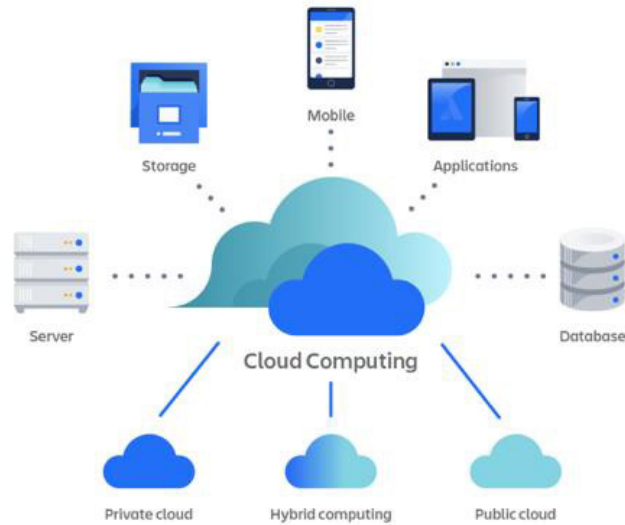
3.2.4.5.3 Cloud Híbrida

Según el autor Zettler Kev en un artículo para la compañía Atlassian (Zettler, 2022) considera a una Cloud Híbrida como lo siguiente:

“Las nubes híbridas son un modelo de TI en donde se combinan al menos una nube pública y una nube privada. Este enfoque de computación es muy extendido ya que la mayoría de las empresas no depende por completo de la nube pública.”

A continuación, en la Figura 6 se presentan los 3 tipos de Cloud Computing según Zettler Kev en un artículo para la compañía Atlassian (Zettler, 2022):

Figura 6
Tipos de Cloud Computing



Nota. Fuente: Atlassian (2022)

3.2.4.6 Servicios de Cloud Computing

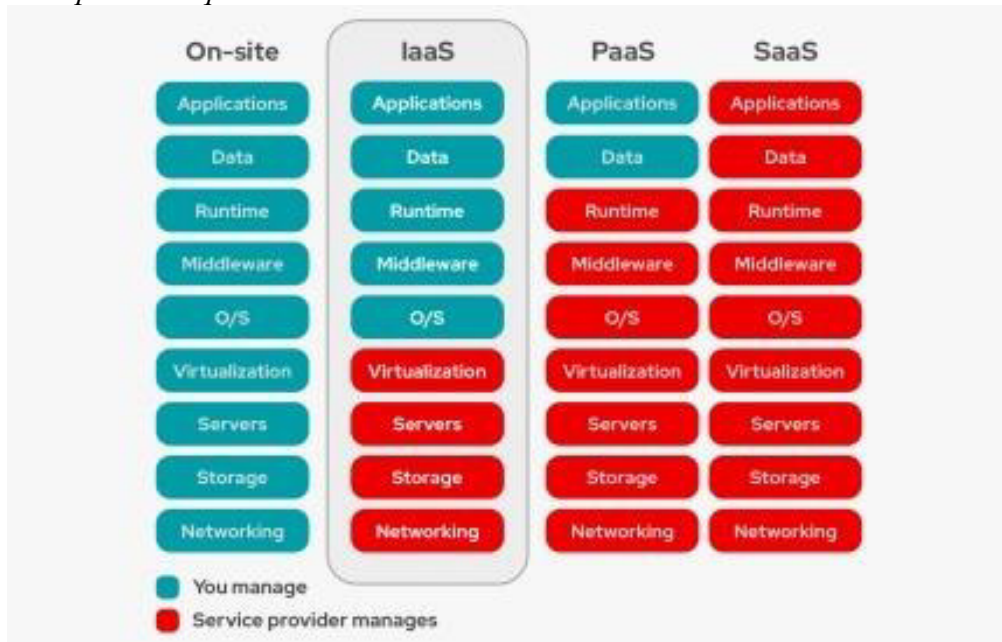
3.2.4.6.1 Infraestructura como servicio (IaaS)

Según lo definido por la compañía Red Hat (Red Hat, 2022) se considera lo siguiente:

“En este tipo de servicio en la nube, es el proveedor de servicios quien gestiona la infraestructura para sus clientes y es usualmente usado a través de Internet. Se paga por la infraestructura y el usuario accede a ella a través de APIs o un panel de administración. El usuario es encargado de administrar el sistema operativo, aplicaciones y el middleware, mientras que los proveedores son responsables de la parte las redes, el hardware (los discos duros y lugar donde se almacenen los datos) y los servidores. Además, el proveedor es el que se encarga de solucionar y mitigar las caídas, hacer las reparaciones y solucionar los problemas en el hardware.”

En la Figura 7, se presentan los componentes que administra un cliente y su proveedor en un modelo de IaaS, según la compañía Red Hat (Red Hat, 2022).

Figura 7
Componentes que se administran en un modelo de IaaS



Nota. Fuente: Red Hat (2022)

3.2.4.6.2 Plataforma como servicio (PaaS)

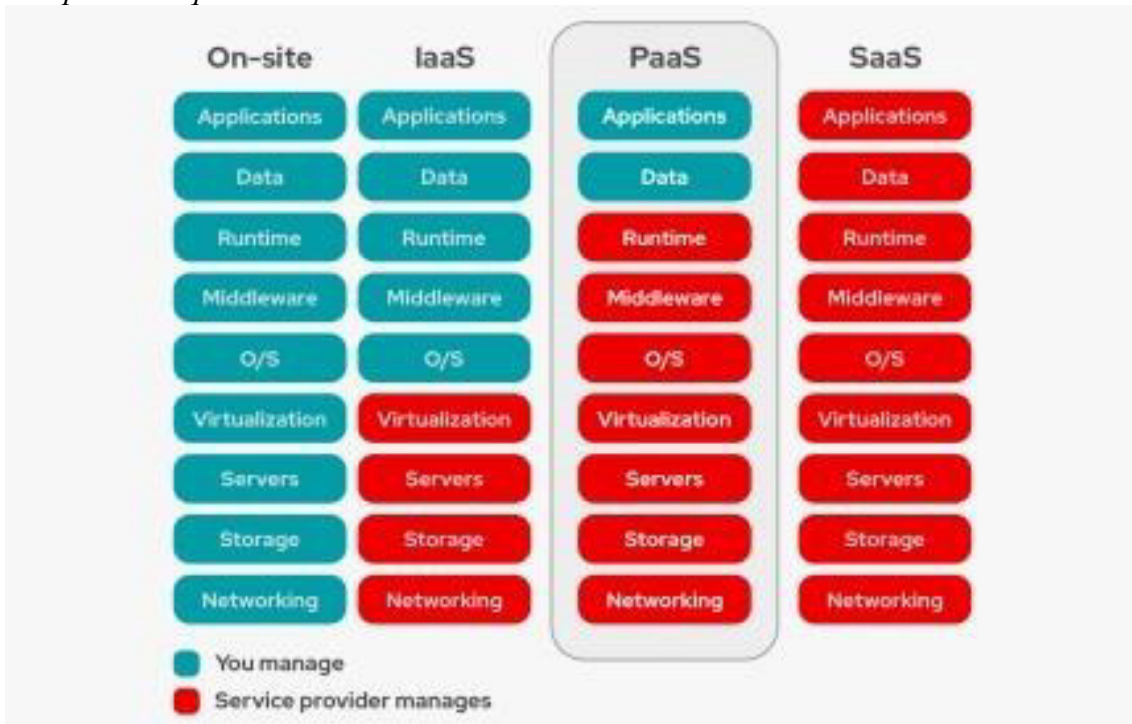
Según lo definido por la compañía Red Hat (Red Hat, 2022) se considera lo siguiente:

“En este tipo de servicio en la nube, es el proveedor de servicios quienes proporcionan y gestionan el hardware, la plataforma de software de aplicaciones, pero el usuario es quien tiene el control de las aplicaciones que se ejecutan en ella y los datos en los que se basan. Normalmente los clientes pagan una tarifa mensual para adquirir estos servicios, pero también pueden elegir lo que desean consumir.”

En la Figura 8, se presentan los componentes que administra un cliente y su proveedor en un modelo de PaaS, según la compañía Red Hat (Red Hat, 2022).

Figura 8

Componentes que se administran en un modelo de PaaS



Nota. Fuente: Red Hat (2022)

Estas son algunas de las características del modelo de PaaS:

- Permite que los clientes que adquieren este modelo se dediquen al desarrollo sin preocuparse de la infraestructura.
- Puedes sacar al mercado rápidamente tus productos o servicios.
- Los proveedores se encargan de realizar copias de seguridad, software de servidor y de los sistemas operativos.
- La escalabilidad se puede realizar de manera muy sencilla y es rentable.
- Tiene un gran acceso económico en general.

3.2.4.6.3 Software como servicio (SaaS)

Según lo definido por la compañía Red Hat (Red Hat, 2022) se considera lo siguiente:

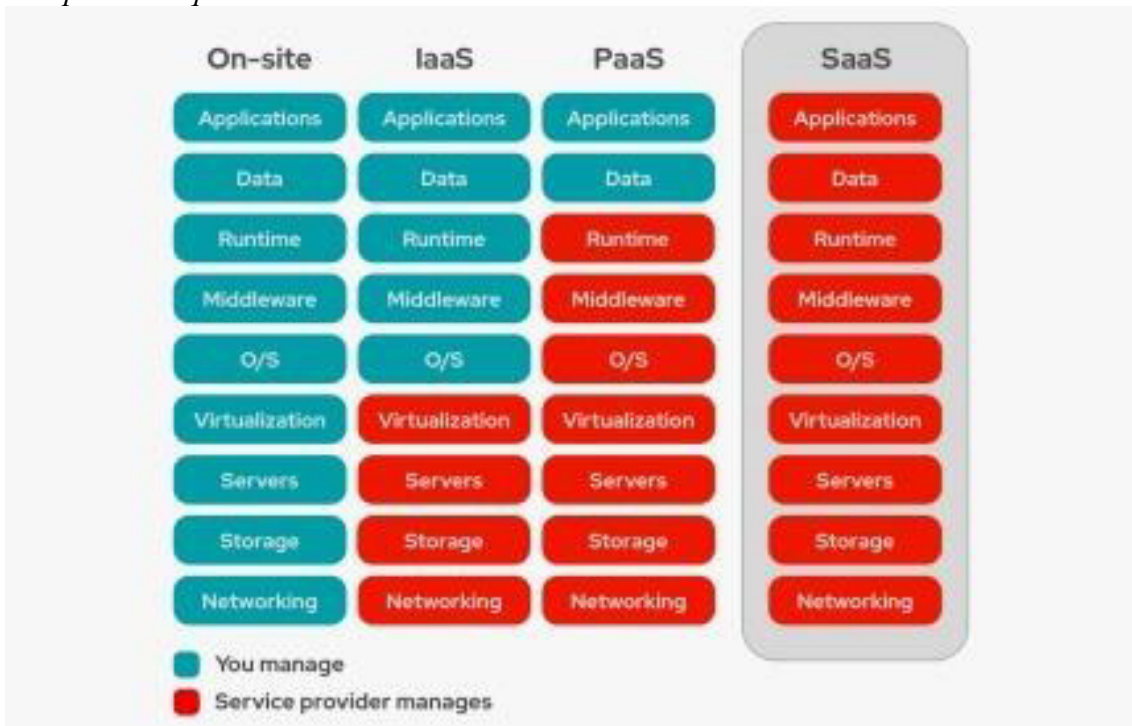
“Este tipo de servicio en la nube provee a sus usuarios una aplicación de software del cual es el proveedor el que administra a este servicio que usualmente se coloca en la nube. Por lo general, las aplicaciones SaaS son aplicaciones de tipo web o aplicaciones móviles a las que los usuarios acceden a través de un portal de administración o una API. Las actualizaciones de software, las correcciones de fallos y otros mantenimientos

generales del software están a cargo del usuario, y se conectan a las aplicaciones de la nube a través de un panel o una API.”

En la Figura 9, se presentan los componentes que administra un cliente y su proveedor en un modelo de SaaS, según la compañía Red Hat (Red Hat, 2022).

Figura 9

Componentes que se administran en un modelo de SaaS



Nota. Fuente: RedHat (2022)

Estas son algunas de las características del modelo de SaaS:

- Se establece un modelo de suscripción para el acceso de los clientes.
- Son los proveedores los responsables de gestionar y mantener actualizado el software.
- Los datos se encuentran seguros en la nube.
- El escalamiento de los recursos se puede realizar según la necesidad del cliente.

3.2.4.7 Flujo de trabajo con DevOps

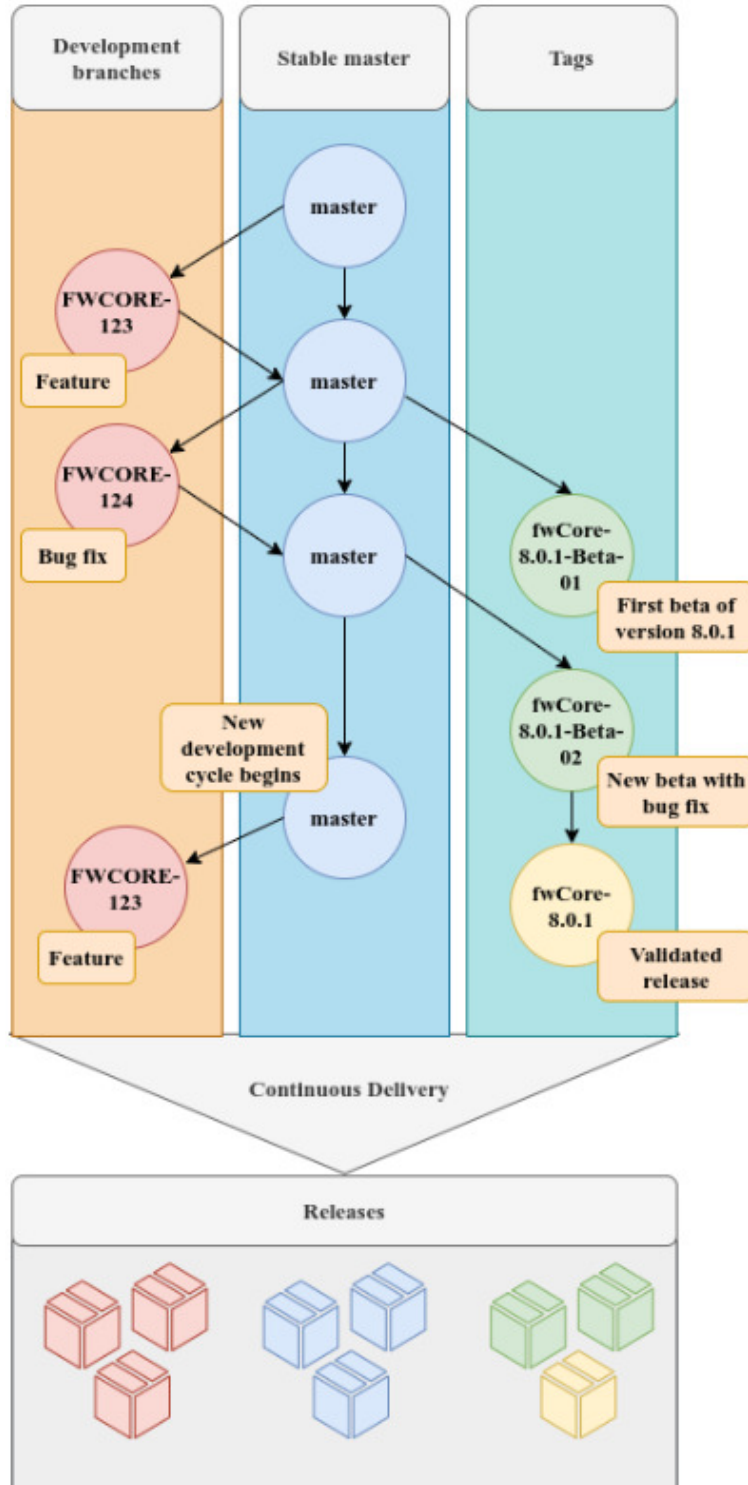
Según lo definido por los autores Silvola y Sargsyan (Silvola & Sargsyan, 2021) se considera lo siguiente:

“Al decidir un flujo de trabajo para el desarrollo con Git, se adoptó el flujo Git estándar de la industria, que posteriormente se acercó más al flujo de GitHub. Para

cualquier proyecto, la rama maestra se considera estable, aunque no necesariamente de calidad de producción. No hay una rama de desarrollo, sólo ramas de corrección de errores y de características, todas las cuales tienden a ser de corta duración. Las fusiones frecuentes promueven un desarrollo dirigido y granular, reduciendo la ocurrencia y la probabilidad de conflictos de fusión complicados y que consumen mucho tiempo. Para iniciar el desarrollo de una corrección de errores o una característica, el desarrollador crea manualmente un ticket en el sistema de tickets. El nombre de ese ticket se utiliza como nombre de la rama. En esa rama, cada confirmación produce una versión que puede pasar a ser validada por los usuarios. Una vez que se crea una solicitud de fusión, las pruebas unitarias se ejecutan automáticamente, y cada empuje posterior desencadenará un nuevo conjunto de pruebas que se ejecutarán. Después de una revisión y con las pruebas realizadas con éxito, la rama se fusiona con master, dando lugar a una construcción instantánea. Una etiqueta produce una versión que se despliega automáticamente en un conjunto de repositorios. Todos los pasos del proceso pueden realizarse desde la interfaz de línea de comandos o desde una única pestaña del navegador.”

A continuación, en la Figura 10, se presenta el flujo de trabajo de desarrollo indicado por los autores Silvola y Sargsyan (Silvola & Sargsyan, 2021).

Figura 10
Gráfica de WorkFlow



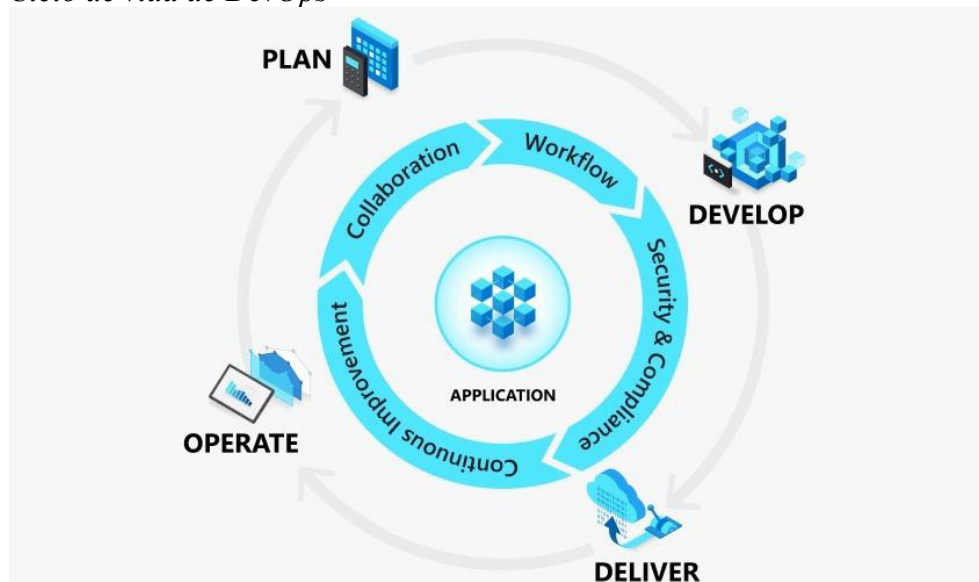
Nota. Fuente: (Silvola & Sargsyan, 2021).

3.2.4.8 Ciclo de vida de DevOps

El ciclo de vida de DevOps guarda una correspondencia con el ciclo de vida de desarrollo de software, de esta forma se reconocen a las siguientes etapas: planificación, desarrollo, entrega y operación. Cada una de estas etapas tiene una relación con otra de ellas, además estas etapas trabajan y se relacionan juntas. Las etapas no definen a un rol o a una sola persona en específico, sino más bien son equipos los que trabajan juntos en cada etapa.

En la Figura 11, se presenta el ciclo de vida de la implementación de DevOps, sus fases y los principales pilares de desarrollo según la compañía Microsoft (MICROSOFT, 2022).

Figura 11
Ciclo de vida de DevOps



Nota. Fuente: Microsoft Azure (2022)

3.2.4.8.1 Fase de Planificación

Según lo definido por Microsoft (MICROSOFT, 2022) se considera a la Fase de Planificación de la siguiente manera:

“En la fase de planificación, los encargados de los equipos de DevOps crean las características y definen las capacidades de las aplicaciones y sistemas que están construyendo. En esta etapa de planificación se sugieren crear las siguientes herramientas para los equipos: tableros de Kanban, creación de las épicas y backlog de actividades. También se crean tareas granulares para poder tener visibilidad y agilidad de los

desarrollos, esto permite además poder corregir errores con rapidez y reaccionar frecuentemente a los cambios que se ven en los progresos entregados en cada Sprint.”

3.2.4.8.2 Fase de Desarrollo

Según lo definido por Microsoft (MICROSOFT, 2022) se considera a la Fase de Desarrollo de la siguiente manera:

“La fase de desarrollo es donde los miembros del equipo realizan las actividades de codificación, pruebas, revisión del software ingresado e integración del nuevo código, así como la generación de ese código en artefactos de construcción que puedan desplegarse en los entornos que corresponden. Los equipos ya muy a menudo utilizan los repositorios de control de versiones para integrar el código y trabajan en paralelo y de forma centralizada y única. También en esta etapa se busca innovar las tecnologías de desarrollo rápidamente sin sacrificar la calidad y la productividad de los integrantes del equipo. Para ello, utilizan herramientas altamente productivas y con soporte para automatizar los pasos manuales, e iteran en pequeños incrementos mediante pruebas automatizadas e integración continua.”

3.2.4.8.3 Fase de Entrega

Según lo definido por Microsoft (MICROSOFT, 2022) se considera a la Fase de Entrega de la siguiente manera:

“La fase de entrega es el proceso donde se despliegue el software desarrollado hacia los entornos productivos, esto idealmente se debería realizar con la entrega continua. Los equipos de DevOps crean y definen un proceso de gestión de despliegues con procesos y documentos de aprobación de forma manual. Además, establecen procesos automatizadas que mueven los sistemas entre las etapas hasta que se ponen a disposición de los clientes. La automatización de todos estos procesos los hace ser repetibles, escalables, controlados y bien probados. De este modo, los equipos que practican DevOps pueden realizar entregas frecuentes con seguridad, tranquilidad y facilidad.”

3.2.4.8.4 Fase de Operación

Según lo definido por Microsoft (MICROSOFT, 2022) se considera a la Fase de Operación de la siguiente manera:

“La fase de operación implica el control, el soporte y la solución de problemas de las aplicaciones ya desplegadas en entornos productivos. Al adoptar las prácticas de

DevOps, los equipos trabajan para garantizar la confiabilidad del sistema, la mayor disponibilidad, al tiempo que refuerzan la seguridad de los sistemas. Los equipos de DevOps emplean prácticas de despliegue seguras para identificar los problemas antes de que afecten a la experiencia del cliente y mitigar los problemas inmediatamente cuando se evidencian. Mantener esta vigilancia requiere herramientas de medición altamente rápidas, que puedan generar las alertas que corresponden y una visibilidad total de las aplicaciones y los sistemas que están siendo implicados.”

3.2.4.9 Servicios de Azure DevOps

3.2.4.9.1 Azure Boards

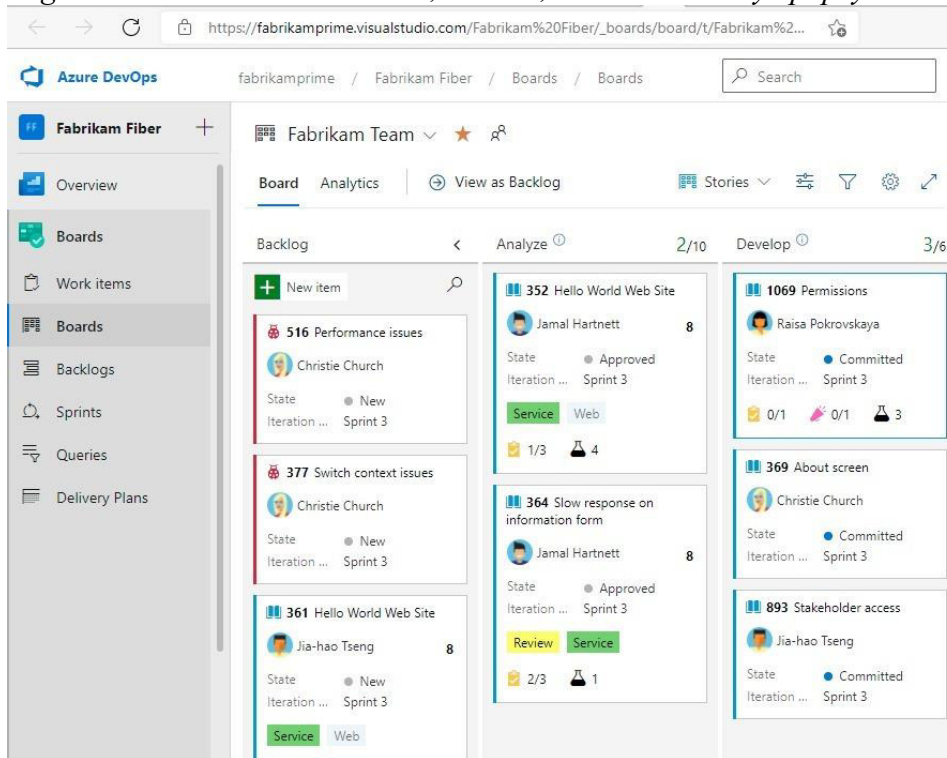
Azure Boards es una gran herramienta que aporte gran valor a las ceremonias que se llevan a cabo en Scrum, tiene muchos tableros que aportan mucha visibilidad e interacción entre los integrantes del equipo, además de poseer un historial de todos los cambios realizados en los tableros que se van generando. Además, esta herramienta es de gran flexibilidad porque tiene muchas vistas y tableros que se pueden configurar según el objetivo y presenta informes que están integrados a las actividades y épicas que se han generado.

Según lo definido por la compañía Microsoft (MICROSOFT, 2022) se considera lo siguiente:

“Azure Boards es un servicio que proporciona a los equipos de desarrollo de software las herramientas dinámicas y fáciles de personalizar que necesitan para gestionar sus proyectos. Estas herramientas además se acomodan a la necesidad de cada compañía. Es posible generar una revisión de las actividades del trabajo que se va llevando a cabo día tras día así como de los bugs y problemas encontrados por los propios desarrolladores.”

En la Figura 12, se muestra un tablero de servicio de Azure Boards con tarjetas creadas como Historias de Usuario, según la compañía Microsoft (MICROSOFT, 2022).

Figura 12
Seguimiento de casos de usuario, errores, características y epopeyas



Nota. Fuente: Microsoft Azure (2022).

3.2.4.9.2 Azure Pipelines

Azure Pipeline es probablemente la herramienta de mayor valor para el trabajo automatizado, porque tiene todos los componentes para generar e integrar procesos de desarrollo y de despliegue. Para el presente trabajo fue fundamental determinar claramente el objetivo que debe de tener cada Pipeline, pero una vez que se pudo determinar los objetivos, los Pipeline son tan flexibles que ayudaron en la construcción de la Integración Continua (CI) y Entrega Continua (CD) de Software.

Un punto muy importante en el presente trabajo con respecto a los Pipelines ha sido la de implementar los proyectos para las pruebas automatizadas, esto sin duda ayudo a detectar errores que pudieran llegar a producción antes de ser lanzados. Con esto seguimos alineados a la agilidad que exige el marco agile de Scrum al cual llevamos de la mano.

Según lo definido por la compañía Microsoft (MICROSOFT, 2022) se considera lo siguiente:

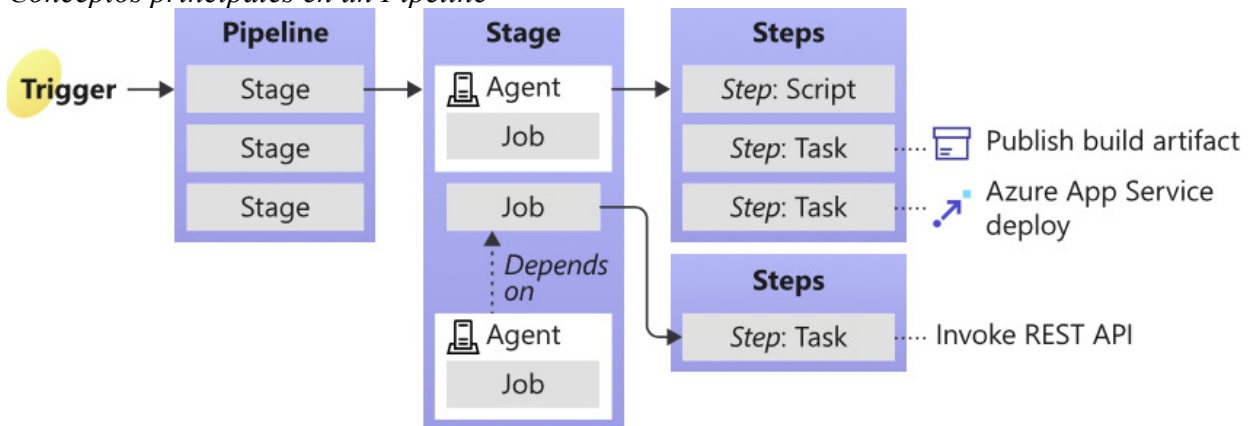
“Azure Pipelines es un servicio que permite construir y probar de forma automática proyectos de código que están escritos en muchos lenguajes y que tienen como

objetivo servir a otros Pipeline o realizar una acción en específico. Azure Pipelines trabaja con CI y CD para crear y probar su código y enviarlo a cualquier destino o acción.”

En la Figura 13 se describe los conceptos principales de un Pipeline según la compañía Microsoft (MICROSOFT, 2022).

Figura 13

Conceptos principales en un Pipeline



Nota. Fuente: Microsoft Azure (2022).

3.2.4.9.3 Azure Repos

Según lo definido por la compañía Microsoft (MICROSOFT, 2022) se considera lo siguiente:

“Azure Repos es un servicio que proporciona un conjunto de herramientas para el seguimiento del control de versiones y así administrar y controlar mejor el código de una aplicación. El control de versiones mantiene un historial del desarrollo a lo largo del tiempo para que pueda tener transparencia y visibilidad e incluso revertir a una versión anterior en la línea de tiempo principal. Azure Repos de Microsoft brinda 2 tipos de control de versiones: Git y Team Foundation (TFVC).”

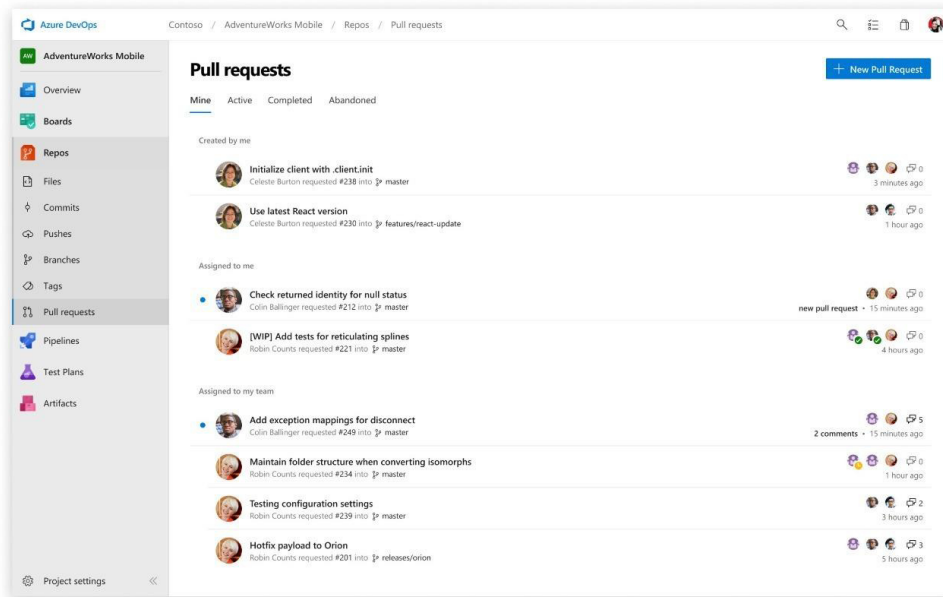
En la experiencia obtenido con esta herramienta de Microsoft, se puede indicar varios beneficios que se tomaron en cuenta para estar mayormente enterados de lo que se ingresa como código por los demás desarrolladores, en el actual proyecto identificamos los siguientes beneficios de usar Azure Repos:

- Permite identificar visualmente las líneas de tiempo por cada rama que se genera. También fue posible regresar en una versión anterior del código cuando se detectó algún inconveniente en el código.

- Nos permitió colocar políticas de integración de Pipeline a cada rama como condición para ingresar a revisión.
- Nos permitió colocar a los aprobadores de los cambios que van a ingresar.

En la Figura 14 se muestra una lista de Pull Request con el servicio de Azure Repos de Microsoft (MICROSOFT, 2022).

Figura 14
Lista de Pull Request generados desde Azure Repos



Nota. Fuente: Azure de Microsoft (2022).

3.2.4.9.4 Azure Test Plans

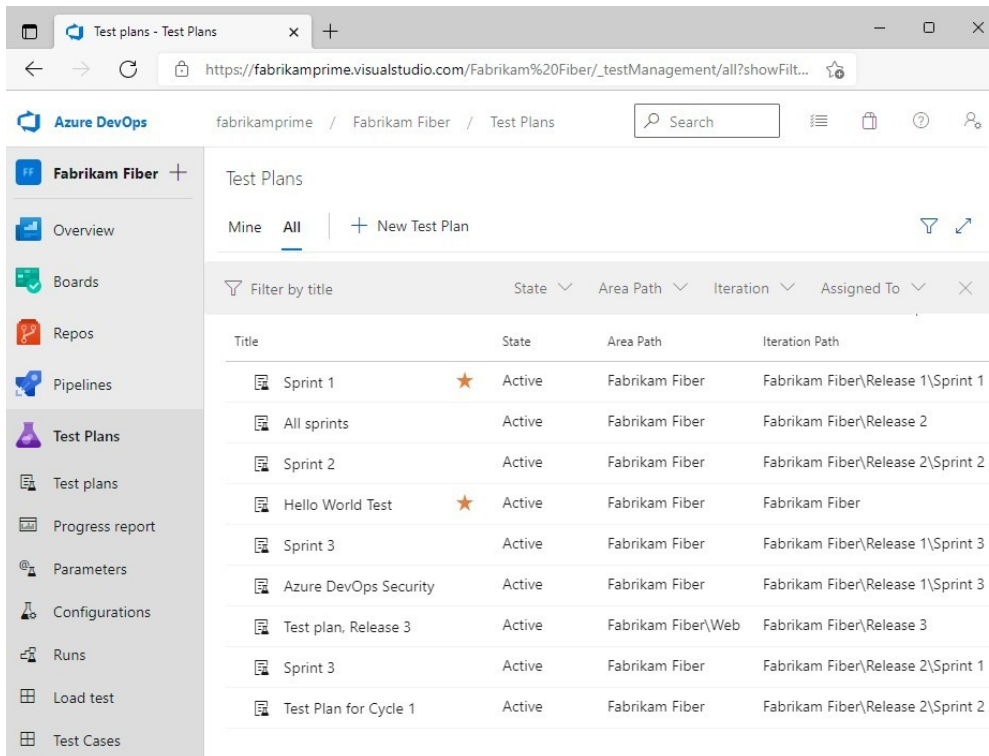
Según lo definido por la compañía Microsoft (MICROSOFT, 2022) se considera lo siguiente:

“Azure Test Plans es un servicio que proporciona un conjunto de herramientas importantes para los equipos de Testing de la compañía ya que permite tener visibilidad de la calidad software que se va desarrollando además de dar comunicación y colaboración entre los integrantes del equipo. La administración de las funciones entregadas por este servicio se va en todos los tipos de pruebas que realizamos tales como pruebas manuales, pruebas de aceptación de usuario y pruebas de exploración.”

En la Figura 15, se muestra una lista de Planes de Prueba generados con el servicio de Azure Test Plans de Microsoft (MICROSOFT, 2022).

Figura 15

Lista de Planes de Prueba generados desde Azure Test Plans.



The screenshot shows the Azure DevOps Test Plans interface for the 'Fabrikam Fiber' project. The left sidebar contains navigation options: Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report, Parameters, Configurations, Runs, Load test, and Test Cases. The main area displays a table of test plans with columns for Title, State, Area Path, and Iteration Path. The table lists ten test plans, including 'Sprint 1', 'All sprints', 'Sprint 2', 'Hello World Test', 'Sprint 3', 'Azure DevOps Security', 'Test plan, Release 3', 'Sprint 3', and 'Test Plan for Cycle 1'. The 'Hello World Test' and 'Sprint 1' entries are marked with a star icon.

Title	State	Area Path	Iteration Path
Sprint 1	Active	Fabrikam Fiber	Fabrikam Fiber\Release 1\Sprint 1
All sprints	Active	Fabrikam Fiber	Fabrikam Fiber\Release 2
Sprint 2	Active	Fabrikam Fiber	Fabrikam Fiber\Release 2\Sprint 2
Hello World Test	Active	Fabrikam Fiber	Fabrikam Fiber
Sprint 3	Active	Fabrikam Fiber	Fabrikam Fiber\Release 1\Sprint 3
Azure DevOps Security	Active	Fabrikam Fiber	Fabrikam Fiber\Release 1\Sprint 3
Test plan, Release 3	Active	Fabrikam Fiber\Web	Fabrikam Fiber\Release 3
Sprint 3	Active	Fabrikam Fiber	Fabrikam Fiber\Release 2\Sprint 1
Test Plan for Cycle 1	Active	Fabrikam Fiber	Fabrikam Fiber\Release 2\Sprint 2

Nota. Fuente: Azure de Microsoft (2022).

3.2.4.9.5 Azure Artifacts

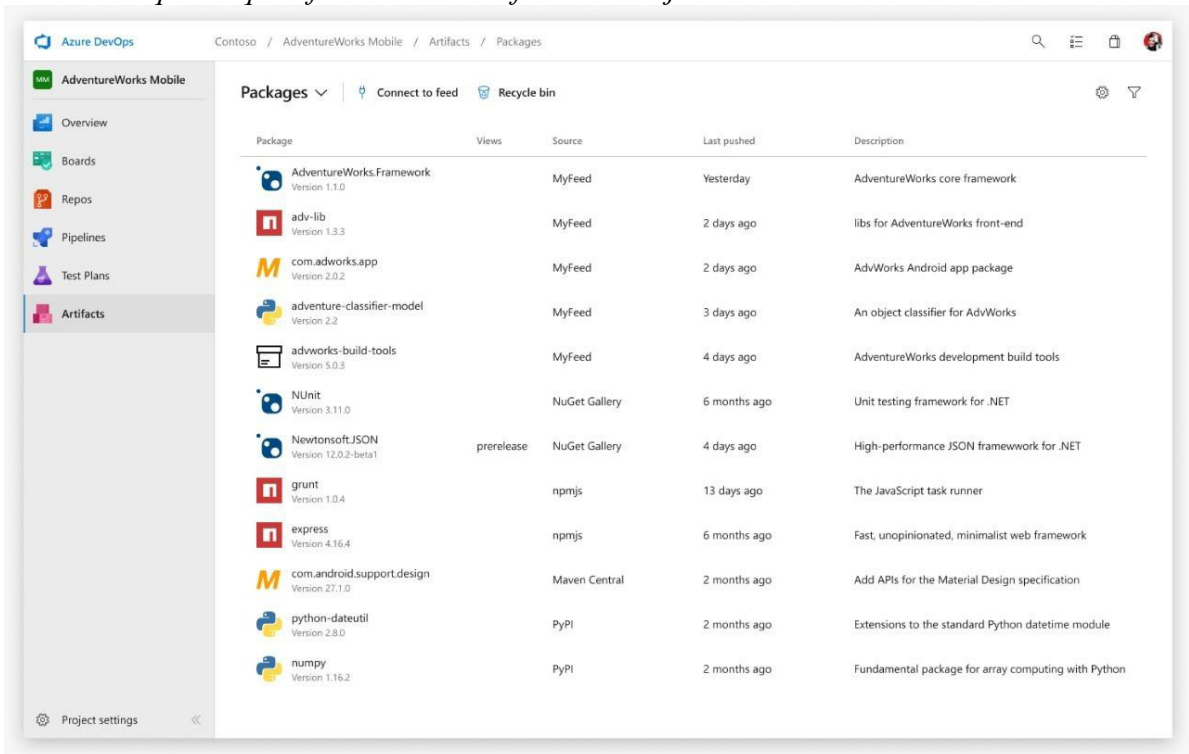
Según lo definido por la compañía Microsoft (MICROSOFT, 2022) se considera lo siguiente:

“Azure Artifacts es un servicio de Microsoft que ofrece un conjunto de herramientas para construir librerías que puedan ser reutilizadas por los desarrolladores de software. El objetivo de este servicio es brindar a los desarrolladores todos los componentes y herramientas para crear sus propias librerías y que estas las puedan compartir con los integrantes de sus propios equipos o con los integrantes que deseen de su propia compañía, en este sentido las librerías pueden crearse de forma pública o privada. Los desarrolladores tienen la opción de consumir paquetes de diferentes orígenes como, por ejemplo: nuget.org o npmjs.com. Algunos tipos de paquetes más frecuentes son: NuGet, NPM, Python y Maven.”

En la Figura 16 se muestra una lista de Paquetes generados con el servicio de Azure Artifacts de Microsoft (MICROSOFT, 2022).

Figura 16

Lista de Paquetes que ofrece Azure Artifacts en la ejecución de sus tareas



Package	Views	Source	Last pushed	Description
AdventureWorks.Framework Version 1.1.0		MyFeed	Yesterday	AdventureWorks core framework
adv-lib Version 1.3.3		MyFeed	2 days ago	libs for AdventureWorks front-end
com.adworks.app Version 2.0.2		MyFeed	2 days ago	AdvWorks Android app package
adventure-classifier-model Version 2.2		MyFeed	3 days ago	An object classifier for AdvWorks
adworks-build-tools Version 5.0.3		MyFeed	4 days ago	AdventureWorks development build tools
NUnit Version 3.11.0		NuGet Gallery	6 months ago	Unit testing framework for .NET
Newtonsoft.JSON Version 12.0.2-beta1	prerelease	NuGet Gallery	4 days ago	High-performance JSON framework for .NET
grunt Version 1.0.4		npmjs	13 days ago	The JavaScript task runner
express Version 4.16.4		npmjs	6 months ago	Fast, unopinionated, minimalist web framework
com.android.support.design Version 27.1.0		Maven Central	2 months ago	Add APIs for the Material Design specification
python-dateutil Version 2.8.0		PyPI	2 months ago	Extensions to the standard Python datetime module
numpy Version 1.16.2		PyPI	2 months ago	Fundamental package for array computing with Python

Nota. Fuente: Azure de Microsoft (2022).

3.2.4.10 Practicas recomendadas para DevOps

DevOps se entiende como una metodología que nos lleva a seguir una serie de pasos, pero, lo que da vida a esta metodología es algunas prácticas recomendadas que ayudan a acelerar, automatizar y mejorar las fases del desarrollo y puesta en producción. De esta forma nos ayuda a mejorar la productividad de la compañía.

A continuación, se presentan algunas prácticas recomendadas para implementar DevOps según la compañía Atlassian.

3.2.4.10.1 Gestión ágil de proyectos

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“En los últimos años las metodologías ágiles han venido tomando mucha presencia en las organizaciones en general y más aun en aquellas que se ocupan del desarrollo de software, pues el principal aporte de este marco de trabajo son las iteraciones cortas que generen el mayor valor para un proyecto. Estas iteraciones de corta duración favorecen a los equipos para crear interacción entre los integrantes e interesados del

proyecto. Cabe destacar que al aumento de interacciones entre los participantes se favorecen las rápidas respuestas y retorno de respuestas.”

3.2.4.10.2 Enfoque Shift Left con CI/ CD

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“Este es un enfoque también conocido como desplazamiento a la izquierda y define que primero debemos realizar pruebas antes y durante todo el proceso de desarrollo de código, esto de la mano de la integración y entrega continua aseguran poder ir un paso adelante de la detección de errores en el código. Se asegura que la realización de esta práctica de pruebas mejora la calidad del código ingresado.”

3.2.4.10.3 Implementa la automatización

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“La automatización de software significa integrar y desplegar el nuevo código de forma inmediata (Integración y Entrega continuas). La CI/ CD tiene como objetivo mejorar y eliminar los procesos de revisión manual que tienen todos los cambios que sufre la rama principal. Para llevar a cabo con éxito un proceso de automatización de software nos apoyamos de pruebas automatizadas que pueden estar definidas por pruebas integrales, prueba unitarias y pruebas de rendimiento.”

3.2.4.10.4 Supervisa la canalización y las aplicaciones DevOps

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“Es muy importante poder contar con la revisión continua de la canalización del flujo de desarrollo con DevOps. Se sugieren que cuando se encuentren con errores en la compilación de los procesos de test automatizados se realizan las pruebas manuales de la funcionalidad para no afectar el ambiente productivo del cliente. Sabemos que la automatización de software mejora el tiempo de entrega de los requerimientos, pero es tan importante además entregar un código con las pruebas superadas y sin errores.”

3.2.4.10.5 Recopila feedback continuo

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“La recopilación del feedback continuo sirve para que el equipo de desarrollo pueda recibir las notificaciones de observaciones de cambios de forma oportuna y a tiempo. La práctica continua de la recopilación sirve además para poder corregir y mejorar el rendimiento de las aplicaciones en producción, esto hace un equilibrio entre la calidad y la velocidad del desarrollo. Durante mucho tiempo atrás se conocía que los procesos de desarrollo de software podían tener calidad o velocidad en la entrega, y este practica ayuda a mejorar ambos puntos.”

3.2.4.10.6 Cambia la cultura

Según lo definido por el autor Hall Tom en un artículo para Atlassian (Hall, 2022) se considera lo siguiente:

“DevOps inicialmente debe de contar con los valores de comunicación confianza, empatía y transparencia, esto facilita el trabajo y las aclara las responsabilidades de los involucrados. Es difícil romper y dejar atrás los conceptos de responsabilidad única cuando tú eres quien ha creado una información, y dejar este concepto atrás permite que los equipos trabajemos con responsabilidades únicas pero colaborativas y que además fortalezca la comunicación.”

3.2.4.11 DevOps y Metodología Ágil

Tanto la metodología DevOps como cualquier metodología ágil ofrecen una serie de pasos y un marco de trabajo que pueden acelerar la entrega de software y mejorar la productividad y comunicación de los equipos. Un enfoque ágil es bueno con respecto a los lineamientos para ordenar a los equipos mientras que DevOps ofrece una cultura más amplia de entrega de software, y ambos alineados con la rapidez y confiabilidad, permite generar madurez a la compañía generando automatización y auto organización. La cuestión de estas 2 metodologías es como integrarlas para que trabajen de la mano y optimicen procesos manuales.

3.2.5 Implementación de las áreas de proceso y sus buenas practicas

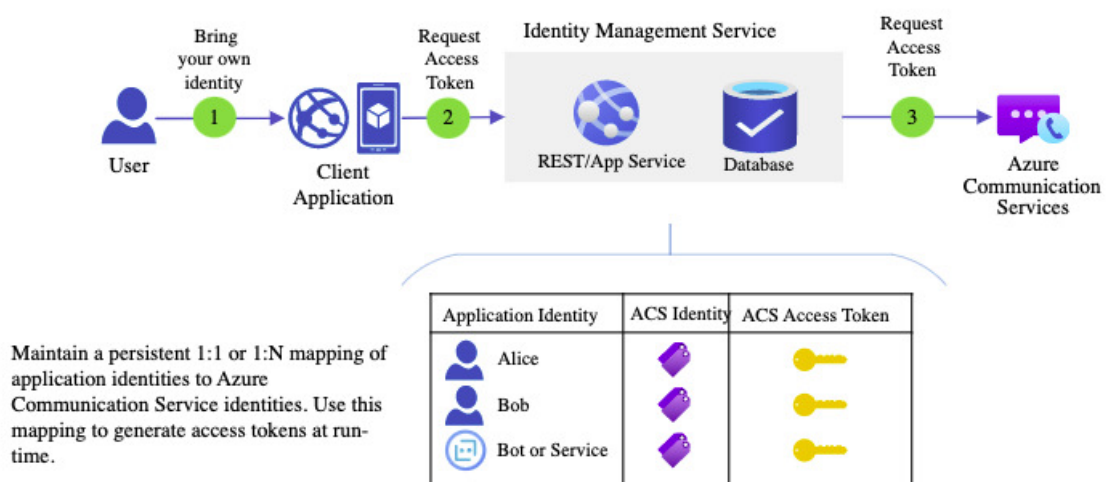
3.2.5.1 Diseño de la Arquitectura AS-IS

La arquitectura existente corresponde al modelo de cliente servidor, esta consiste en la instalación de manera On Premise, de esta manera cada cliente mantenía una instalación única y propia. Este modelo de arquitectura contempla una comunicación de red local donde los datos pertenecen a cada cliente.

En la Figura 17, se presenta la descripción del modelo cliente servidor según la compañía Microsoft (MICROSOFT, 2022).

Figura 17

Diagrama de la Topología Cliente Servidor



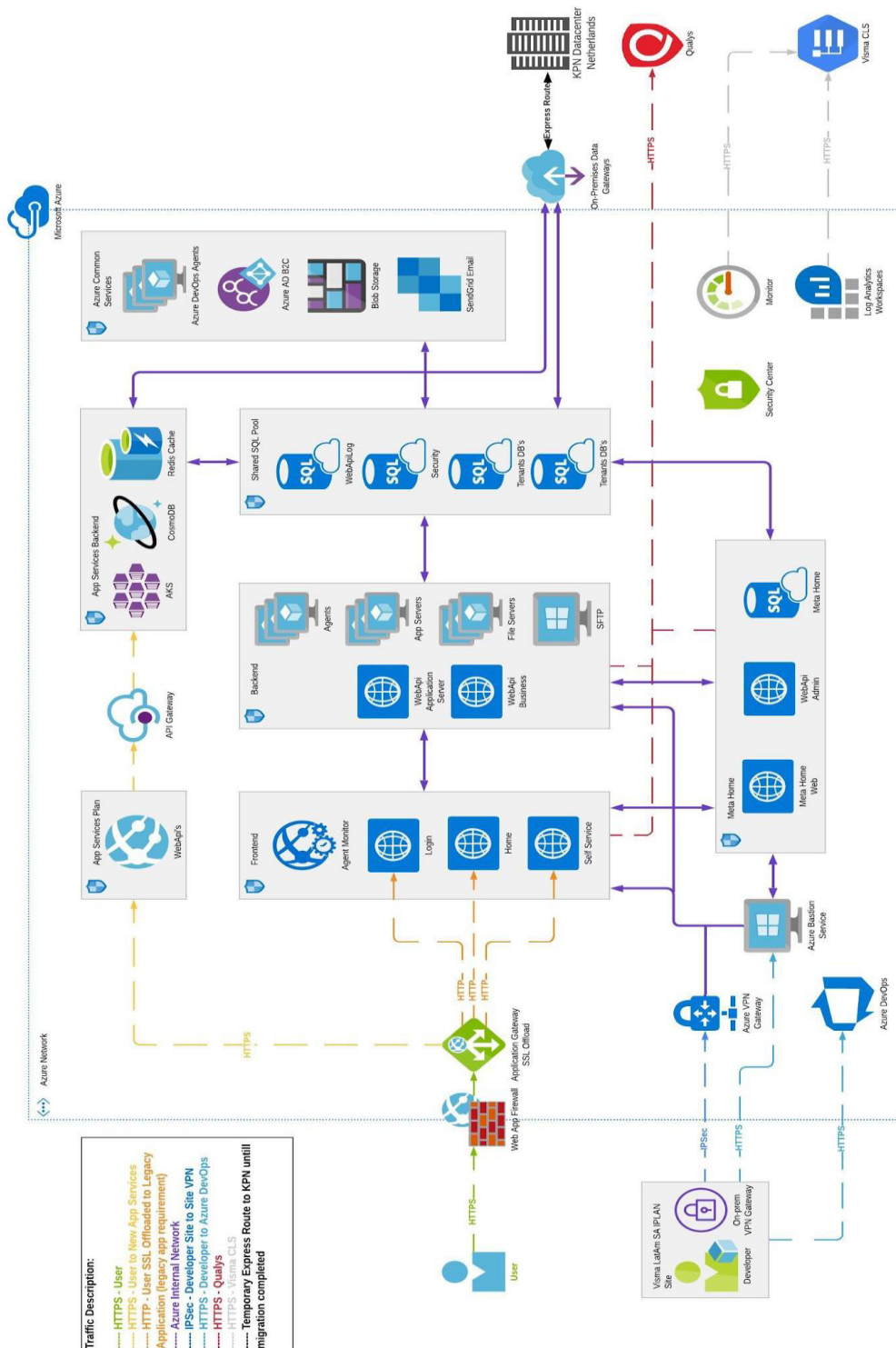
Nota. Fuente. Tomada Azure Microsoft, (2022).

3.2.5.2 Diseño de la Arquitectura TO-BE

En la arquitectura propuesta se contempló primero contar con una VPN Gateway además de un Firewall con protocolo de seguridad HTTPS que pueda garantizar a los clientes el mismo nivel de seguridad y privacidad en sus sistemas, además se propuso trabajar en la nube de Azure con un modelo basado totalmente en la nube y los frentes de desarrollo de FrontEnd y BackEnd.

A continuación, en la Figura 18, se presenta el Diagrama de la Topología de la arquitectura para el presente trabajo presentado por la compañía.

Figura 18
Diagrama de la Topología de infraestructura de la migración



Nota. Fuente. Tomada de la compañía, (2021).

3.2.5.3 Planificación de actividades del equipo DevOps.

Según la arquitectura mostrada en el punto anterior se desprenden las actividades y tareas para los equipos de desarrollo de BackEnd, FrontEnd, además de contar con las actividades para el equipo de DevOps.

En la Tabla 14 que se presenta a continuación, se visualizan las actividades principales que realizará el equipo de DevOps.

Tabla 14
Desarrollo de actividades del equipo de DevOps

Categoría	Descripción	Tipo	Prioridad
Arquitectura	Mapa de Arquitectura	Crear	Alta
WebApi Admin	Pasar a AppService WebApi	Mejoras	Alta
Agentes	Pasar a GIT	Mejoras	Baja
VLCA	Despliegue de DEV, TEST y PROD	Configurar	Media
Azure DevOps	Actualizar APIM por Release	Mejoras	Media
Azure DevOps	Tokenizar YAML AKS CD MS	Mejoras	Media
	Producción Auto escalamiento		

Nota. Fuente: Tomado de la compañía (2022).

En la Tabla 15 que se presenta a continuación, se visualizan las actividades principales que realizará el equipo de BackEnd.

Tabla 15
Desarrollo de actividades del equipo de BackEnd

Categoría	Descripción	Tipo	Prioridad
Agentes	Abortar procesos trabados	Mejoras	Alta
Agentes	Bug fixing	Mejoras	Alta
VLCA	Servicio Database	Crear	Baja
VLCA	Auditoria	Configurar	Media
VLCA	Servicio Subscriptions	Crear	
VLCA	Servicio IntegrationTool	Crear	
VLCA	Servicio CloudAdmin	Crear	
VLCA	Servicio ProcessManager	Mejoras	
VLCA	Servicio SystemNotifications	Crear	
VLCA	Servicio Users	Crear	
VLCA	Servicio Metrics	Crear	
VLCA	Servicio Config Tenants Schema	Crear	
WebApi Negocio	Servicio Files	Crear	
WebApi Negocio	Pasar a AppService	Mejoras	
VLCA	Copiar conceptos	Crear	Baja

VLCA	Copiar conceptos	Crear	Baja
------	------------------	-------	------

Nota. Fuente: Tomado de la compañía (2022).

En la Tabla 16 que se presenta a continuación, se visualizan las actividades principales que realiza el equipo de FrontEnd.

Tabla 16

Desarrollo de actividades del equipo de FrontEnd

Categoría	Descripción	Tipo	Prioridad
VLCA	System Notifications	Crear	Baja
VLCA	Suscripciones refactor	Mejoras	Baja
VLCA	Cientes refactor	Mejoras	Baja
VLCA	Full - alta BBDD	Crear	Media
VLCA	Security	Mejoras	
VLCA	Tags	Crear	
VLCA	Tenants	Crear	
VLCA	Motores	Crear	
VLCA	Servers	Crear	
VLCA	Cientes	Crear	
VLCA	Database	Crear	
VLCA	Subscriptions	Crear	
VLCA	Query Mgmt	Crear	
VLCA	Recurring Jobs	Crear	
VLCA	Settings	Crear	
VLCA	Users	Crear	
VLCA	IntegrationTool	Crear	
VLCA	Agents	Crear	
VLCA	Metrics	Crear	
VLCA	Applications	Crear	
VLCA	Tools/Monitor/Processess	Crear	
VLCA	Oauth2	Crear	

Nota. Fuente: Tomado de la compañía (2022).

3.2.5.4 Planificación de las APIs para la migración

En la Tabla 17 que se presenta a continuación, se visualizan los Microservicios y el listado de Apis que se desarrollan por parte del equipo de BackEnd.

Tabla 17

Listado de APIs que se crearon en el proyecto de migracion

Servicio	Controller	Verbo	EndPoint
agents	ProcessInstances	PUT	process-instances/{id}
agents	ProcessInstances	PATCH	process-instances/{id}
agents	ProcessInstances	POST	process-instances/{id}/process-logs
agents	ProcessInstances	POST	process-instances/sync
clients	Clients	GET	clients
clients	Clients	PATCH	clients/{id}
clients	Clients	DELETE	clients/{id}
clients	Clients	POST	clients
clients	Clients	GET	clients/{id}
cloud-admin	Servers	GET	servers
cloud-admin	Servers	PATCH	servers/{id}
cloud-admin	Servers	DELETE	servers
cloud-admin	Servers	POST	servers
cloud-admin	Servers	GET	servers/{id}
cloud-admin	DatabaseEngines	GET	database-engines
cloud-admin	DatabaseEngines	POST	database-engines
cloud-admin	DatabaseEngines	DELETE	database-engines
cloud-admin	DatabaseEngines	PATCH	database-engines/{id}
cloud-admin	DatabaseEngines	GET	database-engines/{id}
cloud-admin	DatabaseEngines	GET	database-engines/{id}/tenats {filters}
cloud-admin	DatabaseEngines	GET	database-types
cloud-admin	DatabaseEngines	GET	database-versions
cloud-admin	Tenants	GET	tenants
cloud-admin	Tenants	POST	tenants
cloud-admin	Tenants	PATCH	tenants/{id}
cloud-admin	Tenants	GET	tenants/{id}
cloud-admin	Tenants	GET	tenant-types
cloud-admin	Client	GET	clients
cloud-admin	Client	PATCH	clients/{id}
cloud-admin	Client	GET	partners
cloud-admin	Client	PUT	clients
cloud-admin	Client	POST	clients
cloud-admin	Client	GET	clients/{id}
cloud-admin	Settings	GET	settings
cloud-admin	Settings	POST	settings
cloud-admin	Settings	PUT	settings

cloud-admin	Settings	PATCH	settings
cloud-admin	Settings	GET	settings/{Key}
cloud-admin	Settings	DELETE	settings/{Key}
cloud-admin	Settings	PUT	settings/{id}
cloud-admin	Settings	DELETE	settings/{id}
cloud-admin	Settings	GET	tenants/{tenantId}/settings
cloud-admin	Settings	PUT	tenants/{tenantId}/settings/{id}
cloud-admin	Settings	DELETE	tenants/{tenantId}/settings/{id}
cloud-admin	Settings	POST	tenants/{tenantId}/settings
cloud-admin	Settings	GET	tenants/{tenantId}/settings
cloud-admin	Settings	PUT	tenants/{tenantId}/settings/{id}
cloud-admin	QueryMgmt	GET	categories
cloud-admin	QueryMgmt	GET	categories/{categoryId}/templates
cloud-admin	QueryMgmt	GET	templates/{templateId}/cutom-fields
cloud-admin	QueryMgmt	GET	executions
cloud-admin	QueryMgmt	POST	executions
database	Database	GET	databases
database	Database	POST	databases
database	Database	DELETE	databases
database	Database	GET	databases/{id}
integration-tool		GET	tenants/{tenantId}/settings
integration-tool		POST	tenants/{tenantId}/settings
integration-tool		PATCH	tenants/{tenantId}/settings
integration-tool		GET	tenants/{tenantId}/settings/{id}
integration-tool		POST	tenants/{tenantId}/settings
integration-tool		PATCH	tenants/{tenantId}/settings/{id}
integration-tool		GET	interfaces
integration-tool		POST	interfaces
integration-tool		PATCH	interfaces
integration-tool		GET	interfaces/{interfaceId}/instances
integration-tool		POST	interfaces/{interfaceId}/instances
files	Files	DELETE	files
files	Files	POST	files
files	Files	GET	files
files	Files	PATCH	files
files	Files	GET	files/path
files	Templates	GET	templates
files	Templates	POST	templates
files	Files	GET	files/employee-photos
integration-tool		PATCH	interfaces/{interfaceId}/instances
integration-tool		GET	interfaces/{interfaceId}/instances/{instanceId}/steps
integration-tool		POST	interfaces/{interfaceId}/instances/{instanceId}/steps

integration-tool		PATCH	interfaces/{interfaceId}/instances/{instanceId}/steps/{id}
process-manager		GET	jobs
process-manager		PATCH	types
subscriptions	Applications	GET	applications
subscriptions	Applications	GET	applications/{id}
subscriptions	Applications	POST	applications
subscriptions	Applications	PATCH	applications/{id}
subscriptions	Subscriptions	GET	subscriptions
subscriptions	Subscriptions	POST	subscriptions
processes		GET	processes
processes		PATCH	processes
processes		GET	process-instances
processes		POST	process-instances
processes		GET	process-instances/{id}
processes		GET	tenants/{id}/process-instances/{id}
processes		PATCH	tenants/{id}/process-instances/{id}
subscriptions	Subscriptions	PUT	subscriptions/{id}
subscriptions	Subscriptions	GET	tenants/{tenantId}/subscriptions
subscriptions	Subscriptions	PATCH	tenants/{tenantId}/subscriptions/{id}
subscriptions	Subscriptions	GET	clients/{clientId}/subscriptions
subscriptions	Subscriptions	PATCH	clients/{clientId}/subscriptions/{id}
users		GET	clients
users		GET	common-policies
users		GET	common-roles
users		PATCH	internal-users/{userId}
users		GET	tenants/{tenantId}/internal-users
users		PATCH	tenants/{tenantId}/internal-users
users		GET	users
users		POST	users
users		PUT	users
users		GET	users-gruped-roles
users		GET	users/{userId}
users		POST	users/{userId}
users		PATCH	users/{userId}
users		GET	users/{userId}/not-assigned-tenants
users		DELETE	users/{userId}/policy
users		PUT	users/{userId}/policy
users		DELETE	users/{userId}/roles
users		PUT	users/{userId}/roles

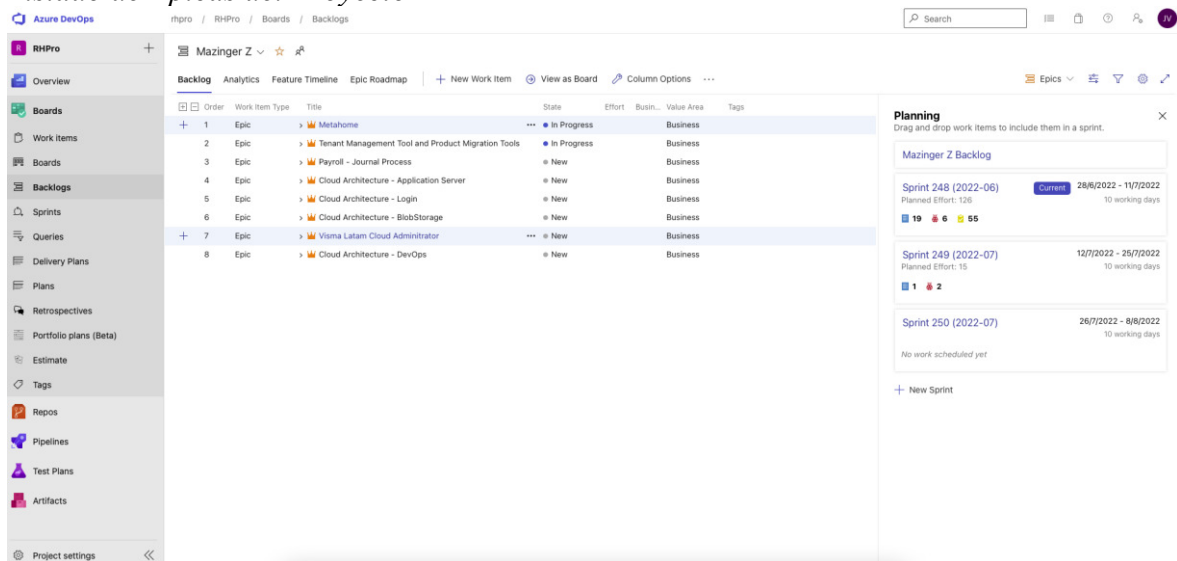
Nota. Fuente: Tomado de la compañía (2022).

3.2.5.5 Épicas, Historias de Usuario y Sprints del Proyecto

Azure DevOps nos proporciona las herramientas y opciones necesarias para poder cumplir con los lineamientos de Scrum en la gestión del Proyecto. Se hace uso del servicio de Azure Boards para crear los Backlogs, Épicas e Historias de Usuario para el presente sistema.

En la Figura 19 que se presenta a continuación, se describe el listado y la creación de Épicas para el presente proyecto a través de la opción de Backlogs.

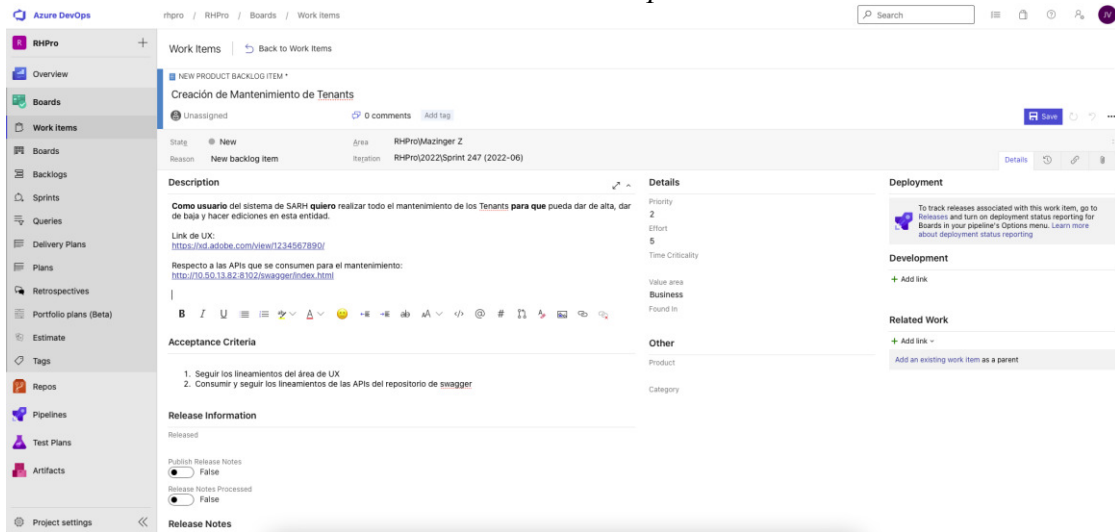
Figura 19
Listado de Épicas del Proyecto



Nota. Fuente: Tomada de la compañía (2022)

En la Figura 20 que se presenta a continuación, se describe la creación de una Historia de Usuario y los criterios de aceptación, así como el esfuerzo que se llevará a cabo para la actividad registrada.

Figura 20
Creación de Historia de Usuario con Azure DevOps Boards



Nota. Fuente: Elaborado por el autor (2022).

3.2.5.6 Creación de App Services para FrontEnd

En la Figura 21 que se presenta a continuación, se visualiza la creación y configuración del AppService que alojara el proyecto de FrontEnd.

Figura 21
Creación de AppService para FrontEnd

Microsoft Azure

Inicio > App Services >

Crear aplicación web

Detalles del proyecto

Seleccione una suscripción para administrar los recursos implementados y los costos. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción *

Grupo de recursos * [Crear nuevo](#)

Detalles de instancia

¿Necesita una base de datos? [Pruebe la nueva experiencia de web y base de datos.](#)

Nombre * .azurewebsites.net

Publicar * Código Contenedor Docker Aplicación web estática

Pila del entorno en tiempo de ejecución *

Sistema operativo * Linux Windows

Región * ¿No encuentra su plan de App Service? Pruebe otra región o seleccione su App Service Environment.

Plan de App Service

El plan de tarifa de App Service determina la ubicación, las características, los costos y los recursos del proceso asociados a la aplicación. [Más información](#)

Plan de Linux (Central US) * [Crear nuevo](#)

SKU y tamaño * **Básico B1**
Total de ACU: 100, 1.75 GB de memoria
[Cambiar el tamaño](#)

[Revisar y crear](#) [< Anterior](#) [Siguiente: Implementación >](#)

Nota. Fuente: Elaborado por el autor (2022).

3.2.5.7 Implementación de repositorios con Azure Repos

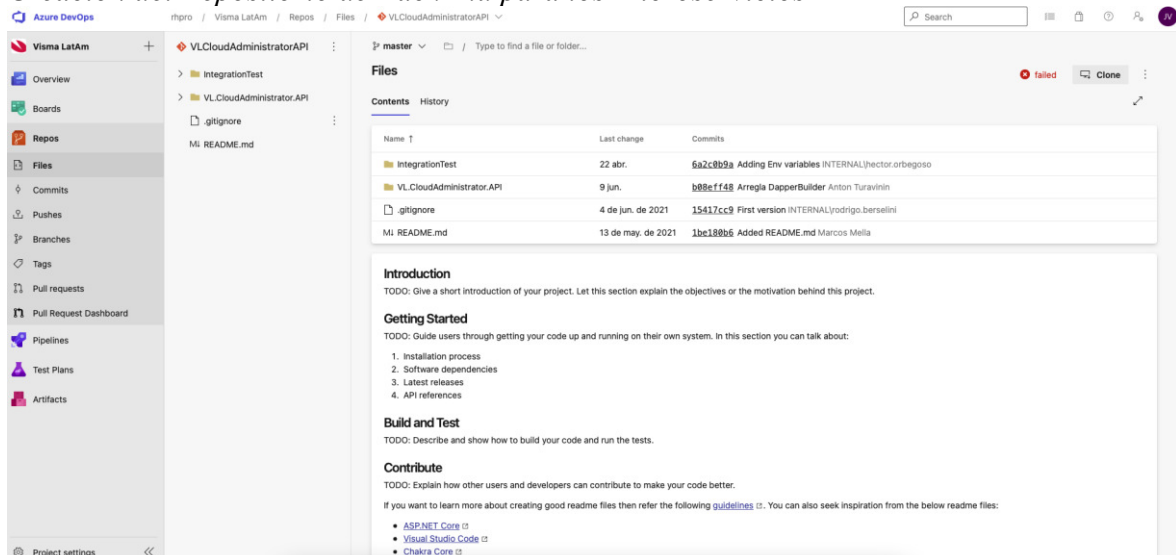
Para la creación de los repositorios se tuvo en cuenta como configuraciones las siguientes características:

- Indicar un número mínimo de 2 revisores cuando se ingrese un Pull Request hacia la rama master.
- Indicar un Pipeline de validación a la rama master cuando ingresen nuevos cambios a través del Pull Request.

En la Figura 22 que se presenta a continuación, se muestra el repositorio de BackEnd para los Microservicios que expondrán las Apis del sistema.

Figura 22

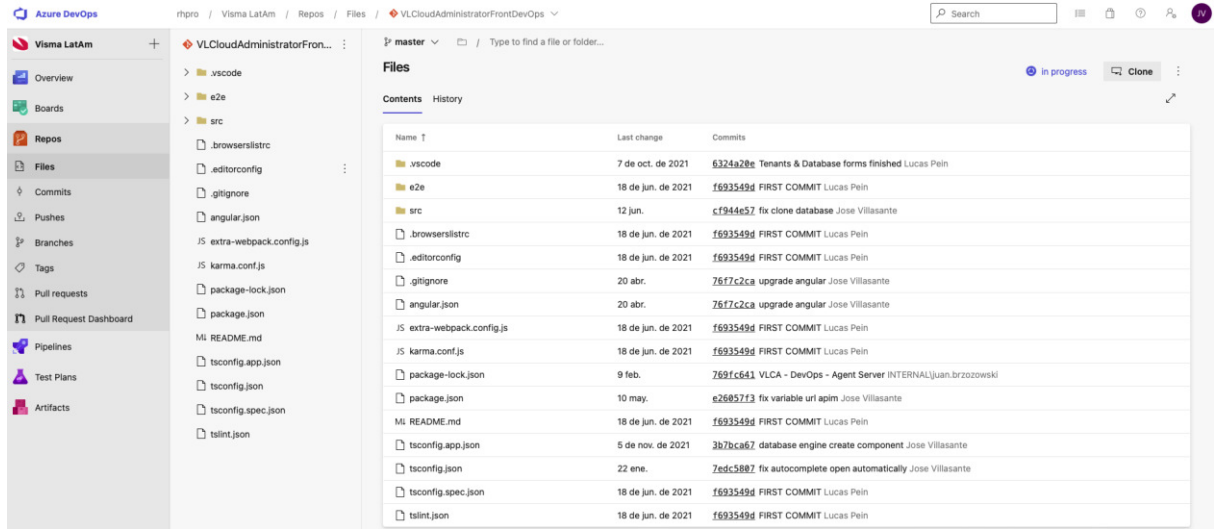
Creación del Repositorio de BackEnd para los Microservicios



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 23 que se presenta a continuación se muestra el repositorio de FrontEnd para el proyecto de Angular del sistema.

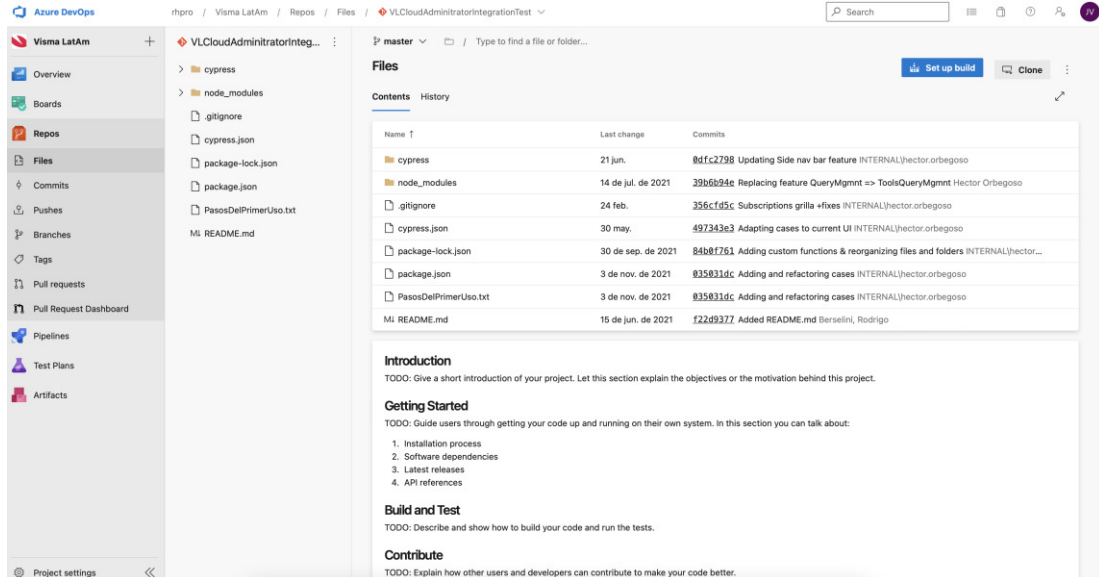
Figura 23
Creación del Repositorio de FrontEnd para los MicroFrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 24 que se presenta a continuación se muestra el repositorio de QA para el proyecto de Cypress que ejecutaran las pruebas automatizadas.

Figura 24
Creación del Repositorio de QA para las pruebas con Cypress

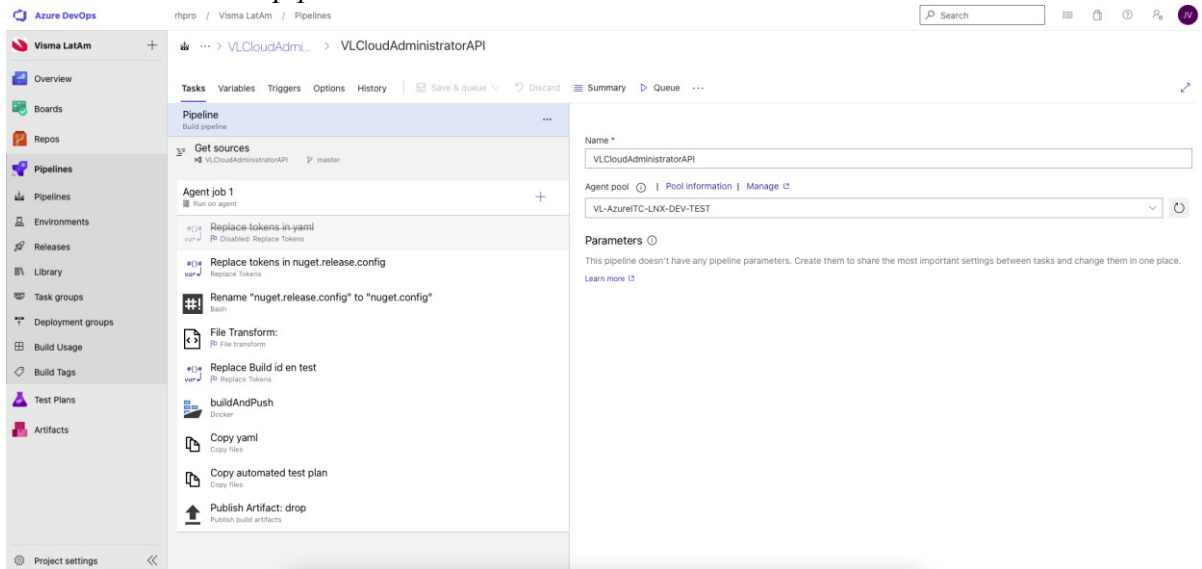


Nota. Fuente: Tomada de la compañía (2022).

3.2.5.8 Pipelines y despliegues de BackEnd y FrontEnd con CI.

En la Figura 25 que se presenta a continuación, se muestra la creación de Pipeline de BackEnd con las tareas que se lanzaran durante su ejecución.

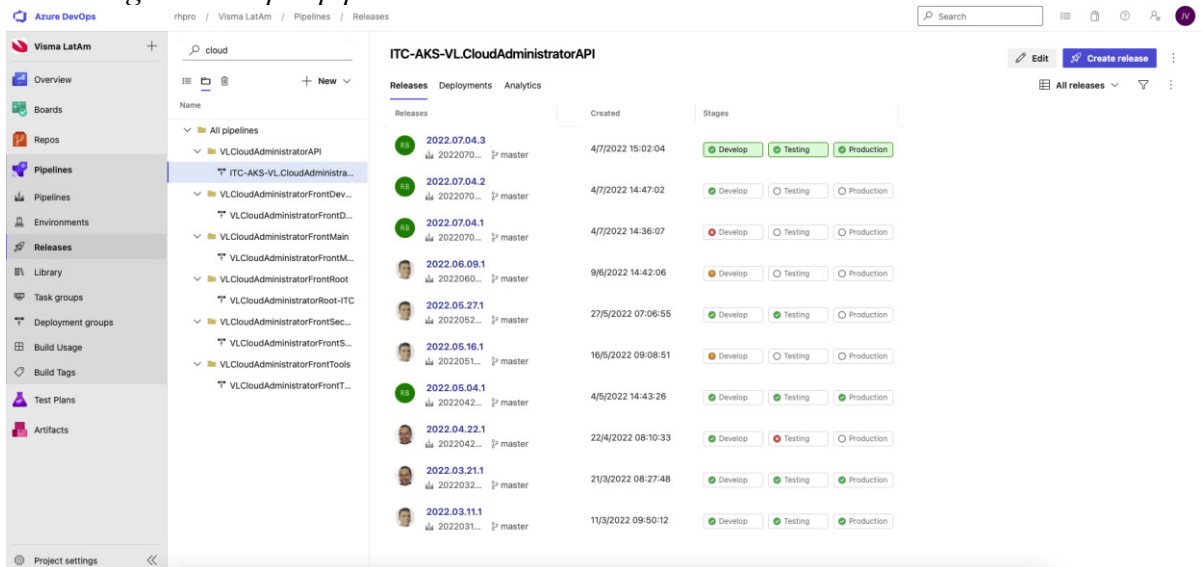
Figura 25
Listado de tareas de pipeline de BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 25 que se presenta a continuación, se visualiza los Releases generados con cada ejecución de Pipeline en BackEnd.

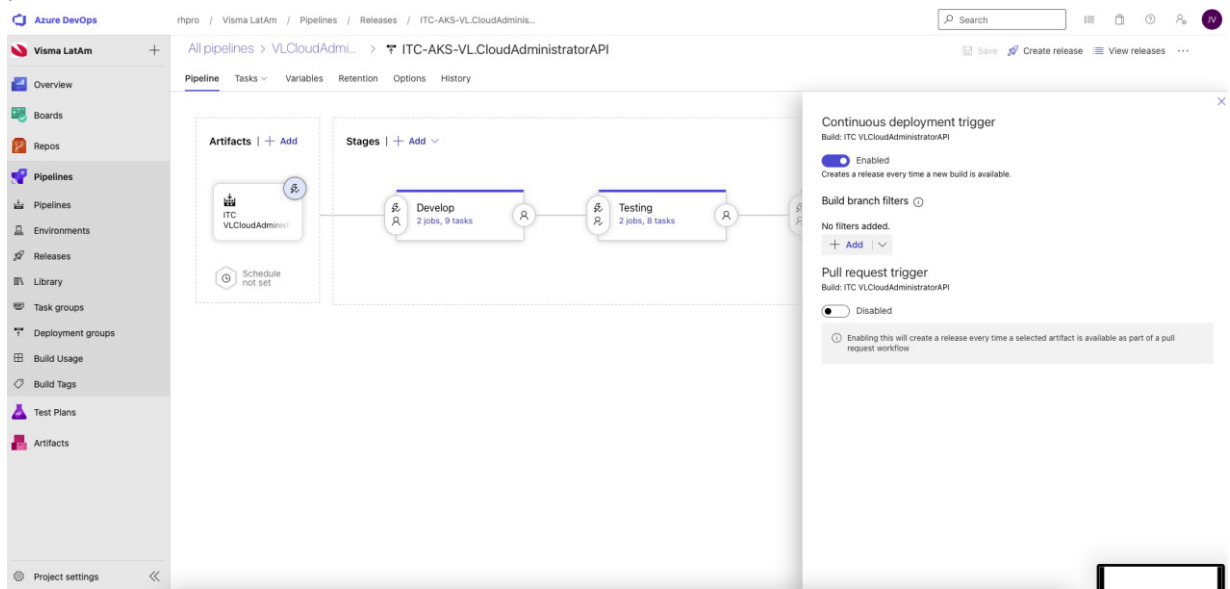
Figura 26
Releases generados por pipeline de BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 27 que se presenta a continuación, se visualiza la configuración para la Integración continua en los Pipeline de BackEnd.

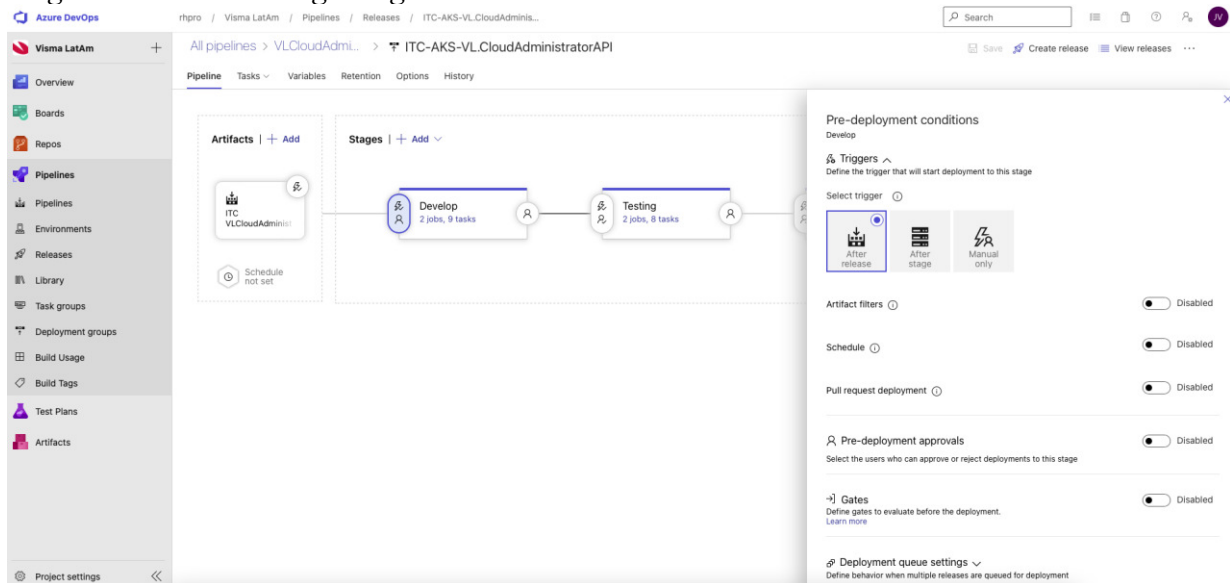
Figura 27
Integración continua de artefacto de pipeline de BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 28 que se presenta a continuación, se muestra la configuración del Stage Develop del BackEnd.

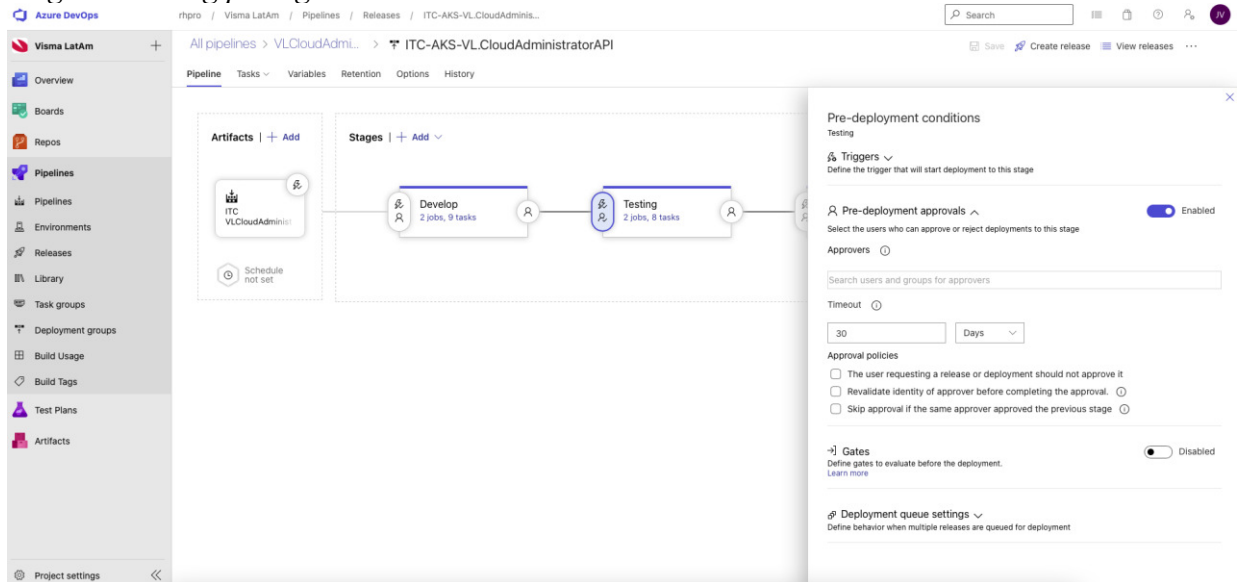
Figura 28
Stage de Desarrollo luego de generada una release en BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 29 que se presenta a continuación, se muestra la configuración del Stage Testing del BackEnd.

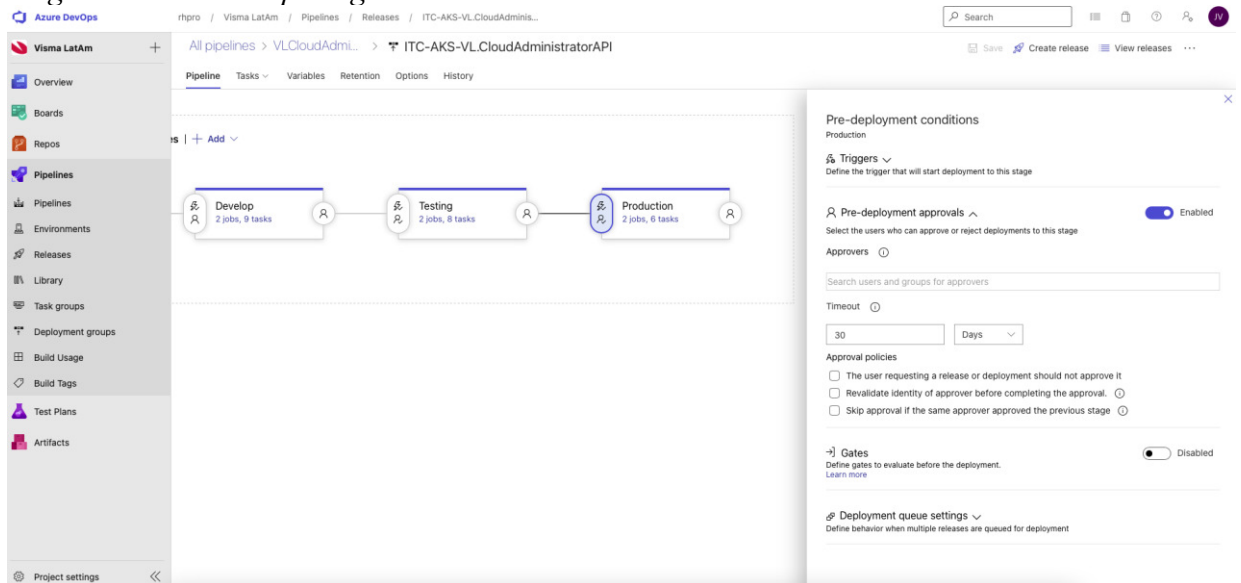
Figura 29
Stage de Testing para generar release en BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 30 que se presenta a continuación, se muestra la configuración del Stage Production del BackEnd.

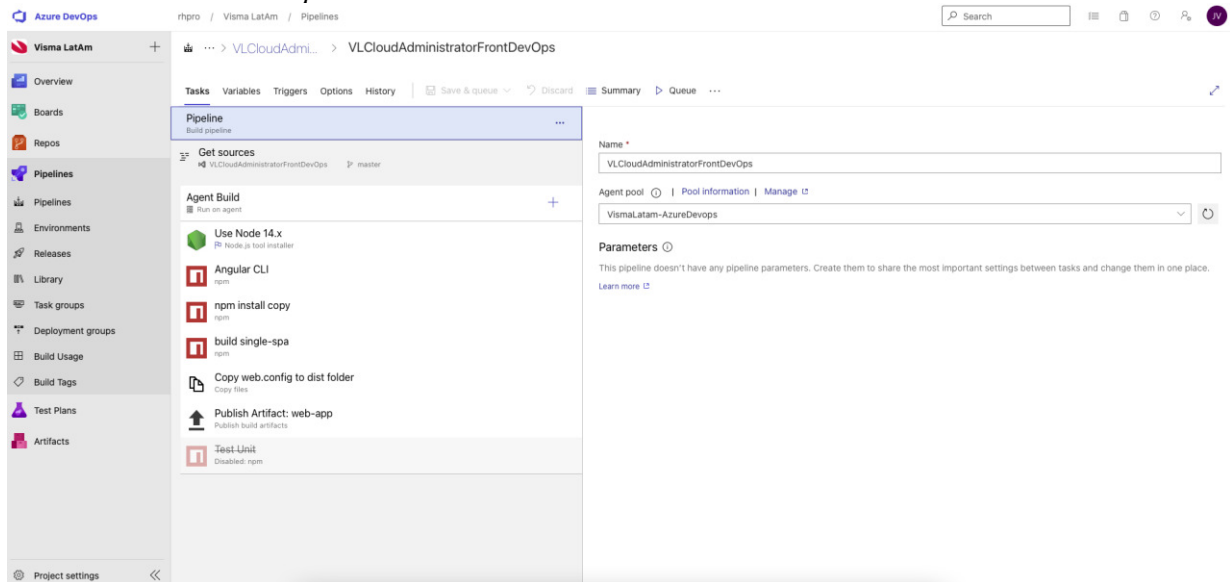
Figura 30
Stage de Producción para generar release en BackEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 31 que se presenta a continuación, se muestra la creación de Pipeline de FrontEnd con las tareas que se lanzaran durante su ejecución.

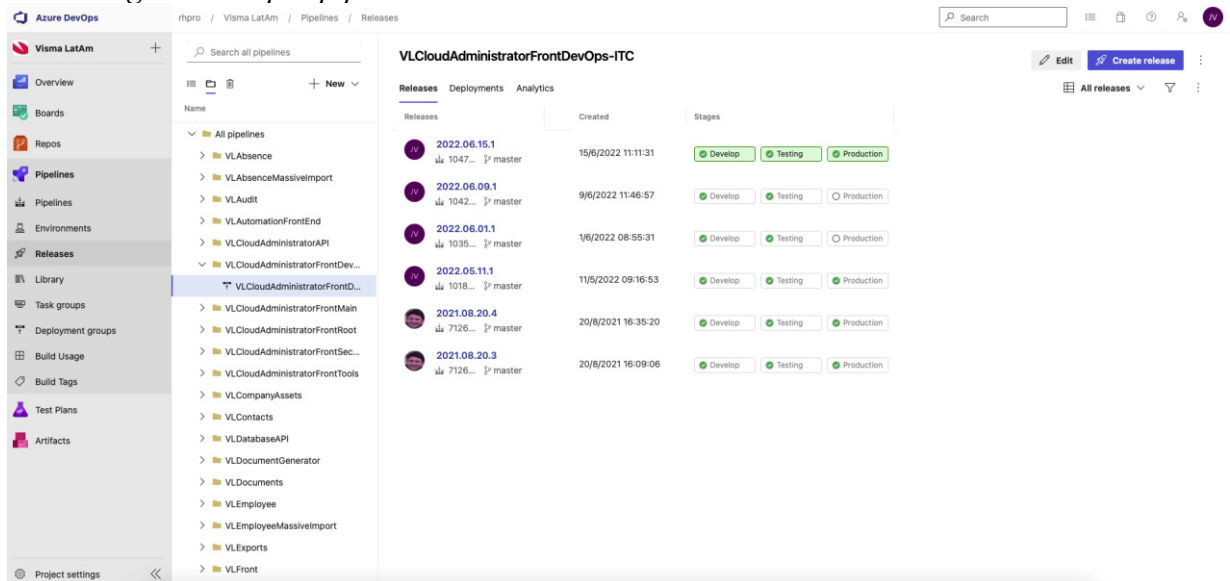
Figura 31
Listado de tareas de Pipeline de FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 32 que se presenta a continuación, se visualiza los Releases generados con cada ejecución de Pipeline en FrontEnd.

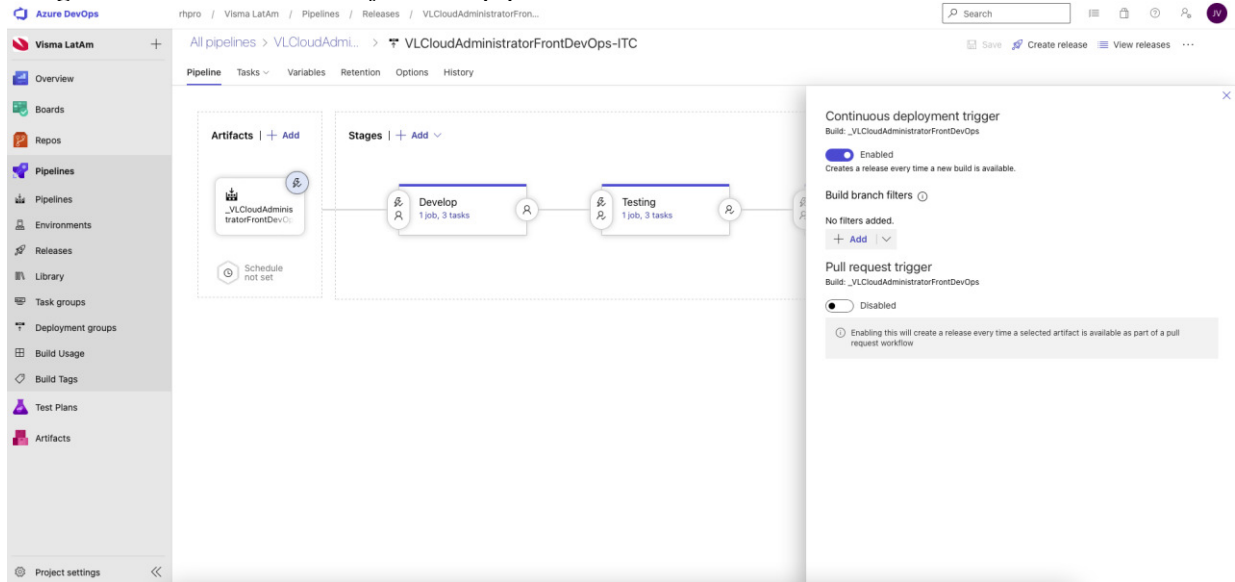
Figura 32
Releases generados por pipeline de FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 33 que se presenta a continuación, se visualiza la configuración para la Integración continua en los Pipeline de FrontEnd.

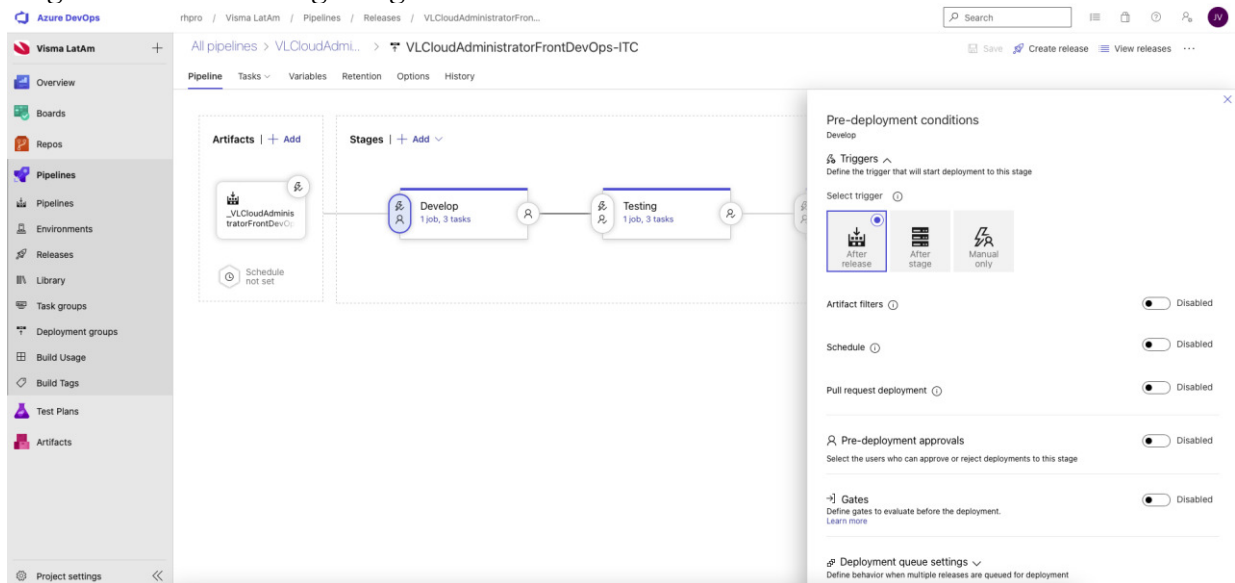
Figura 33
Integración continua de artefacto de pipeline de FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 34 que se presenta a continuación, se muestra la configuración del Stage Develop del FrontEnd.

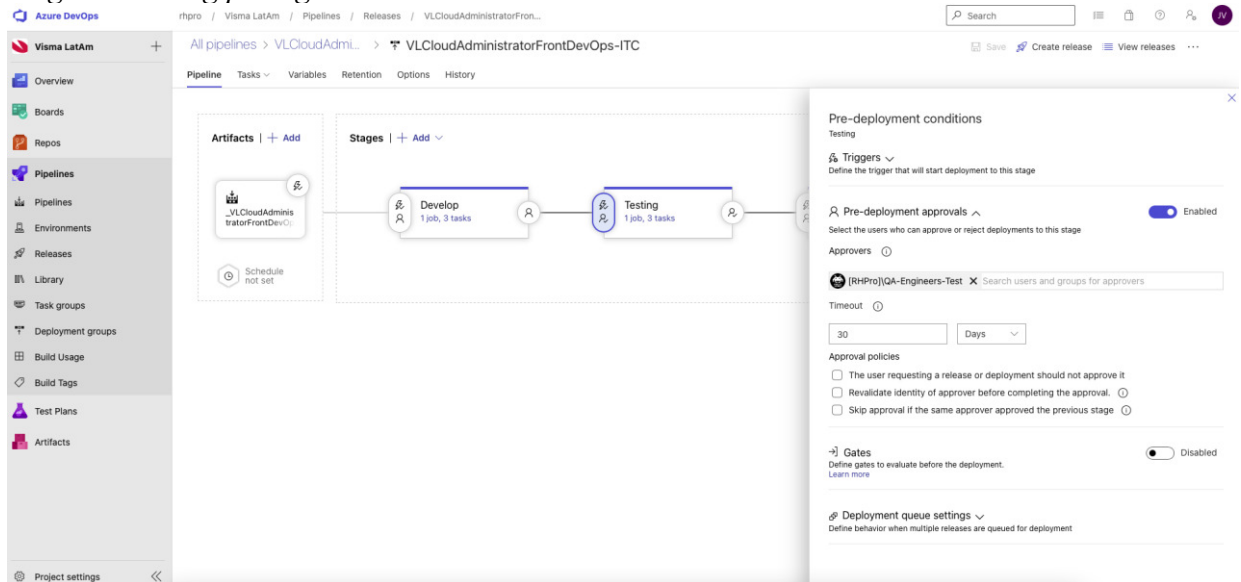
Figura 34
Stage de Desarrollo luego de generada una release en FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 35 que se presenta a continuación, se muestra la configuración del Stage Testing del FrontEnd.

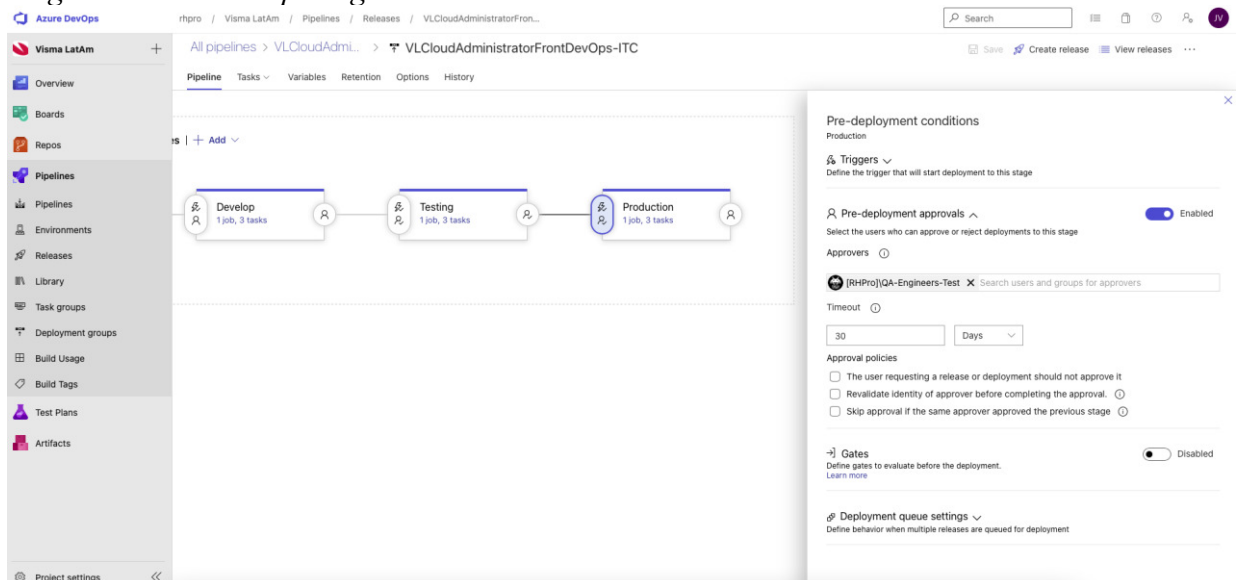
Figura 35
Stage de Testing para generar release en FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 36 que se presenta a continuación, se muestra la configuración del Stage Production del FrontEnd.

Figura 36
Stage de Producción para generar release en FrontEnd



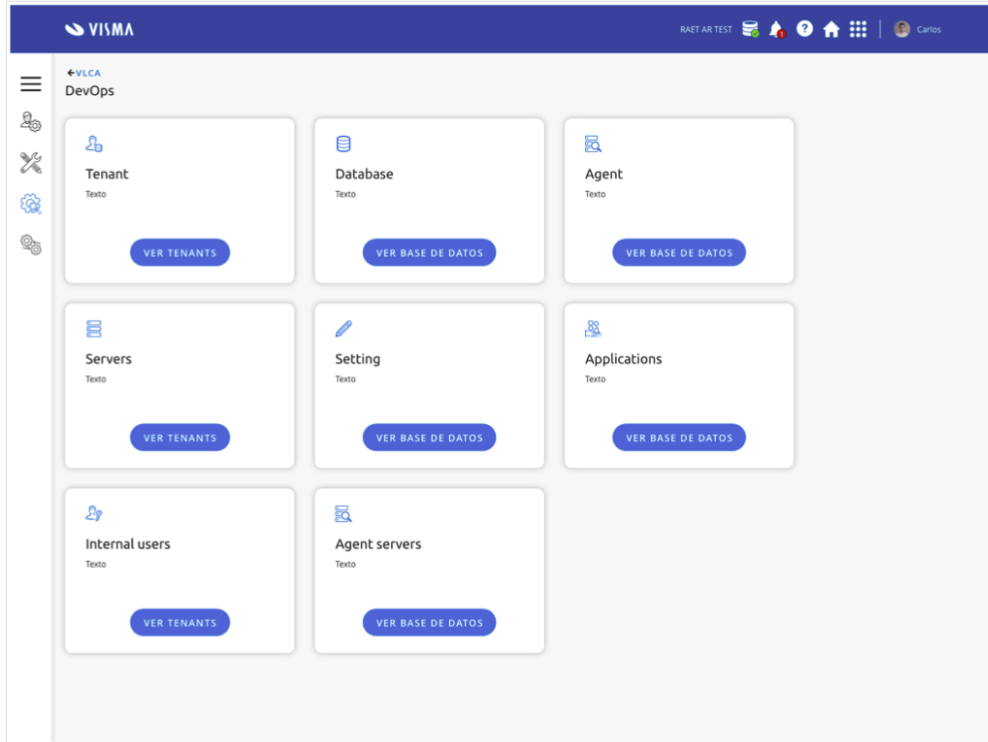
Nota. Fuente: Tomada de la compañía (2022).

3.2.5.9 Pruebas Manuales y Automatizadas

En la Figura 37 que se presenta a continuación se muestra el listado de opciones del menú de DevOps del SARH para la realización de las pruebas manuales.

Figura 37

Prueba manual de listado de opciones de menú de FrontEnd



Nota. Fuente: Tomada de la compañía (2022).

En la Figura 38 que se presenta a continuación se muestra el listado de Tenants del menú de Tenants en DevOps del SARH para la realización de las pruebas manuales.

Figura 38

Prueba manual de listado de Tenants de MicroFrontEnd

Id	Catalog	Description	Client	Type	Engine	Status
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●
34344	BO_DEV	DEV Bolivia	Vioma Latam EX RAET CON ...	Test	vlatamsqldev.database.windows.net	●

Nota. Fuente: Tomada de la compañía (2022).

En la Figura 39 que se presenta a continuación se muestra la creación de Pipeline para Cypress y la lista de tareas que ejecutara el Pipeline.

Figura 39

Lista de tareas de pipeline de Cypress para pruebas automatizadas

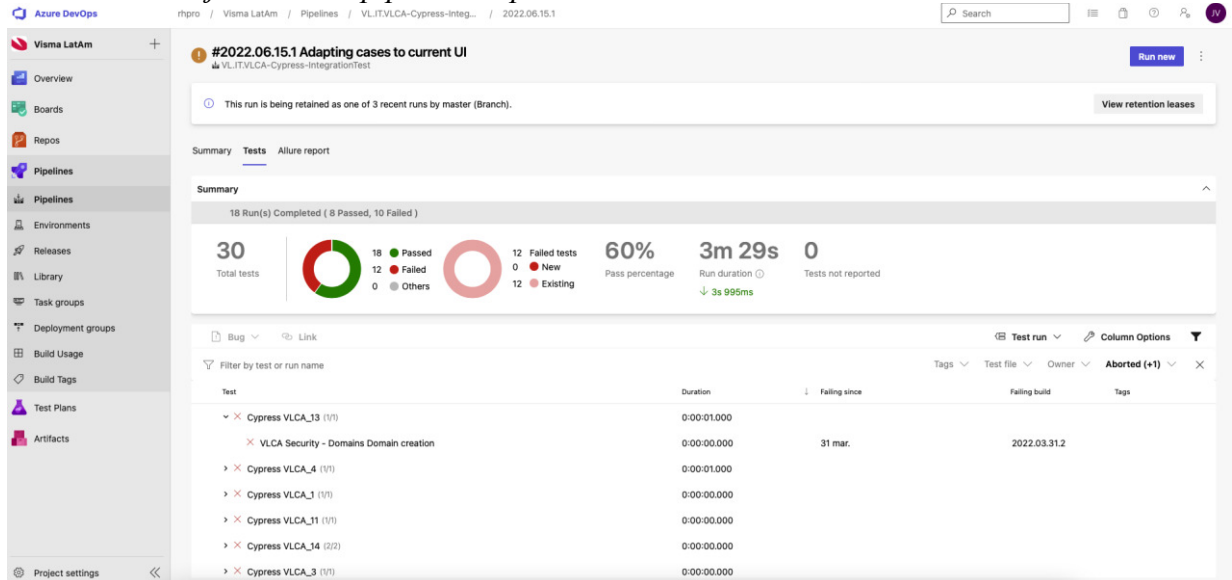
Task Name	Description	Status
Get sources	Build pipeline	Active
Cypress VLCA	Run on agent	Active
Command Line Script	Command line	Active
Publish Test Results *xml	Publish Test Results	Active
Copy Files to: C:\Users\QA01\Desktop\ProcessPostman...	Disabled: Copy files	Disabled
Publish Test Results **test-results.xml	Disabled: Publish Test Results	Disabled
Copy Files to: \$(exportReportFolder)	Disabled: Copy files	Disabled
Publish HTML	Disabled: Publish HTML	Disabled

Nota. Fuente: Tomada de la compañía (2022).

En la Figura 40 que se presenta a continuación, se muestra los resultados de la ejecución de Pipeline de Pruebas Automatizadas.

Figura 40

Resultados de ejecución de pipeline de pruebas automatizadas



Nota. Fuente: Tomada de la compañía (2022).

3.3 Evaluación

3.3.1 Evaluación Económica / Evaluación Costo - Beneficio

La Tabla 18, se muestra la evaluación económica que contempla a los profesionales involucrados de la propia compañía.

Tabla 18

Costos estimados de profesionales en el proyecto de migración

Descripción	Costo Individual	Cantidad	Monto/ mes
Desarrolladores	\$ 3,000.00	8	\$ 24,000.00
Arquitecto de Software	\$ 4,000.00	1	\$ 4,000.00
Líder Técnico	\$ 4,000.00	1	\$ 4,000.00
Product Owner	\$ 3,500.00	1	\$ 3,500.00
Scrum Master	\$ 3,000.00	1	\$ 3,000.00
Líder de Calidad	\$ 3,500.00	1	\$ 3,500.00
Analista de Calidad	\$ 3,000.00	2	\$ 6,000.00
Diseñadores UX/ UI	\$ 2,500.00	2	\$ 5,000.00
Analista de Operaciones	\$ 2,500.00	2	\$ 5,000.00
		Total	\$ 58,000.00

Nota. Fuente: Elaborado por el autor (2022).

La Tabla 19, se muestra la evaluación económica de los costos operativos que asume la compañía para el presente desarrollo del SARH.

Tabla 19

Costos estimados operativos del proyecto de migración

Tipo	Costo Individual	Cantidad	Monto/ mes
Licencias Microsoft Azure DevOps	\$ 500.00	9	\$ 4,500.00
		Total	\$ 4,500.00

Nota. Fuente: Elaborado por el autor (2022).

De acuerdo con la información mostrada en el proyecto se tuvo una inversión mensual aproximada de \$ 62,500.00 en costos de personal y costos operativos. Y realizando el cálculo de los 10 meses que duró la migración y donde se lanzó el primero release a producción, sería una inversión total de \$ 625,000.00 en total que costó el proyecto de migración del SARH en esta compañía.

3.3.2 Interpretación del VAN y del TIR

El proyecto tiene una inversión total de \$ 625,000.00 y se consideró que generará flujos económicos por año durante 5 años considerando una tasa de descuento del 10% anual.

En la siguiente Figura 41, se detallan los cálculos para realizar la evaluación de la viabilidad económica del proyecto con el VAN y el TIR.

Figura 41

Cálculo de VAN y TIR del proyecto

Años	0	1	2	3	4	5
Inversión inicial	\$ 625,000.00					
Flujo caja economico	-\$ 625,000.00	\$ 130,000.00	\$ 150,000.00	\$ 183,000.00	\$ 223,000.00	\$ 261,000.00
COK	10%					
VAN	\$ 69,011.83					
TIR	14%					

Nota. Fuente: Elaboracion propia.

Dado que el VAN > 0 a una tasa de descuento de 10 % y dado que el TIR > k, donde K=10%. Por lo tanto, el proyecto es considerado como un proyecto rentable.

CAPÍTULO IV – REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

El autor del actual trabajo de suficiencia participo en el proyecto de migración del SARH y perteneció al equipo de FrontEnd desarrollándose activamente en todas las fases del proyecto.

Como parte de las actividades del autor han sido la de realizar la codificación y maquetación de componentes visuales, también la verificación y seguimiento de los Pipelines que involucran su desarrollo, así como la revisión del aplicativo en los entornos de desarrollo y testing. También participo de las reuniones de entrega de diseño por parte del equipo de UX/ UI y de las ceremonias de Scrum: Scrum Daily, Sprint Planning, Sprint Review y Retrospective.

A lo largo de toda esta implementación el autor aprendió a llevar a cabo lineamientos de procesos y mejoras para el desarrollo de software, así como la integración y mejora de comunicación con otros desarrolladores y no solo de FrontEnd. Así mismo, aumento los conocimientos de su experiencia y permitió en corto tiempo poder entregar mejoras de valor al sistema, además de poder apoyar a los compañeros de otros equipos.

Como parte final de toda esta experiencia se resalta el trabajo en equipo que mejoro por llevar consigo la metodología de Scrum y cumplir con las características y líneas que este mismo exige, frente a inconvenientes se pudo llevar adelante el propósito de cumplir con el lanzamiento del SARH, con todas las funcionalidades que se comprometido desde un inicio y generando el alto valor para el usuario y sin duda otro punto importante como la automatización de software incremento la agilidad de la entrega al usuario final con ciclos cortos y visibles hacia los entornos de desarrollo y testing principalmente.

CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Se estableció un modelo de arquitectura para la migración a la nube de Azure, esto permitió aclarar y definir las actividades primordiales de todos los equipos. Esto fue determinante para establecer un correcto cronograma de entregas durante cada Sprint, así además aclarar la infraestructura, entornos, aplicaciones y herramientas adecuadas para la compañía.

Al establecer un correcto planeamiento de actividades acorde al plan de migración inicial de la compañía, se ingresaron en Azure DevOps a las Épicas, Historias de Usuario, los Sprints del proyecto y toda la información de apoyo para iniciar el trabajo diario, con lo cual se hizo de mayor visibilidad de los avances y permitió también responder rápidamente frente a los retrasos o demoras

Se realizó todo el trabajo de desarrollo con Azure DevOps, esto ayudó a cumplir con los tiempos establecidos para las entregas en cada Release en cada Sprint. Las ejecuciones de los Jobs del Pipeline tuvieron demoras con respecto a inconvenientes externos a la compañía y de responsabilidad exclusivamente del proveedor de servicios respecto a caídas regionales de los servidores, pero se tuvo rápida respuesta de parte de Microsoft y un adecuado conducto de procesos para mitigar los retrasos e inconvenientes causados.

Se realizaron las pruebas automatizadas y manuales con validaciones necesarias previas al lanzamiento de cada Release del proyecto de migración. Se realizó el seguimiento de la plataforma con la implementación de herramientas como Cypress que permite realizar pruebas de tipo E2E del sistema esto permitió poder corregir a tiempo los inconvenientes del sistema.

5.2 Recomendaciones

Con la experiencia obtenida en el proyecto sobre migración del SARH se brinda como primera recomendación contar con tiempo más amplios para los desarrollos que permitan cumplir con los acuerdos de la planificación, y se toma en cuenta que al ser una primera migración a la nube de Azure conlleva a tener más experiencia en todos los integrantes del equipo.

Se recomienda brindar las capacitaciones respectivas sobre el uso de Azure DevOps y de la metodología Scrum de manera interna, para incrementar los conocimientos de todos los equipos y para generar la cultura del uso de las herramientas que nos proporciona la plataforma de Microsoft y así aprovechar todo el potencial que nos brinda como opción a este servicio.

Además, durante las ejecuciones de pruebas se recomienda incrementar el porcentaje de las pruebas unitarias en todos los desarrollos de BackEnd y FrontEnd, para entregar software con mejor calidad, así reducir las deudas técnicas y también para servir de apoyo al equipo de calidad en la entrega de un producto con menores errores. Estas actividades de pruebas unitarias se pueden colocar en un Pipeline para que se realicen de forma automática.

5.3 Fuentes de Información

MICROSOFT. (07 de 03 de 2022). *Que es Azure DevOps?* Obtenido de Azure DevOps: <https://docs.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops>

MICROSOFT. (01 de 04 de 2022). *Que es Azure Pipelines?* Obtenido de Azure DevOps: <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>

MICROSOFT. (27 de 06 de 2022). *Que es Azure Repos?* Obtenido de Azure DevOps: <https://docs.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos?view=azure-devops>

MICROSOFT. (09 de 05 de 2022). *Resumen Azure Artifacts.* Obtenido de Azure DevOps: <https://docs.microsoft.com/en-us/azure/devops/artifacts/start-using-azure-artifacts?view=azure-devops>

SCRUM Study. (2022). *Scrum en 6 minutos.* Obtenido de Scrum Study: <https://www.scrumstudy.com/whyscrum/scrum-in-six-minutes>

Red Hat. (2022). *Que es SaaS?* Obtenido de Entendiendo Cloud Computing: <https://www.redhat.com/en/topics/cloud-computing/what-is-saas>

Red Hat. (2022). *Que es PaaS?* Obtenido de Entendiendo Cloud Computing: <https://www.redhat.com/en/topics/cloud-computing/what-is-paas>

Red Hat. (2022). *Que es IaaS?* Obtenido de Entendiendo Cloud Computing: <https://www.redhat.com/en/topics/cloud-computing/what-is-iaas>

AMAZON. (2022). *Que es DevOps?* Obtenido de Amazon Web Services: <https://aws.amazon.com/devops/what-is-devops/>

IBM. (02 de 10 de 2019). *DevOps.* Obtenido de IBM Cloud Learn Hub: <https://www.ibm.com/cloud/learn/devops-a-complete-guide>

Silvola, R.-P., & Sargsyan, L. (2021). DevOps and CI/CD for WinCC Open Architecture Applications and Frameworks. *CERN Document Server*, 5.

ATLASSIAN. (2020). *Atlassian Survey 2020 - DevOps Trends.* Obtenido de Atlassian Whitepapers: <https://www.atlassian.com/es/whitepapers/devops-survey-2020>

MICROSOFT. (2022). *Que es Azure Test Plans?* Obtenido de Azure DevOps: <https://docs.microsoft.com/es-es/azure/devops/test/overview?view=azure-devops>

- MICROSOFT. (2022). *Informacion general sobre Azure Artifacts*. Obtenido de Azure DevOps: <https://docs.microsoft.com/es-es/azure/devops/artifacts/start-using-azure-artifacts?view=azure-devops>
- MICROSOFT. (07 de 2022). *¿Qué es Azure Boards?* Obtenido de Azure DevOps: <https://docs.microsoft.com/es-es/azure/devops/boards/get-started/what-is-azure-boards?view=azure-devops>
- MICROSOFT. (07 de 2022). *Arquitectura de cliente y servidor*. Obtenido de Azure: <https://docs.microsoft.com/es-es/azure/communication-services/concepts/client-and-server-architecture>
- Hall, T. (2022). *Prácticas recomendadas de DevOps*. Obtenido de ATlassian: <https://www.atlassian.com/es/devops/what-is-devops/devops-best-practices>
- Drumond, C. (2022). *¿Qué es scrum?* Obtenido de Atlassian: <https://www.atlassian.com/es/agile/scrum>
- Zettler, K. (2022). *What is cloud computing? An overview of the cloud*. Obtenido de Atlassian: <https://www.atlassian.com/es/microservices/cloud-computing>
- Atlassian. (2022). *¿Qué es DevOps?* Obtenido de Atlassian: <https://www.atlassian.com/es/devops>
- MICROSOFT. (19 de 8 de 2022). *¿Qué es DevOps?* Obtenido de MICROSOFT: <https://docs.microsoft.com/en-us/devops/what-is-devops>

5.4 Glosario

- **SARH:** Sistema de Administración de Recursos Humanos.
- **FrontEnd:** Parte visual de una aplicación, definido también como todo componente con el que el usuario puede interactuar. En la programación web también está definido como la Tecnología que se ejecuta en el navegador.
- **BackEnd:** Parte no visual de una aplicación, generalmente son las acciones que se desembocan cuando un usuario interactúa con la parte visual de una aplicación o sistema.
- **SPA (Single Page):** Aplicación que consta de una sola página.
- **Single SPA:** Es una librería que permite aplicar SPA a través de JavaScript y es actualmente de gran aceptación por desarrolladores y empresas.
- **RoadMap:** Es un plan de programación u hoja de ruta del desarrollo de un software.
- **CI:** Del inglés de Integración Continua.
- **CD:** Del inglés de la palabra Entrega Continua.
- **WorkFlow:** Flujo de trabajo, y es el conjunto de actividades relacionadas.

ANEXOS

Anexo 1. Políticas de Branch

Branch Policies
Note: If any required policy is enabled, this branch cannot be deleted and changes must be made via pull request.

On **Require a minimum number of reviewers**
Require approval from a specified number of reviewers on pull requests.

Minimum number of reviewers

Allow requestors to approve their own changes
 Prohibit the most recent pusher from approving their own changes
 Allow completion even if some reviewers vote to wait or reject
 When new changes are pushed:

Off **Check for linked work items**
Encourage traceability by checking for linked work items on pull requests.

On **Check for comment resolution**
Check to see that all comments have been resolved on pull requests.

Required
Block pull requests from being completed while any comments are active.

Optional
Warn if any comments are active, but allow pull requests to be completed.

Off **Limit merge types**
Control branch history by limiting the available types of merge when pull requests are completed.

Anexo 2. Políticas de Branch de Build Validation

Project Settings
Visma LatAm

General
Overview
Teams
Permissions
Notifications
Service hooks
Dashboards

Boards
Project configuration
Team configuration
GitHub connections

Pipelines
Agent pools
Parallel jobs
Settings
Test management
Release retention
Service connections
XAML build services

Repos
Repositories

← VLCloudAdministratorFrontDev...

Filter by keywords

- feature
 - SP230-PBI81869-FeatureTenants
 - Task-82615-FixNavigationTenants
 - Task-82723-FiltersTenantComponent
 - Task-85108-EditComponentSetting
 - Task-91386-POCUnitTest
- 88493_DevOps_fix
- CypressTags-DevOps
- master | Default | Compare
- MejorasIntegrales-Task-84605
- MejorasVLCA-85134-DevOps
- Task-84251-BreadCrumbsDevOps
- Task-86604-DevOps-Improvements
- VLCA-AgentServer-MejorasVisuales
- VLCA-Internal-User

Build Validation
Validate code by pre-merging and building pull request changes.

Enabled	Name ↑	Path filter	Trigger	Inheritance
<input checked="" type="checkbox"/>	VLCloudAdministratorFrontDev...		Automatic Expires when code changes	

Status Checks
Require other services to post successful status to complete pull requests.
No status checks found, but you can use the add button to create one!

Automatically included reviewers
Designate code reviewers to automatically include when pull requests change certain areas of code.
No automatic reviewer policies found, but you can use the add button to create one!

Anexo 3. Release de FrontEnd de Desarrollo

```
{
  "source":2,
  "revision":30,
  "description": null,
  "createdBy": {
    "displayName": "Jose Villasante",
    "id":"46b1ccd5-7cac-66b4-b3e9-3287db156b59",
    "uniqueName": "jose.villasantem@gmail.com",
  }
  "descriptor":"aad.YWNmM2E3YmEtODJhMy03YzViLWJjMDYtOWExOWUzZTk2MDI1"
},
  "createdOn":"2021-06-22T13:49:00.110Z",
  "isDeleted":false,
  "lastRelease":{
    "id":7451,
    "name":"2022.06.15.1",
    "artifacts":[
      ],
    " links":{
      },
    },
  "description":"Triggered by VLCloudAdministratorFrontDevOps 104768.",
  "releaseDefinition":{
    "id":104,
    "projectReference":null,
    " _links":{
      }
    },
  "createdOn":"2022-06-15T16:11:31.273Z",
  "createdBy":{
    "displayName":"Jose Villasante",
    "url":"https://spsprodweu2.vssps.visualstudio.com/Ab94c77fa-95f6-429a-8405-
b686fb54d2b1/_apis/Identities/21efa127-f1eb-6f1a-a2c8-ac1429ddf891",
    " links":{
      "avatar":{
        "href":"https://dev.azure.com/rhpro/_apis/GraphProfile/MemberAvatars/aad.YTMxNTYyZjEtY
zE5NS03MDNlTlJmMmN2U3Y2UzMzYxZmUz"
      }
    },
    "id":"21efa127-f1eb-6f1a-a2c8-ac1429ddf891",
    "uniqueName":"jose.villasantem@gmail.com",
  }
  "imageUrl":"https://dev.azure.com/rhpro/_apis/GraphProfile/MemberAvatars/aad.YTMxNTYy
ZjEtYzE5NS03MDNlTlJmMmN2U3Y2UzMzYxZmUz",
  "descriptor":"aad.YTMxNTYyZjEtYzE5NS03MDNlTlJmMmN2U3Y2UzMzYxZmUz"
}
},
"variables":{
```

```

"version": {
  "value": "$(Build.BuildId)"
},
"variableGroups": [
],
"environments": [
  {
    "id": 254,
    "name": "Develop",
    "rank": 1,
    "owner": {
      "displayName": "Jose Villasante",
      "url": "https://spsprodweu2.vssps.visualstudio.com/Ab94c77fa-95f6-429a-8405-
b686fb54d2b1/ apis/Identities/46b1ccd5-7cac-66b4-b3e9-3287db156b59",
      "_links": {
        "avatar": {
        }
      },
      "id": "46b1ccd5-7cac-66b4-b3e9-3287db156b59",
      "uniqueName": "jose.villasantem@gmail.com",
    },
    "descriptor": "aad.YWNmM2E3YmEtODJhMy03YzViLWJjMDYtOWExOWUzZTk2MDI1"
  },
  "variables": {
    "apiEndpointAdmin": {
      "value": "https://dev.webapiadmin.vismalatam.com"
    },
    "apiManagement": {
      "value": "https://vlatam-dev-apimanagement.azure-api.net"
    },
    "instrumentationKey": {
      "value": "bb137364-ea15-3466-a509-6b8df714cfed"
    },
    "ocpApiSubscriptionKey": {
      "value": "5fcef54926fc46d4b758e000cd17b5b4"
    }
  },
  "variableGroups": [
],
"preDeployApprovals": {
  "approvals": [
    {
      "rank": 1,
      "isAutomated": true,
      "isNotificationOn": false,
      "id": 819
    }
  ],
  "approvalOptions": {
    "requiredApproverCount": null,

```

```

    "releaseCreatorCanBeApprover":false,|
    "autoTriggeredAndPreviousEnvironmentApprovedCanBeSkipped":false,
    "enforceIdentityRevalidation":false,
    "timeoutInMinutes":0,
    "executionOrder":1
  }
},
"deployStep":{
  "id":822
},
"postDeployApprovals":{
  "approvals":[
    {
      "rank":1,
      "isAutomated":true,
      "isNotificationOn":false,
      "id":823
    }
  ],
  "approvalOptions":{
    "requiredApproverCount":null,
    "releaseCreatorCanBeApprover":false,
    "autoTriggeredAndPreviousEnvironmentApprovedCanBeSkipped":false,
    "enforceIdentityRevalidation":false,
    "timeoutInMinutes":0,
    "executionOrder":2
  }
},
"deployPhases":[
  {
    "deploymentInput":{
      "parallelExecution":{
        "parallelExecutionType":0
      },
      "agentSpecification":null,
      "skipArtifactsDownload":false,
      "artifactsDownloadInput":{
        "downloadInputs":[

          ]
      },
      "queueId":128,
      "demands":[

      ],
      "enableAccessToken":false,
      "timeoutInMinutes":0,
      "jobCancelTimeoutInMinutes":1,
      "condition":"succeeded()",
      "overrideInputs":{

      }
    }
  ],
},

```

```

"rank":1,|
  "phaseType":1,
  "name":"Run on agent",
  "refName":null,
  "workflowTasks":[
    {
      "environment":{
      },
      "taskId":"8ce97e91-56cc-4743-bfab-9a9315be5f27",
      "version":"1.*",
      "name":"Replace config.json with variables",
      "refName":"",
      "enabled":true,
      "alwaysRun":false,
      "continueOnError":false,
      "timeoutInMinutes":0,
      "retryCountOnTaskFailure":0,
      "definitionType":"task",
      "overrideInputs":{
      },
      "condition":"succeeded()",
      "inputs":{
"folderPath":"$(System.DefaultWorkingDirectory)/_VLCLOUDAdministratorFrontDevOps/web-
app/assets",
      "enableXmlTransform":"false",
      "xmlTransformationRules":"-transform **\\*.Release.config -xml **\\*.config",
      "fileType":"json",
      "targetFiles":"config.json"
      }
    },
    {
      "environment":{
      },
      "taskId":"e213ff0f-5d5c-4791-802d-52ea3e7be1f1",
      "version":"2.*",
      "name":"PowerShell Script",
      "refName":"",
      "enabled":true,
      "alwaysRun":false,
      "continueOnError":false,
      "timeoutInMinutes":0,
      "retryCountOnTaskFailure":0,
      "definitionType":"task",
      "overrideInputs":{
      },
      "condition":"succeeded()",
      "inputs":{
        "targetType":"inline",

```

```

    "filePath": "",
    "arguments": "",
    "script": "\n((Get-Content -path
$(System.DefaultWorkingDirectory)/_VLCLOUDAdministratorFrontDevOps/web-app/main.js -
Raw) -replace 'http://localhost:4207','https://dev-vlca-front-devops.azurewebsites.net') | Set-
Content -Path $(System.DefaultWorkingDirectory)/_VLCLOUDAdministratorFrontDevOps/web-
app/main.js",
    "errorActionPreference": "stop",
    "warningPreference": "default",
    "informationPreference": "default",
    "verbosePreference": "default",
    "debugPreference": "default",
    "failOnStderr": "false",
    "showWarnings": "false",
    "ignoreLASTEXITCODE": "false",
    "pwsh": "false",
    "workingDirectory": "",
    "runScriptInSeparateScope": "false"
  }
},
{
  "environment": {
  },
  "taskId": "497d490f-eea7-4f2b-ab94-48d9c1acdcbl",
  "version": "4.*",
  "name": "Azure App Service Deploy: dev-vlca-front-devops",
  "refName": "",
  "enabled": true,
  "alwaysRun": false,
  "continueOnError": false,
  "timeoutInMinutes": 0,
  "retryCountOnTaskFailure": 0,
  "definitionType": "task",
  "overrideInputs": {
  },
  "condition": "succeeded()",
  "inputs": {
    "ConnectionType": "AzureRM",
    "ConnectedServiceName": "b609bacd-2984-49d2-8746-9709e7b09cb4",
    "PublishProfilePath": "$(System.DefaultWorkingDirectory)/**/*.pubxml",
    "PublishProfilePassword": "",
    "WebAppKind": "webApp",
    "WebAppName": "dev-vlca-front-devops",
    "DeployToSlotOrASEFlag": "false",
    "ResourceGroupName": "",
    "SlotName": "production",
    "DockerNamespace": "",
    "DockerRepository": "",
    "DockerImageTag": "",
    "VirtualApplication": ""
  }
}
}
}

```

```

{"Package":"$(System.DefaultWorkingDirectory)/_VLCloudAdministratorFrontDevOps/web-
app",
  "RuntimeStack":"","
  "RuntimeStackFunction":"","
  "StartupCommand":"","
  "ScriptType":"","
  "InlineScript":""," You can provide your deployment commands here. One
command per line.",
  "ScriptPath":"","
  "WebConfigParameters":"","
  "AppSettings":"","
  "ConfigurationSettings":"","
  "UseWebDeploy":"false",
  "DeploymentType":"webDeploy",
  "TakeAppOfflineFlag":"true",
  "SetParametersFile":"","
  "RemoveAdditionalFilesFlag":"false",
  "ExcludeFilesFromAppDataFlag":"true",
  "AdditionalArguments":"-retryAttempts:6 -retryInterval:10000",
  "RenameFilesFlag":"true",
  "XmlTransformation":"false",
  "XmlVariableSubstitution":"false",
  "JSONFiles":""
}
]
},
"environmentOptions":{
  "emailNotificationType":"OnlyOnFailure",
  "emailRecipients":"release.environment.owner;release.creator",
  "skipArtifactsDownload":false,
  "timeoutInMinutes":0,
  "enableAccessToken":false,
  "publishDeploymentStatus":true,
  "badgeEnabled":false,
  "autoLinkWorkItems":false,
  "pullRequestDeploymentEnabled":false
},
"demands":[
],
"conditions":[
  {
    "name":"ReleaseStarted",
    "conditionType":1,
    "value":""
  }
],
"executionPolicy":{
  "concurrencyCount":1,
  "queueDepthCount":0
},
}

```



```

    },
    "badgeUrl": "https://vsrm.dev.azure.com/rhpro/_apis/public/Release/badge/da33e6f7-3570-42e4-ab00-6c0f84bc48b6/104/254"
  }
},
"artifacts": [
  {
    "sourceId": "da33e6f7-3570-42e4-ab00-6c0f84bc48b6:403",
    "type": "Build",
    "alias": "_VLCLOUDADMINISTRATORFRONTDEVOPS",
    "definitionReference": {
      "artifactSourceDefinitionUrl": {
        "id": "https://dev.azure.com/rhpro/_permalink/_build/index?collectionId=74f7e0a0-8352-460f-9ff9-18767341206a&projectId=da33e6f7-3570-42e4-ab00-6c0f84bc48b6&definitionId=403",
        "name": ""
      },
      "defaultVersionBranch": {
        "id": "",
        "name": ""
      },
      "defaultVersionSpecific": {
        "id": "",
        "name": ""
      },
      "defaultVersionTags": {
        "id": "",
        "name": ""
      },
      "defaultVersionType": {
        "id": "latestType",
        "name": "Latest"
      },
      "definition": {
        "id": "403",
        "name": "VLCLOUDADMINISTRATORFRONTDEVOPS"
      },
      "definitions": {
        "id": "",
        "name": ""
      },
      "IsMultiDefinitionType": {
        "id": "False",
        "name": "False"
      },
      "project": {
        "id": "da33e6f7-3570-42e4-ab00-6c0f84bc48b6",
        "name": "Visma LatAm"
      },
      "repository": {
        "id": "",
        "name": ""
      }
    }
  }
]

```

```

    },
    "isPrimary":true,
    "isRetained":false
  }
},
"triggers":[
  {
    "artifactAlias":"_VLCloudAdministratorFrontDevOps",
    "triggerConditions":[

    ],
    "triggerType":1
  }
],
"releaseNameFormat":"$(date:yyyy).$(date:MM).$(date:dd)$(rev:r)",
"tags":[

],
"properties":{
  "DefinitionCreationSource":{
    "Type":"System.String",
    "Svalue":"ReleaseClone"
  },
  "IntegrateBoardsWorkItems":{
    "Type":"System.String",
    "Svalue":"False"
  },
  "IntegrateJiraWorkItems":{
    "Type":"System.String",
    "Svalue":"false"
  }
},
"id":104,
"name":"VLCloudAdministratorFrontDevOps-ITC",
"path":"\\VLCloudAdministratorFrontDevOps",
"projectReference":null,
"url":"https://vsrm.dev.azure.com/rhpro/da33e6f7-3570-42e4-ab00-6c0f84bc48b6/_apis/Release/definitions/104",
  "_links":{
    "self":{
      "href":"https://vsrm.dev.azure.com/rhpro/da33e6f7-3570-42e4-ab00-6c0f84bc48b6/_apis/Release/definitions/104"
    },
    "web":{
      "href":"https://dev.azure.com/rhpro/da33e6f7-3570-42e4-ab00-6c0f84bc48b6/_release?definitionId=104"
    }
  }
}
}

```

Anexo 4. Pipeline FrontEnd con Integración Continua

```
{
  "options":[
    {
      "enabled":false,
      "definition":{
        "id":"5d58cc01-7c75-450c-be18-a388ddb129ec"
      },
      "inputs":{
        "branchFilters":["\\+refs/heads/*\\"],
        "additionalFields":{}"
      }
    },
    {
      "enabled":false,
      "definition":{
        "id":"a9db38f9-9fdc-478c-b0f9-464221e58316"
      },
      "inputs":{
        "workItemType":"Task",
        "assignToRequestor":"true",
        "additionalFields":{}"
      }
    }
  ],
  "triggers":[
    {
      "branchFilters":[
        "+refs/heads/master"
      ],
      "pathFilters":[
      ],
      "batchChanges":false,
      "maxConcurrentBuildsPerBranch":1,
      "pollingInterval":0,
      "triggerType":2
    }
  ],
  "variables":{
    "system.debug":{
      "value":"false",
      "allowOverride":true
    }
  },
  "properties":{
  },
  "tags":[
  ],
  "links":{
```

```

    "self":{
      "href":"https://dev.azure.com/front-
end/0123456789/_apis/build/Definitions/403?revision=6"
    },
    "web":{
      "href":"https://dev.azure.com/front-end/0123456789/_build/definition?definitionId=403"
    },
    "editor":{
      "href":"https://dev.azure.com/front-end/0123456789/_build/designer?id=403&_a=edit-
build-definition"
    },
    "badge":{
      "href":"https://dev.azure.com/front-end/0123456789/_apis/build/status/403"
    }
  },
  "jobAuthorizationScope":1,
  "jobTimeoutInMinutes":60,
  "jobCancelTimeoutInMinutes":5,
  "process":{
    "phases":[
      {
        "steps":[
          {
            "environment":{

            },
            "enabled":true,
            "continueOnError":false,
            "alwaysRun":false,
            "displayName":"Use Node 14.x",
            "timeoutInMinutes":0,
            "retryCountOnTaskFailure":0,
            "condition":"succeeded()",
            "task":{
              "id":"31c75bbb-bcdf-4706-8d7c-4da6a1959bc2",
              "versionSpec":"0.*",
              "definitionType":"task"
            },
            "inputs":{
              "versionSpec":"14.x",
              "checkLatest":"false",
              "force32bit":"false"
            }
          },
          {
            "environment":{

            },
            "enabled":true,
            "continueOnError":false,
            "alwaysRun":false,
            "displayName":"Angular CLI",
            "timeoutInMinutes":0,]
        }
      }
    ]
  }
}

```

```

"retryCountOnTaskFailure":0,
"condition":"succeeded()",
"task":{
  "id":"fe47e961-9fa8-4106-8639-368c022d43ad",
  "versionSpec":"1.*",
  "definitionType":"task"
},
"inputs":{
  "command":"custom",
  "workingDir":"",
  "verbose":"false",
  "customCommand":"install -g @angular/cli",
  "customRegistry":"useNpmrc",
  "customFeed":"",
  "customEndpoint":"",
  "publishRegistry":"useExternalRegistry",
  "publishFeed":"",
  "publishPackageMetadata":"true",
  "publishEndpoint":""
}
},
{
  "environment":{
  },
  "enabled":true,
  "continueOnError":false,
  "alwaysRun":false,
  "displayName":"npm install copy",
  "timeoutInMinutes":0,
  "retryCountOnTaskFailure":0,
  "condition":"succeeded()",
  "task":{
    "id":"fe47e961-9fa8-4106-8639-368c022d43ad",
    "versionSpec":"1.*",
    "definitionType":"task"
  },
  "inputs":{
    "command":"install",
    "workingDir":"",
    "verbose":"false",
    "customCommand":"",
    "customRegistry":"useNpmrc",
    "customFeed":"",
    "customEndpoint":"",
    "publishRegistry":"useExternalRegistry",
    "publishFeed":"",
    "publishPackageMetadata":"true",
    "publishEndpoint":""
  }
}
},
{
  "environment":{

```

```

    },
    "enabled":true,
    "continueOnError":false,
    "alwaysRun":false,
    "displayName":"build single-spa",
    "timeoutInMinutes":0,
    "retryCountOnTaskFailure":0,
    "condition":"succeeded()",
    "task":{
      "id":"fe47e961-9fa8-4106-8639-368c022d43ad",
      "versionSpec":"1.*",
      "definitionType":"task"
    },
    "inputs":{
      "command":"custom",
      "workingDir":"",
      "verbose":"false",
      "customCommand":"run build:single-spa:VLCA-DevOps",
      "customRegistry":"useNpmrc",
      "customFeed":"",
      "customEndpoint":"",
      "publishRegistry":"useExternalRegistry",
      "publishFeed":"",
      "publishPackageMetadata":"true",
      "publishEndpoint":""
    }
  },
  {
    "environment":{
    },
    "enabled":true,
    "continueOnError":false,
    "alwaysRun":false,
    "displayName":"Copy web.config to dist folder",
    "timeoutInMinutes":0,
    "retryCountOnTaskFailure":0,
    "condition":"succeeded()",
    "task":{
      "id":"5bfb729a-a7c8-4a78-a7c3-8d717bb7c13c",
      "versionSpec":"2.*",
      "definitionType":"task"
    },
    "inputs":{
      "SourceFolder":"src",
      "Contents":"web.config",
      "TargetFolder":"dist",
      "CleanTargetFolder":"false",
      "OverWrite":"false",
      "flattenFolders":"false",
      "preserveTimestamp":"false",
      "retryCount":"0",

```

```

    "delayBetweenRetries": "1000",
    "ignoreMakeDirErrors": "false"
  }
},
{
  "environment": {
  },
  "enabled": true,
  "continueOnError": false,
  "alwaysRun": false,
  "displayName": "Publish Artifact: web-app",
  "timeoutInMinutes": 0,
  "retryCountOnTaskFailure": 0,
  "condition": "succeeded()",
  "task": {
    "id": "2ff763a7-ce83-4e1f-bc89-0ae63477cebe",
    "versionSpec": "1.*",
    "definitionType": "task"
  },
  "inputs": {
    "PathtoPublish": "dist",
    "ArtifactName": "web-app",
    "ArtifactType": "Container",
    "TargetPath": "",
    "Parallel": "false",
    "ParallelCount": "8",
    "FileCopyOptions": "",
    "StoreAsTar": "false"
  }
}
],
"name": "Agent Build",
"refName": "Job_1",
"condition": "succeeded()",
"target": {
  "queue": {
    "_links": {
      "self": {
        "href": "https://dev.azure.com/front-end/_apis/build/Queues/61"
      }
    }
  },
  "id": 61,
  "url": "https://dev.azure.com/front-end/_apis/build/Queues/61",
  "pool": null
},
"agentSpecification": {
  "identifier": "vs2017-win2016"
},
"executionOptions": {
  "type": 0
},
"allowScriptsAuthAccessOption": false,

```

```

        "type":1
      },
      "jobAuthorizationScope":1
    }
  ],
  "type":1
},
"repository":{
  "properties":{
    "cleanOptions":"0",
    "labelSources":"0",
    "labelSourcesFormat":"$(build.buildNumber)",
    "reportBuildStatus":"true",
    "gitLfsSupport":"false",
    "skipSyncSource":"false",
    "checkoutNestedSubmodules":"false",
    "fetchDepth":"0"
  },
  "id":"a310bf2d-da70-438a-b778-cfac5fecbf0d",
  "type":"TfsGit",
  "name":"VLCLOUDAdministratorFrontDevOps",
  "url":"https://dev.azure.com/front-end/frontend/_git/VLCLOUDAdministratorFrontDevOps",
  "defaultBranch":"refs/heads/master",
  "clean":"false",
  "checkoutSubmodules":false
},
"processParameters":{
},
"quality":1,
"authoredBy":{
  "displayName":"Jose Villasante",
  "url":"https://spsprodweu2.vssps.visualstudio.com/Ab94c77fa-95f6-429a-8405-
b686fb54d2b1/_apis/Identities/21efa127-f1eb-6f1a-a2c8-ac1429ddf891",
  "_links":{
    "avatar":{
      "href":"https://dev.azure.com/front-
end/_apis/GraphProfile/MemberAvatars/aad.YTMxNTYyZjEtYzE5NS03MDNILTljYmMtN2U
3Y2UzMzYxZmUz"
    }
  },
  "id":"21efa127-f1eb-6f1a-a2c8-ac1429ddf891",
  "uniqueName":"jose.villasante@gmail.com",
  "imageUrl":"https://dev.azure.com/front-
end/_apis/GraphProfile/MemberAvatars/aad.YTMxNTYyZjEtYzE5NS03MDNILTljYmMtN2U
3Y2UzMzYxZmUz",
  "descriptor":"aad.YTMxNTYyZjEtYzE5NS03MDNILTljYmMtN2U3Y2UzMzYxZmUz"
},
"drafts":{
},
"queue":{
  "_links":{

```



```

    "self":{
      "href":"https://dev.azure.com/front-end/_apis/build/Queues/128"
    }
  },
  "id":128,
  "name":"RepoFrontEnd-AzureDevops",
  "url":"https://dev.azure.com/front-end/_apis/build/Queues/128",
  "pool":{
    "id":28,
    "name":"RepoFrontEnd-AzureDevops"
  }
},
"id":403,
"name":"VLCloudAdministratorFrontDevOps",
"url":"https://dev.azure.com/front-end/0123456789/_apis/build/Definitions/403?revision=6",
"uri":"vstfs:///Build/Definition/403",
"path":"\\VLCloudAdministratorFrontDevOps",
"type":2,
"queueStatus":0,
"revision":6,
"createdDate":"2022-06-30T22:42:32.307Z",
"project":{
  "id":"0123456789",
  "name":"Repo FrontEnd",
  "description":"Repo FrontEnd solution",
  "url":"https://dev.azure.com/frontend/_apis/projects/0123456789",
  "state":1,
  "revision":214,
  "visibility":0,
  "lastUpdateTime":"2022-06-06T12:38:42.340Z"
}
}

```