

Área de Conocimiento de Tecnología de la
Información y Comunicación

Desarrollo del Sistema Web Helpdesk para la División de Tecnología de la Información de la Universidad Nacional de Ingeniería

**Trabajo Monográfico para optar al título de
Ingeniero en Computación.**

Elaborado por:

Br. Ingrid Jahoska
Trujillo Maltez
Carnet: 2016-0585U

Br. Steven Alexander
Mendez Paiz
Carnet: 2019-0048U

Tutor:

Msc. Ing. Luis Eduardo
Chávez Mairena



Área de Conocimiento de
Tecnología de la Información
y Comunicación

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DEL CONOCIMIENTO DE
TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

TRUJILLO MALTEZ INGRID JAHOSKA

Carné: **2016-0585U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los doce días del mes de junio del año dos mil veinte y cuatro.

Atentamente,

Ing. Cedrick Elksnherr DallarTorre Parrales
SECRETARIO DE ÁREA ACADÉMICA



Móvil: (505) 8588 8333



Recinto Universitario Simón Bolívar
Avenida Universitaria.
Managua, Nicaragua.
Apdo: 5595



www.unf.edu.ni

IMPRESO POR SISTEMA DE REGISTRO ACADEMICO EL 12-jun. 2024



Área de Conocimiento de
Tecnología de la Información
y Comunicación

SECRETARIA DE ÁREA ACADÉMICA

F-8: CARTA DE FINALIZADO PLAN DE ASIGNATURA

El Suscrito Secretario del **ÁREA DEL CONOCIMIENTO DE
TECNOLOGÍA DE LA INFORMACIÓN Y COMUNICACIÓN** hace constar que:

MÉNDEZ PAIZ STEVEN ALEXANDER

Carné: **2019-0048U** Turno: **Diurno** Plan de Asignatura: **2015** de conformidad con el Reglamento Académico vigente en la Universidad, ha aprobado todas las asignaturas correspondientes a la carrera de **INGENIERÍA EN COMPUTACIÓN**, en el año 2023 y solo tiene pendiente la realización de una de las formas de culminación de estudio.

Se extiende la presente **CARTA DE FINALIZADO PLAN DE ASIGNATURA**, a solicitud del interesado en la ciudad de Managua, a los doce días del mes de junio del año dos mil veinte y cuatro.

Atentamente,

Ing. Cedrick Elksnherr DallaTorre Parrales
SECRETARIO DE ÁREA ACADÉMICA



Móvil: (505) 8588 8333



Recinto Universitario Simón Bolívar
Avenida Universitaria,
Managua, Nicaragua.
Apdo: 5595



www.uni.edu.ni

IMPRESO POR SISTEMA DE REGISTRO ACADEMICO EL 12-jun.-2024

Managua, 26 de noviembre de 2024.

MSc. Claudia Benavidez Rugama

**Directora Área de Conocimiento de Tecnología de la Información y
Comunicación
DACTIC**

Su despacho

Reciba un cordial saludo,

A través de la presente, me dirijo a usted, para hacerle de su conocimiento que los bachilleres Ingrid Jahoska Trujillo Maltez con numero de carne 2016-0558U y Steven Alexander Méndez Paiz con carne 2019-0048U, quienes realizaron el trabajo monográfico titulado **Desarrollo del Sistema Web Heldesk para la División de Tecnología de Información de la Universidad Nacional de Ingeniería**, se encuentran preparados para que le programe la fecha de la defensa final.

Sin más a que referirme, esperando una pronta respuesta a dicha solicitud, me despido agradeciendo de antemano la atención prestada.

Atentamente,



MSc. Luis Eduardo Chávez Mairena
Tutor del trabajo monográfico.

Cc. Parte Interesada



Área de Conocimiento de Tecnología de la Información y Comunicación

Managua, 18 de julio 2024

Bachilleres

Br. Ingrid Jahoska Trujillo Maltéz 2016-0585U

Br. Steven Alexander Méndez Paiz 2019-0048U

Egresados Programa académico Ingeniería en Computación

Sus manos.

Estimados Bachilleres:

Reciban cordiales saludos de mi parte y deseándole el mejor de los éxitos en sus actividades diarias.

Por medio de la presente, les comunico la aprobación e inscripción del Protocolo de trabajo monográfico, titulado: **"Desarrollo del sistema web Helpdesk para la División de Tecnología de la Información de la Universidad Nacional de Ingeniería"**, el cual cumple con los requisitos establecidos en el capítulo II de la normativa para los trabajos Monográficos de la Universidad Nacional de Ingeniería (UNI) como forma de culminación de estudios.

No omito manifestar, que el Maestro **MSc. Luis Eduardo Chávez Mairena**, es el docente encargado de acompañarlos con responsabilidad y compromiso como tutor en el proceso de desarrollo de la monografía.

Así mismo, en correspondencia con la Normativa para los Trabajos Monográficos, a partir de la fecha de aprobación e inscripción tendrán un máximo de 12 meses para la ejecución y entrega de los ejemplares para la organización de la pre-defensa y defensa del mismo.

Sin más a que hacer referencia, les deseo el mejor de los éxitos en la culminación de esta etapa, les saludo.

Atentamente,

MSc. Claudia Benavidez Rugama

Directora Área de Conocimiento de
Tecnología de la Información y Comunicación



CC: MSc. Luis Eduardo Chávez Mairena – Tutor

MSc. Cedrick DallaTorre Parrales – Secretario Académico

Archivo DACTIC 2024



Móvil: (505) 8588 8333



Recinto Universitario Simón Bolívar
Avenida Universitaria,
Managua, Nicaragua.
Apdo: 5595



www.uni.edu.ni

DEDICATORIA

En primer lugar, expresamos nuestra profunda gratitud a Dios, quien ha sido nuestra guía constante y la fuente de fortaleza en cada etapa de este enriquecedor viaje académico. Su amor inquebrantable y su infinita sabiduría nos han inspirado a enfrentar y superar los retos que se nos han presentado, permitiéndonos alcanzar nuestras metas con determinación y esfuerzo.

A nuestras queridas familias, les dedicamos nuestro más sincero agradecimiento por su apoyo constante, amor y paciencia a lo largo de este proceso. Sin su aliento y los sacrificios que han realizado. Su confianza en nosotros ha sido un pilar fundamental que nos ha impulsado a seguir adelante.

Queremos también expresar nuestra gratitud a la Universidad, por brindarnos un ambiente de aprendizaje estimulante y por su compromiso con la excelencia académica. Las oportunidades de desarrollo personal y profesional que nos han ofrecido han sido clave en nuestra formación y nos han preparado para enfrentar los desafíos del futuro.

Expresamos un agradecimiento especial al Ing. Manuel Alejandro Godínez Duriez, director de la División de Tecnología de la Información, por su liderazgo excepcional y su apoyo constante en nuestras iniciativas académicas. Su visión innovadora y su dedicación nos han motivado a superarnos y a alcanzar nuevos horizontes.

De igual manera, extendemos nuestro reconocimiento al Msc. Ing. Luis Eduardo Chávez Mairena, nuestro tutor monográfico, cuya orientación experta y valiosos consejos han sido imprescindibles para el desarrollo y la culminación de este trabajo. Su dedicación a nuestra formación ha dejado una huella indeleble en nuestro aprendizaje.

Finalmente, queremos reconocer al Ing. Rodrigo Ignacio Díaz Briceño, responsable del Área de Sistemas, y al Ing. Carlos Emilio Rodríguez Parajon, responsable del Área de Infraestructura, por su entrega y colaboración incondicional en este proyecto.

RESUMEN

El presente trabajo monográfico aborda el desarrollo de un Sistema Web Helpdesk para mejorar la gestión de incidencias y solicitudes de soporte en la División de Tecnología de la Información (DTI) de la Universidad Nacional de Ingeniería (UNI). El proyecto surge a partir de la necesidad de optimizar los procesos de soporte técnico debido a la creciente complejidad de las gestiones sin un sistema centralizado eficiente. El objetivo principal es desarrollar una plataforma personalizada que se ajuste a los requerimientos específicos de la DTI, mejorando la eficiencia operativa y la experiencia del usuario final.

El enfoque metodológico adoptado es el modelo en cascada, que permite una planificación detallada y estructurada desde la fase de requisitos hasta el despliegue. Durante el desarrollo, se emplearon tecnologías modernas como React, C#, y SQL Server para garantizar la escalabilidad y robustez del sistema. El sistema permite gestionar tickets, asignar prioridades, generar reportes, y optimizar la distribución de tareas, ofreciendo una solución integral que supera las limitaciones de herramientas previas como Redmine y OTRS.

Los resultados obtenidos demostraron que el nuevo sistema facilita una gestión más eficiente de incidencias, reduciendo los tiempos de resolución y mejorando la satisfacción de los usuarios. Se concluye que la implementación de este sistema es factible tanto técnica como económicamente, y se recomienda su adopción institucional.

Palabras clave: Sistema Helpdesk, gestión de incidencias, tecnología de la información, DTI, Universidad Nacional de Ingeniería.

INDICE

I. Introducción.....	1
II. Antecedentes	2
III. Justificación.....	3
IV. Objetivos	4
4.1 Objetivo General	4
4.2 Objetivos Específicos.....	4
V. Marco teórico conceptual	5
5.1 Conceptos Básicos	6
5.1.1 Sistemas de Información.....	6
5.2 Sistema informático	8
5.2.1 Diferencia entre Sistema de Información y Sistema Informático.....	8
5.2.2 Helpdesk.....	9
5.3 Herramientas de desarrollo.....	10
5.3.1 Frontend	11
5.3.1.1 HTML	11
5.3.2 Backend.....	13
5.4 Versionado	16
5.4.1 Git.....	16

5.4.2	Gitlab	16
5.5	Infraestructura y Despliegue	16
5.5.1	Docker	17
5.5.2	Kubernetes	17
5.6	Metodología de Desarrollo de Software	17
5.6.1	Metodología	17
5.6.2	Metodología Cascada	18
VI.	Análisis y presentación de resultados	21
6.1	Diseño Metodológico	21
6.1.1	Adaptación al Modelo en Cascada	21
6.2	Fase de Requisitos	27
6.2.1	Requerimientos Funcionales	27
6.2.2	Requerimientos No funcionales	37
6.3	Fase de Diseño	38
6.3.1	Arquitectura del Sistema - Clean Architecture	38
6.4	Fase de Desarrollo	81
6.4.1	Preparación del Entorno de Desarrollo	81
6.4.2	Implementación de los Módulos	82
6.5	Fase de Pruebas y Validación	88

6.5.1	Objetivos de la Fase de Pruebas	88
6.5.2	Estrategia de Pruebas	88
6.6	Fase de Despliegue	94
6.6.1	Preparación del Entorno de Producción	94
6.6.2	Configuración del Servidor Frontend	94
6.6.3	Configuración del Servidor Backend.....	95
6.6.4	Ajustes Finales y Optimización	95
6.6.5	Diagrama de Despliegue	96
6.6.6	Integración y Entrega continua.....	97
6.6.7	Capacitación del Personal	97
VII.	Conclusiones y recomendaciones	98
7.1	Conclusiones	98
7.2	Recomendaciones	99
VII.	Bibliografía.....	100
IX.	Anexos	105
9.1	Diccionario De Datos	105
9.2	Hojas Asistencia De Capacitación	111
9.3	Estudio de Factibilidad.....	116
9.3.1	Factibilidad Técnica.....	116

9.3.2 Factibilidad Económica	117
9.3.3 Factibilidad Operativa	119
9.3.4 Evaluación de Riesgos:.....	119
9.3.5 Estudio de Factibilidad Legal	120
9.4 cronograma.....	122

I. INTRODUCCIÓN

El presente documento tiene como propósito principal documentar el proceso completo de desarrollo del **Sistema Web HelpDesk**, desde su concepción hasta su implementación, poniendo especial énfasis en las necesidades específicas identificadas por la DTI.

En la actualidad, la **Universidad Nacional de Ingeniería** (UNI) enfrenta dificultades respecto a la gestión de incidencias y solicitudes de soporte en un entorno donde la tecnología de la información desempeña un papel cada vez más crucial. Consciente de esto, la **División de Tecnología de Información** (DTI) reconoce la importancia de implementar un Sistema Web HelpDesk adaptado a las particularidades de la comunidad académica y administrativa de la institución.

A lo largo de este trabajo, se explorarán en detalle diversos aspectos del desarrollo del Sistema Web HelpDesk, tales como la identificación de requisitos, el diseño de la arquitectura del sistema, la selección de tecnologías apropiadas, el proceso de implementación y las pruebas de validación.

El presente documento se estructura en varios apartados que detallan el desarrollo del Sistema Web HelpDesk para la División de Tecnología de Información de la Universidad Nacional de Ingeniería. Se inicia con una sección de introducción, antecedentes y justificación que respaldan la necesidad de este sistema. A continuación, se establecen los objetivos generales y específicos del proyecto. La parte central del documento está dedicada al marco teórico, que aborda conceptos básicos. Posteriormente, se describe el diseño metodológico, adoptando el modelo en cascada, y se presenta un cronograma detallado de actividades. Finalmente, se incluye una bibliografía que respalda los conceptos y metodologías utilizados en el desarrollo del Sistema Web Helpdesk.

II. ANTECEDENTES

La División de Tecnología de la Información (DTI) de la Universidad Nacional de Ingeniería tiene como misión garantizar la operatividad tecnológica de la institución, proporcionando soporte técnico y herramientas que optimicen los procesos internos. Sin embargo, el manejo de proyectos e incidencias ha ganado complejidad debido a la falta de una solución centralizada y eficiente, lo que ha impulsado la búsqueda de un sistema de Helpdesk adaptado a las necesidades específicas del departamento.

En el pasado, la DTI utilizó herramientas como **Redmine** y **OTRS**, que cumplieron roles clave, pero presentan limitaciones importantes:

Redmine es una plataforma de código abierto para la gestión de proyectos, utilizada para coordinar tareas mediante funcionalidades como wikis, foros y control de acceso basado en roles. A pesar de sus aportes en la organización, presenta limitaciones significativas:

- No se adapta eficientemente al crecimiento de proyectos e incidencias.
- Su enfoque en la gestión de proyectos complica su uso para soporte técnico.
- Ofrece opciones limitadas de personalización.
- Los reportes generados son vulnerables a modificaciones

(Overview - Redmine, s.f.).

OTRS es un sistema de gestión de servicios de TI que centraliza y automatiza el soporte técnico, destacándose por el seguimiento detallado de tickets, la automatización de procesos y la generación de reportes analíticos. Sin embargo:

- Posee una interfaz poco intuitiva.
- Ofrece opciones de personalización moderadas, insuficientes para las necesidades específicas de la DTI.

(Rengifo, 2023)

III. JUSTIFICACIÓN

La implementación del **Sistema Web Helpdesk** para la **División de Tecnología de la Información** (DTI) de la **Universidad Nacional de Ingeniería** (UNI) responde a la necesidad de mejorar la gestión de incidencias y el soporte técnico, superando las limitaciones de herramientas previas como **Redmine** y **OTRS**. Esta justificación se fundamenta en los siguientes puntos clave:

- **Optimización de Procesos Internos:** Un sistema Helpdesk diseñado específicamente para la DTI permitirá centralizar y agilizar el registro, asignación, seguimiento y resolución de incidencias. Esto reducirá tiempos de respuesta y mejorará la eficiencia operativa, alineándose con las necesidades de la UNI.
- **Adaptación a Requisitos Específicos:** Las herramientas genéricas del mercado no satisfacen completamente los flujos de trabajo y requerimientos de la DTI. Un desarrollo propio garantiza funcionalidad personalizada, reportes seguros y flexibles, y una mayor integración con los procesos institucionales.
- **Interfaz Intuitiva:** Una plataforma fácil de usar fomenta su adopción por parte del personal técnico y administrativo, mejorando la experiencia tanto de los operadores como de los usuarios finales al proporcionar un canal accesible y eficiente para reportar incidencias.
- **Análisis de Rendimiento:** El sistema permitirá recopilar y analizar datos clave sobre tiempos de resolución, patrones de incidencias y desempeño del personal. Esto facilitará la toma de decisiones basadas en evidencia, identificando oportunidades para optimizar el soporte técnico.

En resumen, el desarrollo de un Sistema Web Helpdesk para la DTI de la UNI permitirá una gestión más eficiente y adaptada de incidencias, mejorando tanto los procesos internos como la experiencia de los usuarios, a la vez que se fortalece la capacidad de análisis y planificación estratégica.

IV. OBJETIVOS

4.1 OBJETIVO GENERAL

Desarrollar un sistema web de Helpdesk para la División de Tecnología de Información de la Universidad Nacional.

4.2 OBJETIVOS ESPECÍFICOS

1. Analizar los requerimientos funcionales y no funcionales del Sistema Web Helpdesk.
2. Diseñar una interfaz de usuario que cumpla con los requerimientos del Sistema Web Helpdesk.
3. Codificar el módulo de tickets para el Sistema Web Helpdesk utilizando los lenguajes de programación JavaScript y C#, junto con el motor de base de datos SQL Server.
4. Implementar un sistema de control de acceso para garantizar la privacidad y restricción de funciones según el rol de cada usuario.
5. Implantar el Sistema Web en la División de Tecnología de la Información para su puesta en producción.

V. MARCO TEÓRICO CONCEPTUAL

En esta sección, se explorarán los fundamentos teóricos que respaldan el desarrollo del sistema web Helpdesk para la División de Tecnología de Información, centrándonos en conceptos básicos, herramientas de desarrollo, y control de versiones, así como la metodología de desarrollo de software a utilizar. Estos elementos proporcionan el marco conceptual necesario para comprender el entorno en el que se desenvuelve el proyecto en cuestión.

En primer lugar, se examinarán los conceptos fundamentales relacionados con los sistemas de información y los sistemas informáticos. Se definirán y diferenciarán estos términos clave, así como se explorará el papel de los sistemas Helpdesk.

Posteriormente, se analizarán las herramientas de desarrollo utilizadas en la creación de sistemas de información. Se detallarán tanto el frontend como el backend, así como se abordará la importancia de las bases de datos en el almacenamiento y gestión de la información.

Continuando, se explorarán aspectos cruciales del control de versiones y la gestión del proyecto, incluyendo infraestructura y despliegue de aplicaciones. Se examinarán tecnologías que juegan un papel fundamental en la creación y mantenimiento de entornos de desarrollo y producción estandarizados y eficientes.

Finalmente, se presentarán la metodología de desarrollo de software que guía el proceso de desarrollo del sistema, facilitando la organización, planificación y ejecución de proyectos de manera efectiva y colaborativa.

Mediante este análisis de los aspectos teóricos relacionados con los sistemas de información y el desarrollo de software, se sentará una base sólida permitiendo una comprensión profunda del proyecto.

5.1 CONCEPTOS BÁSICOS

5.1.1 Sistemas de Información

Un sistema de información es una estructura organizada que recopila, procesa, almacena y distribuye datos de manera eficiente para ayudar en la toma de decisiones y en la gestión de una organización. Este sistema está compuesto por personas, tecnología y procesos que trabajan en conjunto para gestionar la información de la manera más efectiva posible, permitiendo a la organización obtener conocimientos útiles y tomar decisiones informadas.

"Sistemas de Información (SI) Conjunto de componentes interrelacionados que reúnen, procesan, almacenan y distribuyen datos e información y proporcionan un mecanismo de retroalimentación con el fin de cumplir un objetivo." (M. Stair & Reynolds, 2010)

5.1.1.1 Componentes de los Sistemas de Información

Los sistemas de información constan de varios componentes que trabajan en conjunto para recopilar, procesar, almacenar y distribuir datos e información. Estos componentes suelen incluir:

- **Hardware:** Son los dispositivos físicos que forman parte del sistema, como computadoras, servidores, dispositivos de almacenamiento, impresoras, escáneres, etc.
- **Software:** Se refiere a los programas y aplicaciones que permiten el funcionamiento del sistema, incluyendo sistemas operativos, software de base de datos, aplicaciones específicas para procesamiento de datos, entre otros.
- **Base de Datos:** Es donde se almacenan los datos de manera organizada y estructurada, permitiendo su acceso, consulta y actualización. Puede ser desde una simple hoja de cálculo hasta una compleja base de datos relacional.
- **Redes de Comunicación:** Son los medios a través de los cuales se transmiten los datos e información dentro y fuera del sistema, como internet, redes locales (LAN), redes de área amplia (WAN), entre otros.

- **Procedimientos:** Son las reglas y pasos definidos para el uso y operación del sistema, incluyendo políticas de seguridad, procedimientos de respaldo de datos, protocolos de acceso, entre otros.
- **Personal:** Son las personas que interactúan con el sistema, incluyendo usuarios finales, administradores de sistemas, personal de soporte técnico, entre otros.
- **Datos e Información:** Son el elemento fundamental de un sistema de información. Los datos son hechos crudos y sin procesar, mientras que la información es el resultado de procesar y organizar esos datos de manera significativa para la toma de decisiones.

Estos componentes interactúan entre sí de manera coordinada para lograr los objetivos del sistema de información y satisfacer las necesidades de la organización. (C. Laudon & P. Laudon, 2012)

5.1.1.2 Tipos de Sistemas de Información

Los sistemas de información pueden clasificarse en varios tipos según sus funciones y objetivos.

- **Sistemas de Información Transaccionales (SIT):** Registran y procesan transacciones diarias, como ventas y compras.
- **Sistemas de Soporte a la Toma de Decisiones (DSS):** Ayudan a los gerentes en decisiones no estructuradas o semiestructuradas mediante análisis de datos.
- **Sistemas de Información Ejecutiva (EIS):** Proporcionan información resumida para ejecutivos y altos directivos.
- **Sistemas de Apoyo a Procesos de Negocio (BPM):** Automatizan y gestionan procesos empresariales, como CRM y ERP.
- **Sistemas de Información Colaborativos (CIS):** Facilitan la comunicación y colaboración entre individuos y grupos.

(KYOCERA, 2024)

5.2 SISTEMA INFORMÁTICO

Sistema Informático es un término que se refiere a un conjunto interrelacionado de componentes hardware, software, datos y procedimientos, diseñados para procesar y gestionar información de manera eficiente. Estos sistemas están diseñados para realizar funciones específicas de procesamiento, almacenamiento, recuperación y transmisión de datos en una organización o entorno informático. (Entendiendo el entorno digital - Aspectos técnicos, s.f.)

5.2.1 Diferencia entre Sistema de Información y Sistema Informático

La diferencia entre un Sistema de Información y un Sistema Informático radica principalmente en su enfoque y alcance:

Sistema de Información: Se centra en la gestión de datos e información en el contexto de una organización. Este sistema abarca no solo los componentes tecnológicos como hardware y software, sino también los procesos, las personas y la estructura organizacional que trabajan juntos para recopilar, procesar, almacenar y distribuir información de manera eficiente para facilitar la toma de decisiones y la gestión de la organización.

Sistema Informático: Se refiere específicamente al conjunto de componentes tecnológicos, como hardware, software, datos y procedimientos, diseñados para procesar y gestionar información de manera eficiente. Aunque los sistemas informáticos forman parte integral de los sistemas de información, su enfoque se limita más a los aspectos técnicos y tecnológicos de la gestión de la información.

En resumen, mientras que un Sistema de Información abarca un enfoque más holístico que incluye aspectos tecnológicos y organizacionales, un Sistema Informático se centra específicamente en los aspectos tecnológicos y técnicos de la gestión de la información.

5.2.2 Helpdesk

Un sistema de Helpdesk es una herramienta de soporte técnico diseñada para gestionar y resolver eficientemente las consultas, problemas y solicitudes de los usuarios. Estos sistemas son fundamentales para organizaciones que ofrecen servicios de atención al cliente, ya que ayudan a mejorar la comunicación, la productividad y la satisfacción del cliente. A continuación, se presentan algunas funciones típicas de un sistema de Helpdesk:

1. **Registro de Tickets:** Permite a los usuarios registrar consultas, problemas o solicitudes de soporte técnico en el sistema.
2. **Seguimiento de Tickets:** Facilita el seguimiento y la gestión de tickets desde su creación hasta su resolución, incluyendo la asignación de responsabilidades y la actualización del estado del ticket.
3. **Priorización de Tickets:** Permite la asignación de prioridades a los tickets según su impacto en el negocio y su urgencia.
4. **Generación de Informes y Métricas:** Ofrece capacidades de generación de informes para monitorear el desempeño del equipo de soporte técnico, identificar tendencias y áreas de mejora.

En resumen, un sistema de Helpdesk permite a los usuarios registrar consultas, problemas o solicitudes de soporte técnico, facilita el seguimiento y gestión de tickets desde su creación hasta su resolución, incluyendo la asignación de responsabilidades y la actualización del estado del ticket, además de la priorización de tickets según su impacto en el negocio y urgencia. También ofrece capacidades de generación de informes y métricas para monitorear el desempeño del equipo de soporte técnico, identificar tendencias y áreas de mejora. (Schlicht, 2019)

5.2.2.1 Importancia en la Organización

Un sistema de Helpdesk es esencial para cualquier organización, ya que actúa como un punto centralizado de contacto y soporte para los usuarios. Permite a los clientes o empleados reportar problemas técnicos y recibir asistencia de manera rápida y eficiente, lo que mejora significativamente su experiencia y satisfacción.

Al mismo tiempo, optimiza los recursos de soporte al asignarlos de manera eficiente, priorizando y distribuyendo los tickets según su urgencia y complejidad. Esto garantiza que los problemas críticos se aborden rápidamente, mejorando la eficiencia operativa de la organización en general.

Por otro lado, el Helpdesk promueve la transparencia y la responsabilidad dentro del equipo de soporte técnico al mantener un registro detallado de todas las interacciones de soporte. Esto facilita la identificación de áreas de mejora y la optimización de procesos, contribuyendo a un servicio de soporte más efectivo y satisfactorio para los usuarios. (Schlicht, 2019)

5.3 HERRAMIENTAS DE DESARROLLO

En esta sección, nos adentraremos en el entorno de desarrollo del proyecto, explorando las diversas herramientas y tecnologías utilizadas tanto en el Frontend como en el Backend, así como aquellas relacionadas con la gestión de la Base de Datos. Analizaremos en detalle cómo estas herramientas contribuyen al desarrollo y funcionamiento del sistema, proporcionando una visión integral de la infraestructura tecnológica empleada en el proyecto.

5.3.1 Frontend

El término "Frontend" se refiere a la parte de un software o aplicación web que se encuentra en contacto directo con los usuarios. Es la interfaz de usuario visible y accesible a través de la cual los usuarios interactúan con el sistema.

El desarrollo del Frontend implica la creación de interfaces de usuario atractivas y fáciles de usar, así como la presentación efectiva del contenido para garantizar una experiencia de usuario fluida en diversos dispositivos. Sus funciones principales incluyen el diseño de interfaces intuitivas, la implementación de interactividad y la optimización del rendimiento para ofrecer una experiencia óptima al usuario final. (Lemonaki, 2022)

5.3.1.1 HTML

HTML (HyperText Markup Language) es el lenguaje estándar utilizado para crear y diseñar páginas web. Se utiliza para estructurar el contenido de una página web, definiendo diferentes elementos como encabezados, párrafos, listas, enlaces, imágenes, formularios, entre otros. HTML utiliza una sintaxis de marcado que consiste en etiquetas rodeadas por corchetes angulares, las cuales indican al navegador web cómo mostrar el contenido. (Robbins, 2012)

5.3.1.2 CSS

CSS (Cascading Style Sheets) es un lenguaje de diseño utilizado para controlar el aspecto visual y la presentación de las páginas web que se crean con HTML. Mientras que HTML se utiliza para estructurar el contenido de una página web, CSS se utiliza para definir cómo se ve ese contenido.

Con CSS, los desarrolladores pueden definir propiedades como colores, fuentes, márgenes, espaciados, tamaños de texto, diseños de caja y muchas otras características visuales. Además, CSS permite la creación de diseños responsivos, lo que significa que las páginas web pueden adaptarse y mostrar correctamente en diferentes dispositivos y tamaños de pantalla. (Robbins, 2012)

5.3.1.3 JavaScript

JavaScript es un lenguaje de programación de alto nivel que se utiliza principalmente para crear contenido interactivo en páginas web. Es conocido por ser un lenguaje de programación interpretado, lo que significa que se ejecuta directamente en el navegador web del usuario sin necesidad de compilación. (Crockford)

JavaScript es esencial debido a su capacidad para hacer que las páginas sean interactivas y dinámicas. Desde validar formularios hasta crear efectos visuales llamativos y comunicarse con el servidor para cargar datos de manera asíncrona, JavaScript proporciona la base para una experiencia de usuario moderna y envolvente en la web.

5.3.1.4 React

React es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y dinámicas. Se destaca por su enfoque en la creación de componentes reutilizables que representan diferentes partes de la interfaz de usuario. React utiliza un modelo de programación declarativa, lo que significa que los desarrolladores describen cómo debería lucir la interfaz en función del estado de la aplicación y React se encarga de actualizar eficientemente esos cambios. (React, s.f.)

Funciona como una herramienta fundamental para el desarrollo de aplicaciones web modernas y se utiliza en una variedad de casos de uso, desde la creación de sitios web simples hasta aplicaciones web complejas de una sola página (SPA).

5.3.1.5 Bootstrap

Bootstrap es un framework de desarrollo web Frontend que ofrece un conjunto de herramientas y componentes para simplificar la creación de sitios y aplicaciones web responsivas y móviles. Con su sistema de rejilla adaptable, permite diseñar interfaces flexibles que se ajustan automáticamente a diferentes dispositivos. (Bootstrap, Otto, & Thornton, s.f.)

Además, ofrece una amplia gama de componentes predefinidos y plugin JavaScript que facilitan la integración de funciones interactivas y dinámicas. Bootstrap es altamente personalizable, lo que permite a los desarrolladores adaptar fácilmente el aspecto y la funcionalidad para satisfacer las necesidades específicas de sus proyectos, convirtiéndolo en una opción popular y poderosa en el desarrollo web moderno.

5.3.2 Backend

Sus funciones principales incluyen la gestión de datos mediante la manipulación de la base de datos, la implementación de la lógica de negocio para procesar solicitudes de usuarios, la garantía de la seguridad de los datos y del sistema, la integración con servicios externos y la optimización del rendimiento para ofrecer una experiencia eficiente y fiable. (Lemonaki, 2022)

5.3.2.1 C Sharp

C# es un lenguaje de programación desarrollado por Microsoft como parte de su plataforma .NET. Es un lenguaje orientado a objetos y de propósito general que se utiliza principalmente para desarrollar aplicaciones de escritorio, aplicaciones web, aplicaciones móviles y servicios en la nube. (Albahari, 2021)

Algunos de los usos comunes de C# incluyen el desarrollo de software empresarial, aplicaciones de bases de datos, juegos, aplicaciones móviles para dispositivos Windows, Android y iOS, así como aplicaciones web utilizando tecnologías como ASP.NET.

5.3.2.2 ASP.NET

ASP.NET es un marco de desarrollo web creado por Microsoft para construir aplicaciones web dinámicas y robustas. Ofrece un entorno unificado para desarrollar aplicaciones web utilizando varios lenguajes de programación como C#, Visual Basic.NET, etc. Su integración con tecnologías de Microsoft, características de seguridad integradas y su enfoque en rendimiento y escalabilidad lo convierten en una opción popular para desarrolladores y empresas que buscan crear aplicaciones web de alta calidad. (Price, 2020)

Quienes utilizan ASP.NET son principalmente desarrolladores y empresas que desean construir aplicaciones web, desde sitios simples hasta aplicaciones empresariales complejas. La flexibilidad de ASP.NET para trabajar con diferentes lenguajes de programación y su sólida integración con otras tecnologías de Microsoft lo convierten en una opción atractiva para aquellos que prefieren el ecosistema de desarrollo de Microsoft para sus proyectos web.

5.3.2.3 Entity Framework

Entity Framework es un framework desarrollado por Microsoft que simplifica el acceso y la manipulación de datos en aplicaciones .NET al proporcionar una capa de abstracción sobre la base de datos, permitiendo a los desarrolladores trabajar con datos en forma de objetos y propiedades sin preocuparse por la complejidad del almacenamiento subyacente en una base de datos relacional. (Albahari, 2021)

5.3.2.4 FastReport

FastReport es una herramienta de generación de informes para desarrolladores de software. Permite a los desarrolladores crear rápidamente informes interactivos y ricos en datos para sus aplicaciones. Se utiliza comúnmente en entornos de desarrollo de software para integrar capacidades de generación de informes en aplicaciones empresariales, sistemas de gestión, aplicaciones web y más. (Fast Reports, s.f.)

5.3.3 Base de Datos

Una base de datos es un conjunto organizado de datos relacionados entre sí, estructurados y almacenados de manera que puedan ser accedidos y gestionados de forma eficiente. Estos datos pueden ser de diversos tipos, como texto, números, imágenes, vídeos, etc., y se utilizan para almacenar información sobre un dominio específico, como clientes, productos, transacciones, etc. Las bases de datos son ampliamente utilizadas en aplicaciones informáticas para almacenar, recuperar y manipular información de manera sistemática. (Elmasri & Navathe)

Es importante destacar que las bases de datos son fundamentales en la gestión de la información en prácticamente todos los ámbitos, desde empresas y organizaciones gubernamentales hasta aplicaciones personales. Permiten realizar consultas complejas, garantizar la integridad de los datos, compartir información entre usuarios, y proporcionan un medio eficiente para el almacenamiento y recuperación de datos.

5.3.3.1 SQL Server

SQL Server es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Se utiliza para almacenar y recuperar datos según las solicitudes de otras aplicaciones de software. SQL Server es especialmente popular en entornos empresariales debido a su robustez, escalabilidad y amplia gama de características. (Petkovic, 2016)

5.3.3.2 SQL Server Management Studio

SQL Server Management Studio (SSMS) es una herramienta de administración y desarrollo para el sistema de gestión de bases de datos Microsoft SQL Server. Permite a los usuarios administrar, configurar, desplegar y controlar bases de datos SQL Server. Además, proporciona un entorno integrado para escribir consultas SQL, diseñar bases de datos, gestionar seguridad, realizar copias de seguridad y restauraciones, entre otras funciones. (Microsoft, 2023)

5.4 VERSIONADO

El versionado es el proceso de gestionar y controlar los cambios realizados en el código fuente, documentos u otros elementos de un proyecto a lo largo del tiempo. Permite mantener un historial de las modificaciones, facilitando la colaboración entre los miembros del equipo y asegurando la integridad y trazabilidad del proyecto.

5.4.1 Git

Git es un sistema de control de versiones distribuido de código abierto, diseñado para gestionar proyectos de cualquier tamaño con rapidez y eficiencia. Permite realizar un seguimiento de los cambios en el código, trabajar en ramas independientes, fusionar cambios entre ramas y colaborar con otros desarrolladores de manera efectiva. (Jon Loeliger, 2012)

5.4.2 Gitlab

Gitlab es una plataforma de desarrollo de software colaborativo basada en Git, que ofrece herramientas para la gestión de repositorios de código, seguimiento de problemas, integración continua y despliegue automatizado. Permite a los equipos trabajar de manera eficiente y transparente en proyectos de software. (Gomez, 2022)

5.5 INFRAESTRUCTURA Y DESPLIEGUE

La infraestructura y despliegue se refiere al conjunto de recursos físicos y virtuales necesarios para ejecutar y mantener una aplicación o servicio en funcionamiento. Incluye la configuración de servidores, redes, bases de datos, sistemas de almacenamiento y otros componentes, así como el proceso de implementación y actualización de la aplicación en estos entornos.

5.5.1 Docker

Docker es una plataforma de código abierto que permite empaquetar, distribuir y ejecutar aplicaciones en contenedores ligeros y portátiles. Utiliza tecnologías de virtualización a nivel de sistema operativo para garantizar la consistencia del entorno de ejecución, facilitando la implementación y escalabilidad de aplicaciones en diferentes entornos. (Matthias, 2015)

5.5.2 Kubernetes

Kubernetes es una plataforma de código abierto para la gestión automatizada de aplicaciones en contenedores. Proporciona herramientas para el despliegue, escalado y operación de aplicaciones en clústeres de servidores, facilitando la orquestación de contenedores y la gestión de recursos de manera eficiente y escalable. (Beda, Hightower, & Burns, 2017)

5.6 METODOLOGÍA DE DESARROLLO DE SOFTWARE

En esta sección se describe la metodología a utilizar para el desarrollo del sistema web Helpdesk.

5.6.1 Metodología

La metodología en el contexto del desarrollo de software se refiere al enfoque sistemático y estructurado utilizado para planificar, diseñar, implementar y mantener sistemas de software. Es un conjunto de prácticas, procesos y herramientas que guían el desarrollo de un proyecto desde su inicio hasta su conclusión, con el objetivo de lograr un resultado final de alta calidad que satisfaga las necesidades del cliente.

Una metodología efectiva proporciona un marco de trabajo claro y coherente que ayuda a los equipos de desarrollo a gestionar eficientemente los recursos, minimizar los riesgos y mantener el control sobre el progreso del proyecto. Al establecer pasos definidos y roles específicos para cada fase del ciclo de vida del desarrollo de software, la metodología facilita la colaboración entre los miembros del equipo y fomenta la comunicación clara y transparente.

La elección de la metodología adecuada depende de varios factores, incluyendo las características del proyecto, los requisitos del cliente, los recursos disponibles y las preferencias del equipo de desarrollo. Entre las metodologías más comunes se encuentran el enfoque en cascada, el desarrollo ágil, el modelo en espiral y otros enfoques híbridos que combinan elementos de diferentes metodologías para adaptarse a las necesidades específicas del proyecto.

En el contexto del desarrollo del sistema web Helpdesk para la División de Tecnología de Información de la Universidad Nacional de Ingeniería, la elección de la metodología adecuada es crucial para garantizar el éxito del proyecto y la satisfacción del cliente. A continuación, se detalla la metodología específica seleccionada para este proyecto y cómo se aplicará en cada etapa del desarrollo.

5.6.2 Metodología Cascada

La metodología en cascada, también conocida como modelo en cascada o ciclo de vida en cascada, es uno de los enfoques más tradicionales y secuenciales para el desarrollo de software. Este enfoque se caracteriza por su estructura lineal y ordenada, donde cada fase del proceso debe completarse antes de avanzar a la siguiente. Aunque ha sido objeto de críticas debido a su rigidez frente a cambios y su incapacidad para adaptarse a entornos dinámicos, la metodología en cascada sigue siendo utilizada en una variedad de proyectos, especialmente aquellos con requisitos bien definidos y estables desde el principio.

5.6.2.1 Etapas de la Metodología Cascada

5.6.2.1.1 Requisitos

En esta etapa inicial, se dedica un esfuerzo significativo a la comprensión y documentación exhaustiva de los requisitos del sistema. Esto implica la interacción cercana con los usuarios finales y las partes interesadas para capturar sus necesidades y expectativas. Los requisitos se documentan en detalle en un documento de especificación de requisitos, que servirá como base para todas las etapas subsiguientes del proyecto.

5.6.2.1.2 Diseño

Con los requisitos claros, se procede al diseño del sistema. En esta fase, se define la arquitectura general del sistema, incluidos los componentes de hardware y software necesarios. Se establecen las interfaces de usuario, los diagramas de flujo de datos y se diseñan las bases de datos que respaldarán la aplicación. Este diseño se presenta en documentos técnicos y diagramas que proporcionan una visión clara de la estructura y funcionalidad del sistema.

5.6.2.1.3 Implementación

Una vez completado el diseño, se pasa a la fase de implementación, donde se traducen los diseños en código ejecutable. Los programadores escriben el código siguiendo las especificaciones establecidas en la fase de diseño. Cada módulo del sistema se desarrolla y prueba de manera independiente para garantizar su funcionamiento correcto antes de integrarlo con el resto del sistema.

5.6.2.1.4 Pruebas

Con la implementación completa, se inicia la fase de pruebas. Se llevan a cabo pruebas exhaustivas para verificar que el sistema cumpla con los requisitos definidos en la fase inicial. Esto incluye pruebas de unidad para cada componente individual, pruebas de integración para asegurar que los módulos funcionen juntos correctamente, pruebas de sistema para evaluar el sistema como un todo y pruebas de aceptación del usuario para garantizar que cumpla con las expectativas del cliente.

5.6.2.1.5 Despliegue

Una vez que el sistema ha pasado satisfactoriamente todas las pruebas, se procede al despliegue en el entorno de producción. Se realizan ajustes finales según sea necesario y se proporciona capacitación y soporte al usuario final. El sistema está listo para ser utilizado de manera regular por los usuarios finales.

VI. ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

6.1 DISEÑO METODOLÓGICO

El diseño metodológico del sistema Helpdesk para la División de Tecnología de Información de la Universidad Nacional de Ingeniería se basa en el modelo en cascada, una metodología de desarrollo de software que sigue un enfoque secuencial y estructurado. Este diseño garantiza una planificación detallada y una ejecución ordenada de todas las etapas del proyecto, desde la captura de requisitos hasta la implementación y el despliegue del sistema.

6.1.1 Adaptación al Modelo en Cascada

El modelo en cascada se ajusta a las necesidades del proyecto de Helpdesk al proporcionar una estructura clara y ordenada para el desarrollo del sistema. A continuación, se detallan las adaptaciones específicas de este modelo para el diseño del sistema Helpdesk:

6.1.1.1 Requisitos

En esta etapa inicial, se llevará a cabo un análisis exhaustivo de los requisitos del sistema Helpdesk, un proceso fundamental para el éxito del proyecto. Se establecerá una comunicación estrecha con los usuarios finales y las partes interesadas para comprender completamente sus necesidades y expectativas. Además de documentar todos los requisitos en un documento de especificación de requisitos, se llevará a cabo un proceso de validación y verificación para garantizar la precisión y la integridad de la información recopilada.

Para asegurar una comprensión completa de los requisitos, se utilizarán diversas técnicas de recolección de información, como entrevistas con los usuarios finales, encuestas, análisis de casos de uso y revisión de documentación existente. Además, se prestará especial atención a los requisitos no funcionales, como el rendimiento, la seguridad y la escalabilidad, que son críticos para el éxito a largo plazo del sistema.

Durante este proceso, se identificarán y documentarán los requisitos tanto del sistema como del usuario, incluidos los requisitos funcionales que describen las funciones específicas que debe realizar el sistema y los requisitos no funcionales que definen sus atributos de calidad. Se establecerán criterios de aceptación claros para cada requisito, lo que facilitará la posterior validación del sistema por parte de los usuarios finales.

Además, se realizará un análisis detallado de los casos de uso del sistema, identificando los diferentes actores y sus interacciones con el sistema. Esto permitirá una comprensión más profunda de los flujos de trabajo y los escenarios de uso del sistema, lo que a su vez facilitará el diseño y la implementación de una solución que satisfaga las necesidades reales de los usuarios finales.

6.1.1.2 Diseño

Con los requisitos claros y validados en la etapa de análisis, se procederá al diseño detallado del sistema Helpdesk, una fase crítica que sienta las bases para su implementación exitosa. Durante esta etapa, se realizará una planificación meticulosa para definir la arquitectura general del sistema, incluidos los componentes de hardware y software necesarios para su funcionamiento óptimo.

Se llevará a cabo un análisis exhaustivo de los requisitos funcionales y no funcionales para determinar cómo se traducirán en la estructura y los componentes del sistema. Se prestará especial atención a la escalabilidad y la flexibilidad del diseño, anticipando las necesidades futuras y garantizando que el sistema pueda adaptarse a medida que evolucionen los requisitos del negocio.

Se establecerán las interfaces de usuario de manera intuitiva y eficiente, teniendo en cuenta las mejores prácticas de diseño de experiencia de usuario (UX) para garantizar que la interacción con el sistema sea fluida y satisfactoria para los usuarios finales. Se realizarán prototipos y maquetas para validar el diseño con los usuarios y recopilar retroalimentación temprana que permita realizar ajustes antes de la implementación final.

Además, se elaborarán diagramas de flujo de trabajo detallados que representen los procesos clave del sistema, desde la gestión de tickets hasta la asignación de tareas y la generación de informes. Estos diagramas servirán como guía para el desarrollo de la lógica de negocio y la implementación de funcionalidades específicas del sistema.

6.1.1.3 Implementación

La fase de implementación del sistema Helpdesk es un paso crucial en el proceso de desarrollo, donde los diseños y especificaciones previamente elaborados se traducen en un producto funcional y operativo. Esta etapa requiere un enfoque cuidadoso y meticuloso para garantizar una transición suave y efectiva hacia el uso del sistema por parte de los usuarios finales.

Durante la implementación, los desarrolladores trabajarán en la codificación de los componentes del sistema de acuerdo con las especificaciones establecidas en la fase de diseño. Se seguirán las mejores prácticas de codificación y se fomentará el modularidad y la reutilización de código para facilitar el mantenimiento futuro y la escalabilidad del sistema.

Es fundamental realizar pruebas unitarias de forma continua durante el proceso de desarrollo para detectar y corregir errores de manera temprana. Las pruebas unitarias se centrarán en validar el comportamiento individual de cada componente del sistema, garantizando su funcionalidad y confiabilidad.

Además, se llevará a cabo una integración continua para combinar los cambios realizados por varios desarrolladores en un entorno compartido de forma regular. Se realizarán pruebas de integración para garantizar que los diferentes módulos del sistema funcionen correctamente juntos y mantengan la coherencia en la experiencia del usuario.

Durante la implementación, se establecerán entornos de desarrollo, pruebas y producción para gestionar el ciclo de vida del software de manera efectiva. Cada entorno estará configurado de manera adecuada para reflejar las necesidades específicas de cada etapa del desarrollo y garantizar la consistencia en el comportamiento del sistema.

Se utilizarán herramientas de gestión de versiones y control de cambios para rastrear y gestionar los cambios realizados en el código y la configuración del sistema. Esto permitirá mantener un registro claro de las modificaciones realizadas y facilitará la colaboración entre los miembros del equipo de desarrollo.

6.1.1.4 Pruebas y Validación

La fase de pruebas y validación es un componente crítico del proceso de desarrollo de software, donde se verifica y valida el funcionamiento del sistema en todas sus dimensiones. Esta etapa garantiza que el sistema cumpla con los requisitos establecidos y que esté libre de errores antes de su despliegue en el entorno de producción.

Durante la fase de pruebas, se llevarán a cabo las siguientes actividades:

6.1.1.4.1 Pruebas Unitarias

Se realizarán pruebas unitarias exhaustivas para cada componente individual del sistema. Estas pruebas se centran en validar el comportamiento funcional y técnico de cada unidad de código, identificando y corrigiendo errores a nivel de código.

6.1.1.4.2 Pruebas de Integración

Se realizarán pruebas de integración para verificar que los diferentes módulos del sistema funcionen correctamente juntos. Se evaluará la interoperabilidad y la comunicación entre los distintos componentes, asegurando que no haya conflictos ni errores en la integración.

Durante todas las etapas de las pruebas, se documentarán cuidadosamente los resultados y se seguirá un proceso de seguimiento y resolución de problemas para abordar cualquier defecto o anomalía identificada. Además, se proporcionará capacitación y soporte adicional a los usuarios finales para garantizar una transición suave y exitosa al nuevo sistema.

6.1.1.5 Despliegue:

La fase de despliegue del sistema Helpdesk marca el momento en que el sistema pasa de los entornos de desarrollo y pruebas al entorno de producción, listo para su uso regular por parte de la comunidad universitaria. Esta etapa requiere una planificación cuidadosa y una ejecución precisa para garantizar una transición sin problemas y una experiencia positiva para los usuarios finales.

Durante la fase de despliegue, se llevan a cabo las siguientes actividades:

6.1.1.5.1 Configuración del Entorno de Producción

Se establece el entorno de producción con la configuración adecuada para soportar el sistema Helpdesk en un entorno de uso real. Esto implica configurar servidores, bases de datos y otros componentes de infraestructura de acuerdo con las especificaciones y requisitos del sistema.

6.1.1.5.2 Implementación del Sistema

Se realiza la implementación final del sistema Helpdesk en el entorno de producción. Esto incluye la instalación de la última versión del software, la carga de datos iniciales si es necesario y la configuración de los permisos de acceso para los usuarios finales y administradores.

6.1.1.5.3 Ajustes Finales y Optimización

Se realizan ajustes finales según sea necesario para garantizar que el sistema funcione de manera óptima en el entorno de producción. Esto puede incluir la optimización de la configuración del servidor, la configuración de la seguridad y la realización de pruebas adicionales para verificar el funcionamiento correcto del sistema.

6.1.1.5.4 Capacitación

Se proporciona capacitación adicional a los usuarios finales para familiarizarlos con el sistema y sus funcionalidades. Se ofrece soporte continuo para abordar cualquier pregunta o problema que puedan surgir durante el proceso de adopción del sistema.

6.2 FASE DE REQUISITOS

La Fase de Requisitos establece las bases para el diseño del sistema Helpdesk, identificando y documentando las necesidades de los usuarios y partes interesadas.

6.2.1 Requerimientos Funcionales

Tabla 1

Requerimiento Funcional No.01

Identificador: FR-01		Nombre: Iniciar de Sesión	
Tipo: Necesario		¿Crítico? Si	
		Prioridad: Alta	
Entrada: - Nombre de usuario o correo electrónico. - Contraseña. - Opción de iniciar sesión con Microsoft.		Salida: - Mensaje de éxito y redirección al dashboard del usuario. - Mensaje de error en caso de datos incorrectos o problemas de acceso.	
Descripción: El sistema debe permitir al usuario iniciar sesión ya sea mediante un inicio de sesión local con su nombre de usuario/correo y contraseña, o mediante el uso de OAuth con su cuenta de Microsoft.			
Postcondición: - El usuario es redirigido al dashboard donde se muestra la lista de tickets asignados al agente o un mensaje indicando que no hay tickets disponibles. - El Cliente es redirigido al dashboard donde puede visualizar sus tickets y realizar nuevas solicitudes.			
Manejo de Situaciones Anormales: Si las credenciales son incorrectas en un inicio de sesión local, el sistema debe mostrar un mensaje de error indicando que el nombre de usuario/correo o la contraseña son incorrectos.			
Criterios de Aceptación: - El sistema permite a ambos tipos de usuarios (Agentes y Clientes) iniciar sesión correctamente mediante ambas opciones (local y OAuth con Microsoft). - Después del inicio de sesión exitoso, el sistema redirige al dashboard adecuado según el tipo de usuario. - El sistema maneja correctamente los errores de inicio de sesión, mostrando mensajes informativos en caso de problemas.			

Nota. La tabla describe el proceso de inicio de sesión, incluyendo opciones locales y mediante OAuth, manejo de errores, y redirección según el tipo de usuario.

Tabla 2

Requerimiento Funcional No.02

Identificador: FR-02		Nombre: Ver tickets Asignados al Agente	
Tipo: Necesario		¿Critico? Si	
		Prioridad: Alta	
Entrada: - Identificador del agente. - Lista de servicios designados al agente		Salida: - Lista de tickets asignados a los servicios designados del agente. - Detalles de cada ticket.	
Descripción: Precondición: - El agente debe haber iniciado sesión en el sistema. Descripción: - El sistema debe permitir a los agentes ver una lista de todos los tickets asignados a sus servicios designados. - Para cada ticket, el sistema debe mostrar los detalles del ticket, incluyendo el número de ticket, el asunto del ticket, la descripción del ticket, la prioridad del ticket, el estado del ticket, la fecha y hora de creación del ticket, la fecha y hora de la última actualización del ticket, y la información del usuario que creó el ticket. Postcondición: - Se muestra la lista de tickets asignados al agente o un mensaje que indica que no hay tickets disponibles.			
Manejo de Situaciones Anormales: Si el agente no tiene tickets asignados a sus servicios designados, el sistema debe mostrar un mensaje que indique que no hay tickets disponibles.			
Criterios de Aceptación: - El sistema muestra una lista precisa de tickets asignados a los servicios designados del agente. - Para cada ticket, el sistema muestra los detalles correctos del ticket. - El sistema maneja el caso en que no haya tickets disponibles para mostrar de manera adecuada.			

Nota. La tabla describe cómo el sistema permite a los agentes visualizar los tickets asignados a sus servicios, mostrando detalles completos de cada ticket o indicando la ausencia de tickets disponibles.

Tabla 3

Requerimiento Funcional No.03

Identificador: FR-03		Nombre: Tomar Ticket.	
Tipo: Necesario		¿Crítico? Si	
		Prioridad: Alta	
Entrada: - Identificador del agente - Identificador del ticket		Salida: Detalles del ticket actualizado (si se reclama correctamente)	
Descripción:			
Precondición: - El agente debe haber iniciado sesión en el sistema. - El ticket debe estar asignado a uno de los servicios designados del agente.			
Descripción: El sistema debe permitir a los agentes tomen los tickets, asumiendo la propiedad de estos. Al tomar un ticket, el agente o los agentes se convierten en los responsables de su resolución.			
Postcondición: - Si el ticket se toma correctamente, el sistema debe mostrar un mensaje de éxito y actualizar los detalles del ticket para reflejar que el agente o agentes son los responsables de su resolución. - Si el ticket no se puede reclamar, el sistema debe mostrar un mensaje de error correspondiente.			
Manejo de Situaciones Anormales: Si el agente intenta reclamar un ticket que no está asignado a ninguno de sus servicios designados, el sistema debe mostrar un mensaje de error.			
Criterios de Aceptación: - El sistema permite a los agentes reclamar tickets. - Al tomar un ticket, el agente o los agentes se convierten en el propietario de este. - El sistema muestra mensajes de éxito o error adecuados. - El sistema maneja correctamente las situaciones anormales.			

Nota.La tabla explica cómo los agentes pueden tomar tickets asignados a sus servicios, actualizar su estado y recibir mensajes claros en caso de éxito o error.

Tabla 4

Requerimiento Funcional No.04

Identificador: FR-04		Nombre: Resolver tickets y agregar comentarios al cierre.	
Tipo: Necesario		¿Crítico? Si	Prioridad: Alta
Entrada: - Identificador del ticket. - Resolución seleccionada (por ejemplo, resuelto, rechazado, etc.) - Comentario opcional del agente sobre la resolución.		Salida: - Mensaje de éxito o error. - Detalles del ticket actualizado (estado = cerrado).	
Descripción: Precondición: - El agente debe haber iniciado sesión en el sistema. - El agente debe tener permiso para resolver el ticket (por ejemplo, el ticket debe estar asignado a uno de sus servicios designados o el agente debe ser el propietario del ticket reclamado - opcional, si relacionado con FR-02). - El agente debe tener conocimiento del problema y la solución adecuada. Descripción: - El sistema debe permitir a los agentes resolver tickets y agregar comentarios opcionales al cerrarlos. Al resolver un ticket, el agente indica que se ha encontrado una solución y el problema del usuario se ha resuelto. Los comentarios pueden proporcionar detalles adicionales sobre la resolución o cualquier otra información relevante. Postcondición: - Si el ticket se resuelve correctamente, el sistema debe mostrar un mensaje de éxito y actualizar el estado del ticket a "cerrado". - El comentario del agente (si se proporciona) se debe adjuntar al historial del ticket.			
Manejo de Situaciones Anormales: Si el agente intenta resolver un ticket para el que no tiene permiso, el sistema debe mostrar un mensaje de error que indique que la acción está restringida.			
Criterios de Aceptación: - El sistema permite a los agentes resolver tickets y agregar comentarios opcionales al cierre. - Al resolver un ticket, el estado del ticket cambia a "cerrado". - Los comentarios del agente se adjuntan al historial del ticket. - El sistema maneja correctamente el caso en que el agente no tenga permiso para resolver el ticket.			

Nota. La tabla detalla el proceso para que los agentes resuelvan tickets, cambien su estado a "cerrado" y agreguen comentarios opcionales, asegurando permisos adecuados y manejo de errores.

Tabla 5

Requerimiento Funcional No.05

Identificador: FR-05		Nombre: Acceder a reportes sobre KPIs.	
Tipo: Deseable		¿Crítico? No	Prioridad: Media
Entrada: - Identificador del agente (opcional) - Departamento o área de interés (opcional) - Rango de fechas (opcional)		Salida: Reportes con visualizaciones e información detallada sobre KPIs relevantes	
Descripción: Precondición: - El agente debe haber iniciado sesión en el sistema. Descripción: - El sistema debe permitir a los agentes acceder a reportes que muestren información sobre los KPIs del HelpDesk. Estos reportes pueden proporcionar información valiosa para supervisar el rendimiento del equipo y tomar decisiones informadas para mejorar la eficiencia y la calidad del servicio. Postcondición: - El sistema muestra los reportes de KPI solicitados o un mensaje que indica que no hay datos disponibles para el criterio de búsqueda.			
Manejo de Situaciones Anormales: Si el agente introduce criterios de búsqueda no válidos, el sistema debe mostrar un mensaje de error que indique el problema.			
Criterios de Aceptación: - El sistema permite a los agentes acceder a reportes de KPI. - Los reportes muestran información relevante y actualizada sobre los KPIs del HelpDesk. - Los reportes utilizan visualizaciones claras y fáciles de entender. - Los agentes pueden filtrar o generar reportes personalizados según sus necesidades (opcional). - El sistema maneja correctamente los criterios de búsqueda no válidos.			

Nota. Esta tabla describe el acceso a reportes de KPIs con visualizaciones detalladas y filtros opcionales.

Tabla 6

Requerimiento Funcional No.06

Identificador: FR-06		Nombre: Crear tickets seleccionando un grupo y servicio.	
Tipo: Necesario		¿Crítico? Si	
		Prioridad: Alta	
Entrada: • Grupo (categoría) del ticket (por ejemplo, Sistemas Administrativos) • Servicio del ticket (por ejemplo, Finanzas) • Asunto del ticket (obligatorio)		Salida: • Mensaje de confirmación de creación del ticket • Identificador del ticket creado (opcional)	
Descripción: Precondición: • El usuario debe haber iniciado sesión en el sistema. Descripción: • El sistema debe permitir a los agentes crear nuevos tickets seleccionando un grupo (categoría) y un servicio específico. El grupo representa la categoría general del problema (por ejemplo, Sistemas Administrativos, Red, Software, etc.), y el servicio representa un área específica dentro de esa categoría (por ejemplo, Finanzas, Seguridad, Instalación, etc.). • Postcondición: • El sistema crea un nuevo ticket con la información proporcionada y muestra un mensaje de confirmación. • Opcionalmente, el sistema puede mostrar el identificador del ticket creado para futuras referencias.			
Manejo de Situaciones Anormales: • Si el agente intenta crear un ticket con información incompleta (por ejemplo, falta el asunto o la descripción), el sistema debe mostrar un mensaje de error y solicitar la información faltante.			
Criterios de Aceptación: • El sistema permite a los agentes crear tickets seleccionando un grupo (categoría) y servicio. • El sistema valida la información obligatoria del ticket (asunto y descripción). • El sistema crea un nuevo ticket con la información proporcionada. • El sistema muestra un mensaje de confirmación de la creación del ticket.			

Nota. Esta tabla describe la creación de tickets seleccionando un grupo, servicio y asunto, con confirmación de creación.

Tabla 7

Requerimiento Funcional No.07

Identificador: FR-07		Nombre: Administrar todos los tickets dentro de sus grupos asignados.	
Tipo: Necesario		¿Crítico? Si	
		Prioridad: Alta	
Entrada: Identificador del administrador.		Salida: - Lista de tickets asignados a los grupos del administrador - Detalles completos de cada ticket.	
Descripción: Precondición: - El administrador debe haber iniciado sesión en el sistema. - El administrador debe tener permisos para ver y gestionar tickets. Descripción: - El sistema debe permitir a los administradores gestionar todos los tickets asignados a sus grupos. Los grupos representan áreas de responsabilidad específicas dentro del sistema de soporte técnico (por ejemplo, Soporte Técnico, Seguridad de la Red, Gestión de Cuentas). Postcondición: - El sistema muestra una lista de tickets filtrada según los criterios del administrador (si se usan filtros) o todos los tickets asignados a sus grupos. - Al seleccionar un ticket, el sistema muestra los detalles completos (similar a FR-03) para permitir la revisión y la toma de decisiones informadas.			
Manejo de Situaciones Anormales: Si el administrador introduce criterios de búsqueda no válidos, el sistema debe mostrar un mensaje de error que indique el problema.			
Criterios de Aceptación: - El sistema permite a los administradores ver tickets asignados a sus grupos. - El sistema proporciona filtros de búsqueda para facilitar la localización de tickets específicos. - El sistema muestra detalles completos de cada ticket. - Los administradores pueden realizar acciones adicionales sobre los tickets según sus permisos (por ejemplo, asignar a otro agente, modificar prioridad, etc.). (Nota.Esta funcionalidad no se menciona en la entrada original, pero es común en la gestión de tickets)			

Nota.Esta tabla describe la gestión de tickets asignados a los grupos del administrador, con filtros y detalles completos de cada ticket.

Tabla 8

Requerimiento Funcional No.08

Identificador: FR-08		Nombre: Asignar tickets a agentes específicos.	
Tipo: Necesario		¿Critico? No	Prioridad: Media
Entrada: - Identificador del administrador - Identificador del ticket - Identificador del agente al que se asignará el ticket		Salida: - Mensaje de confirmación de la asignación - Detalles actualizados del ticket (reflejan el nuevo agente asignado)	
Descripción: Precondición: - El administrador debe haber iniciado sesión en el sistema. - El administrador debe tener permisos para gestionar tickets. - El agente al que se asigna el ticket debe estar habilitado para trabajar con el grupo asociado al ticket. Descripción: - El sistema debe permitir a los administradores asignar tickets a agentes específicos dentro de sus grupos asignados. Esta funcionalidad permite una distribución adecuada de la carga de trabajo y garantiza que los tickets sean manejados por agentes con las habilidades y conocimientos necesarios para resolverlos. Postcondición: - El sistema actualiza el ticket para reflejar el nuevo agente asignado. - Se muestra un mensaje de confirmación al administrador. (Opcional) El sistema puede enviar una notificación al agente asignado para informarle sobre el nuevo ticket.			
Manejo de Situaciones Anormales: Si el administrador intenta asignar un ticket a un agente no habilitado para trabajar con el grupo asociado, el sistema debe mostrar un mensaje de error.			
Criterios de Aceptación: - El sistema permite a los administradores asignar tickets a agentes específicos dentro de sus grupos. - El sistema valida si el agente tiene los permisos necesarios para trabajar con el grupo asociado al ticket. - El sistema actualiza el ticket para reflejar el nuevo agente asignado. - El sistema muestra un mensaje de confirmación de la asignación. (Opcional) El sistema envía una notificación al agente asignado para informarle sobre el nuevo ticket.			

Nota. Esta tabla describe la asignación de tickets a agentes específicos con confirmación y notificación.

Tabla 9

Requerimiento Funcional No.09

Identificador: FR-09		Nombre: Modificar el estado del ticket.	
Tipo: Necesario		¿Crítico? No	Prioridad: Alta
Entrada: - Identificador del ticket - Nuevo estado del ticket (por ejemplo, Abierto, Pendiente, Cerrado) (Opcional) Comentario sobre el cambio de estado		Salida: - Mensaje de confirmación de la actualización del estado - Detalles actualizados del ticket (reflejan el nuevo estado)	
Descripción: Precondición: - El agente o administrador debe haber iniciado sesión en el sistema. - El agente o administrador debe tener permisos para modificar tickets. Descripción: - El sistema debe permitir a los agentes y administradores modificar el estado de los tickets. El cambio de estado permite reflejar el progreso del ticket y su ciclo de vida dentro del sistema de soporte. Postcondición: - El sistema actualiza el estado del ticket en la base de datos. - Se muestra un mensaje de confirmación al agente o administrador. (Opcional) El sistema puede registrar un comentario con la razón del cambio de estado. (Opcional) El sistema puede enviar una notificación al usuario o a otros agentes relevantes sobre el cambio de estado.			
Manejo de Situaciones Anormales: - Si el agente o administrador intenta modificar el estado a uno no válido, el sistema debe mostrar un mensaje de error. - Si el cambio de estado implica restricciones adicionales (por ejemplo, cerrar un ticket requiere una resolución final), el sistema debe mostrar las restricciones y solicitar la información necesaria al agente o administrador.			
Criterios de Aceptación: - El sistema permite a los agentes y administradores modificar el estado de los tickets. - El sistema valida si el nuevo estado es válido. - El sistema actualiza el estado del ticket en la base de datos. - El sistema muestra un mensaje de confirmación de la actualización. (Opcional) El sistema registra un comentario con la razón del cambio de estado. (Opcional) El sistema envía una notificación al usuario o a otros agentes relevantes sobre el cambio de estado.			

Nota. Esta tabla describe la modificación del estado de un ticket.

Tabla 10

Requerimiento Funcional No.10

Identificador: FR-10		Nombre: Acceder a informes sobre el rendimiento de los agentes dentro de su grupo.	
Tipo: Necesario		¿Crítico? No	
Prioridad: Media			
Entrada: - Identificador del agente (opcional, para ver el rendimiento de un único agente) - Identificador del grupo (opcional, para ver el rendimiento del grupo) - Filtros de tiempo (por ejemplo, periodo del informe)		Salida: - Informe detallado sobre el rendimiento del agente o grupo - Visualizaciones de métricas clave.	
Descripción: Precondición: - El agente o administrador debe haber iniciado sesión en el sistema. - El agente o administrador debe tener permisos para acceder a informes de rendimiento. - Los filtros de tiempo seleccionados deben ser válidos. Descripción: - El sistema debe permitir a los agentes y administradores con los permisos adecuados acceder a informes sobre el rendimiento de los agentes dentro de su grupo. Esta funcionalidad proporciona información valiosa para supervisar la actividad, identificar áreas de mejora y reconocer el desempeño positivo. Postcondición: - El sistema genera un informe detallado que muestra métricas clave de rendimiento para el agente o grupo seleccionado dentro del periodo de tiempo especificado. - El sistema puede mostrar visualizaciones de las métricas clave para facilitar el análisis y la comprensión de los datos.			
Manejo de Situaciones Anormales: - Si el agente o administrador introduce filtros de tiempo no válidos, el sistema debe mostrar un mensaje de error. - Si el agente o administrador no tiene permisos para acceder a informes de rendimiento de un grupo específico, el sistema debe mostrar un mensaje de acceso denegado.			
Criterios de Aceptación: - El sistema permite a los agentes y administradores autorizados acceder a informes sobre el rendimiento de los agentes dentro de su grupo. - El sistema valida los filtros de tiempo introducidos por el usuario. - El sistema genera un informe detallado con métricas clave de rendimiento para el agente. - El informe incluye visualizaciones de las métricas clave para facilitar el análisis. - El sistema restringe el acceso a los informes según los permisos de usuario.			

Nota. Esta tabla describe el acceso a informes de rendimiento con métricas y gráficos.

6.2.2 Requerimientos No funcionales

- **Rendimiento:**

- El sistema debe ser capaz de manejar un alto volumen de tickets y usuarios sin comprometer la velocidad de respuesta.
- El sistema debe ser escalable para adaptarse a un crecimiento futuro en la cantidad de usuarios y datos.

- **Seguridad:**

- El sistema debe proteger la información confidencial de los usuarios.
- El sistema debe implementar medidas de seguridad para prevenir accesos no autorizados, e intrusiones.

- **Usabilidad:**

- La interfaz del sistema debe ser intuitiva y fácil de usar para agentes, administradores y usuarios finales.
- El sistema debe proporcionar herramientas de ayuda y documentación para facilitar su aprendizaje y uso

6.3 FASE DE DISEÑO

La fase de diseño convierte los requisitos en un plan técnico detallado que guiará la construcción del sistema Helpdesk. En esta etapa se define la arquitectura del software, el diseño de la base de datos, las interfaces de usuario asegurando que el sistema cumpla con los objetivos funcionales y no funcionales.

6.3.1 Arquitectura del Sistema - Clean Architecture

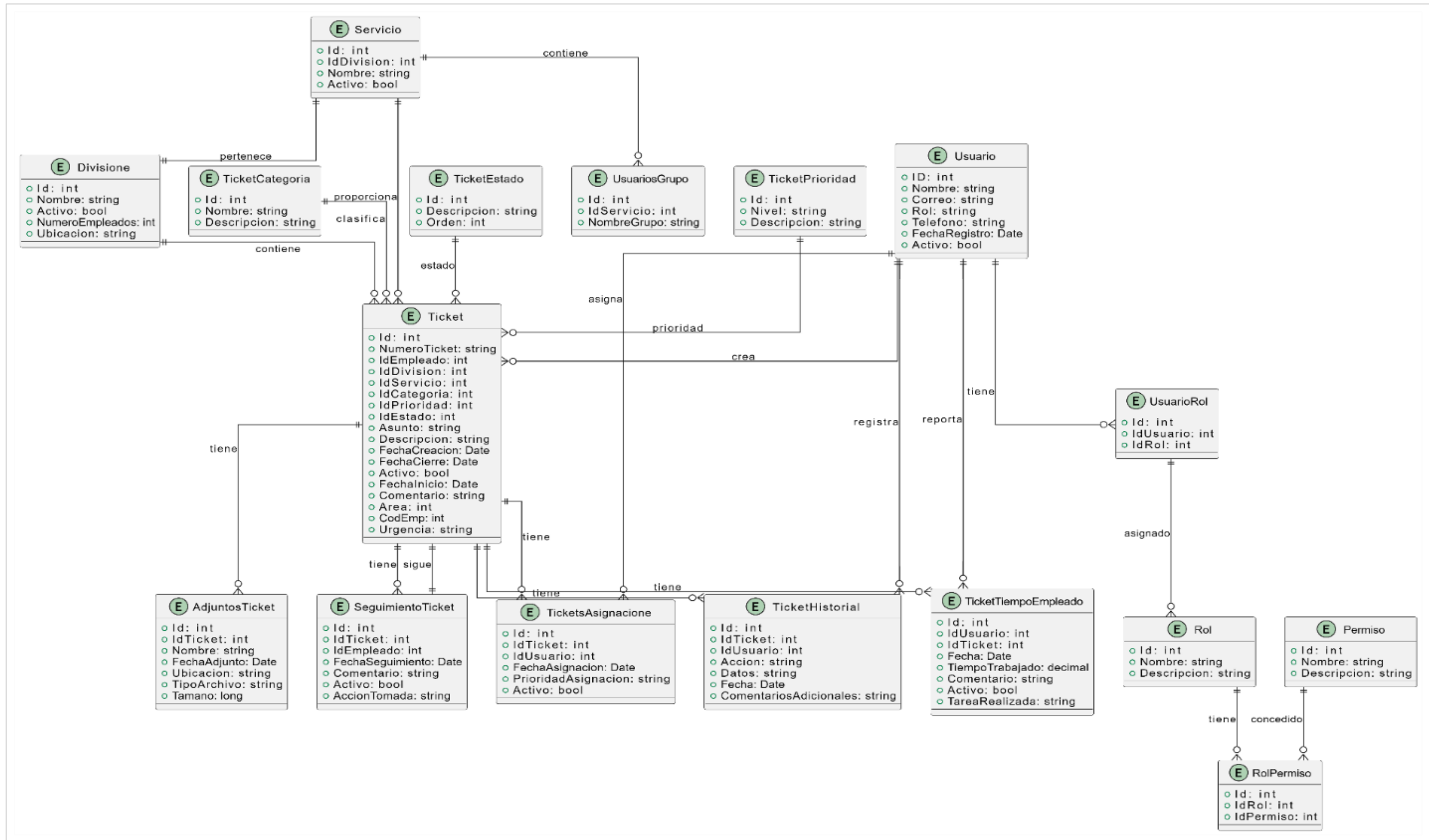
Clean Architecture sigue un enfoque en capas que separa las preocupaciones, lo que facilita la escalabilidad y el mantenimiento. La descripción de tu arquitectura debería incluir las siguientes capas:

6.3.1.1 Capa de Dominio (Domain)

La capa de dominio es el núcleo del sistema, donde se definen las entidades y las reglas de negocio que rigen su comportamiento. Aquí se concentran las clases principales que representan los conceptos clave del sistema Helpdesk.

- **Entidades:** Se incluyen clases como Ticket, Usuario, Departamento, que encapsulan tanto los atributos (por ejemplo, ID, Título, Descripción, Estado, Fecha Creación para Ticket) como los métodos que implementan la lógica de negocio relacionada.
- **Servicios de dominio:** Esta capa también puede contener servicios que implementan lógica compleja que involucra múltiples entidades, como la asignación de tickets a usuarios o la validación de estados.
- **Interfaces:** Se definen interfaces para los repositorios y otros servicios, garantizando que las implementaciones concretas en capas inferiores no afecten la lógica de negocio.

Figura 1: Diagrama de Clases



Nota. Esta figura ilustra el diagrama de clases del sistema Helpdesk, detallando las relaciones entre entidades como usuarios, roles, permisos, tickets, asignaciones y seguimiento de estos.

6.3.1.1.1 Reglas de Negocio del Sistema Helpdesk

1. Gestión de Usuarios y Roles

- a. **Registro de Usuario:** Los usuarios son registrados en el sistema, con un rol que define sus permisos y funcionalidades dentro de la plataforma.
- b. **Asignación de Roles:** Cada usuario puede tener uno o más roles, determinados a través de la entidad UsuarioRol, lo que define sus capacidades en el sistema.
- c. **Gestión de Permisos:** Los roles pueden tener uno o más permisos asociados, gestionados a través de RolPermiso, lo que determina las acciones que un usuario puede realizar.

2. Creación y Gestión de Tickets

- a. **Creación de Tickets:** Los usuarios pueden crear tickets con detalles como Asunto, Descripción, Categoría, Prioridad y Estado.
- b. **Estado del Ticket:** Cada ticket tiene un estado que puede cambiar a lo largo de su ciclo de vida (por ejemplo, "Abierto", "En Progreso", "Cerrado"), gestionado a través de TicketEstado.
- c. **Clasificación de Tickets:** Los tickets se clasifican en categorías mediante TicketCategoría, facilitando su gestión.

3. Historial de Tickets

- a. **Registro de Historial:** Cada acción sobre un ticket se registra en TicketHistorial, permitiendo un seguimiento detallado del ciclo de vida del ticket.
- b. **Tiempo Reportado por el Empleado:** Los usuarios registran el tiempo trabajado en cada ticket, almacenado en TicketTiempoEmpleado, facilitando la evaluación del esfuerzo invertido.

4. Asignaciones de Tickets

- a. **Asignación a Usuarios:** Los tickets se pueden asignar a diferentes usuarios mediante TicketsAsignacione, distribuyendo las tareas y seguimiento de responsabilidades.

- b. **Priorización de Asignaciones:** Se especifica una prioridad al asignar un ticket, influyendo en el orden de atención.

5. Gestión de Divisiones y Servicios

- a. **Clasificación por División:** Cada ticket se asocia a una división específica, como Soporte Técnico, Servidores, o Sistemas. Esto permite gestionar solicitudes según el área organizativa correspondiente.
- b. **Servicios Asociados:** Los tickets están vinculados a un servicio específico ofrecido por la división, lo que permite una atención más especializada y eficiente.

6. Validaciones de Datos

- a. **Validaciones de Entrada:** Se implementan validaciones para asegurar que todos los campos obligatorios sean completados correctamente al crear o modificar tickets.
- b. **Control de Estado Activo:** Se manejan correctamente los estados Activo de usuarios, tickets y otras entidades, asegurando que solo se procesen elementos relevantes.

6.3.1.2 Capa de Aplicación (Application)

La Capa de Aplicación actúa como un intermediario entre la Capa de Dominio y las demás capas del sistema. Su responsabilidad principal es coordinar las acciones que los usuarios pueden realizar y gestionar la lógica de negocio necesaria para cumplir con esos requerimientos.

Responsabilidades de la Capa de Aplicación

- **Casos de Uso:** Se detallan los principales casos de uso, como Crear Ticket, Asignar Ticket y Cerrar Ticket, que describen cómo los usuarios interactúan con el sistema. Cada caso de uso se define con sus respectivos actores, acciones y resultados esperados.

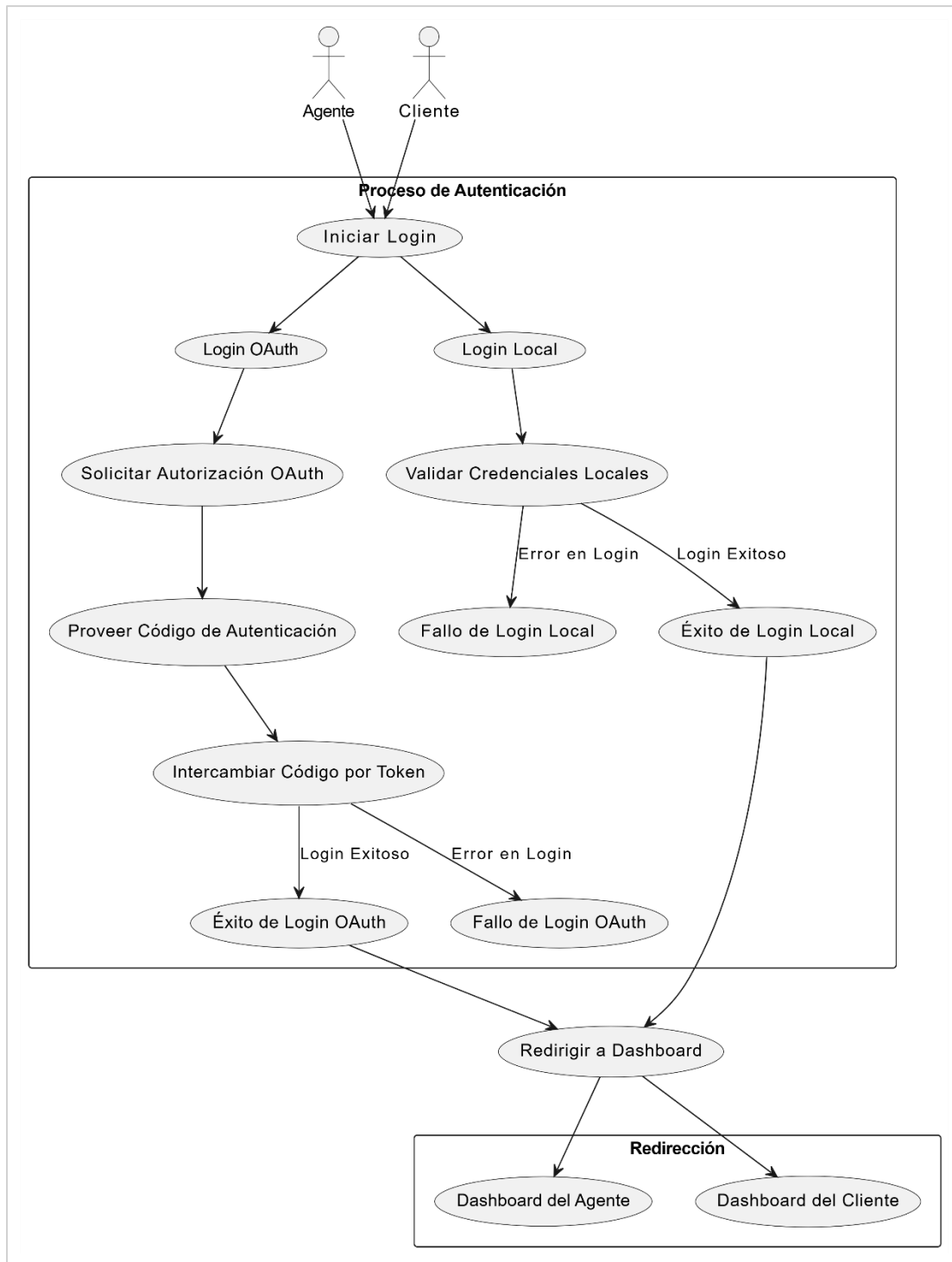
- **Servicios de Aplicación:** Esta capa implementa la lógica que orquesta los casos de uso, invocando a las entidades del dominio y a los repositorios necesarios para completar las acciones solicitadas. Los servicios de aplicación se encargan de coordinar las diferentes operaciones y garantizar que se cumplan las reglas de negocio definidas.
- **Interacción con Repositorios:** La Capa de Aplicación interactúa con las interfaces de repositorios para acceder a los datos necesarios sin conocer su implementación. Esto permite que la lógica de negocio esté separada del acceso a datos, facilitando el mantenimiento y las pruebas.

Diagramas de Casos de Uso

Para ilustrar cómo los usuarios interactúan con el sistema, se incluirán diagramas de caso de uso. Estos diagramas visualizan las interacciones entre los actores y el sistema, proporcionando una representación clara de las funcionalidades disponibles y los flujos de trabajo.

Figura 2

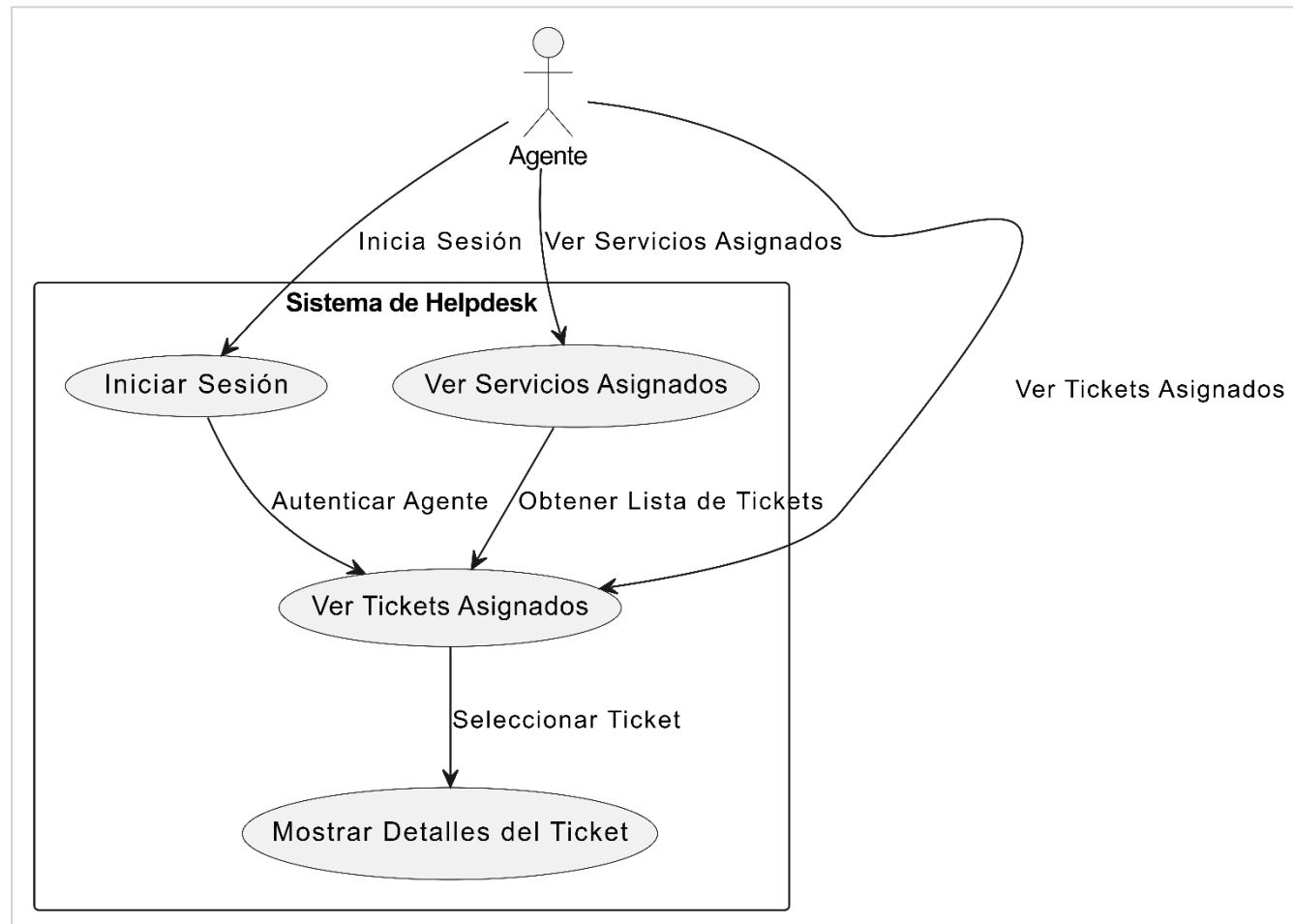
Diagrama de Caso de Uso No.1



Nota. La figura ilustra el caso de uso correspondiente a la autenticación de usuario.

Figura 3

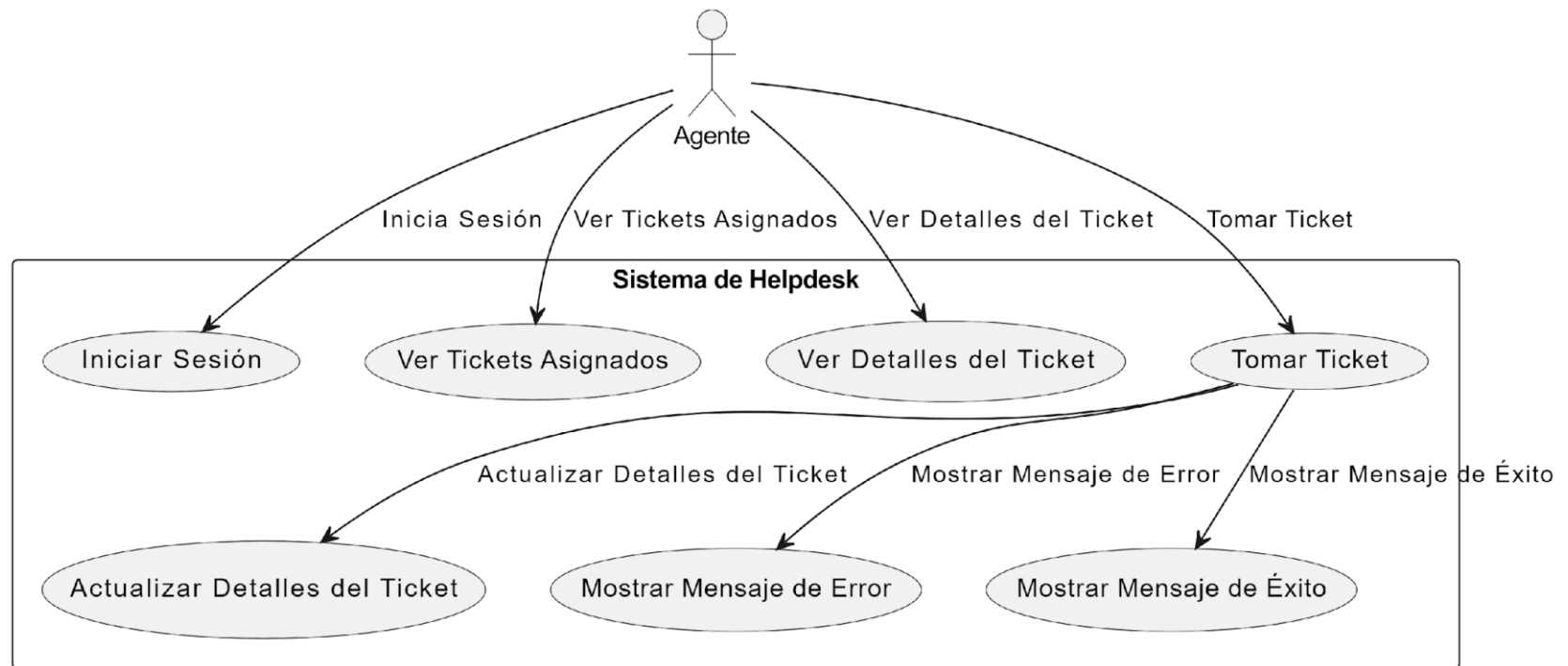
Diagrama de Caso de Uso No.2



Nota. La figura ilustra el caso de uso que corresponde a ver los tickets asignados.

Figura 4

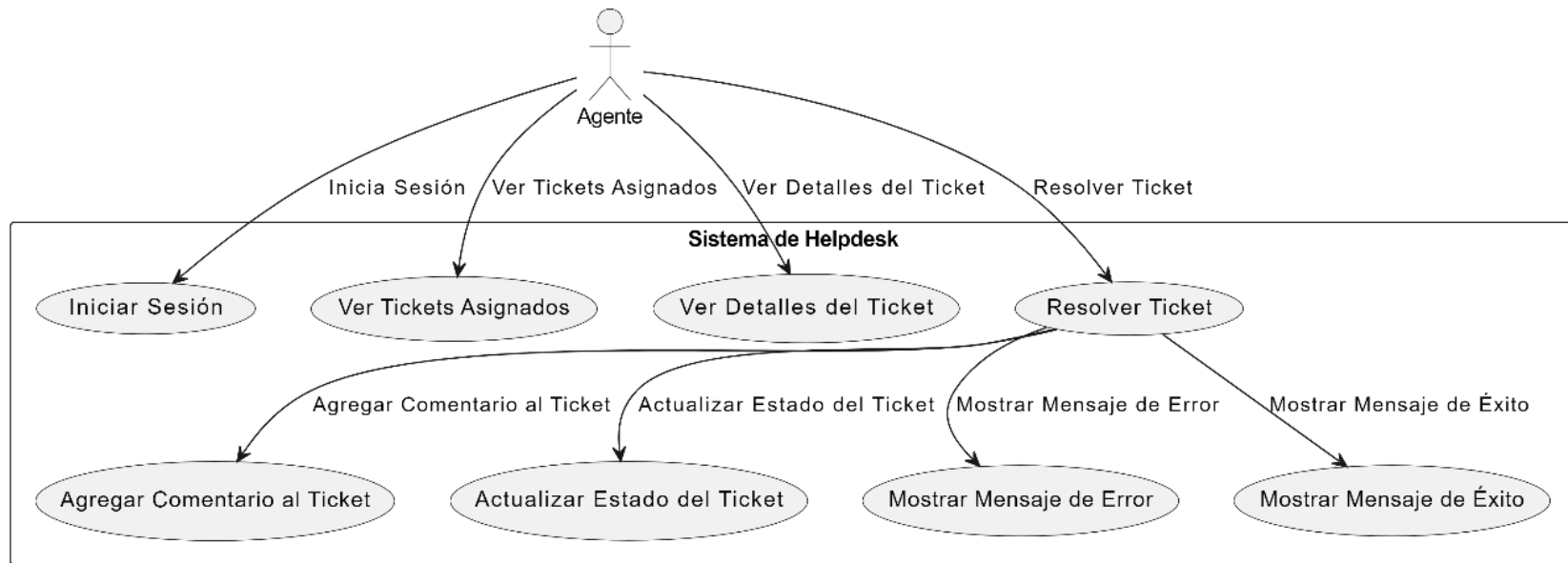
Diagrama de Caso de Uso No.3



Nota. La figura ilustra el caso de uso que corresponde a tomar tickets.

Figura 5

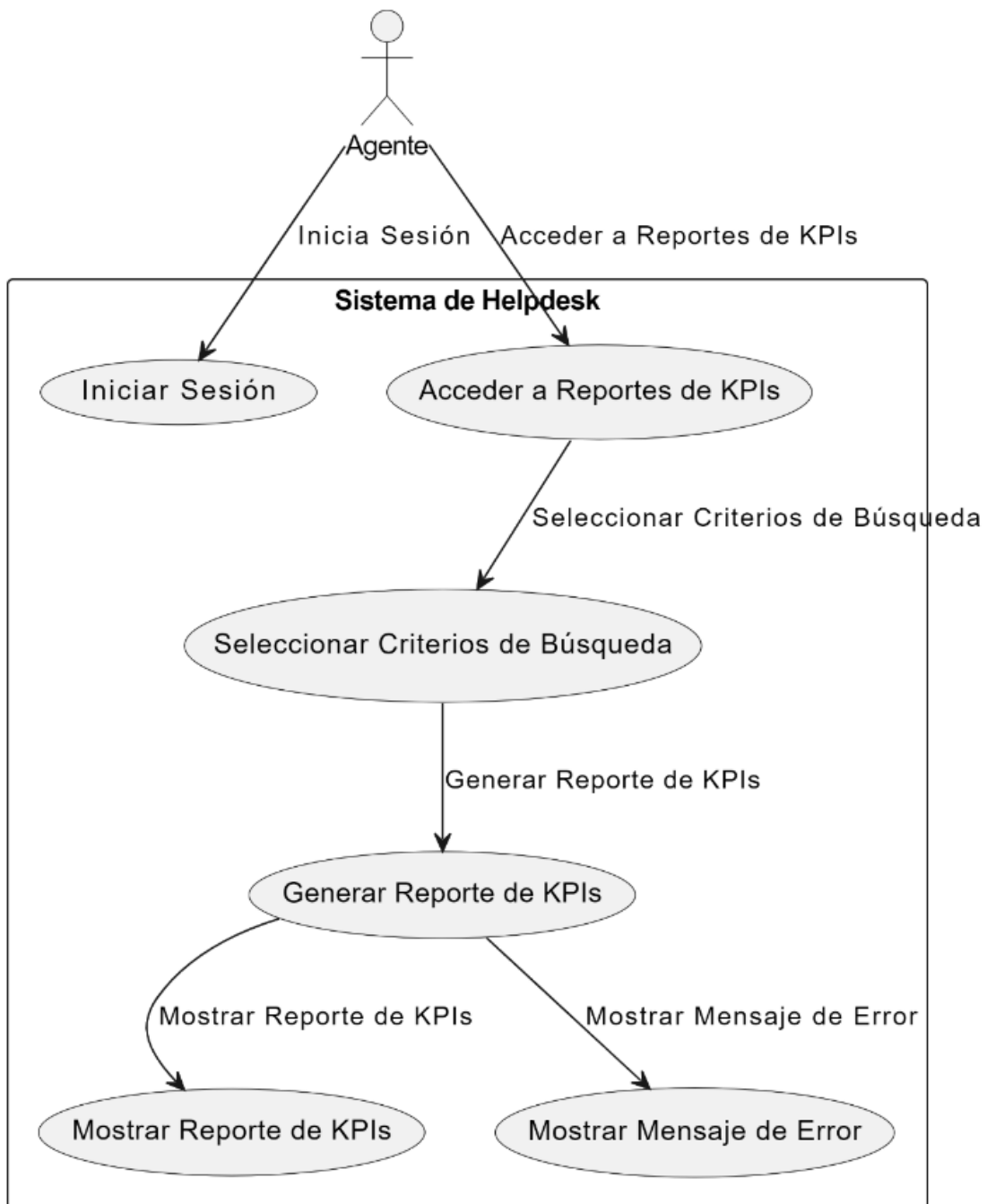
Diagrama de Caso de Uso No.4



Nota. La figura ilustra el caso de uso que corresponde a resolver tickets.

Figura 6

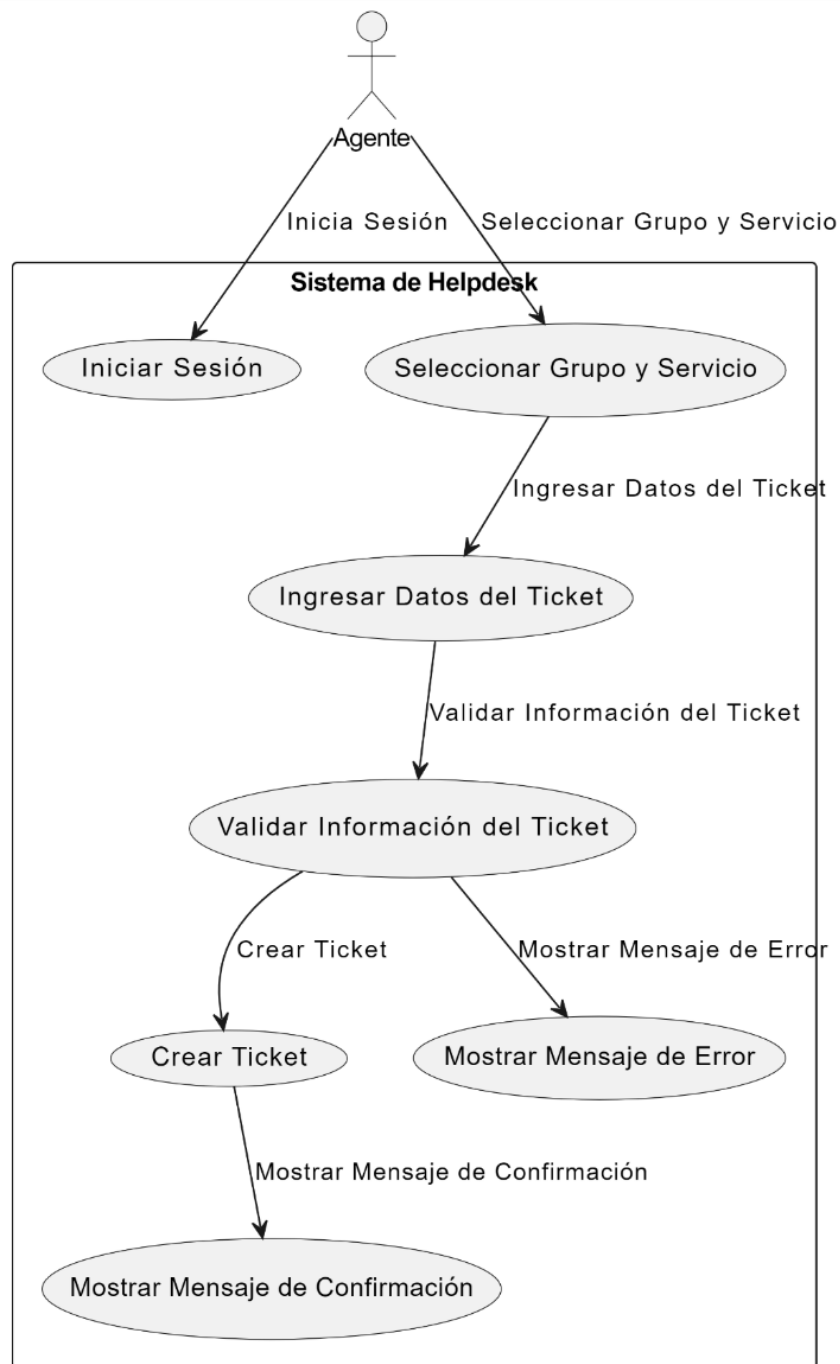
Diagrama de Caso de Uso No.5



Nota. La figura ilustra el caso de uso que corresponde a generar reportes.

Figura 7

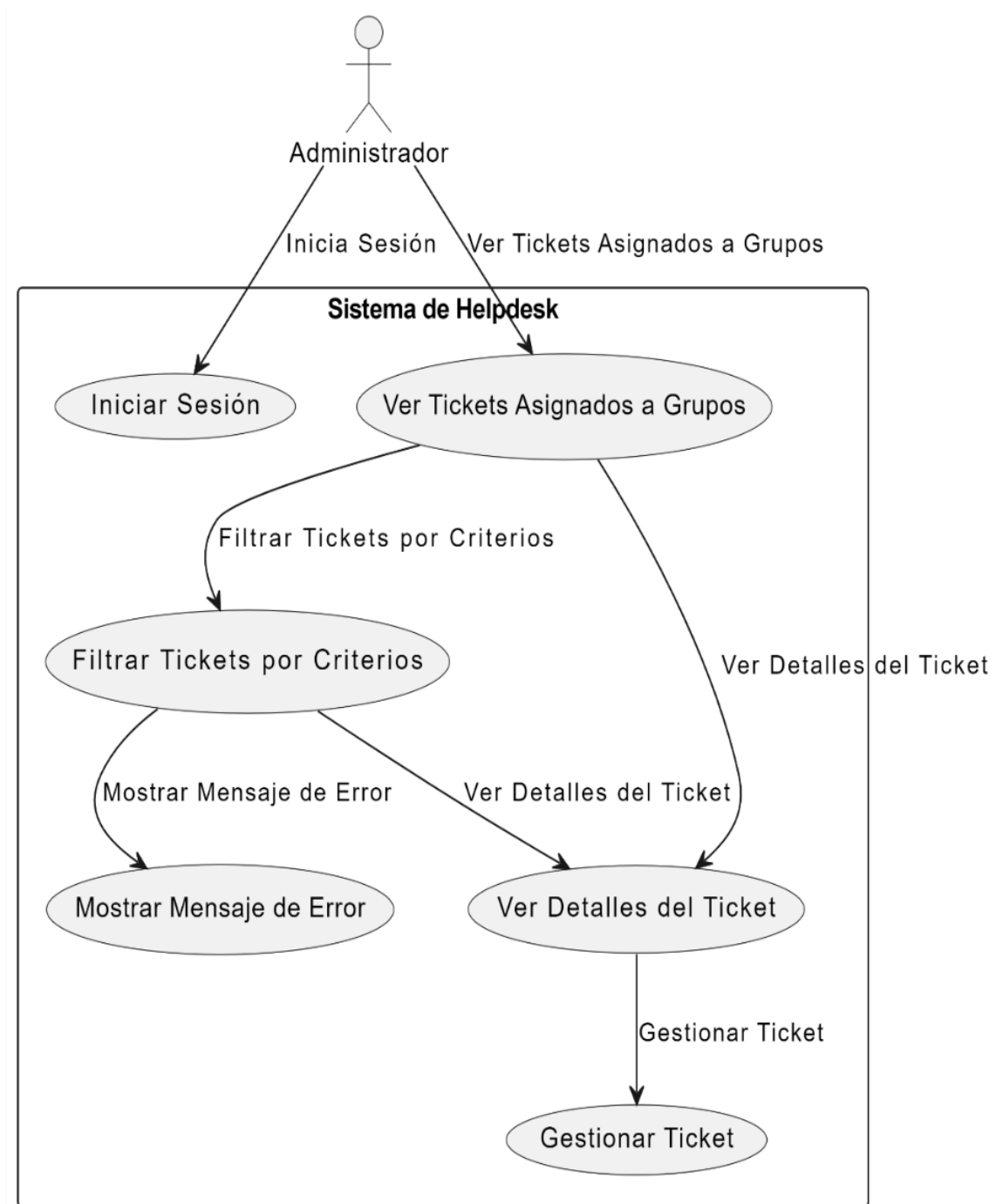
Diagrama de Caso de Uso No.6



Nota. La figura ilustra el caso de uso que corresponde creación de tickets.

Figura 8

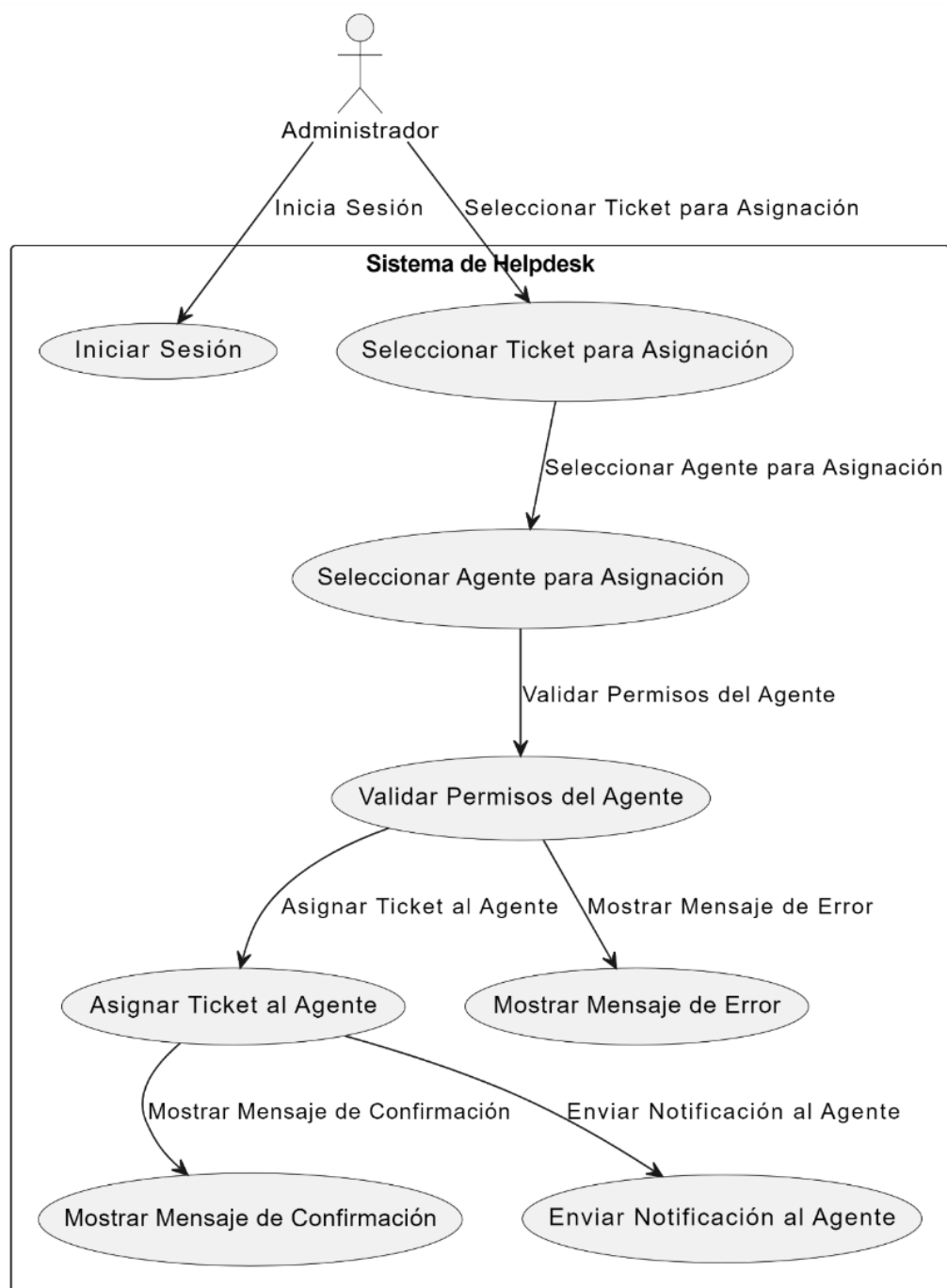
Diagrama de Caso de Uso No.7



Nota. La figura ilustra el caso de uso que corresponde a filtrar por criterios.

Figura 9

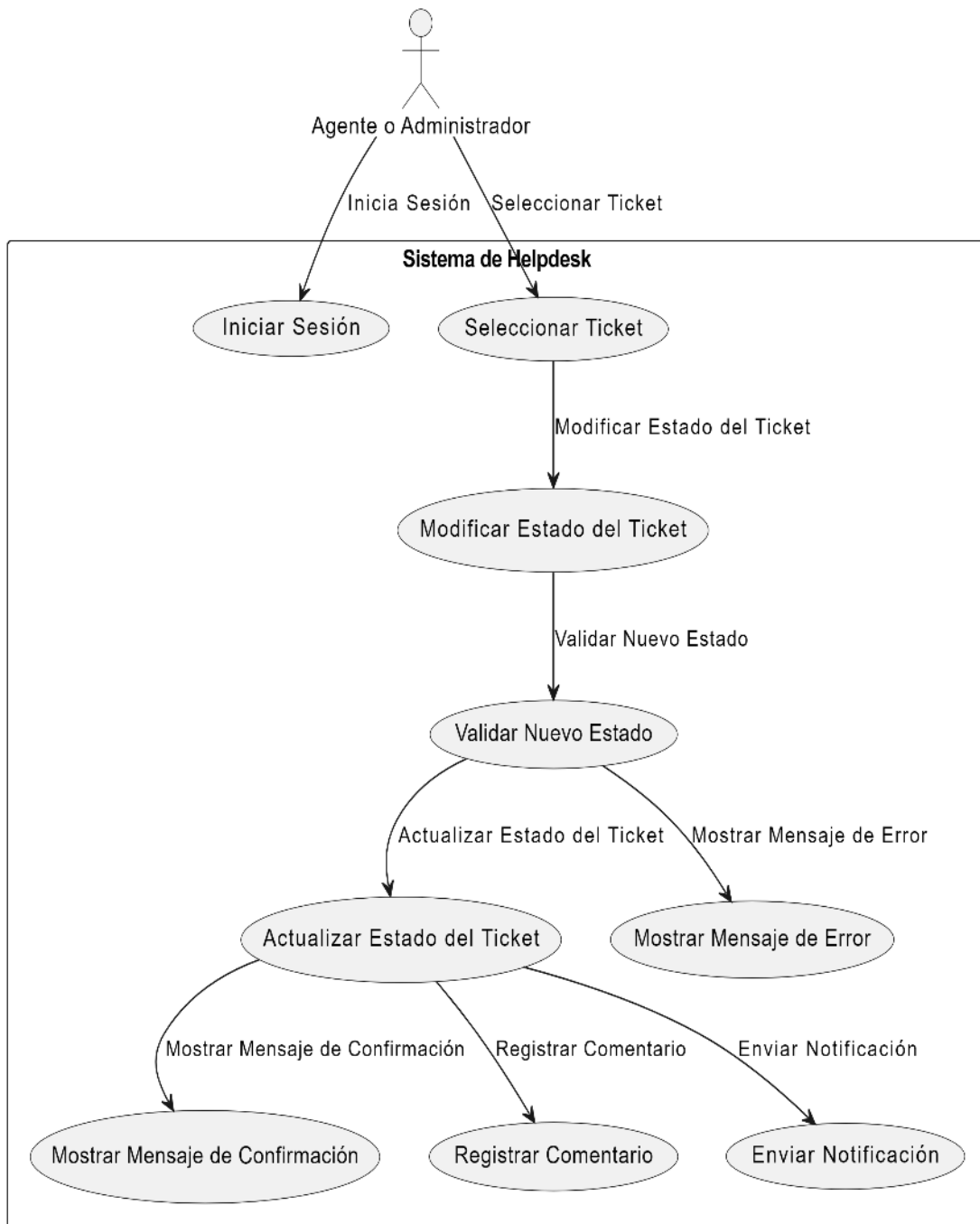
Diagrama de Caso de Uso No.8



Nota. La figura ilustra el caso de uso que corresponde a asignar un agente.

Figura 10

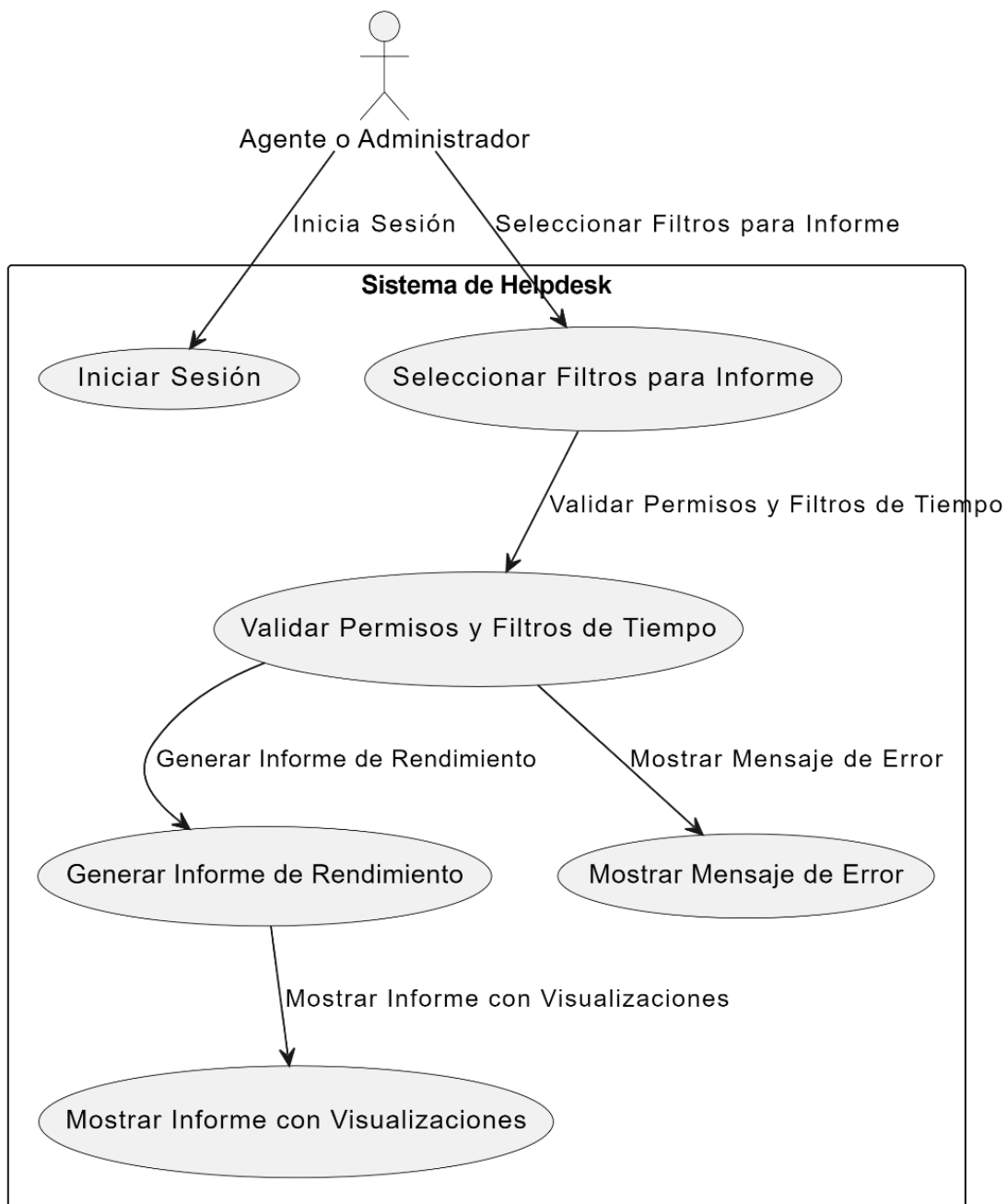
Diagrama de Caso de Uso No. 9



Nota. La figura ilustra al caso de uso que corresponde a modificar el estado de un ticket.

Figura 11

Diagrama de Caso de Uso No.10



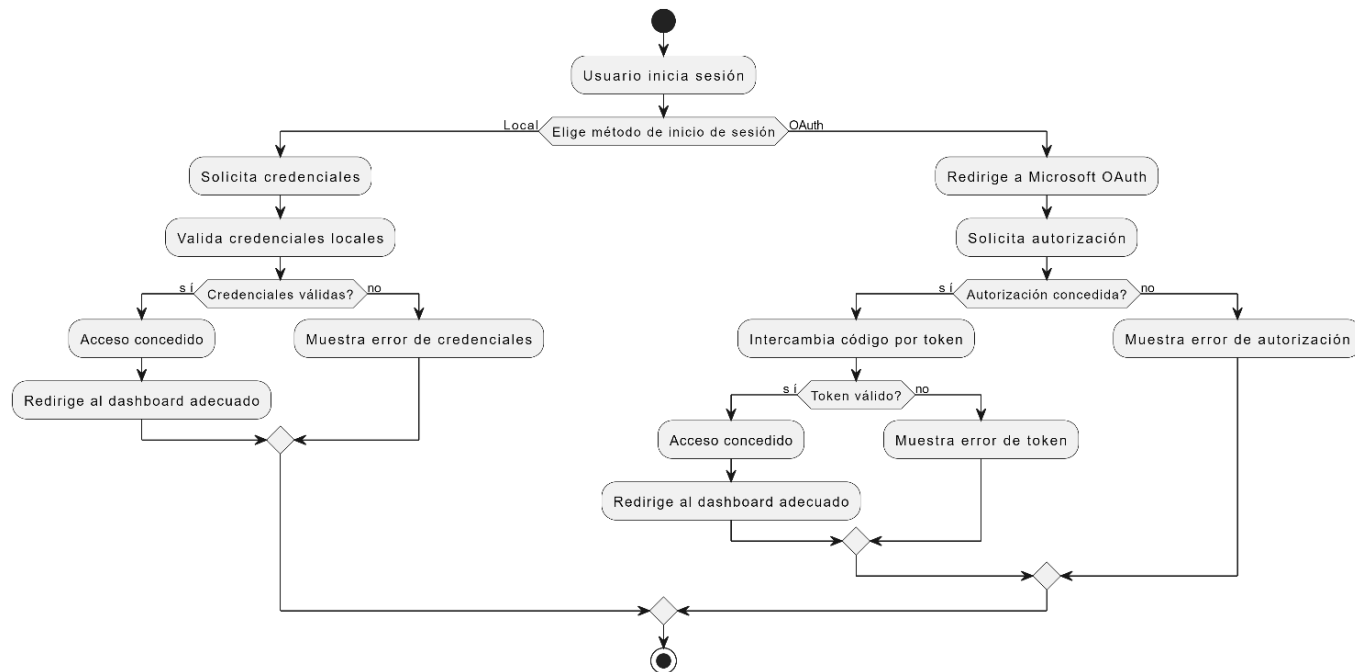
Nota. La figura ilustra el caso de uso que corresponde a seleccionar los criterios de reportes.

Diagramas de Actividades

Los diagramas de actividades detallan los flujos de trabajo dentro del sistema, mostrando visualmente las secuencias de acciones y decisiones de los usuarios y el sistema.

Figura 12

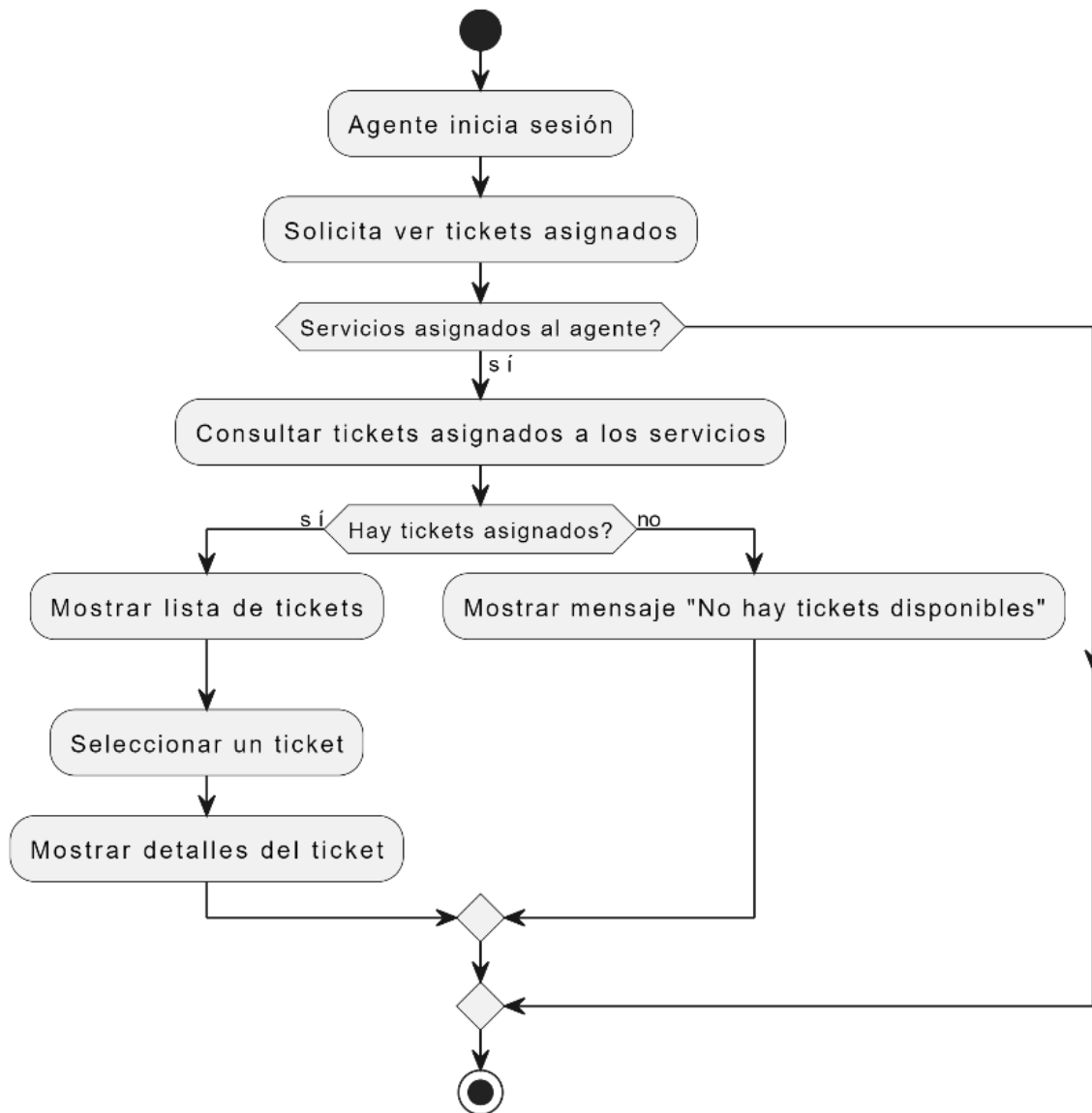
Diagrama de Actividades No.1



Nota. La figura muestra el diagrama de actividades correspondiente a autenticación y autorización.

Figura 13

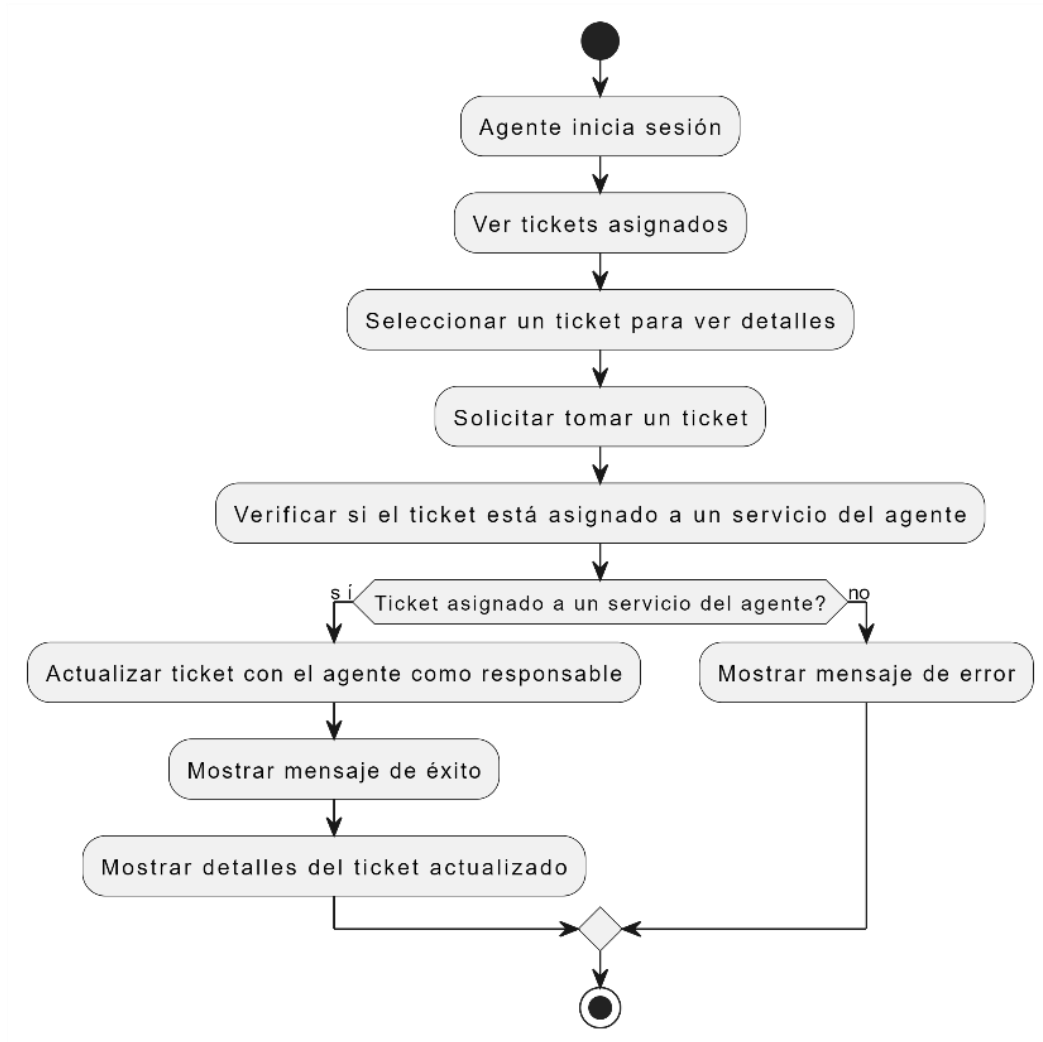
Diagrama de Actividades No.2



Nota. La figura ilustra el diagrama de actividades correspondiente a ver tickets asignados.

Figura 14

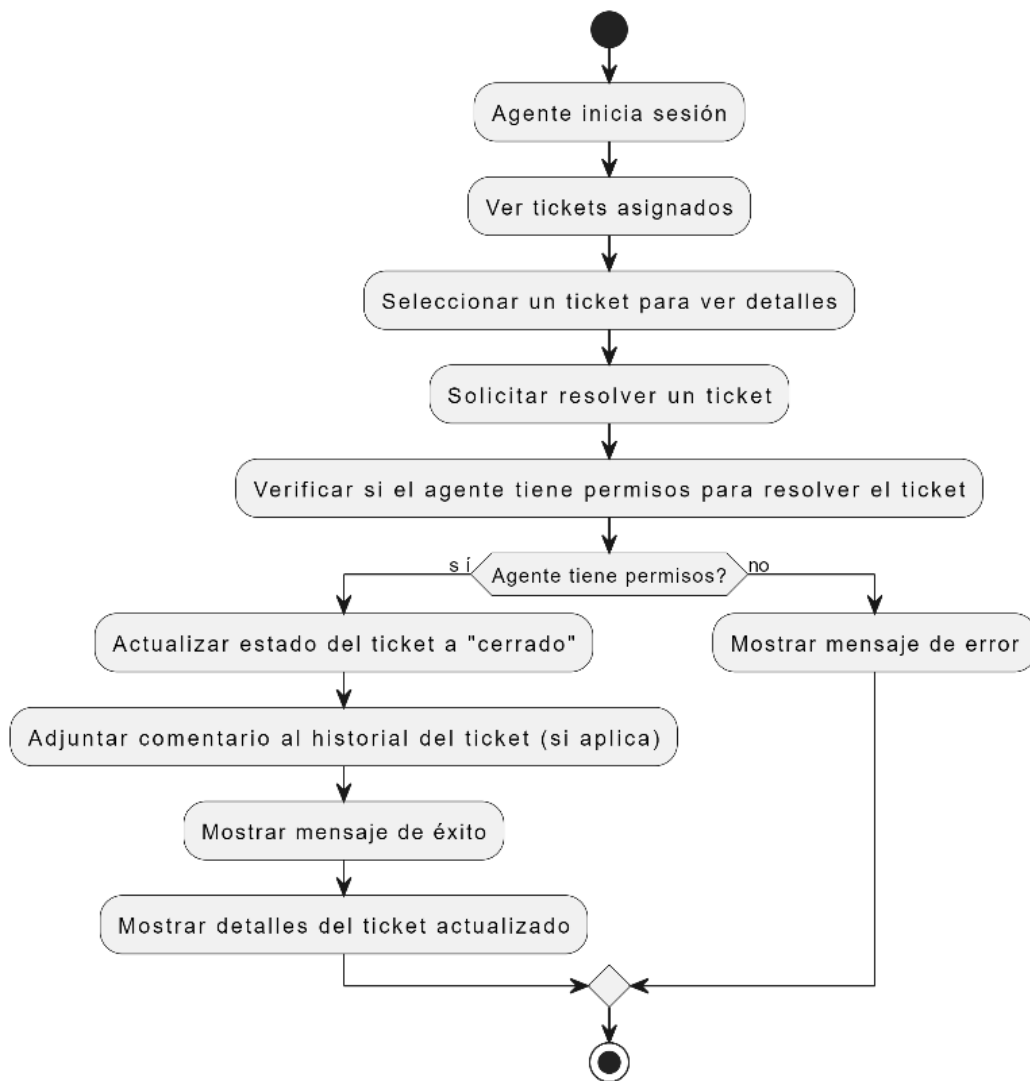
Diagrama de Actividades No.3



Nota. La figura muestra el diagrama de actividades correspondiente a tomar un ticket.

Figura 15

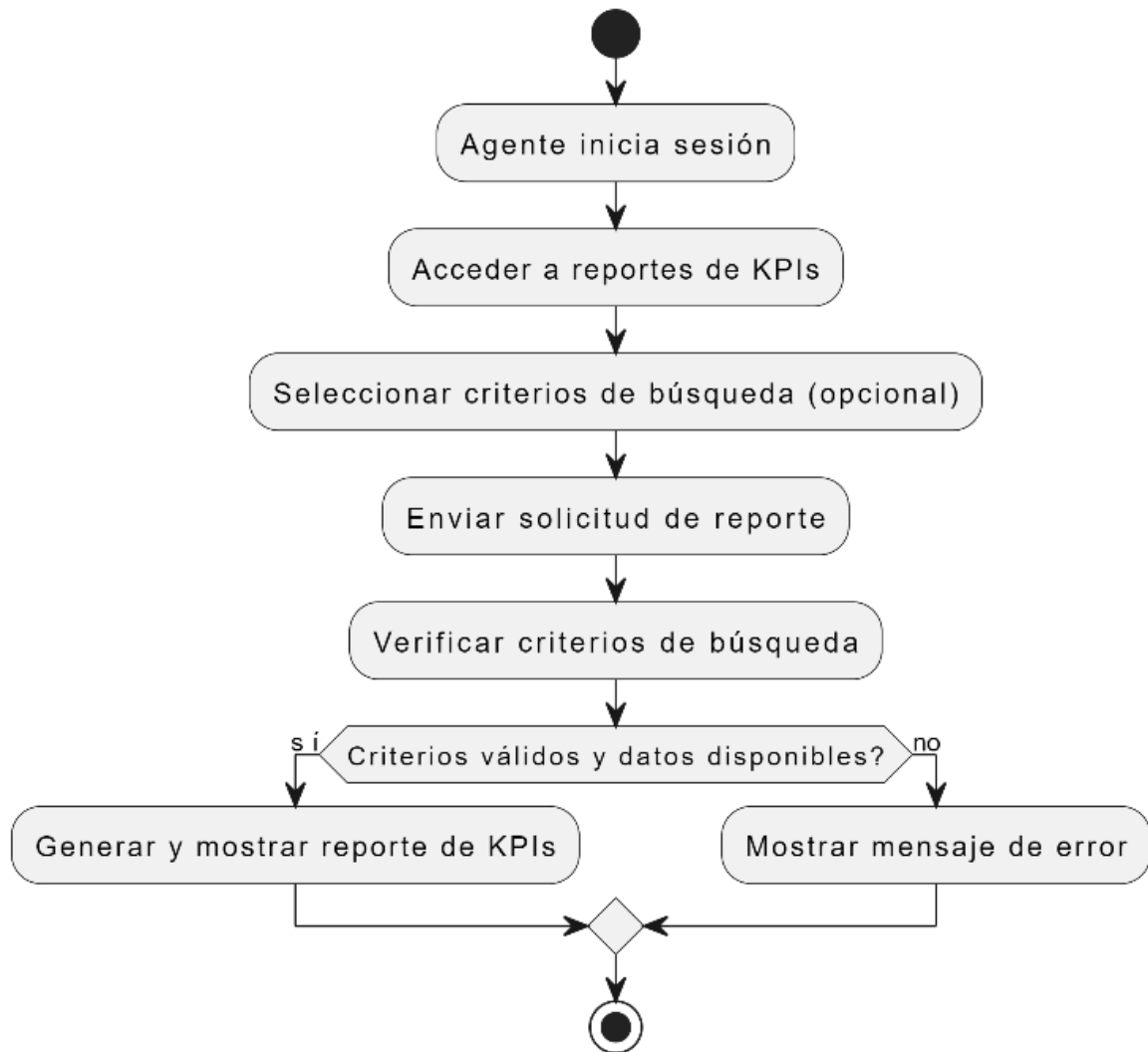
Diagrama de Actividades No.4



Nota. La figura ilustra el diagrama de actividades correspondiente a resolución de tickets.

Figura 16

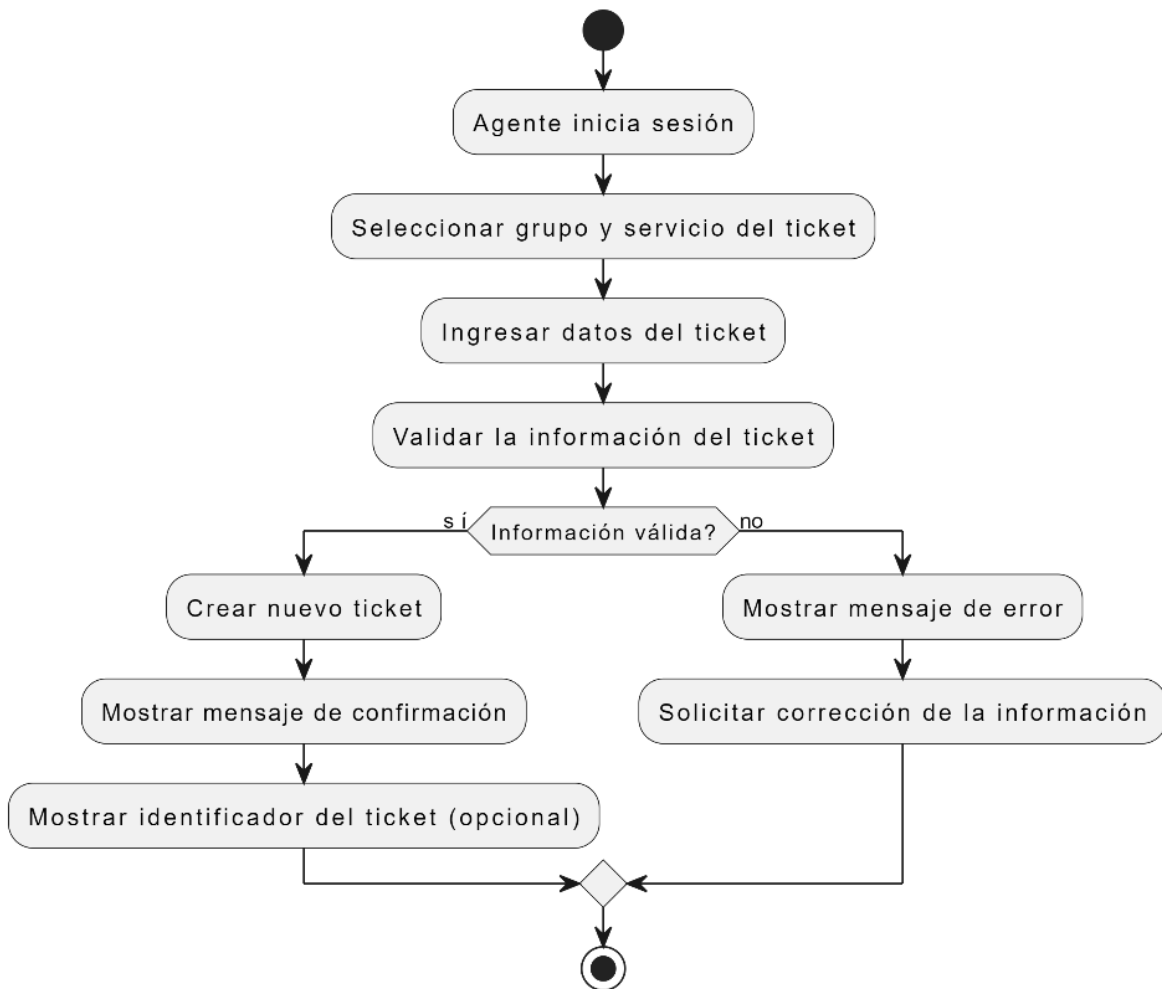
Diagrama de Actividades No.5



Nota. La figura muestra el diagrama de actividades correspondiente

Figura 17

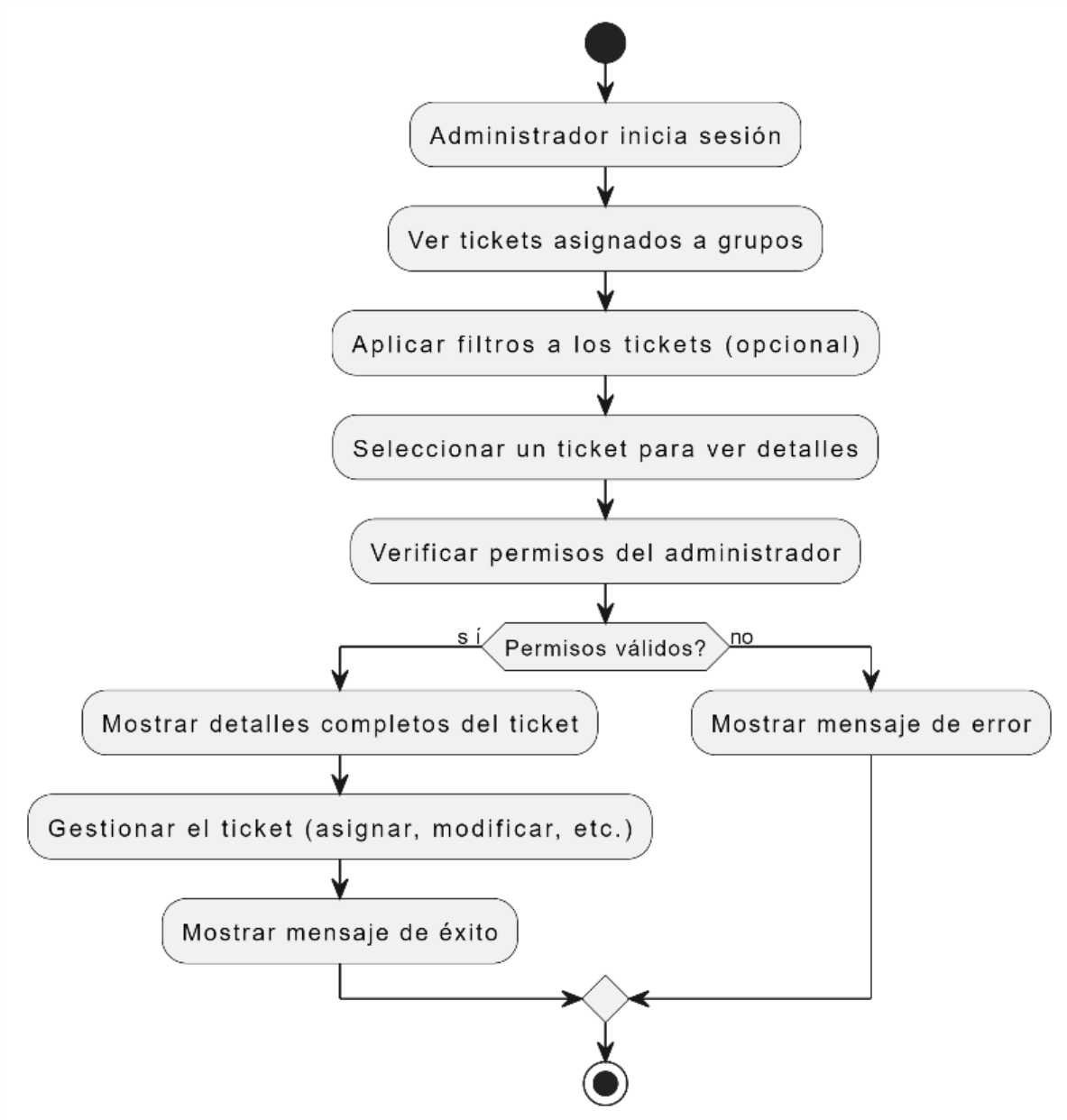
Diagrama de Actividades No.5



Nota. La figura ilustra el diagrama de actividades correspondiente a ingresar ticket.

Figura 18

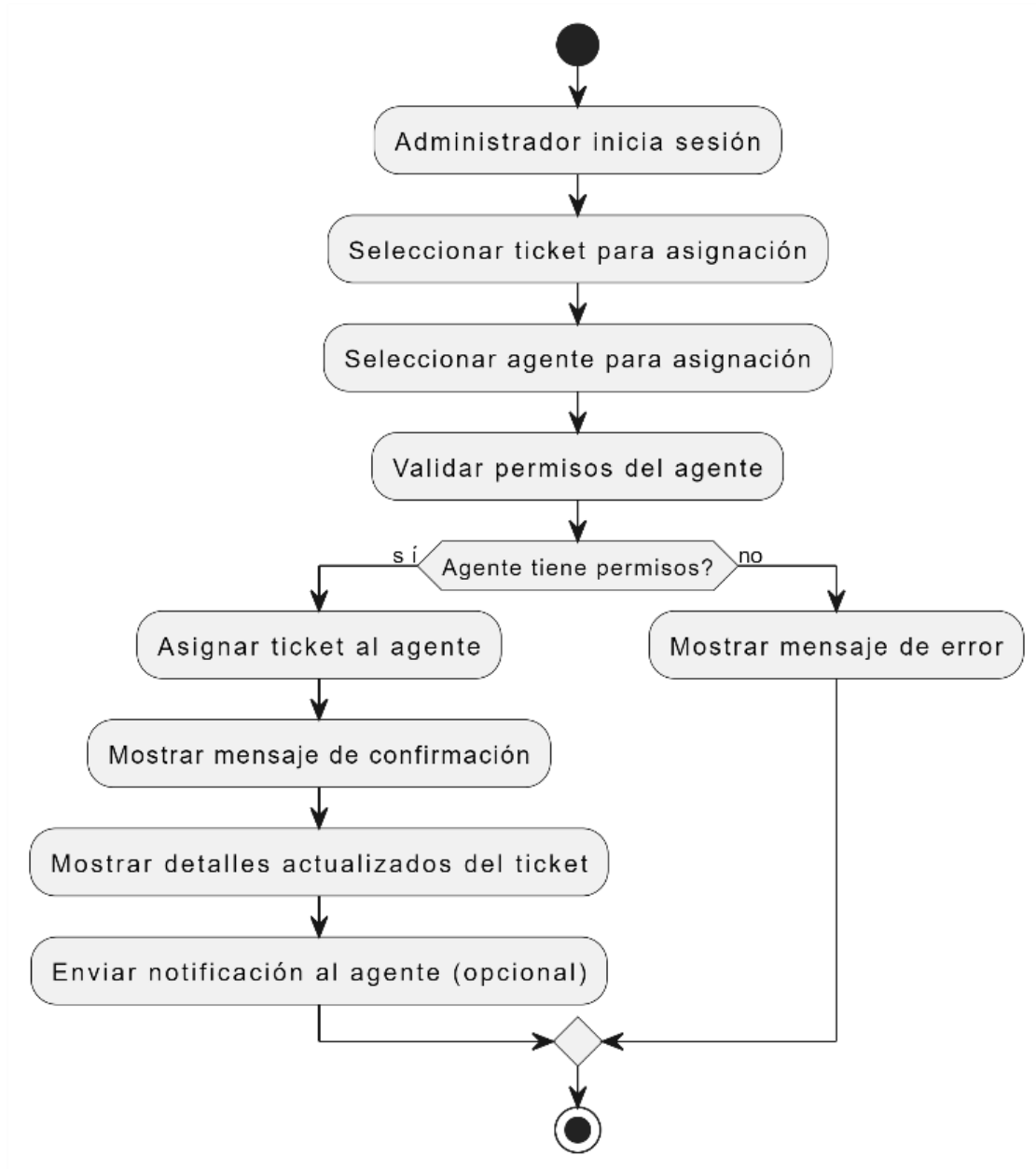
Diagrama de Actividades No.6



Nota. La figura refleja el diagrama de actividades correspondiente a ver tickets asignados como administrador

Figura 19

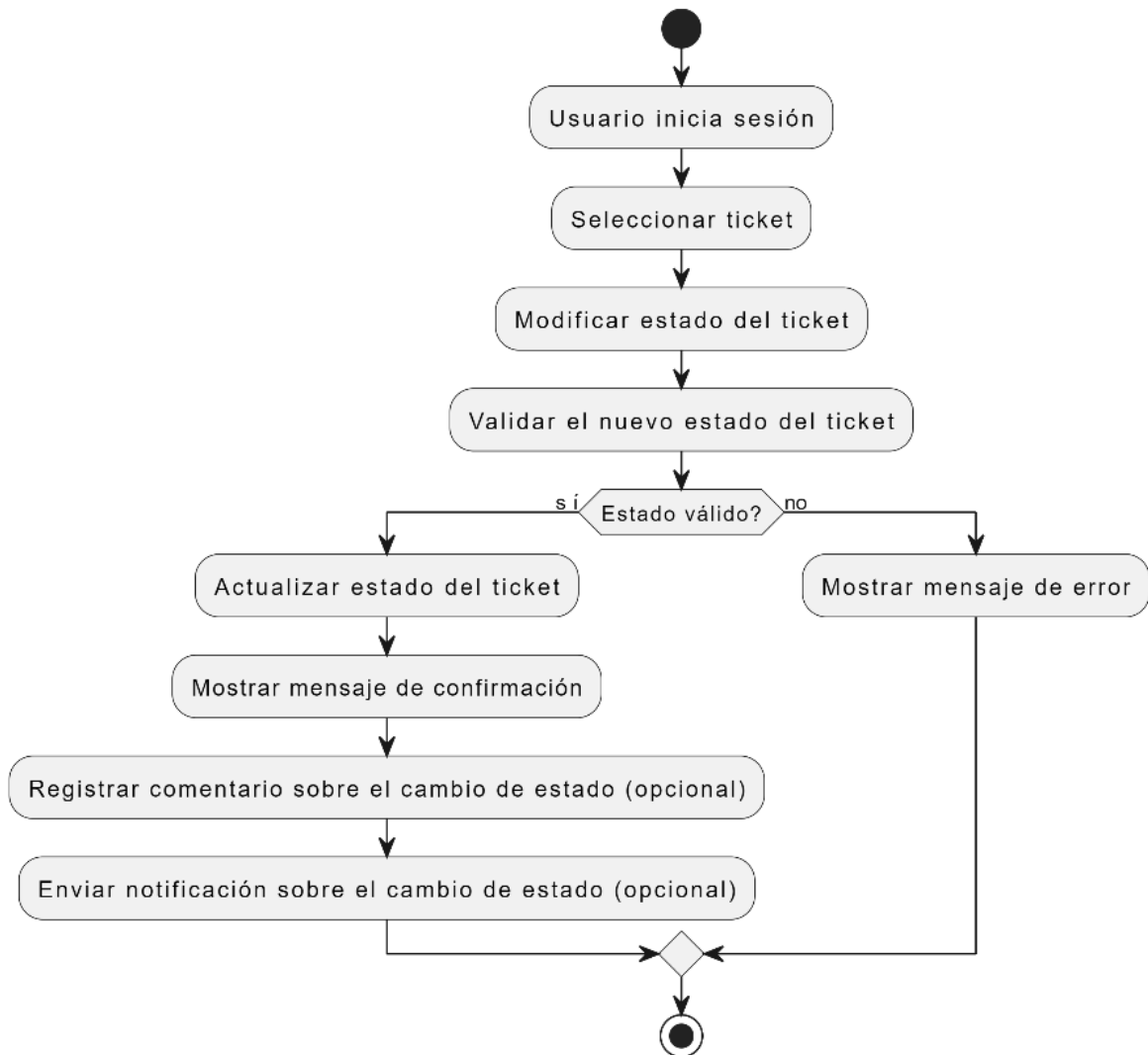
Diagrama de Actividades No.7



Nota. La figura ilustra el diagrama de actividades correspondiente a asignar un agente a un ticket.

Figura 20

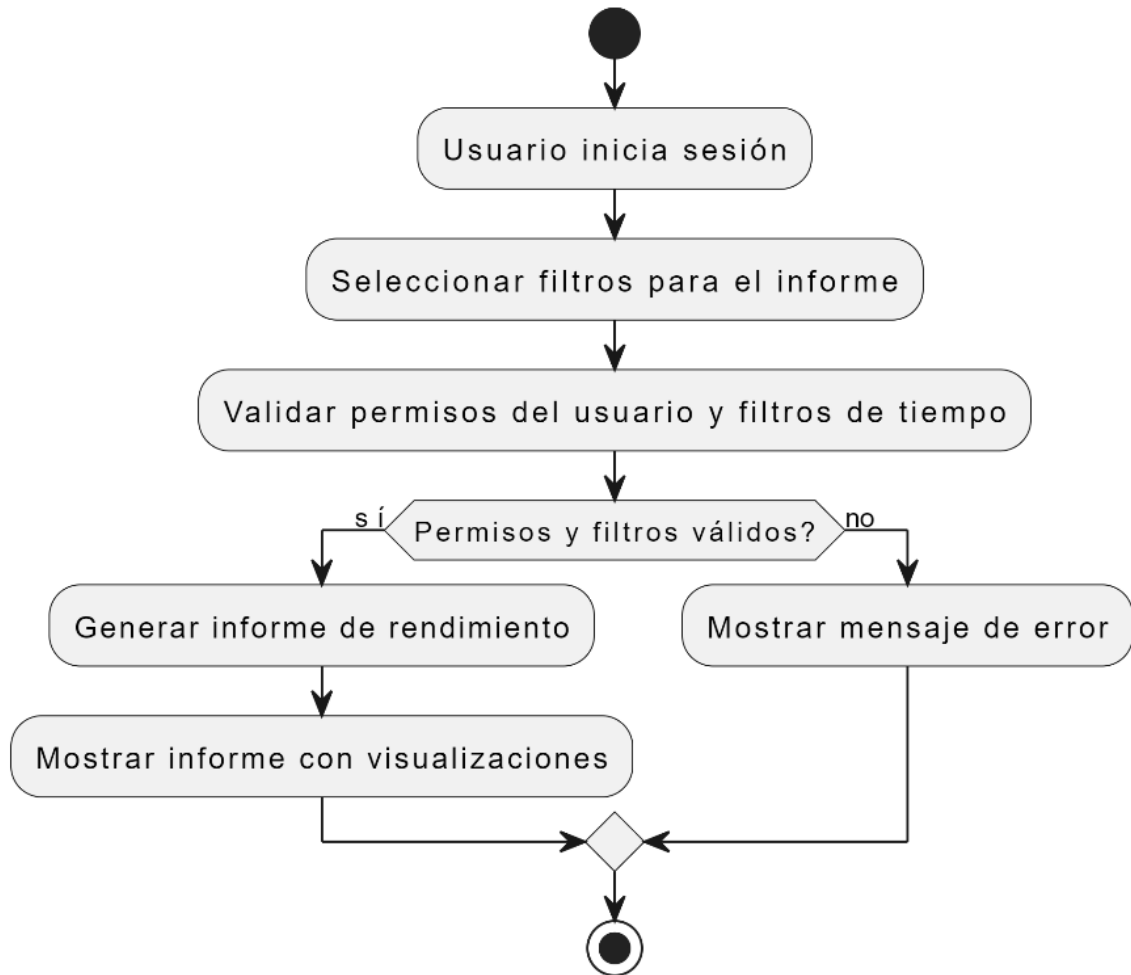
Diagrama de Actividades No.8



Nota. La figura ilustra el diagrama de actividades correspondiente a modificar el estado del ticket.

Figura 21

Diagrama de Actividades No.9



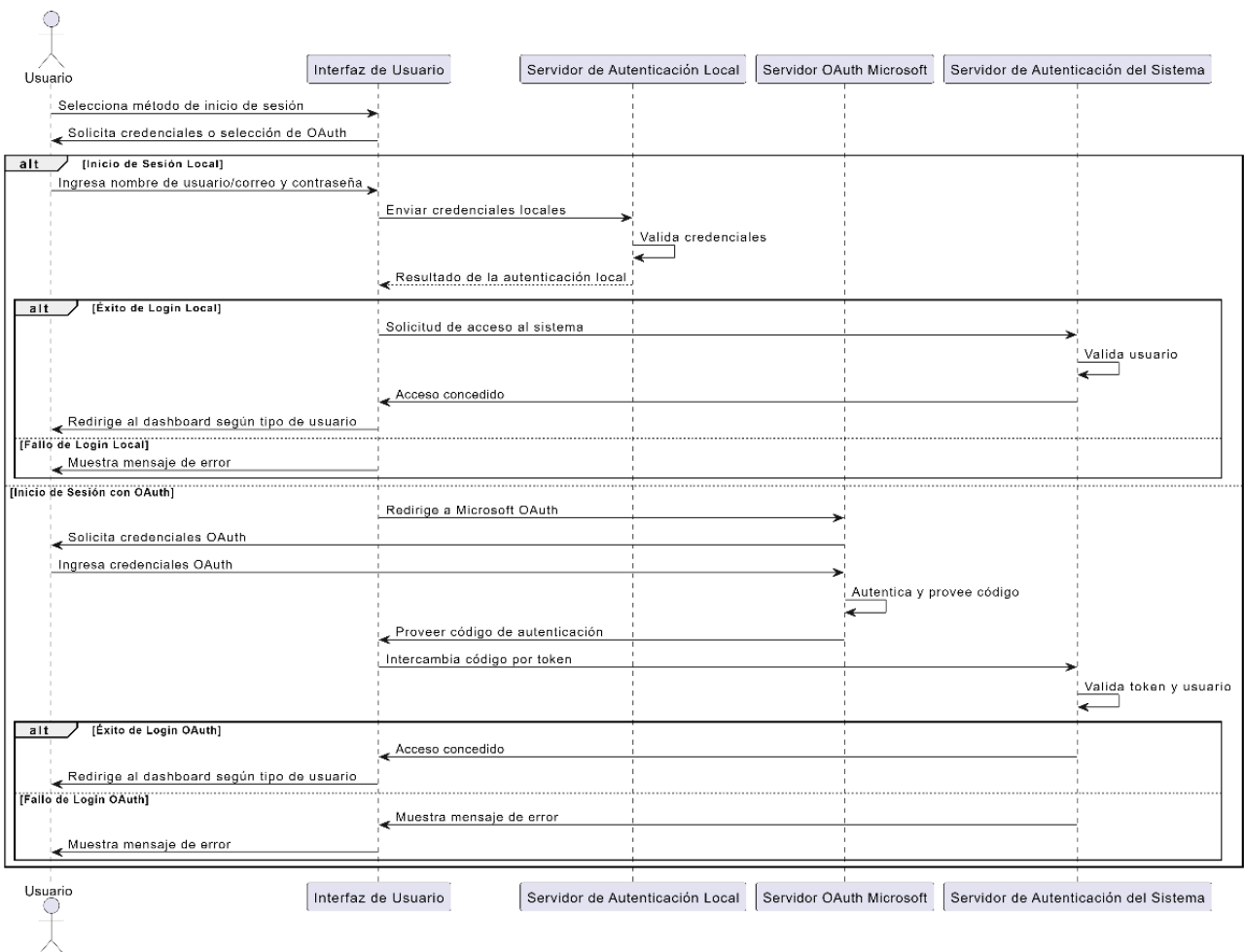
Nota. La figura muestra el diagrama de actividades correspondiente a la generación de informes.

Diagramas de Secuencia

Representan la interacción entre usuarios y el sistema a través de sus capas, mostrando el orden de llamadas y respuestas para entender la lógica de negocio y procesos.

Figura 22

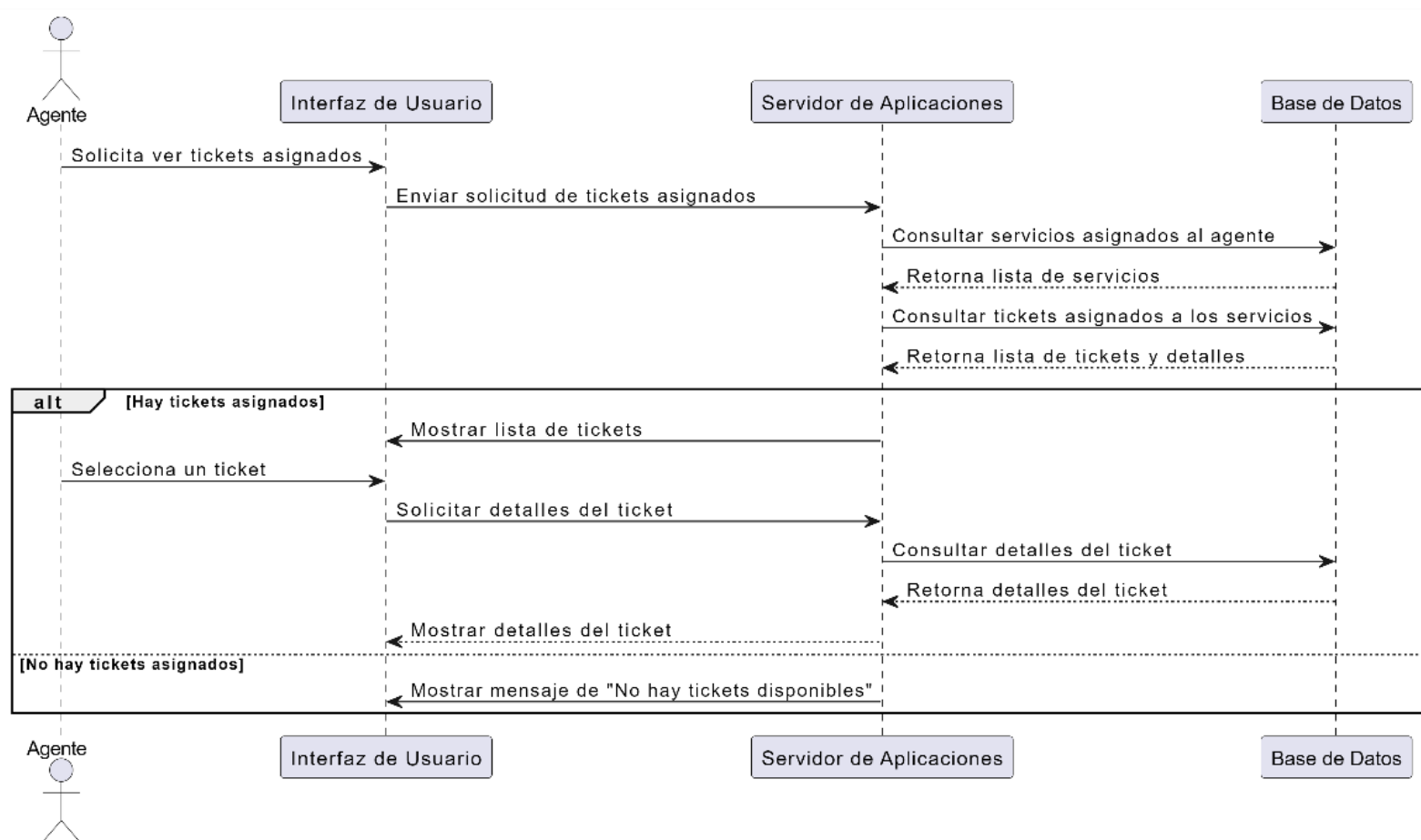
Diagrama de Secuencia No.1



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de autenticación y autorización.

Figura 23

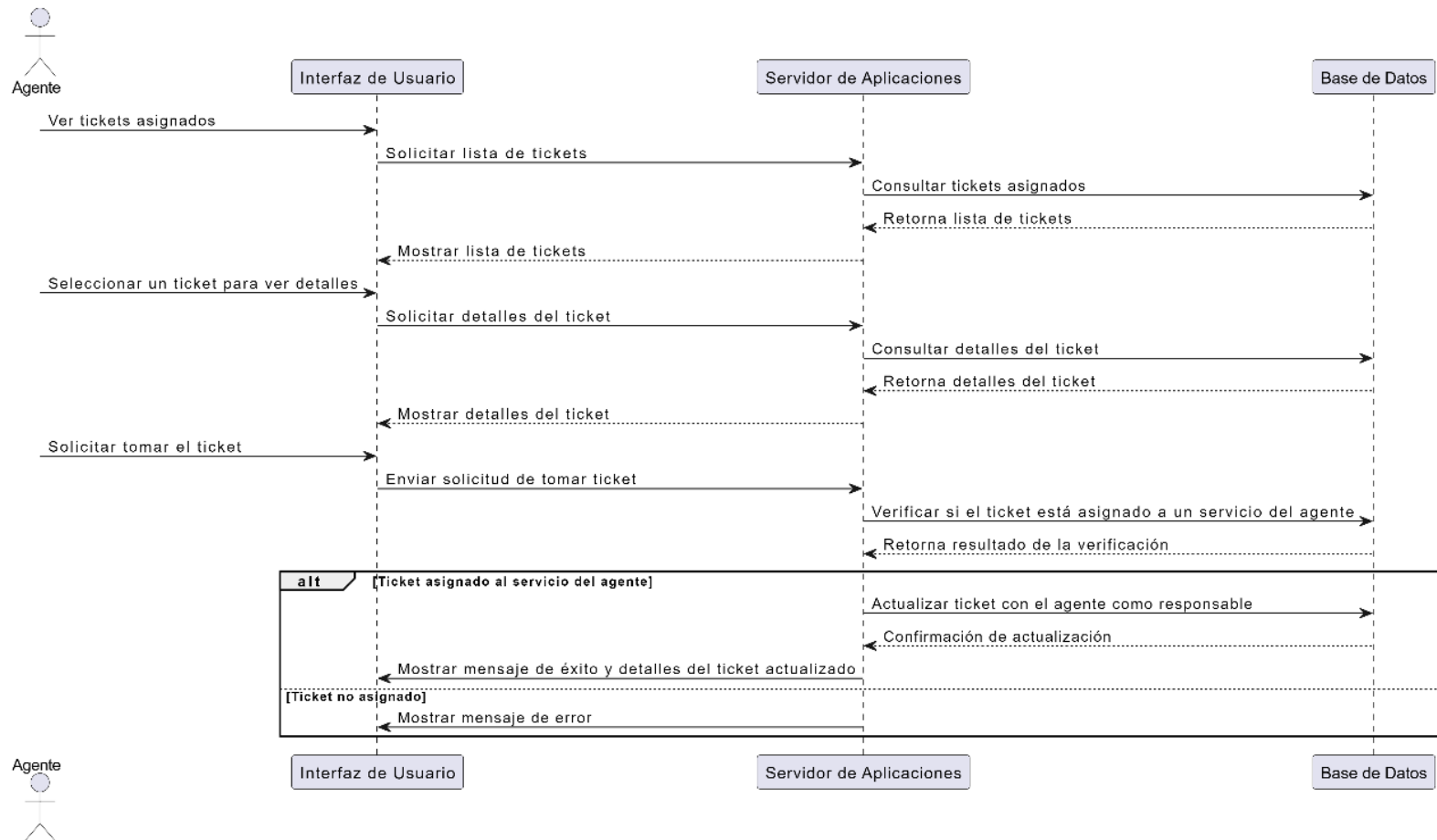
Diagrama de Secuencia No. 2



Nota. La figura ilustra el diagrama de secuencia correspondiente a la solicitud de ver tickets asignados.

Figura 24

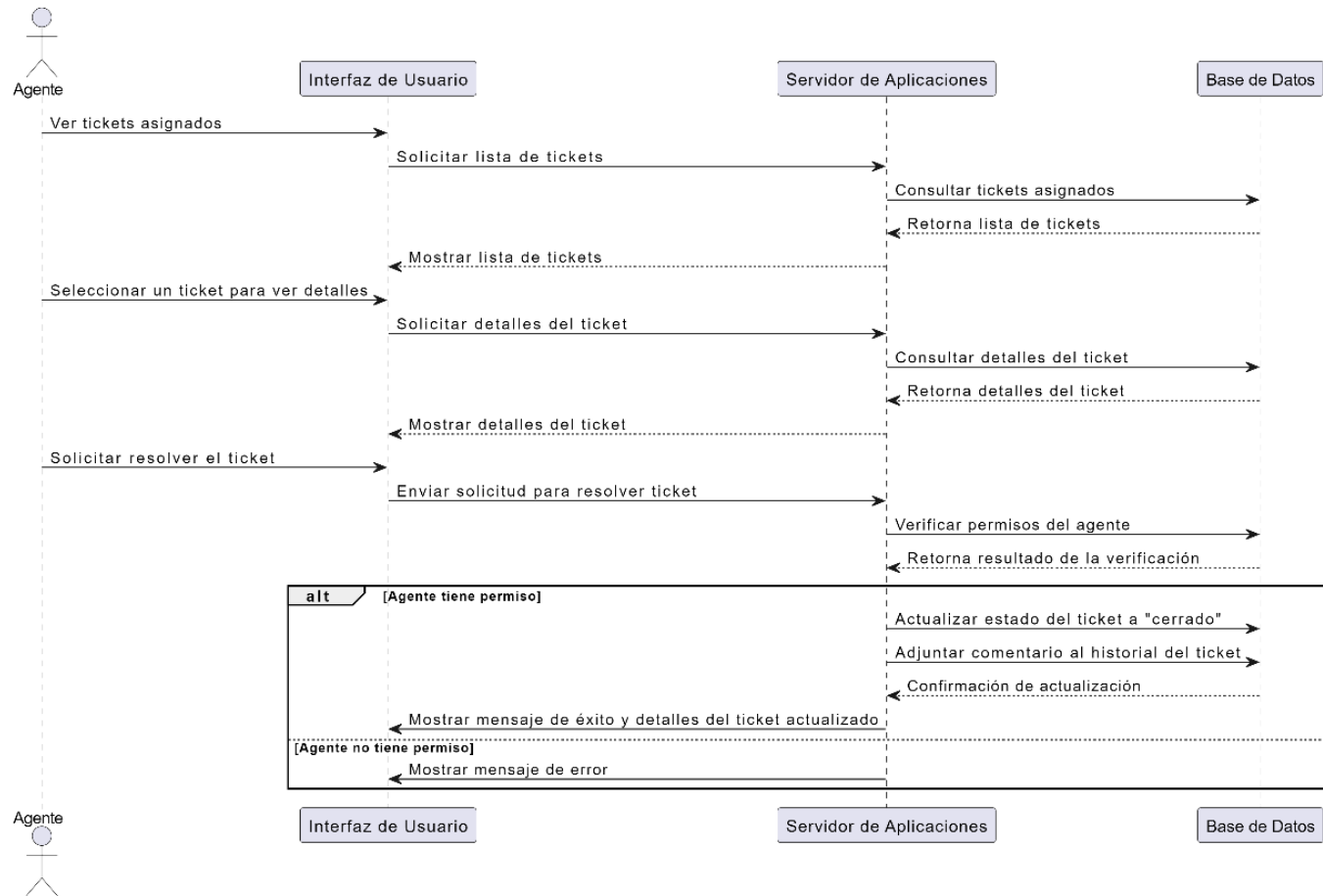
Diagrama de Secuencia No. 3



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de tomar tickets.

Figura 25

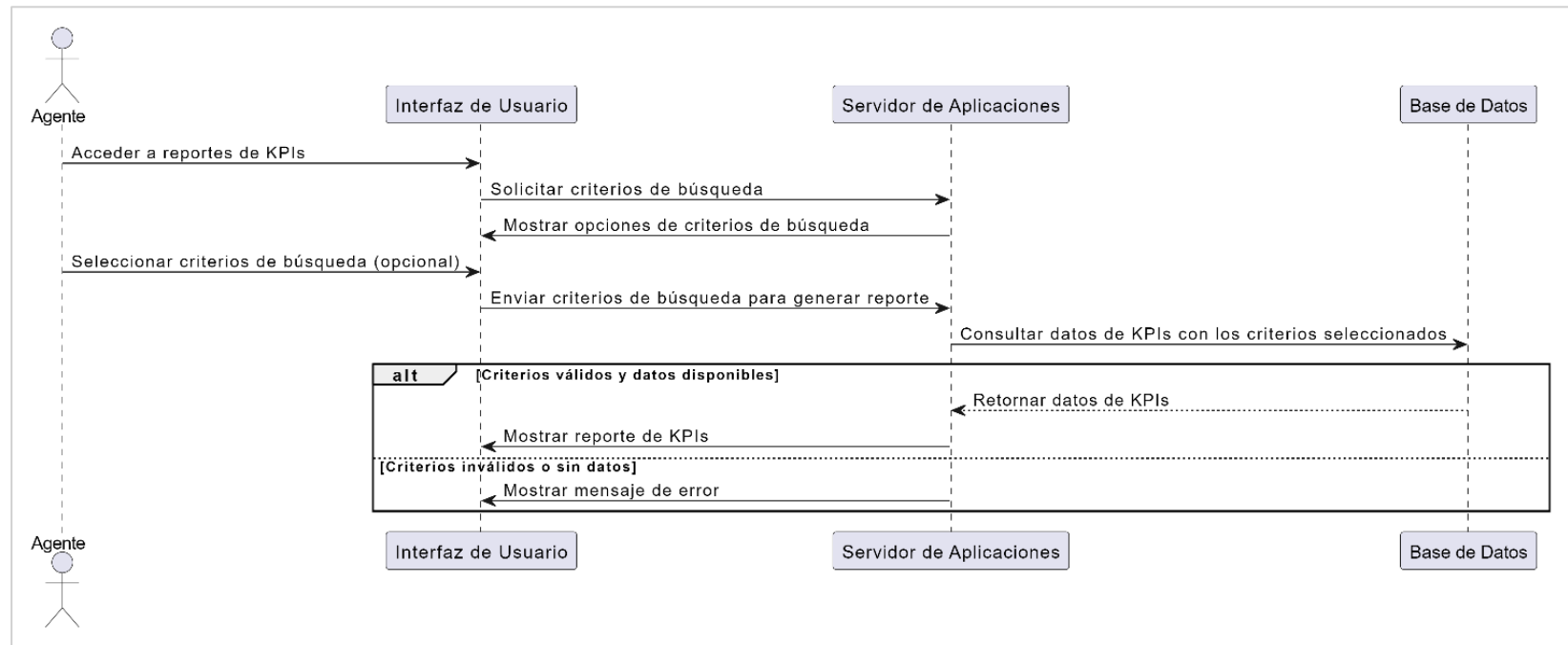
Diagrama de Secuencia No.4



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de resolución

Figura 26

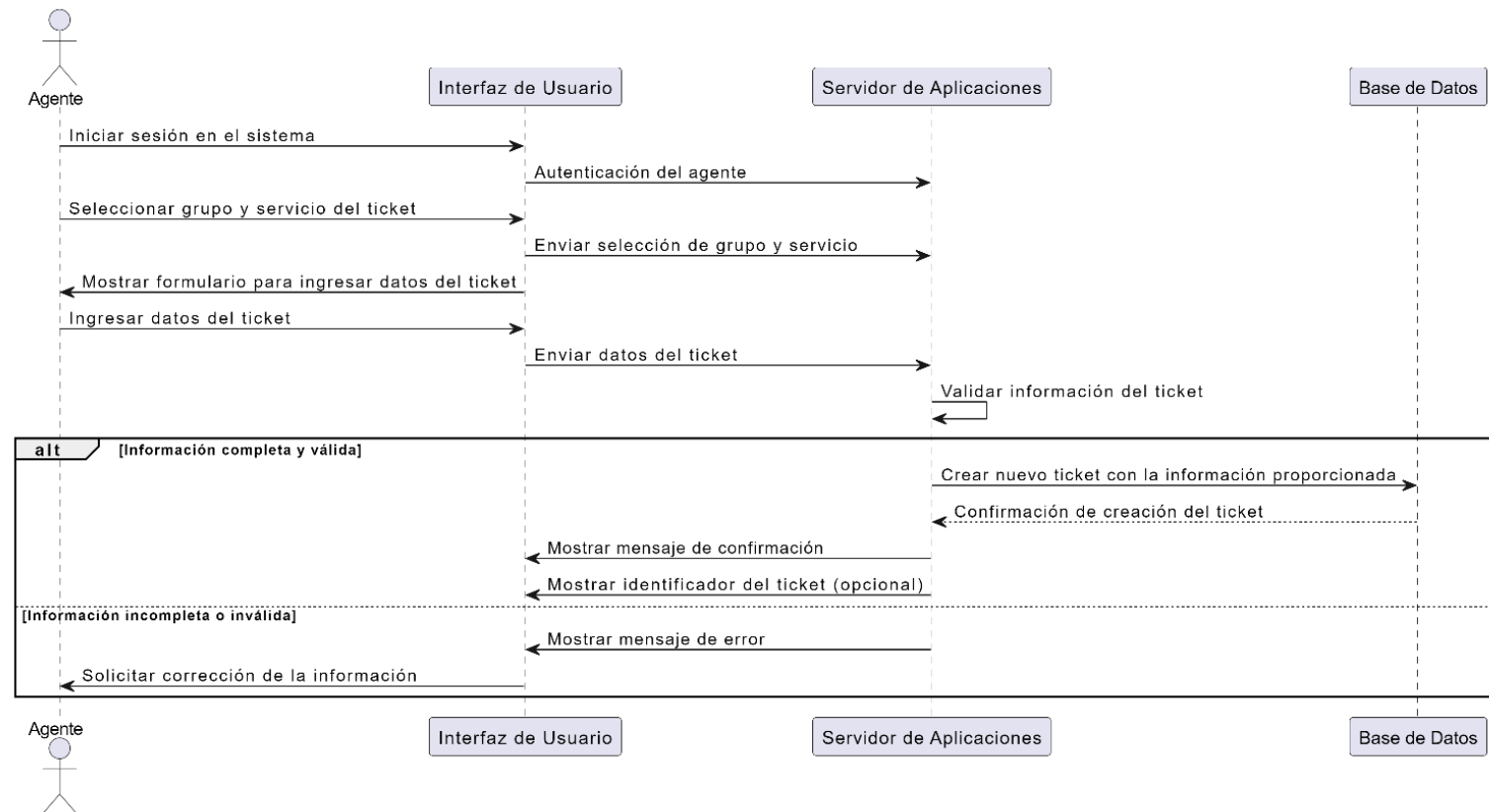
Diagrama de Secuencia No.5



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de acceso a reportes.

Figura 27

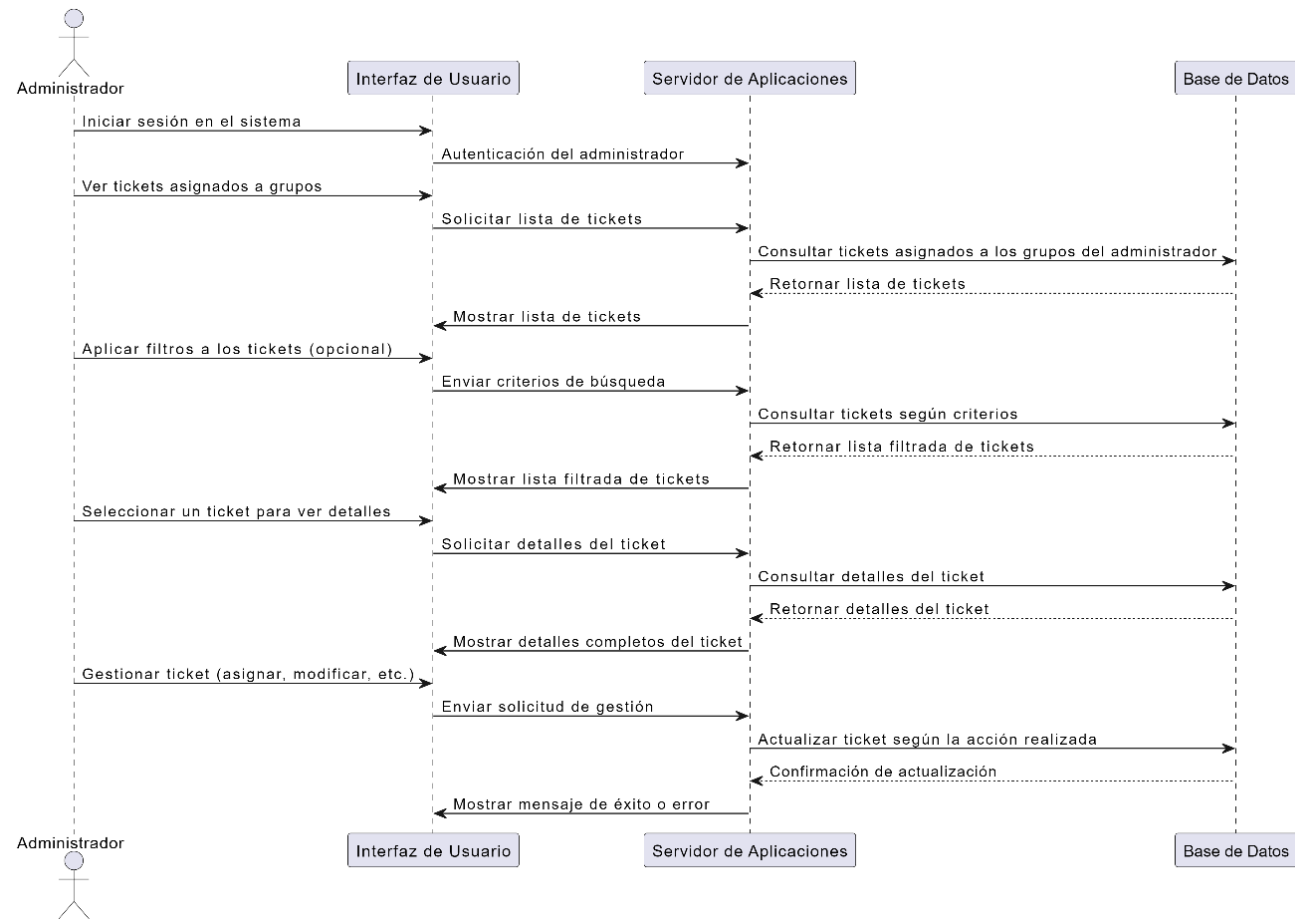
Diagrama de Secuencia No.6



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de enviar tickets a un grupo específico.

Figura 28

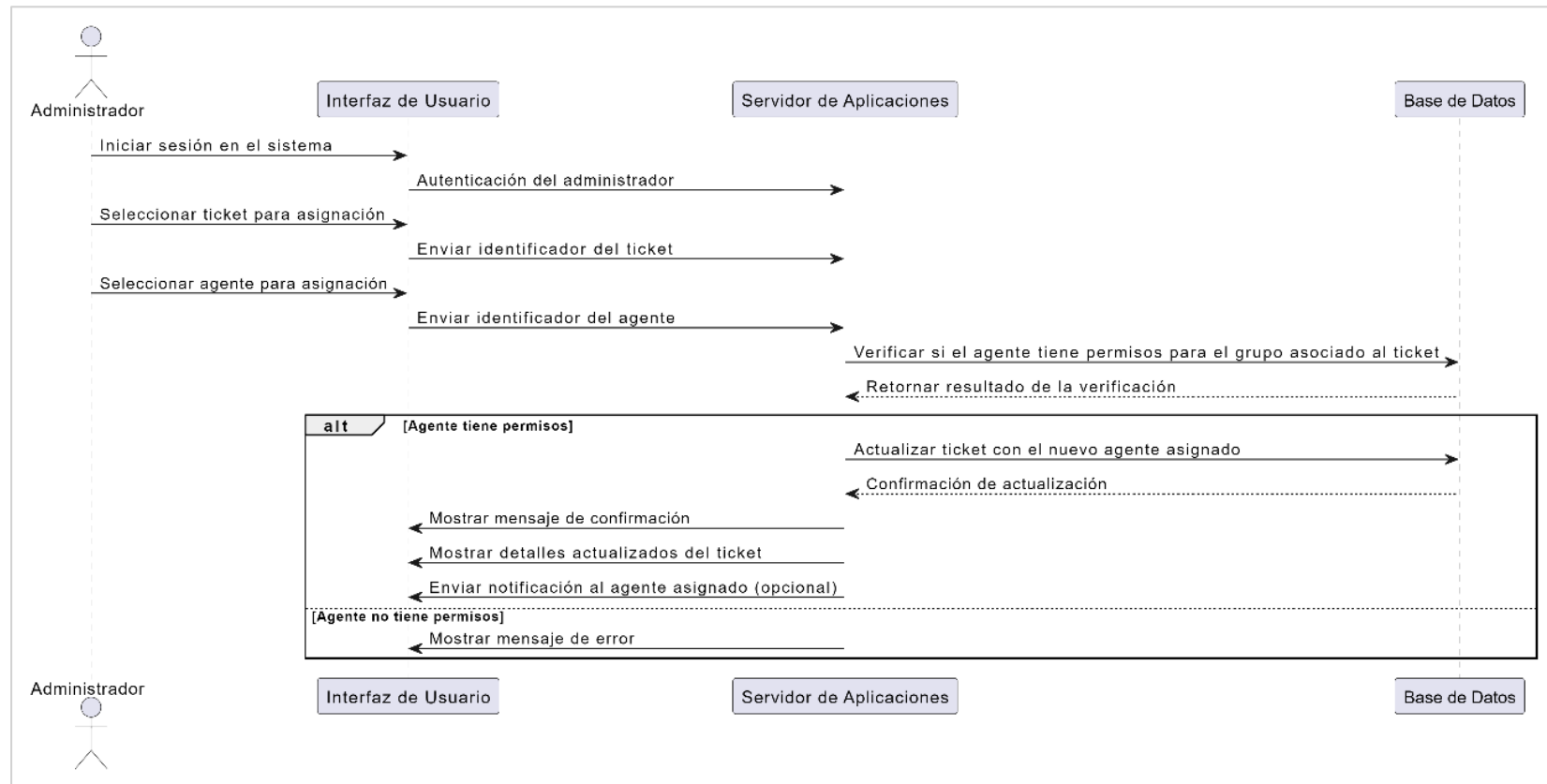
Diagrama de Secuencia No.7



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de

Figura 29

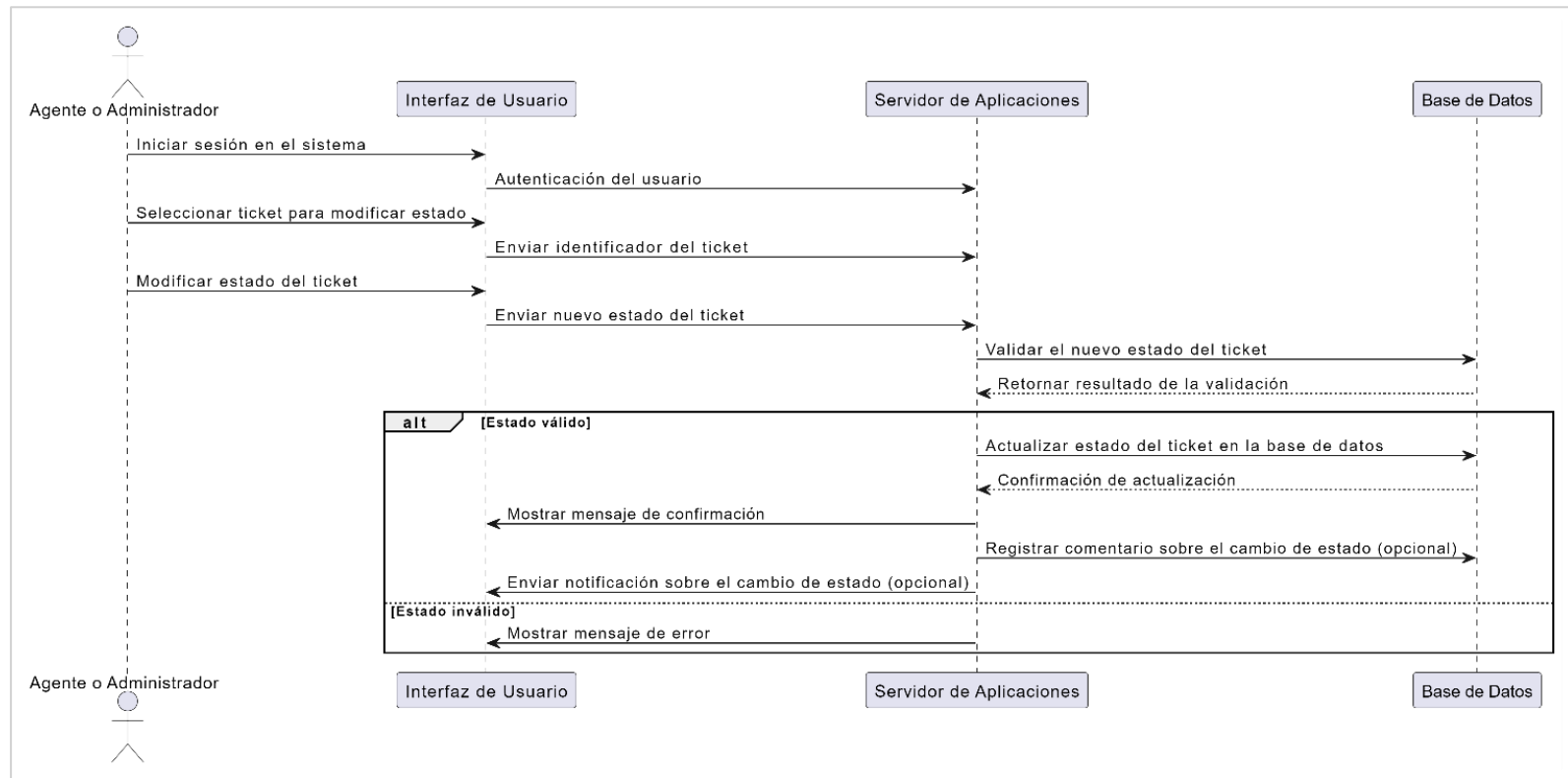
Diagrama de Secuencia No.8



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de asignar un agente como administrador.

Figura 30

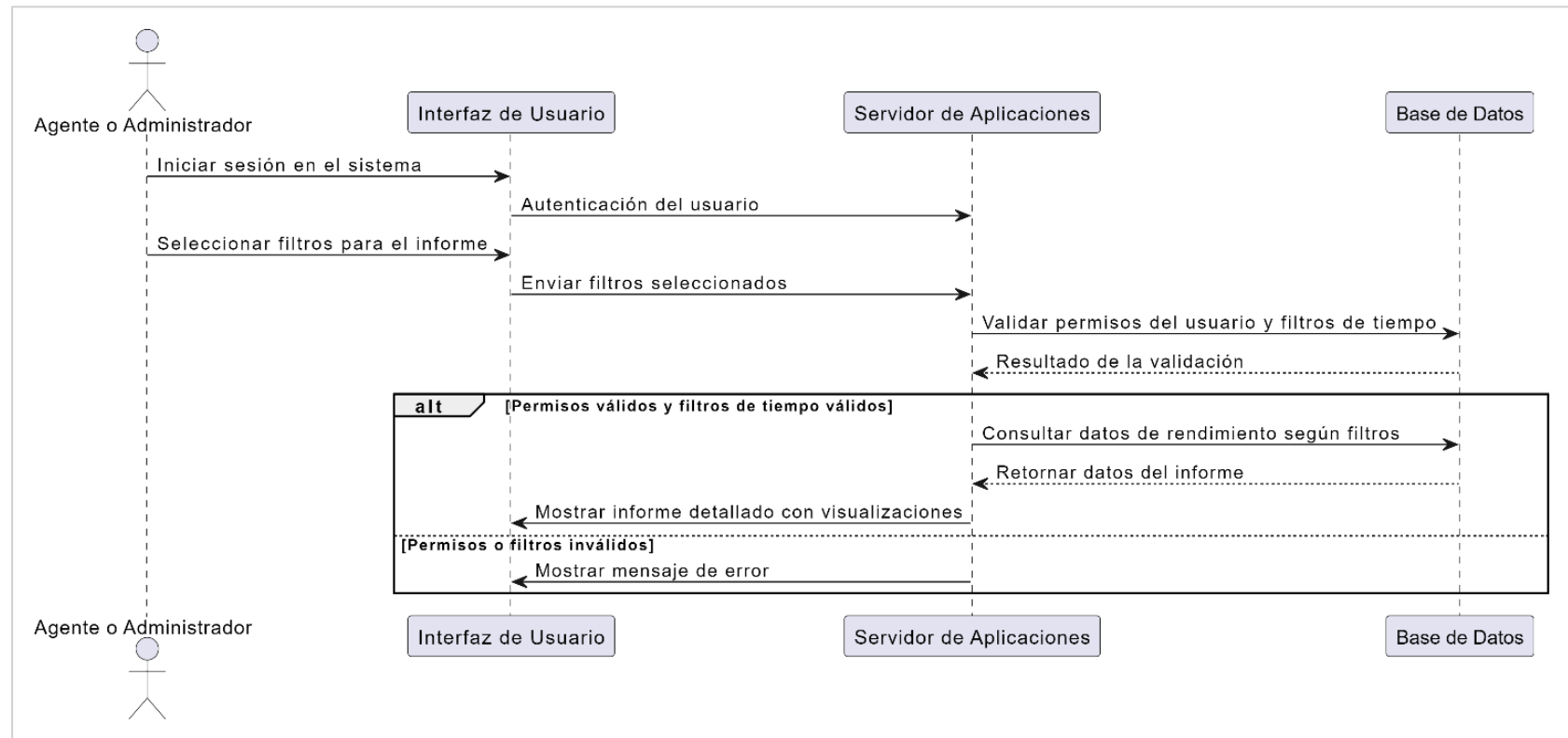
Diagrama de Secuencia No.9



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de modificar el estado de un ticket.

Figura 31

Diagrama de Secuencia No.10



Nota. La figura ilustra el diagrama de secuencia correspondiente al proceso de generar informes mediante filtros.

6.3.1.3 Capa de Infraestructura (Infrastructure)

La Capa de Infraestructura es un componente crítico en la arquitectura del sistema Helpdesk, ya que se encarga de gestionar la comunicación con los recursos externos, como bases de datos, servicios web y sistemas de almacenamiento. Esta capa proporciona los medios necesarios para que las otras capas del sistema accedan a estos recursos de manera eficiente y segura.

Responsabilidades de la Capa de Infraestructura

La Capa de Infraestructura tiene varias responsabilidades clave, que incluyen:

- **Gestión de Datos:** Esta capa es responsable de la implementación de repositorios que manejan las operaciones de acceso a los datos. Esto incluye la creación, lectura, actualización y eliminación (CRUD) de entidades relacionadas con los tickets, usuarios y otros elementos del sistema.
- **Integración con Servicios Externos:** La infraestructura se encarga de la comunicación con servicios externos, como APIs de terceros, sistemas de autenticación y otras aplicaciones que puedan ser necesarias para el funcionamiento del sistema Helpdesk.
- **Configuración y Gestión de Recursos:** Esta capa gestiona la configuración de la infraestructura necesaria, como la conexión a bases de datos, la configuración de servicios en la nube y otros recursos técnicos que soportan el funcionamiento del sistema.

Repositorios

- Los repositorios en la Capa de Infraestructura son implementaciones concretas de las interfaces definidas en la Capa de Dominio. Estos repositorios son responsables de:
- Acceso a Bases de Datos: Proporcionan métodos para acceder y manipular los datos en la base de datos, utilizando tecnologías como Entity Framework.
- Implementación de Patrones de Diseño: Se pueden aplicar patrones de diseño, como el patrón Unit of Work y el patrón Repository, para facilitar la gestión de transacciones y el acceso a los datos.

Comunicación con la Capa de Aplicación

La Capa de Infraestructura interactúa con la Capa de Aplicación a través de interfaces, lo que permite:

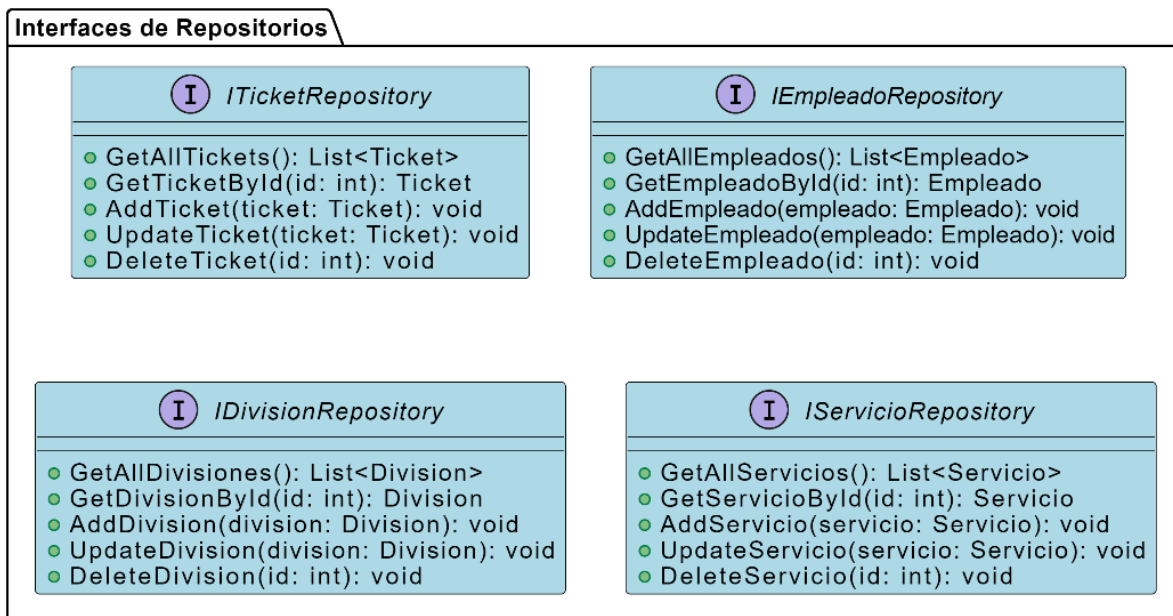
- Desacoplamiento: Al usar interfaces, se logra un desacoplamiento entre la lógica de negocio y la lógica de acceso a datos, lo que facilita la realización de pruebas y el mantenimiento del sistema.
- Flexibilidad: Se puede cambiar la implementación de los repositorios sin afectar a las capas superiores, lo que permite una fácil adaptación a nuevas tecnologías o cambios en los requisitos del sistema.

Diagramas de Infraestructura

A continuación, se presentarán los diagramas relevantes que ilustran la arquitectura de la Capa de Infraestructura, incluyendo el Diagrama de Arquitectura, las Interfaces de Repositorios, las Implementaciones de Repositorios y el Diagrama de base de datos. Estos diagramas proporcionarán una visión clara de la estructura y la interacción entre los componentes, facilitando la comprensión del flujo de datos y las interacciones técnicas en el sistema Helpdesk.

Figura 32

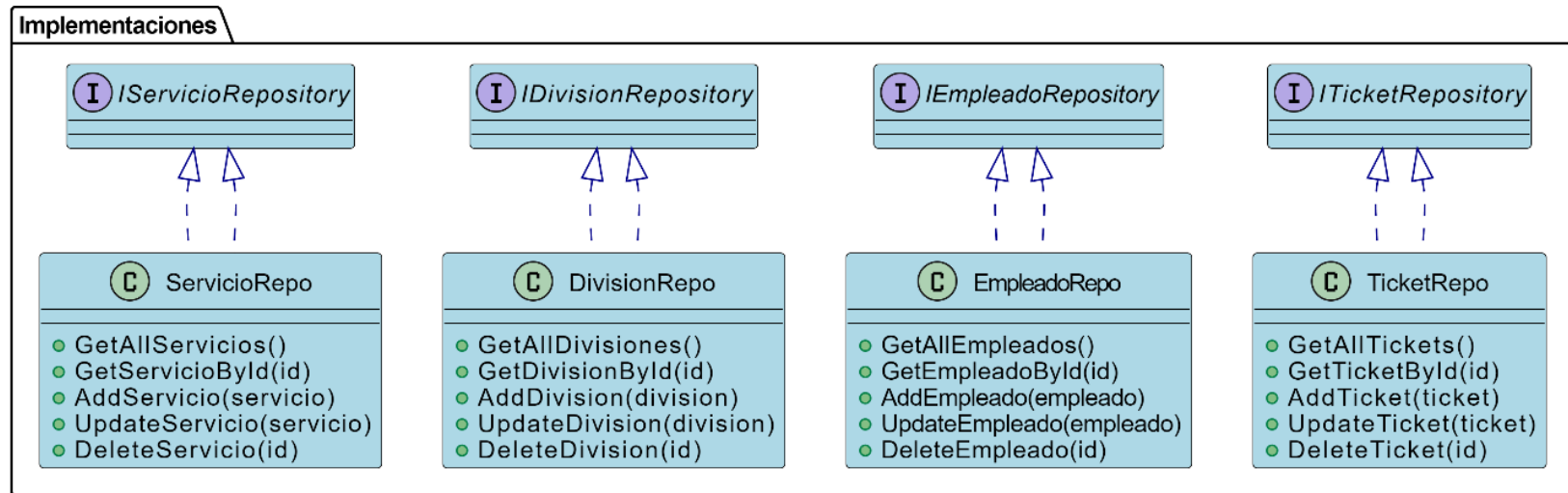
Diagrama de Interfaces de Repositorios



Nota. La figura muestra un conjunto de interfaces de repositorios para gestionar entidades como Ticket, Empleado, División y Servicio. Estas interfaces encapsulan las operaciones de acceso a datos para mantener un diseño limpio y desacoplado.

Figura 33

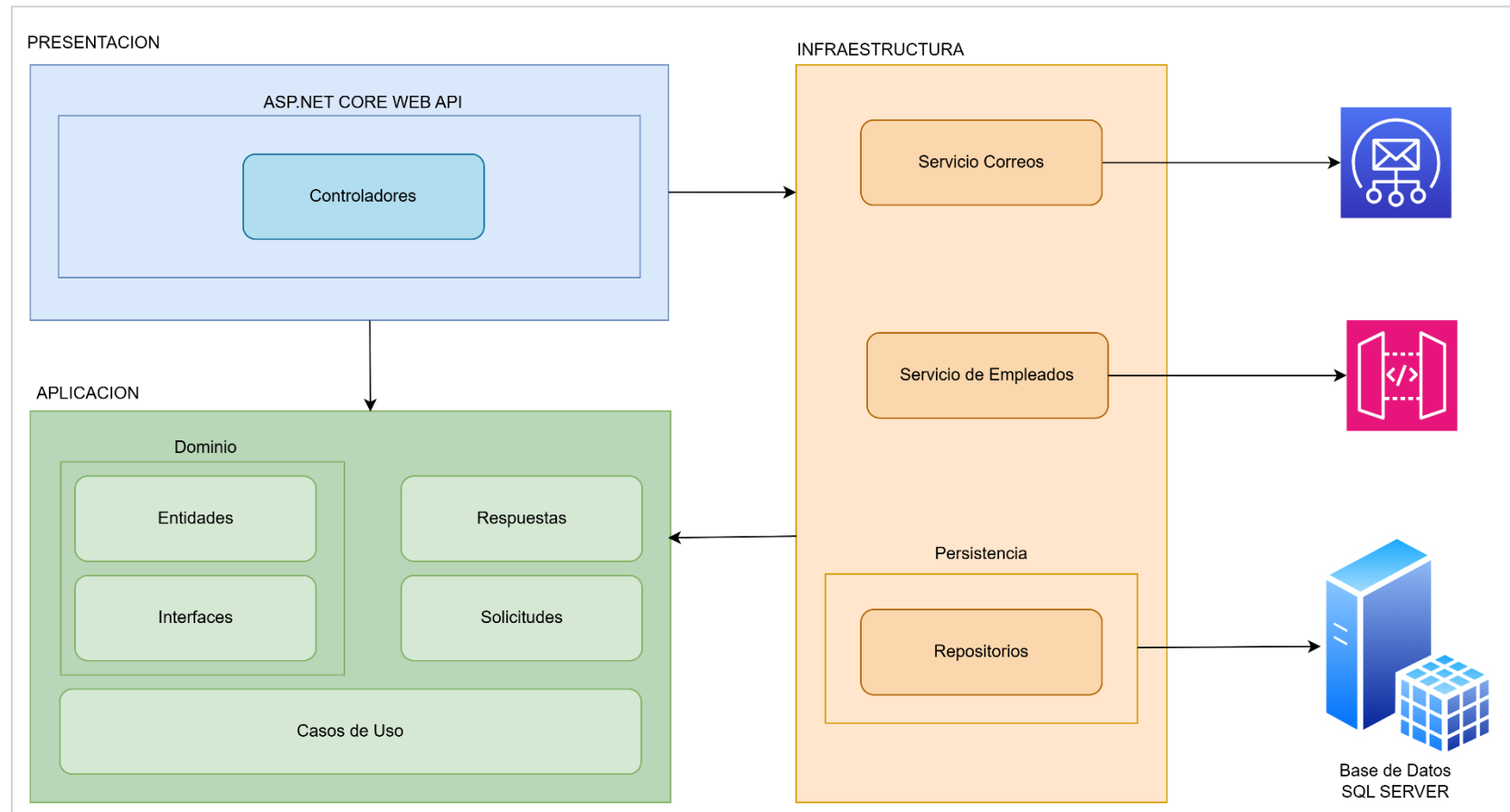
Diagrama de Implementaciones de Repositorios



Nota. La imagen muestra las implementaciones de los repositorios

Figura 34

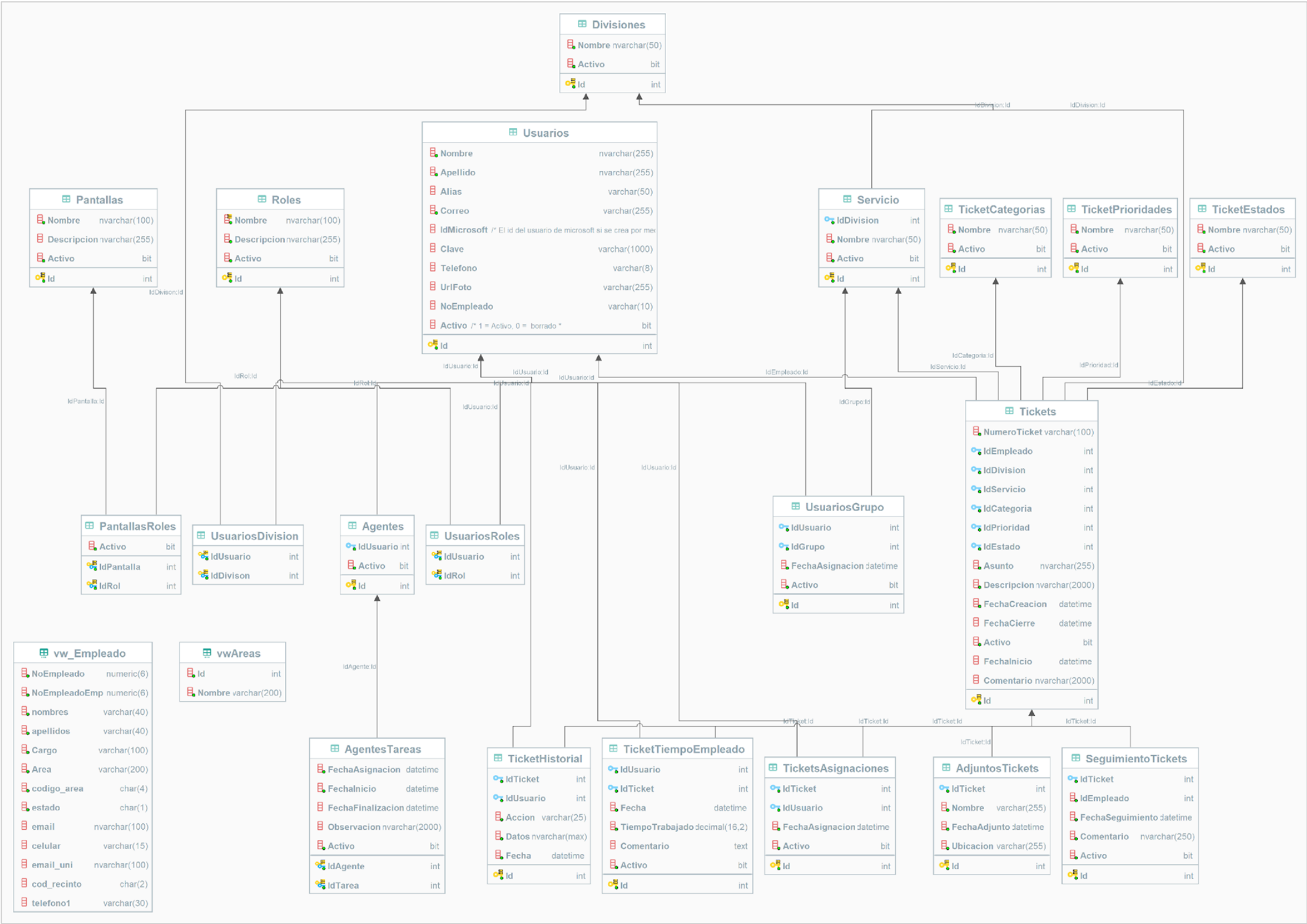
Diagrama de Arquitectura de Software



Nota. La figura ilustra una arquitectura limpia, separando responsabilidades en capas: API, dominio (núcleo de negocio) y servicios/persistencia, asegurando independencia y bajo acoplamiento.

Figura 35

Diagrama de Base de Datos



Nota. La figura ilustra el esquema de base de datos para el Sistema Helpdesk.

6.3.1.4 Capa de Presentación (Presentation)

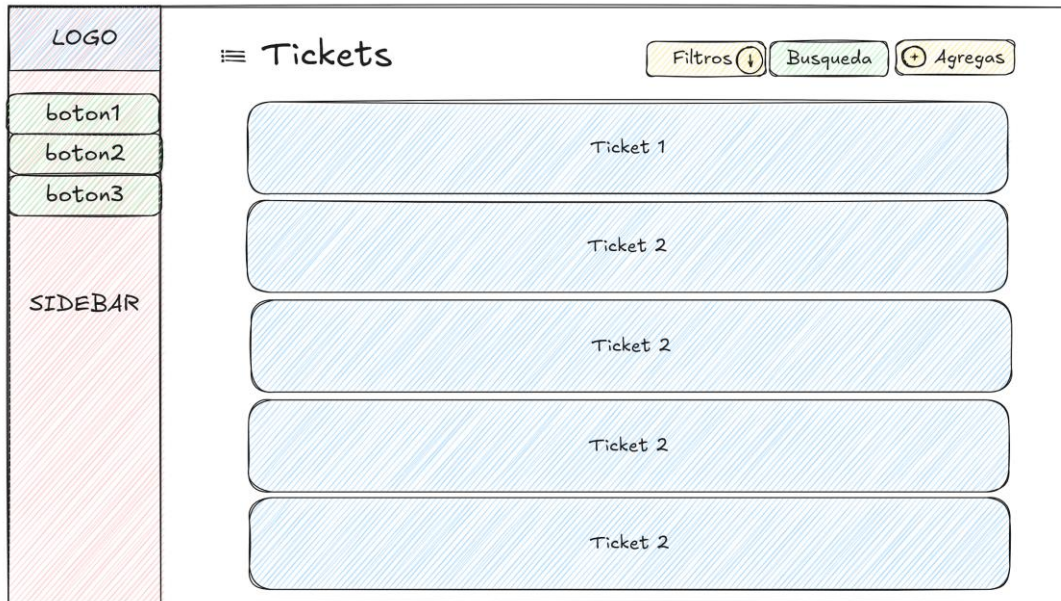
La Capa de Presentación es la interfaz del sistema Helpdesk que permite a los usuarios interactuar con la aplicación. Esta capa puede implementarse como una API que proporciona servicios a otras aplicaciones o como un frontend que ofrece una interfaz gráfica directa al usuario. Su objetivo principal es facilitar la comunicación entre el usuario y la lógica del negocio, asegurando que las funcionalidades del sistema sean accesibles y fáciles de usar.

La Capa de Presentación asume varias responsabilidades clave, que incluyen:

- **Interfaz de Usuario:** Proporciona una interfaz gráfica intuitiva (en el caso del frontend) o endpoints RESTful (en el caso de la API) que permiten a los usuarios realizar acciones como crear, asignar y cerrar tickets. La interfaz debe ser amigable y responder de manera eficiente a las acciones del usuario.
- **Validación de Entradas:** Realiza la validación de los datos introducidos por los usuarios antes de enviarlos a la Capa de Aplicación. Esto ayuda a prevenir errores y asegurar que la información que se procesa sea correcta.
- **Presentación de Resultados:** Muestra los resultados de las acciones realizadas, como la confirmación de la creación de un ticket o el estado de un ticket existente. Esta retroalimentación es crucial para la experiencia del usuario.
- **Gestión de Navegación:** Controla la navegación entre las diferentes vistas y componentes de la aplicación (en el caso del frontend) o facilita el acceso a recursos a través de la API.

Figura 36

Wireframe de Modulo Gestión de Tickets



Nota. La figura ilustra el wireframe de una interfaz con un menú lateral (logo, botones y sidebar) y una sección principal con una lista de tickets, acompañada de opciones para filtros, búsqueda y agregar en la parte superior.

Interacción con la Capa de Aplicación

La Capa de Presentación se comunica con la Capa de Aplicación para ejecutar los casos de uso. Esta interacción se basa en:

- **Llamadas a Servicios:** La Capa de Presentación invoca servicios de la Capa de Aplicación para llevar a cabo acciones solicitadas por los usuarios. Por ejemplo, al crear un nuevo ticket, la interfaz de usuario (o la API) enviará los datos a la Capa de Aplicación, que procesará la solicitud.

Recepción de Respuestas: Una vez que la Capa de Aplicación procesa la solicitud, devuelve una respuesta a la Capa de Presentación, que a su vez se encarga de mostrar la información correspondiente al usuario.

6.4 FASE DE DESARROLLO

La fase de desarrollo del sistema Helpdesk es una de las etapas más cruciales dentro del modelo en cascada, ya que es donde se lleva a cabo la construcción del software basado en el diseño previamente establecido. Esta etapa sigue una serie de pasos que garantizan una correcta implementación de los requisitos y un desarrollo ordenado.

6.4.1 Preparación del Entorno de Desarrollo

Antes de iniciar la codificación, es esencial establecer un entorno de desarrollo que permita a los desarrolladores trabajar de manera eficiente. En el caso del sistema Helpdesk, se seleccionarán herramientas y tecnologías adecuadas para el desarrollo, como:

- **Lenguajes de programación:** Para el backend se utilizará C# con .NET 6 debido a su robustez y soporte para el desarrollo de aplicaciones escalables y seguras. Para el frontend, se utilizarán JavaScript y React para crear una interfaz de usuario dinámica y moderna.
- **Entornos de desarrollo integrados (IDE):** Los desarrolladores utilizarán herramientas como Visual Studio, Visual Studio Code y JetBrains Rider. Visual Studio y Rider se usarán principalmente para el desarrollo en C# y .NET 6, mientras que Visual Studio Code se empleará para trabajar en el frontend con JavaScript y React.
- **Control de versiones:** Para la gestión del código, se empleará Git y el repositorio será alojado en GitLab. Esta configuración permitirá llevar un control detallado de las versiones del software.
- **Base de datos:** Se configurará una base de datos SQL Server, donde se almacenará toda la información relacionada con los tickets, usuarios y administración del sistema. SQL Server garantiza una alta confiabilidad y eficiencia en la gestión de grandes volúmenes de datos.

Este entorno de desarrollo garantizará que los desarrolladores trabajen de manera coordinada y eficiente, facilitando el seguimiento de las tareas y la correcta integración del código.

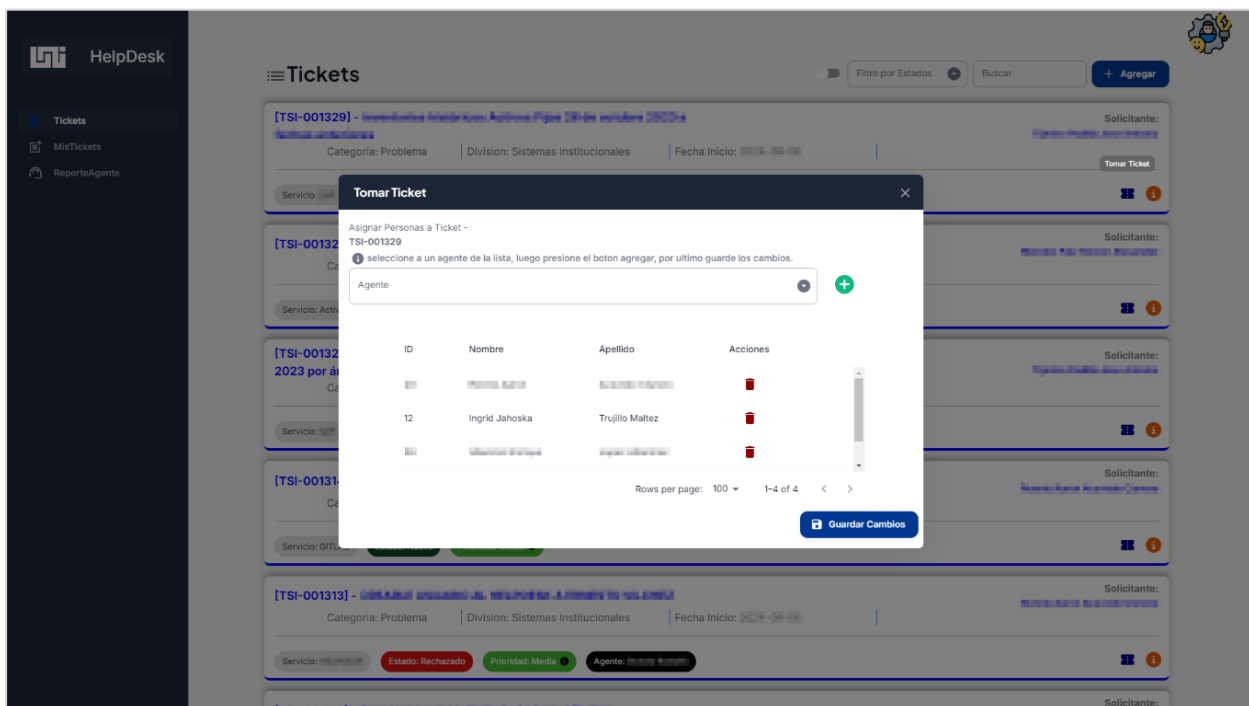
6.4.2 Implementación de los Módulos

La implementación del sistema se llevará a cabo mediante la creación de módulos independientes, cada uno representando una funcionalidad específica del sistema Helpdesk. Los módulos principales que se desarrollarán son los siguientes:

Gestión de Tickets: Este es el núcleo del sistema Helpdesk, donde los usuarios podrán crear, actualizar y cerrar tickets. Los agentes de soporte podrán gestionar el estado de los tickets y asignar prioridades según las necesidades del usuario.

Figura 37

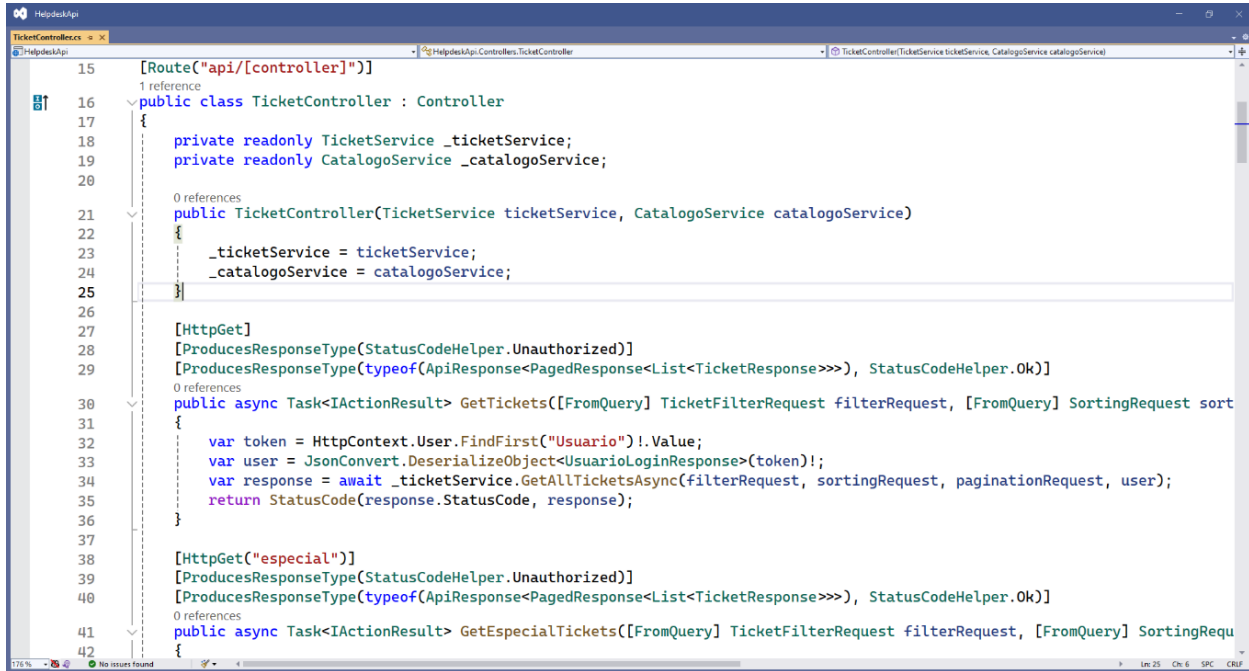
Pantalla Modulo Gestión de Tickets



Nota. La figura muestra el módulo de gestión de tickets del sistema Helpdesk.

Figura 38

Fragmento de Código Endpoints Modulo Gestión de Tickets

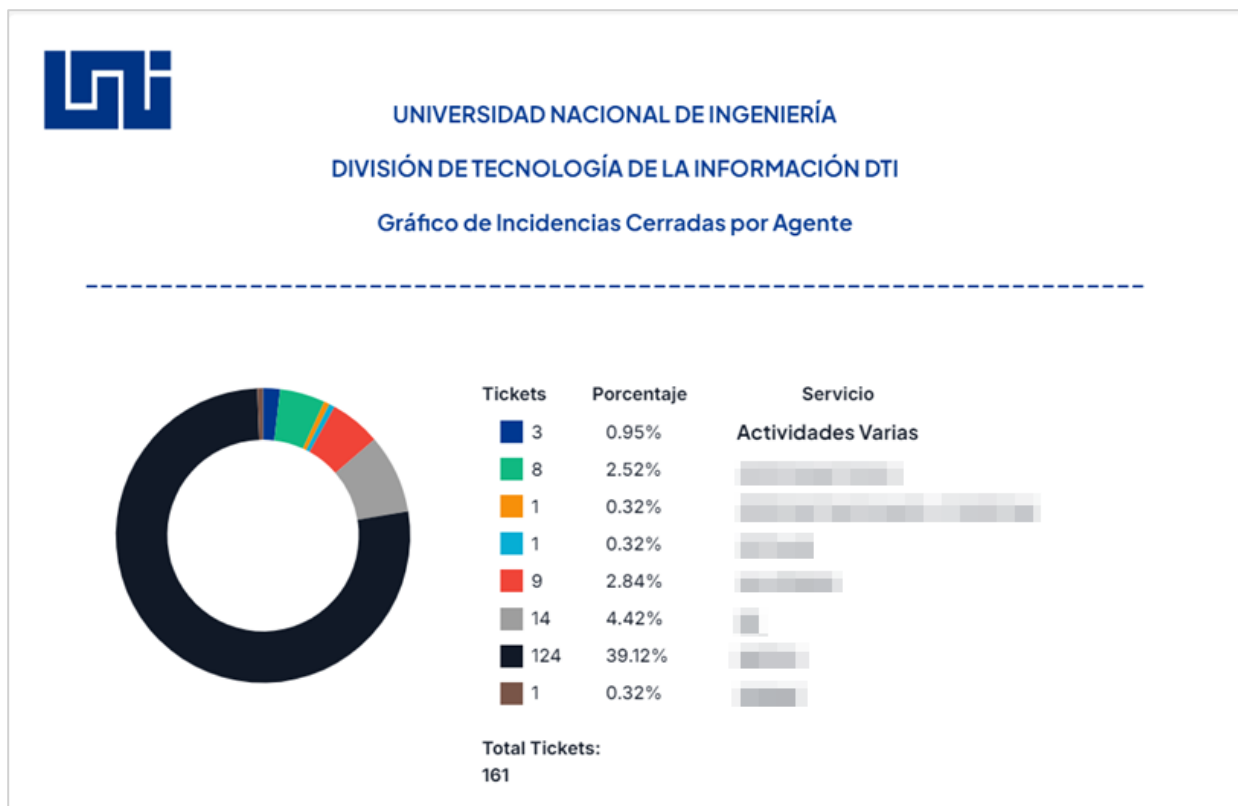


Nota. La figura ilustra una captura de pantalla del fichero TicketController.cs

Reportes: Se implementó un módulo de reportes que generará estadísticas e informes detallados sobre el número de tickets gestionados, rendimiento del equipo de soporte y otros indicadores clave para la toma de decisiones.

Figura 39

Captura de Pantalla Grafico de Incidencias Cerradas por Agente



Nota. La figura ilustra el grafico de incidencias cerradas por agente.

Figura 40

Captura de Pantalla Reporte de Incidencias Cerradas Con exito Entre fechas

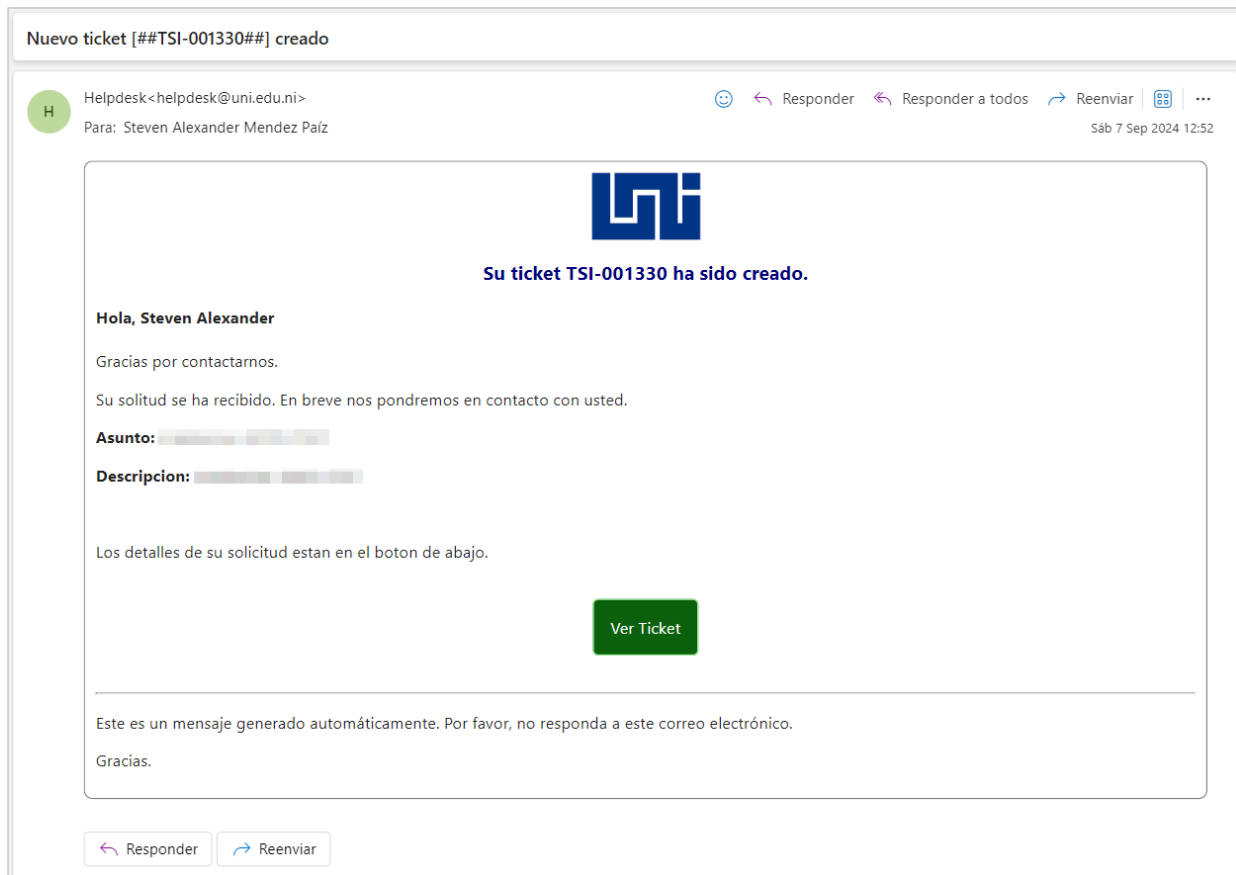
 UNIVERSIDAD NACIONAL DE INGENIERÍA DIVISIÓN DE TECNOLOGÍA DE LA INFORMACIÓN DIVISIÓN DTI				
Managua septiembre 27, 2024				
UNI-RUSB-RUPAP				
Nombre Agente: Steven Alexander Mendez				
Reporte de incidencias cerradas con éxito del 15/4/2024 al 6/9/2024				
Asunto	Solicitante	Ticket	Fecha Inicio	Fecha Fin
		TSyS-000002	15/4/2024	15/4/2024
		TSyS-000004	15/4/2024	15/4/2024
		TSyS-000018	16/4/2024	17/4/2024
		TSyS-000030	16/4/2024	17/4/2024

Nota. La figura ilustra el reporte de incidencias cerradas con éxito entre fechas.

Notificaciones: Se incluyó un módulo para notificaciones que mantendrá a los usuarios y agentes informados sobre el estado de los tickets a través de correos electrónicos.

Figura 41

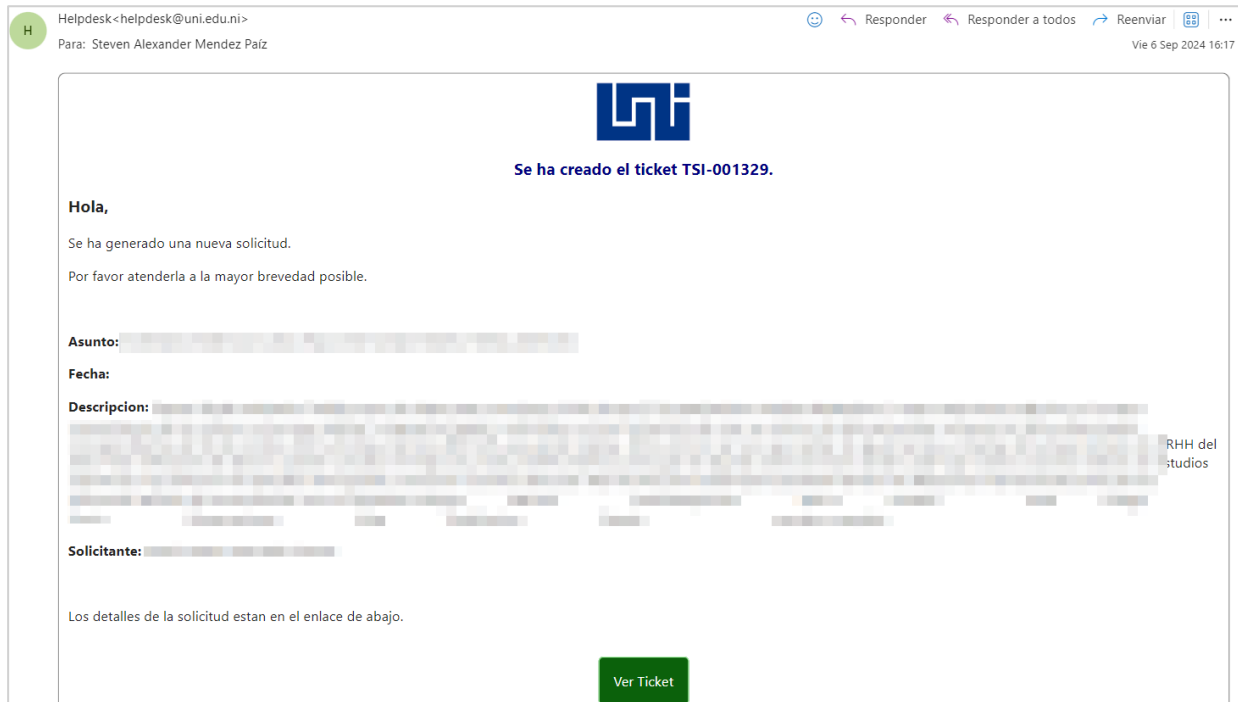
Captura de Pantalla Notificación por Correo Electrónico (Cliente)



Nota. La figura muestra el correo de notificación que se le proporciona al cliente.

Figura 42

Captura de Pantalla Notificación por Correo Electrónico (Agente)



Nota. La figura muestra el correo de notificación que se le proporciona al agente.

6.5 FASE DE PRUEBAS Y VALIDACIÓN

La fase de pruebas y validación es esencial para garantizar que el sistema Helpdesk para la División de Tecnología de Información de la Universidad Nacional de Ingeniería cumple con los requisitos establecidos y satisface las expectativas de los usuarios finales. Esta fase se ejecutará de manera sistemática y se enfocará en identificar y corregir errores, así como en validar la funcionalidad del sistema antes de su implementación final.

6.5.1 Objetivos de la Fase de Pruebas

Los objetivos principales de esta fase son:

- **Identificación y corrección de errores:** A través de pruebas exhaustivas, se detectan fallos o errores en el sistema. Estos se documentan, corrigen y se vuelven a probar para asegurar su resolución.
- **Validación de las funcionalidades del sistema:** Se comprueba que todas las funcionalidades implementadas cumplen con los escenarios de uso reales, asegurando que el sistema funcione según lo previsto.
- **Aseguramiento de la calidad:** Se verifica que el sistema cumple con los estándares de calidad en términos de rendimiento, seguridad y escalabilidad, garantizando una operación eficiente y segura.

6.5.2 Estrategia de Pruebas

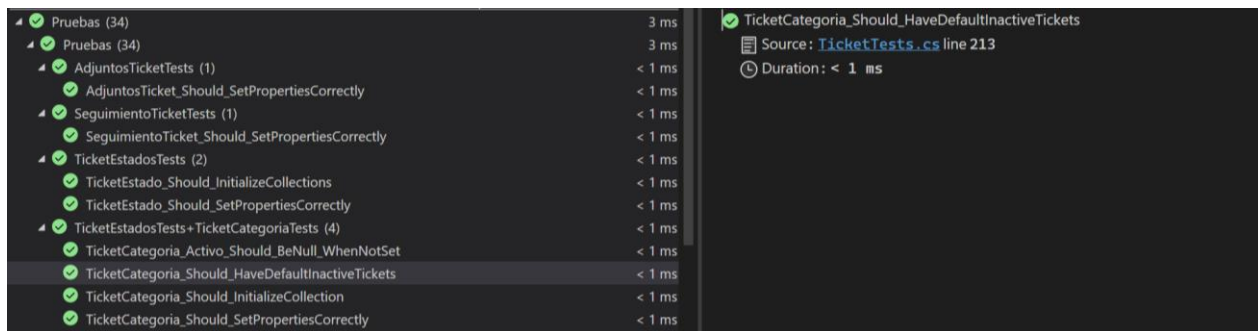
La estrategia de pruebas se desarrolla en varias etapas que abarcan desde la verificación de módulos individuales hasta la validación del sistema completo. Esto asegura que cada componente del sistema funcione de manera adecuada tanto de forma aislada como integrado con otros módulos.

6.5.2.1 Pruebas Unitarias

Una vez que cada módulo está desarrollado, se realizan pruebas unitarias para asegurar que los componentes individuales funcionen correctamente. Estas pruebas verifican que cada unidad de código cumpla con los requisitos establecidos y se comporte como se espera. Se utilizan frameworks de pruebas automatizadas como **xUnit** para facilitar este proceso, garantizando la eficacia y repetibilidad de las pruebas.

Figura 43

Captura de Pantalla de Resultado de Pruebas Unitarias



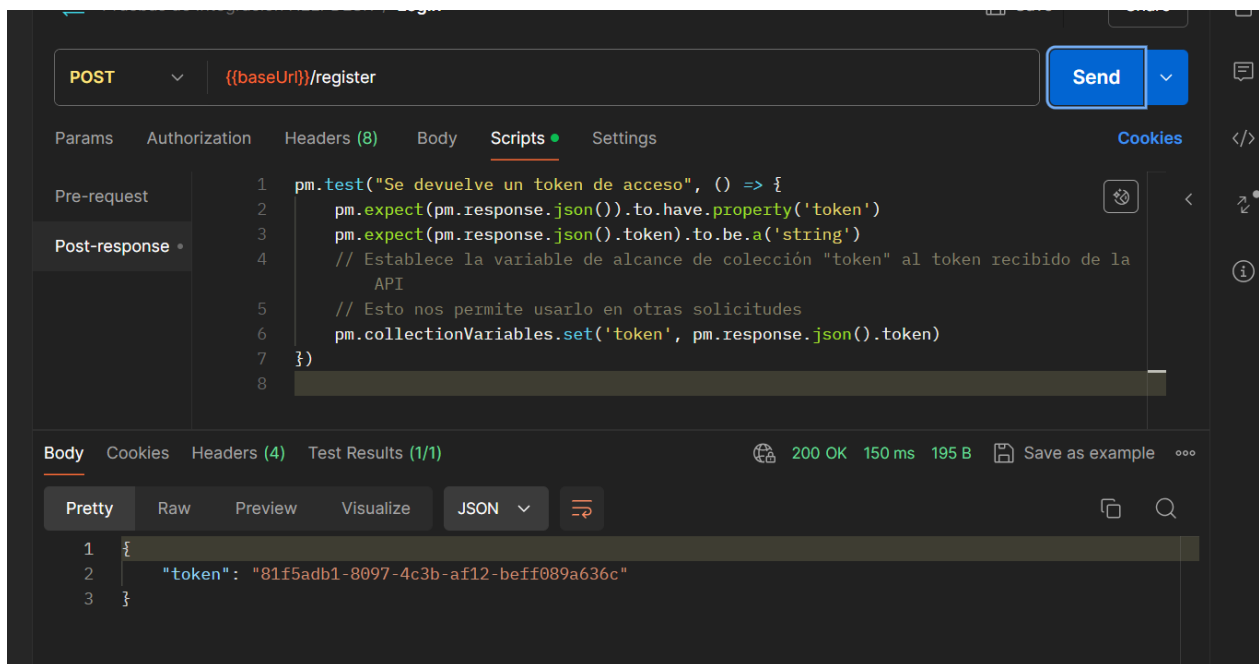
Nota. La figura ilustra el resultado de la ejecución de pruebas unitarias.

6.5.2.2 Pruebas de Integración

Después de las pruebas unitarias, se ejecutan pruebas de integración para asegurar que los módulos del sistema interactúen correctamente. Se evalúa cómo los datos fluyen entre los componentes y se corrigen posibles fallos en la comunicación entre ellos.

Figura 44

Captura de Pantalla Pruebas de Integración



Nota. La figura ilustra una captura de pantalla de postman donde se ejecuta la prueba de integración.

6.5.2.3 Pruebas de Seguridad

6.5.2.3.1 Prueba de Acceso No Autenticado (Sin Token)

La API del sistema requiere de un token de autenticación, por lo que cualquier consulta a endpoints autorizados deberá retornar el código de estado “404 – Forbidden”

Figura 45

Prueba de Acceso No Autenticado

```
1 [Fact]
2 public async Task ProbarApiSinToken_DeberiaRetornarProhibido()
3 {
4     var respuesta = await cliente.GetAsync(UrlApi);
5
6     respuesta.StatusCode.Should().Be(System.Net.HttpStatusCode.Forbidden);
7 }
```

6.5.2.3.2 Prueba de Inyección de SQL

El objetivo de esta prueba es validar que el sistema sea resistente a un ataque de inyección SQL en el endpoint de inicio de sesión.

Figura 46

Prueba de Inyección SQL

```
1 [Fact]
2 public async Task ProbarInyeccionSqlEnLogin_DeberiaRetornarNoAutorizado()
3 {
4     var cargaMaliciosa =
5         "{\"username\": \"admin' OR '1'='1\", \"password\": \"pruebaMaliciosa\"}";
6     var contenido =
7         new StringContent(cargaMaliciosa, Encoding.UTF8, "application/json");
8     var respuesta = await cliente.PostAsync(UrlApi, contenido);
9     respuesta.StatusCode.Should().Be(System.Net.HttpStatusCode.Unauthorized);
10 }
11
```

6.5.2.3.3 Prueba de Respuesta a Entrada Maliciosa (XSS)

El propósito de esta prueba es garantizar que la API sane adecuadamente las entradas para evitar ataques de XSS (Cross-Site Scripting).

Figura 47

Prueba de Ataque XSS.

```
1 [Fact]
2 public async Task ProbarRespuestaApi_DeberiaSerSanitizadaContraXSS()
3 {
4     var entradaMaliciosa = "<script>alert('Ataque XSS');</script>";
5     var contenido = new StringContent($"{{\"titulo\": \"{entradaMaliciosa}\"}}",
6     Encoding.UTF8, "application/json");
7     var respuesta = await cliente.PostAsync(UrlApi, contenido);
8     var contenidoRespuesta = await respuesta.Content.ReadAsStringAsync();
9     contenidoRespuesta.Should().NotContain("<script>");
10 }
```

6.5.2.3.4 Pruebas de Usabilidad

El sistema **Helpdesk** fue puesto a prueba a partir del **6 de junio de 2024**, con el objetivo de evaluar su usabilidad en un entorno real. Durante este período, los **usuarios finales** interactuaron con el sistema, llevando a cabo tareas representativas de su trabajo diario. Estas tareas incluyeron la creación de tickets, seguimiento de incidencias y la gestión de solicitudes, reflejando el uso cotidiano del sistema en un contexto práctico.

Los usuarios finales utilizaron el sistema de manera natural, integrándolo en sus rutinas diarias, lo que permitió obtener una visión auténtica de su eficacia y facilidad de uso. Tras esta fase de pruebas, se realizaron **encuestas de satisfacción** para obtener retroalimentación directa de los usuarios sobre su experiencia con el sistema.

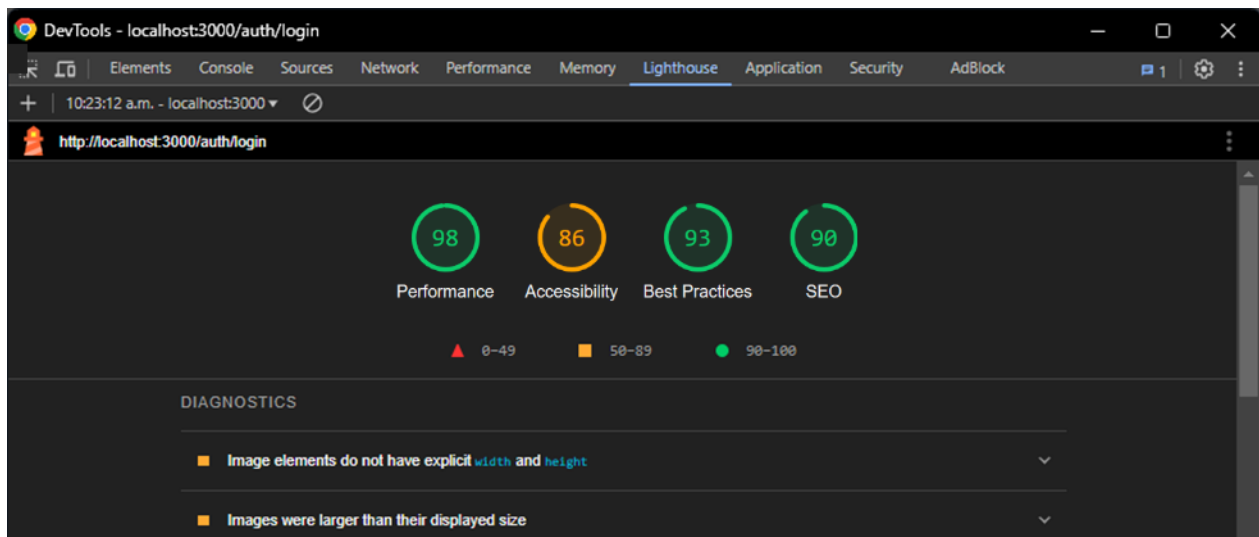
Los resultados de las encuestas y los hallazgos derivados de las pruebas de usabilidad han sido analizados en detalle. Las encuestas realizadas durante las pruebas pueden ser encontrados en el **apartado de anexos**.

6.5.2.3.5 Pruebas de Rendimiento

Las pruebas de rendimiento tienen como objetivo evaluar la capacidad de respuesta. La herramienta **Lighthouse** permite detectar posibles cuellos de botella y optimizar el rendimiento, asegurando una experiencia fluida y rápida para los usuarios finales.

Figura 48

Prueba de rendimiento de Lighthouse



Nota. La prueba fue realizada en ambiente de desarrollo.

6.6 FASE DE DESPLIEGUE

La **fase de despliegue** es crucial, ya que consiste en trasladar el sistema desde el entorno de pruebas al entorno de producción, asegurando que el sistema esté completamente funcional para los usuarios finales. A continuación, se detallan los pasos realizados para el despliegue del **Sistema Helpdesk** para la **División de Tecnología de la Información (DTI)** de la **Universidad Nacional de Ingeniería (UNI)**.

6.6.1 Preparación del Entorno de Producción

Cada uno de los componentes del sistema fue desplegado en su propia máquina virtual:

- **Frontend:** Desplegado en **IIS** en una máquina virtual con **Windows Server**.
- **Backend:** Desplegado en **Nginx** sobre una máquina virtual con **Linux**.
- **Base de Datos:** Alojada en **SQL Server** en una máquina virtual separada con **Windows Server**.

Esta configuración garantiza que cada componente del sistema esté aislado y optimizado según sus necesidades específicas.

6.6.2 Configuración del Servidor Frontend

El **frontend**, desarrollado en **React**, fue desplegado en **IIS** (Internet Information Services) en su propia máquina virtual con **Windows Server**. Los pasos principales fueron:

- **Instalación de IIS:** Configuración de IIS para servir archivos estáticos de la aplicación React.
- **Despliegue del Frontend:** Compilación del código del frontend utilizando `npm run build` y despliegue de los archivos estáticos en IIS.
- **Configuración de URL Rewrite:** Configuración de reglas en IIS para gestionar correctamente las rutas en la aplicación React de una sola página (SPA).

6.6.3 Configuración del Servidor Backend

El **backend**, desarrollado en **ASP.NET Core**, fue desplegado en un contenedor **Docker** sobre una máquina virtual con **Linux**. Los pasos principales fueron:

1. **Configuración de Docker:**
 - a. Creación de un contenedor Docker para ejecutar el backend, permitiendo un entorno aislado y escalable.
 - b. Configuración de Nginx dentro del contenedor para actuar como **proxy inverso**, dirigiendo las peticiones del frontend hacia el backend.
2. **Despliegue del Backend:**
 - a. Uso de `dotnet publish` para empaquetar la aplicación ASP.NET Core y luego transferirla al contenedor Docker.
 - b. Configuración del contenedor para que se inicie automáticamente con la máquina virtual.
3. **Configuración de la Conexión con la Base de Datos:**
 - a. Configuración del backend para conectarse a la base de datos SQL Server, asegurando que las consultas y transacciones se realicen de manera segura.

6.6.4 Ajustes Finales y Optimización

Una vez configurados los servidores, se procedió a integrar ambas partes del sistema:

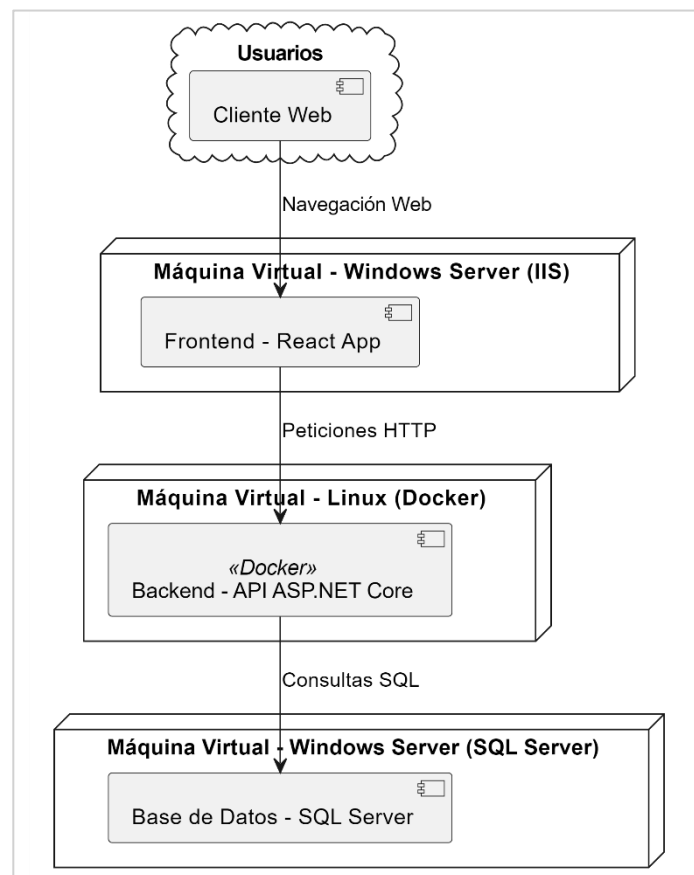
1. **Configuración de CORS (Cross-Origin Resource Sharing):** Se habilitaron las políticas de CORS en el backend, permitiendo que el frontend, alojado en IIS, pudiera interactuar con la API en Nginx.
2. **Pruebas de Conectividad:** Se realizaron pruebas de conectividad para asegurar que el frontend pudiera enviar solicitudes correctamente al backend, y que las respuestas fueran manejadas sin problemas.

6.6.5 Diagrama de Despliegue

El siguiente diagrama ilustra la arquitectura de despliegue del sistema Helpdesk, mostrando la interacción entre las máquinas virtuales y los contenedores:

Figura 49

Diagrama de Despliegue



Nota. La figura muestra una arquitectura distribuida con un cliente web, un frontend en React (Windows Server IIS), un backend en ASP.NET Core (Docker/Linux) y una base de datos SQL Server (Windows Server).

6.6.6 Integración y Entrega continua

El CI/CD (Integración y Entrega Continua) automatiza la integración de código y su despliegue. La **Integración Continua (CI)** verifica frecuentemente el código con pruebas automáticas. La **Entrega Continua (CD)** despliega automáticamente las versiones válidas a producción. Las fases clave del proceso son:

- **Desarrollo:** Los desarrolladores escriben y suben el código al repositorio.
- **Construcción:** El código se compila y genera artefactos ejecutables.
- **Pruebas:** Se ejecutan pruebas automáticas (unitarias, integración, etc.) para validar el código.
- **Despliegue:** El código validado se despliega a los entornos de prueba o producción.

Figura 50

Captura de Pantalla del árbol de Pipelines.

	#900		 release  c76eb41b  FIX: Gitlab CI/CD Entornos ...	  	 00:05:20  6/6/2024
	#899		 test  c76eb41b  FIX: Gitlab CI/CD Entornos ...	  	 00:04:14  6/6/2024
	#898		 dev  97652455  Fix: Dev URLS	  	 00:04:44  5/5/2024

Nota. Muestra cómo se estructuran y se ejecutan los pipelines en el proceso de CI/CD/

6.6.7 Capacitación del Personal

Para asegurar una transición fluida al nuevo sistema, se realizaron sesiones de capacitación con el personal de la DTI, enseñando el uso de las principales funcionalidades del sistema Helpdesk. También se proporcionó documentación detallada para facilitar la administración y resolución de problemas futuros.

VII. CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

Durante el desarrollo del Sistema Web Helpdesk para la División de Tecnología de la Información de la Universidad Nacional de Ingeniería (UNI) cumplió satisfactoriamente tanto con el objetivo general como con los objetivos específicos planteados.

1. **Análisis de requerimientos:** Se realizó un análisis detallado que permitió definir las funcionalidades necesarias para gestionar eficientemente las incidencias dentro de la DTI.
2. **Diseño de la interfaz:** Se creó una interfaz intuitiva y fácil de usar, que facilita la gestión de incidencias y mejora la experiencia del usuario en el registro y seguimiento de solicitudes.
3. **Desarrollo del módulo de tickets:** Se codificó un módulo robusto para la gestión de solicitudes de soporte, utilizando tecnologías como JavaScript, C# y SQL Server, lo que permitió una mejor administración de las incidencias.
4. **Control de acceso:** Se implementó un sistema de control de acceso según roles, garantizando la seguridad y privacidad de la información.
5. **Implantación:** El sistema fue implantado con éxito en la DTI, donde ya está operativo, contribuyendo a una mayor eficiencia en la atención de solicitudes.

A través del sistema, se logra una administración más eficiente de las solicitudes de soporte, permitiendo su registro, seguimiento y resolución de forma más ágil y organizada. En la actualidad, el sistema ya está siendo utilizado a nivel de la División de Tecnología de la Información (DTI), donde los usuarios lo emplean para gestionar incidencias, contribuyendo a una mayor productividad y mejora en la atención de las solicitudes.

Si bien el sistema está completamente funcional y preparado para ser implementado en toda la Universidad Nacional de Ingeniería (UNI), su implementación a nivel de institución aún no se ha llevado a cabo debido a decisiones administrativas ajenas a los desarrolladores del proyecto. Esto significa que, a nivel técnico y de diseño, el sistema está listo para su expansión, pero su implementación a nivel institucional está sujeta a procesos y aprobaciones administrativos que no dependen del equipo desarrollador.

7.2 RECOMENDACIONES

Para maximizar el potencial del Sistema Web Helpdesk y asegurar su evolución dentro de la UNI, se sugieren las siguientes recomendaciones:

- **Capacitación continua:** Es fundamental seguir capacitando a los usuarios actuales de la DTI, y en el futuro, a los usuarios de otras áreas de la UNI, para asegurar el uso correcto y eficiente del sistema.
- **Mejoras continuas:** Aunque el sistema actualmente cubre la gestión de tickets, sería recomendable expandir su funcionalidad con la adición de nuevos módulos. Por ejemplo, un **módulo de gestión de proyectos** podría atender las necesidades más especializadas del equipo de desarrollo de software dentro de la DTI. Además, se podría desarrollar un **módulo para el área de cobranzas**, que permitiría gestionar, asignar y realizar el seguimiento de tareas relacionadas con la recaudación, optimizando el control de los flujos de trabajo en esa área.
- **Seguridad:** Mantener los certificados HTTPS actualizados para garantizar la protección de los datos y la seguridad en la transferencia de información.

Respaldo de información: Se recomienda implementar rutinas regulares de respaldo de la base de datos para asegurar la continuidad del servicio en caso de fallos o eventualidades, protegiendo así los datos críticos del sistema.

VII. BIBLIOGRAFÍA.

Albahari, J. (2021). C# 9 in a Nutshell The Definitive Reference. «O'Reilly Media, Inc.»

Asamblea Nacional de Nicaragua. (2011). Ley No. 787: Ley de Protección de Datos Personales. Publicado en La Gaceta, Diario Oficial, No. 124.

Asamblea Nacional de Nicaragua. (2009). Ley No. 698: Ley de Acceso a la Información Pública. Publicado en La Gaceta, Diario Oficial, No. 118.

Burns, B., Beda, J., & Hightower, K. (2019). Kubernetes: Up and Running: Dive Into the Future of Infrastructure. O'Reilly Media.

Crockford, D. (2008). JavaScript: the good parts. <http://ci.nii.ac.jp/ncid/BA86771438>

Elmasri, R., & Navathe, S. (2007). Fundamentos de sistemas de bases de datos. ADDISON WESLEY.

Entendiendo el entorno digital - Aspectos Técnicos. (s. f.). https://micrositios.inai.org.mx/marcocompetencias/?page_id=372

Fast Reports. (s. f.). <https://www.fast-report.com/products/fast-report-net/>

Gomez, I. (2022, 8 marzo). Conoce qué es GitLab y gestiona efectivamente los cambios que realices en tus proyectos. <https://www.crehana.com>. <https://www.crehana.com/blog/transformacion-digital/que-es-gitlab/>

Kane, S., & Matthias, K. (2023). Docker: Up and Running: Shipping Reliable Containers in Production. O'Reilly Media.

KYOCERA Document Solutions. (2024, 30 mayo). Los 6 principales tipos de sistemas de información | Kyocera. KYOCERA Document Solutions. <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/the-cloud/los-6-principales-tipos-sistemas-informacion.html>

Laudon, K. C., & Laudon, J. P. (2004). Sistemas de información gerencial: administración de la empresa digital. Pearson Education.

Lemonaki, D. (2022, 18 marzo). Frontend VS Backend – What's the Difference? freeCodeCamp.org. <https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/>

Loeliger, J., & McCullough, M. (2012). Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development. «O'Reilly Media, Inc.»

Microsoft. (2023, 23 mayo). SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS). Microsoft Learn. <https://learn.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>

Otto, M., Thornton, J., & Bootstrap. (s. f.). Bootstrap. <https://getbootstrap.com/>

Overview - Redmine. (s. f.). <https://www.redmine.org/projects/redmine/wiki>

Petkovic, D. (2016). Microsoft SQL Server 2016: A Beginner's Guide, Sixth Edition. McGraw Hill Professional.

Price, M. J. (2020). C# 9 and .NET 5 - Modern Cross-Platform Development - Fifth Edition: Build Intelligent Apps, Websites, and Services with Blazor, ASP.NET Core, and Entity Framework Core Using Visual Studio Code.

React. (s. f.). <https://react.dev/>

Rengifo, J. P. (2023, 18 abril). OTRS: Sistema abierto para la gestión de solicitudes. ITSoftware. <https://itsoftware.com.co/content/otrs-sistema-abierto-para-la-gestion-de-solicitudes/>

Robbins, J. N. (2012). Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. «O'Reilly Media, Inc.»

Schlicht, W. (2019). *Help Desk Management: How to run a computer user support Service Desk effectively*. Build a Help Desk Consulting LLC.

Stair, R. M., & Reynolds, G. W. (2010). *Principios de sistemas de información*(s.f.). Obtenido de React: <https://react.dev/>

Albahari, J. (2021). *C# 9 in a Nutshell The Definitive Reference*. Sebastopol: O'Reilly Media, Inc.

Beda, J., Hightower, K., & Burns, B. (2017). *Kubernetes: Up and Running*. O'Reilly Media, Inc.

Bootstrap, Otto, M., & Thornton, J. (s.f.). Obtenido de Bootstrap: <https://getbootstrap.com/>

C. Laudon, K., & P. Laudon, J. (2012). *Sistemas de Información Gerencial: Administración de la Empresa Digital*. México: PEARSON EDUCACIÓN. Obtenido de <https://juanantonioleonlopez.files.wordpress.com/2017/08/sistemas-de-informacic3b3n-gerencial-12va-edicic3b3n-kenneth-c-laudon.pdf>

Crockford, D. (s.f.). *JavaScript: The Good Parts*. O'Reilly.

Elmasri, E. A., & Navathe, S. B. (s.f.). *Fundamentos de Sistemas*.

Entendiendo el entorno digital - Aspectos técnicos. (s.f.). Obtenido de Micrositios: https://micrositios.inai.org.mx/marcocompetencias/?page_id=372

Fast Reports. (s.f.). Obtenido de Fast Reports: <https://www.fast-report.com/en/product/fast-report-net/>

Gomez, I. (8 de Marzo de 2022). *Conoce qué es GitLab y gestiona efectivamente los cambios que realices en tus proyectos*. Obtenido de crehana: <https://www.crehana.com/blog/transformacion-digital/que-es-gitlab/>

Jon Loeliger, M. M. (2012). *Version Control with Git, 2nd Edition*. O'Reilly Media, Inc.

KYOCERA. (30 de Mayo de 2024). *Los 6 principales tipos de sistemas de información*.
Obtenido de KYOCERA Document Solutions:
<https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/the-cloud/los-6-principales-tipos-sistemas-informacion.html>

Lemonaki, D. (18 de Marzo de 2022). *Frontend VS Backend – What's the Difference?*
Obtenido de freecodecamp: <https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/>

M. Stair, R., & Reynolds, G. W. (2010). *Principios de Sistemas de Información: Un Enfoque Administrativo*. Boston, Massachusetts, Estados Unidos: CENGAGE Learning.
Obtenido de <https://books.google.com.ni/books?id=8vVYewAACAAJ>

Matthias, K. (2015). *Docker Up and Running*.

Microsoft. (23 de Mayo de 2023). *¿Qué es SQL Server Management Studio (SSMS)?*
Obtenido de Microsoft Learn: <https://learn.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>

Overview - Redmine. (s.f.). Recuperado el 10 de 04 de 2024, de
<https://www.redmine.org/projects/redmine/wiki>

Petkovic, D. (2016). *Microsoft SQL Server 2016: A Beginner's Guide, Sixth Edition 6th Edition*. McGraw-Hill Education.

Price, M. J. (2020). *C# 9 and .NET 5 – Modern Cross-Platform Development - Fifth Edition*. Packt.

Rengifo, J. P. (18 de Abril de 2023). *ITSoftware*. Obtenido de
<https://itsoftware.com.co/content/otrs-sistema-abierto-para-la-gestion-de-solicitudes/>

Robbins, J. N. (2012). *Learning Web Desing*. Sebastopol: Littlechair, Inc. Obtenido de
<https://wtf.tw/ref/robbins.pdf>

Schlicht, W. (2019). *Help Desk Management: How to run a computer user support Service Desk effectively*. Build a Help Desk Consulting LLC.

IX. ANEXOS

9.1 DICCIONARIO DE DATOS

Tabla 11

Tabla: Usuarios

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDUsuario	INT	Identificador único del usuario.	PK, NOT NULL	N/A	1
Nombre	VARCHAR(50)	Nombre del usuario.	NOT NULL	Texto	"Juan"
Apellido	VARCHAR(50)	Apellido del usuario.	NOT NULL	Texto	"Pérez"
Correo	VARCHAR(100)	Correo electrónico del usuario.	UNIQUE, NOT NULL	Formato de correo	"juan@example.com"
Contraseña	VARCHAR(256)	Contraseña encriptada del usuario.	NOT NULL	N/A	(Hash encriptado)
IDRol	INT	Identificador del rol asignado al usuario.	FK, NOT NULL	N/A	2
FechaCreacion	DATETIME	Fecha y hora en que se creó el registro.	NOT NULL	N/A	28/8/2024 10:00
FechaModificacion	DATETIME	Fecha y hora de la última modificación del registro.	NULLABLE	N/A	28/8/2024 12:00
Anulado	BIT	Indica si el registro ha sido anulado (1 = Anulado, 0 = Activo).	NOT NULL	0, 1	0

Tabla 12

Tabla: Roles

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDRol	INT	Identificador único del rol.	PK, NOT NULL	N/A	2
Nombre	VARCHAR(50)	Nombre del rol.	NOT NULL	Texto	"Administrador"
Descripcion	VARCHAR(255)	Descripción del rol.	NULLABLE	Texto	"Administra todo"
FechaCreacion	DATETIME	Fecha y hora en que se creó el registro.	NOT NULL	N/A	28/8/2024
FechaModificacion	DATETIME	Fecha y hora de la última modificación del registro.	NULLABLE	N/A	28/8/2024
Anulado	BIT	Indica si el registro ha sido anulado (1 = Anulado, 0 = Activo).	NOT NULL	0, 1	0

Tabla 13

Tabla: Pantallas

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDPantalla	INT	Identificador único de la pantalla.	PK, NOT NULL	N/A	1
Nombre	VARCHAR(50)	Nombre de la pantalla.	NOT NULL	Texto	"Gestión Usuarios"
Descripcion	VARCHAR(255)	Descripción de la funcionalidad de la pantalla.	NULLABLE	Texto	"Administra usuarios del sistema"
Activo	BIT	Indica si la pantalla está activa (1 = Activo, 0 = Inactivo).	NOT NULL	0, 1	1
FechaCreacion	DATETIME	Fecha y hora en que se creó el registro.	NOT NULL	N/A	28/8/2024
FechaModificacion	DATETIME	Fecha y hora de la última modificación del registro.	NULLABLE	N/A	28/8/2024

Tabla 14

Tabla: Tickets

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDTicket	INT	Identificador único del ticket.	PK, NOT NULL	N/A	1001
IDUsuario	INT	Identificador del usuario que generó el ticket.	FK, NOT NULL	N/A	1
IDCategoria	INT	Identificador de la categoría del ticket.	FK, NOT NULL	N/A	3
IDPrioridad	INT	Identificador de la prioridad del ticket.	FK, NOT NULL	N/A	2
IDEstado	INT	Identificador del estado actual del ticket.	FK, NOT NULL	N/A	1
Titulo	VARCHAR(255)	Título o resumen del ticket.	NOT NULL	Texto	"Error en login"
Descripcion	TEXT	Descripción detallada del problema o solicitud.	NOT NULL	Texto	"No puedo acceder a mi cuenta"
FechaCreacion	DATETIME	Fecha y hora en que se creó el ticket.	NOT NULL	N/A	28/8/2024 09:00
FechaModificacion	DATETIME	Fecha y hora de la última modificación del ticket.	NULLABLE	N/A	28/8/2024 10:00
Anulado	BIT	Indica si el ticket ha sido anulado (1 = Anulado, 0 = Activo).	NOT NULL	0, 1	0

Tabla 15

Tabla: TicketEstados

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDEstado	INT	Identificador único del estado.	PK, NOT NULL	N/A	1
Nombre	VARCHAR(50)	Nombre del estado (Ej: Abierto, Cerrado, En Proceso).	NOT NULL	Texto	"Abierto"
Descripcion	VARCHAR(255)	Descripción del estado.	NULLABLE	Texto	"Ticket está abierto y en proceso de atención"
FechaCreacion	DATETIME	Fecha y hora en que se creó el registro.	NOT NULL	N/A	28/8/2024
FechaModificacion	DATETIME	Fecha y hora de la última modificación del registro.	NULLABLE	N/A	28/8/2024
Anulado	BIT	Indica si el registro ha sido anulado (1 = Anulado, 0 = Activo).	NOT NULL	0, 1	0

Tabla 16

Tabla: TicketCategorias

Nombre de la Columna	Tipo de Dato	Descripción	Restricciones	Valores Permitidos	Ejemplo
IDCategoria	INT	Identificador único de la categoría.	PK, NOT NULL	N/A	3
Nombre	VARCHAR(50)	Nombre de la categoría.	NOT NULL	Texto	"Sistema"
Descripcion	VARCHAR(255)	Descripción de la categoría.	NULLABLE	Texto	"Problemas relacionados con sistemas"
FechaCreacion	DATETIME	Fecha y hora en que se creó el registro.	NOT NULL	N/A	28/8/2024
FechaModificacion	DATETIME	Fecha y hora de la última modificación del registro.	NULLABLE	N/A	28/8/2024
Anulado	BIT	Indica si el registro ha sido anulado (1 = Anulado, 0 = Activo).	NOT NULL	0, 1	0

9.2 HOJAS ASISTENCIA DE CAPACITACIÓN

Figura 51

Asistencia de Capacitación a Usuarios #1



División de
Tecnología de
la Información

Lista de Asistencia

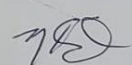
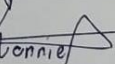
Nombre del encuentro: Capacitación Para Uso Helpdesk

Objetivo: _____

Lugar: Sala Reunion DTI / Reunion Virtual

Fecha: 21/06/24

Hora inicio/final: 11:00 am. 12:00 p.m.

No	Nombre y apellido	Nombre del área a que pertenece	Cargo del participante	Cédula	Firma
1.	Nubia Kema Altamirano	DTI	Respons	6162306831003E	
2.	Carlos Rodriguez	DTI	Prop.	201212600131	
3.	Jean Valencia	DTI	Técnico Informático	001-200589- 0069E	
4.	Ronnie Acevedo	DTI	Analista Sistema	001-180498- 1009W	
5.	Martin Mendoza	DTI	Analista Sistema	001-090893- 0009T	

📞 Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

📍 Recinto Universitario Simón Bolívar
Avenida Universitaria,
Managua, Nicaragua,
Apdo: 5595

✉️ manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

Figura 52

Asistencia de Capacitación a Usuarios #2



División de
Tecnología de
la Información

Lista de Asistencia

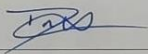
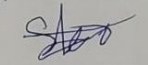
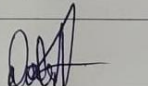
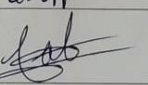
Nombre del encuentro: _____

Objetivo: _____

Lugar: _____

Fecha: _____

Hora inicio/final: _____

No	Nombre y apellido	Nombre del área a que pertenece	Cargo del participante	Cédula	Firma
1.	Danny Caliz	DTI	Analista Sistema.	001 260394 10047	
2.	Steven Mendez	DTI	Analista Sist	001 220302 1024B	
3.	Rollman Mendoza	DTI	Espec. Sist	281-080114- 002969	Rollman MD
4.	Darling Lopez	DTI	Técnico Senior	001-050392- 0004F	
5.	José Gabriel García	DTI	Analista de Sistemas	041-191000- 1002)	

📞 Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

📍 Recinto Universitario Simón Bolívar
Avenida Universitaria,
Managua, Nicaragua.
Apdo: 5595

✉️ manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

Figura 53

Asistencia de Capacitación a Usuarios #2

 División de Tecnología de la Información	
INFORME DE REUNIÓN	
Acuerdo:	Se les brinda Capacitación Sobre uso de Helpdesk Para que los equipos de desarrollo, servidores, soporte técnico e infraestructura puedan registrar sus actividades diarias
Desarrollo:	Donald Barrantes Steven Mendez Jose Gabriel Garcia Ronnie Acevedo
Elaborado por:	
Copia: Archivo	

Teléfonos: (505) 2278 3141
(505) 2278 4984 Ext: 102 -103

Recinto Universitario Simón Bolívar
Avenida Universitaria,
Managua, Nicaragua.
Apdo: 5595

manuel.godinez@dti.uni.edu.ni
www.uni.edu.ni

Figura 55

Encuesta de Satisfacción del Sistema Web Helpdesk #1

Encuesta de Satisfacción sobre el Sistema Web Helpdesk

Esta encuesta tiene como propósito medir el nivel de satisfacción y recopilar retroalimentación sobre el uso del Sistema Web Helpdesk.

Nombre: Jesly Joseth Alvarez Cargo: Analista de Sistema

1. ¿Considera que la interfaz del sistema es intuitiva y fácil de usar?

☒ De acuerdo ☐ Neutral ☐ En desacuerdo

2. ¿Tuvo alguna dificultad para aprender a usar el sistema?

☐ Sí ☐ Parcialmente ☒ No

Si su respuesta es "Sí" o "Parcialmente", por favor explique:

3. ¿El sistema cumple con las funciones necesarias para gestionar sus tickets?

☒ Sí ☐ Parcialmente ☐ No

Si respondió "Parcialmente" o "No", por favor indique qué funciones faltan o necesitan mejorar:

4. ¿Ha mejorado su productividad desde que comenzó a utilizar este sistema?

☒ Sí ☐ Parcialmente ☐ No

Argumente su respuesta: He mejorado considerablemente ya que ahora logro medir y revisar que tan activo estoy en mi area, de igual manera me sirvió para documentar mis avances en las asignaciones encomendadas.

5. ¿Qué tan satisfecho está con el sistema Helpdesk en general?

☒ Muy satisfecho ☐ Satisfecho ☐ Neutral ☐ Insatisfecho ☐ Muy insatisfecho

Agradecemos mucho su tiempo y disposición para completar esta encuesta. Sus respuestas serán consideradas para optimizar y mejorar continuamente el Sistema Web Helpdesk.

Figura 56

Encuesta de Satisfacción del Sistema Web Helpdesk #2

Encuesta de Satisfacción sobre el Sistema Web Helpdesk

Esta encuesta tiene como propósito medir el nivel de satisfacción y recopilar retroalimentación sobre el uso del Sistema Web Helpdesk.

Nombre: Maurin Gomez Torres Cargo: Analista de Sistemas

1. ¿Considera que la interfaz del sistema es intuitiva y fácil de usar?

☒ De acuerdo ☐ Neutral ☐ En desacuerdo

2. ¿Tuvo alguna dificultad para aprender a usar el sistema?

☐ Sí ☐ Parcialmente ☒ No

Si su respuesta es "Sí" o "Parcialmente", por favor explique:

3. ¿El sistema cumple con las funciones necesarias para gestionar sus tickets?

☒ Sí ☐ Parcialmente ☐ No

Si respondió "Parcialmente" o "No", por favor indique qué funciones faltan o necesitan mejorar:

4. ¿Ha mejorado su productividad desde que comenzó a utilizar este sistema?

☒ Sí ☐ Parcialmente ☐ No

Argumente su respuesta:

Todas las tareas que se me han asignado tengo que justificarlas con el sistema helpdesk y así se ve mi trabajo y cumplimiento de tareas.

5. ¿Qué tan satisfecho está con el sistema Helpdesk en general?

☒ Muy satisfecho ☐ Satisfecho ☐ Neutral ☐ Insatisfecho ☐ Muy insatisfecho

Agradecemos mucho su tiempo y disposición para completar esta encuesta. Sus respuestas serán consideradas para optimizar y mejorar continuamente el Sistema Web Helpdesk.

9.3 ESTUDIO DE FACTIBILIDAD

9.3.1 Factibilidad Técnica

Análisis de Requisitos Técnicos:

- **Hardware y Software:** La implementación del sistema requiere servidores robustos para manejar el alto volumen de tickets y usuarios esperados. Se utilizarán tecnologías como HTML, CSS, JavaScript (con React para el frontend), C# y ASP.NET para el backend, junto con SQL Server para la gestión de bases de datos.
- **Infraestructura:** El sistema se desplegará utilizando Docker para ejecutar los contenedores y Kubernetes para la orquestación, asegurando escalabilidad y flexibilidad.
- **Control de Versiones:** Git y GitLab serán utilizados para el control de versiones y la gestión de cambios, facilitando la colaboración entre los desarrolladores.
- **Seguridad:** Implementación de medidas de seguridad robustas para proteger la información confidencial de los usuarios y prevenir accesos no autorizados.

Evaluación de Competencia Técnica:

- El equipo de desarrollo está compuesto por profesionales con experiencia en las tecnologías seleccionadas y en la implementación de sistemas similares, lo que asegura la capacidad técnica para llevar a cabo el proyecto.

9.3.2 Factibilidad Económica

Costos Estimados:

- **Desarrollo:** Costos asociados a la contratación de desarrolladores, adquisición de hardware y software necesarios, y gastos operativos.
- **Mantenimiento:** Costos recurrentes de mantenimiento del sistema, actualizaciones, y soporte técnico.
- **Capacitación:** Inversión en la capacitación de los usuarios finales y del personal de soporte para garantizar una adopción efectiva del sistema.

Tabla 17

Servicios de Mantenimiento o Infraestructura

Concepto	Descripción	Rango de Costo Estimado (USD)	Rango de Costo Estimado (C\$)	Observaciones
Hardware	Servidores, equipos de red, almacenamiento	\$2,000/año	C\$74,000/año	Basado en una capacidad de 16GB de RAM, 4 vCPUs, y 100GB de SSD
Software	Licencias, herramientas de desarrollo	\$500/año	C\$18,500/año	Incluye herramientas de desarrollo, sistemas operativos, etc.
Imprevistos	Contingencias no anticipadas	\$500/año	C\$18,500/año	

Nota. La tabla está basada en estimaciones de servicios de renta en la nube como Azure, Digital Ocean, AWS y licencias de productos para instituciones.

Tabla 18

Costos de Desarrollo

Concepto	Descripción	Rango de Costo Estimado (USD)	Rango de Costo Estimado (C\$)
Desarrollador Backend	Salario de un desarrollador Backend	\$2.21/hora	C\$82.02/hora
Desarrollador Frontend	Salario de un desarrollador Frontend	\$2.21/hora	C\$82.02/hora
Total (480 horas)	Costo total para 480 horas	\$2,128	C\$78,739.2

Nota. Los costos están basados en el salario promedio nacional devengado por actividad económica del banco central de nicaragua.

Tabla 19

Costo Total

Categoría	Costo Estimado (USD)	Costo Estimado (C\$)
Totales de Mantenimiento o Infraestructura (Anual)	\$3,000/año	C\$111,000/año
Totales de Desarrollo (Pago Único)	\$2,128 (único)	C\$78,739.2 (único)

Nota. Costo total de las estimaciones de precios anteriores.

Las estimaciones presentadas se basan en el supuesto de que la universidad no cuenta con ningún recurso previo. La implementación del Sistema Web Helpdesk no genera costos adicionales para la institución, ya que se asume que dispone del hardware y software necesarios para su puesta en marcha. Además, el trabajo se realiza como parte de un proyecto monográfico, por lo que no se incurre en costos de mano de obra. Este esfuerzo representa una retribución a la sociedad, en agradecimiento por el apoyo económico del 6%.

9.2.1 Análisis de Costos-Beneficios:

- **Beneficios:** Mejora en la eficiencia operativa de la DTI, reducción del tiempo de resolución de incidencias, y aumento en la satisfacción del usuario final.
- **Retorno de Inversión (ROI):** Se espera que los beneficios a largo plazo, en términos de ahorro de tiempo y mejora en la productividad, superen significativamente los costos iniciales y recurrentes.

9.3.3 Factibilidad Operativa

9.3.3.1 Adaptación a las Necesidades del Usuario:

- **Interfaz de Usuario:** El diseño de una interfaz intuitiva y fácil de usar que cumpla con los requerimientos específicos de la DTI.
- **Funcionalidades Clave:** Implementación de funcionalidades esenciales como la creación y seguimiento de tickets, generación de reportes de KPIs, y gestión de usuarios y permisos.
- **Soporte y Mantenimiento:** Establecimiento de un plan de soporte técnico continuo para resolver problemas y realizar mejoras basadas en el feedback de los usuarios.

9.3.4 Evaluación de Riesgos:

- **Riesgos Técnicos:** Posibles problemas técnicos durante el desarrollo y la implementación, mitigados mediante pruebas exhaustivas y un plan de contingencia.
- **Riesgos Operativos:** Resistencia al cambio por parte de los usuarios, mitigada mediante un programa de capacitación integral y soporte continuo.
- **Riesgos Financieros:** Desviaciones presupuestarias, mitigadas mediante una gestión financiera rigurosa y la evaluación periódica de costos.

9.3.5 Estudio de Factibilidad Legal

9.3.5.1 Cumplimiento de Normativas y Regulaciones

Nuestro sistema cumple con todas las normativas legales nacionales e institucionales aplicables en Nicaragua, incluyendo leyes de protección de datos, acceso a la información pública y regulación tecnológica. Entre estas, destacan:

- **Ley 787:** Ley de Protección de Datos Personales, garantizando la seguridad, privacidad y correcto manejo de la información de los usuarios (Ley de Protección de Datos Personales, 2011).
- **Ley 698:** Ley de Acceso a la Información Pública, asegurando la transparencia en las operaciones del sistema cuando corresponda (Ley de Acceso a la Información Pública, 2009).
- **Regulaciones Institucionales:** Cumplimiento de políticas internas de la Universidad Nacional de Ingeniería respecto a la infraestructura tecnológica y la seguridad de datos.

9.3.5.2 Propiedad Intelectual

El sistema es un desarrollo interno realizado por estudiantes de la Universidad Nacional de Ingeniería como parte de un proyecto monográfico. Esto asegura que:

- No se vulneran derechos de terceros ni existen conflictos de propiedad intelectual.
- Todo el software empleado utiliza licencias de código abierto (como React y Docker) o ya está licenciado por la universidad, lo que elimina posibles infracciones de derechos.

9.3.5.3 Validez Jurídica de Operaciones Electrónicas

El sistema está diseñado para garantizar la validez de las operaciones electrónicas, incluyendo la autenticación de usuarios mediante credenciales seguras y mecanismos compatibles con estándares internacionales, lo que asegura la trazabilidad y responsabilidad de las acciones realizadas en el sistema.

9.3.5.4 Contratos y Licencias

Dado que el sistema no incurre en costos adicionales para la universidad, no se requieren nuevos contratos o adquisiciones de licencias, respetando las limitaciones presupuestarias de la institución y optimizando los recursos existentes.

El Sistema Web Helpdesk es legalmente factible, dado que cumple con todas las regulaciones aplicables y respeta la propiedad intelectual de los componentes utilizados. Su implementación refuerza la seguridad, eficiencia y transparencia institucional, alineándose con los valores y objetivos de la Universidad Nacional de Ingeniería.

9.4 CRONOGRAMA

Tabla 20

Diagrama de Grant Fase 1 – 3

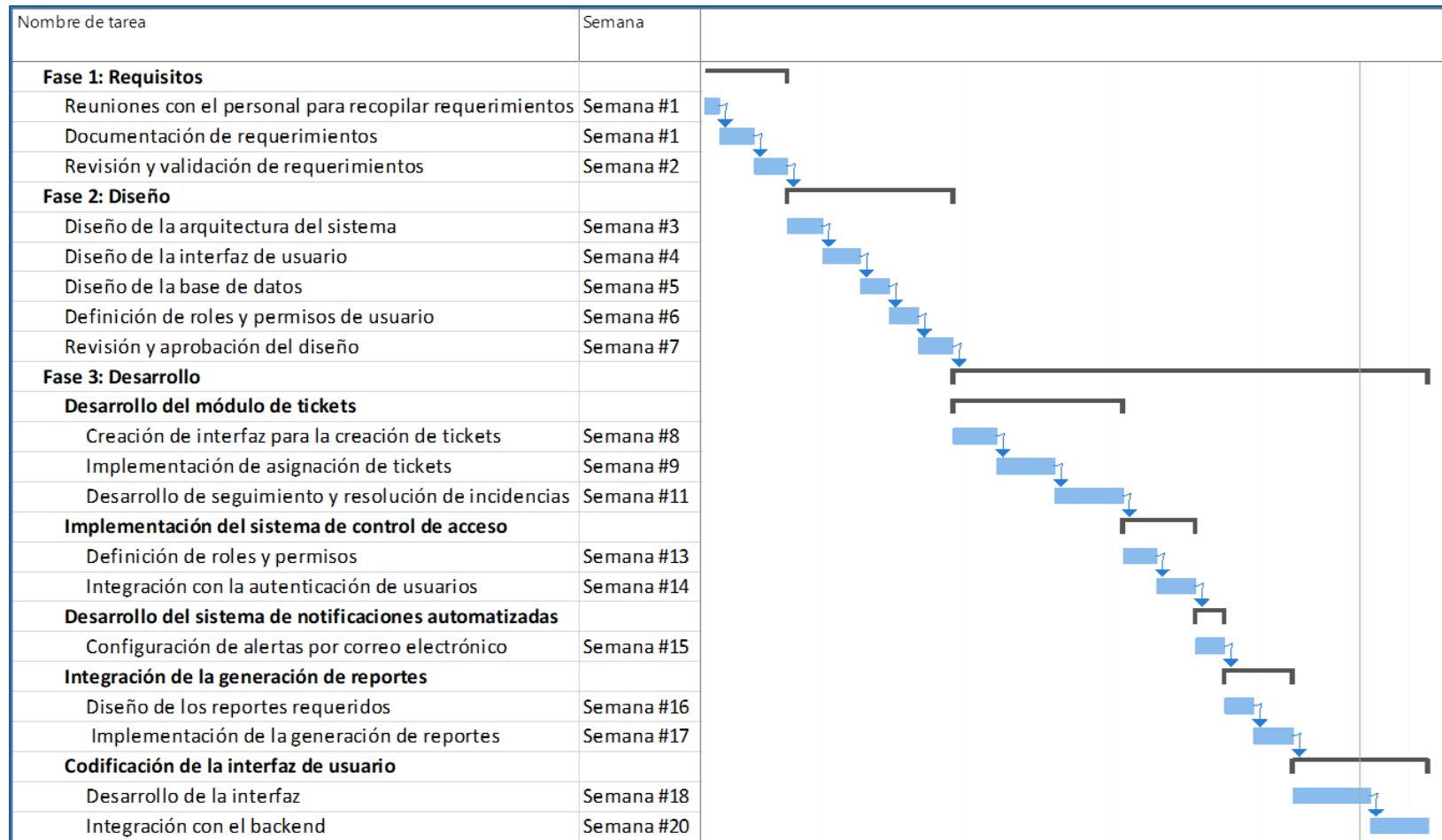


Tabla 21

Diagrama de Grant Fase 4 – 5

