

Área de Conocimiento de Tecnología de la  
Información y Comunicación

# **Sistema web para el control de solicitudes de servicios de telecomunicaciones en la empresa Claro Nicaragua**

Trabajo Monográfico para optar al título de  
Ingeniero en Computación

**Elaborado por:**

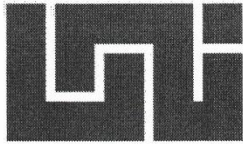
Br. Jesús Enmanuel  
Hernández González  
Carnet: 2017-0505U

Br. Danilo José  
Acevedo Flores  
Carnet: 2017-0016U

**Tutor:**

TeknL. Ing. Anayanci  
López Poveda





Managua, 05 de febrero de 2026.

**MSc. Claudia Lucia Benavidez Rugama**

Directora DACTIC

Sus manos.

Estimada MSc. Benavidez,

**Asunto: Carta Aval de Tutor (Defensa Monográfica)**

A través de la presente hago de su conocimiento que los estudiantes **Jesús Emmanuel Hernández González (2017- 0505U)** y **Danilo José Acevedo Flores (2017-0016U)** han finalizado su proyecto monográfico para optar al título de **Ingeniería en Computación**, el cual se titula **"Sistema web para el control de solicitudes de servicios de telecomunicaciones en la empresa Claro-Nicaragua"**.

He dado seguimiento a este proyecto monográfico y he revisado el informe final del mismo. El informe final cumple con todos los requisitos indicados en la normativa de Culminación de Estudio de la UNI y los lineamientos establecidos por la DACTIC; así también incorpora las indicaciones brindadas por el jurado en el proceso de predefensa. Por tanto, los estudiantes mencionados **están listos para pasar al proceso de defensa final de esta monografía**.

Sin más que agregar, me despido agradeciendo su amable atención a esta comunicación.

Cordialmente,

**TeknL. Ing. Anayanci López P.**

Profesor Titular

Ingeniería en Computación

DACTIC

Universidad Nacional de Ingeniería

 Recinto Universitario Simón Bolívar  
Avenida Universitaria.  
Managua, Nicaragua.

 [anayanci.lopez@uni.edu.ni](mailto:anayanci.lopez@uni.edu.ni)



## Área de Conocimiento de Tecnología de la Información y Comunicación

Managua, 05 de noviembre 2024

### Bachilleres

**Br. Jesús Enmanuel Hernández González** 2017-0505U

**Br. Danilo José Acevedo Flores** 2017-0016U

**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: **“Sistema web para el control de solicitudes de servicios de telecomunicaciones en la empresa Claro Nicaragua”**, 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 **TeknL. Ing. Anayanci López Poveda**, 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: TeknL. Ing. Anayanci López Poveda – 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](http://www.uni.edu.ni)

## **Dedicatoria**

*Le dedico este trabajo monográfico, en primer lugar, a Dios, por la gran bendición de haberme permitido llegar a esta etapa, siendo mi guía y fortaleza en todo momento.*

*A mis padres, Francisco Hernández y María Enriqueta, por su esfuerzo incansable que me impulsó a alcanzar mis metas; y a mis hermanos, Ivania, Lester, Sarah y Josué, por estar siempre a mi lado con su apoyo incondicional y palabras de aliento que nunca faltaron.*

*A mi tutora, TeknL. Ing. Anayanci López Poveda, por su orientación y entrega durante este proceso, cuyo acompañamiento fue esencial para convertir este proyecto en una realidad.*

**Jesús Enmanuel Hernández González.**

*Con profundo agradecimiento, dedico este trabajo en primer lugar a Dios, por haberme permitido llegar hasta esta etapa de mi vida, siendo mi guía, fortaleza y fuente de sabiduría en cada momento de este camino académico y personal.*

*A mis amados padres, María Flores y Danilo Acevedo, por su esfuerzo incansable, apoyo incondicional y ejemplo de perseverancia. Ustedes han sido la base sobre la cual se han construido mis sueños y logros; todo lo alcanzado es reflejo de su dedicación y amor.*

*A mi tutora, TeknL. Ing. Anayanci López Poveda, cuya orientación, paciencia y compromiso fueron fundamentales para dar forma, estructura y solidez a esta monografía. Su acompañamiento ha sido clave en la culminación de este esfuerzo académico.*

**Danilo José Acevedo Flores**

## **Resumen del tema**

En Claro-Nicaragua, la gestión de solicitudes de servicios de telecomunicaciones se realizaba mediante un sistema HelpDesk interno con limitaciones significativas, como ausencia de notificaciones automáticas, seguimiento inadecuado de solicitudes y falta de funcionalidades esenciales como priorización, reapertura, redirección de solicitudes y generación de reportes personalizados.

Para resolver estas deficiencias, se desarrolló un sistema web de control de solicitudes, basado en buenas prácticas de ITIL 2011 y adaptado al modelo de negocio de Claro-Nicaragua. Este sistema centraliza la gestión de solicitudes, mejora funciones críticas como notificaciones, reportes e indicadores, y fortalece la seguridad de los datos para todas las áreas de la empresa, optimizando tiempos de atención y apoyando la toma de decisiones.

El sistema ofrece los siguientes beneficios: gestión centralizada de solicitudes a lo largo de su ciclo de vida; automatización de notificaciones por correo electrónico; categorización y priorización según tipo de servicio y urgencia; asignación y redireccionamiento entre áreas; generación de reportes personalizados exportables a Excel; dashboards con indicadores de solicitudes, flujos de trabajo y Acuerdos de Nivel de Servicio (SLA / Service Level Agreement); diseño dinámico de formularios sin modificar el código base; y control de acceso basado en roles para resguardar la seguridad y confidencialidad.

El desarrollo del sistema no solo cumple con los objetivos planteados, sino que también establece una base sólida para la mejora continua de los procesos de atención al cliente y la escalabilidad operativa de Claro-Nicaragua.

## Índice

I.	Introducción.....	1
II.	Antecedentes.....	2
III.	Justificación.....	4
IV.	Objetivos.....	5
4.1.	General.....	5
4.2.	Específicos.....	5
V.	Marco Teórico.....	6
5.1.	Metodología Cascada.....	6
5.2.	Control de solicitudes para servicios de Telecomunicaciones de la empresa Claro-Nicaragua.....	8
5.3.	ITIL.....	11
5.4.	Herramientas de desarrollo.....	15
VI.	Informe de costos.....	19
6.1.	Costos de desarrollo.....	19
6.2.	Requerimientos técnicos.....	20
VII.	Análisis y presentación de resultados: Metodología y desarrollo del tema.....	21
7.1.	Diseño Metodológico.....	21
7.1.1.	Metodología Cascada.....	21
7.2.	Resultados.....	23
7.2.1.	Planificación.....	23
7.2.2.	Recopilación de requisitos.....	23
7.2.3.	Diseño del sistema.....	44
7.2.4.	Implementación.....	53
7.2.5.	Pruebas.....	64
7.2.6.	Despliegue.....	79
7.2.7.	Mantenimiento.....	84
VIII.	Conclusiones.....	85
IX.	Recomendaciones.....	86
X.	Bibliografía.....	87

## Índice de Figuras

<b>Figura 1</b> Flujo de atención de solicitudes de telecomunicaciones en Claro-Nicaragua .....	11
<b>Figura 2</b> Flujo de Proceso de Cumplimiento de Solicitudes.....	13
<b>Figura 3</b> Etapas de la metodología Cascada.....	22
<b>Figura 4</b> Roles del sistema.....	25
<b>Figura 5</b> Caso de uso Solicitudes .....	25
<b>Figura 6</b> Caso de uso Gestión de Solicitudes.....	26
<b>Figura 7</b> Caso de uso Formularios.....	26
<b>Figura 8</b> Caso de Uso Reportes.....	27
<b>Figura 9</b> Caso de uso Tipos de Solicitud .....	27
<b>Figura 10</b> Caso de uso Jerarquía de Solicitud .....	28
<b>Figura 11</b> Crear Solicitud.....	43
<b>Figura 12</b> Solicitudes Registradas .....	44
<b>Figura 13</b> Asignar solicitud .....	44
<b>Figura 14</b> Detalle de solicitud.....	45
<b>Figura 15</b> Diagrama de actividad Registrar Solicitud .....	46
<b>Figura 16</b> Diagrama de actividad Eliminar Solicitud .....	47
<b>Figura 17</b> Diagrama de actividad Redirigir Solicitud .....	47
<b>Figura 18</b> Diagrama de actividad Cambiar Estado Solicitud.....	48
<b>Figura 19</b> Diagrama de secuencia Registrar Solicitud .....	49
<b>Figura 20</b> Diagrama de secuencia Eliminar Solicitud.....	50
<b>Figura 21</b> Diagrama de secuencia Redireccionar Solicitud .....	50
<b>Figura 22</b> Diagrama entidad Relación .....	51
<b>Figura 23</b> Configuración del entorno de desarrollo .....	52
<b>Figura 24</b> Modelado de datos con prisma .....	53
<b>Figura 25</b> Configuración de Autenticación con NextAuth.....	54
<b>Figura 26</b> Código Frontend con React (Detalle de solicitud). .....	55
<b>Figura 27</b> Interfaz Listado de solicitudes .....	56
<b>Figura 28</b> Interfaz detalle de solicitud .....	57
<b>Figura 29</b> Página de inicio de sesión .....	58
<b>Figura 30</b> Dashboard general.....	59
<b>Figura 31</b> Crear formularios .....	60
<b>Figura 32</b> Diseño de formulario.....	60
<b>Figura 33</b> Flujos de trabajo.....	61
<b>Figura 34</b> Diseño de flujos de trabajo .....	61
<b>Figura 35</b> Historial de commits en git.....	62
<b>Figura 36</b> Código de prueba ver formulario de inicio de sesión.....	65
<b>Figura 37</b> Resultado de prueba ver formulario de inicio de sesión.....	66
<b>Figura 38</b> Código de prueba manejo de credenciales incorrectas al iniciar sesión ..	67

<b>Figura 39</b> Resultado de prueba manejo de credenciales incorrectas al iniciar sesión .....	68
<b>Figura 40</b> Código de prueba inicio de sesión con credenciales correctas .....	69
<b>Figura 41</b> Resultado de prueba inicio de sesión con credenciales correctas .....	70
<b>Figura 42</b> Prueba unitaria código de la prueba actualizar estado de solicitud .....	72
<b>Figura 43</b> Prueba unitaria resultado de la prueba actualizar estado de solicitud.....	73
<b>Figura 44</b> Prueba unitaria código de prueba actualizar solicitud existente .....	75
<b>Figura 45</b> Prueba unitaria resultado de prueba actualizar solicitud existente .....	75
<b>Figura 46</b> Pruebas unitarias código de prueba ver detalle de solicitud .....	76
<b>Figura 47</b> Pruebas unitarias resultado de prueba ver detalle de solicitud.....	77
<b>Figura 48</b> Diagrama de despliegue del sistema web y base de datos .....	78
<b>Figura 49</b> Prueba de Conectividad (Ping) desde Servidor de Aplicación a Servidor de Base de Datos .....	79
<b>Figura 50</b> Prueba de Conectividad (Ping) del Servidor de Base de Datos al Servidor de Aplicación .....	80
<b>Figura 51</b> Base de datos alojada en SQL Server .....	81
<b>Figura 52</b> Estado operativo del servicio web gestionado por PM2.....	81

## Índice de tablas

<b>Tabla 1</b>	Resumen de Costos de Desarrollo del Sistema .....	20
<b>Tabla 2</b>	Especificaciones Técnicas del Sistema .....	21
<b>Tabla 3</b>	Roles del sistema .....	24
<b>Tabla 4</b>	Explicación de los casos de uso .....	28
<b>Tabla 5</b>	Caso de uso Registrar Solicitud .....	29
<b>Tabla 6</b>	Caso de uso Mostrar Solicitudes .....	30
<b>Tabla 7</b>	Caso de uso Clasificar Solicitud.....	31
<b>Tabla 8</b>	Caso de uso Administrar Requisitos de Solicitudes .....	32
<b>Tabla 9</b>	Caso de uso Redireccionar Solicitud .....	34
<b>Tabla 10</b>	Caso de uso Generar Reportes de Solicitudes .....	35
<b>Tabla 11</b>	Caso de uso Asignar solicitud .....	37
<b>Tabla 12</b>	Caso de uso Mostrar Formularios.....	38
<b>Tabla 13</b>	Requerimientos funcionales .....	41
<b>Tabla 14</b>	Requerimientos no funcionales.....	42
<b>Tabla 15</b>	Criterios de aceptación para pruebas de inicio de sesión.....	64
<b>Tabla 16</b>	Caso de Prueba E2E Ver formulario de inicio de sesión .....	65
<b>Tabla 17</b>	Caso de Prueba E2E Manejo de credenciales incorrectas al iniciar sesión .....	67
<b>Tabla 18</b>	Caso de Prueba E2E Inicio de sesión con credenciales correctas .....	69
<b>Tabla 19</b>	Prueba unitaria criterios de aceptación para el módulo “Configuración” .....	71
<b>Tabla 20</b>	Prueba unitaria: actualizar estado de solicitud .....	72
<b>Tabla 21</b>	Prueba unitaria criterios de aceptación para el módulo de “Solicitudes” .....	73
<b>Tabla 22</b>	Prueba unitaria actualizar solicitud existente.....	74
<b>Tabla 23</b>	Prueba unitaria ver detalle de solicitud .....	76

## **Índice de anexos**

<b>Anexo A</b> Diseño de interfaces gráficas de usuario de sistema.....	1
<b>Anexo B</b> Casos de uso.....	5
<b>Anexo C</b> Explicación de los casos de uso .....	9
<b>Anexo D</b> Diagramas de Secuencias .....	16
<b>Anexo E</b> Diagramas de Actividades.....	21
<b>Anexo F</b> Pruebas unitarias .....	24
<b>Anexo G</b> Pruebas de Sistema.....	26
<b>Anexo H</b> Carta de aprobación de Claro-Nicaragua.....	35
<b>Anexo I</b> Instrumento de recolección de datos: Entrevista a usuarios claves .....	36
<b>Anexo J</b> Cronograma de actividades.....	37

## Índice de Figuras de los Anexos

<b>Figura A. 1</b> Menú de navegación y dashboards.....	1
<b>Figura A. 2</b> Crear Formulario .....	2
<b>Figura A. 3</b> Tipos de Solicitudes .....	3
<b>Figura A. 4</b> Requerimientos de tipos de solicitudes .....	3
<b>Figura A. 5</b> Crear / Editar Tipo Solicitud .....	4
<b>Figura B. 1</b> Caso de uso áreas .....	5
<b>Figura B. 2</b> Caso de uso jerarquía de asignación .....	5
<b>Figura B. 3</b> Caso de uso tipos de solicitud .....	6
<b>Figura B. 4</b> Caso de uso jerarquía de solicitud .....	6
<b>Figura B. 5</b> Caso de uso requisitos de solicitud .....	7
<b>Figura B. 6</b> Caso de uso tipos de requisitos de solicitud .....	7
<b>Figura B. 7</b> Caso de uso prioridades .....	8
<b>Figura B. 8</b> Caso de uso flujos de trabajo .....	8
<b>Figura D. 1</b> Diagrama de secuencia generar reporte de solicitudes.....	16
<b>Figura D. 2</b> Diagrama de secuencia mostrar formularios.....	17
<b>Figura D. 3</b> Diagrama de secuencia mostrar áreas.....	17
<b>Figura D. 4</b> Diagrama de secuencia agregar / editar área.....	18
<b>Figura D. 5</b> Diagrama de secuencia mostrar jerarquía de asignación.....	19
<b>Figura D. 6</b> Diagrama de secuencia mostrar tipos de solicitud.....	20
<b>Figura D. 7</b> Diagrama de secuencia mostrar jerarquía de solicitud.....	20
<b>Figura D. 8</b> Diagrama de secuencia mostrar requisitos de solicitud.....	21
<b>Figura E. 1</b> Diagrama de actividad visualizar reportes.....	22
<b>Figura E. 2</b> Diagrama de actividad mostrar formularios .....	22
<b>Figura E. 3</b> Diagrama de actividad mostrar área.....	23
<b>Figura E. 4</b> Diagrama de actividad mostrar jerarquía de solicitud .....	23
<b>Figura E. 5</b> Diagrama de actividad mostrar requisitos de solicitud.....	24

<b>Figura F. 1</b> Código de prueba unitaria permisos completos para administrador global .....	25
<b>Figura F. 2</b> Resultado de la prueba unitaria permisos completos para administrador global.....	25
<b>Figura G. 1</b> Pruebas de sistema código para crear nueva área con todos los campos requeridos .....	27
<b>Figura G. 2</b> Pruebas de sistema resultado de crear nueva área con todos los campos requeridos .....	27
<b>Figura G. 3</b> Código de la prueba de sistema crear nuevo formulario y redirigir al constructor .....	29
<b>Figura G. 4</b> Resultado de la prueba de sistema crear nuevo formulario y redirigir al constructor .....	29
<b>Figura G. 5</b> Código de la prueba de sistema crear nuevo tipo de requisito con todos los campos.....	30
<b>Figura G. 6</b> Resultado de la prueba de sistema crear nuevo tipo de requisito con todos los campos.....	31
<b>Figura G. 7</b> Código de prueba de sistema crear nueva solicitud .....	33
<b>Figura G. 8</b> Resultado de prueba de sistema crear flujo de trabajo completo con nodos conectados .....	33

## **Índice de Tablas de los Anexos**

<b>Tabla C. 1</b> Caso de uso crear formulario .....	9
<b>Tabla C. 2</b> Caso de uso mostrar áreas .....	11
<b>Tabla C. 3</b> Caso de uso mostrar tipos de solicitud .....	12
<b>Tabla C. 4</b> Caso de uso mostrar requisitos de solicitud .....	13
<b>Tabla C. 5</b> Caso de uso mostrar tipos de requisitos .....	14
<b>Tabla C. 6</b> Caso de uso mostrar usuarios .....	15
<b>Tabla F. 1</b> Prueba unitaria permisos completos para administrador global .....	24
<b>Tabla G. 1</b> Pruebas de sistema criterios de aceptación para módulo “Área” .....	25
<b>Tabla G. 2</b> Prueba de sistema crear nueva área con todos los campos requeridos ..	26
<b>Tabla G. 3</b> Pruebas de sistema criterios de aceptación para pruebas E2E del módulo de creación de formulario .....	27
<b>Tabla G. 4</b> Pruebas de sistema crear nuevo formulario y redirigir al constructor .....	28
<b>Tabla G. 5</b> Prueba de sistema crear nuevo tipo de requisito con todos los campos ..	29
<b>Tabla G. 6</b> Prueba de sistema criterios de aceptación para pruebas E2E del módulo de Solicitudes .....	31
<b>Tabla G. 7</b> Prueba de sistema crear una nueva solicitud .....	32

## **I. INTRODUCCIÓN**

Un servicio de asistencia técnica o mesa de servicio es un punto de contacto único que proporciona información centralizada y un servicio de gestión de soporte para manejar las consultas internas o externas de una empresa. Una solución de software de asistencia técnica permite a las empresas resolver las quejas de los clientes de manera más rápida y eficiente simplemente automatizando el proceso de resolución de quejas con el sistema de gestión de tickets (Ameyo, 2020).

Claro Nicaragua ha tomado la decisión de implementar un nuevo sistema a medida que solventa las limitaciones existentes en el sistema informático que se utiliza actualmente. Esta medida se ha adoptado con el objetivo de mejorar la experiencia del cliente y facilitar una respuesta eficiente, oportuna a sus necesidades y consultas. El sistema personalizado permitirá a la empresa adaptar sus procesos de soporte y atención al cliente de manera específica, favoreciendo una mayor eficacia en la resolución de problemas y una comunicación más fluida.

Para el desarrollo del sistema web se ha elegido la metodología de cascada, la cual es una técnica ampliamente empleada y validada en numerosos proyectos. Este enfoque de desarrollo de software se distingue por su naturaleza secuencial y lineal, dividido en etapas claramente definidas. Estos aspectos aunados al uso de tecnologías de desarrollo web modernas (React, Next.js, Prisma, TypeScript, etc.) aseguran el desarrollo de este proyecto con un enfoque estructurado y eficiente que maximiza las posibilidades de éxito de este. El desarrollo de este sistema ofrece a Claro beneficios a corto plazo como una atención al cliente más eficiente y una gestión más organizada de las solicitudes. A largo plazo, proporciona una base sólida para la escalabilidad del negocio, mejoras continuas en la experiencia del cliente y una mayor eficiencia operativa.

## **II. ANTECEDENTES**

Claro-Nicaragua, previo al despliegue de este proyecto, gestionaba las solicitudes de servicio de telecomunicaciones a través de un sistema de información HelpDesk desarrollado internamente. A lo largo del tiempo, se identificaron múltiples deficiencias en el sistema, ya que no se incorporaron diversas funcionalidades esenciales para facilitar el seguimiento de estos. Esta falta de características necesarias obligaba al personal encargado a realizar tareas complementarias de manera manual para abordar estas deficiencias y subsanar la situación (Torrez, 2024).

Entre las principales deficiencias identificadas se encontraba la ausencia de un mecanismo automatizado de seguimiento y notificación de las solicitudes, lo que obligaba al personal a utilizar canales de comunicación alternos como llamadas telefónicas, correos electrónicos o aplicaciones de mensajería instantánea para informar cambios de estado. Esta práctica generaba falta de trazabilidad, escasa visibilidad del ciclo de vida de las solicitudes y dificultades para evaluar el desempeño operativo y la distribución de la carga laboral.

Asimismo, el sistema no contemplaba validaciones previas al registro de las solicitudes, lo que provocaba reprocesos, revisiones redundantes y pérdida de tiempo una vez que las solicitudes ya se encontraban creadas. De igual manera, se identificó la carencia de funcionalidades fundamentales para una gestión eficiente, tales como la asignación de prioridades, la reasignación de solicitudes entre áreas, la reapertura de solicitudes cerradas y la generación de reportes personalizados para el análisis de la información.

Ante estas limitaciones, se realizó una revisión de algunas soluciones HelpDesk y Service Desk disponibles en el mercado que podrían representar una alternativa para la empresa, considerando especialmente opciones gratuitas o de bajo costo, dado que este factor resulta relevante para Claro Nicaragua. Entre los sistemas analizados se encuentran ProProfs HelpDesk (ProProfs, s.f.), Zendesk (Zendesk, 2020) y SolarWinds HelpDesk (SolarWinds, s.f.). Sin embargo, dicha evaluación evidenció que estas

herramientas también presentan restricciones que limitan su adopción en el contexto específico de la organización.

ProProfs HelpDesk, aunque ofrece funcionalidades básicas para la gestión de consultas, presenta limitaciones en la personalización, integraciones y complejidad de configuración inicial, lo que reduce su flexibilidad para adaptarse a procesos empresariales más específicos (ProProfs, s.f.). Por su parte, Zendesk dispone de herramientas robustas para la gestión de tickets; no obstante, su esquema de licenciamiento y costos restringe el acceso a funcionalidades avanzadas, lo cual puede resultar poco viable para la empresa (Zendesk, 2020). Finalmente, SolarWinds HelpDesk se encuentra orientado principalmente a la gestión de incidencias en áreas de Tecnologías de la Información, con escasas posibilidades de personalización, costos elevados y limitaciones para incorporar nuevas funciones que se ajusten al flujo de negocio requerido (SolarWinds, s.f.).

En este contexto, se evidenció la carencia de una solución tecnológica personalizada que respondiera de manera adecuada a las particularidades operativas de Claro Nicaragua. Las limitaciones del sistema existente y las restricciones de las alternativas comerciales generaban consecuencias adversas, destacándose la pérdida de eficiencia operativa, dificultades en la gestión de las solicitudes y un impacto negativo en la calidad del servicio brindado a los clientes.

### **III. JUSTIFICACIÓN**

El desarrollo de este sistema web se fundamenta en la necesidad de modernizar y fortalecer la gestión de solicitudes de servicios de telecomunicaciones en Claro Nicaragua, superando los esquemas de trabajo manuales y dispersos que limitaban el control operativo. Esta solución tecnológica no se plantea únicamente como una herramienta de automatización, sino como un activo estratégico diseñado bajo los estándares internacionales de buenas prácticas de ITIL. Al adoptar este marco como una guía técnica adaptable y no como un esquema rígido, el sistema logra alinearse de forma precisa con el modelo de negocio y las reglas operativas de la empresa, consolidando la estandarización de cada solicitud bajo un flujo de trabajo definido.

La implementación de esta solución permite centralizar toda la información en un entorno único, reduciendo la dependencia de canales de comunicación informales y habilitando una trazabilidad detallada del ciclo de vida de cada servicio. El sistema facilita la transición de una gestión reactiva a un modelo estructurado, donde es posible monitorear tiempos de respuesta, asignar prioridades y validar requisitos de forma automática antes del registro. Esto optimiza la coordinación entre las áreas involucradas y disminuye significativamente el margen de error humano y las tareas duplicadas, elevando la eficiencia institucional de manera integral.

Además, al tratarse de un desarrollo propio y a medida, el sistema ofrece una flexibilidad y escalabilidad que las soluciones genéricas del mercado no pueden proveer. Esta característica es fundamental, ya que permite que la herramienta evolucione y se ajuste progresivamente a las nuevas demandas operativas de la organización sin restricciones técnicas externas. En última instancia, la relevancia de este proyecto reside en su capacidad para transformar datos en información útil; a través de sus módulos de reportes y métricas, se dota a la gerencia de una base sólida para la toma de decisiones estratégicas, impactando de manera directa y positiva en la calidad del servicio final que se brinda a los usuarios.

## **IV. OBJETIVOS**

### **4.1. General**

1. Desarrollar un sistema web que permita el control de solicitudes para servicios de telecomunicaciones en la empresa Claro-Nicaragua.

### **4.2. Específicos**

2. Analizar el proceso y las prácticas en la gestión de solicitudes de servicios de telecomunicaciones de Claro-Nicaragua incorporando las buenas prácticas de "cumplimiento de solicitudes" conforme a ITIL 2011.
3. Diseñar el sistema web mediante la creación de maquetas para la interfaz de usuario y el uso de diagramas UML para la modelación del sistema.
4. Implementar el sistema utilizando React, Next.js, Prisma, TypeScript, TRPC, NextAuth y SQL Server, asegurando la gestión de cambios mediante control de versiones con Git.
5. Realizar pruebas que garanticen el funcionamiento adecuado del sistema web desarrollado.
6. Desplegar el sistema web en un servidor de la empresa Claro-Nicaragua.

## V. MARCO TEÓRICO

El presente marco teórico establece los fundamentos conceptuales esenciales para abordar de manera efectiva el problema en cuestión y proporciona la base sobre la cual se sustenta todo el trabajo. En este sentido, es necesario tener conocimiento de los términos y conceptos relacionados con el campo de estudio del desempeño.

### 5.1. Metodología Cascada

#### Definición

El modelo Cascada (Waterfall) es un proceso metódico y paso a paso de creación de software de una etapa a otra. Este enfoque requiere completar cada fase antes de que pueda comenzar la siguiente. El objetivo principal del modelo Waterfall es garantizar que se cumplan todos los requisitos del software y que el software funcione según lo previsto. Ofrece un enfoque estructurado y lógico para la creación de software que garantiza alta calidad y eficiencia durante todo el proceso de desarrollo (Kiruthika, 2024).

#### Etapas

El modelo en cascada es un enfoque tradicional de desarrollo de software que sigue un proceso lineal y secuencial. Consta de varias fases distintas, cada una con sus objetivos y entregables (Kiruthika, 2024):

1. **Planificación:** La primera fase, sienta las bases de todo el proyecto. Durante esta fase, el equipo del proyecto define el alcance, los objetivos y las limitaciones del producto de software. Se trata de recopilar información sobre los requisitos, objetivos y expectativas del cliente, que sirve de base para las fases posteriores.
2. **Recopilación de requisitos:** En esta fase, el equipo del proyecto identifica y documenta los requisitos, objetivos y expectativas del cliente para el producto de software.
3. **Diseño del sistema:** La fase de diseño del sistema comienza una vez que se recopilan los requisitos. Implica crear un diseño arquitectónico de alto nivel que describa cómo interactuarán entre sí los diferentes componentes del software.

4. **Implementación(codificación):** La fase de implementación se centra en traducir el diseño del sistema en código real. Los desarrolladores escriben código basándose en las especificaciones proporcionadas durante las fases anteriores.
5. **Pruebas:** Después de la implementación, se realizan pruebas exhaustivas para identificar cualquier defecto o error en el software. Esto incluye pruebas unitarias (probar componentes individuales), pruebas de integración (probar cómo funcionan juntos los diferentes componentes) y pruebas de sistemas (probar todo el sistema). Las pruebas de automatización juegan un papel importante. Nos ayuda a conseguir resultados más rápidos y precisos.
6. **Despliegue:** una vez que se completan todas las pruebas, el software está listo para su despliegue, donde los usuarios finales pueden instalarlo y utilizarlo.
7. **Mantenimiento:** Durante la fase de mantenimiento, se implementan las actualizaciones o correcciones de errores necesarias para garantizar que el software funcione sin problemas y de manera eficiente.

#### **Herramientas del modelo de cascada.**

Algunas de las herramientas que permiten seguir el modelo en cascada durante el desarrollo de software son (Kiruthika, 2024):

1. **El diagrama de Gantt** es una herramienta esencial en el modelo en cascada, que permite a los gerentes de proyecto planificar y rastrear visualmente las fases secuenciales del proyecto, asegurando una asignación eficiente de recursos y una finalización oportuna.
2. **Las herramientas de documentación** desempeñan un papel importante en el modelo en cascada al capturar especificaciones detalladas del proyecto, documentos de diseño, planes de prueba y manuales de usuario, lo que garantiza una documentación exhaustiva para referencia y mantenimiento futuros. Una vez documentados los requisitos, comienza la fase de diseño. Aquí, se pueden utilizar herramientas de modelado UML (Unified Modeling Language), como Enterprise Architect o Lucidchart, para crear documentos de diseño detallados.

Estas herramientas permiten a los desarrolladores representar visualmente la arquitectura del sistema, diagramas de clases, diagramas de secuencia y otros elementos de diseño necesarios para construir el software. Para crear un documento de requisitos completo que describa todos los requisitos funcionales y no funcionales del software, se pueden utilizar programas como Microsoft Word o Google Docs.

3. **Los sistemas de control de versiones** ayudan a administrar diferentes versiones (Git) de software o código a lo largo del proceso de desarrollo en el modelo en cascada, lo que permite a los desarrolladores realizar un seguimiento de los 8 cambios, colaborar de manera efectiva y mantener un repositorio de código fuente confiable.

## **5.2. Control de solicitudes para servicios de Telecomunicaciones de la empresa Claro-Nicaragua**

### **5.2.1. Descripción de la empresa**

La empresa de telecomunicaciones Claro-Nicaragua con sede central en Ciudad de México, tiene por objetivo prescindir las fronteras y relacionar con los clientes. La compañía de telecomunicaciones Claro provee sus servicios en Nicaragua desde 1979. A la actualidad, esta empresa tiene presencia en todo el territorio nacional.

#### **5.2.1.1. Misión**

Proporcionar a nuestros clientes los mejores productos y servicios de Telecomunicaciones con rapidez, eficiencia y a precio competitivos (Claro-Nicaragua, 2011)

#### **5.2.1.2. Visión**

Mantenerse como la empresa líder en Telecomunicaciones en el mercado nicaragüense, expandiendo su gama de productos y servicios en todos los mercados (Claro-Nicaragua, 2011).

### **5.2.1. Proceso actual de gestión de solicitudes en Claro-Nicaragua para servicios de telecomunicaciones**

Previo al desarrollo de este proyecto, el proceso de gestión de solicitudes en Claro-Nicaragua para servicios de telecomunicaciones que implica la activación, renovación o suspensión de servicios para clientes de las categorías Gobierno, Grandes Empresas y Pymes se lleva a cabo mediante la colaboración de diversas áreas, como Post Venta, Venta y Activación, las cuales están encargadas de atender dichas solicitudes. Este proceso se realizaba a través de un sistema HelpDesk, en el cual se registraban las solicitudes con información requerida, incluyendo el asunto, descripción, categoría, subcategoría y área responsable a atender la solicitud (Mendez, 2024). El flujo de cómo funcionaba la empresa para atender dichas solicitudes se encuentra detallado en la figura 1.

Claro-Nicaragua ofrece una amplia gama de servicios de telecomunicaciones orientados tanto a clientes residenciales como a clientes empresariales y gubernamentales. En el contexto de este estudio, los servicios analizados corresponden principalmente a los segmentos de Gobierno, Grandes Empresas y Pequeñas y Medianas Empresas (Pymes), los cuales generan la mayor parte de las solicitudes gestionadas por la empresa.

Entre los principales servicios de telecomunicaciones que ofrece Claro-Nicaragua se encuentran:

- Telefonía móvil corporativa y gubernamental, que incluye la activación de líneas, cambios de plan, suspensión temporal o definitiva del servicio, reposición de tarjetas SIM y renovación de contratos.
- Telefonía fija, orientada a entidades gubernamentales y empresas, la cual contempla solicitudes relacionadas con la instalación de líneas, traslado de servicios, suspensión, reconexión y mantenimiento técnico.
- Servicios de internet, que abarcan conexiones de banda ancha fija y móvil, incluyendo solicitudes de activación, ampliación de ancho de banda, renovación de contratos y atención de incidencias técnicas.

- Transmisión de datos y enlaces dedicados, dirigidos principalmente a clientes corporativos y gubernamentales, utilizados para garantizar conectividad segura y estable entre sedes, lo que genera solicitudes de instalación, monitoreo, mantenimiento y suspensión del servicio.
- Soluciones corporativas, tales como servicios de conectividad empresarial, soluciones en la nube, seguridad de red y servicios administrados, los cuales requieren procesos constantes de gestión, validación y seguimiento por parte de las áreas involucradas.

Todos estos servicios constituyen la base sobre la cual se originan las solicitudes de activación, renovación, suspensión y atención de requerimientos operativos que son gestionados mediante el sistema HelpDesk utilizado por Claro-Nicaragua.

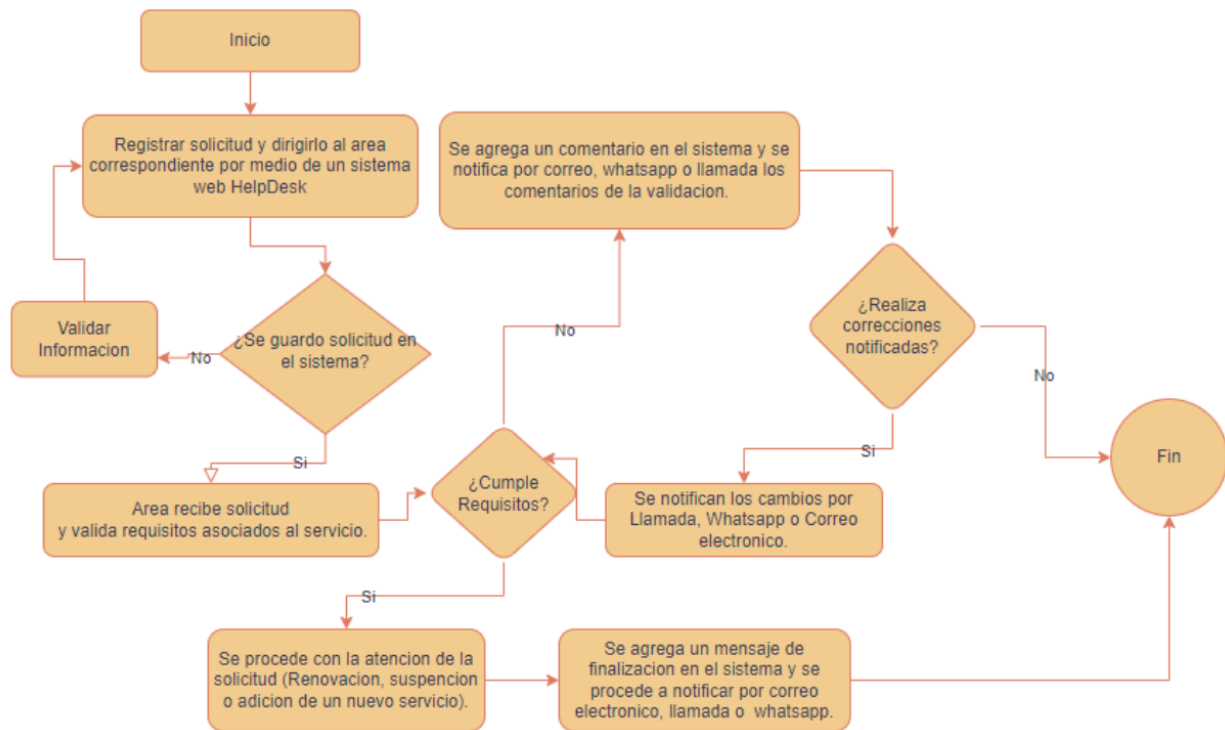
La notificación sobre la creación y cambios de estados o comentarios de los casos se realizaba a través de llamadas telefónicas, correos electrónicos o WhatsApp. Además, en el sistema se registraban solicitudes realizadas por los distribuidores de Claro-Nicaragua.

Estas solicitudes pueden incluir la solicitud de permisos para realizar ventas, la presentación de reclamos, la solicitud de validaciones de clientes, entre otros requerimientos (Mendez, 2024).

Es importante resaltar que solo el personal que trabaja directamente para la empresa ya sea interno o externo, tiene acceso al sistema. Los clientes interesados en un servicio deben dirigirse a un punto de venta autorizado.

**Figura 1**

Flujo de atención de solicitudes de telecomunicaciones en Claro-Nicaragua



*Nota.* Diagrama del flujo de atención de solicitudes, elaborado a partir de las entrevistas realizadas al personal de Claro Nicaragua. Fuente: Elaboración propia, 2025.

### 5.3. ITIL

ITIL es un conjunto de conceptos y mejores prácticas referentes a la gestión de servicios TI (tecnologías de la información), y describe detalladamente un extenso conjunto de funciones y procesos ideados para ayudar a las organizaciones a lograr calidad y eficiencia en las operaciones de TI (Ayala, 2019).

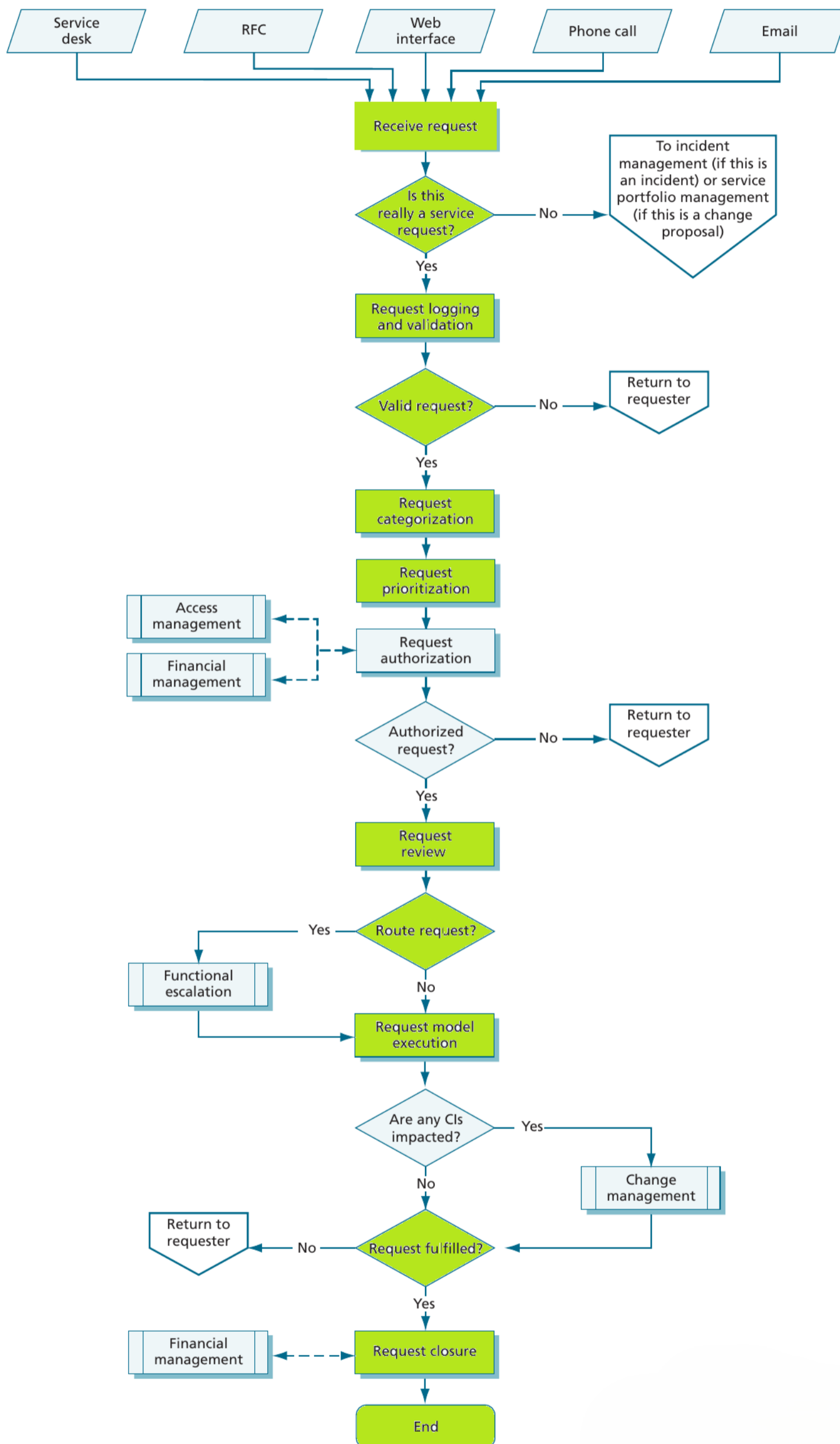
ITIL no es un estándar que deba seguirse; es una guía que debe leerse y entenderse, y utilizarse para crear valor tanto para el proveedor de servicios como para sus clientes. Se alienta a las organizaciones a adoptar las mejores prácticas de ITIL y adaptarlas para trabajar en sus entornos específicos de manera que satisfagan sus necesidades (Steinberg, 2011).

El término "solicitud de servicio" se utiliza como una descripción genérica para muchos tipos diferentes de demandas o peticiones que son colocadas en la organización de TI por los usuarios. Muchas de estas son típicamente solicitudes de

cambios pequeños que tienen bajo riesgo, son realizadas con frecuencia, tienen bajo costo, etc. (por ejemplo, una solicitud para cambiar una contraseña, una solicitud para instalar una aplicación de software adicional en una estación de trabajo específica, una solicitud para reubicar algunos elementos de equipo de escritorio) o pueden ser simplemente una solicitud de información (Steinberg, 2011).

La gestión de las solicitudes de servicio es cubierta en el proceso “Cumplimiento de solicitudes” (conocido como “Request fulfillment” en inglés), el cual es incluido en el libro “Service Operations” de la versión 2011 de ITIL. El proceso “Cumplimiento de solicitudes” propone las buenas prácticas para el cumplimiento de solicitudes de servicio que abarcan la definición de flujos de proceso predefinidos para garantizar la consistencia y eficiencia en la gestión de solicitudes, la centralización de la propiedad de las solicitudes en una función centralizada como la mesa de ayuda, la implementación de cambios estándar para solicitudes que afectan elementos de configuración, el registro y control integral de solicitudes a través de un único sistema, la autorización previa de todas las solicitudes, el cumplimiento de solicitudes basado en criterios de prioridad alineados con los objetivos de servicio, y la facilitación de una comunicación clara y accesible para solicitar y rastrear el estado de las solicitudes (Steinberg, 2011), lo cual se resume en la figura 2.

**Figura 2**  
Flujo de Proceso de Cumplimiento de Solicitudes



**Fuente:** Adaptado de ITIL Service Operation 2011 Edition (p. 90), por Steinberg, 2011. Nota: Se aplicaron tonalidades verdes para resaltar componentes claves.

Se han resaltado en color verde las actividades del proceso de “cumplimiento de solicitudes” que fueron implementadas en el software desarrollado; mientras que en color gris se indican aquellas que, de acuerdo con la forma de trabajo de Claro-Nicaragua, no fue necesario incluir.

En concordancia con las buenas prácticas de cumplimiento de solicitudes propuestas en ITIL 2011 por Steinberg, se consideran las siguientes pautas, adaptadas al modelo de negocio de Claro-Nicaragua, las cuales fueron incorporadas en el sistema desarrollado:

- Se destaca la importancia de registrar, controlar y gestionar todas las solicitudes a lo largo de su ciclo de vida mediante un único sistema. En este sentido, el sistema web a desarrollar se encargará de administrar estas solicitudes, asegurando una gestión eficiente y unificada de todas las actividades relacionadas con ellas.
- Categorizar las solicitudes es una parte esencial del proceso de registro inicial. En el sistema, esta categorización se llevará a cabo respetando el modelo de negocio actual de Claro-Nicaragua, que consiste en clasificar las solicitudes según su tipo.
- Determinar estados de las solicitudes. En este proceso, se empleará el seguimiento del estado de las solicitudes, considerando los siguientes estados:
  - Borrador: una solicitud se crea en estado de borrador.
  - En revisión: solicitudes que han sido autorizadas y están bajo revisión por parte de quienes realizarán las actividades de cumplimiento para ellas.
  - Cancelado: la solicitud ya no es necesaria para el usuario.
  - En progreso: la solicitud está en proceso de ser cumplida.
  - Cerrado: La solicitud ha sido cumplida y la solicitud ha sido cerrada.
- Priorización: Asignar un código de priorización adecuado, ya que esto determinará cómo se maneja la solicitud. En este sentido, se establecerán tres niveles de prioridad: alta, media y baja, adaptándose así a los requisitos y necesidades específicas de la empresa.

- **Revisión de solicitudes:** Esta etapa implica evaluar y determinar cómo se cumplirá cada solicitud de servicio, asegurando que se asignen los recursos adecuados y que se manejen de manera eficiente para satisfacer las necesidades del usuario.
- **Resolución:** implica una serie de pasos que van desde el registro inicial hasta el cierre final, asegurando que se aborden de manera efectiva y eficiente las necesidades y solicitudes de los usuarios.
- **Ejecución del modelo de solicitud:** Es establecer un proceso estandarizado y consistente para manejar y cumplir con las solicitudes de servicio. Esto implica seguir un conjunto definido de pasos y actividades para garantizar que todas las solicitudes se gestionen de manera eficiente y efectiva, minimizando los riesgos de demoras o fallo. Para lograrlo, se contempla el almacenamiento y acceso a los modelos de solicitud en documentos de referencia dentro del sistema.
- **Cierre de solicitudes:** Una vez que se hayan completado las actividades de la solicitud de servicio, el sistema enviará una notificación por correo electrónico a la persona que creó la solicitud, informándole sobre el cambio de estado de esta.

#### **5.4. Herramientas de desarrollo**

Durante el desarrollo del sistema, se implementaron diversas tecnologías, todas ellas de código abierto, lo que permitió garantizar flexibilidad, escalabilidad y un acceso amplio a recursos y comunidades de soporte. A continuación, se describen las herramientas y tecnologías más relevantes utilizadas en el proyecto:

- **Next Js**

Next.js es un marco de interfaz de usuario que facilita la creación de sitios web rápidos con React, que es una biblioteca de JavaScript de interfaz de usuario gratuita y de código abierto para crear interfaces de usuario basadas en componentes de interfaz de usuario (Sigdestad, 2022).

- **React**

React es una biblioteca de desarrollo de UI basada en JavaScript. Aunque React es más una biblioteca que un lenguaje, se usa ampliamente en el desarrollo web. La biblioteca apareció por primera vez en mayo de 2013 y ahora es una de las bibliotecas frontend más utilizadas para el desarrollo web (Deshpande, 2023).

- **TypeScript**

TypeScript es un lenguaje de programación desarrollado por Microsoft que agrega escritura estática a JavaScript, haciéndolo más robusto, mantenible y fácil de escalar. Es un lenguaje de código abierto y ha ganado una inmensa popularidad entre los desarrolladores debido a su perfecta integración con proyectos JavaScript existentes y sus numerosas características que mejoran la experiencia de desarrollo (Williams, 2023).

- **SQL Server**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS). Su función principal es almacenar y recuperar datos solicitados por otras aplicaciones. Admite diversas aplicaciones de procesamiento de transacciones, inteligencia empresarial y análisis, generalmente en entornos de TI corporativos (Monitor, 2023).

- **Prisma**

Prisma es una herramienta de mapeo relacional de objetos (ORM) de código abierto para aplicaciones Node.js o TypeScript. Prisma tiene varios componentes, incluido Prisma Client, que se utiliza como generador de consultas para comunicarse con bases de datos, Prisma Migrate, que actúa como herramienta de migración, y Prisma Studio, que es una herramienta basada en GUI para administrar datos en la base de datos (Prisma ORM, s.f.)

- **Trpc**

Trpc es un marco de llamada a procedimiento remoto (RPC) de código abierto que es rápido, liviano y permite una comunicación eficiente entre las aplicaciones cliente y servidor. Simplifica la creación de API escalables y de alto rendimiento al automatizar muchas tareas cotidianas necesarias para el desarrollo de API a través de HTTP/2 (Adegoke, 2023).

- **NextAuth**

NextAuth.js es una biblioteca de autenticación flexible y completamente segura diseñada para sincronizarse con cualquier servicio OAuth, con soporte completo para el inicio de sesión sin contraseña (Asiuwahu, 2023).

NextAuth.js se puede utilizar con o sin una base de datos y tiene soporte predeterminado para bases de datos populares como MySQL, MongoDB, PostgreSQL y MariaDB. También se puede utilizar sin una base de datos sincronizándolo con servicios como OAuth y JSON Web Token (Asiuwahu, 2023).

- **Visual Studio Code**

Visual Studio Code es un editor de código fuente gratuito, liviano pero potente que se ejecuta en su escritorio y en la web y está disponible para los sistemas operativos Windows, macOS, Linux y Raspberry Pi. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes de programación (como C++, C#, Java, Python, PHP y Go), tiempos de ejecución (como .NET y Unity), entornos (como Docker y Kubernetes) y nubes (como Amazon Web Services, Microsoft Azure y Google Cloud Platform) (Heller, 2022).

- **Visual Paradigm**

Visual Paradigm es una aplicación de software diseñada para que los equipos de desarrollo de software modelen sistemas de información empresarial y gestionen procesos de desarrollo. Además del soporte de modelado, esta tecnología proporciona capacidades de generación de informes e ingeniería de código, incluida la generación de código. Esta tecnología puede realizar ingeniería inversa de diagramas a partir de código y proporcionar ingeniería de ida y vuelta para varios lenguajes de programación. (Department of Veterans Affairs, Office of Information & Technology, & Enterprise Program Management Office., 2010).

- **Case Studio 2**

Case Studio es una herramienta de modelado e información de bases de datos fácil de usar, personalizable y totalmente integrada que permite a los diseñadores y analistas de bases de datos crear y editar visualmente diagramas de relación de entidades (ERD) y diagramas de flujo de datos (DBF) de forma automática, y genera

scripts SQL para las distintas bases de datos disponibles en el ordenador (Filehippo.com, s.f.).

- **Git**

Sistema de control de versiones distribuido muy utilizado en el desarrollo de software. Fue creado por Linus Torvalds y es conocido por su eficiencia, flexibilidad y capacidad para gestionar proyectos de cualquier tamaño. Git les permite a los desarrolladores realizar un seguimiento de los cambios en el código fuente a lo largo del tiempo, facilitando la colaboración entre equipos al proporcionar un historial detallado de modificaciones (KeepCoding, 2023).

- **Cypress**

Cypress es una poderosa herramienta de automatización de pruebas de front-end, de código abierto, diseñada específicamente para probar aplicaciones web modernas. A diferencia de otras soluciones en el mercado, Cypress ofrece una combinación única de pruebas end-to-end y pruebas unitarias (Souza, 2023).

- **Jest**

Jest es un popular marco de pruebas unitarias de JavaScript que es excelente para minimizar el tiempo de configuración de las pruebas. Emplea comparadores para realizar pruebas en el código base de JavaScript. Los comparadores toman un valor esperado y lo comparan con la salida de una función esperada que toma código JavaScript como entrada (Pfeiffer, 2022).

- **Justinmind**

Se especializa en proporcionar prototipos para aplicaciones web y móviles, brindando servicios a diseñadores, gerentes de productos y desarrolladores con un enfoque en la creación de prototipos de alta fidelidad. La herramienta beneficia a los usuarios con su conjunto completo de funciones diseñadas para crear prototipos interactivos y detallados sin necesidad de conocimientos de codificación (Clark, 2024)

## VI. INFORME DE COSTOS

Para el desarrollo del sistema se hizo uso de un conjunto de tecnologías modernas orientadas al desarrollo web (React, Next.js, Prisma, TypeScript, tRPC y NextAuth) bajo licencias de código abierto (open source), lo que permite su implementación libre de costos por adquisición de software.

No obstante, el sistema utiliza SQL Server como motor de base de datos. En lo que respecta a los costos de licenciamiento de este motor y del sistema operativo del servidor, se establece que estos son asumidos en su totalidad por la infraestructura tecnológica existente de la empresa, por lo que no se contabilizan como un costo directo de inversión para este proyecto de desarrollo específico.

### 6.1. Costos de desarrollo

Los costos se determinaron con base en dos factores principales: el salario mensual de los desarrolladores (mismos autores de este trabajo monográfico) y la cantidad total de horas trabajadas durante el período de desarrollo. Se consideraron las siguientes condiciones:

1. **Moneda de referencia:** Todos los costos se presentan en dólares estadounidenses (USD).
2. De acuerdo con la legislación tributaria vigente en la República de Nicaragua, los servicios profesionales o técnicos prestados por personas naturales están sujetos a una retención del diez por ciento (10 %) sobre el monto total facturado (DGI, 2013).
3. **Equipo de trabajo:** El sistema fue desarrollado por dos personas, quienes participaron en las etapas de análisis, diseño, desarrollo, pruebas y documentación.
4. **Salario mensual:** Salario de US\$ 750.00 mensuales por desarrollador.
5. **Jornada laboral:** El horario de trabajo fue de 9 horas diarias, de lunes a viernes (45 horas semanales).
6. **Duración del proyecto:**
  - Inicio: 17 de septiembre de 2024
  - Finalización: 17 de octubre de 2025

### **Cálculo del costo total:**

- a. **Horas trabajadas por mes:**  $9 \text{ horas/día} \times 5 \text{ días/semana} \times 4 \text{ semanas/mes} = 180 \text{ horas/mes}$
- b. **Costo por hora por desarrollador:**  $750 \text{ USD} \div 180 \text{ horas} = \text{US\$ } 4.17 \text{ por hora}$
- c. **Horas totales del proyecto:**  $180 \text{ horas/mes} \times 13 \text{ meses} = 2,340 \text{ horas por desarrollador}$  (Para dos desarrolladores:  $2,340 \times 2 = 4,680 \text{ horas totales}$ )
- d. **Costo total sin impuestos:**  $4,680 \text{ horas} \times 4.17 \text{ USD} = \text{US\$ } 19,515.60$
- e. **Impuesto aplicable** (10 % según la legislación nicaragüense):  $19,515.60 \times 0.10 = \text{US\$ } 1,951.56$
- f. **Costo total del proyecto:**  $19,515.60 + 1,951.56 = \text{US\$ } 21,467.16$

**Tabla 1**

Resumen de Costos de Desarrollo del Sistema

<b>Concepto</b>	<b>Valor</b>
Salario mensual por desarrollador	US\$ 750.00
Costo por hora	US\$ 4.17
Duración total del proyecto	13 (meses)
Horas totales por desarrollador	2,340 h
Horas totales (2 desarrolladores)	4,680 h
Costo total sin impuestos	US\$ 19,515.60
Impuesto (10 %)	US\$ 1,951.56
<b>Costo total del proyecto</b>	<b>US\$ 21,467.16</b>

### **6.2. Requerimientos técnicos**

Para garantizar el correcto desempeño del sistema, es necesario que el entorno de ejecución y los componentes del servidor cumplan con ciertas especificaciones técnicas que permitan el funcionamiento óptimo de la aplicación.

**Tabla 2**  
Especificaciones Técnicas del Sistema

<b>Lenguaje de desarrollo</b>	<b>Base de datos</b>	<b>Maquetación</b>	<b>Framework y librerías</b>
TypeScript en su versión 5.0 o superior	SQL Server en su versión 2019 o superior	HTML5, CSS3 y compatibilidad con ES6+	React en su versión 18+, Next.js en su versión 14+

## **VII. ANÁLISIS Y PRESENTACIÓN DE RESULTADOS: METODOLOGÍA Y DESARROLLO DEL TEMA**

En esta sección se describe el diseño metodológico empleado, incluyendo el enfoque del estudio, las técnicas de recolección de datos y el proceso de análisis aplicado para interpretar la información obtenida.

Seguidamente, se presentan los resultados organizados según los objetivos específicos, acompañados de su respectivo análisis e interpretación, permitiendo una comprensión clara de los hallazgos en relación con la problemática planteada. Por limitaciones de espacio, este documento presenta una muestra de los resultados generados en el cuerpo principal del documento, y los restantes se incluyen en anexos.

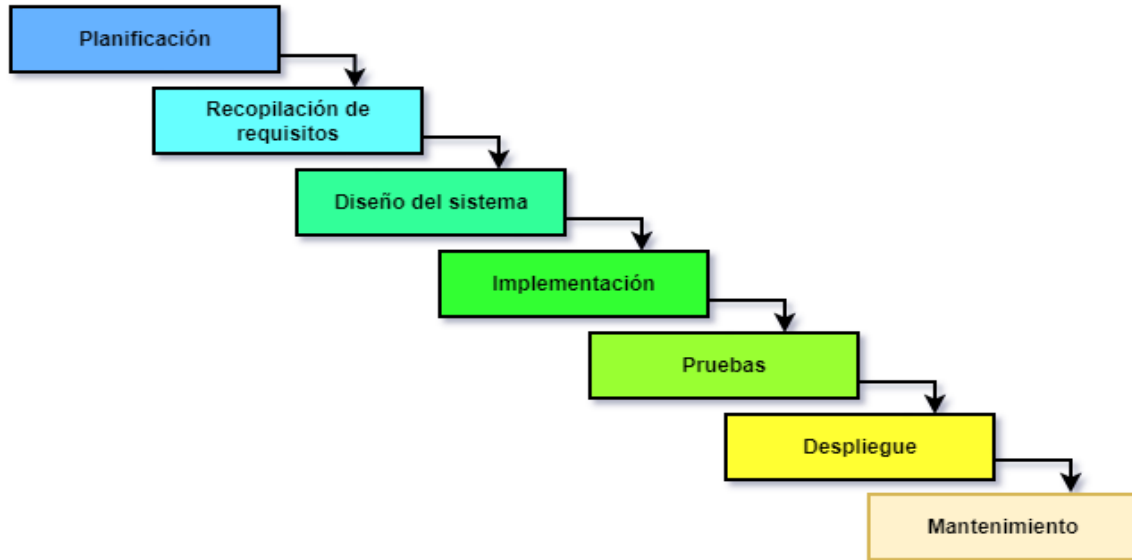
### **7.1. Diseño Metodológico**

Para la ejecución de este estudio se adoptó la metodología de desarrollo en cascada, debido a su enfoque estructurado y secuencial. Este modelo permite avanzar de forma ordenada a través de cada fase del proceso, asegurando que los objetivos definidos se aborden de manera sistemática.

#### **7.1.1. Metodología Cascada**

En la figura 3 se presentan las etapas de la metodología Cascada la cual se aplicó al desarrollo del sistema. Esta metodología permite desarrollar software de manera secuencial, en los cuales se realizaron las fases **Planificación, Recopilación de Requisitos, Diseño del sistema, Implementación, Pruebas, Despliegue y Mantenimiento.**

**Figura 3**  
Etapas de la metodología Cascada



*Nota.* Diagrama basado en las etapas metodológicas del modelo de cascada propuestas por Kiruthika, 2024. Fuente: Elaboración propia.

## **7.2. Resultados**

### **7.2.1. Planificación.**

La aplicación de la primera etapa metodológica del modelo en cascada, correspondiente a la planificación y recopilación de requisitos, permitió establecer una base sólida para el desarrollo del sistema. Esta subetapa se ejecutó a través de un conjunto de actividades dirigidas a estructurar el proyecto y definir sus requerimientos con precisión.

#### **7.2.1.1. Cronograma y planificación inicial**

Se diseñó un cronograma de trabajo detallado utilizando un diagrama de Gantt, el cual organizó las tareas, definió tiempos de ejecución y permitió una correcta asignación de recursos. Este documento facilitó la visualización del progreso del proyecto y la adecuada gestión de sus etapas (ver Anexo J).

### **7.2.2. Recopilación de requisitos**

#### **7.2.2.1. Técnicas utilizadas para la recopilación de requisitos**

Con el objetivo de identificar las necesidades del sistema a desarrollar, se aplicaron diversas técnicas de levantamiento de información, entre ellas:

- Entrevistas con Coordinadores de áreas y Analistas.
- Reuniones con partes interesadas (Coordinadores y Analistas) clave del proyecto.
- Revisión de documentación técnica y operativa.

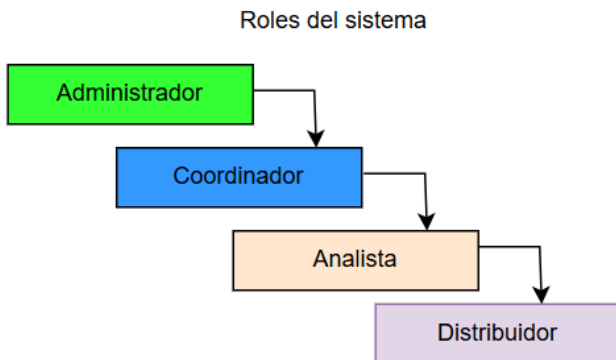
#### **7.2.2.2. Diagramas de Casos de Uso**

Los casos de uso describen las distintas funciones que el sistema realizará a través de la interacción con cada tipo de usuario. Su especificación es fundamental, ya que permite definir claramente la participación de los usuarios en cada proceso y asegura que se cubran todas las necesidades funcionales del sistema.

**Tabla 3**  
Roles del sistema

Actor	Descripción
Administrador	El Administrador es responsable de garantizar la seguridad y el acceso al sistema. Gestiona usuarios, configura el sistema y controla permisos y restricciones para proteger la información. También supervisa tareas como el restablecimiento de contraseñas y la asignación de roles.
Coordinador	El Coordinador es el encargado de supervisar y gestionar las solicitudes de servicios de telecomunicaciones. Actúa como intermediario entre los usuarios y los analistas, asegurando que las solicitudes se asignen y resuelvan de manera oportuna.
Analista	El Analista es el responsable de resolver las solicitudes asignadas por el Coordinador. Debe brindar soluciones efectivas y oportunas, siguiendo los procedimientos establecidos por la empresa.
Distribuidor	Usuario externo autorizado que representa comercialmente a Claro-Nicaragua. Su función principal es registrar solicitudes en nombre de los clientes, ya sea para activaciones, suspensiones, renovaciones u otros servicios relacionados con telecomunicaciones. Además, puede solicitar apoyo en otros temas vinculados a los servicios provistos por la empresa.

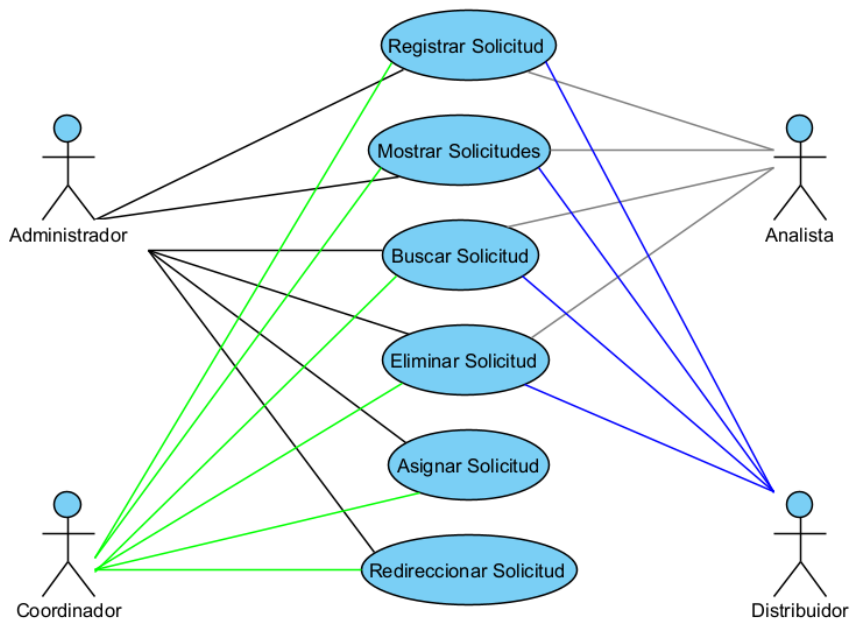
**Figura 4**  
Roles del sistema



**Nota.** Estructura de perfiles de usuarios. Fuente: Elaboración propia, 2025.

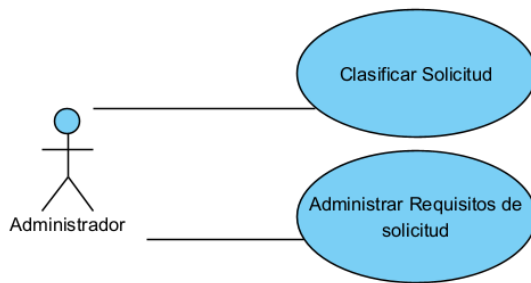
Una vez especificadas las responsabilidades de cada usuario, es fundamental representarlas mediante diagramas UML. Estos diagramas permiten visualizar de manera clara el comportamiento del sistema y su interacción con los diferentes tipos de usuarios, facilitando la comprensión de los procesos y su dinámica operativa.

**Figura 5**  
Caso de uso Solicitudes



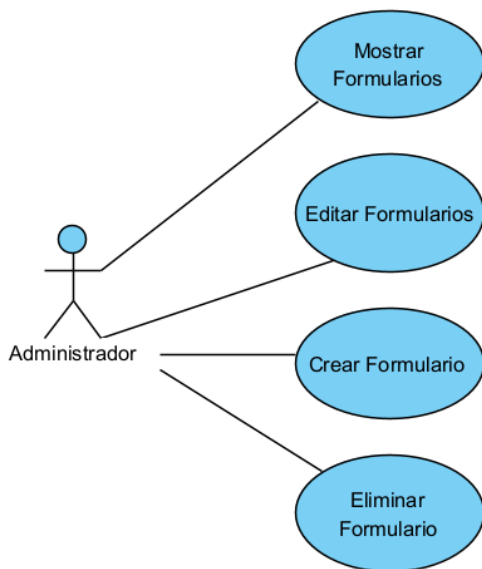
**Nota.** Diagrama UML que detalla las operaciones disponibles y los actores que intervienen en el procesamiento de una solicitud. Fuente: Elaboración propia, 2025.

**Figura 6**  
Caso de uso Gestión de Solicitudes



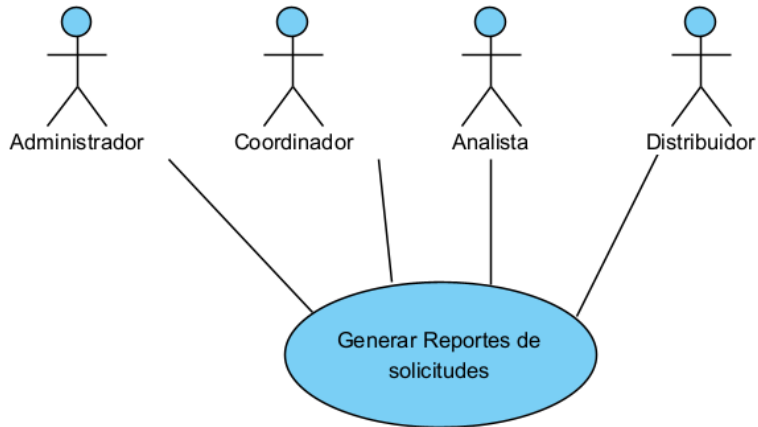
**Nota.** Diagrama UML que detalla las operaciones de clasificación de solicitudes y requisitos asociados. Fuente: Elaboración propia, 2025.

**Figura 7**  
Caso de uso Formularios



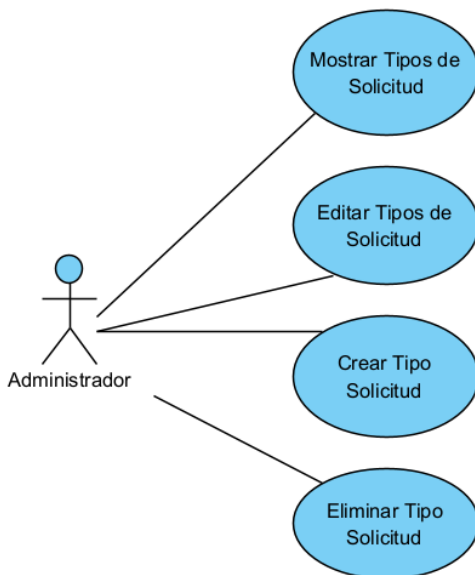
**Nota.** Diagrama UML con las funcionalidades que permiten la definición y estructuración personalizada de los formularios del sistema. Fuente: Elaboración propia, 2025.

**Figura 8**  
Caso de Uso Reportes



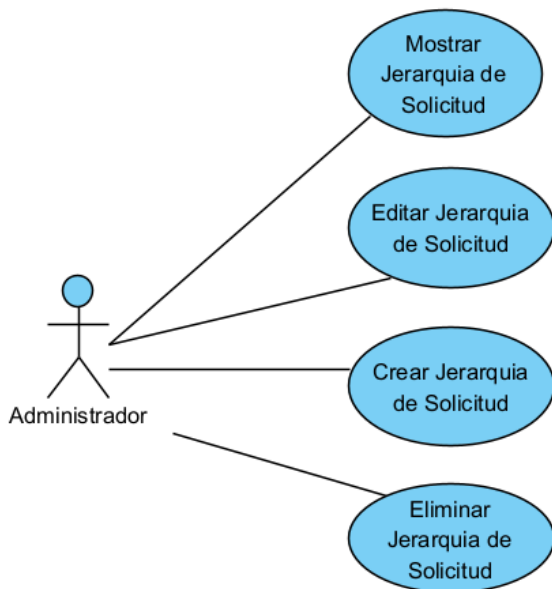
**Nota.** Diagrama UML que detalla las operaciones de visualización y emisión de reportes para los diferentes roles del sistema. Fuente: Elaboración propia, 2025.

**Figura 9**  
Caso de uso Tipos de Solicitud



**Nota.** El diagrama representa la interacción entre los tipos de solicitudes. Fuente: Elaboración propia, 2025.

**Figura 10**  
Caso de uso Jerarquía de Solicitud



**Nota.** Elaboración propia, 2025.

#### 7.2.2.2.1. Explicación de los casos de uso

Anteriormente se definieron los flujos principales con respecto a las funcionalidades que brindara el sistema. A continuación, se presenta la plantilla de caso de usos utilizada para el detalle y especificación de estos.

**Tabla 4**  
Explicación de los casos de uso

<b>Identificador:</b>	Identificador único para cada caso
<b>Nombre:</b>	Nombre completo del caso de uso
<b>Actor(es):</b>	Especificación del (los) usuario (s) que interactúa con el caso de uso
<b>Descripción:</b> Establece las condiciones que deben cumplirse para la ejecución del caso de uso	
<b>Precondiciones:</b> Establece las condiciones que deben cumplirse para la ejecución del caso de uso	
<b>Flujo Principal:</b> Listado de las actividades necesarias para el correcto desarrollo del caso de uso	

<b>Postcondiciones:</b> Establece el estado del sistema cuando el caso de uso se realiza con éxito
<b>Excepciones:</b> Describe funcionalidades inesperadas o excepcionales de un paso a causa de una condición no cumplida
<b>Flujo Alternativo:</b> Describe excepciones o desviaciones del flujo básico

#### 7.2.2.2.2. Detalle de Casos de uso

**Tabla 5**

Caso de uso Registrar Solicitud

<b>Identificador:</b>	CU-01
<b>Nombre:</b>	Registrar Solicitud
<b>Actor(es):</b>	Administrador, Coordinador, Analista, Distribuidor
<b>Descripción:</b>	
Permite a un usuario registrar una solicitud de servicio de telecomunicaciones.	
<b>Precondiciones:</b>	
El usuario debe haber iniciado sesión en el sistema y tener permisos para registrar solicitudes.	
<b>Flujo Principal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede al módulo de solicitudes.</li> <li>2. Selecciona la opción "Nueva Solicitud".</li> <li>3. Completa los campos obligatorios: Canal de venta, Tipo de servicio, Área, Tipo de solicitud, Categoría, Subcategoría, y Detalle del asunto.</li> <li>4. Selecciona los requerimientos específicos (solo si aplica, según el canal de venta: Pyme, Grandes Empresas o Gobierno).</li> <li>5. Establece el estado inicial, prioridad y agrega comentarios.</li> <li>6. Adjunta archivos si es necesario y completa formularios adicionales si la solicitud lo requiere.</li> <li>7. Confirma y guarda la solicitud.</li> </ol>	

<p>8. El sistema genera un identificador único y almacena la información.</p> <p>9. El sistema notifica al usuario que el registro fue exitoso.</p>
<p><b>Postcondiciones:</b></p> <p>La solicitud queda registrada y disponible para su visualización y procesamiento posterior.</p>
<p><b>Excepciones:</b></p> <ul style="list-style-type: none"> <li>- Falla en la conexión con la base de datos.</li> <li>- Campos obligatorios incompletos.</li> <li>- Formato incorrecto o tamaño excedido en archivos adjuntos.</li> <li>- Error interno del sistema.</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>- Si el usuario intenta guardar sin completar todos los campos requeridos, el sistema mostrará un mensaje de validación y resaltará los campos pendientes a completar.</li> <li>- Si se pierde la conexión durante el proceso de guardado, se solicitará al usuario que intente nuevamente.</li> <li>- Si un archivo adjunto supera el límite permitido, se notificará al usuario del tamaño máximo autorizado.</li> </ul>

**Tabla 6**  
Caso de uso Mostrar Solicitudes

<b>Identificador:</b>	CU-02
<b>Nombre:</b>	Mostrar Solicitudes
<b>Actor(es):</b>	Administrador, Coordinador, Analista, Distribuidor
<b>Descripción:</b>	Muestra todas las solicitudes registradas por el usuario o asignadas a un área.
<b>Precondiciones:</b>	El usuario debe haber iniciado sesión en el sistema y tener permisos para ver solicitudes.
<b>Flujo Principal:</b>	1. El usuario accede al módulo de solicitudes.

2. El sistema carga y presenta un listado de solicitudes disponibles según el perfil y área del usuario.
<b>Postcondiciones:</b> Las solicitudes quedan visibles para el usuario, quien puede consultarlas, filtrarlas o realizar otras acciones según su rol.
<b>Excepciones:</b> - Falla en la conexión a la base de datos. - Error en la consulta. - No existen solicitudes disponibles para el usuario.
<b>Flujo Alternativo:</b> - Si no hay solicitudes disponibles, el sistema muestra el mensaje: “No se encontraron solicitudes”.

**Tabla 7**  
Caso de uso Clasificar Solicitud

<b>Identificador:</b>	CU-05
<b>Nombre:</b>	Clasificar solicitud
<b>Actor(es):</b>	Administrador
<b>Descripción:</b> Permite organizar y estructurar las solicitudes según su Área, Categoría, Subcategoría y Tipo de Solicitud.	
<b>Precondiciones:</b> - El usuario debe haber iniciado sesión en el sistema. - Debe contar con permisos de administrador para gestionar clasificaciones.	
<b>Flujo Principal:</b> 1. El usuario accede al módulo de “Configuración” y selecciona la opción “Área”. 2. El sistema muestra la lista de áreas con sus clasificaciones asociadas. 3. El usuario selecciona un área específica y hace clic en la opción “Editar”. 4. El sistema despliega una vista con las categorías y subcategorías asociadas. 5. Se habilitan las opciones: “Eliminar Subcategorías”, “Agregar Subcategorías”, “Agregar Categorías” y “Agregar Tipo de Solicitud”.	

<p>6. Al seleccionar una opción de agregado, el sistema muestra un campo para ingresar el nombre correspondiente.</p> <p>7. Si el campo está vacío y se intenta guardar, el sistema muestra el mensaje: “Campo requerido”.</p> <p>8. Si los datos son válidos, el sistema registra la nueva clasificación y muestra el mensaje: “Registro realizado correctamente”.</p>
<p><b>Postcondiciones:</b></p> <p>La solicitud queda estructurada y clasificada correctamente bajo los parámetros definidos (Área, Categoría, Subcategoría, Tipo de Solicitud).</p>
<p><b>Excepciones:</b></p> <ul style="list-style-type: none"> <li>- Error en la conexión o procesamiento de datos.</li> <li>- Intento de clasificar una solicitud ya clasificada.</li> <li>- Falta de permisos para acceder a esta funcionalidad.</li> </ul>
<p><b>Flujo Alternativo:</b></p> <ul style="list-style-type: none"> <li>- Si se intenta ingresar una clasificación que ya existe, el sistema muestra el mensaje: “Clasificación ya existe” e impide la duplicación.</li> </ul>

**Tabla 8**  
Caso de uso Administrar Requisitos de Solicitudes

<b>Identificador:</b>	CU-06
<b>Nombre:</b>	Administrar Requisitos de Solicitudes
<b>Actor(es):</b>	Administrador
<p><b>Descripción:</b></p> <p>Permite al Administrador gestionar los requerimientos específicos asociados a una solicitud, cuando esta lo requiera. Las operaciones disponibles incluyen crear, editar y eliminar requerimientos según sea necesario para el proceso de atención.</p>	
<p><b>Precondiciones:</b></p> <ul style="list-style-type: none"> <li>- El usuario debe haber iniciado sesión en el sistema.</li> <li>- La solicitud debe admitir requerimientos adicionales.</li> <li>- El usuario debe contar con los permisos necesarios para gestionar esta funcionalidad.</li> </ul>	
<p><b>Flujo Principal:</b></p>	

1. El usuario accede al módulo de “Configuración” y selecciona la opción “Requisitos”.
2. El sistema muestra una lista con los requerimientos existentes. Si no hay registros, muestra el mensaje “Sin requerimientos a mostrar”.
3. El usuario selecciona la opción “Nuevo”.
4. El sistema muestra un formulario con los campos: Nombre, Descripción, Tipo de Requisito y un checkbox “Activo”.
5. El usuario completa los campos y selecciona “Guardar”.
6. Si algún campo obligatorio está vacío, el sistema muestra el mensaje “Campo requerido”.
7. Si todo es válido, el sistema guarda el nuevo requerimiento y muestra el mensaje “Requisito registrado correctamente”.
8. Para modificar un requerimiento existente, el usuario selecciona la opción “Editar” correspondiente.
9. El sistema muestra los campos del requerimiento con su información actual.
10. El usuario realiza los cambios necesarios y selecciona “Actualizar”.
11. Si hay errores en los campos obligatorios, el sistema notifica al usuario.
12. Si todo es correcto, el sistema actualiza la información y muestra el mensaje “Requisito actualizado correctamente”.
13. Para eliminar un requerimiento, el usuario selecciona la opción “Eliminar”.
14. El sistema solicita confirmación y, si el usuario acepta, elimina el requerimiento; si cancela, muestra el mensaje “Operación cancelada”.

**Postcondiciones:**

Los requerimientos de las solicitudes quedan correctamente configurados y disponibles para su uso en el sistema.

**Excepciones:**

- Error en la conexión a la base de datos.
- Faltan permisos para realizar la acción.
- Intento de registrar un requerimiento ya existente con el mismo nombre.

**Flujo Alternativo:**

- Si se intenta crear un requerimiento con un nombre ya registrado, el sistema muestra el mensaje “El requerimiento ya existe” y no permite duplicados.

- Si se cancela alguna operación, el sistema muestra el mensaje “Operación cancelada” y no realiza cambios.

**Tabla 9**

Caso de uso Redireccionar Solicitud

<b>Identificador:</b>	CU-07
<b>Nombre:</b>	Redireccionar Solicitud
<b>Actor(es):</b>	Administrador, Coordinador, Analista
<b>Descripción:</b>	Permite enviar una solicitud a otra área cuando no corresponde al área original.
<b>Precondiciones:</b>	Solicitud sin atención iniciada
<b>Flujo Principal:</b>	<ol style="list-style-type: none"><li>1. El usuario accede al módulo de “Solicitud”</li><li>2. Sistema muestra lista de solicitudes</li><li>3. Usuario selecciona la opción de “Ver” de la solicitud a redirigir de área.</li><li>4. Sistema muestra una vista con los siguientes campos: Asunto, Prioridad, Categoría, Subcategoría, Tipo de servicio, Estado, Fecha Creado, Requerimientos, Asignaciones (a que usuario fue asignada la solicitud), Chat (Comentarios por parte del usuario solicitante y el usuario que atiende solicitud), Envíos (Formularios que funcionan como ayuda cuando se requiere información adicional de solicitud), Guías, Archivos Adjuntos, Relacionado, Historial de la solicitud, opción Editar.</li><li>5. Usuario selecciona la opción de “Editar”.</li><li>6. Sistema muestra nuevamente las opciones de Canal de venta, Tipo Servicio, Área, Tipo de Solicitud, Categoría, Subcategoría.</li><li>7. Usuario selecciona las opciones correspondientes y guarda nuevamente la solicitud.</li><li>8. Sistema guarda los nuevos datos de la solicitud.</li></ol>
<b>Postcondiciones:</b>	La solicitud pasa a la bandeja de la nueva área seleccionada.
<b>Excepciones:</b>	

Error de asignación, área no valida
<b>Flujo Alternativo:</b> Si no se puede redireccionar, sistema detalla motivos (ya está en atención, área no permite recepción).

**Tabla 10**  
Caso de uso Generar Reportes de Solicitudes

<b>Identificador:</b>	CU-08
<b>Nombre:</b>	Generar Reportes de Solicitudes
<b>Actor(es):</b>	Administrador, Coordinador, Analista, Distribuidor
<b>Descripción:</b> Permite a los usuarios con los roles autorizados acceder al módulo de reportes, aplicar filtros personalizados como estado, flujo de trabajo y otros criterios, y seleccionar periodos de tiempo específicos (mes, año, trimestre o un rango personalizado). A partir de estos criterios, el sistema genera y muestra los reportes correspondientes, ofreciendo además la opción de descargar los resultados en formato Excel.	
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>• El usuario debe tener sesión iniciada en el sistema.</li> <li>• El usuario debe tener permisos para acceder al módulo de reportes.</li> </ul>	
<b>Flujo Principal:</b> <ol style="list-style-type: none"> <li>1. El usuario accede al módulo de “Reportes”.</li> <li>2. El sistema presenta una interfaz que incluye un botón denominado “Mostrar Filtros” y un panel informativo con los siguientes elementos: <ul style="list-style-type: none"> <li>• Total de solicitudes registradas.</li> <li>• Tiempo promedio de resolución.</li> <li>• Tasa de resolución.</li> <li>• Cantidad de solicitudes pendientes.</li> <li>• Gráficas estadísticas segmentadas por área, estado, desempeño, flujos de trabajo, ejecución de modelos y alertas.</li> </ul> </li> <li>3. El usuario hace clic en el botón “Mostrar Filtros”.</li> <li>4. El sistema despliega un panel de filtros dividido en secciones:</li> </ol>	

- Flujos de Trabajo: lista de opciones disponibles mediante checkboxes.
  - Estado: conjunto de estados disponibles representados mediante checkboxes.
  - Otros Filtros: incluye campos adicionales como tipo de solicitud, prioridad, responsable, entre otros.
5. El usuario selecciona uno o varios criterios de filtrado en cada sección, según sus necesidades.
6. El usuario elige un periodo de tiempo a través de un menú desplegable, con opciones como:
- Mes
  - Año
  - Trimestre
  - Rango personalizado
7. El usuario hace clic en el botón “Aplicar Filtros”.
8. El sistema procesa los filtros seleccionados, genera los reportes correspondientes, los muestra en pantalla y ofrece la opción de exportar los resultados en formato Excel.

**Postcondiciones:**

El sistema muestra la información filtrada de acuerdo con los criterios seleccionados por el usuario.

**Excepciones:**

- No hay datos que coincidan con los filtros seleccionados, el sistema muestra un mensaje indicando que no se encontraron resultados.
- Error al cargar los datos, el sistema muestra un mensaje de error indicando un problema en la base de datos o en la conexión.

**Flujo Alternativo:**

- Filtros vacíos: Si el usuario no selecciona ningún filtro, el sistema muestra todos los reportes disponibles por defecto.
- Cambio de criterio de filtrado: El usuario puede modificar los filtros seleccionados y volver a ejecutar la búsqueda.

**Tabla 11**

Caso de uso Asignar solicitud

<b>Identificador:</b>	CU-9
<b>Nombre:</b>	Asignar solicitud
<b>Actor(es):</b>	Administrador, Coordinador
<b>Descripción:</b>	Permite que un coordinador asigne una solicitud a un analista.
<b>Precondiciones:</b>	La solicitud debe estar sin "Sin asignar". El usuario debe tener rol de Coordinador o permisos de asignación.
<b>Flujo Principal:</b>	<ol style="list-style-type: none"><li>1. El usuario accede al módulo de "Solicitud"</li><li>2. Sistema muestra lista de solicitudes.</li><li>3. Usuario selecciona la opción de "Asignar Masivamente"</li><li>4. Sistema muestra pantalla con la lista de usuario con el perfil de Analista y las solicitudes sin asignar.</li><li>5. Usuario selecciona usuario al que se le asignara y posterior selecciona la solicitud desde un combo box o un botón de asignar solicitudes, Usuario selecciona opción de "Asignar".</li><li>6. Sistema valida y muestra mensaje de error de campos requeridos.</li><li>7. Sistema guarda la información de las asignaciones.</li></ol>
<b>Postcondiciones:</b>	La solicitud queda registrada con nuevo responsable (usuario asignado) y se cambia el estado a "En Progreso".
<b>Excepciones:</b>	Solicitud ya asignada Error en la base de datos Usuario destino no valido o no disponible
<b>Flujo Alternativo:</b>	Si la solicitud ya está asignada, el sistema lo notifica y bloquea el cambio. Si no hay usuarios disponibles, se recomienda dejar en espera.

**Tabla 12**  
Caso de uso Mostrar Formularios

<b>Identificador:</b>	CU-10
<b>Nombre:</b>	Mostrar Formularios
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	
<p>Permite al usuario acceder al módulo Diseñador de Formularios y visualizar la lista de formularios previamente creados. La interfaz presenta una tabla que resume información clave de cada formulario y proporciona opciones para su gestión (ver, editar o eliminar).</p>	
<b>Precondiciones:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe haber iniciado sesión en el sistema.</li> <li>• El usuario debe contar con los permisos necesarios para acceder al módulo Diseñador de Formularios.</li> </ul>	
<b>Flujo Principal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede al módulo "Diseñador de Formularios".</li> <li>2. El sistema recupera y despliega en pantalla la lista de formularios existentes. En caso de no encontrarse registros disponibles, se muestra el mensaje: "Sin formularios creados"</li> <li>3. La información de los formularios se presenta en una tabla con las siguientes columnas: <ul style="list-style-type: none"> <li>• Nombre del formulario</li> <li>• Descripción</li> <li>• Estado (activo/inactivo)</li> <li>• Número de visitas</li> <li>• Número de envíos</li> <li>• Fecha de creación</li> <li>• Acciones disponibles: Ver, Editar, Eliminar</li> </ul> </li> <li>4. El usuario puede hacer clic en cualquiera de las opciones disponibles para visualizar el detalle del formulario, modificarlo o eliminarlo del sistema, según sus permisos.</li> </ol>	
<b>Postcondiciones:</b>	

- El usuario visualiza correctamente la lista de formularios y puede gestionar cada uno según las acciones permitidas.

**Excepciones:**

- Error de carga: Si ocurre un problema al recuperar la información desde la base de datos, el sistema muestra un mensaje de error.
- Sin formularios disponibles: Si no existen formularios creados, el sistema muestra un mensaje indicando que no hay registros disponibles.

**Flujo Alternativo:**

- Si el usuario no tiene permisos para editar o eliminar formularios, las opciones correspondientes se mostrarán deshabilitadas o no visibles.

El sistema implementa un **módulo de diseñador de formularios** cuya finalidad es permitir la creación dinámica de formularios personalizados. Esta funcionalidad está orientada a brindar flexibilidad al momento de gestionar solicitudes que requieren campos adicionales específicos, sin necesidad de modificar el código fuente del sistema.

A través de este módulo, el usuario puede definir y crear nuevos formularios, seleccionando los campos que necesita de acuerdo con los requerimientos particulares de cada tipo de solicitud. Una vez creado, el formulario puede ser asociado a una solicitud determinada, permitiendo así ampliar su contenido con los datos requeridos en ese contexto específico.

Este enfoque resuelve de manera eficiente la problemática de tener que modificar la estructura del sistema cada vez que surge la necesidad de capturar nueva información en una solicitud. En lugar de implementar nuevos campos en la base de datos o en la interfaz mediante desarrollo adicional, el usuario puede cubrir estos requerimientos desde el módulo de diseño, agilizando los procesos de adaptación y evitando retrabajos en el desarrollo del software.

Cabe destacar que estos formularios no son de uso general para todas las solicitudes, sino que se diseñan y asignan de forma puntual, según la necesidad concreta que se presente.

En concordancia con las buenas prácticas establecidas en ITIL 2011 y siguiendo los lineamientos propuestos por Steinberg, los casos de uso desarrollados para el sistema propuesto reflejan una implementación estructurada y coherente de los principios fundamentales de la gestión de solicitudes, tal como se definió previamente en el marco teórico. Estas prácticas han sido adaptadas al contexto operativo de Claro-Nicaragua, permitiendo que el sistema garantice una administración centralizada.

El sistema contempla el control completo del ciclo de vida de las solicitudes a través de un único módulo, lo que permite registrar, clasificar, actualizar y cerrar solicitudes siguiendo un flujo estandarizado. Para ello, se han definido estados clave como borrador, en revisión, en progreso, cancelado y cerrado, asegurando un seguimiento claro y transparente de cada proceso. Esto se traduce en una mayor eficiencia en la gestión y una reducción en los tiempos de atención.

Asimismo, desde el momento del registro inicial, el sistema permite categorizar las solicitudes según el tipo de canal de venta y requerimiento, respetando el modelo de negocio vigente. Se incorpora también la asignación de prioridades (alta, media o baja), lo cual facilita la atención oportuna de las solicitudes en función de su urgencia y su impacto en la operación. Ambos aspectos contribuyen a una atención más estructurada y alineada con los objetivos estratégicos del negocio.

Los casos de uso incluyen procesos de revisión y asignación, asegurando que cada solicitud sea evaluada antes de ejecutarse, y que se asignen los recursos adecuados para su cumplimiento. Además, se contempla la ejecución de modelos predefinidos para solicitudes recurrentes, lo cual permite estandarizar procedimientos, minimizar errores y acelerar los tiempos de respuesta.

Finalmente, el cierre de solicitudes se realiza mediante la notificación automática al solicitante, informando sobre la finalización del proceso. Esta funcionalidad fortalece la comunicación con el usuario final y garantiza la transparencia del servicio. En conjunto, los casos de uso diseñados cumplen con los principios de ITIL y aportan una solución sólida, adaptada a las necesidades reales de la organización.

### 7.2.2.2.3. Identificación de requisitos del sistema

Para el desarrollo del sistema, se llevó a cabo un proceso de colaboración entre todas las partes involucradas, mediante el cual se identificaron y acordaron los requisitos necesarios para garantizar su correcto funcionamiento y alineación con las necesidades del negocio. A continuación, se presentan los requerimientos acordados:

**Tabla 13**  
Requerimientos funcionales

Identificador	Descripción
RF01	<b>Creación de solicitudes:</b> Permitir a los usuarios crear solicitudes de servicios de telecomunicaciones, ingresando detalles como tipo de solicitud, descripción, área destinada y prioridad.
RF02	<b>Clasificación de solicitudes:</b> Clasificar las solicitudes según el tipo de requerimiento (activaciones, comisiones, soporte técnico, etc.) y el área responsable.
RF03	<b>Asignación de solicitudes:</b> Permitir a los coordinadores asignar solicitudes a los colaboradores correspondientes.
RF04	<b>Cambio de estado de solicitudes:</b> Permitir a los colaboradores cambiar el estado de las solicitudes (por ejemplo, "En Progreso", "Pendiente", "Cerrado").
RF05	<b>Reapertura de solicitudes:</b> Permitir reabrir solicitudes cerradas en caso de que sea necesario realizar ajustes o agregar información adicional.
RF06	<b>Notificaciones automáticas:</b> Enviar notificaciones automáticas por correo electrónico o dentro del sistema cuando: <ul style="list-style-type: none"><li>• Se crea una nueva solicitud.</li><li>• Se asigna una solicitud a un colaborador.</li><li>• Se cambia el estado de una solicitud.</li><li>• Se requiere información adicional.</li></ul>
RF07	<b>Visualización de solicitudes:</b> Permitir a los usuarios, colaboradores y coordinadores ver el estado actual de las solicitudes en tiempo real.

RF08	<b>Historial de solicitudes:</b> Mantener un historial completo de cada solicitud, incluyendo cambios de estado, comentarios y acciones realizadas.
RF09	<b>Tiempos de atención:</b> Registrar y mostrar los tiempos de atención de cada solicitud, desde su creación hasta su cierre.
RF10	<b>Asignación de prioridades:</b> Permitir asignar prioridades a las solicitudes (por ejemplo, baja, media, alta).
RF11	<b>Redireccionamiento de solicitudes:</b> Permitir redirigir solicitudes entre áreas si es necesario.
RF12	<b>Generación de reportes:</b> Permitir generar reportes en formato Excel.

**Tabla 14**  
Requerimientos no funcionales

Identificador	Descripción
RNF01	<b>Facilidad de uso en los procesos:</b> Los procesos del sistema (creación, asignación, seguimiento y cierre de solicitudes) deben ser fáciles de ejecutar, con un flujo de trabajo intuitivo y guiado.
RNF02	<b>Control de acceso basado en roles:</b> El sistema debe implementar un control de acceso basado en roles (RBAC) para garantizar que los usuarios solo puedan acceder a las funcionalidades y datos correspondientes a su perfil.
RNF03	<b>Intuitividad de la interfaz de usuario:</b> La interfaz de usuario debe ser intuitiva, con un diseño claro y consistente que permita a los usuarios navegar y realizar tareas sin necesidad de capacitación extensa.
RNF04	<b>Identidad visual de la empresa:</b> El sistema debe reflejar la identidad visual de Claro-Nicaragua, incluyendo logos, colores corporativos y otros elementos de marca.

### 7.2.3. Diseño del sistema

#### 7.2.3.1. Maquetación del sistema web

Para el diseño visual del sistema web se utilizó la herramienta Justinmind, la cual permitió crear maquetas que representan la estructura y navegación del sistema. Estas maquetas sirvieron como una guía clara para el desarrollo posterior, asegurando que el diseño estuviera alineado con las necesidades del usuario final y facilitando la validación temprana de la interfaz. A continuación, se presenta una muestra de la maqueta creada. Los restantes se incluyen en la sección de Anexos A.

**Figura 11**  
Crear Solicitud













Admin >> Claro >> Solicitudes >> Nuevo

Clasificación	Cumplimiento	Detalles	Adjuntos	Formularios	Resumen
<b>Categoría de Solicitud</b> Seleccione cate... <b>Tipo Servicio</b> Seleccione tipo ... <b>Área</b> Seleccione área <b>Tipo Solicitud</b> Seleccione tipo ... <b>Categoría</b> Seleccione cate... <b>SubCategoría</b> Seleccione sub... Siguiente	<b>Requisitos (0/n)</b> n requisitos en total 0 completados 0 pendientes 0.0% Completado Seleccionar todos Requisito 1 Requisito 2 Requisito 3 Requisito 4 Requisito 5 Requisito 6 Anterior Siguiente	<b>Solicitar información</b> <b>Asunto del problema</b> <b>Prioridad</b> Seleccione prio... <b>Descripción</b> <input checked="" type="checkbox"/> <b>Es Borrador</b> <small>Esta opción le permite guardar esta solicitud sin enviarla al área asignada, para deshabilitar esta opción, debe cumplir con los requisitos requeridos.</small> Anterior Siguiente	Adjuntar archivos Anterior Siguiente	Formularios vinculados Anterior Siguiente	<b>Detalles de solicitud</b> Asunto del problema <b>Asunto</b> Prioridad Prioridad 1 Área Área 1 Categoría Categoría 1 SubCategoría SubCategoría 1 Cumplimineto Total de requisitos N/N Adjuntos Adjunto1.txt Formularios Formulario 1 Anterior Finalizar

**Nota.** Interfaz de usuario para el registro de nuevas solicitudes. Fuente: Elaboración propia, 2025.

**Figura 12**  
Solicitudes Registradas

The screenshot shows a web interface for managing requests. At the top, there is a breadcrumb trail: Admin >> Claro >> Solicitudes. On the right side, there are navigation icons (notifications, globe, star, user) and two buttons: 'Asignar Masivamente' and '+ Nueva'. Below these are three buttons: 'Opciones de vista', 'Filtros', and 'Ordenar'. On the far right, there is an 'Exportar' button. The main content is a table with the following columns: Asunto, Prioridad, Estado, Categoria, Subcategoria, Asignaciones, and Seguimiento de cumplimiento. The table contains six rows of data, all with 'Alta' priority and 'En Proceso' status. At the bottom right, there is a pagination control showing 'Filas por pagina' set to 10 and 'Pagina 1 de 1'.

Asunto	Prioridad	Estado	Categoria	Subcategoria	Asignaciones	Seguimiento de cumplimiento
Solicitud 1	Alta	En Proceso	Internet	Servicio Fijo	0	0  
Solicitud 2	Alta	En Proceso	Internet	Servicio Fijo	0	0  
Solicitud 3	Alta	En Proceso	Internet	Servicio Fijo	0	0  
Solicitud 4	Alta	En Proceso	Internet	Servicio Fijo	0	0  
Solicitud 5	Alta	En Proceso	Internet	Servicio Fijo	0	0  
Solicitud 6	Alta	En Proceso	Internet	Servicio Fijo	0	0  

**Nota.** Módulo “Solicitud” donde se muestran las solicitudes registradas. Elaboración propia, 2025.

**Figura 13**  
Asignar solicitud

The screenshot shows a form titled 'Asignar solicitud'. The form is contained within a larger frame. At the top of the form, it says 'Asignar solicitud 1 a un usuario'. Below this, there is a label 'Area:' followed by the text 'Area 1'. Then, there is a label 'Asignar A:' followed by a dropdown menu showing 'Usuario 1'. Below the dropdown, there is a small instruction: 'Selecciona el usuario al que se asignará la solicitud.' Below that, there is a label 'Comentarios:' followed by a text input field with the placeholder text 'Añade un comentario sobre la asignación'. Below the input field, there is another instruction: 'Agrega un comentario sobre la asignación.' At the bottom right of the form, there is a button labeled 'Asignar'.

**Nota.** Interfaz de asignación de solicitudes. Permite designar a un responsable encargado de la atención, seguimiento y resolución del caso. Elaboración propia, 2025.

**Figura 14**  
Detalle de solicitud

Admin >> Claro >> Solicitudes >> id								
Requerimientos    Modelo de Ejecución    Guías    Submisiones    Adjuntos    Comentarios    Asignaciones    Historial								
<b>Requisitos totales (13/13)</b> <input type="text"/> <input type="button" value="Deseleccionar Todo"/> <b>Seguimiento del progreso</b> 13 Completado 0 Pendiente <input checked="" type="checkbox"/> Requisito 1 <input checked="" type="checkbox"/> Requisito 2 <input checked="" type="checkbox"/> Requisito 3 <input checked="" type="checkbox"/> Requisito 4 . . <input checked="" type="checkbox"/> Requisito 13	<b>Ejecución de Proceso</b> Seguimiento del proceso en tiempo real Progreso total: 0% Paso N de N <input type="text"/> 1 Inicio En Progreso <input type="button" value="Completar"/> 2 paso N Bloqueado Tiempo Estimado 30 minutos Paso Actual Inicio: Inicio Historial de Ejecución Pasos completados: 0 No hay pasos completados aún <input type="button" value="Iniciar"/> <input type="button" value="Reiniciar"/>	Guías de Ejecución de solicitud	Nombre del Formulario Contenido	Archivos Adjuntos	<b>Comentarios</b> User 1: Atender solicitud User 2: solicitud en proceso <input type="text"/> <input type="button" value="Ingresar comentario"/>	<input type="button" value="Reasignar Área"/> <input type="button" value="Asignar Usuario"/> Área Área 1 Usuario Asignado Usuario 1 Fecha Asignación dd / mm / yyyy Categoría Categoría 1 Comentarios Comentario 1	Solicitud Creada dd / mm / yyyy Id Solicitud 1 Fecha Completado dd / mm / yyyy Modificado por user 1 Cambios Cambio 1 Cambio 2 Creado dd / mm / yyyy	<b>Estado de la solicitud</b> Estado actual <input type="button" value="Cambiar"/> <b>En Revisión</b> Prioridad <input type="button" value="Cambiar"/> <b>Alta</b> Progreso Envíos 0/0 <input type="text"/> Requisitos 13/13 <input type="text"/> Modelo de Ejecución 50% <input type="text"/> <b>Detalles</b> Asunto del Problema Asunto Solicitud Descripción Descripción de la solicitud Área <input type="button" value="Reasignar"/> Área 1 Categoría de solicitud Categoría 1 Asignado a <input type="button" value="Reasignar"/> Usuario 1 Solicitante Usuario 2 usuario2@example.com Fechas Creado: dd / mm / yyyy Actualizado: dd / mm / yyyy

**Nota.** Vista de detalle de la solicitud. La pantalla muestra los datos de registro junto con el seguimiento de sus estados, prioridad y la bitácora histórica. Elaboración propia, 2025.

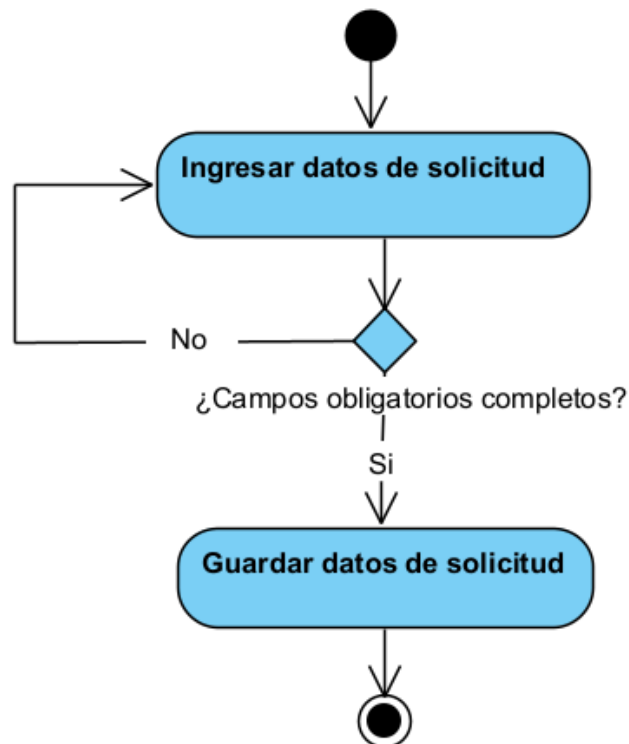
### 7.2.3.2. Diagramas

#### 7.2.3.2.1. Diagramas de Actividades

Los diagramas de actividades son una herramienta visual que permite representar el flujo de procesos dentro de un sistema, facilitando la comprensión de su comportamiento. En esta sección, se describen algunas de las acciones claves modeladas en el sistema.

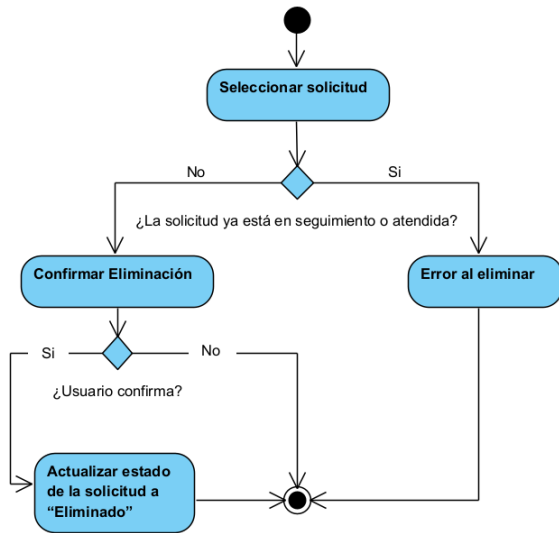
**Figura 15**

Diagrama de actividad Registrar Solicitud



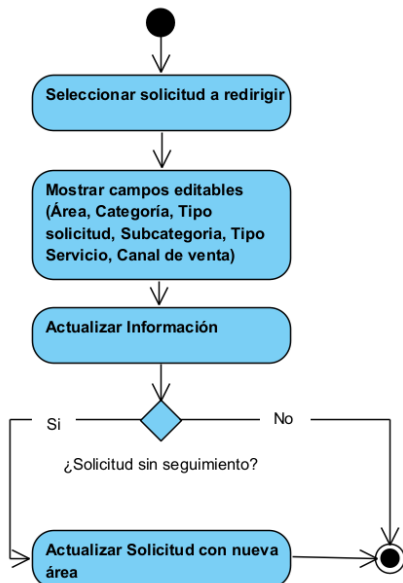
**Nota.** Diagrama de actividad elaborado para representar el proceso "Registrar Solicitud". Fuente: Elaboración propia, 2025.

**Figura 16**  
Diagrama de actividad Eliminar Solicitud



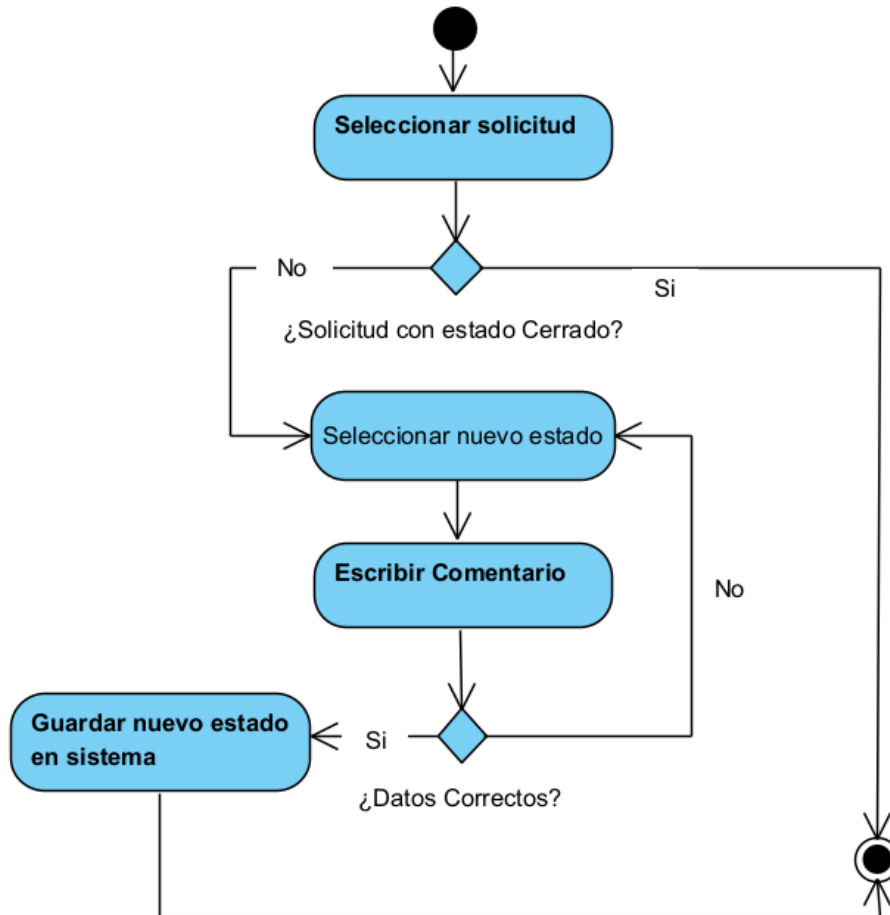
**Nota.** Diagrama de actividad elaborado para representar el proceso "Eliminar Solicitud". Fuente: Elaboración propia, 2025.

**Figura 17**  
Diagrama de actividad Redirigir Solicitud



**Nota.** Diagrama de actividad elaborado para representar el proceso "Redirigir Solicitud". Fuente: Elaboración propia, 2025.

**Figura 18**  
Diagrama de actividad Cambiar Estado Solicitud



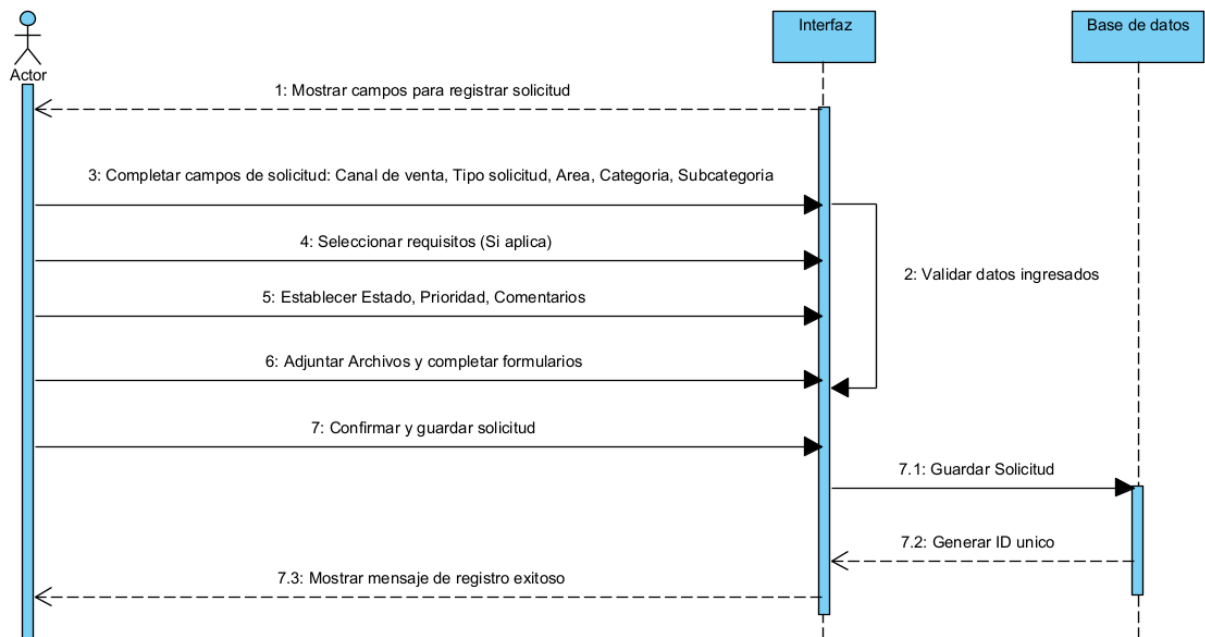
**Nota.** Diagrama de actividad elaborado para representar el proceso “Cambiar Estado Solicitud”. Fuente: Elaboración propia, 2025.

### 7.2.3.3. Diagramas de Secuencias

El Diagrama de Secuencia es una herramienta fundamental para representar visualmente la comunicación entre objetos en un sistema. Permite modelar el flujo de mensajes en un escenario específico, mostrando la interacción entre clases y objetos a lo largo del tiempo. Además, facilita la comprensión del comportamiento del sistema y su implementación.

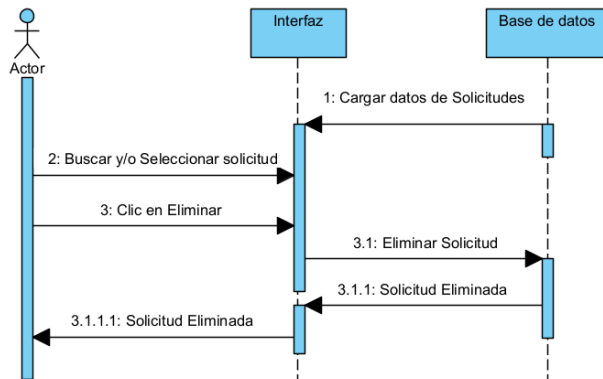
**Figura 19**

Diagrama de secuencia Registrar Solicitud



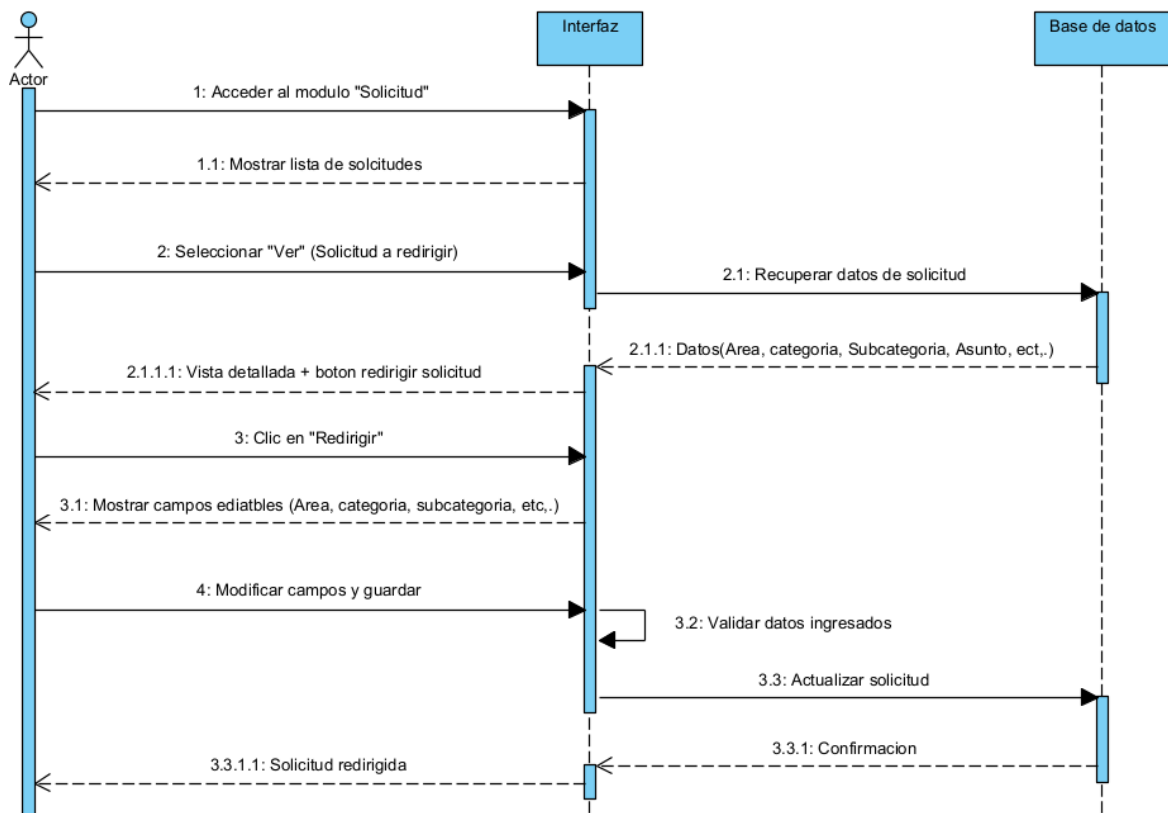
**Nota.** Diagrama de secuencia elaborado para representar el proceso “Registrar Solicitud”. Fuente: Elaboración propia, 2025.

**Figura 20**  
Diagrama de secuencia Eliminar Solicitud



**Nota.** Diagrama de secuencia elaborado para representar el proceso “Eliminar Solicitud”. Fuente: Elaboración propia, 2025.

**Figura 21**  
Diagrama de secuencia Redireccionar Solicitud



**Nota.** Diagrama de secuencia elaborado para representar el proceso “Redireccionar Solicitud”. Fuente: Elaboración propia, 2025.



## 7.2.4. Implementación

En esta fase se desarrollaron los módulos funcionales del sistema, transformando las especificaciones del diseño en código operativo. Se utilizaron herramientas como React y Next.js para la interfaz, Prisma y SQL Server para la gestión de datos, y TRPC, NextAuth y TypeScript para la lógica de negocio y autenticación.

Además, se implementó Git como sistema de control de versiones, lo que permitió organizar el código, registrar avances y facilitar el trabajo colaborativo.

A continuación, se describen las principales etapas del proceso de implementación, con sus respectivos fragmentos de código representativos:

### 7.2.4.1. Configuración del entorno

Figura 23

Configuración del entorno de desarrollo

```
41 "prettier": {
66   "importOrderParserPlugins": [
67     "typescript",
68     "jsx",
69     "decorators-legacy"
70   ],
71   "plugins": [
72     "prettier-plugin-tailwindcss",
73     "@ianvs/prettier-plugin-sort-imports"
74   ],
75   "printWidth": 200,
76   "tailwindStylesheet": "./app/globals.css"
77 },
78 "prisma": {
79   "seed": "tsx prisma/seed.ts"
80 },
81 "dependencies": {
82   "@ably/chat": "^1.1.1",
83   "@base-ui-components/react": "1.0.0-rc.0",
84   "@better-auth/sso": "1.4.7",
85   "@better-fetch/fetch": "^1.1.21",
86   "@date-fns/utc": "^2.1.1",
87   "@dnd-kit/core": "^6.3.1",
88   "@dnd-kit/modifiers": "^9.0.0",
89   "@dnd-kit/sortable": "^10.0.0",
90   "@dnd-kit/utilities": "^3.2.2",
91   "@emoji-mart/data": "^1.2.1",
92   "@floating-ui/react-dom": "^2.1.6",
93   "@google/generative-ai": "^0.24.1",
94   "@headless-tree/core": "^1.6.1",
95   "@headless-tree/react": "^1.6.1",
96   "@hookform/error-message": "^2.0.1",
97   "@hookform/resolvers": "^5.2.2",
98   "@lexical/react": "^0.39.0",
99   "@ngard/tiny-isequal": "^1.1.0",
100  "@polar-sh/checkout": "^0.1.14",
101  "@polar-sh/nextjs": "^0.9.3",
102  "@polar-sh/sdk": "^0.42.1",
103  "@prisma/client": "6.19.0",
104  "@radix-ui/react-accordion": "^1.2.12",
105  "@radix-ui/react-alert-dialog": "^1.1.15",
106  "@radix-ui/react-aspect-ratio": "^1.1.8",
223 },
224 "devDependencies": {
225   "@4tw/cypress-drag-drop": "^2.3.1",
226   "@eslint-community/eslint-plugin-eslint-comments": "^4.5.0",
227   "@eslint-react/eslint-plugin": "^2.3.13",
228   "@eslint/compat": "^2.0.0",
229   "@eslint/eslintrc": "^3.3.3",
230   "@eslint/js": "^9.39.2",
231   "@faker-js/faker": "^10.1.0",
232   "@hookform/devtools": "^4.4.0",
233   "@ianvs/prettier-plugin-sort-imports": "^4.7.0",
234   "@jest/globals": "^30.2.0",
235   "@lingual/i18n-check": "^0.8.16",
236   "@next/eslint-plugin-next": "^16.1.0",
237   "@playwright/test": "^1.57.0",
238   "@svgr/webpack": "^8.1.0",
239   "@tailwindcss/postcss": "^4.1.18",
240   "@tanstack/eslint-plugin-query": "^5.91.2",
241   "@tanstack/react-query-devtools": "^5.91.1",
242   "@testing-library/dom": "^10.4.1",
243   "@testing-library/jest-dom": "^6.9.1",
244   "@testing-library/react": "^16.3.1",
245   "@testing-library/user-event": "^14.6.1",
246   "@types/eslint": "^9.6.1",
247   "@types/jest": "^30.0.0",
248   "@types/lodash": "^4.17.21",
249   "@types/node": "^25.0.3",
250   "@types/react": "^19.2.7",
251   "@types/react-dom": "^19.2.3",
252   "@typescript-eslint/eslint-plugin": "^8.50.0",
253   "@typescript-eslint/parser": "^8.50.0",
254   "@zenstackhq/tanstack-query": "^2.22.1",
255   "autoprefixer": "^10.4.23",
256   "chokidar-cli": "^3.0.0",
257   "concurrently": "^9.2.1",
258   "cross-env": "^10.1.0",
259   "cypress": "^15.8.1",
260   "eslint": "^9.39.2",
261   "eslint-config-next": "^16.1.0",
262   "eslint-config-prettier": "^10.1.0"
}
```

**Nota.** Arquitectura y configuración del entorno de desarrollo basado en Prisma, TypeScript y React, entre otras tecnologías. Fuente: Elaboración propia (2025).

La figura muestra la configuración esencial del entorno de desarrollo establecido para el proyecto. Se definieron las herramientas principales como React, Next.js, Prisma y TypeScript. Además, se implementaron reglas de formateo de código para asegurar la coherencia en todo el equipo. La conexión a la base de datos SQL Server se gestiona de forma segura mediante variables de entorno.

#### 7.2.4.1.1. Modelado de datos con prisma.

**Figura 24**

Modelado de datos con prisma

```

  ▾ datasource db {
    | provider = "sqlserver"
  }

  Generate
  ▾ generator client {
    | provider = "prisma-client-js"
    | output  = "../node_modules/.prisma/client"
  }

  ▾ model User {
    | id          String          @id()
    | name       String
    | email      String
    | emailVerified Boolean
    | image      String?
    | createdAt  DateTime
    | updatedAt  DateTime
    | twoFactorEnabled Boolean?
    | role       String?
    | banned     Boolean?
    | banReason  String?
    | banExpires DateTime?
    | phoneNumber String?
    | phoneNumberVerified Boolean?
    | isAnonymous Boolean?
    | isGlobalAdmin Boolean      @default(false)
    | username   String?
    | displayName String?
    | sessions   Session[]
    | accounts   Account[]
    | twofactors TwoFactor[]
    | ssoproviders SsoProvider[]
    | apikeys    Apikey[]
    | passkeys   Passkey[]

    @@unique([email])
    @@unique([email, username])
    @@map("user")
  }

```

**Nota.** Modelado del esquema de la base de datos con prisma ORM. Fuente: Elaboración propia (2025).

Se presenta el esquema de la entidad *Usuario (User)* utilizando el ORM Prisma. Se define la estructura de la tabla, los tipos de datos de cada atributo y sus relaciones con otras entidades. Este modelo es fundamental para la gestión de usuarios dentro del sistema. Sirve como base para la implementación de funciones relacionadas con perfiles, autenticación y control de acceso.

#### 7.2.4.2. Implementación de autenticación

**Figura 25**

Configuración de Autenticación con NextAuth

```
19 | database: prismaAdapter(db, {
20 |   provider: 'sqlserver',
21 | }),
22 | emailAndPassword: {
23 |   enabled: true,
24 |   password: {
25 |     hash: hashPassword,
26 |     verify: comparePassword,
27 |   },
28 |   sendResetPassword,
29 | },
30 | emailVerification: {
31 |   sendOnSignUp: true,
32 |   autoSignInAfterVerification: true,
33 |   sendVerificationEmail,
34 | },
35 | user: {
36 |   changeEmail: {
37 |     enabled: true,
38 |     sendChangeEmailVerification,
39 |   },
40 |   additionalFields: {
41 |     isGlobalAdmin: {
42 |       input: false,
43 |       type: 'boolean',
44 |       required: false,
45 |       defaultValue: false,
46 |     },
47 |   },
48 | },
49 | plugins: [
50 |   twoFactor({
51 |     issuer: process.env.APP_NAME || 'Requestum',
52 |     otpOptions: {
53 |       async sendOTP({ user, otp }) {
```

**Nota.** Configuración del servicio de autenticación mediante la librería NextAuth.js, detallando la integración de proveedores de identidad (OAuth) y la definición de credenciales de acceso. Fuente: Elaboración propia (2025).

Esta figura evidencia la configuración de la autenticación utilizando NextAuth.js. El código implementa el flujo de inicio de sesión con medidas de seguridad como la verificación de correo electrónico y la autenticación en dos factores (2FA). Se establecen las credenciales necesarias y las opciones de proveedor. Esta lógica refuerza la protección de las cuentas de usuario. Así, se garantiza un acceso seguro y confiable al sistema.

### 7.2.4.3. Código frontend e interfaz

#### Figura 26

Código Frontend con React (Detalle de solicitud).

```
<TabsContent value="requirements" className="flex-1 flex flex-col overflow-hidden">
  <RequirementProgress tenantId={tenantId} requestId={slug} />
</TabsContent>

<TabsContent value="executionModel" className="flex-1 flex flex-col overflow-hidden">
  <ExecutionView
    processFlow={requestDetails.executionFlow?.diagram}
    tenantId={tenantId}
    executionId={requestDetails.executionFlow?.executionId ?? ''}
    locale={locale}
    executionLogs={requestDetails.executionFlow?.logs}
  />
</TabsContent>

<TabsContent value="guides" className="flex-1 flex flex-col overflow-hidden">
  <GuidesViewer guides={requestDetails.guides} />
</TabsContent>

<TabsContent value="submissions" className="flex-1 flex flex-col overflow-hidden">
  <FormSubmissionsViewer tenantId={tenantId} requestId={slug} />
</TabsContent>

<TabsContent value="attachments" className="flex-1 flex flex-col overflow-hidden">
  {requestDetails.dataroom?.id ? (
    <Card className="flex-1 flex flex-col overflow-hidden">
      <DataroomDocuments dataroomId={requestDetails.dataroom?.id} tenantId={tenantId} callbackUrl={callbackUrl} />
    </Card>
  ) : (
    <div className="flex-1 flex items-center justify-center">
      <EmptyState title={t(No documents found)} description={t(Please check back later or contact suppo...)} />
    </div>
  )}
</TabsContent>

<TabsContent value="comments" className="flex-1 flex flex-col overflow-hidden">
  <Messages channel={requestDetails.channel} tenantId={tenantId} />
</TabsContent>
```

**Nota.** Fragmento de código que detalla la implementación de componentes funcionales en React. Se observa la lógica de programación para el sistema de pestañas y la estructura modular del proyecto. Fuente: Elaboración propia (2025).

Se evidencia el desarrollo de una interfaz modular construida con componentes funcionales de React. La implementación utiliza una estructura de pestañas que permite organizar la información de manera clara y dinámica. Este enfoque facilita la reutilización del código y mejora la experiencia del usuario. Además, se promueve la escalabilidad del frontend. La estructura modular contribuye a un mantenimiento más eficiente del sistema.

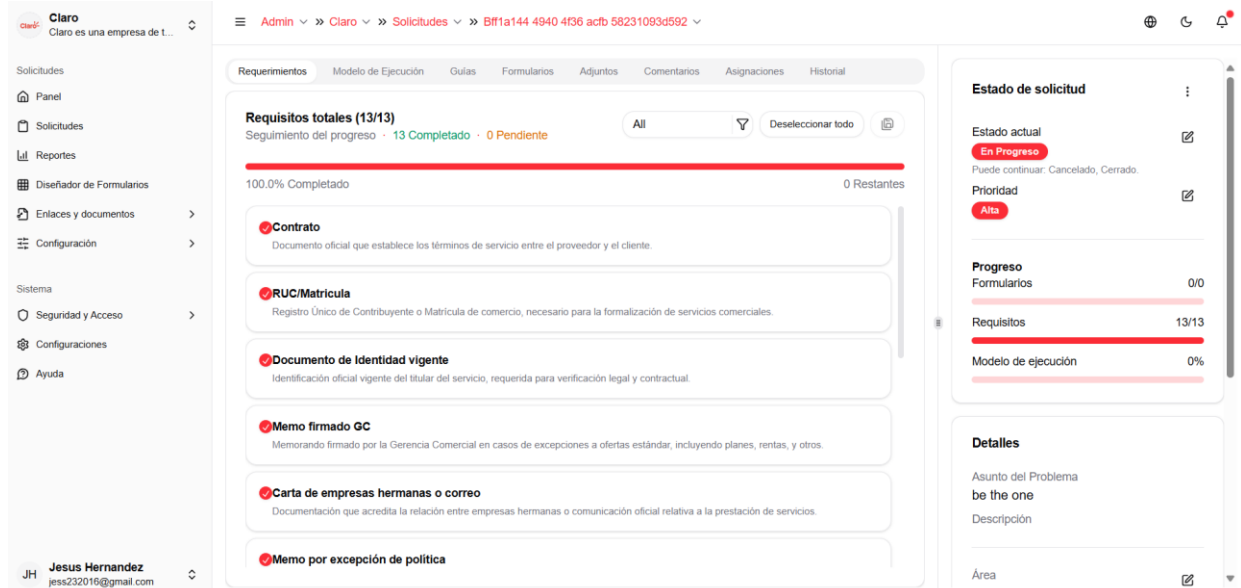
**Figura 27**  
Interfaz Listado de solicitudes

Asunto	Prioridad	Estado	Categoría de Solicitud	Categoría de Asignación	Documentos	Asignaciones de Solicitudes	Seguimientos de Cumplimiento
Reactivacion	Alta	En revisión	Internet Gpon	Reactivacion Casa Claro	1	1	13

**Nota.** Vista principal del módulo "Solicitudes" implementado en el sistema. Fuente: *Elaboración propia (2025).*

Se ilustra la interfaz principal de gestión del sistema. La vista tabular centraliza la información y permite su ordenamiento y filtrado, mientras que los selectores de vista (Tabla, Kanban) ofrecen flujos de trabajo flexibles para la administración de las tareas.

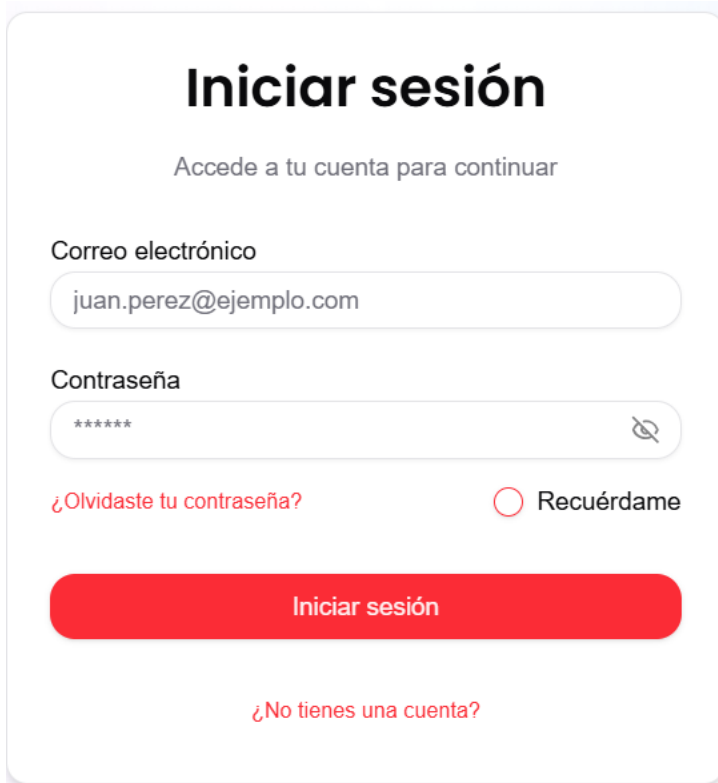
**Figura 28**  
Interfaz detalle de solicitud



**Nota.** Visualización de datos en el módulo "Detalle de solicitud". Fuente: Elaboración propia (2025).

La imagen presenta una vista detallada del seguimiento de los requisitos asociados a una solicitud. Se incorpora un resumen ejecutivo que informa sobre el estado actual y la prioridad de cada ítem. Esta representación permite al usuario una evaluación rápida, clara y precisa. El diseño facilita la toma de decisiones oportunas en función del avance.

**Figura 29**  
Página de inicio de sesión



The image shows a login form with the following elements:

- Title:** "Iniciar sesión" in large, bold black font.
- Subtitle:** "Accede a tu cuenta para continuar" in a smaller, gray font.
- Form Fields:**
  - Correo electrónico:** A text input field containing "juan.perez@ejemplo.com".
  - Contraseña:** A password input field with "\*\*\*\*\*" and a toggle icon on the right.
- Links:**
  - [¿Olvidaste tu contraseña?](#) in red text.
  - Recuérdame** with an unchecked radio button.
- Buttons:**
  - A large red button with the text "Iniciar sesión" in white.
  - A red link at the bottom: [¿No tienes una cuenta?](#)

**Nota.** *Interfaz de inicio de sesión. Fuente: Elaboración propia (2025).*

Se muestra la interfaz de acceso al sistema, diseñada como el punto de entrada seguro para los usuarios. Esta pantalla actúa como el primer nivel de control, validando las credenciales antes de permitir el ingreso. Su diseño integra las configuraciones de autenticación implementadas con NextAuth. A través de ella, se restringe el acceso a funcionalidades protegidas. Representa un componente clave en la seguridad de la aplicación.

# Figura 30

## Dashboard general

### Dashboard

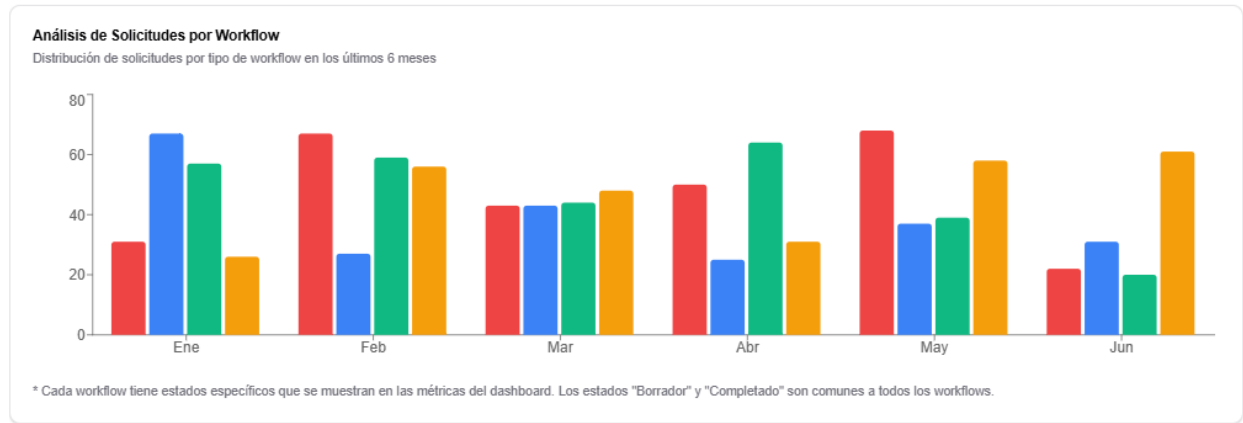
Bienvenido al sistema de gestión de solicitudes de Claro-Nicaragua

[Nueva Solicitud](#)

General Flujo de Trabajo SLAs

#### Métricas Generales

<p>Total de Solicitudes</p> <p><b>1,248</b></p> <p>↑ 12.5% vs. mes anterior</p>	<p>Tiempo Promedio de Resolución</p> <p><b>2.4 días</b></p> <p>↑ 8.1% más rápido</p>	<p>Tasa de Resolución</p> <p><b>94.2%</b></p> <p>↑ 2.1% vs. mes anterior</p>	<p>Solicitudes Pendientes</p> <p><b>72</b></p> <p>↓ 5.3% vs. mes anterior</p>
---	--	--	---



<p><b>Solicitudes Pendientes</b></p> <p>Solicitudes que requieren atención</p> <p><b>12</b></p> <p>4 con prioridad alta</p> <p><a href="#">Ver todas</a> →</p>	<p><b>Solicitudes Completadas</b></p> <p>Solicitudes resueltas este mes</p> <p><b>48</b></p> <p>Tiempo promedio: 2.3 días</p> <p><a href="#">Ver todas</a> →</p>	<p><b>Reportes</b></p> <p>Análisis de desempeño</p> <p></p> <p>Accede a reportes detallados</p> <p><a href="#">Ver reportes</a> →</p>
--	--	---

#### Solicitudes Recientes

ID	Título	Solicitante	Departamento	Workflow	Estado	Prioridad	Fecha	Acciones
REQ-2023-001	Activación de servicio móvil	Juan Pérez	Activaciones	Activaciones	En progreso	Alta	2023-06-15	Ver
REQ-2023-002	Reclamo de comisión no pagada	María López	Comisiones	Comisiones	En revisión	Media	2023-06-14	Ver
REQ-2023-003	Soporte técnico para router	Carlos Ruiz	Soporte	Soporte Técnico	Cerrado	Baja	2023-06-13	Ver
REQ-2023-004	Activación de plan corporativo	Empresa XYZ	Activaciones	Activaciones	Borrador	Alta	2023-06-12	Ver
REQ-2023-005	Ajuste de facturación	Ana Martínez	Facturación	Facturación	Cancelado	Media	2023-06-11	Ver
REQ-2023-006	Cambio de plan empresarial	Tech Corp	Activaciones	Cambios de Plan	Aprobando cambio	Alta	2023-06-10	Ver
REQ-2023-007	Reclamo por facturación incorrecta	Luis García	Reclamos	Reclamos	Investigando	Media	2023-06-09	Ver

**Nota.** Interfaz principal del Dashboard. Fuente: Elaboración propia (2025).

El dashboard muestra de manera visual y centralizada los indicadores clave del sistema de solicitudes. Incluye el total de solicitudes registradas y pendientes, el tiempo promedio de resolución y la tasa de resolución, permitiendo comparar la evolución mensual.

**Figura 31**  
Crear formularios

**Crear nuevo formulario** ✕

Crea un nuevo formulario para comenzar a recopilar respuestas

Nombre

Formulario Clientes

Este es el nombre público de tu formulario

Descripción

Formulario de clientes

Agrega detalles adicionales o una descripción aquí

**Formulario público**  
Haz que este formulario sea público para que cualquier persona pueda verlo y enviar respuestas

**Guardar**

**Nota.** Interfaz de creación de formularios. Fuente: Elaboración propia (2025).

**Figura 32**  
Diseño de formulario

Formulario: Formulario Clientes Vista previa Guardar Publicar

Suelta aquí

Arrastra y suelta elementos

Elementos del formulario

- Text Field
- Number Field
- TextArea Field
- Date Field
- Select Field
- CheckBox Field

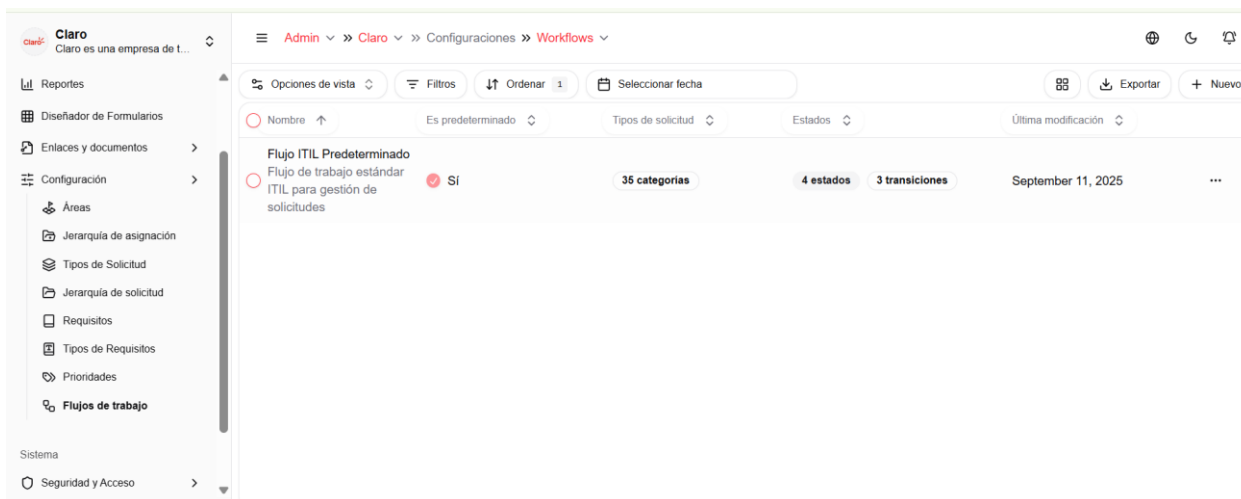
Elementos de diseño

- Title field
- SubTitle field
- Paragraph field

**Nota.** Interfaz de diseño de formularios. Fuente: Elaboración propia (2025).

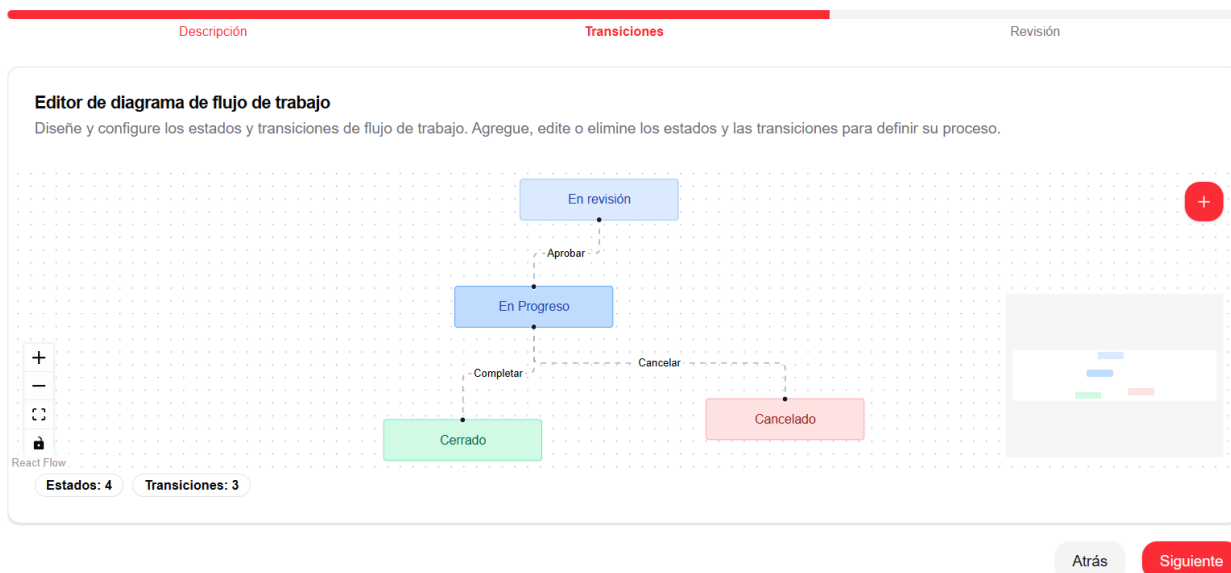
La implementación de formularios (figuras 27 y 28) constituye un recurso fundamental para la recolección de información complementaria durante el proceso de generación de solicitudes. En aquellos casos en que los datos requeridos no se encuentren contemplados en el formulario predeterminado, se posibilita el diseño de formularios específicos que pueden ser asociados a cada tipo de solicitud.

**Figura 33**  
Flujos de trabajo



**Nota.** Interfaz de gestión de flujos de trabajo. Fuente: Elaboración propia (2025).

**Figura 34**  
Diseño de flujos de trabajo



**Nota.** Interfaz de diseño de flujos de trabajo. Fuente: Elaboración propia (2025).

La interfaz de flujos de trabajo permite diseñar y visualizar cómo deben gestionarse las solicitudes dentro del sistema, adaptándose a la forma de trabajo y métricas de Claro-Nicaragua, y adoptando las buenas prácticas de ITIL 2011. Mediante esta herramienta, se pueden definir los pasos y actividades que cada solicitud debe cumplir a lo largo de su ciclo de vida, incluyendo asignación de responsables, estados, prioridades y acciones específicas. Esto asegura una gestión consistente, eficiente y centralizada de las solicitudes, facilitando el seguimiento, la automatización de procesos y la correcta resolución según los criterios de la empresa.

#### 7.2.4.4. Control de versiones con Git

**Figura 35**  
Historial de commits en git

The image shows a Git commit history and a code diff view. The commit history at the top lists several commits with their messages, dates, and hashes. The commit 38f0518 is highlighted, showing its message 'add authentication and user management components' and the author 'jesushg23' committed on Nov 15, 2024. Below the commit history, the code diff for this commit is shown, highlighting changes in the '.eslintrc.cjs' file. The diff shows the removal of an old configuration and the addition of a new one with various ESLint plugins and settings.

Commit Message	Date	Hash
refactor: update import paths and component organization; remove unused Loader component and standardize naming conventions	22 Nov 2024 16:10	91734084
feat: add internationalization support; implement translation hooks and update UI components for localized content	22 Nov 2024 12:07	4b4b50cf
feat: add localized metadata and home page content; enhance translations for English and Spanish	21 Nov 2024 17:13	efbbaeb0
feat: enhance admin navigation with role-based access; update user middleware and improve user avatar handling	21 Nov 2024 12:16	2c28d6c6
refactor: remove unused console logs and components; update user fetching logic and add admin page with role-based access	21 Nov 2024 11:22	0a251a75
refactor: update database import paths and modify login form default values; change default login redirect to admin and fix authentication	20 Nov 2024 23:37	0e1f1bc1
refactor: rename project from 'request-master' to 'request-engine'; update database connection and remove unused chart components; add loading component...	20 Nov 2024 19:30	df9a437c
feat: add loading spinner and skeleton components; implement theme provider and collapsible UI elements	19 Nov 2024 15:18	c7665671
feat: add new components and hooks; implement privacy policy and terms of service pages	18 Nov 2024 16:35	c0d68ca3
feat: add tenant-scoped and auditable models for document management and messaging system	18 Nov 2024 14:53	45b1c40a
chore: update code style and configuration; add VSCode settings, remove unused hooks, add templates for generate admin ui and improve code generation	17 Nov 2024 11:56	721bbc6d
feat: add internationalization support	16 Nov 2024 05:10	ba465a07
refactor: update TRPC imports and enhance Prisma client; add form handling actions	16 Nov 2024 05:01	cf3f80b0
add authentication and user management components	15 Nov 2024 16:16	38f0518d
initialize project with Next.js, Tailwind CSS, and TRPC setup	15 Nov 2024 13:12	524f7d5c

**Commit 38f0518**  
jesushg23 committed on Nov 15, 2024

add authentication and user management components

85 files changed +17061 -3410 lines changed

```

... @@ -1,42 +1,112 @@
1  - /** @type {import("eslint").Linter.Config} */
2  - const config = {
3  -   "parser": "@typescript-eslint/parser",
4  -   "parserOptions": {
5  -     "project": true
6  +   /* eslint-disable import-x/no-named-as-default-member */
7  +   + import comments from '@eslint-community/eslint-plugin-eslint-comments/configs'
8  +   + import react from '@eslint-react/eslint-plugin'
9  +   + import { FlatCompat } from '@eslint/eslintrc'
10  +   + import js from '@eslint/js'
11  +   + import prettierConfig from 'eslint-config-prettier'
12  +   + import eslintPluginImportX from 'eslint-plugin-import-x'
13  +   + import regexPlugin from 'eslint-plugin-regexp'
14  +   + import security from 'eslint-plugin-security'
15  +   + import globals from 'globals'
16  +   + import tseslint from 'typescript-eslint'

```

**Nota.** Control de versiones mediante Git y GitHub. El historial de cambios permite la trazabilidad del código. Fuente: Elaboración propia (2025).

La imagen refleja la implementación del control de versiones mediante Git y la gestión del repositorio en GitHub. Se visualiza el historial de cambios, permitiendo identificar cada modificación realizada durante el desarrollo. También se detalla un commit, práctica clave para la trazabilidad y la revisión del código. Estas acciones fortalecen la colaboración. Asimismo, garantizan un mantenimiento estructurado y eficiente del proyecto.

### **7.2.5. Pruebas**

En esta etapa, se llevaron a cabo pruebas, incluyendo pruebas unitarias, así como pruebas de integración y de sistema validadas en conjunto, con el objetivo de identificar y corregir posibles errores antes del despliegue del proyecto. Para ello, se utilizaron herramientas como Cypress y Jest, las cuales permitieron automatizar las pruebas, obteniendo resultados rápidos y precisos que aseguraron la calidad y funcionalidad del sistema.

Las pruebas unitarias se centraron en validar el funcionamiento correcto de componentes y funciones individuales. Estas pruebas se aplicaron a todos los módulos del sistema, con especial énfasis en los más críticos: el módulo de seguridad (inicio de sesión) y el módulo principal (Solicitudes). Se buscó garantizar que cada función ejecutara correctamente las tareas asignadas y cumpliera con los requisitos definidos en la etapa de diseño. Estas pruebas permitieron detectar y corregir errores de forma aislada, mejorando así la fiabilidad del código fuente.

Por motivos de espacio, en este documento se incluyen únicamente los casos de prueba más representativos.

#### **7.2.5.1. Pruebas de Integración End-to-End (E2E) – Página de Inicio de sesión**

Se llevaron a cabo pruebas de integración end-to-end (E2E) utilizando la herramienta Cypress, con el objetivo de verificar el correcto funcionamiento del flujo completo de autenticación del usuario. Estas pruebas incluyeron la validación de la interfaz, el comportamiento ante credenciales incorrectas, la redirección tras un inicio de sesión exitoso, así como la navegación hacia las páginas de registro y recuperación de contraseña.

**Tabla 15**

Criterios de aceptación para pruebas de inicio de sesión

<b>N.</b>	<b>Rol (“Como...”)</b>	<b>Quiero (“... realizar ...”)</b>	<b>Para que (“... pueda ...”)</b>	<b>Criterios de aceptación</b>
1	Usuario	Ver el formulario de iniciar sesión	Confirmar que el usuario puede ingresar sus credenciales	<ul style="list-style-type: none"> <li>• Mostrar campos de correo y contraseña</li> <li>• Mostrar botón de ingresar</li> <li>• Mostrar enlaces de registro y recuperación</li> </ul>
2	Usuario	Mostrar errores si los campos están vacíos	Avisar al usuario que debe completar los campos	<ul style="list-style-type: none"> <li>• Mensaje “Campo requerido” debajo de cada campo</li> </ul>
3	Usuario	Validar formato de email antes de enviar	Evitar envío de email inválido	<ul style="list-style-type: none"> <li>• Mensaje de formato inválido</li> <li>• No enviar solicitud al servidor</li> </ul>
4	Usuario	Mostrar error por credenciales incorrectas	Informar al usuario que las credenciales no son válidas	<ul style="list-style-type: none"> <li>• Mensaje genérico de error de login</li> <li>• No redirigir</li> </ul>
5	Usuario	Iniciar sesión con credenciales válidas	Acceder al área segura	<ul style="list-style-type: none"> <li>• Redirigir a tablero principal</li> </ul>
6	Usuario	Navegar a la página de registro	Permitir crear una cuenta nueva	<ul style="list-style-type: none"> <li>• Enlace redirige a /auth/register</li> </ul>
7	Usuario	Navegar a recuperación de contraseña	Recuperar acceso si lo olvido	<ul style="list-style-type: none"> <li>• Enlace redirige a /auth/reset</li> </ul>
8	Usuario	Mantener sesión tras recargar	No pedir login nuevamente	<ul style="list-style-type: none"> <li>• Sesión persiste tras recarga</li> </ul>

**Tabla 16**

Caso de Prueba E2E Ver formulario de inicio de sesión

Identificador	PU-E2E-01
Rol	Usuario
Descripción	Validar que la página de inicio de sesión muestre correctamente todos los elementos del formulario.
Prerrequisitos	La URL de la página de login debe estar disponible y accesible desde el navegador.
Pasos	1. Abrir la página del login. 2. Visualizar campos y botones
Resultado esperado	Todos los elementos del formulario (campos, botones y enlaces) se visualizan correctamente sin errores de carga.

**Figura 36**

Código de prueba ver formulario de inicio de sesión



```
1  t('should display login form', () => {
2    cy.get('form').should('exist');
3    cy.get('input[type="email"]').should('exist');
4    cy.get('input[type="password"]').should('exist');
5    cy.get('button[type="submit"]').should('exist');
6  });
```

**Nota.** Código fuente de la prueba End-to-End (PU-E2E-01). Fuente: Elaboración propia (2025).

## Figura 37

### Resultado de prueba ver formulario de inicio de sesión

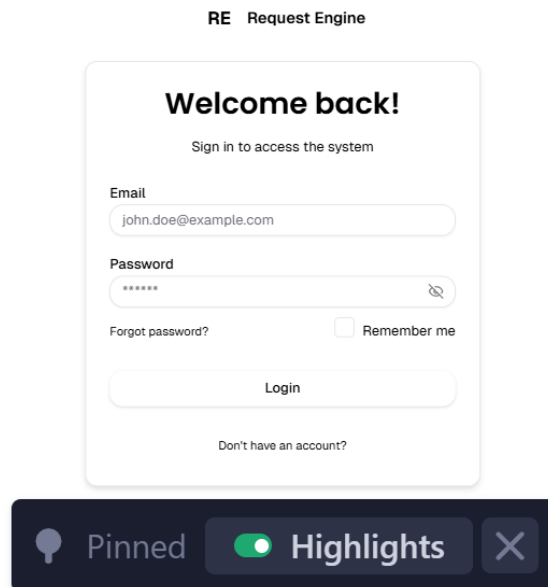
← ● should display login form 989ms

BEFORE EACH

```
1 visit /auth/login
```

TEST BODY

```
1 get form
2 -assert expected <form.space-y-6> to exist in the DOM
3 get input[type="email"]
4 -assert expected <input#_R_mhbsnp5tiqivb_-form-item.border-input.placeholder:text-muted-foreground.focus-visible:ring-ring.flex.h-8.w-full.rounded-md.border.bg-transparent.px-3.py-2.text-sm.shadow-xs.focus-visible:ring-offset-2.focus-visible:outline-hidden.disabled:cursor-not-allowed.disabled:opacity-50.md:text-sm> to exist in the DOM
5 get input[type="password"]
6 -assert expected <input#_R_16hbsnp5tiqivb_-form-item.border-input.placeholder:text-muted-foreground.focus-visible:ring-ring.flex.h-8.w-full.rounded-md.border.bg-transparent.px-3.py-2.text-sm.shadow-xs.focus-visible:ring-offset-2.focus-visible:outline-hidden.disabled:cursor-not-allowed.disabled:opacity-50.md:text-sm.hide-password-toggle.pr-10> to exist in the DOM
7 get button[type="submit"]
8 -assert expected <button.focus-visible:ring-ring.inline-flex.items-center.cursor-pointer.justify-center.gap-2.rounded-md.text-sm.font-medium.whitespace-nowrap.transition-colors.focus-visible:ring-1.focus-visible:outline-hidden.disabled:pointer-events-none.disabled:opacity-50.[&_svg]:pointer-events-none.[&_svg]:size-4.[&_svg]:shrink-0.bg-primary.text-primary-foreground.hover:bg-primary/90.shadow-sm.h-9.px-4.py-2.w-full> to exist in the DOM
(fetch) ● GET 200 /auth/register?_rsc=1jnu1
```



**Nota.** Resultado de la ejecución del caso de prueba PU-E2E-01. Fuente: Elaboración propia (2025).

**Tabla 17**

Caso de Prueba E2E Manejo de credenciales incorrectas al iniciar sesión

Identificador	PU-E2E-04
Rol	Usuario
Descripción	Validar que el sistema notifique al usuario cuando las credenciales ingresadas no coinciden con una cuenta registrada, impidiendo el acceso al sistema.
Prerrequisitos	Debe existir una cuenta válida registrada en el sistema. La página de inicio de sesión debe estar accesible.
Pasos	1. Ingresar una dirección de correo electrónico válida registrada en el sistema. 2. Ingresar una contraseña incorrecta. 3. Hacer clic en el botón "Ingresar".
Resultado esperado	Se muestra un mensaje de error indicando que las credenciales no son válidas. El sistema no permite el acceso ni redirige a otras páginas.

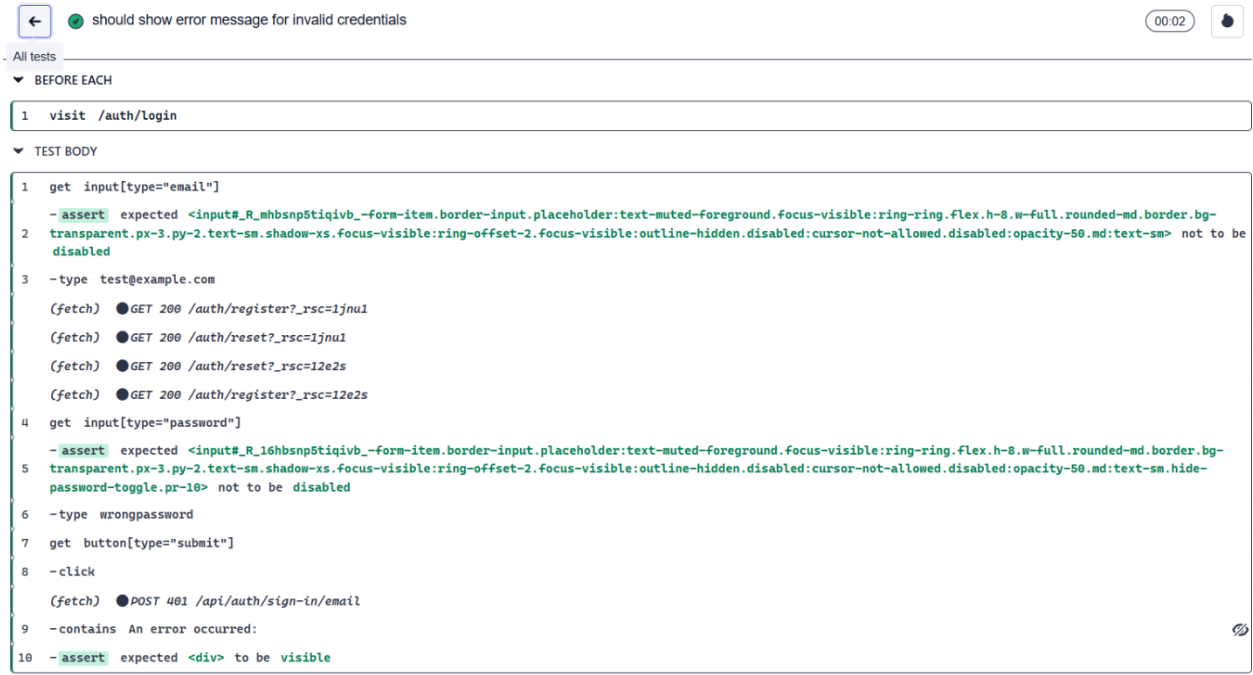
**Figura 38**

Código de prueba manejo de credenciales incorrectas al iniciar sesión

```
1  it('should show error message for invalid credentials', () => {
2    cy.get('input[type="email"]').should('not.be.disabled').type('test@example.com');
3    cy.get('input[type="password"]').should('not.be.disabled').type('wrongpassword');
4    cy.get('button[type="submit"]').click();
5    cy.contains('An error occurred:').should('be.visible');
6  });
```

**Nota.** Notificación al usuario cuando las credenciales ingresadas no coinciden con una cuenta registrada Fuente: Elaboración propia (2025).

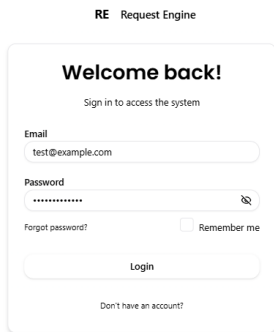
**Figura 39**  
Resultado de prueba manejo de credenciales incorrectas al iniciar sesión



The screenshot shows a test runner interface with the following details:

- Test title: `should show error message for invalid credentials`
- Duration: `00:02`
- Section: `BEFORE EACH`
- Step 1: `visit /auth/login`
- Section: `TEST BODY`
- Step 1: `get input[type="email"]`
- Assertion: `-assert expected <input#_R_mhbsnp5tiqivb_-form-item.border-input.placeholder:text-muted-foreground.focus-visible:ring-ring.flex.h-8.w-full.rounded-md.border.bg-transparent.px-3.py-2.text-sm.shadow-xs.focus-visible:ring-offset-2.focus-visible:outline-hidden.disabled:cursor-not-allowed.disabled:opacity-50.md:text-sm> not to be disabled`
- Step 2: `-type test@example.com`
- Fetches: `(fetch) GET 200 /auth/register?_rsc=1jnu1`, `(fetch) GET 200 /auth/reset?_rsc=1jnu1`, `(fetch) GET 200 /auth/reset?_rsc=12e2s`, `(fetch) GET 200 /auth/register?_rsc=12e2s`
- Step 3: `get input[type="password"]`
- Assertion: `-assert expected <input#_R_16hbsnp5tiqivb_-form-item.border-input.placeholder:text-muted-foreground.focus-visible:ring-ring.flex.h-8.w-full.rounded-md.border.bg-transparent.px-3.py-2.text-sm.shadow-xs.focus-visible:ring-offset-2.focus-visible:outline-hidden.disabled:cursor-not-allowed.disabled:opacity-50.md:text-sm.hide-password-toggle.pr-10> not to be disabled`
- Step 4: `-type wrongpassword`
- Step 5: `get button[type="submit"]`
- Step 6: `-click`
- Fetch: `(fetch) POST 401 /api/auth/sign-in/email`
- Step 7: `-contains An error occurred:`
- Step 8: `-assert expected <div> to be visible`

An error occurred: N/A



Pinned Highlights

**Nota.** Resultado de la prueba PU-E2E-04, notificación de credenciales invalidas.  
Fuente: *Elaboración propia (2025).*

**Tabla 18**

Caso de Prueba E2E Inicio de sesión con credenciales correctas

Identificador	PU-E2E-05
Rol	Usuario
Descripción	Verificar que, al ingresar credenciales válidas, el sistema autentique correctamente al usuario y permita el acceso a la página principal.
Prerrequisitos	Disponer de una cuenta de usuario registrada y activa en el sistema. La página de inicio de sesión debe estar accesible.
Pasos	1. Ingresar una dirección de correo electrónico válida. 2. Ingresar la contraseña correspondiente. 3. Hacer clic en el botón "Ingresar".
Resultado esperado	El sistema autentica correctamente al usuario y redirige a la página principal.

**Figura 40**

Código de prueba inicio de sesión con credenciales correctas

```
1  it('should successfully login with valid credentials', () => {
2    const testEmail = Cypress.env('TEST_USER_EMAIL');
3    const testPassword = Cypress.env('TEST_USER_PASSWORD');
4
5    cy.get('input[type="email"]').type(testEmail);
6    cy.get('input[type="password"]').type(testPassword);
7    cy.get('button[type="submit"]').click();
8
9    cy.url().should('not.include', '/auth/login');
10  });
```

**Nota.** Autenticación de usuario por ingreso de credenciales validas. Fuente: Elaboración propia (2025).

**Figura 41**  
Resultado de prueba inicio de sesión con credenciales correctas

```
← ● should successfully login with valid credentials 00:07 🔒  
All tests  
▼ BEFORE EACH  
1 visit /auth/Login  
▼ TEST BODY  
1 get input[type="email"]  
2 -type jess232016@gmail.com  
(fetch) ● GET 200 /auth/register?_rsc=1jnu1  
(fetch) ● GET 200 /auth/reset?_rsc=1jnu1  
(fetch) ● GET 200 /auth/reset?_rsc=12e2s  
(fetch) ● GET 200 /auth/register?_rsc=12e2s  
3 get input[type="password"]  
4 -type Lamisma123*  
5 get button[type="submit"]  
6 -click  
(fetch) ● POST 200 /api/auth/sign-in/email  
7 url  
8 -assert expected https://request-admin.vercel.app/admin/2DA1FC13-1F87-4A5D-A64C-05823686A111 to not include /auth/login  
(page load) --page loaded--  
(new url) https://request-admin.vercel.app/admin/2DA1FC13-1F87-4A5D-A64C-05823686A111
```

**Dashboard**  
Welcome, Jesus Hernandez, to the Claro request management system.

General Workflow SLAs

**General Metrics**

Total Requests <b>67</b> + 0.0% vs. last month	Average Resolution Time <b>N/A</b> + 0.0% vs. last month	Resolution Rate <b>0.0%</b> + 0.0% vs. last month	Pending Requests <b>67</b> + 0.0% vs. last month
--	--	---	--

**SLA Metrics and Fixed States**

Overdue Requests (SLA) <b>0</b> + 0.0% vs. last month	Average SLA Met <b>0%</b> + 0.0% vs. last month	SLA At Risk <b>0</b> + 0.0% vs. last month	Draft Requests <b>0</b> + 0.0% vs. last month
---	---	--	---

**Requests by Workflow Analysis**  
Requests distribution by workflow type in the last 6 months

JH Jesus Hernandez  
jess232016@gmail.com

Pinned request response

**Nota.** Resultado de pruebas de usuario autenticado por medio de credenciales validas.  
Fuente: Elaboración propia (2025).

### 7.2.5.2. Pruebas Unitarias – Modulo Configuración

Para el diseño y ejecución de las pruebas unitarias del módulo “Configuración”, se identificaron las principales funcionalidades críticas a validar: el cambio de estado de la solicitud, la asignación de prioridades, la gestión de usuarios asignados y la clasificación de las solicitudes.

A continuación, se documentan cuatro pruebas unitarias representativas que validan el comportamiento esperado de dichas funcionalidades.

**Tabla 19**

Prueba unitaria criterios de aceptación para el módulo “Configuración”

N.	Rol (“Como ...”)	Quiero (“... realizar ...”)	Para que (“... pueda ...”)	Criterios de aceptación
1	Coordinador	Cambiar el estado de una solicitud	Mantener el flujo actualizado	<ul style="list-style-type: none"> <li>• Registrar el nuevo estado en la asignación</li> <li>• Agregar un registro en el historial de cambios</li> <li>• Confirmar que el usuario esté autenticado</li> </ul>
2	Coordinador	Cambiar la prioridad de una solicitud	Ajustar el nivel de urgencia	<ul style="list-style-type: none"> <li>• Actualizar la prioridad en la asignación</li> <li>• Confirmar tipo de cambio válido y sesión activa</li> </ul>
3	Analista	Actualizar usuarios asignados	Definir quién atiende la solicitud	<ul style="list-style-type: none"> <li>• Reemplazar la lista de responsables actuales</li> <li>• Guardar los cambios en una sola operación</li> </ul>
4	Coordinador	Actualizar clasificación de la solicitud	Asignar correctamente área y categoría	<ul style="list-style-type: none"> <li>• Cambiar área, categoría y subtipo</li> <li>• Confirmar que el cambio se registre en el historial</li> </ul>

**Tabla 20**

Prueba unitaria: actualizar estado de solicitud

<b>Identificador</b>	<b>PU-CONF-01</b>
<b>Rol</b>	Coordinador
<b>Descripción</b>	Verificar que el coordinador pueda modificar correctamente el estado de una solicitud, registrando el cambio junto con su motivo correspondiente en el historial del sistema.
<b>Prerrequisitos</b>	Usuario autenticado con el rol de Coordinador.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Solicitud con un estado inicial distinto al nuevo estado deseado.</li> <li>2. Cambiar el estado de la solicitud e ingresar el motivo del cambio.</li> <li>3. Verificar que el nuevo estado se haya aplicado correctamente.</li> <li>4. Consultar el historial para confirmar que el cambio fue registrado.</li> </ol>
<b>Resultado esperado</b>	La solicitud cambia al nuevo estado seleccionado y el cambio, junto con el motivo, se registra correctamente en el historial.

**Figura 42**

Prueba unitaria código de la prueba actualizar estado de solicitud

```

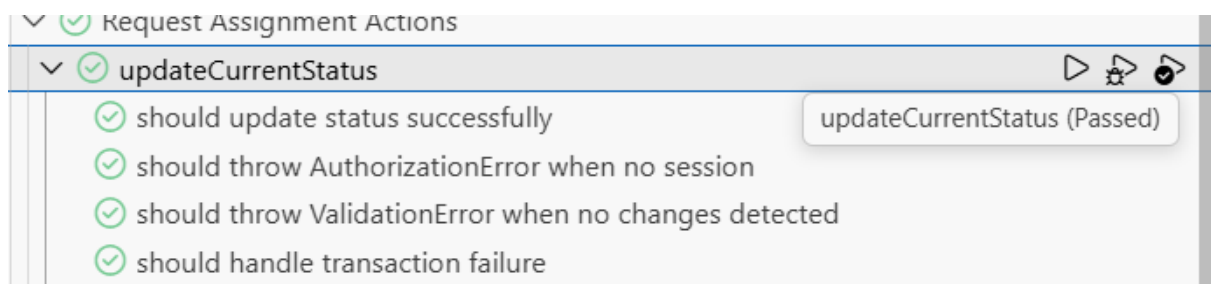
1  it('should update status successfully', async () => {
2      const result = await updateCurrentStatus(mockTenantId, mockRequestId, mockStatusId, mockMetadata);
3
4      expect(result).toEqual({ statusId: mockStatusId });
5      expect(mockDb.$transaction).toHaveBeenCalled();
6      expect(mockDb.requestChangeLog.create).toHaveBeenCalledWith({
7          data: expect.objectContaining({
8              tenantId: mockTenantId,
9              requestId: mockRequestId,
10             updatedBy: mockSession.user.id,
11             fieldName: 'status',
12             oldValue: mockLastAssignment.statusId,
13             newValue: mockStatusId,
14             metadata: JSON.stringify(mockMetadata),
15         }),
16     });
17 });

```

*Fuente: Elaboración propia (2025).*

**Figura 43**

Prueba unitaria resultado de la prueba actualizar estado de solicitud



Fuente: *Elaboración propia, 2025.*

### 7.2.5.3. Pruebas Unitarias – Modulo Solicitud

En esta sección se describen las pruebas unitarias desarrolladas para el módulo de Solicitud, cuyo objetivo es asegurar que cada componente funcional cumpla con los requerimientos definidos. Estas pruebas se centran en validar el correcto funcionamiento de las funcionalidades principales, incluyendo la creación, actualización, obtención por ID, visualización de detalles y listado de prioridades.

**Tabla 21**

Prueba unitaria criterios de aceptación para el módulo de “Solicitudes”

N.	Rol	Quiero (“... realizar ...”)	Para que (“... pueda ...”)	Criterios de aceptación
1	Distribuidor	Crear una nueva solicitud de servicio	Enviar petición de activación o cambio de servicio	<ul style="list-style-type: none"> <li>• Se crea la solicitud con todos los campos obligatorios</li> <li>• Se asigna estado inicial “Borrador”</li> <li>• Se registra fecha y usuario</li> </ul>
2	Analista	Actualizar el estado de una solicitud existente	Reflejar cambios de progreso en la atención	<ul style="list-style-type: none"> <li>• Solo el analista puede mover estados a “En progreso” o “Cerrado”</li> <li>• Se guarda historial de cambios</li> <li>• Error si estado inválido</li> </ul>
3	Coordinador	Consultar los datos básicos de una solicitud por su ID	Ver información resumida para seguimiento	<ul style="list-style-type: none"> <li>• Muestra título, descripción y categoría</li> <li>• Lanza error si no tiene permiso</li> </ul>

				<ul style="list-style-type: none"> <li>• Incluye nombre del solicitante</li> </ul>
4	Analista	Ver detalles ampliados de una solicitud	Acceder a encuestas, guías y métricas asociadas	<ul style="list-style-type: none"> <li>• Incluye conteos de formularios y encuestas</li> <li>• Muestra lista de documentos y usuarios asignados</li> <li>• Muestra métricas de actividad</li> </ul>
5	Administrador	Listar tipos de prioridad como opciones	Configurar niveles de atención en el sistema	<ul style="list-style-type: none"> <li>• Retorna lista {label, value }</li> <li>• Incluye nombre y código de prioridad</li> </ul>

**Tabla 22**

Prueba unitaria actualizar solicitud existente

<b>Identificador</b>	<b>PU-SOLICITUD-02</b>
Rol	Analista, Distribuidor, Coordinador, Administrador
Descripción	El usuario modifica el estado y detalles de una solicitud para reflejar el progreso y cambios realizados.
Prerrequisitos	Usuario autenticado en el sistema.
Pasos	<ol style="list-style-type: none"> <li>1. Ir al módulo de solicitudes y seleccionar "Editar".</li> <li>2. Actualizar los campos disponibles en la edición: Canal de venta, Área, Tipo solicitud, categoría, Subcategoría, Asunto, Prioridad, Requerimientos (si aplica).</li> <li>3. Guardar cambios.</li> </ol>
Resultado esperado	La solicitud se actualiza correctamente cuando los datos son válidos y se genera un error en caso de datos inválidos.

## Figura 44

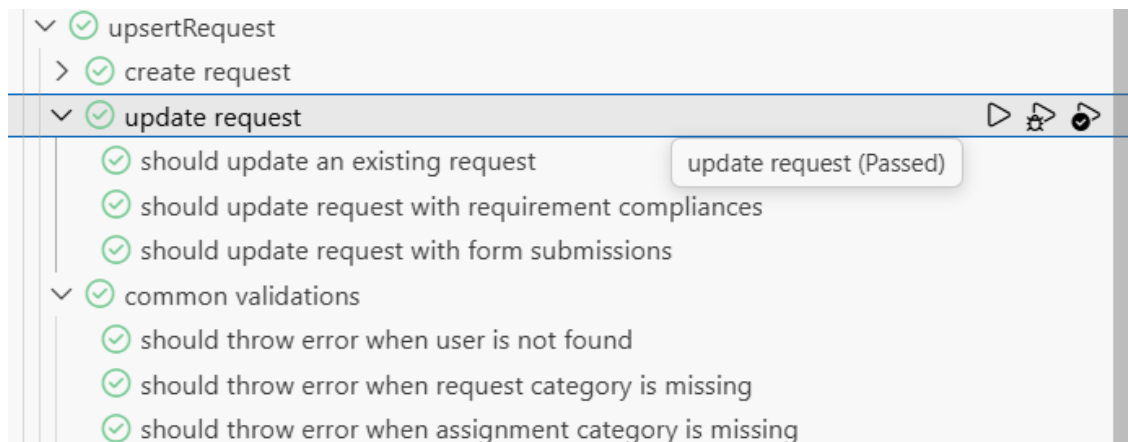
Prueba unitaria código de prueba actualizar solicitud existente

```
1  it('should update request with requirement compliances', async () => {
2      (mockDb.requestCategoryRequirement.findMany as jest.Mock).mockResolvedValue([
3          { requirementId: 'req-1', isActive: true },
4          { requirementId: 'req-2', isActive: false },
5      ]);
6
7      const dataWithCompliances = {
8          ...mockRequestData,
9          requirementCompliances: {
10             'req-1': true,
11             'req-2': false,
12         },
13     };
14
15     const result = await upsertRequest(mockTenantId, dataWithCompliances);
16     expect(result).toBeDefined();
17 });
```

Fuente: Elaboración propia, 2025.

## Figura 45

Prueba unitaria resultado de prueba actualizar solicitud existente



Fuente: Elaboración propia, 2025.

**Tabla 23**

Prueba unitaria ver detalle de solicitud

Identificador	PU-SOLICITUD-03
Rol	Analista, Distribuidor, Coordinador, Administrador
Descripción	El usuario ve el detalle de una solicitud especifica mediante el Id.
Prerrequisitos	Usuario autenticado en el sistema.
Pasos	1. Ir al módulo de solicitudes y seleccionar la opción “Ver”. 2. Sistema muestra el detalle de la solicitud en una vista nueva.
Resultado esperado	Se devuelve la información básica de la solicitud cuando el ID es válido, o se genera un error de ‘no encontrado’ o acceso denegado según el caso

**Figura 46**

Pruebas unitarias código de prueba ver detalle de solicitud

```

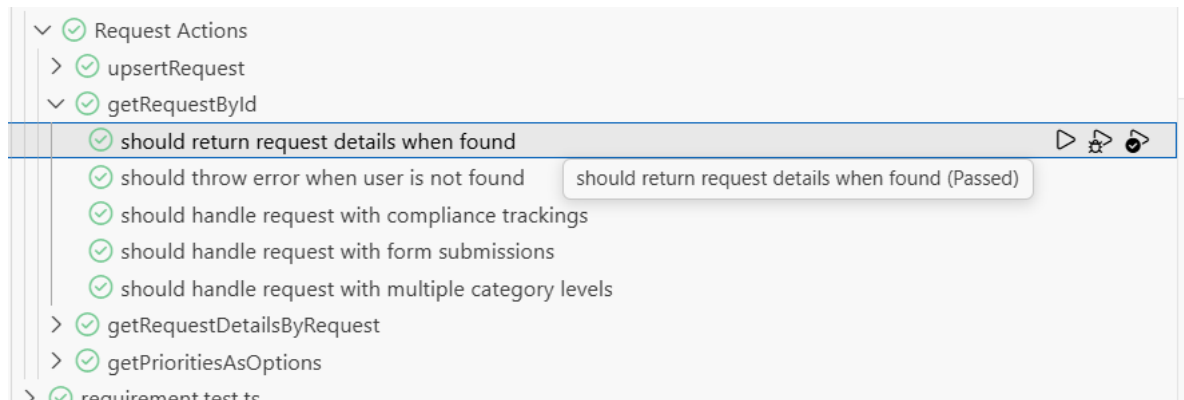
1  ('should return request details when found', async () => {
2      const mockRequest = {
3          id: mockRequestId,
4          requestAssignments: [
5              {
6                  area: { id: 'area-1', name: 'Area 1' },
7                  status: { id: 'status-1', name: 'Status 1' },
8                  priority: { id: 'priority-1', name: 'Priority 1' },
9                  requestCategoryId: 'cat-1',
10                 assignmentCategoryId: 'assign-cat-1',
11             },
12         ],
13         complianceTrackings: [],
14         formSubmission: [],
15         executionFlowDefinitions: [],
16     };
17
18     (mockDb.request.findUniqueOrThrow as jest.Mock).mockResolvedValue(mockRequest);
19     (mockDb.requestCategory.findMany as jest.Mock).mockResolvedValue([
20         { id: 'cat-1', name: 'Category 1', parentCategoryId: null },
21     ]);
22     (mockDb.assignmentCategory.findMany as jest.Mock).mockResolvedValue([
23         { id: 'assign-cat-1', name: 'Assignment Category 1', parentCategoryId: null },
24     ]);
25
26     const result = await getRequestById(mockTenantId, mockRequestId);
27
28     expect(result).toBeDefined();
29     expect(result.id).toBe(mockRequestId);
30
31 });

```

*Fuente: Elaboración propia, 2025.*

## Figura 47

Pruebas unitarias resultado de prueba ver detalle de solicitud



Fuente: *Elaboración propia, 2025.*

### 7.2.6. Despliegue

Durante la fase de despliegue, el sistema web y la base de datos fueron provisionados y configurados en el entorno de producción de la infraestructura tecnológica de Claro-Nicaragua. Este proceso aseguró la operatividad total de los componentes, garantizando su correcta integración con los procesos internos y las políticas de red de la empresa.

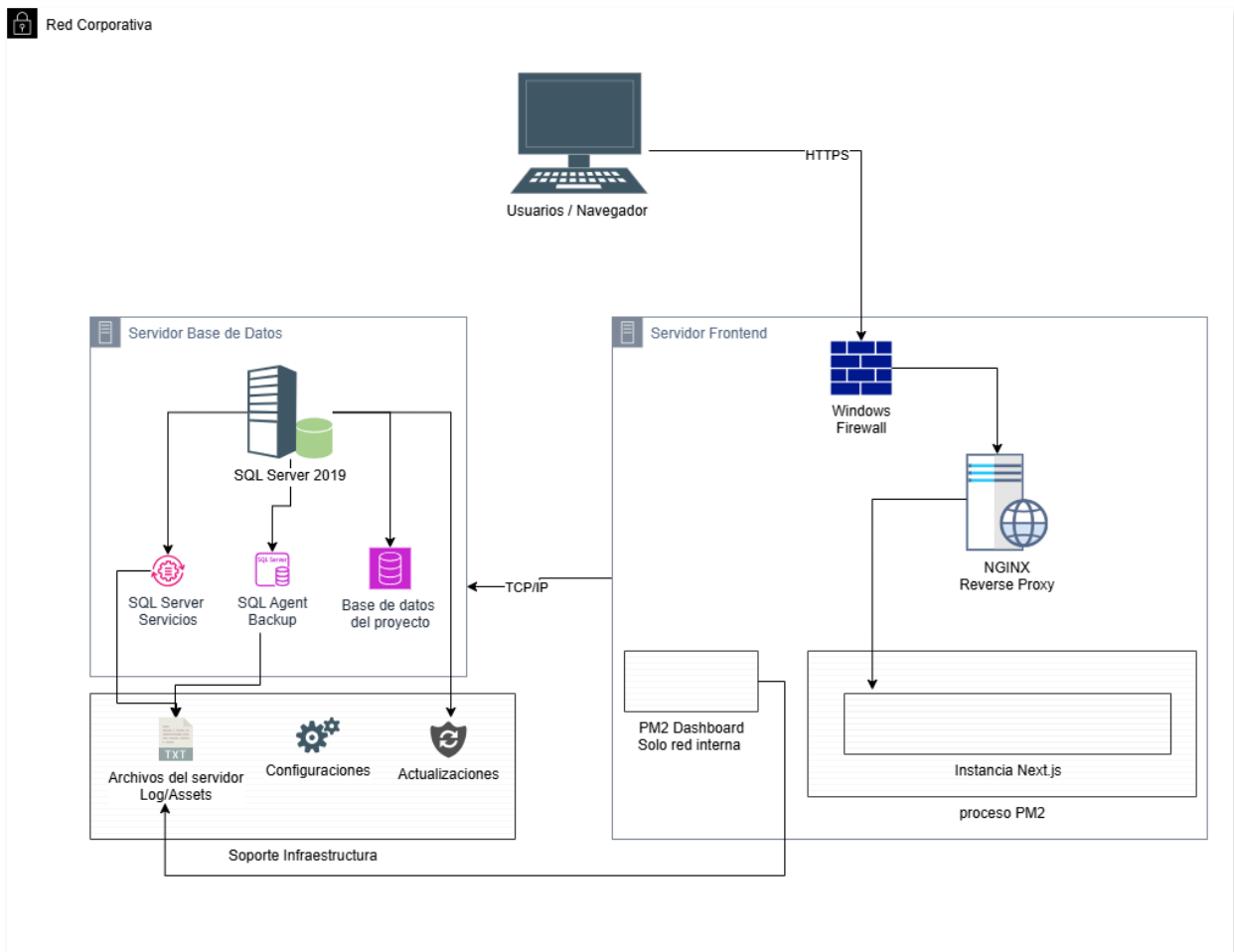
#### 7.2.6.1. Arquitectura y Hardware Utilizado

La solución fue implementada sobre servidores físicos de nivel empresarial, configurados para soportar la carga transaccional del sistema. La infraestructura se dividió en dos nodos principales para garantizar la separación de responsabilidades:

- **Servidor de Aplicación (Frontend/Backend):** Equipado con procesadores Intel Xeon y 64 GB de memoria RAM. En este servidor se configuró el entorno de ejecución Node.js gestionado por PM2, utilizando Nginx como servidor proxy inverso y firewall de aplicación web.
- **Servidor de Base de Datos:** Ejecutándose sobre Windows Server 2019, este nodo aloja el motor SQL Server 2019. Cuenta con unidades de estado sólido (SSD) con tecnología NVMe para optimizar los tiempos de lectura/escritura y conectividad Dual Gigabit Ethernet para evitar cuellos de botella en la red.

A continuación, en la Figura 48, se detalla el diagrama de despliegue que ilustra la interacción lógica y física entre estos componentes.

**Figura 48**  
Diagrama de despliegue del sistema web y base de datos



**Nota.** El diagrama muestra el flujo de comunicación desde el cliente, gestionado por Nginx y PM2, hasta la persistencia de datos en SQL Server. Fuente: Elaboración propia, 2025

### 7.2.6.2. Pruebas de Conectividad

Como parte de la verificación del despliegue, se realizaron pruebas de conectividad entre el servidor de aplicación y el servidor de base de datos, utilizando el comando ping para medir disponibilidad y latencia.

El objetivo fue confirmar que ambos servidores pueden comunicarse de forma continua y estable dentro de la red corporativa.

#### Resultado:

A continuación, se presentan las evidencias de las pruebas realizadas. Por motivos de seguridad y confidencialidad de la información de Claro-Nicaragua, las direcciones IP han sido anonimizadas.

#### Figura 49

Prueba de Conectividad (Ping) desde Servidor de Aplicación a Servidor de Base de Datos

```
D:\Users\admin-web>ping 192.x.x.x

Haciendo ping a 192.x.x.x con 32 bytes de datos:
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127
Respuesta desde 192.x.x.x: bytes=32 tiempo=3ms TTL=127
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127

Estadísticas de ping para 192.x.x.x:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
            (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
            Mínimo = 2ms, Máximo = 3ms, Media = 2ms
```

Fuente: Elaboración propia, 2025.

## Figura 50

### Prueba de Conectividad (Ping) del Servidor de Base de Datos al Servidor de Aplicación

```
E:\Users\admin-bd>ping 192.x.x.x
```

```
Haciendo ping a 192.x.x.x con 32 bytes de datos:
```

```
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127
```

```
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127
```

```
Respuesta desde 192.x.x.x: bytes=32 tiempo=3ms TTL=127
```

```
Respuesta desde 192.x.x.x: bytes=32 tiempo=2ms TTL=127
```

```
Estadísticas de ping para 192.x.x.x:
```

```
Paquetes: enviados = 4, recibidos = 4, perdidos = 0
```

```
(0% perdidos),
```

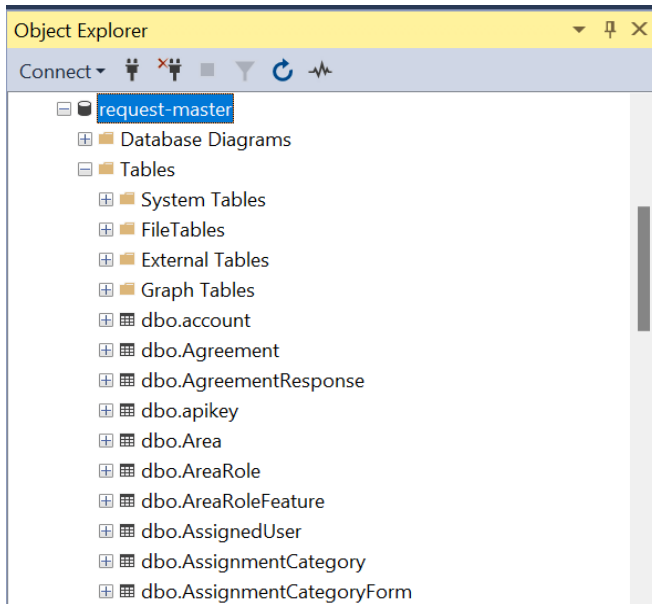
```
Tiempos aproximados de ida y vuelta en milisegundos:
```

```
Mínimo = 2ms, Máximo = 4ms, Media = 2ms
```

**Nota.** La conectividad bidireccional fue verificada exitosamente. Fuente: Elaboración propia, 2025.

Las pruebas mostraron respuesta exitosa, sin pérdida de paquetes y con latencia promedio de 0 ms, lo cual confirma una comunicación óptima para el funcionamiento del sistema.

**Figura 51**  
Base de datos alojada en SQL Server



**Nota.** Visualización del entorno de base de datos en producción. Fuente: Elaboración propia, 2025.

**Figura 52**  
Estado operativo del servicio web gestionado por PM2

```
$ pm2 start node_modules/next/dist/bin/next --name "request-admin" -- start
[PM2] Starting C:\Users\Operaciones\Documents\request-admin\node_modules\next\dist\bin\next in fork_mode (1 instance
)
[PM2] Done.
```

id	name	mode	🔄	status	cpu	memory
0	request-admin	fork	0	online	17.2%	48.3mb

```
Operaciones@OPERACIONES-WEB MINGW64 ~/Documents/request-admin (main)
$ pm2 list
```

id	name	mode	🔄	status	cpu	memory
0	request-admin	fork	0	online	0%	216.4mb

**Nota.** La captura muestra la instancia "request-admin" con ID 0 en estado online (activo), confirmando que el servidor de aplicaciones está recibiendo peticiones correctamente. Fuente: Elaboración propia, 2025.

### **7.2.7. Mantenimiento**

La fase de mantenimiento se ejecutó durante un período de un mes posterior al despliegue del sistema. Durante este tiempo, se realizó una supervisión constante para identificar y corregir posibles errores en el sistema web y su base de datos asociada. El objetivo principal de esta fase fue garantizar el funcionamiento óptimo y continuo del sistema, asegurando su estabilidad y adaptabilidad a las necesidades operativas de Claro-Nicaragua.

## **VIII. CONCLUSIONES**

El presente trabajo monográfico ha permitido subsanar las limitaciones críticas que afectaban la gestión de solicitudes en Claro-Nicaragua, entregando una solución que cumple con todos los requerimientos establecidos, tal como se valida en la carta de aprobación (ver anexo H).

Al analizar el proceso de gestión de solicitudes, se identificaron cuellos de botella como la falta de notificaciones automáticas y la descentralización de la información. El nuevo sistema web resuelve estos problemas al centralizar el flujo de trabajo en una solución única y accesible.

La automatización de las notificaciones por correo electrónico y la capacidad de asignar prioridades y SLAs han transformado la dinámica de atención, pasando de un modelo reactivo a uno proactivo y organizado. Además, el módulo de reportes exportables facilita a las gerencias la auditoría y el control del rendimiento del personal.

El sistema desplegado moderniza el stack tecnológico y, gracias al módulo de Diseñador de Formularios, ofrece una flexibilidad técnica que elimina la dependencia de desarrollo continuo para cambios menores, otorgando autonomía a los administradores.

Finalmente, su despliegue en la infraestructura física de Claro-Nicaragua, en conjunto con los controles operativos y de seguridad definidos, favorece un entorno estable y confiable, orientado a la continuidad operativa, la escalabilidad futura y la mejora continua en la gestión de servicios de telecomunicaciones. Estas buenas prácticas contribuirán, en una etapa de maduración posterior, a la eficiencia, transparencia y reducción de tiempos de resolución para la satisfacción del cliente.

Todos los elementos antes mencionados evidencian el cumplimiento satisfactorio de los objetivos establecidos en este proyecto monográfico.

## **IX. RECOMENDACIONES**

A partir de los resultados obtenidos en el desarrollo e implementación del sistema web para el control de solicitudes de servicios de telecomunicaciones en la empresa Claro-Nicaragua, se proponen las siguientes recomendaciones orientadas a garantizar el correcto funcionamiento, mantenimiento y evolución futura de la solución:

### **1. Capacitación Continua y Gestión del Cambio:**

Realizar talleres de inducción periódicos para los nuevos analistas y coordinadores sobre el uso del sistema, especialmente en el aprovechamiento del "**Diseñador de Formularios**" y la configuración de "**Flujos de Trabajo**". Esto permitirá que la herramienta evolucione orgánicamente según las necesidades del negocio sin requerir intervenciones de código constantes.

### **2. Plan de Mantenimiento Evolutivo y Seguridad:**

Establecer un protocolo formal de actualizaciones para las librerías del entorno (Next.js, Node.js) y copias de seguridad en SQL Server. Dada la rápida evolución del ecosistema JavaScript, se recomienda una revisión semestral de dependencias para mitigar vulnerabilidades y asegurar que el sistema se mantenga bajo los estándares tecnológicos actuales.

### **3. Escalabilidad a Otras Áreas:**

Evaluar la factibilidad de extender el uso del sistema a otras áreas de Claro-Nicaragua que manejen flujos de solicitudes similares. La arquitectura basada en roles y la parametrización de formularios ya implementada permite que esta transición sea de bajo impacto técnico y alto valor operativo, convirtiendo la solución en un estándar interno para la gestión de servicios.

## X. BIBLIOGRAFÍA

- Adegoke, D. (11 de Jul de 2023). *tRPC vs. GraphQL: Which is better for your projects?* Obtenido de tRPC vs. GraphQL: Which is better for your projects?: <https://blog.logrocket.com/trpc-vs-graphql-better-projects/#:~:text=tRPC%20is%20an%20open%20source,API%20development%20over%20HTTP%2F2>.
- Ameyo. (31 de December de 2020). *What is a helpdesk?* Obtenido de What is a helpdesk?: <https://www.ameyo.com/what-is-helpdesk/#>
- Asiuwhu, E. (13 de Jul de 2023). *How to use NextAuth.js for client-side authentication in Next.js*. Obtenido de How to use NextAuth.js for client-side authentication in Next.js: <https://blog.logrocket.com/how-to-use-nextauth-js-client-side-authentication-next-js/#:~:text=js%3F-,NextAuth.,MongoDB%2C%20PostgreSQL%2C%20and%20MariaDB>.
- Ayala, C. (2 de Noviembre de 2019). *¿Qué es ITIL? - Conceptos y principios de las buenas prácticas ITIL*. Obtenido de ¿Qué es ITIL? - Conceptos y principios de las buenas prácticas ITIL: [https://www.servicetonic.com/es/itil/3-itil-conceptos-y-principios/#%C2%BFQue\\_es\\_ITIL\\_Conceptos\\_y\\_principios](https://www.servicetonic.com/es/itil/3-itil-conceptos-y-principios/#%C2%BFQue_es_ITIL_Conceptos_y_principios)
- Clark, H. (21 de January de 2024). *Justinmind Prototyping Software In-Depth*. Obtenido de Justinmind Prototyping Software In-Depth: <https://theproductmanager.com/tools/justinmind-review/#:~:text=Justinmind%20specializes%20in%20providing%20prototyping,on%20creating%20high%2Dfidelity%20prototypes>.
- Claro-Nicaragua. (18 de Enero de 2011). *Misión, Visión y Valores*. Obtenido de Misión, Visión y Valores: <https://claronica.blogspot.com/2011/01/mision-vision-y-valores.html>
- Department of Veterans Affairs, Office of Information & Technology, & Enterprise Program Management Office. (2010). *Visual Paradigm*. Obtenido de Visual Paradigm: <https://www.oit.va.gov/Services/TRM/ToolPage.aspx?tid=10208#:~:text=Description%3A,engineering%20capabilities%20including%20code%20generation>.

Deshpande, C. (5 de October de 2023). *The Best Guide to Know What Is React*. Obtenido de The Best Guide to Know What Is React: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>

DGI, D. G. (22 de enero de 2013). *Reglamento de la Ley No. 822, Ley de Concertación Tributaria: Artículo 44, numeral 2.5 literal a*. Obtenido de Ministerio de Hacienda y Crédito Público de Nicaragua: <https://www.dgi.gob.ni/getDoc/1041>

Filehippo.com. (s.f.). *Descargar CASE Studio 2.25 para Windows*. Obtenido de Descargar CASE Studio 2.25 para Windows: [https://filehippo.com/es/download\\_case-studio/](https://filehippo.com/es/download_case-studio/)

Heller, M. (08 de Jul de 2022). *What is Visual Studio Code? Microsoft's extensible code editor*. Obtenido de What is Visual Studio Code? Microsoft's extensible code editor: <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>

KeepCoding. (13 de Noviembre de 2023). *¿Qué es Git y por qué es tan importante?* Obtenido de ¿Qué es Git y por qué es tan importante?: <https://keepcoding.io/blog/que-es-git-y-por-que-es-tan-importante/>

Kiruthika, D. (27 de March de 2024). *Waterfall Model In Software Testing*. Obtenido de Waterfall Model In Software Testing: <https://testsigma.com/blog/waterfall-model-in-software-testing/>

Mendez, E. d. (24 de Septiembre de 2024). *Entrevista de Evaluación del Sistema de HelpDesk y Análisis del Flujo de Procesos para la Atención de Solicitudes de Servicios de Telecomunicación. (D. Acevedo, Entrevistador)*

Monitor, T. (06 de Enero de 2023). *What is SQL Server?* Obtenido de What is SQL Server?: <https://techmonitor.ai/what-is/what-is-sql-server>

Pfeiffer, S. (04 de Feb de 2022). *Start Unit Testing Your Javascript with Jest*. Obtenido de Start Unit Testing Your Javascript with Jest: <https://dev.to/codesphere/start-unit-testing-your-javascript-with-jest-4lid>

*Prisma ORM*. (s.f.). Obtenido de Prisma ORM: <https://docs.yugabyte.com/preview/drivers-orms/nodejs/prisma/>

ProProfs. (s.f.). *Online Help Desk Software*. Obtenido de Online Help Desk Software: <https://www.proprofsdesk.com/pricing/>

Sigdestad, T. (03 de Noviembre de 2022). *What is Next.js?* Obtenido de What is Next.js?: <https://www.enonic.com/blog/what-is-next-js>

*SolarWinds*. (s.f.). Obtenido de SolarWinds: <https://www.solarwinds.com/es/free-tools/free-help-desk-software>

Souza, J. (27 de Junio de 2023). *CYPRESS: LA HERRAMIENTA QUE AUTOMATIZA TUS PRUEBAS Y GARANTIZA LA CALIDAD DE TUS PROYECTOS*. Obtenido de CYPRESS: LA HERRAMIENTA QUE AUTOMATIZA TUS PRUEBAS Y GARANTIZA LA CALIDAD DE TUS PROYECTOS: <https://itequia.com/es/cypress-la-herramienta-que-automatiza-tus-pruebas-y-garantiza-la-calidad-de-tus-proyectos/>

Steinberg, R. (2011). ITIL Service Operation. En R. Steinberg, *ITIL Service Operation* (págs. 86-87-88-89-90-91-92-93-94-95). TSO (The Stationery Office).

Torrez, E. A. (23 de Septiembre de 2024). Entrevista Inicial para Identificar Deficiencias en el Sistema Actual de Helpdesk. (D. J. Acevedo, Entrevistador)

Williams, P. (03 de Julio de 2023). *WHAT IS TYPESCRIPT?* Obtenido de WHAT IS TYPESCRIPT?: <https://medium.com/@pryncwill819/what-is-typescript-35f4e6f37bd5>

*Zendesk*. (15 de Octubre de 2020). Obtenido de Zendesk: <https://www.zendesk.com.mx/service/>

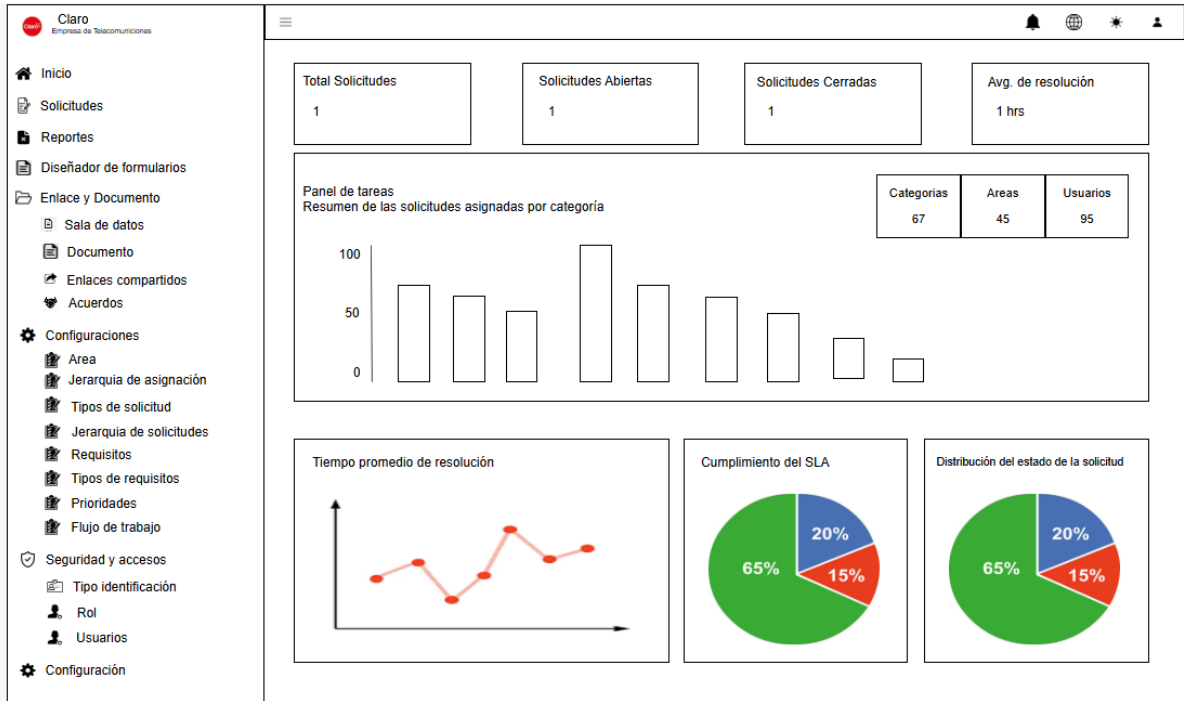
# ANEXOS

## Anexo A

Diseño de interfaces gráficas de usuario de sistema

### Figura A. 1

Menú de navegación y dashboards



Fuente: Elaboración propia, 2025.

**Figura A. 2**  
Crear Formulario

Crear nuevo formulario  
Crea un nuevo formulario para comenzar a recibir respuestas

Nombre  
[Input field]

Descripción  
[Input field]

Formulario público  
Haz que este formulario sea público para que cualquier persona pueda verlo y enviar respuestas

Guardar

Formulario: Formulario 1

Vista Previa Guardar Publicar

Suelta aquí

Arrastra y suelta elementos

Elementos del formulario

- Text field
- Number field
- TextArea field
- Date field
- Select field
- Checkbox field

Elementos de diseño

- Title field
- Subtitle field
- Paragraph field
- Separator field
- Spacer field

*Fuente: Elaboración propia, 2025.*

**Figura A. 3**  
Tipos de Solicitudes

The screenshot shows a web interface for managing request types. At the top, there is a breadcrumb trail: Admin >> Claro >> Configuraciones >> Tipo de Solicitudes. On the right, there are icons for notifications, a globe, a star, and a user profile. Below the breadcrumb, there are buttons for 'Exportar' and '+ Nuevo'. On the left, there are buttons for 'Opciones de vista' and 'Ordenar'. The main content is a table with the following columns: Nombre, Descripción, Elegibles para nuevos clientes, SubCategorías, Formularios, Requisitos, and Fecha Creación. The table lists five request types, all with 0 subcategories, 0 forms, and 2 requirements. Below the table is a modal titled 'Tipos de servicios' with columns: Nombre, Descripción, Elegible para nuevos clientes, Formularios, Subcategorías, and Requisitos. This modal lists five service types with varying numbers of requirements (1, 11, 4, 6, 9). At the bottom right, there is a pagination control showing 'Filas por pagina' set to 10, and 'Pagina 1 de 1' with navigation arrows.

Nombre	Descripción	Elegibles para nuevos clientes	SubCategorías	Formularios	Requisitos	Fecha Creación
<input type="checkbox"/> Tipo Solicitud 1	Descripción tipo solicitud 1	<input type="checkbox"/>	0	0	2	dd / mm / yyyy
<input type="checkbox"/> Tipo Solicitud 2	Descripción tipo solicitud 2	<input type="checkbox"/>	0	0	2	dd / mm / yyyy
<input type="checkbox"/> Tipo Solicitud 3	Descripción tipo solicitud 3	<input type="checkbox"/>	0	0	2	dd / mm / yyyy
<input type="checkbox"/> Tipo Solicitud 4	Descripción tipo solicitud 4	<input type="checkbox"/>	0	0	2	dd / mm / yyyy
<input checked="" type="checkbox"/> Tipo Solicitud 5	Descripción tipo solicitud 5	<input type="checkbox"/>	0	0	2	dd / mm / yyyy

Nombre	Descripción	Elegible para nuevos clientes	Formularios	Subcategorías	Requisitos
Tipo servicio 1	Descripción 1	<input type="radio"/>	0	0	1
Tipo servicio 2	Descripción 2	<input type="radio"/>	0	0	11
Tipo servicio 3	Descripción 3	<input type="radio"/>	0	0	4
Tipo servicio 4	Descripción 4	<input type="radio"/>	0	0	6
Tipo Servicio 5	Descripción 5	<input type="radio"/>	0	0	9

Fuente: Elaboración propia, 2025.

**Figura A. 4**  
Requerimientos de tipos de solicitudes

The screenshot shows a modal window titled 'Requerimientos asociados con tipo de Solicitud' with a red close button (X) in the top right corner. Below the title, it says 'A continuación, se muestran los requerimientos asociados con este tipo de Solicitud:'. The modal contains a table with two columns: Nombre and Descripción. The table lists two requirements: 'Requerimiento 1' and 'Requerimiento 2'.

Nombre	Descripción
Requerimiento 1	Descripción requerimiento 1
Requerimiento 2	Descripción requerimiento 2

Fuente: Elaboración propia, 2025.

**Figura A. 5**  
**Crear / Editar Tipo Solicitud**

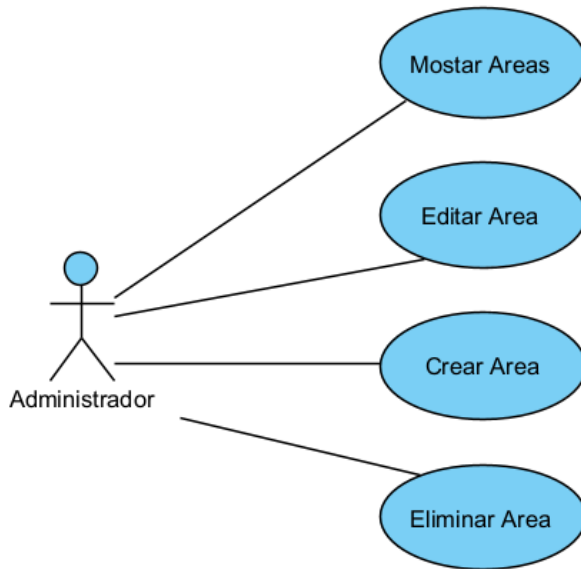
Admin >> Claro >> Configuraciones >> Tipo de Solicitudes >> 1 >> Editar

Categorías	Información Básica	Requisitos	SLA	Formularios	Pasos de ejecución	Guías
<p>Nombre tipo solicitud</p> <p>Tipo solicitud 1 <input checked="" type="checkbox"/> </p> <p>Tipo solicitud 2 <input checked="" type="checkbox"/> </p> <p>Tipo solicitud 3 <input checked="" type="checkbox"/> </p> <p>Tipo solicitud 4 <input checked="" type="checkbox"/> </p> <p>Tipo solicitud 5 <input checked="" type="checkbox"/> </p>	<p>Información general</p> <p>Nombre: <input type="text" value="Nombre Categoría"/></p> <p>Descripción: <input type="text" value="Descripción de cat..."/></p> <p><input type="checkbox"/> Activo</p> <p><input type="checkbox"/> Nuevos clientes</p> <p><input type="button" value="Crear Categoría"/></p>	<p>Requerimientos</p> <p>Requisitos</p> <p>+ Añadir requisitos</p> <p>+ Requisito 1</p> <p>+ Requisito 2</p> <p>+ Requisito 3</p> <p>+ Requisito 4</p> <p><input type="button" value="Crear Categoría"/></p>	<p>Acuerdo de Nivel de Servicio (SLA)</p> <p>Tiempo de Resolución (horas)</p> <p><input type="text" value="hrs"/></p> <p>Tiempo de Escalación (horas)</p> <p><input type="text" value="hrs"/></p> <p><input type="button" value="Crear Categoría"/></p>	<p>Formularios asociados</p> <p>Formularios</p> <p>+ Añadir formularios</p> <p>+ Formulario 1</p> <p>+ Formulario 2</p> <p>+ Formulario 3</p> <p>+ Formulario 4</p> <p><input type="button" value="Crear Categoría"/></p>	<p>Load Simulate [] Guardar</p> <p>Node Palette Drag nodes to the canvas</p> <p>Basic Blocks</p> <p>start end step</p> <p>condition loop</p> <p>Advanced Blocks</p> <p>subprocess task</p> <p>approval notification</p> <p>timer gateway</p> <p>message annotation</p>	<p>+ Agregar Guía</p>

Fuente: Elaboración propia, 2025.

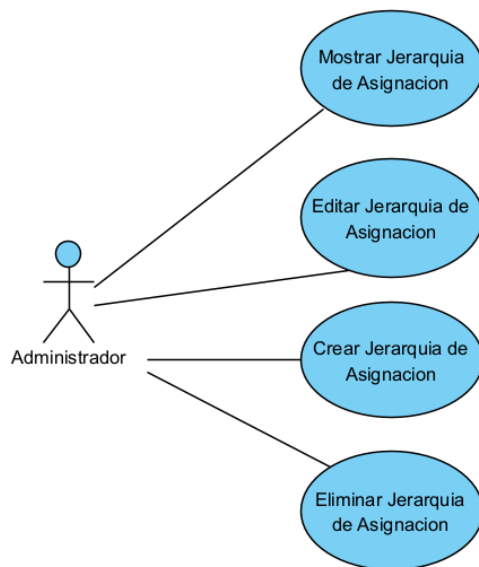
**Anexo B**  
Casos de uso

**Figura B. 1**  
Caso de uso áreas



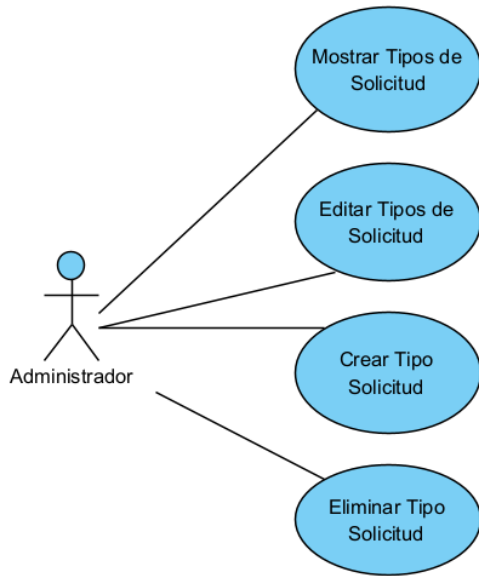
*Fuente: Elaboración propia, 2025.*

**Figura B. 2**  
Caso de uso jerarquía de asignación



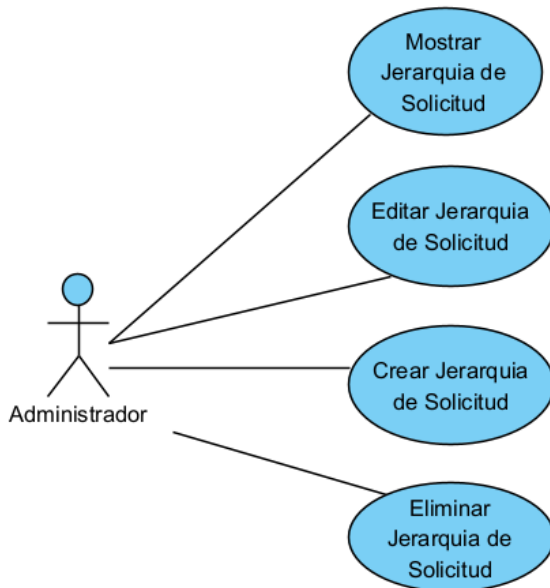
*Fuente: Elaboración propia, 2025.*

**Figura B. 3**  
Caso de uso tipos de solicitud



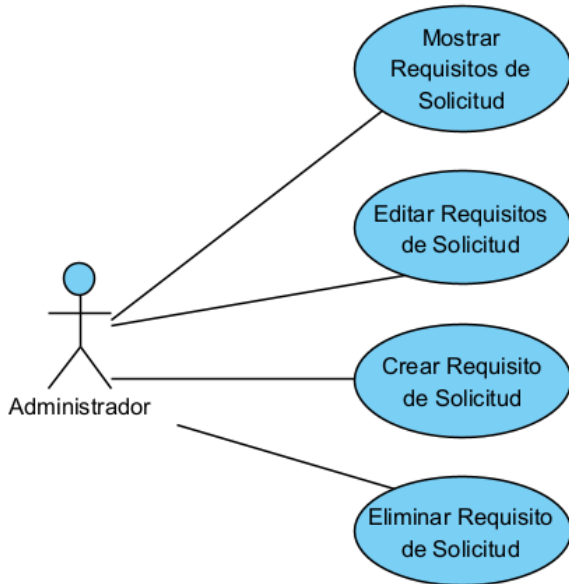
*Fuente: Elaboración propia, 2025.*

**Figura B. 4**  
Caso de uso jerarquía de solicitud



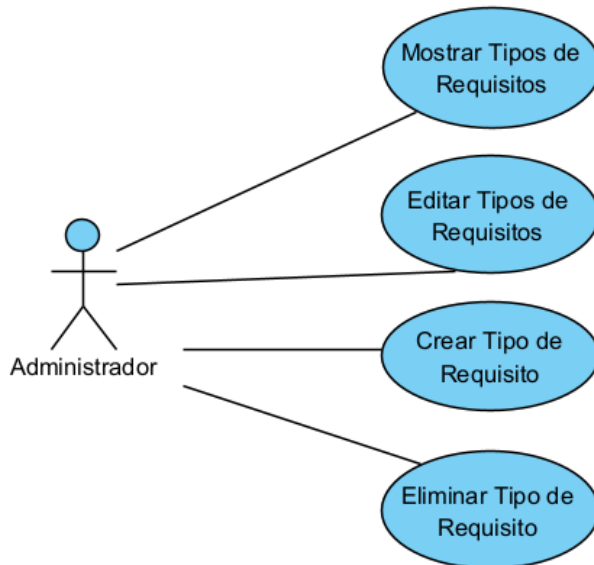
*Fuente: Elaboración propia, 2025.*

**Figura B. 5**  
Caso de uso requisitos de solicitud



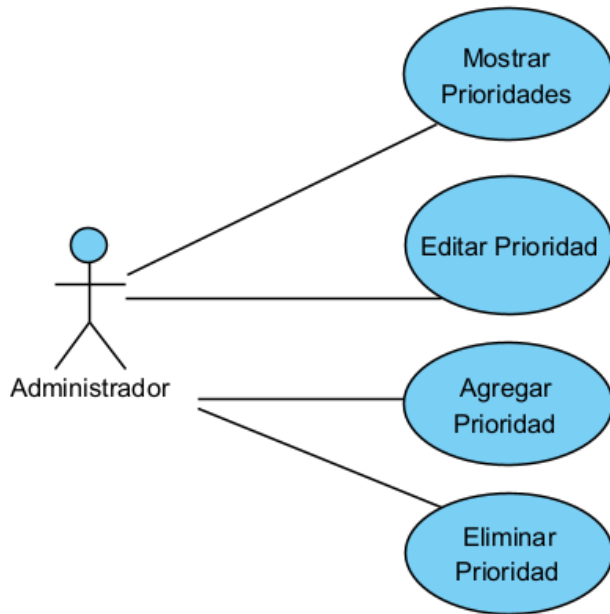
*Fuente: Elaboración propia, 2025.*

**Figura B. 6**  
Caso de uso tipos de requisitos de solicitud



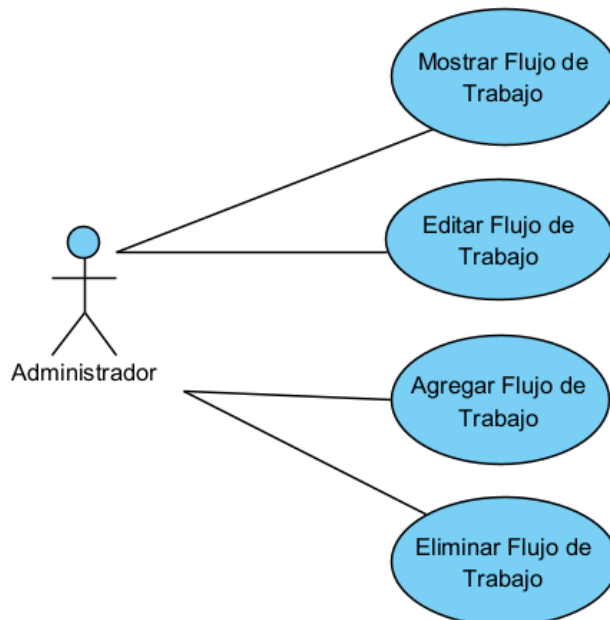
*Fuente: Elaboración propia, 2025.*

**Figura B. 7**  
Caso de uso prioridades



*Fuente: Elaboración propia, 2025.*

**Figura B. 8**  
Caso de uso flujos de trabajo



*Fuente: Elaboración propia, 2025.*

## Anexo C

Explicación de los casos de uso

### Tabla C. 1

Caso de uso crear formulario

<b>Identificador:</b>	CU-12
<b>Nombre:</b>	Crear Formulario
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	<p>El usuario puede crear un nuevo formulario dentro del sistema. A través de una interfaz gráfica, el usuario tiene la posibilidad de definir los datos básicos del formulario (como nombre, descripción y visibilidad pública), así como diseñar su estructura mediante un conjunto de elementos personalizables. El formulario puede ser guardado como borrador o publicado para ser utilizado en otros módulos del sistema.</p>
<b>Precondiciones:</b>	<ul style="list-style-type: none"><li>• El usuario debe haber iniciado sesión en el sistema.</li><li>• El usuario debe contar con permisos para crear formularios.</li></ul>
<b>Flujo Principal:</b>	<ol style="list-style-type: none"><li>1. El usuario accede al módulo "Diseñador de Formularios".</li><li>2. El usuario hace clic en el botón "Agregar Nuevo" (o "Nuevo").</li><li>3. El sistema despliega un modal con los siguientes campos:<ul style="list-style-type: none"><li>• Nombre: campo de texto obligatorio.</li><li>• Descripción: campo de texto opcional.</li><li>• Formulario Público: checkbox con texto explicativo: "Haz que este formulario sea público para que cualquier persona pueda verlo y enviar respuestas."</li></ul></li><li>4. El usuario completa los campos correspondientes.</li><li>5. El usuario hace clic en el botón "Guardar".</li><li>6. El sistema valida que el campo Nombre no esté vacío.</li></ol>

7. Si la validación es exitosa, el sistema guarda la información del nuevo formulario y lo agrega a la lista de formularios existentes.
8. El sistema cierra el modal y redirige al usuario a la vista de diseño del formulario.
9. El usuario puede seleccionar y arrastrar elementos para diseñar el formulario:
  - Elementos del formulario: Text Field, Number Field, TextArea Field, Date Field, Select Field, CheckBox Field.
  - Elementos de diseño: Title Field, Sub Title Field, Paragraph Field, Separator Field, Spacer Field.
10. Al insertar un elemento, el sistema habilita sus propiedades configurables (Nombre, Placeholder, Helper Text, Campo requerido).
11. El usuario puede realizar acciones finales:
  - Hacer clic en "Vista Previa".
  - Hacer clic en "Guardar" (para guardar cambios sin publicar).
  - Hacer clic en "Publicar" para habilitar el formulario.

**Postcondiciones:**

- El formulario queda registrado en el sistema y visible en la lista de formularios.
- El estado del formulario queda como "Activo" por defecto.

**Excepciones:**

- Si el campo "Nombre" no es completado, el sistema muestra un mensaje de error indicando que es obligatorio.
- Si hay un error al guardar la información en la base de datos, se muestra un mensaje de error y se invita a reintentar.

**Flujo Alternativo:**

Si el usuario cancela la operación antes de guardar, el sistema cierra el modal sin guardar ningún cambio.

**Tabla C. 2**

Caso de uso mostrar áreas

<b>Identificador:</b>	CU-14
<b>Nombre:</b>	Mostrar Áreas
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	<p>Este caso de uso permite al Administrador visualizar el listado de áreas disponibles en el sistema desde el módulo de Configuraciones. El sistema carga y muestra todas las áreas registradas; si no existen áreas, muestra un mensaje informativo indicando que no hay registros disponibles.</p>
<b>Precondiciones:</b>	<ul style="list-style-type: none"><li>• El administrador ha iniciado sesión en el sistema.</li><li>• El usuario tiene acceso al módulo de Configuraciones.</li></ul>
<b>Flujo Principal:</b>	<ol style="list-style-type: none"><li>1. El usuario accede al sistema.</li><li>2. El usuario despliega el botón de Configuraciones.</li><li>3. El sistema muestra un submenú con la opción Áreas.</li><li>4. El usuario selecciona la opción Áreas.</li><li>5. El sistema accede al módulo de Áreas.</li><li>6. El sistema consulta las áreas registradas en la base de datos.</li><li>7. El sistema muestra la lista de áreas disponibles al administrador.</li></ol>
<b>Postcondiciones:</b>	<p>El administrador puede visualizar todas las áreas actualmente registradas.</p>
<b>Excepciones:</b>	<p>Si ocurre un error en la conexión con la base de datos, el sistema muestra un mensaje de error: "Error al cargar las áreas. Por favor, intente nuevamente más tarde."</p>
<b>Flujo Alternativo:</b>	<ul style="list-style-type: none"><li>• No existen áreas registradas en el sistema.</li><li>• El sistema muestra el mensaje: "No hay áreas registradas actualmente. Por favor, agregue una nueva área desde el módulo correspondiente."</li></ul>

**Tabla C. 3**

Caso de uso mostrar tipos de solicitud

<b>Identificador:</b>	CU-22
<b>Nombre:</b>	Mostrar Tipos de Solicitud
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	Este caso de uso permite al Administrador visualizar la lista de Tipos de Solicitud registrados en el sistema, junto con sus características y opciones de gestión (ver, editar, eliminar)
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El administrador ha iniciado sesión en el sistema.</li> <li>• Existen uno o más Tipos de Solicitud registrados.</li> </ul>
<b>Flujo Principal:</b>	<ol style="list-style-type: none"> <li>1. El administrador ingresa al sistema.</li> <li>2. El administrador accede al menú “Configuraciones”.</li> <li>3. El administrador selecciona la opción “Tipos de Solicitud”.</li> <li>4. El sistema muestra una tabla con los Tipos de Solicitud disponibles, incluyendo los siguientes campos: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Descripción</li> <li>• Elegibles para nuevos clientes</li> <li>• Subcategorías</li> <li>• Formularios</li> <li>• Requisitos</li> <li>• Fecha de creación</li> <li>• Opciones de gestión: Ver, Editar, Eliminar</li> </ul> </li> </ol>
<b>Postcondiciones:</b>	El sistema muestra correctamente la lista de Tipos de Solicitud registrados y disponibles para su gestión.
<b>Excepciones:</b>	Si no existen Tipos de Solicitud registrados, el sistema muestra el mensaje: “No hay Tipos de Solicitud registrados actualmente.”
<b>Flujo Alternativo:</b>	<p>El sistema presenta un error al intentar obtener los datos:</p> <ul style="list-style-type: none"> <li>• Muestra un mensaje: “Error al cargar los Tipos de Solicitud. Intente nuevamente más tarde.”</li> </ul>

**Tabla C. 4**

Caso de uso mostrar requisitos de solicitud

<b>Identificador:</b>	CU-30
<b>Nombre:</b>	Mostrar Requisitos de Solicitud
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	
Permite al usuario visualizar en una tabla todos los requisitos de solicitud registrados en el sistema, incluyendo su información relevante y opciones para editar o eliminar cada uno.	
<b>Precondiciones:</b>	
<ul style="list-style-type: none"> <li>• El administrador ha iniciado sesión correctamente en el sistema.</li> <li>• Existen requisitos de solicitud registrados en la base de datos.</li> </ul>	
<b>Flujo Principal:</b>	
<ol style="list-style-type: none"> <li>1. El Administrador accede al menú principal del sistema.</li> <li>2. Hace clic en la opción "Configuración".</li> <li>3. El sistema muestra un submenú con varias opciones.</li> <li>4. El Administrador selecciona la opción "Requisitos".</li> <li>5. El sistema muestra una tabla con los requisitos de solicitud registrados, con las siguientes columnas: <ul style="list-style-type: none"> <li>○ Nombre</li> <li>○ Descripción</li> <li>○ Tipo de Requisito</li> <li>○ Creado (fecha de creación)</li> <li>○ Acciones (botones de Editar y Eliminar)</li> </ul> </li> </ol>	
<b>Postcondiciones:</b>	
El sistema presenta de forma clara y estructurada todos los requisitos existentes, listos para ser gestionados.	
<b>Excepciones:</b>	
<b>No hay requisitos registrados</b>	
<ul style="list-style-type: none"> <li>• El sistema muestra el mensaje: "No hay requisitos de solicitud registrados."</li> </ul>	
<b>Error de conexión o carga de datos</b>	
<ul style="list-style-type: none"> <li>• El sistema muestra el mensaje: "No se pudo cargar la información. Intente nuevamente o contacte al administrador."</li> </ul>	

**Flujo Alternativo:**

- La tabla debe permitir funciones de búsqueda, filtrado y ordenamiento por columnas (por ejemplo, por nombre o fecha).
- Si no existen requisitos registrados, el sistema muestra un mensaje informativo.

**Tabla C. 5**

Caso de uso mostrar tipos de requisitos

<b>Identificador:</b>	CU-34
<b>Nombre:</b>	Mostrar Tipos de requisitos
<b>Actor(es):</b>	Administrador
<b>Descripción:</b> Este caso de uso permite al administrador visualizar la lista de tipos de requisitos registrados en el sistema mediante una tabla interactiva.	
<b>Precondiciones:</b> <ul style="list-style-type: none"><li>• El usuario ha iniciado sesión correctamente en el sistema.</li><li>• El usuario tiene el rol de Administrador.</li></ul>	
<b>Flujo Principal:</b> <ol style="list-style-type: none"><li>1. El usuario ingresa al sistema.</li><li>2. El usuario hace clic en el botón "Configuración".</li><li>3. El sistema despliega un submenú.</li><li>4. El usuario selecciona la opción "Tipos de requisitos".</li><li>5. El sistema muestra una tabla con la lista de tipos de requisitos registrados.</li><li>6. La tabla incluye los campos: Nombre, Descripción, Fecha de creación, y las opciones "Ver" y "Eliminar".</li></ol>	
<b>Postcondiciones:</b> Se muestra correctamente la lista de tipos de requisitos disponibles en el sistema.	
<b>Excepciones:</b> <ul style="list-style-type: none"><li>• Si ocurre un error de conexión con la base de datos, el sistema muestra un mensaje de error.</li><li>• Si no existen tipos de requisitos registrados, se muestra un mensaje indicando "No hay registros disponibles".</li></ul>	

**Flujo Alternativo:**

Si el usuario no tiene permisos de administrador:

1. El sistema no muestra la opción "Tipos de requisitos" en el submenú.
2. El usuario no podrá acceder a esta funcionalidad.

**Tabla C. 6**

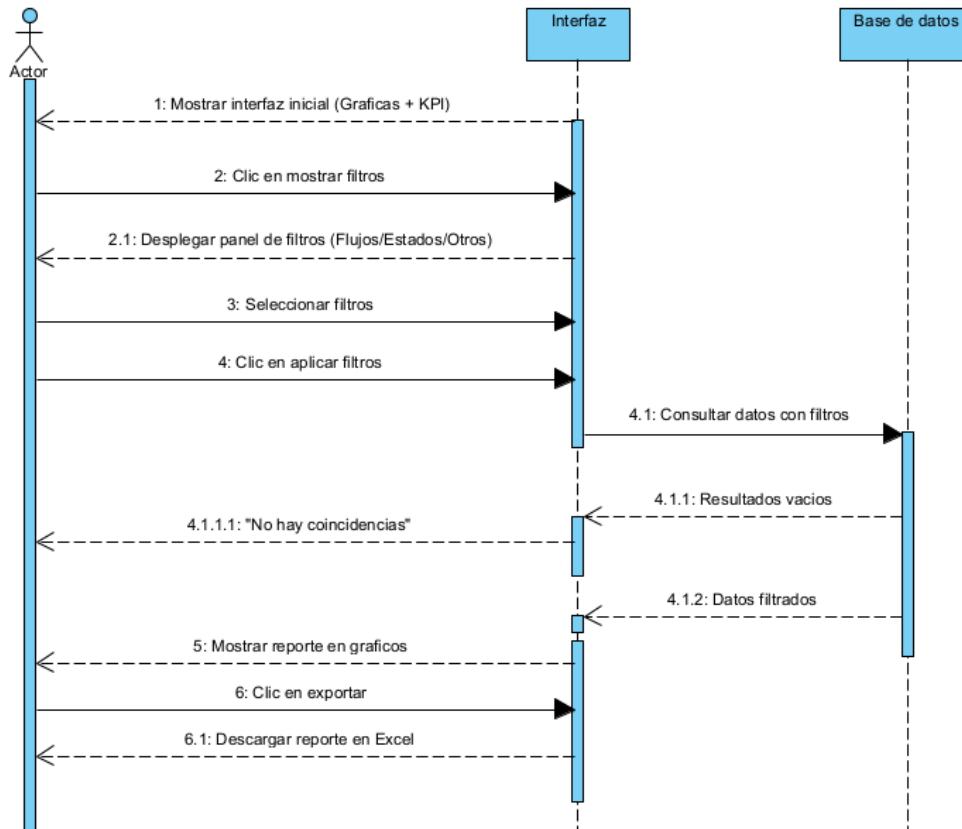
Caso de uso mostrar usuarios

<b>Identificador:</b>	CU-54
<b>Nombre:</b>	Mostrar Usuarios
<b>Actor(es):</b>	Administrador
<b>Descripción:</b>	
Permite al administrador visualizar la lista de usuarios registrados en el sistema.	
<b>Precondiciones:</b>	
<ul style="list-style-type: none"> <li>• El usuario ha iniciado sesión correctamente en el sistema.</li> <li>• El usuario tiene el rol de Administrador.</li> <li>• Existen usuarios registrados.</li> </ul>	
<b>Flujo Principal:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede al sistema.</li> <li>2. Hace clic en el botón "Seguridad y Acceso".</li> <li>3. El sistema despliega el submenú correspondiente.</li> <li>4. El usuario selecciona la opción "Usuarios".</li> <li>5. El sistema muestra una tabla con los usuarios registrados.</li> <li>6. La tabla incluye las siguientes columnas:</li> </ol>	
<b>Postcondiciones:</b>	
El usuario visualiza correctamente la lista actualizada de usuarios del sistema.	
<b>Excepciones:</b>	
<ul style="list-style-type: none"> <li>• Si ocurre un error al recuperar los datos, se muestra un mensaje de error.</li> <li>• Si no hay usuarios registrados, se muestra el mensaje "No hay registros disponibles".</li> </ul>	
<b>Flujo Alternativo:</b>	
El usuario no tiene permisos para ver la sección de usuarios:	
<ol style="list-style-type: none"> <li>1. El sistema oculta la opción del submenú "Usuarios".</li> <li>2. El acceso queda restringido.</li> </ol>	

## Anexo D

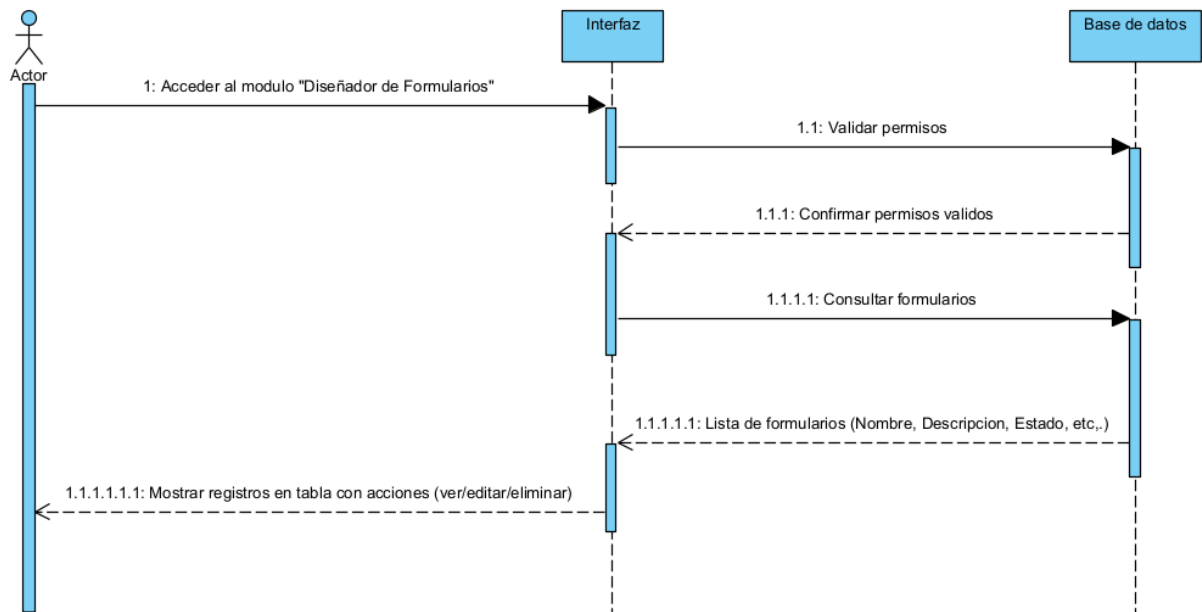
### Diagramas de Secuencias

**Figura D. 1**  
Diagrama de secuencia generar reporte de solicitudes



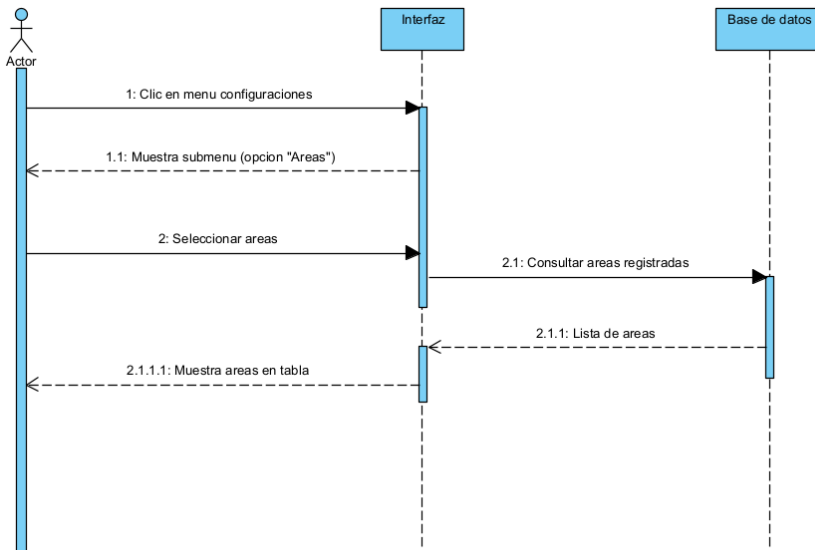
*Fuente: Elaboración propia, 2025.*

**Figura D. 2**  
Diagrama de secuencia mostrar formularios



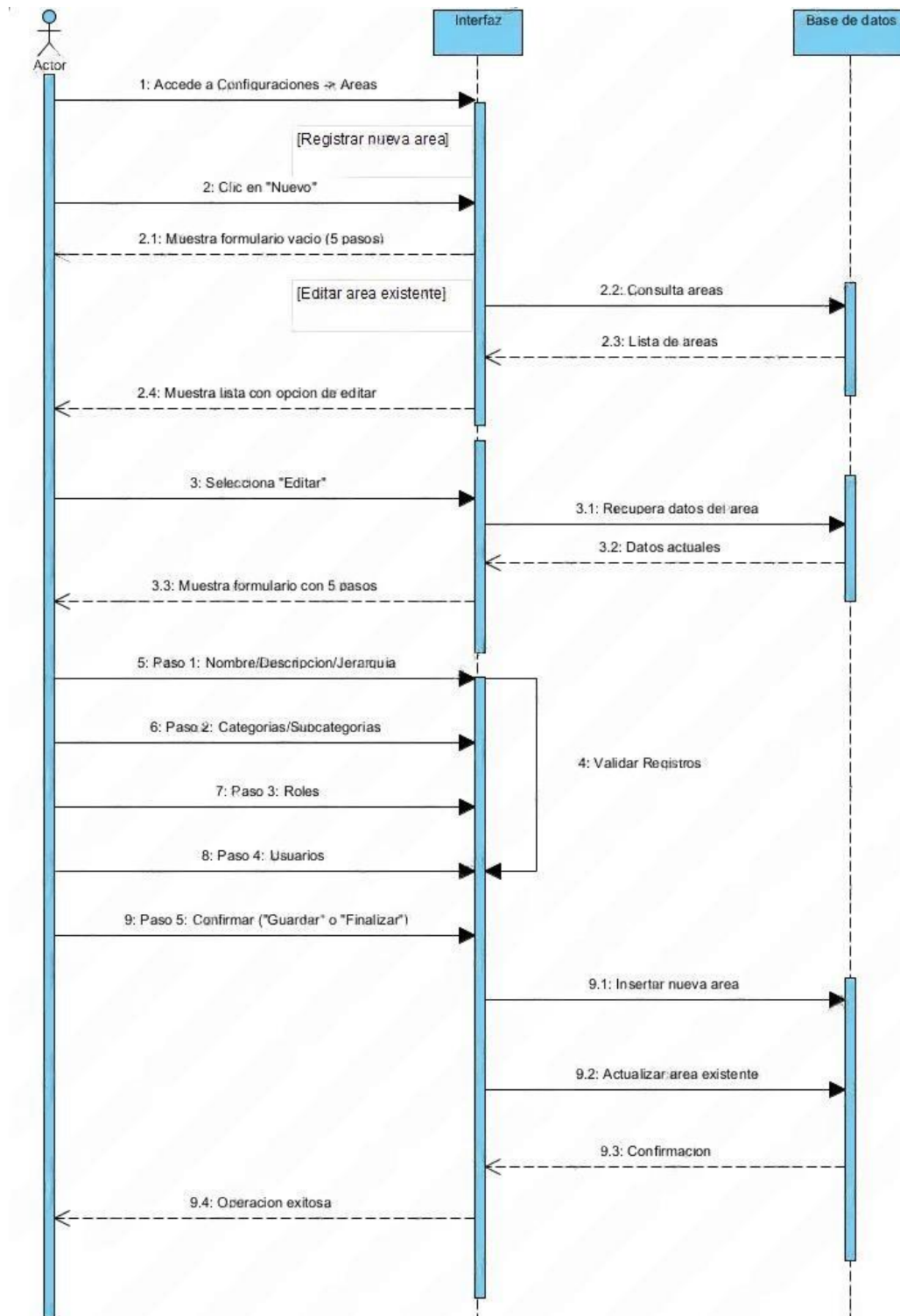
*Fuente: Elaboración propia, 2025.*

**Figura D. 3**  
Diagrama de secuencia mostrar áreas



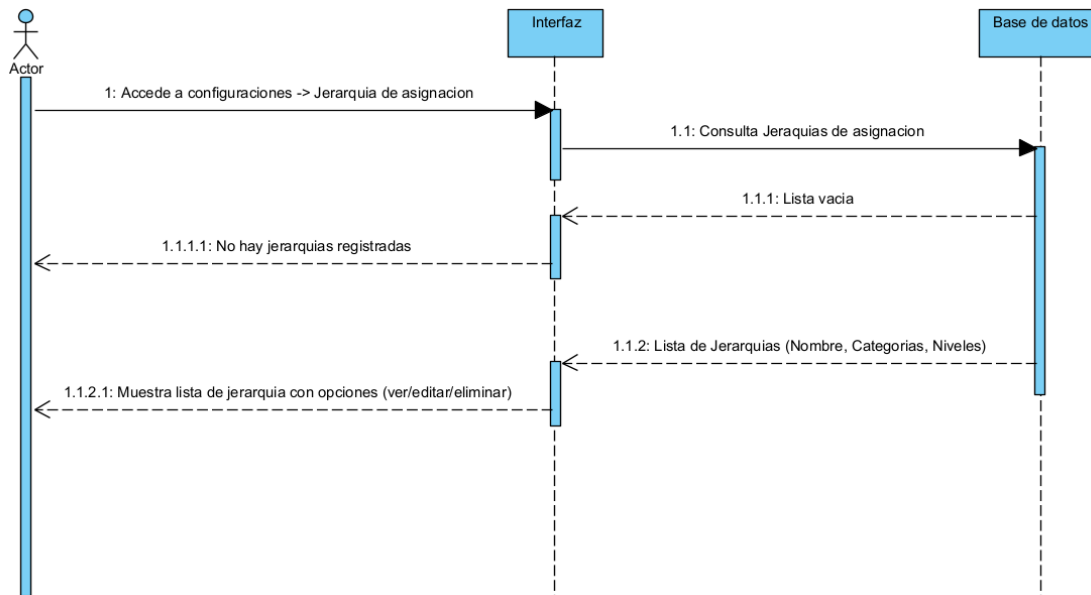
*Fuente: Elaboración propia, 2025.*

**Figura D. 4**  
Diagrama de secuencia agregar / editar área



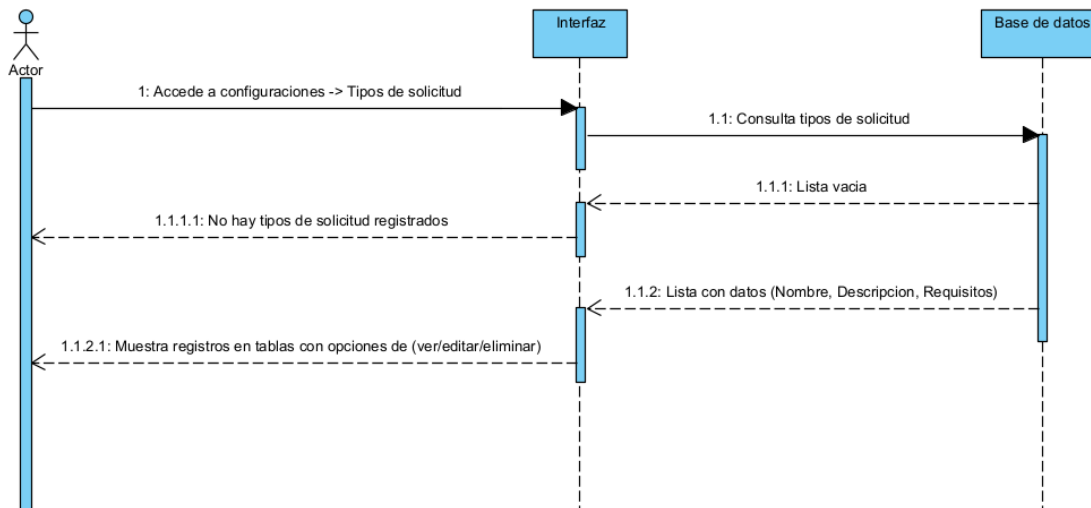
Fuente: Elaboración propia, 2025.

**Figura D. 5**  
Diagrama de secuencia mostrar jerarquía de asignación



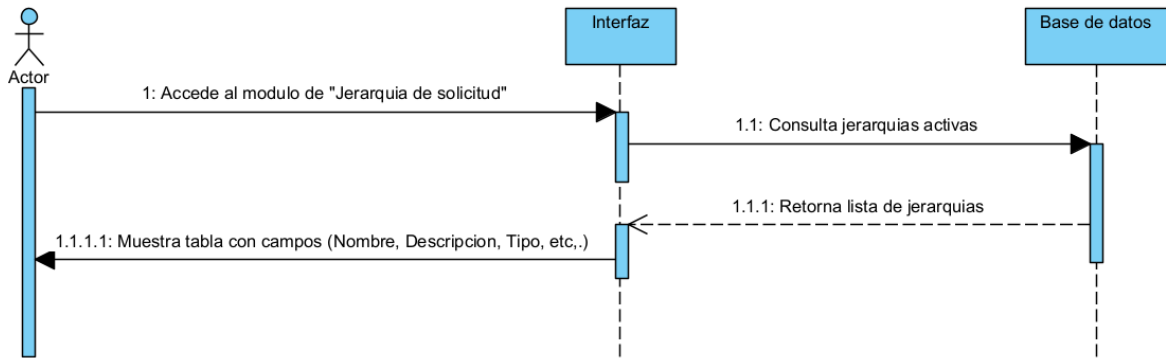
*Fuente: Elaboración propia, 2025.*

**Figura D. 6**  
Diagrama de secuencia mostrar tipos de solicitud



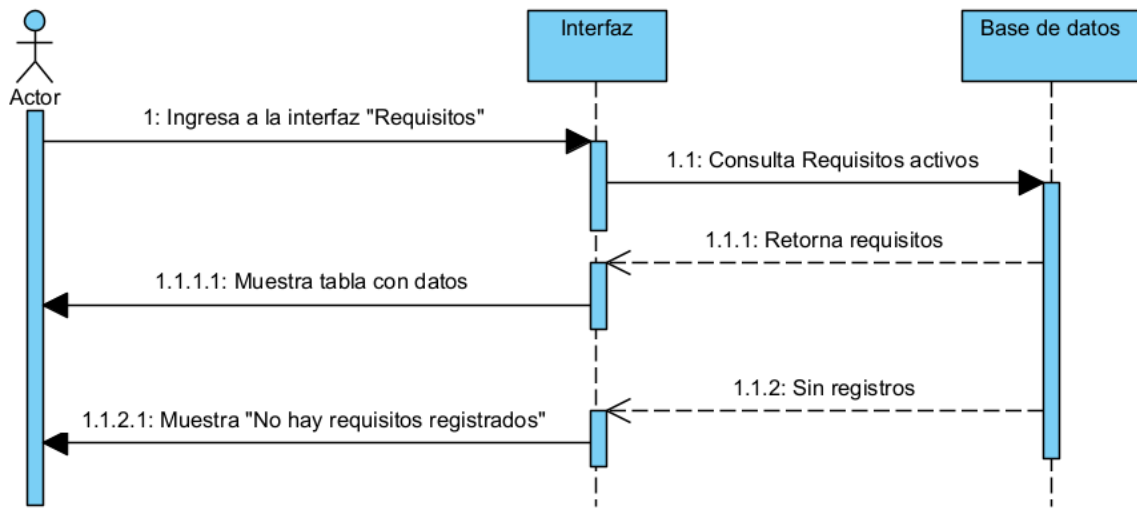
*Fuente: Elaboración propia, 2025.*

**Figura D. 7**  
Diagrama de secuencia mostrar jerarquía de solicitud



*Fuente: Elaboración propia, 2025.*

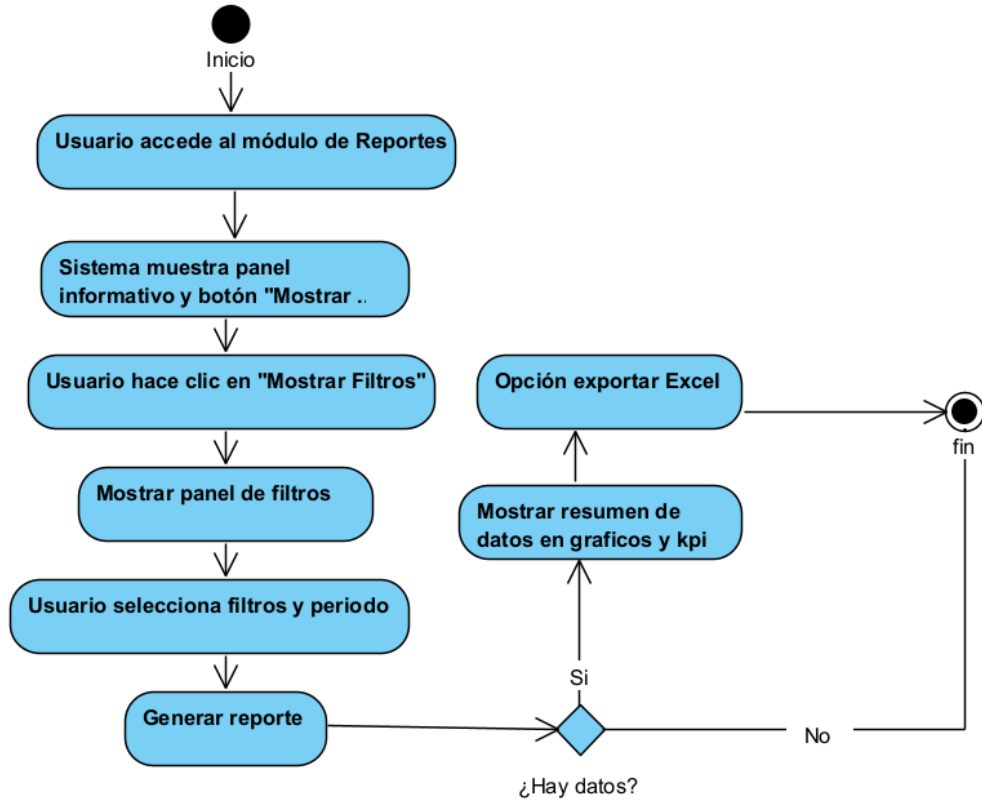
**Figura D. 8**  
Diagrama de secuencia mostrar requisitos de solicitud



*Fuente: Elaboración propia, 2025.*

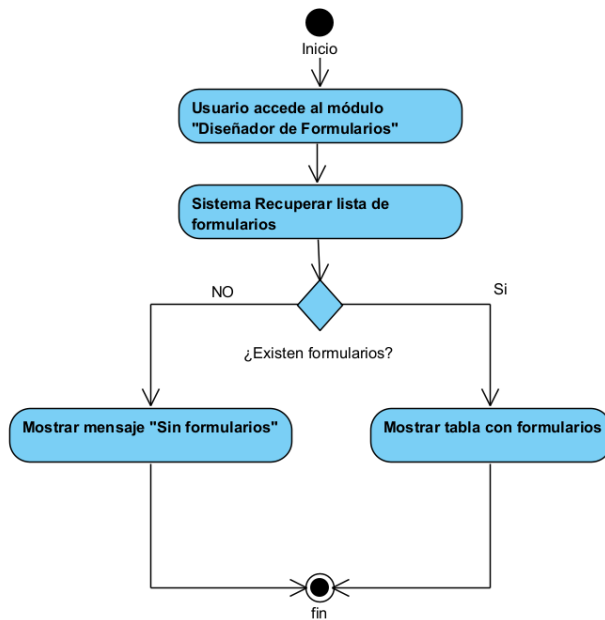
**Anexo E**  
Diagramas de Actividades

**Figura E. 1**  
Diagrama de actividad visualizar reportes



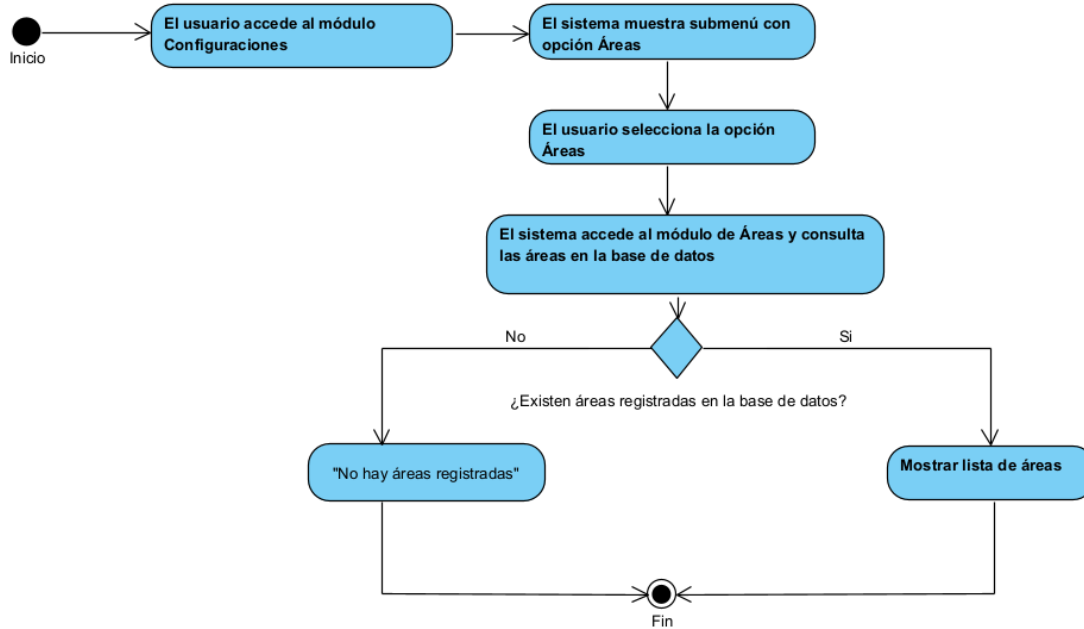
*Fuente: Elaboración propia, 2025.*

**Figura E. 2**  
Diagrama de actividad mostrar formularios



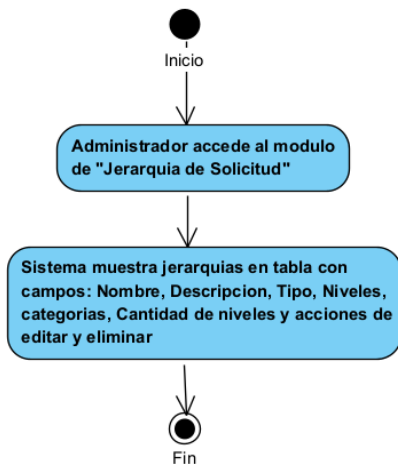
*Fuente: Elaboración propia, 2025.*

**Figura E. 3**  
Diagrama de actividad mostrar área



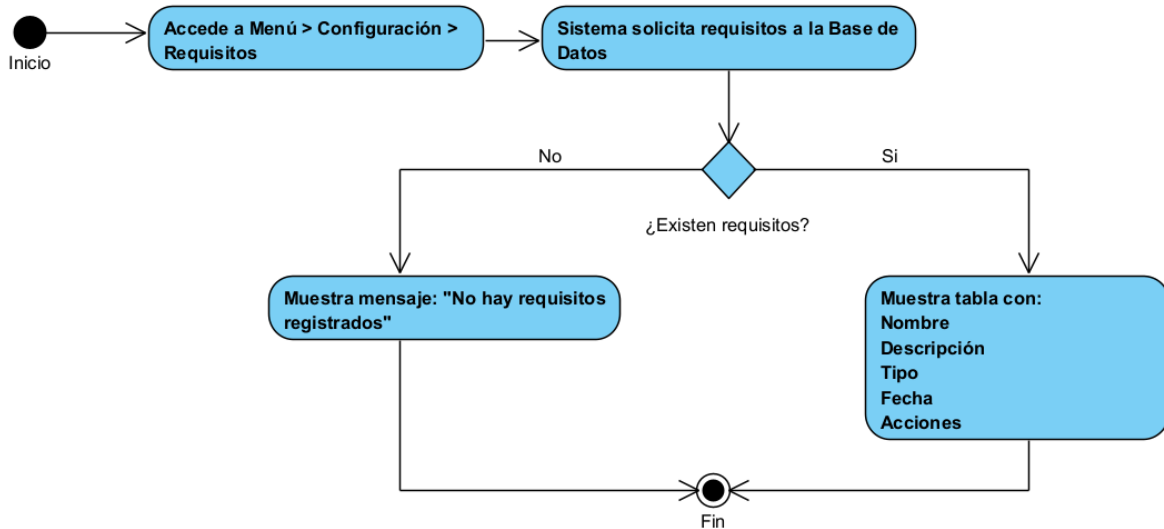
*Fuente: Elaboración propia, 2025.*

**Figura E. 4**  
Diagrama de actividad mostrar jerarquía de solicitud



*Fuente: Elaboración propia, 2025.*

**Figura E. 5**  
Diagrama de actividad mostrar requisitos de solicitud



*Fuente: Elaboración propia, 2025.*

## Anexo F

### Pruebas unitarias

**Tabla F. 1**  
Prueba unitaria permisos completos para administrador global

Identificador	UT-AUTH-002
Rol	Administrador
Descripción	Verificar que un usuario con el rol de administrador global (isGlobalAdmin: true) tenga todos los permisos, independientemente del permiso específico solicitado.
Prerrequisitos	La función <code>currentSession</code> está mockeada para devolver una sesión con un usuario <code>isGlobalAdmin: true</code> .
Pasos	1. Configurar <code>currentSession</code> para resolver con un objeto de sesión donde <code>user.isGlobalAdmin</code> es <code>true</code> . 2. Llamar a la función <code>getAuthContext</code> con un <code>mockTenantId</code> . 3. Verificar <code>auth.hasPermissions(['ANY_PERMISSION'])</code> y <code>auth.hasAreaPermissions('any-area', ['ANY_PERMISSION'])</code> .
Resultado Esperado	Ambas verificaciones de permisos ( <code>hasPermissions</code> y <code>hasAreaPermissions</code> ) deben devolver <code>true</code> .

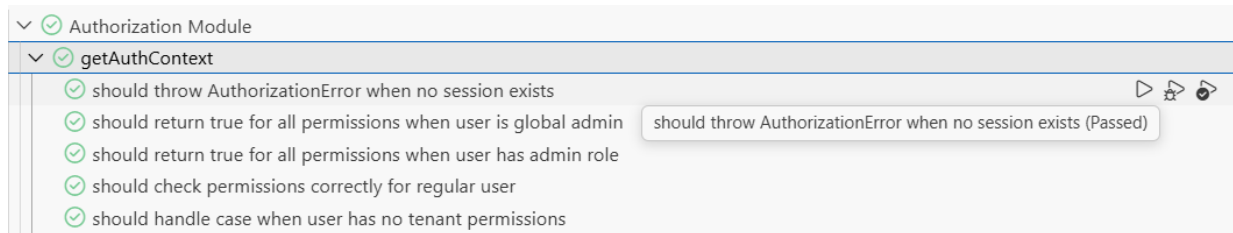
**Figura F. 1**  
Código de prueba unitaria permisos completos para administrador global

```
1 it('should return true for all permissions when user is global admin', async () => {
2   (currentSession as jest.Mock).mockResolvedValue({
3     user: {
4       id: mockUserId,
5       isGlobalAdmin: true,
6     },
7   });
8
9   const auth = await getAuthContext(mockTenantId);
10  expect(auth.hasPermissions(['ANY_PERMISSION'])).toBe(true);
11  expect(auth.hasAreaPermissions('any-area', ['ANY_PERMISSION'])).toBe(true);
12  });
```

*Fuente: Elaboración propia, 2025.*

## Figura F. 2

Resultado de la prueba unitaria permisos completos para administrador global



Fuente: *Elaboración propia, 2025.*

## Anexo G

### Pruebas de Sistema

Documentación de Pruebas E2E: Creación de Área, Formulario, Prioridad, Tipo de Requisito, Solicitud y Flujo de Trabajo

Este documento presenta los casos de prueba End-to-End (E2E) para las funcionalidades de creación de áreas, creación de formularios, creación de prioridades, gestión de tipos de requisitos, creación de solicitudes y creación de flujos de trabajo, detallando los criterios de aceptación, los pasos a seguir y los resultados esperados para cada prueba.

### Tabla G. 1

Pruebas de sistema criterios de aceptación para módulo "Área"

N.	Rol	Quiero ("Como...")	Para que ("... pueda ...")	Criterios de aceptación
1	Administrador	Crear una nueva área	Organizar las solicitudes y asignaciones dentro de una estructura lógica	La nueva área se crea exitosamente con todos los campos requeridos, y el usuario es redirigido a la página de listado de áreas con un mensaje de éxito.
2	Administrador	Validar campos requeridos al crear un área	Prevenir la creación de áreas con información incompleta	El sistema debe mostrar mensajes de validación claros para los campos obligatorios si se intenta

				enviar el formulario sin completarlos.
--	--	--	--	--

**Tabla G. 2**

Prueba de sistema crear nueva área con todos los campos requeridos

Identificador	E2E-AREA-001
Rol	Administrador
Descripción	Verificar que una nueva área se pueda crear exitosamente completando todos los campos requeridos y navegando a través del formulario de varios pasos.
Prerrequisitos	El usuario ha iniciado sesión y ha navegado a la URL de creación de una nueva área (/en/admin/{tenantId}/configurations/areas/new).
Pasos	<ol style="list-style-type: none"> <li>1. Esperar 2 segundos para que la página cargue completamente.</li> <li>2. Rellenar el campo de entrada de nombre con un nombre de área generado (testData.name).</li> <li>3. Si el selector de jerarquía no está deshabilitado, hacer clic en él y seleccionar la primera opción disponible.</li> <li>4. Rellenar el campo de texto de descripción con una descripción generada (testData.description).</li> <li>5. Hacer clic en el interruptor "Active" para alternar su estado.</li> <li>6. Hacer clic en el botón "Next" o "Continue" para avanzar al siguiente paso (Categoría de Asignación).</li> <li>7. Repetir 3 veces: Hacer clic en el botón "Next" o "Continue" y esperar 0.5 segundos para saltar los pasos de Categoría de Asignación, Rol y Usuario.</li> <li>8. En el paso final, hacer clic en el botón "Create", "Save" o "Finish" para enviar el formulario.</li> <li>9. Verificar que la URL cambie para incluir /admin/{tenantId}/configurations/areas.</li> </ol>

Resultado Esperado	La URL debe incluir /admin/{tenantId}/configurations/areas. La página debe mostrar el nombre del área recién creada (testData.name) y un mensaje de éxito (ej. "success").
--------------------	--

### Figura G. 1

Pruebas de sistema código para crear nueva área con todos los campos requeridos

```
it('should create a new area with all required fields', () => {
  cy.wait(2000);

  // Fill name
  cy.get('input[placeholder="Enter your name"], input[placeholder*="name" i]').first().type(testData.name);

  // Check if hierarchy field is enabled and interact with it if so
  cy.get('div')
    .contains('label', /hierarchy/i)
    .parent()
    .find('.css-11vi78b-control, .react-select__control')
    .then(($el: JQuery<HTMLElement>) => {
      // Check if the control is not disabled
      if (!$el.hasClass('is-disabled') && !$el.attr('aria-disabled')) {
        cy.wrap($el).click();
        cy.get('.css-bio7mv-option, .react-select__option').first().click();
      } else {
        // If disabled, just log that we're skipping it
        cy.log('Hierarchy field is disabled, skipping selection');
      }
    });

  // Fill description
  cy.get('textarea[placeholder*="description" i]').type(testData.description);
});
```

Fuente: Elaboración propia, 2025.

### Figura G. 2

Pruebas de sistema resultado de crear nueva área con todos los campos requeridos



Fuente: Elaboración propia, 2025.

### Tabla G. 3

Pruebas de sistema criterios de aceptación para pruebas E2E del módulo de creación de formulario

N.	Rol	Quiero ("Como...")	Para que ("... pueda ...")	Criterios de aceptación
1	Administrador	Crear un nuevo formulario	Diseñar formularios personalizados para diversas necesidades	El nuevo formulario se crea exitosamente, y el usuario es redirigido al constructor/editor de formularios con el nombre del formulario visible.

**Tabla G. 4**

Pruebas de sistema crear nuevo formulario y redirigir al constructor

Identificador	E2E-FORM-001
Rol	Administrador
Descripción	Verificar que se pueda crear un nuevo formulario exitosamente y que el usuario sea redirigido a la página del constructor/editor de formularios.
Prerrequisitos	El usuario ha iniciado sesión.
Pasos	<ol style="list-style-type: none"> <li>1. Visitar la URL de creación de un nuevo formulario (/admin/{tenantId}/form-designer/new).</li> <li>2. Rellenar el primer campo de entrada (nombre del formulario) con un nombre generado (testData.name).</li> <li>3. Rellenar el primer campo de texto (descripción del formulario) con una descripción generada (testData.description).</li> <li>4. Marcar la casilla de verificación "Is public form".</li> <li>5. Hacer clic en el botón "save".</li> </ol>
Resultado Esperado	La URL debe coincidir con el patrón /admin/{tenantId}/form-designer/.+/edit. El nombre del formulario recién creado (testData.name) debe ser visible en el encabezado del constructor de formularios.

### Figura G. 3

Código de la prueba de sistema crear nuevo formulario y redirigir al constructor

```
it('should create a new form and redirect to the form builder', () => {
  cy.visit(url);

  // Fill in the name
  cy.get('input').first().type(testData.name);

  // Fill in the description
  cy.get('textarea').first().type(testData.description);

  // Toggle the public form checkbox
  cy.contains('Is public form').parent().find('input[type="checkbox"]').check({ force: true });

  // Submit the form
  cy.get('button').contains(/save/i).should('be.visible').click();

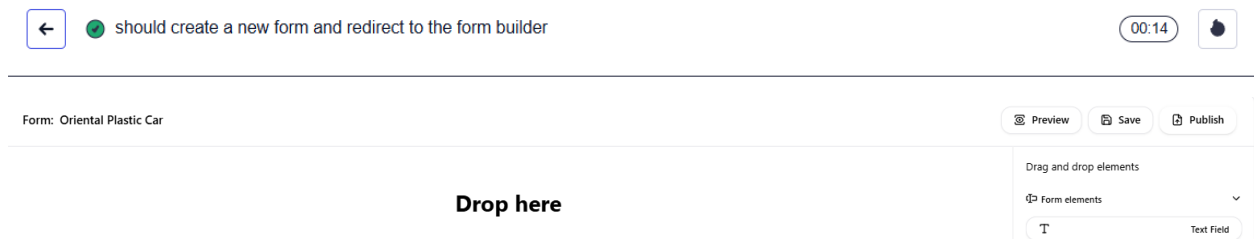
  // Should redirect to the form builder/editor
  cy.url({ timeout: 10000 }).should('match', new RegExp(`/admin/${tenantId}/form-designer/./edit`));

  // Check that the form name appears in the builder header
  cy.contains(testData.name).should('be.visible');
});
```

Fuente: Elaboración propia, 2025.

### Figura G. 4

Resultado de la prueba de sistema crear nuevo formulario y redirigir al constructor



### Tabla G. 5

Prueba de sistema crear nuevo tipo de requisito con todos los campos

Identificador	E2E-REQTYPE-001
Rol	Administrador
Descripción	Verificar que un nuevo tipo de requisito se pueda crear exitosamente completando todos los campos disponibles en el formulario.
Prerrequisitos	El usuario ha iniciado sesión y ha navegado a la URL de creación de un nuevo tipo de requisito (/en/admin/{tenantId}/configurations/requirement-types/new).
Pasos	1. Esperar 3 segundos para que la página cargue completamente.

	<ol style="list-style-type: none"> <li>2. Rellenar el campo de entrada de nombre con un nombre generado (<code>testData.name</code>).</li> <li>3. Rellenar el campo de texto de descripción con una descripción generada (<code>testData.description</code>).</li> <li>4. Hacer clic en el interruptor "Active" para alternar su estado.</li> <li>5. Hacer clic en el interruptor "Default" para alternar su estado.</li> <li>6. Hacer clic en el botón "Create" para enviar el formulario.</li> <li>7. Verificar que la URL cambie para incluir <code>/admin/{tenantId}/configurations/priorities</code> (según la lógica actual de la aplicación).</li> </ol>
Resultado Esperado	La URL debe incluir <code>/admin/{tenantId}/configurations/priorities</code> . La página debe mostrar un mensaje de éxito ("saved successfully").

### Figura G. 5

Código de la prueba de sistema crear nuevo tipo de requisito con todos los campos

```

it('should create a new requirement type with all fields', () => {
  // Debug: Check current URL
  cy.url().then((url) => {
    cy.log(`Current URL: ${url}`);
  });

  // Wait for page to load
  cy.wait(3000);

  // Debug: Check for form elements
  cy.get('body').then($body => {
    cy.log(`Form elements found: ${$body.find('form').length}`);
    cy.log(`Input elements found: ${$body.find('input').length}`);
  });

  // Fill requirement type name
  cy.get('input[placeholder="Enter the name"]').should('be.visible').type(testData.name);

  // Fill description
  cy.get('textarea[placeholder="Enter the description"]').should('be.visible').type(testData.description);

  // Toggle switches
  cy.contains('label', 'Active').parents('div.space-y-2').find('button[role="switch"]').click({ force: true });
  cy.contains('label', 'Default').parents('div.space-y-2').find('button[role="switch"]').click({ force: true });

  // Submit form
  cy.get('button')
    .contains(/create/i)
    .click();
}

```

Fuente: *Elaboración propia, 2025.*

## Figura G. 6

Resultado de la prueba de sistema crear nuevo tipo de requisito con todos los campos

The screenshot shows a Cypress test runner interface. At the top, the browser address bar displays 'create-area.cypress'. Below it, the test title is 'should create a new area with all required fields' with a green status indicator and a timer showing '00:24'. A dropdown menu is open, showing 'BEFORE ALL' with the following code:

```

1  Log Starting login process...
2  visit /en/auth/login -> 307: https://request-admin.vercel.app/auth/login
3  url
  
```

Fuente: Elaboración propia, 2025.

## Tabla G. 6

Prueba de sistema criterios de aceptación para pruebas E2E del módulo de Solicitudes

N.	Rol	Quiero ("Como...")	Para que ("... pueda ...")	Criterios de aceptación
1	Administrador	Crear una nueva solicitud completando el flujo	Registrar solicitudes correctamente en el sistema	El flujo avanza por todas las etapas y, al finalizar, la URL cambia y ya no es <code>/admin/{tenantId}/requests/new</code> .
2	Administrador	Completar la clasificación con selects dinámicos	Clasificar la solicitud antes de continuar	Los selectores cargan opciones válidas, permiten seleccionar sin "Loading..." y habilitan el botón Next.
3	Administrador	Validar clasificación obligatoria	Evitar solicitudes sin clasificación	Si no se completan los selects requeridos, no se puede continuar o se muestra validación.
4	Administrador	Seleccionar cumplimiento	Registrar el cumplimiento requerido	Al usar <code>Select all</code> , se seleccionan todos los ítems y se permite avanzar.
5	Administrador	Ingresar datos de solicitud	Guardar asunto y descripción	<code>issueSubject</code> y <code>description</code> aceptan texto y permiten avanzar.
6	Administrador	Seleccionar un valor adicional en detalles	Completar campos obligatorios del detalle	El selector muestra opciones y permite seleccionar una válida.

N.	Rol	Quiero ("Como...")	Para que ("... pueda ...")	Criterios de aceptación
7	Administrador	Avanzar en adjuntos sin archivos	No bloquear el flujo si no hay documentos	El paso "Attach Documents" permite continuar sin subir archivos.
8	Administrador	Completar el paso de formularios si aparece	Finalizar el flujo cuando hay formularios	Si el paso "Forms" existe, permite avanzar con Next/Finish.
9	Administrador	Finalizar el resumen	Confirmar la solicitud	El paso "Summary" se muestra y el botón Finish completa el proceso.
10	Administrador	Verificar creación exitosa	Confirmar que se creó la solicitud	La URL final es distinta de la URL de creación.

**Tabla G. 7**

Prueba de sistema crear una nueva solicitud

Campo	Detalle
Identificador	E2E-REQUEST-001
Rol	Administrador
Descripción	Verificar el flujo completo de creación de una solicitud desde la clasificación hasta el resumen y finalización.
Prerrequisitos	El usuario ha iniciado sesión y ha navegado a la URL de creación de solicitud (/admin/{tenantId}/requests/new). Debe existir data para los selects dinámicos y el cumplimiento.
Pasos	<ol style="list-style-type: none"> <li>1. Visitar la URL de creación de solicitud.</li> <li>2. Paso 1: Clasificación: <ol style="list-style-type: none"> <li>a. Verificar que existan selects dinámicos.</li> <li>b. Seleccionar opciones válidas en 6 selects (uno por uno).</li> <li>c. Hacer clic en Next cuando esté habilitado.</li> </ol> </li> <li>3. Paso 2: Compliance: <ol style="list-style-type: none"> <li>a. Verificar que existe la sección "Compliance".</li> <li>b. Hacer clic en <input type="button" value="Select all"/>.</li> <li>c. Hacer clic en Next.</li> </ol> </li> <li>4. Paso 3: Details: <ol style="list-style-type: none"> <li>a. Verificar "Request Information".</li> <li>b. Llenar <input type="text" value="issueSubject"/> y <input type="text" value="description"/>.</li> <li>c. Seleccionar opción del selector adicional.</li> </ol> </li> </ol>

Campo	Detalle
	d. Hacer clic en Next. 5. Paso 4: Attachments: a. Verificar “Attach Documents”. b. Hacer clic en Next. 6. Paso 5: Forms (si aparece): a. Verificar “Forms”. b. Hacer clic en Next/Finish. 7. Paso 6: Summary: a. Verificar “Summary”. b. Hacer clic en Finish.
Resultado Esperado	La URL final no es <code>/admin/{tenantId}/requests/new</code> , indicando que la solicitud se creó correctamente.

**Figura G. 7**  
Código de prueba de sistema crear nueva solicitud

```

1
2 // Step 1: Classification - dynamic selects
3 cy.get('[id$="-form-item"] .css-11vi78b-control').should('exist');
4
5 const selectOption = (index) => {
6   cy.get('[id$="-form-item"] .css-11vi78b-control').filter(':not([aria-disabled="true"])').eq(index).should('be.visible').click();
7   cy.get('.css-100ebu3-menu', { timeout: 10000 }).should('be.visible');
8   cy.get('.css-100ebu3-menu', { timeout: 10000 }).should('not.contain', 'Loading...');
9   cy.get('.css-100ebu3-menu .css-bio7mv-option:not([aria-disabled="true"])', { timeout: 10000 }).should('exist');
10  cy.get('.css-100ebu3-menu').within(() => {
11    cy.get('.css-bio7mv-option').not('[aria-disabled="true"]').first().click();
12  });
13 };
14
15 for (let i = 0; i < 6; i++) {
16   selectOption(i);
17   cy.wait(1000);
18 }
19
20 cy.get('button').contains(/next/i).should('be.visible').should('not.be.disabled').click();
21
22 // Step 2: Compliance
23 cy.contains('Compliance').should('exist');
24 cy.get('button')
25   .contains(/select all/i)
26   .click({ force: true });
27 cy.get('button').contains(/next/i).click();
28
29 // Step 3: Details
30 cy.contains('Request Information').should('exist');
31 cy.get('input[name="issueSubject"]').type(testData.issueSubject);
32 cy.get('textarea[name="description"]').type(testData.description);
33

```

Fuente: Elaboración propia, 2025.

**Figura G. 8**  
Resultado de prueba de sistema crear flujo de trabajo completo con nodos conectados



Fuente: Elaboración propia, 2025.

## Anexo H

### Carta de aprobación de Claro-Nicaragua

Managua, 7 de octubre de 2025

Estimados Br. Danilo Acevedo y Br. Jesús Hernández.

Me dirijo a ustedes en mi calidad como Subgerente de Operaciones Comerciales de la Empresa Nicaraguense de Telecomunicaciones, ENITEL – Claro, con el propósito de notificarle formalmente la **aceptación y validación del proyecto de software para apoyo a gestión operativo**, denominado **Sistema Web para Control de Solicitudes de Servicios de Telecomunicaciones**, Gerencia de Operaciones. Nos complace informarles que, tras la revisión de los módulos presentados que se describen en la tesis monográfica, son suficientes para cumplir con la necesidad operacional del área de Fábrica Corporativa, de la Gerencia de Operaciones Comerciales.



MBA. Ing. Israel Antonio Treminio Ruiz

Subgerente de Operaciones Comerciales

Israel.treminio@claro.com.ni



## Anexo I

### Instrumento de recolección de datos: Entrevista a usuarios claves



#### Empresa de Telecomunicaciones Claro Nicaragua

#### Gerencia de Operaciones Comerciales

Nombre del entrevistado: \_\_\_\_\_

Cargo: \_\_\_\_\_

Fecha: \_\_\_\_\_

#### Objetivo de la entrevista:

Recopilar información sobre las necesidades funcionales del sistema de gestión de solicitudes para la atención de servicios de telecomunicaciones, con el fin de desarrollar una solución eficiente, escalable y alineada con los procesos internos de la organización.

#### Preguntas de la entrevista

1. ¿Cuál es el proceso actual que siguen los usuarios o clientes internos para solicitar servicios técnicos o administrativos?
2. ¿Qué funcionalidades considera prioritarias en un sistema que gestione estas solicitudes?
3. ¿Quién se encarga de gestionar o asignar las solicitudes una vez recibidas?
4. ¿Qué estados puede tener una solicitud a lo largo de su ciclo de vida?
5. ¿Es necesario que el sistema envíe notificaciones? ¿Qué tipo de eventos deberían generarlas?
6. ¿Quiénes deben tener acceso para visualizar las solicitudes y en qué nivel?
7. ¿Considera importante que se registre el tiempo de atención de las solicitudes? ¿Por qué?
8. ¿Qué niveles de prioridad deberían poder asignarse a una solicitud?
9. En caso de que una solicitud haya sido asignada incorrectamente, ¿el sistema debe permitir redireccionarla a otra área?
10. ¿Necesitan generar reportes de las solicitudes? ¿Con qué tipo de información y en qué formato?
11. ¿Qué otras observaciones o sugerencias consideran relevantes para mejorar el proceso de gestión de solicitudes?

**Anexo J**  
Cronograma de actividades

**CRONOGRAMA DE ACTIVIDADES**

