

**UNIVERSIDAD DE PANAMÁ**

**FACULTAD DE INFORMÁTICA, ELECTRÓNICA Y COMUNICACIÓN**

**ESCUELA DE INFORMÁTICA**

**LIC. INGENIERÍA EN INFORMÁTICA**

**PARTICIPANTES:**

*David Chalmers | 8-915-761 Daglio Filos |* *8-920-623*

**ASIGNATURA:**

*Análisis y Diseño*

**TRABAJO:**

*Entregable - Parte III*

*Proyecto Final*

**PROFESOR:**

*Saulo Aizprúa A.*

**FECHA DE ENTREGA:**

*03 de diciembre del 2024*

**Índice**

[Introducción 6](#_Toc184066183)

[PARTE I 7](#_Toc184066184)

[Definición del problema 7](#_Toc184066185)

[Objetivo General 8](#_Toc184066186)

[Objetivos Específicos 8](#_Toc184066187)

[Descripción del proyecto 9](#_Toc184066188)

[Alcance o delimitación 10](#_Toc184066189)

[Limitaciones y restricciones 11](#_Toc184066190)

[Requisitos o especificaciones 13](#_Toc184066191)

[Metodología A Utilizar (2 Pagina) Agil, Scrum, Xp, Jira Software … 16](#_Toc184066192)

[Sistema de Propuesto 18](#_Toc184066193)

[Justificación del Estudio 21](#_Toc184066194)

[A quién beneficia 21](#_Toc184066195)

[Importancia del Proyecto 21](#_Toc184066197)

[Retorno y Costos-Beneficios 21](#_Toc184066199)

[Definición de procesos 22](#_Toc184066201)

[Diagrama de caso de Uso (De todos los requerimientos de su proyecto) 26](#_Toc184066202)

[Diagrama de Secuencia (De todos los requerimientos de su proyecto) 57](#_Toc184066203)

[Parte II 84](#_Toc184066206)

[Diagrama actividad (igual a la cantidad de dcu) 84](#_Toc184066207)

[Diagrama de componente (modulo o submodulos o subsitemas o componente interno o externo) 108](#_Toc184066208)

[Diagrama clases () y sus descripciones 110](#_Toc184066211)

[Diagrama de navegación (como se mueve los usuarios por la diferente interface e/s) 113](#_Toc184066212)

[Diagrama de entidad relación (son las relaciones de todas la entidades de la base de datos) 113](#_Toc184066213)

[Diagrama de estructura de datos (data model con sus relaciones) 114](#_Toc184066214)

[Diagrama de arquitectura de la aplicación (es el diagrama de donde reside el sistema de información diseñado) 115](#_Toc184066215)

[Diseño de las interfaces usando una herramientas ui o ux (de todo el proyecto) 116](#_Toc184066216)

[Parte III 119](#_Toc184066217)

[Linea grafica del proyecto – logo tipo de sus software 119](#_Toc184066218)

[Creación de su base de datos y todos sus DDL y un backup de la base de datos (La base de dato que usted o su equipo seleccionó para implementar sus proyecto) 120](#_Toc184066219)

[Programación de los requerimientos del proyectos (Los requerimientos modelados en cada diagrama de casos de uso) 128](#_Toc184066220)

[Programación de las interfaces de entradas y procesos(transacción DML BD o en la apps). DML (Insert, delete, update, select) 138](#_Toc184066221)

[Programación de las interfaces de Salidas los reportes deben de exportarse en formato: xls, txt y pdf 144](#_Toc184066222)

[Casos de Pruebas del Productos desarrollado (Probar las funcionabilidad e integración de sistemas). Tipos de pruebas las documentas los tipos pruebas que le hiciste, debe se seleccionar los datos pruebas. Determiné que la prueba satisfactorias (Certificación de pruebas ++) 146](#_Toc184066223)

[Conclusión 150](#_Toc184066224)

[Referencias 151](#_Toc184066225)

**Índice de ilustración**

[-Figura 1: procesos 22](#_Toc184066992)

[-Figura 2: Gestión de Accesibilidad 23](#_Toc184066993)

[-Figura 3: Análisis de tendencia 24](#_Toc184066994)

[-Figura 4: Indicentes Registrados 24](#_Toc184066995)

[-Figura 5: Registro y seguimiento de solicitudes de servicio 25](#_Toc184066996)

[- Figura 6: interacción con la gestion de generación de Ticket 26](#_Toc184066997)

[-Figura 7: Administración de usuario de una mesa de servicio 28](#_Toc184066998)

[-Figura 8: Generación de reportes y analisis 30](#_Toc184066999)

[-Figura 9: Manejo de prioridades 32](#_Toc184067000)

[-Figura 10: Evaluación de satisfacción de usuario 34](#_Toc184067001)

[-Figura 11: Seguimiento de Ticket 36](#_Toc184067002)

[-Figura 12: Gestión para el escalamiento de Ticket 37](#_Toc184067003)

[-Figura 13: Gestión de Acuerdos de nivel de servicio SLA 39](#_Toc184067004)

[-Figura 14: Gestión de reportes de rendimiento 40](#_Toc184067005)

[-Figura 15: Integración con otros sistemas 42](#_Toc184067006)

[-Figura 16: Gestión de Conocimiento 44](#_Toc184067007)

[-Figura 17: Gestion de notificaciones 46](#_Toc184067008)

[-Figura 18: Gestión de seguridad y privacidad 48](#_Toc184067009)

[-Figura 19: Gestion de recursos humanos para la mesa 50](#_Toc184067010)

[-Figura 20: Analisis de Desempeño del equipo 52](#_Toc184067011)

[-Figura 21: Seguimiento de Ticket 57](#_Toc184067012)

[-Figura 22: Gestión de Ticket 58](#_Toc184067013)

[-Figura 23: Proceso de Creación y Gestión de Ticket 60](#_Toc184067014)

[-Figura 24: Manejo de Prioridades 62](#_Toc184067015)

[-Figura 25: Evaluación de Satisfacción del usuario 64](#_Toc184067016)

[-Figura 26: Evaluación de Satisfacción del usuario 66](#_Toc184067017)

[-Figura 27: Manejo de Solicitudes 68](#_Toc184067018)

[-Figura 28: Escalamiento de Tickets 69](#_Toc184067019)

[-Figura 29: Monitorización de Sla 71](#_Toc184067020)

[-Figura 30: Gestión de Ticket 72](#_Toc184067021)

[-Figura 31: Integración con sistemas de gestión activos 74](#_Toc184067022)

[-Figura 32: Configuración y Notificación de incidentes 76](#_Toc184067023)

[-Figura 33: Gestión de Seguridad y Acceso 78](#_Toc184067024)

[-Figura 34: Gestión de Horarios y Carga de Trabajo 80](#_Toc184067025)

[-Figura 35: Gestión de Métrica y Estadísticas 82](#_Toc184067026)

[-Figura 36. Gestion de solicitudes 84](#_Toc184067027)

[-Figura 37: Gestion de problemas 85](#_Toc184067028)

[-Figura 38: Gestion de incidentes 86](#_Toc184067029)

[-Figura 39: Evaluación de satisfacción 87](#_Toc184067030)

[-Figura 40: Seguimiento de Ticket 89](#_Toc184067031)

[-Figura 41: Escalamiento de Ticket 90](#_Toc184067032)

[-Figura 42: Monitorización y gestion de SLA 91](#_Toc184067033)

[-Figura 43: Generacion de reportes 92](#_Toc184067034)

[-Integracion con otros sistemas 93](#_Toc184067035)

[-Figura 45: Gestion de conocimientos 94](#_Toc184067036)

[-Figura 46: Gestion de notificación 95](#_Toc184067037)

[-Figura 47: Gestion de Seguridad y Privacidad 96](#_Toc184067038)

[-Figura 48: Recursos Humanos para la mesa de servicio 97](#_Toc184067039)

[-Figura 49: Analisis de desempeño del equipo 98](#_Toc184067040)

[-Figura 50: Gestión de incidentes 100](#_Toc184067041)

-[Figura 51: Diagrama de colaboración 101](#_Toc184067042)

[-Figura 52: Colaboracion Gestion de tickets 102](#_Toc184067043)

[-Figura 53: Sistema de gestiones 102](#_Toc184067044)

[-Figura 54: Manejo de prioridades en el sistema de ticket 103](#_Toc184067045)

[-Figura 55: Evaluacion de satisfaccion del usuario 103](#_Toc184067046)

[-Figura 56: manejo de Solicitud 104](#_Toc184067047)

[-Figura 57: Manejo de Solicitudes 104](#_Toc184067048)

[-Figura 58: Diagrama de Componente 108](#_Toc184067049)

# **Introducción**

En el entorno actual de las organizaciones, donde la tecnología juega un rol fundamental en la operación diaria, la gestión eficiente de una mesa de servicio se ha convertido en una necesidad crítica para garantizar la continuidad y calidad de los servicios prestados. La mesa de servicio actúa como el principal punto de contacto entre los usuarios y el equipo de soporte técnico, resolviendo incidencias, gestionando solicitudes y asegurando la satisfacción del cliente. Sin embargo, la creciente complejidad de los sistemas y el volumen de solicitudes requieren soluciones avanzadas para gestionar de manera óptima estos procesos.

Este proyecto tiene como objetivo el desarrollo e implementación de un Sistema de Análisis y Gestión para la Administración de una Mesa de Servicio, que permita centralizar la información, automatizar procesos y proporcionar herramientas de análisis en tiempo real. Este sistema busca no solo optimizar la resolución de incidencias, sino también generar reportes y estadísticas clave para la toma de decisiones estratégicas, mejorando así la eficiencia operativa y la experiencia de los usuarios.

Mediante la integración de tecnologías modernas y buenas prácticas en la gestión de servicios de TI (ITSM), el sistema propuesto proporcionará una solución robusta y escalable, adaptable a las necesidades actuales y futuras de la organización. Esto contribuirá significativamente a reducir tiempos de respuesta, mejorar la asignación de recursos y elevar los estándares de calidad en la prestación de soporte.

# **PARTE I**

# **Definición del problema**

La definición del problema para un sistema de análisis y gestión para la administración de una mesa de servicio radica en la ineficiente gestión de solicitudes e incidencias que impacta negativamente en la calidad del servicio y en la satisfacción de los usuarios. En muchas organizaciones, la mesa de servicio enfrenta dificultades para manejar un creciente volumen de tickets de manera ágil y efectiva, debido a la falta de herramientas adecuadas que permitan automatizar procesos, realizar un seguimiento detallado de cada solicitud, y generar reportes en tiempo real.

La ausencia de un sistema centralizado para gestionar la mesa de servicio impide la correcta priorización y asignación de recursos, lo que se traduce en retrasos en la atención de incidencias críticas y una baja capacidad de respuesta. Además, la falta de datos precisos y análisis en tiempo real limita la toma de decisiones estratégicas y la optimización del servicio a largo plazo. Sin un sistema adecuado, la organización no puede evaluar de manera efectiva el rendimiento del equipo de soporte ni identificar áreas de mejora clave, lo que perpetúa la ineficiencia y afecta los objetivos operativos.

Por tanto, el problema central es la necesidad de un sistema que integre la automatización, el análisis de datos y la generación de reportes para mejorar la eficiencia en la gestión de la mesa de servicio, optimizar la asignación de recursos y elevar los niveles de satisfacción de los usuarios. (Indeed, 2022)

# **Objetivo General**

Analizar, y diseñar sistemas para la administración de una mesa de servicio con la finalidad de mejorar los servicios y atención brindada a sus clientes.

# **Objetivos Específicos**

Los objetivos específicos para el proyecto de análisis y diseño de sistemas para la administración serían los siguientes:

1. **Identificar** las necesidades y requerimientos funcionales de la mesa de servicio, evaluando los principales puntos críticos en la atención y gestión de solicitudes para definir áreas de mejora clave.
2. **Especificar** los requerimientos solicitados para establecer un plan de desarrollo.
3. **Desarrollar** un sistema automatizado de gestión de incidencias y solicitudes, que permita el registro, seguimiento y resolución eficiente de tickets, optimizando tiempos de respuesta y asignación de recursos.
4. **Implementar** herramientas de análisis y generación de reportes en tiempo real, que faciliten la evaluación del rendimiento del equipo de soporte y proporcionen métricas clave para la toma de decisiones estratégicas.
5. **Diseñar** una interfaz de usuario intuitiva y accesible para los operadores de la mesa de servicio, que simplifique la gestión de tareas y la interacción con los clientes.
6. **Establecer** mecanismos de priorización y escalamiento de solicitudes, asegurando que las incidencias críticas reciban atención inmediata y que los recursos se distribuyan de manera efectiva.
7. **Integrar** un sistema de retroalimentación para los clientes, que permita evaluar la satisfacción del servicio recibido y aplicar mejoras continuas en base a los comentarios obtenidos.
8. **Capacitar** al personal de la mesa de servicio en el uso del nuevo sistema, garantizando una adopción exitosa y una utilización eficiente de las nuevas herramientas y funcionalidades. (Desk, 2023)

# **Descripción del proyecto**

En el entorno actual, donde las empresas dependen en gran medida de la tecnología para sus operaciones diarias, la eficiencia en la administración de las mesas de servicio se ha convertido en un factor clave para asegurar la calidad del soporte técnico y la satisfacción de los usuarios. El presente proyecto tiene como finalidad desarrollar un Sistema de Análisis y Gestión que optimice la operación de una mesa de servicio, mejorando la calidad, rapidez y efectividad en la resolución de incidencias y la atención a solicitudes de los usuarios.

El sistema propuesto se enfocará en automatizar y centralizar la gestión de tickets, facilitando el registro, seguimiento y cierre de incidencias de manera eficiente. Esto se logrará a través de la implementación de un software que integre herramientas de análisis en tiempo real, generación de reportes automáticos, y capacidades de priorización de tareas. Al permitir una mayor visibilidad sobre el estado de las solicitudes, los administradores podrán tomar decisiones más informadas y asignar recursos de manera óptima. (IBM, 2023)

**Tareas básicas que va a realizar el sistema:**

* Llevar un control de los usuarios, tickets, información, procedimientos.
* Permitirá generar tickets de solicitud de servicio.
* Permitirá generar derivar procesos.
* Permitirá generar informes para las auditorías internas.

**Beneficios esperados**

Los beneficios esperados incluyen una mejora significativa en la eficiencia operativa y la reducción de los tiempos de respuesta ante incidencias y solicitudes de los usuarios. Al centralizar y automatizar la gestión de tickets, se facilitará la priorización de las tareas y la asignación de recursos de forma más eficiente, lo que permitirá una atención más rápida y enfocada en las áreas críticas.

Además, la implementación de herramientas de análisis en tiempo real y la generación de reportes automáticos proporcionará a los administradores datos precisos y actualizados sobre el desempeño del equipo y las tendencias en las solicitudes. (IBM, 2023)

# **Alcance o delimitación**

1. **Gestión centralizada de tickets**: El sistema permitirá la recepción, registro y seguimiento de todas las incidencias y solicitudes realizadas por los usuarios, proporcionando una plataforma unificada para su administración.
2. **Automatización de procesos clave**: Se implementarán funciones automáticas para la asignación de tareas, priorización de incidencias según su criticidad, y escalamiento de problemas complejos a niveles superiores.
3. **Generación de reportes en tiempo real**: El sistema generará informes automáticos sobre el rendimiento del equipo de soporte, tiempos de resolución y cantidad de tickets gestionados, permitiendo un análisis continuo de los resultados.
4. **Análisis de datos**: Integración de herramientas que analicen los datos recopilados en tiempo real para identificar patrones, mejorar la toma de decisiones y optimizar los procesos de atención al cliente.
5. **Interfaz intuitiva para operadores**: Se desarrollará una interfaz sencilla y accesible que facilite el uso diario del sistema por parte de los operadores de la mesa de servicio, reduciendo la curva de aprendizaje y mejorando la productividad.
6. **Funcionalidades de retroalimentación del cliente**: Se incluirá un módulo para que los usuarios finales puedan evaluar el servicio recibido, proporcionando información valiosa para la mejora continua del soporte.
7. **Capacitación del personal**: Se realizará la capacitación del equipo de la mesa de servicio y los administradores del sistema para garantizar una implementación eficiente y un uso adecuado de todas las funcionalidades del sistema.
8. **Mantenimiento y soporte técnico**: El proyecto incluirá un plan básico de mantenimiento y soporte técnico post-implementación para asegurar el correcto funcionamiento del sistema y resolver posibles incidencias técnicas. (Software, 2019)

# **Limitaciones y restricciones**

El sistema de análisis y gestión para la mesa de servicio presenta ciertas limitaciones que deben ser consideradas para garantizar su viabilidad y éxito. Estas restricciones abarcan tanto aspectos técnicos como operativos, que definirán el alcance real de su implementación. (service, 2014)

**Limitaciones/Restricciones:**

1. Restricción presupuestaria: El proyecto deberá ajustarse al presupuesto disponible, limitando el uso de tecnologías avanzadas o licencias de software costosas.
2. Tiempo de implementación: El desarrollo y despliegue del sistema estará restringido a un cronograma predefinido, lo que podría limitar la incorporación de funcionalidades adicionales.
3. Integración con sistemas existentes: El sistema solo integrará con las plataformas internas esenciales, limitando su conexión con sistemas externos complejos.
4. Capacidad técnica del personal: La implementación estará sujeta al nivel de conocimiento técnico del equipo, lo que requerirá capacitación adicional para ciertas funcionalidades.
5. Requerimientos de hardware y software: Las soluciones implementadas deberán ajustarse a la infraestructura tecnológica existente, limitando la adquisición de nuevo hardware o actualizaciones significativas.
6. Número de usuarios concurrentes: El sistema tendrá un límite en la cantidad de usuarios que pueden acceder simultáneamente, debido a restricciones en la infraestructura disponible.
7. Soporte post-implementación: El soporte técnico y mantenimiento serán limitados a un período específico y no cubrirán mejoras o expansiones futuras.
8. Alcance geográfico: La implementación inicial estará limitada a una sola sede o área operativa, sin cubrir filiales o oficinas en otras ubicaciones.
9. Cumplimiento normativo: El sistema deberá ajustarse a las normativas locales y de seguridad de la información, lo que podría limitar ciertas funcionalidades.
10. Escalabilidad limitada: Aunque el sistema será modular, la capacidad de expansión estará limitada por las restricciones presupuestarias y de infraestructura.

Las limitaciones y restricciones del sistema de análisis y gestión para la mesa de servicio son aspectos críticos que deben ser gestionados cuidadosamente para asegurar el éxito del proyecto. La restricción presupuestaria, el tiempo de implementación y la capacidad técnica del personal son factores clave que influirán en el desarrollo y operación del sistema. Además, las limitaciones en la integración con sistemas existentes y el soporte post-implementación podrían afectar la funcionalidad y sostenibilidad a largo plazo del sistema. Es esencial que el equipo del proyecto aborde estas limitaciones de manera proactiva, buscando soluciones viables que maximicen el impacto positivo del sistema dentro de los parámetros establecidos. Con una planificación adecuada, será posible minimizar los efectos negativos de estas restricciones y garantizar una implementación exitosa. (service, 2014).

# **Requisitos o especificaciones**

1. Objetivos del Sistema

* Mejorar la eficiencia en la gestión de incidentes y solicitudes.
* Proveer una plataforma unificada para la comunicación entre usuarios y técnicos.
* Automatizar procesos como la asignación de tickets y el seguimiento del estado de los mismos.
* Facilitar la documentación y el reporte de incidencias para la generación de informes y análisis de rendimiento. (Tuffley, Software Requirements Specifications, 2010)

2. Requisitos Funcionales

2.1. Gestión de Incidentes

Registro de Incidentes:

* Permitir a los usuarios registrar incidentes a través de un formulario intuitivo.
* Incluir campos obligatorios como título, descripción, categoría y prioridad. (Tuffley, 2010)

Clasificación y Priorización:

* Clasificar automáticamente los incidentes según criterios predefinidos (por ejemplo, hardware, software, red).
* Permitir a los usuarios seleccionar la prioridad (alta, media, baja) al registrar un incidente.

Asignación de Tickets:

* Asignar automáticamente los tickets a los técnicos disponibles basándose en su especialización y carga de trabajo.
* Permitir la reasignación de tickets en caso de que un técnico no pueda atenderlos.

Historial de Incidentes:

* Almacenar el historial de cada incidente, incluyendo detalles de las interacciones y soluciones aplicadas.
* Permitir a los usuarios y técnicos consultar el historial en cualquier momento.

2.2. Gestión de Solicitudes de Servicio

Registro de Solicitudes:

* Permitir a los usuarios registrar solicitudes de servicio de manera sencilla.
* Incluir campos como tipo de solicitud, descripción, y fecha requerida.
* Seguimiento de Solicitudes:
* Permitir a los usuarios consultar el estado de sus solicitudes en cualquier momento.
* Proveer notificaciones automáticas a los usuarios sobre actualizaciones del estado de sus solicitudes.

Cierre de Solicitudes:

* Permitir el cierre de solicitudes una vez que han sido atendidas, registrando la solución implementada.
* Ofrecer la opción a los usuarios de proporcionar retroalimentación sobre la atención recibida.

2.3. Informes y Análisis

Generación de Informes:

* Permitir la generación de informes sobre el rendimiento del servicio, tiempos de respuesta, y tasas de resolución.
* Proveer plantillas de informes predefinidos que los administradores puedan utilizar. (Tuffley)

Análisis de Tendencias:

* Incluir herramientas para analizar tendencias en incidentes y solicitudes, facilitando la identificación de áreas de mejora.
* Ofrecer gráficos y visualizaciones para representar los datos analizados.

3. Requisitos No Funcionales

3.1. Usabilidad

Interfaz Intuitiva:

* La interfaz del sistema debe ser fácil de usar para usuarios no técnicos.
* Proveer guías y tutoriales para facilitar la navegación por el sistema.

Accesibilidad:

* Debe ser accesible desde diferentes dispositivos (PC, tablets, smartphones).
* Cumplir con estándares de accesibilidad web para usuarios con discapacidades.

3.2. Rendimiento

Tiempo de Respuesta:

* El sistema debe responder a las solicitudes del usuario en un tiempo no mayor a 2 segundos.
* Garantizar que el sistema funcione de manera óptima durante las horas pico.

Capacidad de Carga:

* Debe soportar un alto volumen de usuarios simultáneos sin afectar el rendimiento.
* Realizar pruebas de carga para asegurar la estabilidad del sistema.

3.3. Seguridad

Control de Acceso:

* Implementar un control de acceso basado en roles, asegurando que solo el personal autorizado pueda acceder a funciones críticas.
* Realizar auditorías de acceso regularmente para garantizar la seguridad de la información.

Cifrado de Datos:

* Todos los datos sensibles deben ser cifrados tanto en tránsito como en reposo.
* Implementar autenticación multifactor para aumentar la seguridad de acceso. (Sommerville, 2011)

4. Requisitos Técnicos

4.1. Plataforma

Lenguaje de Programación:

* Desarrollar el sistema utilizando tecnologías web modernas (por ejemplo, JavaScript, HTML5, CSS3).
* Utilizar frameworks de desarrollo que faciliten la creación de aplicaciones escalables. (Lerdorf)

Base de Datos:

* Se requiere el uso de una base de datos relacional (como MySQL o PostgreSQL) para almacenar la información de los usuarios y los tickets.
* Implementar un sistema de respaldo de datos para evitar pérdidas de información.

(Tahaghoghi, 2007)

# **Metodología A Utilizar (2 Pagina) Agil, Scrum, Xp, Jira Software …**

**Introducción**

La metodología Kanban para la gestión de nuestro proyecto de análisis y gestión de la mesa de servicio es esencial para el enfoque en la visualización del trabajo y la flexibilidad. Kanban permite adaptar rápidamente las prioridades y optimizar el flujo de tareas, lo que es crucial en un entorno dinámico. Además, su énfasis en la mejora continua y la colaboración del equipo favorece un ambiente de trabajo más eficiente y transparente. Esta metodología se alinea perfectamente con nuestras metas de ofrecer un servicio ágil y de alta calidad.

Utilizaríamos Kanban en nuestro proyecto porque nos ofrece una gestión de trabajo más fluida, lo que es esencial para el desarrollo ágil y coherente de nuestras tareas. Al visualizar el trabajo por fases, no solo obtenemos una visión clara del progreso, sino que también podemos identificar y evitar potenciales sobrecargas en el equipo. Esta visualización constante asegura que todos los miembros del equipo estén al tanto de sus compromisos actuales y lo que se espera de ellos. Además, al tener metas claras y un flujo de trabajo bien definido, buscamos establecer un sistema de trabajo más confiable y predecible. La capacidad de rastrear el tiempo invertido en comparación con el tiempo previsto nos da una herramienta adicional para hacer ajustes en tiempo real, garantizando que el proyecto se mantenga alineado con nuestros objetivos globales. Kanban sería una herramienta valiosa para mejorar nuestra eficiencia, comunicación y adaptabilidad en el proyecto. (Kanbantool, 2023).

Algunas razones clave por las que he elegido Kanban para este proyecto son:

* **Visualización del flujo de trabajo:** Kanban permite visualizar las tareas en un tablero, facilitando el seguimiento del progreso y la identificación de cuellos de botella.
* **Flexibilidad:** Kanban es altamente adaptable. Puedes ajustar las prioridades y agregar nuevas tareas según las necesidades del servicio, lo cual es crucial en un entorno dinámico como una mesa de servicio.
* **Mejora continua:** Fomenta la revisión constante de procesos y resultados, lo que ayuda a identificar áreas de mejora y optimizar el flujo de trabajo.
* **Limitación del trabajo en progreso (WIP):** Kanban ayuda a gestionar la carga de trabajo limitando la cantidad de tareas que se pueden realizar simultáneamente, lo que reduce el estrés del equipo y mejora la calidad del trabajo.
* **Enfoque en la entrega continua:** Permite entregar mejoras y actualizaciones de manera continua, lo que es especialmente útil para adaptarse a las necesidades cambiantes de los usuarios.
* **Colaboración del equipo:** Fomenta la comunicación y colaboración dentro del equipo, ya que todos pueden ver el estado de las tareas y participar en la toma de decisiones.
* **Transparencia:** Todos los involucrados tienen una visión clara del progreso y los problemas, lo que mejora la comunicación con las partes interesadas

Jira: La Herramienta de Elección para la Gestión Kanban

Ahora, para llevar a cabo la metodología Kanban de manera efectiva, es fundamental contar con las herramientas adecuadas. Es aquí donde Jira entra en escena. Jira, desarrollado por Atlassian, es una plataforma líder en la gestión de proyectos que ofrece soluciones específicas para implementar Kanban de forma digital. (Formacion, 2024)

Razones para optar por Jira:

1. Tableros Personalizables: Jira permite la creación de tableros Kanban personalizados que se adaptan a las necesidades específicas del proyecto de mesa de servicio.
2. Integraciones: Jira se integra con una amplia gama de otras herramientas y plataformas, lo que significa que puede adaptarse y expandirse según las necesidades del proyecto.
3. Seguimiento en tiempo real: Con Jira, todo el equipo puede ver el estado actual del proyecto en tiempo real, lo que ayuda a la transparencia y la colaboración.
4. Informes y Métricas: Jira ofrece una serie de informes útiles que pueden ayudar a analizar el rendimiento del proyecto y detectar áreas de mejora.
5. Escalabilidad: A medida que el proyecto crece y se expande, Jira puede manejar el aumento de la carga de trabajo sin problemas. (Formacion, 2024)

# **Sistema de Propuesto**

El DRU describe las funcionalidades y características que el sistema de gestión de mesa de servicio debe proporcionar desde la perspectiva del usuario final. Estas especificaciones son esenciales para asegurar que el sistema cubra todas las necesidades del cliente y de los usuarios que interactuarán con él.

1. **Requerimientos del Usuario Final:**

* Facilidad de uso: La interfaz debe ser intuitiva, simple y fácil de navegar para cualquier empleado, independientemente de su nivel técnico.
* Acceso a través de múltiples dispositivos: Los usuarios deben poder acceder al sistema desde sus dispositivos de escritorio y móviles, permitiendo reportar incidencias de manera inmediata.
* Sistema de notificaciones: Los usuarios deben recibir notificaciones automáticas sobre el estado de sus incidencias, asegurando transparencia y manteniendo la comunicación abierta con el equipo de soporte.
* Historial de incidencias: Los usuarios deben poder consultar el estado actual y el historial de sus tickets de soporte, junto con las soluciones implementadas.
* Tiempo de respuesta rápido: El sistema debe garantizar tiempos de respuesta optimizados para la resolución de problemas, con tiempos de espera mínimos. (Shneiderman, 1992)

1. **Flujo de Usuario Esperado:**

* El usuario se autentica en el sistema con sus credenciales.
* Desde el panel principal, el usuario puede crear un nuevo ticket de incidencia, completando los campos necesarios, como categoría del problema, descripción y prioridad.
* El sistema confirma la creación del ticket y envía notificaciones sobre el estado de su incidencia (por ejemplo, cuando ha sido asignada a un técnico o resuelta).
* El usuario puede revisar el estado de sus incidencias en cualquier momento o actualizar la información del ticket si es necesario.
* Al cierre de la incidencia, el usuario recibe una notificación final y una encuesta de satisfacción opcional. (Ambrós Mendioroz, 2017)

1. **Diseño del Requerimiento Técnico (DRT)**

El DRT especifica los aspectos técnicos que deben ser implementados para cumplir con los requerimientos funcionales y no funcionales del sistema, y garantizar su operatividad y eficiencia.

**Requerimientos Técnicos:**

* Arquitectura basada en la nube: El sistema debe implementarse en una arquitectura escalable en la nube, garantizando acceso seguro y rápido desde cualquier dispositivo. Esto también facilita la expansión futura del sistema y la gestión eficiente de grandes volúmenes de datos.
* Base de datos relacional: Se utilizará una base de datos que permita gestionar los usuarios, incidencias, y las respuestas del equipo técnico. MySQL o PostgreSQL podrían ser opciones recomendadas por su robustez y capacidad de gestión de grandes volúmenes de datos.
* Sistema de autenticación y roles: El sistema debe tener un sistema seguro de autenticación basado en roles. Solo los usuarios autorizados deben poder acceder a las funciones críticas del sistema, como la creación de tickets o la administración de incidencias.
* Sistema de notificaciones automáticas: Implementación de un sistema de notificaciones que pueda enviar correos electrónicos o notificaciones push a los usuarios sobre el estado de sus incidencias en tiempo real.
* Seguridad y encriptación de datos: Todo el tráfico de información, especialmente los datos personales y de incidencias, debe estar encriptado mediante HTTPS y protocolos de seguridad para evitar vulnerabilidades. (Kendall, 2005)

# **Justificación del Estudio**

# **A quién beneficia**

# El desarrollo de este sistema beneficiará directamente a los empleados de la organización al proporcionarles una herramienta confiable y eficiente para reportar problemas técnicos y recibir asistencia de manera oportuna. También beneficiará al equipo de soporte técnico, quienes podrán gestionar y resolver incidencias de manera más organizada y rápida, lo que mejorará su productividad y capacidad de respuesta. (Blut, 2022)

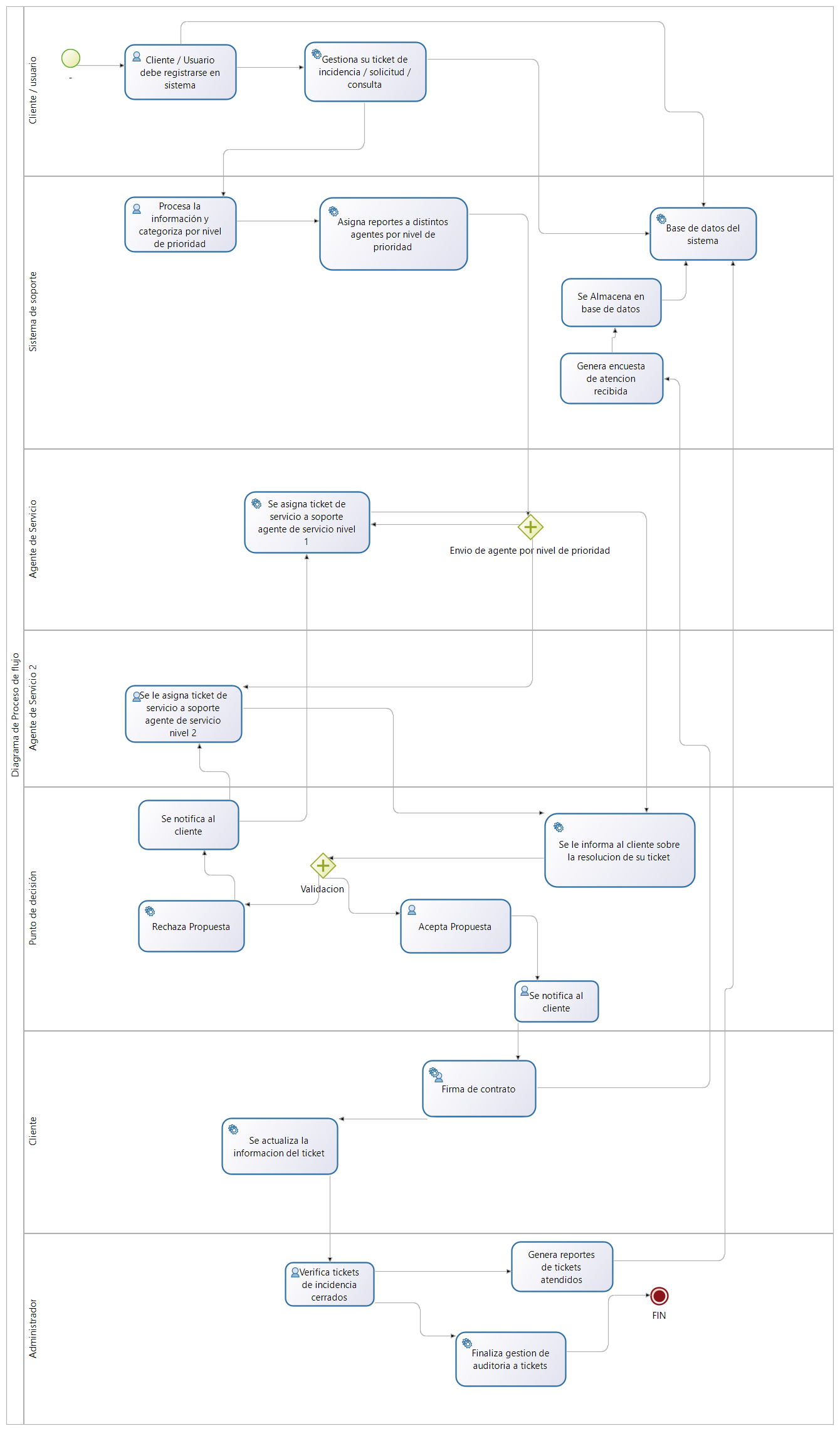
# **Importancia del Proyecto**

# Este sistema es crucial para mejorar la eficiencia de los procesos internos y asegurar un alto nivel de satisfacción de los empleados al recibir soporte técnico. En lugar de depender de sistemas manuales o desorganizados, la mesa de servicio proporcionará una plataforma centralizada que optimiza la comunicación y permite una gestión más eficiente de los problemas técnicos. (Fitzsimmons, 1997)

# **Retorno y Costos-Beneficios**

# El costo inicial de desarrollo del sistema será recuperado rápidamente a través de la mejora en la productividad general de la organización. La reducción de los tiempos de inactividad debido a problemas técnicos y la mayor capacidad del equipo de soporte para resolver incidencias a tiempo resultarán en una mejora notable de la eficiencia operativa. A largo plazo, los beneficios superarán ampliamente los costos asociados con la implementación, dado que el sistema continuará aportando valor con cada incidencia resuelta de manera más rápida y eficaz. (Hollins, 2006)

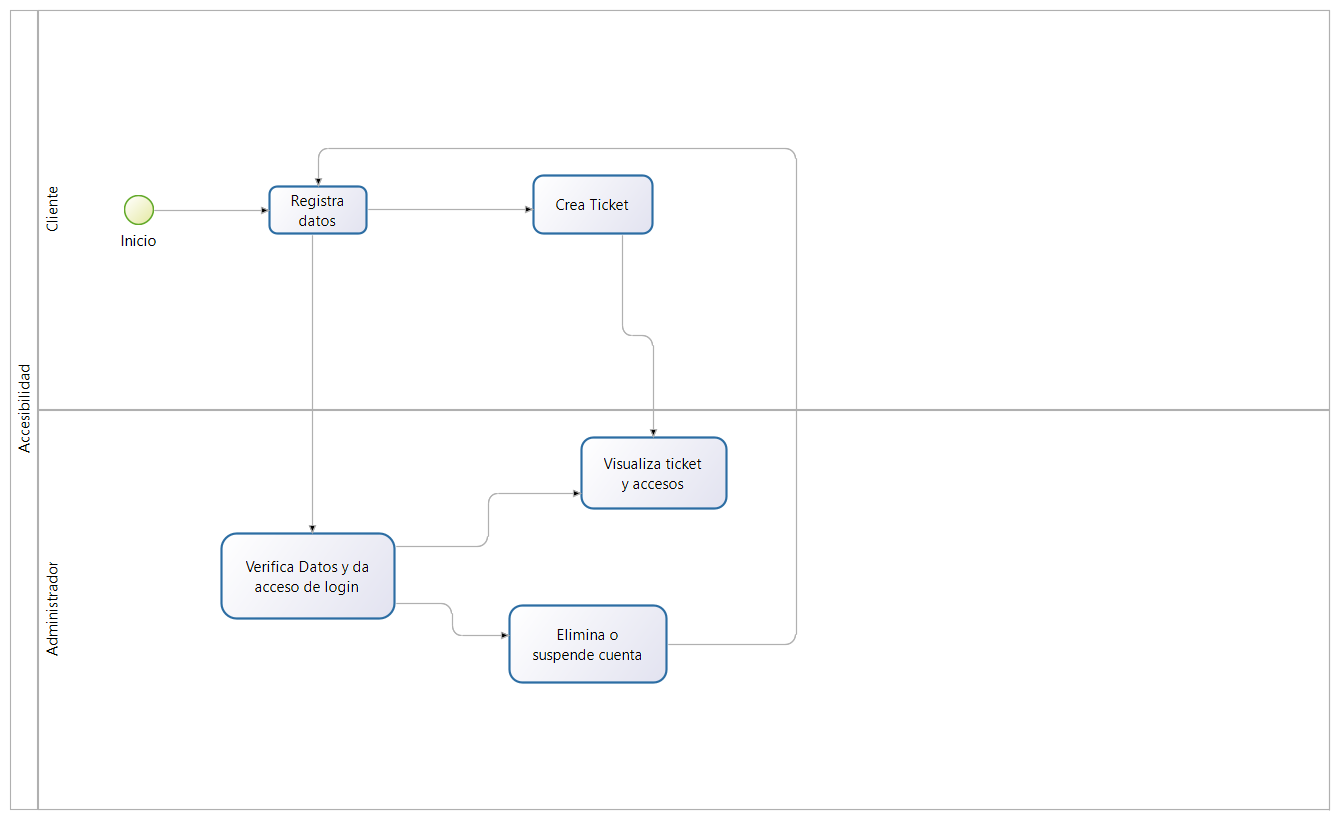
# **Definición de procesos**



1-Figura 1: procesos

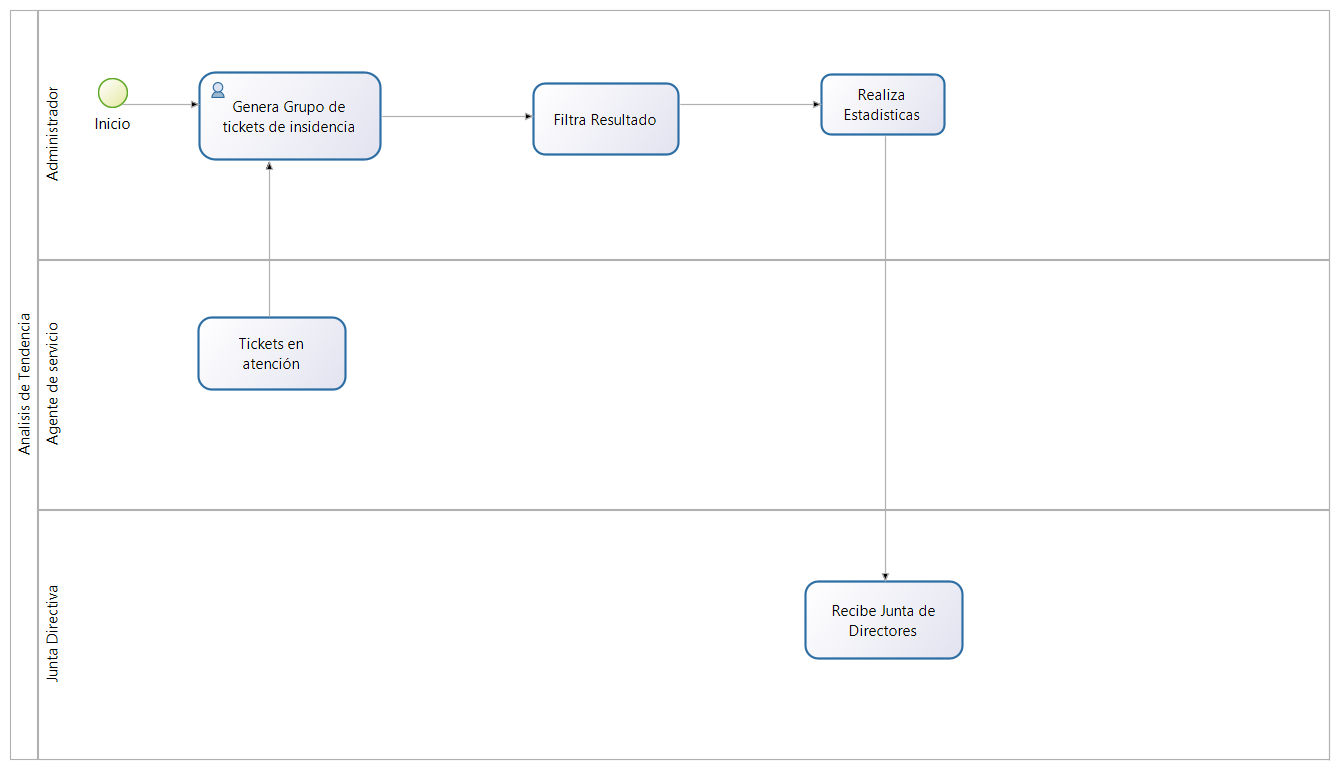
**Diagrama de procesos**

Accesibilidad



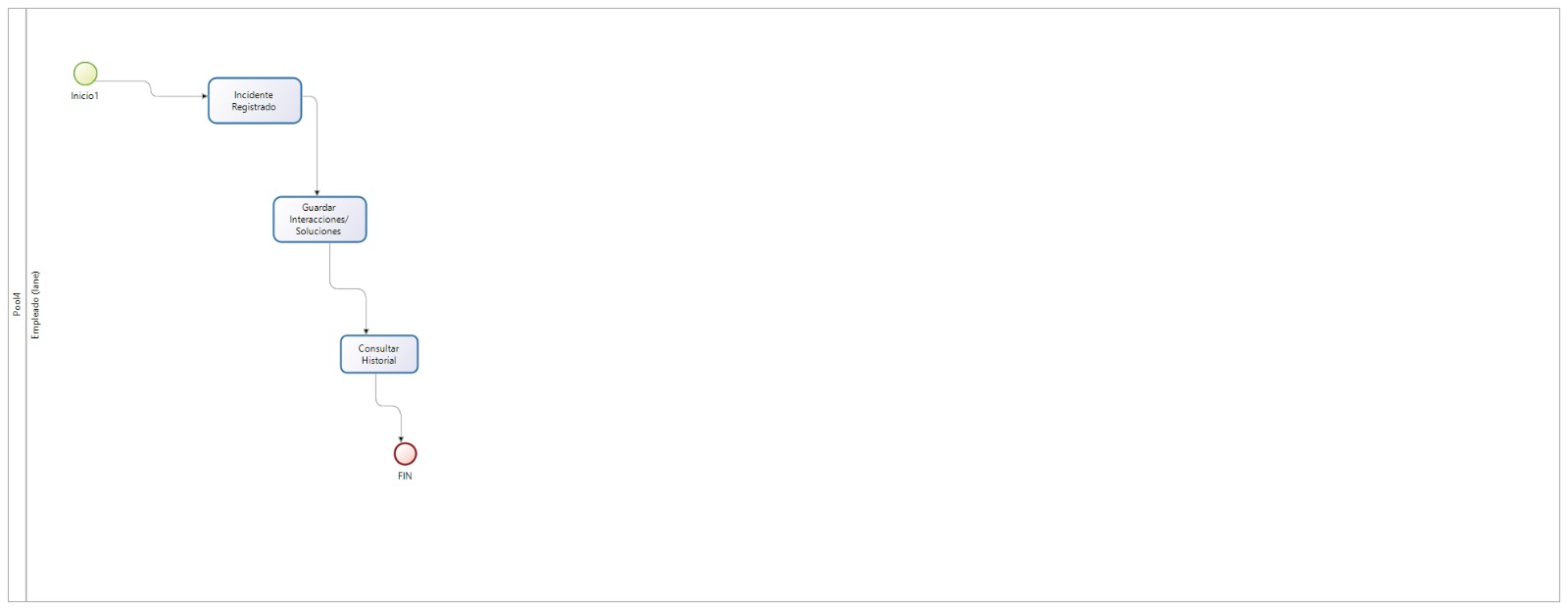
2- Figura 2: Gestión de Accesibilidad

Análisis de tendencia



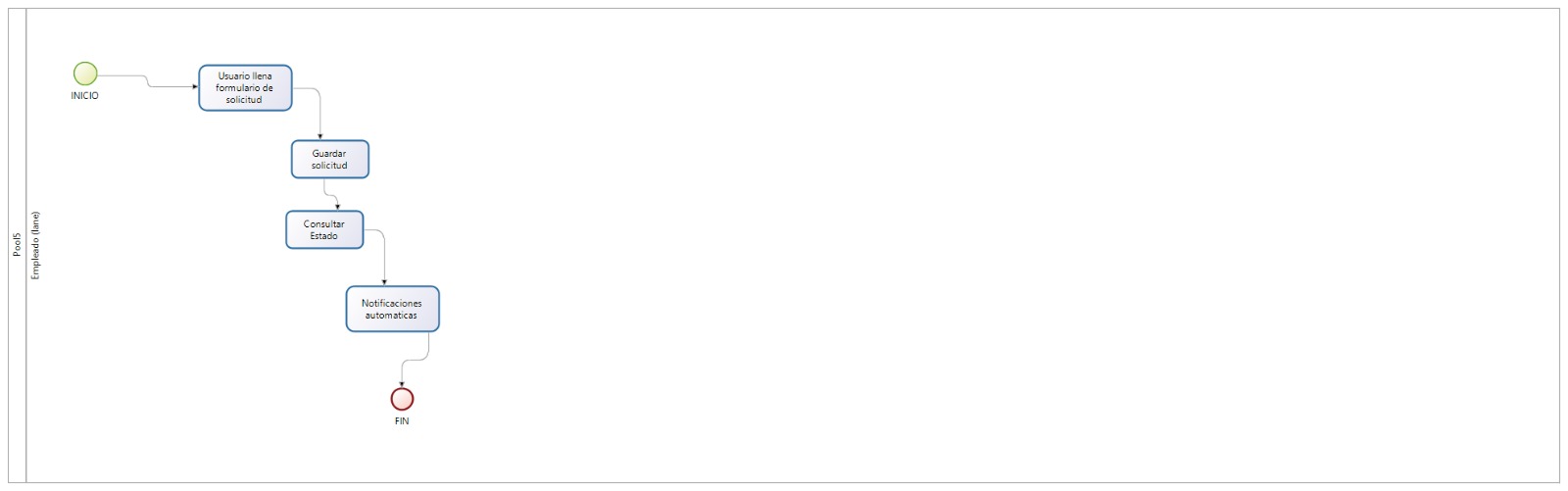
3-Figura 3: Análisis de tendencia

Incidente Registrado



4-Figura 4: Indicentes Registrados

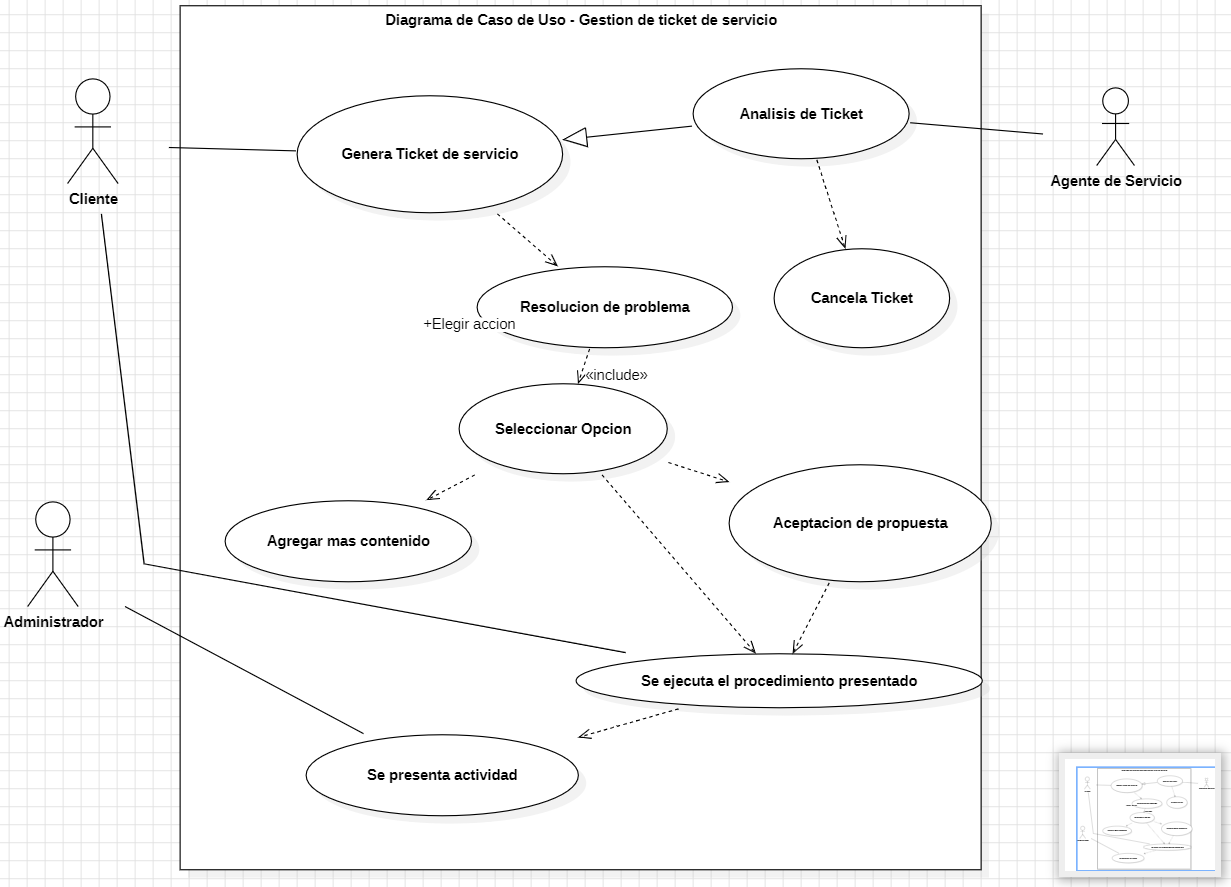
Registro y seguimiento de solicitudes de servicio



5-Figura 5: Registro y seguimiento de solicitudes de servicio

# **Diagrama de caso de Uso (De todos los requerimientos de su proyecto)**

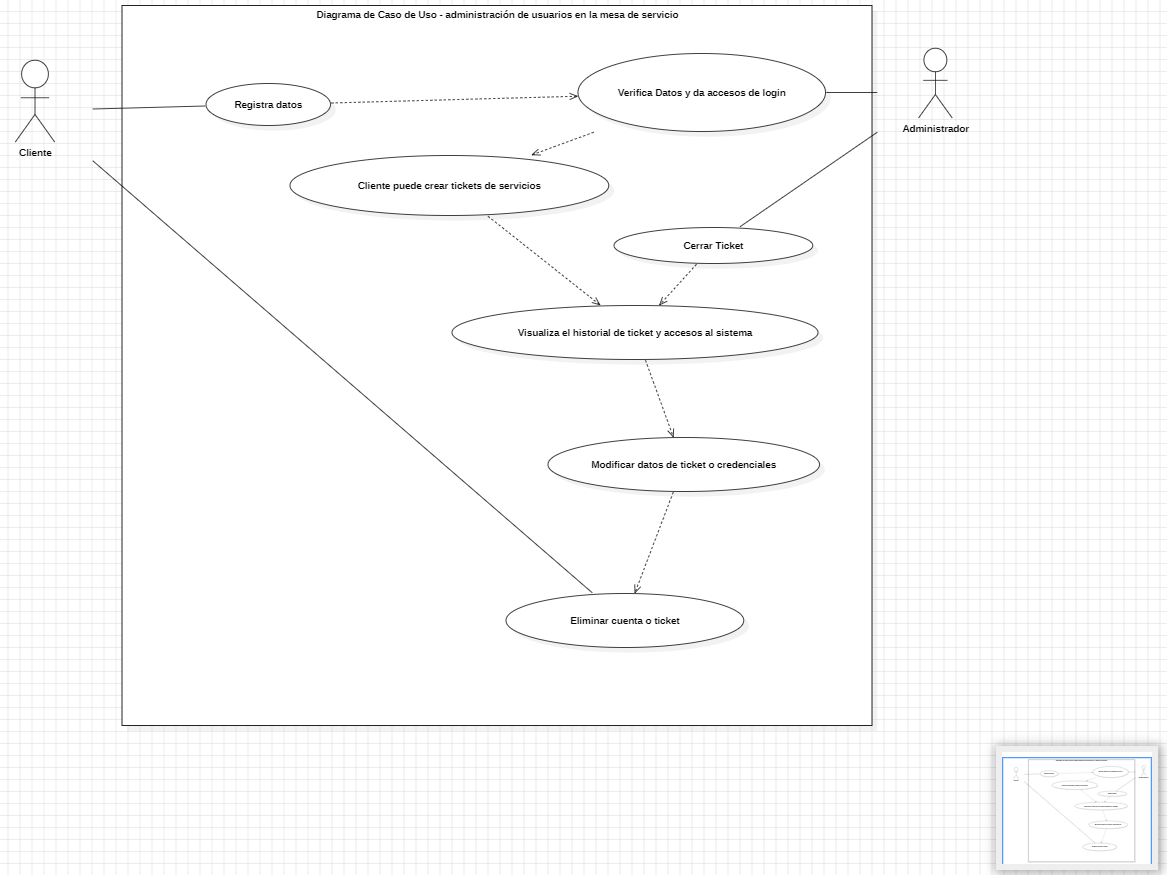
1. Caso de uso 1: Diagrama de caso de uso para la gestión de tickets



6- Figura 6: interacción con la gestion de generación de Ticket

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Interacción con la gestión de generación de tickets | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema debe tener la capacidad de generar, administrar y procesar los tickes de solicitud de clientes. | | | | |
| Precondición | | El cliente debe estar pre- registrado a la plataforma. | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Cliente) genera ticket de servicio | | |
| 2 | | El actor (agente de servicio) analiza lo detallado en el ticket | | |
|  | | 3 | | El actor (agente de servicio) valida la resolución del ticket presentado | |
| 4 | | El actor (cliente) puede agregar mas contenido al ticket o aceptar la solución brindada. | |
| 5 | | Se procesa lo aceptado por el cliente. | |
| 6 | | Se presenta actividad realizada al actor (Administrador). | |
| Post condición | | Salir del sistema | | | |
| Excepciones | | Paso | | Acción | |
| 2 | | El cliente debe estar registrado en la plataforma de generación de ticket | |
|  | |  | |
| Respuestas brindadas esperados | | 240 frecuencia por mes. | | | |
| Importancia | | Alta | | | |
| Estado | | En Ejecución | | | |
| Estabilidad | | Alta | | | |

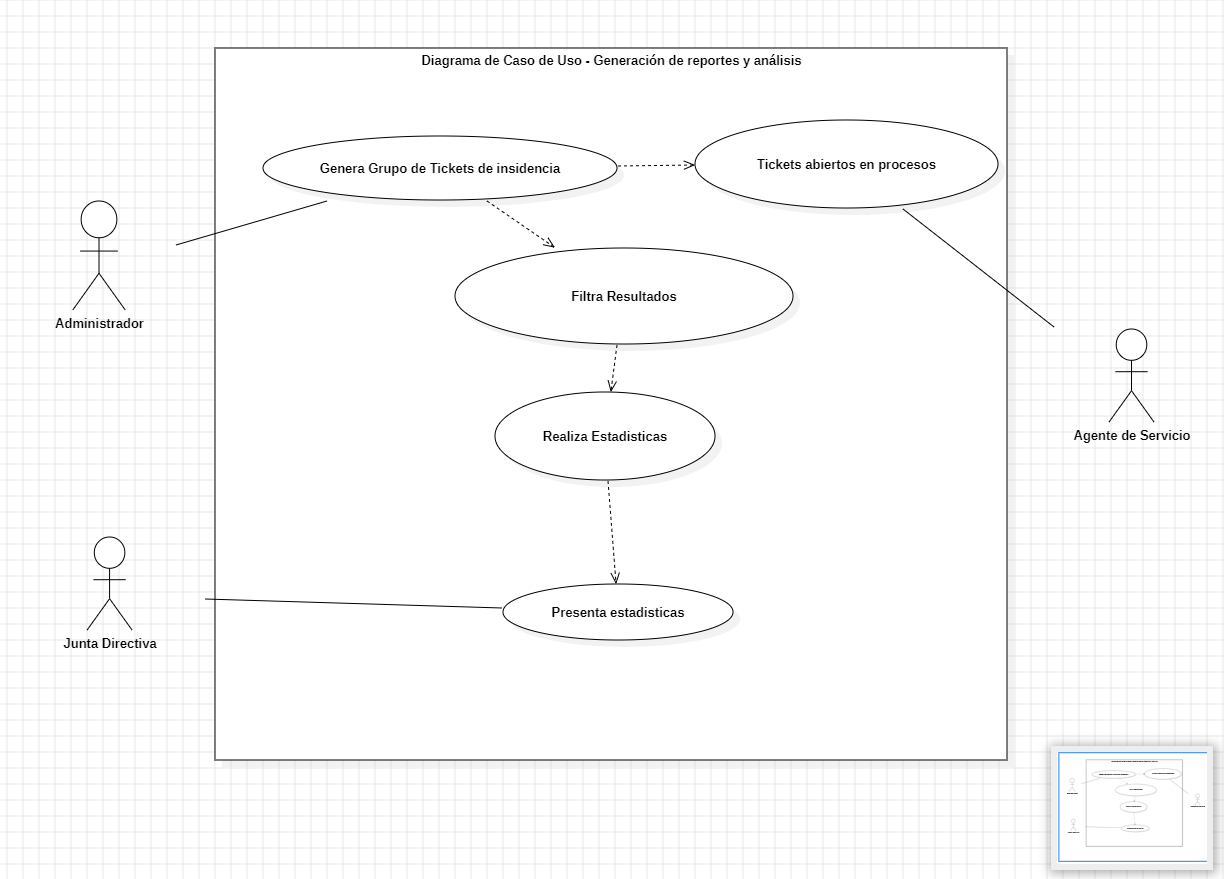
1. Caso de uso 2: Administración de usuarios de una mesa de servicio



7-Figura 7: Administración de usuario de una mesa de servicio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Administración de usuarios en la mesa de servicio | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema administrador los accesos y las solicitudes de los clientes | | | | |
| Precondición | |  | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Cliente) ingresa sus datos para registrarse en sistema. | | |
| 2 | | El actor (administrador) verifica los datos y da acceso al cliente al sistema. | | |
|  | | 3 | | El actor (cliente) puede crear ticket de servicio. | |
| 4 | | El administrador puede visualizar tickets de servicio y eliminarlos | |
| 5 | | El Cliente puede dar seguimiento a sus reportes realizados. | |
| 6 | | Es permitido modificar ticket en secciones especificas o modificar credenciales | |
| 7 | | El cliente y administrador puede cerrar su cuenta de sistema. | |
| Post condición | | Salir del sistema | | | |
| Excepciones | | Paso | | Acción | |
| Nuevos clientes esperados | | 100 frecuencia por mes. | | | |
| Importancia | | Alta | | | |
| Estado | | En Ejecución | | | |
| Estabilidad | | Alta | | | |

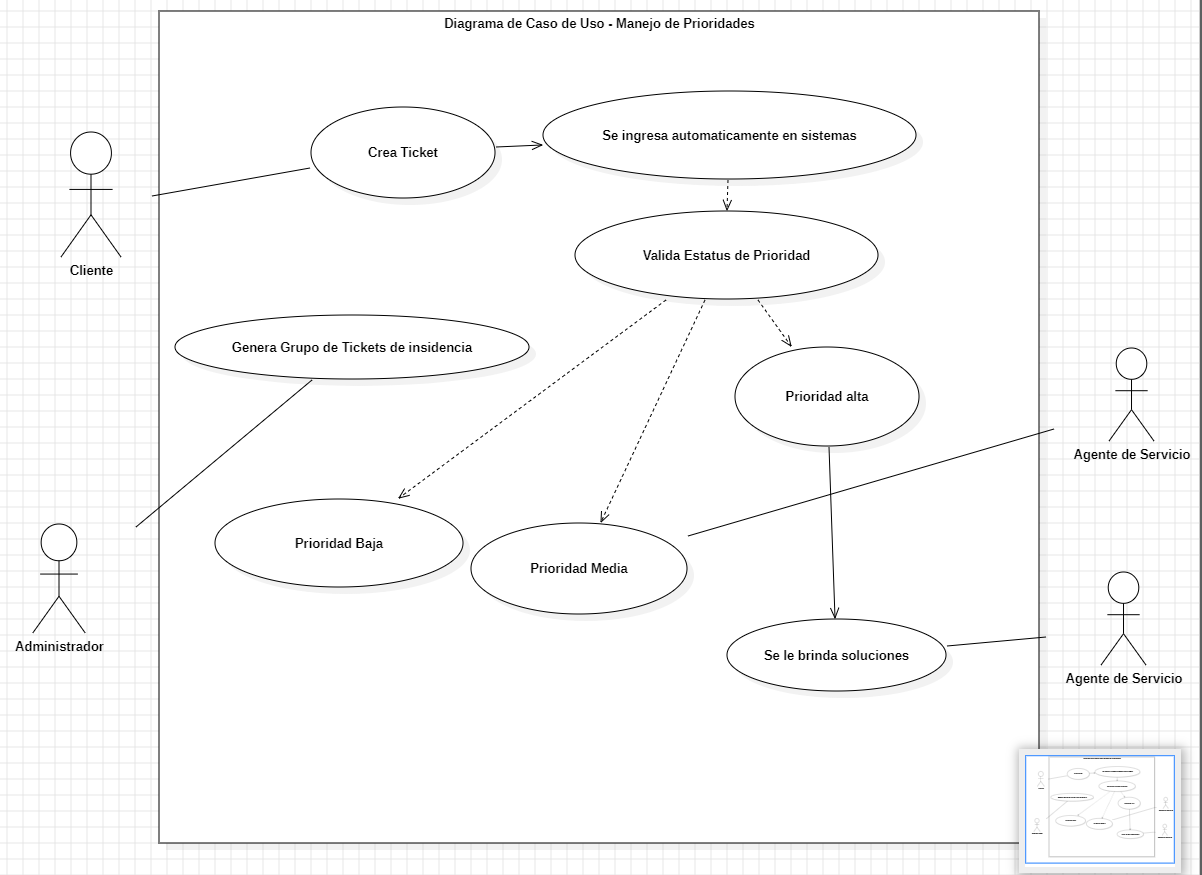
1. Caso de uso 3: Reportes y su análisis



8- Figura 8: Generación de reportes y analisis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Generación de reportes y análisis | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema debe ser capaz de generar diversos tipos de reporte. Los mismos son solicitados por los administradores. Reportes en diversos formatos. | | | | |
| Precondición | | Reportes validos para administradores | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Administrador) ingresa a la plataforma | | |
| 2 | | El actor (administrador) puede generar listado de tickets de incidencias | | |
|  | | 3 | | El actor (administrador) puede visualizar los tickets abiertos y en proceso. | |
| 4 | | Se realizan filtros. | |
| 5 | | Se filtraron los resultados y se presentan al actor ( Junta de directores) | |
| Post condición | | Salir del sistema | | | |
| Excepciones | | Paso | | Acción | |
| 2 | | Apartado valido para administradores | |
| Importancia | | Media | | | |
| Estado | | En Ejecución | | | |
| Estabilidad | | Alta | | | |
| Comentarios | | El sistema tiene la capacidad de generar reportes en pdf. | | | |

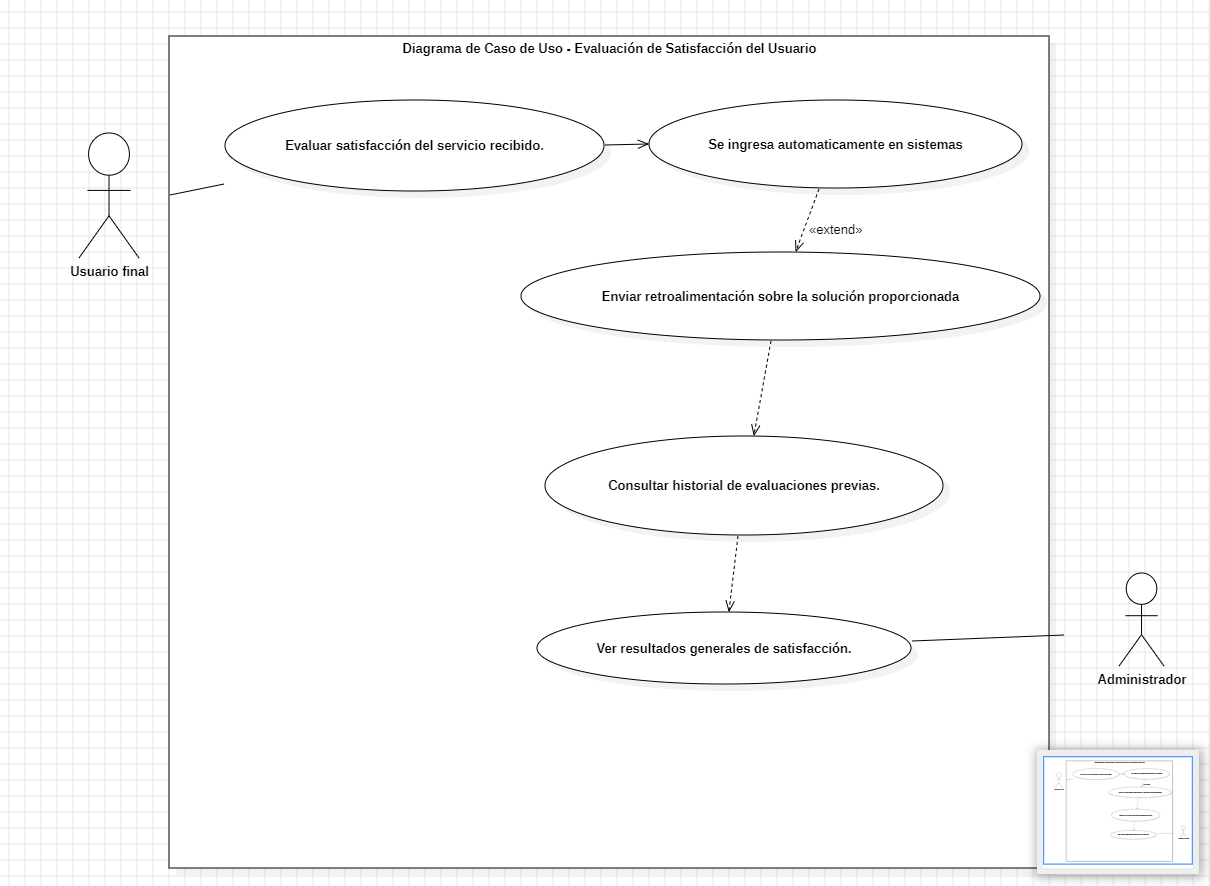
1. Caso de uso 4: Manejo de Prioridades



9-Figura 9: Manejo de prioridades

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Manejo de prioridades | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema debe ser capaz de categorizar los tickets por medio de su nivel de prioridad, baja, media o alta. | | | | |
| Precondición | | El cliente debe estar pre- registrado a la plataforma. | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Cliente) ingresa a la plataforma | | |
| 2 | | El actor (agente de servicio) debe filtrar los reportes por su nivel de prioridad | | |
|  | | 3 | | Categoría baja mantendrá una condiciones básica | |
| 4 | | Categoría media será atentado por un agente de servicio | |
| 5 | | Categoría alta puede ser atendido por un administrador o un agente de servicio. | |
| 6 | | Se les brinda a soluciones a los tickets. | |
| Post condición | | Activo 24/7 | | | |
| Excepciones | | Paso | | Acción | |
| 2 | | El cliente debe estar registrado en la plataforma de compra de vuelo. | |
| Atención esperada | | Aumentar la productividad y dismunir el tiempo de espera | | | |
| Importancia | | Alta | | | |
| Estado | | En Ejecución | | | |
| Estabilidad | | Alta | | | |
| Comentarios | | Enfocado a la atención rápida y segura. | | | |

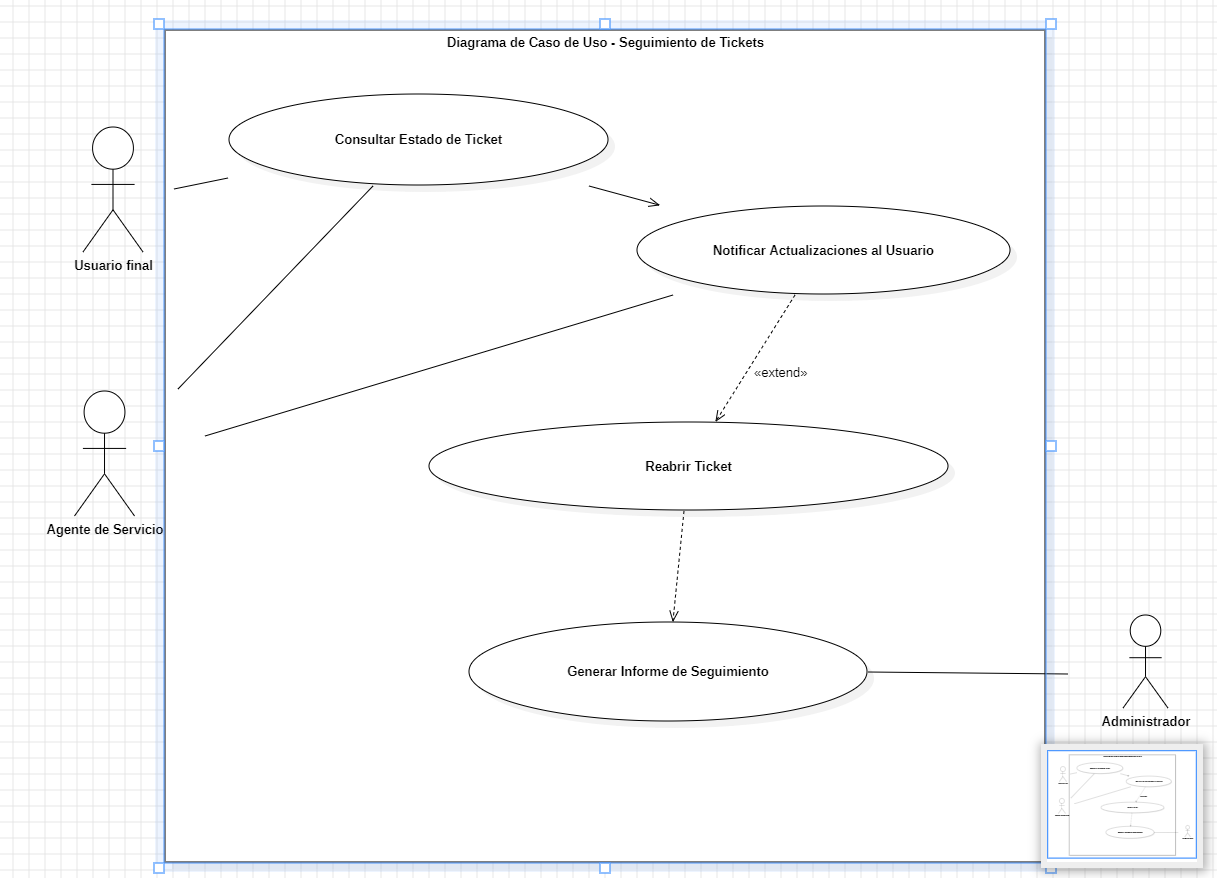
1. Caso de uso 5: Evaluación de Satisfacción del Usuario



10-Figura 10: Evaluación de satisfacción de usuario

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Evaluación de satisfacción de usuario | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema debe hacer capaz de enviar automáticamente una encuenta de satisfacción al cliente. Debe ser procesada y almacenada en el centro de datos y evaludada por el administrador. | | | | |
| Precondición | | El cliente debe haber abierto un ticket de servicio | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Usuario final-cliente) finaliza la creación de un ticket | | |
| 2 | | Recibe encuenta de servicios. | | |
|  | | 3 | | El cliente puede visualizar sus tickets o encuestas previas. | |
| 4 | | El actor (Administrador) puede visualizar los resultados finales de las encuestas. | |
| Importancia | | | Media | | | | |
| Estado | | | En Ejecución | | | | |
| Estabilidad | | | Alta | | | | |

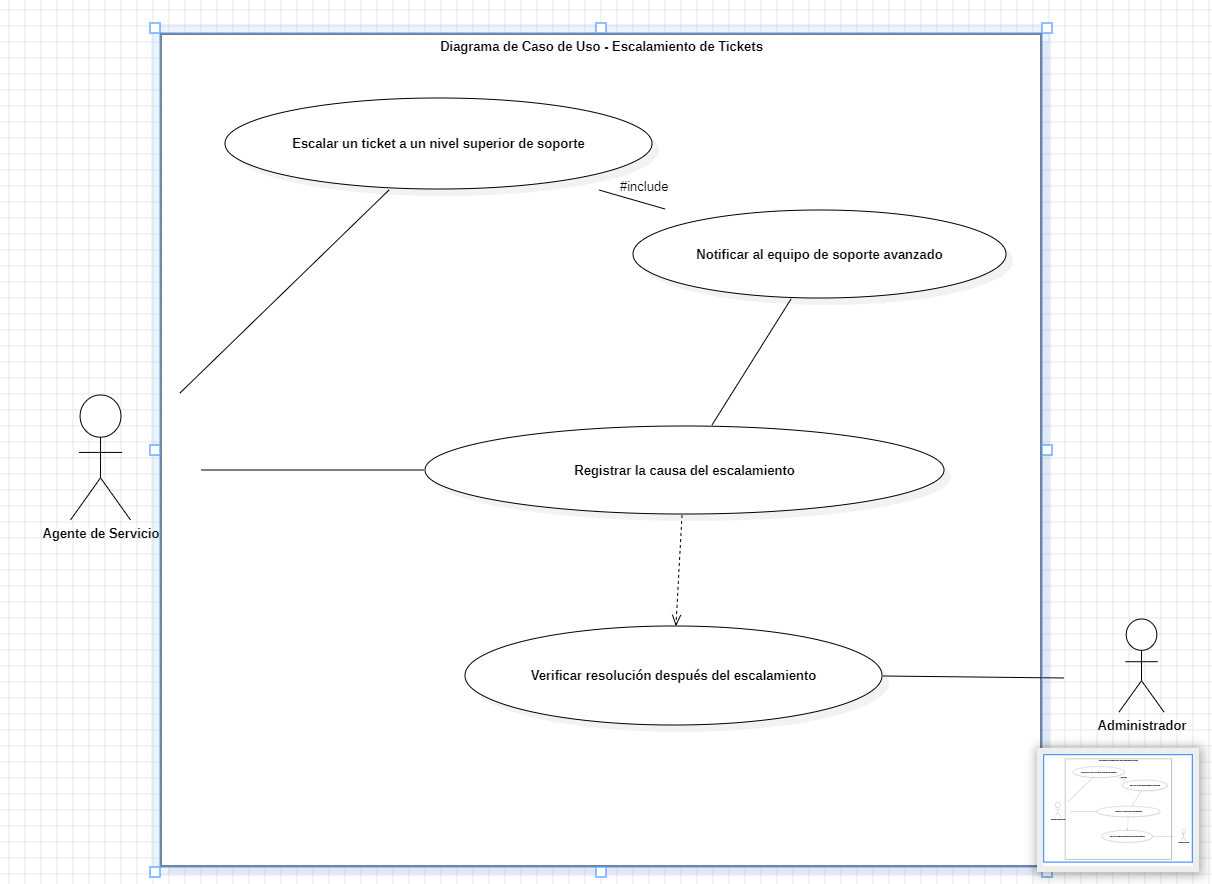
1. Caso de uso 6: Seguimiento de Tickets



11-Figura 11: Seguimiento de Ticket

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| GV-A0-00 | | Seguimiento de Ticket | | | | |
| Versión | | 1.0 | | | | |
| Descripción | | El sistema debe ser capaz de proporcionar información almacenada para la debida gestión y seguimiento de los reportes por parte de los agentes de servicio, administradores y clientes. | | | | |
| Precondición | | El cliente debe haber abierto un ticket de servicio | | | | |
| Secuencia normal | | Paso | | Acción | | |
| 1 | | El actor (Cliente) ingresa a sistema. | | |
| 2 | | Consulta tickets abiertos | | |
|  | | 3 | | El actor (Agente de servicio) mantiene actualizada la información de los tickets | |
| 4 | | El actor (Administrador) puede generar los informes abiertos. | |
| Importancia | | | Media | | | | |
| Estado | | | En Ejecución | | | | |
| Estabilidad | | | Alta | | | | |

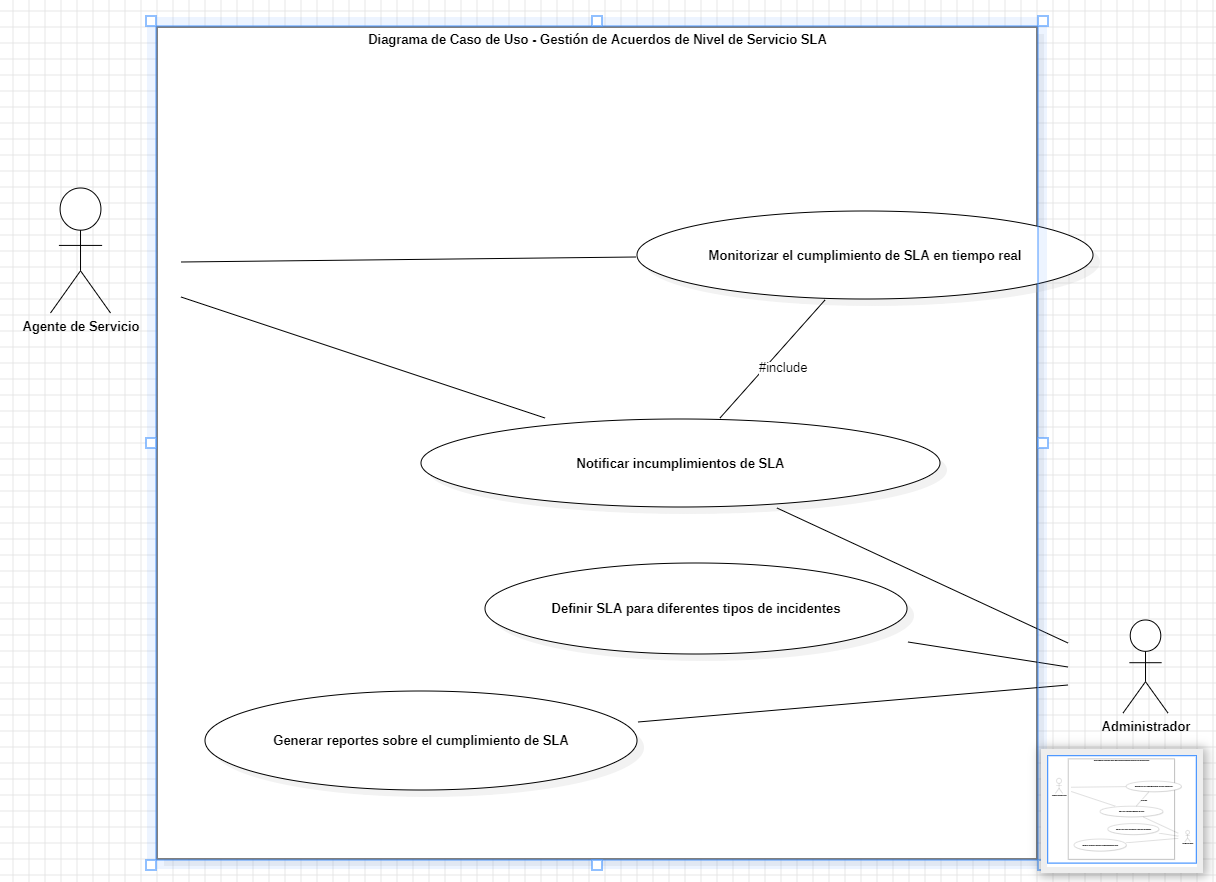
1. Caso de uso 7: Escalamiento de Tickets



12-Figura 12: Gestión para el escalamiento de Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión para el escalamiento de ticket | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de gestionar el escalamiento de ticket de ser necesario por los agentes de servicio. | | | | |
| Precondición | El cliente debe haber abierto un ticket de servicio | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El actor (Agente de servicio) ingresa a sistema. | | |
| 2 | | Se escala el ticket a nivel superior | | |
|  | 3 | Registra causas de escalamiento | |
| 4 | El actor (Administrador) verificar información por escalamiento. | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

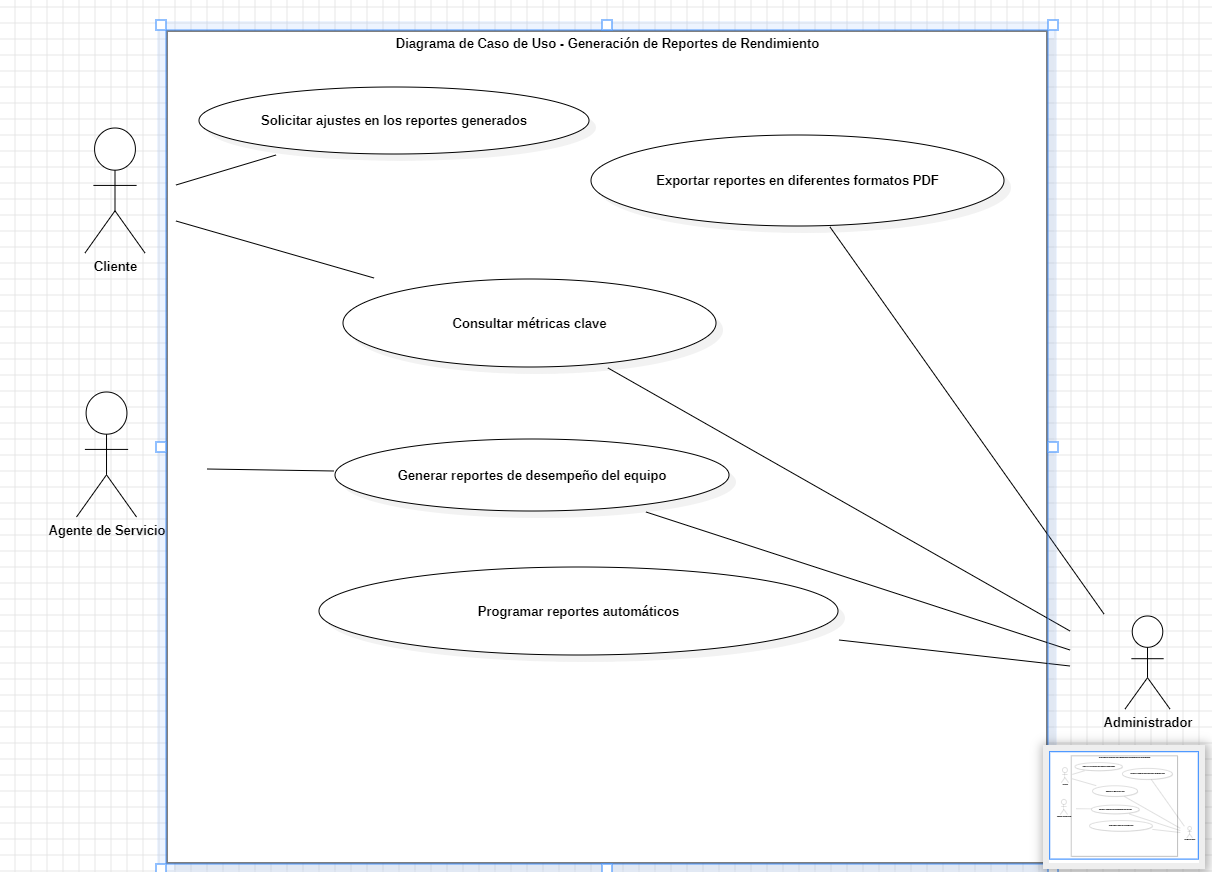
1. Caso de uso 8: Gestión de Acuerdos de Nivel de Servicio (SLA)



13-Figura 13: Gestión de Acuerdos de nivel de servicio SLA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de Acuerdos de Nivel de servicio SLA | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de cumplir con los servicios SLA. | | | | |
| Precondición | Parametrización actualizada. | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El actor (Agente de servicio) Monitoriza el cumplimiento en tiempo real. | | |
| 2 | | Se le notifica el incumplimiento SLA al Actor ( Administrador) | | |
|  | 3 | El actor (Administrador) define los SLA para diferentes tipos de incidentes. | |
| 4 | El actor (Administrador) puede generar los reportes abiertos de cumplimiento SLA. | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

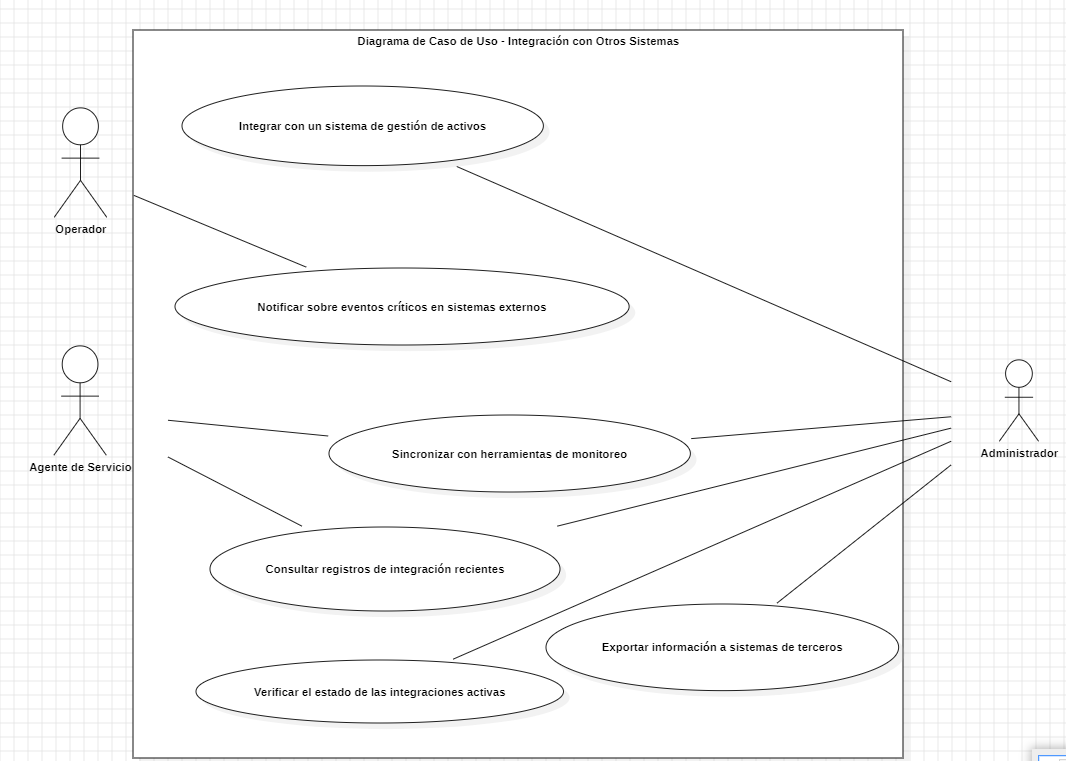
1. Caso de uso 9: Reportes de Rendimiento



14-Figura 14: Gestión de reportes de rendimiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de reportes de rendimiento | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de presentar reportes para medir y analizar la eficiencia y efectividad del equipo de servicio en la mesa de ayuda. Este tipo de reporte es utilizado para identificar áreas de mejora, supervisar el rendimiento de los agentes, y optimizar los procesos de atención al cliente. | | | | |
| Precondición | El administrador debe estar autenticado en el sistema y tener los permisos necesarios para acceder a los reportes de desempeño. Debe existir un registro histórico de tickets atendidos en la mesa de servicio. Los indicadores de rendimiento deben estar configurados previamente en el sistema. | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El actor (Administrador) genera reportes de desempeño de equipo. | | |
| 2 | | Exportar reportes en diferentes formatos PDF | | |
|  | 3 | Puede configurar el sistema para que genere y envíe reportes automáticamente en horarios predefinidos | |
| 4 | Los clientes tienen la opción de solicitar cambios o ajustes en los reportes para incluir información adicional o para modificar la presentación de los datos. | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

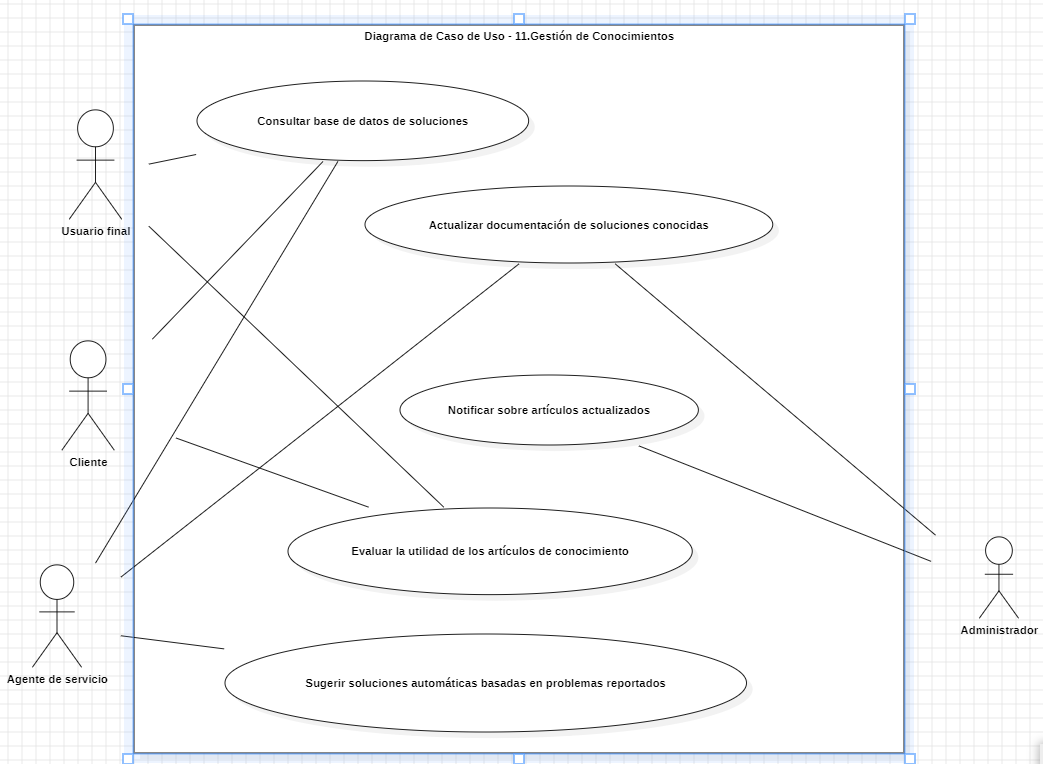
1. Caso de uso 10: Integración con Otros Sistemas



15-Figura 15: Integración con otros sistemas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Integración con sistemas externos | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe tener la capacidad de ser requerido poder establecer enlaces con sistemas externos. Esta integración permite que la mesa de servicio funcione de manera más eficiente al intercambiar datos automáticamente y gestionar eventos críticos. | | | | |
| Precondición | El sistema debe estar configurado para permitir integraciones con otros sistemas. Los usuarios deben tener los permisos adecuados para gestionar o consultar las integraciones. | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El administrador puede conectar el sistema de la mesa de servicio con un sistema de gestión de activos. | | |
| 2 | | Puede integrarse con herramientas de monitoreo para recibir alertas automáticas sobre fallos, errores o cualquier anomalía en la infraestructura tecnológica. | | |
|  | 3 | Los agente de servicio de la mesa de servicio pueden enviar notificaciones a otros sistemas o equipos externos | |
| 4 | Exportar información a sistemas de terceros (Administrador) | |
|  | 5 | El administrador y el agente de servicio pueden acceder a un historial de las integraciones recientes para verificar si los datos se han sincronizado correctamente o si ha habido errores en la comunicación con otros sistemas. | |
|  | 6 | El administrador puede supervisar el estado de las integraciones actuales para asegurarse de que están funcionando correctamente. | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

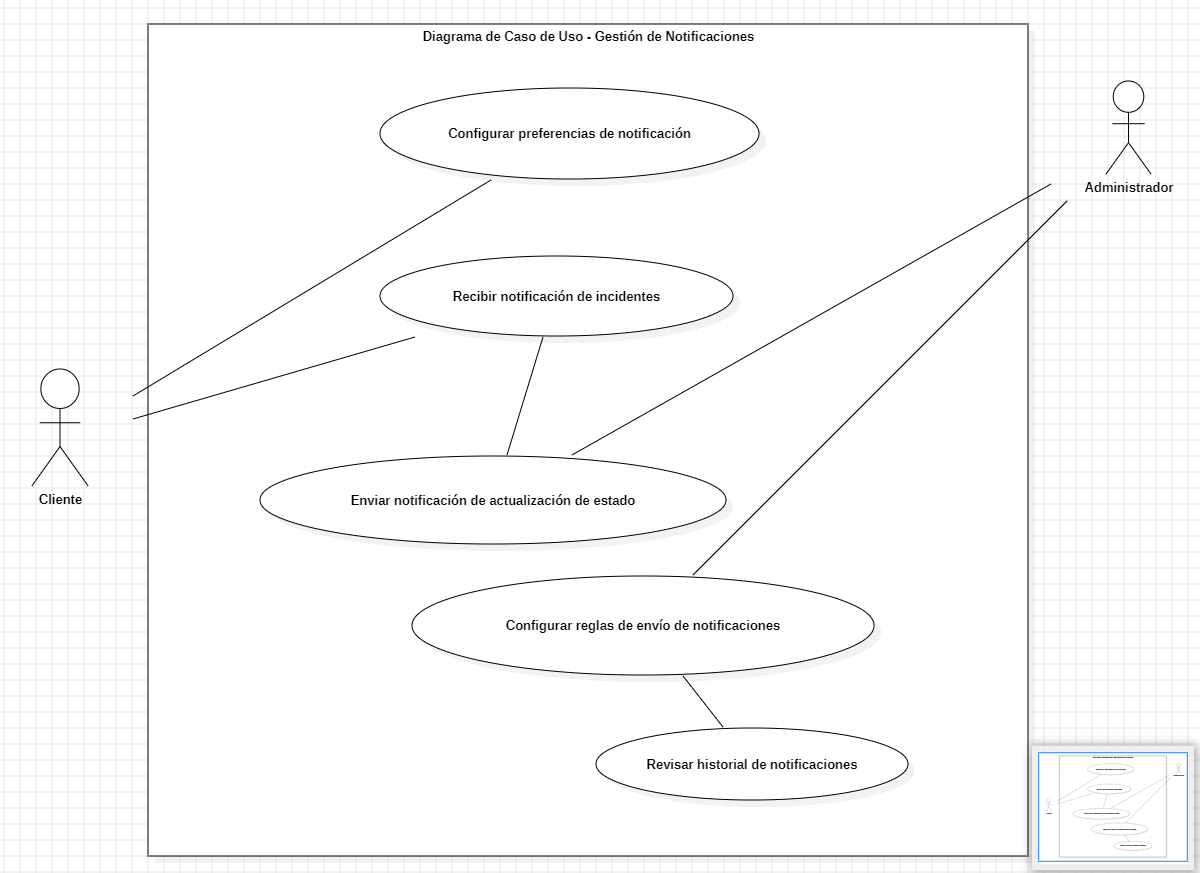
1. Caso de uso 11: Gestión de Conocimientos



16-Figura 16: Gestión de Conocimiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de conocimientos (Base de Datos) | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de permitir crear, gestionar y acceder a la información almacenada en la base de conocimiento de la mesa de servicio. | | | | |
| Precondición | La base de conocimiento debe estar activa y accesible para los usuarios. Los actores deben tener permisos adecuados para crear, editar, aprobar o eliminar artículos. | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | Los agentes de servicio pueden crear nuevos artículos para la base de conocimiento, documentando soluciones a problemas recurrentes, procedimientos técnicos o cualquier información útil | | |
| 2 | | Los agentes tienen la capacidad de actualizar o modificar artículos ya existentes para mantener la información relevante y precisa. | | |
|  | 3 | Antes de que un artículo se publique en la base de conocimiento, el administrador revisa el contenido y decide si lo aprueba para su publicación o lo rechaza si no cumple con los estándares de calidad o relevancia. | |
| 4 | El administrador tiene la autoridad para eliminar artículos que ya no son relevantes, están obsoletos o contienen información incorrecta.. | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

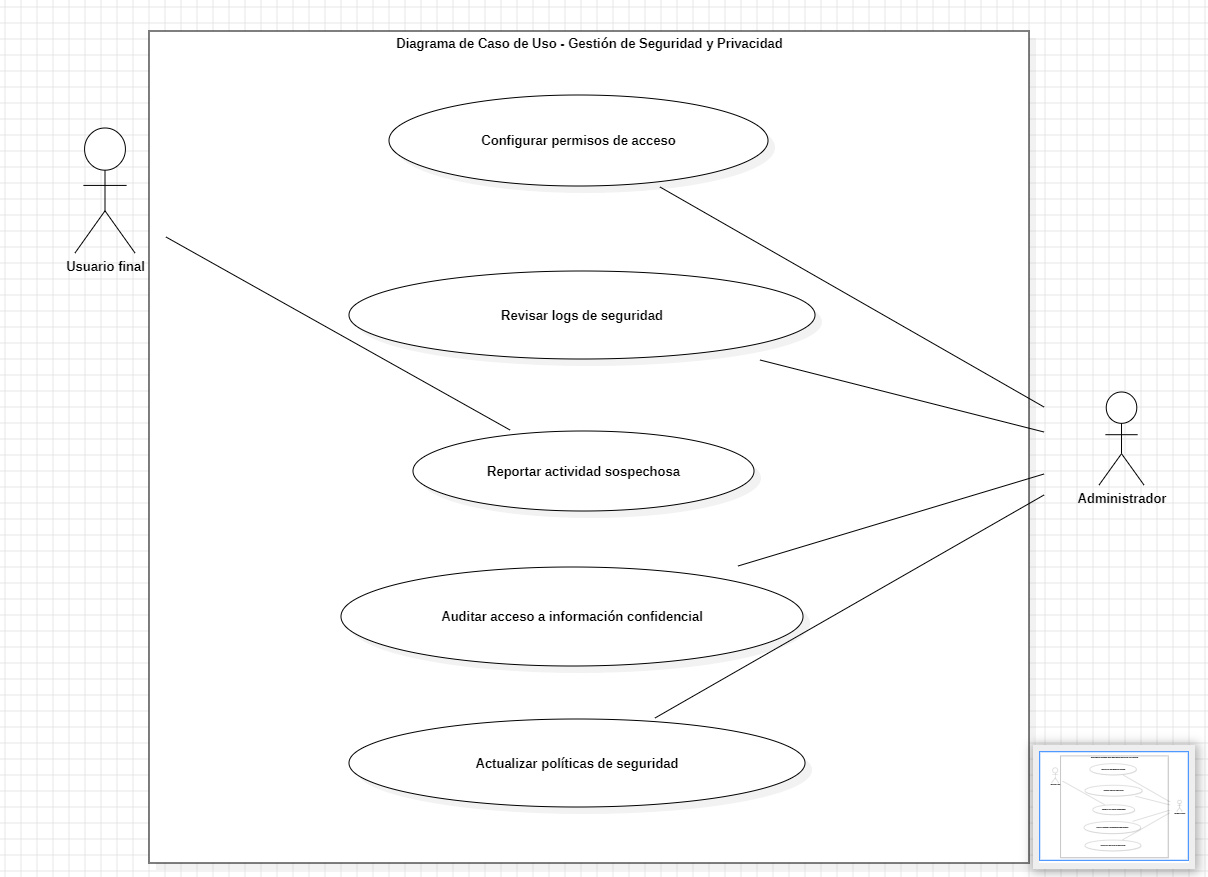
1. Caso de uso 12: Gestión de Notificaciones



17-Figura 17: Gestion de notificaciones

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de Notificaciones | | | | |
| Versión | 1.0 | | | | |
| Descripción | La función de notificaciones se centra en la configuración, envío y recepción de notificaciones relacionadas con la gestión de la mesa de servicio. Las notificaciones son mensajes automáticos que se envían para informar a los usuarios sobre cambios importantes, actualizaciones en tickets o eventos críticos. | | | | |
| Precondición | El cliente debe haber abierto un ticket de servicio | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | Configurar preferencias de notificación (Cliente) | | |
| 2 | | Configurar preferencias de notificación (Cliente) | | |
|  | 3 | Enviar notificación de actualización de estado (Administrador) | |
| 4 | Los clientes reciben notificaciones cuando ocurren incidentes relevantes que afectan sus solicitudes. | |
|  | 5 | Configurar reglas de envío de notificaciones (Administrador): | |
|  | 6 | Revisar historial de notificaciones (Administrador): | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

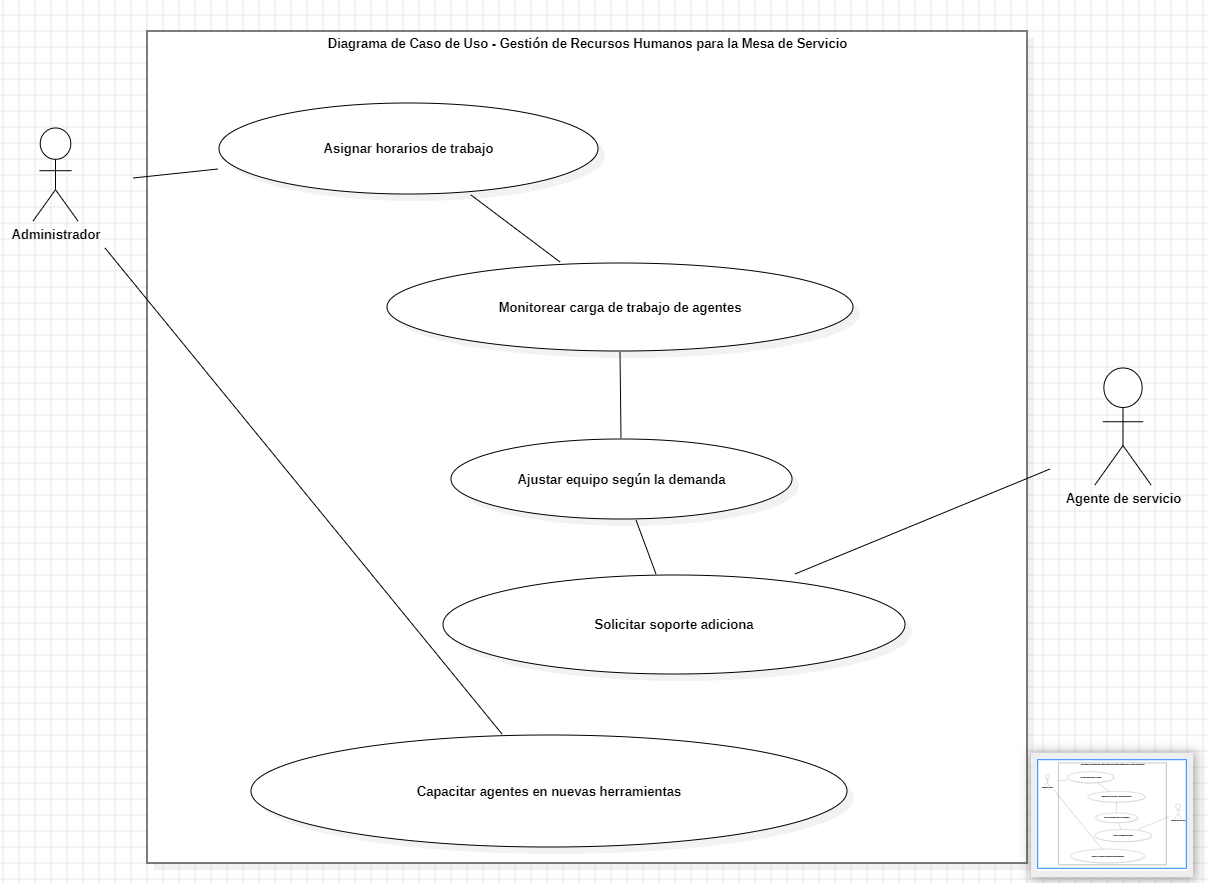
1. Caso de uso 13: Gestión de Seguridad y Privacidad



18-Figura 18: Gestión de seguridad y privacidad

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de Seguridad y Privacidad | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de gestionar y administrar las clases de acceso a sistema y módulos. Generar reportes de login, entre otros. | | | | |
| Precondición | Los autores deben estar registrado en sistemas. | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El administrador configura los permisos de acceso | | |
| 2 | | Se revisan los logs de seguridad de cada cliente, agente y administradores. | | |
|  | 3 | Actualiza políticas de seguridad | |
| 4 | Se notifica a cliente final sobre alguna incidencia sospechosa | |
| Importancia | Alta | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

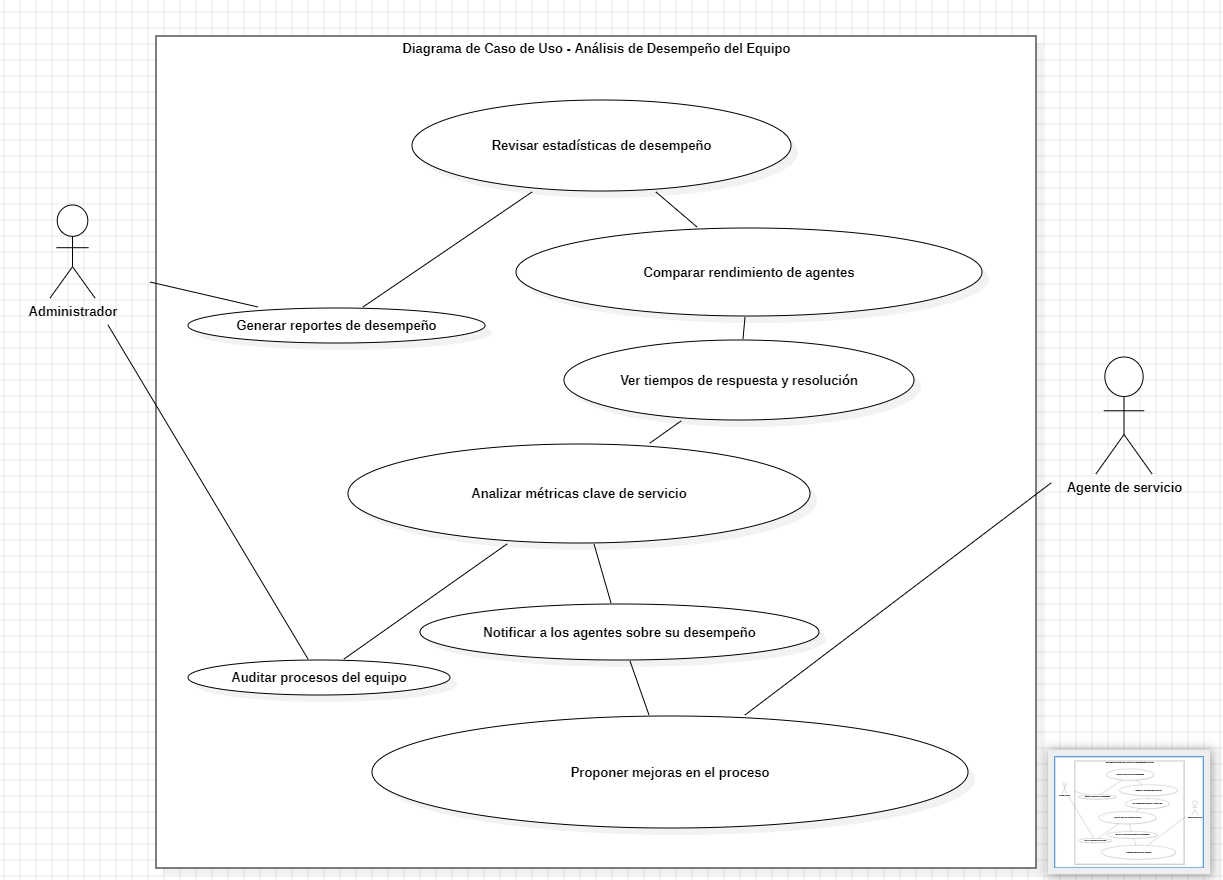
1. Caso de uso 14: Gestión de Recursos Humanos para la Mesa de Servicio



19-Figura 19: Gestion de recursos humanos para la mesa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Gestión de recursos humanos para la mesa de servicio | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe ser capaz de administrador el flujo de trabajo de cada agente de servicio registrado en sistema debe poder relevar y asignar funciones. | | | | |
| Precondición | El cliente debe haber abierto un ticket de servicio | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El actor administrador asigna horarios de trabajo | | |
| 2 | | Monitorea la carga de trabajo asignada a cada agente | | |
|  | 3 | El actor administrador puede ajustar la carga de trabajo | |
| 4 | El actor (Administrador) debe mantener a su personal capacitado | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

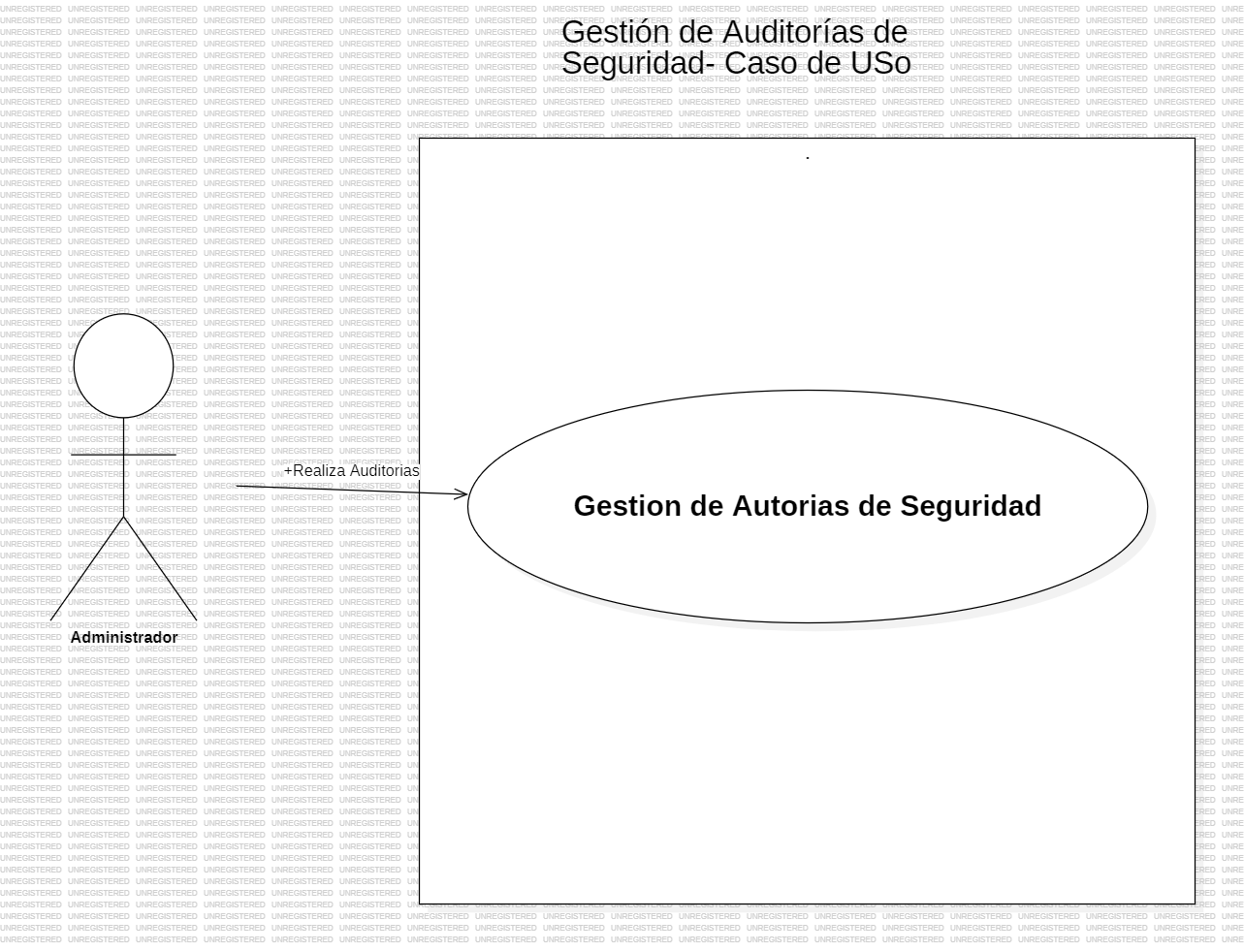
1. Caso de uso 15: Análisis de Desempeño del Equipo



20-Figura 20: Analisis de Desempeño del equipo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GV-A0-00 | Análisis y desempeño de equipo | | | | |
| Versión | 1.0 | | | | |
| Descripción | El sistema debe evaluar y emitir reportes del desempeño del equipo encargado para la mesa de servicio. | | | | |
| Precondición | No existe precondición | | | | |
| Secuencia normal | Paso | | Acción | | |
| 1 | | El actor Administrador genera reportes de desempeño | | |
| 2 | | Revisa los reportes generados | | |
|  | 3 | Compara rendimiento de equipo | |
| 4 | Evalúa tiempo de respuesta de cada ticket finalizado | |
|  | 5. | El agente de servicio propone mejorar en los procesos operativos. | |
|  | 5. | El administrador notifica a los agentes sobre su desempeño | |
| Importancia | Media | | | |
| Estado | En Ejecución | | | |
| Estabilidad | Alta | | | |

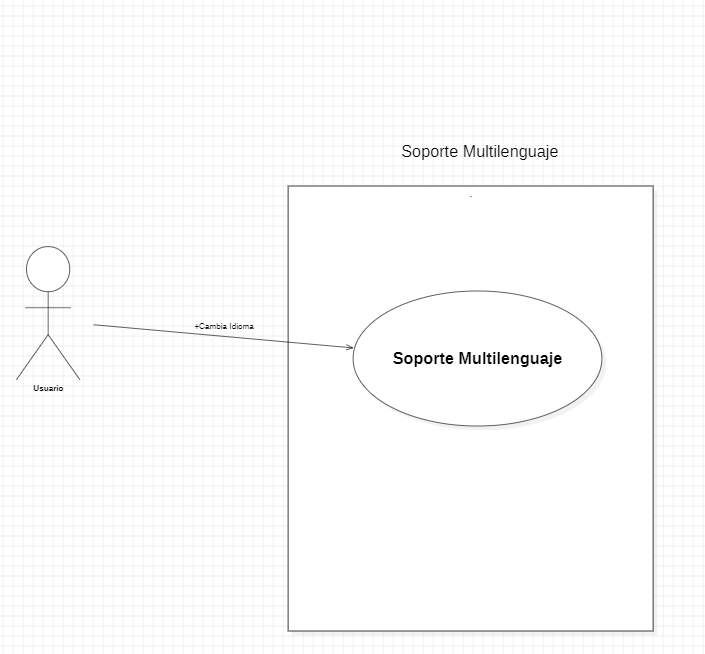
1. Caso de uso 16:



1. Caso de uso 17:



1. Caso de uso 18:



# **Diagrama de Secuencia (De todos los requerimientos de su proyecto)**

# Escala de tiempo Descripción generada automáticamente con confianza media

21-Figura 21: Seguimiento de Ticket

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de proporcionar información almacenada para la debida gestión y seguimiento de los reportes por parte de los agentes de servicio, administradores y clientes. |

# Seguimiento de Ticket

|  |  |
| --- | --- |
| Paso | Acción |
| 1 | El actor (Cliente) ingresa al sistema. |
| 2 | Consulta tickets abiertos. |
| 3 | El actor (Agente de servicio) mantiene actualizada la información de los tickets. |
| 4 | El actor (Administrador) puede generar los informes abiertos. |
| Importancia | Media |
| Estado | En Ejecución |
| Estabilidad | Alta |

Diagrama

Descripción generada automáticamente

22-Figura 22: Gestión de Ticket

Diagrama de Caso de Uso - Gestión de Tickets

GV-A0-00 Gestión de Tickets

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de permitir la gestión de tickets, generando grupos de tickets, realizando estadísticas y presentando informes que luego pueden ser analizados por la junta directiva. |
| Precondición | El administrador debe tener acceso al sistema de gestión de tickets. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Administrador) genera un grupo de tickets. |
| 2 | El sistema de gestión de tickets guarda el grupo de tickets y lo confirma. |
| 3 | El actor (Agente de Servicio) realiza estadísticas de los tickets. |
| 4 | El sistema de gestión de tickets filtra los resultados y presenta estadísticas. |
| 5 | El actor (Junta Directiva) analiza las estadísticas proporcionadas. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

Diagrama

Descripción generada automáticamente

23- Gestión 23: Proceso de Creación y Gestión de Ticket

Diagrama de Secuencia - Proceso de Creación y Gestión de Tickets

|  |  |
| --- | --- |
| Actor | Acción |
| Cliente | Crear Ticket |
| Sistema de Gestión de Tickets | Validar Datos del Ticket |
| Base de Datos | Guardar Datos del Ticket |
| Administrador | Validar Datos del Ticket |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de permitir la creación y gestión de tickets, validando y guardando los datos del ticket, y confirmando las acciones realizadas. |
| Precondición | El administrador debe tener acceso al sistema de gestión de tickets. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Cliente) crea un ticket. |
| 2 | El sistema de gestión de tickets valida los datos del ticket. |
| 3 | El sistema de gestión de tickets guarda los datos del ticket. |
| 4 | La base de datos guarda los datos del ticket. |
| 5 | El sistema de gestión de tickets confirma el guardado de los datos. |
| 6 | El actor (Administrador) valida los datos del ticket. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

Diagrama

Descripción generada automáticamente

24-Figura 24: Manejo de Prioridades

Diagrama de Secuencia - Manejo de Prioridades en el Sistema de Tickets

|  |  |
| --- | --- |
| Actor | Acción |
| Cliente | Crear Ticket |
| Sistema de Gestión de Tickets | Ingresar Ticket |
| Sistema de Gestión de Tickets | Validar Datos del Ticket |
| Sistema de Gestión de Tickets | Determinar Prioridad del Ticket |
| Sistema de Gestión de Tickets | Notificar al Agente de Servicio (si es alta prioridad) |
| Sistema de Gestión de Tickets | Proceder con Procedimientos Estándar (si es prioridad media) |
| Sistema de Gestión de Tickets | Proceder con Procedimientos de Baja Urgencia (si es baja prioridad) |
| Base de Datos | Guardar Datos del Ticket |
| Base de Datos | Confirmar Guardado |
| Agente de Servicio | Recibir Notificación de Alta Prioridad |
| Agente de Servicio | Brindar Soluciones |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de manejar las prioridades de los tickets, determinando la urgencia y notificando al agente de servicio según la prioridad asignada. |
| Precondición | El administrador debe tener acceso al sistema de gestión de tickets. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Cliente) crea un ticket. |
| 2 | El sistema de gestión de tickets ingresa el ticket. |
| 3 | El sistema de gestión de tickets valida los datos del ticket. |
| 4 | El sistema de gestión de tickets determina la prioridad del ticket. |
| 5 | El sistema de gestión de tickets notifica al agente de servicio si el ticket es de alta prioridad. |
| 6 | El sistema de gestión de tickets procede con procedimientos estándar si el ticket es de prioridad media. |
| 7 | El sistema de gestión de tickets procede con procedimientos de baja urgencia si el ticket es de baja prioridad. |
| 8 | La base de datos guarda los datos del ticket. |
| 9 | La base de datos confirma el guardado. |
| 10 | El actor (Agente de Servicio) recibe la notificación de alta prioridad. |
| 11 | El actor (Agente de Servicio) brinda soluciones. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

Diagrama

Descripción generada automáticamente

25-Figura 25: Evaluación de Satisfacción del usuario

Diagrama de Secuencia - Evaluación de Satisfacción del Usuario

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de evaluar la satisfacción del usuario, ingresando y confirmando evaluaciones, enviando encuestas, y presentando resultados para su análisis. |
| Precondición | El administrador debe tener acceso al sistema de gestión de satisfacción. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Usuario Final) evalúa la satisfacción. |
| 2 | El actor (Usuario Final) envía retroalimentación. |
| 3 | El sistema de gestión de satisfacción ingresa la evaluación. |
| 4 | El sistema de gestión de satisfacción confirma el ingreso. |
| 5 | El sistema de gestión de satisfacción envía una encuesta. |
| 6 | El sistema de gestión de satisfacción consulta el historial. |
| 7 | La base de datos devuelve el historial. |
| 8 | El sistema de gestión de satisfacción consulta los resultados. |
| 9 | El sistema de gestión de satisfacción presenta los resultados. |
| 10 | El actor (Administrador) consulta los resultados. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

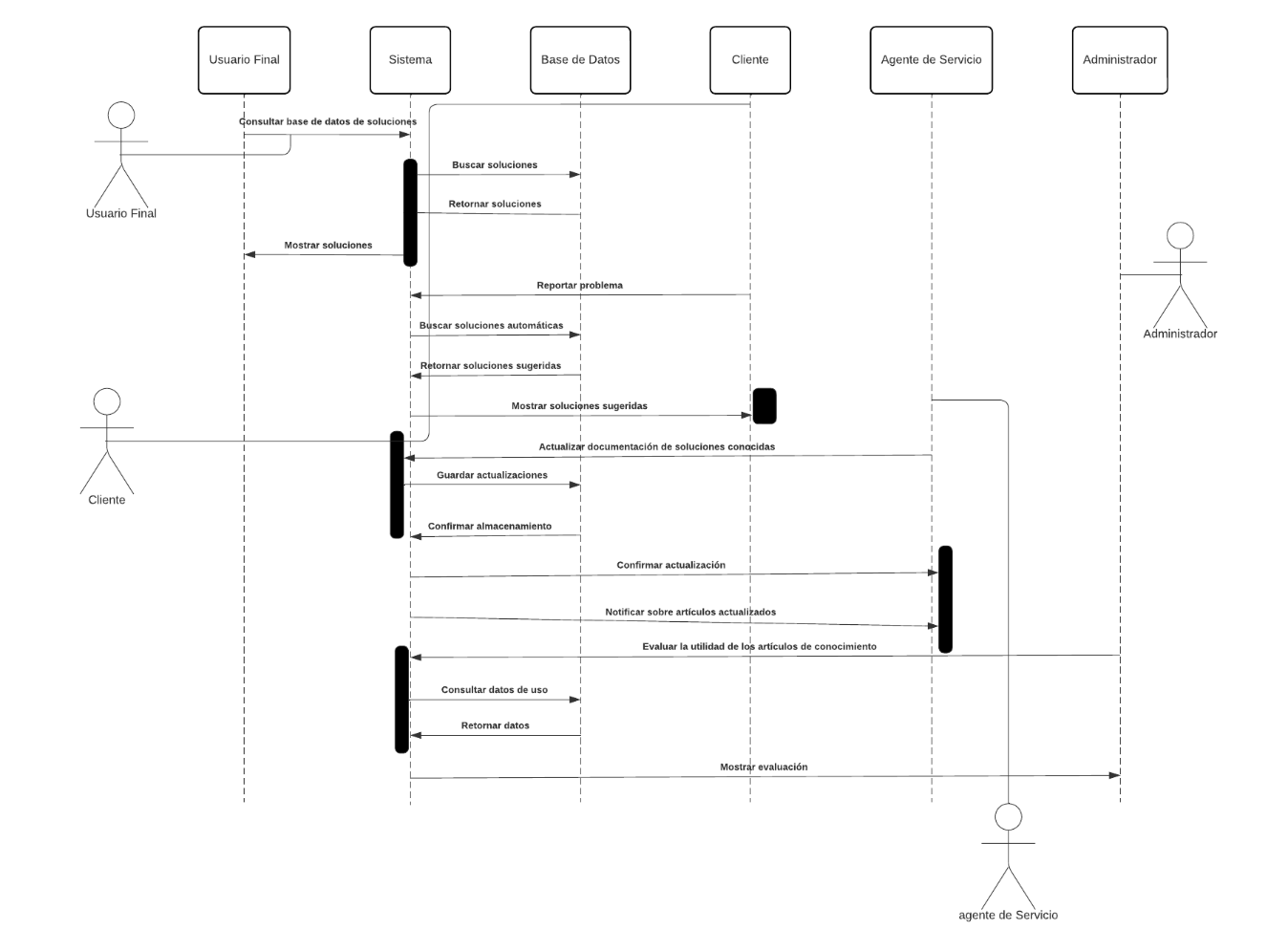
Diagrama

Descripción generada automáticamente

26-Figura 26: Evaluación de Satisfacción del usuario

Diagrama de Secuencia - Evaluación de Satisfacción del Usuario

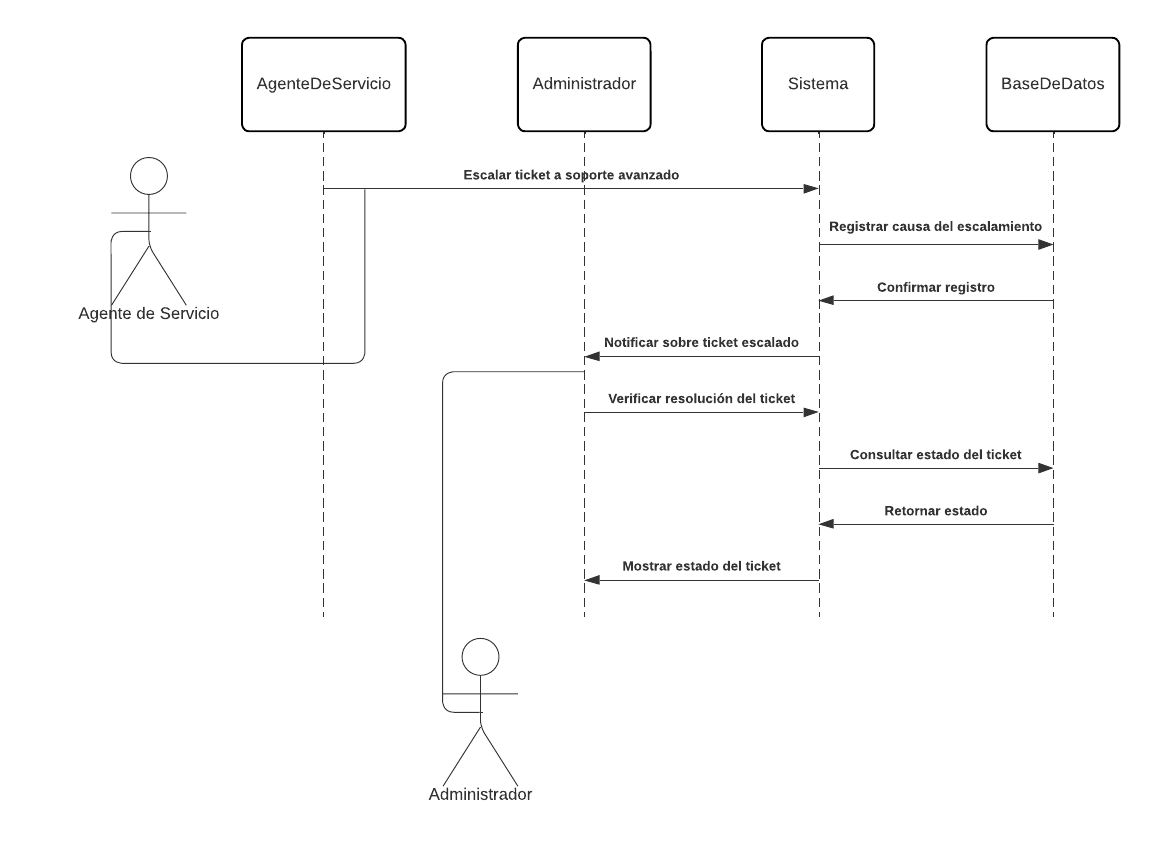
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de evaluar la satisfacción del usuario, ingresando y confirmando evaluaciones, enviando encuestas, y presentando resultados para su análisis. |
| Precondición | El administrador debe tener acceso al sistema de gestión de satisfacción. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Usuario Final) evalúa la satisfacción. |
| 2 | El actor (Usuario Final) envía retroalimentación. |
| 3 | El sistema de gestión de satisfacción ingresa la evaluación. |
| 4 | El sistema de gestión de satisfacción confirma el ingreso. |
| 5 | El sistema de gestión de satisfacción envía una encuesta. |
| 6 | El sistema de gestión de satisfacción consulta el historial. |
| 7 | La base de datos devuelve el historial. |
| 8 | El sistema de gestión de satisfacción consulta los resultados. |
| 9 | El sistema de gestión de satisfacción presenta los resultados. |
| 10 | El actor (Administrador) consulta los resultados. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



27-Figura: Manejo de Solicitudes

Diagrama de Secuencia - Manejo de Solicitudes

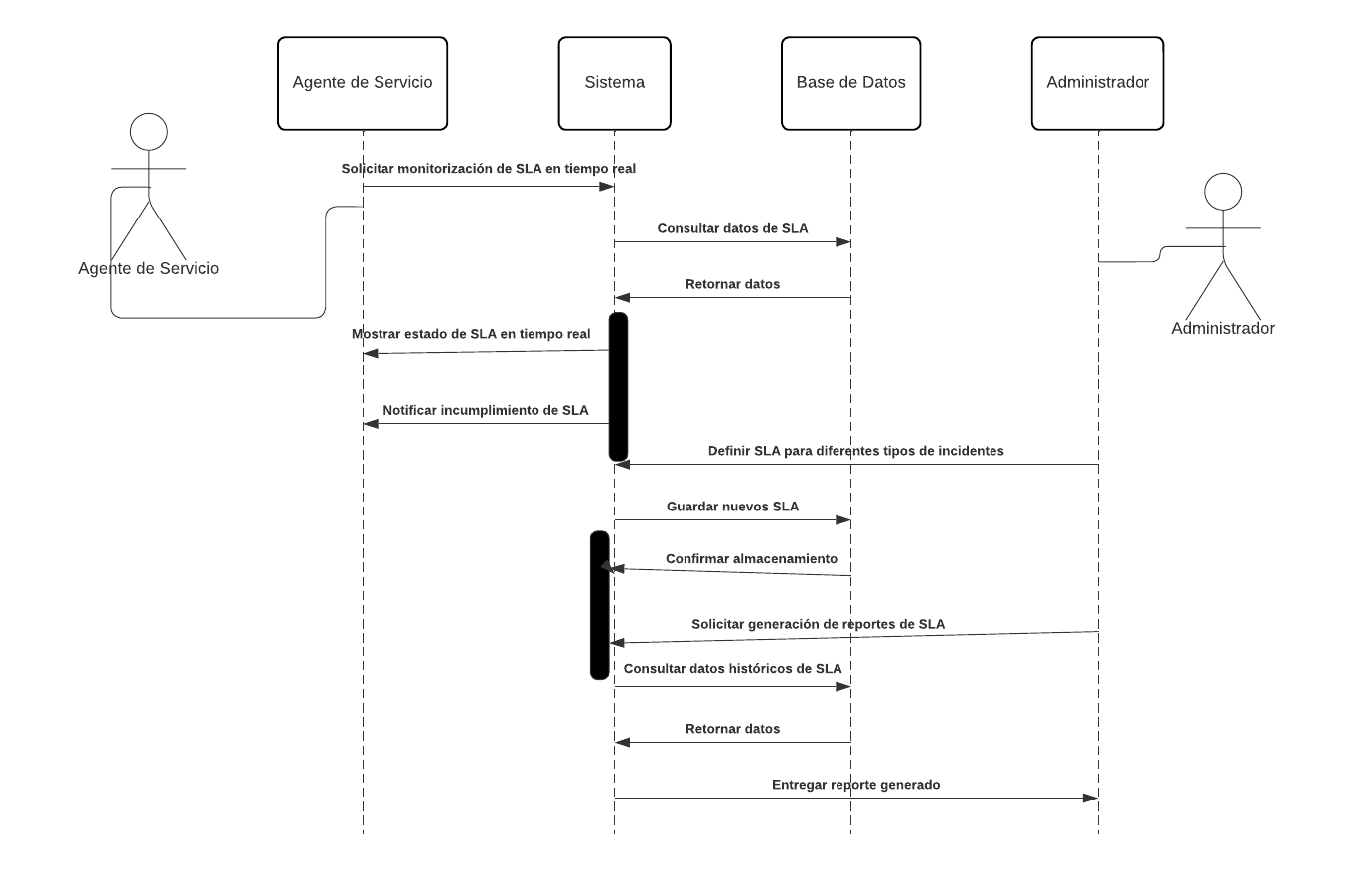
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de manejar las solicitudes, buscando y mostrando soluciones, y confirmando actualizaciones. |
| Precondición | El administrador debe tener acceso al sistema de gestión de solicitudes. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Usuario Final) recibe una solicitud. |
| 2 | El sistema busca la solicitud. |
| 3 | El sistema recibe soluciones. |
| 4 | El sistema muestra las soluciones. |
| 5 | La base de datos consulta la información. |
| 6 | La base de datos confirma la actualización. |
| 7 | El actor (Cliente) confirma la actualización. |
| 8 | El actor (Agente de Servicio) recibe las soluciones. |
| 9 | El actor (Administrador) consulta la información. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



28-Figura 28: Escalamiento de Tickets

Diagrama de Secuencia - Escalamiento de Tickets a Soporte Avanzado

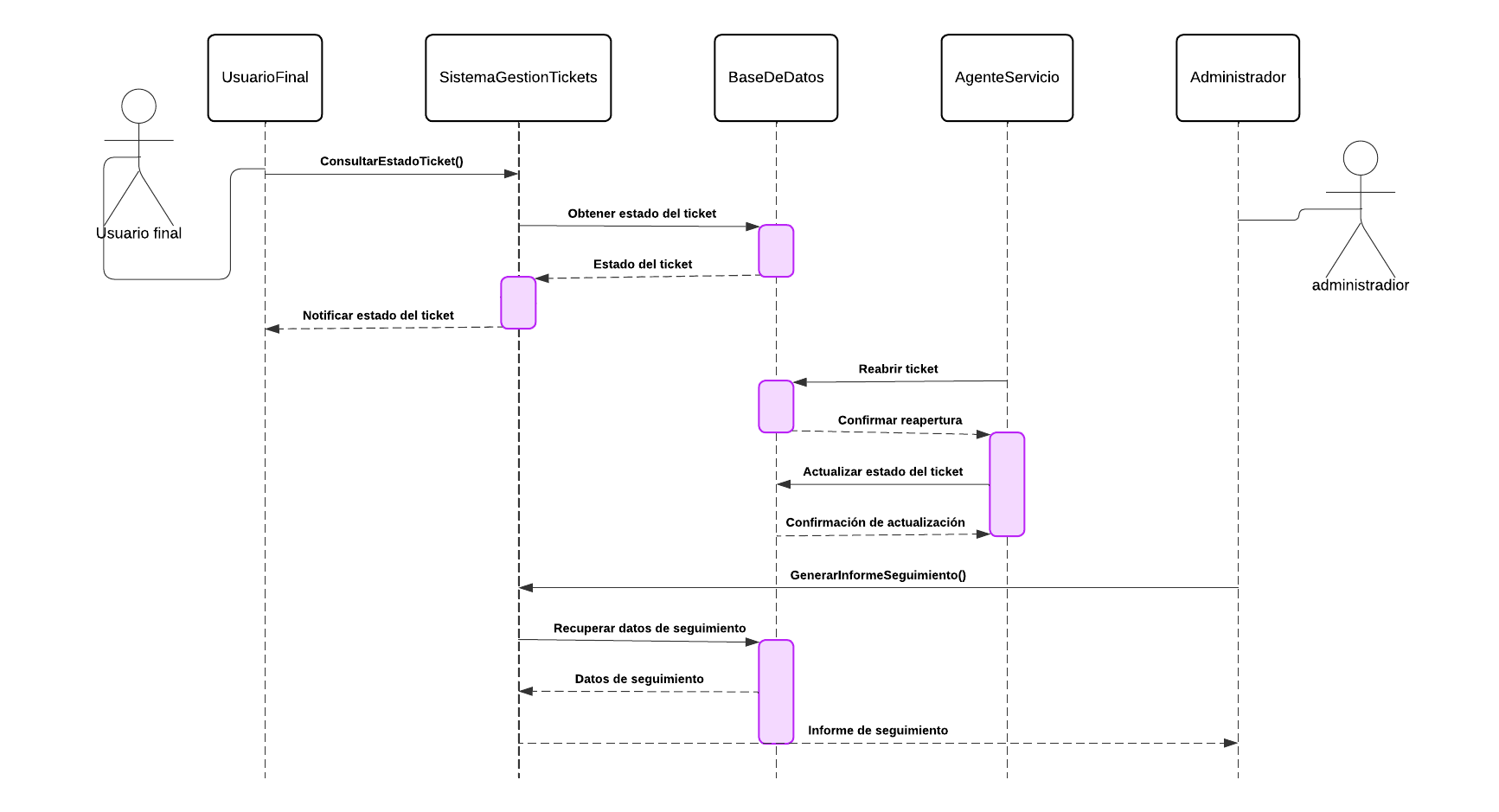
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de manejar el escalamiento de tickets a soporte avanzado, registrando la causa del escalamiento, notificando a los agentes y consultando el estado del ticket. |
| Precondición | El administrador debe tener acceso al sistema de gestión de tickets. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Agente de Servicio) escala un ticket a soporte avanzado. |
| 2 | El sistema registra la causa del escalamiento. |
| 3 | El sistema confirma el registro. |
| 4 | El sistema notifica sobre el ticket escalado. |
| 5 | El actor (Administrador) verifica la resolución del ticket. |
| 6 | El sistema consulta el estado del ticket. |
| 7 | La base de datos retorna el estado del ticket. |
| 8 | El sistema muestra el estado del ticket. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



29-Figura 29: Monitorización de Sla

Diagrama de Secuencia - Monitorización de SLA en Tiempo Real

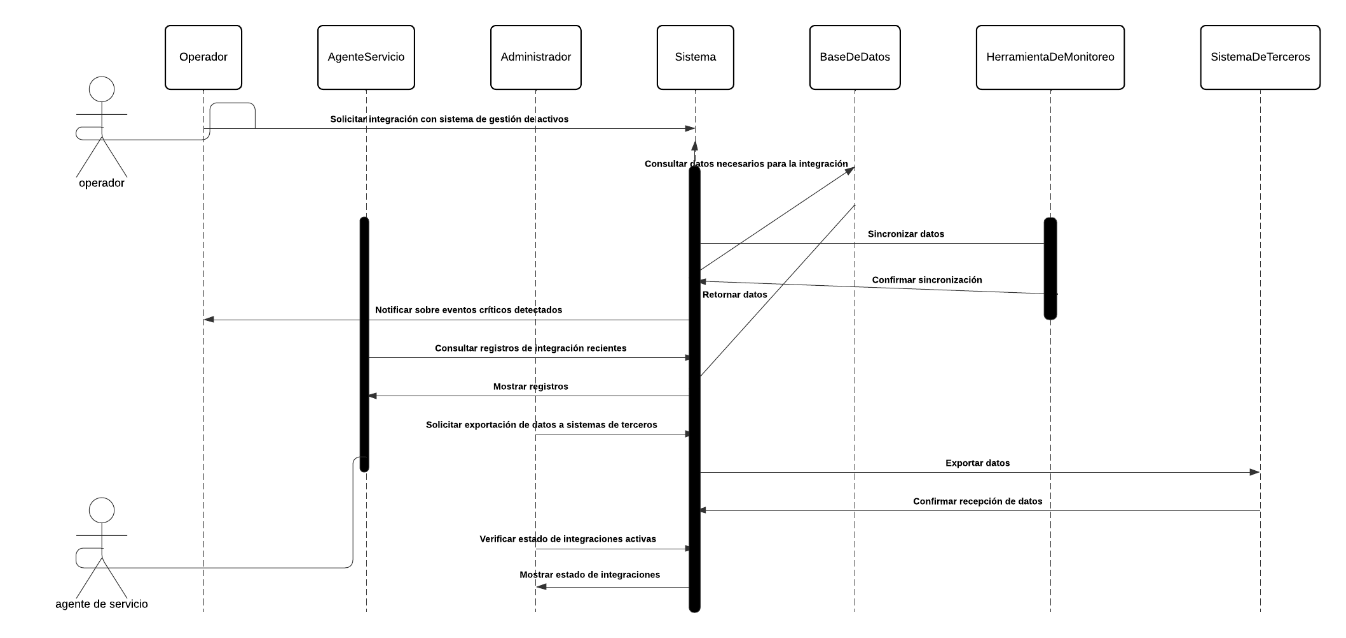
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de monitorizar los SLA en tiempo real, consultar y mostrar datos, notificar incumplimientos, definir nuevos SLA, y generar reportes. |
| Precondición | El administrador debe tener acceso al sistema de gestión de SLA. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Agente de Servicio) solicita la monitorización de SLA en tiempo real. |
| 2 | El sistema consulta los datos de SLA. |
| 3 | La base de datos retorna los datos. |
| 4 | El sistema muestra el estado de SLA en tiempo real. |
| 5 | El actor (Administrador) notifica el incumplimiento de SLA. |
| 6 | El actor (Administrador) define SLA para diferentes tipos de incidentes. |
| 7 | El sistema guarda los nuevos SLA. |
| 8 | El sistema confirma el almacenamiento. |
| 9 | El actor (Administrador) solicita la generación de reportes de SLA. |
| 10 | El sistema consulta los datos históricos de SLA. |
| 11 | La base de datos retorna los datos históricos. |
| 12 | El sistema entrega el reporte generado. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



30-Figura 30: Gestión de Ticket

Diagrama de Secuencia - Gestión de Tickets

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de gestionar los tickets, consultando y mostrando el estado, notificando a los administradores, reabriendo tickets, y generando informes de seguimiento. |
| Precondición | El administrador debe tener acceso al sistema de gestión de tickets. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Usuario Final) obtiene el estado del ticket. |
| 2 | El sistema de gestión de tickets consulta el estado del ticket. |
| 3 | La base de datos retorna el estado del ticket. |
| 4 | El sistema de gestión de tickets muestra el estado del ticket. |
| 5 | El actor (Administrador) notifica el estado del ticket. |
| 6 | El actor (Agente de Servicio) reabre el ticket. |
| 7 | El sistema de gestión de tickets confirma la reapertura. |
| 8 | El sistema de gestión de tickets actualiza el estado del ticket. |
| 9 | El sistema de gestión de tickets confirma la actualización. |
| 10 | El actor (Administrador) recupera los datos de seguimiento. |
| 11 | La base de datos retorna los datos de seguimiento. |
| 12 | El actor (Administrador) genera el informe de seguimiento. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

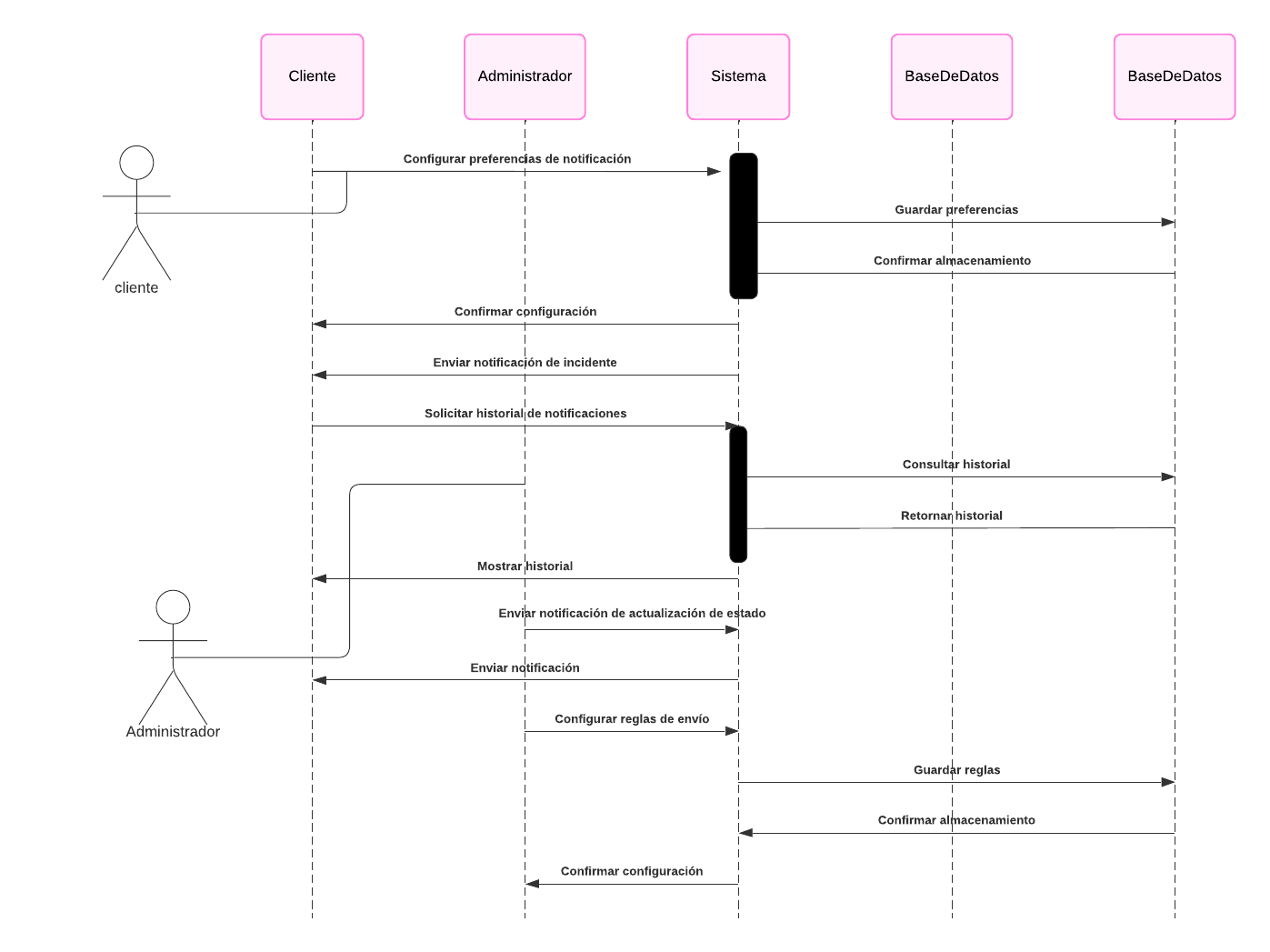


31-Figura 31: Integración con sistemas de gestión activos

Diagrama de Secuencia - Integración con Sistema de Gestión de Activos

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de integrar con el sistema de gestión de activos, sincronizando datos, consultando y mostrando registros, y monitoreando el estado de las integraciones. |
| Precondición | El administrador debe tener acceso al sistema de gestión de activos. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Operador) solicita la integración con el sistema de gestión de activos. |
| 2 | El sistema consulta los detalles necesarios para la integración. |
| 3 | La base de datos sincroniza los datos. |
| 4 | La base de datos confirma la sincronización. |
| 5 | El sistema retira los datos. |
| 6 | El actor (Agente de Servicio) revisa el historial de eventos críticos delegados. |
| 7 | El actor (Administrador) consulta los registros de migración recientes. |
| 8 | El sistema muestra los registros. |
| 9 | El sistema exporta los datos. |
| 10 | El sistema de terceros confirma la recepción de los datos. |
| 11 | El actor (Administrador) verifica el estado de las integraciones. |
| 12 | El actor (Administrador) monitorea el estado de las integraciones. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

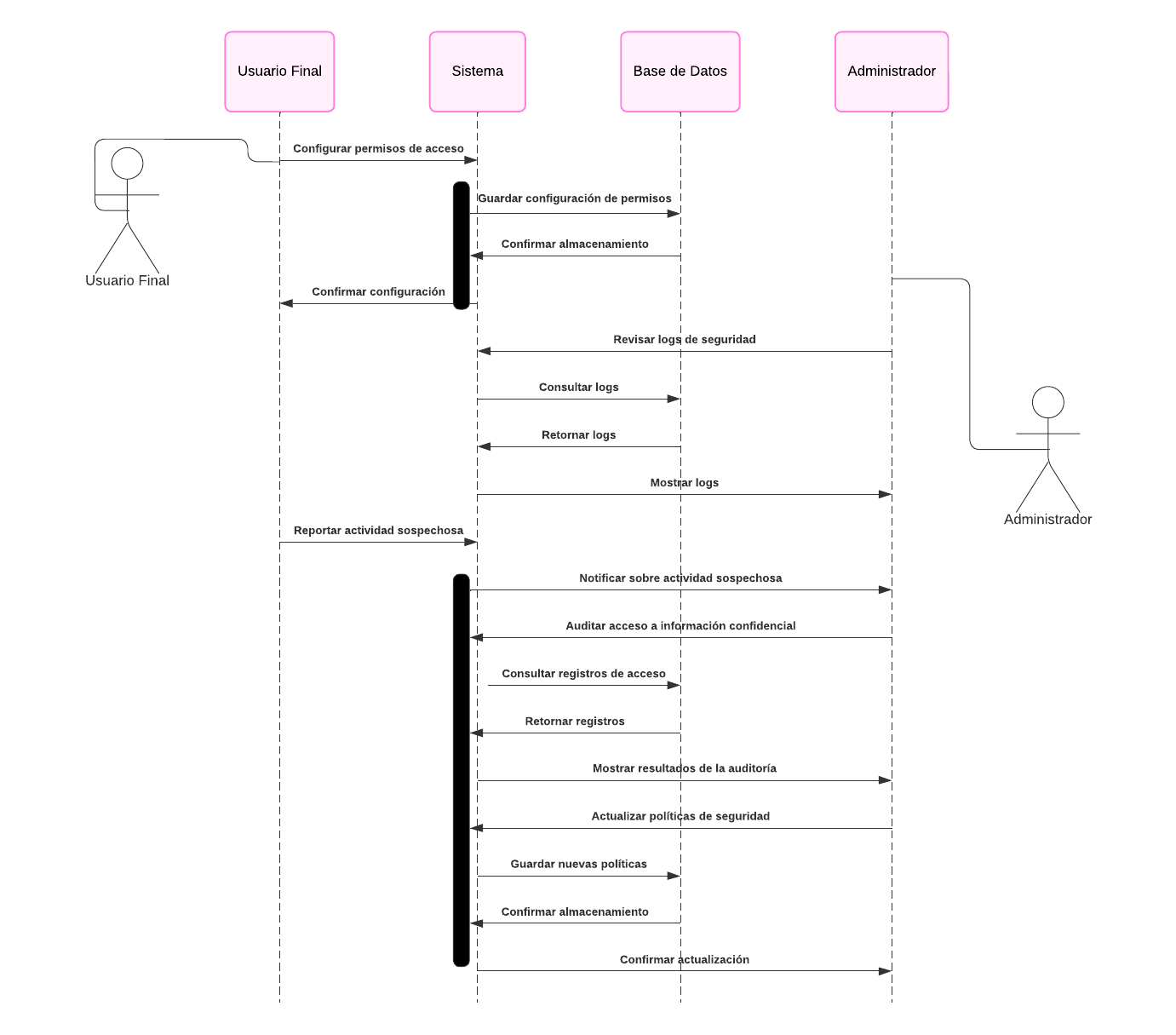
Diagrama de Secuencia - Configuración y Notificación de Incidentes



32-Figura 32: Configuración y Notificación de incidentes

Diagrama de Secuencia - Configuración y Notificación de Incidentes

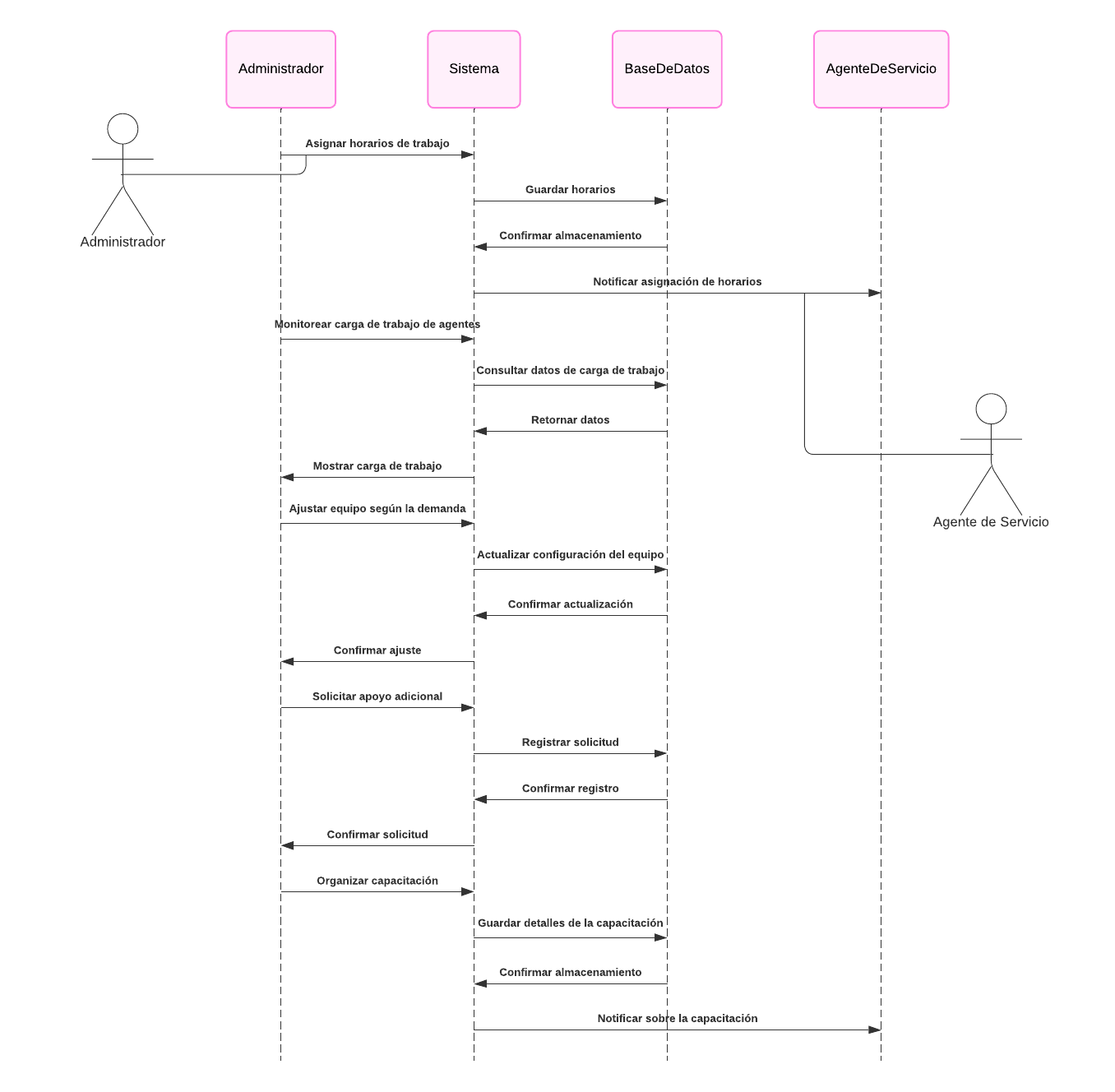
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de configurar y notificar incidentes, guardando preferencias y reglas, consultando y mostrando historiales, y enviando notificaciones. |
| Precondición | El administrador debe tener acceso al sistema de gestión de notificaciones. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Cliente) configura las preferencias de notificación. |
| 2 | El sistema guarda las preferencias. |
| 3 | El sistema confirma el almacenamiento. |
| 4 | El actor (Administrador) continúa con la configuración. |
| 5 | El actor (Administrador) envía una notificación de incidente. |
| 6 | El actor (Administrador) solicita el historial de notificaciones. |
| 7 | El sistema consulta el historial. |
| 8 | El sistema muestra el historial. |
| 9 | El actor (Administrador) envía una notificación de actualización de estado. |
| 10 | El actor (Administrador) envía una notificación. |
| 11 | El actor (Administrador) configura las reglas de envío. |
| 12 | El sistema guarda las reglas. |
| 13 | El sistema confirma el almacenamiento. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



33-Figura 33: Gestión de Seguridad y Acceso

Diagrama de Secuencia - Gestión de Seguridad y Acceso a Información

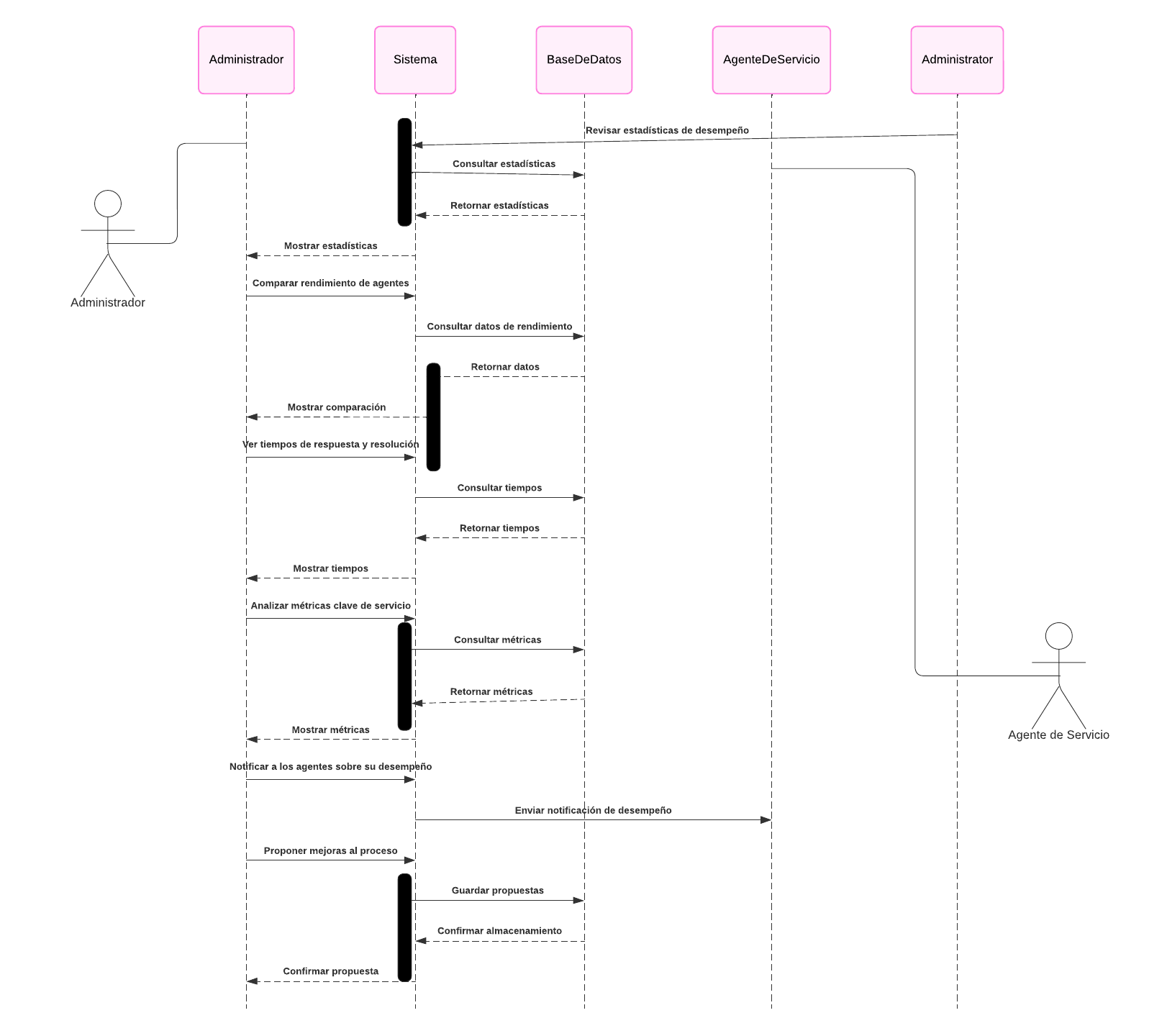
|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de gestionar la seguridad y el acceso a la información, configurando permisos, revisando logs, auditando accesos, y actualizando políticas de seguridad. |
| Precondición | El administrador debe tener acceso al sistema de gestión de seguridad. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Usuario Final) configura los permisos de acceso. |
| 2 | El sistema guarda la configuración de permisos. |
| 3 | El sistema confirma el almacenamiento. |
| 4 | El actor (Usuario Final) confirma la configuración. |
| 5 | El actor (Administrador) revisa los logs de seguridad. |
| 6 | El sistema consulta los logs. |
| 7 | La base de datos retorna los logs. |
| 8 | El sistema muestra los logs. |
| 9 | El actor (Usuario Final) reporta actividad sospechosa. |
| 10 | El actor (Administrador) notifica sobre la actividad sospechosa. |
| 11 | El actor (Administrador) audita el acceso a información confidencial. |
| 12 | El sistema consulta los registros de acceso. |
| 13 | La base de datos retorna los registros. |
| 14 | El sistema muestra los resultados de la auditoría. |
| 15 | El actor (Administrador) actualiza las políticas de seguridad. |
| 16 | El sistema guarda las nuevas políticas. |
| 17 | El sistema confirma el almacenamiento. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



34-Figura 34: Gestión de Horarios y Carga de Trabajo

Diagrama de Secuencia - Gestión de Horarios y Carga de Trabajo

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de gestionar los horarios y la carga de trabajo, monitoreando y ajustando la carga de trabajo de los agentes, notificando asignaciones, y organizando capacitaciones. |
| Precondición | El administrador debe tener acceso al sistema de gestión de horarios y carga de trabajo. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Administrador) guarda los horarios. |
| 2 | El sistema confirma el almacenamiento. |
| 3 | El sistema notifica la asignación de horarios. |
| 4 | El actor (Administrador) monitorea la carga de trabajo de los agentes. |
| 5 | El sistema consulta los datos de carga de trabajo. |
| 6 | La base de datos retorna los datos. |
| 7 | El sistema muestra la carga de trabajo. |
| 8 | El actor (Administrador) ajusta el equipo según la demanda. |
| 9 | El actor (Agente de Servicio) actualiza la configuración del equipo. |
| 10 | El sistema confirma la actualización. |
| 11 | El actor (Administrador) confirma el ajuste. |
| 12 | El actor (Administrador) solicita apoyo adicional. |
| 13 | El sistema registra la solicitud. |
| 14 | El sistema confirma el registro. |
| 15 | El actor (Administrador) confirma la solicitud. |
| 16 | El actor (Administrador) organiza la capacitación. |
| 17 | El sistema guarda los detalles de la capacitación. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |



35-Figura 35: Gestión de Métrica y Estadísticas

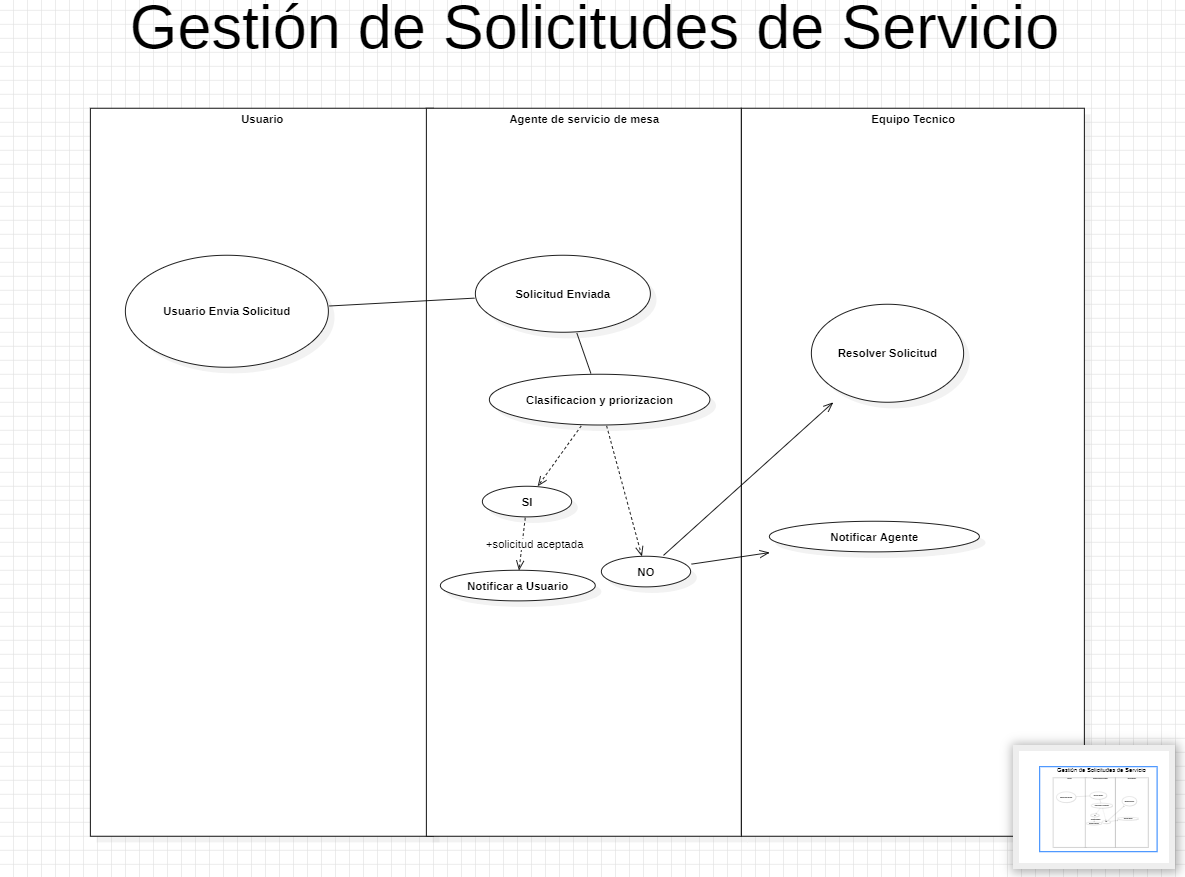
Diagrama de Secuencia - Gestión de Métricas y Estadísticas

|  |  |
| --- | --- |
| Versión | 1.0 |
| Descripción | El sistema debe ser capaz de gestionar las métricas y estadísticas, consultando y mostrando datos, comparando rendimientos, y proponiendo mejoras al proceso. |
| Precondición | El administrador debe tener acceso al sistema de gestión de métricas y estadísticas. |
| Secuencia normal |  |
| Paso | Acción |
| 1 | El actor (Administrador) consulta las estadísticas. |
| 2 | El sistema muestra las estadísticas. |
| 3 | El actor (Administrador) compara el rendimiento de los agentes. |
| 4 | El actor (Administrador) postula cambios. |
| 5 | El sistema confirma las métricas. |
| 6 | La base de datos retorna las métricas. |
| 7 | El sistema muestra las métricas. |
| 8 | El actor (Agente de Servicio) revisa la calificación de desempeño. |
| 9 | El actor (Administrador) propone mejoras al proceso. |
| Importancia | Alta |
| Estado | En Ejecución |
| Estabilidad | Alta |

# **Parte II**

# **Diagrama actividad (igual a la cantidad de dcu)**

1. Gestión de solicitudes de servicio



36-Figura 36. Gestion de solicitudes

El usuario reporta el incidente.

El agente registra el incidente.

Se clasifica y asigna prioridad.

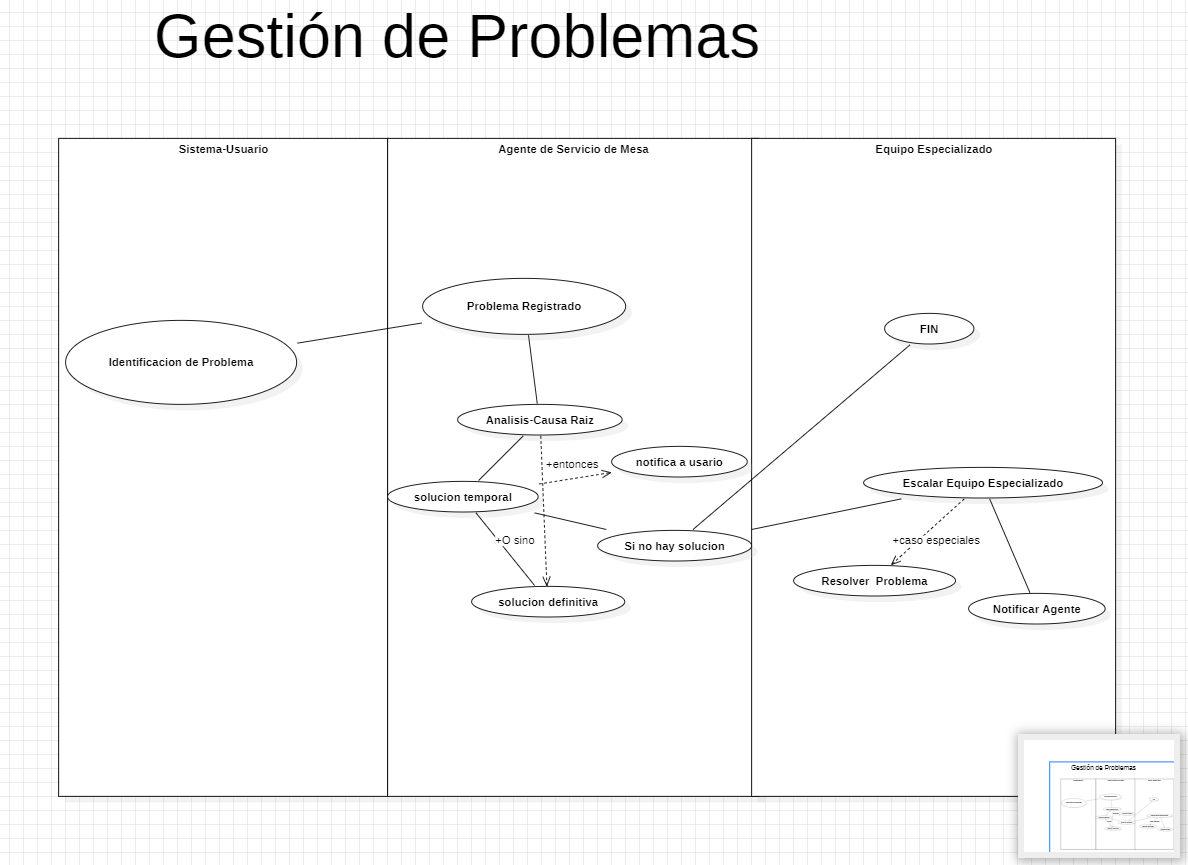
Se investiga y diagnostica.

Se resuelve o se escala a otro equipo.

Se notifica al usuario.

Se cierra el incidente.

1. Gestión de problemas



37-Figura 37: Gestion de problemas

Identificación del problema.

Registro del problema.

Análisis de la causa raíz.

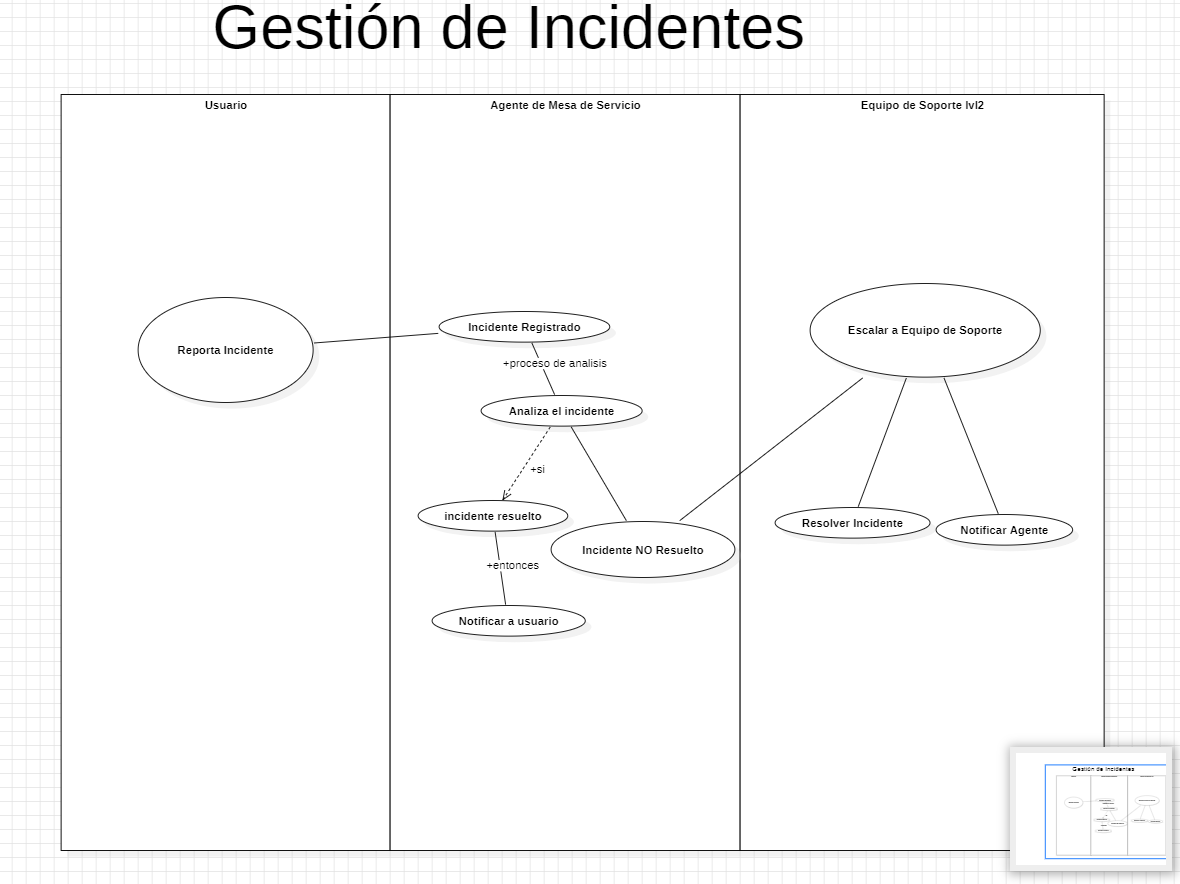
Implementación de solución temporal (si es posible).

Escalación a un equipo especializado (si no se resuelve).

Implementación de solución definitiva.

Cierre del problema.

1. Gestión de incidencias



38-Figura 38: Gestion de incidentes

El usuario solicita el servicio.

Registro y clasificación de la solicitud.

Se realiza una evaluación de la solicitud.

Aprobación o rechazo.

Se implementa el servicio solicitado.

Notificación al usuario.

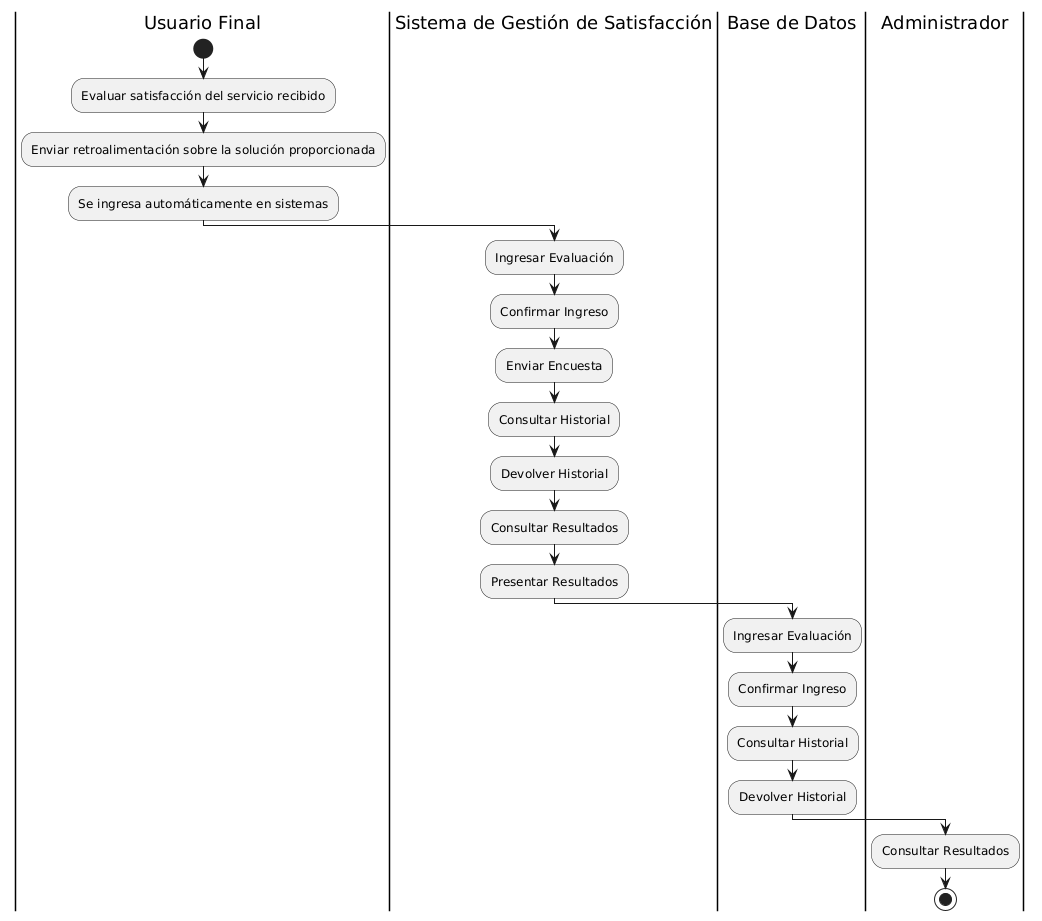
Cierre de la solicitud.

**4. Evaluación de Satisfacción del Usuario**Diagrama

Descripción generada automáticamente

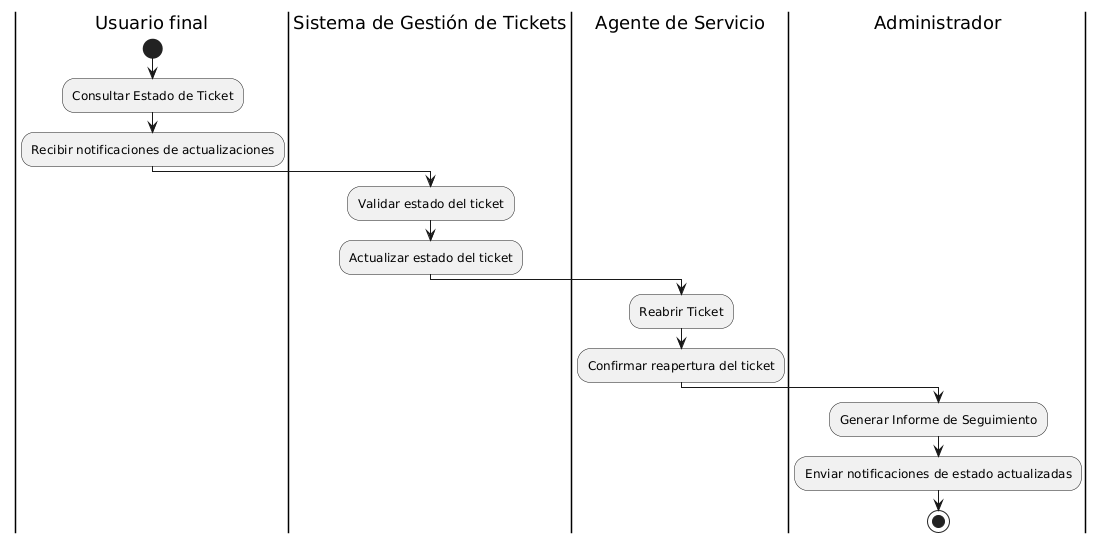
**39**-Figura 39: Evaluación de satisfacción

| **Actor** | **Actividad** |
| --- | --- |
| Usuario Final | Evaluar satisfacción del servicio recibido |
| Usuario Final | Enviar retroalimentación sobre la solución proporcionada |
| Sistema de Gestión de Satisfacción | Ingresar Evaluación |
| Sistema de Gestión de Satisfacción | Confirmar Ingreso |
| Sistema de Gestión de Satisfacción | Enviar Encuesta |
| Sistema de Gestión de Satisfacción | Consultar Historial |
| Sistema de Gestión de Satisfacción | Devolver Historial |
| Sistema de Gestión de Satisfacción | Consultar Resultados |
| Sistema de Gestión de Satisfacción | Presentar Resultados |
| Base de Datos | Ingresar Evaluación |
| Base de Datos | Confirmar Ingreso |
| Base de Datos | Consultar Historial |
| Base de Datos | Devolver Historial |
| Administrador | Consultar Resultados |

**5. Seguimiento de Tickets  
  
**

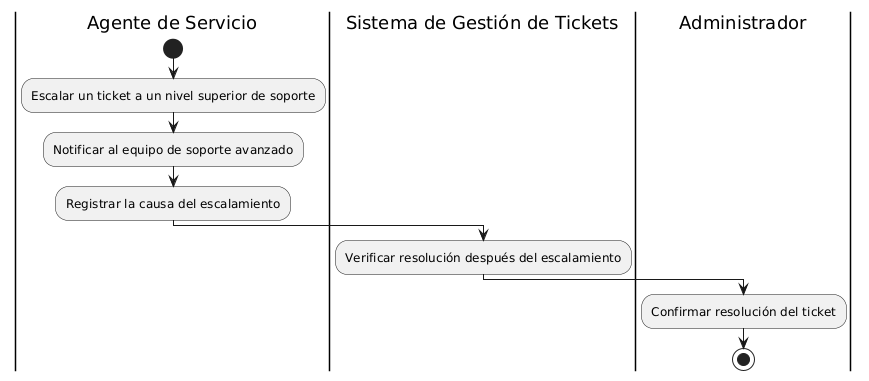
**40**-Figura 40: Seguimiento de Ticket

| **Actor** | **Actividad** |
| --- | --- |
| Usuario Final | Consultar Estado de Ticket |
| Usuario Final | Recibir notificaciones de actualizaciones |
| Sistema de Gestión de Tickets | Validar estado del ticket |
| Sistema de Gestión de Tickets | Actualizar estado del ticket |
| Agente de Servicio | Reabrir Ticket |
| Agente de Servicio | Confirmar reapertura del ticket |
| Administrador | Generar Informe de Seguimiento |
| Administrador | Enviar notificaciones de estado actualizadas |

**6. Escalamiento de Tickets  
  
  
**

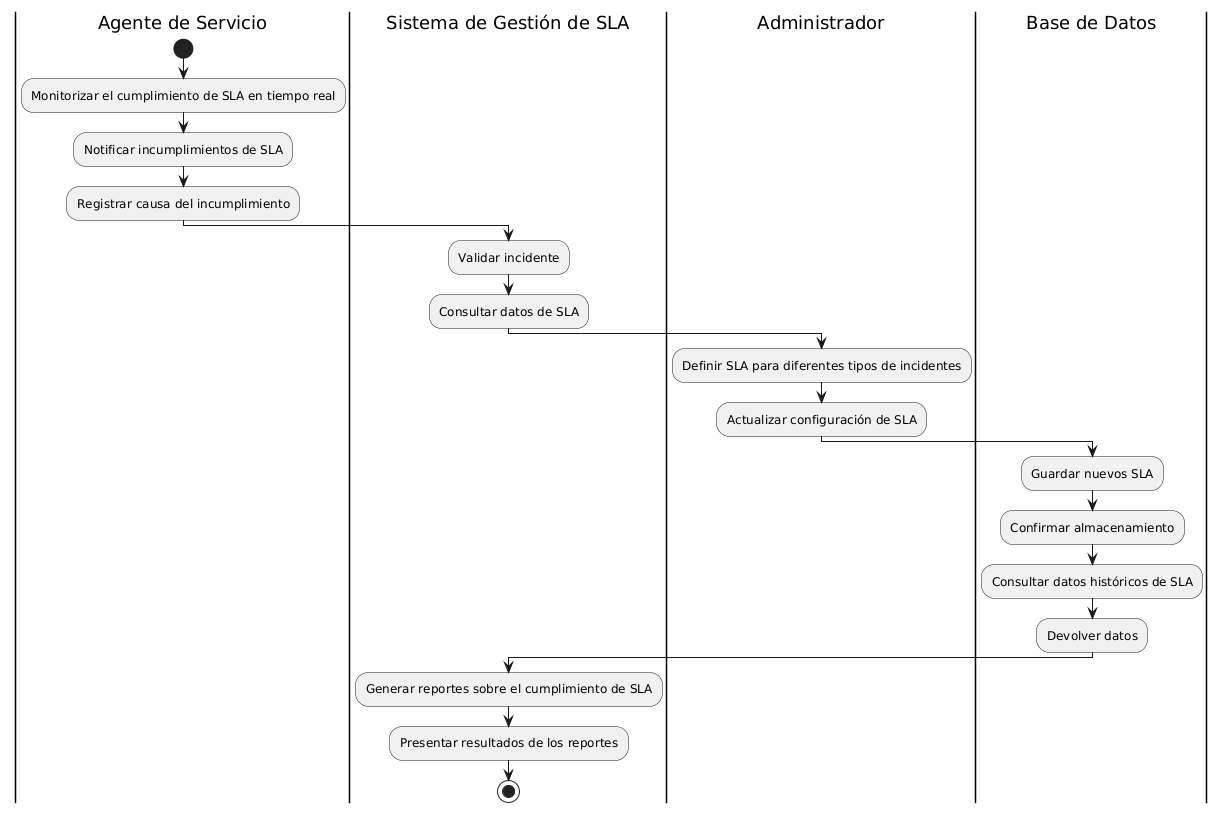
**41**-Figura 41: Escalamiento de Ticket

| **Actor** | **Actividad** |
| --- | --- |
| Agente de Servicio | Escalar un ticket a un nivel superior de soporte |
| Agente de Servicio | Notificar al equipo de soporte avanzado |
| Agente de Servicio | Registrar la causa del escalamiento |
| Sistema de Gestión de Tickets | Verificar resolución después del escalamiento |
| Administrador | Confirmar resolución del ticket |

**7. Monitorización y Gestión de SLA  
  
**

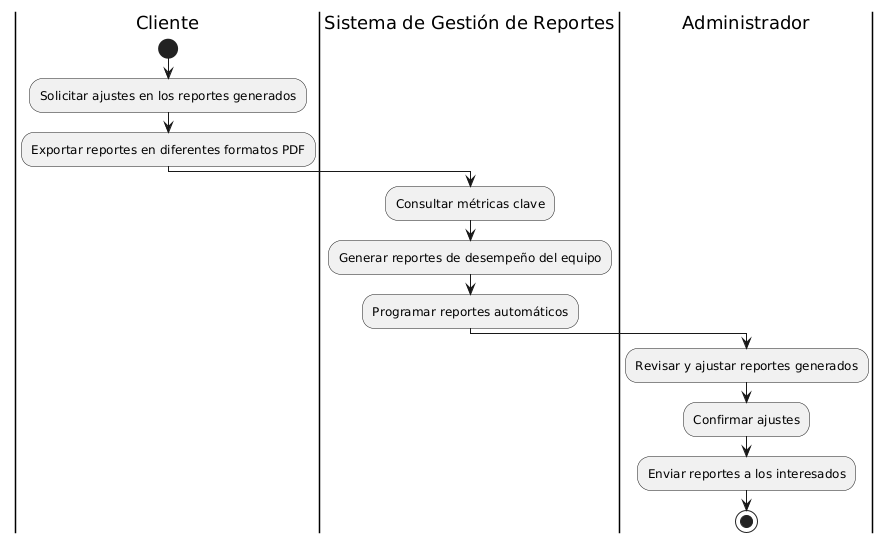
**42**-Figura 42: Monitorización y gestion de SLA

| **Actor** | **Actividad** |
| --- | --- |
| Agente de Servicio | Monitorizar el cumplimiento de SLA en tiempo real |
| Agente de Servicio | Notificar incumplimientos de SLA |
| Agente de Servicio | Registrar causa del incumplimiento |
| Sistema de Gestión de SLA | Validar incidente |
| Sistema de Gestión de SLA | Consultar datos de SLA |
| Administrador | Definir SLA para diferentes tipos de incidentes |
| Administrador | Actualizar configuración de SLA |
| Base de Datos | Guardar nuevos SLA |
| Base de Datos | Confirmar almacenamiento |
| Base de Datos | Consultar datos históricos de SLA |
| Base de Datos | Devolver datos |
| Sistema de Gestión de SLA | Generar reportes sobre el cumplimiento de SLA |
| Sistema de Gestión de SLA | Presentar resultados de los reportes |

**8. Generación de Reportes de Rendimiento  
  
  
**

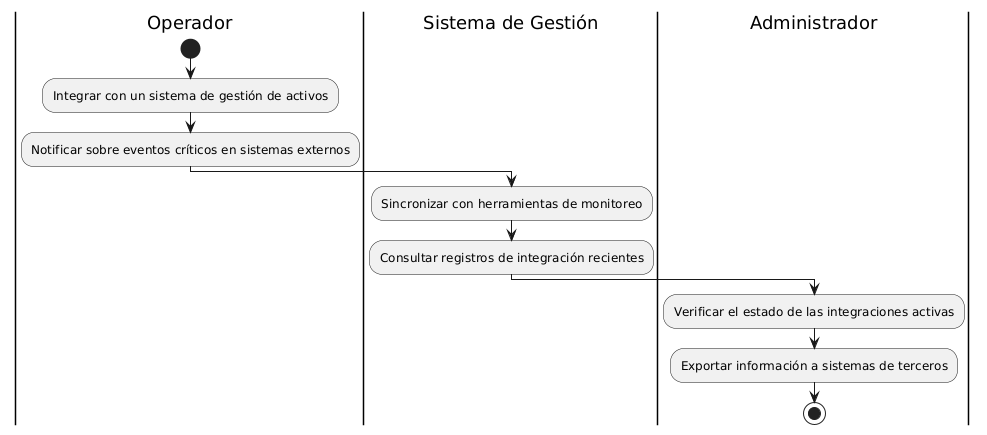
**43**-Figura 43: Generacion de reportes

| **Actor** | **Actividad** |
| --- | --- |
| Cliente | Solicitar ajustes en los reportes generados |
| Cliente | Exportar reportes en diferentes formatos PDF |
| Sistema de Gestión de Reportes | Consultar métricas clave |
| Sistema de Gestión de Reportes | Generar reportes de desempeño del equipo |
| Sistema de Gestión de Reportes | Programar reportes automáticos |
| Administrador | Revisar y ajustar reportes generados |
| Administrador | Confirmar ajustes |
| Administrador | Enviar reportes a los interesados |

**9. Integración con Otros Sistemas  
  
  
**

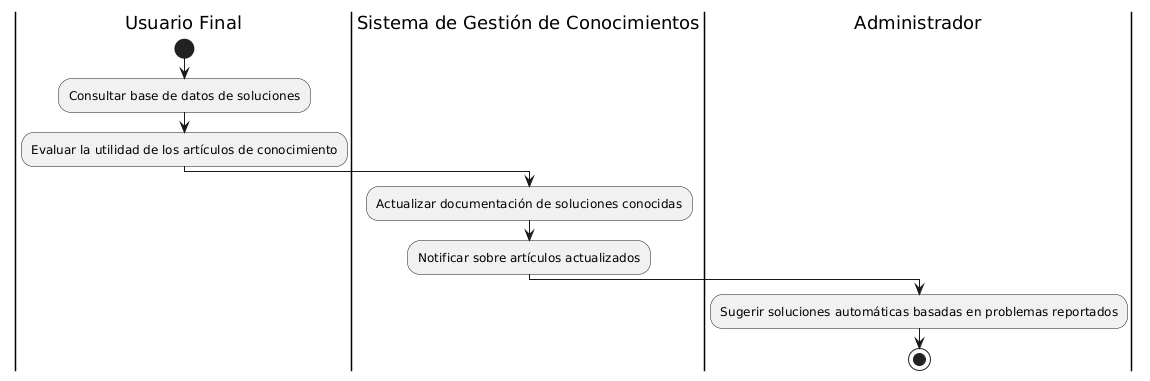
**44**:-Integracion con otros sistemas

| **Actor** | **Actividad** |
| --- | --- |
| Operador | Integrar con un sistema de gestión de activos |
| Operador | Notificar sobre eventos críticos en sistemas externos |
| Sistema de Gestión | Sincronizar con herramientas de monitoreo |
| Sistema de Gestión | Consultar registros de integración recientes |
| Administrador | Verificar el estado de las integraciones activas |
| Administrador | Exportar información a sistemas de terceros |

**10. Gestión de Conocimientos  
**

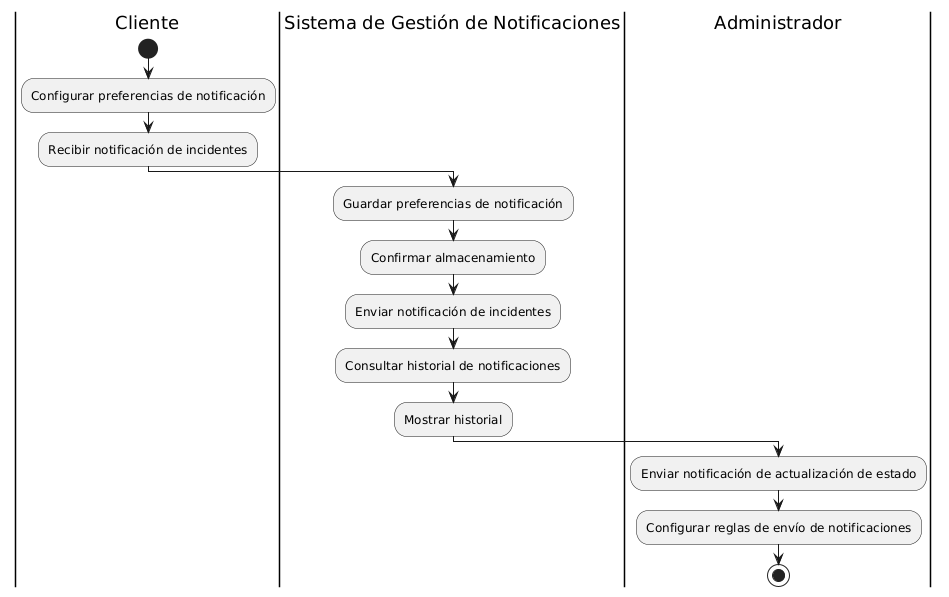
**45**-Figura 45: Gestion de conocimientos

| **Actor** | **Actividad** |
| --- | --- |
| Usuario Final | Consultar base de datos de soluciones |
| Usuario Final | Evaluar la utilidad de los artículos de conocimiento |
| Sistema de Gestión de Conocimientos | Actualizar documentación de soluciones conocidas |
| Sistema de Gestión de Conocimientos | Notificar sobre artículos actualizados |
| Administrador | Sugerir soluciones automáticas basadas en problemas reportados |

**11. Gestión de Notificaciones  
  
**

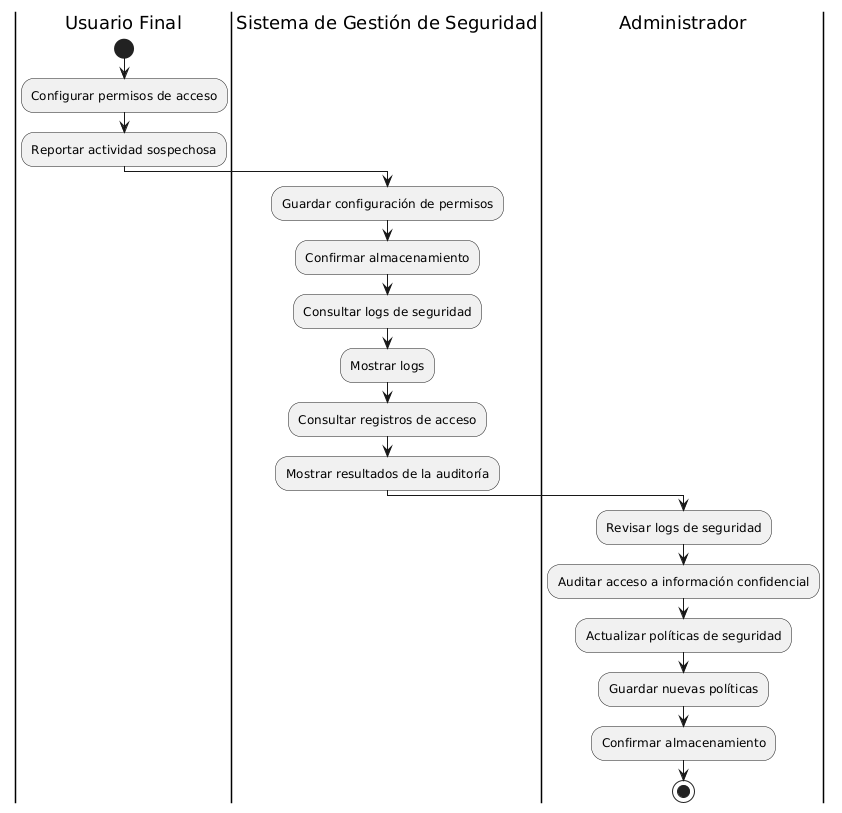
**46**-Figura 46: Gestion de notificación

| **Actor** | **Actividad** |
| --- | --- |
| Cliente | Configurar preferencias de notificación |
| Cliente | Recibir notificación de incidentes |
| Sistema de Gestión de Notificaciones | Guardar preferencias de notificación |
| Sistema de Gestión de Notificaciones | Confirmar almacenamiento |
| Sistema de Gestión de Notificaciones | Enviar notificación de incidentes |
| Sistema de Gestión de Notificaciones | Consultar historial de notificaciones |
| Sistema de Gestión de Notificaciones | Mostrar historial |
| Administrador | Enviar notificación de actualización de estado |
| Administrador | Configurar reglas de envío de notificaciones |

**12. Gestión de Seguridad y Privacidad  
  
**

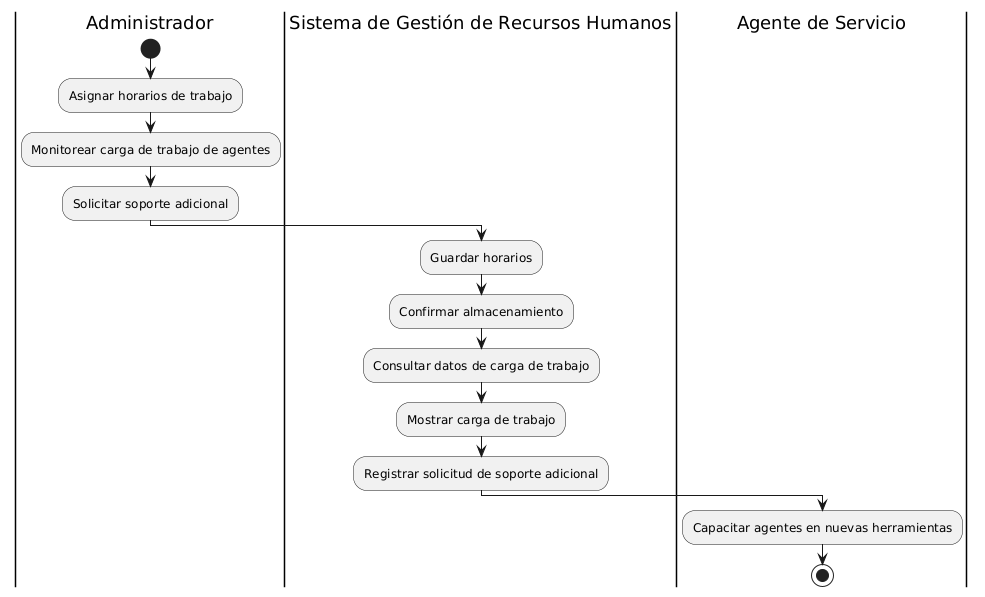
**47**-Figura 47: Gestion de Seguridad y Privacidad

| **Actor** | **Actividad** |
| --- | --- |
| Usuario Final | Configurar permisos de acceso |
| Usuario Final | Reportar actividad sospechosa |
| Sistema de Gestión de Seguridad | Guardar configuración de permisos |
| Sistema de Gestión de Seguridad | Confirmar almacenamiento |
| Sistema de Gestión de Seguridad | Consultar logs de seguridad |
| Sistema de Gestión de Seguridad | Mostrar logs |
| Sistema de Gestión de Seguridad | Consultar registros de acceso |
| Sistema de Gestión de Seguridad | Mostrar resultados de la auditoría |
| Administrador | Revisar logs de seguridad |
| Administrador | Auditar acceso a información confidencial |
| Administrador | Actualizar políticas de seguridad |
| Administrador | Guardar nuevas políticas |
| Administrador | Confirmar almacenamiento |

**13. Gestión de Recursos Humanos para la Mesa de Servicio  
  
  
**

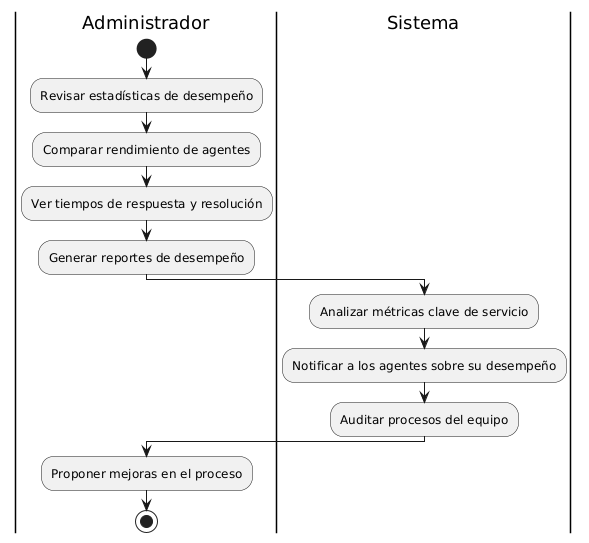
**48**-Figura 48: Recursos Humanos para la mesa de servicio

| **Actor** | **Actividad** |
| --- | --- |
| Administrador | Asignar horarios de trabajo |
| Administrador | Monitorear carga de trabajo de agentes |
| Administrador | Solicitar soporte adicional |
| Sistema de Gestión de Recursos Humanos | Guardar horarios |
| Sistema de Gestión de Recursos Humanos | Confirmar almacenamiento |
| Sistema de Gestión de Recursos Humanos | Consultar datos de carga de trabajo |
| Sistema de Gestión de Recursos Humanos | Mostrar carga de trabajo |
| Sistema de Gestión de Recursos Humanos | Registrar solicitud de soporte adicional |
| Agente de Servicio | Capacitar agentes en nuevas herramientas |

**14. Análisis de Desempeño del Equipo  
  
**

**49**-Figura 49: Analisis de desempeño del equipo

| **Actor** | **Actividad** |
| --- | --- |
| Administrador | Revisar estadísticas de desempeño |
| Administrador | Comparar rendimiento de agentes |
| Administrador | Ver tiempos de respuesta y resolución |
| Administrador | Generar reportes de desempeño |
| Sistema | Analizar métricas clave de servicio |
| Sistema | Notificar a los agentes sobre su desempeño |
| Sistema | Auditar procesos del equipo |
| Administrador | Proponer mejoras en el proceso |

**15. Gestión de Incidentes  
**

**50**-Figura 50: Gestión de incidentes

| **Actor** | **Actividad** |
| --- | --- |
| Usuario | Reportar incidente |
| Usuario | Recibir confirmación de reporte |
| Sistema de Gestión de Incidentes | Registrar incidente |
| Sistema de Gestión de Incidentes | Asignar prioridad |
| Sistema de Gestión de Incidentes | Notificar al equipo de soporte |
| Equipo de Soporte | Investigar incidente |
| Equipo de Soporte | Proponer solución |
| Equipo de Soporte | Implementar solución |
| Sistema de Gestión de Incidentes | Actualizar estado del incidente |
| Sistema de Gestión de Incidentes | Notificar al usuario |
| Usuario | Confirmar resolución |

**Diagrama de colaboración (igual a la cantidad de secuencia)**Escala de tiempo

Descripción generada automáticamente

**51** Figura 51: Diagrama de colaboración

Diagrama, Esquemático

Descripción generada automáticamente

**52**-Figura 52: Colaboracion Gestion de tickets

Diagrama

Descripción generada automáticamente

**53**-Figura 53: Sistema de gestiones

Gráfico

Descripción generada automáticamente

54- Figura 54: Manejo de prioridades en el sistema de ticket

Imagen que contiene Diagrama

Descripción generada automáticamente

55-Figura 55: Evaluacion de satisfaccion del usuario

Diagrama de colaboración - Evaluación de Satisfacción del Usuario  
Diagrama

Descripción generada automáticamente

56-Figura 56: manejo de Solicitud

Diagrama de Colaboración - Manejo de Solicitudes  
Un mapa de color blanco

Descripción generada automáticamente con confianza baja

57-Figura 57: Manejo de Solicitudes

Diagrama

Descripción generada automáticamente  
  
Diagrama

Descripción generada automáticamente

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

Diagrama, Esquemático

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente con confianza media

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente  
Diagrama, Esquemático

Descripción generada automáticamente

# **Diagrama de componente (modulo o submodulos o subsitemas o componente interno o externo)**

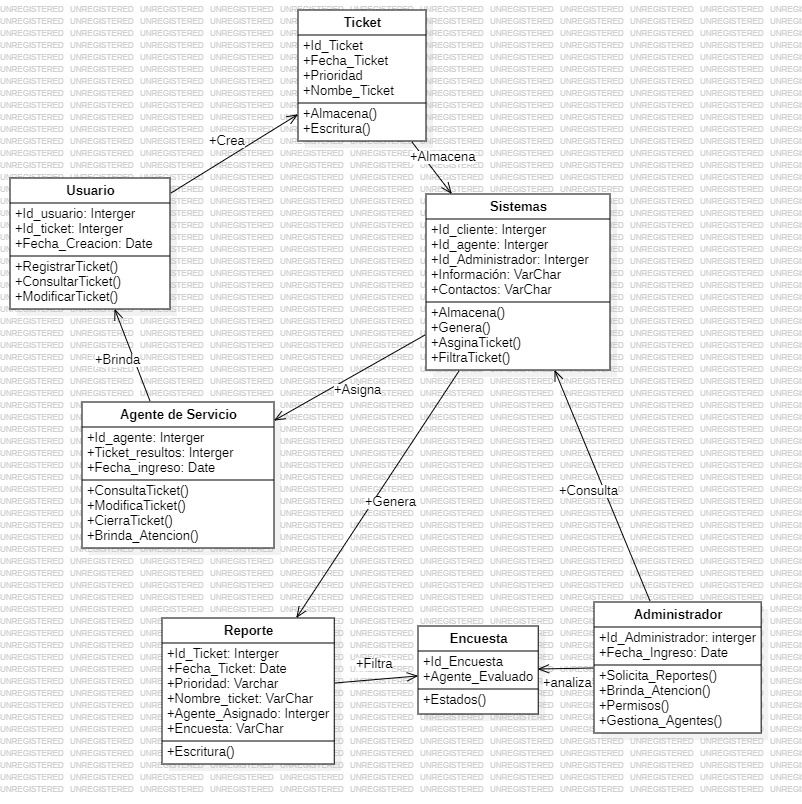
# 

58-Figura 58: Diagrama de Componente

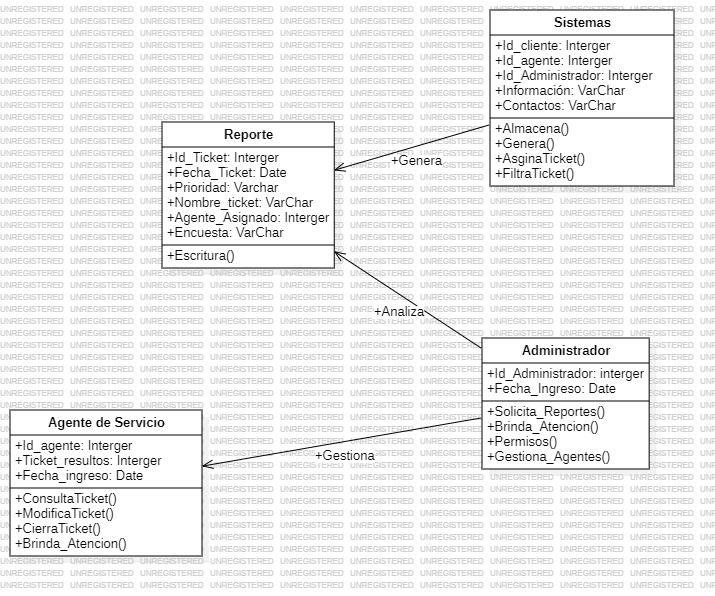
# El diagrama de componente muestra la estructura y las interacciones entre los diferentes elementos de un sistema de gestión de mesa de servicio. Los componentes principales incluyen Usuario, Sistema, Agente de servicio, Administrador, Base de Datos, Base de Datos Racional, Base de datos RDF, Ticket, Escritura, Filtra ticket, Asigna ticket, Almacena, Reporte, Encuesta, Estados, y Solicita Reportes. El Usuario puede registrar, consultar y modificar tickets, interactuando con el Agente de servicio. El Sistema se encarga de asignar y generar tickets, almacenando datos en la Base de Datos, que a su vez se conecta con las bases de datos Racional y RDF. El Administrador analiza encuestas y solicita reportes, mientras que el Reporte filtra información y la Encuesta proporciona datos sobre el estado del sistema. Este diagrama es crucial para comprender cómo los distintos componentes colaboran y se relacionan para gestionar eficazmente los tickets y la información en un sistema de mesa de servicio.

# **Diagrama clases () y sus descripciones**

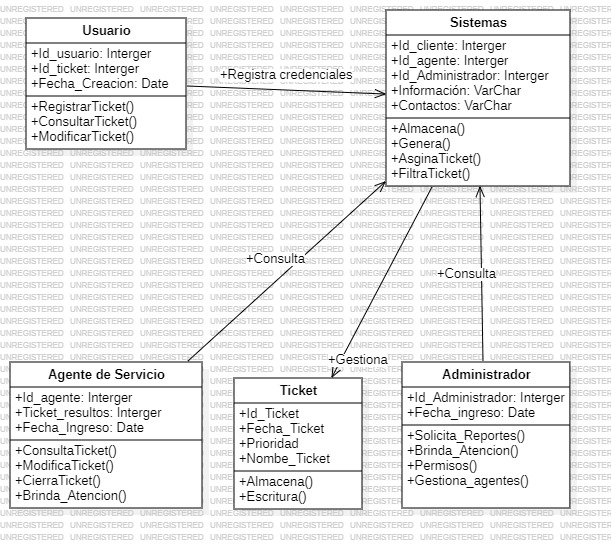
Gestion de encuestas



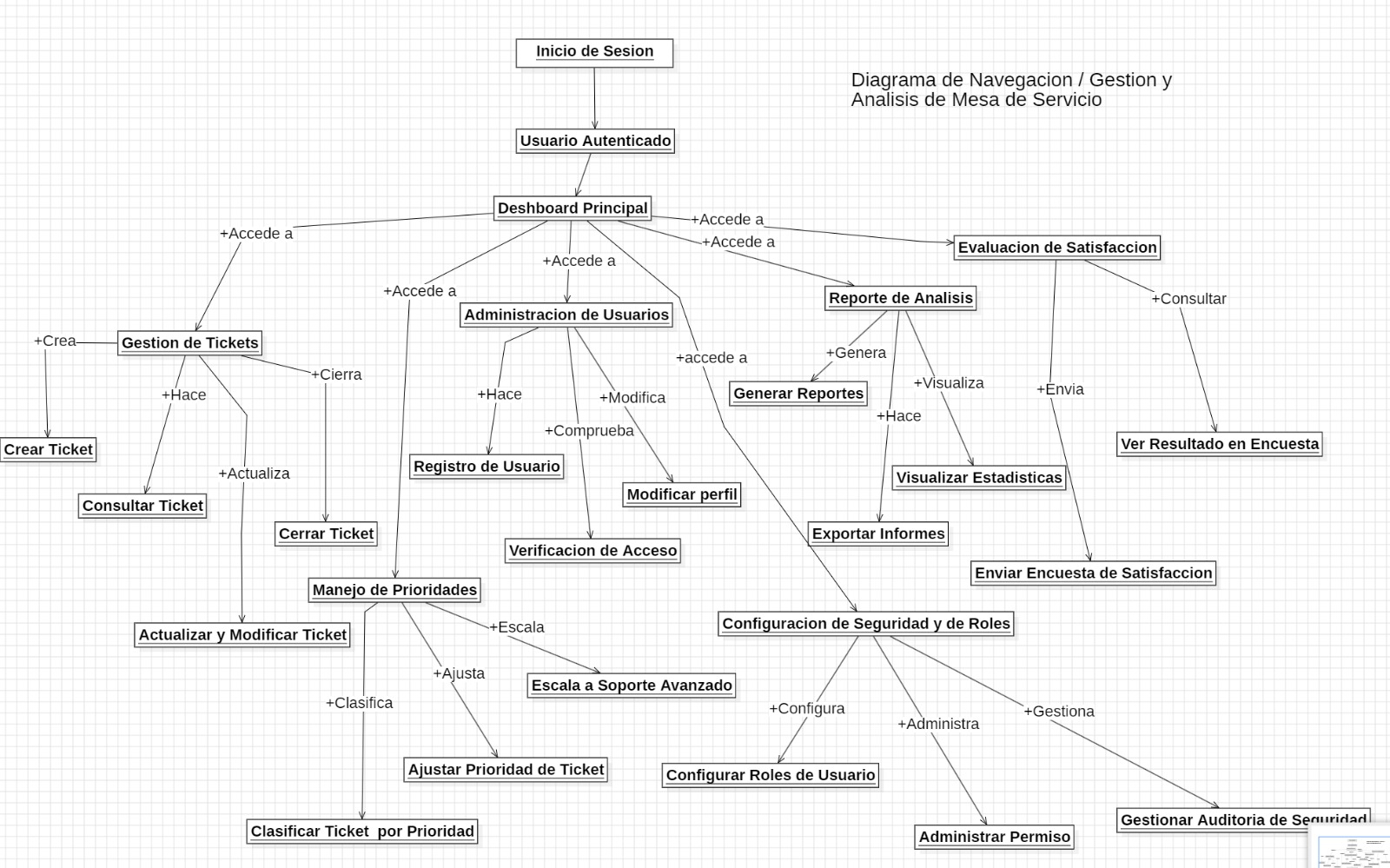
Gestión de tiempo de respuesta



Gestión de seguridad y accesos



# **Diagrama de navegación (como se mueve los usuarios por la diferente interface e/s)**



Descripción:

1. **Inicio de Sesión**:
   * Punto de entrada principal del sistema. Desde aquí, el usuario accede al sistema tras autenticarse.
2. **Usuario Autenticado**:
   * Una vez autenticado, el usuario accede al **Dashboard Principal**, que actúa como un hub para las distintas funcionalidades.
3. **Dashboard Principal**:
   * Permite acceder a diferentes áreas clave del sistema:
     + **Gestión de Tickets**: Incluye las acciones de:
       - Crear, consultar, cerrar, actualizar o modificar un ticket.
       - Manejo de prioridades, incluyendo:
         * Clasificar tickets por prioridad.
         * Ajustar prioridad.
         * Escalar tickets a soporte avanzado.
     + **Administración de Usuarios**:
       - Registro de nuevos usuarios.
       - Modificación del perfil del usuario.
       - Verificación de accesos.
     + **Configuración de Seguridad y Roles**:
       - Configuración de roles de usuario.
       - Administración de permisos.
       - Gestión de auditoría de seguridad.
     + **Evaluación de Satisfacción**:
       - Enviar encuestas de satisfacción.
       - Consultar resultados de encuestas.
     + **Reporte de Análisis**:
       - Generar reportes.
       - Visualizar estadísticas.
       - Exportar informes.
4. **Evaluación de Satisfacción**:
   * Centrada en la retroalimentación del usuario, permitiendo generar y enviar encuestas, y analizar sus resultados.
5. **Reporte de Análisis**:
   * Herramienta para la generación de reportes, visualización de estadísticas e informes exportables.
6. **Configuración de Seguridad y Roles**:
   * Proporciona herramientas para ajustar los permisos y roles del sistema, así como gestionar auditorías.

# **Diagrama de entidad relación (son las relaciones de todas la entidades de la base de datos)**

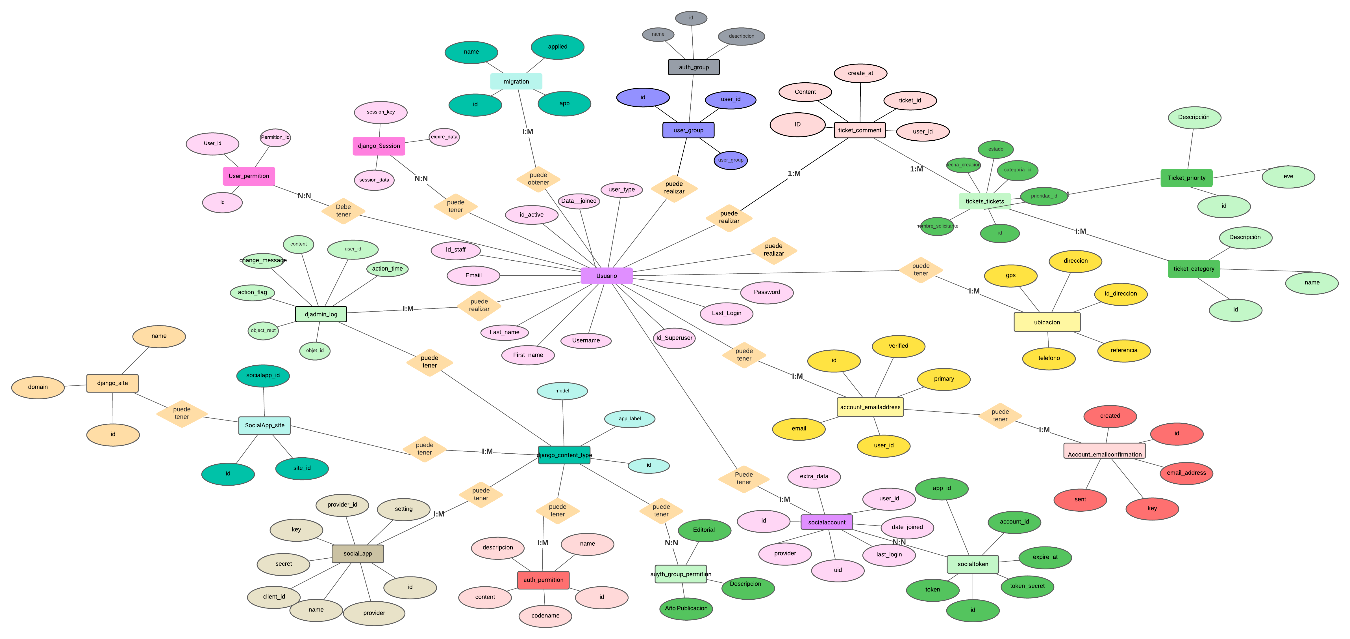


Figura 59- Diagrama entidad Relación

[Enlace](https://lucid.app/lucidspark/9661c28f-27f7-49bc-a23c-a06835fe0210/edit?viewport_loc=-2136%2C-1022%2C4854%2C2280%2C0_0&invitationId=inv_978dd243-898e-4acf-91c8-baaeb8ae1f5c)

# **Diagrama de estructura de datos (data model con sus relaciones)**

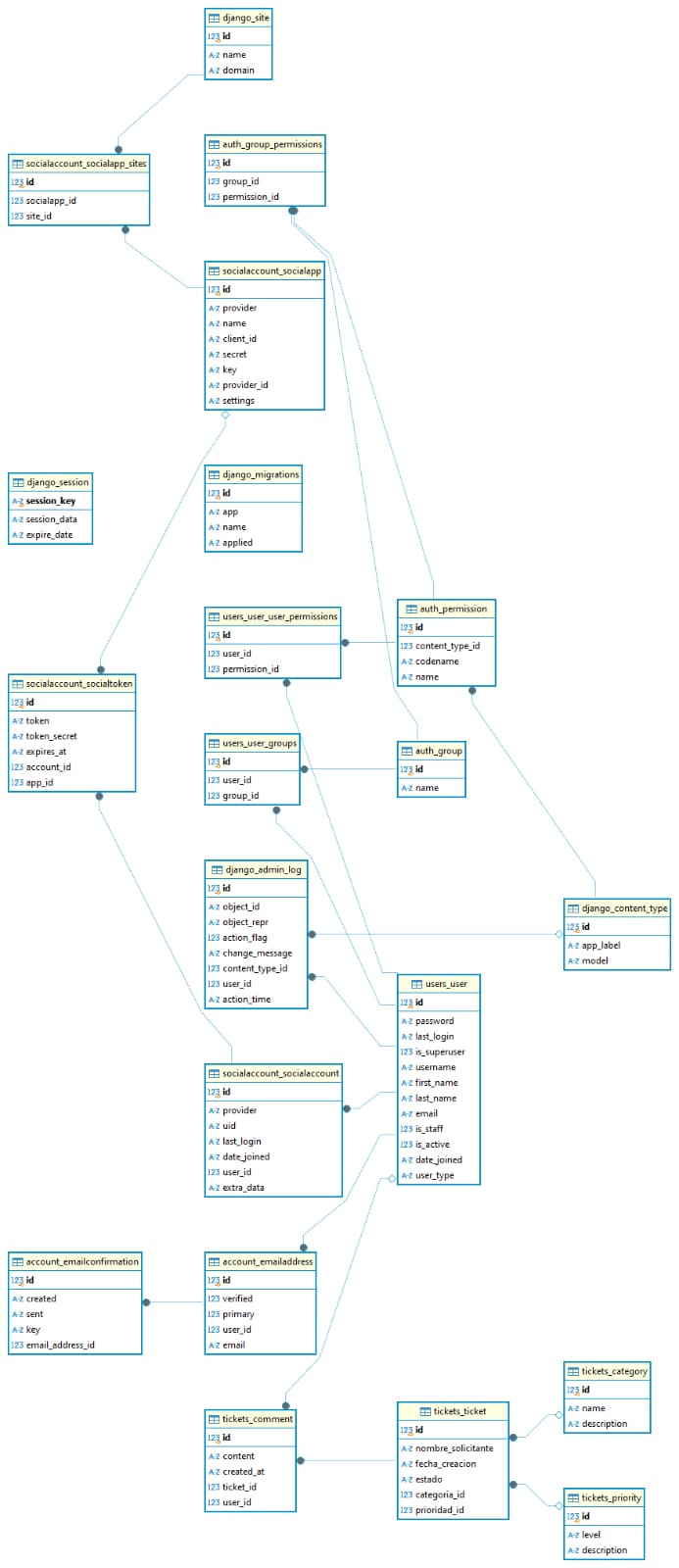


Figura 60 - Data model

# **Diagrama de arquitectura de la aplicación (es el diagrama de donde reside el sistema de información diseñado)**

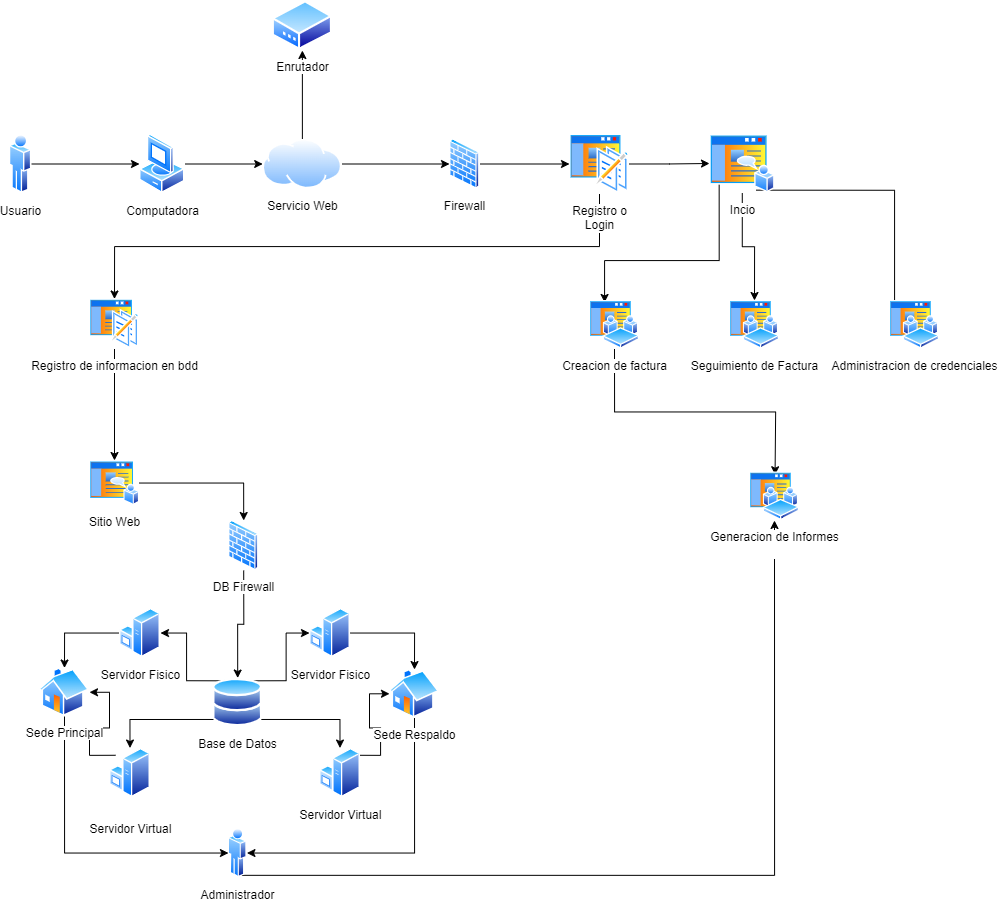


Figura 61: Diagrama de navegación

Descripción:

1. **Inicio con el Usuario**:
   * El **usuario** interactúa con el sistema a través de una **computadora**, la cual se conecta al **servicio web**.
2. **Paso por el Enrutador y Firewall**:
   * La solicitud pasa por un **enrutador** y un **firewall**, asegurando la protección y control de tráfico antes de llegar al sistema.
3. **Acceso al Sistema**:
   * El usuario accede al sistema a través de una página de **registro o login**.
   * Desde aquí, el sistema ofrece varias funcionalidades clave:
     + **Creación de factura**: Permite generar nuevas facturas.
     + **Seguimiento de factura**: Para monitorear el estado y progreso de las facturas existentes.
     + **Administración de credenciales**: Gestión de usuarios y permisos.
4. **Registro de Información en la Base de Datos**:
   * Las acciones realizadas son registradas en una **base de datos** (bdd) a través de un sistema protegido por un **DB Firewall**.
5. **Sitio Web**:
   * La información registrada se puede visualizar o interactuar a través de un **sitio web**, conectado también a la base de datos.
6. **Infraestructura de la Base de Datos**:
   * La base de datos está respaldada por una infraestructura distribuida que incluye:
     + **Servidores físicos** y **servidores virtuales**:
       - En una **sede principal** para las operaciones centrales.
       - En una **sede de respaldo** para garantizar alta disponibilidad y recuperación ante fallos.
7. **Generación de Informes**:
   * El sistema permite generar informes, recopilando y presentando datos procesados para análisis y toma de decisiones.
8. **Administrador**:
   * Un **administrador** tiene acceso directo al sistema para gestionar y supervisar las operaciones.

# **Diseño de las interfaces usando una herramientas ui o ux (de todo el proyecto)**

[Enlace](https://www.figma.com/design/GSrsWR2YbndZ5uvyvIBWxs/SOLUTICKET?node-id=5-33&node-type=canvas&t=py5MhuZYKLmS8PQA-0)

****

-Figura: Figma

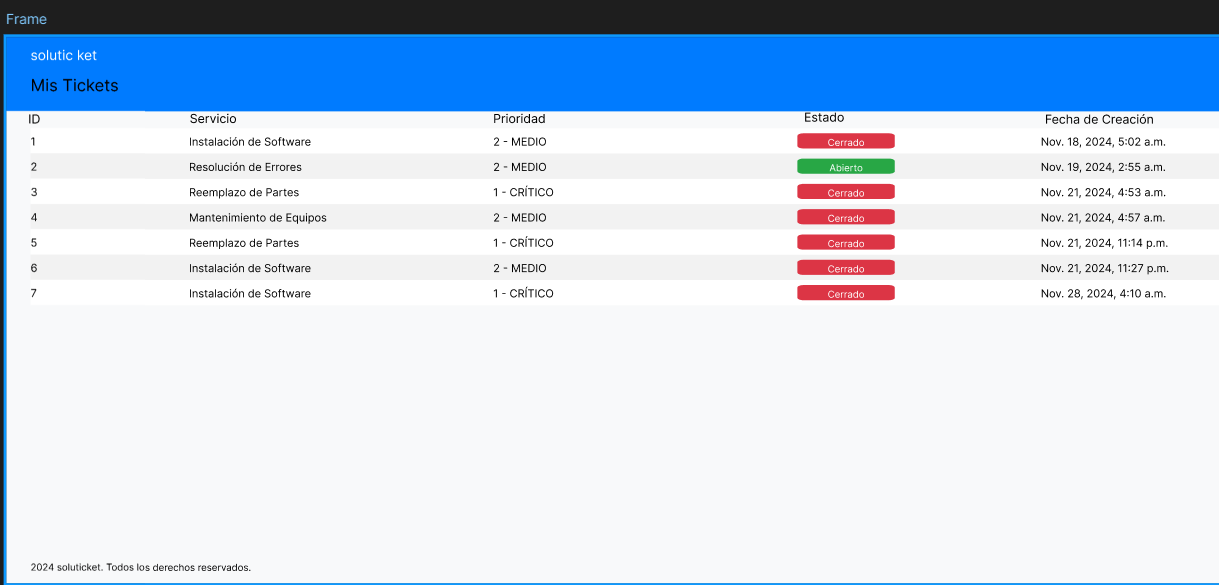
****

Figura: Figma

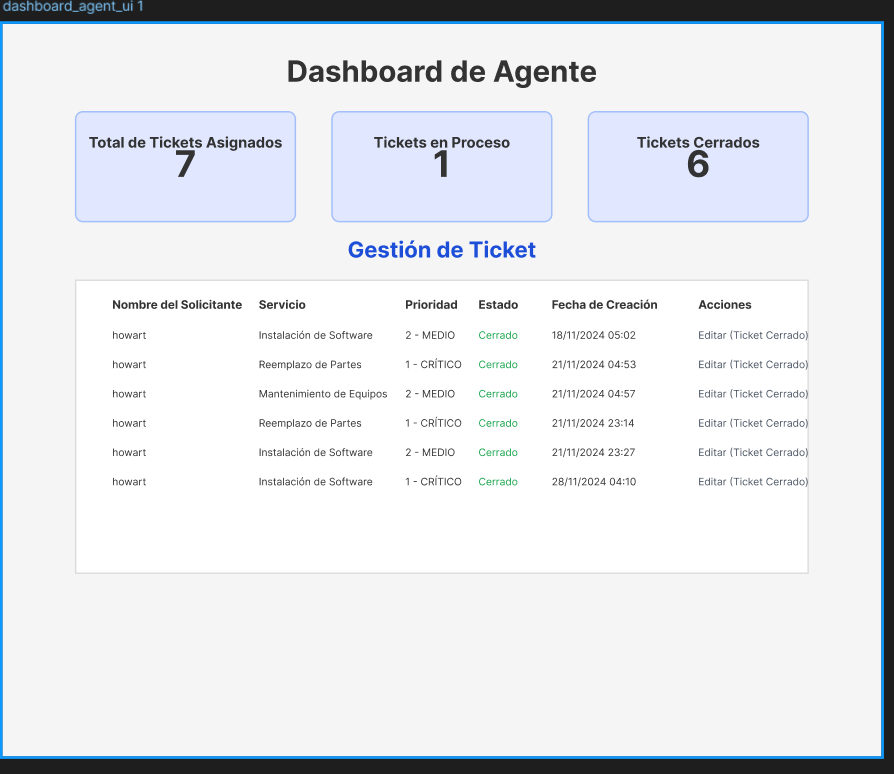
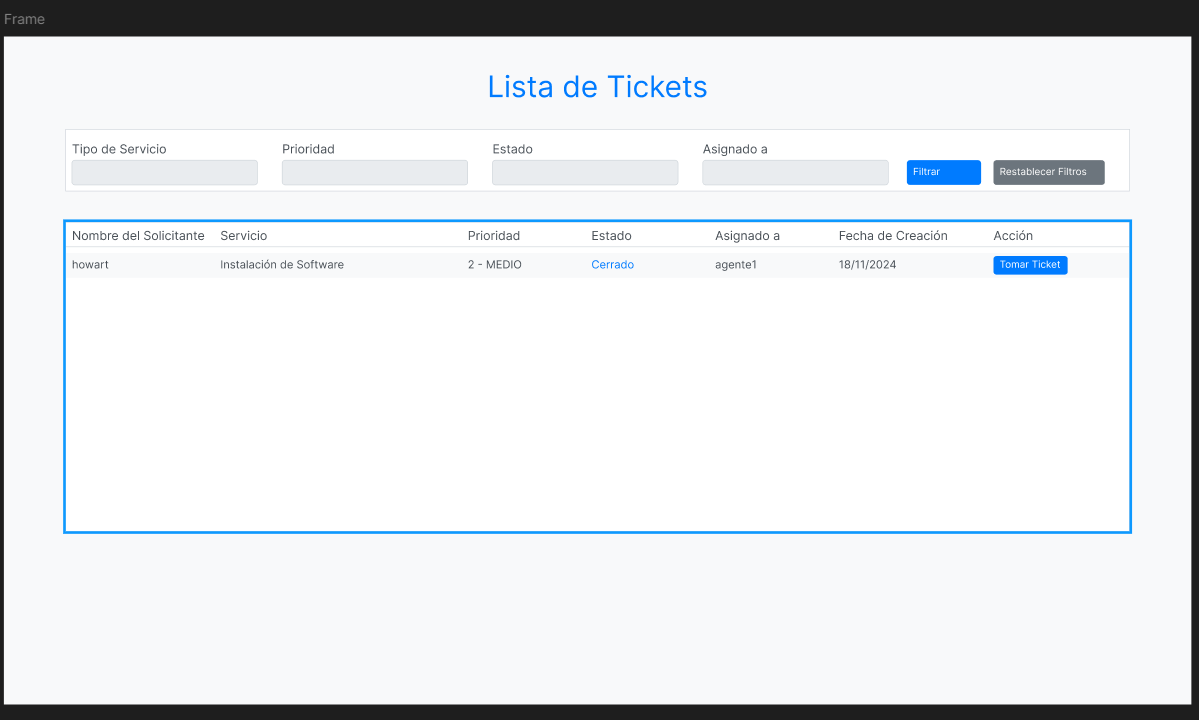
****

Figura: Figma

****

59-Figura: Figma

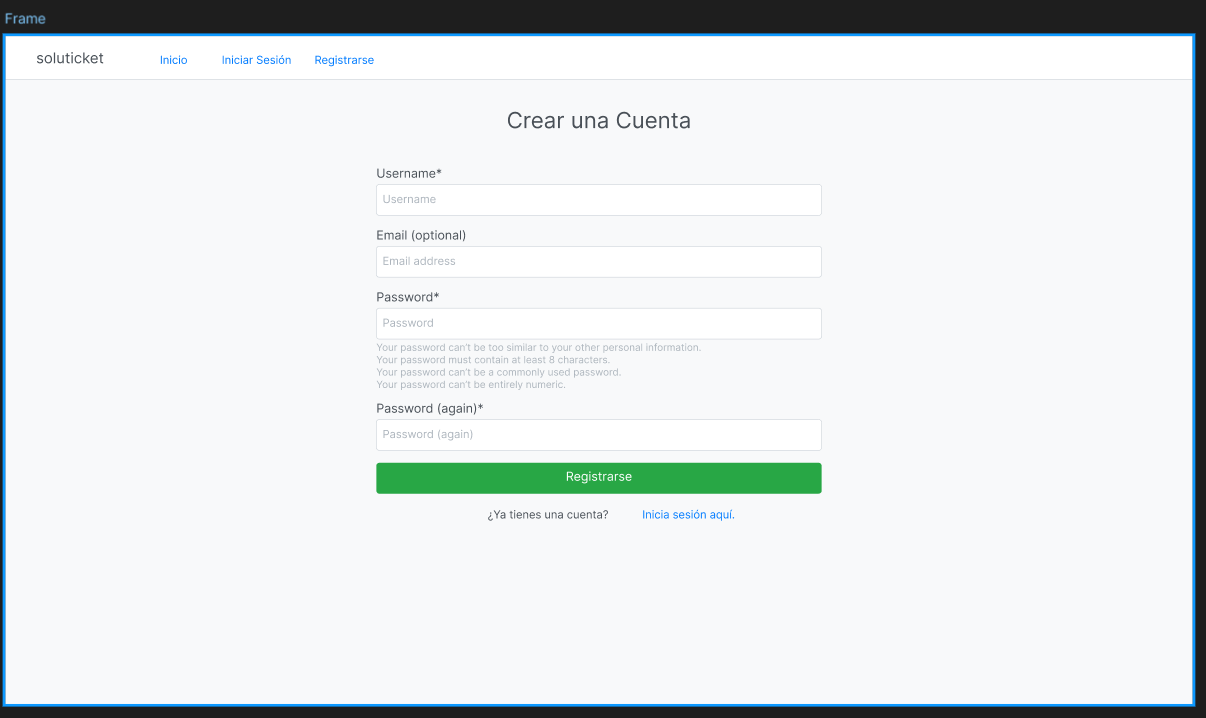
****

Figura: Figma

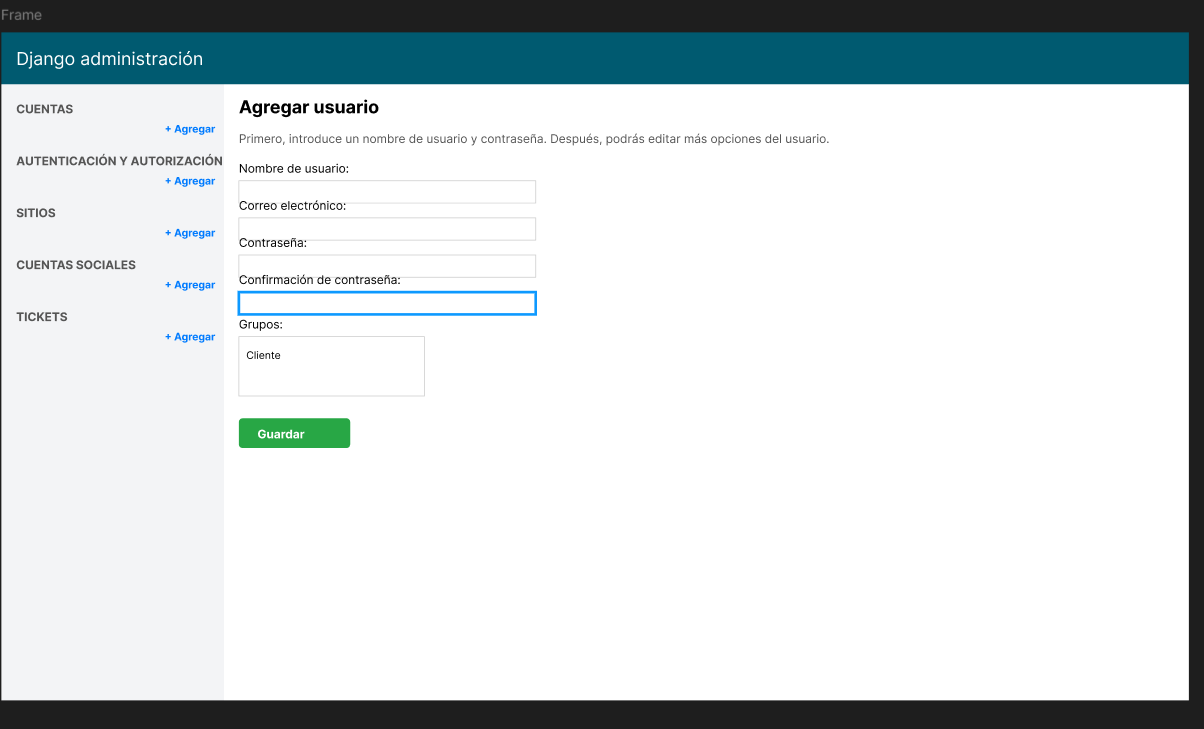
****

Figura: Figma

# **Parte III**

# **Linea grafica del proyecto – logo tipo de sus software**



-Figura: Logotipo

1. **Identidad Visual Coherente**: La identidad visual de *Soluticket* se refleja en su logotipo simple, profesional y moderno. Utiliza un diseño minimalista con elementos gráficos que transmiten claridad y funcionalidad. La fuente es limpia, lo que ayuda a generar confianza y profesionalismo.
2. **Colores y Estilo**: El fondo azul pastel evoca tranquilidad, tecnología y modernidad, lo que es ideal para una solución digital. Este color puede asociarse con accesibilidad y comodidad, características esenciales para un software de gestión como parece ser *Soluticket*.
3. **Iconografía y Elementos Gráficos**: La silueta de una pantalla y un mouse, ambos minimalistas, representan claramente la funcionalidad tecnológica del software. Este elemento gráfico simplificado comunica que *Soluticket* está relacionado con el soporte técnico o herramientas digitales.
4. **Aplicación Práctica**: El diseño puede adaptarse fácilmente a distintos formatos: aplicaciones móviles, páginas web, material publicitario, tarjetas de presentación, etc. Su simplicidad facilita la integración en diferentes plataformas y medios.
5. **Adaptabilidad**: El logotipo y los elementos gráficos son suficientemente versátiles para ser utilizados en múltiples contextos, desde redes sociales hasta software o documentos impresos. La elección de colores pastel también asegura que no compita visualmente con otros elementos de diseño.
6. **Personalización del Software**: Aunque no hay detalles del software, el diseño minimalista sugiere que *Soluticket* puede enfocarse en la usabilidad y la experiencia del usuario, algo crucial en una solución tecnológica. Esto se puede reforzar con opciones de personalización que permitan a los clientes adaptar la plataforma a sus necesidades.
7. **Consistencia en la Documentación**: El diseño profesional y unificado del logotipo puede trasladarse fácilmente a la documentación del software, garantizando coherencia visual. Esto incluye manuales, informes y guías, lo que facilita el reconocimiento de la marca y genera confianza en el producto.

# **Creación de su base de datos y todos sus DDL y un backup de la base de datos (La base de dato que usted o su equipo seleccionó para implementar sus proyecto)**

CREATE TABLE sqlite\_sequence (

name TEXT,

seq TEXT

);

CREATE TABLE django\_migrations (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

app VARCHAR(255) NOT NULL,

name VARCHAR(255) NOT NULL,

applied DATETIME NOT NULL

);

CREATE TABLE django\_content\_type (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

app\_label VARCHAR(100) NOT NULL,

model VARCHAR(100) NOT NULL

);

CREATE TABLE auth\_group\_permissions (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

group\_id INTEGER NOT NULL REFERENCES auth\_group (id) DEFERRABLE INITIALLY DEFERRED,

permission\_id INTEGER NOT NULL REFERENCES auth\_permission (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE auth\_permission (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

content\_type\_id INTEGER NOT NULL REFERENCES django\_content\_type (id) DEFERRABLE INITIALLY DEFERRED,

codename VARCHAR(100) NOT NULL,

name VARCHAR(255) NOT NULL

);

CREATE TABLE auth\_group (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

name VARCHAR(150) NOT NULL UNIQUE

);

CREATE TABLE users\_user (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

password VARCHAR(128) NOT NULL,

last\_login DATETIME NULL,

is\_superuser BOOL NOT NULL,

username VARCHAR(150) NOT NULL UNIQUE,

first\_name VARCHAR(150) NOT NULL,

last\_name VARCHAR(150) NOT NULL,

email VARCHAR(254) NOT NULL,

is\_staff BOOL NOT NULL,

is\_active BOOL NOT NULL,

date\_joined DATETIME NOT NULL

);

CREATE TABLE users\_user\_groups (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

user\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

group\_id INTEGER NOT NULL REFERENCES auth\_group (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE users\_user\_user\_permissions (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

user\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

permission\_id INTEGER NOT NULL REFERENCES auth\_permission (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE account\_emailconfirmation (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

created DATETIME NOT NULL,

sent DATETIME NULL,

key VARCHAR(64) NOT NULL UNIQUE,

email\_address\_id INTEGER NOT NULL REFERENCES account\_emailaddress (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE account\_emailaddress (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

verified BOOL NOT NULL,

primary BOOL NOT NULL,

user\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

email VARCHAR(254) NOT NULL

);

CREATE TABLE django\_admin\_log (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

object\_id TEXT NULL,

object\_repr VARCHAR(200) NOT NULL,

action\_flag SMALLINT UNSIGNED NOT NULL CHECK (action\_flag >= 0),

change\_message TEXT NOT NULL,

content\_type\_id INTEGER NULL REFERENCES django\_content\_type (id) DEFERRABLE INITIALLY DEFERRED,

user\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

action\_time DATETIME NOT NULL

);

CREATE TABLE django\_session (

session\_key VARCHAR(40) NOT NULL PRIMARY KEY,

session\_data TEXT NOT NULL,

expire\_date DATETIME NOT NULL

);

CREATE TABLE django\_site (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

name VARCHAR(50) NOT NULL,

domain VARCHAR(100) NOT NULL UNIQUE

);

CREATE TABLE socialaccount\_socialapp\_sites (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

socialapp\_id INTEGER NOT NULL REFERENCES socialaccount\_socialapp (id) DEFERRABLE INITIALLY DEFERRED,

site\_id INTEGER NOT NULL REFERENCES django\_site (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE socialaccount\_socialapp (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

provider VARCHAR(30) NOT NULL,

name VARCHAR(40) NOT NULL,

client\_id VARCHAR(191) NOT NULL,

secret VARCHAR(191) NOT NULL,

key VARCHAR(191) NOT NULL,

provider\_id VARCHAR(200) NOT NULL,

settings TEXT NOT NULL CHECK (JSON\_VALID(settings) OR settings IS NULL)

);

CREATE TABLE socialaccount\_socialtoken (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

token TEXT NOT NULL,

token\_secret TEXT NOT NULL,

expires\_at DATETIME NULL,

account\_id INTEGER NOT NULL REFERENCES socialaccount\_socialaccount (id) DEFERRABLE INITIALLY DEFERRED,

app\_id INTEGER NULL REFERENCES socialaccount\_socialapp (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE socialaccount\_socialaccount (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

provider VARCHAR(200) NOT NULL,

uid VARCHAR(191) NOT NULL,

last\_login DATETIME NOT NULL,

date\_joined DATETIME NOT NULL,

user\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

extra\_data TEXT NOT NULL CHECK (JSON\_VALID(extra\_data) OR extra\_data IS NULL)

);

CREATE TABLE tickets\_category (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

name VARCHAR(100) NOT NULL,

description TEXT NULL

);

CREATE TABLE tickets\_priority (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

level VARCHAR(50) NOT NULL,

description TEXT NULL

);

CREATE TABLE tickets\_comment (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

content TEXT NOT NULL,

created\_at DATETIME NOT NULL,

user\_id BIGINT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

ticket\_id BIGINT NOT NULL REFERENCES tickets\_ticket (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE tickets\_type (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

name VARCHAR(100) NOT NULL,

description TEXT NULL,

category\_id BIGINT NOT NULL REFERENCES tickets\_category (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE tickets\_ticket (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

estado VARCHAR(10) NOT NULL,

fecha\_creacion DATETIME NOT NULL,

prioridad\_id BIGINT NULL REFERENCES tickets\_priority (id) DEFERRABLE INITIALLY DEFERRED,

servicio\_id BIGINT NULL REFERENCES tickets\_type (id) DEFERRABLE INITIALLY DEFERRED,

asignado\_a\_id BIGINT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED,

descripcion TEXT NULL,

nombre\_solicitante\_id BIGINT NOT NULL REFERENCES users\_user (id) DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE tickets\_feedback (

id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,

satisfaccion INTEGER NOT NULL,

comentarios TEXT NULL,

fecha\_creacion DATETIME NOT NULL,

ticket\_id BIGINT NOT NULL REFERENCES tickets\_ticket (id) DEFERRABLE INITIALLY DEFERRED

);

**Back up:**

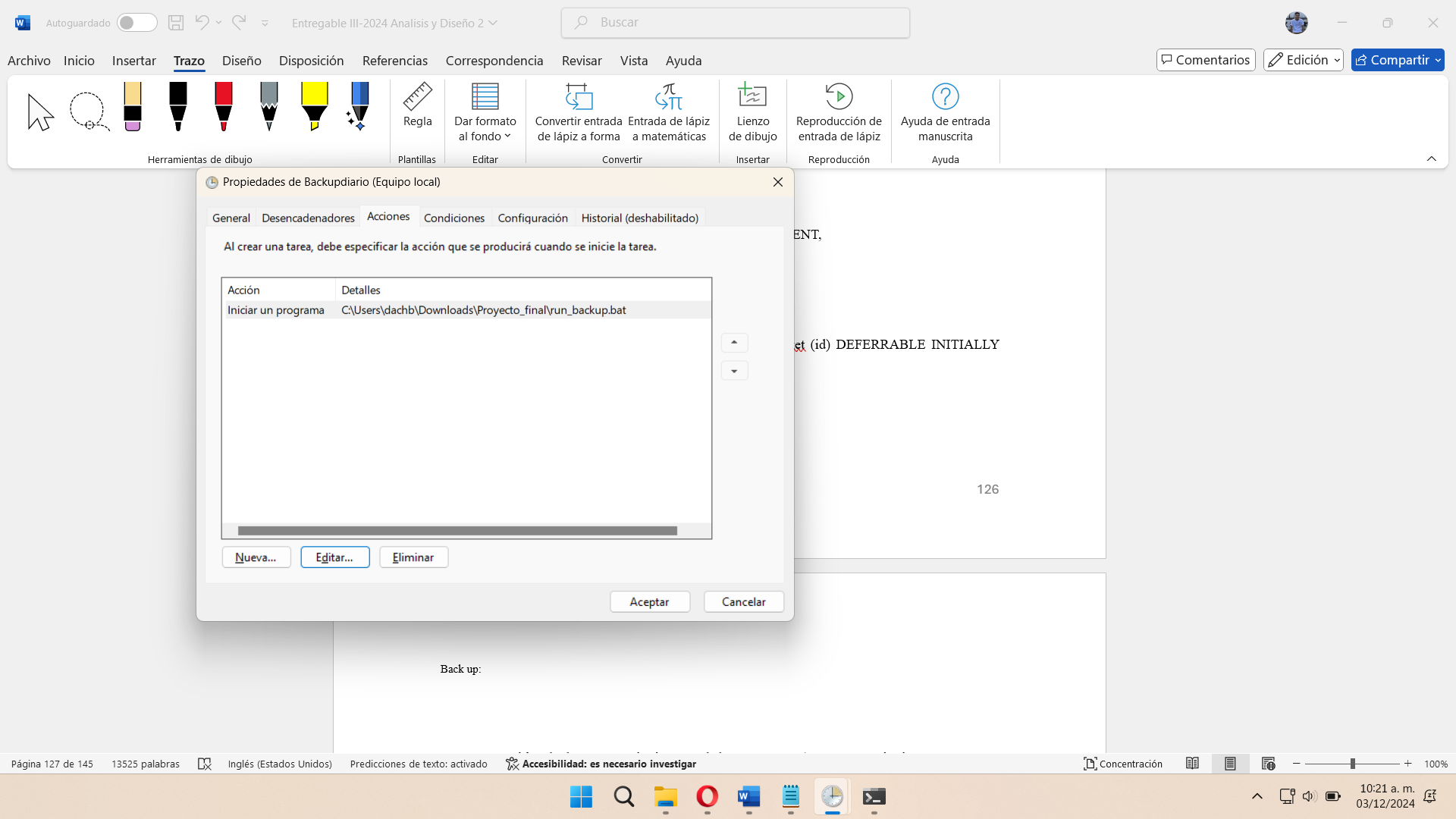
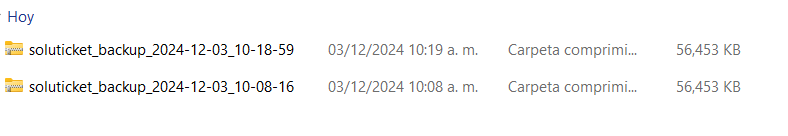


Figura 66: respaldo



# Programación de los requerimientos del proyectos (Los requerimientos modelados en cada diagrama de casos de uso)

**Pagina de Inicio**

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>{% block title %}Soluticket - Conócenos{% endblock %}</title>

    {% load static %}

    {% load crispy\_forms\_tags %}

    <link rel="stylesheet" href="{% static 'css/main.css' %}">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

    {% block extra\_head %}{% endblock %}

    <style>

        body {

            background-color: #f8f9fa;

        }

        .navbar-light .navbar-brand, .nav-link, h2, p {

            color: #1a73e8;

        }

        .navbar-light .navbar-nav .nav-link.active, .navbar-light .navbar-nav .nav-link:hover {

            color: #0047ab;

        }

        .btn-primary {

            background-color: #1a73e8;

            border-color: #1a73e8;

        }

        .btn-primary:hover {

            background-color: #0047ab;

            border-color: #0047ab;

        }

        .btn-crear-ticket, .btn-ver-tickets {

            color: #fff;

            background-color: #28a745;

            border-radius: 5px;

            padding: 8px 16px;

            font-weight: bold;

        }

        .btn-crear-ticket:hover, .btn-ver-tickets:hover {

            background-color: #218838;

        }

        .btn-cerrar-sesion {

            color: #fff;

            background-color: #dc3545;

            border-radius: 5px;

            padding: 8px 16px;

            font-weight: bold;

            margin-left: 10px;

        }

        .btn-cerrar-sesion:hover {

            background-color: #c82333;

        }

        .btn-iniciar-sesion, .btn-registrarse {

            color: #fff;

            border-radius: 5px;

            padding: 8px 16px;

            font-weight: bold;

            margin-left: 10px;

        }

        .btn-iniciar-sesion {

            background-color: #1a73e8;

        }

        .btn-iniciar-sesion:hover {

            background-color: #0047ab;

        }

        .btn-registrarse {

            background-color: #17a2b8;

        }

        .btn-registrarse:hover {

            background-color: #138496;

        }

        .shadowed-container {

            background-color: #ffffff;

            border-radius: 8px;

            padding: 20px;

            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

            margin-top: 20px;

        }

    </style>

</head>

<body>

    <!-- Barra de navegación -->

    <nav class="navbar navbar-expand-lg navbar-light bg-light">

        <a class="navbar-brand" href="/">Soluticket</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"

                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ms-auto">

                {% if user.is\_authenticated %}

                    {% if is\_cliente %}

                    <li class="nav-item ms-2">

                        <a class="nav-link btn-crear-ticket" href="/Tickets/crear-ticket/">Crear Ticket</a>

                    </li>

                    <li class="nav-item ms-2">

                        <a class="nav-link btn-ver-tickets" href="{% url 'ver\_mis\_tickets' %}">Ver Mis Tickets</a>

                    </li>

                    {% elif is\_agente %}

                    <li class="nav-item">

                        <a class="nav-link btn btn-primary text-white" href="{% url 'agente\_dashboard' %}">Dashboard Agente</a>

                    </li>

                    <li class="nav-item ms-2">

                        <a class="nav-link btn-ver-tickets" href="{% url 'consultar\_tickets' %}">Incidencias</a>

                    </li>

                    {% endif %}

                    <li class="nav-item">

                        <a class="nav-link btn-cerrar-sesion" href="{% url 'account\_logout' %}">Cerrar Sesión</a>

                    </li>

                {% else %}

                    <li class="nav-item">

                        <a class="nav-link btn-iniciar-sesion" href="{% url 'account\_login' %}">Iniciar Sesión</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link btn-registrarse" href="{% url 'account\_signup' %}">Registrarse</a>

                    </li>

                {% endif %}

            </ul>

        </div>

    </nav>

    <!-- Contenido principal -->

    <div class="container mt-4">

        <!-- Logo centrado -->

        <div class="text-center mb-5">

            <img src="{% static 'img/logo.png' %}" alt="Logo Soluticket" class="img-fluid" style="max-width: 200px;">

        </div>

        <!-- Sección Conócenos -->

        <div class="shadowed-container my-5">

            <section id="conocenos">

                <h2>Conócenos</h2>

                <p>

                    Soluticket, nos especializamos en brindar soluciones eficientes de Helpdesk para mejorar la gestión de tickets y la atención al cliente. Nuestro equipo de expertos se dedica a ofrecer un servicio de soporte técnico rápido y eficaz, ayudando a las empresas a resolver incidentes con mayor agilidad y eficiencia. Nos comprometemos a optimizar la experiencia del usuario, asegurando que cada solicitud sea atendida con la máxima prioridad.

                </p>

            </section>

        </div>

        <!-- Sección Quiénes Somos -->

        <div class="shadowed-container my-5">

            <section id="quienes-somos">

                <h2>Quiénes Somos</h2>

                <p>

                    Soluticket es una empresa especializada en soluciones de gestión de tickets y Helpdesk, con una sólida trayectoria en el sector. Nuestra misión es simplificar y agilizar el proceso de atención al cliente, proporcionando herramientas eficientes para que las empresas puedan gestionar solicitudes y resolver incidencias de manera rápida y efectiva. Nos esforzamos por ofrecer un servicio accesible, confiable y adaptado a las necesidades de nuestros usuarios.

                </p>

            </section>

        </div>

        <!-- Tarjetas con imágenes -->

        <div class="row">

            <div class="col-md-4">

                <div class="card">

                    <img src="{% static 'img/ejemplo2.png' %}" class="card-img-top" alt="Equipo 1">

                    <div class="card-body">

                        <h5 class="card-title">Nuestro Equipo</h5>

                        <p class="card-text">Conoce a los profesionales detrás de Soluticket.</p>

                    </div>

                </div>

            </div>

            <div class="col-md-4">

                <div class="card">

                    <img src="{% static 'img/ejemplo.png' %}" class="card-img-top" alt="Equipo 2">

                    <div class="card-body">

                        <h5 class="card-title">Nuestros Valores</h5>

                        <p class="card-text">Nos guiamos por la integridad y la innovación.</p>

                    </div>

                </div>

            </div>

            <div class="col-md-4">

                <div class="card">

                    <img src="{% static 'img/ejemplo3.png' %}" class="card-img-top" alt="Misión">

                    <div class="card-body">

                        <h5 class="card-title">Nuestra Misión</h5>

                        <p class="card-text">Transformar y resolver tus problemas de forma ágil, precisa y rapida.</p>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <!-- Pie de página -->

    <footer class="footer bg-light py-3 mt-auto">

        <div class="container text-center">

            <span class="text-muted">© 2024 Soluticket. Todos los derechos reservados.</span>

        </div>

    </footer>

    <!-- Scripts JS -->

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

    {% block extra\_scripts %}{% endblock %}

</body>

</html>

**Gestión de generación de ticket:**

{% extends "base.html" %}

{% block title %}Lista de Tickets{% endblock %}

{% block content %}

<div class="container my-4 p-4" style="background-color: #f7faff; border-radius: 8px; box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);">

    <h2 class="text-center text-primary mb-4">Lista de Tickets</h2>

    <!-- Formulario de filtros -->

    <form method="get" class="mb-4">

        <div class="row">

            <div class="col-md-3">

                <label for="tipo\_servicio" class="form-label">Tipo de Servicio</label>

                <select name="tipo\_servicio" id="tipo\_servicio" class="form-control">

                    <option value="">Todos</option>

                    <option value="servicio1" {% if tipo\_servicio == "servicio1" %}selected{% endif %}>Servicio 1</option>

                    <option value="servicio2" {% if tipo\_servicio == "servicio2" %}selected{% endif %}>Servicio 2</option>

                </select>

            </div>

            <div class="col-md-3">

                <label for="prioridad" class="form-label">Prioridad</label>

                <select name="prioridad" id="prioridad" class="form-control">

                    <option value="">Todas</option>

                    <option value="alta" {% if prioridad == "alta" %}selected{% endif %}>Alta</option>

                    <option value="media" {% if prioridad == "media" %}selected{% endif %}>Media</option>

                    <option value="baja" {% if prioridad == "baja" %}selected{% endif %}>Baja</option>

                </select>

            </div>

            <div class="col-md-3">

                <label for="estado" class="form-label">Estado</label>

                <select name="estado" id="estado" class="form-control">

                    <option value="">Todos</option>

                    <option value="abierto" {% if estado == "abierto" %}selected{% endif %}>Abierto</option>

                    <option value="en\_proceso" {% if estado == "en\_proceso" %}selected{% endif %}>En Proceso</option>

                    <option value="cerrado" {% if estado == "cerrado" %}selected{% endif %}>Cerrado</option>

                </select>

            </div>

            <div class="col-md-3">

                <label for="asignado\_a" class="form-label">Asignado a</label>

                <input type="text" name="asignado\_a" id="asignado\_a" class="form-control" placeholder="Usuario" value="{{ asignado\_a }}">

            </div>

        </div>

        <div class="row mt-3">

            <div class="col-md-12 text-center">

                <button type="submit" class="btn btn-primary">Filtrar</button>

                <a href="{% url 'consultar\_tickets' %}" class="btn btn-secondary">Restablecer Filtros</a>

            </div>

        </div>

    </form>

    <!-- Tabla de tickets -->

    <table class="table table-hover">

        <thead style="background-color: #007bff; color: white;">

            <tr>

                <th>Nombre del Solicitante</th>

                <th>Servicio</th>

                <th>Prioridad</th>

                <th>Estado</th>

                <th>Asignado a</th>

                <th>Fecha de Creación</th>

                <th>Acción</th>

            </tr>

        </thead>

        <tbody>

            {% for ticket in tickets %}

            <tr style="background-color: {% cycle '#ffffff' '#f0f8ff' %};">

                <td>{{ ticket.nombre\_solicitante }}</td>

                <td>{{ ticket.servicio }}</td>

                <td>{{ ticket.prioridad }}</td>

                <td>

                    <span class="badge" style="background-color: #007bff; color: white;">

                        {{ ticket.get\_estado\_display }}

                    </span>

                </td>

                <td>

                    {% if ticket.asignado\_a %}

                        {{ ticket.asignado\_a.get\_full\_name|default:ticket.asignado\_a.username }}

                    {% else %}

                        <span class="text-muted">Sin asignar</span>

                    {% endif %}

                </td>

                <td>{{ ticket.fecha\_creacion|date:"d/m/Y H:i" }}</td>

                <td>

                    <div class="d-flex flex-column gap-2">

                        <button type="button" class="btn btn-primary btn-sm" data-bs-toggle="modal" data-bs-target="#confirmModal" data-ticket-id="{{ ticket.id }}">

                            Tomar Ticket

                        </button>

                        <button type="button" class="btn btn-info btn-sm" data-bs-toggle="modal" data-bs-target="#descriptionModal" data-ticket-id="{{ ticket.id }}" data-ticket-description="{{ ticket.descripcion|default:'Sin descripción' }}">

                            Ver Descripción

                        </button>

                    </div>

                </td>

            </tr>

            {% empty %}

            <tr>

                <td colspan="7" class="text-center text-muted">No hay tickets disponibles.</td>

            </tr>

            {% endfor %}

        </tbody>

    </table>

    <!-- Modal para mostrar la descripción -->

    <div class="modal fade" id="descriptionModal" tabindex="-1" aria-labelledby="descriptionModalLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="descriptionModalLabel">Descripción del Ticket</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body" id="ticketDescriptionContent">

                    <!-- Aquí se mostrará la descripción -->

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>

                </div>

            </div>

        </div>

    </div>

    <!-- Modal para tomar ticket -->

    <div class="modal fade" id="confirmModal" tabindex="-1" aria-labelledby="confirmModalLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="confirmModalLabel">Confirmación</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                    ¿Estás seguro de que deseas tomar este ticket?

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

                    <button type="button" class="btn btn-primary" id="confirmAcceptButton">Aceptar</button>

                </div>

            </div>

        </div>

    </div>

    <!-- Formulario oculto para tomar ticket -->

    <form id="takeTicketForm" action="" method="post" style="display: none;">

        {% csrf\_token %}

    </form>

    {% block extra\_scripts %}

    <script>

        document.addEventListener('DOMContentLoaded', function () {

            // Modal para tomar ticket

            var confirmModal = document.getElementById('confirmModal');

            var confirmAcceptButton = document.getElementById('confirmAcceptButton');

            var takeTicketForm = document.getElementById('takeTicketForm');

            confirmModal.addEventListener('show.bs.modal', function (event) {

                var button = event.relatedTarget;

                var ticketId = button.getAttribute('data-ticket-id');

                takeTicketForm.action = `{% url 'tomar\_ticket' 0 %}`.replace('/0/', `/${ticketId}/`);

            });

            confirmAcceptButton.addEventListener('click', function () {

                takeTicketForm.submit();

            });

            // Modal para mostrar descripción

            var descriptionModal = document.getElementById('descriptionModal');

            var descriptionContent = document.getElementById('ticketDescriptionContent');

            descriptionModal.addEventListener('show.bs.modal', function (event) {

                var button = event.relatedTarget; // Botón que activó el modal

                var description = button.getAttribute('data-ticket-description'); // Obtener la descripción

                descriptionContent.textContent = description ? description : "Sin descripción"; // Asignar al modal

            });

        });

    </script>

    {% endblock %}

    {% endblock %}

# Programación de las interfaces de entradas y procesos(transacción DML BD o en la apps). DML (Insert, delete, update, select)

@group\_required('agente')

def consultar\_tickets(request):

    # Obtiene los valores de los filtros

    filtro\_servicio = request.GET.get('tipo\_servicio', '')

    filtro\_prioridad = request.GET.get('prioridad', '')

    filtro\_estado = request.GET.get('estado', '')

    filtro\_asignado\_a = request.GET.get('asignado\_a', '')

    # Obtiene todos los tickets inicialmente

    tickets = Ticket.objects.all()

    # Filtra por servicio si el usuario selecciona una opción

    if filtro\_servicio:

        tickets = tickets.filter(servicio=filtro\_servicio)

    # Filtra por prioridad si el usuario selecciona una opción

    if filtro\_prioridad:

        tickets = tickets.filter(prioridad=filtro\_prioridad)

    # Filtra por estado si el usuario selecciona una opción

    if filtro\_estado:

        tickets = tickets.filter(estado=filtro\_estado)

    # Filtra por asignado\_a si el usuario ingresa un nombre de usuario

    if filtro\_asignado\_a:

        tickets = tickets.filter(asignado\_a\_\_username=filtro\_asignado\_a)  # Puedes ajustar para usar ID si es necesario

    # Pasa los tickets filtrados y los valores de los filtros al contexto

    context = {

        'tickets': tickets,

        'tipo\_servicio': filtro\_servicio,

        'prioridad': filtro\_prioridad,

        'estado': filtro\_estado,

        'asignado\_a': filtro\_asignado\_a,

    }

    return render(request, 'get\_tickets.html', context)

@group\_required('agente')

@login\_required

def tomar\_ticket(request, ticket\_id):

    print("Ticket ID:", ticket\_id)  # Agrega esta línea para verificar el ID en la consola

    ticket = get\_object\_or\_404(Ticket, id=ticket\_id)

    if ticket.asignado\_a:

        messages.warning(request, "Este ticket ya ha sido tomado por otro usuario.")

    else:

        ticket.asignado\_a = request.user

        ticket.estado = 'en\_proceso'

        ticket.save()

        messages.success(request, "Has tomado el ticket exitosamente.")

    tickets = Ticket.objects.all()

    return render(request, 'get\_tickets.html', {'tickets': tickets})

@group\_required('agente')

@login\_required

def agente\_dashboard(request):

    agente = request.user

    tickets\_asignados = Ticket.objects.filter(asignado\_a=agente)

    # Filtrar tickets por estado

    abiertos = tickets\_asignados.filter(estado="abierto")

    en\_proceso = tickets\_asignados.filter(estado="en\_proceso")

    cerrados = tickets\_asignados.filter(estado="cerrado")

    # Instancias de los formularios para el estado y el comentario

    ticket\_form = TicketForm()  # Formulario para el campo de estado

    comment\_form = CommentForm()  # Formulario para agregar un comentario

    context = {

        "total\_tickets": tickets\_asignados.count(),

        "abiertos": abiertos,

        "en\_proceso": en\_proceso,

        "cerrados": cerrados,

        "tickets\_asignados": tickets\_asignados,

        "ticket\_form": ticket\_form,

        "comment\_form": comment\_form,

    }

    return render(request, "agente\_dashboard.html", context)

@group\_required('agente')

@login\_required

def editar\_ticket(request, ticket\_id):

    ticket = get\_object\_or\_404(Ticket, id=ticket\_id, asignado\_a=request.user)

    if request.method == "POST":

        ticket\_form = TicketForm(request.POST, instance=ticket)

        comment\_form = CommentForm(request.POST)

        if ticket\_form.is\_valid():

            ticket\_form.save()

            if comment\_form.is\_valid():

                comment = comment\_form.save(commit=False)

                comment.ticket = ticket

                comment.user = request.user

                comment.save()

            return redirect(reverse("agente\_dashboard"))

    else:

        ticket\_form = TicketForm(instance=ticket)

        comment\_form = CommentForm()

    return render(request, "agente\_dashboard.html", {

        "ticket\_form": ticket\_form,

        "comment\_form": comment\_form,

        "tickets\_asignados": Ticket.objects.filter(asignado\_a=request.user),

    })

@login\_required

@group\_required('cliente')

def crear\_ticket(request):

    if request.method == 'POST':

        form = CrearTicketForm(request.POST)

        if form.is\_valid():

            tipo\_name = request.POST.get('tipo\_ticket')  # Obtener el nombre del tipo desde el formulario

            if not tipo\_name:

                messages.error(request, "Por favor, selecciona un tipo de ticket válido.")

                return redirect('crear\_ticket')

            try:

                tipo = Type.objects.get(name=tipo\_name)  # Buscar el objeto Type correspondiente

            except Type.DoesNotExist:

                messages.error(request, "El tipo de ticket seleccionado no existe.")

                return redirect('crear\_ticket')

            ticket = form.save(commit=False)  # No guardar todavía

            ticket.servicio = tipo  # Asignar el objeto Type al campo 'servicio'

            ticket.nombre\_solicitante = request.user  # Asignar el usuario autenticado como dueño

            ticket.save()  # Guardar el ticket con el usuario asignado

            messages.success(request, "El ticket se creó correctamente.")

            return redirect('ver\_mis\_tickets')  # Cambia a la vista que prefieras tras éxito

        else:

            messages.error(request, "Por favor, corrige los errores del formulario.")

    else:

        form = CrearTicketForm()

    # Separar lógica para obtener tipos por categoría

    tipos\_software, tipos\_hardware = obtener\_tipos\_por\_categoria()

    return render(request, 'post\_ticket.html', {

        'form': form,

        'tipos\_software': tipos\_software,

        'tipos\_hardware': tipos\_hardware

    })

def obtener\_tipos\_por\_categoria():

    """Obtiene los tipos de ticket agrupados por categoría."""

    tipos\_software = Type.objects.filter(category\_\_name="Software")

    tipos\_hardware = Type.objects.filter(category\_\_name="Hardware")

    return tipos\_software, tipos\_hardware

def generar\_pdf\_xhtml2pdf(request):

    # Datos para la plantilla

    context = {

        'titulo': 'Reporte de Ejemplo',

        'contenido': 'Este es un reporte generado con xhtml2pdf.',

    }

    # Renderizar la plantilla HTML

    html = render\_to\_string('reporte.html', context)

    # Crear el archivo PDF

    response = HttpResponse(content\_type='application/pdf')

    response['Content-Disposition'] = 'attachment; filename="reporte.pdf"'

    pisa.CreatePDF(html, dest=response)

    return response

from django.views.generic import ListView

from .models import Ticket

class MisTicketsView( ListView):

    model = Ticket

    template\_name = "mis\_tickets.html"

    context\_object\_name = "tickets"

    def get\_queryset(self):

        # Filtrar tickets solo para el usuario autenticado

        return Ticket.objects.filter(nombre\_solicitante=self.request.user)

def dar\_feedback(request):

    if request.method == "POST":

        ticket\_id = request.POST.get('ticket\_id')

        satisfaccion = request.POST.get('satisfaccion')

        comentarios = request.POST.get('comentarios')

        # Validar si el ticket existe

        ticket = get\_object\_or\_404(Ticket, id=ticket\_id)

        # Verificar si ya existe un feedback para este ticket

        if Feedback.objects.filter(ticket=ticket).exists():

            messages.error(request, "Ya has enviado un feedback para este ticket.")

            return redirect('ver\_mis\_tickets')

        # Crear el feedback

        try:

            Feedback.objects.create(

                ticket=ticket,

                satisfaccion=int(satisfaccion),

                comentarios=comentarios

            )

            messages.success(request, "Gracias por tu feedback.")

        except Exception as e:

            messages.error(request, f"Ocurrió un error al guardar tu feedback: {e}")

            return redirect('ver\_mis\_tickets')

    return redirect('mis\_tickets')

from django.http import JsonResponse

from .models import Comment

def obtener\_comentarios(request, ticket\_id):

    try:

        comentarios = Comment.objects.filter(ticket\_id=ticket\_id).select\_related('user')

        comentarios\_json = [

            {

                "user": comentario.user.get\_full\_name() if comentario.user else None,

                "content": comentario.content,

                "created\_at": comentario.created\_at,

            }

            for comentario in comentarios

        ]

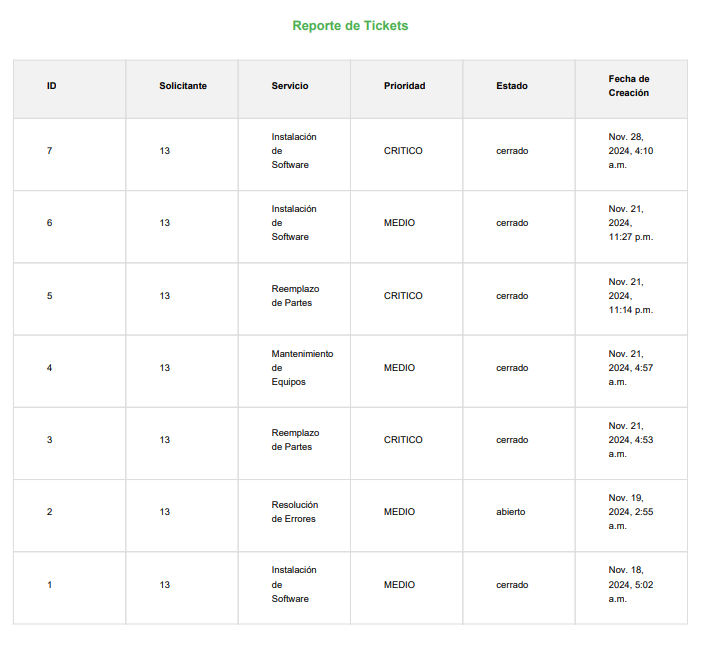
        return JsonResponse({"comments": comentarios\_json})

    except Exception as e:

        return JsonResponse({"error": str(e)}, status=500)

# Programación de las interfaces de Salidas los reportes deben de exportarse en formato: xls, txt y pdf

El Sistema mantiene la capacidad de generar reportes en pdf de los tickets gestionados



<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>{{ titulo }}</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 20px;

        }

        h1 {

            text-align: center;

            color: #4CAF50;

        }

        table {

            width: 100%;

            border-collapse: collapse;

            margin-top: 20px;

        }

        th, td {

            border: 1px solid #ddd;

            padding: 8px;

            text-align: left;

        }

        th {

            background-color: #f2f2f2;

        }

    </style>

</head>

<body>

    <h1>{{ titulo }}</h1>

    <table>

        <thead>

            <tr>

                <th>ID</th>

                <th>Solicitante</th>

                <th>Servicio</th>

                <th>Prioridad</th>

                <th>Estado</th>

                <th>Fecha de Creación</th>

            </tr>

        </thead>

        <tbody>

            {% for ticket in tickets %}

            <tr>

                <td>{{ ticket.id }}</td>

                <td>{{ ticket.nombre\_solicitante }}</td>

                <td>{{ ticket.servicio\_\_name }}</td> <!-- Mostrar el campo name del servicio -->

                <td>{{ ticket.prioridad\_\_description }}</td>

                <td>{{ ticket.estado }}</td>

                <td>{{ ticket.fecha\_creacion }}</td>

            </tr>

            {% endfor %}

        </tbody>

    </table>

</body>

</html>

# Casos de Pruebas del Productos desarrollado (Probar las funcionabilidad e integración de sistemas). Tipos de pruebas las documentas los tipos pruebas que le hiciste, debe se seleccionar los datos pruebas. Determiné que la prueba satisfactorias (Certificación de pruebas ++)

Creación de agente:

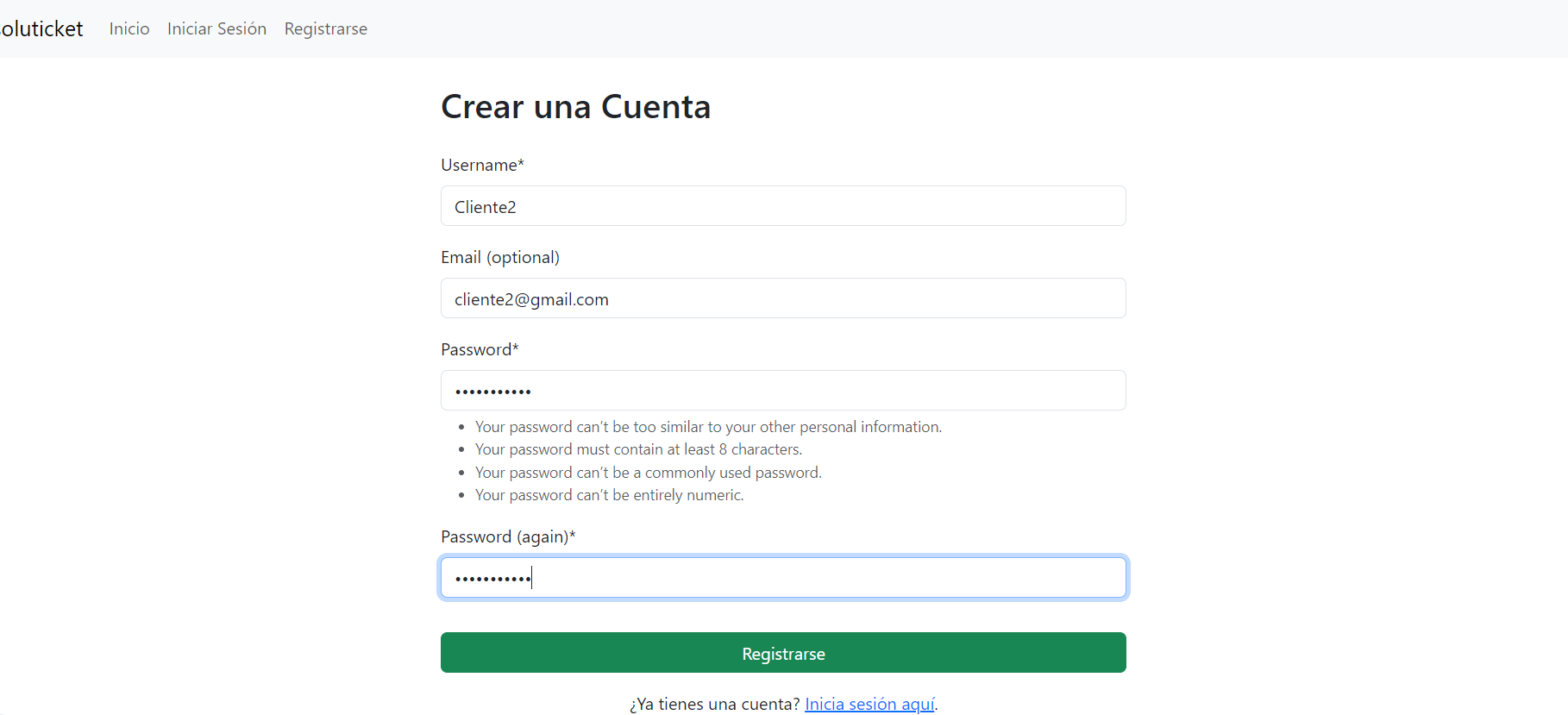
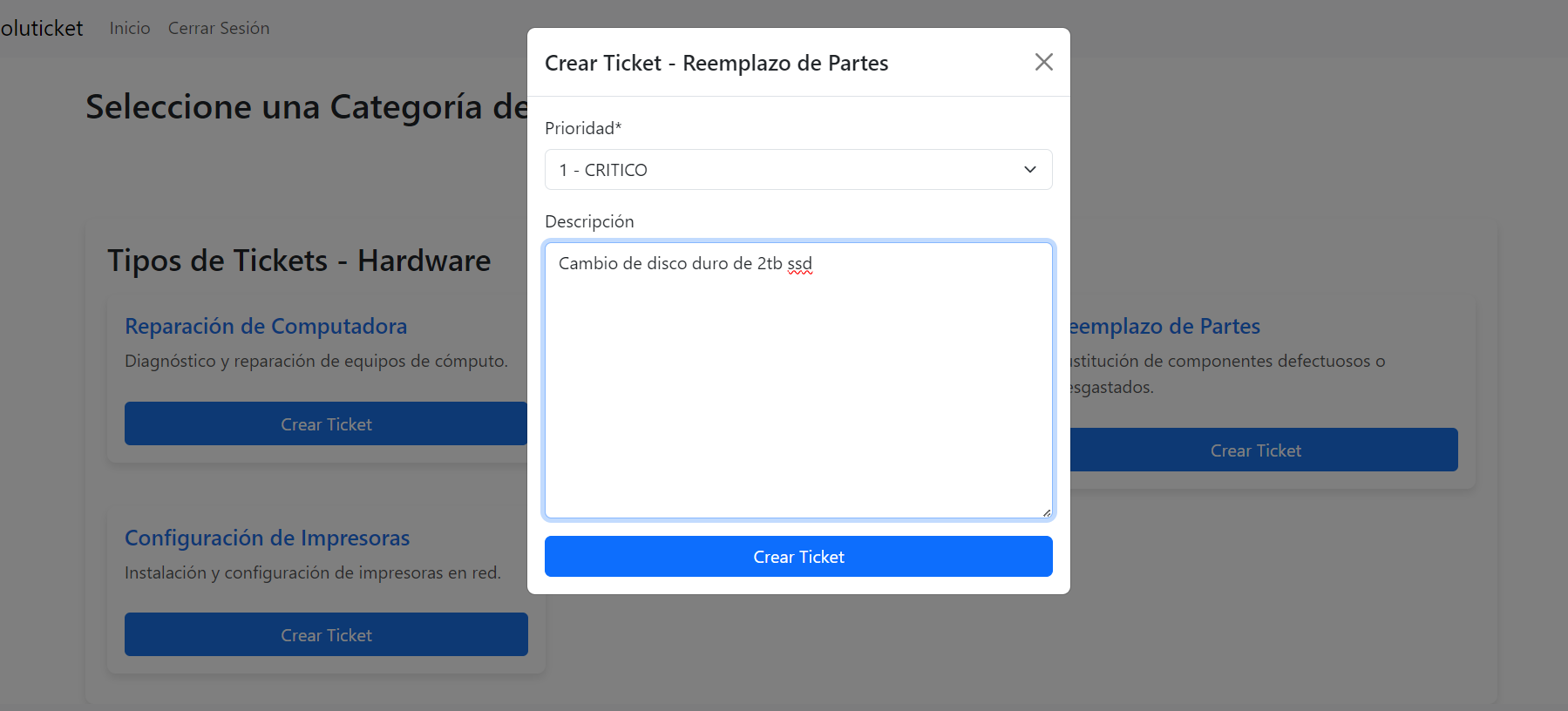
Creación de Cliente:  
  


Figura: Login

Creación de Ticket



-Figura: registro ticket

Generación de reporte actualizado:



Figura: Ticket visibles

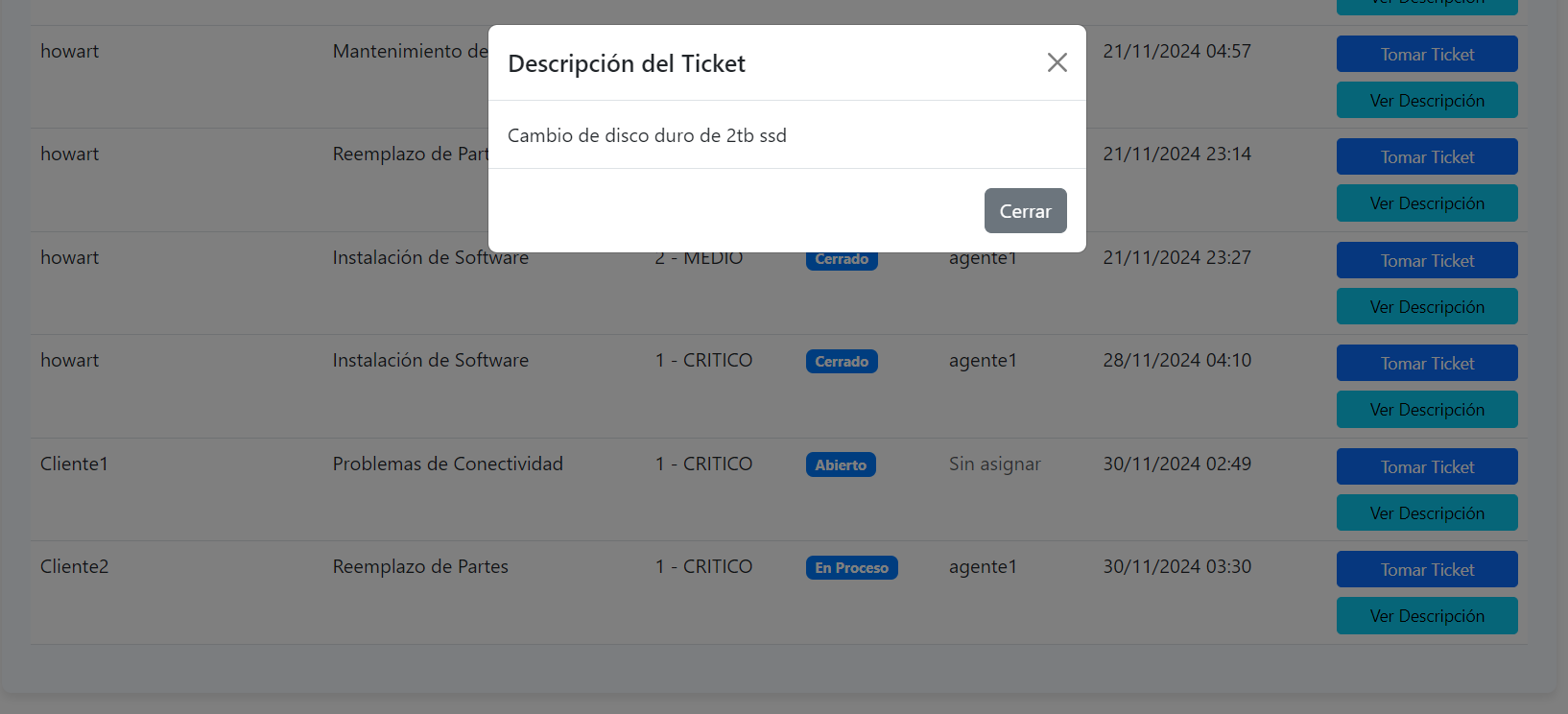


Figura: Verificacion de informacion

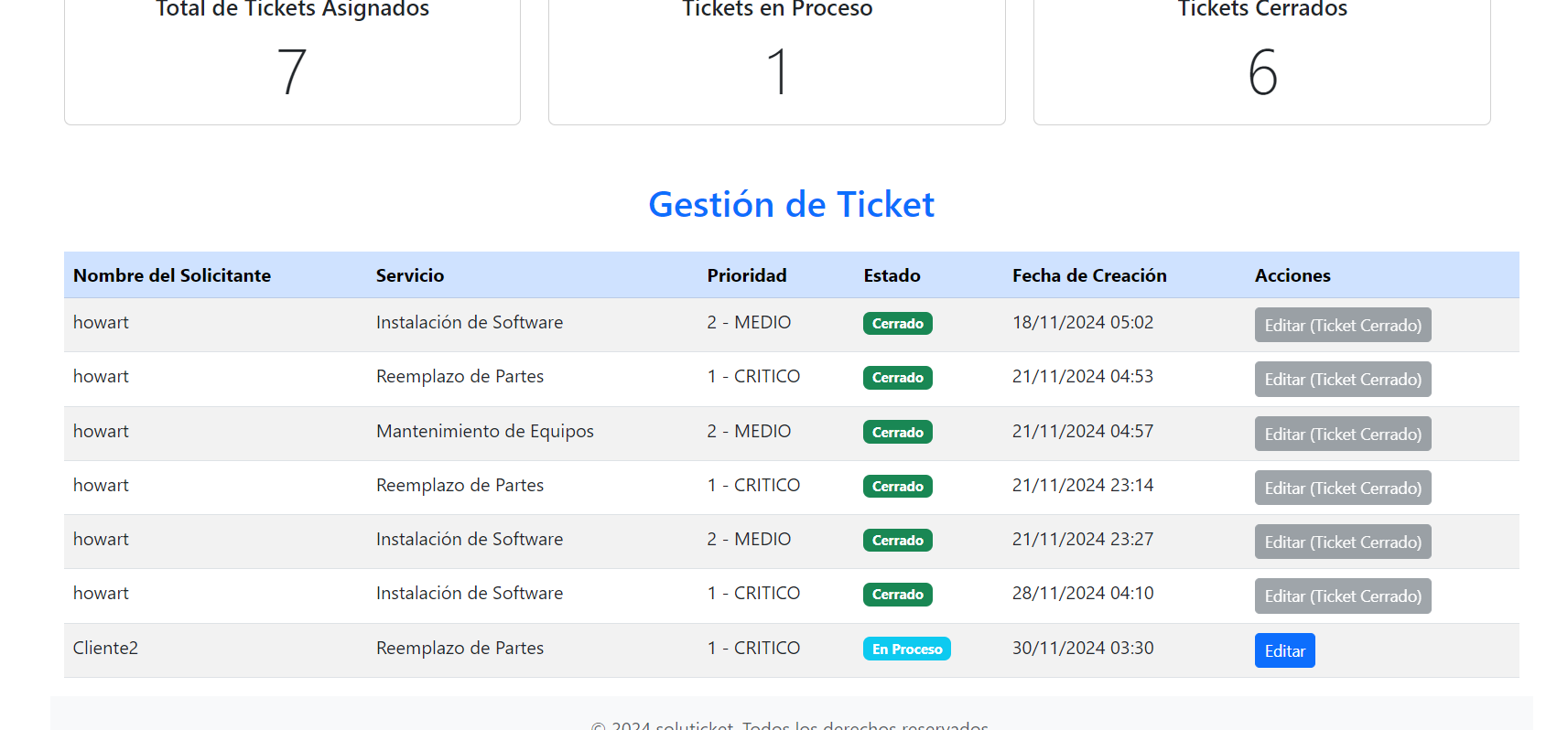


Figura: Dashboard de agente

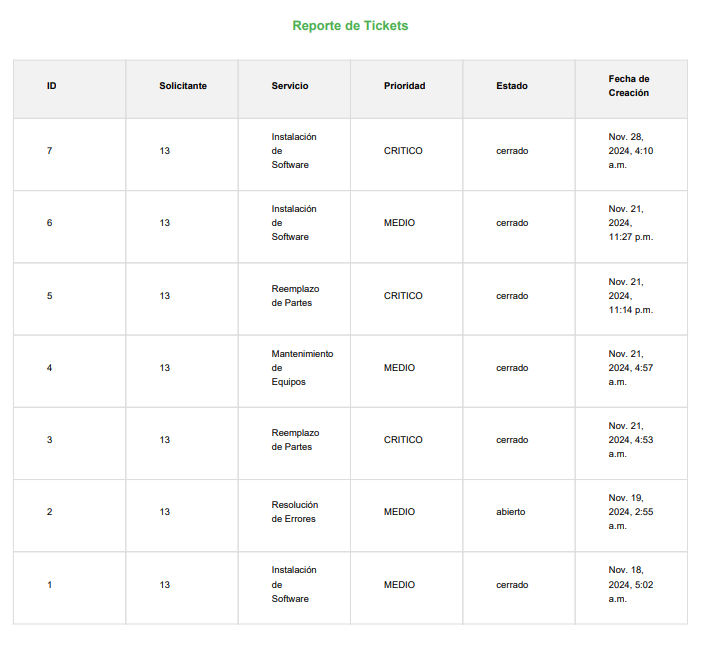


Figura: Reporteria

# **Conclusión**

La implementación del sistema de análisis y gestión para la mesa de servicio ha resultado en una notable mejora en la eficiencia operativa. Al adoptar la metodología Kanban, logramos visualizar el flujo de trabajo de manera clara, lo que facilitó la identificación de cuellos de botella y permitió priorizar tareas de manera efectiva. Esto se tradujo en una reducción significativa en los tiempos de respuesta y en la resolución de incidencias, mejorando la satisfacción del usuario final.

Además, el sistema ha promovido una cultura de mejora continua dentro del equipo. La capacidad de revisar y ajustar constantemente nuestros procesos ha permitido implementar cambios basados en datos reales, lo que ha llevado a optimizar la asignación de recursos y a aumentar la productividad. La colaboración y la comunicación entre los miembros del equipo también se han fortalecido, lo que ha generado un ambiente de trabajo más cohesivo y motivado.

Por último, la transparencia que ofrece el sistema ha facilitado una mejor gestión de las expectativas de las partes interesadas. Con informes y métricas claras, hemos podido comunicar de manera efectiva los avances y resultados alcanzados, estableciendo un ciclo de retroalimentación que beneficia tanto al equipo como a los usuarios del servicio. En conjunto, estos resultados demuestran que la implementación del sistema ha sido un paso crucial hacia la excelencia en la gestión de nuestra mesa de servicio. (IDRIC, 2022)

# **Referencias**

1. Aranda, M. (2019, October 1). *¿Por qué mi empresa necesita una Mesa de Servicio?* Aranda Software. https://arandasoft.com/blog/por-que-mi-empresa-necesita-una-mesa-de-servicio/
2. Atlassian. (n.d.). *Introducción a Jira: guía completa para principiantes*. Atlassian. Retrieved October 11, 2024, from https://www.atlassian.com/es/software/jira/guides/getting-started/introduction
3. Formacion, A. D. R. (2024, June 14). *¿Cuáles son los beneficios clave de Jira? - ADR Formación*. Adrformacion.com; ADR Formacion. https://www.adrformacion.com/knowledge/transformacion-digital/\_cuales\_son\_los\_beneficios\_clave\_de\_jira\_.html
4. Freshis. (2022, June 29). *Propiedades y beneficios del melón*. freshis. https://freshis.com/blog/propiedades-beneficios-del-melon
5. ITIL - Biblioteca de infraestructura de TI. (2023, March 14). *Ibm.com*. https://www.ibm.com/es-es/topics/it-infrastructure-library
6. Martins, J. (2024, January 19). *¿Qué es la metodología Kanban y cómo funciona?* Asana. https://asana.com/es/resources/what-is-kanban
7. Romero, A. (2014, November 26). 5 Pecados comunes en la gestión de un Service Desk - OpenService. *OpenService - Servicios Administrados de TI*. https://openservice.mx/5-pecados-comunes-en-la-gestion-de-un-service-desk/
8. *Service desk: ¿Qué es? Sus características y ventajas*. (2023, October 11). Zendesk. https://www.zendesk.com.mx/blog/mesa-de-servicio-que-es/
9. *Tutorial de Oracle*. (n.d.). Desarrolloweb.com. Retrieved October 10, 2024, from https://desarrolloweb.com/manuales/tutorial-oracle.html
10. (N.d.). Com.Mx. Retrieved October 8, 2024, from <https://www.idric.com.mx/blog/post/la-importancia-de-contar-con-una-mesa-de-servicio>
11. *Bonitasoft*. (n.d.). Bonitasoft.com. Retrieved December 6, 2024, from https://es.bonitasoft.com
12. *El tutorial de Python*. (n.d.). Python documentation. Retrieved December 6, 2024, from https://docs.python.org/es/3/tutorial/
13. Febrer, M. (2022, October 20). *DBeaver: software para la gestión de bases de datos*. Geoinnova; GEOINNOVA SL. https://geoinnova.org/blog-territorio/dbeaver-gestion-de-bases-de-datos/
14. Fernández, Y. (2019, October 30). *Qué es Github y qué es lo que le ofrece a los desarrolladores*. Xataka.com; Xataka Basics. https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores
15. Formaci�n, A. D. R. (2023, November 14). *Por qu� Python es tan popular como lenguaje de programaci�n*. Adrformacion.com; ADR Formaci�n. https://www.adrformacion.com/knowledge/programacion/por\_que\_python\_es\_tan\_popular\_como\_lenguaje\_de\_programacion.html
16. *Framework Web Django (Python)*. (n.d.). MDN Web Docs. Retrieved December 6, 2024, from https://developer.mozilla.org/es/docs/Learn/Server-side/Django
17. *Login*. (n.d.). Figma. Retrieved December 6, 2024, from https://www.figma.com/files/team/1445176207748611950/recents-and-sharing?fuid=1445176206303710771
18. *No title*. (n.d.). Google.com. Retrieved December 6, 2024, from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwies9aa\_JGKAxWKSjABHUfDBvoQFnoECB8QAQ&url=https%3A%2F%2Fopenservice.mx%2Fmesa-de-servicio-ti%2F&usg=AOvVaw2Axgg1KQMmEpzQhkxqoty7&opi=89978449
19. *Service desk: ¿Qué es? Sus características y ventajas*. (2023, October 11). Zendesk. https://www.zendesk.com.mx/blog/mesa-de-servicio-que-es/
20. *SQLite: La base de datos embebida*. (n.d.). SG Buzz. Retrieved December 6, 2024, from https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida
21. *SQLite: la famosa biblioteca en detalle*. (2023, June 27). IONOS Digital Guide; IONOS. https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/sqlite/?srsltid=AfmBOoq7mKXO7zC9yk\_rgMI7hRt77hvLQr9WWDQzxMLGpTc3yF2o9qW7
22. *Visual Studio: IDE y Editor de código para desarrolladores de software y Teams*. (2018, September 26). Visual Studio. https://visualstudio.microsoft.com/es/
23. (N.d.). Amazon.com. Retrieved December 6, 2024, from <https://aws.amazon.com/es/what-is/django/>
24. *ChatGPT*. (n.d.). Chatgpt.com. Retrieved December 6, 2024, from https://chatgpt.com
25. *Guía de Desarrollo Web*. (n.d.). MDN Web Docs. Retrieved December 6, 2024, from https://developer.mozilla.org/es/docs/Web/Guide
26. *Manuales*. (n.d.). Desarrolloweb.com. Retrieved December 6, 2024, from https://desarrolloweb.com/manuales
27. Martins, J. (2024, February 15). *Scrum: conceptos clave y cómo se aplica en la gestión de proyectos*. Asana. https://asana.com/es/resources/what-is-scrum
28. *SQLite en PHP*. (2023, April 27). Desarrolloweb.com. https://desarrolloweb.com/articulos/sqlite-php
29. (N.d.). Dreamhost.com. Retrieved December 6, 2024, from <https://help.dreamhost.com/hc/es/articles/217294097-SQLite>
30. Hostinger. (n.d.). *Compra y Registra Dominios Web En Minutos*. Hostinger. Retrieved December 6, 2024, from <https://www.hostinger.es/comprar-dominio?utm_campaign=Brand-Phrase|NT:Se|LO:Other-LATAM-t1&utm_medium=ppc&gad_source=1&gclid=Cj0KCQiAu8W6BhC-ARIsACEQoDCsplVLk1mwSJ6xb7weP6MjgFUN0iBz4K4YhM565VL_TLwWkcEJJTwaAsGvEALw_wcB>