

**Análisis y Desarrollo de sistema de información Help Desk basado en ITIL para  
MiPymes**

**Deiby Andrey Rodriguez Guerrero**

**Trabajo de grado para optar el título de ingeniero mecatrónico**

**Director**

**Luis Fernando Perico Remolina**

**Magíster en controles industriales**

**Universidad Santo Tomás, Bucaramanga**

**División de ingenierías y arquitectura**

**Ingeniería mecatrónica**

**2025**

### **Dedicatoria**

Quiero dedicar este trabajo principalmente a mis amados padres, Olga Guerrero y Hernán Rodríguez Q. E. P. D., quienes, por medio de su amor, educación y esfuerzo (a pesar de las adversidades), me permitieron alcanzar esta meta. Soy lo que ustedes sembraron en mí. A ustedes les debo todo.

A la institución Gonzalo Jiménez Navas, que fue parte fundamental de mi proceso académico y personal.

A mi abuelo José Guerrero (Q. E. P. D.) y a mis tías Maira Guerrero y Marcela Guerrero, quienes me apoyaron en los momentos más difíciles y fueron un respaldo incondicional durante mi etapa universitaria.

A mis amistades de años: Nicolás J. Villamizar, Edwin Suárez, Rafael Tarazona y Jheysson Ortiz, quienes, entre charlas y risas, me han acompañado y ayudado a crecer como ser humano.

A todas esas personas que hicieron parte del proceso y estuvieron cerca apoyándome: compañeros como Silvia y Moisés, con quienes compartimos sobre la vida y nos impulsamos mutuamente a dar lo mejor de nosotros mismos. A William, Edwin Ebrath y Yeison Pacheco, por ser colegas y amigos que brindaron su apoyo en este camino.

Y agradezco especialmente a Laura Camila Oñate Pérez: amiga, colega y mi compañera de vida, a quien tanto admiro; aquella que ha creído en mí incluso cuando yo no lo hacía, y que me ha ayudado a crecer, a amar y sentirme amado. Ha sido fundamental para alcanzar este objetivo.

Agradecido con la vida y con las personas que me han acompañado. Este logro es para todos ustedes.

### **Agradecimientos**

Agradezco principalmente a mi madre, a mis familiares y amigos cercanos, quienes me brindaron su apoyo incondicional a lo largo de esta etapa de mi vida, incluso en los momentos más difíciles.

Agradezco también a la Universidad Santo Tomás por el conocimiento compartido y por brindarme herramientas académicas y humanas que hoy me permiten culminar este proceso.

Asimismo, expreso mi gratitud a los magísteres Deisy Páez, Luis Fernando Perico, José Monroy y Félix Pérez, quienes influyeron tanto en mi formación profesional como personal, y fueron una fuente de inspiración para lograr este objetivo.

## Contenido

|  |    |
|--|----|
| Introducción .....   | 15 |
| 1. Análisis y Desarrollo de sistema de información Help Desk basado en ITIL para MiPymes | 16 |
| 1.1 Planteamiento del problema .....   | 16 |
| 1.2 Justificación.....   | 16 |
| 1.3 Objetivos .....  | 17 |
| 1.3.1 Objetivo general .....   | 17 |
| 1.3.2 Objetivos específicos.....   | 17 |
| 2. Marco referencial .....   | 18 |
| 2.1 Marco teórico .....  | 18 |
| 2.1.1 ITIL.....  | 18 |
| 2.1.2 Help Desk .....  | 20 |
| 2.1.3 Metodologías ágiles (XP).....  | 20 |
| 2.1.3 Normalización .....  | 21 |
| 2.2 Marco conceptual .....   | 23 |
| 2.3 Marco legal.....   | 26 |
| 2.4 Marco Tecnológico y Científico .....   | 27 |
| 3. Método .....  | 29 |
| 3.1 Criterios de Diseño.....   | 29 |
| 3.1.1 Framework Backend .....  | 29 |
| 3.1.2 Gestor de Base de datos .....  | 30 |
| 3.1.3 Historias de usuario.....  | 31 |

|   |    |
|---|----|
| 3.2 Descripción de etapas y tareas .....                              | 32 |
| 3.2.1 Recolección de requerimientos .....                             | 32 |
| 3.2.2 Diseño de base de datos .....                                   | 33 |
| 3.2.3 Desarrollo Frontend y Backend.....                              | 34 |
| 3.3 Descripción de los productos finales.....                         | 35 |
| 4. Resultados .....   | 36 |
| 4.1 Diseño.....   | 37 |
| 4.1.1 Análisis metodología ITIL .....                                 | 38 |
| 4.1.2 Identificación de roles y perfiles .....                        | 43 |
| 4.1.3 Identificación de requerimientos .....                          | 43 |
| 4.1.4 Arquitectura de proyecto .....                                  | 45 |
| 4.1.5 Diseño base de datos .....                                      | 47 |
| 4.1.6 Casos de uso del sistema .....                                  | 47 |
| 4.1.7 Configuración de proyecto con Django .....                      | 48 |
| 4.1.7.1 Configuración servidor.....                                   | 50 |
| 4.1.7.2 Configuración de correo.....                                  | 51 |
| 4.1.7.2 Creación de aplicaciones.....                                 | 54 |
| 4.2 Validación de requerimientos.....                                 | 56 |
| 4.3 Pruebas y resultados .....  | 74 |
| 4.3.1 Pruebas a usuarios y valoración de experiencia de usuario ..... | 75 |
| 4.3.2 Resultados de encuesta de satisfacción .....                    | 82 |
| 4.3.3 Impacto en la productividad.....                                | 83 |
| 4.3.4 Resultados esperados.....                                       | 85 |

|   |    |
|---|----|
| ANÁLISIS Y DESARROLLO DE SISTEMA DE INFORMACIÓN HELP DESK | 6  |
| 4.3.5 Casos de uso e historias de usuario .....           | 85 |
| 4.3.6 Diagrama de base de datos .....                     | 85 |
| 4.3.7 Software funcional .....                            | 86 |
| 5. Discusión.....   | 86 |
| 6. Conclusiones .....                                     | 87 |
| 7. Trabajos futuros .....                                 | 88 |
| Referencias.....  | 89 |

**Lista de tablas**

|   |    |
|---|----|
| <b>Tabla 1.</b> <i>Comparación Framework Backend.</i> .....                     | 29 |
| <b>Tabla 2.</b> <i>Comparación de gestor de bases de datos.</i> .....           | 30 |
| <b>Tabla 3.</b> <i>Requerimientos funcionales del sistema.</i> .....            | 43 |
| <b>Tabla 4.</b> <i>Requerimientos no funcionales del sistema.</i> .....         | 45 |
| <b>Tabla 5.</b> <i>Resultados de encuesta de satisfacción al usuario.</i> ..... | 82 |
| <b>Tabla 6.</b> <i>Comparación de impacto en la productividad.</i> .....        | 83 |
| <b>Tabla 7.</b> <i>Resultados esperados del proyecto.</i> .....                 | 85 |

**Lista de figuras**

|   |    |
|---|----|
| <b>Figura 1.</b> <i>Cadena de valor ITIL.</i> .....                                       | 19 |
| <b>Figura 2.</b> <i>Las 4 dimensiones de ITIL.</i> .....                                  | 20 |
| <b>Figura 3.</b> <i>Desarrollo ágil.</i> .....  | 21 |
| <b>Figura 4.</b> <i>Ciclo de liberación XP.</i> .....                                     | 21 |
| <b>Figura 5.</b> <i>Tarjetas de historias de usuario.</i> .....                           | 31 |
| <b>Figura 6.</b> <i>Proceso de identificación de requerimientos.</i> .....                | 33 |
| <b>Figura 7.</b> <i>Proceso para creación y normalización de base de datos.</i> .....     | 34 |
| <b>Figura 8.</b> <i>Procedimiento de desarrollo de software.</i> .....                    | 35 |
| <b>Figura 9.</b> <i>Flujo de resolución de incidentes.</i> .....                          | 37 |
| <b>Figura 10.</b> <i>Prácticas de gestión tecnológica marco ITIL.</i> .....               | 38 |
| <b>Figura 11.</b> <i>Roles y funciones del sistema.</i> .....                             | 43 |
| <b>Figura 12.</b> <i>Flujo de información patrón de arquitectura MVC.</i> .....           | 45 |
| <b>Figura 13.</b> <i>Flujo de información patrón de arquitectura MTV de Django.</i> ..... | 46 |
| <b>Figura 14.</b> <i>Diseño de base de datos relacional.</i> .....                        | 47 |
| <b>Figura 15.</b> <i>Casos de uso del sistema.</i> .....                                  | 48 |
| <b>Figura 16.</b> <i>Creación de entorno virtual con Python.</i> .....                    | 48 |
| <b>Figura 17.</b> <i>Instalación de Django.</i> .....                                     | 49 |
| <b>Figura 18.</b> <i>Creación de proyecto con Django.</i> .....                           | 49 |
| <b>Figura 19.</b> <i>Archivos al crear proyecto.</i> .....                                | 49 |
| <b>Figura 20.</b> <i>Creación de base de datos en PostgreSQL.</i> .....                   | 50 |
| <b>Figura 21.</b> <i>Configuración de base de datos en Django.</i> .....                  | 51 |
| <b>Figura 22.</b> <i>Configuración de correo en Django.</i> .....                         | 52 |

|  |    |
|--|----|
| <b>Figura 23.</b> <i>Configuración contraseña de aplicaciones Google.</i> .....                    | 53 |
| <b>Figura 24.</b> <i>Creación de contraseña de aplicaciones.</i> .....                             | 53 |
| <b>Figura 25.</b> <i>Contraseña de aplicación de Google.</i> .....                                 | 53 |
| <b>Figura 26.</b> <i>Creación de aplicaciones en Django.</i> .....                                 | 54 |
| <b>Figura 27.</b> <i>Archivos de una aplicación en Django.</i> .....                               | 54 |
| <b>Figura 28.</b> <i>Sección para agregar aplicaciones en Django.</i> .....                        | 55 |
| <b>Figura 29.</b> <i>Configuración de templates en Django.</i> .....                               | 56 |
| <b>Figura 30.</b> <i>Requerimiento funcional 1.</i> .....  | 56 |
| <b>Figura 31.</b> <i>Login.</i> .....  | 57 |
| <b>Figura 32.</b> <i>Requerimiento funcional 2.</i> .....  | 57 |
| <b>Figura 33.</b> <i>Formulario de creación de usuarios.</i> .....                                 | 58 |
| <b>Figura 34.</b> <i>Listado de usuarios registrados con edición y estado de activación.</i> ..... | 58 |
| <b>Figura 35.</b> <i>Requerimiento funcional 3.</i> .....  | 58 |
| <b>Figura 36.</b> <i>Formulario para recuperar contraseña.</i> .....                               | 59 |
| <b>Figura 37.</b> <i>Correo enviado para recuperar la contraseña.</i> .....                        | 59 |
| <b>Figura 38.</b> <i>Formulario de cambio de contraseña.</i> .....                                 | 59 |
| <b>Figura 39.</b> <i>Requerimiento funcional 4.</i> .....  | 60 |
| <b>Figura 40.</b> <i>Formulario para notificar incidentes.</i> .....                               | 60 |
| <b>Figura 41.</b> <i>Clase del incidente.</i> .....  | 61 |
| <b>Figura 42.</b> <i>Requerimiento Funcional 5.</i> .....  | 61 |
| <b>Figura 43.</b> <i>Historial de tickets generados por el usuario.</i> .....                      | 61 |
| <b>Figura 44.</b> <i>Requerimiento funcional 6.</i> .....  | 62 |
| <b>Figura 45.</b> <i>Formulario de creación o registro de activos TI.</i> .....                    | 62 |

|   |    |
|---|----|
| ANÁLISIS Y DESARROLLO DE SISTEMA DE INFORMACIÓN HELP DESK                                   | 10 |
| <b>Figura 46.</b> <i>Requerimiento funcional 7.</i> .....                                   | 62 |
| <b>Figura 47.</b> <i>Pestaña para creación de activos TI.</i> .....                         | 63 |
| <b>Figura 48.</b> <i>Requerimiento funcional 8.</i> .....                                   | 63 |
| <b>Figura 49.</b> <i>Calificación del servicio del empleado.</i> .....                      | 64 |
| <b>Figura 50.</b> <i>Calificación guardada.</i> .....                                       | 64 |
| <b>Figura 51.</b> <i>Calificación recibida en el área técnica.</i> .....                    | 65 |
| <b>Figura 52.</b> <i>Requerimiento funcional 9.</i> .....                                   | 65 |
| <b>Figura 53.</b> <i>Pestaña de asignación de acuerdos de niveles de servicio.</i> .....    | 66 |
| <b>Figura 54.</b> <i>Requerimiento funcional 10.</i> .....                                  | 66 |
| <b>Figura 55.</b> <i>Requerimiento funcional 11.</i> .....                                  | 66 |
| <b>Figura 56.</b> <i>Chat de resolución de incidentes.</i> .....                            | 67 |
| <b>Figura 57.</b> <i>Requerimiento funcional 12.</i> .....                                  | 67 |
| <b>Figura 58.</b> <i>Hoja de vida de los activos de TI.</i> .....                           | 68 |
| <b>Figura 59.</b> <i>Requerimiento funcional 13.</i> .....                                  | 68 |
| <b>Figura 60.</b> <i>Generación de ticket.</i> .....  | 69 |
| <b>Figura 61.</b> <i>Correo de generación de ticket.</i> .....                              | 69 |
| <b>Figura 62.</b> <i>Correo de cierre de ticket.</i> .....                                  | 69 |
| <b>Figura 63.</b> <i>Requerimiento funcional 14.</i> .....                                  | 69 |
| <b>Figura 64.</b> <i>Decoradores antes de cada función para restringir el acceso.</i> ..... | 70 |
| <b>Figura 65.</b> <i>Error 404 para acceso a módulos.</i> .....                             | 70 |
| <b>Figura 66.</b> <i>Requerimiento funcional 15.</i> .....                                  | 71 |
| <b>Figura 67.</b> <i>Presentación de información de incidentes.</i> .....                   | 71 |
| <b>Figura 68.</b> <i>Requerimiento funcional 16.</i> .....                                  | 71 |

|  |    |
|--|----|
| ANÁLISIS Y DESARROLLO DE SISTEMA DE INFORMACIÓN HELP DESK                      | 11 |
| <b>Figura 69.</b> <i>Creación de áreas de trabajo.</i> .....                   | 72 |
| <b>Figura 70.</b> <i>Requerimiento funcional 17.</i> .....                     | 72 |
| <b>Figura 71.</b> <i>Formulario de mantenimiento a realizar.</i> .....         | 73 |
| <b>Figura 72.</b> <i>Requerimiento no funcional 1.</i> .....                   | 73 |
| <b>Figura 73.</b> <i>Login responsivo y presentación de información.</i> ..... | 74 |
| <b>Figura 74.</b> <i>Requerimiento no funcional 2.</i> .....                   | 74 |
| <b>Figura 75.</b> <i>Personas encuestadas y sus roles definidos.</i> .....     | 75 |
| <b>Figura 76.</b> <i>Primera pregunta de la encuesta.</i> .....                | 76 |
| <b>Figura 77.</b> <i>Resultados primera pregunta de la encuesta.</i> .....     | 76 |
| <b>Figura 78.</b> <i>Segunda pregunta de la encuesta.</i> .....                | 77 |
| <b>Figura 79.</b> <i>Resultados segunda pregunta de la encuesta.</i> .....     | 77 |
| <b>Figura 80.</b> <i>Tercera pregunta de la encuesta.</i> .....                | 78 |
| <b>Figura 81.</b> <i>Resultados tercera pregunta de la encuesta.</i> .....     | 78 |
| <b>Figura 82.</b> <i>Cuarta pregunta de la encuesta.</i> .....                 | 79 |
| <b>Figura 83.</b> <i>Resultados cuarta pregunta de la encuesta.</i> .....      | 79 |
| <b>Figura 84.</b> <i>Quinta pregunta de la encuesta.</i> .....                 | 80 |
| <b>Figura 85.</b> <i>Resultados quinta pregunta de la encuesta.</i> .....      | 80 |
| <b>Figura 86.</b> <i>Sexta pregunta de la encuesta.</i> .....                  | 81 |
| <b>Figura 87.</b> <i>Resultados sexta pregunta de la encuesta.</i> .....       | 81 |
| <b>Figura 88.</b> <i>Séptima pregunta de la encuesta.</i> .....                | 82 |
| <b>Figura 89.</b> <i>Resultados séptima pregunta de la encuesta.</i> .....     | 82 |

### **Resumen**

En la actualidad, la gestión eficiente de los recursos tecnológicos es un factor clave para el crecimiento y la competitividad de las empresas. Este proyecto de tesis propone el desarrollo de un sistema de información Help Desk, basado en la metodología ITIL, orientado a micro, pequeñas y medianas empresas (MiPymes) en Colombia. El objetivo es optimizar la gestión de incidentes, solicitudes e inventario tecnológico, mejorando así la eficiencia operativa y la productividad de estas organizaciones. Se plantea el desarrollo de esta herramienta tecnológica como una aplicación web, debido a su alta escalabilidad y accesibilidad, lo que permite que el sistema sea de fácil uso para cualquier MiPyme. En la etapa inicial del desarrollo, se realiza un análisis de requerimientos con base en la metodología ITIL, contemplando las principales necesidades de las MiPymes. Posteriormente, a partir del levantamiento de requerimientos, se desarrolla la herramienta utilizando Django para el backend, y HTML, CSS, Bootstrap y JavaScript para el frontend y usando como gestor de base de datos se emplea PostgreSQL. En la etapa final, se llevaron a cabo pruebas de funcionamiento sobre la herramienta, lo que permitió implementar mejoras y optimizar su desempeño.

*Palabras clave:* Help Desk, ITIL, Desarrollo Web, MiPymes, Gestión tecnológica

### **Abstract**

Inowadays, the efficient management of technological resources is a key factor for the growth and competitiveness of companies. This thesis project proposes the development of a Help Desk information system, based on the ITIL methodology, oriented to micro, small and medium enterprises (MSMEs) in Colombia. The objective is to optimize the management of incidents, requests and technological inventory, thus improving the operational efficiency and productivity of these organizations. The development of this technological tool is proposed as a web application, due to its high scalability and accessibility, which allows the system to be easy to use for any MSME. In the initial stage of development, a requirements analysis is carried out based on the ITIL methodology, considering the main needs of MSMEs. Subsequently, from the requirements survey, the tool is developed using Django for the backend, and HTML, CSS, Bootstrap and JavaScript for the frontend and using PostgreSQL as database manager. In the final stage, functional tests were carried out on the tool, which allowed implementing improvements and optimizing its performance.

*Keywords:* Help Desk, ITIL, Web Development, Small and Medium Businesses, Technology Management

## Glosario

*MiPymes*: Las micro, pequeñas y medianas empresas (Mipymes) son actores estratégicos en el crecimiento de la economía. (Departamento Nacional de Planeación, 2024).

*Incidentes tecnológicos*: La gestión de incidentes se refiere a la práctica de gestionar los servicios de TI que causan interrupciones. (SolarWinds, 2023).

*Django*: Es un marco web Python de alto nivel que fomenta el desarrollo rápido. (Django Software Foundation, 2024).

*Gestión de servicios de TI*: Es la forma en que los equipos de TI gestionan la prestación integral de los servicios de TI. (Atlassian, 2024b).

*Gestor de base de datos*: Consiste en una agrupación de datos interrelacionados y un conjunto de programas para acceder a dichos datos (Silberschatz, Korth & Sudarshan, 2006).

*ITIL*: Es una guía adoptada para la implementación de la gestión de servicios de TI (AXELOS, 2019).

*Activos TI*: Es hardware, sistemas de software o información que tienen valor para una organización. (Atlassian, 2024a).

*Help Desk*: Captura la demanda de resolución de incidentes y solicitudes tecnológicas. La idea principal de dicha práctica es centralizar la información (AXELOS, 2019).

## **Introducción**

El aprovechamiento de los recursos tecnológicos es un elemento clave para la mejora continua de los procesos y el aumento de la eficiencia operativa en las compañías. En este contexto, el desarrollo de un sistema de información Help Desk tiene como objetivo centralizar y gestionar la infraestructura tecnológica de las organizaciones, al tiempo que genera valor.

Este trabajo se fundamenta en la metodología ITIL y propone el desarrollo de un sistema de información de bajo costo orientado a las micro, pequeñas y medianas empresas (MiPymes), implementado en un entorno web. La solución planteada se basa en principios de seguridad de la información y busca optimizar los procesos tecnológicos internos de las organizaciones.

El proyecto surge como respuesta a la necesidad de ofrecer alternativas tecnológicas accesibles y adaptables, dependiendo del nivel de madurez digital de cada empresa. A lo largo del documento, se exploran las necesidades básicas para una gestión tecnológica eficiente, así como los beneficios esperados con la implementación del sistema.

Finalmente, se desglosa la metodología de desarrollo empleada para alcanzar los objetivos del proyecto, abarcando las etapas de investigación, diseño, implementación y evaluación de la funcionalidad del sistema.

## **1. Análisis y Desarrollo de sistema de información Help Desk basado en ITIL para MiPymes**

### **1.1 Planteamiento del problema**

En muchas micro, pequeñas y medianas empresas (MiPymes) en Colombia, la gestión de incidencias y solicitudes de soporte técnico se realizan de manera informal o con sistemas inadecuados, lo que resulta en una baja eficiencia operativa, largos tiempos de resolución de problemas y manejo inadecuado de la información. La falta de un sistema estructurado y estandarizado conforme las mejores prácticas internacionales, como ITIL, contribuye a estos problemas, generando desorganización, falla en las respuestas de incidencias y respuestas inadecuadas a las necesidades de soporte técnico. Esta situación no afecta solo la productividad de las empresas, sino también su competitividad en el mercado.

Debido a esta problemática se tiene como objetivo principal centralizar y gestionar la infraestructura tecnológica de cualquier compañía. Además de gestionar la infraestructura tecnológica se busca generar valor a la organización, mediante la implementación de la metodología ITIL. Para esto es importante responder la pregunta ¿Cómo se puede desarrollar un sistema que cumpla con la metodología ITIL y que sea escalable para cualquier organización?

### **1.2 Justificación**

La tecnología actualmente avanza muy rápido; la computación en la nube y la inteligencia artificial son componentes tecnológicos importantes que han afectado a todas las empresas, esto abre la puerta a la gestión de estos servicios y demás recursos dentro de las organizaciones, sino se tiene una buena gestión puede afectar temas como productividad y competitividad.

El marco de gestión de servicios de TI (tecnologías de la información) que ofrece ITIL, incluye procesos estandarizados sobre la gestión de incidencias, inventario, solicitudes, cambios y problemas. Adoptar este marco dentro de una organización mejorara significativamente la eficiencia operativa estableciendo procedimientos claros y asegurar una comunicación efectiva entre las diversas áreas de las empresas.

Este proyecto no solo aborda las deficiencias en el sector colombiano de las MiPymes en la gestión de los recursos tecnológicos, sino que también busca proporcionar una oportunidad para mejorar la productividad contribuyendo al crecimiento y desarrollo económico de la región.

### **1.3 Objetivos**

#### ***1.3.1 Objetivo general***

Desarrollar un sistema de información Help desk basado en ITIL para MiPymes, con el fin de optimizar la gestión de incidentes, inventario y solicitudes, mejorando la eficiencia operativa.

#### ***1.3.2 Objetivos específicos***

Identificar requerimientos funcionales y no funcionales de acuerdo a las deficiencias en los procesos actuales de gestión de incidentes y solicitudes en las MiPymes, para el desarrollo del nuevo sistema basado que debe cumplir las mejores prácticas basadas en ITIL.

Construir base de datos relacional cumpliendo reglas de normalización para mantener la integridad de la información manejadas en el sistema con el fin de cumplir con normas de seguridad de la información en las MiPymes.

Desarrollar sistema de información en entorno web conforme a las mejores prácticas de ITIL, que incluya módulos de gestión de incidencias, inventario y solicitudes, para garantizar una solución integrada y adaptada a las necesidades de las MiPymes.

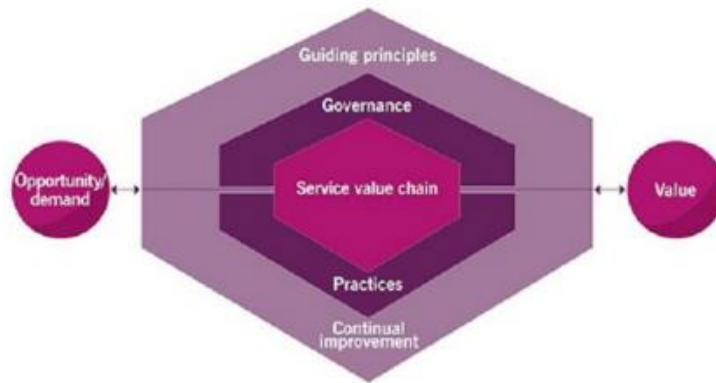
## **2. Marco referencial**

### **2.1 Marco teórico**

#### ***2.1.1 ITIL***

Las siglas ITIL en inglés significan Information Technology Infrastructure Library, es una guía adoptada para la implementación de la gestión de servicios de TI (ITSM) para cualquier organización. Esta guía es ampliamente usada ya que ofrece una visión clara y mejores prácticas para la gestión de servicios creando flujos de valor, transformación digital, entre otros beneficios para la organización [3].

La creación de valor de ITIL, proporciona un modelo operativo para la creación, entrega y mejora continua de los servicios. Este modelo lo que ofrece es guiar a la toma de decisiones y garantizar un entendimiento común para la gestión de servicios de la organización. Partiendo de la organización de los recursos de TI, documentación de los procesos, funciones y roles del IT Service Management (ITSM) [3].

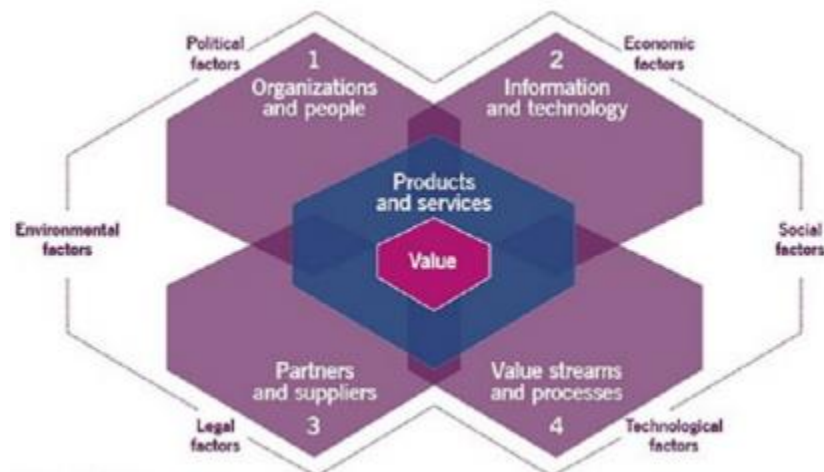
**Figura 1.** Cadena de valor ITIL.

Adaptado de ITIL 4 [3].

Para la implementación de la gestión de servicios de TI se debe realizar una planificación y preparación de un efectivo uso de las cuatro dimensiones fundamentales de ITIL o también conocido como las 4 P's:

- Organizations and people (Personas): Habilidades y competencias de los miembros para los servicios de TI.
- Information and technology (Productos): Componentes tecnológicos y sistemas de gestión usados para la administración de los servicios de TI.
- Partners and suppliers (Asociados): Vendedores, fabricantes o proveedores que se requieren para asistir o soportar el suministro de los servicios de TI.
- Value streams and process (Proceso): Procesos, actividades y roles que se encuentran involucrados para la gestión de los servicios de TI [3].

**Figura 2.** Las 4 dimensiones de ITIL.



Adaptado de ITIL 4 [3].

### 2.1.2 Help Desk

La mesa de ayuda o también nombrado Service Desk por ITIL, es capturar la demanda de resolución de incidentes y solicitudes tecnológicos. La idea principal de dicha práctica es centralizar la información y que sea el único punto de contacto para no generar brechas de seguridad.

Según ITIL, este Service Desk debe manejar acuerdos de niveles de servicio (ANS) para distribuir los incidentes y exista una mayor organización en su estructura de gestión [3].

### 2.1.3 Metodologías ágiles (XP)

El desarrollo de software mediante metodologías ágiles considera el diseño y la implementación como actividades importantes en el proceso del software. De esta parten varias líneas de las cuales se encuentran el diseño, implementación, adquisición de requerimientos y pruebas. Es un proceso cíclico el cual produce mejoras en el desarrollo [21].

**Figura 3.** *Desarrollo ágil.*

Adaptado de Ingeniería de software [21].

Una de las metodologías más utilizadas es la programación extrema (XP) donde su enfoque es el desarrollo iterativo, donde las practicas se llevan a niveles “extremos” en los cuales se agiliza y prioriza la entrega de mejoras.

**Figura 4.** *Ciclo de liberación XP.*

Adaptado de Ingeniería de software [21].

Usando la programación extrema se debe tener en cuenta la ingeniería de requerimientos y lo plasmamos en historias de usuario las cuales se desglosan en tareas para el desarrollo (ver Figura 4). Luego de implementar se evalúa si se realiza una mejora a la implementación o se sigue con otra historia de usuario. Esta metodología permite el desarrollo ágil con entregas casi inmediatas, dando la facilidad de elaborar un proyecto en menos tiempo comparado a otras metodologías de ágiles de desarrollo y en la cual el cliente siempre está involucrado para la aceptación del producto. [21].

### 2.1.3 Normalización

El objeto principal de la normalización es diseñar bases de datos relaciones las cuales permitan almacenar información sin redundancias innecesarias, además dicha información se debe recuperar fácilmente. Esto se logra mediante la normalización de las bases de datos las cuales son reglas que ofrecen minimizar la redundancia, centralizar e información entendible y de fácil acceso. Las formas normales más importantes son:

**Primera forma normal (1FN):** Esta primera forma normal habla de la atomicidad en la cual se considera que los elementos son unidades indivisibles, es decir, que un conjunto de datos debe ser único y no se debe repetir grupos de manera innecesaria.

**Segunda forma Normal (2FN):** Esta forma normal busca eliminar la dependencia transitiva con datos dentro de una misma tabla. Las dependencias transitivas son datos importantes que describen otros datos que no dependen de la llave primaria. Esta redundancia es importante separarla para obtener una base de datos más estructurada y accesible.

**Tercera forma normal (3FN):** Para esta forma normal indirectamente debe estar en la 2FN, ya que esta consiste en eliminar las dependencias transitivas definidas anteriormente como aquellas que son datos que dependen de otros que no son llave primaria. En esta forma normal, se garantiza la funcionalidad dependiente de las llaves primarias, esto provee un esquema limpio y de fácil acceso.

Existen más formas normales para normalizar una base de datos, pero esto puede llegar a generar un esquema complejo e ineficiente. Por lo tanto, en este trabajo se define hasta la tercera forma normal, ya que con esto se cumple con lo suficiente para una base de datos normalizada [9].

## 2.2 Marco conceptual

*Gestor de base de datos:* Un sistema gestor de bases de datos (SGBD) consiste en una agrupación de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Dicha agrupación de datos, normalmente es llamada base de datos, la cual contiene información importante para una empresa u organización. El principal objetivo de un SGBD es tener una forma de almacenar la información y recuperar la información de una base de datos de manera práctica y eficiente [9].

*MiPymes:* Las micro, pequeñas y medianas empresas (Mipymes) son actores estratégicos en el crecimiento de la economía, la transformación del aparato productivo nacional y el mejoramiento de la posición competitiva del país. Además, las Mipymes contribuyen a reducir la pobreza y la inequidad, al ser alternativas de generación de empleo, ingresos y activos para un gran número de personas [5].

*Software libre:* «Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el «software libre» es una cuestión de libertad, no de precio.

Promovemos estas libertades porque todos merecen tenerlas. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que este hace. Cuando los usuarios no controlan el programa, decimos que dicho programa «no es libre», o que es «privativo». Un programa que no es libre controla a los usuarios, y el programador controla el programa, con lo cual el programa resulta ser un instrumento de poder injusto [23].

*Incidentes tecnológicos:* La gestión de incidentes se refiere a la práctica de gestionar los servicios de TI que causan interrupciones. También implica restaurar los servicios a su estado

normal sin afectar a los ANS. El proceso comienza cuando el usuario final informa de un problema y termina cuando se resuelve mediante una respuesta o acción rápida del servicio de TI. Entre las principales ventajas de la gestión de incidentes se encuentran la identificación y prevención proactivas de incidentes graves, la mejora de la productividad, unos niveles de servicio constantes, una mayor visibilidad de los problemas conocidos, etc. La gestión de incidentes ayuda a mitigar los daños y a prevenir futuros incidentes, y puede permitir a las empresas cumplir las normas de conformidad y reglamentación [14].

*Historias de usuario:* Las historias de usuario son descripciones cortas y simples de una característica contada desde la perspectiva de la persona que desea la nueva capacidad, generalmente un usuario o cliente del sistema. Por lo general, siguen una plantilla simple:

- Como <Usuario>
- Quiero <algún objetivo>
- Para que <motivo>

Las historias de los usuarios a menudo se escriben en fichas o notas adhesivas, se almacenan en una caja y se organizan en paredes o mesas para facilitar la planificación y el debate. Como tal, cambian fuertemente el enfoque de escribir sobre las características a discutir. De hecho, estas discusiones son más importantes que cualquier texto que se escriba [13].

*Casos de uso:* Los casos de uso se crean para refinar un conjunto de requisitos basados en un rol o tarea. En lugar de la lista tradicional de requisitos que pueden no abordar directamente el uso de la solución, los casos de uso agrupan requisitos comunes basados en el tipo de rol u objetivo. Los casos de uso definen lo que los usuarios o roles están haciendo en la solución, un proceso de negocio define cómo realizan estas funciones.

Un caso de uso representa la lista de tareas que los actores pueden realizar y está directamente relacionado con los requisitos del proceso de negocio. Los casos de uso son un reconocimiento de los requisitos que debe cumplir el proyecto. Para documentar un caso de uso, defina los requisitos de propósito, proporcione una introducción y liste los distintos actores o roles para un escenario determinado [4].

*Activos TI:* En palabras sencillas, un activo de TI es hardware, sistemas de software o información que tienen valor para una organización [1]. La gestión de activos de TI es el proceso de gestión eficiente de los equipos de TI durante su ciclo de vida o durante el período de propiedad para maximizar su valor para el negocio.

También conocida como «gestión del ciclo de vida de activos de TI» o simplemente «gestión del ciclo de vida del activo», la ITAM es un marco diseñado para gestionar de manera proactiva y estratégica la adquisición, el uso, el mantenimiento y la retirada de los activos de TI [10].

*Gestión de servicios de TI (ITSM):* La gestión de servicios de TI o ITSM por sus siglas en inglés (IT Service Managment), es sencillamente la forma en que los equipos de TI gestionan la prestación integral de los servicios de TI [2].

*Framework Django:* Django es un marco web Python de alto nivel que fomenta el desarrollo rápido. Django se encarga de gran parte del desarrollo web o molestias que puede generar este permite un código limpio y escribir un entrono web en cuestión de horas. Este framework es gratis y de código abierto [7].

### **2.3 Marco legal**

*Decreto 1078 de 2015:* Este decreto único reglamentario del sector de tecnologías de la información y las Comunicaciones. En el cual se da las normativas para la infraestructura tecnológica y de comunicaciones a nivel nacional, determinando entes regulatorios y responsables de la administración y el cumplimiento del decreto. Este decreto habla del manejo de medios audiovisuales, el espectro radioeléctrico, comunicaciones, desarrollo tecnológico, etc. Además, se prioriza la producción de tecnologías y avance en el sector tecnológico colombiano [18].

*Decreto 1412 de 2017:* Esta es una actualización al estatuto tributario del decreto 1078 de 2015, donde se definen nuevos términos relacionados al desarrollo de software, diseño 2d, diseño 3d, desarrollo de videojuegos, multimedia, etc. Además, se define la categorización de software y los servicios de educación virtual, para el desarrollo de contenidos digitales. También cabe resaltar que el ministerio de tecnologías de la información establecerá, si un curso virtual cumple con lo señalado en el decreto [19].

*Ley 1341 de 2009:* La presente ley determina el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, desde su ordenamiento, cobertura, calidad de servicio, competencia, inversión, etc. Buscando facilitar el libre acceso y sin discriminación a los habitantes del territorio nacional.

La ley promueve el desarrollo de tecnologías para así contribuir al crecimiento económico, competitividad, generación de empleo y las exportaciones. Se deben proteger los derechos de información de los ciudadanos, incentivando la seguridad y la inversión tecnológica en cuento a los desarrollos o cambios en la infraestructura tecnológica [17].

## **2.4 Marco Tecnológico y Científico**

De la investigación se conocen trabajos relacionados al desarrollo o implementación de sistemas para la gestión de servicios para diferentes empresas. Sin embargo, hay pocas investigaciones relacionadas a la implementación de sistemas de información Help desk para MiPymes y el impacto sobre este territorio.

En la investigación sobre el tema relacionado se encontró el trabajo de grado de Huerta Julca, L. J (2014) [16]. La cual toca el tema del desarrollo e implementación de un sistema de Help Desk para la atención de incidentes en el área de sistemas de una empresa en Perú, esto con el fin de optimizar los procesos y aumentar la productividad y así incrementar la atención eficiente para dar soporte técnico a incidentes de hardware o software a las diversas áreas de la organización. El proyecto se construye desde los antecedentes de la empresa, recolectando requerimientos para el software el cual lo desarrollan Open Source usando lenguajes de programación como PHP y MySQL; además se construyen encuestas para verificar la calidad del servicio de gestión de incidencias antes de ser implementado y para finalizar con la información recolectada se genera el modelo de negocio y la construcción del software con las necesidades encontradas [16].

El proceso de gestión de solicitudes de atención técnica es un proceso complejo y muchas veces por desconocimiento, dificultad o no contar con procesos definidos eso aporta a la insatisfacción del cliente o de miembros de una organización. Este proyecto plantea el uso de la metodología ITIL para el mejoramiento de los procesos de acuerdo a buenas prácticas de gestión de servicios, usando la metodología XP (Extreme programming) para el desarrollo del sistema, identificando requerimientos funcionales, no funcionales, diseño de interfaces, niveles de acceso e información manejada. Se identifica la aceptación del software mediante encuestas de

satisfacción y concluye con que la mejora de los procesos es significativa de acuerdo a los resultados obtenidos de dichas encuestas [20].

En la cruz roja ecuatoriana se implementó un servicio de Cloud con un proveedor llamado Spiceworks, el cual brinda soluciones informáticas con soporte técnico para la gestión de incidentes tecnológicos. Se identificó el problema de la alta demanda y se identificó que algunas herramientas superan su vida útil u operan de manera ineficiente en entornos los cuales son saturados por su alta demanda; buscar alternativas en la nube e implementando la mejora a partir de las practicas ITIL donde se estructura mejor el área de TI de una organización y sus procesos se optimizan [6].

Las MiPymes (micro, pequeñas y medianas empresa), han cambiado su propuesta de valor debido al impacto de la tecnología en el mundo empresarial. Según este trabajo de grado, las practicas ITIL se deben adaptar para cualquier sector empresarial donde la tecnología está presente para mejorar los procesos y la estructura organizacional. Se plantea una guía de cómo se implementa ITIL en MiPymes dependiendo de la planeación estratégica que se tenga y estas deben ir alineadas a los procesos de tecnología para mayor efectividad de este marco de trabajo. Esta guía de implementación se desarrolla en 5 fases: Entendimiento del cliente, gestión de portafolio de servicios, Políticas de prestación de servicios, Gestión Financiera y Gestión de demanda; estas 5 fases en el proyecto se explican y se implementan dando un mayor entendimiento del alcance de dicha metodología [11].

El proyecto de desarrollo de un sistema de información para la gestión universitaria desarrollado con el framework de Django para el backend y utilizando Bootstrap para el desarrollo del frontend, proveen la visión de un desarrollo ágil y la seguridad para el manejo de información.

Donde el backend se simplifica con funcionalidades como son el manejo de bases de datos, URL's, autenticación de usuarios y sistemas de plantillas [22].

### 3. Método

En este proyecto se usará una metodología de desarrollo ágil el cual brinda entregas rápidas el cual se llama programación extrema (Extreme Programming), el cual lleva al “extremo” las prácticas de desarrollo iterativo [21] (ver Figura 4).

Este enfoque ayuda en un desarrollo de sistemas de información de manera óptima y rápida, ya que el cliente solo se encuentra en la recolección de requerimientos y cuando se requiera evaluar el sistema para mejoras, lo que nos hace centrarnos en la programación y el funcionamiento del sistema.

#### 3.1 Criterios de Diseño

Para la construcción del sistema Help Desk se tuvieron en cuenta los siguientes criterios de diseño:

##### 3.1.1 Framework Backend

Para la selección del Framework Backend se escogió de acuerdo a varios elementos como los siguientes:

**Tabla 1.** *Comparación Framework Backend.*

| Framework | Lenguaje Base | Curva de aprendizaje | de Velocidad de desarrollo | de Seguridad | Licencia           |
|-----------|---------------|----------------------|----------------------------|--------------|--------------------|
| Django    | Python        | Baja                 | Alta                       | Alta         | Libre y Opensource |

| Framework   | Lenguaje Base | Curva de aprendizaje | de Velocidad de desarrollo | de Seguridad | Licencia           |
|-------------|---------------|----------------------|----------------------------|--------------|--------------------|
| Spring Boot | Java          | Alta                 | Media                      | Muy alta     | Apache 2.0         |
| Laravel     | PHP           | Media                | Alta                       | Alta         | Libre y Opensource |
| Express.js  | Javascript    | Baja                 | Alta                       | Media        | Libre y Opensource |

*Nota:* La tabla fue adaptada a partir de distintas fuentes de información citadas [21].

De acuerdo a la comparativa que se identifica en la tabla 1, para el desarrollo de este proyecto se escogió Django por su curva de aprendizaje baja, por el lenguaje que es muy práctico, además la seguridad que ofrece para los productos desarrollados es muy alta y su velocidad para desarrollar es alta y se puede construir cualquier sistema de manera rápida y eficiente.

### 3.1.2 Gestor de Base de datos

Para escoger el gestor de base de datos se tuvieron en cuenta las siguientes consideraciones:

**Tabla 2.** Comparación de gestor de bases de datos.

| Gestor     | Rendimiento                            | Escalabilidad                          | Licencia                                      |
|------------|--|--|---|
| PostgreSQL | Alto rendimiento en sistemas complejos | Muy alta (ideal para grandes sistemas) | Existe licencia libre tipo MIT                |
| SQLite     | Excelente para apps pequeñas           | Muy limitada                           | Libre   |
| MySQL      | Muy rápido en lecturas                 | Alta                                   | Existe de comunidad y/o propietaria de Oracle |

*Nota:* La tabla fue adaptada a partir de distintas fuentes de información citadas [9].

Para el desarrollo de este proyecto se usará PostgreSQL ya que es un gestor que ofrece una mayor escalabilidad, mejor rendimiento y cuenta con una licencia MIT de uso libre, esto con respecto a las 3 opciones encontradas se ajusta mejor al proyecto y es la que se usará para el desarrollo.

### 3.1.3 Historias de usuario

Para representar los requerimientos se diseñaron las siguientes historias de usuario:

**Figura 5.** Tarjetas de historias de usuario.

|                    |                |
|--------------------|----------------|
| <b>Código:</b>     |                |
| <b>Prioridad</b>   | <b>Estado:</b> |
| <b>Rol:</b>        |                |
| <b>Título:</b>     |                |
| <b>Descripción</b> |                |

*Nota:* La figura fue adaptada a partir de distintas fuentes de información citadas [21].

A continuación, se explican los campos de este modelo:

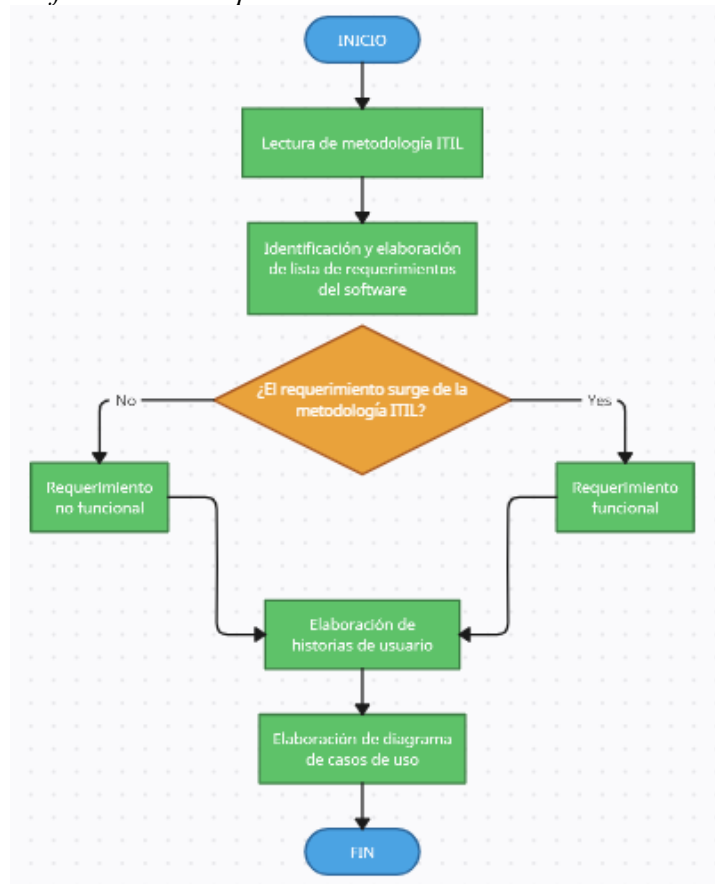
- **Código:** Este elemento sirve para identificar cada requisito. Asignándose de forma única y su formato es el siguiente “RF#” o “RNF#”, don RF indica si es un requerimiento funcional, RNF indica si es un requisito no funcional y # indica el número del requisito.
- **Prioridad:** Este elemento tiene que ver con el orden de implementación del requisito, de esta forma puede tomar los valores siguientes: Alto, Medio o Baja.

- **Estado:** Punto en el que se encuentra el requisito dentro de la aplicación puede ser “implementado” o “sin implementar”.
- **Rol:** Ya que este software está basado en roles, se identifica que rol está afectado con este requisito en caso de no tener un rol se indica con NA o no aplica.
- **Título:** Breve resumen del contenido del requisito.
- **Descripción:** Explicación detallada y clara del requisito.

### **3.2 Descripción de etapas y tareas**

#### **3.2.1 Recolección de requerimientos**

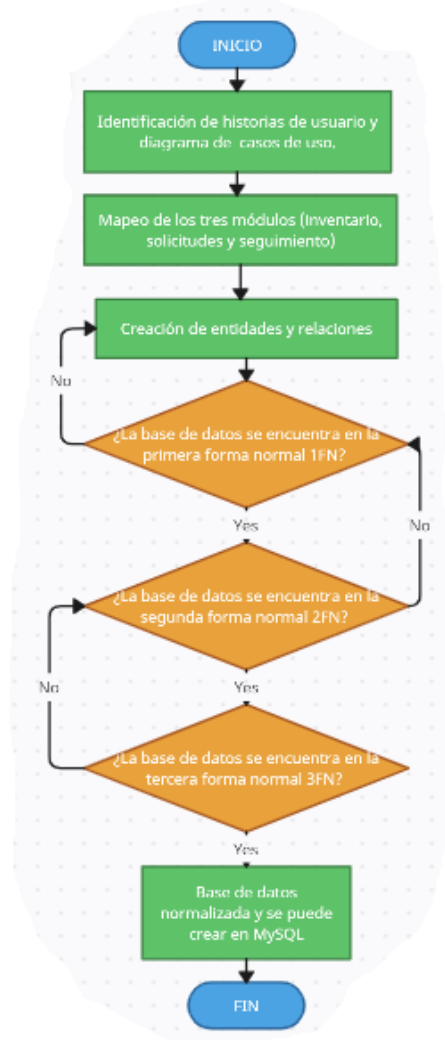
De acuerdo a la lectura de metodología ITIL y el desarrollo de los 3 módulos gestión de incidencias, inventario y solicitudes. Se obtiene un proceso para la identificación de requerimientos funcionales y no funcionales de acuerdo a la metodología ITIL y a las necesidades de los 3 módulos (ver Figura 6).

**Figura 6.** *Proceso de identificación de requerimientos.*

Adaptada según el primer objetivo específico de este trabajo.

### 3.2.2 Diseño de base de datos

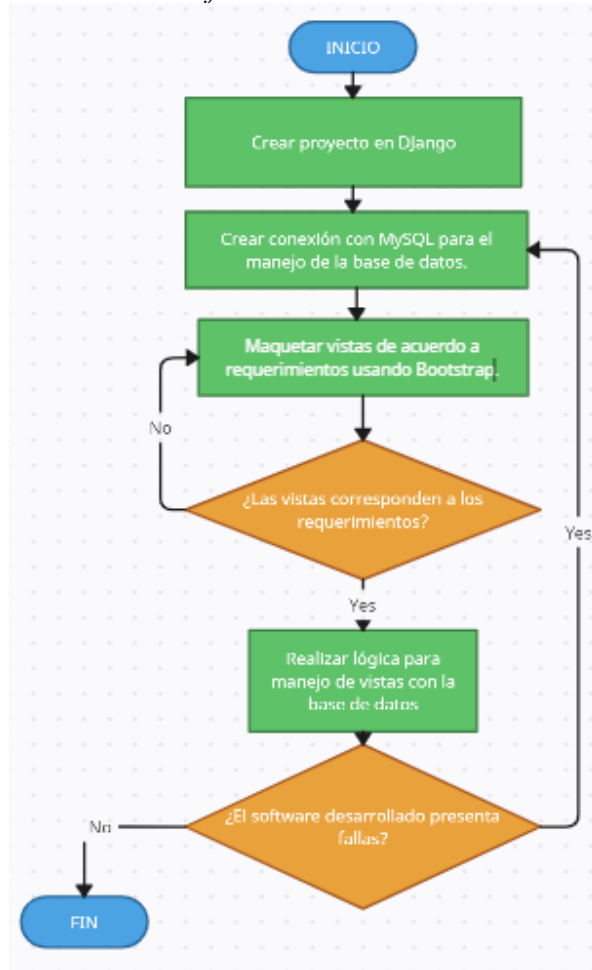
Luego de tener las historias de usuario definidas y los requerimientos funcionales y no funcionales mapeados, procederemos a diseñar la base de datos de acuerdo a reglas de normalización. Ya teniendo actores y funcionalidades definidos será más sencillo la creación con lenguaje SQL de las tablas y sus respectivas relaciones en dicha base de datos (ver Figura 7).

**Figura 7.** *Proceso para creación y normalización de base de datos.*

Adaptada según el segundo objetivo específico de este trabajo.

### 3.2.3 Desarrollo Frontend y Backend

Posteriormente, como parte final se desarrolla la maquetación del frontend con HTML, CSS, Javascript y Bootstrap, para que sea una interfaz amigable para el usuario, donde dependiendo de los requerimientos dados se realizan cambios de acuerdo al gusto del cliente. Y para el backend se realizan conexiones y la lógica de consultas y acceso a la información, donde se pueda crear tickets, hacer seguimientos, asignar equipos, etc (ver Figura 8).

**Figura 8.** *Procedimiento de desarrollo de software.*

Adaptada según el tercer objetivo específico de este trabajo.

### 3.3 Descripción de los productos finales

A lo largo del desarrollo de este proyecto, se han creado varios entregables o productos finales que dan solución al problema planteado. Estos representan una contribución al sector tecnológico de las MiPymes en Colombia.

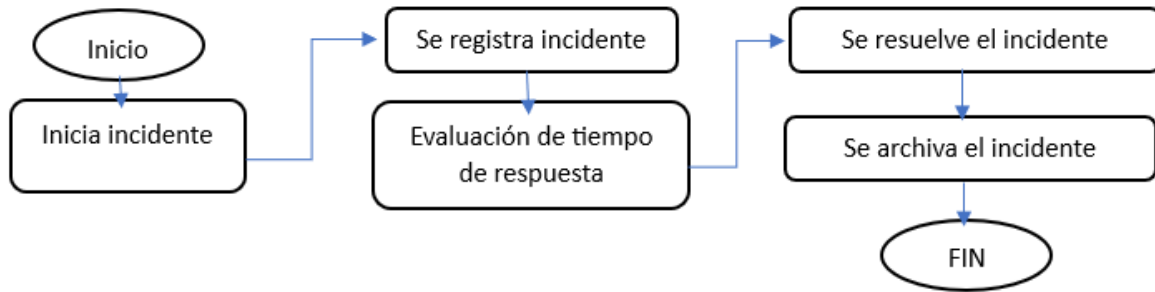
En la primera fase, se cuenta con dos productos finales dentro de los cuales son las historias de usuario y los casos de uso, estos representan los requerimientos funcionales y no funcionales tomados a partir del estudio de la metodología ITIL. Y estos son la base para la construcción del

sistema. En la siguiente fase se diseña la base de datos relacional de acuerdo a los requerimientos levantados usando el gestor de base de datos PostgreSQL, esto proporciona la idea de cómo debe estar estructurado el sistema en el flujo de información. Finalmente se desarrolla el sistema usando el Framework Django el cual gestiona vistas, urls y base de datos; además, usando HTML, CSS, Bootstrap y Javascript se diseñan las vistas, las cuales son las pantallas que el usuario final puede observar y utilizar. Estos productos finales se detallan en el capítulo 4, en el cual se incluyen imágenes y pruebas; esto representa el funcionamiento del sistema en entorno web. Lo cual tendrá un impacto en la gestión tecnológica en las MiPymes optimizando los procesos de las organizaciones.

#### **4. Resultados**

El proceso para la atención de un incidente se muestra en la figura 9 este da inicio cuando:

- El usuario solicita atención por parte del área técnica mediante la herramienta disponible, ya sea a través de documentos o un software.
- La solicitud se registra en la herramienta correspondiente y se notifica al área técnica.
- El técnico evalúa la solicitud, clasificándola según los Acuerdos de Nivel de Servicio (ANS) establecidos por la política de la organización, lo cual determina el tiempo de respuesta.
- El técnico procede a resolver el incidente, documentando la solución en la herramienta correspondiente, según aplique.
- Una vez solucionado el incidente, se archiva la solución, se notifica al usuario que el problema ha sido resuelto y se retoman las labores habituales.

**Figura 9.** *Flujo de resolución de incidentes.*

Adaptado de ITIL 4 [3]

En el presente trabajo se utilizó la metodología ágil de *programación extrema (XP)* para desarrollo de software, como ya se explicó en capítulos anteriores donde se identifican las etapas de planificación, diseño, desarrollo y pruebas, que serán ejecutadas en este trabajo de esa manera descrita.

Inicialmente se preparan las historias de usuario en la etapa de planificación, ya que estas contienen los requerimientos funcionales y no funcionales con los que nuestro sistema debe contar, considerando que este trabajo es la base para un sistema HelpDesk para cualquier tamaño de organización se recolectan los requerimientos de la lectura y análisis de la metodología ITIL, enfocándonos en tres módulos fundamentales como lo son: gestión de incidentes, solicitudes e inventario.

Cabe aclarar que este sistema se desarrolla con la intención clara de ser *escalable* y *adaptable* a cualquier organización MiPyme, buscando optimizar los procesos de gestión de la infraestructura tecnológica.

## 4.1 Diseño

### 4.1.1 Análisis metodología ITIL

Itil v4 divide su marco de trabajo en 3 pilares importantes para la gestión de servicios tecnológicos, estas se dividen en gestión general, gestión de servicios y gestión técnica (ver Figura 10). Para el software que vamos a desarrollar se analizaran las prácticas para la gestión del servicio ya que nuestra aplicación de help desk o mesa de ayuda está basada en estas prácticas.

**Figura 10.** *Prácticas de gestión tecnológica marco ITIL.*

| Prácticas Gestión General  | Prácticas Gestión de Servicios  | Prácticas Gestión Técnica   |
|--|---|---|
| <ul style="list-style-type: none"> <li>■ Gestión de la arquitectura</li> <li>■ Mejora continua</li> <li>■ Gestión de seguridad de la información</li> <li>■ Gestión del conocimiento</li> <li>■ Medición y reporte</li> <li>■ Gestión del cambio organizacional</li> <li>■ Gestión del portafolio</li> <li>■ Gestión de proyectos</li> <li>■ Gestión de relaciones</li> <li>■ Gestión de riesgos</li> <li>■ Gestión financiera del servicio</li> <li>■ Gestión de la estrategia</li> <li>■ Gestión de proveedores</li> <li>■ Gestión del personal y talento</li> </ul> | <ul style="list-style-type: none"> <li>■ Gestión de la disponibilidad</li> <li>■ Análisis del negocio</li> <li>■ Gestión de la capacidad y desempeño</li> <li>■ Control de cambios</li> <li>■ Gestión de incidentes</li> <li>■ Gestión de activos de TI</li> <li>■ Gestión de eventos y monitoreo</li> <li>■ Gestión de problemas</li> <li>■ Gestión de liberación</li> <li>■ Gestión del Catálogo de Servicios</li> <li>■ Gestión de configuración del servicio</li> <li>■ Gestión de continuidad del servicio</li> <li>■ Diseño de servicio</li> <li>■ Service Desk</li> <li>■ Gestión de niveles de servicio</li> <li>■ Gestión de solicitudes de servicio</li> <li>■ Validación y pruebas del servicio</li> </ul> | <ul style="list-style-type: none"> <li>■ Gestión de implementación</li> <li>■ Gestión de infraestructura y plataformas</li> <li>■ Desarrollo y gestión de software</li> </ul> |

Adaptado de ITIL 4 [3].

De estas prácticas de gestión de servicios los elementos fundamentales para el desarrollo de nuestro sistema son la gestión de incidentes, gestión de activos de TI, Gestión de eventos y monitoreo y Gestión de solicitudes de servicio; estos elementos son importantes para una MiPyme y mejorar la eficiencia operativa y garantizar la continuidad del negocio.

*Gestión de incidentes:* Un incidente se define como cualquier interrupción no planificada o reducción en la calidad de un servicio de TI, como puede ser el fallo de un dispositivo tecnológico o la caída de una aplicación crítica. El propósito principal de esta práctica es minimizar el impacto negativo de los incidentes en las operaciones del negocio mediante la restauración del servicio en el menor tiempo posible. Una gestión eficaz de incidentes contribuye significativamente a la continuidad operativa, mejora la experiencia del usuario y fortalece la percepción del servicio.

Para lograrlo, es fundamental implementar procesos sistemáticos de registro, categorización, priorización y resolución de incidentes. Asimismo, deben establecerse Acuerdos de Nivel de Servicio (ANS o SLA, por sus siglas en inglés) que definan los tiempos de respuesta y resolución esperados según el impacto y la urgencia del incidente, permitiendo una atención oportuna y alineada con los objetivos de la organización.

*Gestión de activos de TI:* La gestión de activos de TI es la práctica encargada de planificar y controlar el ciclo de vida completo de todos los activos tecnológicos utilizados en la prestación de servicios, con el objetivo de maximizar su valor para la organización, optimizar los costos, mitigar riesgos y apoyar la toma de decisiones relacionadas con adquisición, mantenimiento y reutilización. Un activo de TI se define como cualquier recurso valioso que contribuya a la entrega de productos o servicios tecnológicos, incluyendo hardware, software, redes, servicios en la nube y dispositivos periféricos.

Para una gestión eficiente, se recomienda implementar políticas que aseguren la trazabilidad y el control de cada activo. Entre las directrices clave se encuentran:

- *Identificación y etiquetado:* Cada activo físico (hardware) debe estar identificado mediante etiquetas con información relevante, lo que permite su seguimiento y protección frente a robos, daños o pérdidas.
- *Control de licencias de software:* Todo software debe estar debidamente registrado y monitoreado para evitar el uso indebido de licencias. Además, deben conservarse los comprobantes de adquisición como respaldo legal y administrativo.
- *Asignación de responsables:* Los activos deben estar asignados a un usuario responsable, quien debe comprometerse formalmente con su cuidado y conservación al momento de su entrega.

Asimismo, se debe realizar un seguimiento continuo que contemple los siguientes aspectos:

- Definir y mantener actualizado el inventario de activos, estructurado según criterios claros y normalizados.
- Controlar el ciclo de vida de los activos, desde su adquisición hasta su retiro, para prevenir la obsolescencia tecnológica y mantener la eficiencia operativa.
- Generar informes periódicos con información actualizada sobre el estado, ubicación y uso de los activos.
- Implementar mecanismos de monitoreo para detectar a tiempo posibles fallos, deterioros o pérdidas, y permitir una intervención oportuna.

Una adecuada gestión de activos de TI no solo asegura el uso eficiente de los recursos tecnológicos, sino que también fortalece la seguridad, la trazabilidad y el cumplimiento normativo dentro de la organización.

*Gestión de eventos y monitoreo:* Esta práctica tiene como objetivo principal detectar, analizar y responder de forma oportuna a eventos que puedan afectar la continuidad y el desempeño de los servicios tecnológicos, con el fin de prevenir, minimizar o eliminar impactos negativos sobre la organización. Un evento se refiere a cualquier cambio de estado que tenga relevancia para la gestión de un servicio o activo de TI, y su monitoreo permite anticipar incidentes antes de que afecten al negocio.

Para implementar esta práctica de manera efectiva, se deben considerar las siguientes directrices:

- *Implementación de herramientas de monitoreo:* Se deben utilizar soluciones tecnológicas que permitan el seguimiento continuo de los sistemas, redes, aplicaciones y dispositivos,

facilitando la identificación temprana de anomalías, fallas o patrones inusuales en los activos de TI.

- *Definición de políticas de gestión de eventos:* Es fundamental establecer lineamientos claros sobre cómo deben categorizarse, priorizarse y gestionarse los eventos detectados, diferenciando entre eventos informativos, advertencias y excepciones. Estas políticas deben garantizar una respuesta coherente y eficaz ante cada tipo de evento.

Una adecuada gestión de eventos y monitoreo no solo mejora la disponibilidad y el rendimiento de los servicios, sino que también fortalece la capacidad de reacción ante incidentes, optimiza el uso de los recursos y proporciona visibilidad en tiempo real del estado operativo de la infraestructura tecnológica.

*Gestión del nivel del servicio:* Esta práctica tiene como objetivo definir, documentar, supervisar y gestionar acuerdos de nivel de servicio (ANS o SLA, por sus siglas en inglés) que establezcan compromisos medibles respecto a la calidad y disponibilidad de los servicios de TI. Los ANS deben incluir métricas claras y alcanzables que reflejen el impacto que la indisponibilidad tecnológica puede generar en la operación de la organización, permitiendo así evaluar el rendimiento del servicio en relación con las expectativas del cliente o usuario.

Estas métricas deben enfocarse especialmente en aspectos como el tiempo de respuesta y resolución de incidentes, tiempos de disponibilidad del servicio, frecuencia de fallos, entre otros indicadores clave. Además, deben estar alineadas con los objetivos del negocio y ser comprensibles tanto para los proveedores del servicio como para los usuarios finales.

La correcta aplicación de esta práctica permite medir y mejorar continuamente la calidad del servicio ofrecido, establecer expectativas realistas con los usuarios, identificar brechas en la

entrega del servicio y fomentar una cultura de mejora continua basada en datos objetivos y verificables.

*Gestión de solicitudes de servicio:* Esta práctica tiene como objetivo garantizar la atención oportuna, ordenada y eficiente de las solicitudes realizadas por los usuarios, las cuales no necesariamente están relacionadas con incidentes, sino con requerimientos estándar como accesos, instalaciones, configuraciones o información. Su propósito es respaldar el cumplimiento de los acuerdos de nivel de servicio (ANS), promoviendo una experiencia positiva del usuario y una operación continua del negocio.

Cada solicitud debe ser registrada con información mínima estandarizada, incluyendo un título claro, una descripción detallada del requerimiento y la prioridad o nivel de importancia asociado a su resolución. Esta información permite su clasificación, asignación y seguimiento adecuado dentro del sistema de gestión.

Asimismo, las organizaciones deben establecer políticas y flujos definidos para cada tipo de solicitud recurrente, tales como: solicitud de equipos, acceso a recursos restringidos, recuperación de información, generación de documentos o cualquier otro servicio definido previamente. Estas políticas deben estar alineadas con los controles de seguridad, niveles de autorización y procedimientos internos, facilitando una atención eficiente y trazable en el marco de una gestión estructurada.

Una correcta gestión de solicitudes contribuye a la transparencia del servicio, mejora los tiempos de respuesta y asegura la alineación entre las necesidades del usuario y las capacidades del área de TI.

### 4.1.2 Identificación de roles y perfiles

Para el desarrollo de nuestro sistema se definen tres roles bases los cuales se pueden escalar para cualquier organización en los cuales se identifica el flujo de trabajo en la Figura 11.

**Figura 11.** Roles y funciones del sistema.



Adaptado de ITIL 4 [3].

*Administrador:* Responsable de la gestión de usuarios, incluyendo su creación y desactivación.

*Técnico:* Encargado de la gestión de incidentes, desde su inicio hasta su resolución. Además, se ocupa de la asignación de equipos a los usuarios y de la administración de la base de conocimientos generada a partir de los incidentes registrados.

*Empleado:* Responsable de reportar incidentes cuando estos se presenten.

### 4.1.3 Identificación de requerimientos

Teniendo en cuenta el análisis de la metodología ITIL se obtuvo la siguiente lista de requerimientos funcionales y no funcionales.

**Tabla 3.** Requerimientos funcionales del sistema.

| Número | Descripción  | Prioridad |
|--------|--|-----------|
| RF1    | El sistema debe permitir la autenticación de los usuarios mediante un nombre de usuario y contraseña. Cada usuario debe tener un rol asignado que determine su nivel de acceso dentro del sistema. | Alta      |

| Número | Descripción   | Prioridad |
|--------|---|-----------|
| RF2    | El usuario con rol de administrador debe poder crear, actualizar y desactivar usuarios dentro del sistema. Los usuarios desactivados no pueden eliminarse, sino quedar inactivos en el sistema para generar trazabilidad. | Alta      |
| RF3    | El sistema debe permitir la restauración de contraseñas para todos los usuarios registrados.  | Media     |
| RF4    | Todos los usuarios con rol empleado deben poder generar un ticket en el sistema, donde incluya descripción del incidente, activo afectado si es el propio o es algo inherente al equipo, prioridad y título.              | Alta      |
| RF5    | Los empleados deben poder visualizar el estado actual y el avance de sus solicitudes registradas en el sistema.   | Alta      |
| RF6    | El administrador debe tener la capacidad de registrar y actualizar la información de los equipos en el sistema. Los equipos no pueden eliminarse; solo pueden ser desactivados, lo que los marcará como inactivos.        | Alta      |
| RF7    | El administrador podrá agregar tipos de activos diferentes.   | Baja      |
| RF8    | El sistema debe permitir a los usuarios calificar a los técnicos que han atendido sus solicitudes.  | Baja      |
| RF9    | El administrador del sistema debe permitir el registro, actualización y eliminación de las prioridades asignadas a los tickets.   | Alta      |
| RF10   | El sistema debe mostrar claramente quién resolvió la solicitud del ticket.  | Media     |
| RF11   | El sistema debe generar una trazabilidad de los tickets, incluyendo un hilo de conversaciones entre el usuario y el técnico en caso de no dar solución inmediata.   | Baja      |
| RF12   | El sistema debe poder generar una pestaña de administrador y técnico donde se pueda evidenciar la hoja de vida del equipo y si ha tenido algún incidente.   | Media     |
| RF13   | El sistema debe enviar un correo al empleado cuando genere un ticket y cuando este se resuelva con la información correspondiente.  | Media     |
| RF14   | El sistema debe restringir el acceso de los usuarios a ciertos módulos dependiendo de su rol.   | Media     |
| RF15   | El sistema debe generar gráficos en la pantalla de inicio para administrador y técnico donde se vean los incidentes resueltos y pendientes.   | Bajo      |
| RF16   | El rol administrador debe poder gestionar las diferentes áreas de la organización y esta ubicación también se les asigna a los equipos y al personal.   | Media     |
| RF17   | El sistema debe poder generar un mantenimiento a un equipo de parte del área técnica o administrativa.  | Media     |

*Nota:* esta tabla fue elaborada teniendo en cuenta como referencia la metodología ITIL [3].

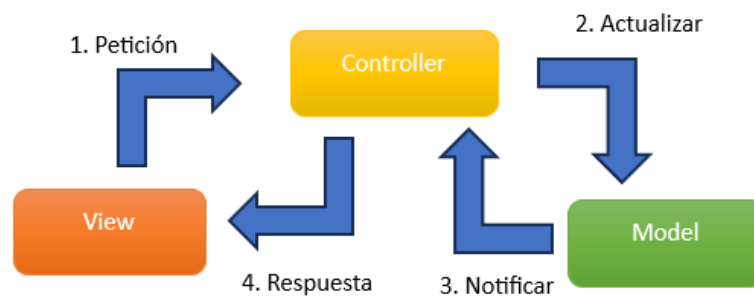
**Tabla 4.** *Requerimientos no funcionales del sistema.*

| Número | Descripción  |
|--------|--|
| RNF1   | El sistema debe contar con Responsive design para el manejo en otros dispositivos. |
| RNF2   | El sistema debe soportar y correr la aplicación sin ningún error.                  |

*Nota:* esta tabla fue elaborada teniendo en cuenta como referencia las metodologías de desarrollo de software [21].

#### 4.1.4 Arquitectura de proyecto

El patrón de arquitectura en la cual está construido el sistema de este proyecto, se basa en MTV (model, template, View) que a su vez está basada en el MVC (model, view, controller), que su funcionamiento principal se basa en separar los datos y la lógica de la aplicación.

**Figura 12.** *Flujo de información patrón de arquitectura MVC.*

Adaptada de Ingeniería de software [21].

El patrón de arquitectura MVC se compone de tres elementos fundamentales (ver Figura 12) como lo son modelo, vista y controlador; esto da la representación de la información y busca la reutilización de código y separación de conceptos facilitando el desarrollo y el mantenimiento de los sistemas. Estos tres componentes se pueden definir de la siguiente manera:

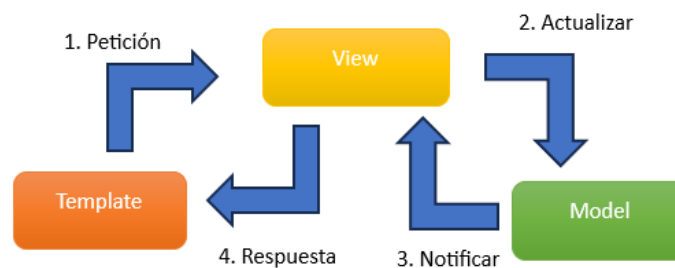
*Modelo:* De manera sencilla se explica como el acceso a la información del sistema, el cual permite acceder a la base de datos y manipularla dentro del sistema.

*Controlador:* Este elemento responde a los eventos del usuario por medio de la vista y realiza peticiones al modelo o base de datos para manipular la información y por último presentarla al usuario.

*Vista:* Este elemento presenta la información del modelo al usuario de manera clara y concisa con la cual poder interactuar.

En Django se usa la arquitectura MTV (Modelo, vista, template) que es similar al MVC con ciertas diferencias (ver Figura 13).

**Figura 13.** Flujo de información patrón de arquitectura MTV de Django.



Adaptado de Django Documentation [7].

La analogía del MVC con MTV es la siguiente:

**Modelo:** Es lo mismo que el modelo en MVC, encargado de la gestión de la información y la comunicación con la base de datos.

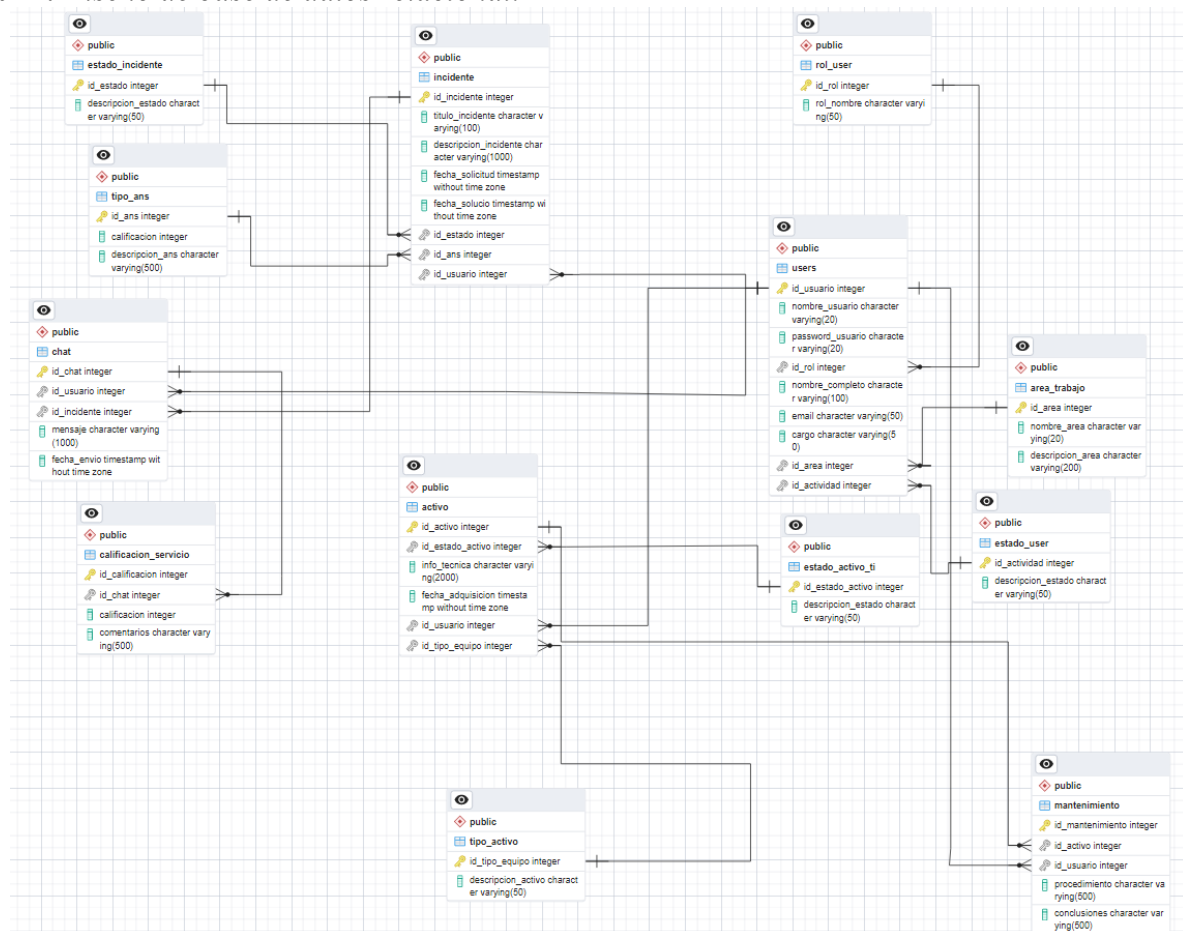
**Vista o View:** Este representa la función del controlador en MVC, encargado de la gestión de peticiones al usuario y manipular la información.

**Template o Plantilla:** Este representa la función de vista en MVC, encargado de presentar la información de manera amigable al usuario.

### 4.1.5 Diseño base de datos

Para el diseño de la base de datos se usó PostgreSQL basándonos en los requerimientos funcionales y cumpliendo las reglas de normalización de las bases de datos relacionales, asegurando la integridad de la información.

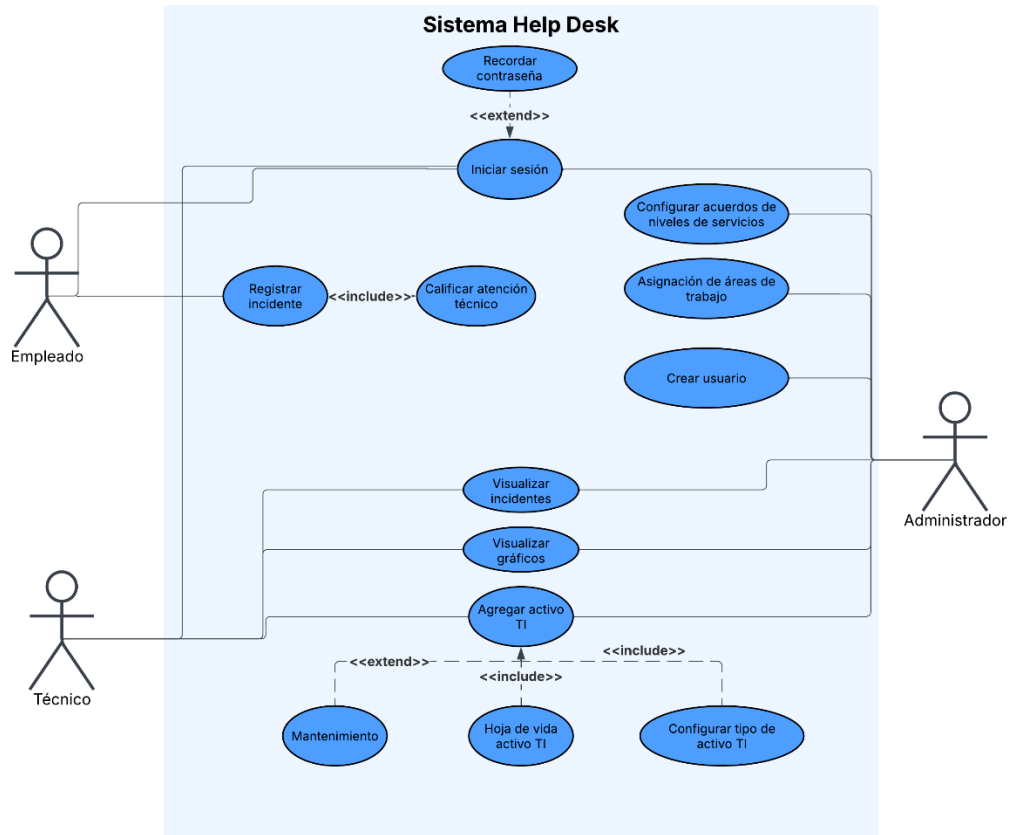
**Figura 14.** *Diseño de base de datos relacional.*



Adaptado de acuerdo a la tabla 3 y tabla 4.

### 4.1.6 Casos de uso del sistema

De acuerdo a los requerimientos previamente definidos se definieron los siguientes casos de uso del sistema.

**Figura 15.** Casos de uso del sistema.

Adaptado de acuerdo a la tabla 3 y tabla 4.

#### 4.1.7 Configuración de proyecto con Django

Django ofrece la facilidad de crear aplicaciones web de manera rápida y eficiente (para su uso se debe tener Python instalado en tu sistema operativo). Para el desarrollo de cualquier software usando Python lo recomendable es usar entornos virtuales para aislar los paquetes o dependencias de un proyecto y mantener un sistema limpio y usando prácticas de desarrollo profesional. Esto se realiza con el siguiente comando (ver Figura 16).

**Figura 16.** Creación de entorno virtual con Python.

```
PS F:\Sistema Help Desk> python -m venv venv
```

Esto crea una carpeta con una copia de la versión de Python instalada y de pip (sistema de gestión de paquetes de Python), así ahora puedes instalar diferentes versiones de dependencias o paquetes dentro de este entorno sin ver afectado a nivel global tu equipo.

Terminando la explicación anterior, se procede a instalar Django usando el sistema de gestión de paquetes de Python con el siguiente comando (ver Figura 17).

**Figura 17.** *Instalación de Django.*

```
PS F:\HelpDesk> pip install django
```

Luego de tener django instalado se puede comenzar a desarrollar el proyecto. Para iniciar el proyecto se utiliza el siguiente comando (ver Figura 18).

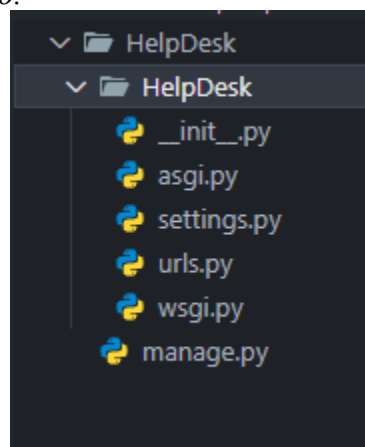
**Figura 18.** *Creación de proyecto con Django.*

```
PS F:\HelpDesk> django-admin startproject HelpDesk
```

En dicho comando se puede cambiar el nombre “*HelpDesk*” por el nombre que se desee ya que va a ser el núcleo principal de configuración del sistema.

Al crear el proyecto, este crea automáticamente la siguiente carpeta con los siguientes archivos (ver Figura 19).

**Figura 19.** *Archivos al crear proyecto.*



Dentro de los cuales los más importantes son los siguientes y sus funcionalidades:

*Urls.py*: encargado de gestionar las urls y acceso a todas las funcionalidades del sistema.

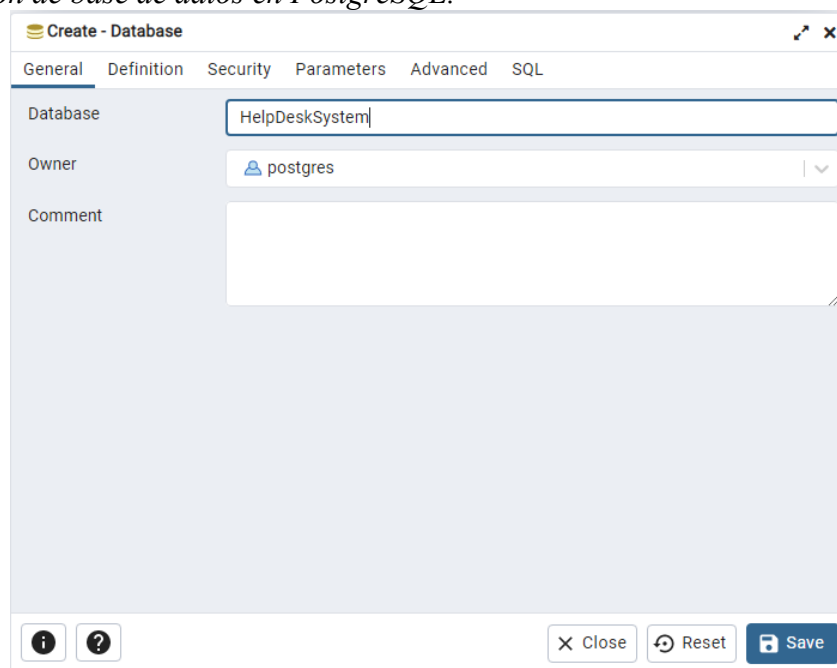
*Settings.py*: Encargado de la configuración de todo el proyecto como envío de emails, conexión con base de datos e incluir aplicaciones con funcionalidades diversas.

*Manage.py*: encargada de la ejecución del sistema.

Para poder iniciar con el desarrollo, se deben configurar temas como que motor de base de datos se usara (PostgreSQL), también se debe definir si se usara algún método para envío de correos electrónicos, agregar aplicaciones del proyecto y por último el uso de los templates.

**4.1.7.1 Configuración servidor.** Para configurar nuestro servidor usaremos PostgreSQL por la facilidad de uso, sin embargo, la configuración es similar para casi todas las opciones mencionadas anteriormente. Primero se debe crear la base de datos en nuestro motor de base de datos (ver Figura 20).

**Figura 20.** Creación de base de datos en PostgreSQL.



The image shows a screenshot of the PostgreSQL 'Create - Database' dialog box. The dialog has a title bar with a yellow icon and the text 'Create - Database'. Below the title bar are tabs for 'General', 'Definition', 'Security', 'Parameters', 'Advanced', and 'SQL'. The 'General' tab is selected. The 'Database' field contains the text 'HelpDeskSystem'. The 'Owner' field is a dropdown menu showing 'postgres'. The 'Comment' field is a large empty text area. At the bottom of the dialog, there are three buttons: 'Close', 'Reset', and 'Save'.

Luego de creada esta base de datos su configuración en Django consta de lo siguiente (Ver Figura 21).

**Figura 21.** Configuración de base de datos en Django.

```
DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql",
        "NAME": "HelpDeskSystem",
        "USER": "postgres",
        "PASSWORD": "1234",
        "HOST": "localhost",
        "PORT": "5432",
    }
}
```

*ENGINE*: De acuerdo a la documentación de django esto es el texto que se debe agregar para el uso de PostgreSQL.

*NAME*: El nombre de la base de datos que debe estar creada en PostgreSQL.

*USER*: Usuario que se tiene en PostgreSQL

*PASSWORD*: La contraseña para poder tener acceso a esta base de datos.

*HOST*: Donde se encuentra ubicada la base de datos en este caso es de manera local, en el caso de que estuviera en la nube se debe colocar la dirección donde se encuentra alojada.

*PORT*: El puerto funcional de dicha base de datos, esto está por defecto con el uso de PostgreSQL.

**4.1.7.2 Configuración de correo.** En nuestro caso debemos configurar el envío de correos para notificar incidentes a los usuarios dentro del sistema, incluso para recuperar la contraseña si esta se llega a olvidar. Dentro de los protocolos el más usado es el SMTP el cual es responsable de enviar correos electrónicos entre cuentas, existen otros protocolos los cuales no tocaremos en este trabajo.

Django para la configuración de envío de correos por el protocolo SMTP el cual utilizaremos el servicio de correo electrónico Gmail, requiere la siguiente información (ver Figura 22).

**Figura 22.** Configuración de correo en Django.

```
EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"
EMAIL_HOST = "smtp.gmail.com"
EMAIL_PORT = 587
EMAIL_USE_TLS = True

EMAIL_HOST_USER = "systemhelpdesk1234@gmail.com"
EMAIL_HOST_PASSWORD = "hyyh1qkmpwnanosx"
```

*EMAIL\_BACKEND*: En la documentación de Django recomienda poner esta información para el uso del protocolo SMTP.

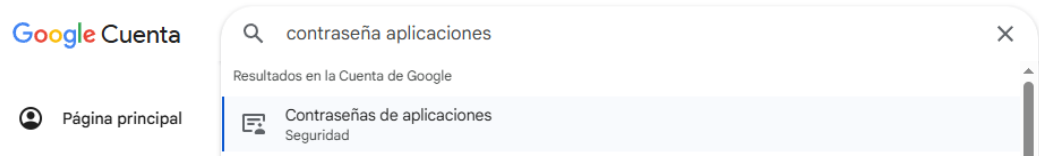
*EMAIL\_HOST*: Es el servidor o proveedor de servicios de correos electrónico que usaremos, que en este caso en Gmail.

*EMAIL\_PORT*: Este puerto está definido para el protocolo y se puede usar el puerto 25 o en este caso el 587.

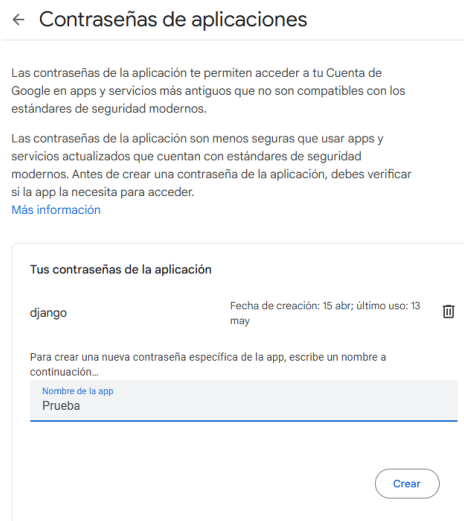
*EMAIL\_USE\_TLS*: El TLS es un protocolo el cual ofrece seguridad en el transporte de información, esto se puede definir como True o False, en el caso que sea necesario.

*EMAIL\_HOST\_USER*: Correo electrónico usado para el envío de correos electrónicos este correo solo será para envío de correos no se podrá responder a dicho correo.

*EMAIL\_HOST\_PASSWORD*: Esta contraseña se agrega de la siguiente manera; Se debe ingresar a la administración de tu cuenta Gmail y en el buscador escribir “*contraseña de aplicaciones*”.

**Figura 23.** Configuración contraseña de aplicaciones Google.

Luego te permitirá crear una contraseña para que aplicaciones externas puedan tener acceso al correo y se puedan enviar correos de manera automática (ver Figura 24).

**Figura 24.** Creación de contraseña de aplicaciones.

Al crear la contraseña se mostrará algo similar a la Figura 25, esta contraseña se debe copiar y pegar sin espacios dentro del espacio designado en `EMAIL_HOST_PASSWORD`.

**Figura 25.** Contraseña de aplicación de Google.

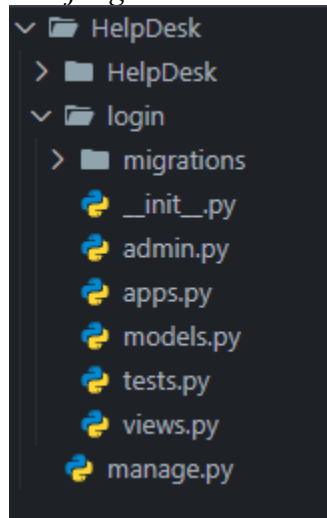
**4.1.7.2 Creación de aplicaciones.** Para la creación de aplicaciones el cual es el uso principal de Django en el cual se desarrolla por módulos cada funcionalidad del sistema, la creación de aplicaciones se hace de la siguiente manera (ver Figura 26).

**Figura 26.** Creación de aplicaciones en Django.

```
PS F:\HelpDesk> py manage.py startapp login
```

Este comando logra crear una carpeta en la cual se identifica lo siguiente (ver Figura 27).

**Figura 27.** Archivos de una aplicación en Django.



*Admin.py:* En este se registran los modelos o las tablas que se vayan a utilizar o crear en este módulo o aplicación.

*Models.py:* Creación de tablas y la base de datos en general, usando el ORM que maneja Django.

*Views.py:* Acá se encuentra la funcionalidad total de la aplicación, es decir en esta sección se manejan las consultas con las bases de datos y el redireccionamiento a los templates o vistas para el usuario.

Además, por buenas prácticas es recomendable agregar un archivo `urls.py` en cada aplicación para manejar las url de cada aplicación por separado y no generar conflictos.

Por último, para hacer uso de tu aplicación la debes registrar en el archivo `settings.py` en la sección `INSTALLED_APPS`, sino esta no tendrá cambio alguno en el proyecto (ver Figura 28).

**Figura 28.** Sección para agregar aplicaciones en Django.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'login',  
    'dashboard',  
    'usuarios',  
    'activosti',  
    'incidentes',  
]
```

Django ofrece la ventaja de usar archivos HTML para la creación de las interfaces del usuario, en el entorno de Django se llaman templates, en el cual se visibilizan todas las interfaces a las cuales el usuario podrá tener acceso, usando HTML, CSS y Javascript. Esto se debe configurar dentro del archivo `settings.py` en la sección `TEMPLATES` y en el elemento `DIRS` se debe colocar `BASE_DIR/templates`, esto con el fin de que en cada aplicación se ubiquen los archivos para las interfaces de manera eficiente (ver Figura 29).

**Figura 29.** Configuración de templates en Django.

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR/"templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

Esto concluye con la configuración total del proyecto para el desarrollo del sistema de información Help Desk. En los siguientes capítulos nos enfocaremos en la validación de requerimientos y análisis de resultados.

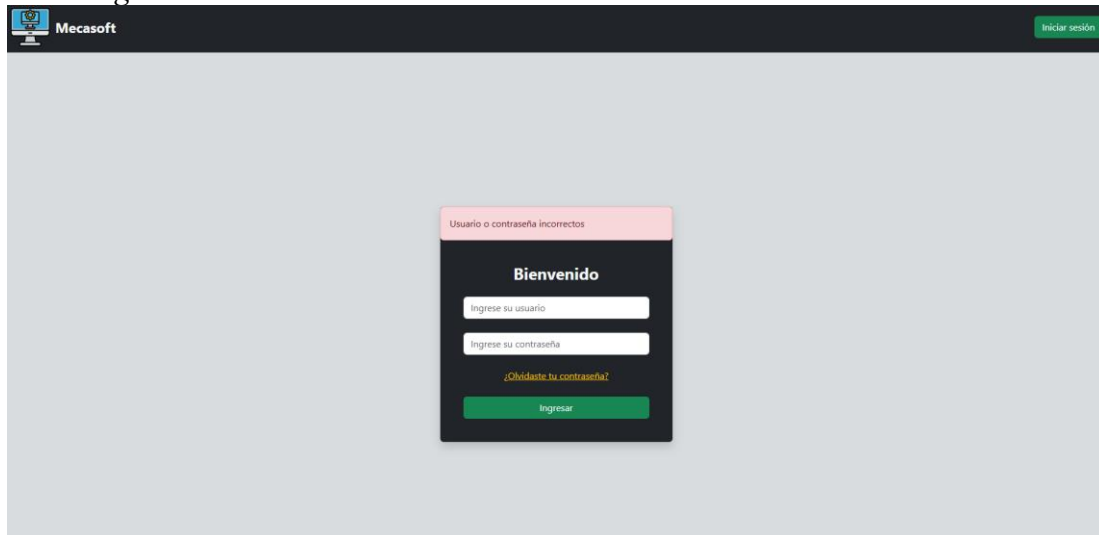
## 4.2 Validación de requerimientos

Para este capítulo se desarrolló el sistema siguiendo los requerimientos definidos previamente; y a para validar estos requerimientos se va a representar cada requerimiento con su respectiva implementación en el sistema desarrollado.

**Figura 30.** Requerimiento funcional 1.

|   |   |
|---|---|
| <b>Código:</b> RF01                           |   |
| <b>Prioridad:</b> Alta                        | <b>Estado:</b> Implementado   |
| <b>Rol:</b> Empleado, Técnico y Administrador |   |
| <b>Título:</b> Acceso a la aplicación         |   |
| <b>Descripción</b>                            | Para acceder a la aplicación es necesario tener un usuario registrado dentro del sistema mediante nombre de usuario y contraseña. El sistema debe notificar en la autenticación de los usuarios: <ul style="list-style-type: none"> <li>• Si los datos son incorrectos, se mostrará un mensaje de error.</li> </ul> |

Adaptado de figura 5.

**Figura 31.** *Login.*

Adaptado de HelpDeskSystem [12].

**Figura 32.** *Requerimiento funcional 2.*

|                           |   |
|---------------------------|---|
| <b>Código:</b> RF02       |   |
| <b>Prioridad:</b> Alta    | <b>Estado:</b> Implementado   |
| <b>Rol:</b> Administrador |   |
| <b>Título:</b> Usuarios   |   |
| <b>Descripción</b>        | <p>Los usuarios del sistema deben estar registrados en la base de datos con los siguientes campos:</p> <ul style="list-style-type: none"> <li>• Id de usuario</li> <li>• Nombre de usuario</li> <li>• Contraseña</li> <li>• Rol</li> <li>• Nombre completo</li> <li>• Email</li> <li>• Cargo</li> <li>• Área de trabajo</li> <li>• Estado</li> </ul> <p>Los usuarios se deben poder crear, consultar y actualizar; no se pueden eliminar ya que se deben poder manejar en el estado activo o inactivo para temas de trazabilidad.</p> |

Adaptado de figura 5.

Este componente posee varios elementos para dar cumplimiento los cuales se pueden identificar en la figura 33 y la figura 34.

**Figura 33.** *Formulario de creación de usuarios.*

**Formulario para crear un usuario**

Correo electrónico:       Contraseña:

Nombre:       Apellido:

Nombre de usuario:       Cargo:

Rol:       Área de trabajo:

Adaptado de HelpDeskSystem [12].

**Figura 34.** *Listado de usuarios registrados con edición y estado de activación.*

Lista de usuarios registrados en el sistema

| Nombre de usuario | Nombre completo          | Email                  | Cargo                 | Área de trabajo | Rol de usuario | Editar | Estado |
|-------------------|--------------------------|------------------------|-----------------------|-----------------|----------------|--------|--------|
| admin             | Deiby Rodríguez          | deiby_2304@gmail.com   | Gerente               |                 | administrador  |        |        |
| tecnico1          | Laura Camila Oñate Perez | onate@gmail.com        | Ingeniera mecatronica | Sistemas        | tecnico        |        |        |
| empleado1         | Olga Lucía Guerrero      | olgap1972@gmail.com    | Lider de desarrollo   | Sistemas        | empleado       |        |        |
| empleado2         | Andrey Guerrero          | deiby_2304@hotmail.com | Diseñador             | Sistemas        | empleado       |        |        |
| empleado3         | Hernan Rodriguez         | prueba@gmail.com       | Carpintero            | Contabilidad    | empleado       |        |        |

Adaptado de HelpDeskSystem [12].

**Figura 35.** *Requerimiento funcional 3.*

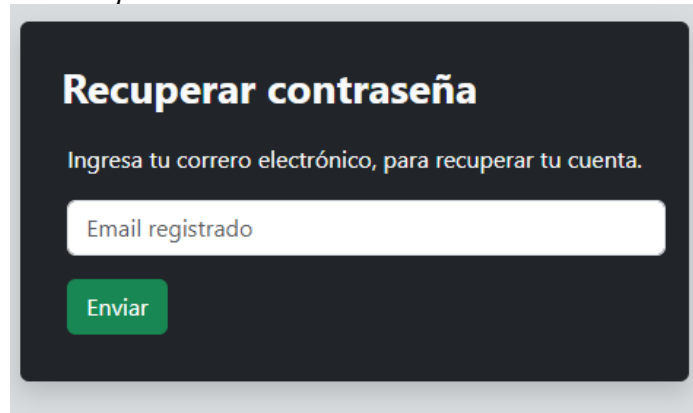
|  |  |
|--|--|
| <b>Código: RF03</b>                          |  |
| <b>Prioridad: Media</b>                      | <b>Estado: Implementado</b>  |
| <b>Rol:</b> Administrador, Usuario y Técnico |  |
| <b>Título:</b> Recuperación de contraseña    |  |
| <b>Descripción</b>                           | Se debe enviar un correo al usuario que lo solicite en el cual se pueda actualizar la contraseña y posteriormente enviarme a la pantalla de login para ingresar. |

Adaptado de figura 5.

El usuario puede acceder a través del login, en un link que dice “¿Olvidaste tu contraseña?” (ver Figura 31). En la cual lo redirige a un formulario que solicita el correo del

usuario registrado (ver Figura 36), posteriormente a el correo se envía un link al cual acceder y cambiar la contraseña (ver Figura 37).

**Figura 36.** *Formulario para recuperar contraseña.*



Recuperar contraseña

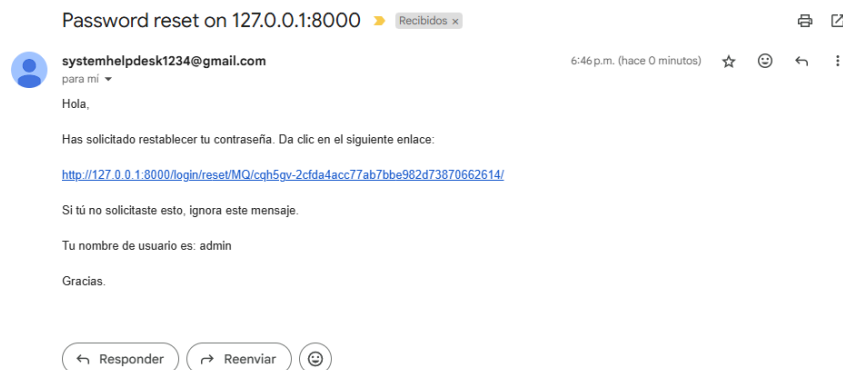
Ingresa tu correo electrónico, para recuperar tu cuenta.

Email registrado

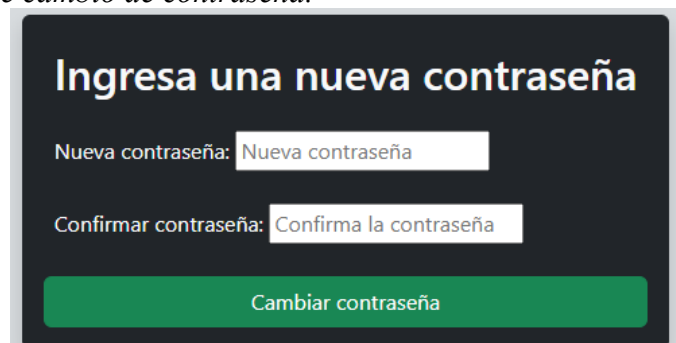
Enviar

Adaptado de HelpDeskSystem [12].

**Figura 37.** *Correo enviado para recuperar la contraseña.*



**Figura 38.** *Formulario de cambio de contraseña.*



Ingresa una nueva contraseña

Nueva contraseña: Nueva contraseña

Confirmar contraseña: Confirma la contraseña

Cambiar contraseña

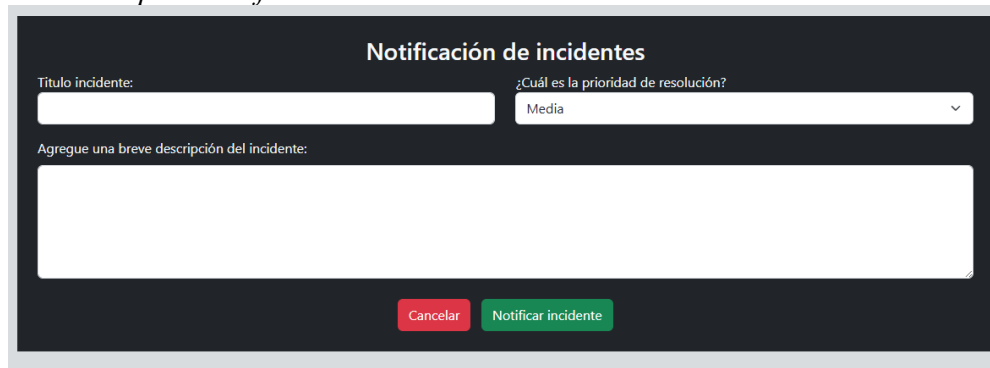
Adaptado de HelpDeskSystem [12].

**Figura 39.** *Requerimiento funcional 4.*

|                                      |  |
|--------------------------------------|--|
| <b>Código:</b> RF04                  |  |
| <b>Prioridad:</b> Alta               | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Usuario                  |  |
| <b>Título:</b> Generación de tickets |  |
| <b>Descripción</b>                   | <p>El sistema debe poder generar un ticket de incidencia con la siguiente información:</p> <ul style="list-style-type: none"> <li>• Título del incidente.</li> <li>• Descripción del incidente.</li> <li>• Selección de prioridad de atención del incidente de acuerdo a los acuerdos de niveles de servicio (ANS).</li> </ul> <p>Esto se debe enviar y se debe mostrar en pantalla el número del ticket generado. Además, en la base de datos se debe agregar esta información:</p> <ul style="list-style-type: none"> <li>• Fecha de solicitud del incidente</li> <li>• Fecha de solución de la solicitud</li> <li>• Estado del incidente</li> </ul> <p>Esto con el fin de tener una trazabilidad del incidente.</p> |

Adaptado de figura 5.

En la figura 40, se identifica la información que registra el usuario para la generación del ticket, en la capa del modelo internamente se genera la demás información y esta va cambiando de acuerdo a lo que el usuario técnico decida (ver Figura 41).

**Figura 40.** *Formulario para notificar incidentes.*


**Notificación de incidentes**

Título incidente:

¿Cuál es la prioridad de resolución?

Agregue una breve descripción del incidente:

Adaptado de HelpDeskSystem [12].

**Figura 41.** Clase del incidente.

```
class incidente(models.Model):
    titulo_incidente=models.CharField(max_length=100)
    descripcion_incidente=models.CharField(max_length=4000)
    fecha_solicitud=models.DateTimeField(auto_now_add=True)
    fecha_solucion=models.DateField(null=True)
    id_estado=models.ForeignKey(estadoIncidente,on_delete=models.SET_NULL,null=True)
    id_ans=models.ForeignKey(ans,on_delete=models.SET_NULL,null=True)
    usuario_remitente=models.ForeignKey(User,on_delete=models.SET_NULL,null=True)
```

Adaptado de HelpDeskSystem [12].

En la Figura 41, se puede identificar que la fecha de solicitud del incidente contiene el elemento `auto_now_add=True`, esto permite que cada que se agregue un elemento a la tabla se registre la hora de la zona horaria donde la base de datos se encuentre ubicada.

**Figura 42.** Requerimiento Funcional 5.

|                                       |  |
|---------------------------------------|--|
| <b>Código:</b> RF05                   |  |
| <b>Prioridad:</b> Alta                | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Usuario                   |  |
| <b>Título:</b> Seguimiento de tickets |  |
| <b>Descripción</b>                    | <p>El sistema debe tener una pestaña para poder verificar el avance de los tickets generados por el usuario que este loggeado, donde se verifique lo siguiente:</p> <ul style="list-style-type: none"> <li>• Tickets abiertos que estén en proceso de solución.</li> <li>• Tickets cerrados que ya están resueltos.</li> </ul> <p>Estos tickets no se pueden eliminar para tener trazabilidad del sistema.</p> |

Adaptado de figura 5.

**Figura 43.** Historial de tickets generados por el usuario.


The screenshot shows the Mecasoft user interface. At the top, there is a navigation bar with the Mecasoft logo on the left and the user name 'Bienvenido, Olga Lucia Guerrero' and a 'Cerrar sesión' button on the right. Below the navigation bar, there is a sidebar with two menu items: 'Notificar incidentes' and 'Historial de incidentes', with the latter being highlighted in yellow. The main content area displays a table titled 'Lista de incidentes registrados en el sistema.' with the following data:

| Id incidente | Título | Descripción        | Fecha de solicitud     | Estado   | Prioridad | Responder |
|--------------|--------|--------------------|------------------------|----------|-----------|-----------|
| 21           |        | Esto es una prueba | May 9, 2025, 4:43 a.m. | activo   | Alta      |           |
| 22           |        | Esto es una prueba | May 9, 2025, 4:45 a.m. | activo   | Alta      |           |
| 23           |        | Esto es una prueba | May 9, 2025, 4:45 a.m. | resuelto | Alta      |           |

Adaptado de HelpDeskSystem [12].

**Figura 44.** *Requerimiento funcional 6.*

|                                      |  |
|--------------------------------------|--|
| <b>Código:</b> RF06                  |  |
| <b>Prioridad:</b> Alta               | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Administrador y Técnico  |  |
| <b>Título:</b> Gestión de activos TI |  |
| <b>Descripción</b>                   | <p>El sistema debe permitir agregar equipos o activos de TI con la siguiente información:</p> <ul style="list-style-type: none"> <li>• Id equipo</li> <li>• Tipo de equipo</li> <li>• Estado</li> <li>• Información técnica del equipo</li> <li>• Usuario asignado</li> <li>• Fecha de adquisición</li> </ul> <p>Esta información es importante para trazabilidad tener la mayor cantidad de información. Estos activos de TI se pueden desactivar si ya no tienen un usuario asignado o si ya cumplieron su ciclo de vida útil.</p> |

Adaptado de figura 5.

**Figura 45.** *Formulario de creación o registro de activos TI.*

Adaptado de HelpDeskSystem [12].

**Figura 46.** *Requerimiento funcional 7.*

|                                       |  |
|---------------------------------------|--|
| <b>Código:</b> RF07                   |  |
| <b>Prioridad:</b> Baja                | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Administrador             |  |
| <b>Título:</b> Tipos de activos de TI |  |
| <b>Descripción</b>                    | <p>El sistema debe permitir agregar diferentes tipos de activos en los cuales debe ir la siguiente información:</p> <ul style="list-style-type: none"> <li>• Id de tipo de activo TI</li> <li>• Descripción del activo TI</li> </ul> |

Adaptado de figura 5.

**Figura 47.** Pestaña para creación de activos TI.

Formulario para crear tipos de activo de TI

Tipo de activo tecnologico o equipo:

Describe el tipo de activo que registrara

Agregar tipo

Lista de tipos de activos registrados en el sistema.

| Tipo activo | Editar | Eliminar |
|-------------|--------|----------|
| computador  |        |          |
| impresora   |        |          |
| Perifericos |        |          |

Adaptado de HelpDeskSystem [12].

**Figura 48.** Requerimiento funcional 8.

|                                       |  |
|---------------------------------------|--|
| <b>Código:</b> RF08                   |  |
| <b>Prioridad:</b> Baja                | <b>Estado:</b>   |
| <b>Rol:</b> Empleado y Técnico        |  |
| <b>Título:</b> Evaluación del técnico |  |
| <b>Descripción</b>                    | <p>El sistema debe permitir que el usuario empleado pueda emitir una calificación cuando se cierre un ticket y esta calificación debe contener lo siguiente:</p> <ul style="list-style-type: none"> <li>• Técnico que resolvió la incidencia</li> <li>• Calificación</li> <li>• Comentarios</li> </ul> <p>El rango de calificación debe ser de 1 a 5 siendo 5 satisfactorio.</p> |

Adaptado de figura 5.

Cuando el ticket se encuentra en esta “*resuelto*” se activa la calificación del servicio (ver Figura 49), se registra la calificación y no se puede editar (ver Figura 50) y finalizando esta calificación se envía al área técnica para cerrar el ciclo de prestación del servicio (ver Figura 51).

**Figura 49.** *Calificación del servicio del empleado.*

Chat incidente # 1

**Deiby**  
Atendiendo  
09/05/2025 05:35

**Deiby**  
Ticket resuelto  
09/05/2025 05:35

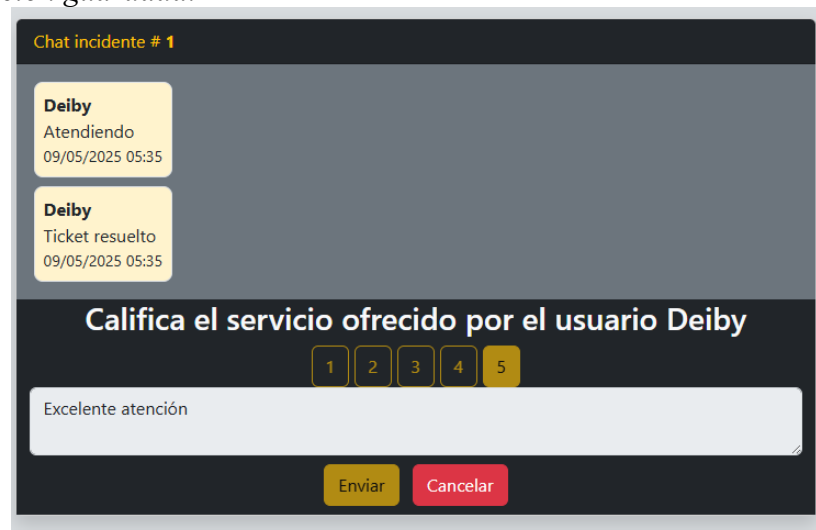
**Califica el servicio ofrecido por el usuario Deiby**

1 2 3 4 5

Escribe tu comentario sobre la atención...

Enviar Cancelar

Adaptado de HelpDeskSystem [12].

**Figura 50.** *Calificación guardada.*

Chat incidente # 1

**Deiby**  
Atendiendo  
09/05/2025 05:35

**Deiby**  
Ticket resuelto  
09/05/2025 05:35

**Califica el servicio ofrecido por el usuario Deiby**

1 2 3 4 5

Excelente atención

Enviar Cancelar

Adaptado de HelpDeskSystem [12].

**Figura 51.** Calificación recibida en el área técnica.

Adaptado de HelpDeskSystem [12].

**Figura 52.** Requerimiento funcional 9.

|  |  |
|--|--|
| <b>Código:</b> RF09                            |  |
| <b>Prioridad:</b> Alta                         | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Administrador                      |  |
| <b>Título:</b> Acuerdos de niveles de servicio |  |
| <b>Descripción</b>                             | <p>El sistema debe permitir asignar las políticas de acuerdos de niveles de servicio con la siguiente información:</p> <ul style="list-style-type: none"> <li>• Id de ANS</li> <li>• Calificación de ANS</li> <li>• Descripción de ANS</li> </ul> <p>Esto debe contener un calificativo de la prioridad de acuerdo a las políticas establecidas con una breve descripción de dicho calificativo.</p> |

Adaptado de figura 5.

**Figura 53.** Pestaña de asignación de acuerdos de niveles de servicio.

| Prioridad | Descripción de prioridad                    | Editar | Eliminar |
|-----------|---|--------|----------|
| Alta      | Respuesta inmediata                         |        |          |
| Media     | Respuesta moderada entre dos a cuatro horas |        |          |
| Baja      | Respuesta lenta entre 8 a 12 horas          |        |          |

Adaptado de HelpDeskSystem [12].

**Figura 54.** Requerimiento funcional 10.

|   |  |
|---|--|
| <b>Código:</b> RF010                          |  |
| <b>Prioridad:</b> Media                       | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Empleado y técnico                |  |
| <b>Título:</b> Información de solución ticket |  |
| <b>Descripción</b>                            | El sistema debe permitir identificar quien resolvió un incidente y también al que resolvió el incidente se le debe agregar que incidente resolvió. |

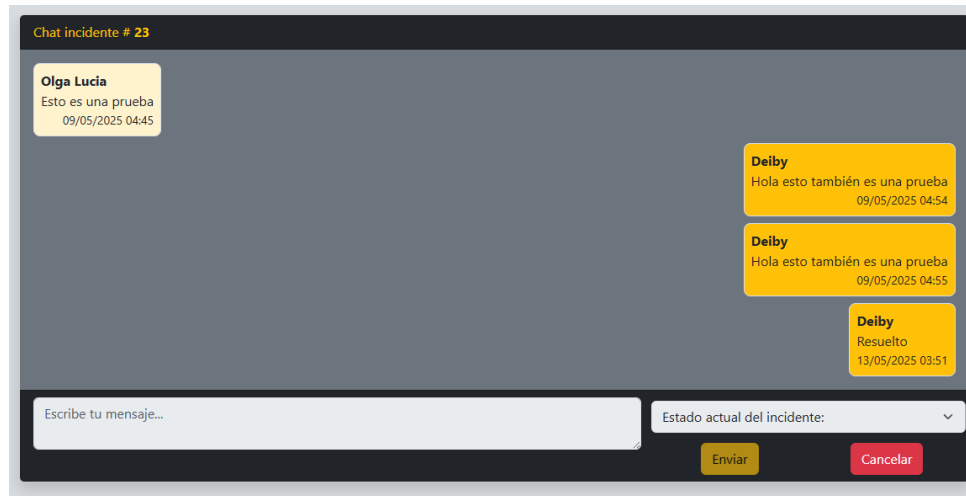
Adaptado de figura 5.

**Figura 55.** Requerimiento funcional 11.

|  |   |
|--|---|
| <b>Código:</b> RF011                     |   |
| <b>Prioridad:</b> Baja                   | <b>Estado:</b> Implementado   |
| <b>Rol:</b> Empleado y técnico           |   |
| <b>Título:</b> Historial de conversación |   |
| <b>Descripción</b>                       | El sistema debe permitir identificar la conversación de los tickets es decir se debe identificar la respuesta del empleado y del técnico. |

Adaptado de figura 5.

En el caso de los requerimientos de la figura 52 y figura 53, se implementó en un chat el cumplimiento de estos requerimientos se realizó lo siguiente (ver Figura 54).

**Figura 56.** Chat de resolución de incidentes.

Adaptado de HelpDeskSystem [12].

**Figura 57.** Requerimiento funcional 12.

|   |   |
|---|---|
| <b>Código: RF012</b>                      |   |
| <b>Prioridad: Media</b>                   | <b>Estado: Implementado</b>   |
| <b>Rol: Administrador y técnico</b>       |   |
| <b>Título: Hoja de vida de activos TI</b> |   |
| <b>Descripción</b>                        | <p>El sistema debe permitir generar una hoja de vida del equipo en caso de todos los posibles casos que pueden suceder como los siguientes:</p> <ul style="list-style-type: none"> <li>• Incidentes afectando al activo</li> <li>• Mantenimiento realizado al activo</li> </ul> <p>Esto para generar una trazabilidad del ciclo de vida del equipo.</p> |

Adaptado de figura 5.

**Figura 58.** Hoja de vida de los activos de TI.

**i** Recuerda que, para registrar mantenimientos, primero deben haberse creado los activos de TI correspondientes. Una vez registrado un mantenimiento, este no podrá ser eliminado del sistema, ya que se conserva como parte del historial para fines de auditoría e integridad de la información.

Registrar mantenimiento ↗

Filtrar por equipo:  
 Selecciona el equipo

Buscar Cancelar

Historial de equipos a los que se les ha realizado mantenimiento

| Identificador Equipo | Procedimiento realizado | Usuario que realizo el mantenimiento | Fecha de mantenimiento  |
|----------------------|-------------------------|--------------------------------------|-------------------------|
| IP1                  | Cambio de tinta         | admin                                | May 5, 2025, 12:19 a.m. |
| CP1                  | Cambio de cargador      | admin                                | May 5, 2025, 12:42 a.m. |
| CP1                  | cambio de memoria ram   | admin                                | May 5, 2025, 12:43 a.m. |
| CP1                  | asdasd                  | admin                                | May 5, 2025, 12:45 a.m. |
| IP1                  | asdasd                  | admin                                | May 5, 2025, 12:47 a.m. |
| CP1                  | dasdasd                 | admin                                | May 5, 2025, 12:50 a.m. |
| CP1                  | dasdasd                 | admin                                | May 5, 2025, 12:53 a.m. |
| CP1                  | asda                    | admin                                | May 5, 2025, 12:55 a.m. |
| IP1                  | dasd                    | admin                                | May 5, 2025, 12:55 a.m. |

Adaptado de HelpDeskSystem [12].

**Figura 59.** Requerimiento funcional 13.

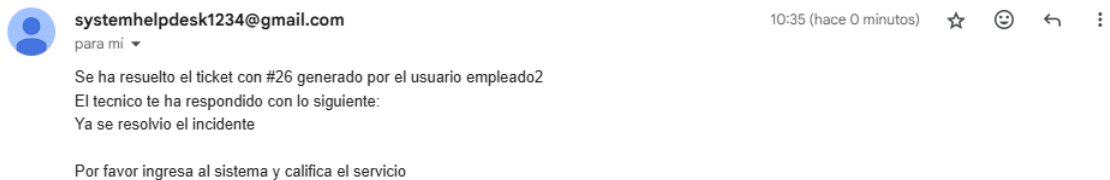
|                                   |   |
|-----------------------------------|---|
| <b>Código: RF013</b>              |   |
| <b>Prioridad: Media</b>           | <b>Estado: Implementado</b>   |
| <b>Rol: Empleado</b>              |   |
| <b>Título: Correo electrónico</b> |   |
| <b>Descripción</b>                | <p>El sistema debe permitir enviar un correo electrónico cuando se genere un ticket y cuando se cierre, y debe contener la siguiente información:</p> <ul style="list-style-type: none"> <li>• Id del ticket generado</li> <li>• Título del ticket</li> <li>• Descripción del ticket</li> <li>• Prioridad del ticket</li> </ul> <p>En caso de que se cierre el ticket debe contener lo siguiente:</p> <ul style="list-style-type: none"> <li>• Id del ticket generado</li> <li>• Título del ticket</li> <li>• Respuesta del técnico</li> <li>• Nombre del técnico que resolvió el incidente</li> </ul> <p>Si se cierra el ticket se habilita la calificación al técnico respectivo.</p> |

Adaptado de figura 5.

**Figura 60.** *Generación de ticket.*

The image shows a web interface for incident notification. At the top, there is a form titled 'Notificación de incidentes' with fields for 'Título incidente:' and a dropdown for '¿Cuál es la prioridad de resolución?' set to 'Media'. Below the form is a large text area for 'Agregue una breve descripción del incidente:'. In the center, a dark modal dialog box displays a success message: 'Operación exitosa' with a green checkmark icon, followed by 'ticket generado con éxito y correo enviado con éxito'. At the bottom of the dialog is a blue 'Aceptar' button.

Adaptado de HelpDeskSystem [12].

**Figura 61.** *Correo de generación de ticket.***Figura 62.** *Correo de cierre de ticket.***Figura 63.** *Requerimiento funcional 14.*

|   |  |
|---|--|
| <b>Código:</b> RF014                          |  |
| <b>Prioridad:</b> Media                       | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Empleado, administrador y técnico |  |
| <b>Título:</b> Acceso a módulos               |  |
| <b>Descripción</b>                            | El sistema debe restringir el acceso a ciertos módulos limitando la modificación de ciertos valores, esto se debe definir cuando se crea el usuario dependiendo del rol. |

Adaptado de figura 5.

Para el cumplimiento de este requerimiento (ver Figura 60), se utilizan los decoradores en las vistas, estos permiten restringir el acceso a los diferentes módulos del sistema (ver Figura 61).

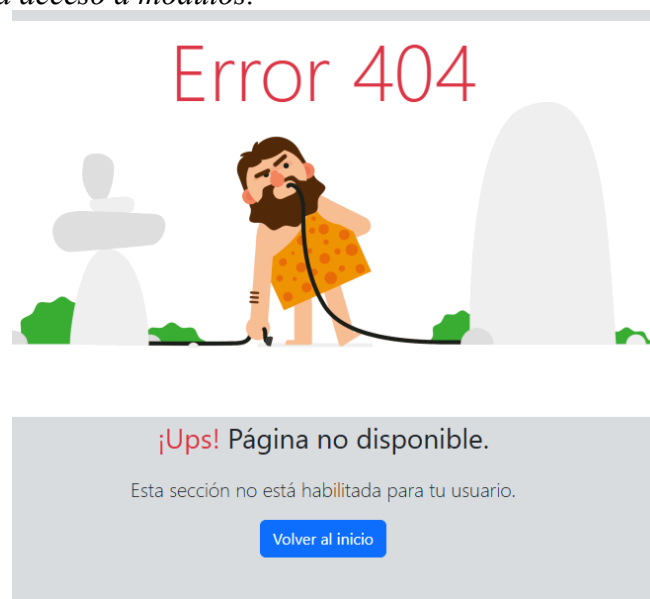
**Figura 64.** Decoradores antes de cada función para restringir el acceso.

```
@login_required
@user_passes_test(is_admin_or_tecnico, login_url=reverse_lazy("pagina_no_disponible"))
def administrador(request):
    contexto=graficos()
    return render(request, "graficos/graficos.html", contexto)
```

Adaptado de HelpDeskSystem [12].

En la figura 61, se requiere que este logueado el usuario y además debe validarse si el usuario es técnico, empleado o administrador para el acceso. Si no hay acceso a este módulo se muestra lo que se identifica en la figura 62.

**Figura 65.** Error 404 para acceso a módulos.

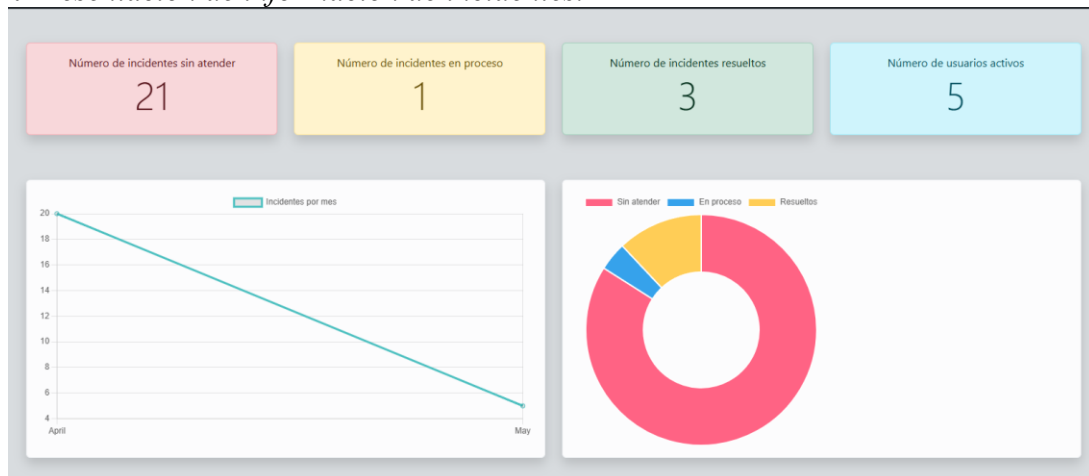


Adaptado de HelpDeskSystem [12].

**Figura 66.** *Requerimiento funcional 15.*

|                                     |  |
|-------------------------------------|--|
| <b>Código:</b> RF015                |  |
| <b>Prioridad:</b> Bajo              | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Administrador y técnico |  |
| <b>Título:</b> Gráficos             |  |
| <b>Descripción</b>                  | <p>El sistema debe permitir mostrar información en un dashboard como la siguiente:</p> <ul style="list-style-type: none"> <li>• Incidentes pendientes</li> <li>• Incidentes resueltos</li> <li>• Incidentes en proceso</li> <li>• Gráfico de incidentes resueltos por mes</li> </ul> <p>Esto con el fin de identificar cierta información de manera gráfica.</p> |

Adaptado de figura 5.

**Figura 67.** *Presentación de información de incidentes.*

Adaptado de HelpDeskSystem [12].

**Figura 68.** *Requerimiento funcional 16.*

|   |  |
|---|--|
| <b>Código:</b> RF016                          |  |
| <b>Prioridad:</b> Media                       | <b>Estado:</b> Implementado  |
| <b>Rol:</b> Administrador                     |  |
| <b>Título:</b> Asignación de áreas de trabajo |  |
| <b>Descripción</b>                            | <p>El sistema debe permitir crear, actualizar o eliminar zonas de trabajo con la siguiente información:</p> <ul style="list-style-type: none"> <li>• Id área de trabajo</li> <li>• Nombre área de trabajo</li> <li>• Descripción del área</li> </ul> <p>Es importante que estas áreas se asignan a personal y a activos de TI.</p> |

Adaptado de figura 5.

**Figura 69.** Creación de áreas de trabajo.

Formulario para crear áreas de trabajo

Nombre del área:  
Escriba el nombre del área...

Descripción de área:  
Describe que actividades se realizan en esta área brevemente...

Crear área

Lista de áreas de trabajo registradas en el sistema

| Nombre del área | Descripción del área                                      | Editar | Eliminar |
|-----------------|---|--------|----------|
| Sistemas        | area encargada para el desarrollo de soporte tecnico      |        |          |
| Contabilidad    | área encargada de la parte financiera de la organización. |        |          |

Adaptado de HelpDeskSystem [12].

**Figura 70.** Requerimiento funcional 17.

|                                      |   |
|--------------------------------------|---|
| <b>Código:</b> RF017                 |   |
| <b>Prioridad:</b> Media              | <b>Estado:</b> Implementado   |
| <b>Rol:</b> Administrador y técnico  |   |
| <b>Título:</b> Mantenimiento equipos |   |
| <b>Descripción</b>                   | <p>El sistema debe permitir realizar mantenimiento a equipos de la organización de la siguiente manera:</p> <ul style="list-style-type: none"> <li>• Id mantenimiento</li> <li>• Activo TI al que se le realiza mantenimiento</li> <li>• Usuario que realizo el mantenimiento</li> <li>• Hallazgos y procedimiento</li> <li>• Conclusiones</li> </ul> <p>Esto con el fin de generar una trazabilidad y mantener optimo el ciclo de vida de los activos.</p> |

Adaptado de figura 5.

**Figura 71.** *Formulario de mantenimiento a realizar.*

Registrar mantenimiento

**Formulario para realizar mantenimiento**

Equipos registrados:  
 Seleccione el equipo

Procedimiento realizado:  
 Explique el procedimiento realizado...

Conclusiones del procedimiento:  
 De sus conclusiones sobre el procedimiento...

Cancelar Registrar mantenimiento

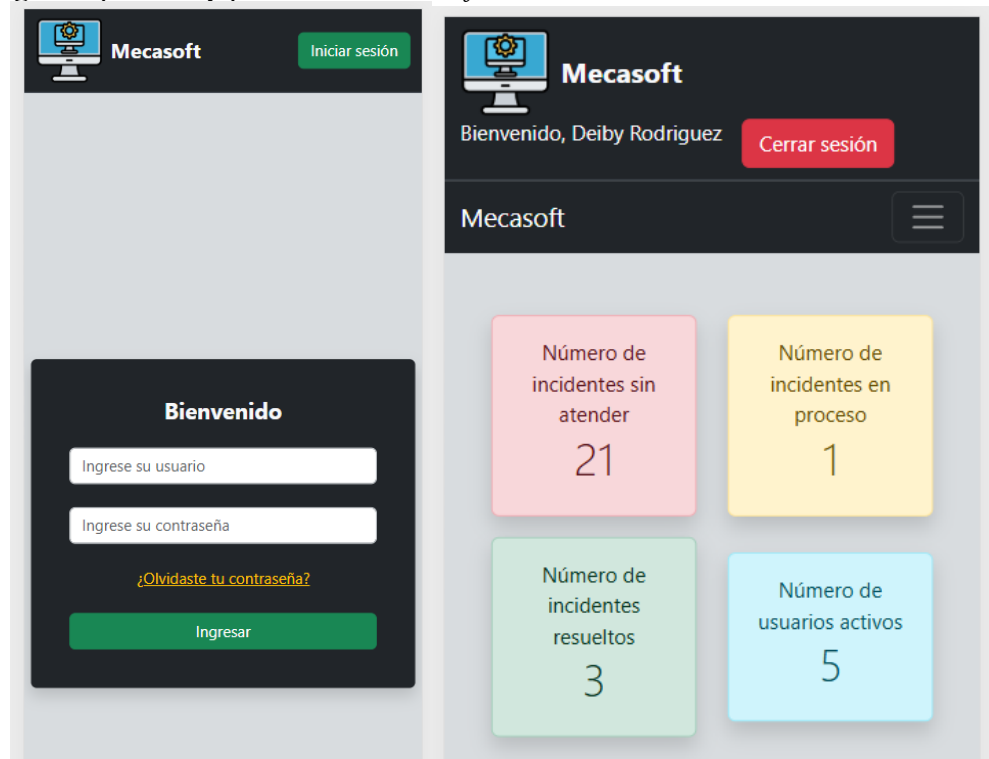
Adaptado de HelpDeskSystem [12].

**Figura 72.** *Requerimiento no funcional 1.*

|                              |   |
|------------------------------|---|
| <b>Código:</b> RNF01         |   |
| <b>Prioridad:</b> Media      | <b>Estado:</b> Implementado   |
| <b>Rol:</b> NA               |   |
| <b>Título:</b> Responsividad |   |
| <b>Descripción</b>           | El sistema debe poder adaptarse a cualquier dispositivo siendo responsivo para equipos de escritorio y/o móviles. |

Adaptado de figura 5.

De acuerdo a la figura 69, se usó Bootstrap que es un framework de CSS muy fácil de implementar el cual ofrece la responsividad para cualquier dispositivo (ver Figura 70 y Figura 71).

**Figura 73.** Login responsivo y presentación de información.

Adaptado de HelpDeskSystem [12].

**Figura 74.** Requerimiento no funcional 2.

|                             |  |
|-----------------------------|--|
| <b>Código:</b> RNF02        |  |
| <b>Prioridad:</b> Media     | <b>Estado:</b> Implementado                                      |
| <b>Rol:</b> NA              |  |
| <b>Título:</b> Optimización |  |
| <b>Descripción</b>          | El sistema debe ejecutarse de manera óptima sin bugs ni errores. |

Adaptado de figura 5.

De acuerdo a la figura 71, el sistema cumple con este requerimiento ya que se optimizo el funcionamiento y no se encontraron errores ni bugs dentro del sistema.

### 4.3 Pruebas y resultados

### 4.3.1 Pruebas a usuarios y valoración de experiencia de usuario

Las pruebas de usuario se realizaron con seis participantes, utilizando las interfaces presentadas en el capítulo anterior. Estos usuarios se encuentran representados de la siguiente manera:

**Figura 75.** *Personas encuestadas y sus roles definidos.*

| Usuario                 | Cantidad |
|-------------------------|----------|
| Personal Administrativo | 2        |
| Personal Técnico        | 2        |
| Empleados               | 2        |
| Total                   | 6        |

Adaptado de encuesta [24].

Estas pruebas se realizaron sobre el sistema en funcionamiento, considerando los diferentes roles definidos, tal como se muestra en la tabla anterior. A cada participante se le asignó un usuario y una contraseña, los cuales fueron entregados previamente. Posteriormente, mediante monitoreo, se permitió el uso autónomo del sistema, asignando ciertas actividades específicas como: creación de tickets, asignación de equipos, verificación de inventario, entre otras. Con ello, se evaluaron todas las interfaces explicadas en el capítulo anterior.

Después de la prueba, se aplicó una encuesta para evaluar la experiencia de usuario, tomando como referencia seis áreas esenciales: facilidad de uso, claridad de la interfaz, rendimiento del sistema (rapidez/respuesta), cumplimiento de funcionalidades esperadas, utilidad del sistema y satisfacción general.

Se analizaron las respuestas a siete preguntas de la encuesta final, enfocándose en las áreas mencionadas anteriormente.

*Facilidad de uso:* Para determinar si el sistema presenta un nivel de complejidad adecuado o si requiere de una capacitación previa, se aplicó la primera pregunta de la encuesta, la cual se muestra en la Figura 76.

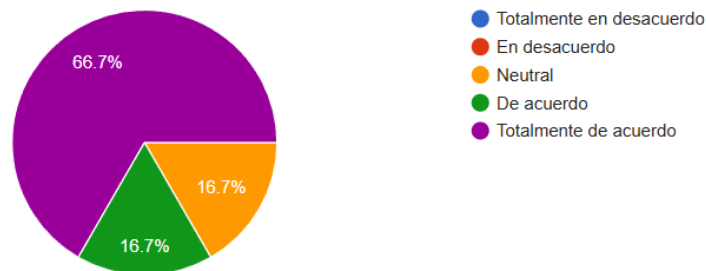
**Figura 76.** *Primera pregunta de la encuesta.*

¿El sistema se puede manejar y entender de forma sencilla?

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 66,7 % de los encuestados manifestó estar totalmente de acuerdo; el 16,7 % indicó estar de acuerdo, mientras que el 16,7 % restante se mostró neutral.

**Figura 77.** *Resultados primera pregunta de la encuesta.*

Adaptado de encuesta [24].

*Claridad de la interfaz:* Se evaluó si la interacción con la interfaz del sistema resulta agradable y comprensible para el usuario. Para ello, se aplicó la segunda pregunta de la encuesta, la cual se muestra en la Figura 78.

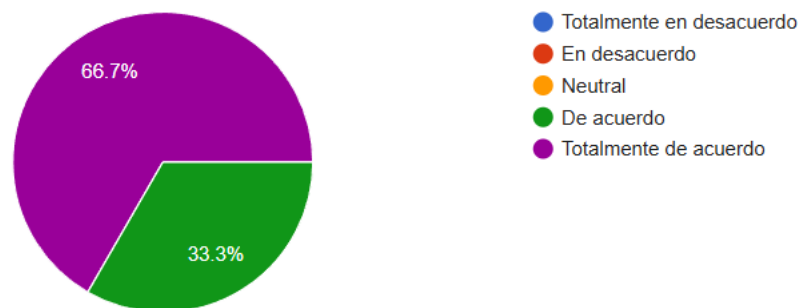
**Figura 78.** Segunda pregunta de la encuesta.

¿El diseño del sistema (colores, Botones, textos) es claro y agradable al uso?

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 66,7 % de los encuestados manifestó estar totalmente de acuerdo con el diseño del sistema, mientras que el 33,3 % restante indicó estar de acuerdo.

**Figura 79.** Resultados segunda pregunta de la encuesta.

Adaptado de encuesta [24].

*Rendimiento del sistema:* Se evaluó la rapidez y capacidad de respuesta del sistema ante las solicitudes realizadas por el usuario. Para ello, se aplicó la tercera pregunta de la encuesta, la cual se muestra en la Figura 80.

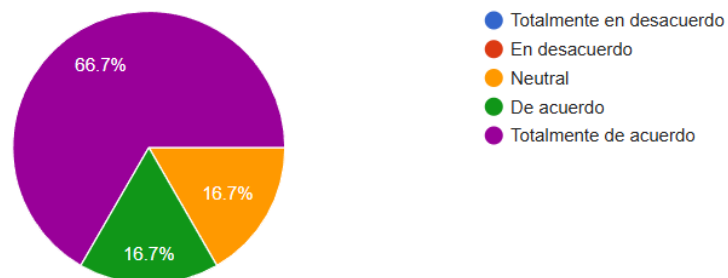
**Figura 80.** Tercera pregunta de la encuesta.

¿El sistema respondió rápidamente a mis acciones?

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 66,7 % de los encuestados manifestó estar totalmente de acuerdo; el 16,7 % indicó estar de acuerdo, mientras que el 16,6 % restante se mostró neutral.

**Figura 81.** Resultados tercera pregunta de la encuesta.

Adaptado de encuesta [24].

*Utilidad del sistema:* Se evaluó la utilidad del sistema en cuanto a su estructura y contenido, con el fin de validar su usabilidad. Para ello, se aplicó la cuarta pregunta de la encuesta, la cual se muestra en la Figura 82.

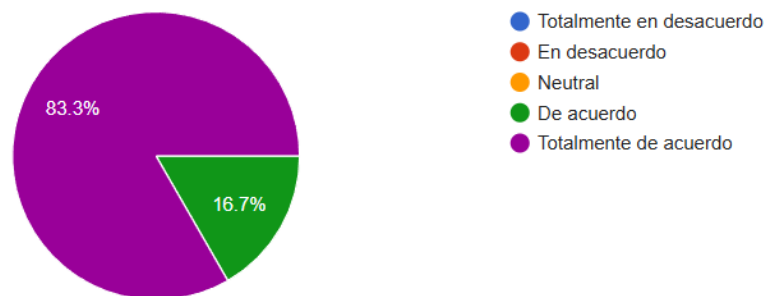
**Figura 82.** *Cuarta pregunta de la encuesta.*

¿El sistema mantiene una estructura visual coherente en todas las pantallas?

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 83,3 % de los encuestados manifestó estar totalmente de acuerdo, mientras que el 16,7 % restante indicó estar de acuerdo.

**Figura 83.** *Resultados cuarta pregunta de la encuesta.*

Adaptado de encuesta [24].

*Funcionalidad:* Se evaluaron las funciones del sistema y su ejecución óptima por parte de los usuarios. Para ello, se aplicó la quinta pregunta de la encuesta, la cual se muestra en la Figura 84.

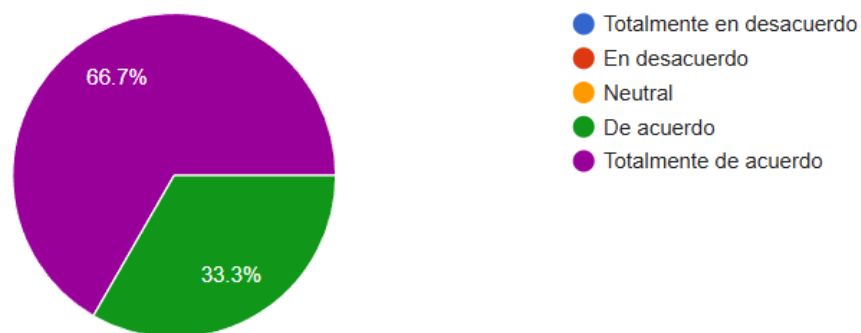
**Figura 84.** *Quinta pregunta de la encuesta.*

¿El sistema ofreció todas las funcionalidades necesarias para realizar mis tareas?

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 66,7 % de los encuestados manifestó estar totalmente de acuerdo, mientras que el 33,3 % restante indicó estar de acuerdo.

**Figura 85.** *Resultados quinta pregunta de la encuesta.*

Adaptado de encuesta [24].

*Evaluación de errores y funcionamiento:* Se evaluaron los posibles errores que podría presentar el sistema y se verificó su funcionamiento óptimo. Para ello, se aplicó la sexta pregunta de la encuesta, la cual se muestra en la Figura 86.

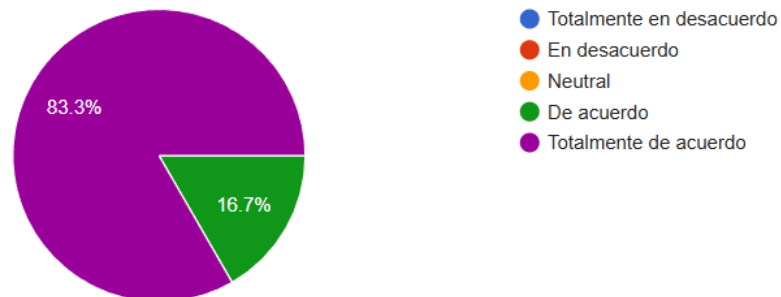
**Figura 86.** *Sexta pregunta de la encuesta.*

No encontré errores técnicos mientras usaba el sistema

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 83,3 % de los encuestados manifestó estar totalmente de acuerdo, mientras que el 16,7 % restante indicó estar de acuerdo.

**Figura 87.** *Resultados sexta pregunta de la encuesta.*

Adaptado de encuesta [24].

*Satisfacción general:* Se evaluó la disposición del usuario a recomendar el sistema, de acuerdo con su experiencia previa. Para ello, se formuló la séptima pregunta de la encuesta, la cual se muestra en la Figura 4.

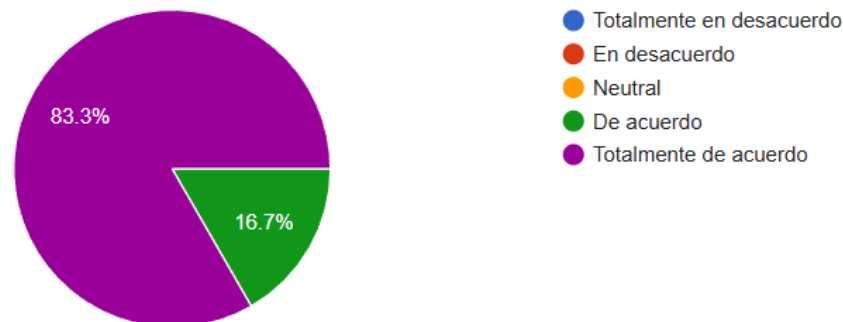
**Figura 88.** *Séptima pregunta de la encuesta.*

Recomendaría el uso de este sistema a otras personas o empresas

- Totalmente en desacuerdo
- En desacuerdo
- Neutral
- De acuerdo
- Totalmente de acuerdo

Adaptado de encuesta [24].

El 83,3 % de los encuestados manifestó estar totalmente de acuerdo, mientras que el 16,7 % restante indicó estar de acuerdo.

**Figura 89.** *Resultados séptima pregunta de la encuesta.*

Adaptado de encuesta [24].

#### 4.3.2 Resultados de encuesta de satisfacción

A partir del análisis de los resultados, se elaboró una tabla general en la que se identifica el nivel de aceptación de cada una de las áreas esenciales evaluadas:

**Tabla 5.** *Resultados de encuesta de satisfacción al usuario.*

| Área evaluada    | % de acuerdo o totalmente de acuerdo |
|------------------|--------------------------------------|
| Facilidad de uso | 83.3 % (5 de 6 personas)             |

| Área evaluada                   | % de acuerdo o totalmente de acuerdo |
|---------------------------------|--------------------------------------|
| Claridad de interfaz            | 100% (6 de 6 personas)               |
| Rendimiento del sistema         | 83.3% (5 de 6 personas)              |
| Cumplimiento de funcionalidades | 100% (6 de 6 personas)               |
| Utilidad del sistema            | 100% (6 de 6 personas)               |
| Satisfacción general            | 100% (6 de 6 personas)               |

Nota: Esta tabla se realizó de acuerdo a los resultados obtenidos de la encuesta [24].

En términos generales, el sistema cumple con los requisitos básicos de usabilidad para el usuario, lo cual se refleja en una buena aceptación por parte de los mismos.

### 4.3.3 Impacto en la productividad

Para analizar el impacto del sistema en la productividad de una organización, se presenta a continuación una tabla comparativa que permite visualizar los posibles cambios generados por su implementación:

**Tabla 6.** *Comparación de impacto en la productividad.*

| Actividad evaluada     | Situación actual  | Sistema desarrollado   | Mejora estimada              |
|------------------------|---|--|------------------------------|
| Registro de incidentes | Registro manual mediante hojas de Excel. El proceso toma entre 10 y 20 minutos.                   | Registro directo a través del sistema, con una duración aproximada de 3 minutos. | Ahorro del 70% en tiempo     |
| Gestión de inventario  | Administración mediante Excel, con actualizaciones esporádicas o manuales. Toma entre 1 a 2 días. | Control en tiempo real desde el sistema.   | Mejora del 70% en eficiencia |

| <b>Actividad evaluada</b>                           | <b>Situación actual</b>   | <b>Sistema desarrollado</b>  | <b>Mejora estimada</b>                     |
|---|---|--|--|
| Comunicación entre usuarios                         | Se realiza por correo electrónico o medios externos, lo cual genera demoras. El tiempo de respuesta es entre 4 a 6 horas. | El sistema incorpora un módulo de chat interno y notificaciones por correo. Tiempo de respuesta: 30 min – 1 h. | Mejora del 80% en fluidez comunicativa     |
| Seguimiento de tickets                              | No existe trazabilidad clara ni seguimiento eficiente de los incidentes reportados.                                       | El sistema permite visualizar el estado, tiempos de respuesta y responsables asignado.                         | 100 % de trazabilidad                      |
| Consulta de historial de incidentes o mantenimiento | Revisión manual de documentos o archivos dispersos. El proceso puede tomar entre 1 a 2 días.                              | Consulta centralizada e inmediata desde el sistema.  | Mejora del 90% en acceso a la información. |

Nota: Esta tabla se realizó de acuerdo a los resultados obtenidos de la encuesta y a la información dada en la metodología ITIL [24],[3].

La tabla 6, fue elaborada a partir de estimaciones obtenidas durante las pruebas realizadas con usuarios reales del sistema, contrastando estos resultados con los métodos tradicionales que actualmente se utilizan para la gestión tecnológica. En ella se evidencian mejoras significativas, tales como: una reducción de hasta un 70% en el tiempo requerido para el registro de incidentes, una mejora del 70% en la eficiencia de la gestión de inventario, un 80% en la fluidez de la comunicación entre los miembros de la organización, un 100% en la trazabilidad de los incidentes registrados y hasta un 90% en el acceso a la información. Estos indicadores reflejan un impacto positivo en la productividad y la eficiencia operativa de las organizaciones que implementen el sistema.

#### 4.3.4 Resultados esperados

**Tabla 7.** *Resultados esperados del proyecto.*

| <b>Resultado</b>                 | <b>Indicador</b>   | <b>Objetivo relacionado</b> |
|----------------------------------|--------------------|-----------------------------|
| Diagrama de casos de uso         | Grafico            | Objetivo específico 1       |
| Historias de usuario             | Grafico            | Objetivo específico 1       |
| Diagrama de base de datos        | Gráfico y software | Objetivo específico 2       |
| Software funcional con 3 módulos | Software           | Objetivo específico 3       |

Nota: Esta tabla se realizó teniendo en cuenta los objetivos específicos de este proyecto.

#### 4.3.5 Casos de uso e historias de usuario

Como resultado de la representación de requerimientos se entregan los casos de uso e historias de usuario, los cuales están en capítulos anteriores entre la página 43, 44 y 48.

Teniendo en cuenta los resultados esperados para el objetivo específico 1, se puede decir que se cumplió en su totalidad.

#### 4.3.6 Diagrama de base de datos

Como resultado del diseño y desarrollo de base de datos demostrando integridad y confiabilidad en la información, se puede tener acceso al diseño de la base datos en la página 47, además de poder acceder al código SQL de creación de la base de datos y el esquema creado los cuales se pueden acceder a través de diseño de base de datos [8].

Teniendo en cuenta los resultados esperados para el objetivo específico 2, se puede decir que se cumplió en su totalidad.

#### **4.3.7 Software funcional**

Como resultado del software de help desk que se puede ejecutar en cualquier dispositivo móvil, Tablet o PC, ya que está desarrollado en entorno web. La evidencia que se desarrolló se encuentra entre las páginas 56 a la 74, donde se identifica el cumplimiento de los requerimientos previamente definidos. Para acceder al código base de todo el sistema se puede descargar y ejecutar desde HelpDeskSystem [12].

Teniendo en cuenta los resultados esperados para el objetivo específico 3, se puede decir que se cumplió en su totalidad.

### **5. Discusión**

Durante el desarrollo del proyecto y al analizar los objetivos propuestos, se identificó que para escalar el sistema a organizaciones más grandes es altamente recomendable utilizar frameworks con mayor optimización y seguridad, que garanticen un funcionamiento estable según el flujo de información que la organización requiera.

Además, también es importante considerar el uso de un framework de frontend que ofrezca mayor velocidad y eficiencia, con el fin de brindar una experiencia más cómoda al usuario. En este proyecto no se implementó dicha solución debido al tiempo considerable que implica el aprendizaje de un framework frontend. No obstante, herramientas como React.js o Vue.js podrían ser opciones adecuadas.

Cabe destacar que Django es un framework escalable, optimizado y con altos estándares de seguridad. Sin embargo, es pertinente evaluar otras alternativas que podrían ofrecer mejores resultados dependiendo del contexto y las necesidades específicas del proyecto.

## 6. Conclusiones

La recolección de requerimientos basada en las buenas prácticas de ITIL permitió estructurar un sistema adaptable a distintas organizaciones, identificando de manera clara los requerimientos funcionales y no funcionales. Este enfoque facilitó abordar las principales deficiencias en la gestión tecnológica de las MiPymes, asegurando que el sistema respondiera efectivamente a las necesidades operativas reales.

El desarrollo del sistema Help Desk basado en la metodología ITIL permitió crear una solución web escalable, accesible y adaptada a las necesidades fundamentales de las MiPymes, integrando en una sola plataforma la gestión de solicitudes, el seguimiento de incidentes y el control de inventario. Esta herramienta facilita el flujo eficiente de información, destacando el uso de notificaciones por correo electrónico para mantener actualizados a los usuarios sobre el estado de sus solicitudes.

Las pruebas realizadas al sistema confirmaron la integridad y confidencialidad de la información, lo que fortalece la confianza en la plataforma desarrollada. Asimismo, se logró mejorar los procesos de gestión tecnológica al garantizar una administración segura, eficiente y organizada de los datos, generando beneficios tanto para los usuarios como para la continuidad operativa de las empresas.

Las interfaces de usuario se desarrollaron de manera intuitiva y adaptable a cualquier dispositivo. Se realizaron pruebas en diversos entornos y plataformas, lo que permitió garantizar una experiencia de uso fluida y accesible desde computadores de escritorio, portátiles, tabletas y teléfonos móviles, mejorando así la usabilidad y facilitando la adopción del sistema a cualquier usuario.

## **7. Trabajos futuros**

Una forma de mejorar este proyecto es integrarlo con un framework de frontend, lo que permitiría hacer el sistema más dinámico y ofrecer a los usuarios una experiencia de uso más cómoda. Para ello, también es importante considerar elementos clave como el diseño UX (experiencia de usuario) y UI (interfaz de usuario), disciplinas enfocadas en optimizar la interacción y la usabilidad de las interfaces. Con estos aspectos, se lograría una herramienta más atractiva y funcional, con mayor potencial comercial.

### Referencias

- [1] Atlassian, “¿Qué es la gestión de activos de TI? Una guía”. [En línea]. Disponible en: <https://www.atlassian.com/es/itsm/it-asset-management>. [Consultado: 16-sep-2024].
- [2] Atlassian, “¿Qué es la gestión de servicios de TI (ITSM)?”. [En línea]. Disponible en: <https://www.atlassian.com/es/itsm>. [Consultado: 18-sep-2024].
- [3] AXELOS, *ITIL Foundation: ITIL 4 Edition*. Norwich, Inglaterra: TSO, 2019.
- [4] “Definición de Procesos de Casos de Uso,” IBM. [En línea]. Disponible en: <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>. [Accedido: 13-sep-2024].
- [5] Departamento Nacional de Planeación, "Micro, Pequeña y Mediana Empresa." [En línea]. Disponible en: [https://www.dnp.gov.co/LaEntidad\\_/subdireccion-general-prospectiva-desarrollo-nacional/direccion-innovacion-desarrollo-empresarial/Paginas/micro-pequena-y-mediana-empresa.aspx](https://www.dnp.gov.co/LaEntidad_/subdireccion-general-prospectiva-desarrollo-nacional/direccion-innovacion-desarrollo-empresarial/Paginas/micro-pequena-y-mediana-empresa.aspx). [Accedido: 13-sep-2024].
- [6] J. K. Díaz Hidalgo, “Diseño de software de Help Desk en la nube basado en ITIL para la Cruz Roja Ecuatoriana,” Universidad Politécnica Salesiana, Quito, 2024. [En línea]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/27279/4/TTS1795.pdf>.
- [7] “Django overview,” Django Project. [En línea]. Disponible en: <https://www.djangoproject.com/start/overview/>. [Consultado: 24-sep-2024].
- [8] “Diseño base de datos,” D. R., Google Docs. [En línea]. Disponible en: [https://drive.google.com/drive/folders/1g74p0AynLBbExbIayFjJznIfcELckmAR?usp=drive\\_link](https://drive.google.com/drive/folders/1g74p0AynLBbExbIayFjJznIfcELckmAR?usp=drive_link).
- [9] “Fundamentos de bases de Datos (5a. Ed.) - Silberschatz : Free download, Borrow, and streaming,” *Internet Archive*. [En línea]. Disponible en:

- <https://archive.org/details/fundamentosdebasesdedatos5a.ed.abrahamsilberschatzhenryf.korths.sudarshan/page/n3/mode/2up>. [Accedido: 1-abr-2024].
- [10] “Gestión de Activos de TI,” Hewlett Packard Enterprise. [En línea]. Disponible en: <https://www.hpe.com/lamerica/es/what-is/it-asset-management.html>. [Accedido: 13-sep-2024].
- [11] “Gestión de servicios de TI usando ITIL en las MIPYME,” J. C. Gutiérrez Cantor, B. N. Guzmán Prieto, D. S. Chisco Quintero. Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, 2017. [En línea]. Disponible en: <https://repositorio.escuelaing.edu.co/bitstream/handle/001/703/Chisco%20Quintero,%20David%20Santiago%20-%202017.pdf?sequence=1>.
- [12] “HelpDeskSystem,” D. R., GitHub. [En línea]. Disponible en: <https://github.com/DeibyRG23/HelpDeskSystem.git>.
- [13] “Historias de Usuario,” Scrum México. [En línea]. Disponible en: <https://scrum.mx/informate/historias-de-usuario#queesunahdu>. [Accedido: 13-sep-2024].
- [14] “IT Incident,” SolarWinds. [En línea]. Disponible en: <https://www.solarwinds.com/es/resources/it-glossary/it-incident>. [Accedido: 13-sep-2024].
- [15] Icontec, “Seguridad de la información, ciberseguridad y protección de la privacidad. Sistemas de gestión de seguridad de la información. Requisitos.” [En línea]. Disponible en: <https://tienda.icontec.org/gp-ntc-iso-iec-seguridad-de-la-informacion-ciberseguridad-y-proteccion-de-la-privacidad-sistemas-de-gestion-de-seguridad-de-la-informacion-requisitos-ntc-iso-iec27001-2022.html>. [Consultado: 23-ago-2024].

- [16] L. J. Huerta Julca, “Implantación de un sistema Help Desk para el proceso de atención de incidencias de hardware y software bajo la modalidad Open Source en la empresa Mixercon S.A,” Universidad Peruana de Integración Global, Perú, 2014. [En línea]. Disponible en: <https://cazova.wordpress.com/wp-content/uploads/2015/01/tesis-help-desk.pdf>.
- [17] “Ley 1341 de 2009 - Gestor Normativo,” Gov.co. [En línea]. Disponible en: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36913>.  
[Consultado: 1-oct-2024].
- [18] “Decreto 1078 de 2015 Sector de Tecnologías de la Información y las Comunicaciones - Gestor Normativo,” Gov.co. [En línea]. Disponible en: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=77888>.  
[Consultado: 1-oct-2024].
- [19] “Decreto 1412 de 2017 - Gestor Normativo,” Gov.co. [En línea]. Disponible en: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=83247>.  
[Consultado: 23-ago-2024].
- [20] R. C. Palomares, “Análisis y diseño del sistema Help Desk para la gestión de incidencias en una empresa de TI,” Universidad Continental, Huancayo, 2020. [En línea]. Disponible en: [https://repositorio.continental.edu.pe/bitstream/20.500.12394/7472/2/IV\\_FIN\\_103\\_TE\\_Cuchula\\_Palomares\\_2020.pdf](https://repositorio.continental.edu.pe/bitstream/20.500.12394/7472/2/IV_FIN_103_TE_Cuchula_Palomares_2020.pdf).
- [21] I. Sommerville y C. V. Olguín, *Ingeniería de Software*. México: Addison-Wesley, 2011.
- [22] R. S. Solarte Coral y C. A. Calpa Arellano, “Implementación de un sistema de gestión universitaria utilizando Python y Django,” Corporación Universitaria Remington, Ipiales, 2024. [En línea]. Disponible en:

<https://repositorio.uniremington.edu.co/server/api/core/bitstreams/605c9242-0892-4857-a3b2-b2f18ef4dfd0/content>.

[23] “Software Libre,” Universidad Complutense de Madrid. [En línea]. Disponible en:

<https://www.ucm.es/oficina-de-software-libre/software-libre>. [Accedido: 13-sep-2024].

[24] D. A. Rodríguez Guerrero, *Valoración del sistema* [Encuesta de Google], 2025. [En línea].

Disponible en:

[https://docs.google.com/forms/d/e/1FAIpQLSc\\_9EUPTeAfcevuKWS1HtTQdLKG8z6gwt\\_ADjK7u7RqsRMKcQ/viewform?usp=sharing&ouid=112072050585406375988](https://docs.google.com/forms/d/e/1FAIpQLSc_9EUPTeAfcevuKWS1HtTQdLKG8z6gwt_ADjK7u7RqsRMKcQ/viewform?usp=sharing&ouid=112072050585406375988).