



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

**HELPDESK USANDO HERRAMIENTAS OPEN SOURCE PARA LA
GESTIÓN DE INCIDENCIAS DE SOPORTE TÉCNICO EN EL GAD
MUNICIPAL DE PELILEO**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Tecnologías de la Información

ÁREA: Gestión de Tecnologías de la Información.

LÍNEA DE INVESTIGACIÓN: Software, Tecnologías de la Información y
Ciencia de Datos.

AUTOR: Josué Israel Valverde Endara

TUTOR: Ing. Dennis Vinicio Chicaiza Castillo, Mg.

Ambato - Ecuador

julio – 2026

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: HELPDESK USANDO HERRAMIENTAS OPEN SOURCE PARA LA GESTIÓN DE INCIDENCIAS DE SOPORTE TÉCNICO EN EL GAD MUNICIPAL DE PELILEO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Josué Israel Valverde Endara, estudiante de la Carrera de Tecnologías de la Información de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, julio 2026.

Ing. Dennis Vinicio Chicaiza Castillo, Mg.
TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: **HELPDESK USANDO HERRAMIENTAS OPEN SOURCE PARA LA GESTIÓN DE INCIDENCIAS DE SOPORTE TÉCNICO EN EL GAD MUNICIPAL DE PELILEO** es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, julio 2026.



Josué Israel Valverde Endara

C.C. 1805316062

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, julio 2026.



Josué Israel Valverde Endara

C.C. 1805316062

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Josué Israel Valverde Endara, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado HELPDESK USANDO HERRAMIENTAS OPEN SOURCE PARA LA GESTIÓN DE INCIDENCIAS DE SOPORTE TÉCNICO EN EL GAD MUNICIPAL DE PELILEO, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, julio 2026.

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Oscar Fernando Ibarra Torres, Mg.

PROFESOR CALIFICADOR

Ing. Clay Fernando Aldas Flores, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

A mis padres, Rosita y Jaime, quienes con su amor incondicional, sacrificio diario y ejemplo de vida me enseñaron que los sueños se construyen con esfuerzo y perseverancia. Todo lo que soy y lo que he logrado lleva su huella.

A mis hermanos, mis cómplices y confidentes, cuya confianza inquebrantable en mí fue la brújula que me orientó en los momentos de duda. Su motivación constante me recordó siempre por qué empecé.

A mis amigos y compañeros de vida, con quienes compartí las alegrías de los logros y la fortaleza necesaria para superar las adversidades. La amistad que construimos vale más que cualquier título.

A mi compañera, persona extraordinaria que eligió estar a mi lado en los momentos más difíciles de este camino. Tu presencia, paciencia y apoyo fueron el ancla que me dio estabilidad cuando más lo necesitaba. Gracias por creer en mí incluso cuando yo dudaba.

A todos ustedes, les dedico este trabajo con profunda gratitud y amor.

AGRADECIMIENTO

Agradezco a Dios por guiarme hacia el camino correcto por darme la sabiduría y pensamiento correcto en los momentos difíciles de mi vida.

A mi familia y mis padres que fueron los pilares fundamentales en este largo camino quienes me apoyaron en todo lo posible.

A mis amigos y compañeros que estuvieron conmigo en este largo y retador camino. Cada uno ha contribuido en mi fortaleza y ánimo de una u otra manera.

A mi Tutor el Ing. Dennis Chicaiza. Por su invaluable guía y dedicación, cuyo conocimiento y apoyo fueron esenciales para llevar a cabo y culminar con éxito este proyecto.

A todos ustedes, les dedico este trabajo con profunda gratitud y amor.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xvi
ÍNDICE DE ANEXOS	xix
RESUMEN EJECUTIVO	xx
ABSTRACT	xxi
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	5

1.3.1 Gestión de servicios de tecnologías de la información	5
1.3.2 Gestión de incidencias.....	5
1.3.3 Mesa de ayuda (help desk) vs. mesa de servicio.....	6
1.3.4 Software open source en la administración pública.....	7
1.3.5 Stack tecnológico seleccionado.....	7
1.3.6 Resumen de la arquitectura del sistema	9
1.4 Objetivos	9
1.4.1 Objetivo general.....	9
1.4.2 Objetivos específicos	10
CAPÍTULO II.- METODOLOGÍA.....	11
2.1 Materiales.....	11
2.2 Métodos.....	16
2.2.1 Modalidad de la investigación	16
2.2.2 Población y muestra	17
2.2.3 Recolección de información.....	18
2.2.4 Procesamiento y análisis de datos	34
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	37
3.1 Análisis.....	37
3.1.1 Análisis de los procesos involucrados de incidencias de soporte técnico.....	38
3.1.2 Matriz de evaluación de riesgos del proceso actual de gestión de incidencias .	39
3.2 Herramientas Open Source para el desarrollo.....	43

3.2.1 Selección del Framework Frontend	45
3.2.2 Selección del framework backend	48
3.2.3 Sistema gestor de base de datos	50
3.2.4 Resumen del stack tecnológico seleccionado	51
3.3 Desarrollo.....	53
3.3.1 Proceso identificado para la asistencia técnica	53
3.3.2 Metodología de desarrollo para aplicaciones web	56
Descripción detallada de cada fase	59
3.3.3 Desarrollo de la propuesta.....	60
3.4 Implantación.....	115
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	121
4.1 Conclusiones	121
4.2 Recomendaciones	123
REFERENCIAS BIBLIOGRÁFICAS	124
ANEXOS	127

ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda	3
Tabla 2. Criterios de inclusión y exclusión	3
Tabla 3. Encuesta al personal administrativo.....	11
Tabla 4. Ficha de observación para el personal técnico de TI	15
Tabla 5. Población y Muestra.....	18
Tabla 6. Niveles de confiabilidad según el coeficiente Alfa de Cronbach	19
Tabla 7. Calculo Alfa de Cronbach (α)	20
Tabla 8. Valores del coeficiente Alfa de Cronbach	21
Tabla 9. Resultados de la Entrevista — Jefe del Departamento de TI.....	28
Tabla 10. Resultados de la Entrevista — Técnico de Soporte de TI.....	30
Tabla 11. Resultados de la ficha de observación	32
Tabla 12. Descripción del proceso actual de gestión de incidencias	39
Tabla 13. Escala de valoración de probabilidad e impacto	40
Tabla 14. Clasificación del nivel de riesgo	40
Tabla 15. Mapa de calor — Probabilidad vs. Impacto.....	41
Tabla 16. Matriz de evaluación de riesgos	41
Tabla 17. Esquema de ponderación para el cálculo del puntaje de herramientas Open Source.....	45
Tabla 18. Criterios de evaluación y escala de puntuación	46

Tabla 19. Matriz de selección — Framework Frontend.....	47
Tabla 20. Criterios de evaluación y escala de puntuación	48
Tabla 21. Matriz de selección — Framework Backend.....	49
Tabla 22. Stack tecnológico seleccionado para el sistema Helpdesk.....	51
Tabla 23. Descripción del proceso optimizado mediante el sistema Helpdesk	54
Tabla 24. Comparativa de metodologías de desarrollo ágil.....	57
Tabla 25. Fases de la metodología XP y su aplicación en el sistema Helpdesk	58
Tabla 26. Roles y permisos del sistema Helpdesk	61
Tabla 27. Historia de usuario 1 — acceso al sistema.....	61
Tabla 28. Historia de usuario 2 — recuperar contraseña	62
Tabla 29. Historia de usuario 3 — gestión de usuarios.....	62
Tabla 30. Historia de usuario 4 — Gestión de roles	63
Tabla 31. Historia de usuario 5 — gestión de categorías y prioridades.....	63
Tabla 32. Historia de usuario 6 — gestión de departamentos.....	63
Tabla 33. Historia de usuario 7 — configuración de SLA.....	64
Tabla 34. Historia de usuario 8 — cerrar sesión	64
Tabla 35. Historia de usuario 9 — crear ticket	64
Tabla 36. Historia de usuario 10 — gestión de tickets.....	65
Tabla 37. Historia de usuario 11 — asignación de tickets	65
Tabla 38. Historia de usuario 12 — reapertura de ticket	65
Tabla 39. Historia de usuario 13 — cálculo automático de SLA.....	66

Tabla 40. Historia de usuario 14 — base de conocimiento.....	66
Tabla 41. Historia de usuario 15 — módulo AnyDesk por departamento.....	66
Tabla 42. Historia de usuario 16 — notificaciones automáticas.....	67
Tabla 43. Historia de usuario 17 — reportes y métricas KPIs.....	67
Tabla 44. Historia de usuario 18 — calificación del servicio	67
Tabla 45. Historias de usuario y estimación de esfuerzo	68
Tabla 46. Plan de entrega del sistema Helpdesk	69
Tabla 47. Equipo de trabajo del proyecto	69
Tabla 48. Tarjeta CRC — acceso al sistema	70
Tabla 49. Tarjeta CRC — recuperar contraseña	70
Tabla 50. Tarjeta CRC — gestión de usuarios.....	71
Tabla 51. Tarjeta CRC — gestión de roles	71
Tabla 52. Tarjeta CRC — gestión de categorías y prioridades.....	71
Tabla 53. Tarjeta CRC — gestión de departamentos.....	71
Tabla 54. Tarjeta CRC — configuración de SLA.....	72
Tabla 55. Tarjeta CRC — cerrar sesión	72
Tabla 56. Tarjeta CRC — crear ticket.....	72
Tabla 57. Tarjeta CRC — gestión de tickets.....	72
Tabla 58. Tarjeta CRC — asignación de tickets	73
Tabla 59. Tarjeta CRC — reapertura de ticket.....	73
Tabla 60. Tarjeta CRC — cálculo automático de SLA.....	73

Tabla 61. Tarjeta CRC — base de conocimiento.....	73
Tabla 62. Tarjeta CRC — módulo AnyDesk	74
Tabla 63. Tarjeta CRC — notificaciones automáticas.....	74
Tabla 64. Tarjeta CRC — reportes y métricas KPIs	74
Tabla 65. Tarjeta CRC — calificación del servicio	74
Tabla 66. Resumen de la iteración 1	81
Tabla 67. Tarea 1 — acceso al sistema	81
Tabla 68. Tarea 2 — Recuperar contraseña	84
Tabla 69. Tarea 3 — gestión de usuarios	87
Tabla 70. Tarea 4 — gestión de roles.....	91
Tabla 71. Resumen iteración 2.....	93
Tabla 72. Tarea 5 — gestión de categorías y prioridades	94
Tabla 73. Tarea 6 — gestión de departamentos	95
Tabla 74. Tarea 7 — configuración de SLA	96
Tabla 75. Tarea 8 — cerrar sesión	98
Tabla 76. Resumen iteración 3	99
Tabla 77. Tarea 9 — crear ticket.....	100
Tabla 78. Tarea 10 — gestión de tickets.....	102
Tabla 79. Tarea 11 — Asignación de tickets	103
Tabla 80. Tarea 12 — reapertura de ticket.....	105
Tabla 81. Resumen iteración 4.....	106

Tabla 82. Tarea 14 — base de conocimiento	106
Tabla 83. Tarea 15 — módulo AnyDesk por departamento	107
Tabla 84. Tarea 16 — notificaciones automáticas	108
Tabla 85. Resumen iteración 5	109
Tabla 86. Tarea 17 — reportes y métricas KPIs	109
Tabla 87. Indicadores de desempeño	110
Tabla 88. Tarea 18 — calificación del servicio	111
Tabla 89. Pruebas de aceptación — iteración 1	113
Tabla 90. Pruebas de aceptación — iteración 2	113
Tabla 91. Pruebas de aceptación — iteración 3	114
Tabla 92. Resumen consolidado de pruebas de aceptación	114
Tabla 93. Cronograma de capacitaciones al personal del GAD Municipal de Pelileo	119

ÍNDICE DE FIGURAS

Figura 1. Canal disponible para reporte de incidencias	22
Figura 2. Medios de comunicación utilizados para soporte.....	23
Figura 3. Dificultad para comunicar incidencias	23
Figura 4. Frecuencia de atención del departamento de TI	24
Figura 5. Evaluación de efectividad del soporte técnico	25
Figura 6. Tiempo de atención de requerimientos técnicos.....	25
Figura 7. Retroalimentación técnica al usuario tras la resolución	26
Figura 8. Frecuencia de actualizaciones sobre el progreso de incidencias	27
Figura 9. Aceptación de implementación del sistema Helpdesk	27
Figura 10. Estadísticas de respuestas de la encuesta.....	35
Figura 11. Gestión de incidencias identificado mediante la ficha de observación ...	36
Figura 12. Gestión de incidencias identificado	38
Figura 13. Entorno tecnológico del sistema Helpdesk desarrollado para el GAD Municipal de Pelileo.	52
Figura 14. Proceso optimizado con sistema Helpdesk.....	54
Figura 15. Diagrama BPMN del proceso optimizado.....	56
Figura 16. Modelo entidad-relación del sistema helpdesk.....	75
Figura 17. Prototipo — pantalla de acceso al sistema	76
Figura 18. Prototipo — dashboard principal del sistema helpdesk.....	77
Figura 19. Prototipo — módulo de gestión de tickets.....	78

Figura 20. Prototipo — vista de detalle del ticket.....	78
Figura 21. Prototipo — módulo de reportes y KPIs analíticos	79
Figura 22. Prototipo — gestor de usuarios	80
Figura 23. Pantalla de acceso al sistema	82
Figura 24. Pantalla de recuperación de contraseña del sistema Helpdesk	85
Figura 25. Módulo de gestión de usuarios — vista ADMIN	88
Figura 26. Módulo de gestión de roles — asignación de rol	92
Figura 27. Módulo de gestión de categorías y prioridades — Vista ADMIN	94
Figura 28. Módulo de gestión de departamentos	96
Figura 29. Módulo historia de usuario	97
Figura 30. Botón de cierre de sesión — menú principal del sistema.....	99
Figura 31. Formulario de creación de ticket — vista usuario.....	101
Figura 32. Panel de gestión de tickets — vista ADMIN.....	103
Figura 33. Modal de asignación de ticket a técnico	104
Figura 34. Módulo AnyDesk — listado de códigos por departamento	108
Figura 35. Ejemplo de notificación automática enviada al usuario — cierre de ticket	109
Figura 36. Módulo de reportes KPIs — dashboard analítico del sistema Helpdesk	110
Figura 37. Formulario de calificación del servicio — cierre de ticket	112
Figura 38. Configuración del servidor físico institucional del GAD Municipal de Pelileo.....	116

Figura 39. Configuración de conexión con Oracle Database 19c.....	117
Figura 40. Sistema Helpdesk en producción — Red institucional del GAD Municipal de Pelileo	118
Figura 41. Sistema Helpdesk en producción.....	119

ÍNDICE DE ANEXOS

Anexo A. Manual de usuario	127
----------------------------------	-----

RESUMEN EJECUTIVO

El presente trabajo trata sobre un sistema web HelpDesk usando herramientas Open Source para la gestión de incidencias de soporte técnico en el Gobierno Autónomo Descentralizado Municipal de Pelileo. El diagnóstico inicial, realizado mediante encuestas, entrevistas y observación directa a una población censal de 75 funcionarios, identifica dos cuellos de botella principales: la ausencia de un registro formal de incidencias y la inexistencia de control de tiempos de atención y acuerdos de nivel de servicio (SLA), gestionados a través de canales informales como llamadas telefónicas y mensajes de WhatsApp. La solución se alinea con las buenas prácticas de gestión de servicios, centralizando la atención mediante un Punto Único de Contacto (SPOC) digital que permitió registrar, categorizar, priorizar y resolver incidencias de forma estructurada, incorporando módulos de gestión de tickets, cálculo automático de SLA, base de conocimiento institucional y reportes analíticos en tiempo real. El stack tecnológico empleado incluyó Next.js 14, Node.js con Express.js, NextAuth.js con JWT, Tailwind CSS y Oracle Database 19c seleccionadas en base a los requerimientos empresariales. El desarrollo sigue la metodología de Programación Extrema (XP), organizado en 5 iteraciones con 18 historias de usuario, entregando el sistema con pruebas de aceptación aprobadas. El uso exclusivo de herramientas Open Source elimina costos de licenciamiento y aprovechó la infraestructura existente, garantizando la sostenibilidad del proyecto.

Palabras clave: Gestión de incidencias, HelpDesk, Open Source, GAD Municipal de Pelileo, sistema web, Next.js, Programación Extrema (XP).

ABSTRACT

This thesis presents the development of a web-based HelpDesk system using Open Source tools for the management of technical support incidents at the Gobierno Autónomo Descentralizado Municipal de Pelileo. The initial diagnosis, conducted through surveys, interviews, and direct observation on a census population of 75 municipal employees, identified two major bottlenecks: the lack of a formal incident registration system and the absence of control over response times and Service Level Agreements (SLA), which were managed through informal channels such as phone calls and WhatsApp messages. The proposed solution was aligned with IT service management best practices, implementing a digital Single Point of Contact (SPOC) that enables the structured registration, categorization, prioritization, and resolution of incidents. The system includes key modules such as ticket management, automatic SLA calculation, institutional knowledge base, and real-time analytical reports. The technological stack consisted of Next.js 14, Node.js with Express.js, NextAuth.js with JWT, Tailwind CSS, and Oracle Database 19c, selected according to the institution's requirements. The development followed the Extreme Programming (XP) methodology, organized in 5 iterations with 18 user stories, achieving user acceptance testing approval.

Keywords: HelpDesk, incident management, ITIL, IT service management, municipal government, web system, Next.js, Extreme Programming (XP).

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

HELPDESK USANDO HERRAMIENTAS OPEN SOURCE PARA LA GESTIÓN DE INCIDENCIAS DE SOPORTE TÉCNICO EN EL GAD MUNICIPAL DE PELILEO

1.1.1 Planteamiento del problema

A nivel global, la gestión eficiente de los servicios de Tecnologías de la Información (TI) se ha convertido en un pilar fundamental para la operatividad de las instituciones públicas. Sin embargo, el alto costo de licenciamiento de software propietario representa una barrera significativa. Según el informe de The State of Enterprise Open Source 2022 de Red Hat, el 82% de las organizaciones empresariales y gubernamentales optan por soluciones de código abierto (Open Source) para reducir costos y evitar la dependencia de proveedores específicos, permitiendo modernizar sus infraestructuras de soporte sin comprometer el presupuesto [1]. La falta de herramientas adecuadas de gestión de servicios de TI (ITSM) provoca que los departamentos de soporte pierdan hasta un 30% de su tiempo resolviendo incidentes recurrentes que podrían automatizarse o documentarse [2].

En el contexto latinoamericano y ecuatoriano, la transformación digital en el sector público enfrenta desafíos de organización y presupuesto. En Ecuador, la Ley Orgánica para la Transformación Digital y Audiovisual, así como las directrices del Ministerio de Telecomunicaciones y de la Sociedad de la Información (MINTEL), instan a los Gobiernos Autónomos Descentralizados (GAD) a implementar soluciones tecnológicas eficientes priorizando la soberanía tecnológica. No obstante, un reporte del Banco Interamericano de Desarrollo (BID) señala que la madurez digital en los municipios de la región aún es baja, donde muchos procesos de atención interna siguen siendo manuales o informales, lo que afecta directamente la calidad del servicio al ciudadano [3]. El uso de herramientas Open Source se presenta como una directriz

estratégica en Ecuador para cumplir con la eficiencia administrativa sin incurrir en gastos recurrentes de licencias [4].

A nivel local, el GAD Municipal de Pelileo enfrenta una problemática operativa en su departamento de tecnologías. Actualmente, la gestión de incidencias de soporte técnico carece de una plataforma centralizada y automatizada (Helpdesk). El reporte de requerimientos se realiza frecuentemente a través de canales informales como llamadas telefónicas, mensajes de WhatsApp o visitas presenciales a la oficina de TI. Esta informalidad conlleva a la pérdida de trazabilidad de los incidentes, la ausencia de una base de conocimiento para problemas recurrentes y métricas clave de desempeño (KPIs, del inglés Key Performance Indicators) como el tiempo de respuesta. Esta situación genera insatisfacción en los funcionarios municipales, retrasos en los procesos administrativos y una carga laboral desorganizada para el personal técnico, evidenciando la necesidad crítica de implementar una solución de gestión basada en herramientas Open Source.

1.2 Antecedentes investigativos

Para el desarrollo de la presente investigación, se realizó una revisión sistemática de la literatura en bases de datos científicas de alto impacto y repositorios académicos regionales (Google Scholar, IEEE Xplore y Red de Repositorios de Acceso Abierto). La búsqueda se centró en estudios publicados entre los años 2021 y 2024 que abordaran la implementación de sistemas de mesa de ayuda (Helpdesk) en entornos gubernamentales y corporativos, utilizando criterios de inclusión como: "gestión de incidencias", "soporte técnico municipal" y "herramientas Open Source". A continuación, se detallan los hallazgos más relevantes que sustentan la viabilidad técnica y operativa de este proyecto.

Tabla 1. Cadenas de búsqueda

Base de datos	Criterios de búsqueda	Número de artículos
IEEE Xplore	"Help desk" OR "Service Desk") AND ("Open Source" OR "Free Software") AND "Management"	45
Google Scholar	"Gestión de incidencias" AND ("Municipalidad" OR "Gobierno Local")	120
La Referencia	"Mesa de ayuda" AND "Software Libre" AND "Soporte Técnico"	38
Total		157

Tabla 2. Criterios de inclusión y exclusión

Criterios de inclusión	Criterios de exclusión
Proyectos en el dominio de la gobernanza pública (municipios, GAD o administración territorial).	Archivos repetidos o con enfoques metodológicos ambiguos (ensayos, opiniones o revisiones no basadas en datos).
Trabajos en entornos regionales o con restricciones tecnológicas, idealmente en Latinoamérica o Ecuador.	Investigaciones no ligadas directamente a la administración municipal o a la productividad digital.
Estudios que examinen la productividad operativa, la estandarización de procesos o la implementación de soluciones tecnológicas orientadas a la gestión de incidencias y soporte técnico en instituciones públicas.	Documentos previos a 2022 o sin texto íntegro disponible.

En el contexto internacional, Osman y Jimu [8] desarrollaron un sistema de soporte Help Desk para un Consejo Municipal (City Council), con el objetivo de gestionar las quejas y solicitudes de servicios de los ciudadanos y funcionarios. Su estudio destaca que la implementación de una plataforma web centralizada, basada en metodologías ágiles, logró automatizar los procesos administrativos que anteriormente se realizaban de forma manual. Los resultados obtenidos demostraron una reducción significativa en los tiempos de respuesta y una mejora en la eficiencia operativa del municipio al contar con una base de datos histórica de los requerimientos, lo que valida la importancia de digitalizar el soporte en entidades públicas.

En el ámbito regional latinoamericano, Pérez Romero y Mendoza Callupe [9] evaluaron el impacto de un sistema web de Help Desk en la gestión de incidencias tecnológicas. Su investigación, de enfoque cuantitativo y diseño experimental, arrojó datos contundentes sobre la eficacia de estas herramientas: la tasa de resolución de problemas se incrementó del 37,2% al 85,9% tras la implementación del sistema. Asimismo, lograron reducir la reincidencia de fallos técnicos del 38,9% al 14,7%. Este antecedente es crucial para el presente proyecto, pues proporciona evidencia estadística de cómo una herramienta de gestión bien estructurada puede duplicar la capacidad de resolución de un departamento de TI.

Específicamente sobre la aplicación de buenas prácticas en gobiernos locales, Sánchez Casanova y Valles Coral [10] analizaron la influencia del marco de trabajo en la gestión de incidencias de una municipalidad peruana. El estudio concluyó que la adopción de procesos estandarizados, apoyados por software de gestión, permitió disminuir las incidencias de TI en un 20% y mejorar el cumplimiento de los tiempos de atención establecidos en los Acuerdos de Nivel de Servicio (SLA) en un 84%. Este trabajo sirve como referente directo para el GAD Municipal de Pelileo, demostrando que la combinación de software Open Source con es una estrategia exitosa para la administración pública.

Finalmente, en el contexto nacional, Faria [11] en su investigación de posgrado para la Escuela Superior Politécnica del Litoral (ESPOL), abordó la implementación de una mesa de ayuda para un Centro de Operaciones de Seguridad (SOC) que prestaba servicios a entidades públicas y privadas. El autor identificó que la falta de un sistema formal generaba pérdida de trazabilidad en los requerimientos. La solución implementada permitió no solo organizar el flujo de trabajo, sino también generar métricas de rendimiento para la toma de decisiones. Este antecedente local respalda la necesidad de abandonar los canales informales (como WhatsApp o correo directo) en favor de plataformas auditables que garanticen la calidad del servicio técnico. En el caso específico del GAD Municipal de Pelileo, esta necesidad se acentúa por la existencia de múltiples departamentos y unidades operativas distribuidas, tales como el Departamento de Servicios Públicos, el Mercado República de Argentina (centro de abasto con locales comerciales y actividades diarias que requieren soporte constante) y el Camal Municipal (instalación dedicada al faenamiento de animales de abasto, con

procesos regulados y necesidades técnicas específicas). Estas dependencias, al formar parte integral de la estructura organizacional del municipio, generan un volumen diversificado de incidencias y solicitudes de soporte (desde mantenimiento de infraestructura hasta resolución de problemas tecnológicos o administrativos), lo que justifica aún más la implementación de un sistema de Help Desk centralizado y estandarizado para mejorar la trazabilidad, eficiencia y calidad del servicio en toda la entidad pública.

1.3 Fundamentación teórica

1.3.1 Gestión de servicios de tecnologías de la información

La Gestión de Servicios de TI (ITSM, por sus siglas en inglés) se define como un conjunto de capacidades organizativas especializadas para aportar valor a los clientes en forma de servicios. A diferencia de la gestión tecnológica tradicional centrada en el hardware, ITSM se enfoca en la entrega de servicios que satisfagan las necesidades del negocio [12].

El marco de referencia más utilizado a nivel mundial para ITSM es ITIL (Information Technology Infrastructure Library). En su versión más reciente, evoluciona el concepto de "procesos" hacia "prácticas de gestión", enfatizando la co-creación de valor. ITIL define el (SVS) Sistema de Valor del Servicio como un modelo operativo que asegura que la demanda de servicios se convierta efectivamente en valor para la organización, mediante la integración de flujos de trabajo ágiles y mejora continua [13]. Para el GAD Municipal, adoptar estas prácticas implica dejar de ver al departamento de sistemas como un centro de costos y gestionarlo como un proveedor de servicios internos críticos para la administración pública.

1.3.2 Gestión de incidencias

Dentro de las prácticas, la Gestión de Incidencias es la más visible y crítica para la operación diaria. Una incidencia se define como cualquier interrupción no planificada de un servicio de TI o una reducción en la calidad de este. El objetivo principal de esta

práctica es restaurar la operación normal del servicio lo más rápido posible para minimizar el impacto negativo en las operaciones del negocio [14].

El ciclo de vida de una incidencia, que debe ser soportado por la herramienta Helpdesk, sigue un flujo estandarizado [15]:

- **Identificación y Registro:** Detección del fallo reportado por el usuario o herramientas de monitoreo.
- **Categorización:** Clasificación del incidente para facilitar su asignación y análisis posterior.
- **Priorización:** Asignación de un orden de atención basado en el Impacto (número de usuarios afectados) y la Urgencia (tiempo crítico para la resolución).
- **Diagnóstico y Resolución:** Investigación técnica y aplicación de la solución.
- **Cierre:** Confirmación de satisfacción del usuario y documentación en la base de conocimientos.

1.3.3 Mesa de ayuda (help desk) vs. mesa de servicio

Es fundamental distinguir el alcance de la solución propuesta. Según la literatura especializada, un Help Desk (Mesa de Ayuda) es una unidad funcional enfocada tácticamente en la resolución rápida de incidencias y requerimientos inmediatos ("break-fix"). Actúa como el Punto Único de Contacto (SPOC) entre los usuarios y el departamento de TI, garantizando que ninguna solicitud se pierda en canales informales [16].

A diferencia de un Service Desk, que posee un alcance estratégico más amplio (incluyendo gestión de cambios, activos y niveles de servicio complejos), el Help Desk es la solución ideal para entidades que inician su formalización de procesos, como es

el caso de municipios medianos, pues permite centralizar el soporte con una implementación menos costosa y compleja [17].

1.3.4 Software open source en la administración pública

El Software de Código Abierto (Open Source) se refiere al software cuyo código fuente está disponible públicamente bajo una licencia que permite a los usuarios estudiar, cambiar y distribuir el software a cualquier persona y para cualquier propósito. Este modelo de desarrollo fomenta la colaboración descentralizada y la transparencia tecnológica [18].

En el contexto ecuatoriano, el uso de estas herramientas en el sector público no es solo una opción técnica, sino una directriz estratégica. El Estado ecuatoriano promueve la Soberanía Tecnológica y la eficiencia en el gasto público, priorizando el uso de software libre para evitar la dependencia de proveedores extranjeros (vendor lock-in) y eliminar los costos recurrentes de licenciamiento. Investigaciones recientes señalan que la adopción de herramientas Open Source en gobiernos locales de Latinoamérica permite redirigir el presupuesto ahorrado hacia la mejora de infraestructura crítica, garantizando además la seguridad y auditabilidad de los datos ciudadanos [19].

1.3.5 Stack tecnológico seleccionado

La selección de las herramientas tecnológicas para el desarrollo del sistema Helpdesk se fundamentó en criterios de código abierto, compatibilidad, rendimiento, escalabilidad y alineación con las capacidades de la infraestructura existente en el GAD Municipal de Pelileo. A continuación, se describe el stack tecnológico adoptado, organizado por capas de arquitectura.

Backend: Node.js y Express.js

Node.js es un entorno de ejecución de JavaScript del lado del servidor basado en el motor V8 de Google Chrome. Su arquitectura asíncrona y orientada a eventos lo hace especialmente eficiente para aplicaciones con múltiples solicitudes concurrentes, como los sistemas de gestión de tickets. Express.js es el framework web minimalista

más utilizado sobre Node.js, que facilita la definición de rutas, middlewares y controladores de forma estructurada [20].

Base de datos: Oracle Database 19c

Oracle Database 19c es un sistema gestor de bases de datos relacional de nivel empresarial, ampliamente utilizado en instituciones públicas del Ecuador por su robustez, soporte a transacciones ACID, y capacidades avanzadas de seguridad y auditoría. Su elección para este proyecto responde al hecho de que el GAD Municipal de Pelileo ya cuenta con licenciamiento institucional de Oracle, lo que elimina costos adicionales de adquisición y garantiza compatibilidad con los sistemas existentes [21].

Para el sistema Helpdesk se diseñó un esquema de base de datos específico que incluye, entre otras, las siguientes entidades principales: tabla de tickets, tabla de usuarios, tabla de categorías, tabla de comentarios, tabla de artículos de base de conocimiento, y la nueva tabla

Frontend: Next.js 14 con App Router y Tailwind CSS

Next.js es un framework de React desarrollado por Vercel que incorpora renderizado del lado del servidor (SSR) y generación estática (SSG), optimizando el rendimiento y la experiencia del usuario. La versión 14 introduce el App Router, una arquitectura basada en componentes de servidor que mejora significativamente los tiempos de carga y la organización del código [19].

Tailwind CSS es un framework de utilidades CSS que permite construir interfaces modernas y responsivas directamente en el marcado HTML, sin necesidad de escribir hojas de estilo personalizadas. Su integración con Next.js garantiza un tamaño de bundle optimizado al eliminar las clases no utilizadas en producción.

Autenticación y Control de Acceso: NextAuth.js y JWT

NextAuth.js es la solución estándar de autenticación para aplicaciones Next.js, compatible con múltiples proveedores de identidad y estrategias de sesión. Para este proyecto se implementa la estrategia de credenciales personalizadas con tokens JWT

(JSON Web Token), alineada con la arquitectura REST (Representational State Transfer) del backend [22].

Módulo AnyDesk — gestión de acceso remoto institucional

AnyDesk es una herramienta de escritorio remoto de alto rendimiento, ampliamente utilizada por equipos de soporte técnico para asistencia a distancia. El GAD Municipal de Pelileo utiliza AnyDesk como solución de acceso remoto a los equipos de sus diferentes dependencias.

El módulo permite a cualquier usuario autenticado consultar y copiar el código AnyDesk de cualquier departamento con un solo clic (función de copiado al portapapeles), agilizando la asistencia remota. La edición de los códigos está restringida exclusivamente al rol ADMIN, garantizando la integridad y trazabilidad de los datos de acceso.

1.3.6 Resumen de la arquitectura del sistema

El sistema Helpdesk para el GAD Municipal de Pelileo se estructura bajo una arquitectura de tres capas desacopladas: el frontend desarrollado en Next.js 14 mediante una API (Application Programming Interface) REST protegida por JWT, y este a su vez interactúa con la base de datos Oracle 19c para la persistencia de datos [19] [20] [21]. Esta arquitectura garantiza escalabilidad, mantenibilidad y compatibilidad con la infraestructura tecnológica existente en la institución.

1.4 Objetivos

1.4.1 Objetivo general

Implantar un sistema web helpdesk usando herramientas open source para la gestión de incidencias de soporte técnico en el GAD municipal de Pelileo.

1.4.2 Objetivos específicos

- Analizar los procesos involucrados en la identificación, registro, diagnóstico y resolución de interrupciones o problemas no planificados en el soporte técnico en el GAD municipal de Pelileo.
- Investigar las diferentes herramientas Open source adecuadas para el desarrollo de plataformas web.
- Desarrollar un sistema web helpdesk usando herramientas open source para la gestión de incidencias de soporte técnico en el GAD municipal de Pelileo.

CAPÍTULO II.- METODOLOGÍA

2.1 Materiales

Los materiales aplicados en esta investigación para la recolección de la información fueron una encuesta dirigida al personal del GAD municipal de Pelileo, una entrevista al personal del Departamento de TI y una ficha de observación con el fin de identificar los procesos.

Los objetivos principales de estos instrumentos fueron evaluar la eficiencia y calidad del soporte técnico, con el fin de analizar la factibilidad de implementar un sistema que permita optimizar la gestión de incidencias y elevar la calidad de la atención brindada al personal de la institución pública.

a. Encuesta dirigida al personal de la institución

La encuesta se aplica a todo el personal que labora en el GAD de Pelileo y sus respectivos departamentos, con el propósito de recabar datos sobre la percepción del equipo de Soporte Técnico del Departamento de Tecnologías de la Información (TI).

El cuestionario posee nueve preguntas las cuales se detallan a continuación:

Tabla 3. Encuesta al personal administrativo

Encuesta al personal	
Area / Departamento: _____	
Objetivo: Recopilar información sobre el proceso actual de reporte y atención de incidencias técnicas en el GAD Municipal de Pelileo, con el fin de identificar necesidades y validar la implementación de un sistema Helpdesk Open Source.	
Instrucción: Marque con una X la opción que mejor describa su situación. La encuesta es anónima y sus respuestas serán utilizadas únicamente con fines académicos.	
Pregunta 1. ¿Cuándo se presentan problemas informáticos, tiene algún canal o sistema disponible para ingresar y reportar incidencias?	
[]	Sí

☐	No
Pregunta 2. Durante el horario de oficina, ¿cuál es el medio que utiliza para comunicarse con el área de soporte técnico cuando se presenta un problema?	
☐	Llamada telefónica
☐	Correo electrónico
☐	Mensaje de WhatsApp / SMS
☐	Visita presencial al área de TI
☐	Otro: _____
Pregunta 3. ¿Qué tan complicado le resulta comunicar sus incidencias al área de soporte técnico?	
☐	Muy fácil
☐	Fácil
☐	Difícil
☐	Muy difícil
Pregunta 4. ¿Con qué frecuencia el departamento de TI atiende sus solicitudes relacionadas con los problemas reportados?	
☐	Muy frecuentemente
☐	Frecuentemente
☐	A veces
☐	Rara vez
Pregunta 5. ¿Cómo evalúa usted la efectividad del área de soporte técnico para solucionar los problemas que reporta?	
☐	Muy efectivo
☐	Efectivo
☐	Poco efectivo
☐	Inefectivo
Pregunta 6. Cuando usted genera un reporte de problema técnico, ¿cuánto tiempo suele tardar en ser atendido su requerimiento?	
☐	De 1 a 3 horas
☐	De 3 a 5 horas

☐	De 5 a 8 horas
☐	Más de 8 horas
Pregunta 7. Cuando el técnico soluciona una falla, ¿le proporciona alguna explicación sobre la causa del problema para evitar que vuelva a ocurrir?	
☐	Sí, siempre
☐	A veces
☐	Casi nunca
☐	No, nunca
Pregunta 8. Una vez generado el reporte de un problema, ¿con qué frecuencia recibe actualizaciones sobre el progreso de la incidencia por parte del personal de soporte técnico?	
☐	Muy frecuentemente
☐	Frecuentemente
☐	Poco frecuente
☐	Nunca
Pregunta 9. ¿Estaría a favor de la implementación de un sistema web que le permita reportar, dar seguimiento y consultar el estado de los problemas informáticos que experimente durante su jornada laboral?	
☐	Sí, definitivamente
☐	Probablemente sí
☐	Probablemente no
☐	No, prefiero el método actual

b. Entrevista al personal del departamento de TI

Se aplicó una entrevista al personal del Departamento de TI, utilizando el instrumento de recolección de información diseñado, con el propósito de obtener detalles sobre los procesos involucrados en la gestión de incidencias y conocer los aspectos técnicos relevantes.

La entrevista se compone de diez preguntas, las cuales se detallan a continuación:

1. ¿Qué canales utilizan los funcionarios para solicitar ayuda?
2. ¿Con qué frecuencia aproximada reciben solicitudes de soporte técnico durante una semana de trabajo?
3. ¿Cómo llevan registro de estas solicitudes actualmente?
4. ¿Cuánto tiempo toma en promedio la resolución de una incidencia desde que es reportada hasta que se soluciona?
5. ¿Qué problemas o dificultades ha identificado en el proceso actual de gestión de incidencias?
6. ¿Ha ocurrido que alguna solicitud de soporte quede sin atención o se pierda el registro de la misma?
7. ¿Existe actualmente alguna forma de medir o evaluar el desempeño del equipo de TI?
8. ¿Cómo afecta la situación actual a la carga laboral del personal de TI y a la productividad de los funcionarios administrativos?
9. ¿Cuáles considera que son las funcionalidades más importantes que debería tener un sistema de Helpdesk para satisfacer las necesidades del departamento?
10. ¿Qué aspectos del proceso actual considera que deben mantenerse o adaptarse al momento de diseñar el nuevo sistema?

c. Ficha de observación (asistentes técnicos)

El propósito de la ficha de observación, presentada en la Tabla 4, fue identificar los procesos involucrados en la gestión de asistencias técnicas brindadas al personal de los diferentes departamentos del GAD Municipal de Pelileo.

Tabla 4. Ficha de observación para el personal técnico de TI

Ficha de observación				
Fecha y hora de la observación: _____ N° de ficha: _____				
Técnico observado: _____ Cargo: _____				
Objetivo: Determinar los procesos involucrados en la gestión de incidencias de soporte técnico en el GAD Municipal de Pelileo.				
Indicadores	Observación			Novedades
Recepción de la solicitud				
Canal de recepción de la solicitud (llamada telefónica, mensaje de WhatsApp, visita presencial, correo electrónico).				
Se registra formalmente la solicitud al momento de recibirla (numero, descripción, solicitante).				
Tiempo entre recepción e inicio del proceso:	Inmediata ☐	15 min ☐	30 min ☐	Más de 30 min ☐
Identificación y diagnóstico del problema				
Acciones realizadas por el técnico para identificar el tipo y origen de la incidencia.				
Se consulta algún historial previo o base de conocimiento sobre incidencias similares.				
Resolución del problema				
Se documenta el procedimiento o solución aplicada durante o después de la atención.				
Verificación final con el funcionario para confirmar que el problema fue resuelto satisfactoriamente.				
Seguimiento y cierre				
Se ofrece alguna indicación adicional al funcionario para prevenir futuros problemas.				

Se registra el cierre de la incidencia en algún sistema, bitácora o documento físico.		
Tiempo total del proceso de atención: _____ minutos		
Interacción con el funcionario		
Canales utilizados para comunicarse (presencial, llamada, WhatsApp, correo, otro).		
El técnico mantiene informado al funcionario sobre el avance y estado de la solución.		
Notas adicionales		
Problemas críticos observados: Procesos sin formalizar: Observaciones generales del investigador: 		

2.2 Métodos

2.2.1 Modalidad de la investigación

El presente proyecto de investigación se enmarca en la modalidad de investigación de campo y aplicada.

Investigación de campo

Se trató de una investigación de campo, ya que el investigador estudió directamente la problemática en su contexto real y lugar de origen: el GAD Municipal de Pelileo. Mediante la aplicación de entrevistas al personal del Departamento de TI, encuestas dirigidas al personal administrativo de los diferentes departamentos y fichas de observación de los procesos técnicos, se logró recopilar información de primera mano. Estas técnicas permitieron identificar las causas principales, las consecuencias derivadas de las deficiencias en la gestión de incidencias técnicas y evaluar la eficiencia y calidad del servicio de soporte técnico actual [23].

Investigación aplicada

Esta investigación tiene un carácter aplicado, orientado hacia la resolución de un problema específico detectado en el GAD Municipal de Pelileo. Lejos de perseguir únicamente fines teóricos, el estudio se centra en evaluar la situación actual del servicio de soporte técnico y la gestión de incidencias, con el fin de generar propuestas concretas y accionables. Los hallazgos derivados de las encuestas, entrevistas y fichas de observación servirán de base para determinar la viabilidad de implementar un sistema de gestión de incidencias (Helpdesk) que permita optimizar los procesos, reducir tiempos de respuesta y mejorar la satisfacción del personal usuario del servicio.

2.2.2 Población y muestra

Población

La población objetivo de este estudio está conformada por el personal del GAD Municipal de Pelileo que interactúa de manera directa con los servicios de soporte técnico del Departamento de Tecnologías de la Información. Esta población incluye tanto a los usuarios finales (personal administrativo y operativo cuyos equipos informáticos son esenciales para el desarrollo de sus funciones) como a los proveedores del servicio (personal técnico del Departamento de TI).

Específicamente, se consideró al personal que realiza funciones administrativas o que, en caso de presentarse fallas en los equipos, ve directamente afectado el cumplimiento de sus responsabilidades laborales.

Según la nómina institucional, la población total ascendió a 75 personas, distribuidas de la siguiente forma:

Tabla 5. Población y Muestra

Población	Número	Porcentaje
Departamento de TI	5	6,7%
Departamento de Talento Humano	6	8,0%
Secretaría General	4	5,3%
Asesoría Jurídica	5	6,7%
Departamento de Dirección Financiera	7	9,3%
Departamento de Obras Públicas	8	10,7%
Departamento de Planificación Territorial	6	8,0%
Departamento de Avalúos y Catastros	5	6,7%
Departamento de Gestión Ambiental	4	5,3%
Departamento de Comisaría Municipal	6	8,0%
Departamento de Desarrollo Social	6	8,0%
Camal Municipal de Pelileo	5	6,7%
Mercado República de Argentina	8	10,7%
TOTAL	75	100%

Dado que la población total ascendió únicamente a 75 personas, no se aplicaron técnicas de muestreo. En su lugar, se trabajó con la totalidad de la población a través de un censo, garantizando de esta manera la máxima representatividad y confiabilidad de los datos recolectados [23].

2.2.3 Recolección de información

La recolección de información se realizó mediante tres instrumentos principales: una encuesta dirigida al personal administrativo de los diferentes departamentos del GAD Municipal de Pelileo, una entrevista presencial aplicada al personal del Departamento de TI (jefe de sistemas y técnicos de soporte), y una ficha de observación orientada a identificar y documentar los procesos actuales de gestión del soporte técnico. Estos instrumentos permitieron obtener datos tanto cuantitativos como cualitativos sobre la percepción del servicio, los flujos de trabajo vigentes y las posibles áreas de mejora.

a. Validación de la encuesta dirigida al personal de la institución

Para evaluar la confiabilidad y consistencia interna del instrumento de encuesta, se calculó el coeficiente Alfa de Cronbach. Esta medida estadística determina qué tan bien los ítems de una escala miden un mismo constructo. En este caso, se aplicó a las 8 preguntas con escala ordinal (preguntas 2 a 9), excluyendo la pregunta 1 de tipo dicotómico (Sí/No).

El cálculo se realizó con apoyo de la herramienta Microsoft Excel, utilizando la siguiente fórmula:

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right]$$

Donde:

- α : coeficiente de Alfa de Cronbach
- K : número de ítems del instrumento
- $\sum S_i^2$: sumatoria de las varianzas de los ítems
- S_T^2 : varianza total del instrumento

Interpretación de resultados – Alfa de Cronbach

Tabla 6. Niveles de confiabilidad según el coeficiente Alfa de Cronbach

Rango de Alfa de Cronbach	Nivel de confiabilidad
Mayor a 0,9	Excelente
De 0,8 a 0,9	Bueno
De 0,7 a 0,8	Aceptable
De 0,6 a 0,7	Cuestionable
De 0,5 a 0,6	Pobre
Menor a 0,5	Inaceptable

Coeficiente de confiabilidad

Una vez identificadas las respuestas medibles de la encuesta, se procedió a construir una matriz en Microsoft Excel con las respuestas de 20 participantes seleccionados de la población total de 75 personas del GAD Municipal de Pelileo, con el propósito de calcular los valores necesarios para aplicar la fórmula del Alfa de Cronbach y determinar la confiabilidad del instrumento antes de su aplicación definitiva.

Tabla 7. Calculo Alfa de Cronbach (α)

Sujeto	Í1	Í2	Í3	Í4	Í5	Í6	Í7	Í8	Suma
Sujeto 1	3	4	4	3	4	3	4	3	28
Sujeto 2	1	3	2	2	2	4	4	2	20
Sujeto 3	1	3	3	3	4	2	2	3	21
Sujeto 4	1	4	3	3	4	3	4	3	25
Sujeto 5	4	3	4	3	3	4	3	3	27
Sujeto 6	3	3	3	3	2	1	3	4	22
Sujeto 7	3	2	3	3	2	2	3	3	21
Sujeto 8	3	2	3	2	2	1	2	3	18
Sujeto 9	1	3	1	3	2	3	4	2	19
Sujeto 10	1	2	2	2	2	1	3	3	16
Sujeto 11	2	3	4	3	3	2	4	4	25
Sujeto 12	1	2	3	3	2	3	1	2	17
Sujeto 13	1	2	2	2	2	2	3	2	16
Sujeto 14	1	3	2	3	2	1	3	3	18
Sujeto 15	1	3	3	3	2	2	3	3	20
Sujeto 16	3	4	4	3	4	3	4	4	29
Sujeto 17	2	3	2	2	3	3	3	3	21
Sujeto 18	1	4	4	4	4	4	4	4	29
Sujeto 19	1	2	4	3	3	1	3	1	18
Sujeto 20	2	2	3	3	2	2	3	3	20
Varianza	0.73	0.53	0.75	0.26	0.69	0.73	0.41	0.75	15.19

FÓRMULA

$$\alpha = \frac{K}{K-1} \left[1 - \frac{\sum S_i^2}{S_T^2} \right]$$

RESULTADO

K (Núm. ítems)	8.00
Σi^2 (Varianza ítem)	4.84
St ² (Varianza total)	15.19
α (Alfa)	0.78

CONFIABILIDAD ACEPTABLE

Rango: 0,7 – 0,8

Resultados del cálculo

Los valores previamente calculados fueron incorporados en la fórmula del Alfa de Cronbach.

A continuación, se presentan los valores calculados y su sustitución en la fórmula:

$$\alpha = \frac{8}{8-1} \left[1 - \frac{4,84}{15,19} \right] = 1,1429 \times 0,6814 = 0,78$$

Tabla 8. Valores del coeficiente Alfa de Cronbach

Ítem	Valor
K (Número de ítems)	8,00
$\sum S_i^2$ (Sumatoria de varianzas de los ítems)	4,84
S_T^2 (Varianza total del instrumento)	15,19
α (Alfa de Cronbach)	0,78

El coeficiente Alfa de Cronbach obtenido fue de 0,78 como se muestra en la Tabla 8, valor que se ubica en el rango de 0,7 a 0,8. Por tanto, el instrumento presenta un nivel de confiabilidad aceptable. Este resultado indica que los ítems del cuestionario miden de manera consistente la percepción del personal administrativo respecto a la calidad, eficiencia y procesos del servicio de soporte técnico en el GAD Municipal de Pelileo, validando así su uso en la presente investigación.

Resultados de la encuesta

Pregunta 1. ¿Cuándo se presentan problemas informáticos, tiene algún canal o sistema disponible para ingresar y reportar incidencias?

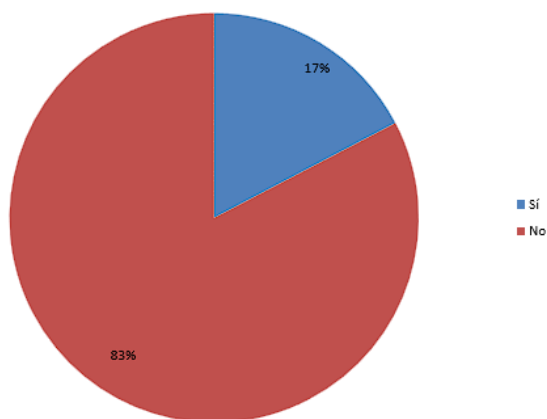


Figura 1. Canal disponible para reporte de incidencias

Análisis e interpretación de resultados

La Figura 1 evidencia que el 83% de los funcionarios del GAD Municipal de Pelileo no dispone de ningún canal formal o sistema habilitado para reportar incidencias técnicas, mientras que únicamente el 17% indica que sí cuenta con algún medio. Estos datos confirman la ausencia de una plataforma centralizada de soporte y validan la necesidad urgente de implementar un sistema Helpdesk que formalice el proceso de reporte.

Pregunta 2. Durante el horario de oficina, ¿cuál es el medio que utiliza para comunicarse con el área de soporte técnico cuando se presenta un problema?

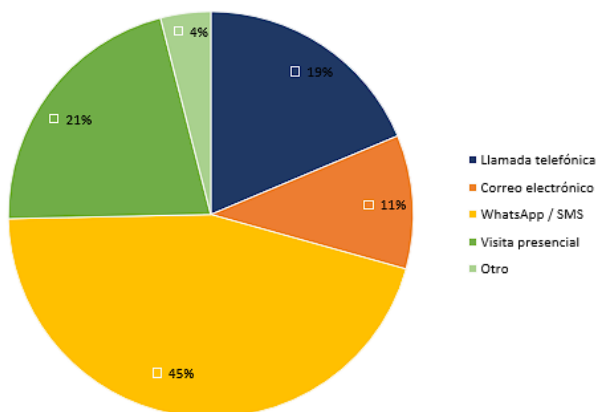


Figura 2. Medios de comunicación utilizados para soporte

Análisis e interpretación de resultados

Como se observa en la Figura 2, el 45% de los funcionarios utiliza mensajes de WhatsApp como principal canal de comunicación con el área de TI, seguido de visitas presenciales con un 21% y llamadas telefónicas con un 19%. Solo el 11% emplea el correo electrónico. Esta dispersión en los canales de comunicación confirma la falta de un punto único de contacto, lo que genera pérdida de trazabilidad y dificulta el seguimiento de las incidencias reportadas.

Pregunta 3. ¿Qué tan complicado le resulta comunicar sus incidencias al área de soporte técnico?

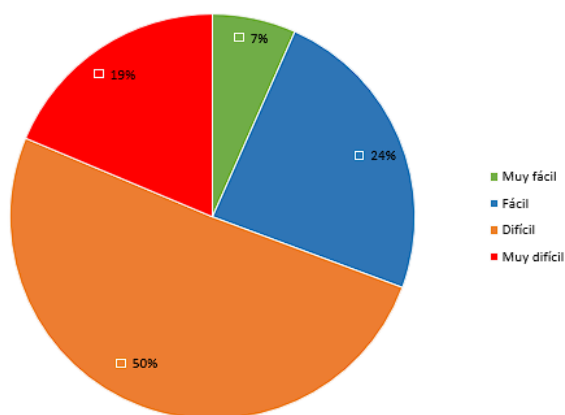


Figura 3. Dificultad para comunicar incidencias

Análisis e interpretación de resultados

La Figura 3 muestra que el 51% de los funcionarios considera difícil comunicar sus incidencias al área de soporte técnico, y un 19% adicional lo califica como muy difícil, sumando el 70% con dificultades. Solo el 31% lo considera fácil o muy fácil. Estos datos reflejan que los canales informales actuales no son accesibles ni eficientes, lo que justifica la implementación de un sistema web de gestión de incidencias con interfaz amigable.

Pregunta 4. ¿Con qué frecuencia el departamento de TI atiende sus solicitudes relacionadas con los problemas reportados?

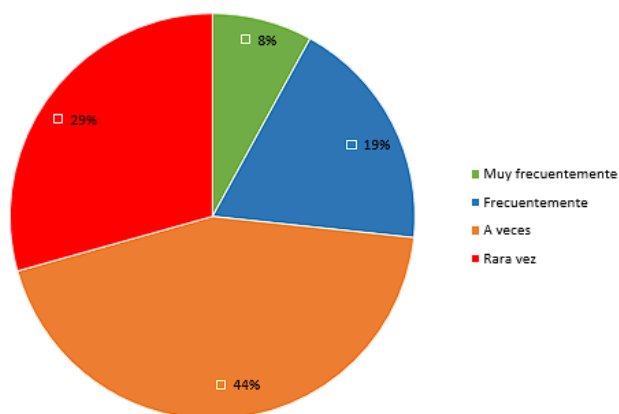


Figura 4. Frecuencia de atención del departamento de TI

Análisis e interpretación de resultados

En la Figura 4 se evidencia que el 44% de los encuestados indica que sus solicitudes son atendidas solo a veces, y el 29% afirma que rara vez reciben atención oportuna, representando juntos el 73% de insatisfacción en la frecuencia de atención. Solo el 27% considera que sus solicitudes son atendidas frecuentemente o muy frecuentemente. Este resultado refleja la sobrecarga y desorganización del departamento de TI al gestionar incidencias sin un sistema formal de ticketing.

Pregunta 5. ¿Cómo evalúa usted la efectividad del área de soporte técnico para solucionar los problemas que reporta?

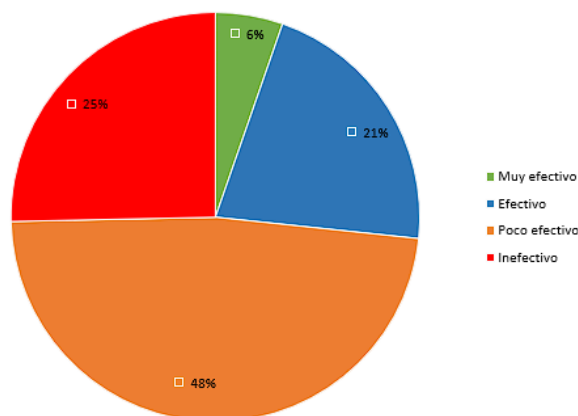


Figura 5. Evaluación de efectividad del soporte técnico

Análisis e interpretación de resultados

La Figura 5 revela que el 48% de los funcionarios califica el soporte técnico como poco efectivo y el 25% como inefectivo, acumulando el 73% de evaluaciones negativas. El 21% lo considera efectivo y apenas el 5% muy efectivo. Estos resultados evidencian deficiencias estructurales en el proceso de resolución de incidencias, asociadas directamente a la falta de seguimiento, priorización y base de conocimiento que un sistema Helpdesk permitiría subsanar.

Pregunta 6. Cuando usted genera un reporte de problema técnico, ¿cuánto tiempo suele tardar en ser atendido su requerimiento?

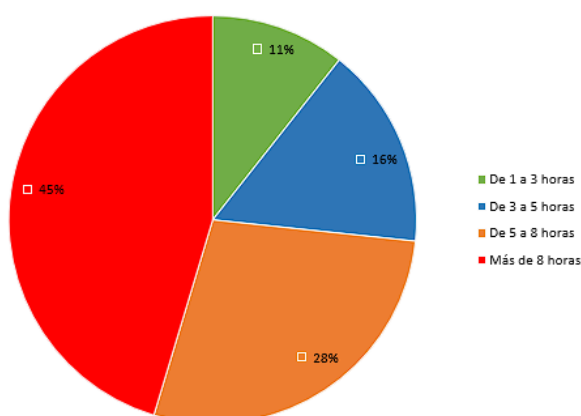


Figura 6. Tiempo de atención de requerimientos técnicos

Análisis e interpretación de resultados

Como se muestra en la Figura 6, el 45% de los funcionarios reporta que sus requerimientos tardan más de 8 horas en ser atendidos, y el 28% indica tiempos de entre 5 y 8 horas, representando el 73% con tiempos de espera superiores a 5 horas. Solo el 27% recibe atención en menos de 5 horas. Estos tiempos de respuesta excesivos afectan directamente la productividad institucional y demuestran la necesidad de implementar acuerdos de nivel de servicio (SLA) gestionados mediante un sistema Helpdesk.

Pregunta 7. Cuando el técnico soluciona una falla, ¿le proporciona alguna explicación sobre la causa del problema para evitar que vuelva a ocurrir?

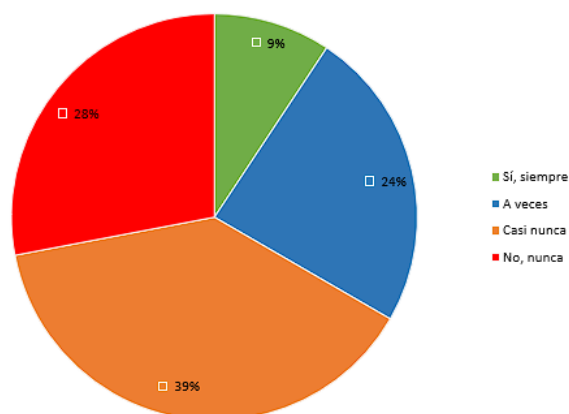


Figura 7. Retroalimentación técnica al usuario tras la resolución

Análisis e interpretación de resultados

La Figura 7 indica que el 39% de los funcionarios afirma que casi nunca recibe una explicación sobre la causa del problema resuelto, y el 28% señala que nunca la recibe, sumando el 67% sin retroalimentación técnica. Solo el 9% recibe siempre una explicación. Esta ausencia de retroalimentación impide la generación de una base de conocimiento institucional y contribuye a la reincidencia de los mismos problemas técnicos, situación que un sistema Helpdesk con módulo de base de conocimiento podría resolver.

Pregunta 8. Una vez generado el reporte de un problema, ¿con qué frecuencia recibe actualizaciones sobre el progreso de la incidencia por parte del personal de soporte técnico?

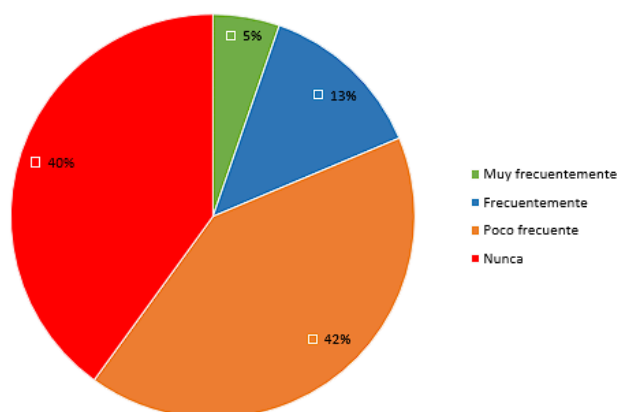


Figura 8. Frecuencia de actualizaciones sobre el progreso de incidencias

Análisis e interpretación de resultados

En la Figura 8 se observa que el 40% de los encuestados nunca recibe actualizaciones sobre el estado de su incidencia y el 41% las recibe con poca frecuencia, representando el 81% sin seguimiento efectivo. Solo el 18% recibe actualizaciones frecuentes o muy frecuentes.

Pregunta 9. ¿Estaría a favor de la implementación de un sistema web que le permita reportar, dar seguimiento y consultar el estado de los problemas informáticos que experimente durante su jornada laboral?

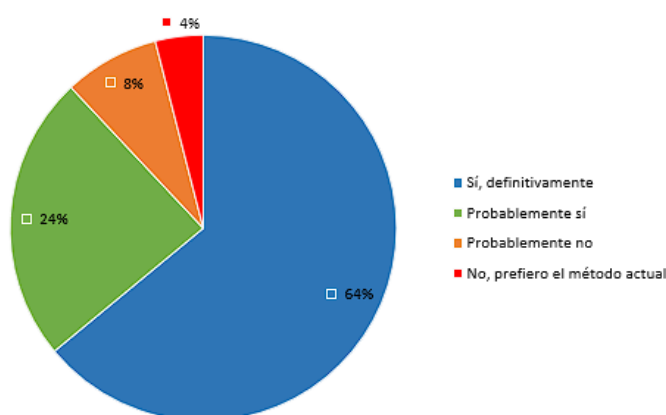


Figura 9. Aceptación de implementación del sistema Helpdesk

Análisis e interpretación de resultados

La Figura 9 demuestra que el 64% de los funcionarios está definitivamente a favor de implementar un sistema web de gestión de incidencias, y el 24% probablemente lo

estaría, sumando el 88% de aceptación positiva. Solo el 12% muestra dudas o rechazo. Este alto nivel de aceptación valida la pertinencia del proyecto y confirma que el personal administrativo del GAD Municipal de Pelileo adoptaría favorablemente la herramienta Helpdesk propuesta.

Conclusión de los resultados de la encuesta

Los resultados de la encuesta aplicada a los 75 funcionarios del GAD Municipal de Pelileo evidencian de forma contundente las deficiencias del proceso actual de soporte técnico: el 83% no dispone de ningún canal formal para reportar incidencias, el 45% recurre a WhatsApp como medio principal de comunicación, el 73% califica el servicio como poco efectivo o inefectivo, el 73% reporta tiempos de espera superiores a 5 horas, el 81% nunca o casi nunca recibe actualizaciones sobre el estado de su solicitud, y el 67% no obtiene retroalimentación técnica tras la resolución del problema; ante este panorama, el 88% de los funcionarios manifestó estar a favor de implementar un sistema web de gestión de incidencias, validando la pertinencia y viabilidad operativa del sistema Helpdesk propuesto.

b. Aplicación de la entrevista al personal del departamento de TI

Tabla 9. Resultados de la Entrevista — Jefe del Departamento de TI

Datos del Entrevistado			
Institución: GAD Municipal del Cantón Pelileo			
Cargo: Jefe del Departamento de Tecnologías de la Información			
Nº	Pregunta	Respuesta	Observación
1	¿Qué canales utilizan los funcionarios para solicitar ayuda?	Los funcionarios reportan las incidencias principalmente a través de llamadas telefónicas directas al técnico, mensajes de WhatsApp y visitas presenciales al Departamento de TI. No existe un canal formal ni centralizado para el reporte.	

2	¿Con qué frecuencia aproximada reciben solicitudes de soporte técnico durante una semana de trabajo?	Se reciben entre 15 y 25 solicitudes semanales aproximadamente, distribuidas entre los diferentes departamentos de la institución. En épocas de mayor carga administrativa la cantidad puede incrementarse.	
3	¿Cómo llevan registro de estas solicitudes actualmente?	Actualmente no se lleva un registro formal de las incidencias. El control se realiza de manera informal a través de mensajes en WhatsApp y la memoria del personal técnico. No existe una bitácora ni sistema de tickets.	
4	¿Cuánto tiempo toma en promedio la resolución de una incidencia desde que es reportada hasta que se soluciona?	El tiempo varía considerablemente según la complejidad. Problemas simples (reinicio, configuración) pueden resolverse en 30 minutos, mientras que fallas de hardware o red pueden tardar más de 8 horas o incluso días.	La falta de priorización y categorización incrementa los tiempos de resolución.
5	¿Qué problemas o dificultades ha identificado en el proceso actual de gestión de incidencias?	• Pérdida de solicitudes por los canales informales utilizados. • Ausencia de priorización formal de incidencias según urgencia o impacto. • Imposibilidad de medir métricas de desempeño del equipo de TI.	
6	¿Ha ocurrido que alguna solicitud de soporte quede sin atención o se pierda el registro de la misma?	Sí, ocurre con frecuencia. Al no existir un sistema de tickets, las solicitudes enviadas por WhatsApp se pierden entre otros mensajes o simplemente no son vistas a tiempo. El funcionario debe volver a reportar el problema.	
7	¿Existe actualmente alguna forma de medir o evaluar el desempeño del equipo de TI?	No existe ningún mecanismo formal de medición. No se calculan indicadores como tiempo promedio de respuesta, tasa de resolución en primer contacto ni satisfacción del usuario. La evaluación es completamente subjetiva.	
8	¿Cómo afecta la situación actual a la carga laboral del personal de TI y a la productividad de los funcionarios administrativos?	La desorganización genera una carga laboral impredecible y estresante para el equipo técnico. Los funcionarios administrativos pierden tiempo productivo al no recibir atención oportuna, lo que retrasa procesos	El impacto se extiende a la calidad del servicio público brindado por la institución.

		administrativos críticos de la institución.	
9	¿Cuáles considera que son las funcionalidades más importantes que debería tener un sistema de Helpdesk para satisfacer las necesidades del departamento?	• Sistema de tickets con categorización y priorización automática. • Base de conocimiento para soluciones a problemas recurrentes. • Notificaciones automáticas al usuario y al técnico asignado.	
10	¿Qué aspectos del proceso actual considera que deben mantenerse o adaptarse al momento de diseñar el nuevo sistema?	Se debe mantener la comunicación directa con el usuario para la confirmación de resolución. El sistema debe adaptarse al flujo de trabajo actual del departamento, permitiendo la asignación manual de tickets en casos especiales y respetando la jerarquía de atención ya establecida.	La implementación progresiva facilitará la adopción del sistema por parte del personal.

Tabla 10. Resultados de la Entrevista — Técnico de Soporte de TI

Datos del Entrevistado	
Institución: GAD Municipal del Cantón Pelileo Cargo: Técnico de Soporte — Departamento de Tecnologías de la Información	

Nº	Pregunta	Respuesta	Observación
1	¿Qué canales utilizan los funcionarios para solicitar ayuda?	Los funcionarios se comunican principalmente de manera presencial o por WhatsApp. En algunas ocasiones llaman directamente al celular del técnico. No se utiliza ningún sistema formal de tickets.	La dependencia del celular personal del técnico genera disponibilidad irregular del servicio.
2	¿Con qué frecuencia aproximada reciben solicitudes de soporte técnico durante una semana de trabajo?	Aproximadamente entre 10 y 20 solicitudes semanales, concentradas principalmente en los departamentos de Dirección Financiera, Obras Públicas y Secretaría General por el volumen de trabajo con equipos.	Los departamentos con mayor número de equipos generan más incidencias.
3	¿Cómo llevan registro de estas solicitudes actualmente?	No se lleva ningún registro formal. Los mensajes de WhatsApp sirven como referencia temporal, pero se	

		pierden con el tiempo. No existe ninguna bitácora física ni digital.	
4	¿Cuánto tiempo toma en promedio la resolución de una incidencia desde que es reportada hasta que se soluciona?	Problemas de software o configuración suelen resolverse entre 1 y 3 horas. Problemas de hardware o red pueden tardar más de un día, especialmente si requieren adquisición de repuestos.	
5	¿Qué problemas o dificultades ha identificado en el proceso actual de gestión de incidencias?	• Interrupciones constantes por solicitudes no planificadas. • Los mismos problemas se repiten porque no existe base de conocimiento. • El usuario no sabe en qué estado está su solicitud.	
6	¿Ha ocurrido que alguna solicitud de soporte quede sin atención o se pierda el registro de la misma?	Sí, especialmente las que llegan fuera del horario de atención o cuando el técnico está atendiendo otro problema presencialmente. Al no existir un sistema de cola de espera, las solicitudes se olvidan con facilidad.	La falta de sistema de cola incrementa el riesgo de solicitudes no atendidas.
7	¿Existe actualmente alguna forma de medir o evaluar el desempeño del equipo de TI?	No existe ninguna evaluación formal. Ocasionalmente se reciben comentarios verbales de los jefes departamentales, pero no hay indicadores ni reportes de desempeño.	
8	¿Cómo afecta la situación actual a la carga laboral del personal de TI y a la productividad de los funcionarios administrativos?	La desorganización provoca interrupciones constantes que reducen la capacidad de planificar el trabajo. Los funcionarios administrativos se ven afectados cuando sus equipos no funcionan y no reciben atención oportuna.	La carga imprevisible afecta la calidad y puntualidad de la atención.
9	¿Cuáles considera que son las funcionalidades más importantes que debería tener un sistema de Helpdesk para satisfacer las necesidades del departamento?	• Registro automático de tickets con número de seguimiento. • Categorización por tipo de incidencia (hardware, software, red). • Asignación automática según disponibilidad del técnico. • Notificación al usuario cuando se actualiza su ticket. • Historial de soluciones aplicadas para consulta futura.	
10	¿Qué aspectos del proceso actual considera que deben	El nuevo sistema debe permitir la atención presencial cuando sea	

	mantenerse o adaptarse al momento de diseñar el nuevo sistema?	necesaria, sin obligar al técnico a completar todos los pasos digitalmente en ese momento. La interfaz debe ser simple para que los funcionarios no necesiten capacitación extensa.	
--	--	---	--

Conclusión de la entrevista:

Las entrevistas realizadas al Jefe de TI y al Técnico de Soporte confirman y complementan los resultados de la encuesta, evidenciando que el actual proceso de gestión de incidencias en el GAD Municipal de Pelileo carece de formalización, registro, trazabilidad y métricas de control. La dispersión de los canales de comunicación genera pérdida de solicitudes, interrupciones constantes y una elevada sobrecarga laboral que afecta la calidad del servicio. Ambos entrevistados coinciden en la necesidad urgente de implementar un sistema Helpdesk que centralice el registro, permita la categorización y priorización de incidencias y proporcione indicadores de desempeño para mejorar la eficiencia y la mejora continua del soporte técnico.

c. *Recolección de información ficha de observación*

En esta sección se presentan los resultados obtenidos a partir de la ficha de observación aplicada en el Departamento de Tecnologías de la Información del GAD Municipal de Pelileo. Los datos recolectados mediante la observación directa corroboran los hallazgos de la encuesta y las entrevistas, y permiten identificar con mayor precisión las deficiencias del proceso actual de gestión de incidencias.

Tabla 11. Resultados de la ficha de observación

Ficha de Observación — GAD Municipal del Cantón Pelileo		
Fecha de observación: 26/03/2026	Hora inicio: 10:00	Hora fin: 12:00
Lugar: Departamento de TI — GAD Municipal de Pelileo		
N° de ficha: 1		

Indicadores	Observación	Novedades
Recepción de la solicitud		
Canal de recepción utilizado (llamada, WhatsApp, visita presencial, correo).	Las solicitudes llegan principalmente por WhatsApp y visitas presenciales. No existe un canal oficial habilitado.	La dispersión de canales impide la trazabilidad de las solicitudes.
Se registra formalmente la solicitud al recibirla (número, descripción, solicitante).	No se realiza ningún registro formal. El técnico retiene la información en la memoria o en mensajes de WhatsApp.	La ausencia de registro genera pérdida de solicitudes.
Tiempo entre recepción e inicio del proceso	Varía entre inmediata y más de 30 minutos dependiendo de si el técnico está atendiendo otra incidencia.	Sin sistema de cola, las esperas son impredecibles.
Identificación y diagnóstico del problema		
Acciones realizadas por el técnico para identificar el tipo y origen de la incidencia.	El diagnóstico se realiza in situ o por descripción verbal/escrita del usuario. No se usa ningún sistema de diagnóstico estandarizado.	La falta de estandarización incrementa el tiempo de diagnóstico.
Se consulta historial previo o base de conocimiento sobre incidencias similares.	No existe base de conocimiento ni historial de incidencias. El técnico resuelve desde cero cada vez.	Esto incrementa la reincidencia de los mismos problemas.
Resolución del problema		
Se documenta el procedimiento o solución aplicada durante o después de la atención.	No se documenta ninguna solución aplicada. La información queda únicamente en la memoria del técnico.	La falta de documentación impide la generación de una base de conocimiento.
Verificación final con el funcionario para confirmar que el problema fue resuelto satisfactoriamente.	La verificación se realiza verbalmente en persona o mediante un mensaje de WhatsApp. No existe confirmación formal.	Sin confirmación formal no hay cierre oficial del ticket.
Seguimiento y cierre		
Se ofrece indicación adicional al funcionario para prevenir futuros problemas.	Ocasionalmente el técnico da indicaciones verbales al usuario. No existe proceso estandarizado de retroalimentación.	La falta de retroalimentación contribuye a la reincidencia.
Se registra el cierre de la incidencia en algún sistema, bitácora o documento.	No se registra el cierre de ninguna incidencia. No existe evidencia	Sin registro de cierre es imposible generar métricas de desempeño.

	del trabajo realizado por el Departamento de TI.	
Interacción con el funcionario		
Canales utilizados para comunicarse durante la atención (presencial, llamada, WhatsApp, correo).	La comunicación durante la atención se realiza principalmente de forma presencial o por WhatsApp. No se utiliza ningún sistema de mensajería institucional.	El uso de canales personales mezcla lo institucional con lo personal.
El técnico mantiene informado al funcionario sobre el avance y estado de la solución.	En la mayoría de casos el técnico no informa sobre el avance. El funcionario desconoce si su solicitud está siendo atendida.	Esta situación genera incertidumbre y reclamos reiterados al área de TI.
Notas adicionales del investigador		
Observaciones generales: El Departamento de TI muestra disposición para adoptar una herramienta tecnológica que formalice su proceso de trabajo. La infraestructura institucional disponible es compatible con el despliegue de un sistema Helpdesk Open Source.		

Conclusión de la ficha de observación

En general, el proceso de gestión de incidencias en el GAD Municipal de Pelileo presenta múltiples áreas críticas que requieren mejora. Actualmente no existe un sistema formal para registrar, gestionar ni dar seguimiento a las incidencias técnicas, lo que limita la trazabilidad, la eficiencia en la resolución y la generación de métricas de desempeño. La comunicación entre el técnico y el funcionario se realiza por canales personales e informales, sin confirmación formal ni retroalimentación documentada. Estos hallazgos corroboran los resultados de la encuesta y las entrevistas, y ponen de manifiesto la necesidad urgente de implementar un sistema Helpdesk basado en herramientas Open Source que estandarice, centralice y automatice el proceso de soporte técnico en la institución.

2.2.4 Procesamiento y análisis de datos

Una vez recolectada la información mediante los instrumentos descritos, se procedió al procesamiento y análisis de los datos siguiendo las etapas que se describen a continuación:

Para el análisis estadístico de los datos obtenidos a través de la encuesta aplicada a los 75 funcionarios del GAD Municipal de Pelileo, se elaboraron gráficos de frecuencias que permitieron visualizar y resumir las respuestas por cada pregunta (Figura 10). Por su parte, los procesos identificados mediante la ficha de observación aplicada al Departamento de Tecnologías de la Información se representaron de forma esquemática en la Figura 11.

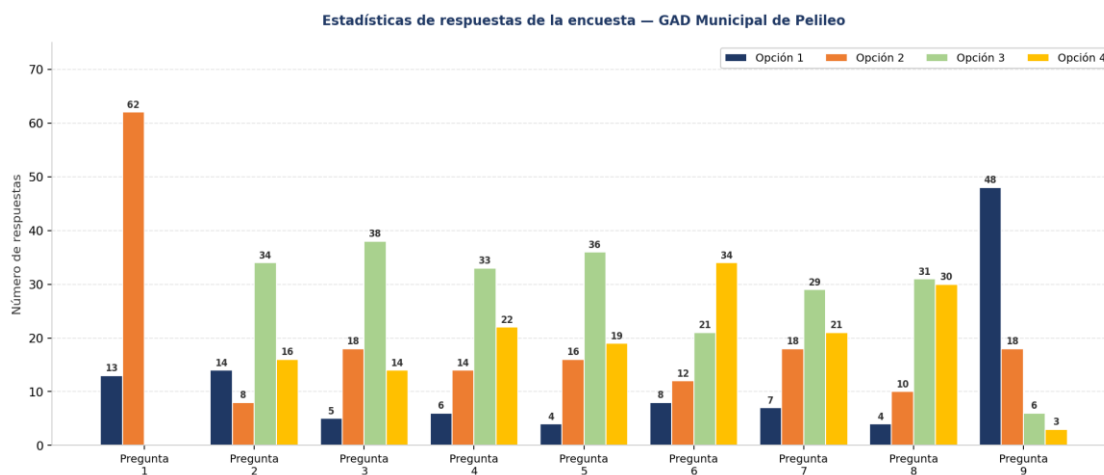


Figura 10. Estadísticas de respuestas de la encuesta

En la Figura 10 se puede evidenciar la frecuencia de las respuestas obtenidas en cada pregunta, denotando los siguientes hallazgos principales:

- Los tiempos de atención (Pregunta 6) son excesivamente altos, con la mayoría de los requerimientos tardando más de 8 horas en ser atendidos, lo que afecta directamente la productividad institucional.
- Las preguntas 7 y 8 confirman la ausencia de retroalimentación técnica y de seguimiento proactivo sobre el estado de las incidencias reportadas, factores clave de insatisfacción del personal.
- La Pregunta 9 refleja una alta aceptación (88%) hacia la implementación de un sistema web Helpdesk, lo que valida la viabilidad operativa y la pertinencia del proyecto en el GAD Municipal de Pelileo.

Mediante la aplicación de la ficha de observación al Departamento de Tecnologías de la Información del GAD Municipal de Pelileo, se identificaron los pasos que

conforman el proceso actual de gestión de incidencias de soporte técnico, tal como se muestra en la Figura 11.

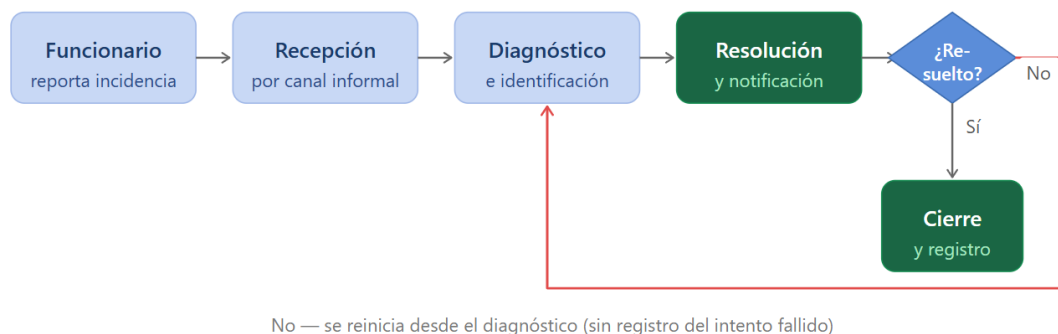


Figura 11. Gestión de incidencias identificado mediante la ficha de observación

En base a los datos recolectados a través de la ficha de observación, se identificaron los siguientes aspectos del proceso actual:

- El diagnóstico e identificación del problema depende exclusivamente del criterio del técnico disponible, sin consulta a bases de conocimiento ni historial de incidencias previas, lo que incrementa el tiempo de diagnóstico.
- La resolución se realiza de manera presencial o mediante acceso remoto, según el tipo de incidencia. Sin embargo, no existe un procedimiento estandarizado ni documentación de las soluciones aplicadas.

En síntesis, se evidencia que la ausencia de un registro formal y sistemático de las incidencias dificulta el seguimiento, el análisis y la gestión eficiente de los problemas técnicos. Esta situación pone de manifiesto la necesidad imperiosa de implementar un sistema que permita registrar, dar seguimiento y resolver las incidencias de manera ordenada, optimizando así el servicio de soporte técnico y mejorando la eficiencia operativa del Departamento de TI del GAD Municipal de Pelileo.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

El presente capítulo expone los resultados obtenidos a lo largo del proceso investigativo y el desarrollo del sistema Helpdesk para el GAD Municipal de Pelileo. La estructura del capítulo sigue una secuencia lógica que parte del análisis diagnóstico del problema identificado, continúa con la selección fundamentada de las herramientas tecnológicas Open Source empleadas y culmina con el desarrollo, pruebas e implementación del sistema propuesto.

En primera instancia, se presentan los resultados del análisis de los procesos actuales de gestión de incidencias, evidenciando las deficiencias operativas que motivaron el proyecto. A continuación, se describe el proceso optimizado que el sistema Helpdesk implementa y validado con los datos recolectados en el Capítulo II. Posteriormente, se justifica la selección de cada componente del stack tecnológico mediante matrices de evaluación ponderadas. Finalmente, se documenta el ciclo de desarrollo bajo la metodología XP, las pruebas de aceptación y la implementación del sistema en la infraestructura institucional del GAD Municipal de Pelileo.

3.1 Análisis

Con base en la información recopilada en el Capítulo II mediante los instrumentos de investigación como encuesta a funcionarios del GAD Municipal de Pelileo, entrevistas al personal del Departamento de TI y fichas de observación del proceso actual, se procedió al análisis del proceso de gestión de incidencias existente, la selección de herramientas tecnológicas Open Source y el diseño del proceso optimizado que sustenta el desarrollo del sistema Helpdesk propuesto.

Los hallazgos obtenidos en la fase de diagnóstico corroboran los antecedentes investigativos revisados en el Capítulo I. Estas referencias avalan la pertinencia de la solución propuesta para el contexto institucional del GAD Municipal de Pelileo.

3.1.1 Análisis de los procesos involucrados de incidencias de soporte técnico

Tal como lo confirman los resultados de la encuesta aplicada, el 83% de los funcionarios no dispone de un canal formal para reportar incidencias (Pregunta 1), el 45% recurre a mensajes de WhatsApp (Pregunta 2) y el 81% nunca o casi nunca recibe actualizaciones sobre el estado de su solicitud (Pregunta 8). Estos indicadores reflejan una situación estructuralmente deficiente que impacta directamente en la productividad institucional.

En la Figura 12 se visualiza el diagrama de flujo del proceso actual de gestión de incidencias identificado mediante la ficha de observación aplicada en el GAD Municipal de Pelileo, donde se evidencian los pasos que conforman el proceso informal vigente.

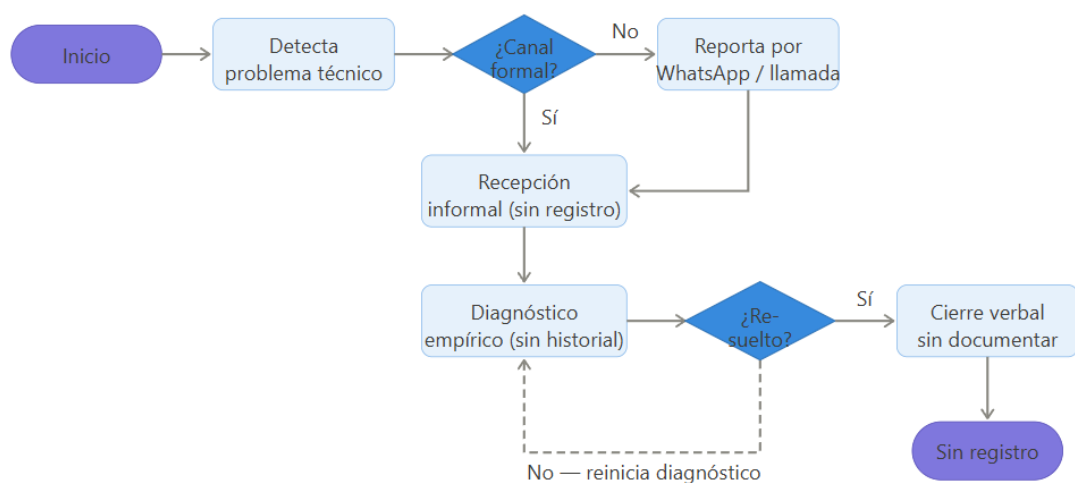


Figura 12. Gestión de incidencias identificado

En base a la Figura 12, se puede identificar el conjunto de procesos que conforman la gestión de incidencias dentro de la institución pública. Esta representación proporciona una referencia clave para estandarizar el proceso de registro de incidencias, permitiendo así optimizar el proceso de seguimiento y resolución de las solicitudes de soporte técnico.

Tabla 12. Descripción del proceso actual de gestión de incidencias

Nº	Proceso / Etapa	Descripción del proceso actual	Problema identificado
1	Reporte de incidencia	El funcionario detecta un problema técnico y lo reporta de forma verbal, por llamada telefónica o mediante mensaje de WhatsApp al técnico de TI. No existe un canal formal ni un formulario estandarizado.	Pérdida de solicitudes. Sin trazabilidad del reporte.
2	Recepción informal	El técnico recibe la solicitud a través del canal informal utilizado. No se registra ningún dato sobre el tipo de problema, área solicitante ni hora de recepción.	Ausencia de registro formal. Imposibilidad de priorizar.
3	Diagnóstico empírico	El técnico realiza el diagnóstico in situ o por descripción verbal del usuario, sin consultar historial previo ni base de conocimiento institucional.	Diagnósticos lentos. Reincidencia de problemas no documentados.
4	Resolución	El técnico aplica la solución de forma presencial o mediante acceso remoto (AnyDesk). No documenta el procedimiento ni la solución implementada.	Sin documentación. Sin generación de base de conocimiento.
5	Cierre verbal	El técnico informa verbalmente al funcionario que el problema fue resuelto. No existe confirmación formal ni registro de cierre.	Sin métricas de tiempo de resolución. Sin evidencia del servicio.
6	Sin seguimiento post-cierre	No se realiza ningún seguimiento posterior. El funcionario no recibe retroalimentación sobre la causa del problema.	El 67% de funcionarios nunca recibe explicación (Pregunta 7, encuesta).

El análisis de este proceso evidencia la existencia de desperdicios operativos que impiden una gestión eficiente de las incidencias. La ausencia de un sistema de registro formal imposibilita la medición de indicadores de desempeño clave como el tiempo promedio de respuesta, la tasa de resolución en primer contacto y el nivel de satisfacción del usuario.

3.1.2 Matriz de evaluación de riesgos del proceso actual de gestión de incidencias

Como parte del cumplimiento del Objetivo Específico analizar los procesos involucrados en la identificación, registro, diagnóstico y resolución de interrupciones de soporte técnico en el GAD Municipal de Pelileo, se realizó una evaluación estructurada de los riesgos operativos identificados a partir de los resultados de la encuesta, las entrevistas al Departamento de TI y la ficha de observación aplicada en

el Capítulo II. La metodología de evaluación sigue el enfoque de análisis de riesgo cualitativo-cuantitativo[25], utilizando la fórmula:

$$\text{Nivel de Riesgo} = \text{Probabilidad (P)} \times \text{Impacto (I)}$$

Donde la probabilidad e impacto se valoran en una escala de 1 a 5, generando una puntuación entre 1 y 25. Los riesgos identificados corresponden directamente a las deficiencias detectadas en cada etapa del proceso actual de gestión de incidencias representado en la Figura 12.

Tabla 13. Escala de valoración de probabilidad e impacto

Valor	Probabilidad / Impacto	Interpretación en el contexto del GAD Municipal de Pelileo
1	Muy baja / Mínimo	El evento es muy poco probable o su impacto es insignificante para la operación.
2	Baja / Bajo	Ocurrencia poco frecuente o impacto menor que puede resolverse rápidamente.
3	Media / Moderado	Ocurrencia ocasional o impacto que afecta parcialmente el servicio de TI.
4	Alta / Alto	Ocurrencia frecuente o impacto significativo en la operatividad institucional.
5	Muy alta / Crítico	Ocurrencia casi segura o impacto que paraliza procesos administrativos críticos.

Tabla 14. Clasificación del nivel de riesgo

Puntuación (P × I)	Nivel de riesgo	Acción requerida
1 – 4	BAJO	Riesgo aceptable. Monitoreo periódico. No requiere acción inmediata.
5 – 9	MEDIO	Requiere plan de mitigación documentado y seguimiento mensual por el Jefe de TI.
10 – 25	ALTO	Riesgo crítico. Acción inmediata requerida. El sistema Helpdesk mitiga directamente este nivel de riesgo.

La siguiente matriz visualiza la distribución del nivel de riesgo según la combinación de probabilidad e impacto. Las celdas en rojo/amarillo (ALTO) corresponden a los

riesgos que el sistema Helpdesk aborda directamente como solución al problema identificado.

Tabla 15. Mapa de calor — probabilidad vs. impacto.

Probabilidad \ Impacto →	1 – Mínimo	2 – Bajo	3 – Moderado	4 – Alto	5 – Crítico
5 – Muy alta	5	10	15	20	25
4 – Alta	4	8	12	16	20
3 – Media	3	6	9	12	15
2 – Baja	2	4	6	8	10
1 – Muy baja	1	2	3	4	5

Análisis consolidado de la evaluación de riesgos

Del total de 10 riesgos identificados, 7 fueron clasificados como ALTO (puntuación $\geq$ 10), 2 como MEDIO (5-9) y 1 como BAJO (1-4). La concentración de riesgos en el nivel ALTO confirma la criticidad del problema identificado en el GAD Municipal de Pelileo y valida la necesidad urgente de implementar el sistema Helpdesk propuesto.

En síntesis, la evaluación de riesgos confirma que el proceso actual de gestión de incidencias del GAD Municipal de Pelileo opera bajo condiciones de alto riesgo operativo en sus etapas críticas (reporte, registro, diagnóstico, resolución y cierre), lo que fundamenta técnica y metodológicamente el desarrollo del sistema Helpdesk Open Source como solución al problema identificado en el Capítulo I.

La siguiente tabla presenta los 10 riesgos identificados en el proceso actual de gestión de incidencias del GAD Municipal de Pelileo, clasificados conforme a la escala definida en las Tablas 13 y 14. Para cada riesgo se especifica su categoría, causa raíz, etapa del proceso afectada

Tabla 16. Matriz de evaluación de riesgos

ID	Categoría	Riesgo identificado	Causa raíz y etapa afectada	Nivel	Estado
				B/M/A	
R-01	Proceso	Pérdida de solicitudes de soporte técnico	Ausencia de un canal formal de recepción y registro de incidencias.	ALTO	Mitigado con sistema Helpdesk
R-02	Proceso	Ausencia de priorización formal de incidencias	No existe criterio de categorización ni priorización por urgencia e impacto.	ALTO	Mitigado con sistema Helpdesk
R-03	Operativo	Reincidencia de problemas técnicos no documentados	Ausencia de documentación de soluciones y base de conocimiento institucional.	ALTO	Mitigado con sistema Helpdesk
R-04	Gestión	Imposibilidad de medir KPIs de desempeño del área de TI	Inexistencia de un sistema de ticketing que genere datos históricos auditables.	ALTO	Mitigado con sistema Helpdesk
R-05	Operativo	Incumplimiento de tiempos de atención (SLA)	Ausencia de Acuerdos de Nivel de Servicio formalizados y herramienta de monitoreo.	ALTO	Mitigado con sistema Helpdesk
R-06	Proceso	Falta de trazabilidad en el ciclo de vida de la incidencia	Inexistencia de un sistema de seguimiento de estado por ticket.	ALTO	Mitigado con sistema Helpdesk
R-07	Tecnológico	Resistencia al cambio por parte del personal administrativo	Cambio en el flujo de trabajo habitual sin capacitación ni acompañamiento.	MEDIO	En gestión
R-08	Tecnológico	Interrupciones en la conectividad de la red institucional	Dependencia de la infraestructura de red LAN del GAD Municipal de Pelileo.	MEDIO	En gestión
R-09	Gestión	Sobrecarga laboral del personal técnico por solicitudes no planificadas	Ausencia de cola de priorización y asignación estructurada de tickets.	ALTO	Mitigado con sistema Helpdesk
R-10	Seguridad	Acceso no autorizado a información de incidencias institucionales	Uso de canales informales y personales (WhatsApp) para gestionar incidencias institucionales.	BAJO	Mitigado con sistema Helpdesk

3.2 Herramientas Open Source para el desarrollo

La selección de las herramientas tecnológicas Open Source, se fundamentó en el análisis de matrices de evaluación multicriterio, considerando seis criterios organizados en tres niveles de importancia. Esta metodología de selección sigue el enfoque aplicado por estudios similares en el contexto universitario ecuatoriano, donde la selección comparativa de herramientas garantizó la sostenibilidad técnica del proyecto.

Justificación de los criterios de evaluación

Previo a la presentación de las matrices de selección, se justifica a continuación la elección de cada criterio de evaluación y su peso relativo, el cual fue asignado en función de la criticidad de cada dimensión para el éxito del proyecto Helpdesk en el GAD Municipal de Pelileo.

Método de asignación de pesos — Ponderación por importancia relativa

La asignación de los pesos porcentuales a cada criterio de evaluación se realizó mediante el método de ponderación por importancia relativa, técnica ampliamente utilizada en proyectos de selección tecnológica en el contexto universitario latinoamericano [31]. Este método consiste en distribuir el 100% del peso total entre los criterios identificados, asignando mayor porcentaje a aquellos factores que representan restricciones no negociables o de mayor impacto directo en el éxito del proyecto [31].

Los pesos fueron determinados considerando tres dimensiones de análisis:

- (1) Las restricciones institucionales del GAD Municipal de Pelileo como la normativa legal vigente, infraestructura tecnológica disponible y limitaciones presupuestarias.
- (2) Las condiciones del contexto de desarrollo como el equipo unipersonal, plazo académico.

- (3) Los requerimientos técnicos del sistema Helpdesk como la integridad transaccional de datos, manejo de concurrencia de usuarios y coherencia del stack tecnológico.

La distribución resultante agrupa los criterios en tres niveles de importancia:

- Nivel crítico — Restricciones no negociables (20% c/u): Licencia Open Source, Rendimiento y escalabilidad, Compatibilidad con el stack. Cualquiera de estos criterios, de no cumplirse, hace inviable la herramienta con independencia de sus otras características.
- Nivel intermedio — Factores de riesgo operativo (15% c/u): Ecosistema y comunidad, Curva de aprendizaje. Impactan en la productividad del desarrollo, pero son parcialmente gestionables.
- Nivel complementario — Factores mitigables (10%): Documentación oficial. Influye en la velocidad de implementación, pero puede compensarse con fuentes secundarias y soporte comunitario.

Fórmula de cálculo del puntaje ponderado

Con el fin de garantizar la objetividad y reproducibilidad de la evaluación multicriterio, se aplica la siguiente fórmula de puntaje ponderado [31] [32] para cada herramienta evaluada:

$$\text{Puntaje Total} / 100 = \Sigma (W_i \times P_i) \times 20$$

Donde:

- **W_i** = Peso del criterio i expresado como decimal (ejemplo: 20% = 0.20)
- **P_i** = Puntuación asignada a la herramienta en el criterio i , en escala de 1 a 5
- **Puntaje máximo posible:** $\Sigma (W_i \times 5) \times 20 = 1.00 \times 20 = 100$ puntos
- La herramienta con mayor Puntaje Total es seleccionada para la capa evaluada.

En la siguiente tabla se resume el esquema de ponderación aplicado en todas las matrices de selección del presente proyecto:

Tabla 17. Esquema de ponderación para el cálculo del puntaje de herramientas open source

Criterio	Peso (W_i)	Puntuación Máxima	Puntuación Obtenida (P_i)	Puntaje Ponderado ($W_i \times P_i$)	Contribución al Total
Licencia O.S.	0.20	5	P_i (1–5)	$0.20 \times P_i$	Hasta 1.00
Rendimiento	0.20	5	P_i (1–5)	$0.20 \times P_i$	Hasta 1.00
Compatibilidad	0.20	5	P_i (1–5)	$0.20 \times P_i$	Hasta 1.00
Ecosistema	0.15	5	P_i (1–5)	$0.15 \times P_i$	Hasta 0.75
Curva aprendizaje	0.15	5	P_i (1–5)	$0.15 \times P_i$	Hasta 0.75
Documentación	0.10	5	P_i (1–5)	$0.10 \times P_i$	Hasta 0.50
PUNTAJE TOTAL MÁXIMO (Suma de todos los aportes):				5.00 puntos (100%)	

3.2.1 Selección del Framework Frontend

Para la capa de presentación del sistema se evaluaron los cuatro frameworks de desarrollo frontend más relevantes del ecosistema actual, aplicando la metodología de selección multicriterio descrita en la sección 3.2 con los seis criterios y pesos definidos en la Tabla 16. La evaluación priorizó la compatibilidad con TypeScript, las capacidades de renderizado del lado del servidor (SSR) y la coherencia de lenguaje con el backend Node.js.

Criterios de evaluación aplicados a frameworks frontend

A continuación, se detalla cómo se aplicó cada uno de los seis criterios de la Tabla 10 al contexto específico de la selección de frameworks frontend, incluyendo la escala de puntuación utilizada en cada caso:

Tabla 18. Criterios de evaluación y escala de puntuación

Criterio	Peso (%)	Puntaje Máx.	Justificación del criterio aplicado a frameworks frontend
Licencia Open Source	20%	5	El framework debe contar con licencia Open Source (MIT, Apache 2.0 o equivalente) para cumplir con la directriz del MINTEL y la Ley Orgánica para la Transformación Digital del Ecuador. Puntuación 5: licencia MIT o Apache 2.0. Puntuación 3: licencia dual con restricciones. Puntuación 1: licencia propietaria o comercial.
Rendimiento y escalabilidad	20%	5	Se evalúa la capacidad del framework de renderizar páginas de forma eficiente bajo carga institucional (75 usuarios concurrentes), considerando las capacidades de SSR (Server-Side Rendering), CSR (Client-Side Rendering) y generación estática. Puntuación 5: SSR + CSR híbrido nativo con optimización automática. Puntuación 3: SSR mediante librería adicional. Puntuación 1: solo CSR sin capacidades de SSR.
Compatibilidad con el stack	20%	5	Evalúa la integración nativa del framework con Node.js (backend), TypeScript (lenguaje del stack) y el ecosistema npm. La compatibilidad garantiza uniformidad de lenguaje en toda la arquitectura y reduce la deuda técnica. Puntuación 5: integración nativa completa con Node.js y TypeScript sin configuración adicional. Puntuación 3: compatible, pero requiere adaptadores. Puntuación 1: lenguaje distinto al stack del proyecto.
Ecosistema y comunidad	15%	5	Se valora la disponibilidad de librerías de componentes UI, herramientas de prueba, recursos de aprendizaje y actividad en repositorios GitHub. Un ecosistema robusto reduce el tiempo de implementación de módulos como dashboards, tablas de datos y formularios del sistema Helpdesk. Puntuación 5: ecosistema mayor del mercado (>1M descargas semanales npm). Puntuación 3: ecosistema activo pero limitado. Puntuación 1: ecosistema reducido o en declive.
Curva de aprendizaje	15%	5	Considerando el plazo académico enero–junio 2026 y el desarrollo individual del proyecto, se evalúa la facilidad de implementación del framework, la disponibilidad de documentación oficial en español y la intuitividad de su API de rutas y componentes. Puntuación 5: documentación extensa, sintaxis familiar para desarrolladores JavaScript, curva baja. Puntuación 3: requiere aprendizaje de conceptos nuevos. Puntuación 1: paradigma completamente diferente, alta curva.
Documentación oficial	10%	5	Se evalúa la calidad, completitud y actualización de la documentación oficial del framework, incluyendo guías de integración con TypeScript, ejemplos de autenticación con JWT y tutoriales de despliegue en entornos Node.js. Puntuación 5: documentación oficial completa, actualizada y con ejemplos de código para todos los casos de uso del proyecto. Puntuación 3: documentación parcial o desactualizada. Puntuación 1: sin documentación oficial.

En la Tabla 17 se presenta la matriz de evaluación multicriterio con los puntajes obtenidos por cada framework en los seis criterios, junto con el total ponderado calculado mediante la fórmula anterior:

Tabla 19. Matriz de selección — Framework Frontend

Herramienta	Licencia	Descripción	Lic. O.S. (20%)	Rend. (20%)	Compat. (20%)	Ecosist. (15%)	Curva (15%)	Doc. (10%)	Total /100
Next.js 14 ★	MIT	Framework React con SSR, App Router y soporte nativo TypeScript. Rutas de API integradas. Renderizado híbrido SSR + CSR.	5	5	5	5	5	5	100.0
Vue.js 3 + Nuxt	MIT	Framework progresivo con SSR vía Nuxt. Ecosistema menor que React. Compatible con TypeScript.	5	4	3	4	4	4	80.0
Angular 17	MIT	Framework empresarial con TypeScript obligatorio. Alta curva de aprendizaje y overhead de configuración para proyectos medianos.	5	4	3	4	2	4	74.0
SvelteKit	MIT	Compila a JS vanilla con mínimo overhead. Ecosistema reducido y compatibilidad limitada con librerías del ecosistema React/Node.	5	5	2	3	3	3	72.0

Next.js 14 fue seleccionado como framework frontend con un puntaje de 100/100, tal como se refleja en la Tabla 17, obteniendo calificación máxima en los seis criterios de evaluación. La ventaja determinante frente a las demás alternativas radica en su compatibilidad nativa con el runtime de Node.js, la coherencia de lenguaje

JavaScript/TypeScript en toda la arquitectura del sistema, las capacidades de renderizado híbrido (SSR + CSR) que garantizan una experiencia de usuario óptima en la red institucional del GAD Municipal de Pelileo, y su ecosistema de librerías Tailwind CSS, NextAuth.js, Lucide React que cubre directamente los requerimientos de interfaz del sistema Helpdesk propuesto.

3.2.2 Selección del framework backend

Para la capa de lógica de negocio se evaluaron los principales frameworks de desarrollo de APIs REST, aplicando la metodología de selección multicriterio con los seis criterios y pesos definidos en la Tabla 16. Los factores determinantes fueron la compatibilidad con el driver node-oracledb, el soporte nativo para TypeScript y la coherencia de lenguaje con el frontend Next.js.

Criterios de evaluación aplicados a frameworks backend

A continuación, se detalla la aplicación de cada criterio al contexto específico de selección de frameworks backend, con la escala de puntuación utilizada:

Tabla 20. Criterios de evaluación y escala de puntuación

Criterio	Peso	Máx.	Justificación y escala de puntuación
Licencia Open Source	20%	5	5 = Licencia MIT, Apache 2.0 o BSD (compatible con directriz MINTEL). 3 = Licencia dual con restricciones de distribución. 1 = Licencia propietaria o comercial.
Rendimiento y escalabilidad	20%	5	5 = Arquitectura asíncrona nativa (non-blocking I/O) con soporte para alto volumen de peticiones concurrentes. 3 = Asíncrono mediante plugins o extensiones. 1 = Arquitectura síncrona bloqueante, limitada para APIs REST.
Compatibilidad con el stack	20%	5	5 = Mismo lenguaje del frontend (JavaScript/TypeScript), driver node-oracledb disponible y ecosistema npm nativo. 3 = Compatible mediante adaptadores pero lenguaje distinto al frontend. 1 = Lenguaje completamente distinto al stack del proyecto, sin driver Oracle nativo.
Ecosistema y comunidad	15%	5	5 = Mayor ecosistema del mercado backend (>1M descargas semanales npm), librerías maduras para JWT, validación y ORM disponibles. 3 = Ecosistema activo pero limitado frente al líder del mercado. 1 = Ecosistema reducido o en declive.
Curva de aprendizaje	15%	5	5 = Sintaxis minimalista familiar para desarrolladores JavaScript, documentación extensa en español, baja curva de adopción dentro del plazo académico. 3 = Requiere aprendizaje de paradigmas nuevos (OOP estricto, decoradores). 1 = Paradigma completamente diferente al stack JavaScript del proyecto.
Documentación oficial	10%	5	5 = Documentación oficial completa con guías de integración para Oracle, JWT y TypeScript. 3 = Documentación parcial, requiere fuentes secundarias. 1 = Sin documentación oficial actualizada.

Tabla 21. Matriz de selección — Framework Backend

Herramienta	Licencia	Descripción	Lic.O.S. (20%)	Rend. (20%)	Compat. (20%)	Ecosist. (15%)	Curva (15%)	Doc. (10%)	Total /100
Node.js + Express ★	MIT	Runtime JS asíncrono + framework minimalista. Compatible con node-oracledb y ecosistema npm. Mismo lenguaje que el frontend.	5	5	5	5	5	5	100.0
Django (Python)	BSD	Framework Python maduro con ORM potente. No comparte lenguaje con el frontend Next.js.	5	4	2	5	4	4	79.0
Spring Boot (Java)	Apache 2.0	Robusto para empresas Java. Alta curva de aprendizaje y overhead de configuración para proyectos medianos.	5	5	2	5	2	4	77.0
FastAPI (Python)	MIT	API asíncrona muy rápida. Lenguaje distinto al stack JavaScript del proyecto.	5	5	2	4	3	3	75.0

Las tres alternativas obtienen calificación crítica de 2/5 en Compatibilidad porque utilizan lenguajes distintos (Python/Java) al stack JavaScript/TypeScript del proyecto. Esto generaría una arquitectura heterogénea que incrementaría la complejidad de mantenimiento y eliminaría la ventaja de uniformidad de lenguaje entre capas. Adicionalmente, Spring Boot obtiene 2/5 en Curva de aprendizaje debido al overhead del paradigma de inyección de dependencias de Java, incompatible con el plazo de desarrollo disponible.

Node.js + Express fue seleccionado como framework backend con un puntaje de 100/100, consolidando la uniformidad del stack JavaScript/TypeScript en toda la arquitectura del sistema Helpdesk y garantizando la integración nativa con Oracle 19c mediante el driver node-oracledb.

3.2.3 Sistema gestor de base de datos

Para la capa de persistencia del sistema Helpdesk se evaluaron las principales opciones de Sistemas gestores de bases de datos (SGBD) relacionales y no relacionales disponibles en el mercado. A diferencia de las capas de frontend y backend, donde la selección fue estrictamente técnica, en esta capa existe un condicionante institucional determinante: el GAD Municipal de Pelileo opera actualmente con Oracle Database 19c como motor de base de datos oficial en toda su infraestructura tecnológica, constituyendo un requisito no negociable establecido por el Departamento de Tecnologías de la Información de la institución. En consecuencia, cualquier sistema desarrollado para el GAD debe integrarse obligatoriamente con Oracle 19c para garantizar la coherencia de la infraestructura tecnológica existente, evitar costos adicionales de licenciamiento y cumplir con las directrices de gestión tecnológica institucional.

No obstante, el condicionante institucional, se aplicó la metodología de evaluación multicriterio definida en la sección 3.2 para validar objetivamente que Oracle Database 19c es la opción técnicamente más adecuada entre las alternativas disponibles, y para documentar las razones por las cuales las otras opciones no se ajustan al contexto del proyecto. Se adaptó el primer criterio de la Tabla 10 de 'Licencia Open Source' a 'Disponibilidad y requisito institucional', dado que el factor determinante en esta capa no es únicamente la naturaleza de la licencia, sino la disponibilidad real y obligatoriedad de uso del motor en la infraestructura tecnológica vigente del GAD Municipal de Pelileo, lo cual impacta directamente en la viabilidad técnica y económica del proyecto.

3.2.4 Resumen del stack tecnológico seleccionado

Como resultado del proceso de selección multicriterio aplicado en las secciones 3.2.1, 3.2.2 y 3.2.3, el stack tecnológico definitivo del sistema Helpdesk para el GAD Municipal de Pelileo quedó conformado conforme se detalla en la Tabla 14. Todas las herramientas seleccionadas son Open Source o cuentan con licenciamiento institucional vigente, garantizando la soberanía tecnológica y eliminando costos recurrentes de licenciamiento, en cumplimiento de las directrices del MINTEL y la Ley Orgánica para la Transformación Digital del Ecuador.

Tabla 22. Stack tecnológico seleccionado para el sistema Helpdesk

Capa del sistema	Herramienta seleccionada	Versión / Licencia	Rol en el sistema
Frontend	Next.js + React	v14 App Router / MIT	Interfaz web responsiva, SSR, rutas de API y dashboard institucional.
Estilos	Tailwind CSS + CSS Modules	v3 / MIT	Diseño responsivo optimizado para producción con estilos modulares por componente.
Cliente HTTP	Axios	v1.x / MIT	Peticiones HTTP asíncronas del frontend al backend REST.
Iconografía	Lucide React	v0.4x / ISC	Íconos SVG Open Source optimizados para React con TypeScript.
Backend	Node.js + Express.js	v20 LTS / MIT	API REST, lógica de negocio, SLA automático y gestión de módulos.
Lenguaje	TypeScript	v5.x / Apache 2.0	Tipado estático para reducir errores y mejorar mantenibilidad.
Autenticación	NextAuth.js + JWT	v4 / ISC + MIT	Autenticación stateless, sesiones JWT y control de roles.
Seguridad	Bcrypt	v5.x / MIT	Hash seguro de contraseñas antes del almacenamiento en base de datos.
Base de datos	Oracle Database 19c	19c / Institucional	Persistencia relacional con soporte ACID, auditoría y analítica avanzada.
Driver BD	node-oracledb	v6.x / Apache 2.0	Conexión Node.js a Oracle con pool de conexiones y soporte async/await.

En la Figura 13 se resume el entorno tecnológico completo del sistema, mostrando la interacción entre las capas de presentación (Next.js + React), lógica de negocio (Node.js + Express) y persistencia de datos (Oracle Database 19c), consolidando una

arquitectura de tres capas desacopladas con uniformidad de lenguaje JavaScript/TypeScript en las capas de cliente y servidor.

Resumen final de todas las tecnologías escogidas entorno al sistema.

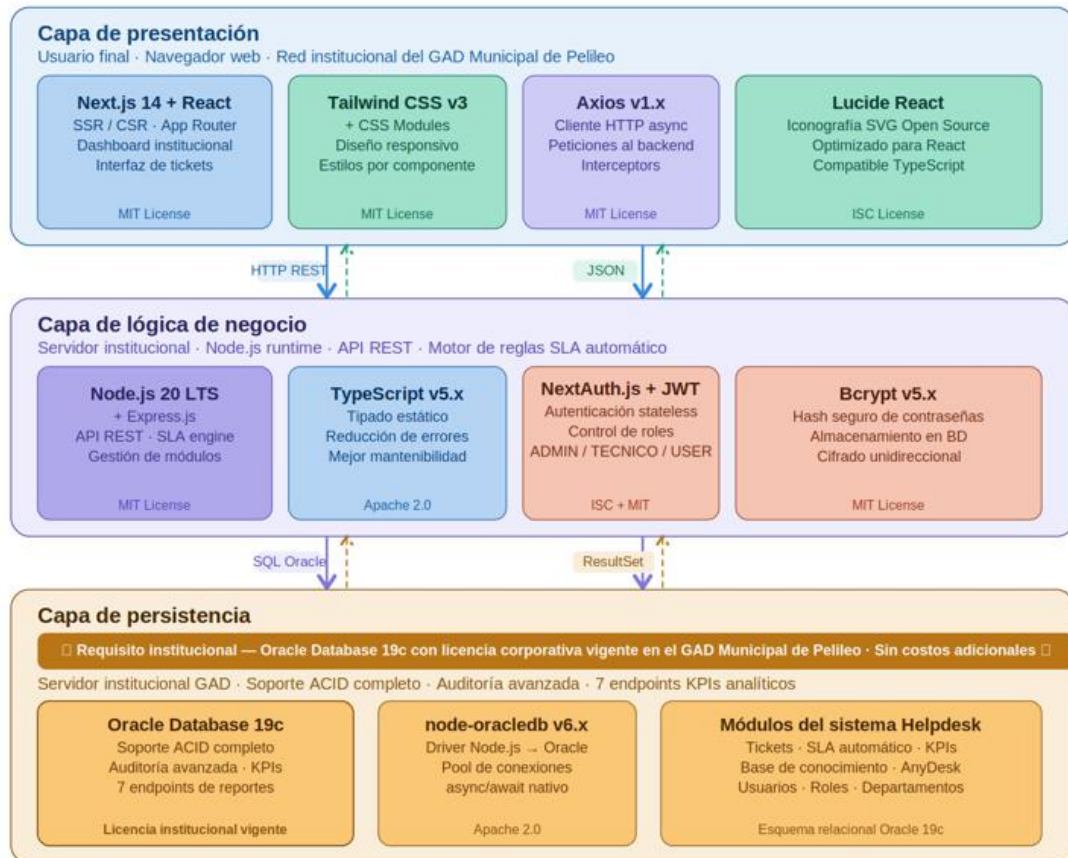


Figura 13. Entorno tecnológico del sistema Helpdesk desarrollado para el GAD Municipal de Pelileo.

La Figura 13 ilustra el entorno tecnológico completo del sistema Helpdesk desarrollado para el GAD Municipal de Pelileo, evidenciando la integración coherente entre las tres capas de la arquitectura propuesta. En la capa de presentación, Next.js 14 con React gestiona el renderizado híbrido SSR/CSR de la interfaz web, apoyado por Tailwind CSS y CSS Modules para el diseño responsivo, Lucide React para la iconografía SVG y Axios para las peticiones HTTP asíncronas hacia el backend. En la capa de lógica de negocio, Node.js 20 LTS con Express.js expone la API REST desarrollada en TypeScript, incorporando NextAuth.js con JWT para la autenticación stateless por roles y Bcrypt para el cifrado seguro de contraseñas. En la capa de persistencia, Oracle Database 19c centraliza el almacenamiento relacional con soporte

ACID completo, conectada al backend mediante el driver node-oracledb v6 con pool de conexiones y soporte async/await. La uniformidad del lenguaje JavaScript/TypeScript entre las capas de presentación y lógica de negocio, combinada con el aprovechamiento del licenciamiento institucional vigente de Oracle en el GAD Municipal de Pelileo, consolida un stack tecnológico coherente, sostenible y alineado con las directrices de soberanía tecnológica del Estado ecuatoriano.

3.3 Desarrollo

3.3.1 Proceso identificado para la asistencia técnica

A partir del análisis de los procesos actuales y los requerimientos identificados mediante los instrumentos de investigación, se diseñaron los procesos optimizados de gestión de incidencias soportados por el sistema Helpdesk desarrollado. Estos procesos adoptan las prácticas recomendadas para la gestión de incidencias, adaptadas a la realidad operativa del GAD Municipal de Pelileo.

En la Figura 14 se visualiza el diagrama de flujo del proceso optimizado implementado en el sistema Helpdesk, el cual establece un Punto Único de Contacto digital que elimina los canales informales de comunicación y garantiza la trazabilidad completa del ciclo de vida de cada incidencia.

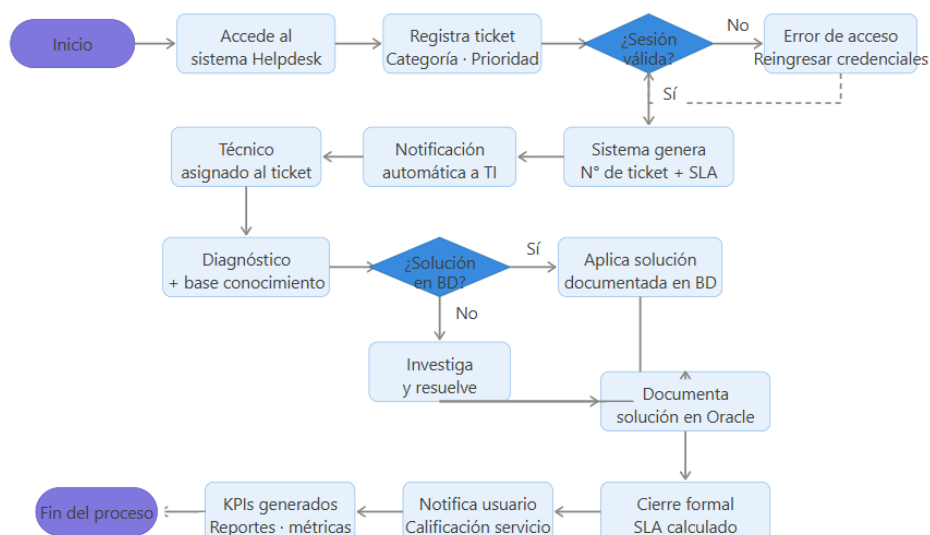


Figura 14. Proceso optimizado con sistema Helpdesk

Como se visualiza en la Figura 14, el proceso optimizado elimina la informalidad de los canales de comunicación y establece un Punto Único de Contacto digital accesible para todos los funcionarios desde cualquier equipo conectado a la red institucional, cumpliendo con el objetivo específico 3 planteado en la investigación.

En la Tabla 22 se describe detalladamente cada etapa del proceso optimizado, especificando las mejoras implementadas respecto al proceso informal identificado en el Capítulo II.

Tabla 23. Descripción del proceso optimizado mediante el sistema Helpdesk

Nº	Proceso optimizado	Descripción con el sistema Helpdesk	Mejora implementada
1	Registro digital del ticket	El funcionario accede al sistema web Helpdesk desde su equipo, inicia sesión con sus credenciales institucionales y completa el formulario de registro de ticket. El sistema asigna automáticamente un número único de seguimiento, categoriza la incidencia por tipo (hardware, software, red) y registra la hora y fecha del reporte.	Canal formal unificado (SPOC). Trazabilidad completa desde el primer contacto.
2	Notificación automática y asignación	El sistema envía una notificación automática al Departamento de TI informando sobre el nuevo ticket. El Jefe de TI o técnico con rol ADMIN asigna el ticket al técnico disponible según disponibilidad y especialización. El funcionario recibe confirmación de que su solicitud fue registrada.	Eliminación de solicitudes perdidas. Asignación transparente y documentada.
3	Diagnóstico asistido por base de conocimiento	El técnico accede al detalle del ticket y consulta la base de conocimiento institucional para verificar si existen soluciones documentadas a incidencias similares previas. El técnico categoriza y prioriza la incidencia según el impacto (número de usuarios afectados) y la urgencia.	Reducción del tiempo de diagnóstico. Disminución de reincidencias mediante conocimiento acumulado.
4	Resolución con documentación	El técnico resuelve la incidencia (presencial o mediante acceso remoto con AnyDesk) y documenta en el ticket: solución aplicada, tiempo dedicado, estado de resolución y observaciones técnicas relevantes.	Generación de historial de soluciones. Insumo directo para la base de conocimiento.
5	Cierre formal con SLA	Al resolver la incidencia, el sistema calcula automáticamente si se cumplió el Acuerdo de Nivel de Servicio (SLA) según el tipo de incidencia. El funcionario recibe notificación de cierre e invitación para calificar la atención recibida.	Medición objetiva de tiempos. Activación del mecanismo de satisfacción del usuario.
6	Seguimiento y métricas	El sistema registra todas las métricas de desempeño: tiempo de respuesta, tiempo de resolución, tasa de resolución en primer contacto y calificación del usuario. El módulo de reportes	Transparencia total del servicio. Mejora continua basada en datos reales.

		genera KPIs analíticos para la toma de decisiones del jefe del Departamento de TI.	
7	Base de conocimiento y AnyDesk	Las soluciones documentadas se incorporan a la base de conocimiento. El módulo AnyDesk centraliza los códigos de acceso remoto por departamento, eliminando demoras en la obtención de credenciales de acceso.	Autoservicio del usuario. Reducción de tiempo en atención remota (nuevo módulo).

La comparación entre los procesos actuales y los procesos optimizados evidencia una mejora sustancial en todas las fases del ciclo de vida de la incidencia. Al igual que lo identificado por Osman y Jimu [8] en su implementación de Helpdesk para un municipio, la sustitución de canales informales por una plataforma centralizada permite automatizar los procesos administrativos y generar una base de datos histórica que facilita la toma de decisiones. Asimismo, el estudio de Faria [11] para la ESPOL confirma que la trazabilidad proporcionada por un sistema de mesa de ayuda es el factor diferencial más significativo en la percepción de calidad del servicio técnico.

En la Figura 15 se resume el flujo completo del sistema mediante el diagrama BPMN (Business Process Model and Notation) del proceso optimizado, donde se identifican los actores, los módulos del sistema y las interacciones entre las capas de la arquitectura propuesta.

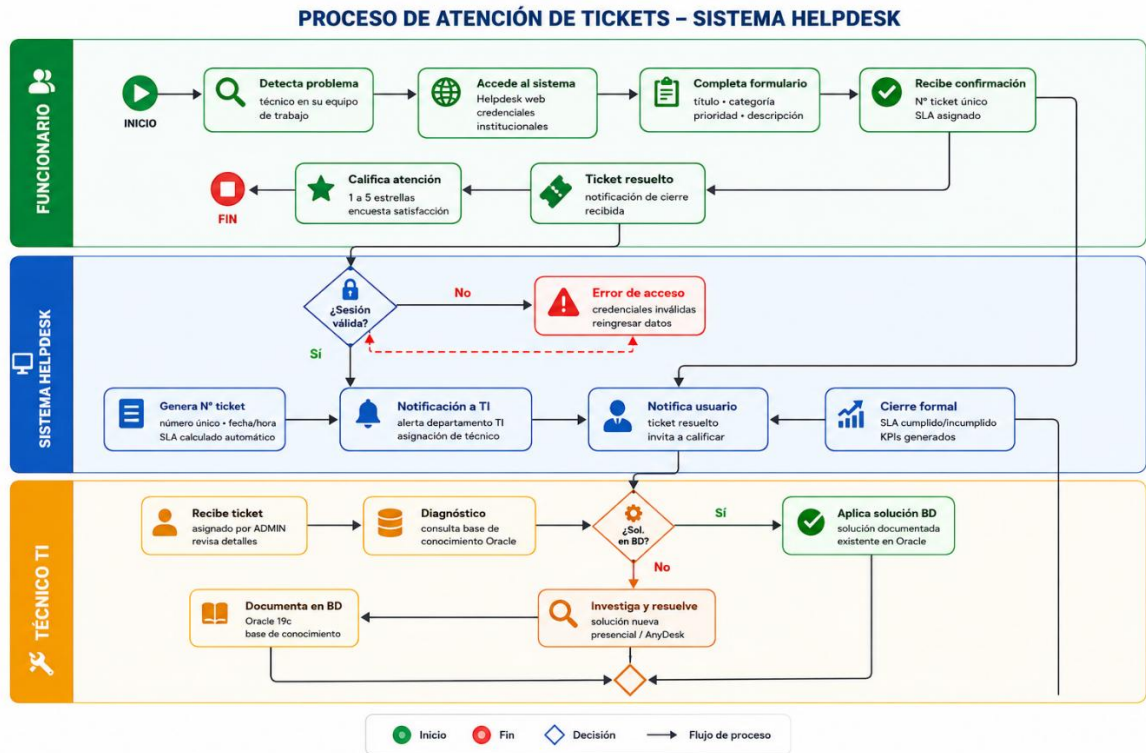


Figura 15. Diagrama BPMN del proceso optimizado

3.3.2 Metodología de desarrollo para aplicaciones web

Las metodologías de desarrollo ágil son esenciales al momento de organizar proyectos de software, ya que permiten implementar cambios de forma continua sin interrumpir el flujo de trabajo y reducen el tiempo de entrega mediante marcos de trabajo iterativos e incrementales. En un entorno de desarrollo de sistemas para la administración pública, donde los requerimientos evolucionan conforme el equipo institucional interactúa con el sistema, la adopción de metodologías ágiles se convierte en un factor crítico para garantizar la calidad del producto final y la satisfacción del cliente. Es entonces que, en la Tabla 23 se presenta un cuadro comparativo de las tres metodologías ágiles más utilizadas en el desarrollo de aplicaciones web como XP,

Scrum y Lean, el cual fundamentó la selección de la metodología más adecuada para el desarrollo del sistema Helpdesk del GAD Municipal de Pelileo.

Tabla 24. Comparativa de metodologías de desarrollo ágil

Criterio de comparación	XP (Extreme Programming)	Scrum	Lean
Tamaño del equipo	Equipos pequeños, menores a 10 personas. Ideal para desarrollo individual o en pares.	Múltiples equipos menores de 10 personas. Requiere roles diferenciados.	Variable, depende del proyecto. Sin restricción de tamaño.
Tamaño del proyecto	Proyectos pequeños y medianos.	Proyectos pequeños, medianos y grandes.	Proyectos pequeños, medianos y grandes.
Complejidad	Baja a media. Orientada a proyectos con requisitos cambiantes.	Alta. Diseñada para gestionar proyectos de gran complejidad.	Media. Aplica a cualquier complejidad sin estructura predefinida.
Metodología base	Retroalimentación continua con el cliente en cada iteración (1–2 semanas).	Sprints de 2–4 semanas con revisión al finalizar cada sprint.	Flujo de trabajo continuo. No posee fases ni ciclos predefinidos.
Roles	Programador, Cliente, Tester, Gestor.	Product Owner, Scrum Master, Equipo de desarrollo.	Gerente del proyecto, Programadores, Equipo de soporte.
Entregas	Cada iteración es puesta en producción según criterio del tester y validación del cliente.	Al finalizar cada Sprint se entrega el avance al Product Owner.	Entrega frecuente basada en métricas de flujo definidas.
Fases	Exploración, Planificación, Diseño, Codificación, Pruebas.	Sprint con 5 subfases: Planificación, Desarrollo, Scrum Diario, Revisión, Retrospectiva.	No posee fases predefinidas. Flujo continuo sin iteraciones formales.
Comunicación	Presencia activa del cliente/usuario en todo el ciclo de desarrollo.	Reunión diaria del equipo (Daily Scrum). Cliente presente en revisiones de sprint.	Presencia continua del cliente/usuario. Comunicación basada en métricas de desempeño.

La metodología que se seleccionó para el desarrollo del proyecto fue XP (Extreme Programing) debido a que al ser una metodología menos compleja de aplicar y estar enfocada hacia proyectos pequeños, fue la más adecuada de entre las metodologías más utilizadas. XP ofrece la flexibilidad necesaria para la implementación de cambios inesperados, además de, que el ciclo de desarrollo que presenta permite realizar ajustes rápidos y pruebas continuas garantizando la calidad del producto.

Fases de la metodología XP aplicadas al sistema Helpdesk

La metodología XP estructura el desarrollo del sistema Helpdesk en cinco fases secuenciales e iterativas, cada una con entregables concretos que garantizan la trazabilidad del proceso de desarrollo y la validación continua con el Departamento de TI del GAD Municipal de Pelileo. En la Tabla 24 se describe cada fase con su aplicación específica al proyecto:

Tabla 25. Fases de la metodología XP y su aplicación en el sistema Helpdesk

Nº	Fase	Descripción general	Aplicación en el sistema Helpdesk — GAD Municipal de Pelileo
1	Exploración	Se recopilan los requisitos del proyecto y se exploran múltiples soluciones, identificando los objetivos iniciales, definiendo el alcance y las iteraciones del sistema. Permite traducir los requisitos en tareas específicas y recopilar las necesidades del cliente.	Se levantaron los requisitos del sistema Helpdesk mediante encuestas al personal del GAD Municipal de Pelileo, entrevistas al Departamento de TI y fichas de observación del proceso actual de gestión de incidencias, conforme se detalla en el Capítulo II.
2	Planificación	Se identifican y priorizan los requisitos del proyecto mediante historias de usuario, guiando el desarrollo y estableciendo tiempos de desarrollo aproximados para cada módulo del sistema.	Se elaboraron las historias de usuario del sistema Helpdesk (Tabla 20), estimando un esfuerzo total de 224 horas distribuidas en los módulos de tickets, SLA, base de conocimiento, reportes, conforme a las necesidades identificadas en el GAD Municipal de Pelileo.
3	Diseño	Se realizan los bosquejos del sistema optando por diseños simples que cumplan con los requisitos y permitan modificaciones fáciles ante cambios. El objetivo es identificar riesgos y refactorizar la estructura del proyecto de ser necesario.	Se diseñó la arquitectura de tres capas desacopladas del sistema Helpdesk (Figura 14), el esquema relacional en Oracle Database 19c con todas las entidades del sistema y el proceso optimizado de gestión de incidencias. (Figura 12).
4	Codificación	Los desarrolladores trabajan en la escritura del código, facilitando la revisión mutua y mejorando la calidad del producto. El cliente se encuentra presente en el proceso para sugerir cambios en base a lo codificado.	El sistema Helpdesk se desarrolló con el stack Next.js 14 + Node.js 20 LTS + Oracle Database 19c bajo TypeScript, con participación activa del Jefe de TI del GAD Municipal de Pelileo en cada iteración para validar la funcionalidad de los módulos implementados.
5	Pruebas	Se realizan las pruebas del software, donde el cliente otorga retroalimentación de cada iteración culminada. Las pruebas de aceptación validan que cada historia de usuario cumple con los criterios definidos en la fase de planificación.	Se ejecutaron pruebas de aceptación por cada historia de usuario del sistema Helpdesk, validadas con el personal del Departamento de TI del GAD Municipal de Pelileo, verificando el correcto funcionamiento del registro de tickets, cálculo de SLA, módulo de reportes y base de conocimiento.

Descripción detallada de cada fase

1. Exploración: En la fase de exploración se recopilaban los requisitos del sistema Helpdesk directamente en el contexto real del GAD Municipal de Pelileo, mediante los tres instrumentos de investigación descritos en el Capítulo II: encuesta al personal administrativo, entrevistas y fichas de observación del proceso actual de gestión de incidencias. Esta fase permitió identificar las deficiencias operativas del proceso informal vigente, definir el alcance del sistema registro de tickets, categorización, priorización y establecer los tres roles de usuario del sistema: ADMIN, TECNICO y USUARIO.

2. Planificación: En la fase de planificación se identificaron y priorizaron los requisitos del sistema mediante historias de usuario. Las prioridades fueron definidas en conjunto con el Jefe del Departamento de TI del GAD Municipal de Pelileo, asignando mayor prioridad a los módulos de registro de tickets, gestión de usuarios y reportes analíticos, por ser los que abordan directamente los riesgos operativos identificados en el análisis del Capítulo II.

3. Diseño: En la fase de diseño se elaboraron los bosquejos del sistema optando por diseños simples y modulares que permiten modificaciones ágiles ante cambios de requerimientos. Se diseñaron:

La arquitectura de tres capas desacopladas del sistema (Figura 12), separando la capa de presentación (Next.js 14 + React), la capa de lógica de negocio (Node.js 20 LTS + Express.js) y la capa de persistencia (Oracle Database 19c).

4. Codificación: En la fase de codificación se implementaron los módulos del sistema Helpdesk utilizando el stack tecnológico seleccionado. La autenticación se implementó con roles diferenciados, y el cifrado de contraseñas. El Jefe de TI del GAD Municipal de Pelileo participó activamente en cada iteración, validando la funcionalidad de los módulos implementados y sugiriendo ajustes basados en el flujo de trabajo real del Departamento de TI.

5. Pruebas: En la fase de pruebas se ejecutaron pruebas de aceptación por cada historia de usuario del sistema Helpdesk, validadas con el personal del Departamento

de TI del GAD Municipal de Pelileo. Cada prueba verificó que los criterios de aceptación definidos en la fase de planificación se cumplieran correctamente, incluyendo el registro y seguimiento de tickets con número único, el cálculo automático del SLA por tipo de incidencia, la generación de reportes analíticos con KPIs institucionales, el funcionamiento de la base de conocimiento con búsqueda full-text y la correcta gestión de roles y permisos de acceso.

3.3.3 Desarrollo de la propuesta

El presente capítulo documenta el desarrollo del sistema web Helpdesk para el GAD Municipal de Pelileo siguiendo las cinco fases de la metodología XP (Extreme Programming): Exploración, Planificación, Diseño, Codificación y Pruebas. Cada fase se ejecutó de manera iterativa e incremental, con retroalimentación continua del Departamento de Tecnologías de la Información de la institución, garantizando que el sistema desarrollado responda directamente a las necesidades identificadas en el Capítulo II.

- ***Fase 1: Exploración***

En la fase de Exploración se recopiló la información necesaria para definir los requerimientos del sistema Helpdesk, partiendo de los datos obtenidos en el Capítulo II mediante encuestas, entrevistas y fichas de observación aplicadas al personal del GAD Municipal de Pelileo. Adicionalmente, se identificaron las herramientas tecnológicas Open Source seleccionadas en la sección 3.2 y se estableció la arquitectura de tres capas desacopladas del sistema (Figura 13).

Con base en los hallazgos del diagnóstico y el proceso optimizado diseñado en la Figura 13, se identificaron tres roles de usuario asociados al proceso de soporte técnico del GAD Municipal de Pelileo, los cuales se presentan en la Tabla 25.

Tabla 26. Roles y permisos del sistema Helpdesk

Rol	Acciones y permisos
ADMIN	• Acceso total al sistema. Gestión de usuarios (crear, editar, activar/desactivar, reset contraseña). • Configuración de categorías, prioridades y tiempos de SLA. • Visualización y gestión de todos los tickets del sistema. • Edición exclusiva de códigos AnyDesk por departamento. • Generación de reportes analíticos con 7 KPIs y exportación CSV.
TÉCNICO	• Visualización y gestión de tickets asignados. • Creación y edición de tickets. Actualización de estado y documentación de soluciones. • Incorporación y consulta de artículos en la base de conocimiento. • Consulta de códigos AnyDesk de todos los departamentos.
USUARIO	• Creación y seguimiento de sus propios tickets con número único de rastreo. • Reapertura de tickets cerrados no resueltos satisfactoriamente. • Consulta de la base de conocimiento institucional. • Visualización de códigos AnyDesk de todos los departamentos. • Calificación de la atención recibida al cierre del ticket (1–5 estrellas).

Historias de usuario

Una historia de usuario es un requerimiento expresado desde la perspectiva del usuario final, redactado de forma clara y no técnica, que describe lo que el sistema debe realizar para satisfacer una necesidad del negocio. Su principal objetivo es establecer los criterios de aceptación que permiten verificar la correcta implementación de cada funcionalidad en las iteraciones de desarrollo [21].

Las historias de usuario del sistema Helpdesk fueron definidas a partir de los requerimientos identificados en el Capítulo II, priorizadas en conjunto con el Jefe del Departamento de TI del GAD Municipal de Pelileo, y se presentan a continuación:

Tabla 27. Historia de usuario 1 — acceso al sistema

Historia de usuario			#1
Usuario:	Todos		
Nombre:	Acceso al sistema		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta

Puntos estimados:	1	Iteración asignada:	1
Programador a cargo:	Josué Valverde		
Descripción:	El usuario accede al sistema Helpdesk con sus credenciales institucionales mediante autenticación JWT.		
Observación:	—		

Tabla 28. Historia de usuario 2 — recuperar contraseña

Historia de usuario			#2
Usuario:	Todos		
Nombre:	Recuperar contraseña		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	1
Programador a cargo:	Josué Valverde		
Descripción:	Todos los usuarios pueden solicitar el restablecimiento de contraseña mediante correo institucional.		
Observación:	El usuario debe tener una cuenta activa en el sistema para poder solicitar el cambio de contraseña.		

Tabla 29. Historia de usuario 3 — gestión de usuarios

Historia de usuario			#3
Usuario:	ADMIN		
Nombre:	Gestión de usuarios		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	1
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN puede crear, editar, activar y desactivar usuarios del sistema Helpdesk.		
Observación:	—		

Tabla 30. Historia de usuario 4 — Gestión de roles

Historia de usuario			#4
Usuario:	ADMIN		
Nombre:	Gestión de roles		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	1
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN asigna roles diferenciados (ADMIN, TÉCNICO, USUARIO) a cada cuenta del sistema.		
Observación:	—		

Tabla 31. Historia de usuario 5 — gestión de categorías y prioridades

Historia de usuario			#5
Usuario:	ADMIN		
Nombre:	Gestión de categorías y prioridades		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	2
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN puede crear, editar y deshabilitar categorías e incidencias (Hardware, Software, Red) y niveles de prioridad (Alta, Media, Baja).		
Observación:	—		

Tabla 32. Historia de usuario 6 — gestión de departamentos

HISTORIA DE USUARIO			#6
Usuario:	ADMIN		
Nombre:	Gestión de departamentos		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	2
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN puede crear, editar y deshabilitar los departamentos del GAD Municipal de Pelileo asociados a los usuarios del sistema.		
Observación:	—		

Tabla 33. Historia de usuario 7 — configuración de SLA

Historia de usuario			#7
Usuario:	ADMIN		
Nombre:	Configuración de SLA		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	2
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN configura los tiempos máximos de resolución por categoría y prioridad que el sistema utilizará para calcular automáticamente el cumplimiento del SLA.		
Observación:	Los tiempos de SLA definidos se aplican automáticamente al crear cada ticket.		

Tabla 34. Historia de usuario 8 — cerrar sesión

Historia de usuario			#8
Usuario:	Todos		
Nombre:	Cerrar sesión		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Baja
Puntos estimados:	1	Iteración asignada:	2
Programador a cargo:	Josué Valverde		
Descripción:	Todos los roles pueden cerrar sesión desde el menú principal del sistema.		
Observación:	—		

Tabla 35. Historia de usuario 9 — crear ticket

Historia de usuario			#9
Usuario:	Todos		
Nombre:	Crear ticket		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Programador a cargo:	Josué Valverde		
Descripción:	Todos los usuarios pueden registrar una incidencia técnica completando el formulario de ticket con título, categoría, prioridad y descripción. El sistema asigna número único y SLA automáticamente.		
Observación:	Al crear el ticket, el sistema genera automáticamente el número de seguimiento, calcula el SLA y notifica al Departamento de TI.		

Tabla 36. Historia de usuario 10 — gestión de tickets

Historia de usuario			#10
Usuario:	ADMIN / TECNICO		
Nombre:	Gestión de tickets		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Programador a cargo:	Josué Valverde		
Descripción:	Los roles ADMIN y TÉCNICO pueden visualizar, editar, asignar y actualizar el estado de todos los tickets registrados en el sistema.		
Observación:	El estado del ticket cambia automáticamente al registrar una actualización.		

Tabla 37. Historia de usuario 11 — asignación de tickets

Historia de usuario			#11
Usuario:	ADMIN / TECNICO		
Nombre:	Asignación de tickets		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Programador a cargo:	Josué Valverde		
Descripción:	El ADMIN asigna los tickets al técnico disponible según disponibilidad y especialización. El TECNICO puede reasignar si es necesario.		
Observación:	—		

Tabla 38. Historia de usuario 12 — reapertura de ticket

Historia de usuario			#12
Usuario:	USUARIO		
Nombre:	Reapertura de ticket		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	4
Programador a cargo:	Josué Valverde		
Descripción:	El USUARIO puede reabrir un ticket cerrado si considera que el problema no fue resuelto satisfactoriamente.		
Observación:	El ticket debe estar en estado 'Cerrado' para poder ser reabierto.		

Tabla 39. Historia de usuario 13 — cálculo automático de SLA

Historia de usuario			#13
Usuario:	Sistema		
Nombre:	Cálculo automático de SLA		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	4
Programador a cargo:	Josué Valverde		
Descripción:	El sistema calcula automáticamente si cada ticket cumple el SLA configurado y genera alertas cuando el tiempo de resolución se aproxima al límite establecido.		
Observación:	El SLA se calcula desde el momento del registro del ticket hasta su cierre formal.		

Tabla 40. Historia de usuario 14 — base de conocimiento

Historia de usuario			#14
Usuario:	Todos		
Nombre:	Base de conocimiento		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	3	Iteración asignada:	4
Programador a cargo:	Josué Valverde		
Descripción:	Los roles ADMIN y TÉCNICO pueden crear y editar artículos en la base de conocimiento con soporte Markdown. Todos los roles pueden consultar y buscar artículos por palabras clave.		
Observación:	—		

Tabla 41. Historia de usuario 15 — módulo AnyDesk por departamento

Historia de usuario			#15
Usuario:	ADMIN / TECNICO		
Nombre:	Módulo AnyDesk por departamento		
Prioridad en el negocio:	Media	Riesgo de desarrollo:	Media
Puntos estimados:	2	Iteración asignada:	5
Programador a cargo:	Josué Valverde		
Descripción:	El sistema centraliza los códigos AnyDesk por departamento del GAD Municipal de Pelileo. El ADMIN puede editar los códigos; todos los roles pueden consultarlos y copiarlos al portapapeles.		
Observación:	Módulo adicional diseñado para optimizar los tiempos de asistencia remota del equipo técnico.		

Tabla 42. Historia de usuario 16 — notificaciones automáticas

Historia de usuario			#16
Usuario:	Sistema		
Nombre:	Notificaciones automáticas		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	5
Programador a cargo:	Josué Valverde		
Descripción:	El sistema envía notificaciones automáticas al usuario y al técnico asignado al crear, actualizar y cerrar cada ticket.		
Observación:	—		

Tabla 43. Historia de usuario 17 — reportes y métricas KPIs

Historia de usuario			#17
Usuario:	Todos		
Nombre:	Reportes y métricas KPIs		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	3	Iteración asignada:	5
Programador a cargo:	Josué Valverde		
Descripción:	Todos los roles pueden visualizar reportes analíticos del sistema: tiempo promedio de respuesta, tasa de resolución, cumplimiento de SLA, tickets por categoría y satisfacción del usuario. El ADMIN puede exportar en formato CSV.		
Observación:	7 endpoints analíticos desarrollados sobre Oracle Database 19c.		

Tabla 44. Historia de usuario 18 — calificación del servicio

Historia de usuario			#18
Usuario:	USUARIO		
Nombre:	Calificación del servicio		
Prioridad en el negocio:	Media	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	5
Programador a cargo:	Josué Valverde		
Descripción:	Al cerrar un ticket, el USUARIO recibe una notificación con invitación a calificar la atención recibida en escala de 1 a 5 estrellas.		
Observación:	—		

- **Fase 2: Planificación del proyecto**

Tras la obtención de las historias de usuario y los roles definidos en la Fase 1, se procedió a realizar la planificación del proyecto mediante la estimación del esfuerzo requerido para cada historia de usuario. Se definieron 8 horas de trabajo diario destinadas al desarrollo del sistema, identificando los requerimientos que mayor tiempo de implementación demandan [24][16].

a. Estimación de historias de usuario

Tabla 45. Historias de usuario y estimación de esfuerzo

Nº	Historia de usuario	Rol	Días	Horas
—	Base de datos (esquema Oracle 19c)	Todos	3	24
1	Acceso al sistema	Todos	1	8
2	Recuperar contraseña	ADMIN	1	8
3	Gestión de usuarios	ADMIN	1	8
4	Gestión de roles	ADMIN	1	8
5	Gestión de categorías y prioridades	ADMIN	1	8
6	Gestión de departamentos	ADMIN	1	8
7	Configuración de SLA	Todos	1	8
8	Cerrar sesión	Todos	1	8
9	Crear ticket	ADMIN/TECNICO	2	16
10	Gestión de tickets	ADMIN/TECNICO	2	16
11	Asignación de tickets	USUARIO	2	16
12	Reapertura de ticket	Sistema	1	8
13	Cálculo automático de SLA	Todos	2	16
14	Base de conocimiento	Todos	3	24
15	Módulo AnyDesk	Sistema	2	16
16	Notificaciones automáticas	Todos	2	16
17	Reportes y métricas KPIs	USUARIO	4	32
18	Calificación del servicio	Sistema	1	8
	TOTAL		28	224

b. Plan de entrega

Una vez definida la estimación de las historias de usuario, se elaboró el plan de entrega que establece el tiempo total requerido para el desarrollo del sistema Helpdesk, organizado en cinco iteraciones. El tiempo se define en semanas, días y horas de trabajo, considerando 8 horas laborables por día.

Tabla 46. Plan de entrega del sistema Helpdesk

Iteración	Nº	Historia de usuario	Semanas	Días	Horas
Iteración 1	—	Base de datos (esquema Oracle 19c)	0.6	3	24
	1	Acceso al sistema	0.2	1	8
	2	Recuperar contraseña	0.2	1	8
	3	Gestión de usuarios	0.2	1	8
	4	Gestión de roles	0.2	1	8
Iteración 2	5	Gestión de categorías y prioridades	0.2	1	8
	6	Gestión de departamentos	0.2	1	8
	7	Configuración de SLA	0.2	1	8
	8	Cerrar sesión	0.2	1	8
Iteración 3	9	Crear ticket	0.4	2	16
	10	Gestión de tickets	0.4	2	16
	11	Asignación de tickets	0.4	2	16
Iteración 4	12	Reapertura de ticket	0.2	1	8
	13	Cálculo automático de SLA	0.4	2	16
	14	Base de conocimiento	0.6	3	24
Iteración 5	15	Módulo AnyDesk	0.4	2	16
	16	Notificaciones automáticas	0.4	2	16
	17	Reportes y métricas KPIs	0.8	4	32
	18	Calificación del servicio	0.2	1	8
		TOTAL	5.6	28	224

c. Equipo de trabajo

Tabla 47. Equipo de trabajo del proyecto

Rol	Responsable
Programador	Josué Israel Valverde Endara
Cliente	Personal del GAD Municipal de Pelileo (Departamento de TI)
Tester	Ing. Dennis Vinicio Chicaiza Castillo, Mg. (Tutor)
Coach	Ing. Dennis Vinicio Chicaiza Castillo, Mg.
Seguimiento	Ing. Dennis Vinicio Chicaiza Castillo, Mg. — Josué Israel Valverde Endara

- **Fase 3: Diseño**

a. Tarjetas CRC

Las tarjetas CRC (Clase-Responsabilidad-Colaboración) son una herramienta de diseño de software que permite organizar y representar las clases del sistema, sus responsabilidades funcionales y sus colaboraciones con otras capas de la arquitectura. En XP, las tarjetas CRC facilitan la comunicación entre el desarrollador y el cliente al expresar el diseño del sistema en términos comprensibles sin requerir conocimientos técnicos. [21][22].

Se elaboraron las tarjetas CRC del sistema Helpdesk a partir de las historias de usuario definidas en la Fase 1, representando cada módulo del sistema con sus responsabilidades en las capas de datos (Oracle Database 19c), lógica de negocio (Node.js + Express.js) y presentación (Next.js + React):

Tabla 48. Tarjeta CRC — acceso al sistema

Acceso al sistema	
Responsabilidad	Colaboradores
• Obtener usuario y contraseña • Validar credenciales institucionales • Generar token JWT de sesión	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 49. Tarjeta CRC — recuperar contraseña

Recuperar contraseña	
Responsabilidad	Colaboradores
• Obtener y validar usuario • Enviar correo de restablecimiento • Guardar nueva contraseña cifrada	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Bcrypt) • Capa de presentación (Next.js)
Observación:	—

Tabla 50. Tarjeta CRC — gestión de usuarios

Gestión de usuarios	
Responsabilidad	Colaboradores
• Obtener parámetros del usuario • Validar datos institucionales • Guardar / actualizar usuario en BD	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 51. Tarjeta CRC — gestión de roles

Gestión de roles	
Responsabilidad	Colaboradores
• Obtener roles definidos • Asignar rol al usuario • Validar permisos por rol	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 52. Tarjeta CRC — gestión de categorías y prioridades

Gestión de categorías y prioridades	
Responsabilidad	Colaboradores
• Obtener parámetros • Validar parámetros • Guardar / deshabilitar categoría o prioridad	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 53. Tarjeta CRC — gestión de departamentos

Gestión de departamentos	
Responsabilidad	Colaboradores
• Obtener parámetros del departamento • Guardar / deshabilitar departamento • Asociar usuarios al departamento	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 54. Tarjeta CRC — configuración de SLA

Configuración de SLA	
Responsabilidad	Colaboradores
• Obtener tiempos por categoría/prioridad • Guardar configuración de SLA • Calcular SLA automáticamente al crear ticket	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • SLA engine (Node.js)
Observación:	—

Tabla 55. Tarjeta CRC — cerrar sesión

Cerrar sesión	
• Invalidar token JWT • Eliminar sesión activa • Redirigir a pantalla de login	• Capa de lógica de negocio (NextAuth.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 56. Tarjeta CRC — crear ticket

Crear ticket	
Responsabilidad	Colaboradores
• Obtener datos del formulario • Validar campos requeridos • Asignar número único y SLA • Guardar ticket en BD • Enviar notificación a TI	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • SLA engine • Sistema de notificaciones • Capa de presentación (Next.js)
Observación:	—

Tabla 57. Tarjeta CRC — gestión de tickets

Gestión de tickets	
Responsabilidad	Colaboradores
• Obtener listado de tickets • Filtrar por estado/categoría/prioridad • Actualizar estado del ticket • Documentar solución aplicada	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 58. Tarjeta CRC — asignación de tickets

Asignación de tickets	
Responsabilidad	Colaboradores
• Obtener técnicos disponibles • Asignar ticket a técnico • Registrar asignación en BD • Notificar al técnico asignado	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Sistema de notificaciones
Observación:	—

Tabla 59. Tarjeta CRC — reapertura de ticket

Reapertura de ticket	
Responsabilidad	Colaboradores
• Validar estado del ticket (Cerrado) • Cambiar estado a Abierto • Notificar al Departamento de TI	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Sistema de notificaciones
Observación:	—

Tabla 60. Tarjeta CRC — cálculo automático de SLA

Cálculo automático de SLA	
Responsabilidad	Colaboradores
• Obtener tiempo límite configurado • Calcular tiempo transcurrido • Generar alerta si se aproxima al límite • Registrar cumplimiento al cierre	• Capa de datos (Oracle 19c) • SLA engine (Node.js) • Sistema de notificaciones
Observación:	—

Tabla 61. Tarjeta CRC — base de conocimiento

Base de conocimiento	
Responsabilidad	Colaboradores
• Obtener artículos existentes • Crear / editar artículo con Markdown • Búsqueda full-text de artículos • Registrar contador de vistas	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)

Tabla 62. Tarjeta CRC — módulo AnyDesk

Módulo anydesk	
Responsabilidad	Colaboradores
• Obtener códigos por departamento • Editar código AnyDesk (solo ADMIN) • Copiar código al portapapeles	• Capa de datos (Oracle 19c) • Capa de presentación (Next.js)
Observación:	—

Tabla 63. Tarjeta CRC — notificaciones automáticas

Notificaciones automáticas	
Responsabilidad	Colaboradores
• Detectar evento del ticket (crear/actualizar/cerrar) • Enviar notificación al usuario • Enviar notificación al técnico asignado	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Sistema de correo institucional
Observación:	—

Tabla 64. Tarjeta CRC — reportes y métricas KPIs

Reportes y métricas KPIS	
Responsabilidad	Colaboradores
• Obtener parámetros de filtro (fechas, departamento) • Ejecutar consultas analíticas en Oracle 19c • Generar visualización de KPIs • Exportar datos en formato CSV	• Capa de datos (Oracle 19c — 7 endpoints analíticos) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)
Observación:	—

Tabla 65. Tarjeta CRC — calificación del servicio

Calificación del servicio	
Responsabilidad	Colaboradores
• Enviar invitación al cierre del ticket • Obtener calificación (1–5 estrellas) • Guardar calificación en BD	• Capa de datos (Oracle 19c) • Capa de lógica de negocio (Express.js) • Capa de presentación (Next.js)

Incluir calificación en KPIs de satisfacción	
Observación:	—

b. Diseño de la base de datos

La Figura 16 presenta el modelo Entidad-Relación del sistema Helpdesk, que define la estructura de la base de datos implementada en Oracle Database 19c. El esquema relacional incluye las entidades principales del sistema: hd_tickets, hd_usuarios, hd_adjuntos, hd_departamentos, hd_categorias, hd_historial_tickets, hd_comentarios, hd_notificaciones y hd_anydesk_departamentos, esta última constituyendo un módulo adicional diseñado específicamente para optimizar los tiempos de asistencia remota del Departamento de TI del GAD Municipal de Pelileo.

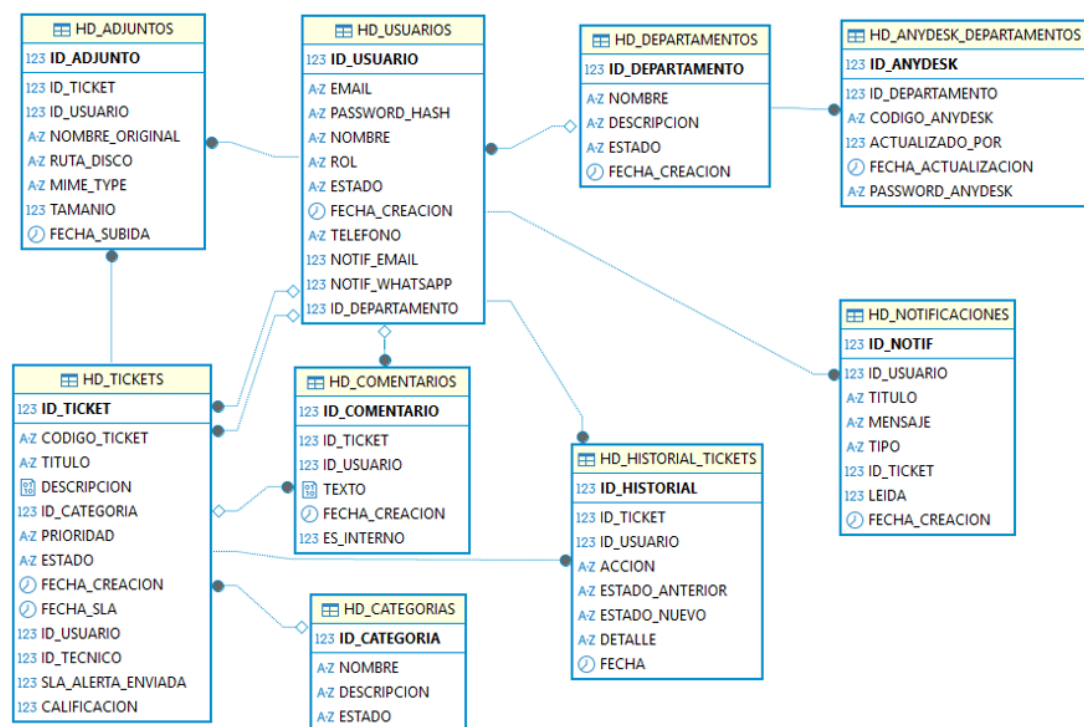


Figura 16. Modelo entidad-relación del sistema helpdesk

c. Diseño de la interacción

El diseño del prototipo de interfaz del sistema Helpdesk se elaboró utilizando la herramienta Figma en su versión gratuita, la cual permite diseñar interfaces web

responsivas con componentes reutilizables. El prototipo sirvió como referencia visual para las reuniones de validación con el Departamento de TI del GAD Municipal de Pelileo durante las iteraciones de desarrollo.

La Figura 16 presenta la pantalla de acceso al sistema, destinada a todos los roles (ADMIN, TÉCNICO, USUARIO), donde el usuario ingresa sus credenciales institucionales para autenticarse mediante JWT.

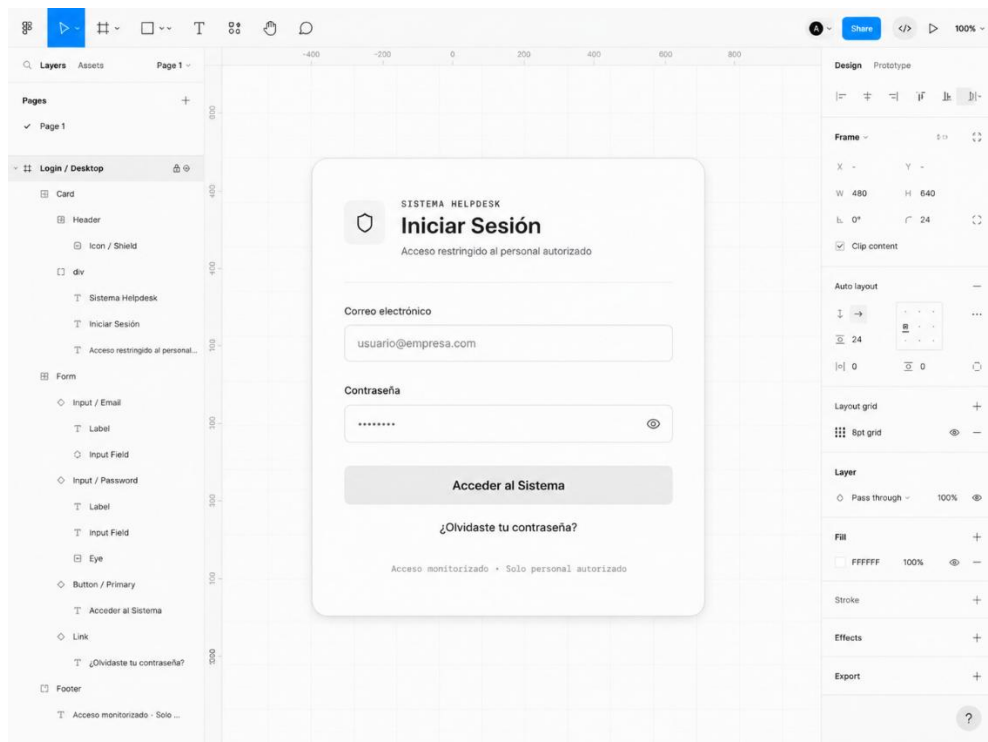


Figura 17. Prototipo — pantalla de acceso al sistema

La Figura 18 presenta el dashboard principal del sistema Helpdesk, visible para todos los roles tras el acceso exitoso. El dashboard muestra los KPIs institucionales más relevantes: tickets abiertos, en proceso y cerrados; tiempo promedio de respuesta; tasa de cumplimiento de SLA; y las incidencias recientes del GAD Municipal de Pelileo.

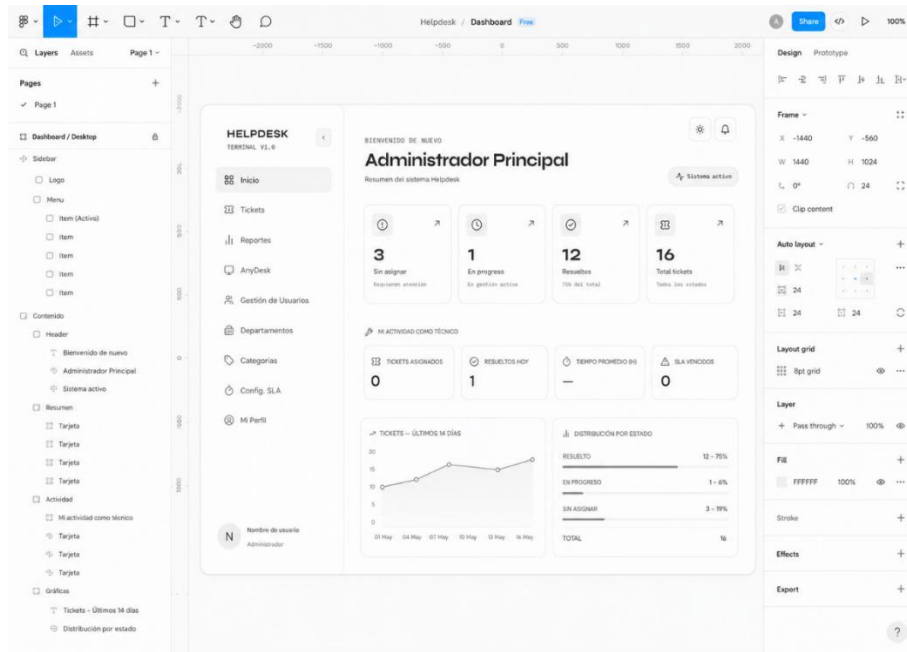


Figura 18. Prototipo — dashboard principal del sistema helpdesk

La Figura 19 presenta el módulo de gestión de tickets, accesible para los roles ADMIN y TÉCNICO. Este módulo muestra el listado completo de tickets con filtros por estado, categoría, prioridad y departamento, junto con las acciones de asignación, actualización de estado y documentación de la solución aplicada.

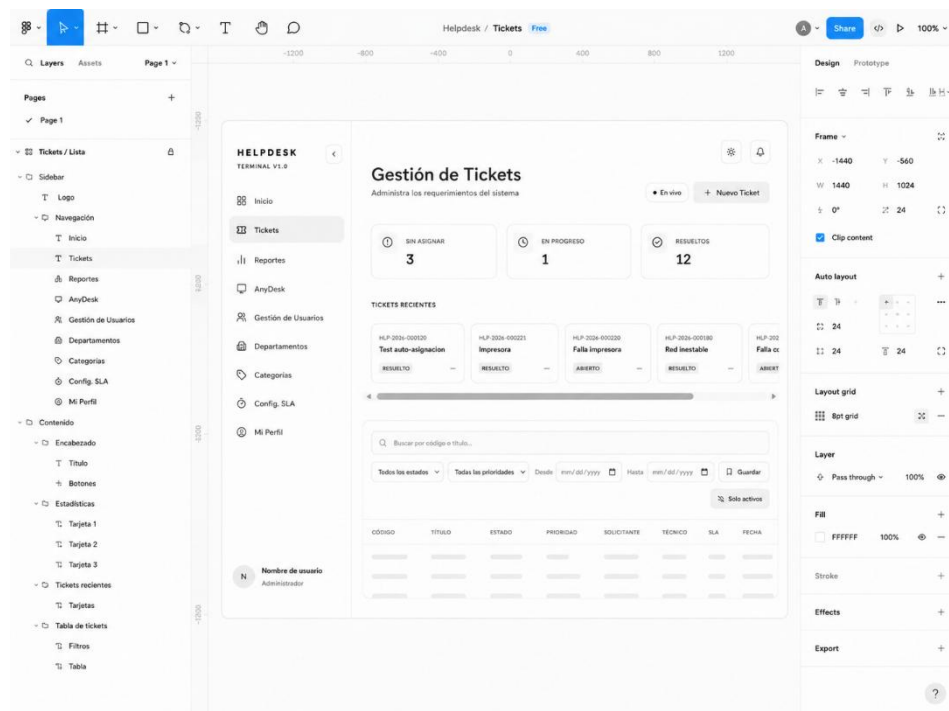


Figura 19. Prototipo — módulo de gestión de tickets

La Figura 20 presenta la vista de detalle de un ticket, accesible para todos los roles. Esta interfaz muestra la información completa del ticket: datos del solicitante, descripción del problema, categoría, prioridad, técnico asignado, historial de actualizaciones, estado del SLA y —exclusivamente para el USUARIO— el botón de calificación del servicio al momento del cierre.

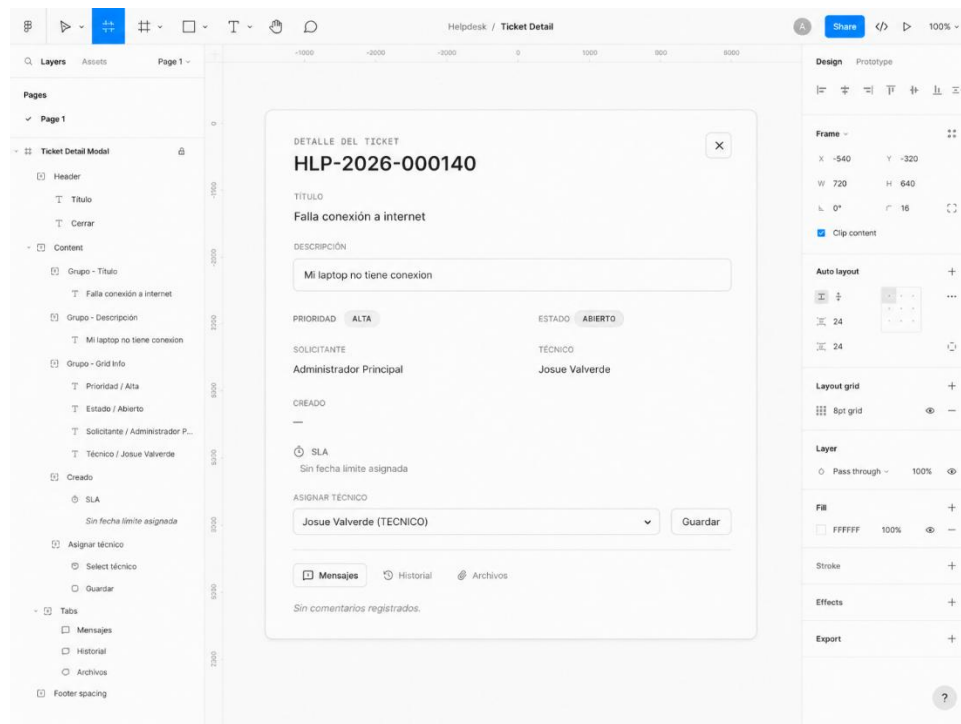


Figura 20. Prototipo — vista de detalle del ticket

La Figura 21 presenta el módulo de reportes y métricas KPIs, que genera visualizaciones analíticas a partir de los 7 endpoints desarrollados sobre Oracle Database 19c. Este módulo permite al Jefe de TI del GAD Municipal de Pelileo medir el desempeño del Departamento de TI mediante indicadores objetivos: tiempo promedio de respuesta, tasa de resolución en primer contacto, porcentaje de cumplimiento de SLA, volumen de tickets por categoría/departamento y calificación promedio del servicio.

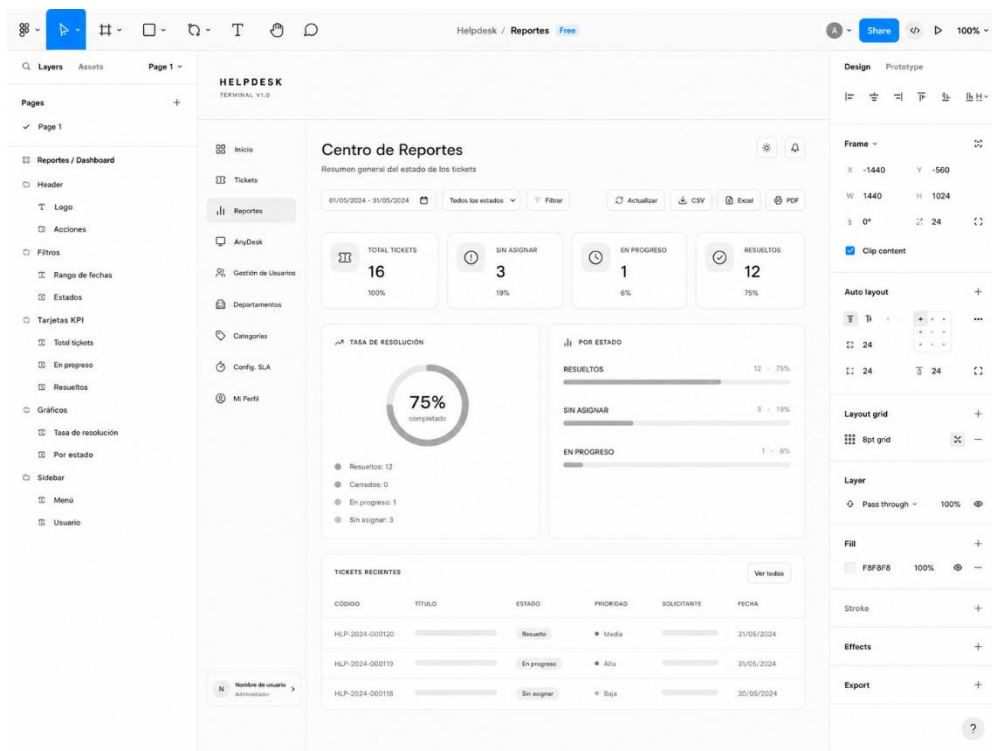


Figura 21. Prototipo — módulo de reportes y KPIs analíticos

La Figura 22 presenta el módulo de gestión de usuarios, accesible exclusivamente para el rol ADMIN del sistema Helpdesk del GAD Municipal de Pelileo. Este módulo centraliza el ciclo de vida completo de las cuentas institucionales, permitiendo al administrador crear nuevos usuarios con asignación de rol y departamento, editar los datos de cuentas existentes, activar o desactivar cuentas según la situación del funcionario, y restablecer contraseñas de forma segura mediante el módulo de recuperación. La interfaz presenta una tabla paginada con filtros por nombre, rol y estado, junto con modales de creación y edición que validan los datos en tiempo real antes de ejecutar la operación sobre la base de datos.

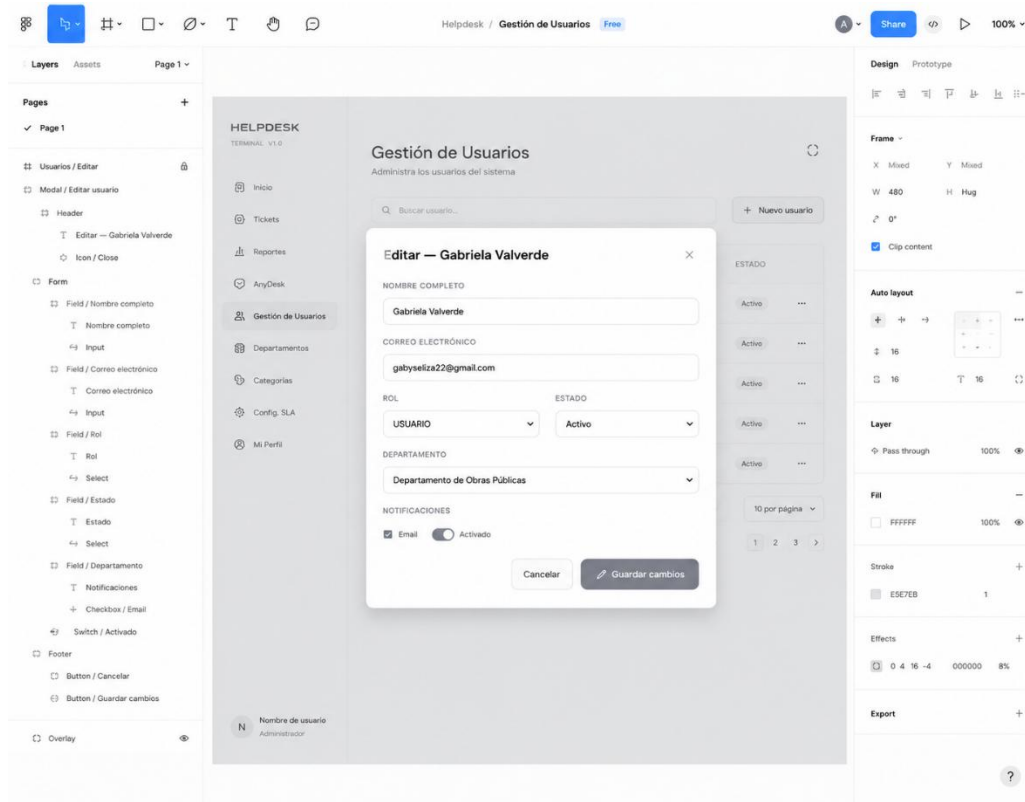


Figura 22. Prototipo — gestor de usuarios

El diseño de la interacción del sistema Helpdesk del GAD Municipal de Pelileo se desarrolló mediante prototipos de alta fidelidad elaborados en Figma, los cuales sirvieron como referencia visual durante las iteraciones de validación con el Departamento de TI. Los prototipos cubren el flujo completo del sistema: desde la pantalla de autenticación mediante JWT hasta el dashboard principal con KPIs institucionales, pasando por el módulo de gestión de tickets con filtros por estado, categoría, prioridad y departamento; la vista de detalle del ticket con historial de actualizaciones y estado del SLA; el módulo de reportes analíticos sobre Oracle Database 19c; y el gestor de usuarios con control de ciclo de vida de cuentas exclusivo para el rol ADMIN. En conjunto, los seis prototipos validan que la arquitectura de navegación del sistema responde a los roles definidos, garantizando que cada actor acceda únicamente a las funcionalidades que le corresponden según los permisos establecidos en la Tabla 25.

d. Fase 4: Codificación

- Iteración 1 — autenticación y configuración del sistema***

La primera iteración abarca el desarrollo de los módulos de autenticación y configuración base del sistema. A continuación, se presenta el resumen de las historias de usuario desarrolladas.

Tabla 66. Resumen de la iteración 1

Iteración	Nº	Historia de usuario
Iteración 1	—	Base de datos — Esquema relacional Oracle Database 19c
	1	Acceso al sistema
	2	Recuperar contraseña
	3	Gestión de usuarios
	4	Gestión de roles

Historia de usuario 1 — acceso al sistema

Tabla 67. Tarea 1 — acceso al sistema

Tarea			#1
Historia de usuario:	#1 — Acceso al sistema		
Usuario:	Todos		
Nombre:	Acceso al sistema		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	1
Fecha inicio:	12/01/2026	Fecha fin:	12/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El usuario accede al sistema Helpdesk con sus credenciales institucionales mediante autenticación JWT.		
Observación:	—		

El módulo de acceso al sistema implementa la autenticación mediante NextAuth.js con estrategia de credenciales y tokens JWT firmados. El usuario ingresa su nombre de usuario y contraseña; el backend verifica las credenciales contra la tabla HD_USUARIOS de Oracle Database 19c, compara la contraseña cifrada con Bcrypt y retorna el token JWT con el rol asignado al usuario. El frontend gestiona la sesión activa y renderiza el dashboard correspondiente según el rol autenticado.

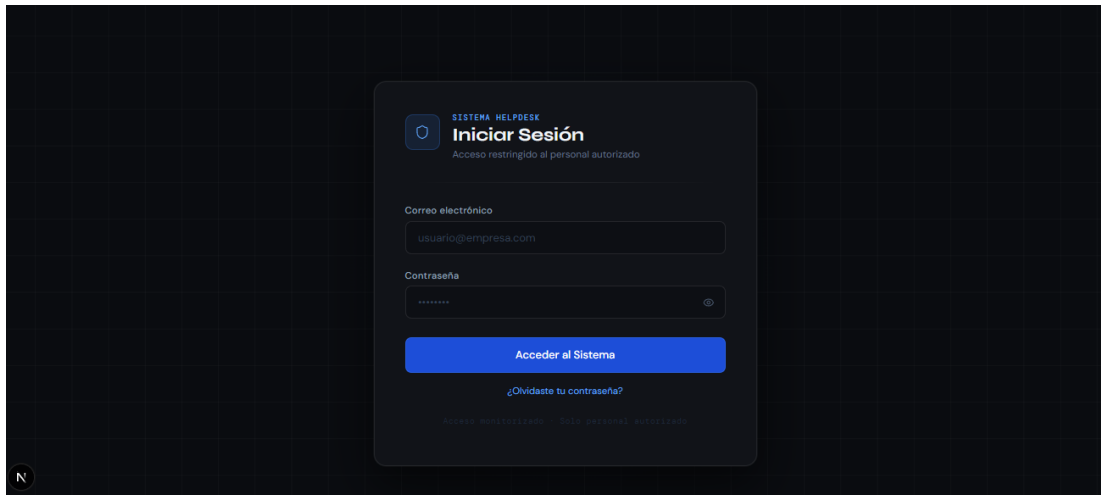


Figura 23. Pantalla de acceso al sistema

Código 1. Endpoint de autenticación — Backend Node.js + Express.js

```
1  'use client';
2
3  import { useState } from 'react';
4  import { useForm } from 'react-hook-form';
5  import { zodResolver } from '@hookform/resolvers/zod';
6  import { z } from 'zod';
7  import { useRouter } from 'next/navigation';
8  import Link from 'next/link';
9  import { Eye, EyeOff, Shield, AlertCircle, Loader2 } from 'lucide-react';
10 import { authService } from '@src/services/auth.service';
11 import styles from './page.module.css';
12
13 // — Validación Zod —
14 const schema = z.object({
15   email: z
16     .string()
17     .min(1, 'El correo es requerido')
18     .email('Correo electrónico inválido'),
19   password: z
20     .string()
21     .min(1, 'La contraseña es requerida')
22     .min(6, 'Mínimo 6 caracteres'),
23 });
24
```

```

25 type LoginForm = z.infer<typeof schema>;
26
27 // --- Componente ---
28 export default function LoginPage() {
29   const router = useRouter();
30   const [showPassword, setShowPassword] = useState(false);
31   const [serverError, setServerError] = useState('');
32
33   const {
34     register,
35     handleSubmit,
36     formState: { errors, isSubmitting },
37   } = useForm<LoginForm>({ resolver: zodResolver(schema) });
38
39   const onSubmit = async (data: LoginForm) => {
40     setServerError('');
41     try {
42       const result = await authService.login({ email: data.email, password: data.password });
43       if (result?.ok) {
44         router.push('/dashboard');
45       } else {
46         setServerError('Correo electrónico o contraseña incorrectos');
47       }
48     } catch {
49       setServerError('No se pudo conectar con el servidor. Intenta de nuevo.');
```

```

> src > controllers > TS auth.controller.ts > ...
export const login = async (req: Request, res: Response): Promise<void> => {
  const parsed = LoginSchema.safeParse(req.body);
  if (!parsed.success) {
    res.status(400).json({ error: parsed.error.issues[0].message });
    return;
  }

  const { email, password } = parsed.data;
  let connection;

  try {
    connection = await oracledb.getConnection();

    const result = await connection.execute(
      `SELECT ID_USUARIO, EMAIL, PASSWORD_HASH, NOMBRE, ROL, ESTADO
      FROM HD_USUARIOS WHERE EMAIL = :email`,
      { email },
      { outFormat: oracledb.OUT_FORMAT_OBJECT }
    );

    const user: any = result.rows?.[0];

    if (!user) { res.status(401).json({ error: 'Credenciales inválidas' }); return; }
    if (user.ESTADO === 'PENDIENTE') {
      res.status(403).json({ error: 'Tu cuenta está pendiente de aprobación. Contacta al administrador.' });
    }
  }
}
```



```

    if (user.ESTADO !== 'ACTIVO') {
      res.status(403).json({ error: 'Cuenta inactiva. Contacta al administrador.' }); return;
    }

    const passwordValida = await bcrypt.compare(password, user.PASSWORD_HASH);
    if (!passwordValida) { res.status(401).json({ error: 'Credenciales inválidas' }); return; }

    const secret = process.env.JWT_SECRET;
    if (!secret) throw new Error('JWT_SECRET no configurado');

    const token = jwt.sign(
      { idUsuario: user.ID_USUARIO, email: user.EMAIL, rol: user.ROL },
      secret,
      { expiresIn: '8h' }
    );

    res.json({
      mensaje: 'Login exitoso',
      token,
      usuario: {
        id: user.ID_USUARIO,
        nombre: user.NOMBRE,
        email: user.EMAIL,
        rol: user.ROL,
      },
    });

  } catch (error: any) {
    console.error('Error en login:', error.message);
    res.status(500).json({ error: 'Error interno del servidor' });
  } finally {
    if (connection) await connection.close();
  }
};

```

Historia de usuario 2 — Recuperar contraseña

Tabla 68. Tarea 2 — Recuperar contraseña

Tarea			#2
Historia de usuario:	#2 — Recuperar contraseña		
Usuario:	Todos		
Nombre:	Recuperar contraseña		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	1
Fecha inicio:	13/01/2026	Fecha fin:	13/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Todos los usuarios pueden solicitar el restablecimiento de contraseña mediante correo institucional.		
Observación:	El usuario debe tener una cuenta activa en el sistema para poder solicitar el cambio de contraseña.		

El módulo verifica la existencia del correo institucional en la tabla HD_USUARIOS de Oracle 19c. Si el usuario existe y está activo, el sistema genera un token temporal

de restablecimiento, lo almacena en la base de datos con un tiempo de expiración de 1 hora y envía al correo institucional del usuario un enlace seguro para definir su nueva contraseña. La nueva contraseña es cifrada con Bcrypt antes de ser almacenada en Oracle 19c.

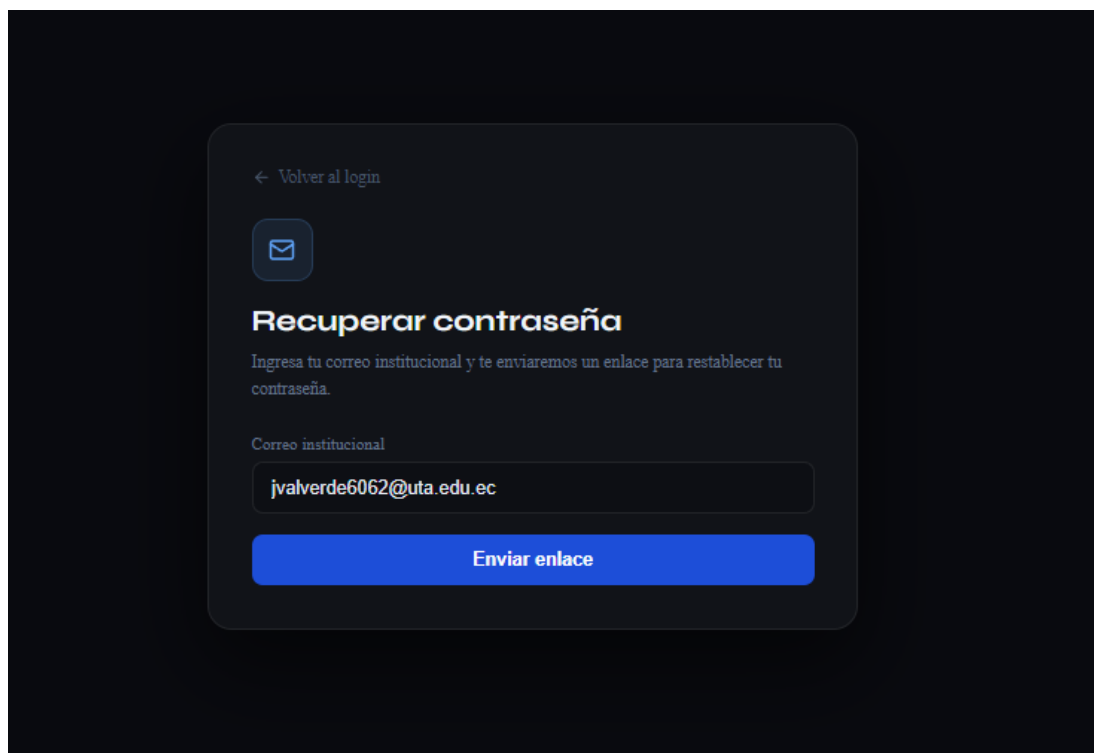


Figura 24. Pantalla de recuperación de contraseña del sistema Helpdesk

Código 2. Endpoint de recuperación de contraseña — backend Express.js

```
1  'use client';
2  import { useState } from 'react';
3  import Link from 'next/link';
4  import api from '@src/services/api';
5  import { Loader2, Mail, ArrowLeft, CheckCircle2 } from 'lucide-react';
6
7  export default function ForgotPasswordPage() {
8    const [email, setEmail] = useState('');
9    const [loading, setLoading] = useState(false);
10   const [sent, setSent] = useState(false);
11   const [error, setError] = useState('');
12
13   const handleSubmit = async (e: React.FormEvent) => {
14     e.preventDefault();
15     setLoading(true); setError('');
16     try {
17       await api.post('/auth/forgot-password', { email });
18       setSent(true);
19     } catch (err: any) {
20       setError(err.response?.data?.error || 'Error al procesar la solicitud');
21     } finally { setLoading(false); }
22   };
23 }
```

```

> src > controllers > TS auth.controller.ts > ...
// — POST /api/auth/forgot-password —————
export const forgotPassword = async (req: Request, res: Response): Promise<void> => {
  const { email } = req.body;
  if (!email || typeof email !== 'string') {
    res.status(400).json({ error: 'Email requerido' }); return;
  }

  let connection;
  try {
    connection = await oracledb.getConnection();
    const result = await connection.execute(
      `SELECT ID_USUARIO, NOMBRE, ESTADO FROM HD_USUARIOS WHERE EMAIL = :email`,
      { email }, { outFormat: oracledb.OUT_FORMAT_OBJECT }
    );
    const user: any = result.rows?.[0];

    // Siempre responder OK para no revelar si el email existe
    if (!user || user.ESTADO !== 'ACTIVO') {
      res.json({ mensaje: 'Si el correo existe, recibirás el enlace en breve' }); return;
    }

    const token = crypto.randomBytes(48).toString('hex');
    const expira = new Date(Date.now() + 60 * 60 * 1000); // 1 hora
    const expiraOracle = expira.toISOString().replace('T', ' ').substring(0, 19);

    // Invalidar tokens anteriores del mismo usuario
    await connection.execute(
      `UPDATE HD_RESET_TOKENS SET USADO = 1 WHERE ID_USUARIO = :id AND USADO = 0`,
      { id: user.ID_USUARIO }, { autoCommit: true }
    );

    await connection.execute(
      `INSERT INTO HD_RESET_TOKENS (ID_USUARIO, TOKEN, EXPIRA_EN)
      VALUES (:id, :token, TO_TIMESTAMP(:expira, 'YYYY-MM-DD HH24:MI:SS'))`,
      { id: user.ID_USUARIO, token, expira: expiraOracle },
      { autoCommit: true }
    );

    const frontendUrl = process.env.FRONTEND_URL ?? 'http://localhost:3000';
    const resetUrl = `${frontendUrl}/reset-password?token=${token}`;

    await sendMail({
      from: `Helpdesk GAD Pelileo` <${process.env.EMAIL_USER}>,
      to: email,
      subject: 'Restablecer contraseña – Helpdesk GAD Pelileo',
      html: `
        <div style="font-family:Arial,sans-serif;max-width:500px;margin:auto">
          <h2 style="color:#1d4ed8">Restablecer contraseña</h2>
          <p>Hola <strong>${user.NOMBRE}</strong>,</p>
          <p>Recibimos una solicitud para restablecer tu contraseña. El enlace expira en <strong>1 hora</strong>
          <a href="${resetUrl}"

```

Historia de usuario 3 — gestión de usuarios

Tabla 69. Tarea 3 — gestión de usuarios

Tarea			#3
Historia de usuario:	#3 — Gestión de usuarios		
Usuario:	ADMIN		
Nombre:	Gestión de usuarios		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	1
Fecha inicio:	14/01/2026	Fecha fin:	14/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN puede crear, editar, activar y desactivar cuentas de usuario del sistema Helpdesk.		
Observación:	—		

El módulo de gestión de usuarios implementa las operaciones CRUD completas sobre la tabla HD_USUARIOS de Oracle 19c. El formulario de creación valida campos requeridos, genera un username automático y asigna una contraseña temporal cifrada con Bcrypt. El ADMIN puede activar o desactivar cuentas sin eliminarlas físicamente, manteniendo el historial de tickets asociados al usuario.

ADMINISTRACIÓN

Gestión de Usuarios

Registra, edita y controla el acceso al sistema.

TOTAL

6

ACTIVOS

6

PENDIENTES

0

TÉCNICOS

1

Q Buscar por nombre, email o rol...

Probar email

Nuevo Usuario

6 de 6 usuarios

NOMBRE	EMAIL	DEPARTAMENTO	ROL	ESTADO	NOTIF.	REGISTRADO	ACCIONES
Gabriela Valverde	gabyseliza22@u...	Departamento de O...	USUARIO	Activo		6/5/2026	Editar Pard Borrar
Dennis Chicaiza	dv.chicaiza@u...	Sin asignar	USUARIO	Activo	—	4/5/2026	Editar Pard Borrar
Josue Valverde	jvalverde6063...	Sin asignar	USUARIO	Activo		13/4/2026	Editar Pard Borrar

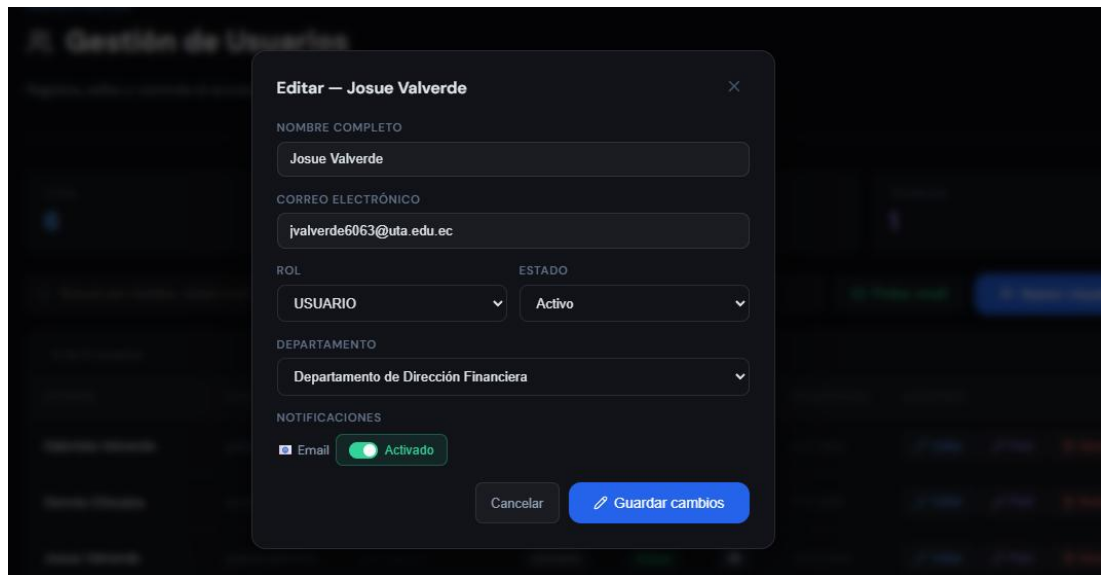


Figura 25. Módulo de gestión de usuarios — vista ADMIN

Código 3. Creación de usuario — Backend Express.js

```

1  "use client";
2  import { useState, useEffect } from 'react';
3  import { useSession } from 'next-auth/react';
4  import {
5    Loader2, UserPlus, ShieldAlert, CheckCircle2, Clock,
6    Pencil, Trash2, KeyRound, X, Search, Users, Eye, EyeOff,
7    Mail,
8  } from 'lucide-react';
9  import { usuariosService } from '@src/services/usuarios.service';
10 import { departamentosService, Departamento } from '@src/services/departamentos.service';
11 import { useToast } from '@src/components/Toast';
12 import styles from './usuarios.module.css';
13
14 const GYE = 'America/Guayaquil';
15
16 interface Usuario {
17   ID_USUARIO:      number;
18   NOMBRE:          string;
19   EMAIL:           string;
20   ROL:             string;
21   ESTADO:          string;
22   FECHA_CREACION:  string;
23   TELEFONO:        string | null;
24   NOTIF_EMAIL:     number;
25   NOTIF_WHATSAPP?: number;
26   ID_DEPARTAMENTO: number | null;
27   NOMBRE_DEPARTAMENTO: string | null;
28 }

```

```

142 // — Página principal —
143 export default function UsuariosPage() {
144   const { data: session, status } = useSession();
145   const { toast } = useToast();
146
147   const [usuarios, setUsuarios] = useState<Usuario[]>([]);
148   const [departamentos, setDepartamentos] = useState<Departamento[]>([]);
149   const [loadingU, setLoadingU] = useState(true);
150   const [search, setSearch] = useState('');
151
152   // — Modals state —
153   const [createOpen, setCreateOpen] = useState(false);
154   const [editTarget, setEditTarget] = useState<Usuario | null>(null);
155   const [pwdTarget, setPwdTarget] = useState<Usuario | null>(null);
156   const [deleteTarget, setDeleteTarget] = useState<Usuario | null>(null);
157
158   // — Saving flags —
159   const [saving, setSaving] = useState(false);
160   const [deleting, setDeleting] = useState(false);
161   const [testingEmail, setTestingEmail] = useState(false);
162
163   // — Formulario crear —
164   const [createForm, setCreateForm] = useState({
165     nombre: '', email: '', password: '', rol: 'USUARIO',
166     telefono: '', notif_email: 1, id_departamento: null as number | null,
167   });

```

```

163 // — Formulario crear —
164 const [createForm, setCreateForm] = useState({
165   nombre: '', email: '', password: '', rol: 'USUARIO',
166   telefono: '', notif_email: 1, id_departamento: null as number | null,
167 });
168
169 // — Formulario editar —
170 const [editForm, setEditForm] = useState({
171   nombre: '', email: '', rol: '', estado: '',
172   telefono: '', notif_email: 1, id_departamento: null as number | null,
173 });
174
175 // — Formulario reset password —
176 const [pwdForm, setPwdForm] = useState({ nueva: '', confirmar: '' });
177 const [showPwd, setShowPwd] = useState(false);
178
179 // — Carga de datos —
180 const cargar = async () => {
181   setLoadingU(true);
182   try {
183     const [dataU, dataD] = await Promise.all([
184       usuariosService.getAll(),
185       departamentosService.getAll(),
186     ]);

```

```

198 // — Guard —————
199 if (status === 'loading') return <div className={styles.container} />;
200 if ((session?.user as any)?.rol !== 'ADMIN') {
201   return (
202     <div className={styles.restricted}>
203       <ShieldAlert size={48} color="#ef4444" />
204       <h2 className={styles.restrictedTitle}>Acceso Restringido</h2>
205       <p>Solo los administradores pueden ver esta sección.</p>
206     </div>
207   );
208 }
209
210 const meId = (session?.user as any)?.id ?? 0;
211
212 // — Filtro —————
213 const term = search.toLowerCase();
214 const filtrados = usuarios.filter(u =>
215   u.NOMBRE.toLowerCase().includes(term) ||
216   u.EMAIL.toLowerCase().includes(term) ||
217   u.ROL.toLowerCase().includes(term) ||
218   (u.NOMBRE_DEPARTAMENTO ?? '').toLowerCase().includes(term)
219 );
220
221 // — Stats rápidas —————
222 const totalActivos = usuarios.filter(u => u.ESTADO === 'ACTIVO').length;
223 const totalPendientes = usuarios.filter(u => u.ESTADO === 'PENDIENTE').length;
224 const totalTecnicos = usuarios.filter(u => u.ROL === 'TECNICO').length;

```

> src > controllers > TS usuarios.controller.ts > ...

```

// — POST /usuarios/crear —————
export const crearUsuario = async (req: AuthRequest, res: Response) => {
  const parsed = CreateUsuarioSchema.safeParse(req.body);
  if (!parsed.success) {
    return res.status(400).json({ error: parsed.error.issues[0].message });
  }

  const { nombre, email, password, rol, estado, telefono, notif_email, notif_whatsapp, id_departamento }
  const estadoFinal = rol === 'PASANTE' && estado === 'ACTIVO' ? 'PENDIENTE' : estado;

  let connection;
  try {
    connection = await oracledb.getConnection();

    const checkEmail = await connection.execute(
      `SELECT ID_USUARIO FROM HD_USUARIOS WHERE EMAIL = :email`,
      { email },
      { outFormat: oracledb.OUT_FORMAT_OBJECT }
    );
    if ((checkEmail.rows ?? []).length > 0) {
      return res.status(400).json({ error: 'El correo electrónico ya está registrado' });
    }

    const hashedPassword = await bcrypt.hash(password, 10);

```

```

    await connection.execute(
      `INSERT INTO HD_USUARIOS
      (NOMBRE, EMAIL, PASSWORD_HASH, ROL, ESTADO, TELEFONO, NOTIF_EMAIL, NOTIF_WHATSAPP, ID_DEPARTAMENTO)
      VALUES
      (:nombre, :email, :password, :rol, :estado, :telefono, :notif_email, :notif_whatsapp, :id_departamen
      { nombre, email, password: hashedPassword, rol, estado: estadoFinal,
        telefono: telefono ?? null, notif_email, notif_whatsapp,
        id_departamento: id_departamento ?? null },
        { autoCommit: true }
      );

    // Enviar credenciales por email - fire and forget
    notificarBienvenida({
      nombre,
      email,
      passwordPlano: password,
      rol,
      notifEmail: notif_email,
    }).catch(() => {});

    return res.status(201).json({
      mensaje: rol === 'PASANTE'
        ? 'Pasante registrado. Estado: PENDIENTE de aprobación.'
        : 'Usuario creado exitosamente',
    });

```

Historia de usuario 4 — gestión de roles

Tabla 70. Tarea 4 — gestión de roles

Tarea			#4
Historia de usuario:	#4 — Gestión de roles		
Usuario:	ADMIN		
Nombre:	Gestión de roles		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	1
Fecha inicio:	15/01/2026	Fecha fin:	15/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN asigna roles diferenciados (ADMIN, TÉCNICO, USUARIO) a cada cuenta del sistema.		
Observación:	—		

El módulo de gestión de roles permite al ADMIN reasignar el rol de cualquier usuario del sistema desde el panel de administración. El middleware verifyRole del backend valida el rol del token JWT en cada endpoint protegido.

Figura 26. Módulo de gestión de roles — asignación de rol

Código 4. Middleware de control de acceso por roles

```
> src > middleware > TS auth.middleware.ts > UserPayload > email
export type Rol = 'ADMIN' | 'TÉCNICO' | 'USUARIO' | 'PASANTE';

export interface UserPayload {
  idUsuario: number;
  nombre: string;
  email: string;
  rol: Rol;
  iat?: number;
  exp?: number;
}
```

```

export const authenticate = (req: AuthRequest, res: Response, next: NextFunction): void => {
  const authHeader = req.headers.authorization;

  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    res.status(401).json({ error: 'Acceso denegado. No se proporcionó token.' });
    return;
  }

  const token = authHeader.split(' ')[1];
  const secret = process.env.JWT_SECRET as string;

  try {
    // Verificamos y casteamos el contenido al tipo UserPayload
    const decoded = jwt.verify(token, secret) as UserPayload;

    // Inyectamos el payload en la petición
    req.user = decoded;
    next();
  } catch (error) {
    res.status(403).json({ error: 'Token inválido o expirado' });
  }
};

export const authorize = (...rolesPermitidos: Rol[]) => {
  return (req: AuthRequest, res: Response, next: NextFunction): void => {
    if (!req.user || !rolesPermitidos.includes(req.user.rol)) {
      res.status(403).json({
        error: `Acceso prohibido. Se requiere rol: ${rolesPermitidos.join(' o ')}`,
      });
      return;
    }
    next();
  };
};

```

- *Iteración 2 - configuración de categorías, prioridades*

La Iteración 2 implementa los módulos de configuración institucional del sistema: gestión de categorías y prioridades de incidencias, gestión de departamentos del GAD Municipal de Pelileo, configuración de tiempos de SLA por categoría/prioridad y el módulo de cierre de sesión. Estos módulos son prerequisites para la creación de tickets en la Iteración 3.

Tabla 71. Resumen iteración 2

Iteración	Nº	Historia de usuario
Iteración 2	5	Gestión de categorías y prioridades
	6	Gestión de departamentos
	7	Configuración de SLA
	8	Cerrar sesión

Historia de usuario 5 — gestión de categorías y prioridades

Tabla 72. Tarea 5 — gestión de categorías y prioridades

Tarea			#5
Historia de usuario:	#5 — Gestión de categorías y prioridades		
Usuario:	ADMIN		
Nombre:	Gestión de categorías y prioridades		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	2
Fecha inicio:	19/01/2026	Fecha fin:	19/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN puede crear, editar y deshabilitar categorías (Hardware, Software, Red) y niveles de prioridad (Alta, Media, Baja).		
Observación:	Las categorías y prioridades son utilizadas por todos los roles al crear tickets.		

El módulo gestiona las tablas HD_CATEGORIAS y HD_PRIORIDADES de Oracle 19c. Las categorías definen el tipo de incidencia reportada y las prioridades determinan el orden de atención (Alta, Media, Baja). Ambas tablas están relacionadas con la tabla HD_SLA_CONFIG, que almacena el tiempo máximo de resolución en horas para cada combinación categoría-prioridad, configurado por el ADMIN.

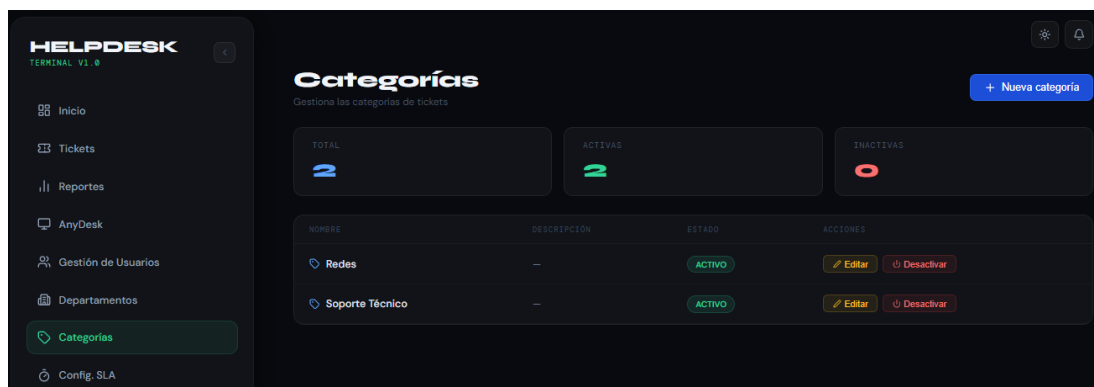


Figura 27. Módulo de gestión de categorías y prioridades — vista ADMIN

Código 5. Consulta de categorías activas — Oracle 19c

```

> src > controllers > TS categorias.controller.ts > ...
// — GET /categorias —
export const getCategorias = async (_req: AuthRequest, res: Response) => {
  const cached = cache.get<unknown[]>(CACHE_KEY);
  if (cached) return res.status(200).json(cached);

  let connection;
  try {
    connection = await oracledb.getConnection();
    const result = await connection.execute(
      `SELECT ID_CATEGORIA, NOMBRE, DESCRIPCION, ESTADO
      FROM HD_CATEGORIAS ORDER BY NOMBRE`,
      {},
      { outFormat: oracledb.OUT_FORMAT_OBJECT }
    );
    const rows = result.rows ?? [];
    cache.set(CACHE_KEY, rows, 300);
    return res.status(200).json(rows);
  } catch (error: any) {
    console.error('Error al obtener categorías:', error.message);
    return res.status(500).json({ error: 'Error al obtener las categorías' });
  } finally {
    if (connection) await connection.close();
  }
};

```

Historia de usuario 6 — gestión de departamentos

Tabla 73. Tarea 6 — gestión de departamentos

Tarea			#6
Historia de usuario:	#6 — Gestión de departamentos		
Usuario:	ADMIN		
Nombre:	Gestión de departamentos		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	2
Fecha inicio:	20/01/2026	Fecha fin:	20/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN puede crear, editar y deshabilitar los departamentos del GAD Municipal de Pelileo.		
Observación:	—		

El módulo gestiona la tabla HD_DEPARTAMENTOS de Oracle 19c, que almacena los 13 departamentos del GAD Municipal de Pelileo identificados en la población de estudio (Tabla 3, Capítulo II): Departamento de TI, Talento Humano, Secretaría General, Asesoría Jurídica, Dirección Financiera, Obras Públicas, Planificación Territorial, Avalúos y Catastros, Gestión Ambiental, Comisaría Municipal, Desarrollo

Social, Camal Municipal y Mercado República de Argentina. Los departamentos son asignados a usuarios y utilizados en los reportes analíticos por área.

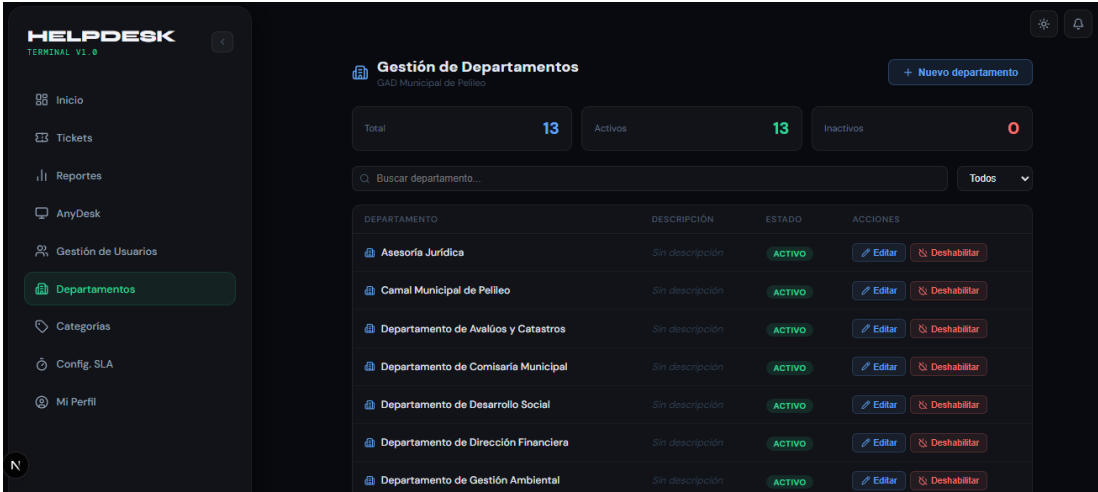


Figura 28. Módulo de gestión de departamentos

Historia de usuario 7 — configuración de SLA

Tabla 74. Tarea 7 — configuración de SLA

Tarea			#7
Historia de usuario:	#7 — Configuración de SLA		
Usuario:	ADMIN		
Nombre:	Configuración de SLA		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	1	Iteración asignada:	2
Fecha inicio:	21/01/2026	Fecha fin:	21/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN configura los tiempos máximos de resolución por categoría y prioridad para el cálculo automático del SLA.		
Observación:	Los tiempos de SLA configurados se aplican automáticamente al momento de crear cada ticket.		

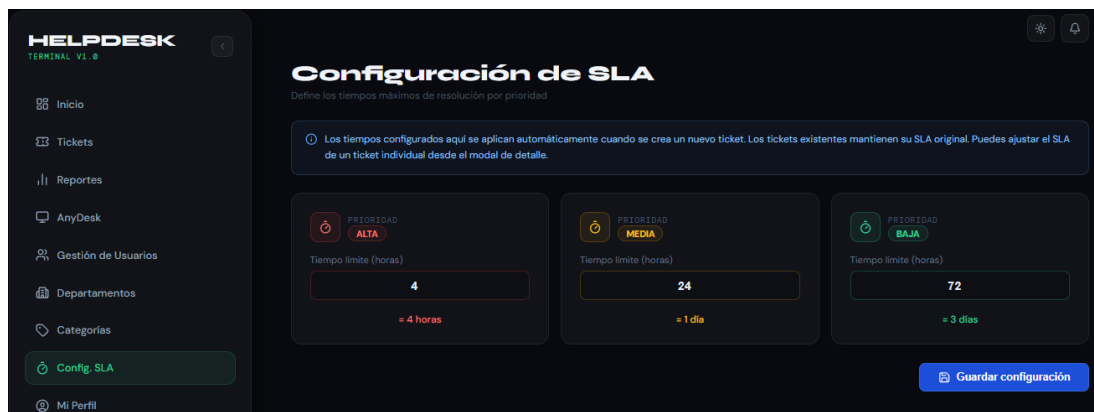


Figura 29. Módulo historia de usuario

El módulo de configuración de SLA permite al ADMIN definir los tiempos máximos de resolución en horas para cada combinación de categoría e incidencia. Estos valores son almacenados en la tabla HD_SLA_CONFIG de Oracle 19c y consultados por el motor de SLA (SLA engine) al momento de crear un ticket, calculando automáticamente la fecha y hora límite de resolución. El sistema genera una alerta visual en el panel del TÉCNICO cuando el tiempo restante es inferior al 20% del SLA configurado.

Código 6. Motor de cálculo de SLA — Node.js

```
> src > jobs > TS slaMonitor.ts > ...
const INTERVALO_MS = 60 * 60 * 1000; // cada hora

async function verificarSLA(): Promise<void> {
  let connection;
  try {
    connection = await oracledb.getConnection();

    // Buscar tickets con SLA vencido, activos, y que aún no recibieron alerta
    const result = await connection.execute(
      `SELECT ID_TICKET FROM HD_TICKETS
       WHERE FECHA_SLA IS NOT NULL
          AND FECHA_SLA < SYSTIMESTAMP
          AND ESTADO NOT IN ('RESUELTO', 'CERRADO')
          AND SLA_ALERTA_ENVIADA = 0`,
      {},
      { outFormat: oracledb.OUT_FORMAT_OBJECT }
    );

    const vencidos = (result.rows as any[]) ?? [];
    if (vencidos.length === 0) return;

    console.log(`🚨 SLA Monitor: ${vencidos.length} ticket(s) con SLA vencido - enviando alertas`);

    for (const row of vencidos) {
      const id: number = row.ID_TICKET;
```

```

// Marcar como alertado primero para evitar doble envío si falla la notificación
await connection.execute(
  `UPDATE HD_TICKETS SET SLA_ALERTA_ENVIADA = 1 WHERE ID_TICKET = :id`,
  { id },
  { autoCommit: true }
);

// Fire and forget – error no afecta al loop
notificarSLAVencido(id).catch((err) =>
  console.error('❌ Error notificación SLA ticket #${id}:', err.message)
);
}
} catch (err: any) {
  console.error('❌ SLA Monitor error:', err.message);
} finally {
  if (connection) await connection.close();
}
}

export function startSLAMonitor(): void {
  console.log('🚀 SLA Monitor iniciado – verificación cada hora');
  // Primera verificación al arrancar (con 10s de delay para que el pool esté listo)
  setTimeout(verificarSLA, 10_000);
  setInterval(verificarSLA, INTERVALO_MS);
}

```

Historia de usuario 8 — cerrar sesión

Tabla 75. Tarea 8 — cerrar sesión

Tarea			#8
Historia de usuario:	#8 — Cerrar sesión		
Usuario:	Todos		
Nombre:	Cerrar sesión		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Baja
Puntos estimados:	1	Iteración asignada:	2
Fecha inicio:	21/01/2026	Fecha fin:	21/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Todos los roles pueden cerrar sesión desde el menú principal del sistema.		
Observación:	—		

El cierre de sesión invalida el token JWT activo del usuario mediante NextAuth.js y redirige al usuario a la pantalla de acceso. El sistema implementa cierre de sesión seguro eliminando la cookie de sesión del navegador y el token del almacenamiento local, garantizando que no sea posible acceder al sistema sin volver a autenticarse.

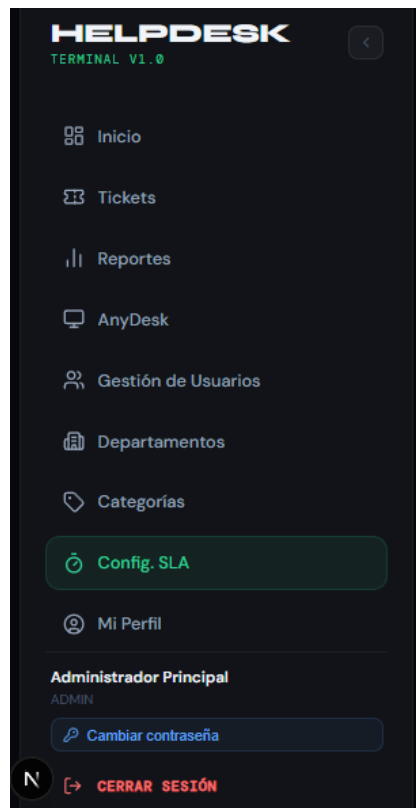


Figura 30. Botón de cierre de sesión — menú principal del sistema

- **Iteración 3**

La Iteración 3 implementa el núcleo funcional del sistema Helpdesk: el registro, gestión y asignación de tickets. Esta iteración es la más crítica del proyecto, ya que aborda directamente los riesgos R-01 (pérdida de solicitudes), R-02 (ausencia de priorización) y R-06 (falta de trazabilidad) identificados en la matriz de evaluación de riesgos (Tabla 16, sección 3.1.2). También se implementa el cálculo automático de SLA al crear cada ticket.

Tabla 76. Resumen iteración 3

Iteración	Nº	Historia de usuario
Iteración 3	9	Crear ticket
	10	Gestión de tickets
	11	Asignación de tickets
	12	Reapertura de ticket
	13	Cálculo automático de SLA

Historia de usuario 9 — Crear ticket

Tabla 77. Tarea 9 — crear ticket

Tarea			#9
Historia de usuario:	#9 — Crear ticket		
Usuario:	Todos		
Nombre:	Crear ticket		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Fecha inicio:	22/01/2026	Fecha fin:	23/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Todos los usuarios pueden registrar una incidencia técnica completando el formulario de ticket con título, categoría, prioridad y descripción. El sistema asigna número único y calcula el SLA automáticamente.		
Observación:	Al crear el ticket el sistema genera el número de seguimiento, calcula la fecha límite de SLA y envía notificación automática al Departamento de TI.		

El módulo de creación de tickets es el componente central del sistema Helpdesk. El formulario valida los campos requeridos (título, categoría, prioridad, descripción) y al confirmar el registro, el backend ejecuta una transacción en Oracle 19c que:

- (1) inserta el ticket en la tabla HD_TICKETS con número único generado por secuencia Oracle (HD_TICKETS_SEQ).
- (2) invoca el motor de SLA para calcular la fecha límite de resolución.
- (3) registra el estado inicial 'Abierto' en la tabla HD_TICKET_ESTADOS.
- (4) genera una notificación automática al Departamento de TI.

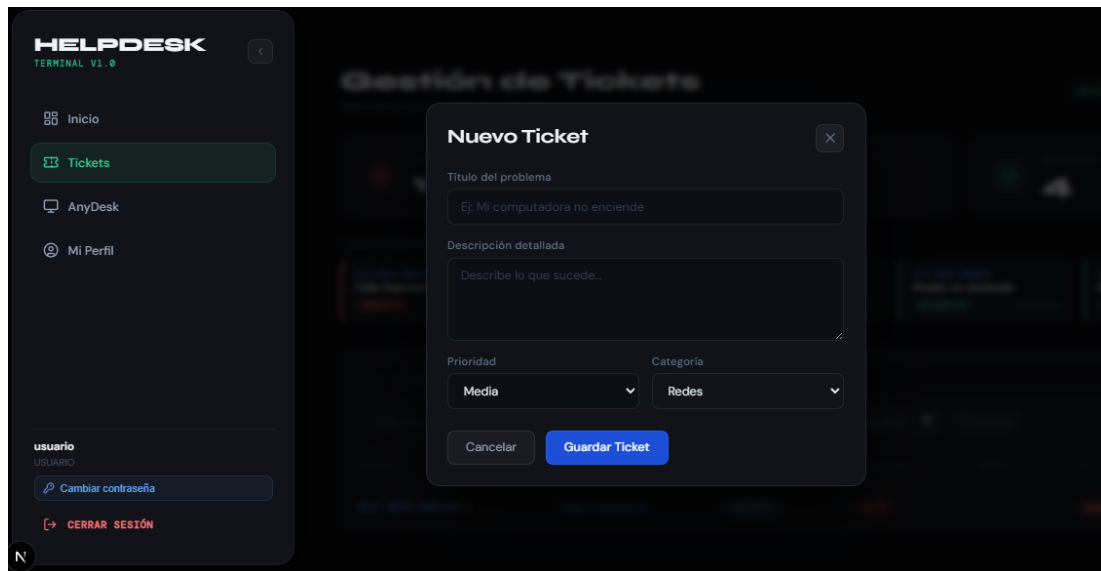


Figura 31. Formulario de creación de ticket — vista usuario

Código 7. Creación de ticket con SLA automático

```
> src > controllers > TS tickets.controller.ts > createTicket > autoRow
// — POST /tickets —
export const createTicket = async (req: AuthRequest, res: Response) => {
  const parsed = CreateTicketSchema.safeParse(req.body);
  if (!parsed.success) return res.status(400).json({ error: parsed.error.issues[0].message });

  const idUsuario = getUserId(req);
  if (!idUsuario) return res.status(401).json({ error: 'Sesión inválida' });

  const { titulo, descripcion, prioridad, id_categoria, id_tecnico, auto_asignar } = parsed.data;

  let connection;
  try {
    connection = await oracledb.getConnection();

    // Resolve technician: explicit > auto-assign > null
    let tecnicoFinal: number | null = id_tecnico ?? null;
    if (!tecnicoFinal && auto_asignar) {
      const autoRow = await connection.execute(
        `SELECT u.ID_USUARIO
        FROM HD_USUARIOS u
        WHERE u.ROL = 'TECNICO' AND u.ESTADO = 'ACTIVO'
        ORDER BY (
          SELECT COUNT(*) FROM HD_TICKETS t
          WHERE t.ID_TECNICO = u.ID_USUARIO AND t.ESTADO NOT IN ('RESUELTO','CERRADO')
        ) ASC
        FETCH FIRST 1 ROW ONLY`,
        {}
      );
    }
  }
}
```

```

    { outFormat: oracledb.OUT_FORMAT_OBJECT }
  );
  tecnicoFinal = (autoRow.rows as any[])?.[0]?.ID_USUARIO ?? null;
}

// SLA desde BD (HD_SLA_CONFIG) con fallback hardcoded
let slaHoras = prioridad === 'ALTA' ? 4 : prioridad === 'BAJA' ? 72 : 24;
try {
  const slaRow = await connection.execute(
    `SELECT HORAS_LIMITE FROM HD_SLA_CONFIG WHERE PRIORIDAD = :p`,
    { p: prioridad }, { outFormat: oracledb.OUT_FORMAT_OBJECT }
  );
  const h = (slaRow.rows as any[])?.[0]?.HORAS_LIMITE;
  if (h) slaHoras = Number(h);
} catch { /* usa fallback */ }

const result = await connection.execute(
  `INSERT INTO HD_TICKETS
  (TITULO, DESCRIPCION, PRIORIDAD, ID_USUARIO, ID_CATEGORIA, ESTADO, ID_TECNICO)
  VALUES (:titulo, :descripcion, :prioridad, :id_usuario, :id_categoria, 'ABIERTO', :id_tecnico)
  RETURNING ID_TICKET INTO :id_ticket`,
  {
    titulo, descripcion, prioridad,
    id_usuario: idUsuario,
    id_categoria,
    id_categoria,
    id_tecnico: tecnicoFinal,
    id_ticket: { dir: oracledb.BIND_OUT, type: oracledb.NUMBER },
  },
  { autoCommit: false }
);

const idTicket = (result.outBinds as any)?.id_ticket?.[0];
const codigo = `HLP-${new Date().getFullYear()}-${String(idTicket).padStart(6, '0')}`;

// Set CODIGO_TICKET (zero-padded) + FECHA_SLA en un solo UPDATE
await connection.execute(
  `UPDATE HD_TICKETS
  SET CODIGO_TICKET = :codigo,
      FECHA_SLA = SYSTIMESTAMP + :sla_h / 24
  WHERE ID_TICKET = :id`,
  { codigo, sla_h: slaHoras, id: idTicket },
  { autoCommit: true }
);

ticketEmitter.emit('ticket.created', { idTicket, idUsuario, codigoTicket: codigo });

return res.status(201).json({ mensaje: 'Ticket creado exitosamente', codigo, idTicket });

```

Historia de usuario 10 — gestión de tickets

Tabla 78. Tarea 10 — gestión de tickets

Tarea			#10
Historia de usuario:	#10 — Gestión de tickets		
Usuario:	ADMIN / TÉCNICO		
Nombre:	Gestión de tickets		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta

Puntos estimados:	2	Iteración asignada:	3
Fecha inicio:	26/01/2026	Fecha fin:	27/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Los roles ADMIN y TÉCNICO pueden visualizar, editar, asignar y actualizar el estado de todos los tickets registrados.		
Observación:	El estado del ticket cambia automáticamente al registrar una actualización.		

El módulo de gestión de tickets implementa el panel principal de administración del sistema Helpdesk, donde los roles ADMIN y TÉCNICO visualizan el listado completo de tickets con filtros por estado (Abierto, En proceso, En espera, Cerrado), categoría, prioridad, departamento y rango de fechas. Cada ticket muestra el indicador de cumplimiento de SLA en tiempo real: verde (dentro del plazo), amarillo (menos del 20% restante) y rojo (SLA vencido).

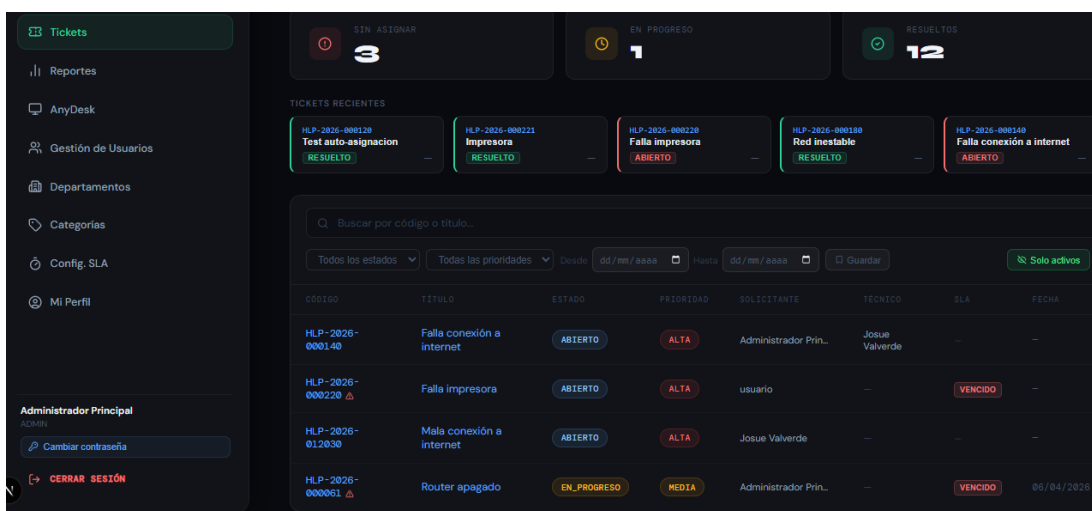


Figura 32. Panel de gestión de tickets — vista ADMIN

Historia de usuario 11 — Asignación de tickets

Tabla 79. Tarea 11 — Asignación de tickets

Tarea		#11
Historia de usuario:	#11 — Asignación de tickets	
Usuario:	ADMIN / TECNICO	
Nombre:	Asignación de tickets	

Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Fecha inicio:	28/01/2026	Fecha fin:	29/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El ADMIN asigna los tickets al técnico disponible según disponibilidad y especialización.		
Observación:	—		

La asignación de tickets actualiza la columna ID_TECNICO_ASIGNADO de la tabla HD_TICKETS en Oracle 19c y cambia el estado del ticket de 'Abierto' a 'En proceso'. El sistema registra la asignación en la tabla HD_TICKET_HISTORIAL con fecha, usuario que realizó la asignación y el técnico asignado, manteniendo la trazabilidad completa del ciclo de vida del ticket. Una notificación automática es enviada al técnico asignado informando sobre el nuevo ticket.

The screenshot shows a dark-themed modal window titled "DETALLE DEL TICKET". At the top, the ticket ID "HLP-2026-000061" is displayed next to a red "SLA VENCIDO" badge. Below this, the "TÍTULO" is "Router apagado" and the "DESCRIPCIÓN" is "No se puede conectar a internet". The "PRIORIDAD" is "MEDIA" and the "ESTADO" is "EN_PROGRESO". The "SOLICITANTE" is "Administrador Principal" and the "CREADO" date is "08/04/2026, 05:28 a. m.". There is an "Editar" button next to the SLA status. The "ASIGNAR TÉCNICO" section features a dropdown menu with options: "— Sin asignar —", "Josue Valverde (TECNICO)", and "Administrador Principal (ADMIN)". A "Guardar" button is next to the dropdown. At the bottom, there is a section for "RESPONDER Y FINALIZAR" with a text input field.

Figura 33. Modal de asignación de ticket a técnico

Historia de usuario 12 — reapertura de ticket

Tabla 80. Tarea 12 — reapertura de ticket

Tarea			#12
Historia de usuario:	#12 — Reapertura de ticket		
Usuario:	USUARIO		
Nombre:	Reapertura de ticket		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	3
Fecha inicio:	30/01/2026	Fecha fin:	30/01/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El USUARIO puede reabrir un ticket cerrado si considera que el problema no fue resuelto satisfactoriamente.		
Observación:	El ticket debe estar en estado Cerrado para poder ser reabierto.		

La reapertura de ticket cambia el estado de 'Cerrado' a 'Abierto' en la tabla HD_TICKETS de Oracle 19c, registra el evento en HD_TICKET_HISTORIAL y notifica automáticamente al Departamento de TI. Esta funcionalidad aborda el riesgo R-01 identificado en el análisis, garantizando que ninguna incidencia quede sin resolución definitiva y permitiendo al usuario expresar su insatisfacción con la solución aplicada.

- **Iteración 4**

La Iteración 4 implementa los módulos de gestión de conocimiento institucional y acceso remoto: la base de conocimiento con búsqueda full-text, el módulo AnyDesk por departamento y las notificaciones automáticas del sistema. Estos módulos abordan directamente los riesgos R-03 (reincidencia de problemas) y R-10 (seguridad de la información) identificados en la matriz de riesgos.

Tabla 81. Resumen iteración 4

Iteración	N°	Historia de usuario
Iteración 4	14	Base de conocimiento
	15	Módulo AnyDesk por departamento
	16	Notificaciones automáticas

Historia de usuario 14 — base de conocimiento**Tabla 82.** Tarea 14 — base de conocimiento

Tarea			#14
Historia de usuario:	#14 — Base de conocimiento		
Usuario:	ADMIN / TÉCNICO / USUARIO		
Nombre:	Base de conocimiento		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	3	Iteración asignada:	4
Fecha inicio:	02/02/2026	Fecha fin:	04/02/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Los roles ADMIN y TÉCNICO pueden crear y editar artículos con soporte Markdown. Todos los roles pueden consultar y buscar artículos.		
Observación:	La base de conocimiento aborda el riesgo R-03: reincidencia de problemas no documentados.		

El módulo de base de conocimiento implementa un repositorio institucional de soluciones documentadas a incidencias recurrentes del GAD Municipal de Pelileo. Los artículos son almacenados en la tabla HD_BASE_CONOCIMIENTO de Oracle 19c con soporte para contenido en formato Markdown, renderizado en el frontend mediante la librería react-markdown. La búsqueda full-text utiliza el operador LIKE de Oracle sobre los campos TITULO_ARTICULO y CONTENIDO_ARTICULO, permitiendo localizar soluciones documentadas en segundos.

Historia de usuario 15 — módulo AnyDesk por departamento

Tabla 83. Tarea 15 — módulo AnyDesk por departamento

Tarea			#15
Historia de usuario:	#15 — Módulo AnyDesk por departamento		
Usuario:	Todos		
Nombre:	Módulo AnyDesk por departamento		
Prioridad en el negocio:	Media	Riesgo de desarrollo:	Media
Puntos estimados:	2	Iteración asignada:	4
Fecha inicio:	05/02/2026	Fecha fin:	06/02/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El sistema centraliza los códigos AnyDesk por departamento. El ADMIN edita los códigos; todos los roles los consultan y copian al portapapeles.		
Observación:	Módulo adicional diseñado para optimizar los tiempos de asistencia remota del equipo técnico del GAD Municipal de Pelileo.		

El módulo AnyDesk es una funcionalidad adicional diseñada específicamente para el contexto del GAD Municipal de Pelileo, donde la asistencia remota mediante AnyDesk es la principal herramienta de soporte a distancia del Departamento de TI. El módulo centraliza los códigos de acceso remoto de los 13 departamentos en la tabla HD_ANYDESK_DEPARTAMENTOS de Oracle 19c, eliminando la pérdida de tiempo que ocurría cuando el técnico debía solicitar el código al funcionario cada vez que iniciaba una sesión remota. Todos los roles pueden copiar el código al portapapeles con un clic.

DEPARTAMENTO	CÓDIGO ANYDESK	CONTRASEÑA	ÚLTIMA ACTUALIZACIÓN	ACTUALIZADO POR	ACCIONES
Asesoría Jurídica	12567899	*****	11/05/2026	Administrador Principal	Conectar Copiar ID Editar
Carnal Municipal de Pelileo	—	—	11/05/2026	—	Editar
Departamento de Avalúos y Catastros	—	—	11/05/2026	—	Editar
Departamento de Comisaría Municipal	—	—	11/05/2026	—	Editar
Departamento de Desarrollo Social	—	—	11/05/2026	—	Editar
Departamento de Dirección Financiera	—	—	11/05/2026	—	Editar
Departamento de Gestión Ambiental	—	—	11/05/2026	—	Editar

Figura 34. Módulo AnyDesk — listado de códigos por departamento

Historia de usuario 16 — notificaciones automáticas

Tabla 84. Tarea 16 — notificaciones automáticas

Tarea			#16
Historia de usuario:	#16 — Notificaciones automáticas		
Usuario:	Sistema		
Nombre:	Notificaciones automáticas		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	2	Iteración asignada:	4
Fecha inicio:	09/02/2026	Fecha fin:	10/02/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	El sistema envía notificaciones automáticas al usuario y al técnico asignado al crear, actualizar y cerrar cada ticket.		
Observación:	—		

El sistema de notificaciones automáticas implementa el envío de correos institucionales mediante Nodemailer con configuración SMTP del servidor del GAD Municipal de Pelileo. Se definen tres eventos que generan notificaciones:

- (1) creación de ticket — notifica al Departamento de TI con el número y descripción del ticket.
- (2) actualización de estado — notifica al usuario solicitante con el nuevo estado y observaciones del técnico.
- (3) cierre de ticket — notifica al usuario con la solución documentada e invitación a calificar el servicio.

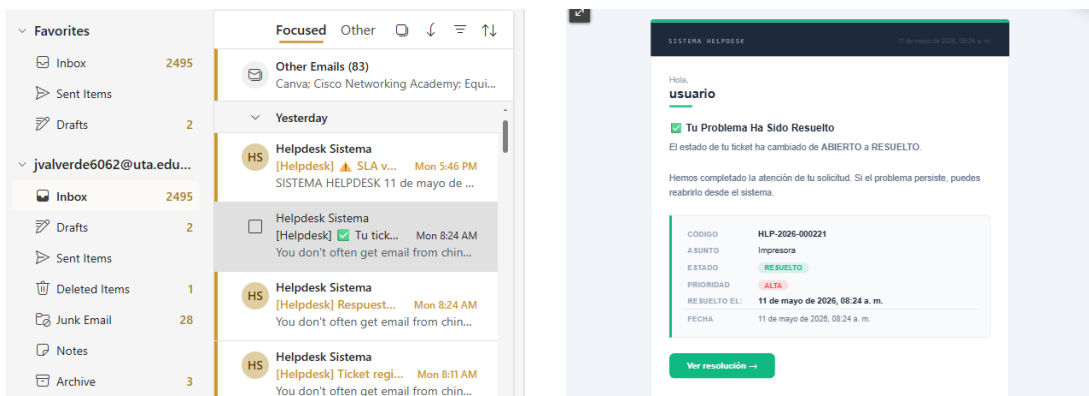


Figura 35. Ejemplo de notificación automática enviada al usuario — cierre de ticket

- **Iteración 5**

La Iteración 5 implementa los módulos analíticos y de retroalimentación del sistema: los reportes y métricas KPIs con 7 endpoints analíticos sobre Oracle Database 19c, la calificación del servicio por parte del usuario y el dashboard principal del sistema. Esta iteración cierra el ciclo de mejora continua y aborda directamente el riesgo R-04 (imposibilidad de medir KPIs de desempeño del área de TI) identificado en el análisis.

Tabla 85. Resumen iteración 5

Iteración	Nº	Historia de usuario
Iteración 5	17	Reportes y métricas KPIs
	18	Calificación del servicio

Historia de usuario 17 — reportes y métricas KPIs

Tabla 86. Tarea 17 — reportes y métricas KPIs

Tarea			#17
Historia de usuario:	#17 — Reportes y métricas KPIs		
Usuario:	Todos		
Nombre:	Reportes y métricas KPIs		
Prioridad en el negocio:	Alta	Riesgo de desarrollo:	Alta
Puntos estimados:	4	Iteración asignada:	5
Fecha inicio:	11/02/2026	Fecha fin:	14/02/2026
Programador a cargo:	Josué Israel Valverde Endara		

Descripción:	Todos los roles visualizan reportes analíticos con 7 KPIs institucionales. El ADMIN puede exportar los datos en formato CSV.
Observación:	7 endpoints analíticos desarrollados sobre Oracle Database 19c.

El módulo de reportes y métricas KPIs implementa 7 endpoints analíticos sobre Oracle Database 19c que generan los indicadores de desempeño clave del Departamento de TI del GAD Municipal de Pelileo:

Tabla 87. Indicadores de desempeño

KPI	Indicador	Descripción
1	Tiempo promedio de respuesta	Tiempo promedio desde la creación hasta la primera actualización del ticket.
2	Tasa de resolución	Porcentaje de tickets cerrados sobre el total registrado en el periodo.
3	Cumplimiento de SLA	Porcentaje de tickets resueltos dentro del tiempo límite de SLA configurado.
4	Tickets por categoría	Distribución de incidencias por tipo: Hardware, Software, Red, Acceso, Otro.
5	Tickets por departamento	Volumen de solicitudes generadas por cada departamento del GAD Municipal de Pelileo.
6	Tiempo promedio de resolución	Tiempo promedio desde la creación del ticket hasta su cierre formal.
7	Calificación promedio del servicio	Promedio de las calificaciones (1–5 estrellas) otorgadas por los usuarios al cierre de tickets.



Figura 36. Módulo de reportes KPIs — dashboard analítico del sistema Helpdesk

Código 8. Endpoint KPI — cumplimiento de SLA por período — Oracle 19c

```
qSafe('SELECT COUNT(*) AS TOTAL_CERRADOS,
SUM(CASE WHEN h.FECHA IS NULL OR h.FECHA <= t.FECHA_SLA THEN 1 ELSE 0 END) AS CUMPLIDOS,
ROUND(SUM(CASE WHEN h.FECHA IS NULL OR h.FECHA <= t.FECHA_SLA THEN 1 ELSE 0 END) * 100.0 / NULLIF(COUNT(*),0), 1) AS PORCENTAJE
FROM HD_TICKETS t
LEFT JOIN (
SELECT ID_TICKET, MIN(FECHA) AS FECHA
FROM HD_HISTORIAL_TICKETS
WHERE ACCION = 'CAMBIO_ESTADO' AND ESTADO_NUEVO IN ('RESUELTO','CERRADO')
GROUP BY ID_TICKET
) h ON h.ID_TICKET = t.ID_TICKET
WHERE t.ESTADO IN ('RESUELTO','CERRADO') AND t.FECHA_SLA IS NOT NULL ${whereT}', bindst);
```

Historia de usuario 18 — calificación del servicio

Tabla 88. Tarea 18 — calificación del servicio

Tarea			#18
Historia de usuario:	#18 — Calificación del servicio		
Usuario:	USUARIO		
Nombre:	Calificación del servicio		
Prioridad en el negocio:	Media	Riesgo de desarrollo:	Media
Puntos estimados:	1	Iteración asignada:	5
Fecha inicio:	17/02/2026	Fecha fin:	17/02/2026
Programador a cargo:	Josué Israel Valverde Endara		
Descripción:	Al cerrar un ticket, el USUARIO recibe notificación con invitación a calificar la atención recibida en escala de 1 a 5 estrellas.		
Observación:	La calificación es incluida en el KPI 7 del módulo de reportes analíticos.		

El módulo de calificación del servicio cierra el ciclo de retroalimentación del sistema Helpdesk. Al cerrar un ticket, el sistema envía automáticamente una notificación al usuario solicitante con un enlace al formulario de calificación (1 a 5 estrellas) junto con un campo de comentario opcional. La calificación es almacenada en la tabla HD_CALIFICACIONES de Oracle 19c y alimenta el KPI 7 del módulo de reportes. Esta funcionalidad responde directamente al hallazgo de la encuesta (Pregunta 5, Capítulo II), donde el 73% de funcionarios calificó el soporte técnico como poco efectivo o inefectivo.

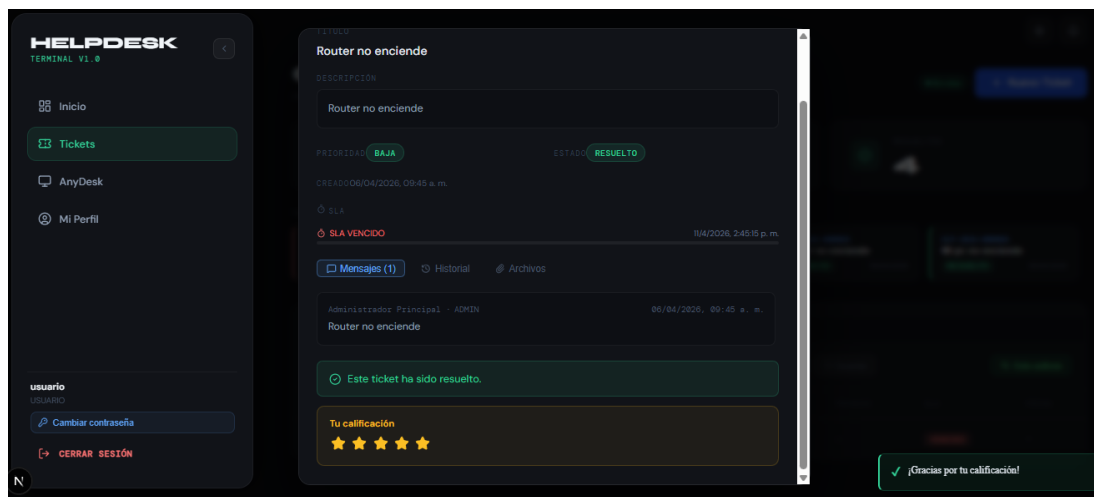


Figura 37. Formulario de calificación del servicio — cierre de ticket

e. Pruebas

Pruebas de aceptación por iteración

En la fase de Pruebas se verificó el correcto funcionamiento de cada historia de usuario definida en la Fase de Planificación, mediante la ejecución de pruebas de aceptación realizadas en conjunto con el personal del Departamento de TI del GAD Municipal de Pelileo al finalizar cada iteración de desarrollo. Cada prueba fue evaluada con uno de tres resultados posibles:

Resultado	Descripción
Exitosa	Funcionalidad implementada correctamente y validada sin observaciones por el Departamento de TI del GAD Municipal de Pelileo.
Neutral	Funcionalidad implementada con observaciones menores de mejora planteadas por el cliente; corregidas en la iteración siguiente sin impactar el cronograma.
Fallida	Funcionalidad no implementada o con errores críticos que impiden su uso. Requiere rediseño en la iteración siguiente.

Las pruebas de aceptación fueron ejecutadas al finalizar cada una de las tres iteraciones de desarrollo, con la participación directa del Jefe del Departamento de TI y el Técnico de Soporte del GAD Municipal de Pelileo, quienes validaron que cada historia de usuario cumpliera con los criterios de aceptación definidos en la Fase de Planificación.

Iteración 1 — módulos de autenticación y configuración institucional

La Iteración 1 cubrió las historias de usuario HU-01 a HU-08, correspondientes a los módulos de autenticación (acceso, recuperación de contraseña, cierre de sesión), gestión de usuarios y roles, y configuración institucional (categorías, prioridades, departamentos y SLA). Se detectaron dos observaciones de mejora en HU-03 y HU-05, las cuales fueron corregidas al inicio de la Iteración 2.

Tabla 89. Pruebas de aceptación — iteración 1

Nº	HU	Historia de usuario	Evaluación	Observaciones
1	HU-01	Acceso al sistema	✓ Exitosa	—
2	HU-02	Recuperar contraseña	✓ Exitosa	—
3	HU-03	Gestión de usuarios	Neutral	Se solicitó agregar validación de correo institucional duplicado. Corregido en iteración 2.
4	HU-04	Gestión de roles	✓ Exitosa	—
5	HU-05	Gestión de categorías y prioridades	Neutral	Se solicitó deshabilitar categoría sin eliminarla físicamente para conservar historial de tickets. Implementado en iteración 2.
6	HU-06	Gestión de departamentos	✓ Exitosa	—
7	HU-07	Configuración de SLA	✓ Exitosa	—
8	HU-08	Cerrar sesión	✓ Exitosa	—

Iteración 2 — módulo central de tickets y SLA

La Iteración 2 incorporó la corrección de las observaciones detectadas en la Iteración 1 (HU-03 y HU-05) y cubrió las historias de usuario del núcleo funcional del sistema: creación de tickets (HU-09), gestión y asignación (HU-10, HU-11), reapertura (HU-12) y cálculo automático de SLA (HU-13). Todas las historias evaluadas en esta iteración obtuvieron resultado Exitosa.

Tabla 90. Pruebas de aceptación — iteración 2

Nº	HU	Historia de usuario	Evaluación	Observaciones
1	HU-01	Acceso al sistema	✓ Exitosa	Corrección de observación iteración 1: validación de correo duplicado implementada.
2	HU-05	Gestión de categorías y prioridades	✓ Exitosa	Corrección de observación iteración 1: opción deshabilitar sin eliminar implementada.
3	HU-08	Cerrar sesión	✓ Exitosa	—
4	HU-09	Crear ticket	✓ Exitosa	—

5	HU-10	Gestión de tickets	✓ Exitosa	—
6	HU-11	Asignación de tickets	✓ Exitosa	—
7	HU-12	Reapertura de ticket	✓ Exitosa	—
8	HU-13	Cálculo automático de SLA	✓ Exitosa	—

Iteración 3 — módulos de conocimiento, acceso remoto y analítica

La Iteración 3 cubrió los módulos complementarios del sistema Helpdesk: base de conocimiento institucional (HU-14), módulo AnyDesk por departamento (HU-15), notificaciones automáticas (HU-16), reportes y métricas KPIs (HU-17) y calificación del servicio (HU-18). Todas las historias evaluadas obtuvieron resultado Exitosa sin observaciones pendientes.

Tabla 91. Pruebas de aceptación — iteración 3

Nº	HU	Historia de usuario	Evaluación	Observaciones
1	HU-14	Base de conocimiento	✓ Exitosa	—
2	HU-15	Módulo AnyDesk por departamento	✓ Exitosa	—
3	HU-16	Notificaciones automáticas	✓ Exitosa	—
4	HU-17	Reportes y métricas KPIs	✓ Exitosa	—
5	HU-18	Calificación del servicio	✓ Exitosa	—

Resumen consolidado de pruebas de aceptación

Tabla 92. Resumen consolidado de pruebas de aceptación

Iteración	HUs evaluadas	✓ Exitosas	~ Neutrales	✗ Fallidas	Resultado
Iteración 1	HU-01 a HU-08 (8 HU)	6	2	0	Observaciones menores corregidas en iteración siguiente
Iteración 2	HU-01, HU-05, HU-08 a HU-13 (8 HU)	8	0	0	Sin observaciones pendientes
Iteración 3	HU-14 a HU-18 (5 HU)	5	0	0	Sin observaciones pendientes
TOTAL	21 evaluaciones realizadas	19	2	0	100% de HUs implementadas y validadas exitosamente

Las pruebas de aceptación confirman que las 18 historias de usuario definidas en la Fase de Planificación fueron implementadas y validadas exitosamente, alcanzando un

porcentaje de cumplimiento del 100%. Las dos observaciones menores detectadas en la Iteración 1 (HU-03: validación de correo duplicado y HU-05: opción de deshabilitar categoría sin eliminación física) fueron corregidas al inicio de la Iteración 2 sin impactar el cronograma del proyecto ni generar retrasos en las entregas planificadas.

Estos resultados son consistentes con los antecedentes investigativos revisados en el Capítulo I. Pérez Romero y Mendoza Callupe [9] reportaron un incremento en la tasa de resolución de problemas del 37.2% al 85.9% tras implementar un sistema de gestión de incidencias estructurado, lo que valida que la correcta implementación y validación de cada módulo del sistema Helpdesk del GAD Municipal de Pelileo producirá mejoras operativas comparables en el Departamento de TI.

3.4 Implantación

Tras la validación exitosa de las pruebas de aceptación en las tres iteraciones de desarrollo, se procedió a la implantación del sistema Helpdesk en la infraestructura tecnológica del GAD Municipal de Pelileo, aprovechando el servidor institucional existente con Oracle Database 19c y la red LAN interna de la institución.

El sistema fue desplegado directamente en el servidor físico institucional del GAD Municipal de Pelileo, aprovechando la infraestructura tecnológica existente y eliminando la necesidad de adquirir servicios de hosting externos, lo que garantiza la soberanía de los datos institucionales y reduce los costos operativos.

a. Instalación del sistema web

El proceso de instalación y despliegue del sistema Helpdesk en el servidor institucional del GAD Municipal de Pelileo se realizó siguiendo los siguientes pasos:

1. Transferencia y configuración de archivos del sistema, tanto del frontend (Next.js 14) como del backend (Node.js + Express.js), fueron transferidos directamente al servidor físico institucional, configurando los directorios correspondientes para cada componente de la arquitectura y el entorno de ejecución Node.js 20 LTS.

2. Configuración de variables de entorno: Se establecieron las variables de entorno (.env) necesarias para la conexión segura a Oracle Database 19c mediante el driver node-oracledb v6, incluyendo credenciales de acceso, host, puerto, parámetros de autenticación JWT y nombre de servicio de la base de datos institucional.
3. Configuración de la base de datos: Se ejecutaron los scripts SQL de creación del esquema relacional en Oracle Database 19c, generando las tablas, secuencias e índices necesarios para el funcionamiento del sistema.
4. Habilitación en red institucional: Se configuró el acceso al sistema en la red LAN del GAD Municipal de Pelileo, garantizando la disponibilidad desde los equipos de los 13 departamentos de la institución a través de la dirección <https://192.168.69.8>.
5. Verificación de acceso y roles: Se confirmó el correcto funcionamiento del sistema desde equipos representativos de los departamentos, validando la autenticación con credenciales institucionales y la disponibilidad de módulos según el rol asignado (ADMINISTRADOR, TÉCNICO, USUARIO).

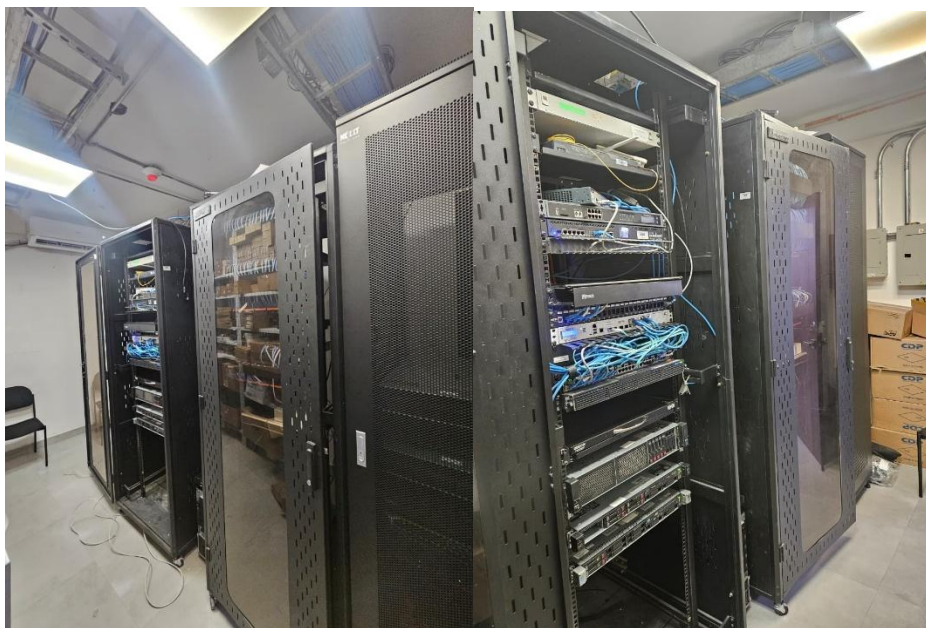


Figura 38. Configuración del servidor físico institucional del GAD Municipal de Pelileo.

La Figura 39 el sistema utiliza un certificado SSL autofirmado para cifrar la comunicación en la red interna, lo cual es una práctica habitual en entornos institucionales locales. Los navegadores pueden mostrar una advertencia de seguridad al acceder por primera vez, la cual puede ignorarse de forma segura al tratarse de una red LAN controlada por el GAD Municipal de Pelileo.

The screenshot displays the Oracle Enterprise Manager interface for configuring a connection to an Oracle Database 19c. The browser address bar shows a local HTTPS connection to 192.168.69.8. The page title is 'Configurar Conexión: 19c'. On the left, there is a sidebar with 'Detalles de Conexión' and 'Resumen'. The main area is titled 'Detalles de Conexión' and contains the following fields and options:

- Nombre del Objetivo:** 19c
- Tipo de Base de Datos:** Base de Datos
- Información de Conexión:**
 - Nombre de Host:** 192.168.69.8
 - Puerto:** 1521
 - Conectar como:** ☒ SYSDBA ☐ SYSOPER
 - Nombre de Servicio:** ORCL (with an example 'orcl' below)
 - Nombre de Usuario:** sys
 - Contraseña:** (masked with dots)
- Configuración de Oracle Management Service:**
 - ☐ Registrar en Oracle Management Service
 - Grupo de Gestión:** MGMT_SERVICE

Buttons for 'Cancelar', 'Siguiente >', and 'Probar Conexión' are visible on the right side of the form.

Figura 39. Configuración de conexión con Oracle Database 19c

La Figura 40 presenta el sistema Helpdesk en producción, accesible desde la red institucional del GAD Municipal de Pelileo, mostrando el dashboard principal con los indicadores KPI activos.

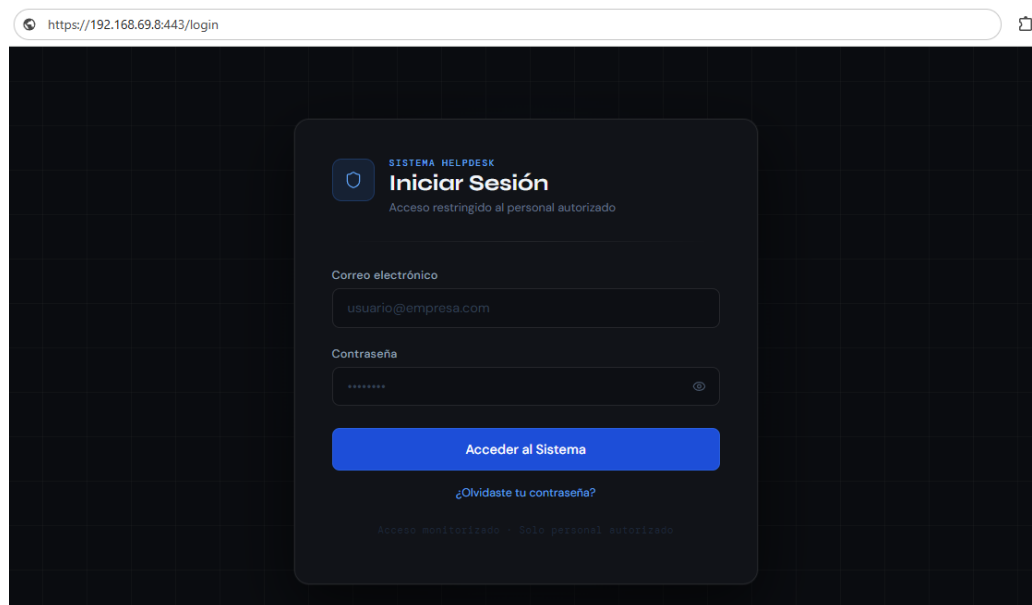


Figura 40. Sistema Helpdesk en producción — Red institucional del GAD Municipal de Pelileo

- ***Capacitaciones***

Tras la implantación del sistema en el servidor institucional, se procedió a realizar las capacitaciones al personal del GAD Municipal de Pelileo. Las sesiones fueron conducidas por el autor del proyecto, Josué Israel Valverde Endara, y se desarrollaron de forma presencial en las instalaciones de la institución durante los días 05 y 06 de mayo de 2026.

Las capacitaciones fueron diferenciadas por rol de usuario, abarcando tanto al personal administrativo (rol Usuario) como al Departamento de Tecnologías de la Información (roles Técnico y Administrador), garantizando que cada grupo recibiera la formación específica correspondiente a sus funciones dentro del sistema. Al finalizar cada sesión, se entregó el manual de usuario como material de apoyo para la consulta posterior de instrucciones y funcionalidades.

La Tabla 93 presenta el cronograma de capacitaciones ejecutadas, organizadas por módulo del sistema y tipo de personal al que fue dirigida cada sesión:

Tabla 93. Cronograma de capacitaciones al personal del GAD Municipal de Pelileo

Tema de capacitación	Encargado	Dirigido	Fecha	Realizado
Acceso al sistema y cierre de sesión	Josué Valverde	Personal administrativo y TI	05/05/2026 06/05/2026	Sí
Creación y seguimiento de tickets	Josué Valverde	Personal administrativo	05/05/2026	Sí
Gestión y asignación de tickets	Josué Valverde	Departamento de TI	05/05/2026	Sí
Configuración del sistema (SLA, categorías, departamentos)	Josué Valverde	Jefe de TI (ADMIN)	05/05/2026	Sí
Base de conocimiento	Josué Valverde	Personal técnico (TECNICO)	05/05/2026	Sí
Módulo AnyDesk por departamento	Josué Valverde	Departamento de TI	05/05/2026 06/05/2026	Sí
Dashboard de reportes y KPIs	Josué Valverde	Jefe de TI	05/05/2026	Sí
Calificación del servicio	Josué Valverde	Personal administrativo	05/05/2026	Sí

Las capacitaciones fueron diferenciadas por rol de usuario, abarcando tanto al personal administrativo (rol Usuario) como al Departamento de Tecnologías de la Información (roles Técnico y Administrador), garantizando que cada grupo recibiera la formación específica correspondiente a sus funciones dentro del sistema. Al finalizar cada sesión, se entregó el manual de usuario como material de apoyo para la consulta posterior de instrucciones y funcionalidades.



Figura 41. Sistema Helpdesk en producción

La entrega del manual de usuario institucional en formato digital complementó las sesiones de capacitación, proporcionando al personal del GAD Municipal de Pelileo una guía de referencia permanente para la resolución de dudas operativas durante el período de adopción del sistema.

Resultado de la Fase 5

Las 18 historias de usuario del sistema Helpdesk para el GAD Municipal de Pelileo fueron implementadas y validadas exitosamente con un porcentaje de cumplimiento del 100%. El sistema fue desplegado exitosamente en la infraestructura institucional y el personal capacitado en los 8 módulos del sistema. El proceso de desarrollo bajo la metodología XP, con retroalimentación continua del Departamento de TI en cada iteración, garantizó que el sistema responda directamente a las necesidades identificadas en el diagnóstico del Capítulo II.

La totalidad de los temas programados fueron impartidos satisfactoriamente, logrando que el personal involucrado comprenda el funcionamiento del sistema acorde a su rol asignado, lo que facilitó la adopción inmediata de la herramienta en el proceso de gestión de incidencias institucional.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

El presente capítulo tiene como propósito presentar las conclusiones derivadas del desarrollo de la investigación y del proyecto de implementación del sistema web Helpdesk en el GAD Municipal de Pelileo.

Asimismo, se incluyen una serie de recomendaciones estratégicas orientadas a garantizar la sostenibilidad, escalabilidad y mejora continua del sistema implementado.

4.1 Conclusiones

- El sistema web HelpDesk desarrollado fue implantado directamente en el GAD Municipal de Pelileo, capacitando al personal involucrado en su uso y administración. El sistema cuenta con la capacidad de gestionar las incidencias de manera estructurada mediante tickets con historial completo, cálculo automático de SLA, base de conocimiento, acceso remoto y reportes analíticos en tiempo real. Su implementación eliminó los procesos manuales e informales de soporte técnico, estableciendo un flujo de atención ordenado, trazable y medible.
- La recolección y análisis de datos mediante encuestas, entrevistas y observación directa permitió identificar claramente las deficiencias del proceso de soporte técnico. Del total de 10 riesgos identificados, 7 fueron clasificados como ALTO (puntuación ≥ 10), 2 como MEDIO (5-9) y 1 como BAJO (1-4). Se determinaron dos cuellos de botella principales: la ausencia de un registro formal de incidencias y la inexistencia de control de tiempos de atención y acuerdos de nivel de servicio (SLA).

- Mediante la evaluación multicriterio de herramientas Open Source se seleccionaron e integraron de forma eficiente las siguientes tecnologías: Next.js 14 como framework de frontend, aprovechando su renderizado híbrido, enrutamiento optimizado y su ecosistema React para construir interfaces dinámicas y responsivas; Node.js como entorno de ejecución del servidor, destacando por su arquitectura orientada a eventos y su capacidad de manejar múltiples solicitudes concurrentes de forma eficiente; y Express.js como framework backend, que permitió estructurar la API REST de manera modular y escalable. Complementariamente, se emplearon NextAuth.js con JWT para la autenticación segura por roles y Tailwind CSS para el diseño de la interfaz. Para la gestión de datos se utilizó Oracle Database 19c, adoptado por disposición institucional del GAD Municipal de Pelileo, que ya contaba con la infraestructura y el licenciamiento vigente, lo que eliminó costos adicionales de implementación. Todas estas herramientas comparten compatibilidad y cuentan con comunidades activas de soporte, y permitieron un desarrollo estable y escalable dentro del entorno tecnológico de la institución.
- La aplicación de la metodología de Programación Extrema (XP), adaptada al desarrollo individual, permitió el desarrollo y la organización del trabajo en 5 iteraciones, cada una con pruebas de aceptación definidas. Esta estructura permitió entregar un sistema completamente funcional en 28 días, distribuyendo las 18 historias de usuario de manera progresiva y priorizando los módulos de mayor impacto para el GAD Municipal de Pelileo. La naturaleza iterativa e incremental de XP garantizó una alta adaptabilidad ante cambios en los requerimientos, una comunicación constante con el cliente y una alineación entre el diseño técnico y las necesidades reales de los usuarios finales.

4.2 Recomendaciones

- Se recomienda mantener la metodología de Programación Extrema (XP), adaptada al desarrollo individual, en las futuras iteraciones de mejora del sistema HelpDesk, dado que demostró alta efectividad durante el desarrollo del presente proyecto al gestionar requerimientos variables con participación activa del usuario clave. Para preservar la calidad del código a largo plazo, se sugiere planificar iteraciones de refactorización con una frecuencia mínima anual, con el objetivo de reducir la deuda técnica acumulada y mantener la escalabilidad del sistema conforme crezcan las necesidades del GAD Municipal de Pelileo.
- Desde el punto de vista metodológico, se sugiere explorar la integración de metodologías ágiles escaladas como Scrum para gestionar equipos de desarrollo más grandes. En cuanto a herramientas, se recomienda evaluar la migración hacia bases de datos Open Source, la incorporación de inteligencia artificial para la clasificación automática de tickets, y la implementación de módulos de analítica avanzada.
- Se sugiere al GAD Municipal de Pelileo implementar un plan formal de capacitación continua y gestión del cambio dirigido al personal administrativo, incluyendo sesiones prácticas por departamento, material de apoyo y un periodo de acompañamiento de cuatro semanas, designando además un usuario clave por cada dependencia.
- Se propone planificar la escalabilidad progresiva del sistema incorporando nuevos módulos como inventario de activos de TI, portal de autoservicio e integración con correo institucional, priorizando las dependencias con mayor volumen de incidencias. De forma paralela, se debe establecer un proceso formal de actualización mensual de la base de conocimiento institucional, obligando a los técnicos a documentar las soluciones al cerrar tickets de prioridad alta y media, con el fin de reducir la reincidencia de problemas, optimizar los tiempos de diagnóstico y maximizar el valor del sistema a largo plazo en el GAD Municipal de Pelileo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Red Hat, "The State of Enterprise Open Source 2022," Red Hat, Inc., 2022. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.redhat.com/en/resources/state-of-enterprise-open-source-report-2022>
- [2] HDI, "The State of Tech Support 2023," Help Desk Institute, 2023. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.thinkhdi.com/library/supportworld/2023/state-of-tech-support-landing>
- [3] A. Cubo, J. L. Hernández Carrión, M. Porrúa y B. Roseth, "Guía de transformación digital del gobierno," Banco Interamericano de Desarrollo (BID), ago. 2022, doi: 10.18235/0004402.
- [4] MINTEL, "Plan Nacional de Telecomunicaciones y Tecnologías de Información del Ecuador 2022-2025," Ministerio de Telecomunicaciones y de la Sociedad de la Información, Quito, 2022. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.telecomunicaciones.gob.ec/plan-nacional-de-telecomunicaciones-y-tecnologias-de-informacion-del-ecuador/>
- [5] J. K. Díaz Hidalgo, "Diseño de software de help desk en la nube basado en ITIL para la Cruz Roja Ecuatoriana," bachelor Thesis, Universidad Politécnica Salesiana, Quito, 2024.
- [6] J. G. Lozada Acuña y C. J. Reyes Chalacan, "Desarrollo de un aplicativo web de mesa de ayuda aplicando el marco de referencia ITIL v4 para la gestión de tickets en el Departamento de Secretaría de la Universidad Técnica de Cotopaxi Extensión La Maná," bachelor Thesis, Universidad Técnica de Cotopaxi Extensión La Maná, La Maná, 2022.
- [7] K. A. Caguana Valle, "Help Desk basado en la arquitectura MVC para la gestión de servicios en el área administrativa de la empresa Gala Importaciones Cía. Ltda.," bachelor Thesis, Universidad Técnica de Ambato, Ambato, feb. 2025.
- [8] N. Osman y C. Jimu, "City Council Help Desk Support System," i-manager's Journal on Software Engineering, vol. 15, n.º 3, 2021.
- [9] S. Pérez Romero y F. J. Mendoza Callupe, "Sistema Help Desk: una solución para la gestión de incidencias," Gaceta Científica, vol. 10, n.º 2, pp. 76-81, abr.-jun. 2024, doi: 10.46794/gacien.10.2.2188.
- [10] F. S. Sánchez Casanova y M. Á. Valles Coral, "Influencia de ITIL V3 en la gestión de incidencias de una municipalidad peruana," Revista Cubana de Ciencias Informáticas, vol. 15, n.º 3, pp. 1-19, jul.-sep. 2021. ISSN: 2227-1899.
- [11] D. A. Faria Dominguez y L. Freire Cobo, "Implementación de un sistema de mesa de ayuda informático Help Desk para la gestión de requerimientos que se presentan en un SOC," bachelor Thesis, Escuela Superior Politécnica del Litoral

- (ESPOL), Guayaquil, 2021. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/53616>
- [12] IBM, "¿Qué es la biblioteca de infraestructura de TI (ITIL)?" IBM Corporation, 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/it-infrastructure-library>
- [13] G. C. Carmen Cristina, "Modelo de gestión basado en la metodología ITIL V3 para el servicio de Help Desk orientada al soporte técnico de la unidad de TI de la empresa eléctrica regional norte S.A. EMELNORTE," bachelor Thesis, Universidad Técnica del Norte, Ibarra, 2021.
- [14] IBM, "¿Qué es la gestión de servicios de TI (ITSM)?" IBM Corporation, 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/it-service-management>
- [15] P. X. Moscoso Collantes, "Helpdesk usando herramientas Open Source para la gestión de asistencias técnicas a clientes en la empresa ConnectaServices," bachelor Thesis, Universidad Técnica de Ambato, Ambato, feb. 2025.
- [16] J. Morales-Carrillo, L. Cedeño-Valarezo, J. S. Cajape Bravo y J. G. Ormaza Calderón, "Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática," Revista Ibérica de Sistemas e Tecnologías de Informação, n.º E47, pp. 29-45, ene. 2022. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.proquest.com/docview/2648273778>
- [17] Red Hat, "¿Qué es el open source o código abierto?" Red Hat, Inc., 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.redhat.com/es/topics/open-source/what-is-open-source>
- [18] IBM, "Open source at IBM," IBM Corporation, 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.ibm.com/opensource/>
- [19] Vercel, Inc., "Next.js 14 Documentation," 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://nextjs.org/docs>
- [20] OpenJS Foundation, "Node.js v20 LTS Documentation," 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://nodejs.org/docs/latest-v20.x/api/>
- [21] Oracle Corporation, "Oracle Database 19c Long Term Release Documentation," Oracle Corporation, 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://docs.oracle.com/en/database/oracle/oracle-database/19/>
- [22] Microsoft Corporation, "TypeScript Documentation," 2024. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.typescriptlang.org/docs/>

- [23] R. Hernández Sampieri y C. P. Mendoza Torres, Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta, 1ra ed. Ciudad de México: McGraw-Hill Education, 2018. ISBN: 978-1-4562-6096-5.
- [24] J. Morales-Carrillo, L. Cedeño-Valarezo, J. S. Cajape Bravo y J. G. Ormaza Calderón, "Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática," Revista Ibérica de Sistemas e Tecnologías de Informação, n.º E47, pp. 29-45, ene. 2022. Consultado: el 29 de noviembre de 2025. [En línea]. Disponible en: <https://www.proquest.com/docview/2648273778>
- [25] ISO, "ISO 31000:2018 — Risk Management Guidelines," International Organization for Standardization, Ginebra, 2018.
- [26] D. Castellanos-Perilla, "Toma de decisiones multicriterio en la selección de tecnologías de software: una revisión sistemática de la literatura," Información Tecnológica, vol. 33, n.º 5, pp. 187-200, 2022, doi: 10.4067/S0718-07642022000500187.
- [27] E. Hernández-Chanto y J. Mora-Vargas, "Revisión sistemática de métodos de toma de decisiones multicriterio aplicados en proyectos de ingeniería de software," Tecnología en Marcha, vol. 36, n.º 1, pp. 3-18, ene.-mar. 2023, doi: 10.18845/tm.v36i1.6307.

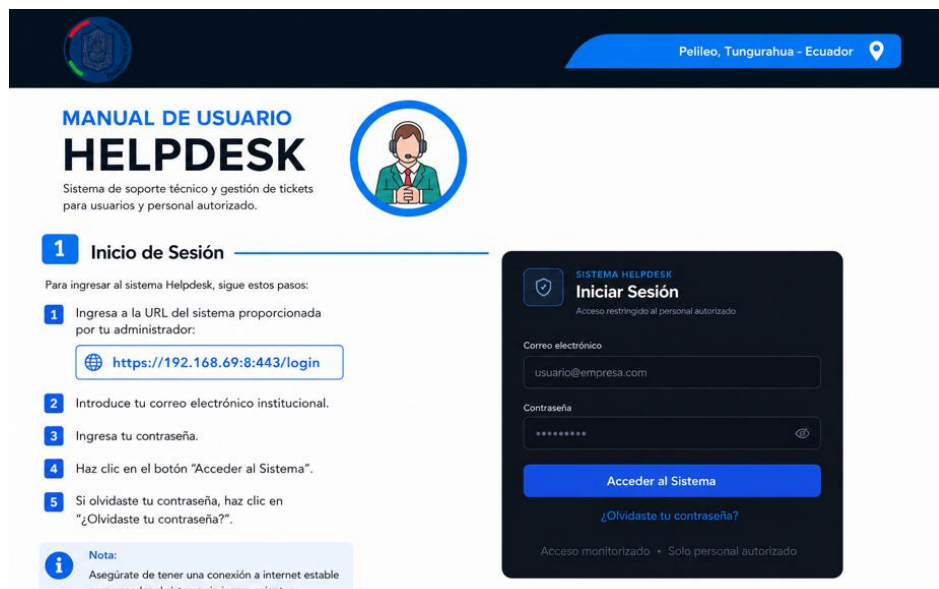
ANEXOS

Anexo A. Manual de usuario



El presente manual describe el funcionamiento del sistema web HelpDesk desarrollado para el Gobierno Autónomo Descentralizado (GAD) Municipal de Pelileo. El sistema centraliza la gestión de incidencias de soporte técnico mediante un Punto Único de Contacto, permitiendo registrar, categorizar, priorizar, asignar y resolver solicitudes.

- *Acceso al sistema*



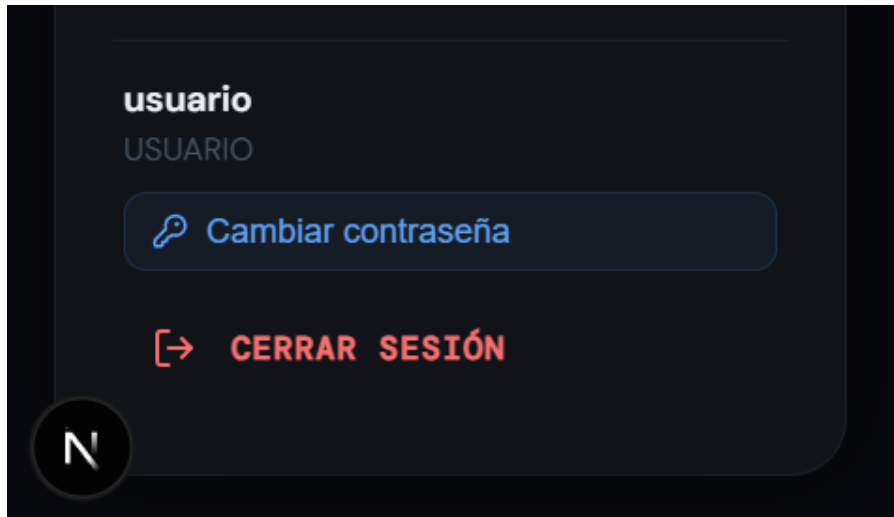
- **Recuperar Contraseña**

En caso de olvidar su contraseña, haga clic en el enlace ¿Olvidó su contraseña? en la pantalla de inicio de sesión. El sistema enviará un correo de recuperación a su cuenta institucional registrada.

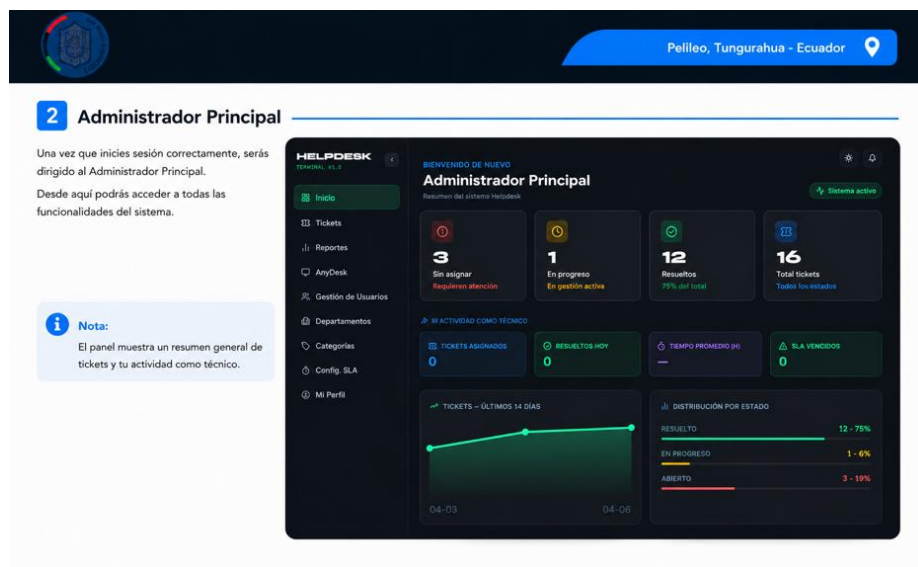
Un correo se enviará con tu nueva contraseña con la cual puedes acceder al sistema.

- **Cierre de sesión**

Para cerrar sesión de forma segura, haga clic inferior izquierda y seleccione la opción Cerrar sesión. Se recomienda siempre cerrar sesión al terminar de usar el sistema, especialmente en equipos compartidos.



- **Pantalla Principal**

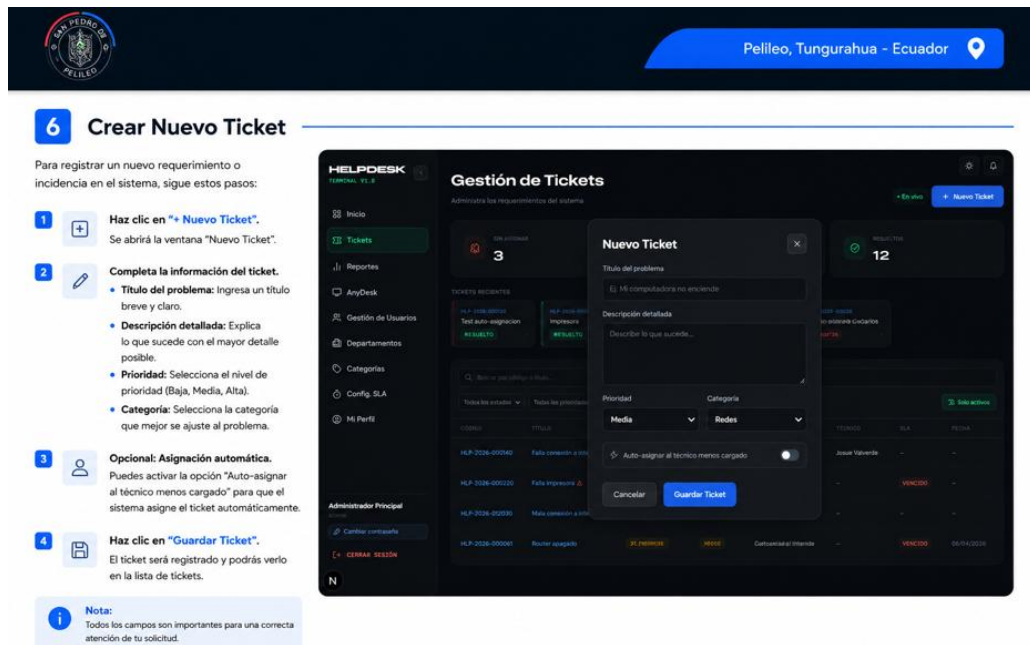


El menú principal está compuesto por dos secciones:

1. Menú de módulos: Sección donde puedes navegar a través del sistema para crear tickets, verificar usuarios y sistemas o realizar reportes

2. Dashboard: Página principal que te muestra los datos más relevantes de tu uso en el sistema.

- *Tickets*



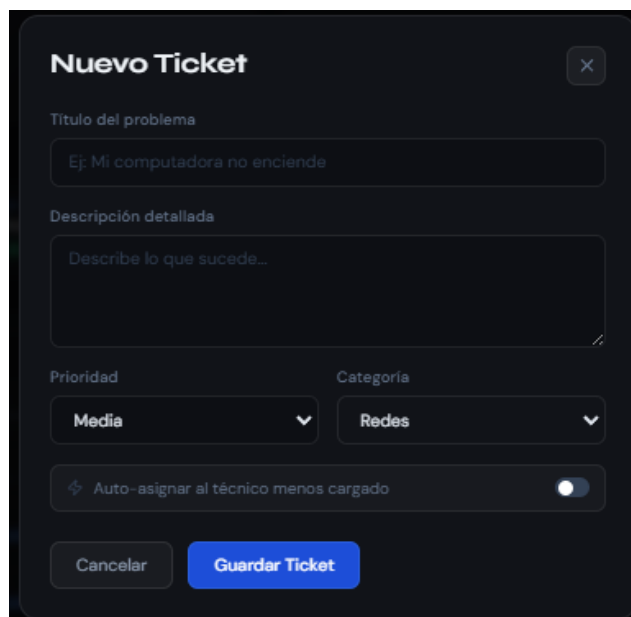
6 Crear Nuevo Ticket

Para registrar un nuevo requerimiento o incidencia en el sistema, sigue estos pasos:

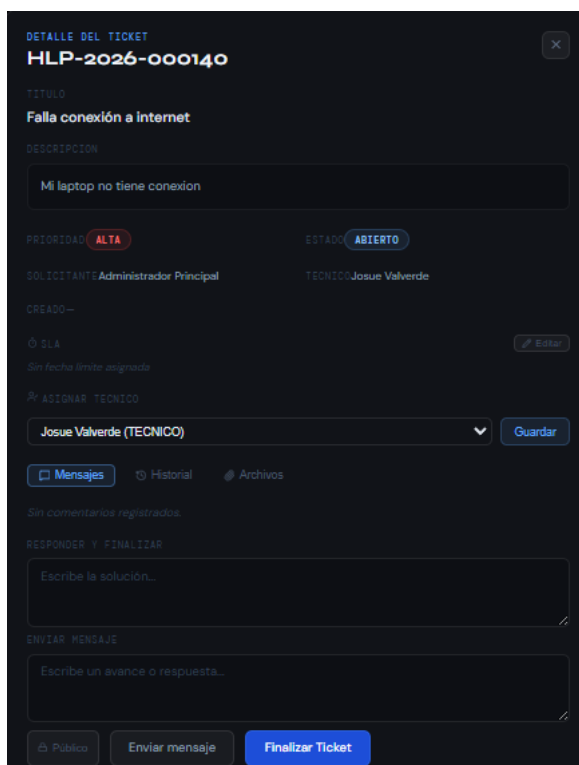
- 1** Haz clic en **"+ Nuevo Ticket"**. Se abrirá la ventana "Nuevo Ticket".
- 2** Completa la información del ticket.
 - **Título del problema:** Ingresar un título breve y claro.
 - **Descripción detallada:** Explica lo que sucede con el mayor detalle posible.
 - **Prioridad:** Selecciona el nivel de prioridad (Baja, Media, Alta).
 - **Categoría:** Selecciona la categoría que mejor se ajuste al problema.
- 3** Opcional: **Asignación automática.** Puedes activar la opción "Auto-asignar al técnico menos cargado" para que el sistema asigne el ticket automáticamente.
- 4** Haz clic en **"Guardar Ticket"**. El ticket será registrado y podrás verlo en la lista de tickets.

Nota: Todos los campos son importantes para una correcta atención de tu solicitud.

Botón para crear ticket: Al hacer clic en este botón, se abrirá una ventana flotante (emergente) que permitirá registrar un nuevo ticket. Todos los datos son obligatorios para poder crear el ticket.



Una vez llenados todos los campos, al hacer clic en el botón “Guardar”, se mostrará el detalle completo del ticket creado.



DETALLE DEL TICKET
HLP-2026-000140

TÍTULO
Falla conexión a internet

DESCRIPCIÓN
Mi laptop no tiene conexión

PRIORIDAD ALTA **ESTADO** ABIERTO

SOLICITANTE Administrador Principal **TECNICO** Josue Valverde

CREADO —

SLA
Sin fecha límite asignada [Editar](#)

ASIGNAR TECNICO
Josue Valverde (TECNICO) [Guardar](#)

[Mensajes](#) [Historial](#) [Archivos](#)

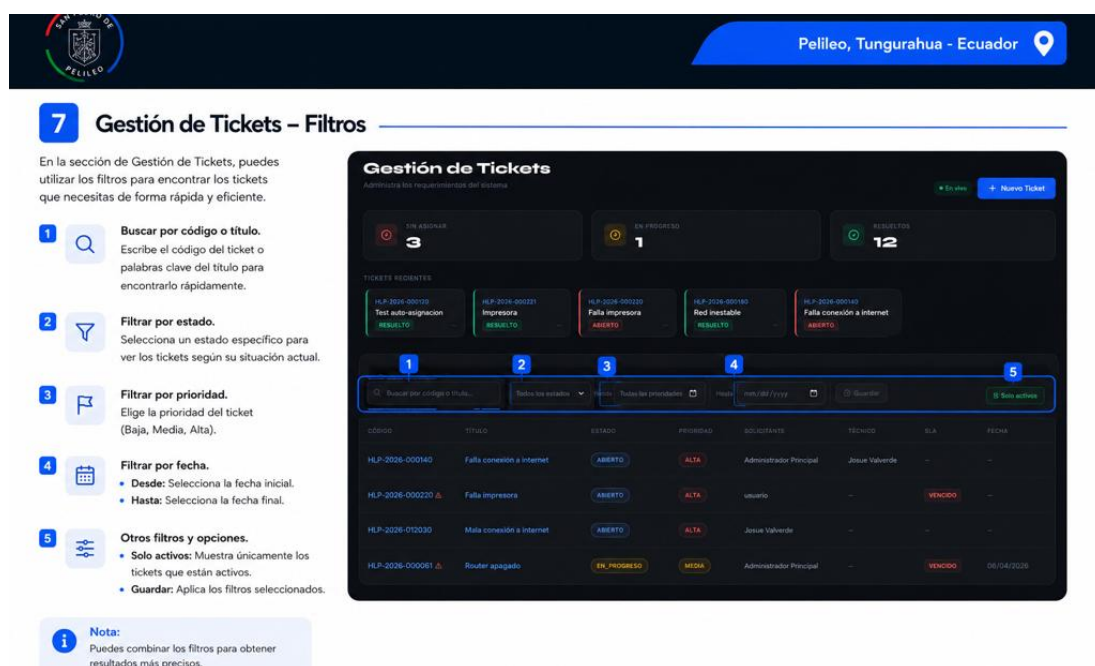
Sin comentarios registrados.

RESPONDER Y FINALIZAR
Escribe la solución...

ENVIAR MENSAJE
Escribe un avance o respuesta...

[Publico](#) [Enviar mensaje](#) [Finalizar Ticket](#)

Para aplicar los filtros con los datos ingresados, debe hacer clic en el botón Buscar (botón azul con símbolo de lupa).



7 Gestión de Tickets – Filtros

En la sección de Gestión de Tickets, puedes utilizar los filtros para encontrar los tickets que necesitas de forma rápida y eficiente.

- 1 Buscar por código o título.**
Escribe el código del ticket o palabras clave del título para encontrarlo rápidamente.
- 2 Filtrar por estado.**
Selecciona un estado específico para ver los tickets según su situación actual.
- 3 Filtrar por prioridad.**
Elige la prioridad del ticket (Baja, Media, Alta).
- 4 Filtrar por fecha.**
 - Desde: Selecciona la fecha inicial.
 - Hasta: Selecciona la fecha final.
- 5 Otros filtros y opciones.**
 - Solo activos: Muestra únicamente los tickets que están activos.
 - Guardar: Aplica los filtros seleccionados.

Nota:
Puedes combinar los filtros para obtener resultados más precisos.

Gestión de Tickets
Administra los tickets reportados por los usuarios

[En vivo](#) [Nuevo Ticket](#)

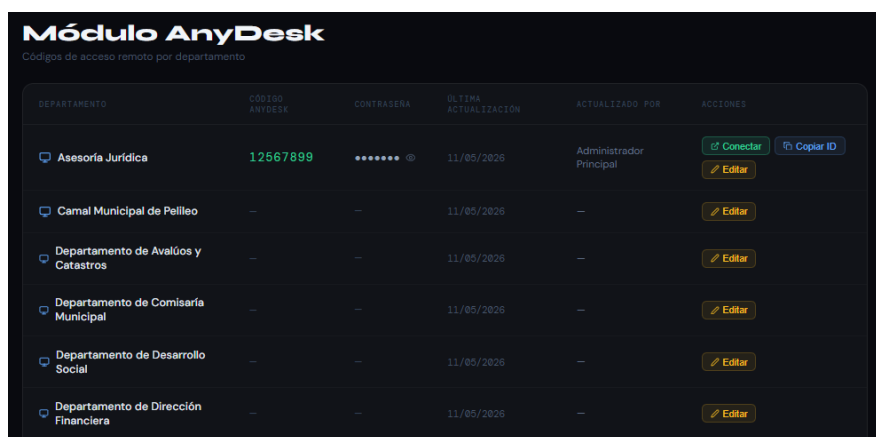
EN ASIGNAR 3 **EN PROGRESO** 1 **RESUELTO** 12

TICKETS VISTANTES

CÓDIGO	TÍTULO	ESTADO	PRIORIDAD	SOLICITANTE	TECNICO	SLA	FECHA
HLP-2026-000140	Falla conexión a internet	ABIERTO	ALTA	Administrador Principal	Josue Valverde	—	—
HLP-2026-000220	Falla impresora	ABIERTO	ALTA	usuario	—	VENCIDO	—
HLP-2026-000230	Mala conexión a internet	ABIERTO	ALTA	Josue Valverde	—	—	—
HLP-2026-000061	Router apagado	EN PROGRESO	MEDIA	Administrador Principal	—	VENCIDO	08/04/2026

- **Módulo AnyDesk**

El sistema integra el módulo de acceso remoto AnyDesk organizado por departamento, permitiendo al técnico identificar rápidamente los equipos disponibles para soporte remoto. Para utilizar este módulo, acceda a la sección AnyDesk del menú lateral, seleccione el departamento correspondiente y conectese al equipo del usuario mediante el ID mostrado.

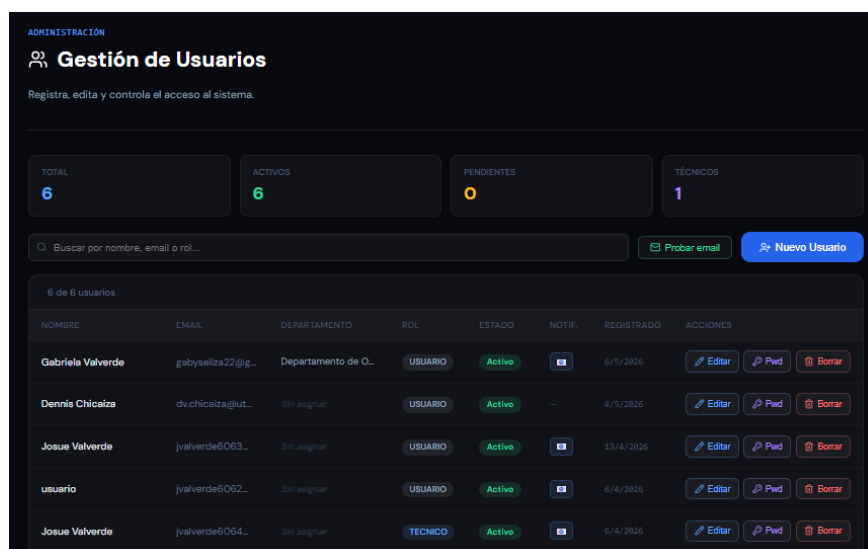


Módulo AnyDesk
Códigos de acceso remoto por departamento

DEPARTAMENTO	CÓDIGO ANYDESK	CONTRASEÑA	ÚLTIMA ACTUALIZACIÓN	ACTUALIZADO POR	ACCIONES
Asesoría Jurídica	12567899	*****@	11/05/2026	Administrador Principal	Conectar Copiar ID Editar
Camal Municipal de Pelileo	—	—	11/05/2026	—	Editar
Departamento de Avalúos y Catastros	—	—	11/05/2026	—	Editar
Departamento de Comisaría Municipal	—	—	11/05/2026	—	Editar
Departamento de Desarrollo Social	—	—	11/05/2026	—	Editar
Departamento de Dirección Financiera	—	—	11/05/2026	—	Editar

- **Módulo Gestión de Usuarios**

El Administrador puede crear, editar y deshabilitar cuentas de usuario desde la sección Gestión de Usuarios. Para crear un nuevo usuario, haga clic en Nuevo Usuario, complete los datos (nombre, correo institucional, departamento y rol asignado) y guarde el registro. El sistema enviará automáticamente las credenciales de acceso al correo del nuevo usuario.



ADMINISTRACIÓN
Gestión de Usuarios
Registra, edita y controla el acceso al sistema.

TOTAL: 6

ACTIVOS: 6

PENDIENTES: 0

TÉCNICOS: 1

Q. Buscar por nombre, email o rol... [Probar email](#) [Nuevo Usuario](#)

6 de 6 usuarios

NOMBRE	EMAIL	DEPARTAMENTO	ROL	ESTADO	NOTIF.	REGISTRADO	ACCIONES
Gabriela Valverde	gabyseliza22@g...	Departamento de O...	USUARIO	Activo	📧	6/5/2026	Editar Pwd Borrar
Dennis Chicaiza	dv.chicaiza@ut...	Sin asignar	USUARIO	Activo	—	4/5/2026	Editar Pwd Borrar
Josue Valverde	jvalverde063...	Sin asignar	USUARIO	Activo	📧	11/4/2026	Editar Pwd Borrar
usuario	jvalverde062...	Sin asignar	USUARIO	Activo	📧	6/4/2026	Editar Pwd Borrar
Josue Valverde	jvalverde064...	Sin asignar	TÉCNICO	Activo	📧	6/4/2026	Editar Pwd Borrar

- **Configuración de SLA**

Los Acuerdos de Nivel de Servicio (SLA) definen los tiempos máximos de respuesta y resolución según la prioridad del ticket. Para configurar los SLA, acceda a Configuración → SLA y defina los tiempos en horas para cada nivel de prioridad (Baja, Media, Alta, Crítica).

- **Dashboard de reportes y KPIs**

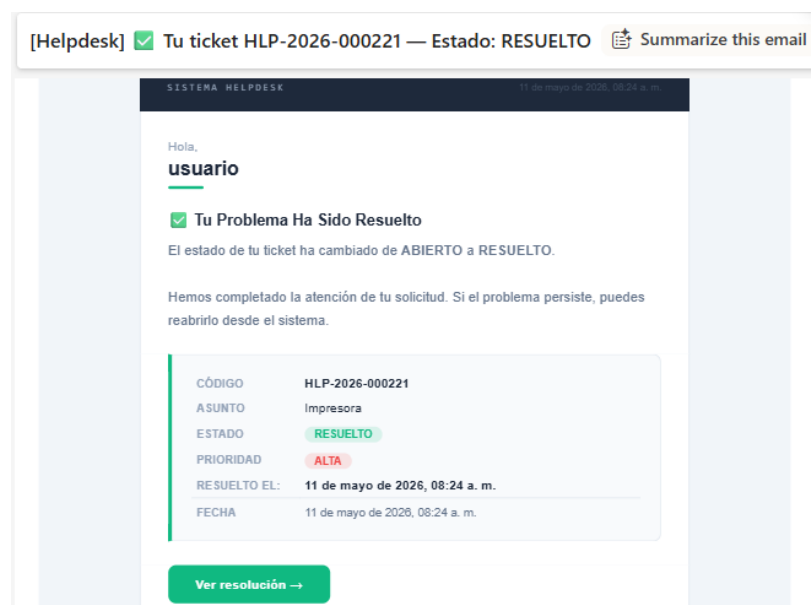
El dashboard de reportes proporciona al Administrador una visión en tiempo real del desempeño del Departamento de TI. Los indicadores clave disponibles incluyen: número de tickets abiertos, en proceso y cerrados; tiempo promedio de resolución por técnico; tickets por categoría y departamento; cumplimiento de SLA; y tendencia de incidencias por período.



Notificaciones del sistema

El sistema envía notificaciones automáticas por correo electrónico en los siguientes eventos:

- Creación de un nuevo ticket: notifica al Departamento de TI.
- Asignación de ticket a un técnico: notifica al técnico asignado.
- Cambio de estado del ticket: notifica al usuario solicitante.



- **Recomendaciones de uso**
- Describa el problema con el mayor detalle posible al crear un ticket, indicando el equipo afectado, el error mostrado y las acciones realizadas antes del fallo.
- No comparta sus credenciales de acceso con otros usuarios. Cada funcionario debe tener su propia cuenta.
- Si el problema es urgente, indique la prioridad Alta o Crítica al crear el ticket y notifique directamente al Departamento de TI.
- Los técnicos deben documentar la solución aplicada en la base de conocimiento al cerrar tickets de prioridad alta o media.