



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

**Implementación de un sistema de información para la
autenticación y autorización centralizada en las
aplicaciones backoffice de una empresa de seguros**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Carlos Hugo TACZA SOTO

ASESOR

Mg. César Augusto ALCÁNTARA LOAYZA

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

Tacza, C. (2022). *Implementación de un sistema de información para la autenticación y autorización centralizada en las aplicaciones backoffice de una empresa de seguros*. [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	CARLOS HUGO TACZA SOTO
Tipo de documento de identidad	DNI
Número de documento de identidad	72755941
URL de ORCID	https://orcid.org/0000-0003-0653-9552
Datos de asesor	
Nombres y apellidos	Alcántara Loayza César Augusto
Tipo de documento de identidad	DNI
Número de documento de identidad	09132297
URL de ORCID	https://orcid.org/0000-0003-3435-4555
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cortez Vásquez Augusto Parcemón
Tipo de documento	DNI
Número de documento de identidad	08634618
Miembro del jurado 1	
Nombres y apellidos	Osorio Beltrán Norberto Antonio
Tipo de documento	DNI
Número de documento de identidad	08799230
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 20:00 horas del día 17 de agosto del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Mg. Cortez Vásquez Augusto Parcemon (Presidente), Ing. Osorio Beltrán Norberto Antonio (Miembro) y el Mg. Alcántara Loayza César Augusto (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/zhf-yrtc-sfc>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA AUTENTICACIÓN Y AUTORIZACIÓN CENTRALIZADA EN LAS APLICACIONES BACKOFFICE DE UNA EMPRESA DE SEGUROS”**, por el Bachiller **Tacza Soto Carlos Hugo**; 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. Cortez Vásquez Augusto Parcemon, declara al Bachiller **Ingeniero de Sistemas**.

Siendo las 21:00 horas, se levantó la sesión.

Presidente

Mg. Cortez Vásquez Augusto Parcemon

Miembro

Ing. Osorio Beltrán Norberto Antonio

Miembro Asesor

Mg. Alcántara Loayza César Augusto



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA
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: "Implementación de un Sistema de Información para la Autenticación y Autorización centralizada en las aplicaciones Backoffice de una empresa de seguros"
 7. Autor del documento
Bach. Tacza Soto, Carlos Hugo
 8. Fecha de recepción del documento **18/09/2022**
 9. Fecha de aplicación del programa informático de similitudes **18/09/2022**
 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 **5 (cinco)%**
 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 **19/09/2022**



Firmado digitalmente por PRO
CONCEPCION Luzmila Elisa FAU
20148092282 soft
Motivo: Soy el autor del documento
Fecha: 14.11.2022 23:13:07 -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 INGENIERIA DE SISTEMAS E INFORMATICA
Escuela Profesional de Ingeniería de Sistemas

ANEXO 1

Fuentes originales de las similitudes encontradas

1. **hdl.handle.net: 4%**
2. **cybertesis.unmsm.edu.pe: <1%**
3. **www.returngis.net: <1%**
4. **docs.microsoft.com: <1%**
5. **www.smv.gob.pe: <1%**



UNMSM

Firmado digitalmente por PRO
CONCEPCION Luzmila Elisa FAU
20148092282 soft
Motivo: Soy el autor del documento
Fecha: 14.11.2022 23:12:28 -05:00

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

DEDICATORIA

Este trabajo lo dedico a mis padres, por el apoyo, enseñanzas y todo el amor que me brindaron, y que ahora me cuidan desde el cielo.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

**Implementación de un sistema de información para la autenticación y autorización
centralizada en las aplicaciones BackOffice de una empresa de seguros**

Autor: Tacza Soto, Carlos Hugo

Asesor: Alcántara Loayza, César Augusto,

**Título: Trabajo de Suficiencia Profesional para optar el Título Profesional de
Ingeniero de Sistemas**

Fecha: Junio de 2022

RESUMEN

Hoy en día existen cada vez más empresas afectadas debido a ciberataques en sus sistemas de información. La constante evolución de los negocios hacia un esquema cada vez más digital exponen estos sistemas a posibles amenazas como el acceso no autorizado, lo cual puede dañar seriamente la empresa a nivel económico y reputacional. El presente trabajo de suficiencia profesional detalla la implementación de un sistema de información que permitirá centralizar la autenticación y autorización de los sistemas BackOffice que usan los colaboradores una empresa de seguros, reforzando así el control de acceso a estas y permitiendo escalar a formas de autenticación más seguras. Para el desarrollo del sistema se utilizaron los protocolos OpenID Connect y OAuth 2.0, además de una herramienta de gestión de identidades y acceso (IAM). El proyecto se realizó bajo la metodología SCRUM y en un equipo multidisciplinario en el área de Arquitectura de Seguridad Informática de una empresa de seguros. El resultado del proyecto fue la implementación del sistema e integración de las principales aplicaciones críticas de la Empresa Aseguradora. Gracias a esto se logró habilitar un componente único y reutilizable para todas las aplicaciones BackOffice, permitiendo minimizar el riesgo al que están expuestas

dichas aplicaciones debido al acceso no autorizado, además se utilizó una herramienta de gestión de identidades y acceso (IAM), la cual permitirá escalar la forma de autenticación de los colaboradores a una mucho más robusta según las tendencias actuales.

Palabras clave: Autenticación, Autorización, Oauth 2.0, OpenID Connect, IAM, seguridad

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

Implementation of an information system for centralized authentication and authorization in the BackOffice's applications of an insurance company.

Author: Tacza Soto, Carlos Hugo

Advisor: Alcántara Loayza, César Augusto

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

Date: June 2022

ABSTRACT

Nowadays there are more and more companies affected by cyber-attacks on their information systems. The constant evolution of business towards an increasingly digital scheme exposes these systems to possible threats such as unauthorized access, which can seriously damage the company at an economic and reputational level. The present work of professional sufficiency details the implementation of an information system that will allow centralizing the authentication and authorization of BackOffice systems used by the employees of an insurance company, thus reinforcing the access control to these systems, and allowing scaling to more secure forms of authentication. The OpenID Connect and OAuth 2.0 protocols were used for the development of the system, in addition to an identity and access management (IAM) tool. The project was carried out under the SCRUM methodology and in a multidisciplinary team in the Information Security Architecture area of an insurance company, which we will call Insurance Company for confidentiality reasons. The result of the project was the implementation of the system and integration of the main critical applications of the Insurance Company. Thanks to this, it was possible to enable a unique and reusable component for all BackOffice

applications, allowing to minimize the risk to which these applications are exposed due to unauthorized access. In addition, an Identity and Access Management (IAM) tool was used, which will allow to scale the authentication form of the collaborators to a much more robust one according to the current trends.

Key words: Authentication, Authorization, Oauth 2.0, OpenID Connect, IAM, security

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPITULO I TRAYECTORIA PROFESIONAL.....	3
1.1 Presentación profesional	3
1.3 Formación Académica	6
CAPITULO II CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA.....	8
2.1 Empresa – Actividad que realiza	8
2.2 Visión	8
2.3 Misión.....	8
2.4 Organización de la empresa.....	9
2.5 Empresa cliente – Actividad que realiza	10
2.6 Organización de la empresa cliente	11
2.7 Equipo del proyecto	13
2.5 Área, cargo y funciones desarrolladas	14
2.6 Experiencia profesional realizada en la organización.....	15
CAPITULO III ACTIVIDADES DESARROLLADAS	16
3.1 Situación Problemática	16
3.1.1 Definición del Problema	17
3.2 Solución	17
3.2.1 Objetivos	18
3.2.2. Alcance.....	19
3.2.3 Etapas y Metodología	19
3.2.4 Fundamentos Utilizados	25
3.2.4.1 Autenticación	25
3.2.4.2 Autorización.....	27
3.2.4.3 OAuth 2.0	28
3.2.4.4 OpenID Connect.....	32
3.2.4.5 Gestión de Identidad y Acceso (IAM)	33

3.2.4.6 Azure Active Directory.....	35
3.2.4.7 JWT	37
3.2.4.8 Token de Identidad o ID Token.....	40
3.2.4.9 Token de acceso o Access Token	41
3.2.4.10 Mecanismos de control de acceso	42
3.2.4.11 SCRUM	44
3.2.4.12 Microservicios.....	47
3.2.4.13 Metodología FAIR	48
3.2.4.14 OWASP (Open Web Application Security Project).....	51
3.2.5 Implementación de las áreas, procesos, sistemas y buenas prácticas.....	53
3.2.5.1 Análisis de aplicaciones BackOffice	53
3.2.5.2 Diseño de la arquitectura propuesta	64
3.2.5.3 Configuración de Azure Active Directory	69
3.2.5.4 Construcción de la Interfaz Web.....	72
3.2.5.5 Construcción de Microservicios	77
3.2.5.6 Pruebas	84
3.2.5.7 Liberación a Producción.....	91
3.2.5.8 Integración de Aplicaciones BackOffice críticas.....	93
3.3 Evaluación	111
3.3.1 Evaluación económica	111
3.3.2 Justificación Económica.....	114
3.3.1 Beneficios para la organización.....	114
CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA.....	116
CAPITULO V CONCLUSIONES Y RECOMENDACIONES.....	117
5.1 Conclusiones	117
5.2 Recomendaciones	118
5.3 Fuentes de Información.....	120
5.4 Glosario de términos.....	122

Índice de Figuras

Figura 1 Organigrama de la empresa HUNDRED - GFI	9
Figura 2 Organigrama de la Empresa Aseguradora cliente	11
Figura 3 Organigrama de la Gerencia de TI y Comunicaciones	12
Figura 4 Organización del equipo del proyecto	13
Figura 5 Etapas del proyecto	20
Figura 6 Scrum Team del proyecto	21
Figura 7 Elementos en el proceso de autorización	28
Figura 8 Diagrama del flujo Authorization code	29
Figura 9 Diagrama del flujo Implicit Flow	30
Figura 10 Diagrama del flujo Resource owner password	31
Figura 11 Diagrama del flujo Client credentials	32
Figura 12 Diagrama de secuencia OpenID Connect y OAuth 2.0 con Azure Active Directory	36
Figura 13 Estructura de un JWT	37
Figura 14 Estructura del JWT - Encabezado	38
Figura 15 Estructura del JWT - Contenido	38
Figura 16 Estructura del JWT - Firma	38
Figura 17 Ciclo de vida de un JWT	39
Figura 18 Diagrama de generación y utilización del ID Token	40
Figura 19 Diagrama de generación y utilización del Access Token	41
Figura 20 Petición usando Token Bearer	43
Figura 21 Flujo de autenticación basada en cookie	44
Figura 22 Ciclo de vida de Scrum	45
Figura 23 Magnitudes Económicas según metodología FAIR	49
Figura 24 Diagrama general de la arquitectura de las aplicaciones BackOffice Ecommerce	54
Figura 25 Pantalla de inicio – BackOffice Ecommerce Vida	55
Figura 26 Petición y respuesta del servicio de autorización – BackOffice Vida	56
Figura 27 Ejemplo de autenticación básica – BackOffice Canales	56
Figura 28 Diagrama general de la arquitectura de la aplicación BackOffice Reembolsos	57
Figura 29 Pantalla de inicio – BackOffice Reembolsos	58
Figura 30 Petición y respuesta del servicio de autorización – BackOffice Vida	59
Figura 31 Ejemplo de autenticación bearer – BackOffice Reembolsos	60

Figura 32 Diagrama general de la arquitectura de la aplicación BackOffice SOAT	61
Figura 33 Pantalla de inicio – BackOffice SOAT	62
Figura 34 Petición con Cookie de sesión – BackOffice SOAT	63
Figura 35 Diagrama de Solución propuesta	64
Figura 36 Diagrama de secuencia - BackOffice Ecommerce (Vida, Viajes, Autos, Hogar y Canales).....	66
Figura 37 Diagrama de secuencia - BackOffice Reembolsos.....	67
Figura 38 Diagrama de secuencia - BackOffice SOAT	68
Figura 39 Diagrama de Flujo Azure Active Directory – Frontend BackOffice.....	69
Figura 40 Portal de Inicio – Microsoft Azure	70
Figura 41 Buscador de Microsoft Azure.....	71
Figura 42 Pantalla de registro de aplicaciones en Azure Active Directory – Microsoft Azure.....	71
Figura 43 Registro de una aplicación Azure Active Directory – Microsoft Azure.....	72
Figura 44 Diagrama de Flujo Usuario corporativo – Interfaz Web	73
Figura 45 Código de implementación de la librería MSAL.js	74
Figura 46 Pantalla de inicio del sistema de autenticación centralizada	75
Figura 47 Pantalla de autenticación de Azure Active Directory	76
Figura 48 Diagrama de flujo - Microservicios	78
Figura 49 Diagrama de Flujo – Endpoint validate-ecommerce	79
Figura 50 Diagrama de Flujo – Endpoint validate-oauth	80
Figura 51 Diagrama de Flujo – Endpoint autoriza-ecommerce.....	81
Figura 52 Diagrama de Flujo – Endpoint autoriza-oauth.....	82
Figura 53 Evidencia Endpoint validate-ecommerce	85
Figura 54 Evidencia Endpoint validate-oauth	86
Figura 55 Evidencia Endpoint autoriza-ecommerce	87
Figura 56 Evidencia Endpoint autoriza-oauth.....	88
Figura 57 Aplicación BackOffice de prueba.....	89
Figura 58 Diagrama de procesos de la ejecución del pase a producción	91
Figura 59 Flujo de Integración BackOffice Ecommerce	94
Figura 60 Sección de código – Módulos del sistema BackOffice Ecommerce	95
Figura 61 Sección de código – Almacenamiento en localStorage – BackOffice Ecommerce.....	96
Figura 62 Sección de código – Envío de ID Token – BackOffice Ecommerce.....	97
Figura 63 Pantalla de inicio del usuario – BackOffice Ecommerce.....	98

Figura 64 Sección de código – Almacenamiento en localStorage de Basic Token – BackOffice Ecommerce	98
Figura 65 Pantalla de inicio del usuario – Envío de petición con Basic Token – BackOffice Ecommerce	99
Figura 66 Pantalla cierre de sesión – BackOffice Ecommerce	99
Figura 67 Flujo de Integración BackOffice Reembolsos	100
Figura 68 Sección de código – Módulos del sistema BackOffice	101
Figura 69 Sección de código – Almacenamiento en localStorage	102
Figura 70 Sección de código – Envío de ID Token	103
Figura 71 Pantalla de inicio del usuario – BackOffice Reembolsos	103
Figura 72 Sección de código – Almacenamiento en localStorage de Bearer Token – BackOffice Ecommerce	104
Figura 73 Envío de petición con Bearer Token – BackOffice Reembolsos	105
Figura 74 Pantalla cierre de sesión – BackOffice Reembolsos	105
Figura 75 Diagrama de flujo backend – BackOffice Reembolsos	106
Figura 76 Flujo de Integración BackOffice SOAT	108
Figura 77 Diagrama de flujo backend – BackOffice Reembolsos	110

Índice de Tablas

Tabla 1	Experiencia Profesional	3
Tabla 2	Formación Académica	6
Tabla 3	Otros conocimientos.....	7
Tabla 4	Datos generales de la empresa HUNDRED - GFI	9
Tabla 5	Datos generales de la empresa cliente	10
Tabla 6	Cronograma del proyecto.....	22
Tabla 7	Roles de Scrum.....	45
Tabla 8	Ceremonias de Scrum	46
Tabla 9	Artefactos de Scrum.....	47
Tabla 10	Formas de pérdida según metodología FAIR.....	50
Tabla 11	Estructura de archivos – Interfaz Web.....	73
Tabla 12	Estructura de archivos – Microservicios	83
Tabla 13	Tipos de pruebas.....	90
Tabla 14	Duración de Integración por aplicación	93
Tabla 15	Costo del Equipo del Proyecto.....	112
Tabla 16	Costo del despliegue de la Infraestructura	112
Tabla 17	Pérdida económica estimada por aplicación	113
Tabla 18	Justificación económica del proyecto	114

INTRODUCCIÓN

Las aplicaciones BackOffice, son sistemas de información que ha creado o adquirido una organización y que son utilizadas única y exclusivamente por los colaboradores de dicha organización o proveedores autorizados.

Para que los colaboradores puedan tener acceso a estas aplicaciones, deben seguir un proceso de registro diferente en cada una de ellas, en consecuencia, mientras necesiten acceso a más aplicaciones, mayor será el número de credenciales que se deben generar.

Esto dio origen a una serie de dificultades para el colaborador y la organización, denominada en adelante Empresa Aseguradora por razones de confidencialidad, como: desorden en la gestión de credenciales, generando a su vez la utilización de contraseñas no seguras, repetidas y compartidas, lo que pueden conllevar a el acceso no autorizado de personas ajenas a la organización y esto a su vez ocasionar daños económicos y reputacionales a la organización.

Para solucionar este problema, se planteó crear un sistema que centralice la autenticación y autorización de las aplicaciones BackOffice a las que acceden los colaboradores. Para ello la solución se basó en el protocolo OpenID Connect para la autenticación y utilizó la herramienta de gestión de identidades y acceso (IAM) de Microsoft, mientras que para la capa de autorización se utilizó el repositorio de roles de la Empresa y se basó en el protocolo OAuth 2.0. Adicional a ello la solución tuvo que adaptarse a los mecanismos de autorización que ya usaban algunas aplicaciones.

Esto permitió que los colaboradores de la Empresa Aseguradora puedan ingresar a las aplicaciones BackOffice solo con sus credenciales de correo corporativo, además

elevó el nivel de seguridad de las aplicaciones y las preparó para en un futuro migrar a formas de autenticación más robustas.

A continuación, se detalla la estructura del trabajo:

En el CAPÍTULO I: Se explica de forma cronológica la trayectoria profesional del autor. En esta sección se especifican los roles y funciones que ha ocupado en cada puesto de su trayectoria profesional. Se incluyen, a nivel académico, los conocimientos adquiridos por el autor a través de diversos estudios y experiencias profesionales.

En el CAPÍTULO II: Se explica la empresa en la cual se desarrolló la experiencia profesional, visión, misión, organigrama, servicios que presta y las funciones que el autor desempeñó dentro de la misma.

En el CAPITULO III Se detalla el trabajo realizado y se destaca la situación problemática, objetivos, métodos utilizados, fundamentos utilizados, desarrollo del proyecto y los aspectos más importantes en el proceso de realización de este trabajo.

En el CAPITULO IV Se describen las reflexiones y contribuciones críticas del autor, las experiencias adquiridas, los conocimientos que requirió y adquirió con la realización del proyecto.

En el CAPÍTULO V: Se describen finalmente las conclusiones y recomendaciones expuestas por el autor.

CAPITULO I TRAYECTORIA PROFESIONAL

1.1 Presentación profesional

El autor del presente trabajo de suficiencia profesional es bachiller en Ingeniería de Sistemas e Informática, con más de 6 años de experiencia en áreas de tecnología de información y ciberseguridad. Posee habilidades de liderazgo, facilidad para trabajar en equipo y alcanzar responsablemente las metas establecidas.

Tiene experiencia en el desarrollo e implementación de sistemas de información en los rubros de banca, seguros y retail, actualmente se ha especializado en el campo de la seguridad informática y ejerce el cargo de Arquitecto de seguridad de aplicaciones en NTT DATA.

1.2 Experiencia Profesional

Tabla 1

Experiencia Profesional

NTTDATA Europe & Latam	
Mayo 2022 - Actualidad	
Cargo	Arquitecto de Seguridad

Funciones	- Definir la arquitectura de seguridad de aplicaciones y velar por su cumplimiento desde el punto de vista técnico para contribuir a mejorar los niveles de seguridad de la compañía. - Participar junto con las otras disciplinas de arquitectura en la creación de un modelo de solución que mejor se adapte a las necesidades de negocio y que cumpla con los estándares de seguridad. - Mejorar los procesos o análisis de seguridad durante todo el ciclo de vida de software. - Formación de seguridad a equipos de desarrollo.
------------------	--

Banco Ripley

Marzo 2021 – Abril 2022

Cargo	Líder de Seguridad Informática
Funciones	- Diseñar e implementar una estrategia de gestión de identidades y accesos de clientes (CIAM) y colaboradores (IAM) desde una perspectiva tecnológica, alineada con la estrategia de Transformación Digital del banco. - Diseñar y definir las iniciativas relacionadas a identidad digital en los sistemas de información de la organización. - Gestionar los riesgos sobre el proceso de autenticación y autorización, así como sobre posibles vulnerabilidades referidas a los métodos de autenticación de clientes.

Banco de Crédito del Perú

Octubre 2020 – Marzo 2021

Cargo	Analista Senior de Seguridad de Aplicaciones
--------------	---

Funciones	- Garantizar la seguridad en las distintas aplicaciones de los equipos ágiles (web, api, mobile). - Participar en el ciclo de vida de desarrollo de las aplicaciones con el fin de asegurar el cumplimiento de políticas, lineamientos y normas de seguridad. - Entrenar y capacitar a los equipos ágiles en prácticas de desarrollo seguro.
HUNDRED - GFI Perú	
Setiembre 2019 – Octubre 2020	
Cargo	Technical Consultant
Funciones	- Diseñar las soluciones para el desarrollo de las aplicaciones de seguridad en una Empresa Aseguradora. - Liderar equipo de desarrollo de acuerdo con los requerimientos del negocio y basados en las buenas prácticas de desarrollo de software y de seguridad. - Asegurarse que el equipo de desarrollo ejecute sus tareas en tiempo y con la debida calidad.
CINEPLANET	
Octubre 2017 – Setiembre 2019	
Cargo	Analista Programador
	- Análisis de requerimientos de las distintas áreas de la empresa. - Desarrollo y mantenimiento de los sistemas internos de los colaboradores. - Desarrollo y mantenimiento de los canales digitales (Web, mobile y boletería) para Perú y Chile.
INDRA Perú	

Mayo 2016 – Setiembre 2017

Cargo**Analista Programador**

- Análisis, desarrollo y mantenimiento de los sistemas del banco BBVA.

- Acompañamiento de pases a producción y resolución de incidentes.

Sypsoft SA

Setiembre 2015 – Abril 2016

Cargo**Practicante de desarrollo**

- Análisis y desarrollo de aplicaciones de Facturación electrónica y sistemas de punto de venta.

- Resolución de incidentes.

Nota. Fuente: Elaboración propia.

1.3 Formación Académica

Tabla 2

Formación Académica

2011 - 2016	Universidad Nacional Mayor de San Marcos - Facultad de Ingeniería de Sistemas e Informática Grado académico de Bachiller en Ingeniería de Sistemas
2018	Scrum.org Scrum Master Profesional I (PSM I)
2021	Microsoft Microsoft Certified: Azure Fundamentals
2021	Certiprof Lead Cybersecurity Professional Certificate LCSPC
2021	API Academy

	API Security Architect
2022	AWS
	AWS Cloud Practitioner

Nota. Fuente: Elaboración propia.

Tabla 3

Otros conocimientos

Lenguajes de programación	C++, Java, Javascript, PHP, Python
Frameworks	React, Vue, Angular, Springboot, Laravel
Cloud	AWS, Microsoft Azure
Sistemas operativos	Windows Server, MacOS, Linux

Nota. Fuente: Elaboración propia.

CAPITULO II CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA

2.1 Empresa – Actividad que realiza

HUNDRED - GFI es una empresa peruana con experiencia Multinacional que brinda servicios informáticos. Cuenta con más de 350 empleados y 15 años de experiencia en el mercado local y regional. Sus clientes son importantes empresas en los rubros de telecomunicaciones, banca y seguros. En el año 2020 la empresa de origen francés Inetum compra la compañía como parte de su proceso de expansión en Latinoamérica.

La empresa Hundred – GFI fue contratada por la Empresa Aseguradora para realizar el proyecto.

2.2 Visión

La misión de la empresa es ayudar a sus clientes con servicios profesionales de calidad, competencia y objetividad, entregando las mejores soluciones para satisfacer sus objetivos tecnológicos, de procesos de negocio y de gestión.

2.3 Misión

Transformar nuestro entorno con soluciones tecnológicas para vivir más felices.

En la Tabla 4, se describen algunos datos generales de la empresa.

Tabla 4

Datos generales de la empresa HUNDRED - GFI

Más de 350 empelados
Presencia en 3 países de Latinoamérica
Más de 10 millones de soles en ingresos anuales
Más de 60% de los empleados están certificados en tecnologías de vanguardia

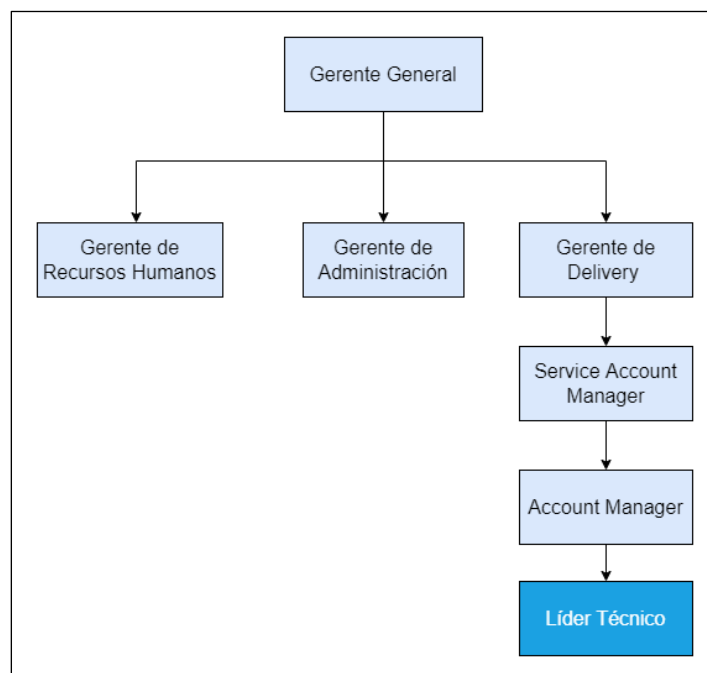
Nota. Fuente: Elaboración propia.

2.4 Organización de la empresa

Se describe el organigrama de la empresa Hundred – GFI.

Figura 1

Organigrama de la empresa HUNDRED - GFI



Nota. Fuente: Elaboración propia.

2.5 Empresa cliente – Actividad que realiza

La Empresa Aseguradora es una organización peruana que comenzó con sus actividades en el año 1992, después de la unificación de dos empresas aseguradoras que pertenecían al mercado peruana desde 1940.

En 2012 la Empresa Aseguradora se consolidó en el mercado asegurador peruano con una participación del 27% en el mercado de Seguros Generales, 46,67% en EPS y 27,8% en Vida y facturando US\$ 1.019,9 millones con una utilidad neta de US\$ 70 millones.

Actualmente forma parte del mercado asegurador peruano y posee tres líneas de negocio: Salud, Vida y Riesgos Generales. Forma parte de un holding financiero nacional. y cuenta con el respaldo de un holding de servicios de salud internacional.

Debido al gran crecimiento en los últimos años la empresa viene realizando un proceso de transformación digital en sus canales digitales y adaptándose para crear nuevos productos para las cambiantes necesidad de sus clientes. Para ello ha creado un centro de innovación con el cual busca acelerar esta creación y transformar la cultura de la organización hacia un camino más ágil.

Tabla 5

Datos generales de la empresa cliente

3, 409 empleados
10 agencias de atención en el país
24 aliados comerciales
1, 274 corredores de seguros
Valor de pólizas: S/. 2,946,621 miles de millones de soles
Patrimonio: 2,867 miles de millones de soles

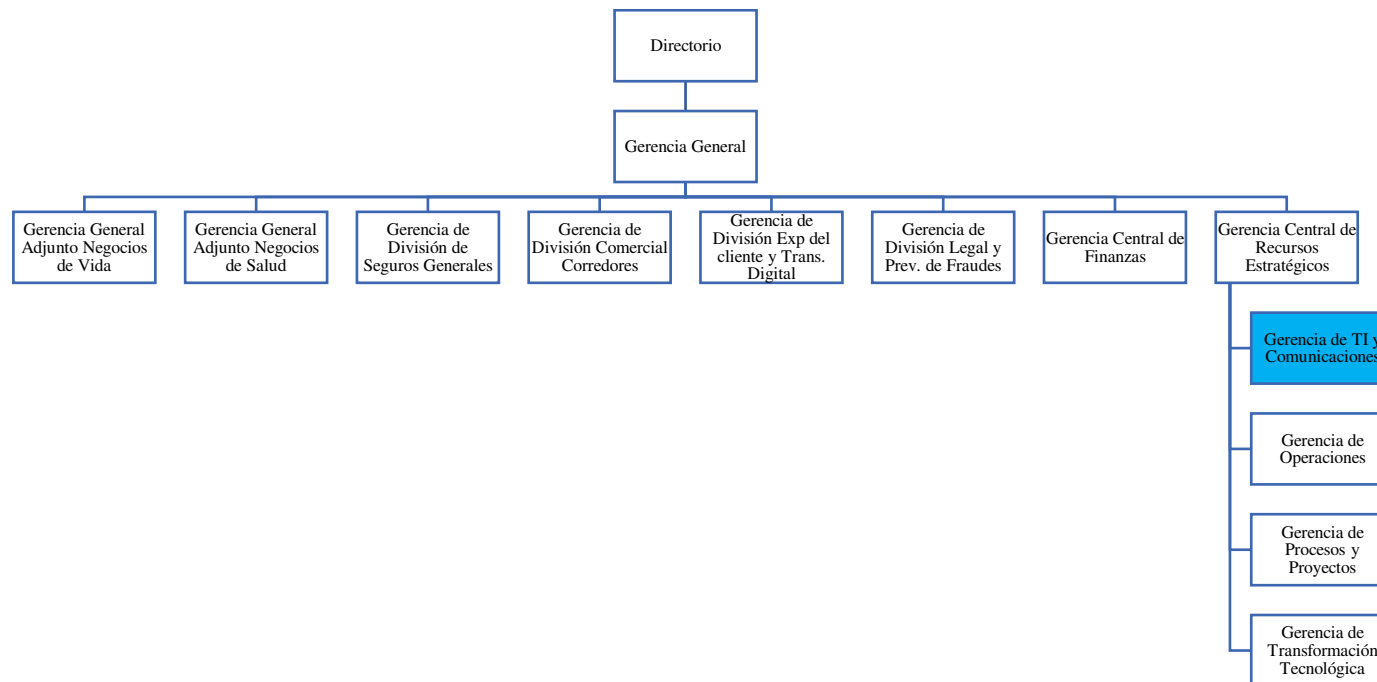
Nota. Fuente: Elaboración propia.

2.6 Organización de la empresa cliente

A continuación se detalla el organigrama de la empresa cliente donde se desarrolló el trabajo de suficiencia profesional.

Figura 2

Organigrama de la Empresa Aseguradora cliente

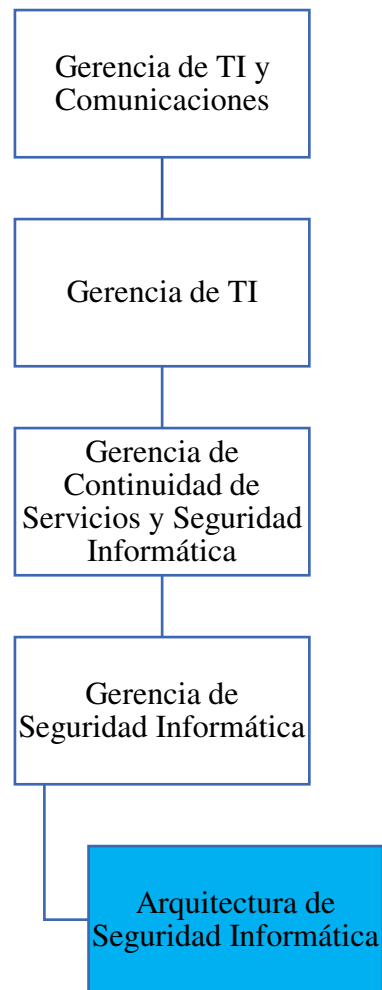


Nota. Fuente: Empresa Aseguradora, 2017.

A continuación, se describe el organigrama de la unidad de la empresa cliente donde se desarrolló la experiencia profesional.

Figura 3

Organigrama de la Gerencia de TI y Comunicaciones



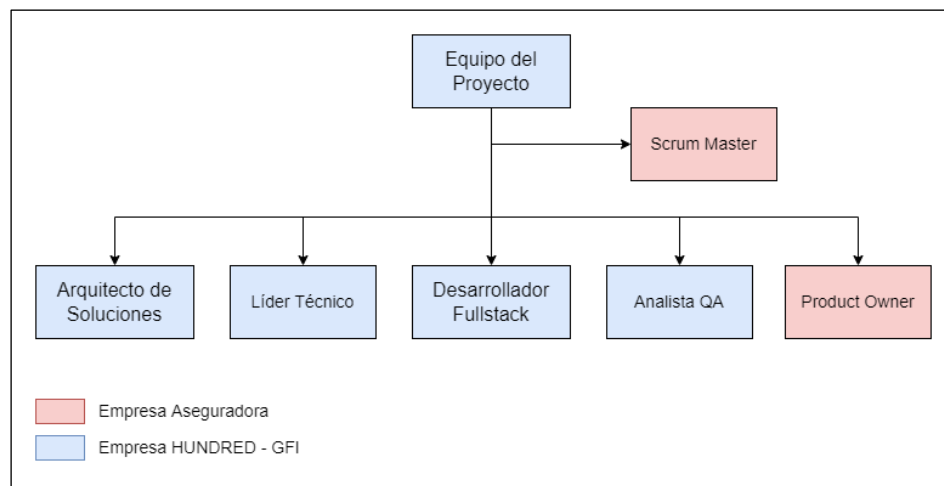
Nota. Fuente: Empresa Aseguradora, 2017.

2.7 Equipo del proyecto

El equipo del proyecto de “Autenticación y Autorización Centralizada para Colaboradores” se integró por personas expertas en el área de tecnología de la empresa Hundred - GFI, con el apoyo y la participación de la unidad de “Arquitectura de Seguridad Informática” de la Empresa Aseguradora. En dicho equipo el autor se desempeñó como Líder Técnico.

Figura 4

Organización del equipo del proyecto



Nota. Fuente: Elaboración propia.

2.5 Área, cargo y funciones desarrolladas

Área: Arquitectura de Seguridad Informática

Cargo: Líder Técnico (Hundred – GFI)

Funciones Desempeñadas:

- Liderar equipo de desarrollo de acuerdo con los requerimientos del negocio y basados en las buenas prácticas de desarrollo de software y de seguridad.
- Asegurarse que el equipo de desarrollo tenga el apropiado entendimiento de los requerimientos del equipo de producto.
- Coordinar con los diferentes equipos para el despliegue y correcto funcionamiento de la solución.
- Revisar y sugerir cambios en la arquitectura de componentes y adopción de frameworks o librerías.
- Contribuir con las políticas de seguridad de la empresa, incentivando un desarrollo seguro de las aplicaciones.

A continuación se nombran los proyectos en los que participó el autor, dentro de la Empresa Aseguradora:

- Proyecto de Autenticación Centralizada (CIAM) para clientes.
- Proyecto de Autenticación y Autorización Centralizada (IAM) para Colaboradores de la empresa. (Seguro de Autos, Viajes, Vida, Hogar, Rembolsos, SOAT, Canales).
- Implementación de MFA para aplicaciones de corredores de seguros.
- Proyecto de desactivación de cuentas de usuarios para corredores de seguros.

2.6 Experiencia profesional realizada en la organización

El autor del presente trabajo de suficiencia profesional se desempeñó como Líder Técnico de la Empresa Hundred - GFI e integró el equipo del proyecto (Squad) “Autenticación y Autorización Centralizada para Colaboradores” en la Empresa Aseguradora.

El equipo estaba orientado a desarrollar soluciones que permitan elevar la postura de seguridad de los sistemas de información y cumplir con los lineamientos de seguridad establecidos por la Empresa Aseguradora y por el holding financiero nacional al que pertenece.

Debido al buen desempeño que el autor tuvo en un proyecto previo denominado Autenticación Centralizada para Clientes, que duró aproximadamente 3 meses, se plantearon nuevas iniciativas entre ellas el proyecto de “Autenticación y Autorización Centralizada para Colaboradores”, el cual generó mucho aprendizaje, entre los cuales se destacan: comprender arquitecturas de varios proyectos, explorar nuevas herramientas y conocer nuevos lenguajes de programación.

Adicional a lo anteriormente mencionado, la solución diseñada debía cumplir todos los requerimientos de seguridad de la Empresa Aseguradora y el equipo del proyecto, ser un referente para los demás en temas de seguridad y autenticación, para lo cual se debió revisar y aprender buenas prácticas en desarrollo seguro basadas en el ranking de vulnerabilidades de OWASP (Open Web Application Security Project).

Gracias al esfuerzo de todo el equipo del proyecto se logró cumplir con el objetivo, el cual era implementar un sistema de información escalable, reutilizable y seguro, e integrar las 7 aplicaciones BackOffice críticas de la Empresa Aseguradora.

CAPITULO III ACTIVIDADES DESARROLLADAS

3.1 Situación Problemática

Las aplicaciones BackOffice de una organización, son sistemas de información que utilizan los colaboradores como parte de sus funciones diarias, ayudando a optimizar su trabajo y por ende generando valor al negocio. Algunas de las funciones de estos sistemas son por ejemplo crear campañas, actualizar datos de clientes, crear nuevos productos, actualizar precios, generar y descargar reportes, etc.

Muchos de estos sistemas actualmente no cuentan con los controles de seguridad adecuados para asegurar que la información que estos manejan no sea accedida por personas externas a la organización y utilizada de manera maliciosa con el fin de perjudicar reputacional o económicamente a la Empresa Aseguradora.

En la investigación realizada por el instituto Ponemon y patrocinada por IBM Security en 17 países y 18 sectores económicos, se halló que el costo por una violación de datos (filtración y exposición de datos sensibles) ha aumentado en 10 % desde el año 2020, llegando a un promedio de 4,24 millones de dólares por violación. (IBM Security, 2021)

A nivel de Latinoamérica, el Perú es el segundo país de América Latina más afectado por incidentes de seguridad, 71 % de los incidentes en América Latina se concentraron en nuestro país. Entre los principales incidentes reportados por las empresas se tienen: El acceso indebido (61 %), el robo de información (58%) y la privacidad de la información (48%). (ESET, 2019)

3.1.1 Definición del Problema

Se identificaron las siguientes vulnerabilidades en las aplicaciones BackOffice de la Empresa Aseguradora:

- No existe la facilidad para aplicar las políticas de contraseñas que brinda como lineamiento la Empresa Aseguradora y que se basa en estándares y buenas prácticas.
- Existe un número elevado de aplicaciones y cada una con un módulo de autenticación y autorización independiente, ocasionando que sea cada vez más difícil para los colaboradores recordar sus credenciales.
- El módulo de autenticación de las aplicaciones no está preparado para soportar formas de autenticación más robustas como doble factor de autenticación o biometría.

Lo mencionado anteriormente ha ocasionado que la gestión de credenciales sea engorrosa y dispersa, pudiendo ocasionar fallas de seguridad en las aplicaciones BackOffice, como el acceso no autorizado, lo que conllevaría a pérdidas económicas importantes para la Empresa Aseguradora.

3.2 Solución

Para resolver el problema identificado se propone crear un componente único que permita reemplazar los módulos de autenticación y autorización de las aplicaciones BackOffice de los colaboradores.

Para ello se desarrolló un sistema de información que utilizó la herramienta Azure Active Directory como proveedor de identidad, la cual resguarda y centraliza toda la información de los colaboradores de la organización.

Además, se utilizó el componente de autorización de la Empresa Aseguradora que guarda toda la información de roles y funciones de los colaboradores.

Finalmente, el colaborador podrá utilizar la contraseña de su correo corporativo como única forma para ingresar a las aplicaciones BackOffice que necesite en el desempeño de sus funciones.

3.2.1 Objetivos

Objetivo General

El principal objetivo del presente proyecto fue crear e implementar un sistema único, seguro, escalable, desacoplado y fácil de integrar para la autenticación y autorización de colaboradores de la Empresa Aseguradora en 7 aplicativos BackOffice. Incrementando la seguridad en dichas aplicaciones y permitiendo que las pérdidas económicas generadas por el acceso no autorizado sean menos factibles.

Una vez implementada la solución, todos los sistemas de información que usan los colaboradores podrán integrarse a este componente y desacoplar su módulo de autenticación.

Objetivos Específicos

- Analizar los módulos de autenticación y autorización de los sistemas de información que utilizan los colaboradores de la Empresa Aseguradora.
- Implementar los componentes necesarios para centralizar la autenticación y autorización de las aplicaciones BackOffice de la Empresa Aseguradora.
- Configurar e integrar la herramienta Azure Active Directory al “Sistema de Autenticación y Autorización Centralizada para Colaboradores”.
- Integrar el componente de autorización de la Empresa Aseguradora, con el fin de obtener la información de roles y funciones de los colaboradores.
- Integrar las 7 aplicaciones BackOffice principales que utilizan los colaboradores, catalogadas como críticas para la Empresa Aseguradora.
- Proporcionar capacitaciones y material de formación a los equipos para que conozcan el sistema de autenticación centralizado y puedan acceder a él.

- Explorar y analizar nuevos mecanismos de autenticación que brinda la herramienta Azure Active Directory.

3.2.2. Alcance

- Implementar un sistema centralizado de autenticación al cual puedan integrarse las aplicaciones de colaboradores de la Empresa Aseguradora.
- Utilizar la herramienta de gestión de identidades y acceso, de la Empresa Aseguradora, para gestionar la autenticación.
- Utilizar el repositorio de roles y funciones de los colaboradores de la Empresa Aseguradora para incorporar la autorización hacia las aplicaciones.
- Integrar las 7 aplicaciones BackOffice catalogadas como críticas para la organización al sistema de autenticación y autorización centralizado.

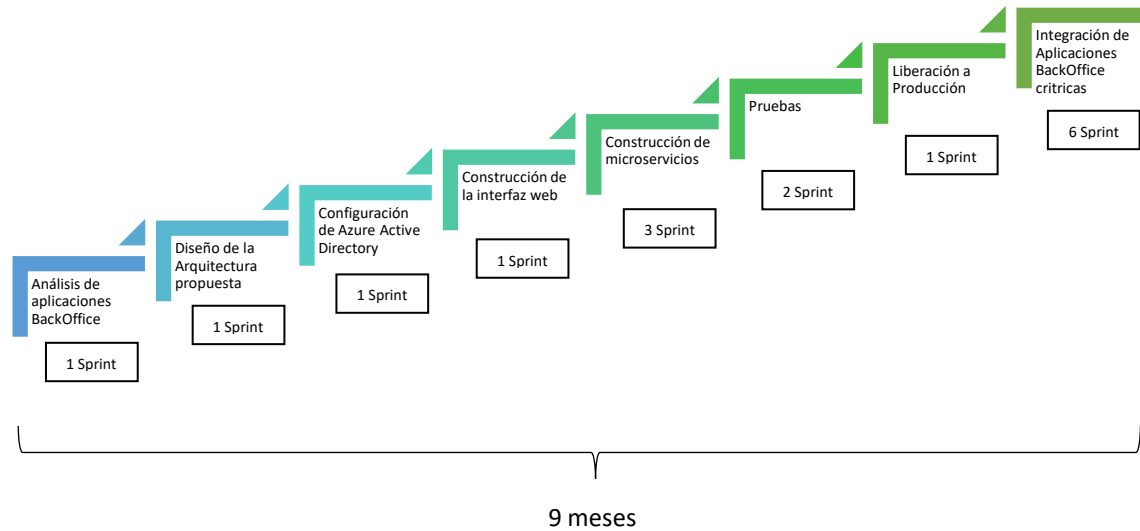
3.2.3 Etapas y Metodología

Para la implementación del proyecto de “Autenticación y Autorización Centralizada para Colaboradores” se tuvieron las siguientes consideraciones:

- El proyecto estuvo en curso por un lapso de 9 meses, desde el 25 de enero del 2022 hasta el 3 de Setiembre de 2022.
- La duración de los sprints fue de 10 días hábiles
- El autor del presente trabajo de suficiencia profesional estuvo presente durante toda la duración del proyecto y tuvo el rol de Líder Técnico.
- Como se puede visualizar en la Figura 5, el proyecto estuvo dividido en 8 etapas, las cuales van desde el análisis de los sistemas BackOffice actuales hasta la integración de las 7 aplicaciones críticas de la Empresa Aseguradora.

Figura 5

Etapas del proyecto

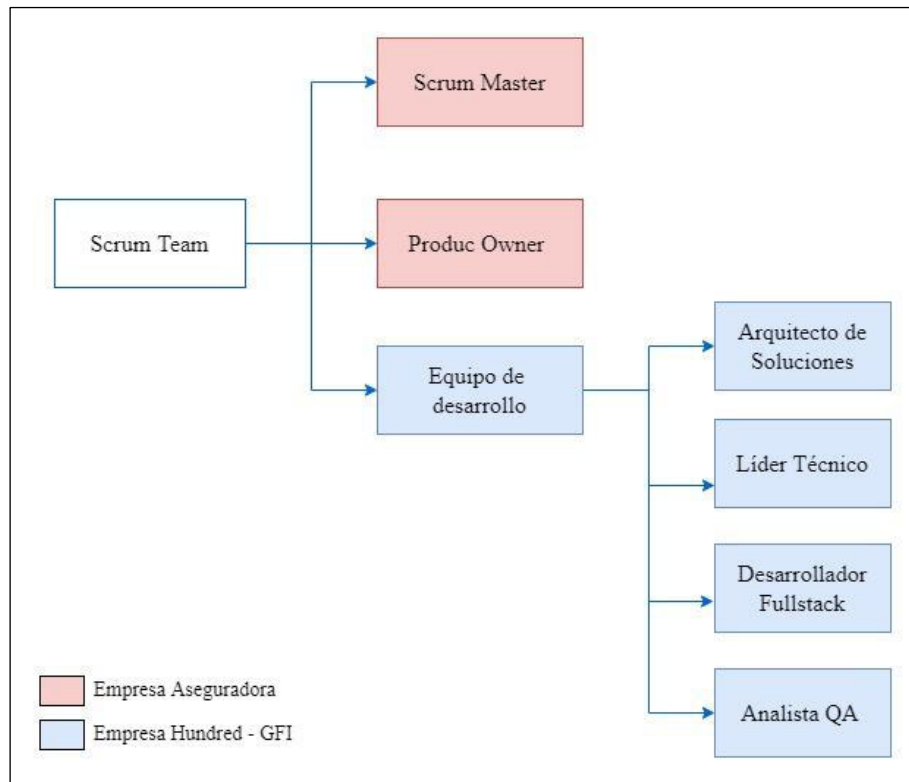


Nota. Fuente: Elaboración propia.

En la Figura 6 se describen los roles que conformaron el proyecto, los cuales pertenecían a la Empresa Aseguradora y a la Empresa Hundred-GFI

Figura 6

Scrum Team del proyecto



Nota. Fuente: Elaboración propia.

A continuación, se detallan los roles del equipo Scrum del proyecto:

Product Owner: Desempeñaba el cargo de Jefe de Arquitectura de Seguridad en la Empresa Aseguradora, fue el experto en los lineamientos y mejores prácticas de autenticación y autorización, en las cuales se basó el sistema, así mismo gestionó el proyecto y priorizaba las historias de usuario que se desarrollaban en cada Sprint.

Scrum Máster: Desempeñaba el cargo de Agile Coach en la Empresa Aseguradora, facilitaba la adopción del marco de trabajo SCRUM participando en cada ceremonia con el fin de que el equipo pueda entender los objetivos, así mismo daba seguimiento de las historias de usuario del proyecto.

Equipo de desarrollo: Estuvo compuesto por los siguientes roles 1 Arquitecto de Soluciones, 1 Líder Técnico, 1 Desarrollador Fullstack y 1 Analista QA, todos los integrantes pertenecían a una Empresa Externa.

En la Tabla 6 se presenta el cronograma del proyecto de “Autenticación y Autorización Centralizada para Colaboradores”, el cual inició el 25 de enero y terminó el 3 de septiembre del año 2020.

Tabla 6
Cronograma del proyecto

Etapas	Sprint	Ceremonia	Inicio	Fin	Duración (En días hábiles)
Análisis de aplicaciones BackOffice	Sprint 1	Sprint Planning	25/01/2020	25/01/2020	1
		Ejecución del sprint	26/01/2020	04/02/2020	8
		Sprint Review	05/02/2020	05/02/2020	1
		Sprint Retrospective	05/02/2020	05/02/2020	
Diseño de la Arquitectura propuesta	Sprint 2	Sprint Planning	08/02/2020	08/02/2020	1
		Ejecución del sprint	09/02/2020	18/02/2020	8
		Sprint Review	19/02/2020	19/02/2020	1
		Sprint Retrospective	19/02/2020	19/02/2020	
Configuración de Azure Active Directory	Sprint 3	Sprint Planning	22/02/2020	22/02/2020	1
		Ejecución del sprint	23/02/2020	4/03/2020	8
		Sprint Review	5/03/2020	5/03/2020	1
		Sprint Retrospective	5/03/2020	5/03/2020	

Construcción de la Interfaz Web	Sprint 4	Sprint Planning	8/03/2020	8/03/2020	1
		Ejecución del sprint	9/03/2020	18/03/2020	8
		Sprint Review	19/03/2020	19/03/2020	1
		Sprint Retrospective	19/03/2020	19/03/2020	
Construcción de Microservicios	Sprint 5	Sprint Planning	22/03/2020	22/03/2020	1
		Ejecución del sprint	23/03/2020	1/04/2020	8
		Sprint Review	2/04/2020	2/04/2020	1
		Sprint Retrospective	2/04/2020	2/04/2020	
	Sprint 6	Sprint Planning	5/04/2020	5/04/2020	1
		Ejecución del sprint	6/04/2020	15/04/2020	8
		Sprint Review	16/04/2020	16/04/2020	1
		Sprint Retrospective	16/04/2020	16/04/2020	
	Sprint 7	Sprint Planning	19/04/2020	19/04/2020	1
		Ejecución del sprint	20/04/2020	29/04/2020	8
		Sprint Review	30/04/2020	30/04/2020	1
		Sprint Retrospective	30/04/2020	30/04/2020	
Pruebas	Sprint 8	Sprint Planning	3/05/2020	3/05/2020	1
		Ejecución del sprint	4/05/2020	13/05/2020	8
		Sprint Review	14/05/2020	14/05/2020	1
		Sprint Retrospective	14/05/2020	14/05/2020	
	Sprint 9	Sprint Planning	17/05/2020	17/05/2020	1
		Ejecución del sprint	18/05/2020	27/05/2020	8

		Sprint Review	28/05/2020	28/05/2020	1
		Sprint Retrospective	28/05/2020	28/05/2020	
Liberación a Producción	Sprint 10	Sprint Planning	31/05/2020	31/05/2020	1
		Ejecución del sprint	1/06/2020	10/06/2020	8
		Sprint Review	11/06/2020	11/06/2020	1
		Sprint Retrospective	11/06/2020	11/06/2020	
Integración de aplicaciones BackOffice críticas	Sprint 11	Sprint Planning	14/06/2020	14/06/2020	1
		Ejecución del sprint	15/06/2020	24/06/2020	8
		Sprint Review	25/06/2020	25/06/2020	1
		Sprint Retrospective	25/06/2020	25/06/2020	
	Sprint 12	Sprint Planning	28/06/2020	28/06/2020	1
		Ejecución del sprint	29/06/2020	8/07/2020	8
		Sprint Review	9/07/2020	9/07/2020	1
		Sprint Retrospective	9/07/2020	9/07/2020	
	Sprint 13	Sprint Planning	12/07/2020	12/07/2020	1
		Ejecución del sprint	13/07/2020	22/07/2020	8
		Sprint Review	23/07/2020	23/07/2020	1
		Sprint Retrospective	23/07/2020	23/07/2020	
	Sprint 14	Sprint Planning	26/07/2020	26/07/2020	1
		Ejecución del sprint	27/07/2020	05/07/2020	8
		Sprint Review	06/08/2020	06/08/2020	1
		Sprint Retrospective	06/08/2020	06/08/2020	

Sprint 15	Sprint Planning	9/08/2020	9/08/2020	1
	Ejecución del sprint	10/08/2020	19/08/2020	8
	Sprint Review	20/08/2020	20/08/2020	1
	Sprint Retrospective	20/08/2020	20/08/2020	
Sprint 16	Sprint Planning	23/08/2020	23/08/2020	1
	Ejecución del sprint	24/08/2020	02/09/2020	8
	Sprint Review	03/09/2020	03/09/2020	1
	Sprint Retrospective	03/09/2020	03/09/2020	

Nota. Fuente: Elaboración propia.

3.2.4 Fundamentos Utilizados

3.2.4.1 Autenticación

La autenticación es la capacidad de demostrar que un usuario o aplicación es realmente quien dice ser. (OWASP, 2018)

Como ejemplo, si consideramos el caso de un usuario que se conecta a un sistema especificando un ID de usuario y una contraseña. El sistema utilizará el ID de usuario para identificar al mismo y autenticará al usuario en el instante de la conexión, verificando que la contraseña proporcionada es correcta y dándole acceso al sistema. (IBM, 2021)

Hoy en día los métodos de autenticación han evolucionado y la tendencia actual es identificar correctamente a los usuarios mediante la autenticación multifactor, la cual se basa en un método de autenticación combinado que va más allá de factores simples como el nombre de usuario y la contraseña.

Métodos de autenticación

Los métodos de autenticación pueden clasificarse en tres grupos (Mendoza Arteaga, Bolaños Burgos, Cedeño Sarmiento, & Saltos Rivas, 2020):

- Algo que sabes: un secreto que solo el usuario tiene conocimiento, como una contraseña o preguntas de seguridad.
- Algo que tienes: se refiere a cualquier dispositivo físico que el usuario posea en el momento de la autenticación.
- Algo que eres: Es el análisis de algún rasgo físico o comportamiento del usuario.

Algo que sabes: Contraseña

El método de usuario y contraseña es el más conocido y utilizado en la mayoría de los sistemas de información, esta se encuentra dentro del grupo algo que sabes, sin embargo, su mayor defecto es que su seguridad depende únicamente del usuario y esto en muchas ocasiones conducen a malas prácticas por parte de ellos como anotar la contraseña en el celular, en un archivo dentro de la computadora y hasta pegada en el monitor.

Algo que tienes: Token

El token es un método de autenticación del grupo algo que tienes, el cual es un dispositivo de seguridad que se utiliza para firmar digitalmente y cifrar mensajes, existen varias clases de tokens, como los que comúnmente conocemos como generadores de contraseñas dinámicas OTP o tokens digitales, los cuales están basados en software y los tokens físicos, los cuales además de permitir almacenar contraseñas y certificados, permiten almacenar la identidad digital de la persona.

Algo que eres: Biometría

Se refiere a la autenticación de una persona por sus rasgos físicos, huellas dactilares, voz, iris, forma de su rostro y otras características únicas. Así, la identificación biométrica implica almacenar, medir y comparar las características de una persona.

3.2.4.2 Autorización

La autorización es el proceso de permitir o denegar a alguien la capacidad de acceder a un recurso de acuerdo con las políticas de control de acceso emitidas por los propietarios de dicho recurso. (Esposito, Massimo, & Brij Bhooshan, 2021)

En los sistemas informáticos, las reglas de autorización forman parte de una disciplina de TI llamada Gestión de acceso e identidad (IAM). En IAM, la autorización y la autenticación ayudan a los administradores del sistema a controlar quién tiene acceso a los recursos del sistema y establecer permisos de clientes. (Auth0, 2022)

Por ejemplo, considere una herramienta de colaboración como Google Docs, esta le permite crear, actualizar y compartir documentos, el propietario puede compartirlo con otra persona y solo permitir que agregue comentarios o definir una serie de políticas de control de acceso para dicho documento. (Auth0, 2022)

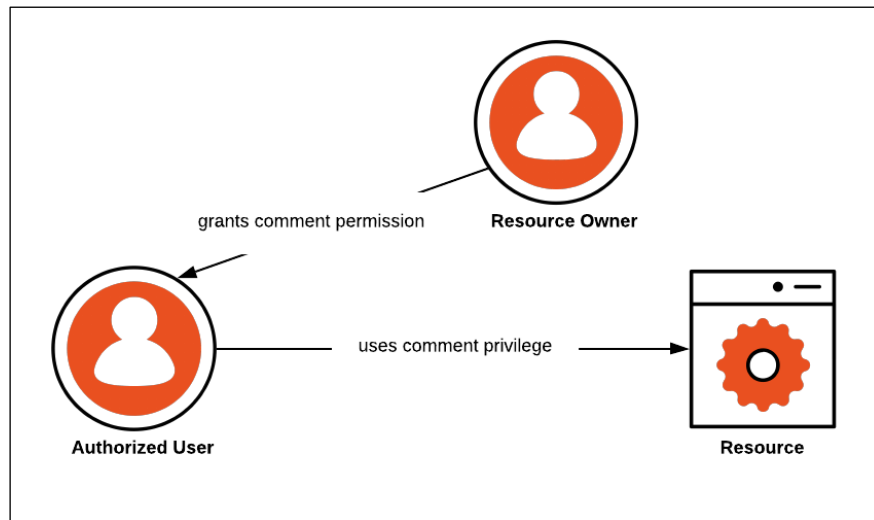
En este escenario, se podrían identificar los siguientes elementos

- Recurso: Documento de Google
- Propietario: Se refiere al usuario que creó el documento
- Usuario Autorizado: Se refiere al usuario al cual el propietario del recurso le otorga derechos de agregar comentarios

En la Figura 7 se representa la autorización de acceso a los recursos y la forma en como interactúan los elementos.

Figura 7

Elementos en el proceso de autorización



Nota. Tomado de www.auth0.com, Introduction To IAM.

3.2.4.3 OAuth 2.0

“El marco de autorización OAuth 2.0 es un protocolo que permite a un usuario otorgar a un sitio web o aplicación de tercero acceso a los recursos protegidos del usuario, sin necesariamente revelar sus credenciales a largo plazo o incluso su identidad.” (OAuth 2.0, 2012)

Roles

Un flujo de oauth contiene los siguientes roles (Mahendrakumaran, 2021):

- **Resource Owner:** Una entidad que puede otorgar acceso a recursos protegidos. Por lo general, este es el usuario final.
- **Resource Server:** Se refiere al recurso que se quiere acceder
- **Client Application:** Hace referencia a una aplicación que necesita acceder a un recurso protegido en nombre del propietario del recurso.

- **Authorization Server:** Es un componente tecnológico que permite autenticar al dueño del recurso y le brinda tokens de acceso una vez finalizada la autorización.

Grand Types o Tipos de concesión

OAuth 2.0 define cuatro flujos para obtener un token de acceso. A estos flujos se les denomina grand types o tipos de concesión. La utilización de cada uno de estos depende del tipo de aplicación. (Mahendrakumaran, 2021)

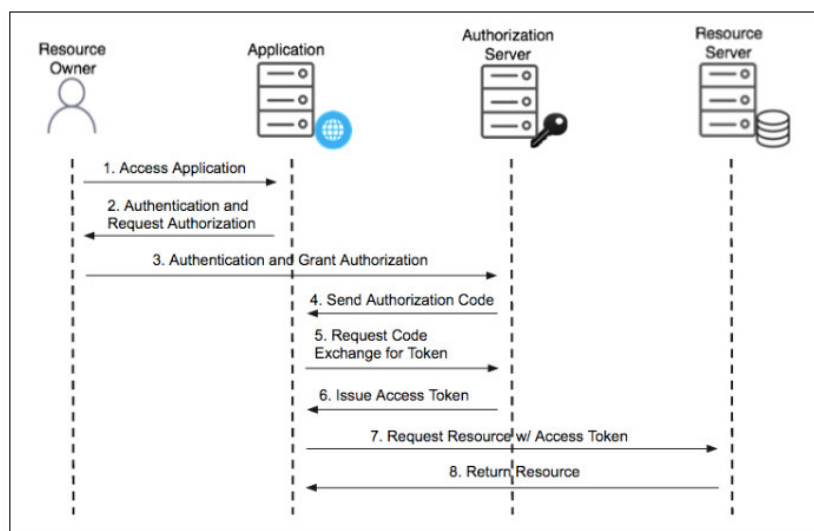
A continuación, se describen los tipos de concesión más importantes:

Authorization code grant o Código de Autorización

Este es el tipo de concesión más completo y seguro, está destinado a lo que se denomina “confidential clients”, que son aplicaciones que tienen la facultad de almacenar secretos de forma segura, en otras palabras, este escenario aplica para aplicaciones que poseen un backend. (Torres, 2019)

Figura 8

Diagrama del flujo Authorization code



Nota. Tomado de www.medium.com, An Introduction to OAuth 2.0.

A continuación, se describen los pasos que debe seguir la aplicación cliente para acceder a los recursos del Resource Owner (Torres, 2019):

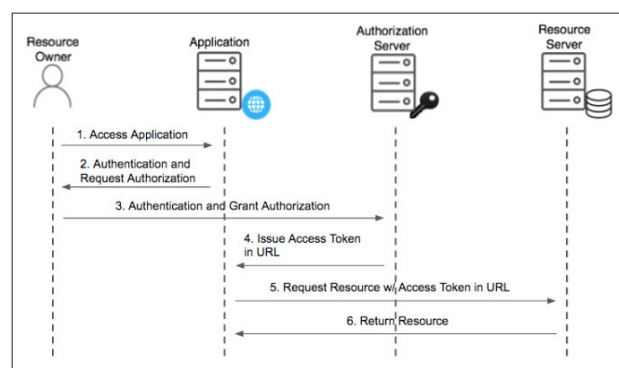
1. El Resource Owner ingresa a la aplicación cliente.
2. La aplicación solicita que el usuario se autentique y lo redirige al Authorization Server.
3. El Resource Owner ingresa sus credenciales y se autentica en el Authorization Server.
4. El Authorization Server redirige al usuario a la aplicación cliente enviando un código de autorización.
5. La aplicación cliente intercambia con el Authorization Server el código de autorización por un access token.
6. La aplicación cliente envía el access token junto con la solicitud del recurso al Resource Server.
7. El Resource Server, le entrega a la aplicación cliente el recurso del Resource Owner.

Implicit Flow

Este tipo de concesión está diseñado para aplicaciones que no pueden almacenar un secreto. Es el caso, por ejemplo, de una aplicación SPA desarrollada en ReactJS, que se comunica con el recurso protegido directamente. Para este caso no existe un código de autorización. (Torres, 2019)

Figura 9

Diagrama del flujo Implicit Flow



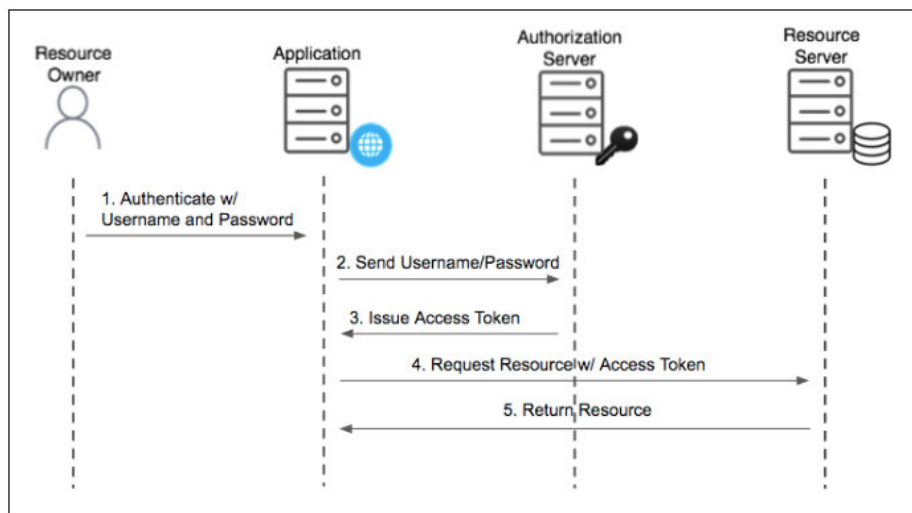
Nota. Tomado de www.medium.com, An Introduction to OAuth 2.0.

Resource owner password

En este tipo de concesión no redirige al usuario al servidor de autorización, sino que la propia aplicación cliente le pedirá al usuario el nombre de usuario y la contraseña del propietario del recurso. Luego la aplicación cliente enviará estas credenciales al servidor de autorización junto con las propias credenciales de la aplicación cliente. (Mahendrakumaran, 2021)

Figura 10

Diagrama del flujo Resource owner password



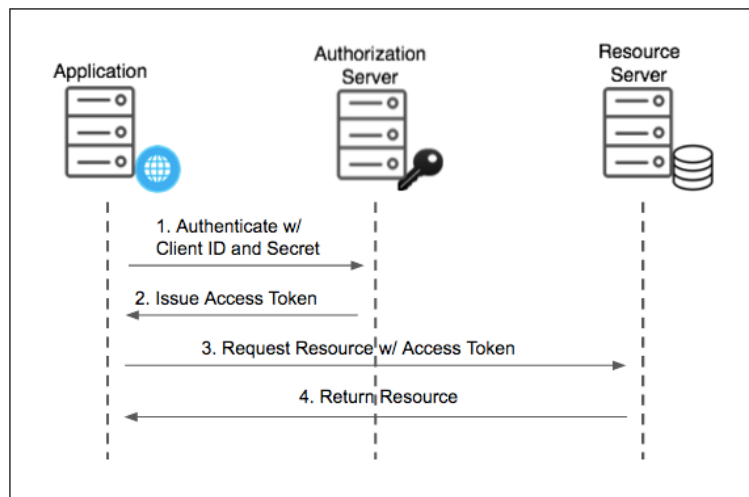
Nota. Tomado de www.medium.com, An Introduction to OAuth 2.0.

Client credentials

Este tipo de concesión está diseñado para las aplicaciones cliente que también cumplen el rol de resource owner y no hay usuarios involucrados en la operación. Es una comunicación máquina a máquina. (Torres, 2019)

Figura 11

Diagrama del flujo Client credentials



Nota. Tomado de www.medium.com, An Introduction to OAuth 2.0

3.2.4.4 OpenID Connect

“OpenID Connect 1.0 es una capa de identidad simple sobre el protocolo OAuth 2.0. Permite a los Clientes verificar la identidad del Usuario final en función de la autenticación realizada por un Servidor de autorización, así como obtener información de perfil básica sobre el Usuario final”. (OpenID, 2014)

OpenID Connect adiciona características que complementan a OAuth (OpenID, 2014) :

- Un mecanismo para identificar al usuario mediante el ID Token.
- Un mecanismo que permite recuperar más información del usuario a través de un punto final denominado UserInfo.
- Atributos denominados “scopes”
- Atributos denominados “claims” o reclamaciones que permiten extraer datos del usuario involucrado en la autenticación.

Esta información está estructurada en formato JSON Web Token, el cual especifica una manera segura de comunicar información entre dos partes utilizando JSON. Debido a que esta digitalmente firmada es posible su verificación. (Torres, 2019)

Hybrid Flow

Gracias a OpenID Connect aparece un nuevo tipo de concesión denominado **Hybrid Flow**. La cual es la fusión entre el flujo “Authorization Code” y el “Implicit Flow”. Este tipo de concesión se practica cuando se requiere que el usuario tenga acceso al token de identidad o Access token de forma inmediata, pero también necesite realizar peticiones al backend para intercambiar un código de autorización por un token de acceso o Access token. (Torres, 2019)

3.2.4.5 Gestión de Identidad y Acceso (IAM)

Un sistema de administración de identidades y accesos (IAM) se define como un conjunto de tecnologías, infraestructura y políticas que permiten a los usuarios administrar sus identidades y controlar su acceso a los recursos corporativos. (Sebastian Castro, 2020)

Los sistemas IAM se puede utilizar para iniciar, capturar, registrar y administrar las identidades de los usuarios y sus correspondientes derechos de acceso de forma automatizada. Esto garantiza que el acceso se conceda de acuerdo con la política para la autenticación, autorización y auditoría adecuadas de todas las personas y servicios. (Sebastian Castro, 2020)

A continuación, se describen las principales ventajas (Sebastian Castro, 2020) :

- **Control de acceso basado en roles o RBAC:** Esto permite que los usuarios accedan solo a aquellos recursos permitidos en base a sus roles dentro de la organización.
- **Autoservicio:** Permiten a los usuarios autogestionar sus identidades digitales, poder recuperar contraseñas, solicitar permisos y más. Esto a su vez ayuda a reducir las tareas administrativas asociadas con el manejo de cuentas de usuario,

registro, cancelación, modificación de asignaciones de permisos dentro de la misma u organización.

- **Automatización:** Este tipo de sistema permite la automatización de todos los procesos relacionados con los usuarios y su acceso, tales como la creación, eliminación y modificación de cuentas de usuario. También permite la aprobación automática de varios permisos, acceso o revocación, asignación de roles y más.
- **Gestión centralizada:** los sistemas IAM centralizan los diversos repositorios de usuarios que pueden existir en una organización en un repositorio de identidad central. Esto facilita la gestión de las cuentas de los usuarios ya que con una sola cuenta se puede acceder a diferentes sistemas informáticos o aplicaciones.

A continuación, se describen las principales desventajas (Sebastian Castro, 2020):

- Aumento de riesgo de seguridad para la organización debido a que el usuario sólo necesita conocer la contraseña para acceder a todos los sistemas y aplicaciones a los que tiene permitido.
- Indisponibilidad de acceso a las aplicaciones dado que el proceso de autenticación y acceso a las diferentes aplicaciones se realiza ante un único repositorio central de identidad.
- Requiere una reorganización sustancial de la definición de roles comerciales y su relación con los roles técnicos, lo que debe mitigar los diferentes accesos y permisos a diferentes aplicaciones y plataformas informáticas.
- La implementación de este tipo de soluciones no es fácil y requiere recursos, reorganización y/o modificación en algunas de las aplicaciones a integrar y procesos de negocio. Los mecanismos de autenticación y autorización del sistema han cambiado mucho y tienen que adaptarse al nuevo sistema. Además, tales soluciones suelen ser propietarias y tienen altos costos.

3.2.4.6 Azure Active Directory

“Azure Active Directory (Azure AD) es un servicio de administración de acceso e identidades basado en la nube. Este servicio ayuda a los empleados a acceder a los recursos externos, como Microsoft 365, Azure Portal y miles de otras aplicaciones SaaS. Azure Active Directory también les ayuda a acceder a recursos internos, como las aplicaciones de la red corporativa de intranet, junto con las aplicaciones en la nube desarrolladas para su propia organización.” (Microsoft, 2022)

Azure Active Directory posee las siguientes características principales que pueden ser implementadas en una organización: (Microsoft, 2022):

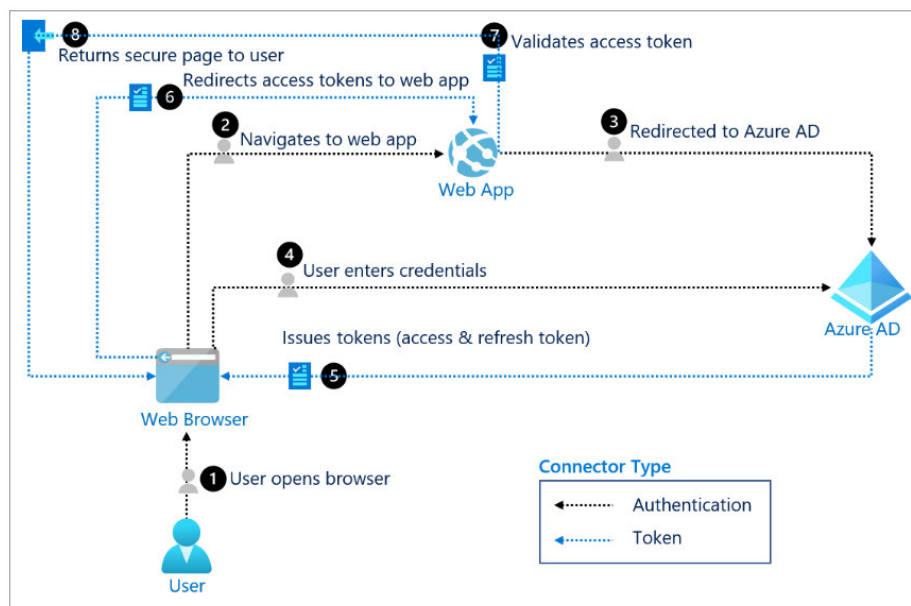
- **Administrar aplicaciones:** Le permite administrar aplicaciones locales en la nube, incluidas aplicaciones SaaS y aplicaciones desarrolladas por la misma organización.
- **Autenticación:** Le permite administrar el restablecimiento de contraseña de Azure Active Directory de autoservicio, la autenticación multifactor, las listas personalizadas de contraseñas prohibidas y el bloqueo inteligente.
- **Administrar dispositivos:** Le permite administrar cómo los dispositivos locales o en la nube acceden a los datos de la empresa
- **Gobierno de identidades:** Le permite administrar la identidad de su organización a través del control de acceso a empleados, socios comerciales, proveedores, servicios y aplicaciones.
- **Protección de identidad:** Le permite detectar posibles vulnerabilidades que afectan la identidad de su organización y configurar políticas para responder a acciones sospechosas

Azure Active Directory como proveedor de identidad de Microsoft utiliza OpenID Connect para la autenticación de los usuarios en las aplicaciones y OAuth para la autorizar los permisos de los usuarios en recursos protegidos.

En la Figura 12 se describe un breve diagrama de secuencia donde interactúan los componentes de OAuth y OpenID Connect con Azure Active Directory. (Microsoft, 2021)

Figura 12

Diagrama de secuencia OpenID Connect y OAuth 2.0 con Azure Active Directory



Nota. Tomado de Microsoft docs, OAuth 2.0 authentication with Azure Active Directory.

Usuario: Solicita un servicio desde la aplicación web (app). El usuario suele ser el propietario del recurso que posee la información y puede permitir que las aplicaciones cliente accedan a su información.

Navegador web: El navegador web con el que interactúa el usuario es el cliente OAuth.

Aplicación cliente: Servidor de recursos o aplicación web, que almacena la información o recursos. Posee una relación de confianza con el servidor de autorización para autenticar y autorizar al cliente de manera segura.

Authorization Server: Azure AD es el servidor de autorización, también llamado proveedor de identidad (IdP). Controla la información del usuario, su acceso y la relación de confianza. Es responsable de emitir los tokens que otorgan y revocan el acceso a los recursos.

3.2.4.7 JWT

“Acrónimo de JSON Web Tokens, es un estándar (RFC 7519) basado en JSON y que podemos definir como un contenedor que almacena toda la información que hace referencia a la autenticación de un usuario y/o intercambio de información.” (Ruiz Mancilla, 2022)

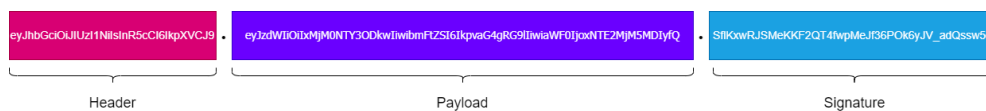
Los JWT son especialmente populares durante la autenticación. Con este estándar, los mensajes cortos se pueden cifrar, brindándoles información sobre el remitente y demostrando que tiene los derechos de acceso requeridos.

Estructura de un JWT

Un JWT consta de tres partes, todas ellas codificadas en Base64 y separadas por un punto:

Figura 13

Estructura de un JWT



Nota. Adaptado de IONOS, Introducción a JSON Web Token (JWT).

- **Header o encabezado**

El encabezado generalmente consta de dos valores y proporciona información importante sobre el token. Contiene el tipo de token y el algoritmo de firma y/o cifrado utilizado. (Ruiz Mancilla, 2022)

Figura 14

Estructura del JWT - Encabezado

```
{ "alg": "HS256", "typ": "JWT" }
```

Nota. Adaptado de IONOS, Introducción a JSON Web Token (JWT).

- **Payload o contenido**

Tiene la información real que se transmitirá a la aplicación, la información se proporciona en forma de pares clave/valor, la clave también se denomina claims del token. (Ruiz Mancilla, 2022)

Figura 15

Estructura del JWT - Contenido

```
{ "sub": "123", "name": "Alicia", "exp": 30 }
```

Nota. Adaptado de IONOS, Introducción a JSON Web Token (JWT).

- **Signature o firma**

Se crea utilizando la codificación base64 de la cabecera, del contenido y el método de firma, para que esta sea eficaz es necesario utilizar una clave que solo es conocida por el servidor de identidad. (Ruiz Mancilla, 2022)

Figura 16

Estructura del JWT - Firma

```
{ 7WK5T79u5mlzjlXXi2ol9Fglmgivv7RAJ7izyj9tUyQ }
```

Nota. Adaptado de IONOS, Introducción a JSON Web Token (JWT).

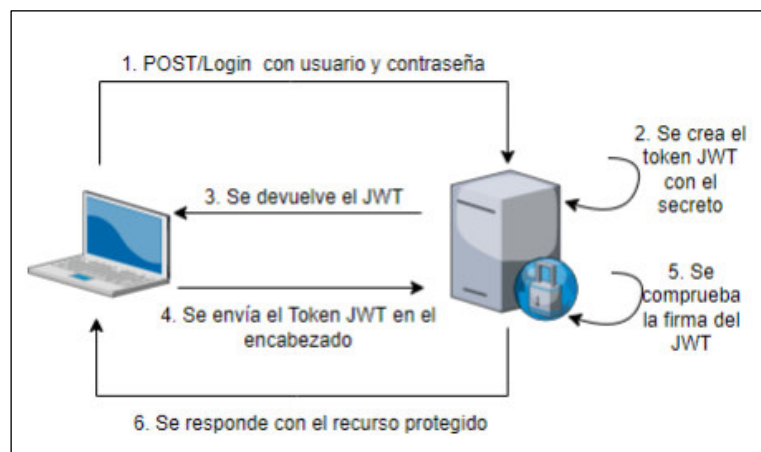
Ciclo de vida de un JWT

Se describe a continuación el ciclo de vida del JWT (Ruiz Mancilla, 2022):

- El proceso se inicia en la aplicación cliente, donde se le solicita al usuario que envíe su usuario y contraseña para iniciar el proceso de autenticación.
- Luego de verificar que los datos sean correctos, se genera un JWT, el cual es generalmente un ID Token o Access Token, que finalmente es enviado al usuario.
- Posterior a ello, la aplicación realizará peticiones de recursos y utilizando el JWT enviado como parte del encabezado.
- Posterior a ello, se verificará en el servidor que el JWT recibido sea seguro, para lo cual se verifica la fecha de expiración, el issuer, el audience y la firma. Adicional a ello en el interior del token se encuentran datos del usuario que realizó la petición.
- Finalmente, y de acuerdo con el mecanismo de control que se haya establecido se responde con el recurso protegido.

Figura 17

Ciclo de vida de un JWT



Nota. Tomado de Ruiz Mancilla, 2022, en el repositorio Cybertesis de la Universidad Nacional Mayor de San Marcos.

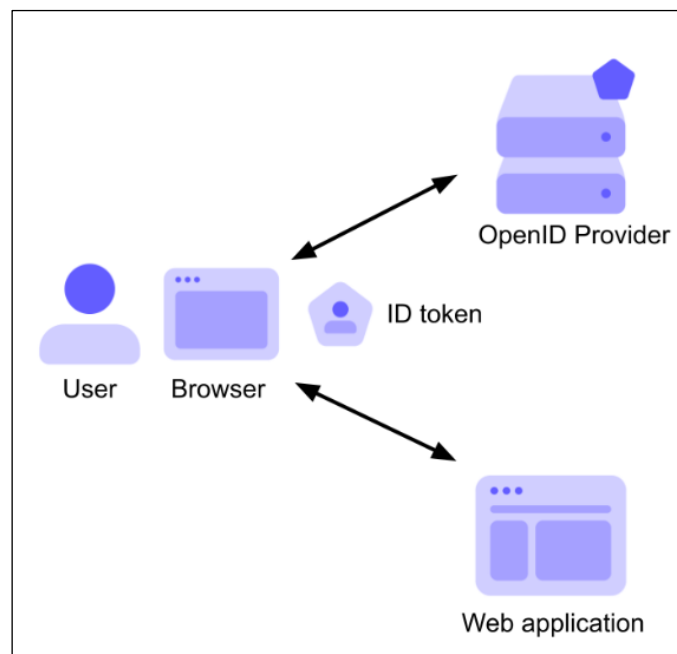
3.2.4.8 Token de Identidad o ID Token

“Un ID Token o Token de acceso es un artefacto que prueba que el usuario ha sido autenticado. Fue presentado por OpenID Connect (OIDC), el cual es un estándar abierto para la autenticación utilizado por muchos proveedores de identidad.” (Chiarelli, 2021)

En el siguiente gráfico, se puede observar que un usuario se autenticará primero en un proveedor de OpenID o proveedor de identidad para obtener un ID token, el cual luego se pasará a la aplicación web como prueba de que el usuario fue exitosamente autenticado.

Figura 18

Diagrama de generación y utilización del ID Token



Nota. Tomado de Auth0, 2021, ID Token and Access Token.

Los tokens de identificación están codificados como JSON Web Tokens (JWT), un formato estándar que permite que su aplicación verifique fácilmente su contenido, se asegure de que provenga del remitente previsto y de que no haya sido alterado.

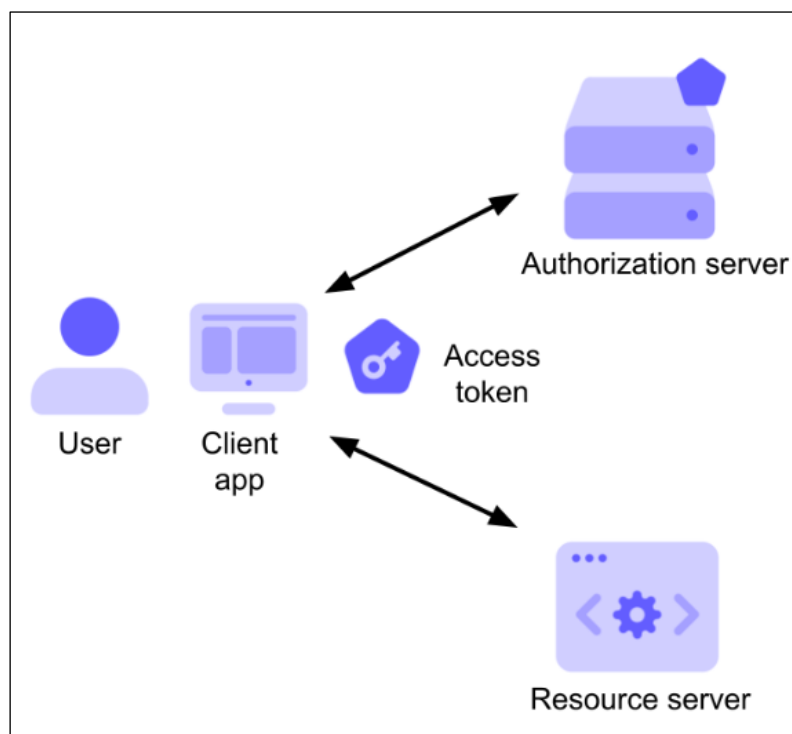
3.2.4.9 Token de acceso o Access Token

“En el contexto de OAuth 2.0, el token de acceso permite que una aplicación cliente acceda a un recurso específico para realizar acciones específicas en nombre del usuario. Eso es lo que se conoce como un escenario de autorización delegada: el usuario delega una aplicación cliente para acceder a un recurso en su nombre.” (Chiarelli, 2021)

En el diagrama de la Figura 19, se puede observar que una aplicación cliente desea acceder a un recurso, como una API u otro recurso que está protegido contra el acceso no autorizado. En este caso, el token de acceso es el artefacto que permite que la aplicación cliente acceda a los recursos del usuario. El token de acceso se emite después de autenticar con éxito al usuario y obtener el consentimiento del usuario.

Figura 19

Diagrama de generación y utilización del Access Token



Nota. Tomado de Auth0, 2021, ID Token and Access Token.

Esto significa, por ejemplo, que puede autorizar su aplicación de LinkedIn para acceder a la API de Twitter en su nombre para realizar publicaciones cruzadas en dos plataformas sociales.

Las especificaciones de OAuth 2 no mencionan el formato que debe tener el token de acceso. Puede ser una cadena en cualquier formato. Un formato común que se usa para los tokens de acceso es JWT. Sin embargo, esto no significa que los tokens de acceso deban tenerlo. (Chiarelli, 2021)

3.2.4.10 Mecanismos de control de acceso

“Hay muchos métodos de seguridad y muchas formas de autenticarse, que pueden depender desde el tipo de dispositivo, el tipo de uso, la confidencialidad de la información, no hay una sola manera de asegurar el API”. (Ruiz Mancilla, 2022)

A continuación, se describen los métodos más utilizados para la autenticación en API Rest (Levin, 2019):

Clave de API o API Key: En este enfoque, a cada usuario se le asigna primero un valor único generado, lo que demuestra que es un usuario confiable. Luego cuando el usuario se autentica nuevamente en el sistema, su clave única (a veces generada por una combinación de su hardware, IP, o veces aleatoriamente por un servidor) se usa para demostrar que es el mismo usuario que antes.

Autenticación Básica o Basic Auth: Actualmente es el menos recomendado debido a sus vulnerabilidades de seguridad inherentes. Con este método el remitente concatena su usuario y contraseña codificado en Base64 en el encabezado de la petición, a esta cadena se le conoce como Basic Token.

Autenticación del portador o Bearer Auth: Esta autenticación se puede traducir como “dar acceso al portador del token”. Los tokens de portador generalmente se emiten a los clientes como resultado de un flujo de autorización de OAuth 2.0, donde se conocen como tokens de acceso. Luego, el cliente usa el token para acceder a un recurso HTTP usando el esquema Bearer definido por RFC 6750. Aunque comúnmente es utilizado en un flujo de OAuth también puede utilizarse fuera de este contexto.

Por ejemplo, supongamos que a una aplicación se le ha emitido la cadena “SLAV32hkKG” como token. Si la aplicación desea utilizar este token para obtener información sobre el usuario, debe mandar en el encabezado tal como se muestra en la Figura 20.

Figura 20

Petición usando Token Bearer

```
GET /userinfo HTTP/1.1
Host: api.example.com
Authorization: Bearer SLAV32hkKG
```

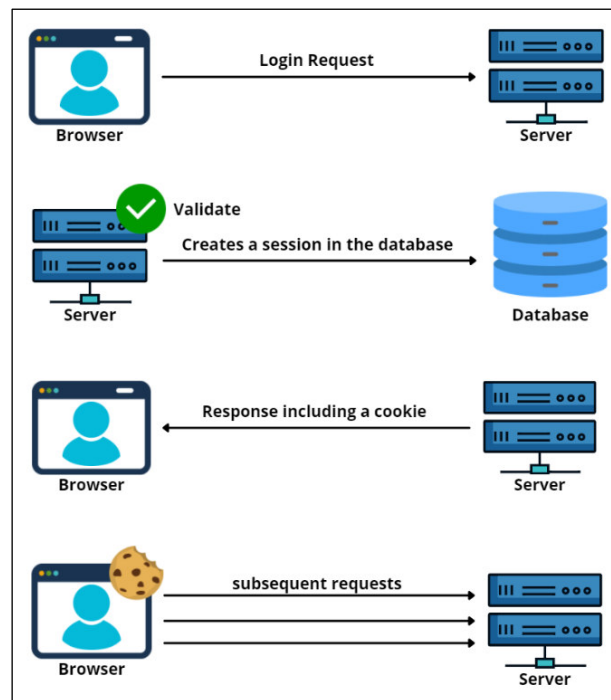
Nota. Fuente: Elaboración propia.

Autenticación basada en Cookies: Esta autenticación consiste en intercambiar las credenciales del usuario por una cadena única generada por el servidor y enviada al navegador. En este tipo de autenticación cuando inicia sesión en una aplicación web, el navegador recibirá una cookie del servidor de su aplicación, y el navegador almacenará y enviará esa cookie con cada solicitud posterior para verificar que las solicitudes provengan del mismo usuario.

En la siguiente Figura se ilustran los pasos para realizar la autenticación basada en cookies.

Figura 21

Flujo de autenticación basada en cookie



Nota. Fuente: Elaboración propia.

3.2.4.11 SCRUM

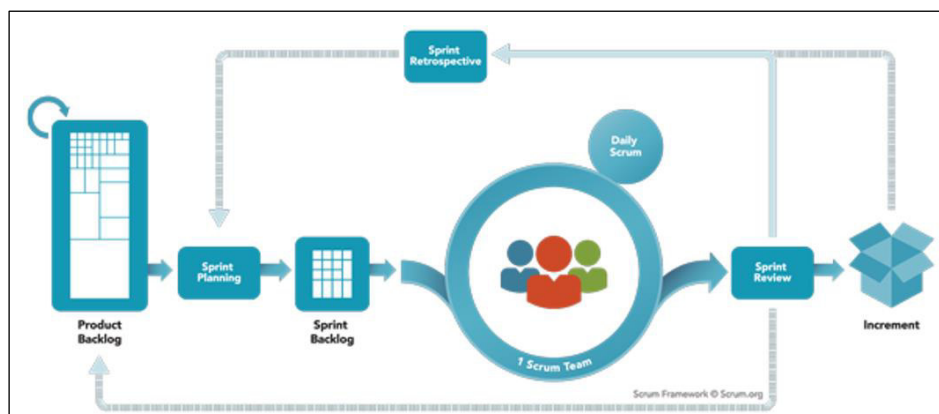
“Scrum es un marco ligero que ayuda a las personas, los equipos y las organizaciones a generar valor a través de soluciones adaptables para problemas complejos.” (Scrum, 2021)

Este marco de forma principal se caracteriza por desarrollos de corta duración, entregando pequeñas funcionalidades que contribuyen al desarrollo de un producto final

En la Figura 22 se observa el ciclo de vida de un sprint, en el cual interactúan roles, artefactos y ceremonias para crear un producto.

Figura 22

Ciclo de vida de Scrum



Nota. Tomado de Scrum Guide, 2021, Scrum org.

En la siguiente tabla se describen los roles que participan en el marco de trabajo Scrum.

Tabla 7

Roles de Scrum

Roles	Descripción
Product Owner	Su principal función es gestionar el Product Backlog, especificando los requisitos a partir de las historias de usuario, asimismo priorizar las historias que se abordaran en cada sprint.
Scrum Máster	Encargado de dotar al equipo Scrum de los recursos necesarios para evitar problemas a la hora de desarrollar las historia de usuario
Scrum Team	Equipo encargado de construir un producto terminado a partir de historias de usuario
Stakeholder	Persona o grupo de persona con interés directo o indirecto en la construcción del producto final

Nota. Adaptado de Ayra Olano, 2021, en el repositorio Cybertesis de la Universidad Nacional Mayor de San Marcos.

En la siguiente tabla se describen las ceremonias que conforman el marco de trabajo Scrum.

Tabla 8

Ceremonias de Scrum

Ceremonias	Descripción
Sprint Planning	En esta fase se planean las historias de usuario a resolver durante el Sprint, las cuales son priorizadas por el Product Owner.
Daily Scrum	Reunión diaria de seguimiento de las tareas trabajadas durante el sprint. Se identifican impedimentos que pueden detener el desarrollo normal de la iteración
Sprint Review	Se refiere a la reunión al final de cada Sprint que sirve para explicar a las partes interesadas el progreso y objetivos alcanzados en todo el Sprint.
Sprint Retrospective	Ceremonia en la cual el equipo comenta su percepción sobre el sprint finalizado, con el fin de identificar áreas de mejora. De esta manera se promueve la mejora continua.

Nota. Adaptado de Ayras Olano, 2021, en el repositorio Cybertesis de la Universidad Nacional Mayor de San Marcos.

En la siguiente tabla se describen los artefactos que conforman el marco de trabajo Scrum.

Tabla 9

Artefactos de Scrum

Artefactos	Descripción
Product Backlog	Lista de tareas denominadas historias de usuario, especifican lo que debe lograrse para cumplir con los objetivos del proyecto
Sprint Backlog	Lista de historias de usuario del Product Backlog, que se resuelven durante el sprint

Nota. Adaptado de Ayra Olano, 2021, en el repositorio Cybertesis de la Universidad Nacional Mayor de San Marcos.

3.2.4.12 Microservicios

Los microservicios son un enfoque arquitectónico para el desarrollo de software, que consta de pequeños servicios independientes que se comunican a través de APIs. La arquitectura de microservicios hace que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite acelerar el tiempo de creación y despliegue de nuevas funciones. (AWS, 2022)

A continuación, se describe una breve comparación entre una arquitectura monolítica y una arquitectura de microservicios. (AWS, 2022):

Con una arquitectura monolítica, los procesos se encuentran demasiado acoplados y se exponen a través de un solo servicio. Es decir, si un proceso recibe un aumento de demanda, se debe escalar toda la arquitectura. Adicional a esto, el esfuerzo para mantener la aplicación se vuelve más tedioso a medida que crece el código de la aplicación.

Con una arquitectura de microservicios, las aplicaciones se crean a partir de componentes separados que exponen cada proceso de la aplicación como un servicio. Los

cuales se comunican a través de interfaces llamadas APIs. Cada servicio expone una función acotada y trabajan de forma independiente del resto.

3.2.4.13 Metodología FAIR

“Es una metodología para cuantificar y gestionar el riesgo en cualquier organización. Proporciona un modelo para comprender, analizar y cuantificar el riesgo cibernético en términos financieros”. (Sealpath, 2022).

Esta metodología propone una clasificación de diferentes elementos que conforman el riesgo cibernético con el fin estimar la pérdida económica en una organización. Esta clasificación está dividida en 2 ramas (Sealpath, 2022):

“Frecuencia de eventos de pérdida” (LEF) Se refiere a la frecuencia, en la que una amenaza puede infligir daño a un activo. Para ello considera dos puntos importantes:

- **“Frecuencia de Eventos de Amenaza” (TEF)** Frecuencia en la que una amenaza tiene éxito o no sobre un activo.
- **“Vulnerabilidad del recurso” (Vuln – Vulnerability)** Probabilidad en que una amenaza materialice una pérdida económica.

“Magnitud de la pérdida” (LM) Establece dos tipos de pérdida:

- **“Pérdida Primaria” (Primary Loss):** Hace referencia a las pérdida directas debido al efecto que causa la amenaza sobre el activo.
- **“Pérdida Secundaria” (Secondary Loss):** Son pérdidas que ocurren a las partes interesadas secundarias.

Para la aplicación de la metodología se debe seguir 4 fases o etapas (Sealpath, 2022):

Etapas 1: Identificar los componentes del escenario, para ello se debe considerar lo siguiente:

- Identificación del activo en riesgo.
- Identificar la comunidad de amenazas.
- Definir el evento de pérdida.

Etapla 2: “Evaluación de la Frecuencia de Eventos de Amenaza” (LEF), para lo cual se consideran las siguientes estimaciones:

- Estimación de la Frecuencia de Eventos de Amenaza (TEF).
- Estimación de la Capacidad de Amenaza (TCAP).
- Estimación de la Fuerza de Resistencia (RS).
- Definir la Vulnerabilidad (Vuln).
- Definir la Frecuencia de Pérdida (LEF).

Etapla 3: “Evaluación de la Magnitud de Pérdida” (LM).

Para elaborar la magnitud de la pérdida la metodología brinda la siguiente clasificación correspondiente a las pérdidas económicas:

Figura 23

Magnitudes Económicas según metodología FAIR

Magnitud	Rango más bajo	Rango más alto
Severa (SV)	\$10.000.000	-
Alta (A)	\$1.000.000	\$9.999.999
Significativa (S)	\$100.000	\$999.999
Moderada (M)	\$10.000	\$99.999
Baja (B)	\$1.000	\$9.999
Muy Baja (MB)	\$0	\$999

Nota. Tomado de Sealpath, 2022.

Las cuales se toman como referencia para evaluar la magnitud de los siguientes escenarios de pérdida.

Tabla 10

Formas de pérdida según metodología FAIR

Forma de Pérdida	Descripción
Productividad	Asociado al costo por la pérdida de la productividad
Respuesta	Asociado al costo por la respuesta ante el incidente
Reemplazo	Asociado al costo por el reemplazo del activo
Multas/Juicios	Asociado al costo por sanciones
Venta comprometida	Asociado al costo por la pérdida de la venta
Reputación	Asociado al costo por la pérdida reputacional

Nota. Tomado de Sealpath, 2022.

Esta evaluación debe realizarse para las pérdidas primarias y secundarias.

Etapas 4: Derivar y Articular el Riesgo

Con todas las estimaciones realizadas en las etapas anteriores y en base a las pérdidas económicas estimadas primarias y secundarias se deriva el riesgo global y por ende la magnitud económica global.

3.2.4.14 OWASP (Open Web Application Security Project)

“El Open Web Application Security Project (OWASP) es una fundación sin fines de lucro dedicada a mejorar la seguridad del software”. (OWASP, 2021)

Mediante software de código abierto escritos en su mayoría por la comunidad, numerosos equipos locales en todo el mundo y educación a través de conferencias, talleres y documentos, la organización OWASP es una fuente para que los profesionales de la tecnología y desarrolladores en todo el mundo protejan sus aplicaciones. (OWASP, 2021)

El OWASP Top 10 es un documento de referencia que describe los 10 problemas de seguridad más críticos para la seguridad de las aplicaciones web. El informe es elaborado por un equipo de expertos en seguridad de todo el mundo, con datos de varias organizaciones. (OWASP, 2021)

A continuación, se describe brevemente el ranking de vulnerabilidades web de OWASP, según la última publicación del documento en el 2021 (OWASP, 2021):

- **Control de acceso roto (A01:2021):** Se refiere a la debilidad en un sistema que permite a un atacante obtener acceso a la cuenta de otro usuario o incluso elevar sus niveles de privilegio y obtener acceso a la cuenta de administrador.
- **Fallas criptográficas (A02:2021):** Las fallas criptográficas ocurren cuando se comprometen datos importantes almacenados o transmitidos (por ejemplo, datos de tarjeta de crédito).
- **Inyección (A03:2021):** Una inyección de código ocurre cuando un atacante envía datos no válidos a una aplicación web para hacer que la aplicación haga algo para lo que no fue diseñada.
- **Diseño Inseguro (A04:2021):** Se refiere a los riesgos relacionados con el diseño y las fallas arquitectónicas que pueden tener una aplicación en su etapa inicial.
- **Configuración incorrecta de seguridad (A05:2021):** Una mala configuración de seguridad es una falla de diseño o configuración que resulta de una mala configuración o un defecto en los elementos de infraestructura que soportan una aplicación.

- **Componentes vulnerables y obsoletos (A06:2021):** Esta categoría se relaciona con componentes o bibliotecas de terceros que plantean riesgos de seguridad conocidos y potenciales.
- **Fallas de Identificación y Autenticación (A07:2021):** Esta vulnerabilidad permite a los atacantes comprometer contraseñas y sesiones, lo que puede conducir al robo de la identidad del usuario.
- **Fallas de integridad de software y datos (A08:2021):** Esta categoría para 2021 está enfocada en actualizaciones de software, datos críticos y canalizaciones de CI/CD utilizadas sin verificar la integridad.
- **Registro de seguridad y fallas de monitoreo (A09:2021):** Esta categoría se refiere al escaso o nulo registro y monitoreo que se realiza en las aplicaciones, que puede conllevar a que acciones potencialmente peligrosas no se detecten a tiempo y por ende causar graves daños en las aplicaciones.
- **Falsificación de solicitud del lado del servidor (A10:2021):** Esta vulnerabilidad puede ocurrir si se obtiene un recurso de forma remota a través de una URL sin validar proporcionada por el usuario. Esto permite a los atacantes hacer que las aplicaciones envíen solicitudes manipuladas a destinos inesperados, incluso si el sistema está protegido por firewalls, VPN u otras listas de control de acceso a la red.

3.2.5 Implementación de las áreas, procesos, sistemas y buenas prácticas

A continuación, se describen las 8 etapas del proyecto.

3.2.5.1 Análisis de aplicaciones BackOffice

Durante esta primera etapa del proyecto se realizó el análisis de las 7 aplicaciones BackOffice críticas para la Empresa Aseguradora. Para este fin se desarrolló una serie de actividades que permitió al equipo conocer a profundidad estas aplicaciones.

Revisión de documentación: Se solicitó a los equipos responsables del proyecto los diagramas de arquitectura, diagrama de flujo, modelo de base de datos, etc. Sin embargo, la proporcionada era muy escasa o nula.

Reunión con los equipos: El equipo del proyecto se reunió con los equipos responsables de las aplicaciones BackOffice con el fin de entender el mecanismo de autenticación, la tecnología y el lenguaje utilizados.

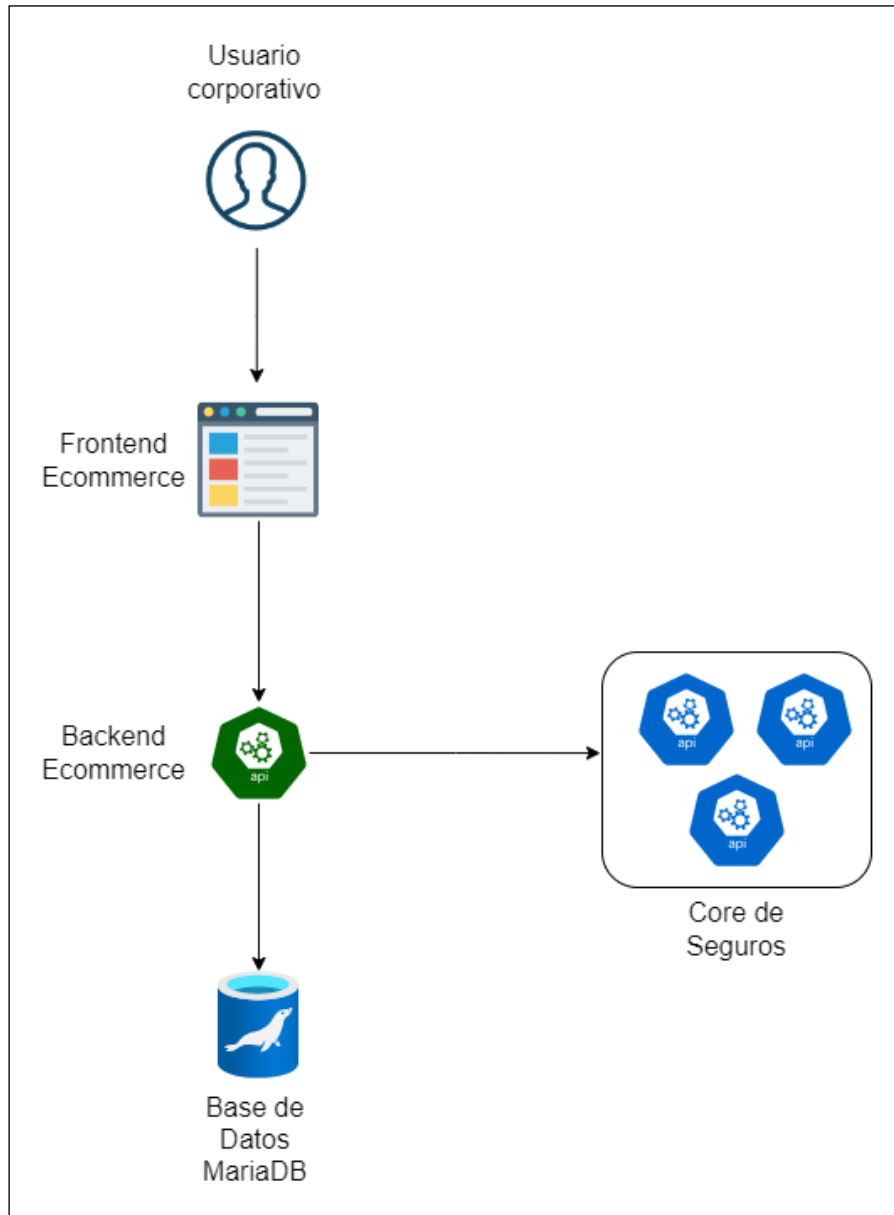
Revisión de código: El equipo del proyecto realizó una revisión detallada de los repositorios de códigos de las aplicaciones BackOffice, tanto del frontend como el backend.

En las 5 aplicaciones BackOffice de Vida, Viajes, Hogar, Autos y Canales se identificaron que utilizaban la misma arquitectura por lo que en adelante se referirá a estas como BackOffice Ecommerce. Estas tienen por objetivo, crear, actualizar y eliminar parámetros de la plataforma online para la venta de seguros de Vida, Viajes, Hogar y Autos. Adicional a ello pueden gestionar campañas de seguros para clientes corporativos.

En la Figura 24 se muestra la arquitectura general de las aplicaciones BackOffice Ecommerce.

Figura 24

Diagrama general de la arquitectura de las aplicaciones BackOffice Ecommerce



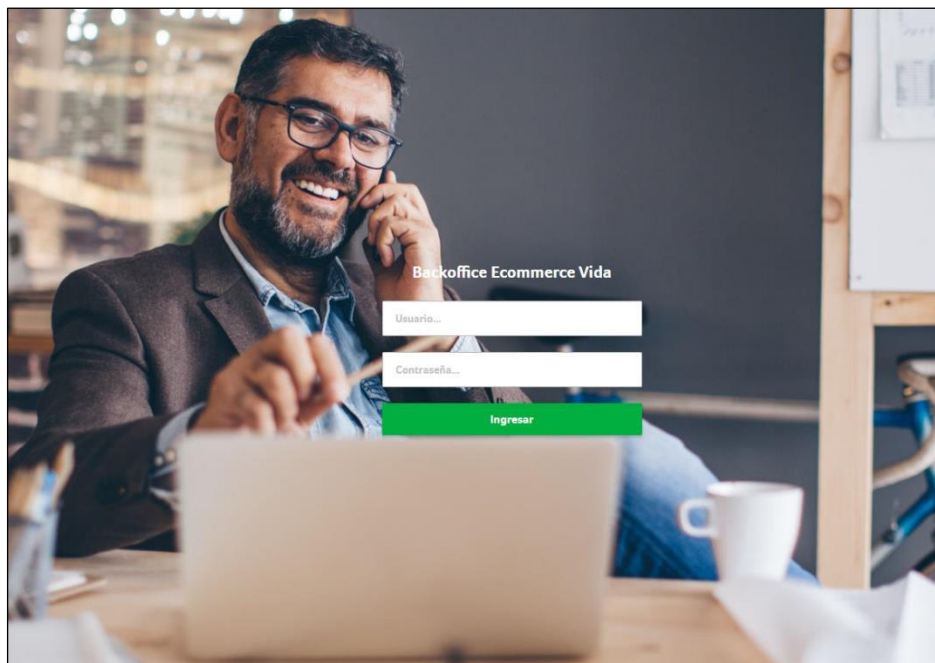
Nota. Adaptado de la Empresa Aseguradora.

El alcance de la revisión solo tomaron en cuenta los componentes de Autenticación y Autorización de los sistemas BackOffice Ecommerce, que se describen a continuación:

- Autenticación: Los 5 sistemas BackOffice Ecommerce realizan la autenticación del usuario bajo el mecanismo de usuario y contraseña. El procedimiento para registrar el usuario es mediante solicitud directa al área técnica responsable de la aplicación, la cual realizará manualmente la generación de las claves e inserción en la base de datos del sistema.

Figura 25

Pantalla de inicio – BackOffice Ecommerce Vida

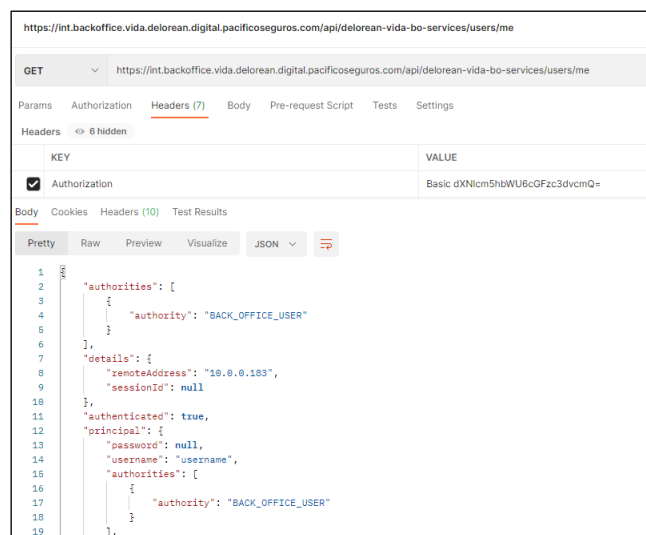


Nota. Adaptado de la Empresa Aseguradora.

- Autorización: La autorización de usuarios se da a través de la API mostrada en la Figura 26, la cual devuelve información sobre el usuario y su rol en la aplicación. Esta API se consume justo después de la autenticación del usuario.

Figura 26

Petición y respuesta del servicio de autorización – BackOffice Vida

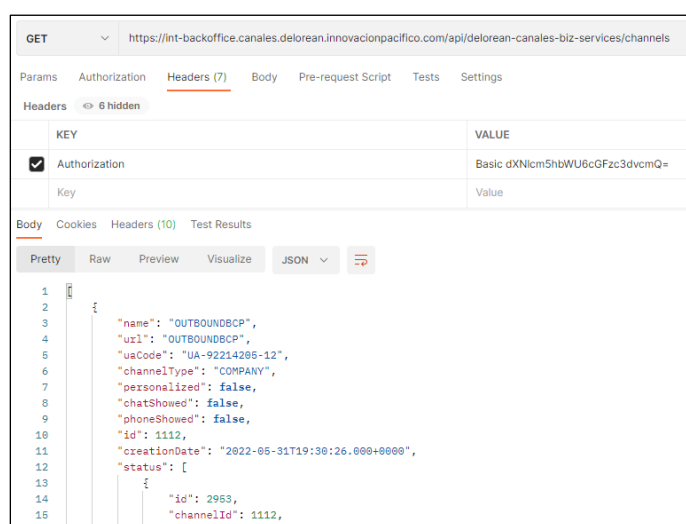


Nota. Adaptado de la Empresa Aseguradora.

- El control de acceso para las APIs utiliza la autenticación básica, como se muestra en la Figura 27 en cada solicitud se envía el usuario y contraseña codificado en Base64 en la cabecera “Authorization”.

Figura 27

Ejemplo de autenticación básica – BackOffice Canales



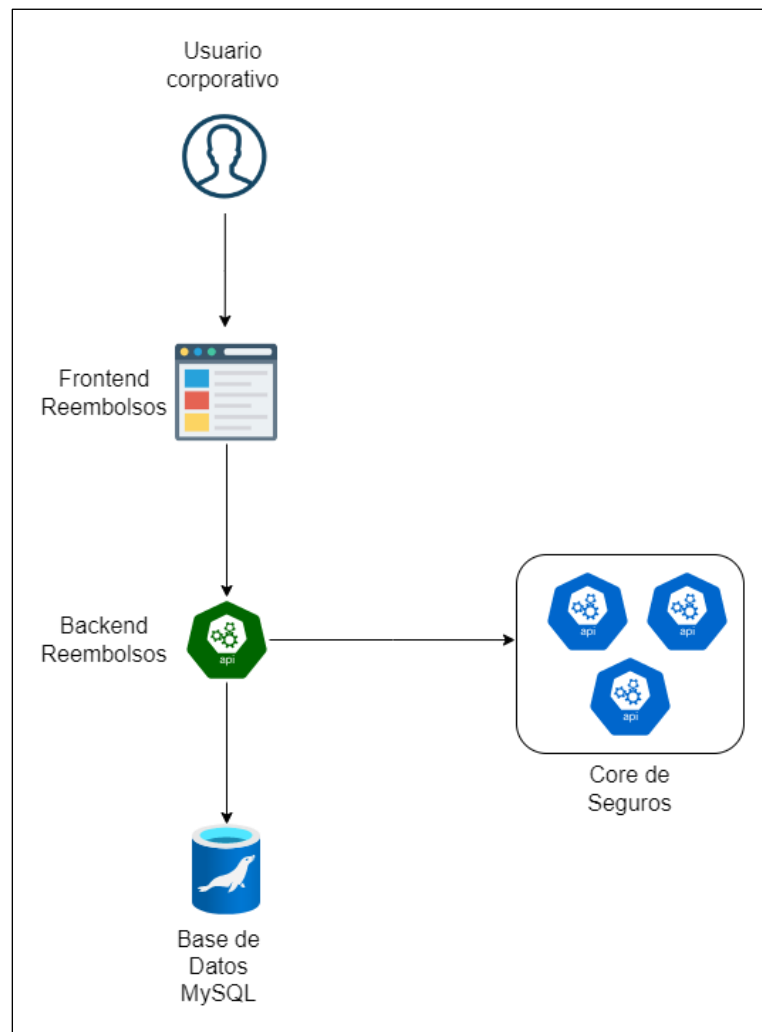
Nota. Adaptado de la Empresa Aseguradora.

La aplicación BackOffice de Reembolsos tiene por objetivo, gestionar el trámite de reembolso solicitado por el cliente, validar la solicitud del cliente y registrar la devolución del dinero gastado por el cliente en la atención médica en un centro de salud particular.

En la Figura 28 se muestra la arquitectura general de la aplicación BackOffice Reembolsos.

Figura 28

Diagrama general de la arquitectura de la aplicación BackOffice Reembolsos



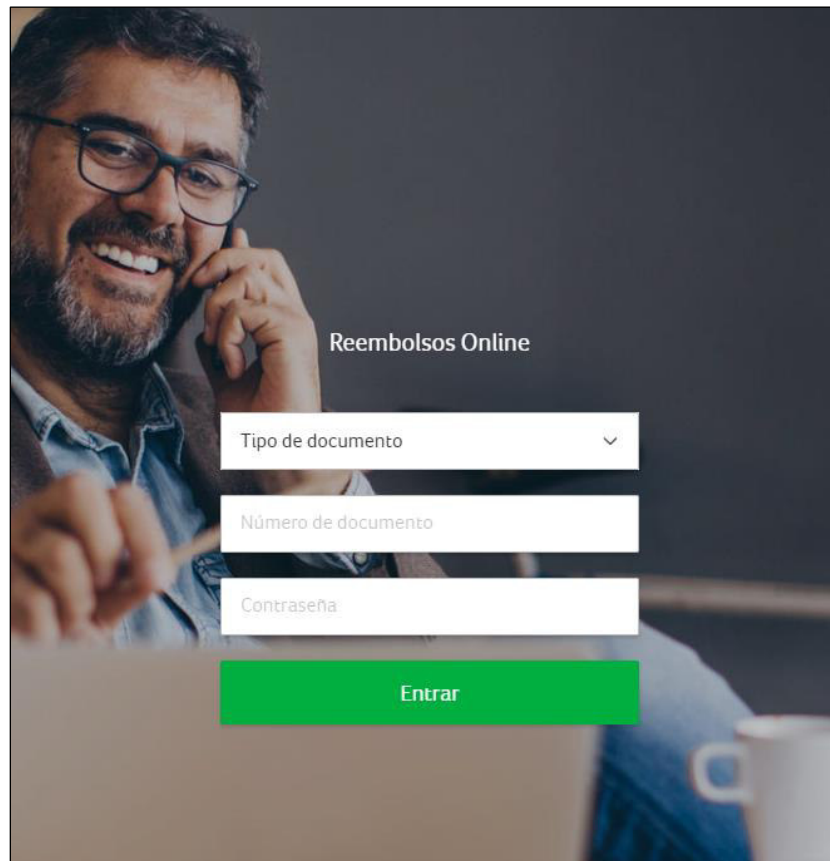
Nota. Adaptado de la Empresa Aseguradora.

El alcance de la revisión solo tomaron en cuenta los componentes de Autenticación y Autorización del sistema BackOffice de Reembolsos, que se describen a continuación:

- Autenticación: La aplicación BackOffice Reembolsos realiza la autenticación del usuario bajo el mecanismo de usuario y contraseña. El procedimiento para crear registrar el usuario es mediante solicitud directa al área técnica responsable de la aplicación, la cual realizará manualmente la generación de las claves e inserción en la base de datos del sistema. Una vez el usuario es autenticado exitosamente, se genera un token que se utilizará para el control de acceso a las APIs.

Figura 29

Pantalla de inicio – BackOffice Reembolsos

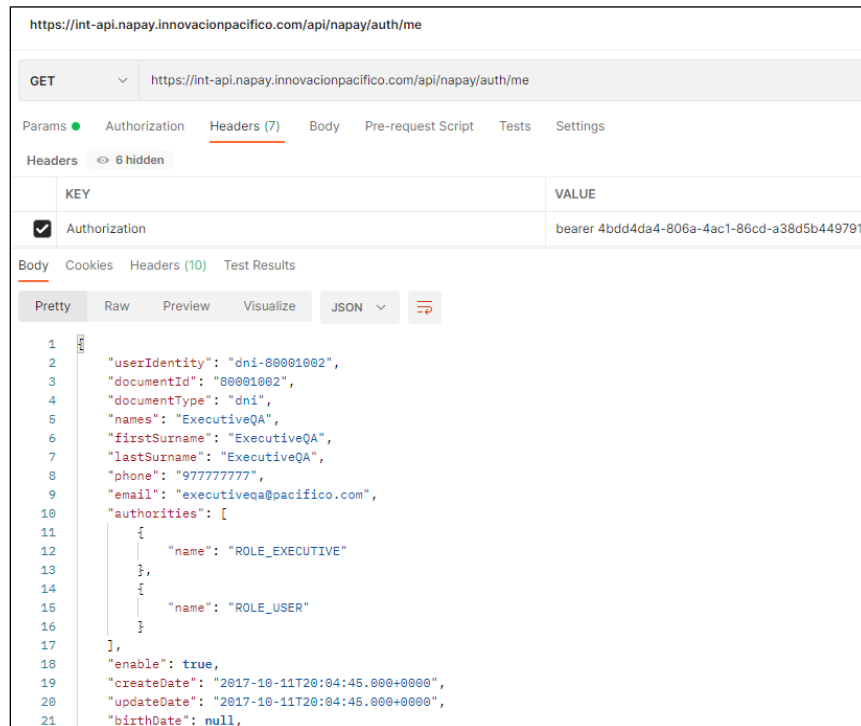


Nota. Adaptado de la Empresa Aseguradora.

- Autorización: La autorización de usuarios se da a través de la API mostrada en la Figura 30, la cual devuelve información sobre el usuario y su rol en la aplicación. Esta API se consume justo después de la autenticación del usuario.

Figura 30

Petición y respuesta del servicio de autorización – BackOffice Vida

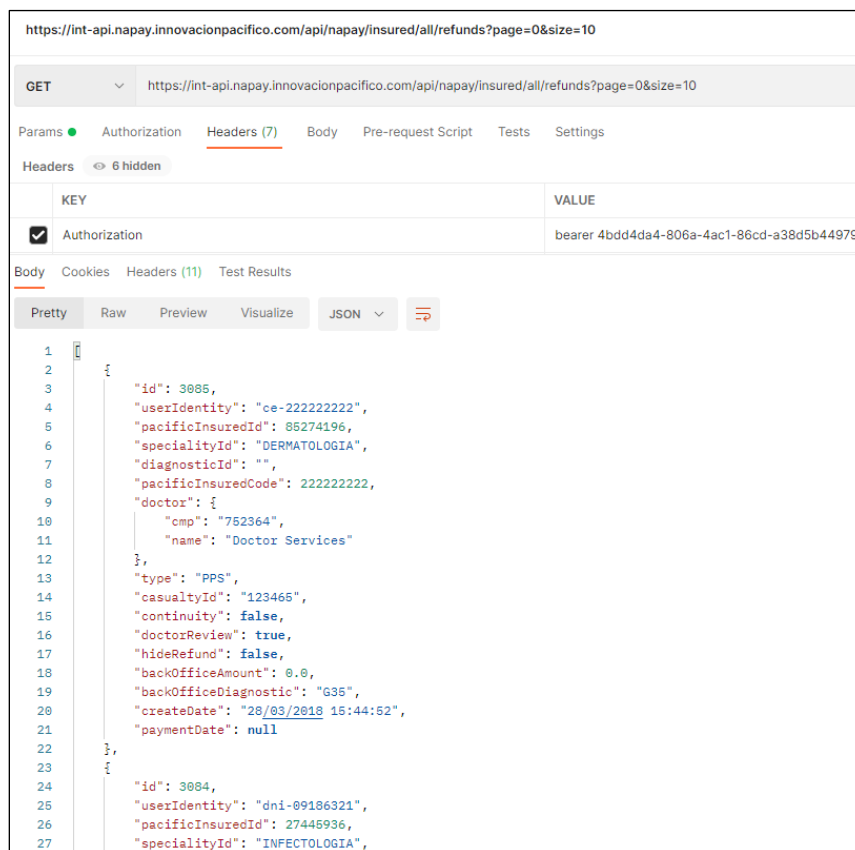


Nota. Adaptado de la Empresa Aseguradora.

- El control de acceso para las APIs utiliza la autenticación del portador o bearer, como se muestra en la Figura 31 en cada solicitud se envía el token bearer en la cabecera Authorization.

Figura 31

Ejemplo de autenticación bearer – BackOffice Reembolsos

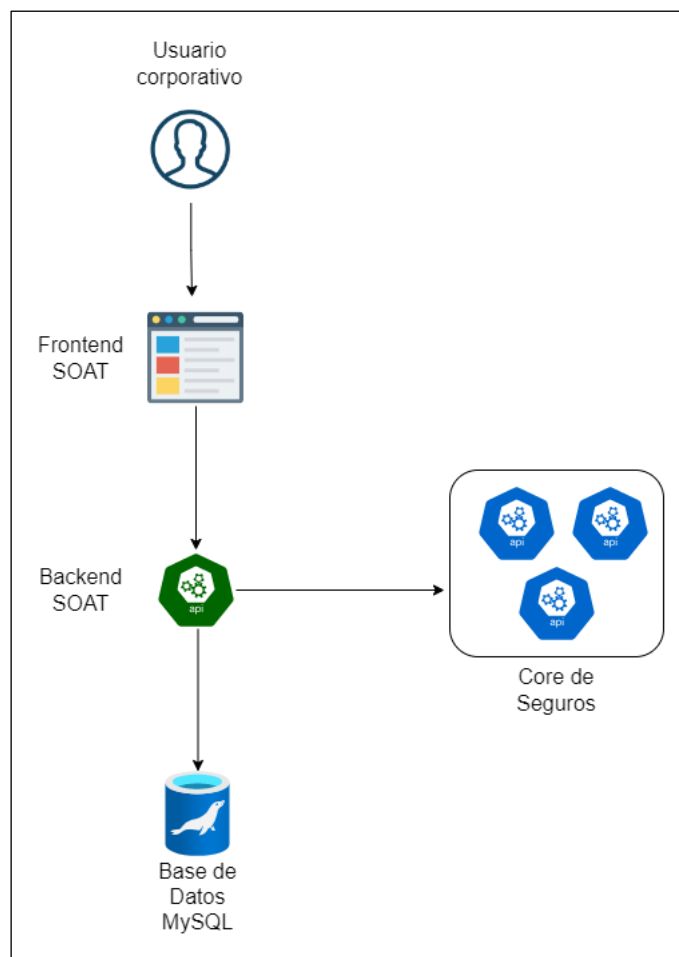


Nota. Adaptado de la Empresa Aseguradora.

La aplicación BackOffice de SOAT tiene por objetivo gestionar los procesos relacionados al seguro SOAT. Entre las cuales se incluyen las siguientes funcionalidades: Gestionar la información mostrada en los canales de venta, agregar códigos de descuento y promociones, buscar vehículo en registros públicos y gestionar la información del cliente.

Figura 32

Diagrama general de la arquitectura de la aplicación BackOffice SOAT



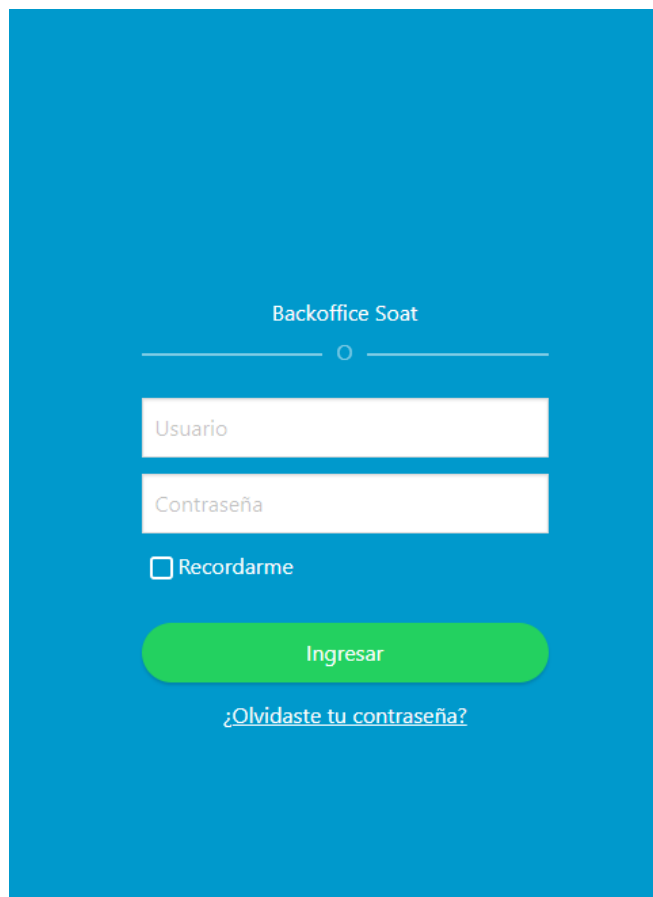
Nota. Adaptado de la Empresa Aseguradora.

El alcance de la revisión solo tomaron en cuenta los componentes de Autenticación y Autorización del sistema BackOffice de SOAT, que se describen a continuación:

- Autenticación: La aplicación BackOffice SOAT realiza la autenticación del usuario bajo el mecanismo de usuario y contraseña. El procedimiento para crear registrar el usuario es mediante una solicitud al jefe directo del área, el cual lo realiza mediante la misma aplicación BackOffice.

Figura 33

Pantalla de inicio – BackOffice SOAT

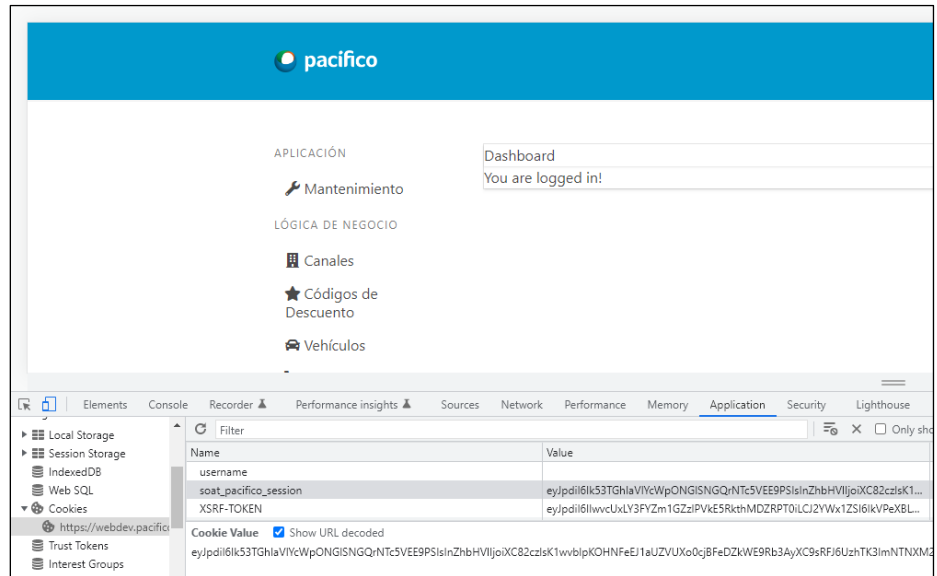
The image shows a login interface for 'Backoffice Soat' on a solid blue background. At the top, the text 'Backoffice Soat' is centered in white. Below it is a horizontal line with a small white circle in the middle. Underneath the line are two white input fields: the first is labeled 'Usuario' and the second is labeled 'Contraseña'. Below these fields is a checkbox labeled 'Recordarme'. A large, rounded green button with the text 'Ingresar' in white is positioned below the checkbox. At the bottom, there is a link that reads '¿Olvidaste tu contraseña?' in white.

Nota. Adaptado de la Empresa Aseguradora.

Posterior a la autenticación exitosa, se genera una cookie de sesión, la cual debe ser enviada en todas las peticiones posteriores y que es verificada por el servidor.

Figura 34

Petición con Cookie de sesión – BackOffice SOAT



Nota. Adaptado de la Empresa Aseguradora.

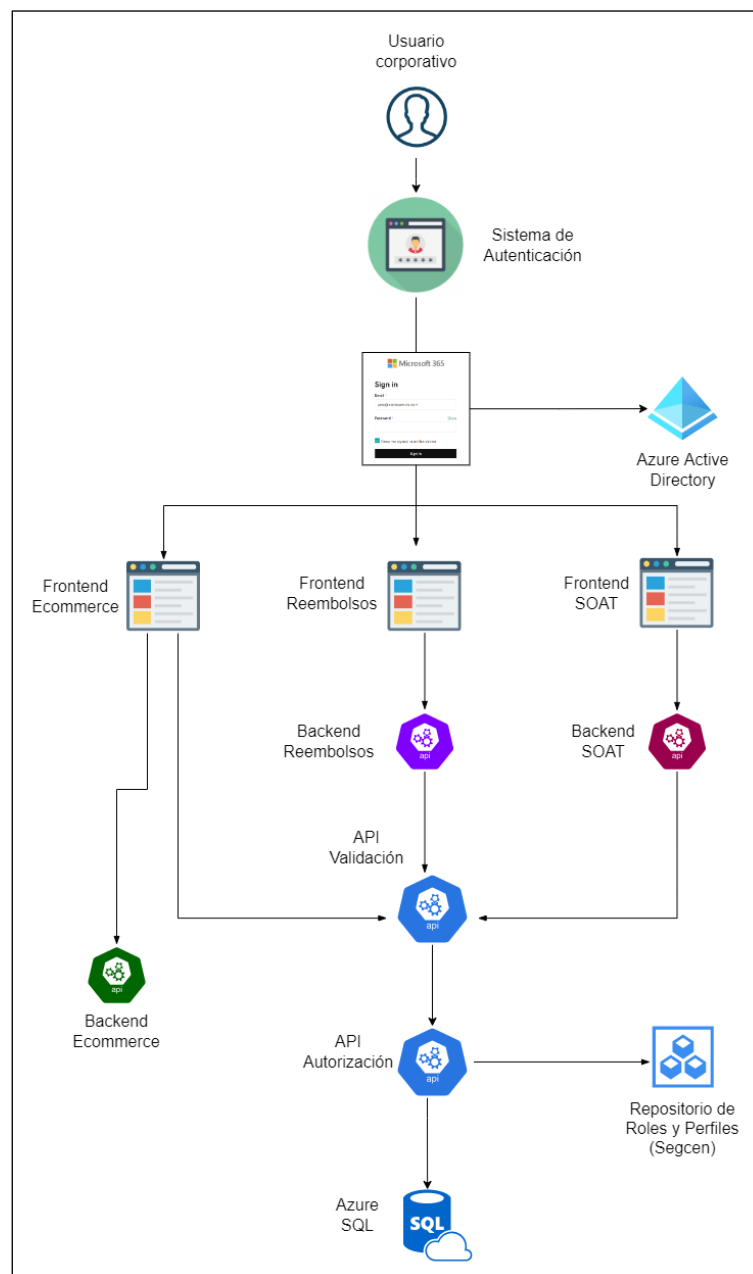
- Autorización: El rol del usuario es proporcionado en el momento en que el usuario es autenticado correctamente y de acuerdo con dicho rol se muestran las opciones a las que tendrá acceso el usuario.

3.2.5.2 Diseño de la arquitectura propuesta

A continuación, se detallan los diagramas de solución y secuencia que se elaboraron después del análisis de las aplicaciones BackOffice.

Figura 35

Diagrama de Solución propuesta



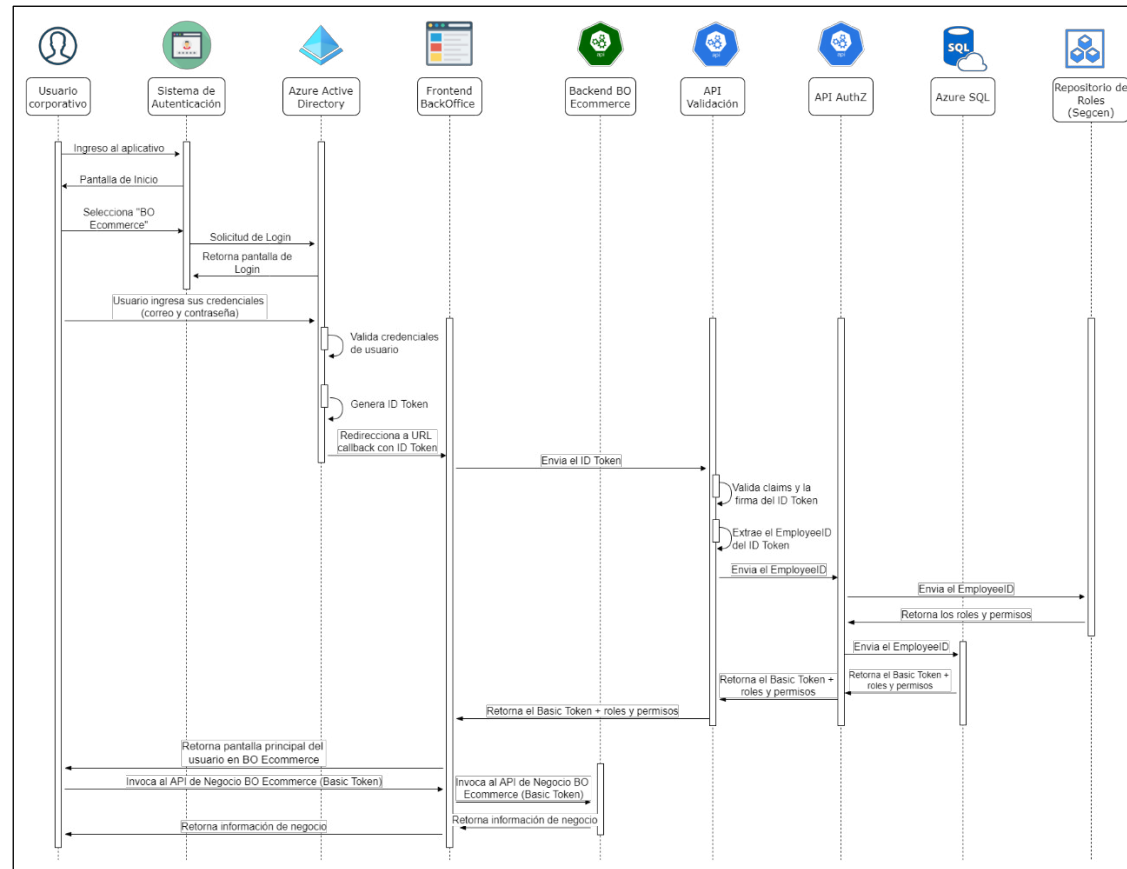
Nota. Adaptado de la Empresa Aseguradora.

En las Figuras 36, 37 y 38 se muestra los diagramas de secuencias de la solución, del cual podemos mencionar que los dos elementos importantes que se utilizaron fueron Azure Active Directory y el repositorio de roles y permisos de la Empresa Aseguradora. Los que han permitido centralizar la autenticación y autorización de los usuarios y disminuir el nivel de riesgo de las aplicaciones BackOffice.

A continuación, se muestra el diagrama de secuencia de los 5 sistemas BackOffice de Ecommerce (Vida, Viajes, Autos, Hogar y Canales) los cuales se representan mediante un solo diagrama debido a la similitud de sus arquitecturas.

Figura 36

Diagrama de secuencia - BackOffice Ecommerce (Vida, Viajes, Autos, Hogar y Canales)

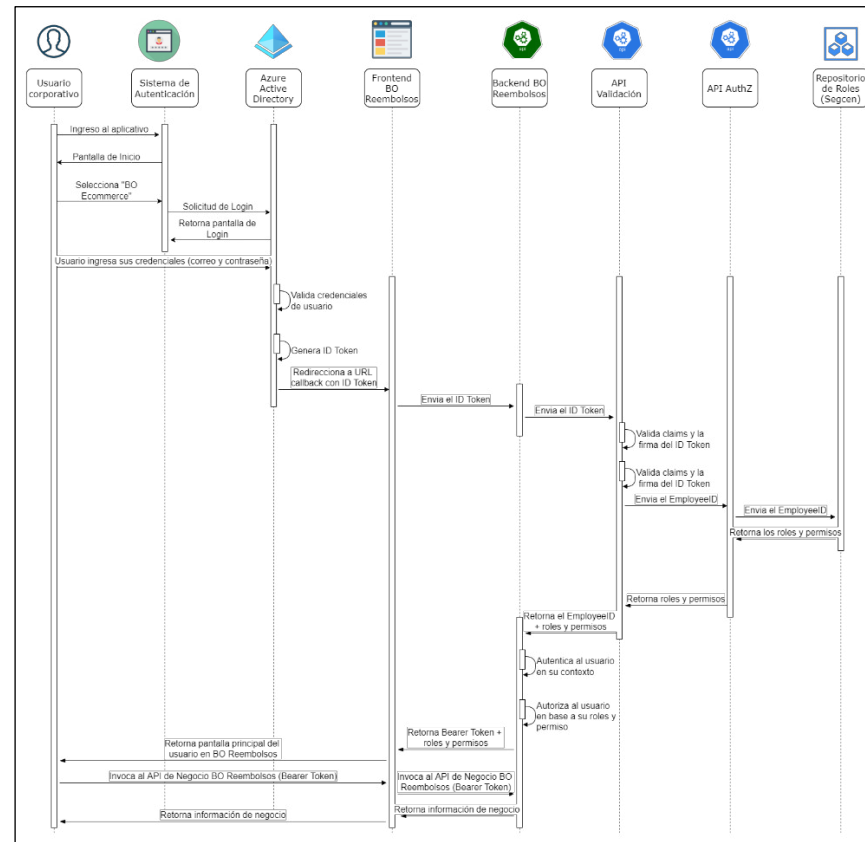


Nota. Adaptado de la Empresa Asegurador.

A continuación, se muestra el diagrama de secuencia del BackOffice Reembolsos.

Figura 37

Diagrama de secuencia - BackOffice Reembolsos

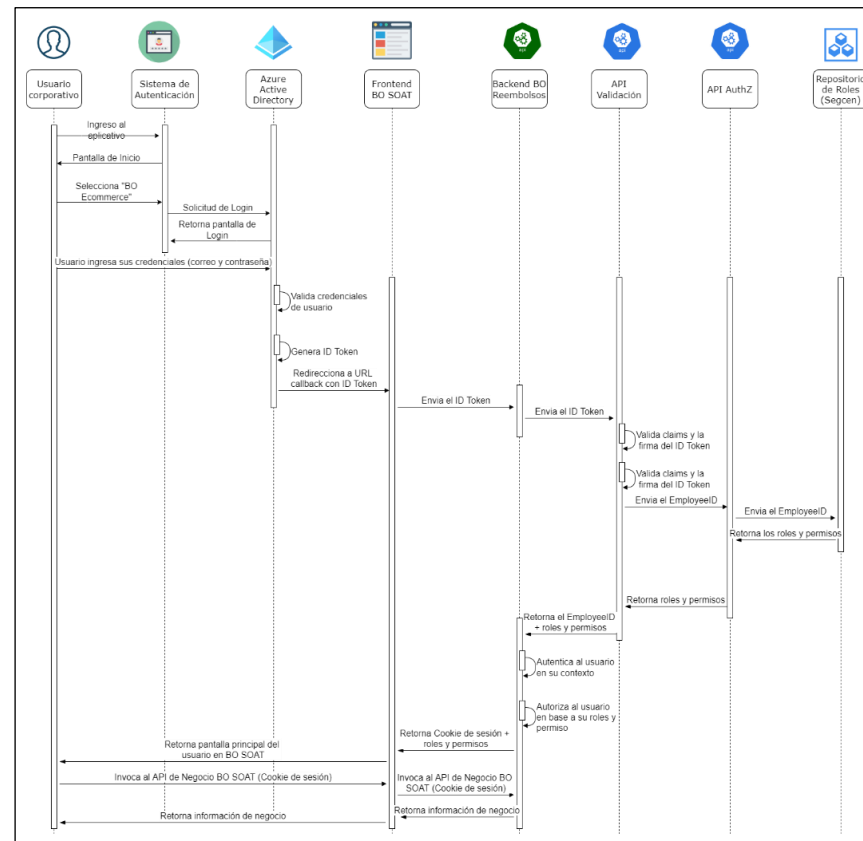


Nota. Adaptado de la Empresa Aseguradora.

A continuación, se muestra el diagrama de secuencia del BackOffice SOAT

Figura 38

Diagrama de secuencia - BackOffice SOAT



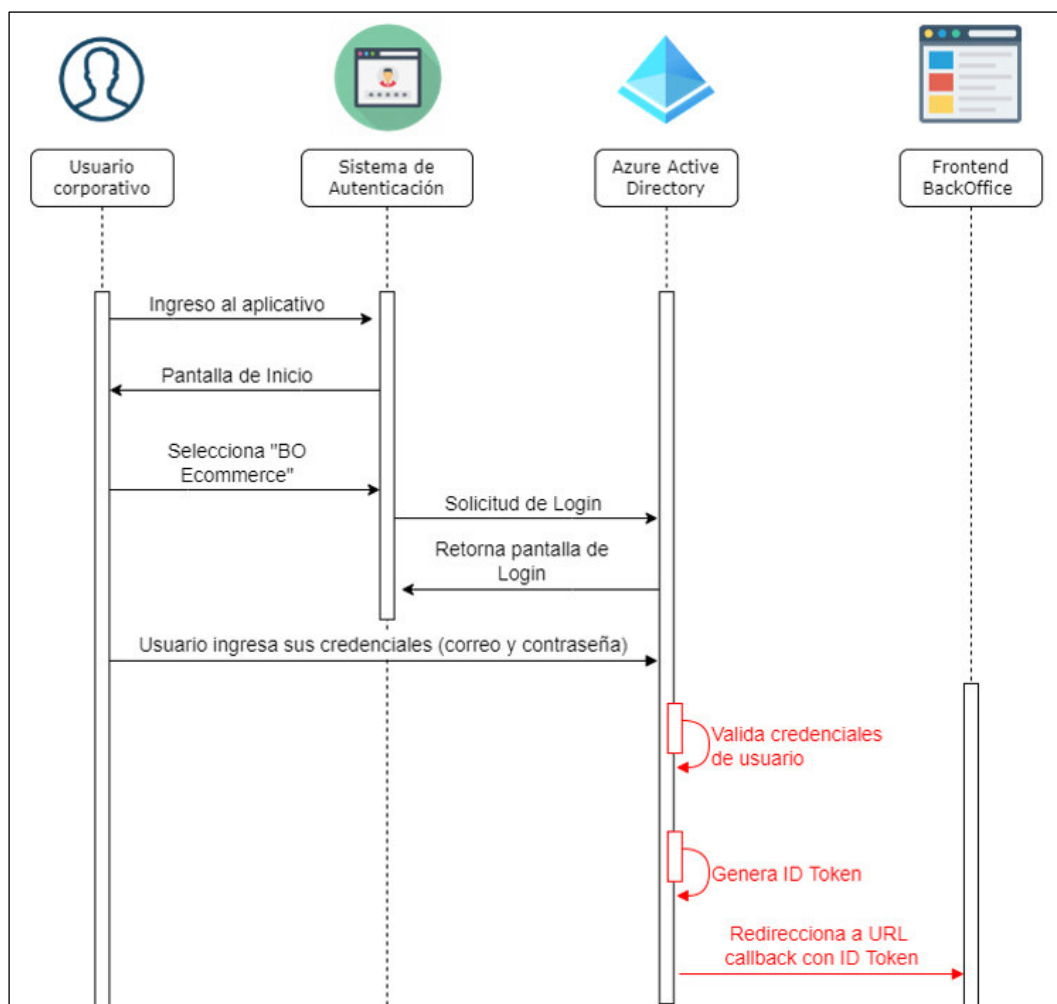
Nota. Adaptado de la Empresa Aseguradora.

3.2.5.3 Configuración de Azure Active Directory

Como se muestra en la Figura 39, Azure Active Directory se comunica con la aplicación BackOffice después de realizar la autenticación exitosa, redireccionando al Frontend de la aplicación BackOffice e incluyendo el ID token (código que indica la confirmación de la autenticación del usuario) en dicha redirección.

Figura 39

Diagrama de Flujo Azure Active Directory – Frontend BackOffice



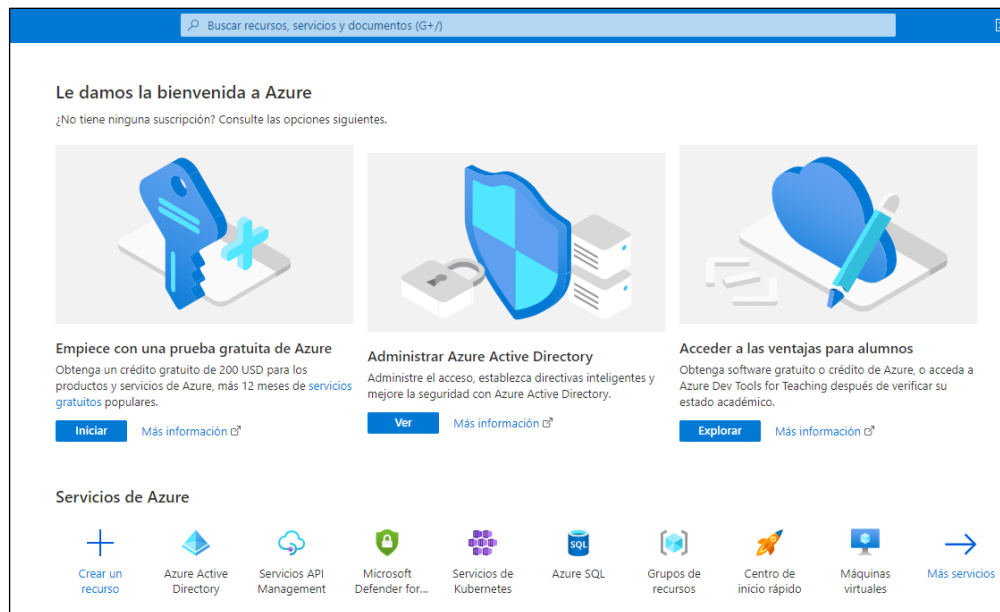
Nota. Adaptado de la Empresa Aseguradora.

Para lograr el objetivo en el párrafo anterior, el equipo de desarrollo del proyecto necesitó realizar las siguientes actividades para establecer la comunicación:

- a. Ingresar al portal de Azure.

Figura 40

Portal de Inicio – Microsoft Azure

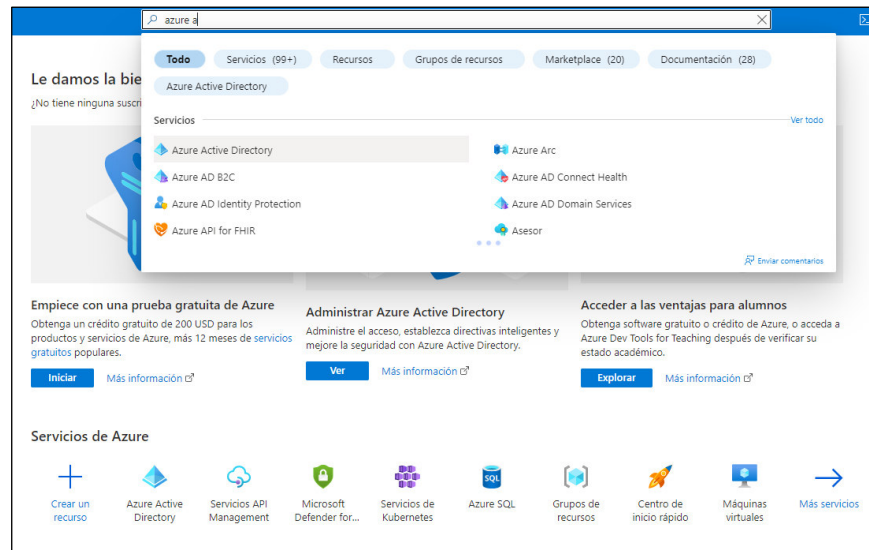


Nota. Adaptado de la Empresa Aseguradora.

- b. Buscar el servicio “Azure Active Directory”.

Figura 41

Buscador de Microsoft Azure

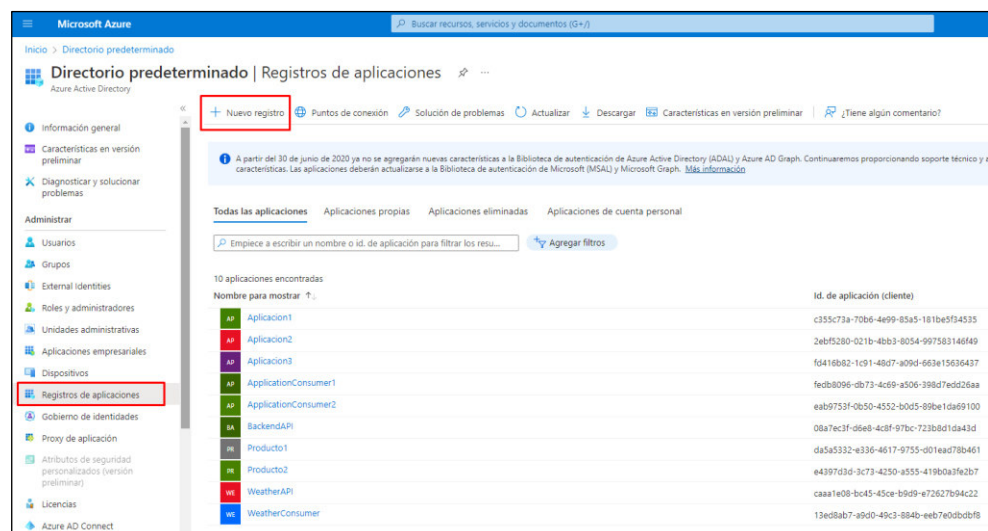


Nota. Adaptado de la Empresa Aseguradora.

- c. Ingresar a la opción “Registro de Aplicaciones” y seleccionar “Nuevo Registro”.

Figura 42

Pantalla de registro de aplicaciones en Azure Active Directory – Microsoft Azure

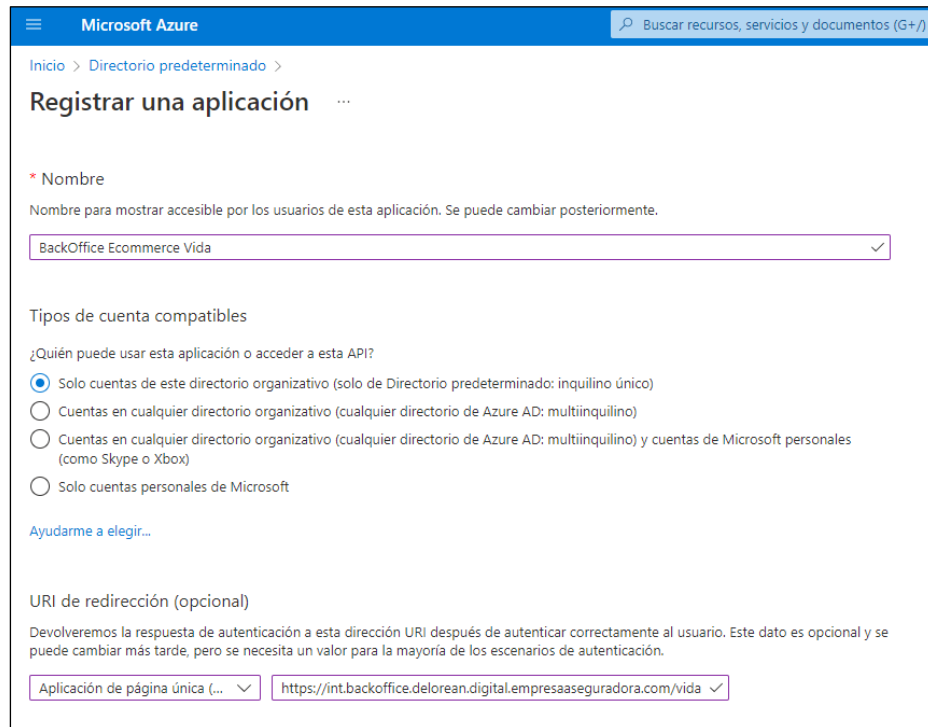


Nota. Adaptado de la Empresa Aseguradora.

- d. En la pantalla de registro como se muestra en la Figura 43 se debe seleccionar el Nombre, el Tipo de cuenta y la URL de redirección, a la que será redirigido el usuario después de la autenticación exitosa en Azure Active Directory.

Figura 43

Registro de una aplicación Azure Active Directory – Microsoft Azure



Nota. Adaptado de la Empresa Aseguradora.

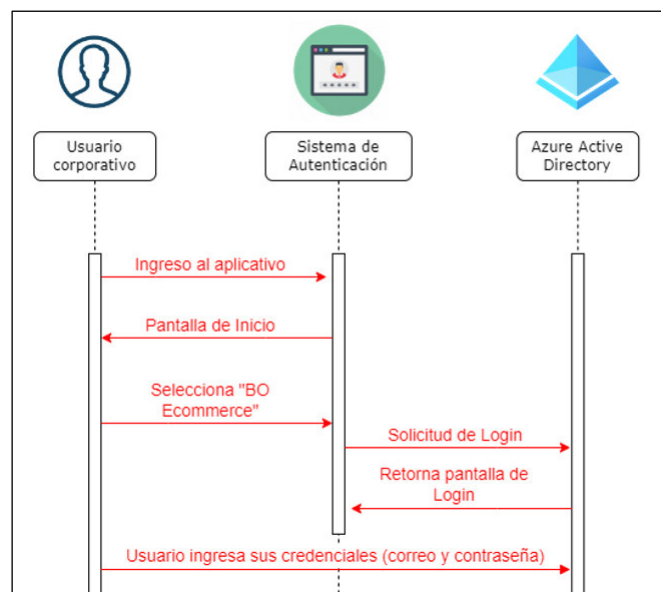
3.2.5.4 Construcción de la Interfaz Web

Luego de realizar las configuraciones en Azure Active Directory, se diseñó una interfaz web para poder brindarle un único canal de acceso a los usuarios de todos los BackOffice, este sería el punto de entrada inicial del usuario de la Empresa Aseguradora.

Como se describe en la Figura 44, el usuario corporativo, como primer paso ingresará a esta interfaz para seleccionar el BackOffice al cual quiere acceder, y posteriormente dicha interfaz será la encargada de redirigir al usuario a la pantalla de autenticación Azure Active Directory donde podrá ingresar sus credenciales corporativas.

Figura 44

Diagrama de Flujo Usuario corporativo – Interfaz Web



Nota. Adaptado de la Empresa Aseguradora.

El desarrollo de la interfaz Web, fue creado utilizando el framework de javascript ReactJS y siguiendo los lineamientos y buenas prácticas de desarrollo de la Empresa Aseguradora.

En la Tabla siguiente, se detalla la estructura de archivos de la interfaz web:

Tabla 11

Estructura de archivos – Interfaz Web

Carpeta	Descripción
assets	Contenido css/fonts
components	Componentes react transversales a la aplicación
containers	Componentes que conforma la estructura de una determinada interfaz de usuario y que generalmente se conecta a proveedores de datos tales como redux
Libs	Librerías utilitarias y de acceso a terceros
models	Modelos e interfaces comunes a la aplicación
api	Implementación de los servicios rest
redux	Implementación de acciones/middlewares

hooks	Implementación de react hooks
index.js	Archivo principal que inicia la aplicación
.env	Archivo de configuración para variables de entorno

Nota. Adaptado de la Empresa Aseguradora.

Para realizar la comunicación entre la interfaz web y Azure Active Directory se utilizó la librería MSAL.js, la cual es una librería recomendada por Microsoft para las integraciones entre un SPA (Single Page Application) y Azure Active Directory.

En la siguiente Figura, se muestra la implementación de la librería MSAL.js en el desarrollo de la Interfaz Web.

Figura 45

Código de implementación de la librería MSAL.js

```
import {UserAgentApplication} from "msal";
import config from "./msal.json";

class AuthService {
  constructor(configuration) {
    this.signInOptions = {
      scopes: configuration.msal.scopes,
    };

    this.msalConfig = {
      auth: {
        authority: process.env.REACT_APP_AUTHORITY,
        clientId: process.env.REACT_APP_CLIENT_ID,
        validateAuthority: true,
        redirectUri: process.env.REACT_APP_REDIRECT_URI,
        postLogoutRedirectUri: process.env.REACT_APP_REDIRECT_URI,
        navigateToLoginRequestUrl: false,
      },
      cache: {
        cacheLocation: "localStorage",
      },
      system: {
        navigateFrameWait: 0,
      },
    };

    this.msalClient = new UserAgentApplication(this.msalConfig);
    this.msalClient.handleRedirectCallback(this.redirectCallback);
  }

  redirectCallback(error, response) {
    if (error) {
      console.log(error);
    } else {
      console.log(response);
    }
  }

  signIn() {
    this.msalClient.loginRedirect(this.signInOptions);
  }

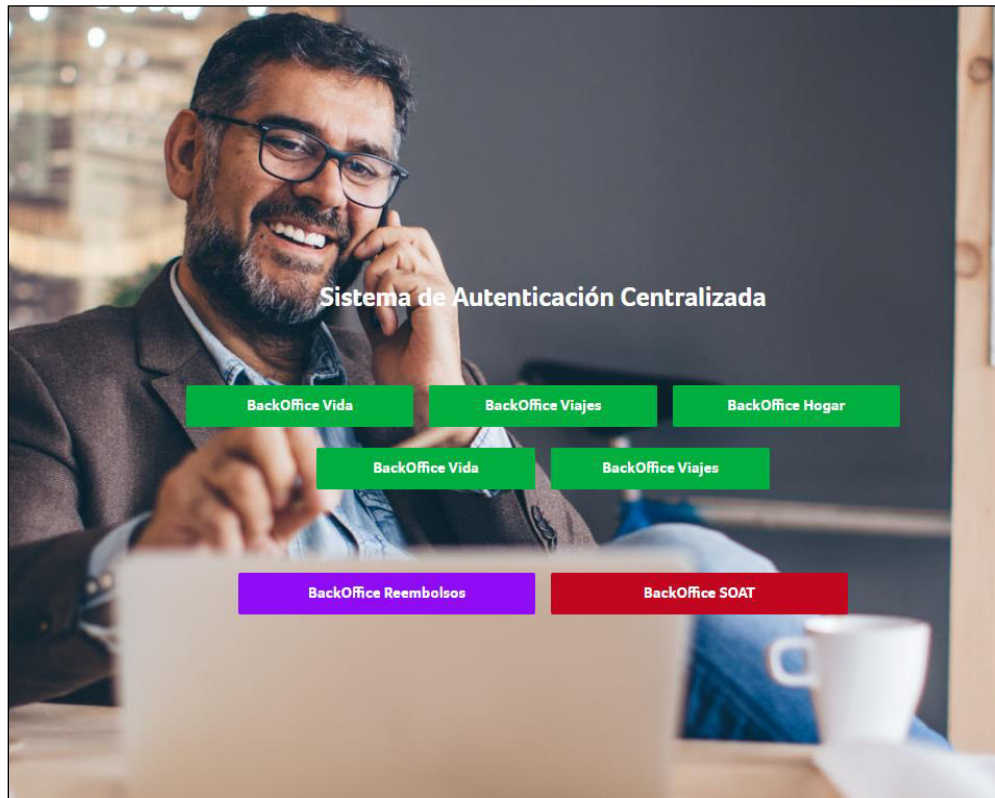
  signOut() {
    this.msalClient.logout();
  }
}
```

Nota. Adaptado de la Empresa Aseguradora.

Finalmente, el resultado del desarrollo de la interfaz web, integrado con Azure Active Directory, se muestra en las siguientes Figuras.

Figura 46

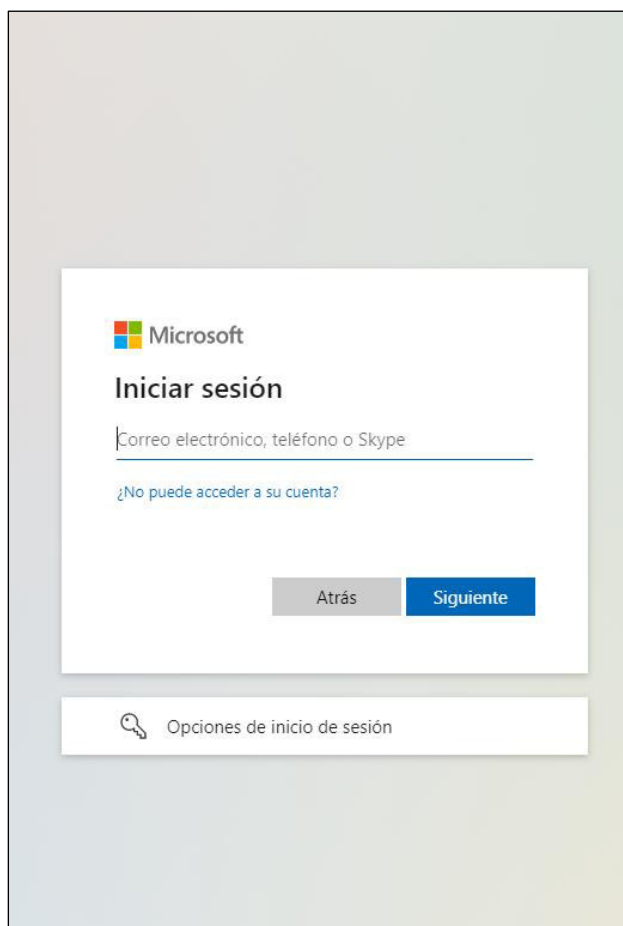
Pantalla de inicio del sistema de autenticación centralizada



Nota. Adaptado de la Empresa Aseguradora.

Figura 47

Pantalla de autenticación de Azure Active Directory



Nota. Adaptado de la Empresa Aseguradora.

Cuando la autenticación ha finalizado exitosamente, Azure Active Directory realizará una redirección hacia la página web configurada en el parámetro “URI de Redirección” de Azure Active Directory y enviará un ID Token, como prueba de que el usuario fue autenticado.

3.2.5.5 Construcción de Microservicios

Una vez realizado el envío del ID Token hacia las aplicaciones BackOffice, este debe pasar por un proceso de validación y autorización.

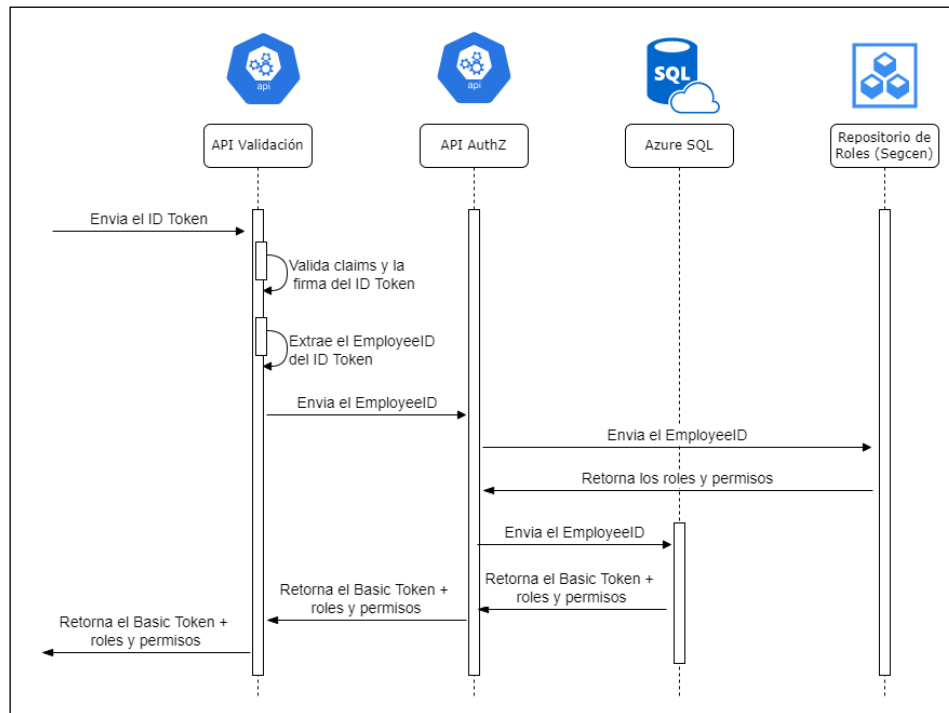
Para validar el ID Token primero es necesario descifrarlo, extraer los valores denominados “claims” (issuer y audience) para verificar que hayan sido emitidos por Azure Active Directory. Adicional a ello se debe validar la firma del ID Token, para lo cual el microservicio debe obtener las llaves públicas de Azure Active Directory, finalmente si este proceso es correcto, se enviará el EmployeeID (también extraído del ID Token) al microservicio de autorización.

Para autorizar al usuario en los aplicativos BackOffice, el microservicio encargado debe recibir el EmployeeID y consultar al repositorio SegCen, los roles y permisos del usuario. Sólo en el caso de los BackOffice Ecommerce, este microservicio también tendrá que obtener de la base de datos el Basic Token.

Como se observa en la Figura 48, se describen los procesos anteriormente mencionados.

Figura 48

Diagrama de flujo - Microservicios



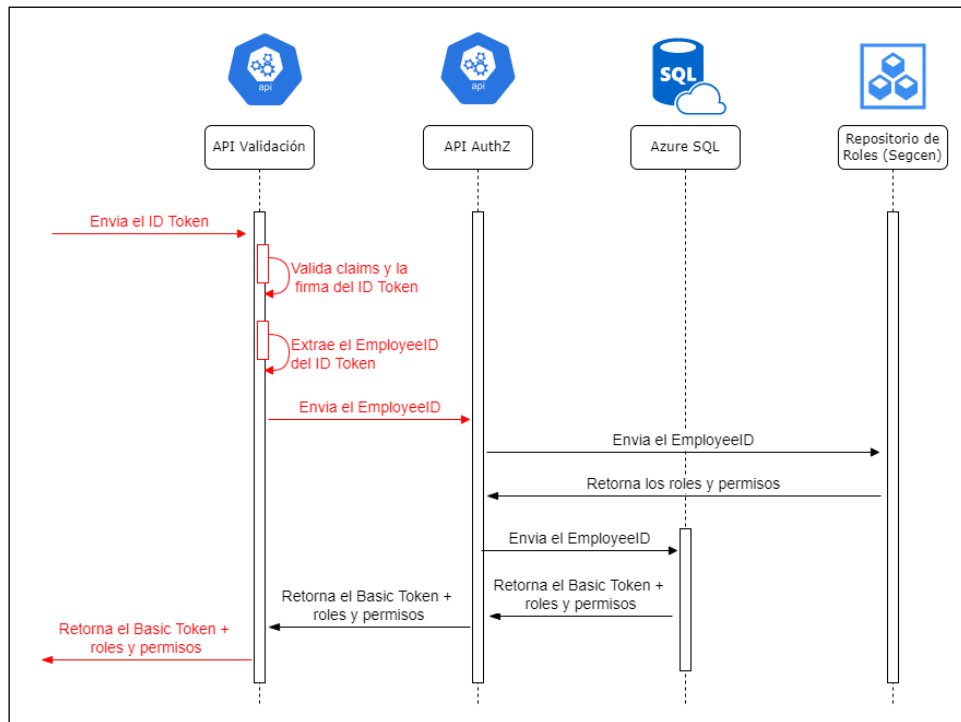
Nota. Adaptado de la Empresa Aseguradora.

El desarrollo del microservicio de Validación incluyó 2 endpoints o puntos finales, los cuales se describen a continuación:

- **validate-ecommerce:** Encargado de validar el ID token generado por los BackOffice Ecommerce y enviarlo al microservicio de Autorización, para finalmente devolver el Basic Token y los roles y permisos del usuario.

Figura 49

Diagrama de Flujo – Endpoint validate-ecommerce



Nota. Adaptado de la Empresa Aseguradora.

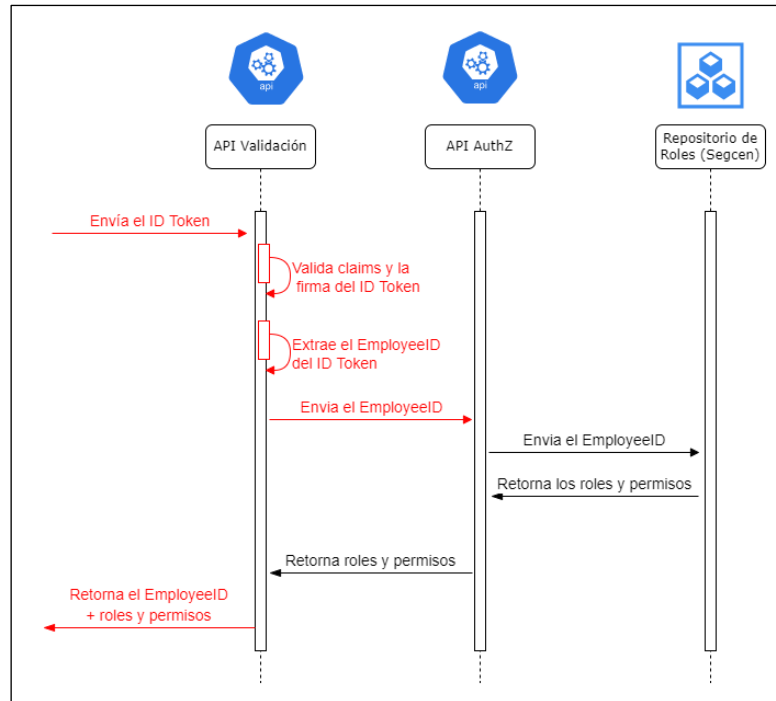
A continuación, se describen los pasos que debió implementar el microservicio de validación de token para los BackOffice Ecommerce:

1. Recibir el ID Token.
2. Descifrar el ID Token, y extraer los claims (issuer y audience).
3. Validar los claims.
4. Realizar una solicitud a Azure Active Directory para obtener las llaves de validación de firma.
5. Validar la firma.
6. Extraer del ID Token el EmployeeID y el tipo de aplicación.
7. Enviar el EmployeeID al endpoint del microservicio de Autorización.
8. Recibir el Basic Token y los roles del microservicio de Autorización.
9. Enviar el Basic Token y los roles al frontend del BackOffice Ecommerce.

- **validate-oauth:** Encargado de validar el ID token generado por los BackOffice Reembolsos y SOAT y enviarlo al microservicio de Autorización, para finalmente devolver el EmployeeID y los roles y permisos del usuario.

Figura 50

Diagrama de Flujo – Endpoint validate-oauth



Nota. Adaptado de la Empresa Aseguradora.

A continuación, se describen los pasos para implementar el microservicio de validación de token para los BackOffice Reembolsos y SOAT

1. Recibir el ID Token.
2. Descifrar el ID Token, y extraer los claims (issuer y audience).
3. Validar los claims.
4. Realizar una solicitud a Azure Active Directory para obtener las llaves de validación de firma.
5. Validar la firma.
6. Extraer del ID Token el EmployeeID y el tipo de aplicación.
7. Enviar el EmployeeID al endpoint del microservicio de Autorización.

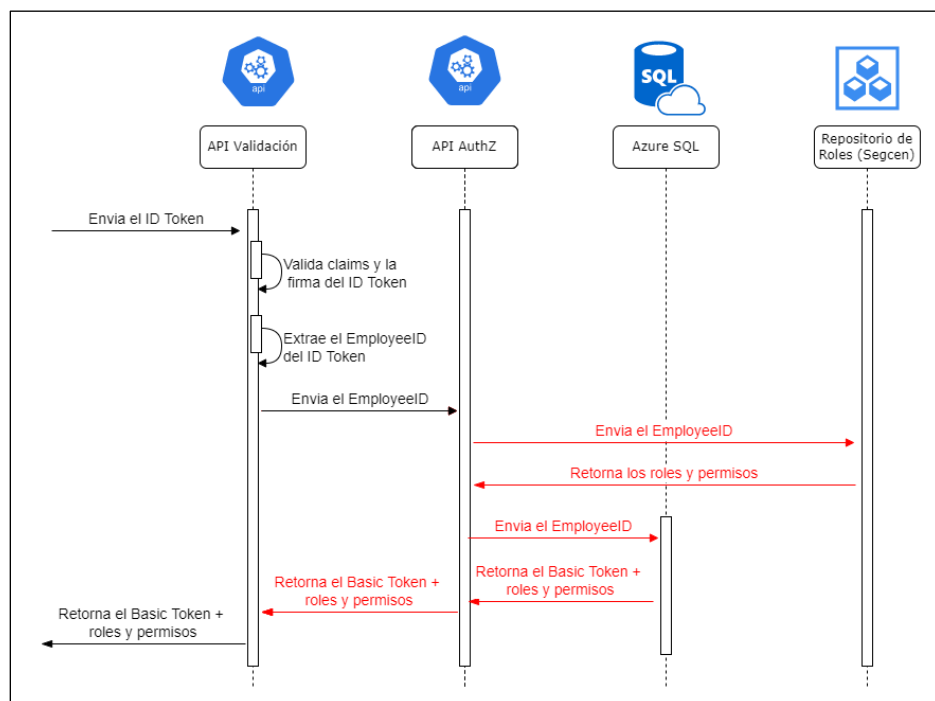
8. Recibir los roles y permisos del microservicio de Autorización.
9. Enviar el EmployeeID, los roles y permisos al backend de los BackOffice Reembolsos o SOAT.

El desarrollo del microservicio de Autorización incluyó 2 endpoints o puntos finales, los cuales se describen a continuación:

- **autoriza-ecommerce:** Encargado de recibir el EmployeeID del servicio de validación, buscar en la base de datos el Basic Token del usuario y consultar al repositorio de roles de la Empresa Aseguradora, para finalmente devolver el Basic Token y los roles y permisos del usuario.

Figura 51

Diagrama de Flujo – Endpoint autoriza-ecommerce



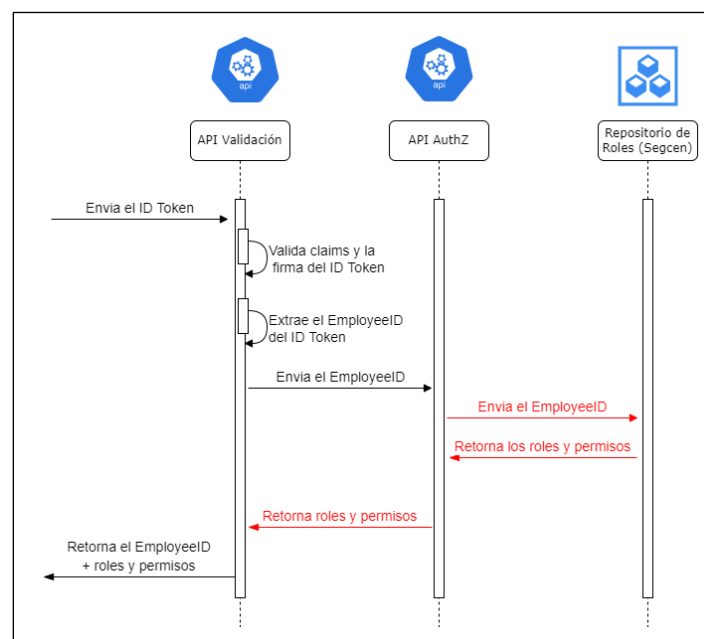
Nota. Adaptado de la Empresa Aseguradora.

A continuación, se describen los pasos que debe implementar el microservicio de autorización, debido al tipo de autorización con el que cuentan los Ecommerce fue necesario almacenar los Basic Token en una base de datos Azure SQL.

1. Recibir el EmployeeID del microservicio de validación.
 2. Enviar el EmployeeID al servicio REST que expone el repositorio de Roles y permisos SegCen.
 3. Recibir los roles y permisos de SegCen.
 4. Realizar la búsqueda del Basic Token del usuario por el EmployeeID en la base de datos.
 5. Retornar el Basic Token y los roles y permisos al microservicio de validación.
- **autORIZA-oauth:** Encargado de recibir el EmployeeID del servicio de validación y consultar al repositorio de roles de la Empresa Aseguradora, para finalmente devolver el EmployeeID, los roles y permisos del usuario.

Figura 52

Diagrama de Flujo – Endpoint autoriza-oauth



Nota. Adaptado de la Empresa Aseguradora.

A continuación, se describen los pasos que debe implementar el microservicio de autorización, debido al tipo de autorización con el que cuentan los BackOffice de Reembolsos y SOAT no fue necesario utilizar una base de datos.

1. Recibir el EmployeeID del microservicio de validación.
2. Enviar el EmployeeID al servicio REST que expone el repositorio de Roles y permisos SegCen.
3. Recibir los roles y permisos de SegCen.
4. Retornar los roles y permisos al microservicio de validación.

El proyecto debió seguir las buenas prácticas y lineamientos brindadas por el área de Arquitectura Digital de la Empresa Aseguradora, en la siguiente Tabla, se detallan la estructura de carpetas del proyecto y la finalidad de cada una de estas.

Tabla 12

Estructura de archivos – Microservicios

Carpeta	Descripción
Business	Destinado a la lógica de negocio
Config	Configuraciones necesarias antes del proceso de compilación
Exception	Excepciones controladas
Expose	Controladores que expondrán los endpoints
Model	Modelos e interfaces
Repository	Comunicación con Base de Datos
Webclient	Comunicación con API o servicios externos

Nota. Adaptado de la Empresa Aseguradora.

3.2.5.6 Pruebas

A continuación, se describen las pruebas realizadas en el proyecto:

Pruebas Unitarias

Fueron realizadas a cargo del equipo de desarrollo, estas consisten en probar fragmentos funcionales de código y comprobar su funcionamiento según lo esperado. Para ello se utilizó la librería Junit y Mock para los microservicios desarrollados en Java; y Jest.js para la interfaz web desarrollada en React.js.

Pruebas Integrales

Fueron realizadas a cargo del equipo de desarrollo y su objetivo fue validar los endpoints de los microservicios, comprobando las entradas y salidas de estos.

- Microservicio de Validación

A continuación, se muestran las entradas y salidas del microservicio de validación, el cual tiene 2 endpoints o puntos finales.

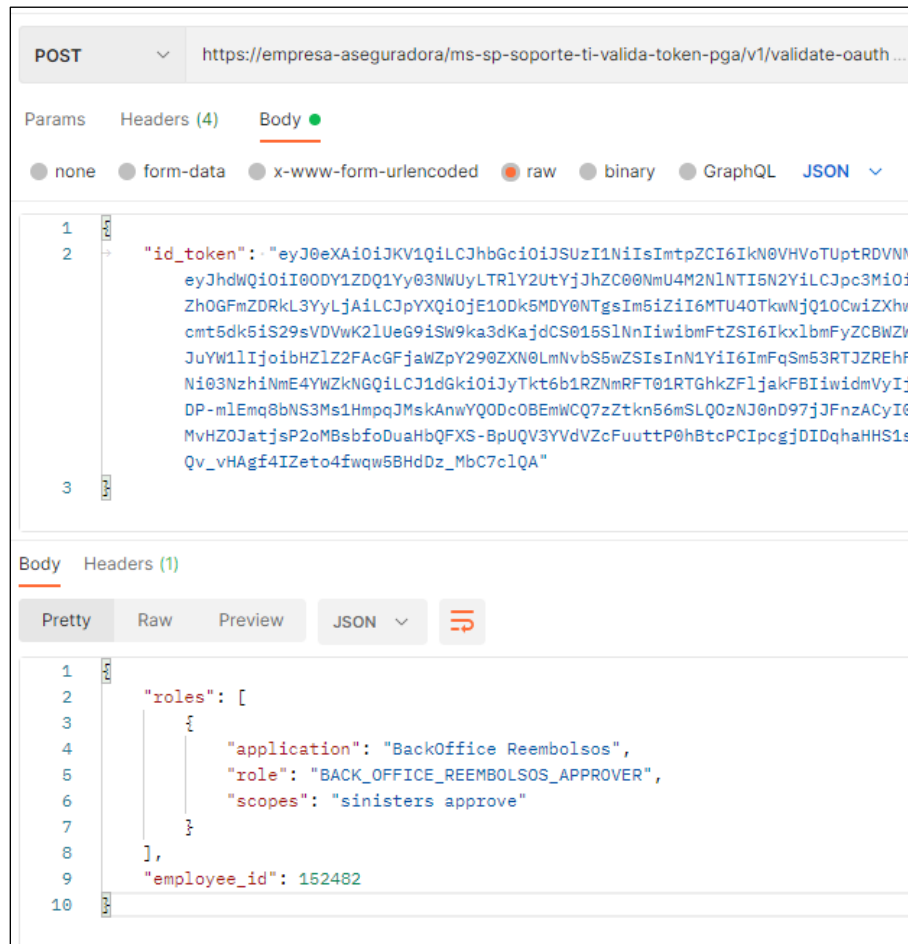
- a. Validate-ecommerce: El cual recibe el id token como entrada y devuelve los roles, permisos y el token basic necesario para los BackOffice de Ecommerce.
- b. Validate-oauth: El cual recibe el id token como entrada y devuelve los roles, permisos y el employeeID del usuario.

Evidencia Endpoint validate-ecommerce



Figura 54

Evidencia Endpoint validate-oauth



Nota. Adaptado de la Empresa Aseguradora.

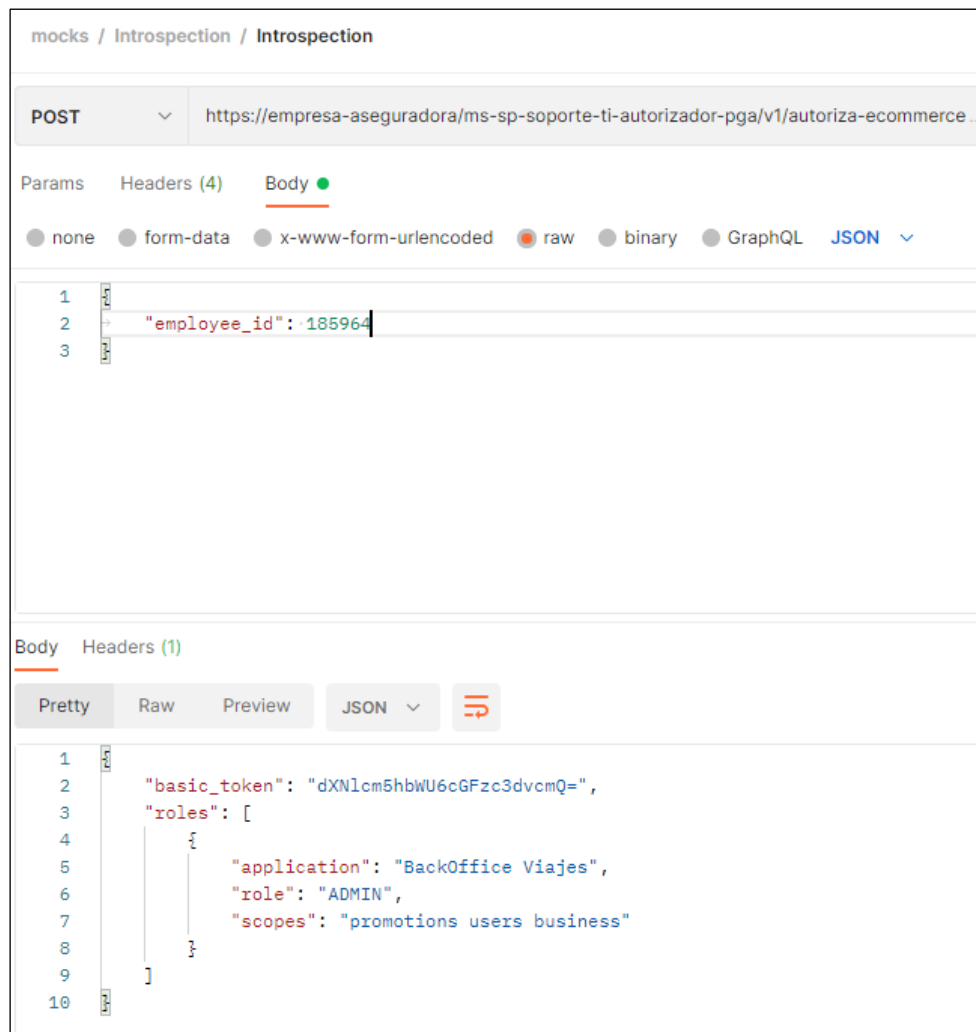
- Microservicio de Autorización

A continuación, se muestran las entradas y salidas del microservicio de autorización, el cual tiene 2 endpoints o puntos finales:

- Autoriza-ecommerce: El cual recibe el `employee_id` como entrada y devuelve los roles, permisos y el token basic necesario para los BackOffice de Ecommerce.
- Autoriza-oauth: El cual recibe el `employee_id` como entrada y devuelve los roles y permisos.

Figura 55

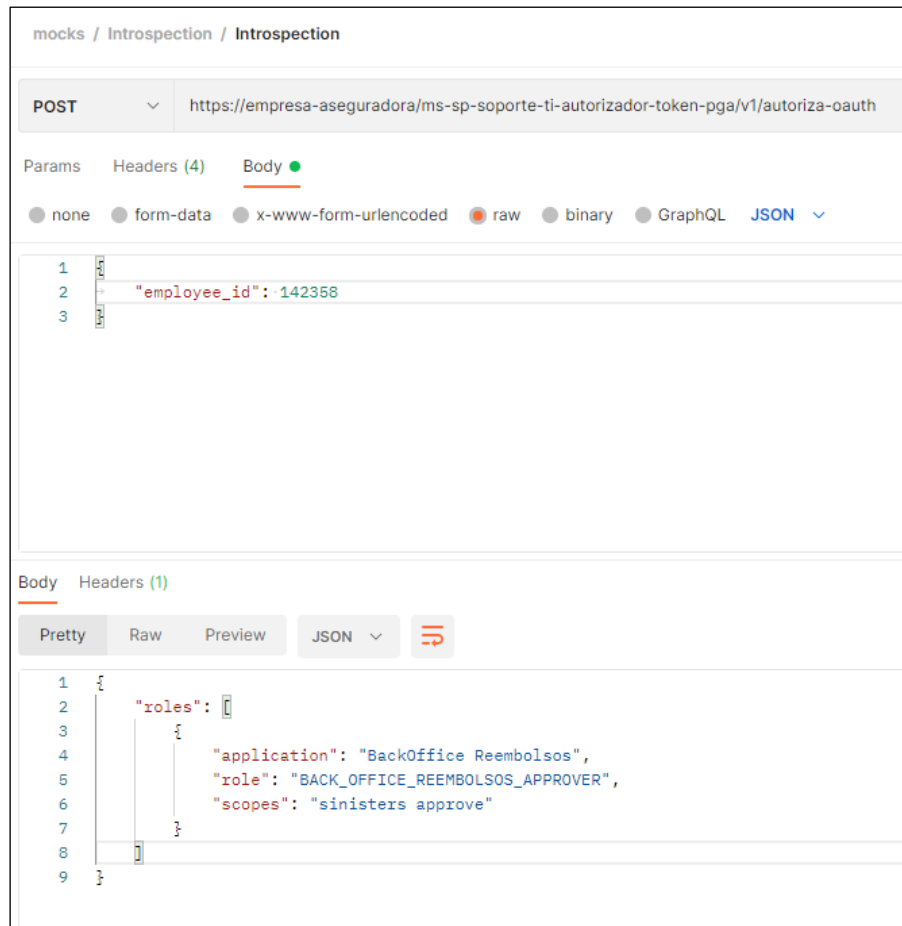
Evidencia Endpoint autoriza-ecommerce



Nota. Adaptado de la Empresa Aseguradora.

Figura 56

Evidencia Endpoint autoriza-oauth



Nota. Adaptado de la Empresa Aseguradora.

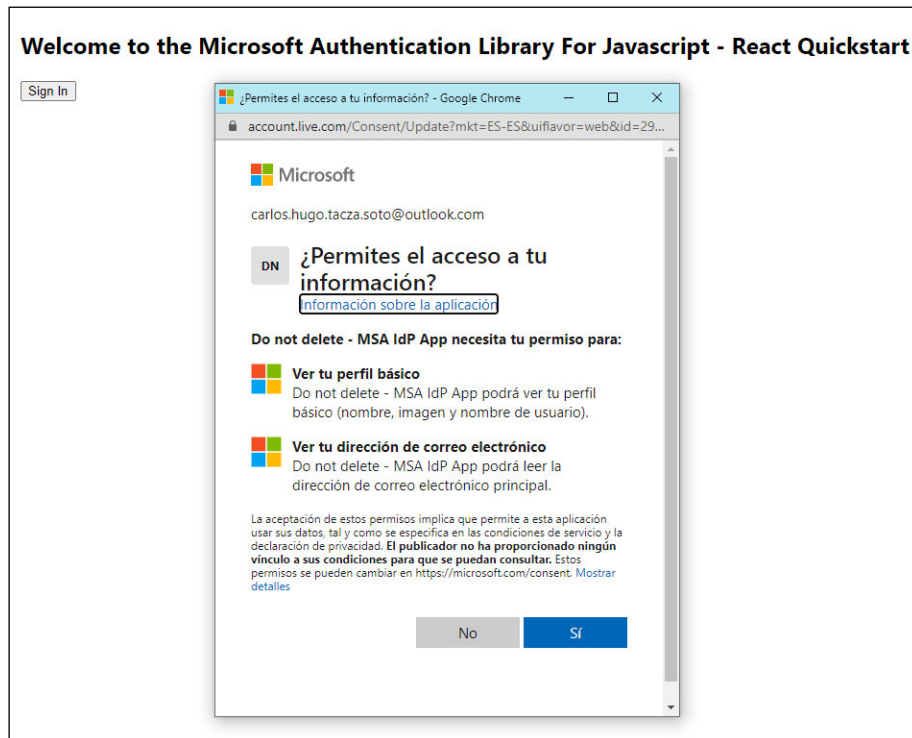
Pruebas de sistema

Las pruebas de sistema fueron realizadas por el equipo de QA del proyecto y tenían el objetivo de probar el flujo completo de autenticación y autorización de un usuario, desde la interfaz de autenticación centralizada hasta la validación y autorización de los microservicios.

Para dicho fin se debió utilizar una aplicación BackOffice de prueba, que fue construida por el equipo, esto debido a que aún no se realizaba la integración con ningún BackOffice.

Figura 57

Aplicación BackOffice de prueba



Nota. Adaptado de la Empresa Aseguradora.

Pruebas de rendimiento

Las pruebas de rendimiento fueron realizadas por el equipo de QA del proyecto y tenían por objetivo probar el sistema en escenarios concurrentes y analizar su comportamiento.

Los tipos de pruebas que se realizaron fueron: Pruebas de carga, estrés y estabilidad al sistema para lo cual se utilizó la herramienta Selenium.

En la Tabla 13 se detalla los tipos de prueba por cada uno de los componentes:

Tabla 13

Tipos de pruebas

Componente	Tipo de Prueba
Interfaz Web	Prueba de carga
Azure Active Directory	Prueba de carga
Microservicio de validación	Prueba de carga y estabilidad
Microservicios de Autorización	Prueba de carga, estrés y estabilidad

Nota. Adaptado de la Empresa Aseguradora.

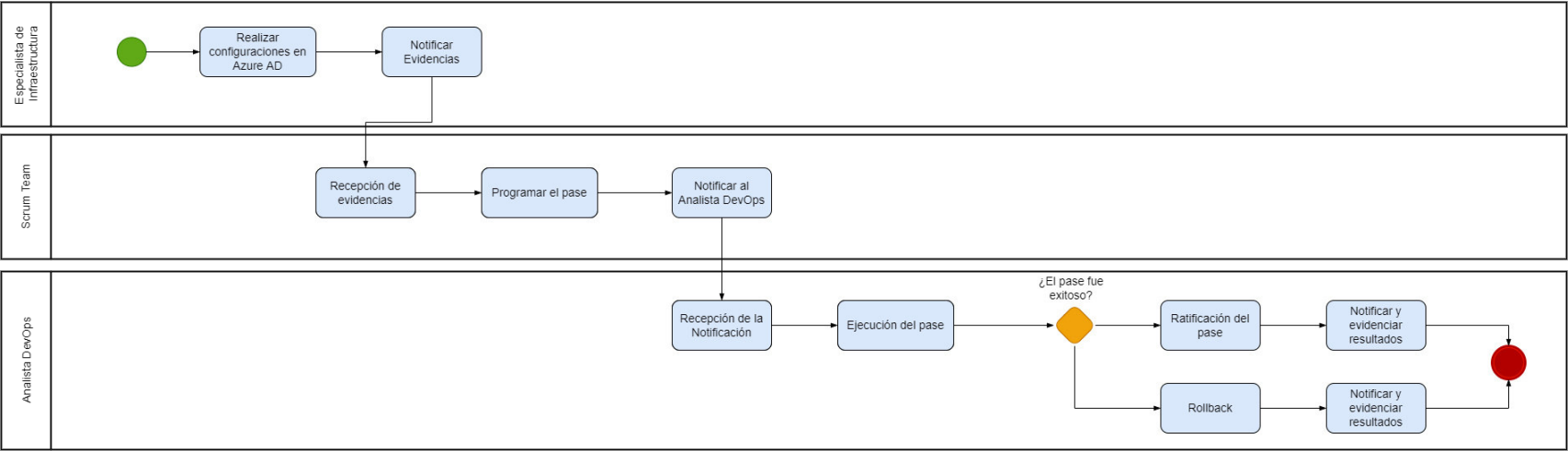
3.2.5.7 Liberación a Producción

El encargado de realizar las actividades necesarias para el pase a producción de todos los equipos de la Empresa Aseguradora es el Analista DevOps.

Sin embargo, debido a que el pase involucraba la herramienta Azure Active Directory, el especialista de Infraestructura debió realizar las configuraciones en dicha herramienta como paso previo al pase a producción.

En la siguiente Figura se describe el proceso que se realizó para ejecutar el pase a producción:

Figura 58
Diagrama de procesos de la ejecución del pase a producción



Nota. Adaptado de la Empresa Aseguradora.

A continuación, se describen las tareas de la figura anterior:

1. Realizar configuraciones en Azure AD: El especialista de Infraestructura fue el encargado de realizar las configuraciones necesarias en Azure Active Directory, en base al manual proporcionado por el equipo del proyecto.
2. Notificar Evidencias: El especialista de infraestructura mediante correo electrónico envió las evidencias de las configuraciones realizadas al equipo del proyecto.
3. Recepción de evidencias: El equipo del proyecto realizó la revisión de las configuraciones implementadas por el especialista de infraestructura.
4. Programar el pase: El Scrum Team realizó las coordinaciones necesarias para la ejecución del pase, en este caso debido a que es un sistema nuevo, solo se debió coordinar la ventana de tiempo disponible del Analista DevOps.
5. Notificar al Analista DevOps: Se notificó mediante correo electrónico al analista DevOps el horario de ejecución del pase y los componentes involucrados en dicho pase.
6. Recepción de la Notificación: El Analista DevOps, confirmó la recepción del correo y programó el pasé para la hora indicada.
7. Ejecución del Pase: El Analista DevOps ejecutó el pase a producción de todos los componentes especificados (Interfaz web y microservicios).
8. Ratificación del Pase: En el caso de que el pase fuese exitoso, el Analista DevOps notifica por correo al equipo del proyecto y envía las evidencias.
9. Rollback: En el caso de que el pase no haya sido exitoso, el Analista DevOps realiza el rollback inmediatamente y notifica con las evidencias necesarias al equipo del proyecto.

Luego de realizar el pase exitosamente, todos los componentes quedaron habilitados para empezar a integrar las aplicaciones BackOffice.

3.2.5.8 Integración de Aplicaciones BackOffice críticas

Luego de culminar el pase a producción de la interfaz web, los microservicios y configurar la herramienta Azure Active Directory, empezó la etapa de integrar las aplicaciones BackOffice críticas, la cual fue una etapa que duró 6 sprints debido a la cantidad y complejidad de cada uno de los sistemas en términos de arquitectura, lenguaje de programación y estructura de código.

Tabla 14

Duración de Integración por aplicación

Aplicación	Duración
BackOffice Ecommerce (Vida, Viajes, Autos, Hogar y Canales)	2 sprints
BackOffice Reembolsos	1 sprint
BackOffice SOAT	3 sprints

Nota. Adaptado de la Empresa Aseguradora.

A continuación, se describe la Integración realizada en las aplicaciones BackOffice.

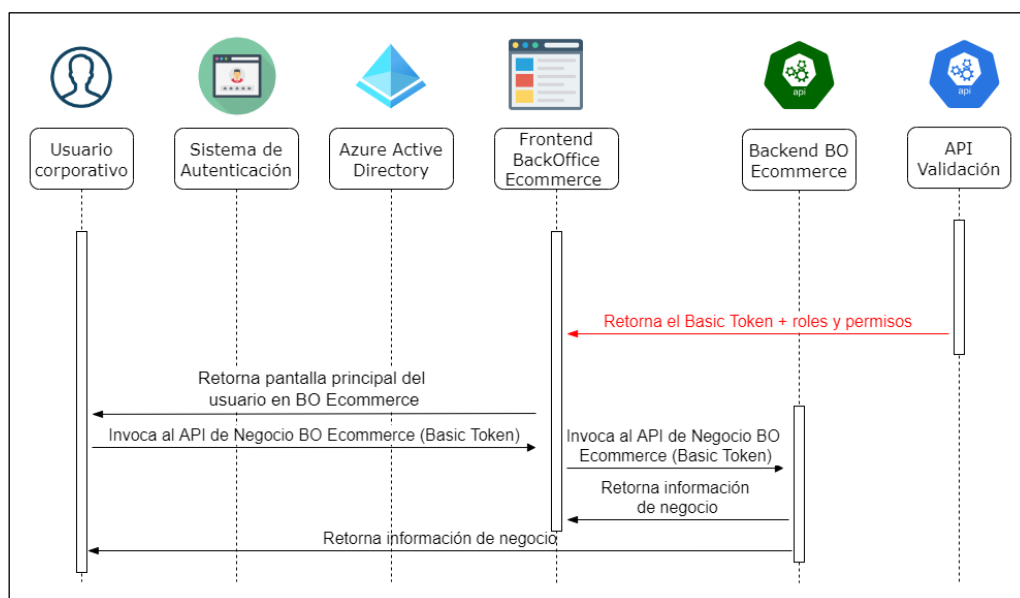
Aplicaciones BackOffice Ecommerce

Como se mencionó anteriormente las aplicaciones BackOffice Ecommerce agrupan los sistemas BackOffice de Vida, Viajes, Hogar, Autos y Canales. Debido a la similitud de estos se desarrollará solo una explicación.

En la Figura 59 se muestra que los roles y Basic Auth devueltos por el microservicio de validación llegan al Frontend del BackOffice Ecommerce, el cual debe confiar en la autenticidad del usuario y permitirle el ingreso sin solicitarle credenciales.

Figura 59

Flujo de Integración BackOffice Ecommerce



Nota. Adaptado de la Empresa Aseguradora.

Para realizar la integración se tuvieron que realizar cambios en el frontend de la aplicación, los cuales se describen a continuación:

1. Eliminar el módulo de Autenticación de la aplicación BackOffice Ecommerce, lo que implicaba quitar el módulo de login, asegurándose que eso no impacte en otros módulos de la aplicación.

Figura 60

Sección de código – Módulos del sistema BackOffice Ecommerce

```
src > containers > App.jsx > App
9  import { Page } from "../components/Page";
10 import { PopulateSession } from "../Auth/components/PopulateSession";
11 import Loading from "../components/Loading";
12
13 const AuthRoutes = lazy(() => import("../Auth/auth.routes"));
14 const CentralRoutes = lazy(() => import("../Central/central.routes"));
15
16 const App = () => {
17   return (
18     <StoreProvider store={store}>
19       <Suspense fallback={<Loading></Loading>}>
20         <Router basename={publicUrl}>
21           <Page>
22             <PopulateSession>
23               <Switch>
24                 <NotLoggedRoute
25                   path={routes.AUTH}
26                   component={AuthRoutes}
27                 ></NotLoggedRoute>
28                 <ProtectedRoute
29                   path={routes.CENTRAL}
30                   component={CentralRoutes}
31                 ></ProtectedRoute>
32                 <Redirect to={routes.CENTRAL}></Redirect>
33               </Switch>
34             </PopulateSession>
35           </Page>
36         </Router>
37       </Suspense>
38     </StoreProvider>
39   );
40 };
41
42 export default App;
```

Nota. Adaptado de la Empresa Aseguradora.

2. Modificar el archivo de inicio de la aplicación para que extraiga el ID Token de localStorage (por defecto es guardado en esa ubicación cuando se realiza la redirección).

Figura 61

Sección de código – Almacenamiento en localStorage – BackOffice Ecommerce

```
src > libs > helpers > JS auth-token.js > AuthToken > getTokenAAD
1  import { KEY_TOKEN, MSAL_TOKEN } from "../../constants/storage";
2  import { skipLoginAzureAD } from "../../libs/security/auth-utils";
3
4  const storage = localStorage;
5
6  class AuthToken {
7    static getToken() {
8      return storage.getItem(KEY_TOKEN) || null;
9    }
10
11   static getTokenAAD() {
12     return storage.getItem(MSAL_TOKEN) || null;
13   }
14
15   static setToken(token) {
16     if (token) {
17       if (skipLoginAzureAD()) {
18         storage.setItem(KEY_TOKEN, window.btoa(token));
19       }
20     } else {
21       storage.setItem(KEY_TOKEN, token);
22     }
23   } else {
24     storage.removeItem(KEY_TOKEN);
25   }
26 }
```

Nota. Adaptado de la Empresa Aseguradora.

3. Modificar el archivo de inicio de la aplicación para que antes de renderizar la página web, envíe el id Token al microservicio de validación.

Figura 62

Sección de código – Envío de ID Token – BackOffice Ecommerce

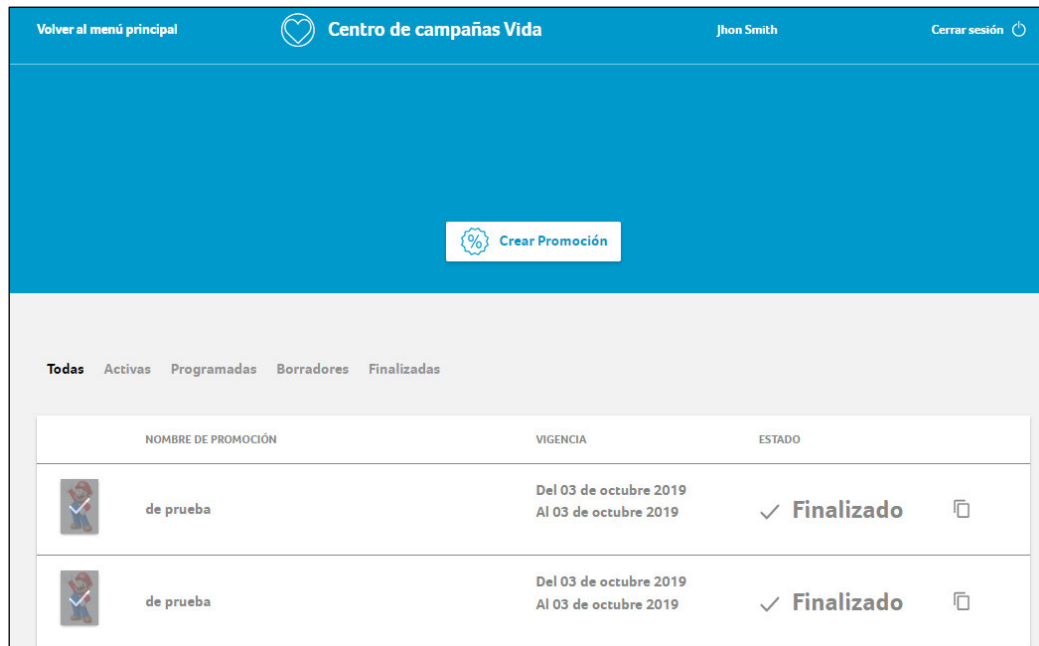
```
src > index.jsx > renderApp
22
23   const rootElement = document.getElementById("root");
24
25   const renderApp = () => {
26     if (rootElement && rootElement.hasChildNodes()) hydrate(<App />, rootElement);
27     else render(<App />, rootElement);
28     serviceWorker.unregister();
29     if (module.hot) {
30       module.hot.accept("./containers/App", () => {
31         render(<App />, rootElement);
32       });
33     }
34   };
35
36   async function runApp() {
37     render(<Loading />, rootElement);
38     const idToken = AuthToken.getTokenAAD();
39     if (idToken) {
40       ajax.post(
41         idToken,
42         ({ response }) => {
43           getUser(
44             response.token,
45             () => {
46               AuthToken.setToken(response.token);
47               renderApp();
48             },
49             () => {
50               AuthToken.clearToken();
51               renderApp();
52             }
53           );
54         }
55       );
56     }
57   }
58 }
```

Nota. Adaptado de la Empresa Aseguradora.

4. Preparar la página principal para mostrar las opciones de roles y permisos devueltos por el microservicio de validación.

Figura 63

Pantalla de inicio del usuario – BackOffice Ecommerce



Nota. Adaptado de la Empresa Aseguradora.

5. Guardar en el localStorage el Basic Token devuelto por el microservicio de validación.

Figura 64

Sección de código – Almacenamiento en localStorage de Basic Token – BackOffice Ecommerce

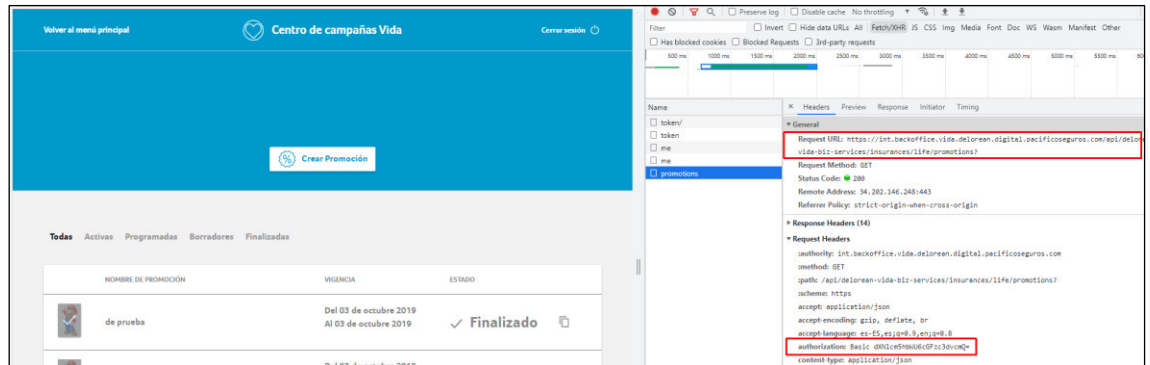
```
src > libs > helpers > JS auth-token.js > AuthToken > getTokenAAD
27
28 static getAuthorization() {
29   const token = AuthToken.getToken();
30   return `Basic ${token}`;
31 }
32
33 static clearToken() {
34   storage.removeItem(KEY_TOKEN);
35 }
36 }
37
38 export default AuthToken;
39
```

Nota. Adaptado de la Empresa Aseguradora.

6. Invocar a la API de negocio con el Basic Token guardado en localStorage

Figura 65

Pantalla de inicio del usuario – Envío de petición con Basic Token – BackOffice Ecommerce

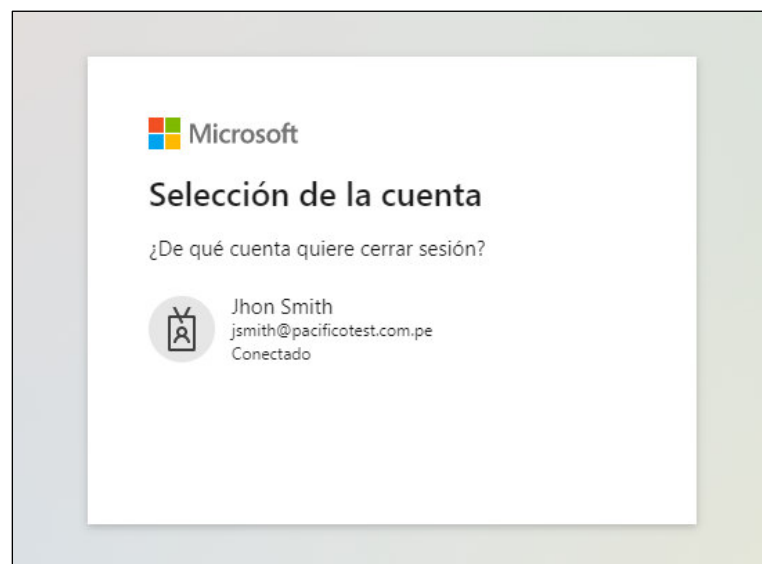


Nota. Adaptado de la Empresa Aseguradora.

7. Modificar el cierre de sesión para que utilice el servicio de Logout de Azure Active Directory.

Figura 66

Pantalla cierre de sesión – BackOffice Ecommerce



Nota. Adaptado de la Empresa Aseguradora.

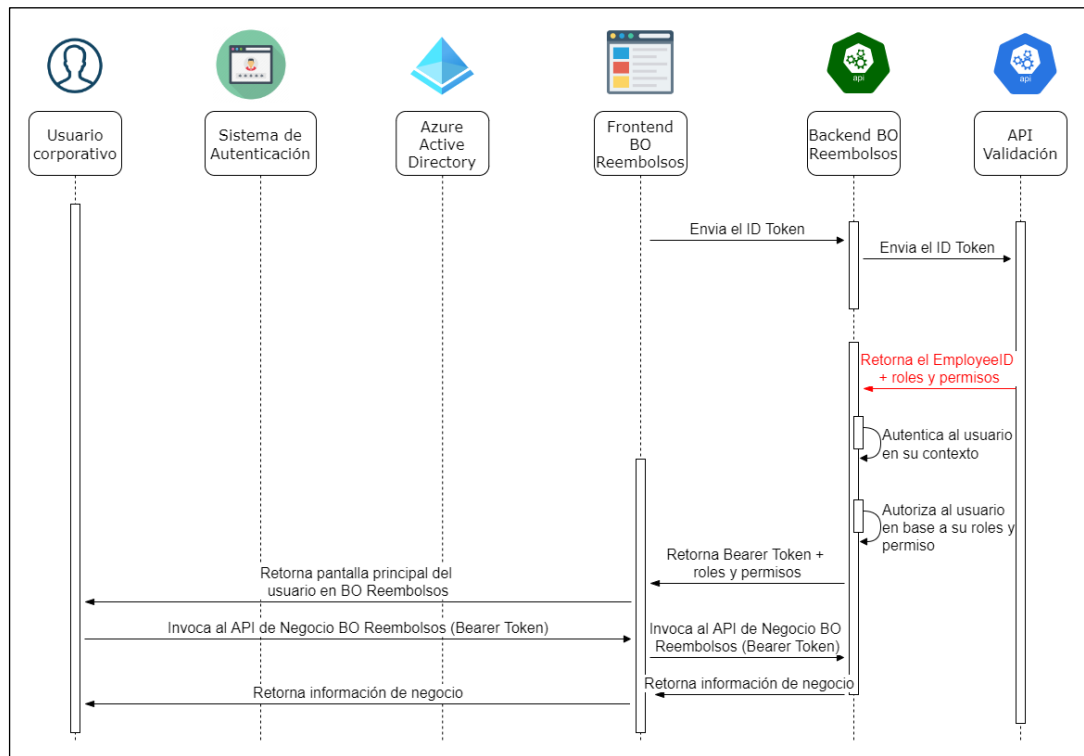
Aplicación BackOffice Reembolsos

En la Figura 67 se muestra que los roles, permisos y EmployeeID devueltos por el microservicio de validación llegan al Backend del BackOffice Reembolsos, el cual debe autenticar y autorizar al usuario en su contexto para finalmente obtener un Bearer Token, el cual le brinda al usuario autorización a las APIs de la aplicación.

Autenticar y Autorizar al usuario en su contexto significa que la propia aplicación debe generar el Bearer Token¹ de acuerdo con la tecnología que esté utilizando, y este Token no existe fuera de la aplicación.

Figura 67

Flujo de Integración BackOffice Reembolsos



Nota. Adaptado de la Empresa Aseguradora.

¹ Bearer Token, se explica con mayor detalle en la sección “Fundamentos Utilizados”

Para realizar la integración primero se tuvieron que realizar cambios en el frontend de la aplicación, los cuales se describen a continuación:

1. Eliminar el módulo de Autenticación de la aplicación BackOffice Reembolsos, lo que implicaba quitar el módulo de login, asegurándose que eso no impacte en otros módulos de la aplicación.

Figura 68

Sección de código – Módulos del sistema BackOffice

```
src > routes.jsx > ...
1  import React from 'react';
2  import { Route, Switch } from 'react-router-dom';
3  import Login from './components/layouts/login';
4  import dashboard from './components/layouts/dashboard';
5  import Error from './components/shared/error/error';
6
7  import {
8    ROUTE_LOGIN,
9    ROUTE_DEFAULT,
10   ROUTE_DASHBOARD,
11 } from './constants/routes/routes-ui';
12
13 const routes = (
14   <Switch>
15     <Route exact path={ROUTE_DEFAULT} component={Login} />
16     <Route exact path={ROUTE_LOGIN} component={Login} />
17     <Route path={ROUTE_DASHBOARD} component={dashboard} />
18     <Route path="*" component={Error} />
19   </Switch>
20 );
21
22 export default routes;
23
```

Nota. Adaptado de la Empresa Aseguradora.

2. Modificar el archivo de inicio de la aplicación para que extraiga el ID Token de localStorage (por defecto es guardado en esa ubicación cuando se realiza la redirección de Azure Active Directory).

Figura 69

Sección de código – Almacenamiento en localStorage

```
src > libs > helpers > JS auth-token.js > AuthToken > getTokenAAD
1  import { KEY_TOKEN, MSAL_TOKEN } from "../../constants/storage";
2  import { skipLoginAzureAD } from "../../libs/security/auth-utils";
3
4  const storage = localStorage;
5
6  class AuthToken {
7    static getToken() {
8      return storage.getItem(KEY_TOKEN) || null;
9    }
10
11   static getTokenAAD() {
12     return storage.getItem(MSAL_TOKEN) || null;
13   }
14
15   static setToken(token) {
16     if (token) {
17       if (skipLoginAzureAD()) {
18         storage.setItem(KEY_TOKEN, window.btoa(token));
19       }
20       else {
21         storage.setItem(KEY_TOKEN, token);
22       }
23     } else {
24       storage.removeItem(KEY_TOKEN);
25     }
26   }
27 }
```

Nota. Adaptado de la Empresa Aseguradora.

3. Modificar el archivo de inicio de la aplicación para que antes de renderizar la página web, envíe el id Token al backend de Reembolsos.

Figura 70

Sección de código – Envío de ID Token

```
src > index.jsx > renderApp
22
23 const rootElement = document.getElementById("root");
24
25 const renderApp = () => {
26   if (rootElement && rootElement.hasChildNodes()) hydrate(<App />, rootElement);
27   else render(<App />, rootElement);
28   serviceWorker.unregister();
29   if (module.hot) {
30     module.hot.accept("./containers/App", () => {
31       render(<App />, rootElement);
32     });
33   }
34 };
35
36 async function runApp() {
37   render(<Loading />, rootElement);
38   const idToken = AuthToken.getTokenAAD();
39   if (idToken) {
40     ajax.post(
41       idToken,
42       ({ response }) => {
43         getUser(
44           response.token,
45           () => {
46             AuthToken.setToken(response.token);
47             renderApp();
48           },
49           () => {
50             AuthToken.clearToken();
51             renderApp();
52           }
53         );
54       }
55     );
56   }
57 }
```

Nota. Adaptado de la Empresa Aseguradora.

4. Preparar la página principal para mostrar las opciones de roles y permisos devueltos por el backend del BackOffice de Reembolsos.

Figura 71

Pantalla de inicio del usuario – BackOffice Reembolsos

Reembolso							ExecutiveQA Es
BUSCAR POR		BUSCAR SOLICITUD			ESTADO DE SOLICITUD		
Número de Siniestro ▾		Introduce búsqueda			Todas ▾		
SINIESTRO	FECHA	TITULAR	PACIENTE	MONTO	ESTADO		
123465	28 mar. hace 3 años			s/. 0	En trámite		M
	28 mar. hace 3 años	Jerico Segura Internacional		s/. 0	En trámite Matrícula lista		
	28 mar. hace 3 años	Jerico Segura Test Eps		s/. 405.98	En trámite Matrícula lista		

Nota. Adaptado de la Empresa Aseguradora.

5. Guardar en el localStorage el Bearer Token devuelto por el backend del BackOffice de Reembolsos.

Figura 72

Sección de código – Almacenamiento en localStorage de Bearer Token – BackOffice Ecommerce

```
src > lib > common > JS auth-token.js > AuthToken > getAuthorization
1  import { CookieStorage } from 'cookie-storage';
2  import { KEY_TOKEN, KEY_EVENT, KEY_SESSION } from '../constants/cookie-storage';
3  import { KEY_TOKEN_MSAL } from '../constants/local-storage';
4
5  const store = new CookieStorage();
6  const localstore = localStorage;
7
8  class AuthToken {
9      static getToken() {
10         return JSON.parse(store.getItem(KEY_TOKEN)) || null;
11     }
12
13     static getTokenAAD() {
14         return localstore.getItem(KEY_TOKEN_MSAL) || null;
15     }
16
17     static setToken(token) {
18         if (token) {
19             store.setItem(KEY_TOKEN, JSON.stringify(token));
20             AuthToken.setSession();
21         } else {
22             store.removeItem(KEY_TOKEN);
23             AuthToken.removeSession();
24             AuthToken.removeLastUserEvent();
25         }
26     }
27 }
```

Nota. Adaptado de la Empresa Aseguradora.

6. Invocar a la API de negocio con el Bearer Token guardado en localStorage.

Figura 73

Envío de petición con Bearer Token – BackOffice Reembolsos

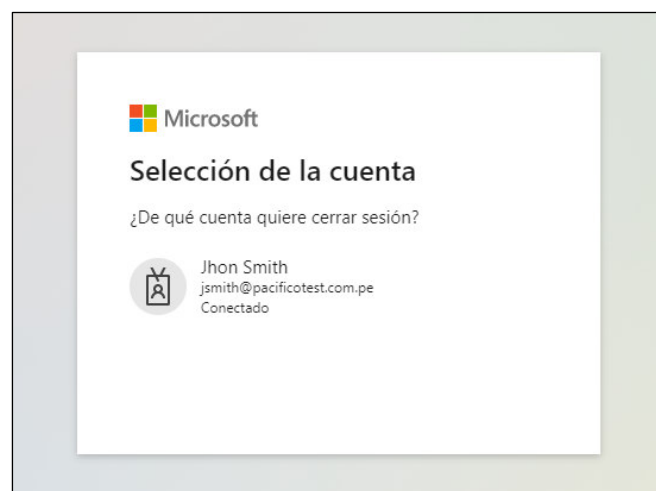
```
src > lib > common > JS header-requestjs > HeaderRequest > getRequestHeader > token
1  import AuthToken from './auth-token';
2
3  export default class HeaderRequest {
4    static getRequestFileHeader() {
5      const requestHeader = {
6        Authorization: AuthToken.getAuthorization(),
7        accept: 'application/json',
8        'Accept-Language': 'en-US,en;q=0.8',
9      };
10     return requestHeader;
11   }
12   static getRequestHeader() {
13     const requestHeader = { 'Content-Type': 'application/json', Accept: 'application/json' };
14     const token = AuthToken.getToken();
15     if (token) {
16       requestHeader.Authorization = AuthToken.getAuthorization();
17     }
18     return requestHeader;
19   }
20   static getPublicRequestHeader() {
21     const requestHeader = { 'Content-Type': 'application/json', Accept: 'application/json' };
22     return requestHeader;
23   }
24 }
```

Nota. Adaptado de la Empresa Aseguradora.

7. Modificar el cierre de sesión para que utilice el servicio de Logout de Azure Active Directory.

Figura 74

Pantalla cierre de sesión – BackOffice Reembolsos

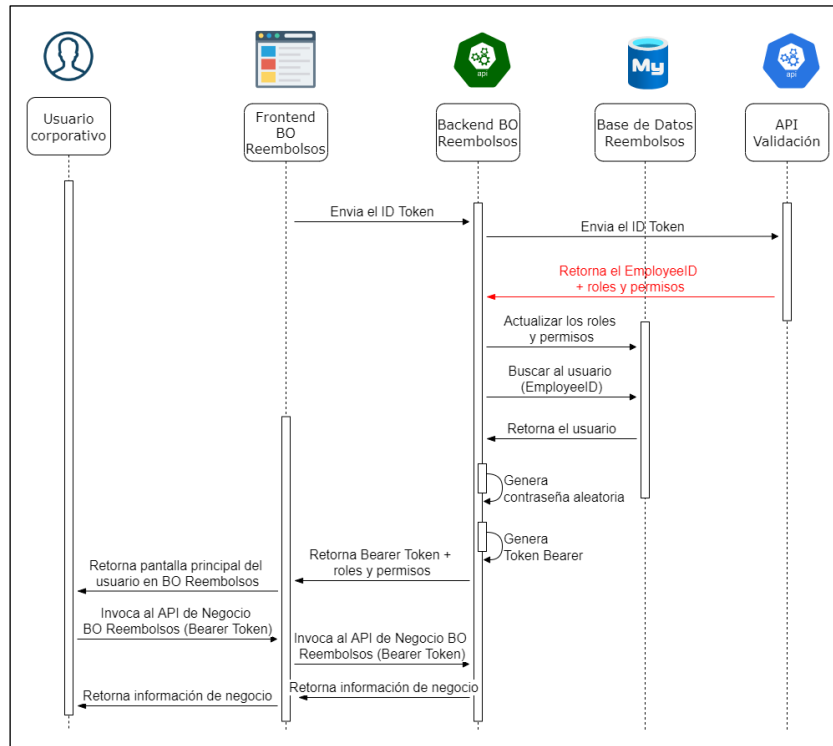


Nota. Adaptado de la Empresa Aseguradora.

También fue necesario realizar cambios en el backend de la aplicación, los cuales se describen a continuación:

Figura 75

Diagrama de flujo backend – BackOffice Reembolsos



Nota. Adaptado de la Empresa Aseguradora.

A continuación, se detallan las modificaciones que se debieron realizar:

1. Crear un endpoint que reciba el ID Token, enviado por el frontend.
2. Enviar el ID Token al microservicio de validación.
3. Recibir la respuesta del microservicio de validación con el EmployeeID y los roles y permisos del usuario.
4. Los roles y permisos deben ser actualizados en la Base de datos de la aplicación Reembolsos.
5. Con el EmployeeID se buscará al usuario en la Base de datos de la aplicación Reembolsos.

6. Una vez encontrado el usuario, se generará una contraseña aleatoria y se guardará en la base de datos de la aplicación Reembolsos.
7. El usuario y la contraseña generada debe mandarse a la función Oauth de la aplicación Reembolsos, la cual generará el Token Bearer de acuerdo con los roles y permisos actualizados en la Base de datos.
8. Finalmente, el Token Bearer será devuelto al frontend de la aplicación Reembolsos.
9. El Token Bearer será enviado en cada petición que realice el usuario hacia las APIs, demostrando que es un usuario autenticado.

Aplicación BackOffice SOAT

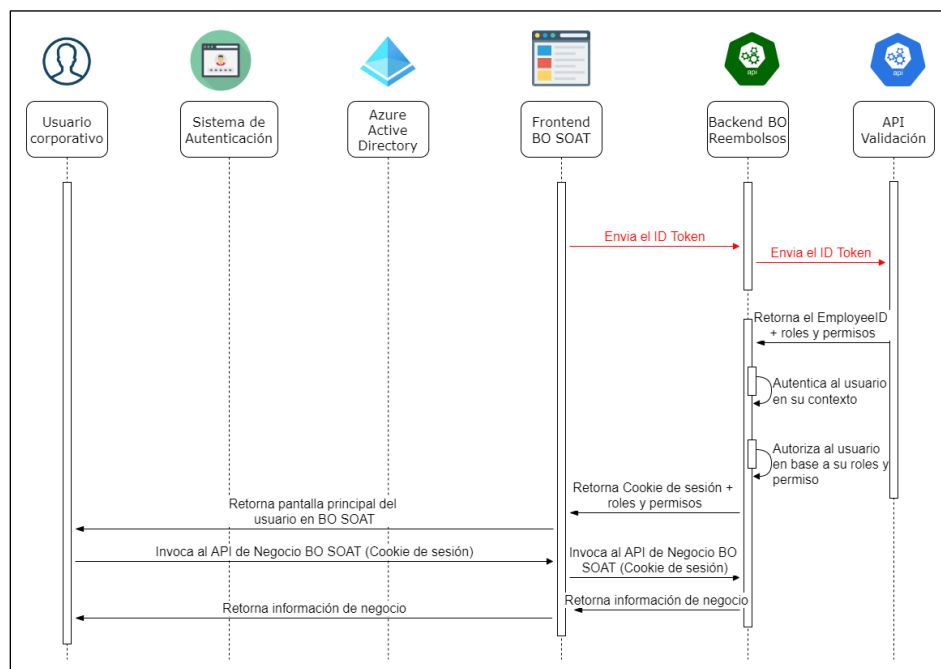
La aplicación SOAT estaba desarrollado en el lenguaje de programación PHP y utilizaba el framework Laravel 6.5, un lenguaje y framework bastante antiguo. Adicional a ello el frontend estaba desarrollado bajo el framework Vue.js. Por lo cual el equipo dedicó 1 sprint para comprender y analizar estas tecnologías.

En la Figura 76 se muestra que los roles, permisos y EmployeeID devueltos por el microservicio de validación llegan al Backend del BackOffice SOAT, el cual debe autenticar y autorizar al usuario en su contexto para finalmente obtener una Cookie de sesión, la cual le brinda al usuario autorización a las APIs de la aplicación SOAT.

Autenticar y Autorizar al usuario en su contexto significa que la propia aplicación debe generar el Cookie de Sesión² de acuerdo con la tecnología que esté utilizando, y esta cookie no existe fuera de la aplicación.

Figura 76

Flujo de Integración BackOffice SOAT



Nota. Adaptado de la Empresa Aseguradora.

² Cookie de Sesión, se explica con mayor detalle en la sección “Fundamentos Utilizados”

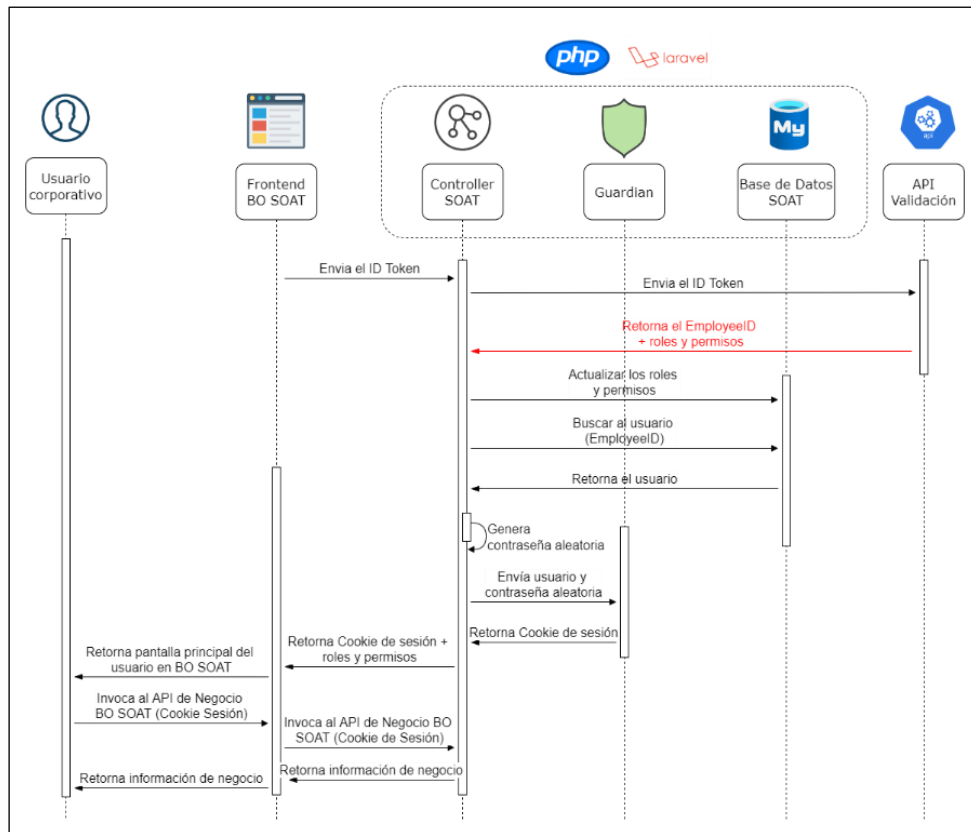
Para realizar la integración primero se tuvieron que realizar cambios en el frontend de la aplicación, los cuales se describen a continuación:

1. Eliminar el módulo de Autenticación de la aplicación BackOffice SOAT, lo que implicaba quitar la capa frontend de login, asegurándose que eso no impacte en otros módulos de la aplicación.
2. Modificar el archivo de inicio de la aplicación para que extraiga el ID Token de localStorage (por defecto es guardado en esa ubicación cuando se realiza la redirección de Azure Active Directory).
3. Modificar el archivo de inicio de la aplicación para que antes de renderizar la página web, envíe el ID Token al backend de SOAT.
4. Modificar el cierre de sesión para que utilice el servicio de Logout de Azure Active Directory.

También fue necesario realizar cambios en el backend de la aplicación, los cuales se describen a continuación:

Figura 77

Diagrama de flujo backend – BackOffice Reembolsos



Nota. Adaptado de la Empresa Aseguradora.

A continuación, se detallan las modificaciones que se debieron realizar:

1. Modificar el endpoint para que reciba el ID Token, enviado por el frontend.
2. Enviar el ID Token al microservicio de validación.
3. Recibir la respuesta del microservicio de validación con el EmployeeID y los roles y permisos del usuario.
4. Los roles y permisos deben ser actualizados en la Base de datos de la aplicación BackOffice SOAT.
5. Con el EmployeeID se buscará al usuario en la Base de datos de la aplicación BackOffice SOAT.
6. Una vez encontrado el usuario, se generará una contraseña aleatoria y se guardará en la base de datos de la aplicación BackOffice SOAT.

7. El usuario y la contraseña generada debe mandarse al módulo Guardian de la aplicación BackOffice SOAT, la cual generará la Cookie de sesión de acuerdo con los roles y permisos actualizados en la Base de datos.
8. Finalmente, la cookie de sesión será devuelta al frontend de la aplicación BackOffice SOAT.
9. La cookie de sesión será enviada en cada petición que realice el usuario hacia las APIs, demostrando que es un usuario autenticado.

Con la finalización de la integración de las 7 aplicaciones BackOffice críticas, se dio por concluido el proyecto, sin embargo, esta etapa fue la más retadora para el equipo, debido a la gran cantidad de información que tuvo que revisar y aprender para poder adaptar las aplicaciones a una forma de autenticación y autorización más moderna y centralizada.

Durante esta etapa el equipo en muchas ocasiones tuvo que laborar fuera de su horario regular e incluso fines de semanas y días festivos, todo esto demostró el gran compromiso y el trabajo en equipo que se forjó durante el proyecto.

Adicional a ello, adoptar Scrum como marco de trabajo le permitió generar valor mucho más rápido y poder evidenciarlo en las revisiones después de cada sprint.

3.3 Evaluación

A continuación, se detalla la evaluación realizada con respecto al proyecto.

3.3.1 Evaluación económica

Se desarrolló el proyecto durante aproximadamente 9 meses y fue conformado por 6 perfiles profesionales, en distintos momentos, en la siguiente tabla se detallan las horas dedicadas al proyecto, el coste por hora y el coste total del equipo.

Tabla 15

Costo del Equipo del Proyecto

Posición	Recurso	Horas Hombre	Costo x Hora (soles)	Costo Total (soles)
1 Scrum Máster	Interno	450 h	S/ 70	S/ 31 500
1 Product Owner	Interno	800 h	S/ 75	S/ 60 000
1 Arquitecto de Soluciones	Externo	600 h	S/ 55	S/ 33 000
1 líder técnico	Externo	1100 h	S/ 44	S/ 48 400
1 Desarrollador Fullstack	Externo	1512 h	S/ 35	S/ 52 920
1 Analista de Calidad	Externo	850 h	S/ 31	S/ 26 350
Total				S/ 252 170

Nota. Adaptado de la Empresa Aseguradora.

Adicional se detallan los costes relacionados al despliegue de la solución en la infraestructura de la Empresa Aseguradora.

Tabla 16

Costo del despliegue de la Infraestructura

Descripción	Costo x Hora (soles)	Costo Total (soles)
Azure Active Directory	-	S/ 20 000
Infraestructura Cloud Azure	-	S/ 14 000
Analista DevOps	S/ 40	S/ 480
Total		S/ 34 480

Nota. Adaptado de la Empresa Aseguradora.

Antes del desarrollo del proyecto el área de seguridad informática había realizado un análisis de riesgo de las aplicaciones BackOffice de la Empresa Aseguradora, llegando a las siguientes conclusiones:

Tabla 17

Pérdida económica estimada por aplicación

Aplicación	Nivel de Riesgo	Amenaza	Pérdida económica estimada (soles)
BackOffice Vida	Crítica	Acceso no autorizado	S/ 250 000
BackOffice Viajes	Crítica	Acceso no autorizado	S/ 120 000
BackOffice Hogar	Crítica	Acceso no autorizado	S/ 85 000
BackOffice Autos	Crítica	Acceso no autorizado	S/ 130 000
BackOffice Canales	Crítica	Acceso no autorizado	S/ 65 000
BackOffice Reembolsos	Crítica	Acceso no autorizado	S/ 180 000
BackOffice SOAT	Crítica	Acceso no autorizado	S/ 200 000
Total			S/ 1 030 000

Nota. Adaptado de la Empresa Aseguradora.

La Empresa Aseguradora estructura su sistema de gestión de seguridad de la información bajo el marco de la norma ISO 27001, con la cual se apoyó para la gestión e identificación de riesgos con la norma ISO 27005, estableciendo así el nivel criticidad y las amenazas relacionadas a cada uno de los sistemas BackOffice.

También realizó la clasificación de cada uno de los datos que manejan los sistemas BackOffice, y estableció su importancia para la Empresa Aseguradora en términos de confidencialidad, integridad y disponibilidad. Complementario a este análisis con el apoyo del área de riesgo operacional utilizó la metodología FAIR (Factor Analysis of Information Risk) para cuantificar la pérdida económica que podría darse si una amenaza de los sistemas BackOffice es materializada. Este análisis también consideró las pérdidas económicas relacionadas a las sanciones debido al incumplimiento de la ley de protección

de datos personales y a las directrices de seguridad que debe seguir por parte de la superintendencia de banca y seguros.

3.3.2 Justificación Económica

En la Tabla 18 se muestra la comparación entre el costo total del proyecto y la pérdida económica estimada de las aplicaciones BackOffice críticas.

Tabla 18

Justificación económica del proyecto

Inversión Total del proyecto (soles)	Pérdida económica estimada Total (soles)	Ahorro (soles)
S/ 286 650	S/ 1 030 000	S/743 350

Nota. Adaptado de la Empresa Aseguradora.

Con lo descrito anteriormente, el proyecto se justifica, pues su costo es menor que las pérdidas que pueden ocasionarse debido falta de controles de seguridad y a la materialización de las amenazas en las aplicaciones BackOffice, adicional a ello al ser una solución escalable y reutilizable permitirá que la totalidad de aplicaciones BackOffice de la empresa aseguradora también puedan integrarse, lo que significará también un ahorro de costos para dichas aplicaciones.

3.3.1 Beneficios para la organización

A continuación, se comentan los impactos positivos que tuvo el desarrollo del proyecto en la Empresa Aseguradora:

La solución implementada ha permitido centralizar la autenticación de los colaboradores a través de una herramienta de gestión de identidades y bajo una arquitectura consistente, lo que permitirá a la organización monitorear los accesos de sus

trabajadores y utilizar mecanismos de autenticación más robustos como el doble factor de autenticación o la autenticación biométrica.

También ha permitido ahorrar en costos debido a que es una solución reutilizable y escalable, a la que pueden integrarse todos los sistemas BackOffice de la Empresa Aseguradora.

La integración de las 7 aplicaciones BackOffice críticas como parte del desarrollo del proyecto, ha permitido que la gerencia de seguridad informática de la Empresa Aseguradora pueda establecer una serie de lineamientos de seguridad, enfocados a la autenticación y autorización, uno de los cuales establece que todos los sistemas internos de la organización deben estar integrados al “Sistema de autenticación y Autorización Centralizada”. Como consecuencia de esto se prevé que en un futuro las aplicaciones nuevas y en desarrollo nazcan con el componente de autenticación ya integrado.

Finalmente debido al gran desempeño y resultados del proyecto se han impulsado más iniciativas enfocadas a reforzar los controles de seguridad informática de la organización, como autenticación biométrica para clientes, doble factor de autenticación para corredores de seguros, Autenticación facial para trámites documentarios, etc.

CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

Los resultados de este proyecto demostraron a la Empresa Aseguradora los beneficios de conformar un equipo de trabajo orientado a mejorar la seguridad en los sistemas informáticos.

La participación en este proyecto permitió al autor desarrollar su capacidad de autoaprendizaje, debido al gran reto que significó aprender nuevas tecnologías en un corto tiempo, destacando una virtud muy importante, la perseverancia, por la cual a pesar de que surgieron muchas dificultades en el camino, el proyecto pudo concluirse satisfactoriamente.

Pero lo que más rescata el autor, fue que, debido a esta experiencia y las enseñanzas de su jefe directo en el proyecto, descubrió su gusto y pasión por la seguridad informática, motivo por el cual se especializó en el campo de la seguridad de aplicaciones e identidad digital y ha podido implementar proyectos como plataformas de autenticación de clientes, tokens digitales, procesos de desarrollo seguro de software, diseñar arquitectura seguras y herramientas de monitoreo de seguridad en importantes Bancos del Perú y el extranjero.

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

El proyecto duró aproximadamente 9 meses, dentro de los cuales hubo mucha dedicación y compromiso de los integrantes del proyecto, motivo por el cual se concluyó satisfactoriamente y se obtuvieron los resultados esperados.

A continuación, se detallan las principales conclusiones:

Se implementó el “Sistema de Autenticación y Autorización Centralizada para Colaboradores”, para el cual se utilizó la herramienta de gestión de identidades y acceso (IAM), denominada Azure Active Directory, y el repositorio de roles y funciones de la Empresa Aseguradora, obteniendo así un componente reutilizable por todas las aplicaciones BackOffice, además se incrementó el nivel de seguridad de las 7 aplicaciones BackOffice críticas el proceso permitiendo que las pérdidas económicas ocasionadas por el acceso no autorizado sean menos factibles.

Se realizó un análisis bastante profundo sobre los módulos de autenticación y autorización que utilizaba la Empresa Aseguradora para que la solución implementada pueda ser escalable y reutilizable.

Se utilizó la herramienta de identidad y acceso de la Empresa Aseguradora, Azure Active Directory, la cual es una solución global muy utilizada y difundida, proporcionada por Microsoft y que cuenta con muchas capacidades en temas de identidad.

El sistema se integró al repositorio de roles y funciones de los colaboradores de la Empresa Aseguradora, lo cual permitió la reutilización de componentes y dando uso a todo el potencial de la herramienta de autorización.

Se integraron las 7 aplicaciones catalogadas como críticas por la Empresa Aseguradora, lo cual demandó mucho aprendizaje, pues algunas se habían desarrollado en lenguajes de programación y arquitecturas que no conocía el equipo.

Se incrementó el nivel de seguridad de las 7 aplicaciones BackOffice críticas el proceso permitiendo que las pérdidas económicas ocasionadas por el acceso no autorizado sean menos factibles.

El conocimiento adquirido por el proyecto desarrollado permitió que el equipo sea un referente para los demás en temas de autenticación y autorización.

Finalmente, la utilización de la herramienta identidad y acceso de la Empresa Aseguradora, Azure Active Directory, permitió explorar nuevos mecanismos de autenticación que actualmente son tendencia a nivel global, como la autenticación biométrica que en un futuro cercano pueda implementar la Empresa Aseguradora.

5.2 Recomendaciones

Debido a la gran cantidad de aplicaciones BackOffice de la Empresa Aseguradora, se recomienda preparar un plan de migración evaluando la arquitectura y criticidad de las aplicaciones con la finalidad que, de manera paulatina se integren al componente de autenticación y autorización centralizado.

Explorar todas las capacidades de la herramienta de gestión e identidad de acceso (IAM), Azure Active Directory, con la finalidad de implementar los mecanismos más actuales de autenticación, como la autenticación biométrica o FIDO (Fast Identity Online), los cuales están alineados a las mejores prácticas y estándares de seguridad actuales.

Brindar capacitaciones y charlas a todos los colaboradores de la Empresa Aseguradora a fin de fomentar una conciencia y cultura de seguridad debido a que en gran

parte los incidentes de seguridad informática provienen de los mismos empleados de una organización.

Brindar capacitaciones a todos los equipos responsables de los sistemas BackOffice de la Empresa Aseguradora para que puedan conocer la forma de como integrarse al sistema de Autenticación y Autorización Centralizado y conocer sus beneficios para la organización.

Finalmente se sugiere compartir toda la información del proyecto con las demás Empresas del grupo financiero al que pertenece la Empresa Aseguradora, pues se detectó que muchas de ellas no han actualizado o migrado sus formas de autenticación desde hace años.

5.3 Fuentes de Información

- Auth0. (2022). *What is Authorization?* Obtenido de Auth0: <https://auth0.com/intro-to-iam/what-is-authorization/>
- AWS. (2022). *¿Qué son los microservicios?* Obtenido de Amazon Web Services: <https://aws.amazon.com/es/microservices/>
- Ayras Olano, J. D. (2021). Implementación de un modelo de datos para la identificación de potenciales clientes y optimización del spread comercial en el negocio de cambio de divisas en un entorno Big Data de una entidad bancaria. *Trabajo de Suficiencia Profesional*. UNMSM, Lima, Perú. Obtenido de <https://hdl.handle.net/20.500.12672/17837>
- Chiarelli, A. (28 de Octubre de 2021). *ID Token and Access Token: What's the Difference?* Obtenido de Auth0: <https://auth0.com/blog/id-token-access-token-what-is-the-difference/>
- ESET. (2019). *Eset Security Report*. Eset Security Report.
- Esposito, C., Massimo, F., & Brij Bhooshan, G. (2021). Blockchain-based authentication and authorization for smart city. *Information Processing and Management*, 58.
- IBM. (20 de 04 de 2021). *Seguridad - Identificación y autenticación*. Obtenido de IBM: <https://www.ibm.com/docs/es/ibm-mq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-sec-doc-q009740--htm>
- IBM Security. (2021). *Cost of a Data Breach Report*. IBM Security.
- Levin, G. (26 de Julio de 2019). *4 Most Used REST API Authentication Methods*. Obtenido de RestCase: <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/>
- Mahendrakumaran, K. (22 de Septiembre de 2021). *An Introduction to OAuth 2.0*. Obtenido de Medium: <https://medium.datadriveninvestor.com/an-introduction-to-oauth-2-0-e86d97036542>
- Mendoza Arteaga, A., Bolaños Burgos, F., Cedeño Sarmiento, C., & Saltos Rivas, W. R. (2020). La importancia de la autenticación multifactor para el usuario final en un. *Informática y Sistemas*, 42-51.
- Microsoft. (29 de Diciembre de 2021). *OAuth 2.0 authentication with Azure Active Directory*. Obtenido de Microsoft docs: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/auth-oauth2>
- Microsoft. (06 de Junio de 2022). *¿Qué es Azure Active Directory?* Obtenido de Microsoft Docs: <https://docs.microsoft.com/es-es/azure/active-directory/fundamentals/active-directory-what-is>
- OAuth 2.0. (2012). *The OAuth 2.0 Authorization Framework*. Obtenido de OAuth 2.0: <https://datatracker.ietf.org/doc/html/rfc6749>

- OpenID. (2014). *Open ID Connect*. Obtenido de OpenID: <https://openid.net/connect/>
- OWASP. (2018). *C6: Implementar Identidad Digital*. Obtenido de OWASP Foundation: <https://owasp.org/www-project-proactive-controls/v3/en/c6-digital-identity>
- OWASP. (2021). *OWASP Top 10:2021*. Obtenido de OWASP: <https://owasp.org/Top10/>
- Ruiz Mancilla, R. P. (2022). Implementación de una solución para consumir servicios Rest desde un frontal de oficinas en una entidad financiera. *TRABAJO DE SUFICIENCIA PROFESIONAL*. UNMSM, Lima, Perú. Obtenido de <https://hdl.handle.net/20.500.12672/18003>
- Scrum. (2021). *Scrum Guide*. Obtenido de Scrum org: <https://www.scrum.org/resources/scrum-guide>
- Sealpath. (14 de Abril de 2022). *Cuantificar el coste de una fuga de datos*. Obtenido de Sealpath: <https://www.sealpath.com/es/blog/como-cuantificar-el-coste-de-una-fuga-de-datos-un-caso-practico/>
- Sebastian Castro, F. (2020). Gestión de identidades y accesos unificados. (*Tesina de Licenciatura*). Universidad Nacional de La Plata, La Plata. Obtenido de <http://sedici.unlp.edu.ar/handle/10915/118390>
- Torres, G. (14 de Abril de 2019). *OAuth 2.0, OpenID Connect y JSON Web Tokens (JWT)*. Obtenido de Returngis: <https://www.returngis.net/2019/04/oauth-2-0-openid-connect-y-json-web-tokens-jwt-que-es-que/>

5.4 Glosario de términos

- **Amenaza**

Una amenaza es cualquier acto que aprovecha una vulnerabilidad para amenazar la seguridad de un sistema de información. En otras palabras, puede potencialmente afectar negativamente a ciertos elementos de un sistema.

- **API (Interfaz de Programación de Aplicaciones)**

Es un conjunto de definiciones y protocolos para diseñar e integrar software de las aplicaciones.

- **Audience o audiencia**

Es un reclamo que identifica a los destinatarios de el JWT.

- **Autenticación Multifactor**

La autenticación multifactor, comúnmente conocida por su acrónimo en inglés MFA, es un método de control de acceso en el que a un usuario se le otorga acceso a un sistema solo después de proporcionar dos o más pruebas distintas que permitan corroborar su identidad.

- **Backend**

Área lógica de toda aplicación. Se refiere a la arquitectura interna del sitio que asegura que todos elementos desarrollen la función correcta.

- **Base64:**

Es un método de codificación aplicado a datos binarios para representar la información obtenida en una cadena de caracteres en código ASCII.

- **Claims o reclamos**

Son atributos en forma de clave/valor que existen dentro de un JWT.

- **Endpoint o punto final API**

Es una URL que recibe una entrada de datos, realiza un proceso informático y produce una salida de datos.

- **Frontend**

Es la parte que ve el usuario y en la que se incluyen los estilos, los colores, los fondos, tamaños y las animaciones del sitio web.

- **Issuer**

Es un claim o reclamo que identifica a la entidad que emitió el JWT.

- **JWT**

Es un objeto JSON (Javascript Object Notation) que se usan para enviar información entre partes, normalmente un cliente web y un servidor, de manera segura.

- **OAuth:**

Es un estándar abierto que permite flujos simples de autorización para sitios web o aplicaciones informáticas.

- **OpenID Connect**

Protocolo de identidad y de estándar abierto basado en el protocolo OAuth 2.0 que brinda la posibilidad a las aplicaciones cliente delegar la autenticación a un proveedor de identidad.

- **Riesgo**

Es la posibilidad que una amenaza pueda materializarse y ocasionando una serie de daños.

- **SaaS**

Software que reside en la nube y que es desarrollado por un proveedor, el cual se encarga de mantenerlo actualizado y lo brinda a disposición de sus clientes a través de Internet mediante un sistema de suscripción.

- **Scope o Alcance**

Es un mecanismo en OAuth 2.0 para limitar los permisos que un usuario brinda a una aplicación.

- **Sistema BackOffice**

Sistema de información que permite al usuario realizar todas las actividades relacionadas con la gestión interna de la empresa. Es decir, tareas administrativas no relacionadas con las ventas o la atención directa al cliente.

- **SPA (Single Page Application o Aplicación de Página Única)**

Es un tipo de aplicación web que ejecuta todo su contenido en una sola página. Funciona cargando el contenido HTML, CSS y JavaScript por completo al abrir la web.

- **Squad**

Término acuñado por la empresa Spotify para hacer referencia al Scrum Team, hoy en día es un término comúnmente utilizado en todas las empresas que practican el marco de trabajo Scrum.

- **Token**

Es una cadena de caracteres que ha sustituido sus datos sensibles por datos no sensibles mediante un proceso de tokenización.

- **Vulnerabilidad**

Debilidad de un sistema que lo faculta de ser dañado, comprometiendo su la integridad, privacidad y disponibilidad de sus datos u otros sistemas asociados.