



Universidad Nacional de Loja

Área de la Energía, las Industrias, y los
Recursos Naturales no Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

**“SISTEMA HELP DESK PARA EL DEPARTAMENTO DE
MANTENIMIENTO Y SOPORTE TÉCNICO DE LA
UNIVERSIDAD NACIONAL DE LOJA”**

TESIS DE GRADO PREVIA LA
OBTENCIÓN DEL TÍTULO DE
INGENIERA EN SISTEMAS

AUTOR:

DIANA ALEXANDRA MOROCHO PUCHAICELA

DIRECTOR:

ING. CARLOS MIGUEL JARAMILLO CASTRO, Mg. Sc.

Loja – Ecuador

2016

CERTIFICACIÓN DEL DIRECTOR

Ingeniero

Carlos Miguel Jaramillo Castro, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD NACIONAL DE LOJA

CERTIFICA:

Haber dirigido y corregido en todas sus partes de desarrollo del presente trabajo de titulación elaborado previo a la obtención del título de Ingeniería en Sistemas, titulado **“SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA”**, autoría de la egresada Diana Alexandra Morocho Puchaicela. En razón de que la misma reúne a satisfacción los requisitos de fondo y forma, exigidos para una investigación de éste nivel.

Por lo tanto autorizo su presentación, sustentación y defensa ante el tribunal designado para el efecto.

Loja, 7 de Diciembre del 2015



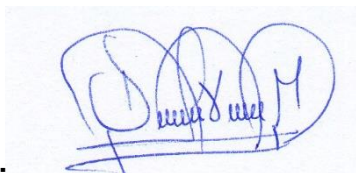
.....
Ing. Carlos Miguel Jaramillo Castro, Mg. Sc.

DIRECTOR DE TESIS

AUTORÍA

Yo, **DIANA ALEXANDRA MOROCHO PUCHAICELA**, declaro ser la autora del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

A handwritten signature in blue ink, appearing to be 'Diana Alexandra Morocho Puchaicela', written over a light blue rectangular background.

Firma:

Cédula: 1104662745

Fecha: 13-01-2016

CARTA DE AUTORIZACIÓN

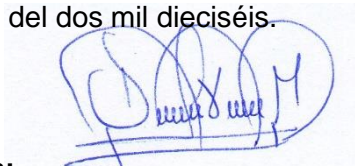
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **DIANA ALEXANDRA MOROCHO PUCHAICELA**, declaro ser la autora de la tesis titulada: “**SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA**”, como requisito para optar al grado de: **INGENIERA EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los trece días del mes de enero del dos mil dieciséis.



Firma:

Autor: Diana Alexandra Morocho Puchaicela

Cédula: 1104662745

Dirección: Loja (Leningrado entre Bucarest y París s/n, Barrio San Cayetano Bajo.)

Correo Electrónico: mor8diana@gmail.com

Teléfono: 072611659 **Celular:** 0990493445

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Carlos Miguel Jaramillo Castro, Mg.Sc.

Tribunal de grado: Ing. Alex Vinicio Padilla Encalada, Mg.Sc.

Ing. Ana Lucía Colala Troya, Mg.Sc.

Ing. Roberth Gustavo Figueroa Díaz, M.Sc.

DEDICATORIA

Por lo que representa en mi historia, dedico el resultado de éste esfuerzo:

A Dios, por concederme la perseverancia, sabiduría y fortaleza para vencer los retos.

A mis padres, la Sra. María Puchaicela y el Sr. Teófilo Morocho, que con amor me han inculcado los mejores valores para mantener la esencia de un ser humano auténtico, porque a lo largo de mi vida han dado su mejor esfuerzo para velar por mi bienestar y siempre han estado ahí para darme la mano, definitivamente, por demostrarme que los problemas son pasajeros y que hasta de lo malo se puede obtener algo positivo.

A mi esposo Nelson Leonardo González, por tu inmenso amor, al compartir el día a día conmigo, en la alegría y en la adversidad, por enseñarme que la felicidad no es una opción sino una decisión de vida, sencillamente, por ser mi motivación más grande para soñar y hacerlos realidad.

Finalmente, a mi pequeño hijo Pablo Andrés, porque cada día aprendo de ti y me motivas a no rendirme porque tú sigues mis pasos.

AGRADECIMIENTO

Es mi deber expresar un sincero agradecimiento a:

La Universidad Nacional de Loja, en la carrera de Ingeniería en Sistemas del Área de la Energía, las Industrias y los Recursos Naturales No Renovables, donde forjé mis conocimientos, y a quien le debo mi formación profesional.

Al Ing. Carlos Jaramillo, al compartir sus conocimientos, tiempo y ayuda desinteresada durante el transcurso de las tutorías para dirigir mi proyecto de titulación.

Al Ing. Milton Labanda, por proveerme los lineamientos técnicos y facilidades logísticas de la Unidad de Telecomunicaciones e Información, para el desarrollo del sistema.

A la Ing. Mabel Rodríguez, por su tiempo al proporcionarme la información necesaria referente al flujo de procesos del departamento de Mantenimiento y Soporte Técnico.

Finalmente, a mis familiares y amigos, especialmente a mi hermana Ángeles por mantenerse siempre al pendiente de mí.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DEL DIRECTOR.....	II
AUTORÍA.....	III
CARTA DE AUTORIZACIÓN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
ÍNDICE DE CONTENIDOS.....	VII
ÍNDICE DE TABLAS.....	X
ÍNDICE DE FIGURAS.....	XII
1. TÍTULO.....	1
2. RESUMEN.....	2
2.1. ABSTRACT.....	3
3. INTRODUCCIÓN.....	4
4. REVISIÓN LITERARIA.....	6
4.1. MARCO REFERENCIAL.....	6
4.1.1. UNIVERSIDAD NACIONAL DE LOJA.....	6
4.1.2. HELP DESK.....	10
4.1.3. ITIL.....	11
4.1.4. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	14
4.1.5. PLAN DE VERIFICACIÓN Y VALIDACIÓN.....	21
4.2. TECNOLOGÍAS DE DESARROLLO.....	22
4.2.1. ARQUITECTURA DEL SISTEMA HELP DESK.....	22
4.2.2. HERRAMIENTAS Y TECNOLOGÍA DEL SISTEMA HELP DESK.....	24
4.2.3. METODOLOGÍA ITIL.....	26
4.3. SISTEMAS HELP DESK - CASO PRÁCTICO.....	30
4.3.1. PROYECTO GLPI.....	30
4.3.2. KMKEY HELP DESK.....	30

4.3.3.	ANÁLISIS COMPARATIVO	32
5.	MATERIALES Y MÉTODOS.....	33
5.1.	RECURSOS	33
5.1.1.	TALENTO HUMANO	33
5.1.2.	BIENES	33
5.1.3.	SERVICIOS.....	33
5.1.4.	IMPREVISTOS	34
5.1.5.	TOTALIDAD DE RECURSOS	34
5.2.	MÉTODOS Y TÉCNICAS	35
5.2.1.	MÉTODOS	35
5.2.2.	TÉCNICAS	35
5.2.3.	METODOLOGÍA DE DESARROLLO DE SOFTWARE	36
6.	RESULTADOS	38
6.1.	FASE DE PLANEACIÓN	38
6.1.1.	HISTORIAS DE USUARIO	38
6.1.2.	VELOCIDAD DEL PROYECTO	43
6.1.3.	ROLES DE XP.....	44
6.1.4.	ROLES DE USUARIO DEL SISTEMA HELP DESK	44
6.1.5.	REQUERIMIENTOS FUNCIONALES.....	45
6.1.6.	REQUERIMIENTOS NO FUNCIONALES.....	47
6.1.7.	PLAN DE ITERACIONES	48
6.2.	FASE DE DISEÑO.....	49
6.2.1.	METÁFORA DEL SISTEMA	49
6.2.2.	DIAGRAMA GENERAL DE CASOS DE USO.....	50
6.2.3.	MODELO CONCEPTUAL.....	51
6.2.4.	TARJETAS CRC	52
6.3.	FASE DE CODIFICACIÓN	56
6.3.1.	DETALLE DE LA IMPLEMENTACIÓN.....	56

6.3.2.	EJECUCIÓN DEL PLAN DE ITERACIONES	65
6.4.	FASE DE PRUEBAS	79
6.4.1.	PLAN DE VALIDACIÓN Y VERIFICACIÓN	79
6.4.2.	EJECUCIÓN DEL PLAN DE VERIFICACIÓN Y VALIDACIÓN	83
7.	DISCUSIÓN.....	112
7.1.	DESARROLLO DE LA PROPUESTA	112
8.	CONCLUSIONES.....	114
9.	RECOMENDACIONES.....	115
10.	BIBLIOGRAFÍA	116
11.	ANEXOS	119
	ANEXO 1. CERTIFICADO DE ACEPTACIÓN DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN.....	119
	ANEXO 2. CERTIFICADO DE ACEPTACIÓN RESPONSABLE DE VERIFICACIÓN DE REQUERIMIENTOS.....	120
	ANEXO 3. CERTIFICADO DE TRADUCCIÓN.	121
	ANEXO 4. ORGANIGRAMA ESTRUCTURAL DE LA UNIVERSIDAD NACIONAL DE LOJA.....	122
	ANEXO 5. ENTREVISTA DIRIGIDA A LA ING. MABEL RODRÍGUEZ.....	123
	ANEXO 6. REGISTRO ANECDÓTICO.....	125
	ANEXO 7. MODELO DEL DOMINIO.....	129
	ANEXO 8. MODELO DE CLASES	130
	ANEXO 9. MODELO DE BASE DE DATOS.....	131
	ANEXO 10. MODELO DE ENCUESTA APLICADA.....	132

ÍNDICE DE TABLAS

Tabla 1. Orgánico Funcional de la Unidad de Telecomunicaciones e Información.....	9
Tabla 2. Comparativa del proyecto GLPI y KMKEY HELP DESK.....	32
Tabla 3. Presupuesto Talento Humano.....	33
Tabla 4. Presupuesto de Recursos Técnicos.....	33
Tabla 5. Presupuesto de Servicios.....	33
Tabla 6. Presupuesto de Imprevistos.....	34
Tabla 7. Presupuesto de Totalidad de Recursos.....	34
Tabla 8. Índice de valoración del campo Prioridad de Negocio.....	39
Tabla 9. Índice de valoración del campo Riesgo De Desarrollo.....	39
Tabla 10. Índice de valoración del campo Puntos Estimados.....	40
Tabla 11. Índice de valoración del campo Iteración asignada.....	40
Tabla 12. Historia de usuario tipos de mantenimiento.....	41
Tabla 13. Historia de usuario tipos de personal.....	41
Tabla 14. Historia de usuario solicitud de asistencia técnica.....	42
Tabla 15. Solicitud de asistencia técnica postergada.....	42
Tabla 16. Historia de usuario registro de equipo.....	43
Tabla 17. Historia de usuario entrega de equipos.....	43
Tabla 18. Velocidad del proyecto.....	44
Tabla 19. Roles de XP.....	44
Tabla 20. Roles de usuario del sistema Help Desk.....	45
Tabla 21. Requerimientos funcionales del sistema Help Desk.....	47
Tabla 22. Requerimientos no funcionales del sistema Help Desk.....	48
Tabla 23. Plan de entrega Sistema Help Desk.....	48
Tabla 24. Metáfora del sistema.....	50
Tabla 25. Tarjeta CRC Centro de Asistencia.....	53
Tabla 26. Tarjeta CRC Perfil.....	53
Tabla 27. Tarjeta CRC Incidencia.....	54
Tabla 28. Tarjeta CRC Estado de incidencia.....	55
Tabla 29. Tarjeta CRC Equipo.....	55
Tabla 30. Tarjeta CRC Evaluación.....	55
Tabla 31. Iteración PE-001.....	66
Tabla 32. Iteración PE-002.....	67
Tabla 33. Iteración PE-003.....	69
Tabla 34. Iteración PE-004.....	70

Tabla 35. Iteración PE-005	71
Tabla 36. Iteración PE-006.	72
Tabla 37. Iteración PE-007.	73
Tabla 38. Iteración PE-008.	74
Tabla 39. Iteración PE-009	75
Tabla 40. Iteración PE-010.	76
Tabla 41. Iteración PE-011.	77
Tabla 42. Iteración PE-012.	79
Tabla 43. Características evaluadas con la prueba de funcionalidad.	81
Tabla 44. Características evaluadas con la prueba de interfaz de usuario.	81
Tabla 45. Características evaluadas con la prueba de carga.	82
Tabla 46. Estrategia de verificación.	82
Tabla 47. Prueba de Funcionalidad PF-001	85
Tabla 48. Prueba de funcionalidad PF-002.	87
Tabla 49. Prueba de funcionalidad PF-003	89
Tabla 50. Prueba de Funcionalidad PF-004	91
Tabla 51. Comparativa de procesos.	93
Tabla 52. Prueba de Interfaz de usuario - Administrar Incidencia – PIU-002.	98
Tabla 53. Prueba de Interfaz de usuario - Asignar Incidencia – PIU-002.	101
Tabla 54. Prueba de Interfaz de usuario – Atender Incidencia – PIU-003	103
Tabla 55. Prueba de Interfaz de usuario – Administrar Sistema – PIU-004.....	105
Tabla 56. Prueba de Carga – URL de Inicio.....	108
Tabla 57. Prueba de Carga – URL Lista de Incidencias.....	111
Tabla 58. Resumen de carga de datos.	111
Tabla 59. Registro anecdótico de las personas y dependencias relacionadas al departamento de Soporte Técnico.....	125
Tabla 60. Registro anecdótico del tipo de asistencia que ofrecen en el departamento de Soporte Técnico.....	126
Tabla 61. Registro anecdótico de solicitud de asistencia técnica.	127
Tabla 62. Registro anecdótico de recepción de equipos.	128

ÍNDICE DE FIGURAS

Figura 1. Orgánico Estructural de la Unidad de Telecomunicaciones e Información	7
Figura 2. Diagrama del proceso solicitud de incidencia.....	10
Figura 3. Soporte al Servicio.....	12
Figura 4. Provisión del servicio ITIL.....	13
Figura 5. Metodologías de desarrollo de software.....	20
Figura 6. Arquitectura cliente servidor.....	22
Figura 7. Arquitectura del sistema Help Desk.....	23
Figura 8. Estándar ITIL Help Desk.....	27
Figura 9. Estructura física centralizada.....	29
Figura 10. Formato de Historia de Usuario.....	38
Figura 11. Diagrama general de casos de uso.....	51
Figura 12. Asistente de instalación de python.....	56
Figura 13. Comando de verificación python.....	56
Figura 14. Comando de verificación pip.....	57
Figura 15. Comandos de instalación de django.....	57
Figura 16. Asistente de instalación de postgresql.....	57
Figura 17. Asistente de instalación de NodeJS.....	58
Figura 18. Servidor Redis.....	58
Figura 19. Repositorio Github del proyecto Help Desk.....	59
Figura 20. Archivo Settings.py 1 de 3.....	60
Figura 21. Archivo Settings.py 2 de 3.....	61
Figura 22. Archivo Settings.py 3 de 3.....	61
Figura 23. Jerarquía de archivos del código fuente.....	62
Figura 24. Archivo models.py.....	62
Figura 25. Archivo views.py.....	63
Figura 26. Archivo forms.py.....	63
Figura 27. Archivo urls.py.....	64
Figura 28. Archivo crear_centro_asistencia.html.....	64
Figura 29. Configuración Inicial.....	65
Figura 30. Administración de Centros de Asistencia.....	66
Figura 31. Administración de acuerdos de nivel de servicio.....	68
Figura 32. Administración de usuarios.....	69
Figura 33. Solicitud de incidencia. Iteración 1.....	70
Figura 34. Solicitud de incidencia. Iteración 2.....	71

Figura 35. Asignación de Incidencias. Iteración 1.	72
Figura 36. Asignación de Incidencias. Iteración 2.	73
Figura 37. Atender Incidencias.	74
Figura 38. Registro de Equipo.	75
Figura 39. Calidad de servicio. Iteración 1.	76
Figura 40. Calidad de servicio. Iteración 2. 1 de 3.	77
Figura 41. Calidad de servicio. Iteración 2. 2 de 3.	78
Figura 42. Calidad de servicio. Iteración 2. 3 de 3.	78
Figura 43. Acceso al sistema.	83
Figura 44. Mensajes de error.	83
Figura 45. Panel del Usuario Jefe Departamento.....	84
Figura 46. Panel del Usuario Administrador.....	84
Figura 47. Panel del usuario Asesor Técnico.....	84
Figura 48. Panel del Usuario Final.	84
Figura 49. Centro de Asistencia.....	85
Figura 50. Mensajes de error.	86
Figura 51. Actualizar Registro.....	86
Figura 52. Eliminar Registro.....	86
Figura 53. Panel del Centro de Asistencia.	87
Figura 54. Búsqueda de usuarios.	88
Figura 55. Asignar Jefe de Departamento.....	88
Figura 56. Asignar Asesor Técnico.	88
Figura 57. Lista de Acuerdos de Nivel de Servicio.	89
Figura 58. Nuevo SLA.....	90
Figura 59. Actualizar SLA.	90
Figura 60. Eliminar SLA.	90
Figura 61. Diagrama del proceso solicitud de incidencia con el sistema Help Desk.	92
Figura 62. Preguntas de la encuesta.	93
Figura 63. Respuesta de usabilidad del sistema Help Desk.....	94
Figura 64. Respuesta de aceptación al sistema Help Desk.....	94
Figura 65. Efectividad del sistema Help Desk.	95
Figura 66. Utilidad del sistema Help Desk.....	95
Figura 67. Administrar incidencia.	96
Figura 68. Bloque de preguntas al Usuario Final.	97
Figura 69. Complejidad de reporte de incidencia.	97

Figura 70. Complejidad de reporte de bienes.....	98
Figura 71. Calificación de incidencias.	98
Figura 72. Asignación de incidencia.....	99
Figura 73. Preguntas dirigidas al Jefe de Departamento.....	99
Figura 74. Complejidad de asignación de incidencia.....	100
Figura 75. Verificación de la calidad de servicio.....	100
Figura 76. Tiempo de apertura máxima.	100
Figura 77. Atención de incidencia.	101
Figura 78. Preguntas dirigidas al Asesor Técnico.	101
Figura 79. Atención de incidencias asignadas.	102
Figura 80. Registro de acciones de incidencia.....	102
Figura 81. Diagnóstico inicial.	102
Figura 82. Administrar sistema	103
Figura 83. Preguntas dirigidas al administrador.	103
Figura 84. Administrar panel de configuraciones	104
Figura 85. Asignación de usuarios.....	104
Figura 86. Administrar centros de asistencia.	104
Figura 87. Prueba de carga – escenario 1.	105
Figura 88. Gráfico de prueba de carga.	105
Figura 89. Prueba de carga – escenario 2.	106
Figura 90. Gráfico de prueba de carga.	106
Figura 91. Prueba de carga – escenario 3.	107
Figura 92. Gráfico de prueba de carga.	107
Figura 93. Prueba de carga – escenario 1.	108
Figura 94. Gráfico de la prueba de carga.....	109
Figura 95. Prueba de carga escenario 2.	109
Figura 96. Gráfico de la prueba de carga.....	110
Figura 97. Prueba de carga – escenario 3.	110
Figura 98. Gráfico de la prueba de carga.....	111
Figura 99. Certificado de aceptación UTI-1.....	119
Figura 100. Certificado de aceptación UTI-2.....	120
Figura 101. Certificado de traducción.	121
Figura 102. Orgánico estructural de la Universidad Nacional de Loja	122
Figura 103. Entrevista dirigida 1 de 2.....	123
Figura 104. Entrevista dirigida 2 de 2.....	124

Figura 105. Ficha de observación 001.	125
Figura 106. Ficha de observación 002.	126
Figura 107. Ficha de observación 004.	128
Figura 108. Modelo del dominio.	129
Figura 109. Modelo de clases.	130
Figura 110. Modelo de base de datos.	131
Figura 111. Formato de la encuesta aplicada	132

1. TÍTULO

“SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA”.

2. RESUMEN

El presente documento refleja la implementación de un sistema denominado Help Desk, para la automatización del proceso de atención de incidencias del Departamento de Soporte y Mantenimiento de la Universidad Nacional de Loja. El software está basado en una arquitectura web cliente servidor, bajo los principios de la programación orientada a objetos y completamente basado en el patrón de diseño Modelo Vista Controlador.

Es así, que previo a la codificación del sistema, se introdujo en el contexto del proceso de atención de incidencias de la dependencia, para recopilar la información necesaria, aplicando métodos y técnicas de investigación científica.

Luego, para la construcción del sistema, se aplicó las fases planificación, diseño, codificación y pruebas de la metodología de desarrollo de software Extreme Programming (XP) y la utilización de determinados artefactos que ésta metodología propone en cada fase.

Finalmente, en éste documento, se realiza una discusión y socialización de los resultados obtenidos frente a los objetivos planteados y definir con ello premisas a modo de conclusiones y recomendaciones.

2.1. ABSTRACT

This document reflects the implementation of a system called Help Desk to support the service via addressing incidents. It is based on a client-server web-architecture and is in keeping with the principles of object-oriented programming. The implementation of this project intends to automate the process of handling incidences for the National University of Loja's Department of Support and Maintenance.

So, in the following sections of this document, and for the achievement of the project's main objective, the problems which need solving were identified through the collection of information and applied research techniques. Consecutively the phases of the XP software implementation methodology and the technological resources that were used were mentioned.

Prior to the implementation of the system, during the exploration phase, the requirements analysis was defined through the use of certain devices. Additionally, to continue with the development during the iterations phase and to validate the requirements, a Validation and Verification Plan was applied to the Help Desk system.

Finally a discussion and socialization of the obtained results were carried out against the defined objectives, and with them, premises were defined by way of conclusions and recommendations.

3. INTRODUCCIÓN

La Universidad Nacional de Loja, es una institución de educación superior dedicada a la formación académica de profesionales altamente capacitados, debido a que su enseñanza se basa en el aprendizaje científico técnico de calidad humana y, para cumplir con la misión de su planeación estratégica, requiere de departamentos administrativos de apoyo institucional que permiten generar varios servicios prestos a la comunidad universitaria.

Así, la Unidad de Telecomunicaciones e Información, es el departamento administrativo encargado de apoyar los procesos institucionales, además de las actividades académicas de gestión y administración, es decir, provee de todos los aspectos del ámbito informático en el campus universitario; ésta sección está compuesta por las dependencias de Desarrollo de Software y, Redes y Equipos Informáticos.

La dependencia de Redes y Equipos Informáticos, dentro de su jerarquía posee al departamento de Mantenimiento y Soporte Técnico, que entre sus principales funciones están: realizar diseños y rediseños para la implementación de equipo informático; intervenir en la adquisición de dispositivos y dar soluciones óptimas en cuanto a mantenimiento y renovación de equipamiento informático. Sin embargo, es considerable el alcance hasta donde se prestan sus servicios, por lo que es importante para el solicitante o receptor del servicio, recibir un soporte efectivo, rápido y cordial. Uno de los principales servicios, es la Mesa de Ayuda, que conjuga el talento humano y recursos tecnológicos para solucionar iteraciones de incidencias de manera integral, es decir, proporciona a los usuarios respuestas y soluciones de soporte técnico acerca de computadores, equipos electrónicos, o software institucional, éste proceso se realiza mediante vía telefónica y por asignación de turnos, así, de acuerdo al tipo de problema, los usuarios son atendidos por los operadores, quienes receptan el equipo en el área de soporte o incluso en algunas ocasiones se trasladan al lugar del incidente, es aquí, que quienes ejercen esta actividad se ven inmersos en una variedad de problemas, que hasta la actualidad han sido tratadas de forma improvisada aplicando métodos tradicionales para cumplir con la gestión de los servicios que ofrecen.

De la premisa anterior y dando cumplimiento del objetivo principal de éste proyecto que es la implementación del sistema de atención de incidencias Help Desk, lo que se ha realizado es: automatizar el proceso de atención de incidencias para reducir el esfuerzo

y tiempo de respuesta del personal operativo de ésta dependencia; establecer con norma los niveles de prioridad entre incidencias, tomando en cuenta la importancia del problema a solucionar, el riesgo y el plazo de entrega; y, evaluar la calidad de servicio del personal operativo frente a la atención de las incidencias.

Y así, de éste modo, mejorar notablemente el grado de satisfacción de operadores y usuarios.

4. REVISIÓN LITERARIA

4.1. MARCO REFERENCIAL

4.1.1. UNIVERSIDAD NACIONAL DE LOJA.

4.1.1.1. ANTECEDENTES HISTÓRICOS.

La Universidad Nacional de Loja, cumple con una larga trayectoria histórica en logros institucionales y académicos en la República del Ecuador, es así, que su recorrido inicia desde el año de 1859, en donde adscritos al actual Colegio Experimental Bernardo Valdivieso, se empiezan a formar los primeros profesionales en Jurisprudencia, Filosofía y Letras, Teología y Medicina, haciendo efectivos sus grados académicos en las ciudades de Quito, Guayaquil y Cuenca; posteriormente en el año 1985 se crea una facultad, para la otorgación de grados a nivel local. [1]

Luego para el 2 de mayo de 1990, se tomó la decisión de impulsar una reforma académica institucional, que permita propiciar una educación emancipadora, creándose así el SAMOT. Para el 16 de septiembre de 2002, se realiza una reestructuración académico-administrativa (Ver Anexo 4.), dividiendo su accionar institucional en cinco áreas, desde ahí se sigue impulsando la formación académica, investigación, vinculación con la colectividad, etc. Actualmente, con la finalidad de velar por el bienestar de la comunidad universitaria, se presta múltiples servicios, uno de éstos es la Unidad de Telecomunicaciones e Información. [1]

❖ UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

▪ MISIÓN

“Apoyar a los procesos institucionales, así como las actividades académicas, de gestión y administración, mediante el uso de tecnologías de información y recursos humanos capaces y comprometidos, para cumplir con los objetivos de la Universidad Nacional de Loja, contribuyendo a la formación integral de la comunidad universitaria”. [1]

▪ VISIÓN

“Innovar los servicios ofrecidos por la Unidad de Telecomunicaciones e Información para mantener la vanguardia tecnológica en la Universidad Nacional de Loja”. [1]

▪ **OBJETIVOS.**

- Apoyar en la consecución de los objetivos estratégicos de la Universidad Nacional de Loja.
- Ofrecer un servicio de calidad como principales proveedores de información y automatización de funciones administrativas y académicas.
- Mantener servicios de información integrados y consistentes para la comunidad universitaria.
- Optimizar el uso y capacidades de la tecnología de la universidad en procura de mantener una relación costo-beneficio ventajosa para la institución.
- Mantener una participación activa en la investigación y descubrimientos tecnológicos en beneficio de la institución y la colectividad.
- Proyectar a la universidad como una institución líder en avances tecnológicos constituyéndose en un aporte constante en la búsqueda de soluciones a las necesidades de información y tecnología de la sociedad.
- Proporcionar apoyo y asesoría informática a todos los usuarios que operen programas o proyectos institucionales.
- Capacitar a la planta docente y administrativa para la utilización correcta y eficiente del software académico institucional. [1]

❖ **ORGANIZACIÓN DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN.**

▪ **ORGÁNICO ESTRUCTURAL**

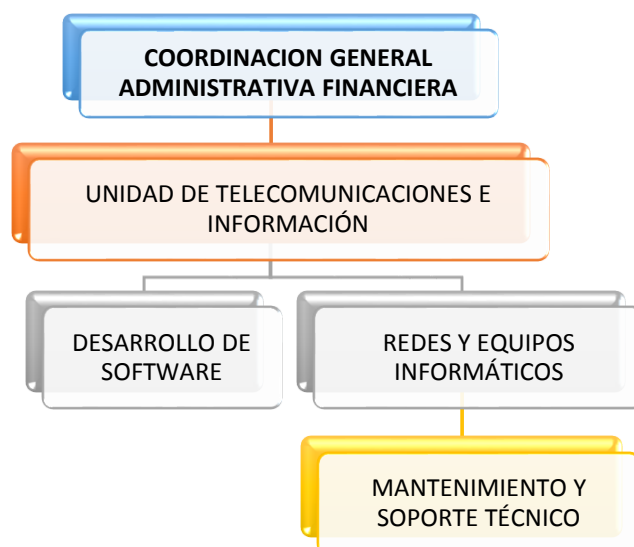


Figura 1. Orgánico Estructural de la Unidad de Telecomunicaciones e Información. [1]

▪ **ORGÁNICO FUNCIONAL**

A continuación, se detalla las funciones de cada área operativa de la Unidad de Telecomunicaciones e Información.

FUNCION	
UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN	<ul style="list-style-type: none"> - Tomar decisiones respecto a los diferentes proyectos considerando la relación costo-beneficio. - Asignar las tareas al personal requerido para la elaboración de un determinado proyecto. - Decisión final en la adquisición de hardware y software. - Planificar, dirigir y controlar las actividades en el ámbito informático. - Planificar, dirigir y controlar los Proyectos de Investigación y Desarrollo en el área de informática. - Determinar y administrar los recursos humanos, técnicos y económicos para la implementación, equipamiento y desarrollo de laboratorios, en coordinación con las correspondientes dependencias. - Planificar y desarrollar políticas, planes, programas, proyectos y estrategias en el área de Informática para el desarrollo del proceso de enseñanza aprendizaje. - Emitir informes de carácter académico y/o administrativo, solicitados por la Administración Central, directores de área o por el Honorable Consejo Académico. - Coordinar y supervisar la ejecución oportuna y completa de los servicios de mantenimiento preventivo y correctivo a los bienes informáticos, equipos de cómputo y de telecomunicaciones, de conformidad con los contratos y/o convenios correspondientes.
DESARROLLO DE SOFTWARE	<ul style="list-style-type: none"> - Analizar, diseñar, adaptar, desarrollar e implantar sistemas automatizados de información que apoyen, tanto las actividades operativas, como la toma de decisiones de la comunidad universitaria e instancias externas. - Realizar el análisis de factibilidad de los sistemas automatizados de información que sean solicitados. - Construir soluciones integrales (aplicaciones) a las necesidades de la institución educativa.

	<ul style="list-style-type: none"> - Analizar, diseñar e implantar páginas o sitios en Internet, que se alojan en el portal de la Universidad, para ofrecer información a la comunidad. - Dar soporte a los sistemas automatizados de información de la UNL.
REDES Y EQUIPOS INFORMÁTICOS	<ul style="list-style-type: none"> - Agregar y configurar nuevas estaciones de trabajo (puntos de red). - Instalar y configurar software y equipo de redes de computadoras y de telecomunicaciones. - Planear el crecimiento y actualización de la infraestructura de red y telecomunicaciones. - Supervisar la operación del equipo de suministro de energía eléctrica, sistemas de respaldo de energía y tierras físicas en las diferentes instalaciones. - Monitorear violaciones de seguridad a la red y tomar acciones correctivas. - Crear claves de acceso para la red y configurar tipos de privilegios de accesos para los usuarios. - Participar en el diseño e instalación de nuevas redes y revisar diariamente el buen funcionamiento de la red. - Recomendar la adquisición de nuevos equipos de red en caso de ser necesarios. - Desarrollar las demás funciones inherentes al área de su competencia.

Tabla 1. Orgánico Funcional de la Unidad de Telecomunicaciones e Información. [1]

▪ **DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO.**

El Departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja, es una sección adherida al Departamento de Redes y Equipos Informáticos cuya principal función es mantener la continuidad en la prestación de servicios del campus universitario en el ámbito informático. Es decir, se encarga del mantenimiento preventivo y correctivo de equipos informáticos que presentan algún desperfecto y que son reportados.

Actualmente, ésta dependencia no dispone documentación de ninguna técnica de diagramación para reflejar el curso de sus procesos. Sin embargo, y para una mejor comprensión del servicio de atención de incidencias, éste se ha reflejado en el diagrama de proceso de la Figura 2, cuyo fundamento está basado en las técnicas de investigación científica como la observación directa y la entrevista del Anexo 5 y 6.

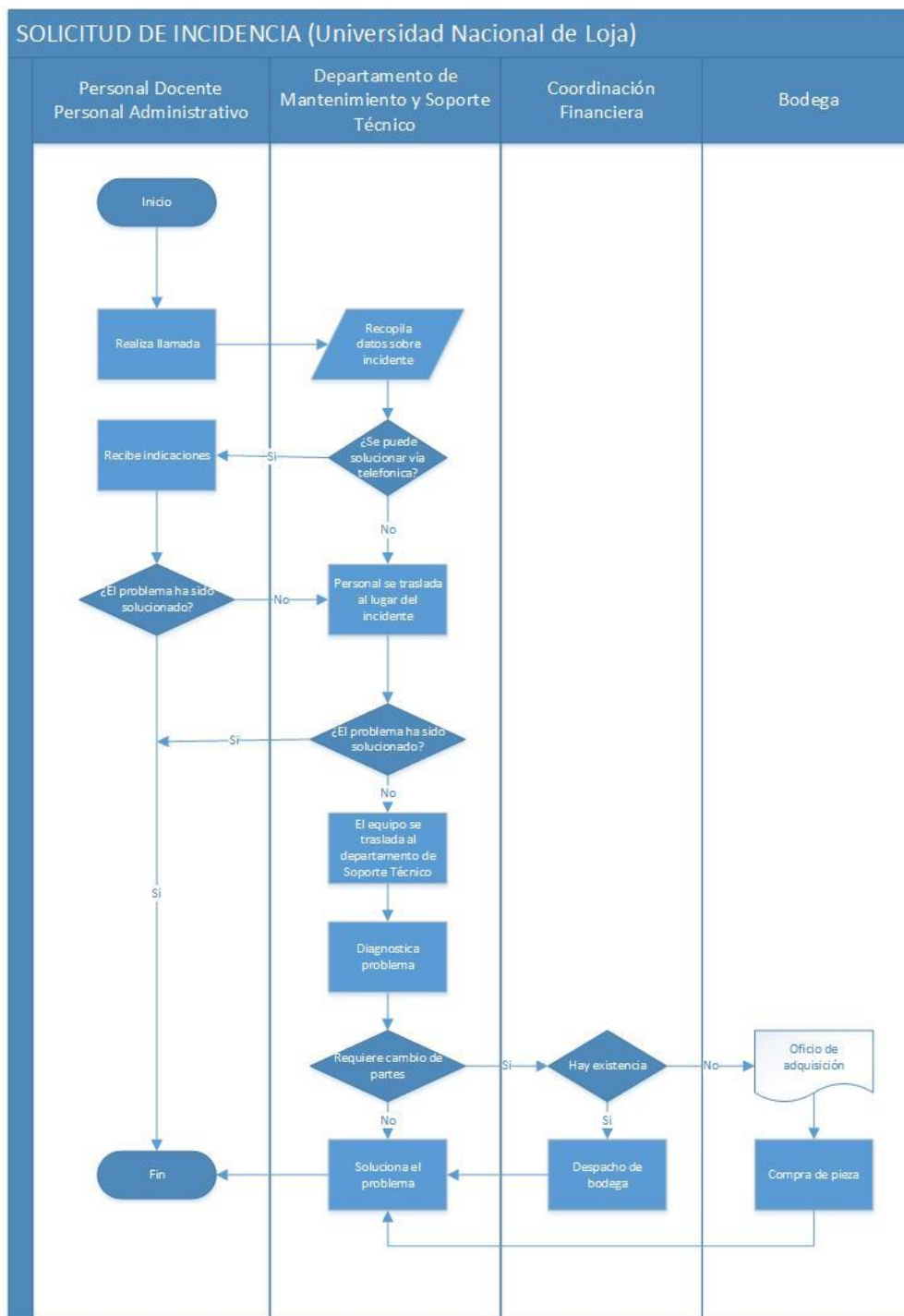


Figura 2. Diagrama del proceso solicitud de incidencia.

4.1.2. HELP DESK

La mesa de ayuda o help desk es un conjunto de servicios destinados a la gestión y solución de todas las posibles incidencias relacionadas con las tecnologías de la información y comunicación.

“La mesa de ayuda ofrece servicios acerca de soporte técnico en la detección de bugs o fallas en el software y hardware. Se basa en un conjunto de recursos tecnológicos y humanos que brindan soporte técnico a los usuarios del área informática de una empresa para incrementar la productividad y la satisfacción de los usuarios internos o externos.” [2]

La mesa de ayuda administra peticiones de los usuarios por software para dar seguimiento a las solicitudes presentadas, a esto se define como “Seguimiento local de fallos”. [2]

Cuando el usuario notifica un problema, la mesa de ayuda le asigna un ticket donde están especificados los detalles del problema. Entre los principales servicios que presta una mesa de ayuda están: atención inmediata a los incidentes, seguimiento y progreso en cada solicitud de servicio.

El analista destinado para la atención debe contar con la habilidad conocimiento y capacidad de brindar atención eficiente al usuario. Además, debe identificar las posibles consecuencias de cada acción para obtener la satisfacción del usuario, que es el objetivo principal de la mesa de ayuda. [2]

4.1.3. ITIL

Desarrollada a finales de 1980, la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL®), se ha convertido en el estándar mundial de de facto en la Gestión de Servicios Informáticos. Iniciado como una guía para el gobierno de UK, la estructura base ha demostrado ser útil para las organizaciones en todos los sectores a través de su adopción por innumerables compañías como base para consulta, educación y soporte de herramientas de software. Hoy, ITIL es conocido y utilizado mundialmente. Pertenece a la Oficina de Comercio del Gobierno Británico, pero es de libre utilización. [3]

ITIL, fue desarrollada al reconocer que las organizaciones dependen cada vez más de la Informática para alcanzar sus objetivos corporativos. Esta dependencia ha dado como resultado una necesidad creciente de servicios informáticos de calidad que se correspondan con los objetivos del negocio, y que satisfagan los requisitos y las expectativas del cliente. A través de los años, el énfasis pasó de estar sobre el desarrollo de las aplicaciones TI a la gestión de servicios TI. La aplicación TI (a veces nombrada

como un sistema de información) sólo contribuye a realizar los objetivos corporativos si el sistema está a disposición de los usuarios y, en caso de fallos o modificaciones necesarias, es soportado por los procesos de mantenimiento y operaciones. [3]

A lo largo de todo el ciclo de los productos TI, la fase de operaciones alcanza cerca del 70-80% del total del tiempo y del coste, y el resto se invierte en el desarrollo del producto (u obtención). De esta manera, los procesos eficaces y eficientes de la Gestión de Servicios TI se convierten en esenciales para el éxito de los departamentos de TI. Esto se aplica a cualquier tipo de organización, grande o pequeña, pública o privada, con servicios TI centralizados o descentralizados, con servicios TI internos o suministrados por terceros. En todos los casos, el servicio debe ser fiable, consistente, de alta calidad, y de coste aceptable. [3]

4.1.3.1. FUNDAMENTOS DE LA GESTIÓN TI

❖ SOPORTE AL SERVICIO

El soporte al servicio se preocupa de todos los aspectos que garanticen la continuidad, disponibilidad y calidad del servicio prestado al usuario. La Figura 3 resume sucintamente los principales aspectos de la metodología de soporte al servicio según los estándares ITIL.

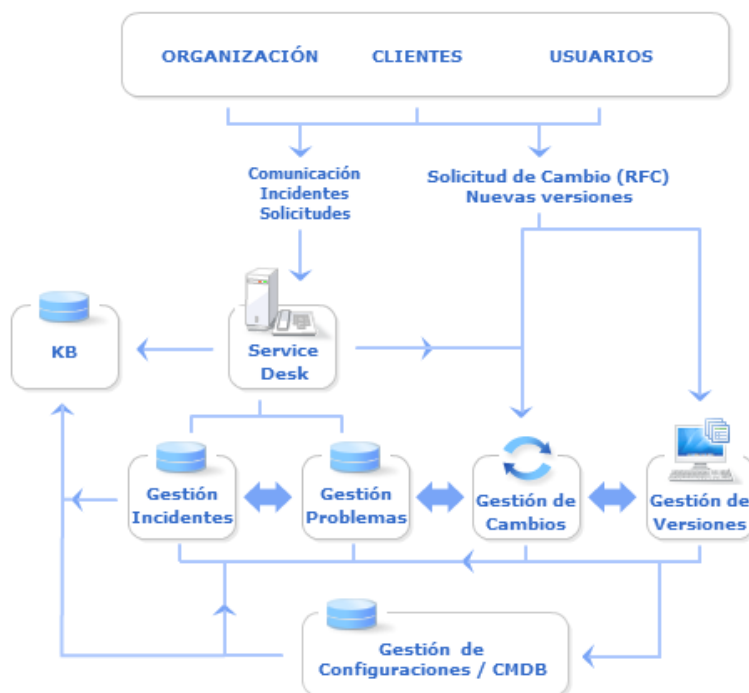


Figura 3. Soporte al Servicio. [3]

❖ PROVISIÓN DEL SERVICIO

La provisión del servicio se ocupa de los servicios ofrecidos en sí mismos. En particular de los Niveles de servicio, su disponibilidad, su continuidad, su viabilidad financiera, la capacidad necesaria de la infraestructura TI y los niveles de seguridad requeridos. [3]

En la Figura 4 se resume sucintamente los principales aspectos de la metodología de provisión del servicio según los estándares ITIL:



Figura 4. Provisión del servicio ITIL.

4.1.3.2. CENTRO DE SERVICIOS

El objetivo primordial, aunque no único, del Centro de Servicios es servir de punto de contacto entre los usuarios y la Gestión de Servicios TI. Un Centro de Servicios, en su concepción más moderna, debe funcionar como centro neurálgico de todos los procesos de soporte al servicio: registrando y monitorizando incidentes; aplicando soluciones temporales a errores conocidos en colaboración con la Gestión de Problemas; Colaborando con la Gestión de Configuraciones para asegurar la actualización de las bases de datos correspondientes; gestionando cambios solicitados por los clientes mediante peticiones de servicio en colaboración con la Gestión de Cambios y Versiones

4.1.3.3. GESTIÓN DE INCIDENTES

La Gestión de Incidentes tiene como objetivo resolver cualquier incidente que cause una interrupción en el servicio de la manera más rápida y eficaz posible. La Gestión de Incidentes no debe confundirse con la Gestión de Problemas, pues a diferencia de ésta última, no se preocupa de encontrar y analizar las causas subyacentes a un determinado

incidente sino exclusivamente a restaurar el servicio. Sin embargo, es obvio, que existe una fuerte interrelación entre ambas. [3]

4.1.3.4. GESTIÓN DE PROBLEMAS

Las funciones principales de la Gestión de Problemas son:

- Investigar las causas subyacentes a toda alteración, real o potencial, del servicio TI.
- Determinar posibles soluciones a las mismas.
- Proponer las peticiones de cambio (RFC) necesarias para restablecer la calidad del servicio.
- Realizar Revisiones Post Implementación (PIR) para asegurar que los cambios han surtido los efectos buscados sin crear problemas de carácter secundario.

La Gestión de Problemas puede ser:

- **Reactiva:** Analiza los incidentes ocurridos para descubrir su causa y propone soluciones a los mismos.
- **Proactiva:** Monitoriza la calidad de la infraestructura TI y analiza su configuración con el objetivo de prevenir incidentes incluso antes de que estos ocurran. [3]

4.1.4. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO DE SOFTWARE

4.1.4.1. ICONIX

Iconix es una metodología pesada-ligera de desarrollo del software que se halla a medio camino entre un RUP (Rational Unified Process) y un XP (Extreme Programming).

“Iconix deriva directamente del RUP y su fundamento es el hecho de que un 80% de los casos pueden ser resueltos tan solo con un uso del 20% del UML, con lo cual se simplifica muchísimo el proceso sin perder documentación al dejar solo aquello que es necesario. Esto implica un uso dinámico del UML de tal forma que siempre se pueden utilizar otros diagramas además de los ya estipulados si se cree conveniente.” [4]

“Iconix se guía a través de casos de uso y sigue un ciclo de vida iterativo e incremental. El objetivo es que a partir de los casos de uso se obtenga el sistema final.” [5]

Las tres características fundamentales de ICONIX son:

Iterativo e incremental: Varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos. [6]

Trazabilidad: Cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos. [6]

Dinámica del UML: La metodología ofrece un uso “dinámico del UML” como los diagramas del caso de uso, diagramas de secuencia y de colaboración. [6]

❖ CICLO DE VIDA

García, López y Amavizca en su paper Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software, definen claramente las fases que se mencionan a continuación. [5]

- **ANÁLISIS DE REQUISITOS.** En esta primera fase se realiza un modelo de dominio, que no es más que un diagrama de clases extremadamente simplificado. Este modelo contiene únicamente aquellos objetos de la vida real cuyo comportamiento o datos deban ser almacenados en el sistema. a partir de este pequeño modelo, se realiza un pequeño prototipo basándose en la storyboard de la interfaz gráfica obtenida previamente, el cual se mostrará al cliente y se refinará en sucesivas reuniones. Normalmente este prototipo suele converger en dos o tres iteraciones. Una vez el prototipo ya es final y se han obtenido todos los requisitos del sistema por parte del cliente, se procede a realizar los casos de uso. Estos diagramas de casos de uso se agrupan en diagramas de paquetes y se asocia cada requisito a un caso de uso para obtener la ya mencionada anteriormente trazabilidad.
- **ANÁLISIS Y DISEÑO PRELIMINAR.** A partir de cada caso de uso se obtienen sus correspondientes fichas de caso de uso. Cabe destacar que estas fichas no pertenecen al UML. Después será necesario realizar lo que se conoce como diagrama de robustez, el cual pertenece al proceso Iconix y tampoco forma parte del UML.
- **DISEÑO.** En esta fase se proceden a realizar los diagramas de secuencia, los cuales derivan directamente de las fichas de caso de uso. Esto implica que una vez finalizado el diseño, tras refinar nuevamente el diagrama de clases, se puede verificar directamente gracias a este factor de trazabilidad, y prepararse para la siguiente fase.

- **IMPLEMENTACIÓN.** Es cuestión de crear un buen software a partir del diseño, y mediante los testeos y pruebas adecuados poder garantizar que el sistema final cumple con los requisitos iniciales y por tanto proceder a su entrega.

4.1.4.2. RUP

Es una metodología cuyo fin es entregar un producto de software. Se estructura todos los procesos y se mide la eficiencia de la organización. Es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. [7]

El RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Describe cómo aplicar enfoques para el desarrollo del software, llevando a cabo unos pasos para su realización. [8]

Se centra en la producción y mantenimiento de modelos del sistema. Entre las principales características están:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software. [7]

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

❖ CICLO DE VIDA

El ciclo de vida de un proyecto elaborado con RUP, consta de las fases de inicio, elaboración, desarrollo y cierre, así como lo menciona Anay en el proyecto “Multimedia de apoyo a la enseñanza de la metodología RUP” [7]

- **FASE DE INICIO.** Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
- **FASE DE ELABORACIÓN.** En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
- **FASE DE DESARROLLO.** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
- **FASE DE CIERRE.** El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

4.1.4.3. EXTREME PROGRAMMING (XP)

La programación extrema o Extreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia “Extreme Programming Explained: Embrace Change” (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. [9]

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [9]

❖ **PRACTICAS BASICAS DE LA PROGRAMACION EXTREMA.**

La programación extrema se basa en doce "prácticas básicas" que deben seguirse.

Equipo completo: Forman parte del equipo todas las personas que tienen algo que ver con el proyecto, incluido el cliente y el responsable del proyecto.

Planificación: Se hacen las historias de usuario y se planifica en qué orden se van a hacer y las mini-versiones. La planificación se revisa continuamente.

Test del cliente: El cliente, con la ayuda de los desarrolladores, propone sus propias pruebas para validar las mini-versiones.

Versiones pequeñas: Las mini-versiones deben ser lo suficientemente pequeñas como para poder hacer una cada pocas semanas. Deben ser versiones que ofrezcan algo útil al usuario final y no trozos de código que no pueda ver funcionando.

Diseño simple: Hacer siempre lo mínimo imprescindible de la forma más sencilla posible. Mantener siempre sencillo el código.

Pareja de programadores: Los programadores trabajan por parejas (dos delante del mismo ordenador) y se intercambian las parejas con frecuencia (un cambio diario).

Integración continua: Deben tenerse siempre un ejecutable del proyecto que funcione y en cuanto se tenga una nueva pequeña funcionalidad, debe recompilarse y probarse. Es un error mantener una versión congelada dos meses mientras se hacen mejoras y luego integrarlas todas de golpe.

El código es de todos: Cualquiera puede y debe tocar y conocer cualquier parte del código. Para eso se hacen las pruebas automáticas.

Normas de codificación: Debe haber un estilo común de codificación (no importa cuál), de forma que parezca que ha sido realizado por una única persona.

Metáforas: Hay que buscar unas frases o nombres que definan cómo funcionan las distintas partes del programa, de forma que sólo con los nombres se pueda uno hacer una idea de qué es lo que hace cada parte del programa.

Ritmo sostenible: Se debe trabajar a un ritmo que se pueda mantener indefinidamente. Esto quiere decir que no debe haber días muertos en que no se sabe qué hacer y que no se deben hacer un exceso de horas otros días.

❖ **CICLO DE VIDA**

Como resultado del estudio del “Caso práctico de la metodología ágil XP al desarrollo de software” elaborado por Echeverry y Delgado, mencionan que El ciclo de vida ideal de XP se compone de las siguientes fases [9]:

- **FASE DE PLANEACIÓN.** En ésta fase inicial, se interactúa con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. En éste punto se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto. Para dar cumplimiento a la fase de planeación se debe tener en cuenta los siguientes elementos: historias de usuario, velocidad del proyecto, plan de iteraciones y roles de XP.
- **FASE DE DISEÑO.** En ésta fase solo se diseñan las historias de usuario que el usuario ha seleccionado para la iteración actual, debido a que: se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio; y, puesto que dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo. Los elementos que se utilizan en ésta fase son: metáfora del sistema, modelos conceptuales y tarjetas CRC.
- **FASE DE CODIFICACION.** En ésta fase se realiza la codificación de los algoritmos en el lenguaje de programación seleccionado. Aquí se detalla la ejecución del plan de iteraciones.
- **FASE DE PRUEBAS.** En ésta fase, XP enfatiza mucho los aspectos relacionados con las pruebas, clasificándolas en diferentes tipos y funcionalidades específicas,

indicando quién, cuándo y cómo deben ser implementadas y ejecutadas. El elemento principal de aceptación en ésta fase es el plan de validación y verificación.

4.1.4.4. RESUMEN COMPARTIVO DE METODOLOGIAS DE DESARROLLO DE SOFTWARE

	Características	Mecanismos	Ventajas y Desventajas
RUP	<ul style="list-style-type: none"> Modelo estándar para el análisis, implementación y documentación de sistemas orientados a objetos. Maneja casos de uso. Arquitectura céntrica. Iterativo e incremental. Manejo de requerimientos, desarrollo basado en componentes, modelo visual utilizando UML, verificación continua de la calidad. Ejecuta la planificación, requerimientos, análisis & diseño, implementación, pruebas, despliegue y evaluación en las fases: concepción, elaboración, construcción y transición. 	<ul style="list-style-type: none"> Casos de uso. Planificación de Pruebas. Gestión de Cambios. Estuerzo - Horario contra fases del RUP 	<p>VENTAJAS</p> <ul style="list-style-type: none"> Evaluación en cada fase que permite cambios de objetivos. Proporcionan un lenguaje común (UML). Seguimiento detallado de cada fase. Es modelado por los casos de uso. Mayor documentación <p>DESVENTAJAS</p> <ul style="list-style-type: none"> Proceso pesado, debido a la extensa documentación. La documentación se ve afectada por los posibles cambios volátiles de funcionalidad. No es adecuado para proyectos pequeños. Ocupa mucho tiempo en la fase de desarrollo. Realización de documentación implica gastos en recursos.
XP	<ul style="list-style-type: none"> Enfatiza las comunicaciones cara a cara en lugar de la documentación. Solicita la colaboración con el cliente en lugar de contratos. Desarrolla software funcional. Facilidad de respuesta frente al cambio. Aplica la reutilización de código desarrollado. Promueve el trabajo en equipo. Entrega de software en el menor tiempo posible. Aplica las fases: exploración, release, iteraciones, producción, mantenimiento y muerte del proyecto. 	<ul style="list-style-type: none"> Historias de Usuario. Release planning iteraciones. Programación en Parejas 	<p>VENTAJAS</p> <ul style="list-style-type: none"> Mayor satisfacción del cliente, cumplimiento de plazos. El cliente o el equipo de desarrollo tiene el control sobre las prioridades. Ejecución continua de pruebas. Utilizada para requerimientos que cambian rápidamente. Programación Organizada Puede ser aplicada a cualquier lenguaje de programación. <p>DESVENTAJAS</p> <ul style="list-style-type: none"> Recomendable para proyectos a corto plazo. Altas comisiones en caso de fallar.
ICONIX	<ul style="list-style-type: none"> Ágil y robusto para proyectos medianos. Utilizado por equipos de trabajo mediano. Modelo iterativo incremental (desarrollo del modelo del dominio e identificación de casos de uso). Dinámica de UML (casos de uso, diagramas de secuencia y colaboración). Trazabilidad. Consta de 4 fases: análisis de requerimientos, análisis y diseño preliminar, diseño detallado, e implementación. 	<ul style="list-style-type: none"> Análisis de Requerimientos Modelo de Dominio Casos de Uso de Diagramas de Secuencia, Robustez y Colaboración. 	<p>VENTAJAS</p> <ul style="list-style-type: none"> Es un modelo pequeño y firme que no desecha el análisis y el diseño. Usa un análisis de robustez para reducir ambigüedad en los casos. Proporciona suficientes requisitos y documentación de diseño, pero sin parar el análisis. Es refinado y actualizado a lo largo del proyecto, por lo que siempre refleja la actual comprensión del problema de espacio. <p>DESVENTAJAS</p> <ul style="list-style-type: none"> No puede ser usado para proyectos grandes. Necesita información rápida y puntual de los requisitos, el diseño y las estimaciones. Se debe conocer los diagramas UML.

Figura 5. Metodologías de desarrollo de software.

4.1.5. PLAN DE VERIFICACIÓN Y VALIDACIÓN

Un plan de verificación y validación, consiste en un conjunto de procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes.

Los objetivos de las actividades de verificación y validación son valorar y mejorar la calidad de los productos del trabajo generados durante el desarrollo y modificación del software. Se corrige todo posible fallo para alcanzar cierto grado de perfección, así mismo, se debe garantizar la consistencia, confiabilidad, utilidad, eficacia y el apego a los estándares del desarrollo de software. [10]

Para ello es necesario encontrar defectos en el sistema y asegurar el sistema es útil para el entorno de trabajo requerido.

4.1.5.1. VERIFICACIÓN Y VALIDACIÓN

Ambos conceptos suelen tratarse como sinónimos, sin embargo, se refieren a cosas completamente distintas:

- La verificación se enfoca más al proceso de evaluación del sistema o de los componentes, permite determinar si los productos de una determinada fase del desarrollo satisfacen las condiciones impuestas en el inicio de la misma.
- La validación también es una evaluación del sistema o componentes, pero solo se efectúa en el transcurso o al final del proceso del desarrollo para determinar si cumple con lo especificado.

Es importante resaltar que nunca se va a poder demostrar que el software está completamente libre de defectos, la verificación y validación más crítica es realizada por los clientes finales.

4.2. TECNOLOGÍAS DE DESARROLLO

4.2.1. ARQUITECTURA DEL SISTEMA HELP DESK

Del axioma de Javier Acuña, donde expresa el concepto de arquitectura de software, se define que: “La arquitectura de software de un programa o de un sistema computacional está definida por la estructura, comprendida por los elementos de software, las propiedades visibles de esos elementos y las relaciones entre ellos” [11], a continuación, se presenta la arquitectura del sistema Help Desk para la Sección de Mantenimiento de la Universidad Nacional de Loja en dos tipos de estilos arquitectónicos, con la finalidad de expresar un esquema de organización estructural del software desarrollado.

4.2.1.1. ARQUITECTURA CLIENTE SERVIDOR

El sistema Help Desk cuenta con una arquitectura cliente servidor, que consiste en una aplicación web con clientes livianos (light clients). En éste tipo de arquitectura se identifican claramente dos posiciones; uno es el **cliente**, donde se encuentra el usuario final que accede a la aplicación por medio de un navegador web, y por otro lado está el **servidor** que es donde realmente están los datos, reglas y lógica de la aplicación. (Ver Figura 6.)

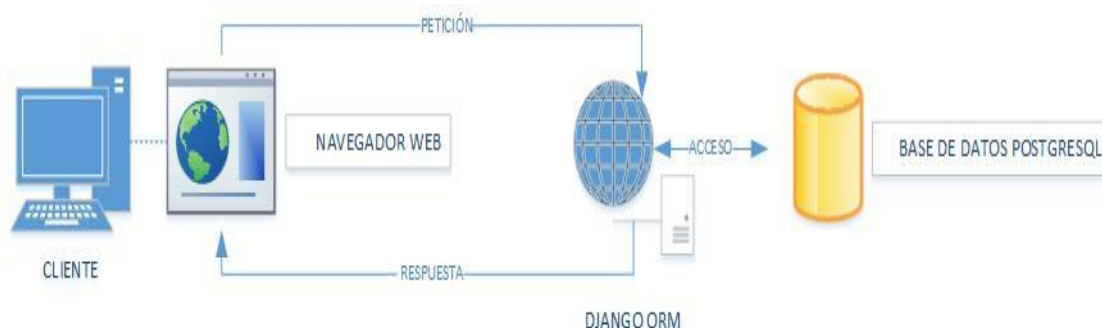


Figura 6. Arquitectura cliente servidor.

4.2.1.2. ARQUITECTURA DEL SOFTWARE

La arquitectura del sistema, provee un grupo de componentes y cada una un conjunto de servicios y, al utilizar el framework Django para la codificación del sistema, los módulos de la aplicación Help Desk se adhieren a la arquitectura definida para Django.

Como se puede observar, en la Figura 7, los componentes del sistema Help Desk son:

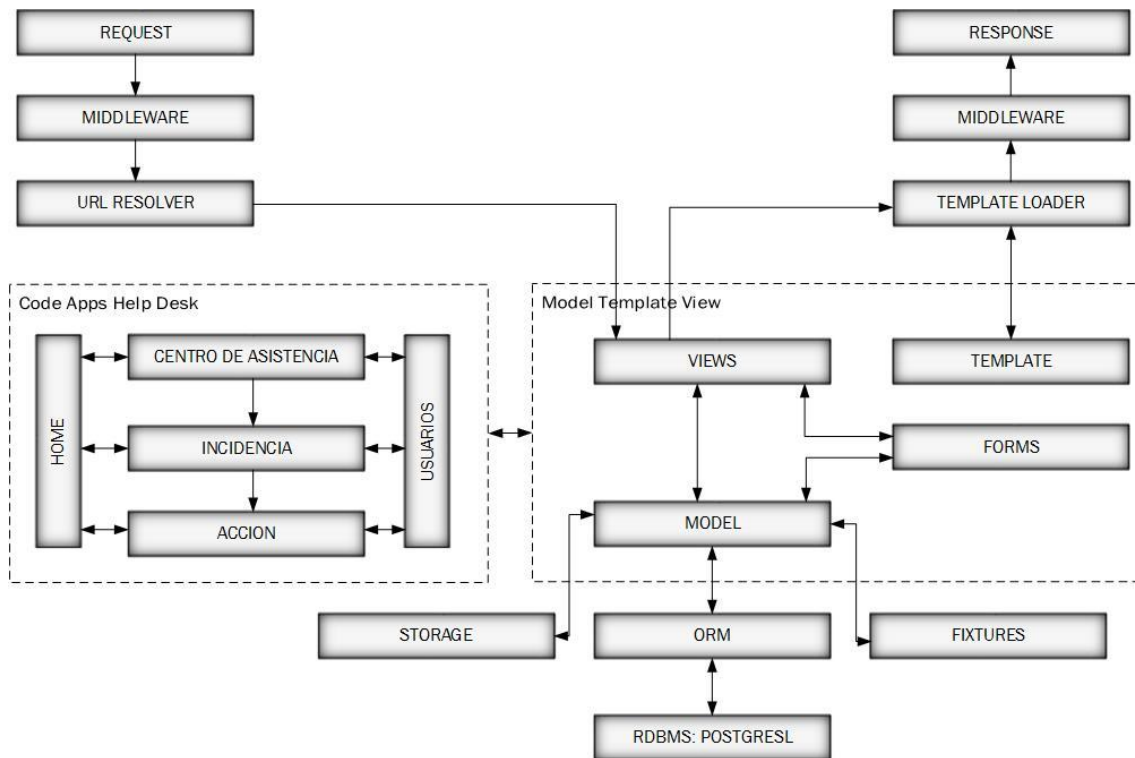


Figura 7. Arquitectura del sistema Help Desk [12].

HOME: Es la aplicación encargada de proveer las funciones generales del sistema, debido a sus procesadores de contexto y configuraciones, siempre está en contacto frecuente con la aplicación de centro de asistencia, incidencia y acción.

CENTRO DE ASISTENCIA: Esta aplicación provee el entorno neurálgico del sistema donde, se determina los usuarios operadores, acuerdos de nivel de servicios, y se reciben las solicitudes de incidencia.

INCIDENCIA: La aplicación administra todo lo referente a la generación de incidencias previas a su atención, como la asignación de asesores técnicos, algoritmos de determinación de prioridad, estados de incidencia, entre otras.

ACCIÓN: En ésta aplicación, se generan las acciones sobre las incidencias, es decir, las funcionalidades para solicitar recursos, diagnosticar equipos y solucionar incidencias.

USUARIOS: Ésta aplicación, al igual que home, interactúa con todas las aplicaciones del sistema Help Desk, debido a que provee las funcionalidades para la autenticación y autorización de usuarios.

4.2.2. HERRAMIENTAS Y TECNOLOGÍA DEL SISTEMA HELP DESK

4.2.2.1. LENGUAJE DE PROGRAMACIÓN PYTHON

De acuerdo a las normativas que propone el departamento de Desarrollo de Software en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, la aplicación web se desarrolló bajo el paradigma de programación Orientado a Objetos en el lenguaje de programación y de código abierto Python en la versión 2.7; y, para facilitar el desarrollo se utilizó el framework Django (versión 1.8) que permitió la construcción rápida de la aplicación web y con menos código.

Django es conocido como un Framework Model Template View en el patrón de diseño MTV.

M significa "Model" (Modelo), la **capa de acceso a la base de datos**. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.

T significa "Template" (Plantilla), la **capa de presentación**. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.

V significa "View" (Vista), la **capa de la lógica de negocios**. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: se puede pensar en esto como un puente entre los modelos y las plantillas. [13]

❖ DJANGO-DBBACKUP

Es una aplicación desarrollada con la colaboración de Django, que facilitó los comandos de gestión para ayudar a respaldar y restaurar la base de datos del sistema Help Desk.

4.2.2.2. BASE DE DATOS POSTGRESQL

En cuanto a la capa de datos, que es en donde reside la información, se utilizó el gestor de base de datos POSTGRESQL 9.3.5, que permitió el almacenamiento de datos, solicitudes de almacenamiento, o recuperación de la información.

4.2.2.3. INTERFAZ GRÁFICA

Para potenciar el funcionamiento y mejorar la apariencia de la interfaz gráfica de usuario, la aplicación contó con la integración de las siguientes herramientas tecnológicas de software:

❖ CSS3 (CASCADING STYLE SHEETS)

Css3, son hojas de estilo en cascada, permiten definir de manera eficiente las páginas web, ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web. [14]

❖ JQUERY.

Es una biblioteca de JavaScript, en la versión 1.11.2., que permitió simplificar la manera de interactuar con los documentos Hipertext Markup Language (HTML), manipular el árbol Document Object Model (DOM), manejar eventos, desarrollar animaciones y agregar interacción con la técnica Asynchronous JavaScript And XML (AJAX) a las páginas web. [14]

❖ HTML5 (HYPERTEXT MARKUP LANGUAGE)

Es el lenguaje básico de la World Wide Web, HTML versión 5 usa un lenguaje de etiquetas para construir páginas web. Además, éste estándar también funciona correctamente con smartphones y tablets. De este modo, también es posible mejorar la velocidad y visualización de las páginas webs en dispositivos móviles. [14]

❖ BOOTSTRAP

Es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. [14]

❖ CHART.JS

Librería de Javascript para generar gráficos estadísticos. Posee un potente paquete de graficación, como diagramas de línea, diagrama de barras, radar, área polar, entre otras, con soporte para varios navegadores web. [15]

4.2.2.4. NPM

Es la herramienta necesaria en cuando a manejo de dependencias para Node Js. [16]

❖ NODE JS

Node.js está basado en el motor V8 de Javascript de Google. Este motor está diseñado para correr en un navegador y ejecutar el código de Javascript de una forma extremadamente rápida. Node.js delega todo el trabajo en un pool de threads. Este pool de threads está construido con la librería libuv. Esta librería dispone de su propio entorno multithread asíncrono. Node.js envía el trabajo que hay que realizar al pool. [16]

4.2.2.5. REDIS

“Es un motor de base de datos libre de tipo clave-valor (key-value) persistentes que residen en memoria ram y posteriormente vuelca el conjunto de datos almacenados al disco duro. Redis es cliente/servidor por lo que levanta su servicio y responde peticiones, cuenta con interfaz de red lo cual hace posible conectar clientes o nodos desde otros host.” [17]

4.2.2.6. ISHOUT.JS

iShout.js es un servidor socket.io + Node.js, utilizado para añadir notificaciones en tiempo real en aplicaciones. A éste proyecto se suma una fácil integración iShout.js de Django, lo que le permite añadir fácilmente notificaciones en tiempo real fuera de las aplicaciones de Django. [18]

4.2.2.7. GITHUB

Servicio para el alojamiento de repositorios de Software gestionado por el sistema de control de versiones Git.

4.2.3. METODOLOGÍA ITIL

Basado en el fundamento que ITIL propone y de acuerdo al análisis de requerimientos para el Departamento de Soporte Técnico de la Universidad Nacional de Loja, el Centro de Soporte Help Desk se desarrolló bajo la siguiente propuesta.

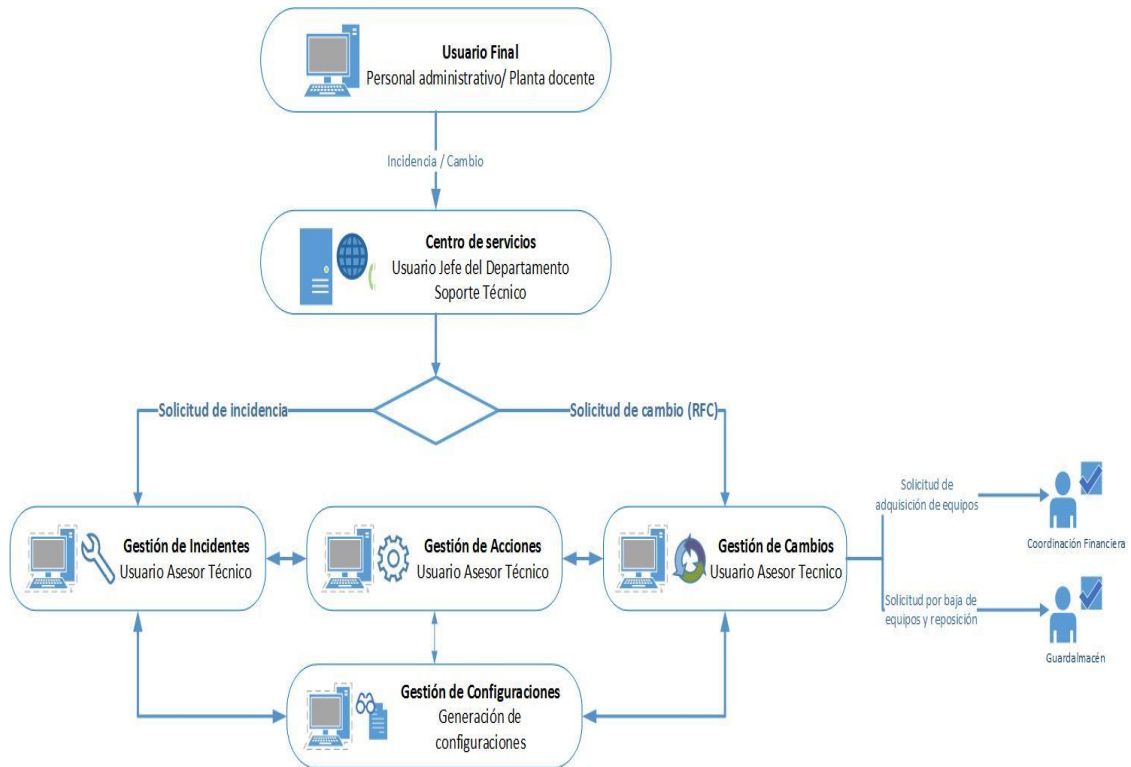


Figura 8. Estándar ITIL Help Desk.

4.2.3.1. DESCRIPCIÓN DE COMPONENTES UTILIZADOS

El sistema Help Desk aplicó el estándar ITIL, resultando el modelo de la Figura 8. Seguidamente, se describe cada uno de los componentes utilizados.

❖ USUARIO FINAL

Este componente representa en el sistema al usuario final, es decir al personal administrativo y planta docente de la Universidad Nacional de Loja que requiera solicitar servicio técnico.

❖ CENTRO DE SERVICIOS

Simula la acción de un conmutador, el usuario Jefe del Departamento Soporte Técnico está en la capacidad de: asignar las solicitudes de incidencia al personal colaborador de soporte técnico; proveer al usuario asesor técnico todas las facilidades para la resolución de incidencias; y, asignar acuerdos de nivel de servicio para la resolución de incidentes.

❖ **GESTIÓN DE INCIDENTES**

El personal técnico brinda solución al incidente generado por el usuario final, con la finalidad de restaurar el servicio con la mayor prontitud, acorde al acuerdo de nivel de servicio establecido.

❖ **GESTIÓN DE ACCIONES**

En el sistema, le permite al usuario Asesor Técnico registrar todas las acciones realizadas sobre la atención del incidente.

❖ **GESTIÓN DE CAMBIOS**

Tiene como objetivo tramitar la solicitud de compra o cambio de piezas y/o partes mediante procedimientos establecidos a las unidades departamentales correspondientes.

❖ **GESTIÓN DE CONFIGURACIONES**

Contiene una base de configuraciones y acciones que se pueden ejecutar en los diferentes tipos de gestión.

4.2.3.2. ESTRUCTURA FÍSICA

ITIL propone tres formatos básicos para determinar la estructura física del Centro de Soporte: Centralizado, Distribuido y Virtual. La estructura física aplicada en éste proyecto es la **ESTRUCTURA CENTRALIZADA**, por las siguientes razones:

El contacto con todos los usuarios finales se canaliza a través de una sola estructura central.

Proporciona ventajas como: reducción de costes, optimización de recursos, y simplicidad en la gestión.

No es un inconveniente la ubicación geográfica de los diferentes centros de apoyo de la Universidad Nacional de Loja (El Instituto de Idiomas, El Centro Binacional de Formación Técnica Zapotepamba, Área de la Salud Humana, etc...) ya que, dependiendo de la circunstancia el personal de soporte técnico aplica los acuerdos de nivel de servicio previamente establecidos

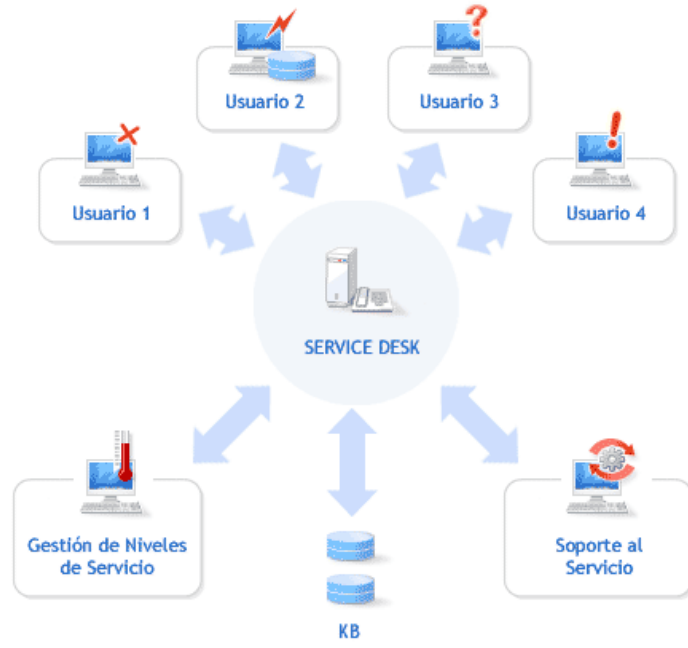


Figura 9. Estructura física centralizada. [3]

4.3. SISTEMAS HELP DESK - CASO PRÁCTICO.

Con la finalidad de realizar un análisis comparativo entre sistemas de mesa de ayuda, en éste apartado, se muestran dos ejemplos de proyectos, tanto libre como comercial.

4.3.1. PROYECTO GLPI

GLPI es software libre distribuido bajo licencia GPL (Licencia pública general), que facilita la administración de recursos informáticos. GLPI es una aplicación basada en Web escrita en PHP, que permite registrar y administrar los inventarios del hardware y el software de una empresa, optimizando el trabajo de los técnicos gracias a su diseño coherente. [19]

GLPI incluye también software de mesa de ayuda para el registro y atención de solicitudes de servicio de soporte técnico, con posibilidades de notificación por correo electrónico a usuarios y al mismo personal de soporte, al inicio, avances o cierre de una solicitud. Y entre las principales funcionalidades de GLPI están articuladas sobre dos ejes: el inventario preciso de todos los recursos informáticos, y el software existente, cuyas características se almacenan en bases de datos; y, la administración e historiales de las diferentes labores de mantenimiento y procedimientos relacionados, llevados a cabo sobre esos recursos informáticos.

Está desarrollada para entornos Apache-PHP-MySQL, usa también directorios LDAP, y en tanto, help desk para el seguimiento de intervenciones, permite a los administradores, y al personal de soporte, vincular las intervenciones realizadas a usuarios y a equipos, generándose así un historial completo del mantenimiento realizado, traducción a 22 idiomas y de coste cero.

4.3.2. KMKEY HELP DESK

Es un software de gestión de incidencias indicado para servicios de mantenimiento, ayuda al usuario y resolución de problemas en cualquier sector. Permite definir flujos de trabajo para abordar problemáticas derivadas de anomalías en servicios y maquinaria. La incidencia puede recibirse de forma automática (e-mail, entrada a través de una web, o desde un dispositivo móvil) o bien ser abierta por el servicio de atención. [20]

4.3.2.1. GESTIÓN DE INCIDENCIAS:

Recepción. Vía mail, entradas por el propio usuario, desde un formulario web etc. Abre de forma automática un expediente y avisa mediante notificación (interna, correo electrónico o SMS) al responsable.

Resolución. Atención y cerrado inmediato o planificación en el tiempo según tipo o urgencia. Asignación de recursos. Re-asignación. Escalado. Listas con el estado y los responsables. Retrasos. Avisos automáticos a quien sea necesario. Posibilidad de añadir notas, documentos, correos, fotos, manuales. Búsquedas en base de datos de conocimiento. Búsquedas indexadas en cualquier ítem, incluso dentro de los documentos. Introducción de horas y material utilizado. Facturación.

4.3.2.2. CONTROL

Generación y envío automático del resumen de la incidencia. Generación de informes y estadísticos por varios criterios: urgencia, tipo, contrato, elemento, técnico. Informes de horas invertidas. Informe de elementos problemáticos. Filtros e informes por periodos, personas, clientes etc.

4.3.2.3. GESTIÓN DE CONTRATOS Y ELEMENTOS. SLA.

Inventario. Elementos organizados por diferentes criterios: tipo, departamento, contratos etc.

SLA. Posibilidad de guardar y controlar documentación relativa a los contratos y condiciones de servicio. Realización de mantenimientos e intervenciones según las condiciones reflejado en las SLA (tiempos, urgencias, visibilidades).

Agenda. Base de datos de empresas y contactos. Calendario actividades. Mailing.

Integración e-mail y SMS. Notificaciones a terceros vía mail de acciones y tareas. Recepción automática de mails. Envío SMS.

Enlaces. Con la Gestión de Calidad o con la Gestión de proyectos. Enlaces con otros sistemas propios mediante desarrollos específicos: ERPs, BI, Blance Score Card etc.

4.3.3. ANÁLISIS COMPARATIVO

Para realizar un análisis comparativo entre las dos herramientas de mesa de ayuda antes descritas, se ha creado la siguiente tabla.

Característica	PROYECTO GLPI	KMKEY HELP DESK
Tipo de Código	Open source	Closed source
Licenciamiento	GPL	GPL
Estandarización	ISO 27001, ISO 20000, ISO 22301, ISO 9001	ITIL
Administración de inventarios	Si	Si
Administración de incidencias	Si	Si
Facturación	No	Si
Administración de acuerdos de nivel de servicio	No	Si
Administración de agenda	Si	Si
Notificaciones	Correo electrónico	Correo electrónico, mensajería telefónica.
Precio	No	Si
Planificación en el tiempo	No disponible	Por urgencias
Base de conocimiento	No	Si

Tabla 2. Comparativa del proyecto GLPI y KMKEY HELP DESK.

La comparativa de características entre el Proyecto GLPI y KMKEY HELP DESK de la Tabla 2, permitió comprobar que el sistema Help Desk para el Departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja cumple con las principales características para ser un sistema de mesa de ayuda y que éste funcione de manera efectiva.

Así, el sistema Help Desk se apega en mayor proporción a las características de KMKEY HELP DESK, puesto que utiliza la estandarización ITIL, y cuenta con módulos para la administración de incidencias, acuerdos de nivel de servicio, planificación en el tiempo, y notificaciones.

5. MATERIALES Y MÉTODOS

5.1. RECURSOS

5.1.1. TALENTO HUMANO

Recurso Humano	Cantidad	Horas	V. Unitario	V. Total
Asesoramiento Profesional	1	10	20.00	200.00
Investigadora	1	600	3.00	1800.00
			Subtotal	2000.00

Tabla 3. Presupuesto Talento Humano.

5.1.2. BIENES

Equipo	Depreciación		V. Depreciación / año	V. Total
	V. Real	T. útil/meses		
Computador portátil Dell C17	1480.00	14	105.71	14.00
Impresora	290.00	5	58.00	5.00
Memory 8 GB Kingston	10.00	3	3.33	3.00
			Subtotal	22.00

Tabla 4. Presupuesto de Recursos Técnicos.

5.1.3. SERVICIOS

Servicio	Cantidad	V. Estimado	V. Total	
Internet	200	0.60 hor.	120.00	
Llamadas Telefónicas	1	0.25 min.	15.00	
Transporte	--	75.00	75.00	
Cartuchos de tinta	4	5.00	20.00	
Resma de papel	2	4.50	9.00	
Anillados	4	1.50	6.00	
Perfiles	4	0.50	2.00	
			Subtotal	135.00

Tabla 5. Presupuesto de Servicios.

5.1.4. IMPREVISTOS

	Porcentaje Costo Directo	V. Total
Previsión de imprevistos (Talento+Bienes+Servicios)	5%	107.85
	Subtotal	107.85

Tabla 6. Presupuesto de Imprevistos

5.1.5. TOTALIDAD DE RECURSOS

Descripción	Total
Talento Humano	2000.00
Bienes	22.00
Servicios	135.00
Imprevistos	107.85
Total	2264.85

Tabla 7. Presupuesto de Totalidad de Recursos.

5.2. MÉTODOS Y TÉCNICAS

Para la ejecución del presente proyecto fue conveniente y necesaria la adopción de métodos que permitieron obtener información relevante y fiable, para ello se utilizó:

5.2.1. MÉTODOS

Método Científico: Permitió ir moldeando los resultados obtenidos del objeto de estudio y consecuentemente la corrección de errores según el avance del proceso de investigación. Así también, el de colaborar con un trabajo planificado y sustentable. Además de aplicar los conocimientos adquiridos mediante la ciencia disponible y alcanzar mayor experiencia.

Método Deductivo: Cuyo estudio parte de lo general a lo particular, ayudó a comprender, desarrollar la planificación y diseño del sistema de gestión de incidencias Help Desk, donde se inició con los procesos administrativos generales, hasta llegar a particularizar y determinar cada uno de los procedimientos inherentes en forma detallada.

Método Inductivo: Estudio que parte de lo particular a lo general, éste singular método se utilizó para la ejecución continua de pruebas de funcionalidad, donde como punto de partida del desarrollo de los componentes más pequeños, que forman parte de un módulo general.

Método híbrido: “los métodos híbridos son un enfoque de investigación cuantitativa y cualitativa en un mismo estudio, así se combina las fortalezas de cada enfoque aprovechando lo mejor de cada uno y superando sus deficiencias” [21]. Método útil para enriquecer y mejorar la comprensión del tema de estudio generando nuevas ideas con relación al mismo. Se aplicó en el sistema Help Desk al momento de realizar el procesamiento cuantitativo y el análisis cualitativo de información.

5.2.2. TÉCNICAS

Análisis de Información: Se usó para recopilar toda la información actual y relevante a cerca de los procesos del departamento de Mantenimiento y Soporte Técnico y su gestión, para determinar las principales necesidades requeridas en cuanto a los procesos administrativos y operativos.

Observación Directa: Mediante una observación de campo se pudo constatar la problemática referente al manejo actual de información, con la finalidad de complementar las historias de usuario obtenidas.

Entrevista: Se la utilizó para obtener mayor detalle de información en la dependencia, para constatar y enriquecer los datos recopilados.

Búsqueda de Información Científica: Se empleó para sustentar el desarrollo del presente trabajo en base a conocimientos adquiridos de múltiples personas que han hecho uso de la investigación científica, así como para plantear la solución, obtener y generar nuevos conocimientos.

5.2.3. METODOLOGÍA DE DESARROLLO DE SOFTWARE

La definición conceptual de las metodologías de desarrollo de software ICONIX, RUP, y XP se pueden visualizar en la Figura 5 de éste documento, sin embargo, la metodología más equilibrada a aplicar en el sistema de gestión de incidencias Help Desk, es la metodología XP, por las siguientes razones:

El equipo de desarrollo: Para ejecutar éste proyecto, el equipo de desarrollo se limitó por el número de integrantes, ya que se redujo a un desarrollador (Tesista), y el Jefe de Departamento, y XP propone la programación en parejas e incluir al usuario final como parte del equipo de desarrollo.

Modelo: Éste proyecto requería trabajar con entregas de software en el menor tiempo posible cumpliendo paulatinamente un ciclo de vida en cada fase. XP propone un modelo iterativo e incremental con funcionamiento interno del modelo en cascada, es por esto que son posible las entregas de software durante la ejecución de cada iteración.

Velocidad: El proyecto de titulación exigía la realización del software en un plazo máximo de diez meses, así, XP propone el desarrollo ágil, velocidad de reacción para la implementación y los cambios y, reducida documentación.

Seguidamente, se detalla las fases de la metodología XP y los elementos éste proyecto aplicó.

FASE DE PLANEACIÓN. En ésta fase se interactuó con el cliente para recopilar información y definir los requerimientos del sistema. (Ver apartado 6.1 Fase de planeación)

FASE DE DISEÑO. En ésta fase se diseñó las historias de usuario, y aplicó la metáfora del sistema, modelos conceptuales y tarjetas CRC. (Ver apartado 6.2 Fase de diseño)

FASE DE CODIFICACIÓN. En ésta fase se configuró el entorno de desarrollo y se codificó los algoritmos con la tecnología y arquitectura propuestas, además se ejecutó el plan de iteraciones. (Ver apartado 6.3 Fase de codificación)

FASE DE PRUEBAS. En ésta fase, se realizó las pruebas para determinar fallos o errores en la aplicación y corregirlos, para ello se ejecutó un plan de validación y verificación. (Ver apartado 6.4 Fase de pruebas)

6. RESULTADOS

En éste apartado se describe las fases de la metodología ágil de desarrollo XP que se utilizaron para el desarrollo del sistema Help Desk, donde se trata de realizar ciclos cortos (iteraciones) con entregables funcionales al finalizar cada ciclo. Cada iteración se realizó en un ciclo completo de planeación, diseño, codificación y pruebas aplicando un conjunto de reglas y practicas propias de XP.

6.1. FASE DE PLANEACIÓN

En esta fase se realizó una panorámica general del proyecto, mediante la redacción sencilla de historias de usuario, definición de la velocidad del proyecto, diseño del plan de iteraciones, roles de XP, roles de usuario y, requerimientos funcionales y no funcionales, además “se definió las herramientas, tecnología y prácticas que se utilizaron, se probó la tecnología, y determinó la arquitectura del sistema construyendo un prototipo”, [6] ésta información fue obtenida gracias a la aplicación de: una entrevista a la Ing. Mabel Rodríguez encargada del departamento de Soporte Técnico de la Universidad Nacional de Loja (Ver Anexo 5); y, la técnica de la observación obteniendo como resultado el registro anecdótico del Anexo 6.

6.1.1. HISTORIAS DE USUARIO

“Las historias de usuario son documentos usados para la especificación de requisitos, sustituyen a los documentos de especificación funcional y a los casos de uso.” [9]. Para efectos de éste estudio se aplicó el formato de historia de usuario de la Figura 10, el mismo que recopiló la información del flujo de procesos del Departamento de Soporte de la Universidad Nacional de Loja. Las historias de usuario, también se utilizaron para estimar el tiempo que el equipo de desarrollo tomó para realizar las entregas.

HISTORIA DE USUARIO	
Número:	Nombre de Historia:
Usuario:	
Prioridad de Negocio: (Alto/Medio/Bajo)	Riesgo de Desarrollo: (Alto/Medio/Bajo)
Puntos estimados:	Iteración Asignada:
Descripción:	
Observaciones:	Firma:

Figura 10. Formato de Historia de Usuario.

6.1.1.1. ÍNDICE DE VALORACIÓN PARA HISTORIAS DE USUARIO

Del formato de la historia de usuario aplicada en éste estudio se realizó un índice de valoración para los campos de “**Prioridad de Negocio**”, “**Riesgo de Desarrollo**”, “**Puntos estimados**” e “**Iteración asignada**”, esto permitió realizar una estimación medible y obtener un desarrollo controlable del proyecto.

Prioridad De Negocio		
La Prioridad de Negocio , es el grado de prioridad para el desarrollo de la historia de usuario. [6] El índice de estimación se define en tres valoraciones cualitativas.		
Alto	Medio	Bajo
Significa que esa historia de usuario se encuentra entre las primeras que se debe desarrollar (etapa inicial del desarrollo). Implica mayor preferencia de desarrollo.	Significa que esta historia de usuario puede desarrollarse con un nivel de prioridad media. Implica menor preferencia en el orden de desarrollo.	Significa que esta historia de usuario está ubicada entre las historias que pueden desarrollarse en la etapa final del desarrollo. Implica preferencia mínima.

Tabla 8. Índice de valoración del campo Prioridad de Negocio.

Riesgo De Desarrollo		
El Riesgo de desarrollo , se define basado en el riesgo que afronta el equipo de desarrollo para satisfacer los requerimientos del cliente. [6]		
Alto	Medio	Bajo
Implica que la historia de usuario debe satisfacer los requerimientos del cliente con un porcentaje igual o mayor al 80%.[80%-100%]	Implica que la historia de usuario debe satisfacer los requerimientos del cliente con un porcentaje igual o superior al 45%.[45%-79%].	Implica que la historia de usuario satisfaga los requerimientos del cliente con un porcentaje menor al 45%.[0%-44%].

Tabla 9. Índice de valoración del campo Riesgo De Desarrollo.

Puntos Estimados																				
<p>Los Puntos Estimados, son puntos de esfuerzo determinados en base a la duración del desarrollo de la historia de usuario. [6] De acuerdo a las buenas prácticas que persigue XP, se debe trabajar como máximo 40 horas por semana y elaborar como máximo 2 historias por mes, por lo que se estimó una escala numérica de puntos del 1 al 20.</p>																				
Puntos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Horas de desarrollo	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160
Nro. Semana	Semana 1.					Semana 2.					Semana 3.					Semana 4.				

Tabla 10. Índice de valoración del campo Puntos Estimados.

Iteración asignada											
<p>La Iteración asignada, es el número productos entregables de la historia de usuario, se puede asignar hasta un número máximo de 4 iteraciones, que lo puede definir el cliente o el equipo de desarrollo.</p> <p>La duración mínima de cada iteración es de 1 semana o 5 puntos.</p> <p>La duración máxima de cada iteración es de 3 semanas o 15 puntos.</p>											
Número de Iteración	1		2		3			4			
Puntos	1-15	1-15	16-20	1-6	7-13	14-20	1-5	6-10	11-15	16-20	

Tabla 11. Índice de valoración del campo Iteración asignada.

6.1.1.2. HISTORIAS DE USUARIO APLICADAS

Debido al número reducido de personal que labora en la sección de Soporte Técnico, las historias de usuario se aplicaron a personas que tienen una relación directa con el flujo de procesos y de las cuales se puede recopilar la mayor y mejor información, la persona seleccionada, representa a cada tipo de funcionario administrativo.

Si bien el cliente no fue quien escribió personalmente las historias de usuario, fue el quien diseñó su contenido y dirigió la redacción de las mismas debido a que no tenía los conocimientos necesarios en formato para elaborarlas y mucho menos estimarlas.

A continuación se muestra las historias de usuario que se obtuvieron del levantamiento de requerimientos.

HISTORIA DE USUARIO	
Número: 001	Nombre de Historia: Tipos de mantenimiento
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de Desarrollo: Bajo (Alto/Medio/Bajo)
Puntos estimados: 8	Iteración Asignada: 1
Descripción:	
<ul style="list-style-type: none"> • Se debe identificar dos tipos de mantenimiento: el preventivo y correctivo. • Cada tipo de mantenimiento debe tener una estimación del tiempo mínimo y máximo de solución. • La estimación de tiempo puede variar dependiendo del lugar desde donde se reporta la incidencia. 	
Observaciones:	
<ul style="list-style-type: none"> • El mantenimiento preventivo se realiza para mantener la conservación de los equipos y garantizar su buen funcionamiento. • El mantenimiento correctivo es aplicado en los equipos informáticos, para corregir ciertos defectos o fallas que puedan presentar. • De acuerdo al tipo de acontecimiento o tipo de problema en el equipo se puede tomar las siguientes acciones: <ul style="list-style-type: none"> - Se traslada el personal hasta el lugar del incidente. - Se lleva las herramientas necesarias y se trata de dar solución al inconveniente en el equipo afectado. - Si el problema persiste o no se puede solucionar en el lugar, se traslada el equipo informático hasta las oficinas. 	

Tabla 12. Historia de usuario tipos de mantenimiento.

HISTORIA DE USUARIO	
Número: 002	Nombre de Historia: Tipos de personal.
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de Desarrollo: Medio (Alto/Medio/Bajo)
Puntos estimados: 13	Iteración Asignada: 1
Descripción:	
<ul style="list-style-type: none"> • Se debe identificar los siguientes tipos de usuario: El usuario que genera la incidencia, el Jefe de Departamento que distribuye el trabajo, el Asesor Técnico quien realiza el trabajo. • El Jefe de Departamento es quien distribuye equitativamente el trabajo a los Asesores Técnicos para cumplir con todas las incidencias que se reporten. 	
Observaciones:	
<ul style="list-style-type: none"> • El trabajo se asigna a la persona que tenga mayor disponibilidad de atender el incidente. 	

Tabla 13. Historia de usuario tipos de personal.

HISTORIA DE USUARIO	
Número: 003	Nombre de Historia: Solicitud de asistencia técnica
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Alto (Alto/Medio/Bajo)	Riesgo de Desarrollo: Alto (Alto/Medio/Bajo)
Puntos estimados: 20	Iteración Asignada: 2
Descripción:	
<ul style="list-style-type: none"> • Cada incidencia que se reporta debe ser registrada. • Cuando se recepta una incidencia se debe clasificar por turno y según el nivel de prioridad. • Los bienes que se pueden reportar en una incidencia son exclusivamente en los equipos de la institución. • Con respecto al equipo afectado: <ul style="list-style-type: none"> - Se recepta los principales datos del equipo. - Se obtienen los síntomas que presenta el equipo afectado para realizar un diagnóstico predictivo e identificar qué tipo de mantenimiento se debe aplicar. 	
Observaciones:	
<ul style="list-style-type: none"> • Cuando se requiere cambiar alguna pieza del equipo afectado se realiza la petición a la Coordinación Financiera Administrativa del área correspondiente. 	

Tabla 14. Historia de usuario solicitud de asistencia técnica.

HISTORIA DE USUARIO	
Número: 004	Nombre de Historia: Solicitud de asistencia técnica postergada
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de Desarrollo: Alto (Alto/Medio/Bajo)
Puntos estimados: 18	Iteración Asignada: 2
Descripción:	
<ul style="list-style-type: none"> • Cuando se ha solicitado la adquisición de una pieza o parte, y mientras el recurso es asignado, la incidencia se posterga. • La incidencia puede estar: Nuevo registro, en espera de recurso, terminada y pendiente. 	
Observaciones:	

Tabla 15. Solicitud de asistencia técnica postergada

HISTORIA DE USUARIO	
Número: 005	Nombre de Historia: Registro de equipo
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Medio (Alto/Medio/Bajo)	Riesgo de Desarrollo: Alto (Alto/Medio/Bajo)
Puntos estimados: 14	Iteración Asignada: 1
Descripción: <ul style="list-style-type: none"> • Antes de ingresar el equipo al área de reparación de la Sección de Mantenimiento, se requiere registrar sus datos, como el nombre del custodio o responsable, el nombre de la persona que trae el equipo, y los síntomas que presenta. • Cada bien institucional puede ser registrado y verificado de forma única, definiendo el número de serie, código institucional, o código del Ministerio de Finanzas. 	
Observaciones:	

Tabla 16. Historia de usuario registro de equipo.

HISTORIA DE USUARIO	
Número: 006	Nombre de Historia: Entrega de equipos
Usuario: Ing. Mabel Rodríguez.	
Prioridad de Negocio: Bajo (Alto/Medio/Bajo)	Riesgo de Desarrollo: Medio (Alto/Medio/Bajo)
Puntos estimados: 10	Iteración Asignada: 1
Descripción: <ul style="list-style-type: none"> • Cuando se finaliza la atención se debe agregar alguna observación sobre el trabajo realizado. • Luego de haber solucionado el problema el equipo es devuelto a su responsable para su uso. • El usuario que recepta el equipo debe calificar la atención recibida. 	
Observaciones: <ul style="list-style-type: none"> • Dependiendo del caso o del tipo de equipo se debe realizar la instalación en el área de trabajo correspondiente. 	

Tabla 17. Historia de usuario entrega de equipos.

6.1.2. VELOCIDAD DEL PROYECTO

El número de historias de usuario realizadas por iteración no fue una buena medida de la velocidad del proyecto, debido a que no todas tienen la misma puntuación y por lo

tanto las mismas horas de desarrollo. Puesto que cada historia me reflejaba una o dos iteraciones y no viceversa, por esto se encontró que en las segundas iteraciones se trabajaba menos horas semanales en comparación con las primeras iteraciones, lo que implicaría un nivel de rendimiento superior, pero no es cierto, el motivo de éste resultado fue que el nivel de dificultad y por lo tanto el número de horas requeridas para la segunda iteración fueron las más bajas de todo el proyecto.

	ITERACIÓN 1	ITERACIÓN 2
Horas	120	40
Semanas	3	1
Horas semanales	40	40
Historias de usuario (Velocidad del proyecto)	1	1

Tabla 18. Velocidad del proyecto.

Si bien ésta medida fue tomada en cuenta, la mejor medida de velocidad y con mayor utilidad de estimar es el número de horas que tomaría implementar cada historia de usuario y planificar las iteraciones en base a ésta medida.

6.1.3. ROLES DE XP

A continuación se definen los roles que cada integrante desempeña durante el transcurso del proyecto.

ROLES DE XP	
ROL	ENCARGADO
Analista	Diana Alexandra Morocho Puchaicela
Diseño de interfaz	Diana Alexandra Morocho Puchaicela
Programador	Diana Alexandra Morocho Puchaicela
Tester	Diana Alexandra Morocho Puchaicela
Cliente	Mabel Rodríguez

Tabla 19. Roles de XP.

6.1.4. ROLES DE USUARIO DEL SISTEMA HELP DESK

El sistema Help Desk para la sección de Mantenimiento de la Universidad Nacional de Loja contó con los siguientes roles de usuario para el acceso al sistema.

ROLES DE USUARIO	
Rol	Función
SUPER ADMINISTRADOR	Es el usuario encargado de la administración general y absoluta del sistema, permitiéndole administrar las funciones relacionadas con el módulo de usuarios, roles y privilegios del sistema, contactos institucionales, control de calidad, gestión de bienes institucionales, incidencias, logs – acciones realizadas, respaldos.
JEFE DE DEPARTAMENTO	A éste usuario se le asignó poder realizar funciones de administración departamental, como las de actualizar la lista de usuarios jefe departamento, y asesor técnico para el centro que se ha asignado, administrar acuerdos de nivel de servicio, administrador de incidencias, asignar, redirigir, actualizar incidencias, despachar solicitudes de incidencia expirada y extensión de tiempo de apertura, medir el rendimiento de <i>Asesor Técnico</i> del centro, generar reportes.
ASESOR TÉCNICO	Es el usuario que realiza las tareas operativas, se le asignó funciones que le permitieron manipular los registros de solicitud de incidencias, efectuar diagnósticos sobre los equipos y determinar las soluciones, incidencias asignadas, atender incidencias, generar solicitudes de incidencia expirada y extensión de tiempo de apertura, administrar acciones y finalizar incidencias.
USUARIO FINAL	Este usuario es el encargado de generar las incidencias hacia el centro de asistencia.

Tabla 20. Roles de usuario del sistema Help Desk.

6.1.5. REQUERIMIENTOS FUNCIONALES

Un requerimiento es una característica que el sistema debe cumplir para ser aceptado por el cliente. El sistema Help Desk dispone de las siguientes características útiles, que se detallan en la siguiente tabla de requerimientos funcionales.

REQUERIMIENTOS FUNCIONALES			
Ref.	Requerimiento Funcional	Visibilidad	Historia de Usuario
El sistema Help Desk permite:			
RF-001	El acceso al sistema mediante los roles de usuario: <i>administrador, jefe de departamento, personal técnico y usuario final.</i>	Evidente	002
RF-002	Al usuario <i>administrador</i> realizar las operaciones de CRUD para los demás roles de usuarios en el sistema.	Evidente	002
RF-003	Al usuario <i>administrador</i> y <i>jefe departamento</i> administrar centros de asistencia.	Evidente	001
RF-004	Al usuario <i>administrador</i> y <i>jefe departamento</i> realizar las operaciones de crear, actualizar, eliminar y buscar acuerdos de nivel de servicio	Evidente	001
RF-005	Al <i>usuario final</i> crear, actualizar, eliminar y buscar incidencias.	Evidente	003
RF-006	Al <i>usuario final</i> visualizar el estado actual de la incidencia generada.	Oculto	003
RF-007	Al <i>usuario final</i> calificar el servicio prestado en la atención a la incidencia generada.	Evidente	005
RF-008	Al usuario <i>jefe de departamento</i> asignar y dirigir las incidencias generadas en el centro de asistencia, hacia el usuario <i>asesor técnico</i> pertinente.	Evidente	002
RF-009	Generar una función para el usuario <i>asesor técnico</i> que permita determinar el tiempo restante de las incidencias que le han sido asignadas.	Evidente	002-004
RF-010	Al <i>asesor técnico</i> crear las acciones realizadas referentes a la atención de la incidencia.	Evidente	005

RF-011	A los usuarios, acceder en el sistema mediante un perfil de usuario que permita su identificación.	Evidente	003
RF-012	Registrar y reportar los datos principales del equipo que requiere de asistencia técnica, en el caso de ser necesario.	Evidente	003
RF-013	Asignar a las incidencias una etiqueta de prioridad de atención.	Oculto	003
RF-014	Al usuario <i>jefe departamento</i> , ejecutar la función de medición de calidad de servicio al <i>asesor técnico</i> perteneciente al centro de asistencia correspondiente.	Evidente	005
RF-015	Al usuario <i>personal técnico</i> finalizar la atención de la incidencia.	Oculto	006
RF-016	Identificar el estado de las incidencias generadas mediante algún identificador.	Evidente	004

Tabla 21. Requerimientos funcionales del sistema Help Desk.

6.1.6. REQUERIMIENTOS NO FUNCIONALES

REQUERIMIENTOS NO FUNCIONALES	
Ref.	Requerimiento No Funcional
RNF-001	La aplicación web está desarrollada con lenguaje y herramientas no privativos.
RNF-002	El diseño de las interfaces de usuario de la aplicación debe desarrollarse lo más intuitivo como sea posible.
RNF-003	El desarrollo de la aplicación debe ajustarse a los lineamientos técnicos que propone el departamento de Software en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.
RNF-004	Debe poseer algún grado de escalabilidad para facilitar su crecimiento en el futuro.
RNF-005	El sistema debe contar con un manual de usuario.

RNF-006	Las contraseñas de los usuarios deben contener algún algoritmo de encriptación.
---------	---

Tabla 22. Requerimientos no funcionales del sistema Help Desk.

6.1.7. PLAN DE ITERACIONES

Es una planificación donde se establecen la prioridad y los tiempos de implementación ideales de las historias de usuario en cada versión del programa. La clasificación de las historias no fue realizada estrictamente por su grado de importancia en el proyecto. Solo se optó por desarrollar la configuración inicial y prueba de herramientas y, la administración de centros de asistencia en las dos primeras iteraciones, por tratarse de las actividades las importantes en el negocio.

CALENDARIO DE TRABAJO		
Código	Semana	Función a desarrollar.
PE-001	[1,3[Configuración inicial y prueba de herramientas de desarrollo.
PE-002	[3,4[Administración de centros de asistencia – primera iteración – RF-003
PE-003	[4,5]	Administración de acuerdos de nivel de servicio en centros de asistencia - segunda iteración – RF-004
PE-004]5,6[Administración de usuarios – Acceso al sistema – primera iteración- RF-001, RF-002
PE-005	[6,7[Solicitud de incidencia – primera iteración – RF-005.
PE-006	[7,8[Solicitud de incidencia – segunda iteración – RF-005 – RF-006
PE-007	[8,9[Asignación de incidencias – primera iteración – RF-008
PE-008	[9,10[Asignación de incidencias – segunda iteración – RF-008
PE-009	[10,13[Atender incidencia – iteración única – RF-009, RF-010
PE-010	[13,14[Registro de equipo – primera iteración – RF-012
PE-011	[14,16[Medición de la calidad de servicio – primera iteración – RF-014
PE-012	[16,17[Medición de la calidad de servicio - Evaluación – segunda iteración – RF-014

Tabla 23. Plan de entrega Sistema Help Desk

6.2. FASE DE DISEÑO

A diferencia de las metodologías pesadas, el diseño se realizó todo el tiempo de vida del proyecto siendo constantemente revisado y probablemente modificado debido a los cambios presentados durante el desarrollo. En ésta fase se diseñó las historias de usuario, la metáfora del sistema, un diagrama de casos de uso, modelos conceptuales y tarjetas CRC.

6.2.1. METÁFORA DEL SISTEMA

La siguiente tabla representa una lista de términos utilizados a lo largo del desarrollo de todo el proyecto, para mejorar la comprensión del entorno y de los contenidos relacionados con el análisis e implementación del sistema Help Desk.

METÁFORA DEL SISTEMA	
METÁFORA	SIGNIFICADO
Acuerdo de Nivel de Servicio (SLA)	Mecanismo de contrato entre el centro de asistencia y el usuario final para el cumplimiento de la calidad del servicio.
Administrador de incidencias	Función desarrollada en el Sistema Help Desk y asignada al usuario <i>Jefe Departamento</i> , que tiene la finalidad de administrar todas las incidencias registradas para el centro de asistencia.
Calidad de Servicio	Se entiende como la satisfacción de las necesidades y expectativas del <i>Usuario Final</i> expresadas a través de una valoración.
Centro de Asistencia	Es la base general del sistema, representando para éste estudio al Departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja.
Incidencia	Circunstancia o suceso secundario que ocurre en el desarrollo de un asunto o negocio, que puede influir e interrumpir en la consecución de los objetivos.
Incidencia Asignada	Los casos de incidencia registrados para un centro y asignados para la atención al <i>Asesor Técnico</i> .

Estado de Incidencia	Representa la forma o cambios que presenta una incidencia durante el transcurso de su atención.
Notificación	Nomenclatura utilizada para registrar una acción comunicada.
Rol de usuario	La función que desarrolla un determinado usuario en el sistema, agrupado en una colección de permisos.
Cuestionario	Instrumento de investigación con una serie de preguntas con el objetivo de recopilar información.
Pregunta	Expresión para solicitar información.
Solicitud de Recurso	Función que tiene por objetivo el pedido de equipos o bienes.
Solicitud de Extensión	Función que tiene por objetivo extender el tiempo de iniciación de una incidencia no aperturada.
Solicitud de Reapertura	Función que tiene por objetivo reaperturar incidencias expiradas.

Tabla 24. Metáfora del sistema.

6.2.2. DIAGRAMA GENERAL DE CASOS DE USO

Para mejorar la comprensión del comportamiento del sistema Help Desk, se aplicó el artefacto de casos de uso, puesto que: “Un caso de uso es una secuencia de interrelaciones entre un sistema y alguien o algo que usa alguno de sus servicios” [22] y por lo tanto define una funcionalidad concreta, entre tanto las historias de usuario solo muestran la silueta de una tarea a realizarse.

Por otra parte, el diagrama de casos de uso es un elemento de esquematización de UML, y la metodología XP puede adoptar las prácticas del UML, lo que actualmente se denomina Modelamiento Ágil (AM).

En fin, los casos de uso son opcionales y no reemplazan a las historias de usuario, pero para éste estudio, resultó útil para analizar, elaborar, comprender el funcionamiento y mejorar la calidad del sistema.

La representación gráfica del uso de requisitos de negocio para el Sistema Help Desk se muestra en el siguiente diagrama general de casos de uso. (Ver Figura 11)

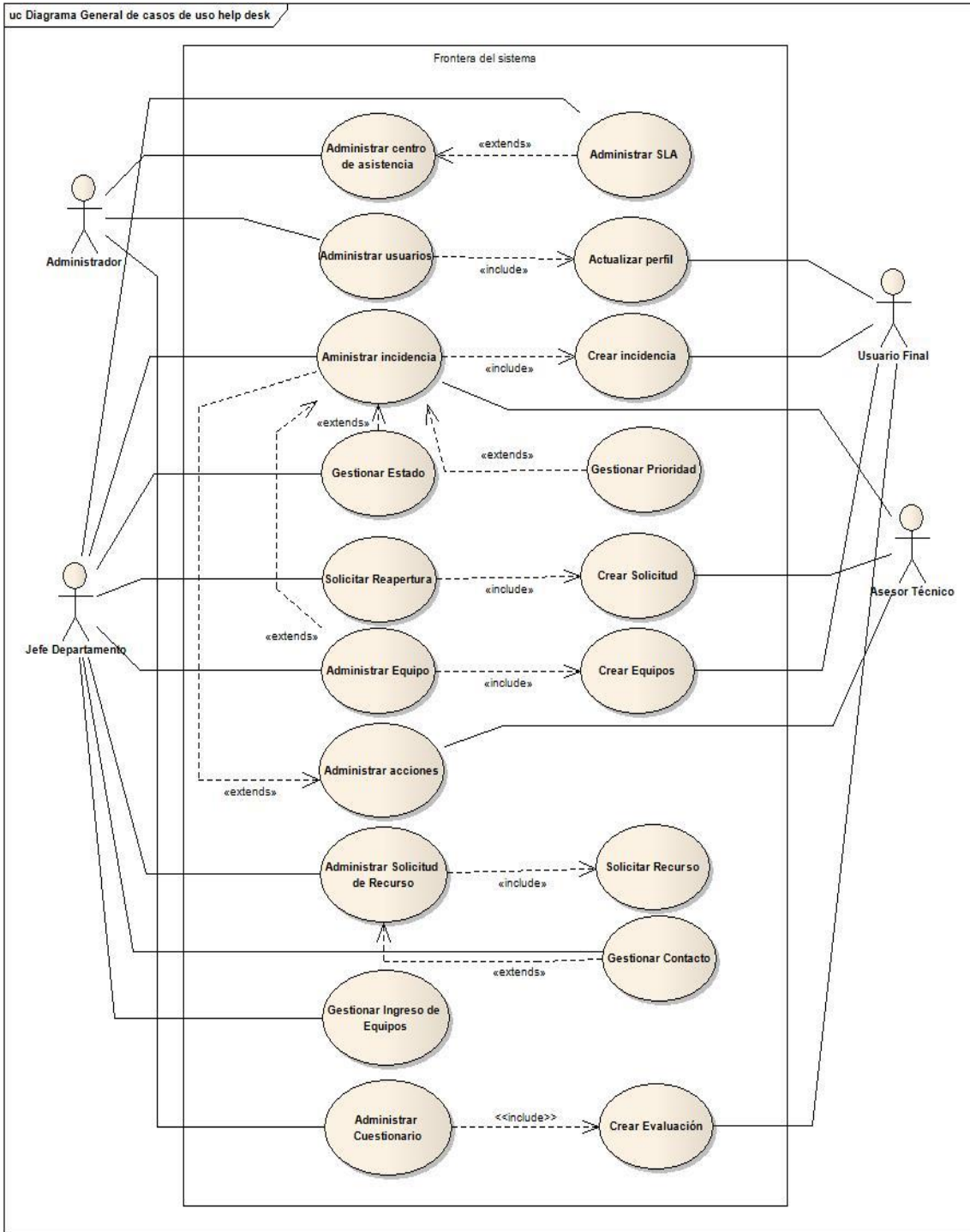


Figura 11. Diagrama general de casos de uso.

6.2.3. MODELO CONCEPTUAL

El modelo conceptual que se aplicó al sistema Help Desk, contempla el diseño de dos diagramas: Modelo de dominio y modelo de base de datos que se describen a continuación.

6.2.3.1. MODELO DEL DOMINIO

El modelo del dominio, es una representación de las clases conceptuales del mundo real, no de componentes de software, en los que se defina alguna operación. En éste diagrama se muestran: objetos del dominio o clases conceptuales, asociación entre clases conceptuales, y atributos entre clases conceptuales. [23] El modelo del dominio general y no parcial aplicado en éste proyecto, se puede ver con detalle en el Anexo 7 de éste documento.

6.2.3.2. MODELO DE CLASES

El modelo de clases, es un tipo de diagrama del UML, donde se muestra la estructura del sistema mostrando las clases, atributos, métodos, y relaciones entre los objetos. Para obtener mayor detalle sobre éste diagrama dirigirse al Anexo 8 de éste documento.

6.2.3.3. MODELO DE BASE DE DATOS

El modelo de base de datos, es un tipo de modelo de datos que determina la estructura lógica de una base de datos y de manera fundamental determina el modo de almacenar, organizar y manipular los datos. [23] Para obtener mayor detalle sobre éste diagrama dirigirse al Anexo 9 de éste documento.

6.2.4. TARJETAS CRC

Una de las principales piezas de diseño empleada en el proyecto fueron las tarjetas CRC que no solo sirvieron como columna vertebral de este, sino también fueron la base para modelar la base de datos. Cada tarjeta CRC se convirtió en un objeto, sus responsabilidades en métodos públicos y sus colaboradores en llamadas a otras clases. [22] Se definió un escenario para cada historia de usuario.

6.2.4.1. ESCENARIO NRO.1 - TIPOS DE MANTENIMIENTO

TARJETA CRC NRO 1.	
Nombre:	CENTRO DE ASISTENCIA
Escenario:	Tipos de mantenimiento
Descripción:	En esta clase se llevó el control del centro de asistencia, así como los acuerdos de nivel de servicio y sus respectivos tiempos de atención para cada tipo de mantenimiento.

RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> - Registrar centros de asistencia - Proveer de acuerdos de nivel de servicio. - Registrar a los usuarios operadores: Jefe de Departamento y Asesor Técnico. 	<ul style="list-style-type: none"> - Estadística SLA. - Servicio - Personal Operativo - Perfil

Tabla 25. Tarjeta CRC Centro de Asistencia.

6.2.4.2. ESCENARIO NRO.2 – TIPOS DE PERSONAL

TARJETA CRC NRO 2.	
Nombre:	PERFIL
Escenario:	Tipos de personal
Descripción:	En esta clase se define los cuatro roles de usuario para el acceso al sistema Help Desk (Usuario Final, Asesor Técnico, Jefe Departamento, Administrador)
RESPONSABILIDADES	COLABORADORES
<ul style="list-style-type: none"> - Registrar datos personales. - Ingresar al sistema. - Generar incidencias. - Asignar incidencias. - Atender incidencias. - Evaluar atención al cliente. - Realizar el diagnóstico inicial de las incidencias asignadas. - Generar reportes. - Medición de la calidad del servicio. - Registrar acciones a las incidencias. - Crear evaluaciones y cuestionarios 	<ul style="list-style-type: none"> - Centro de asistencia - Incidencia - Apertura incidencia - Cierre incidencia - Evaluación - Diagnóstico inicial. - Acción

Tabla 26. Tarjeta CRC Perfil.

6.2.4.3. ESCENARIO NRO.3 – SOLICITUD DE ASISTENCIA TÉCNICA

TARJETA CRC NRO 3.	
Nombre:	INCIDENCIA
Escenario:	Solicitud de asistencia técnica.

Descripción:	En esta clase se administran los datos referentes a las incidencias reportadas, se define prioridades de ejecución, acciones realizadas, solicitud y despacho de recursos.	
RESPONSABILIDADES		COLABORADORES
<ul style="list-style-type: none"> - Reportar incidencias seleccionado el centro de asistencia. - Reporte de bienes institucionales para el mantenimiento. - Solicitar la prioridad de ejecución. - Mostrar las acciones realizadas por el Asesor Técnico. - Cambiar de estado según la circunstancia actual de la incidencia. 		<ul style="list-style-type: none"> - Diagnóstico inicial - Acción. - Solicitud reapertura. - Bien. - Apertura incidencia. - Cierre incidencia. - Historial de incidencia. - Centro de asistencia.

Tabla 27. Tarjeta CRC Incidencia.

6.2.4.4. ESCENARIO NRO.4 – SOLICITUD DE ASISTENCIA TÉCNICA POSTERGADA

TARJETA CRC NRO 4.	
Nombre:	ESTADO DE INCIDENCIA
Escenario:	Solicitud de asistencia técnica postergada.
Descripción:	En esta clase se definen varios estados en los que puede permanecer una incidencia (Nueva, Asignada, Atendiendo, Expirada, Pendiente, Reaperturada, Finalizada)
RESPONSABILIDADES	
COLABORADORES	
<ul style="list-style-type: none"> - Poner en estado Nueva, a la nueva incidencia reportada. - Determinar el estado Asignada, a la incidencia que el usuario jefe departamento ya ha asignado para la solución. - Cambiar al estado Atendiendo, cuando el asesor técnico tenga vigente la atención de la incidencia. - Cambiar al estado Expirada, cuando no se ha cumplido con el plazo de atención establecido. - Cambiar al estado Pendiente, cuando se ha realizado la solicitud de recursos. 	
<ul style="list-style-type: none"> - Incidencia - Historial de incidencia 	

<ul style="list-style-type: none"> - Cambiar al estado Reaperturada, cuando se ha aperturado nuevamente luego del vencimiento de plazo establecido para la atención de la incidencia. - Cambiar al estado Finalizada, cuando la atención ha concluido. 	
--	--

Tabla 28. Tarjeta CRC Estado de incidencia.

6.2.4.5. ESCENARIO NRO.5 – REGISTRO DE EQUIPO

TARJETA CRC NRO 5.	
Nombre:	EQUIPO
Escenario:	Registro de equipo
Descripción:	En esta clase se administra los datos referentes a los equipos o bienes institucionales.
RESPONSABILIDADES	
<ul style="list-style-type: none"> - Registrar el código de serie, código institucional, o el código asignado por el Ministerio de Finanzas. - Incluir al bien institucional en el reporte de incidencias. 	COLABORADORES
	<ul style="list-style-type: none"> - Incidencia. - Perfil.

Tabla 29. Tarjeta CRC Equipo.

6.2.4.6. ESCENARIO NRO.6 – ENTREGA DE EQUIPOS

TARJETA CRC NRO 6.	
Nombre:	EVALUACION
Escenario:	Entrega de equipos
Descripción:	Posterior a la entrega del equipo y de la finalización de la incidencia, el usuario final debe calificar la atención recibida.
RESPONSABILIDADES	
<ul style="list-style-type: none"> - Evaluar la atención recibida por el asesor técnico. - Diseñar el cuestionario de evaluación. - Generar preguntas y métricas. - Definir el sumario de las acciones realizadas en la incidencia. 	COLABORADORES
	<ul style="list-style-type: none"> - Cuestionario. - Pregunta. - Respuesta. - Sumario incidencia.

Tabla 30. Tarjeta CRC Evaluación.

6.3. FASE DE CODIFICACIÓN

En ésta fase de la metodología XP se implementó del código fuente del sistema Help Desk. Seguidamente se muestra como está estructurado el código del aplicativo y la ejecución de las iteraciones definidas en el plan de iteraciones.

6.3.1. DETALLE DE LA IMPLEMENTACIÓN

6.3.1.1. CONFIGURACIÓN DEL ENTORNO

A continuación, se describe de forma general la configuración de la tecnología que se utilizó en el proyecto para el sistema operativo Windows.

❖ INSTALACIÓN DE PYTHON

Ingresa a la página oficial de Python (<https://www.python.org/downloads/>), descargar el instalador, y seguir los pasos de configuración del asistente (Ver Figura 12).



Figura 12. Asistente de instalación de python.

Para verificar la correcta instalación, ejecutar el comando `>> python --versión`

```
C:\Users\Diana>python --version
Python 2.7.10
```

Figura 13. Comando de verificación python.

❖ INSTALACIÓN DE PAQUETES PIP

Primero, descargar el archivo instalador desde este enlace <https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py>. Presiona CTRL + S (o CTRL + G) para guardarlo en el ordenador. Luego, una vez situado con la terminal

(línea de comandos) en donde se ha guardado get-pip.py, ejecuta el siguiente comando:

>> **python get-pip.py**

Finalmente para verificar que la instalación fue correcta ejecutar: >>**pip --version**

```
C:\Users\Diana>pip --version
pip 7.1.2 from c:\python27\lib\site-packages (python 2.7)
```

Figura 14. Comando de verificación pip.

❖ INSTALACIÓN DE DJANGO

Acceder a la terminal de comandos y ejecutar >> **pip install django**

Y para verificar la correcta instalación ejecuta los comandos de la Figura 15.

```
C:\Users\Diana>python
Python 2.7.10 <default, May 23 2015, 09:40:32> [MSC v.1500 32 bit <Intel>] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> print django.get_version()
1.9
```

Figura 15. Comandos de instalación de django.

❖ INSTALACIÓN DE POSTGRESQL

Acceder a la página oficial de postgresql (<http://www.postgresql.org/download/windows/>) y descargar el instalador, ejecutar el archivo y seguir los pasos de instalación. Ver Figura 16.

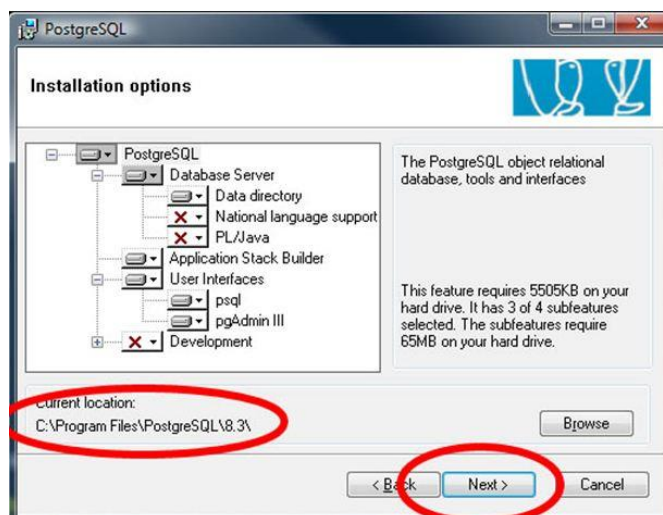


Figura 16. Asistente de instalación de postgresql.

❖ INSTALACIÓN DE NODEJS

Acceder a la página oficial de Nodejs (<https://nodejs.org/en/>), descargar y ejecutar el archivo. Ver Figura 17.



Figura 17. Asistente de instalación de NodeJS

❖ INSTALACIÓN DE REDIS

Acceder a la página oficial de Redis (<http://redis.io/download>), descargar el archivo y descomprimir. Ejecutar el archivo redis-server.exe



Figura 18. Servidor Redis.

❖ INSTALACIÓN DE ISHOUT.JS

Apertura una consola de comandos y ejecutar el comando `>> npm install ishout.js`

❖ DESCARGA DEL CÓDIGO FUENTE

Acceder a la página de github donde se encuentra alojado el código fuente del proyecto (<https://github.com/dianamor8/helpdeskunl>), abrir una consola de comandos y clonar el repositorio ejecutando >> **git clone** <https://github.com/dianamor8/helpdeskunl.git>

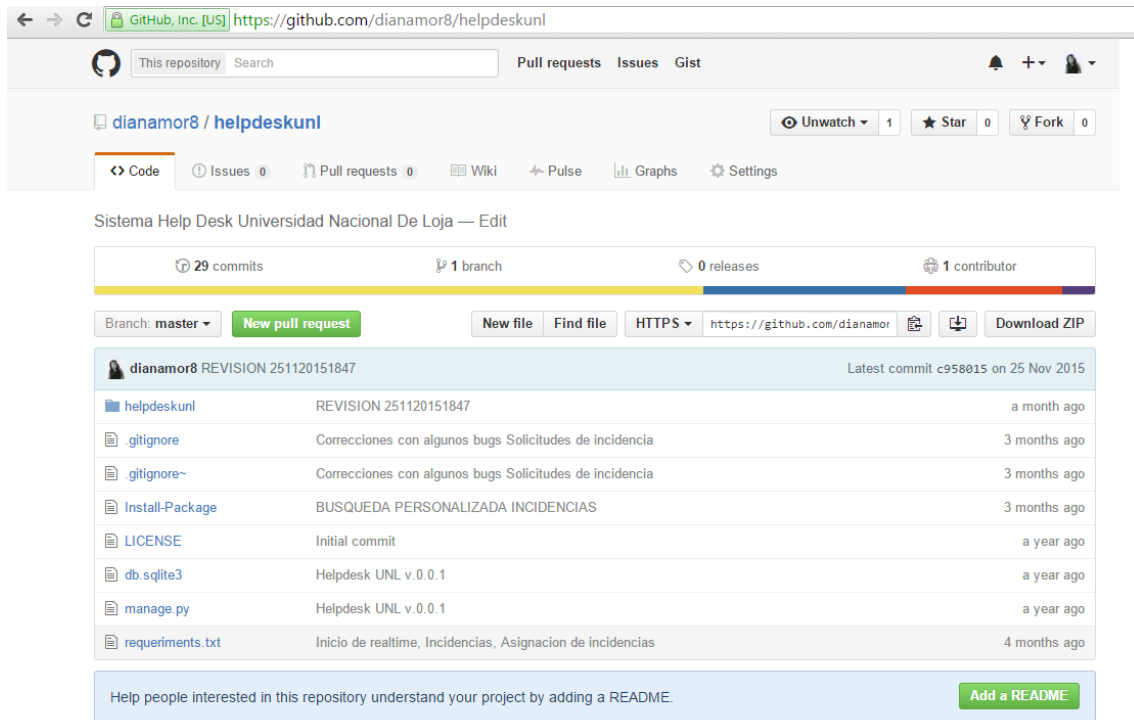


Figura 19. Repositorio Github del proyecto Help Desk.

❖ EJECUCIÓN DEL APLICATIVO

Antes de ejecutar el aplicativo, crear una base de datos en postgresql denominada **helpdeskunl**, y de propietario el rol de usuario **helpdeskunl**, contraseña **helpdeskunl**. Luego, acceder al directorio donde se encuentra el archivo **manage.py** y ejecutar los siguientes comandos para generar las migraciones del proyecto:

>> **python manage.py makemigrations**

>> **python manage.py migrate**

>> **python manage.py runserver localhost:8000**

Finalmente acceder desde el navegador web a la dirección <http://localhost:8000>

Para activar las notificaciones en tiempo real acceder al path de instalación de NodeJs y ejecutar desde una consola el comando:

>> **node node_modules/ishout/server.js**

6.3.1.2. ESTRUCTURA DEL CÓDIGO

El sistema Help Desk está basado en el patrón Model View Template, sin embargo, la configuración principal, está en el archivo **settings.py**. A continuación, en la Figura 20, 21 y 22, se muestra su configuración.

```
2 #encoding:utf-8
3 import os
4 BASE_DIR = os.path.dirname(os.path.dirname(__file__))
5 SECRET_KEY = '(usi7)r_az-w+gf@7u9(892^1a)ji1cf4+sz75m83n&o_67-8)'
6 # SECURITY WARNING: don't run with debug turned on in production!
7 DEBUG = True
8 TEMPLATE_DEBUG = True
9 ALLOWED_HOSTS = ['*']
10 #PERSONALIZACIÓN DE USUARIOS
11 AUTH_USER_MODEL='usuarios.Perfil'
12 #APLICACIONES
13 INSTALLED_APPS = (
14     'django.contrib.admin',
15     'django.contrib.auth',
16     'django.contrib.contenttypes',
17     'django.contrib.sessions',
18     'django.contrib.messages',
19     'django.contrib.staticfiles',
20     'helpdeskunl.apps.usuarios',
21     'helpdeskunl.apps.centro_asistencia',
22     'helpdeskunl.apps.incidencia',
23     'helpdeskunl.apps.home',
24     'helpdeskunl.apps.accion',
25     'drealtime',
26     'dbbackup',
27 )
28 MIDDLEWARE_CLASSES = (
29     #'django.middleware.cache.UpdateCacheMiddleware',
30     'drealtime.middleware.iShoutCookieMiddleware',
31     'django.contrib.sessions.middleware.SessionMiddleware',
32     'django.middleware.common.CommonMiddleware',
33     'django.middleware.csrf.CsrfViewMiddleware',
34     'django.contrib.auth.middleware.AuthenticationMiddleware',
35     'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
36     'django.contrib.messages.middleware.MessageMiddleware',
37     'django.middleware.clickjacking.XFrameOptionsMiddleware',
38     'helpdeskunl.apps.home.current_user.CurrentUserMiddleware',
39     #'django.middleware.cache.FetchFromCacheMiddleware',
40 )
41 #PROCESADORES DE CONTEXTO
42 TEMPLATE_CONTEXT_PROCESSORS = (
43     "django.contrib.auth.context_processors.auth",
44     "django.template.context_processors.i18n",
45     "django.template.context_processors.media",
46     "django.template.context_processors.static",
47     "django.template.context_processors.tz",
48     "django.contrib.messages.context_processors.messages",
49     "helpdeskunl.apps.context_processors.tipo_usuario",
50 )
```

Figura 20. Archivo Settings.py 1 de 3.

```

41 #PROCESADORES DE CONTEXTO
42 TEMPLATE_CONTEXT_PROCESSORS = (
43     "django.contrib.auth.context_processors.auth",
44     "django.template.context_processors.i18n",
45     "django.template.context_processors.media",
46     "django.template.context_processors.static",
47     "django.template.context_processors.tz",
48     "django.contrib.messages.context_processors.messages",
49     "helpdeskunl.apps.context_processors.tipo_usuario",
50 )
51 #CONFIGURACIÓN DE URLS
52 ROOT_URLCONF = 'helpdeskunl.urls'
53 WSGI_APPLICATION = 'helpdeskunl.wsgi.application'
54 #cCONFIGURACIÓN DE BASE DE DATOS
55 DATABASES = {
56     'default': {
57         'ENGINE': 'django.db.backends.postgresql_psycopg2',
58         'NAME': 'helpdesk',
59         'USER': 'helpdeskunl',
60         'PASSWORD': 'helpdeskunl',
61         'HOST': '127.0.0.1',
62         'PORT': '5432',
63     }
64 }
65 #CONFIGURACION DE TEMPLATES
66 TEMPLATE_DIRS = os.path.join(BASE_DIR, 'helpdeskunl/templates'),
67 #CONFIGURACION DE LENGUAJE Y ZONA
68 LANGUAGE_CODE = 'en-EC'
69 TIME_ZONE = 'America/Guayaquil'
70 USE_I18N = True
71 USE_L10N = True
72 STATIC_URL = '/static/'
73
74 #SERVIDOR DE MEDIOS
75 MEDIA_ROOT = os.path.normpath(os.path.join(BASE_DIR, 'helpdeskunl/media/'))
76 MEDIA_URL = '/media/'
77
78 #PARA REDIRECCIONAR
79 LOGIN_URL = '/login/'
80 LOGIN_REDIRECT_URL = '/'
81
82 # CONFIGURACION DE ENVIO DE CORREO
83 EMAIL_HOST = 'smtp.gmail.com'
84 EMAIL_PORT = 587
85 DEFAULT_FROM_EMAIL = 'mihelpdeskunl@gmail.com'
86 EMAIL_HOST_USER = 'mihelpdeskunl@gmail.com'
87 EMAIL_HOST_PASSWORD = 'h3lpd3sk'
88 EMAIL_USE_TLS = True
89 SERVER_EMAIL = 'mihelpdeskunl@gmail.com'

```

Figura 21. Archivo Settings.py 2 de 3.

```

91 LOGGING = {
92     'version': 1,
93     'disable_existing_loggers': True,
94     'formatters': {
95         'verbose': {
96             'format': '[%(levelname)s %(asctime)s %(module)s:%(lineno)s]: %(message)s'},
97         'simple': {
98             'format': '%(levelname)s %(message)s'},
99     },
100     'handlers': {
101         'mail_admins': {
102             'level': 'ERROR',
103             'class': 'django.utils.log.AdminEmailHandler'},
104         'filelogapp': {
105             'level': 'DEBUG',
106             'class': 'logging.FileHandler',
107             'formatter': 'verbose',
108             'filename': '%s%s' % (os.path.dirname(__file__), '/tmp/helpdeskunl.log')},
109     },
110     'loggers': {
111         'django.request': {
112             'handlers': ['mail_admins'],
113             'level': 'ERROR',
114             'propagate': True,},
115         'logapp': {
116             'handlers': ['filelogapp'],
117             'level': 'DEBUG',
118             'propagate': True,},
119     }
120 }
121 #CONFIGURACION DE RESPALDOS
122 DBBACKUP_STORAGE = 'dbbackup.storage.filesystem_storage'
123 DBBACKUP_STORAGE_OPTIONS = {'location': os.path.normpath(os.path.join(MEDIA_ROOT, 'backups/'))}

```

Figura 22. Archivo Settings.py 3 de 3.

El sistema Help Desk dispone de cinco apps que son: Acción, Centro_Asiencia, Home, Incidencia y, Usuarios. Cada una de estas aplicaciones cuenta con la estructura de una app Django es decir los archivos form.py, models.py, reports.py, url.py, views.py. (Ver Figura 23.)



Figura 23. Jerarquía de archivos del código fuente.

❖ EL ARCHIVO MODELS.PY

En este archivo se colocaron las clases del modelo con sus respectivos atributos y métodos. En el ejemplo de la Figura 24 se muestra la clase Centro_Asiencia.

```

44 class Centro_Asiencia(TimeStampedModel):
45     nombre = models.CharField(max_length=100)
46     descripcion = models.CharField(max_length=250)
47     usuarios = models.ManyToManyField(settings.AUTH_USER_MODEL, through='Personal_Operativo', related_name='usuario')
48     email = models.EmailField(max_length=254, blank=True, null=True)
49     contacto = models.CharField(max_length=100, blank=True, null=True)
50
51     objects = Centro_Asiencia_Manager()
52
53     class Meta:
54         verbose_name = "Centro de Asistencia"
55         verbose_name_plural = "Centros de Asistencia"
56         db_table = 'Centro_Asiencia'
57     def __unicode__(self):
58         return u'%s'%(self.nombre)
59     def get_absolute_url(self):
60         return '/centro_asistencia/%i'%(self.id)
61     def get_administradores(self):
62         # administradores = Perfil.jefes_departamento.filter(personal_operativo__centro_asistencia = self).distinct()
63         # return administradores
64         personal_operativo = self.personal_operativo_set.filter(grupo__name='JEFE DEPARTAMENTO')
65         administradores = list()
66         for personal in personal_operativo:
67             administradores.append(personal.usuario)
68         return administradores
69     def get_asesores(self):
70         # asesores = Perfil.asesores_tecnicos.filter(personal_operativo__centro_asistencia = self).distinct()
71         personal_operativo = self.personal_operativo_set.filter(grupo__name='ASESOR TECNICO')
72         asesores = list()
73         for personal in personal_operativo:
74             asesores.append(personal.usuario)
75         return asesores
  
```

Figura 24. Archivo models.py.

❖ EL ARCHIVO VIEWS.PY

En este archivo se colocaron las views con sus respectivos controles. En el ejemplo de la Figura 25 se muestra el controlador para crear objetos de tipo Centro_Asiencia.

```
105
106 class Centro_AsienciaCreate(CreateView):
107     model = Centro_Asiencia
108     template_name = 'centro_asistencia/centro_asistencia_edit_form.html'
109     form_class = form_agregar_centro_asistencia
110
111     def form_valid(self, form):
112         centro_existente = Centro_Asiencia.objects.filter(nombre=form.cleaned_data['nombre'])
113         if centro_existente:
114             raise forms.ValidationError('Ya existe este centro')
115
116         self.object = form.save()
117         LogEntry.objects.log_action(
118             user_id = self.request.user.pk,
119             content_type_id = ContentType.objects.get_for_model(self.object).pk,
120             object_id = self.object.pk,
121             object_repr = force_unicode(self.object),
122             action_flag = ADDITION,
123             change_message = 'Crea nuevo centro de asistencia'
124         )
125         centro_asistencia = self.object
126         usuario = self.request.user
127         if usuario.has_perm('centro_asistencia.change_centro_asistencia'):
128             link = '<a href="{centro_asistencia.get_absolute_url()}+>%s</a>' % (centro_asistencia.nombre)
129             editar = '<div class="col-sm-5"><a data-toggle="modal" href="/centro_asistencia/edit/%s" data-target="#modal_centro
130         else:
131             link = '%s' % (centro_asistencia.nombre)
132             editar = ''
133         if usuario.has_perm('centro_asistencia.delete_centro_asistencia'):
134             eliminar = '<div class="col-sm-5"><a data-toggle="modal" href="/centro_asistencia/%s/delete" data-target="#delele_m
135         else:
136             eliminar = ''
137         if self.request.is_ajax():
138             fila = '<tr id="tr_centro_asistencia%s"><td>%s</td><td> %s</td><td><div class="row-fluid">%s %s</div></td></tr>' % (
139                 ctx = {'respuesta': 'create', 'fila': fila, 'id': centro_asistencia.id,}
140             return HttpResponse(json.dumps(ctx), content_type="application/json")
```

Figura 25. Archivo views.py.

❖ EL ARCHIVO FORMS.PY

En este archivo se colocaron los formularios a renderizar en los archivos .html. Cada archivo form.py posee métodos específicos para validar los fields. En el ejemplo de la Figura 26 se muestra el formulario para actualizar objetos de tipo Centro_Asiencia.

```
46 class Centro_Asiencia_UpdateForm(ModelForm):
47     class Meta:
48         model = Centro_Asiencia
49         fields = 'nombre', 'descripcion', 'email', 'contacto',
50         labels = {
51             'nombre': ('Nombre:'),
52             'descripcion': ('Actividad:'),
53             'email': ('Correo Electrónico:'),
54             'contacto': ('Contacto:'),
55         }
56         error_messages = {
57             'nombre': {'required': u"Este campo no puede estar vacío."},
58             'descripcion': {'required': u"Este campo no puede estar vacío."},
59             'email': {'invalid': u"Ingresa un valor válido."},
60             'NON_FIELD_ERRORS': {'duplicado': u"Ya existe un centro de asistencia con este nombre."},
61         }
62         widgets = {
63             'nombre': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Nombre del centro de asistencia.'}),
64             'descripcion': forms.Textarea(attrs={'class': 'form-control', 'placeholder': 'Descripción del centro de asistencia.'}),
65             # 'administradores': forms.Select(attrs={'class': 'form-control required'}),
66             'email': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Dirección de correo electrónico de contacto.'}),
67             'contacto': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Números de contacto.'}),
68         }
69
70     def registro_nombre_duplicado(self, id_centro):
71         nombre = self.cleaned_data['nombre']
72         try:
73             centro_existente = Centro_Asiencia.objects.get(nombre=nombre)
74         except Centro_Asiencia.DoesNotExist:
75             centro_existente = None
76
77         if centro_existente:
78             if centro_existente.id != id_centro:
79                 raise forms.ValidationError((self.Meta.error_messages['NON_FIELD_ERRORS'])['duplicado'], code='NON_FIELD_ERRORS')
```

Figura 26. Archivo forms.py.

❖ EL ARCHIVO URLS.PY

En este archivo se colocaron las urls del proyecto relacionadas con los controladores del archivo views.py. (Ver Figura 27.)

```
from django.conf.urls import patterns, url
from helpdeskunl.apps.centro_asistencia.views import *

urlpatterns = patterns('helpdeskunl.apps.centro_asistencia',
    # VIEW
    url(r'^lista/centro_asistencia/$', 'views.lista_centro_asistencia', name = 'view_lista_centro_asistencia'),

    # LIST, DETAIL, CREATE, CHANGE, DELETE VIEWS
    # CENTRO DE ASISTENCIAAw
    url(r'^centro_asistencia/(?P<pk>[-\d]+)/$', CentroAsistenciaDetailView.as_view(), name='view_centro_asistencia_detail'),
    url(r'^listageneral/centro_asistencia/$', CentroAsistenciaGeneral.as_view(), name='centro_asistencia_list'),
    url(r'^centro_asistencia/edit/(?P<pk>\d+)/$', CentroAsistenciaUpdate.as_view(), name='centro_asistencia_update'),
    url(r'^centro_asistencia/add$', CentroAsistenciaCreate.as_view(), name='centro_asistencia_add'),
    url(r'^centro_asistencia/(?P<pk>\d+)/delete/$', CentroAsistenciaDelete.as_view(), name='centro_asistencia_delete'),

    #SERVICIOS CENTRO DE ASISTENCIA
    url(r'^servicio/add/(?P<centro>\d+)$', ServicioCreate.as_view(), name='servicio_add'),
    url(r'^servicio/(?P<pk>\d+)/$', ServicioUpdate.as_view(), name='servicio_update'),
    url(r'^servicio/(?P<pk>\d+)/delete/$', ServicioDelete.as_view(), name='servicio_delete'),

    url(r'^estadisticas/list$', EstadisticaList.as_view(), name='estadistica_list'),
    url(r'^estadistica/add$', EstadisticaCreate.as_view(), name='estadistica_add'),
    url(r'^estadistica/edit/(?P<pk>\d+)/$', EstadisticaUpdate.as_view(), name='estadistica_update'),
    url(r'^estadistica/(?P<pk>\d+)/delete/$', EstadisticaDelete.as_view(), name='estadistica_delete'),

    # REST
    # url(r'^centro_asistencia/add$', 'rest.centro_asistencia_add_ajax', name = 'view_centro_asistencia_add'),
    url(r'^centro_asistencia/add_usuario$', 'rest.agregar_usuario', name = 'view_agregar_usuario_centro_asistencia'),
    url(r'^centro_asistencia/remove_usuarios$', 'rest.eliminar_usuario', name = 'view_eliminar_usuario_centro_asistencia'),
    url(r'^rest/centro_asistencia/verificar_duracion$', 'rest.verificar_duracion', name = 'view_verificar_duracion_centro_asistencia'),
    url(r'^rest/centro_asistencia/verificar_duracion/estadisticas$', 'rest.verificar_duracion_dos', name = 'view_verificar_duracion_dos'),
    url(r'^rest/centro_asistencia/verificar_estadistica$', 'rest.verificar_estadistica', name = 'view_verificar_estadistica_centro_asistencia'),
    url(r'^rest/get_notificaciones$', 'rest.get_notificaciones', name = 'view_get_notificaciones_centro_asistencia'),

    url(r'^rest/get_asesores$', 'rest.get_asesores_centro', name = 'view_asesores_centro'),
    url(r'^rest/get_slas$', 'rest.get_slas_centro', name = 'view_slas_centro'),
```

Figura 27. Archivo urls.py.

❖ EL ARCHIVO TEMPLATE.HTML

En este archivo se colocó el código html de las interfaces de usuario.

```
{% url 'centro_asistencia_add' as url_add_centro_asistencia %}
<form id="centro_asistencia_create_form" method="post" class="form form-horizontal" role="form" action="{url_add_centro_asistencia}">
  {{url_add_centro_asistencia}}
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden="true"></button>
    <h4 class="modal-title" id="myModalLabel">
      <strong>Nuevo Centro de Asistencia</strong>
    </h4>
  </div>
  {% endif %}
  <div class="modal-body">
    {% csrf_token %}
    {% for formulario in form %}
      {% if formulario.errors %}
        <div class="alert alert-warning alert-dismissible">
          <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
          {% for error in formulario.errors %}
            <small>{{error}}</small>
          {% endfor %}
        </div>
      {% endif %}
      <div class="form-group">
        <label class="control-label col-sm-3">{{formulario.label}}</label>
        <div class="col-sm-8">
          {{formulario}}
          <p class="help-block small">{{formulario.help_text}}</p>
        </div>
      </div>
    {% endfor %}
  </div>
  <div class="modal-footer">
    <input type="button" class="btn btn-default" data-dismiss="modal" value="Cancelar" />
    <input type="submit" class="btn btn-primary" value="Guardar" style="margin-bottom: 5px;" />
  </div>
</form>
```

Figura 28. Archivo crear_centro_asistencia.html

6.3.2. EJECUCIÓN DEL PLAN DE ITERACIONES

Seguidamente, se ejecutó el plan de iteraciones definido en el Apartado 6.1.7. del plan de iteraciones

6.3.2.1. ITERACIÓN PE-001. CONFIGURACIÓN INICIAL Y PRUEBA DE HERRAMIENTAS

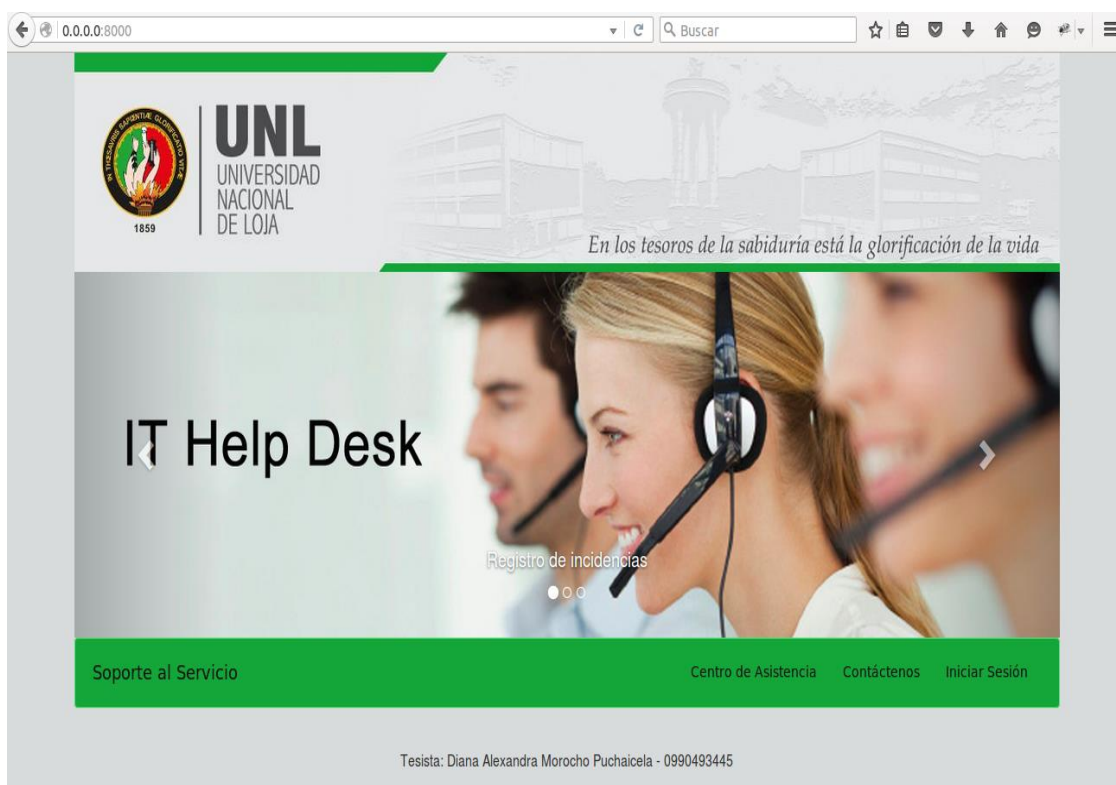


Figura 29. Configuración Inicial.

CONFIGURACIÓN INICIAL Y PRUEBA DE HERRAMIENTAS	
CÓDIGO:	PE-001
ACTORES INVOLUCRADOS:	Usuario Final, Asesor Técnico, Jefe Departamento, Administrador.
CASOS DE USO:	Proceso de configuración. Sin casos de uso aplicados.
REFERENCIAS:	RNF-002, RNF-003
DETALLE:	<p>En ésta iteración, se configuró:</p> <ul style="list-style-type: none"> - Las herramientas de desarrollo. - Clasificación de colores en los templates de acuerdo al rol de usuario.

- Diseño del sidebar y funciones de acceso.	
PRUEBAS DE ACEPTACION NO SUPERADAS:	<ul style="list-style-type: none"> - Tipos de roles de usuario bien definidos. - Aprobación de la gama de colores.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACION ANTERIOR:	Es la primera iteración.

Tabla 31. Iteración PE-001.

6.3.2.2. ITERACIÓN PE-002. ADMINISTRACIÓN DE CENTROS DE ASISTENCIA – PRIMERA ITERACIÓN.

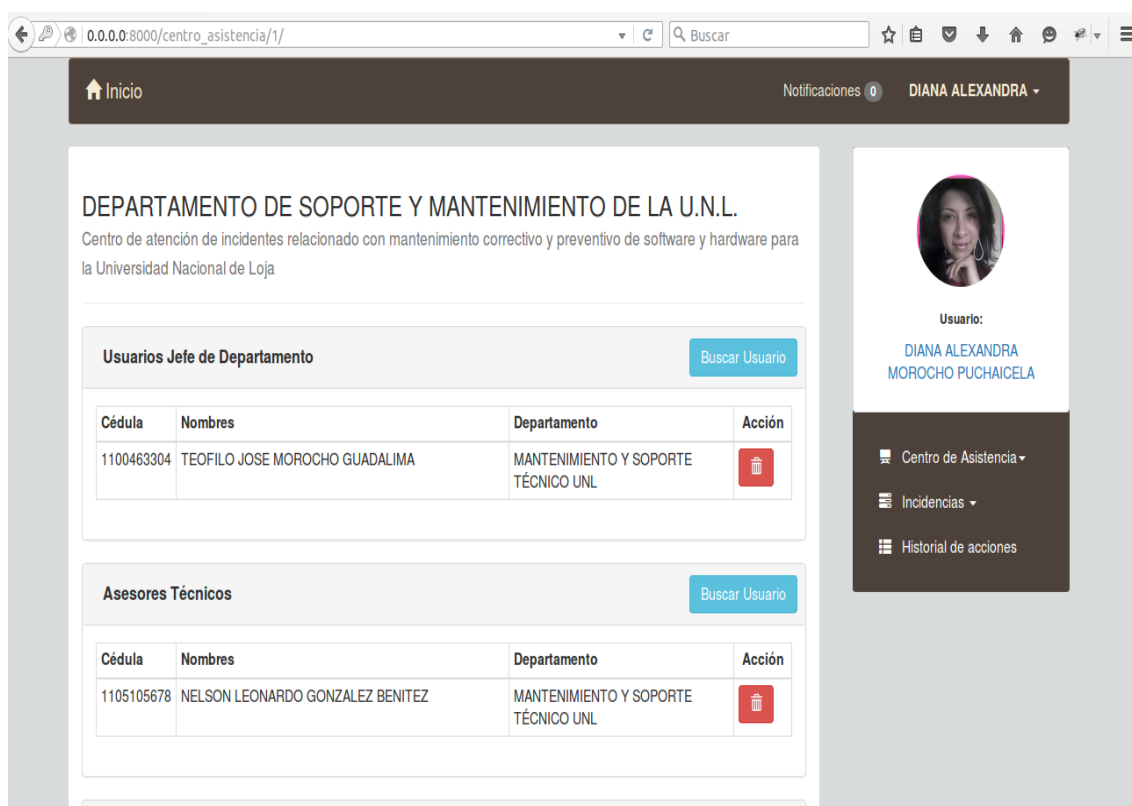


Figura 30. Administración de Centros de Asistencia.

ADMINISTRACIÓN DE CENTROS DE ASISTENCIA IT-1	
CÓDIGO:	PE-002
ACTORES INVOLUCRADOS:	Jefe Departamento, Asesor Técnico, Administrador.
CASOS DE USO:	Administrar centro de asistencia, Administrar SLA

REFERENCIAS:	RF-003, RF-004.
DETALLE:	
La iteración, consistió en desarrollar los principales parámetros para el centro de asistencia, como son las funciones de crear, actualizar y eliminar. Además, agregar usuarios de tipo Asesor Técnico y Jefe Departamento al centro de asistencia respectivamente.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	<ul style="list-style-type: none"> - El borrado físico de centros de asistencia debe ser modificado por borrado lógico. - Almacenado de centros de asistencia bajo el mismo nombre.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Acceso personalizado según el rol de usuario del sidebar. - Definición de cuatro roles de usuario principal para el acceso al sistema. - Aceptación de la gama de colores para el panel en los diferentes tipos de usuario. - Corrección de todas las pruebas de aceptación no superadas.

Tabla 32. Iteración PE-002

6.3.2.3. ITERACIÓN PE-003. ADMINISTRACIÓN DE ACUERDOS DE NIVEL DE SERVICIO EN CENTROS DE ASISTENCIA - SEGUNDA ITERACIÓN.

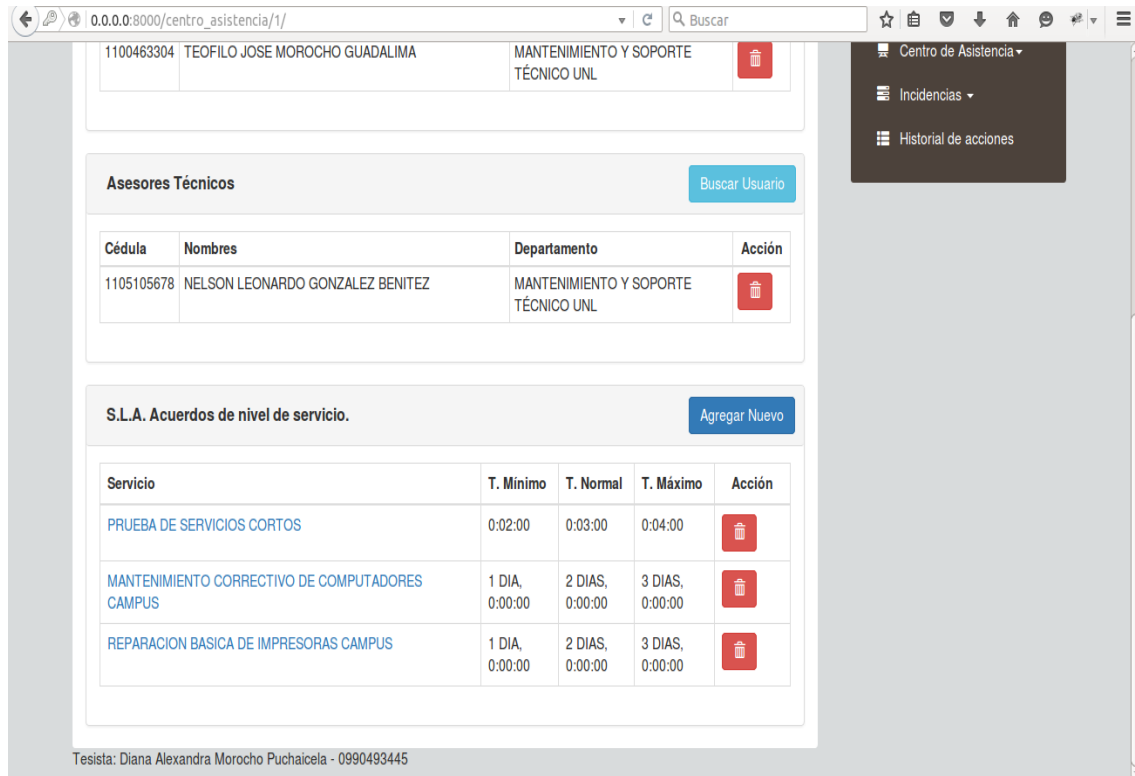


Figura 31. Administración de acuerdos de nivel de servicio.

ADMINISTRACIÓN DE ACUERDOS DE NIVEL DE SERVICIO IT-2	
CÓDIGO:	PE-003
ACTORES INVOLUCRADOS:	Jefe Departamento, Asesor Técnico, Administrador.
CASOS DE USO:	Administrar centro de asistencia, Administrar SLA
REFERENCIAS:	RF-003, RF-004.
DETALLE:	
El objetivo principal de ésta iteración fue crear, actualizar, eliminar acuerdos de nivel de servicio, estructurados en tres tiempos: mínimo, normal y máximo. Éste registro permitió establecer de acuerdo y con la urgencia asignada la prioridad de atención de la incidencia.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	- No se ha establecido una estadística de rangos mínimos y

	máximos para el acuerdo de nivel de servicio.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Borrado lógico de centros de asistencia, e impedimento de borrado cuando se han reportado incidencias en ese centro. - Almacenamiento de centros de asistencia con nombre único.

Tabla 33. Iteración PE-003

6.3.2.4. ITERACIÓN PE-004. ADMINISTRACIÓN DE USUARIOS – ACCESO AL SISTEMA – PRIMERA ITERACIÓN.

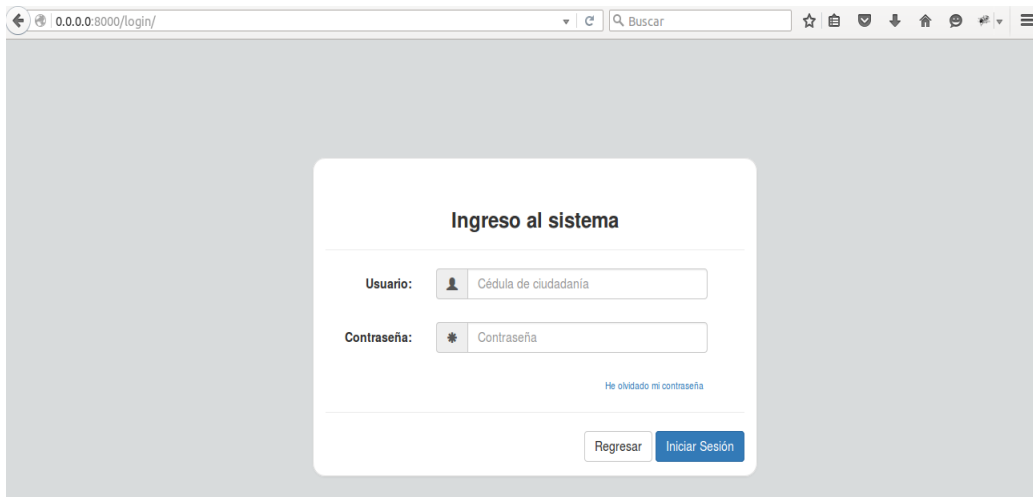


Figura 32. Administración de usuarios.

ADMINISTRACIÓN DE USUARIOS	
CÓDIGO:	PE-004
ACTORES INVOLUCRADOS:	Usuario Final, Asesor Técnico, Jefe de Departamento, Administrador.
CASOS DE USO:	Administrar usuarios.
REFERENCIAS:	RF-001, RF-002, RF-011.
DETALLE:	Esta iteración está diseñada para restringir el acceso a usuarios no registrados en el sistema. Permite autenticar al usuario y renderizar el contenido del panel de acuerdo a los roles de usuario.

HISTORIAS DE USUARIO NO ABORDADAS:	002.
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	- Verificación de cédula no válida.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	- Definición de estadística de SLA. - Corrección de las pruebas de aceptación no superadas.

Tabla 34. Iteración PE-004.

6.3.2.5. ITERACIÓN PE-005. SOLICITUD DE INCIDENCIA – PRIMERA ITERACIÓN.

Figura 33. Solicitud de incidencia. Iteración 1.

SOLICITUD DE INCIDENCIA IT-1	
CÓDIGO:	PE-005
ACTORES INVOLUCRADOS:	Usuario Final
CASOS DE USO:	Administrar Incidencia.
REFERENCIAS:	RF-005, RF-006, RF-012.

DETALLE:	
En ésta iteración, el usuario final pudo generar incidencias y reportar equipos o bienes para la atención. Registró una definición general del problema, lo describe, selecciona un nivel de prioridad de acuerdo a la urgencia requerida en la solución del problema, selecciona el centro de asistencia al que se dirige la incidencia y finalmente se adjunta alguna imagen en caso de ser requerido.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	<ul style="list-style-type: none"> - Registro de bienes a reportar. - Configuración del servidor de medios para las imágenes.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Desarrollo del algoritmo de cédula válida. - Corrección de pruebas de aceptación no superadas.

Tabla 35. Iteración PE-005

6.3.2.6. ITERACIÓN PE-006. SOLICITUD DE INCIDENCIA – SEGUNDA ITERACION

The screenshot displays a web application interface for incident management. At the top, there is a green navigation bar with 'Inicio', 'Notificaciones 4', and the user name 'ANGELES ROCIO'. The main content area is titled 'Detalle de Incidencia' and shows the following information:

- PAPEL ATASCADO** (with a 'Calificar la atención' button)
- Centro de Asistencia: DEPARTAMENTO DE SOPORTE Y MANTENIMIENTO DE LA U.N.L.
- Técnicos Asignados: NELSON LEONARDO GONZALEZ BENITEZ.
- Estado: Finalizada
- Diagnostico Inicial: IMPRESORA CON PAPEL ATASCADO
- Acciones realizadas: Mostar 10 resultados por página. Buscar: [input field]

Fecha	Título	Realizador por
11/20/2015 13:42	LIMPIEZA Y DESATASCO	NELSON LEONARDO GONZALEZ BENITEZ

Below the table are navigation buttons: Anterior, 1, and Siguiente. At the bottom, it states: 'Solicitud de recursos: Aún no se ha registrado solicitudes de recurso.'

Footer: Tesista: Diana Alexandra Morocho Puchacela - 0990493445

Figura 34. Solicitud de incidencia. Iteración 2.

SOLICITUD DE INCIDENCIA IT-2	
CÓDIGO:	PE-006
ACTORES INVOLUCRADOS:	Usuario Final.
CASOS DE USO:	Administrar Incidencia.
REFERENCIAS:	RF-005, RF-006
DETALLE:	
En ésta iteración, se implementó una funcionalidad que permita al usuario visualizar el estado de la incidencia generada.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	Ninguna.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Configuración del servidor de medios. - Corrección de las pruebas de aceptación no superadas.

Tabla 36. Iteración PE-006.

6.3.2.7. ITERACIÓN PE-007. ASIGNACIÓN DE INCIDENCIAS – PRIMERA ITERACIÓN

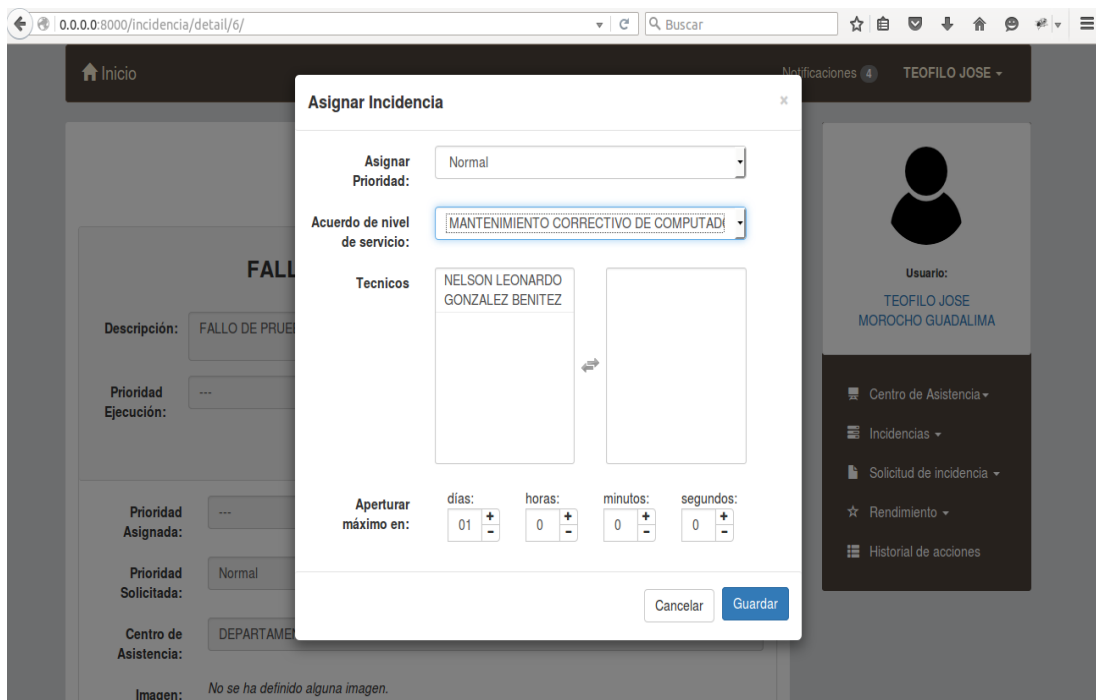


Figura 35. Asignación de Incidencias. Iteración 1.

ASIGNACIÓN DE INCIDENCIAS IT-1	
CÓDIGO:	PE-007
ACTORES INVOLUCRADOS:	Asesor Técnico, Jefe Departamento.
CASOS DE USO:	Administrar incidencia, gestionar prioridad.
REFERENCIAS:	RF-008, RF-013.
DETALLE:	
En ésta iteración, se pretende asignar a uno o varios asesores técnicos la atención de incidencias, seleccionando la prioridad que se asignará, además de escoger el acuerdo de nivel de servicio y determinando el tiempo de apertura máxima para atender la incidencia.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	- Verificación del algoritmo de cálculo de prioridades.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	Ninguna.

Tabla 37. Iteración PE-007.

6.3.2.8. ITERACIÓN PE-008. ASIGNACIÓN DE INCIDENCIAS – SEGUNDA ITERACIÓN

Fecha: Nov. 26, 2015, 6:07 a.m.

Usuario: TEOFILO JOSE MOROCHO GUADALIMA

Bienes: Opciones. Seleccione una opción de búsqueda.

Código Código Sist.Finan. Serie Producto

Prioridad Asignada: Normal

Servicio. SLA.: MANTENIMIENTO CORRECTIVO DE COMPUTADORES CAMPUS

Duración: 2 00:00:00

Asesores Técnicos: NELSON LEONARDO GONZALEZ BENITEZ

Seleccione los técnicos que atenderán la incidencia.

Aperturar máximo en: días: 1 horas: 0 minutos: 0 segundos: 0

Figura 36. Asignación de Incidencias. Iteración 2.

ASIGNACIÓN DE INCIDENCIAS IT-2	
CÓDIGO:	PE-008
ACTORES INVOLUCRADOS:	Jefe Departamento.
CASOS DE USO:	Administrar incidencia, gestionar prioridad.
REFERENCIAS:	RF-008, RF-013.
DETALLE:	
Ésta iteración, permitió administrar la incidencia luego de la asignación.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	<ul style="list-style-type: none"> - Re cálculo de duración de incidencia de acuerdo al acuerdo de nivel de servicio. - Cambio de asignación de Asesor Técnico.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Desarrollo del análisis del cálculo de asignación de prioridad. - Corrección de todas las pruebas de aceptación no superadas.

Tabla 38. Iteración PE-008.

6.3.2.9. ITERACIÓN PE-009. ATENDER INCIDENCIA

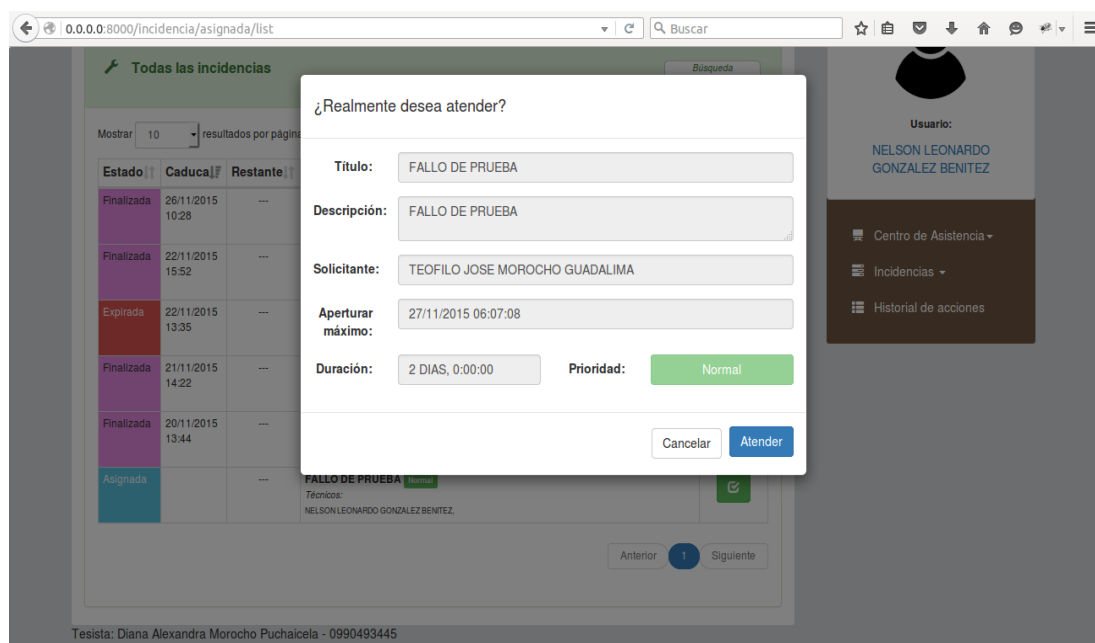


Figura 37. Atender Incidencias.

ATENDER INCIDENCIAS	
CÓDIGO:	PE-009
ACTORES INVOLUCRADOS:	Asesor Técnico.
CASOS DE USO:	Administrar Incidencia.
REFERENCIAS:	RF-009,RF-013, RF-016
DETALLE:	
En ésta iteración, se implementó la función que permitió al usuario Asesor Técnico la capacidad de iniciar la atención de la incidencia que le ha sido asignada.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	Ninguna.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Al modificar acuerdos de nivel de servicio se realiza una nueva valoración de la duración de la incidencia, bajo determinadas condiciones. - Cambio de asignación de Asesor Técnico. - Corrección de todas las pruebas de aceptación no superadas.

Tabla 39. Iteración PE-009

6.3.2.10. ITERACIÓN PE-010. REGISTRO DE EQUIPO – PRIMERA ITERACIÓN

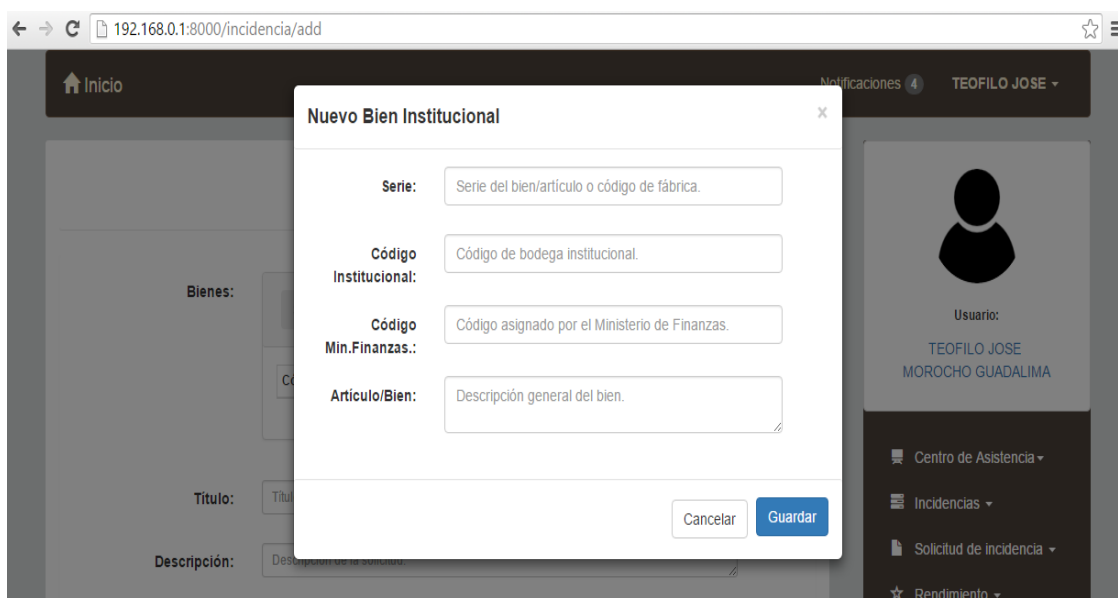


Figura 38. Registro de Equipo.

REGISTRO DE EQUIPO	
CÓDIGO:	PE-010
ACTORES INVOLUCRADOS:	Usuario Final, Asesor Técnico, Jefe Departamento, Administrador.
CASOS DE USO:	Administrar equipos.
REFERENCIAS:	RF-012
DETALLE:	
En esta iteración se implementó el módulo de bienes, cuyo objetivo fundamental, es permitir reportar equipos con detalle en los incidentes. Para la Universidad Nacional de Loja, se ha utilizado tres tipos de codificación: número de serie, código institucional, y el código otorgado por el Ministerio de Finanzas del Ecuador.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	Ninguna.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	Ninguna.

Tabla 40. Iteración PE-010.

6.3.2.11. ITERACIÓN PE-011. MEDICIÓN DE LA CALIDAD DE SERVICIO – PRIMERA ITERACIÓN

Figura 39. Calidad de servicio. Iteración 1.

MEDICIÓN DE CALIDAD DE SERVICIO IT-1	
CÓDIGO:	PE-011
ACTORES INVOLUCRADOS:	Jefe Departamento, Asesor Técnico.
CASOS DE USO:	---
REFERENCIAS:	RF-007, RF-014
DETALLE:	
Ésta iteración, permitió medir la calidad del servicio prestado por el usuario Asesor Técnico, se selecciona los parámetros de validación y se presiona el botón Generar. Previo a la medición, se debe realizar la evaluación de las incidencias.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	- Creación de los modelos de cuestionario, preguntas, y evaluación.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	Ninguna.

Tabla 41. Iteración PE-011.

6.3.2.12. ITERACIÓN PE-012. MEDICIÓN DE LA CALIDAD DE SERVICIO - EVALUACIÓN – SEGUNDA ITERACIÓN

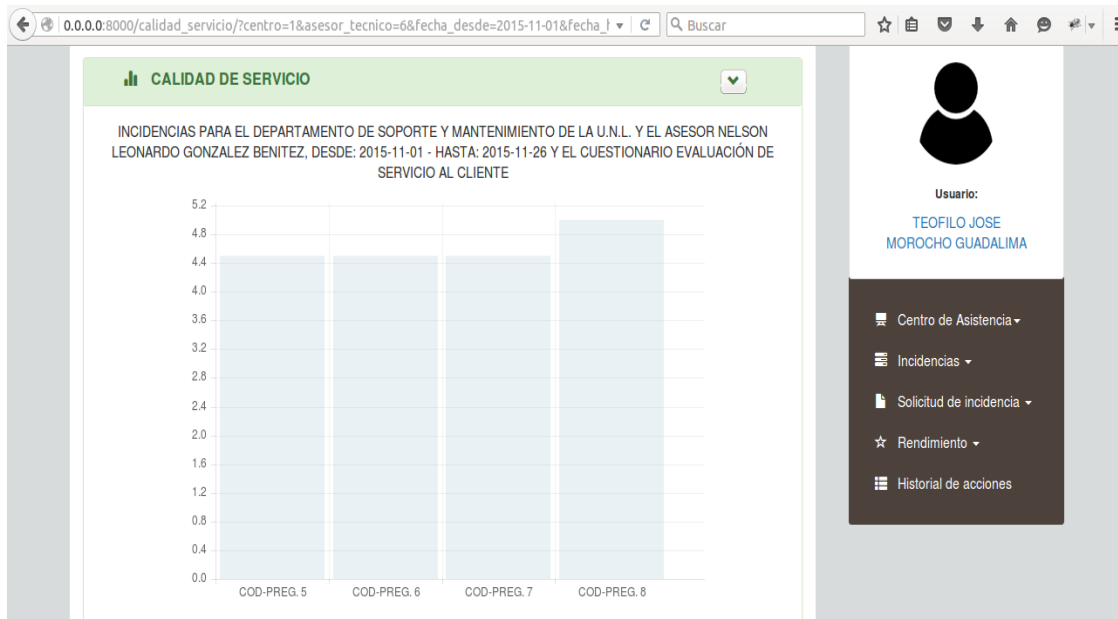


Figura 40. Calidad de servicio. Iteración 2. 1 de 3.

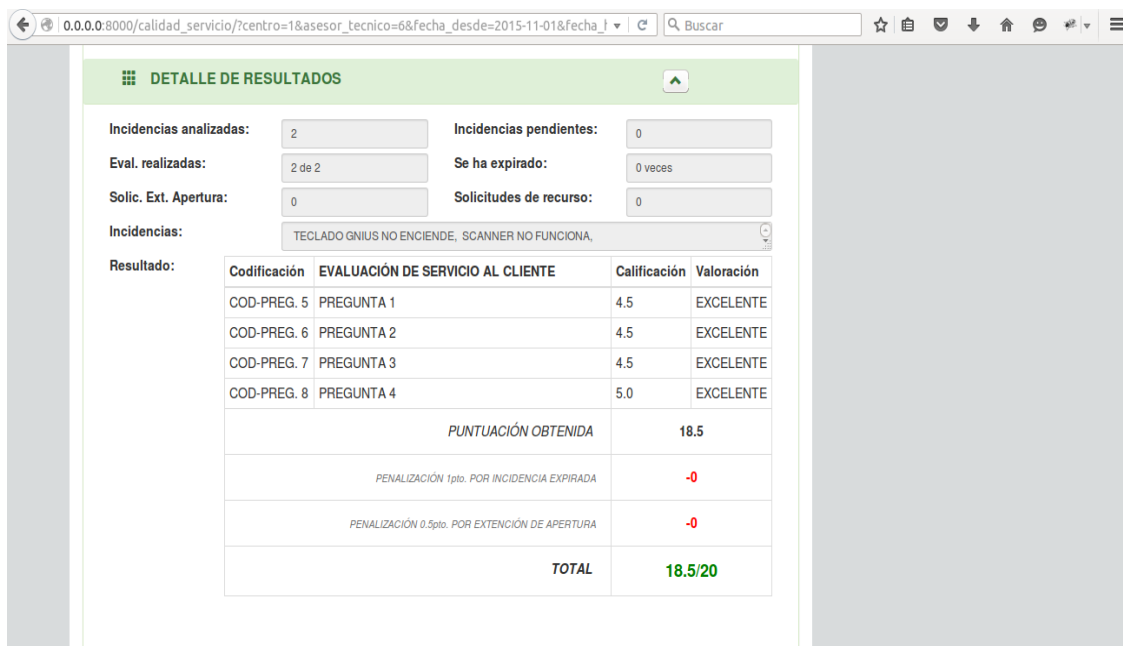


Figura 41. Calidad de servicio. Iteración 2. 2 de 3.

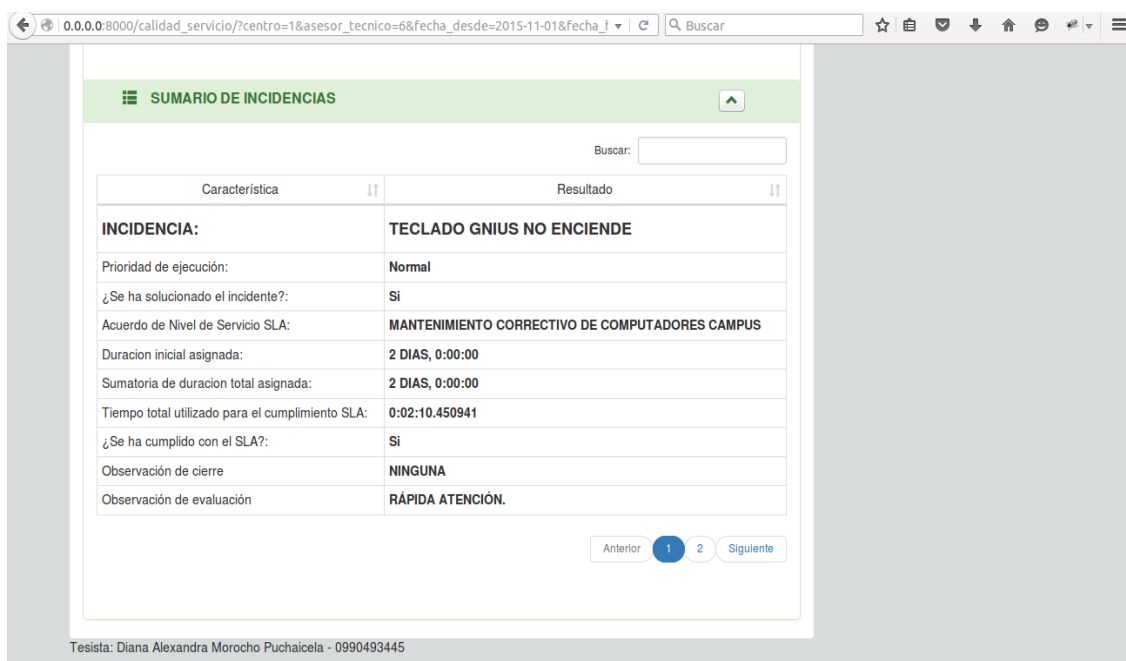


Figura 42. Calidad de servicio. Iteración 2. 3 de 3.

MEDICIÓN DE CALIDAD DE SERVICIO IT-2	
CÓDIGO:	PE-012
ACTORES INVOLUCRADOS:	Jefe Departamento, Asesor Técnico.
CASOS DE USO:	---

REFERENCIAS:	RF-007, RF-014
DETALLE:	
Permitió visualizar el resultado los datos seleccionados.	
PRUEBAS DE ACEPTACIÓN NO SUPERADAS:	Ninguna.
TAREAS REALIZADAS Y NO TERMINADAS EN LA ITERACIÓN ANTERIOR:	<ul style="list-style-type: none"> - Se crea el modelo, vista y templates referentes al cuestionario, preguntas, y evaluación, para poder obtener los datos de medición de calidad de servicio. - Corrección de las prueba de aceptación no superadas.

Tabla 42. Iteración PE-012.

6.4. FASE DE PRUEBAS

XP enfatiza en la realización de un sinnúmero de pruebas a lo largo del proyecto, con el fin de asegurar en todo momento la realización de lo planteado en el diseño. En ésta fase, se estableció el plan de validación y verificación y, su ejecución.

6.4.1. PLAN DE VALIDACIÓN Y VERIFICACIÓN

El plan de verificación y validación de software se define como un conjunto de procedimientos, actividades, técnicas y herramientas que se utilizan paralelamente al desarrollo, para asegurar el cumplimiento de los requerimientos. [10] A continuación se muestra el plan de verificación y validación aplicado en éste proyecto.

6.4.1.1. INTRODUCCIÓN

❖ PROPÓSITO

El plan de validación y verificación para el proyecto “Sistema Help Desk para el Departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja” propone el cumplimiento de los siguientes objetivos:

- Identificar los requerimientos de software que deben ser sometidos al proceso de verificación y validación.

- Describir las estrategias de verificación utilizadas para las pruebas de: funcionalidad, interfaz de usuario y, carga.
- Identificar el personal humano y los roles de usuario que fueron necesarios para el proceso de verificación y validación.
- Ejecutar las pruebas planteadas.

❖ PUNTO DE PARTIDA

El sistema Help Desk, tiene como principal objetivo atender las incidencias referentes al mantenimiento de hardware y software de equipos pertenecientes a la institución y que se reportan por fallas, testeo, etc.

Para esto, el sistema posee una arquitectura cliente servidor y cuenta con múltiples funcionalidades; sin embargo el núcleo fundamental se encuentra en los módulos especializados para gestionar: acuerdos de nivel de servicio, centro de asistencia e incidentes, de aquí que el objetivo de validar y verificar, fue encontrar la mayor cantidad de errores para implementar un software de calidad.

❖ ALCANCE

Las pruebas de funcionalidad, interfaz de usuario y de carga están bajo la responsabilidad del equipo de desarrollo y se aplican únicamente a las funcionalidades del sistema Help Desk (Ver el apartado 6.4.1.2. Requerimientos para verificar).

Quedan exentos de validación y verificación:

- Los requerimientos no funcionales, puesto que éstos obedecen a las características que posee el sistema en cuanto a portabilidad, diseño, escalabilidad, y seguridad del sistema.
- La tecnología con que se desarrolló el software, debido a que esto implica poseer un profundo conocimiento de cada herramienta e incrementar el tiempo para validar y verificar en diferentes escenarios.
- La navegación en varios browsers, debido a que ésta característica obliga a que las pruebas funcionales se realicen varias veces, una vez en cada navegador, relenteciendo el proceso de verificación.

❖ IDENTIFICACIÓN DEL PROYECTO

El artefacto más útil para ejecutar el Plan de Validación, es la Tabla de Requerimientos Funcionales (Ver Tabla 21.), y el Diagrama de Casos de Uso (Ver Figura 11.) de éste documento.

6.4.1.2. REQUERIMIENTOS PARA VERIFICAR

Seguidamente, aquí se define qué prueba aplicar al conjunto de características que son objeto de validación y verificación.

PRUEBA DE FUNCIONALIDAD	
Característica	Resultado Esperado
Por interfaz - acceso al sistema.	Acceder al sistema.
Por interfaz - administrar centros de asistencia.	Crear, actualizar, eliminar y buscar con éxito centros de asistencia
Por interfaz - asignar usuarios a centros de asistencia.	Asignar usuarios a un centro de asistencia determinado.
Por interfaz - administrar acuerdos de nivel de servicio.	Crear, actualizar, eliminar y buscar con éxito acuerdos de nivel de servicio.

Tabla 43. Características evaluadas con la prueba de funcionalidad.

PRUEBA DE INTERFAZ DE USUARIO	
Característica	Resultado Esperado
Administrar incidencia (Con el rol de usuario final)	Mostrar el resultado obtenido de la aplicación de una encuesta (ver anexo 10) sección única al <i>Usuario Final</i> .
Asignación de incidencia	Mostrar el resultado obtenido de la aplicación de una encuesta (ver anexo 10) sección única al usuario <i>Jefe Departamento</i> .
Atención de incidencia	Mostrar el resultado obtenido de la aplicación de una encuesta (ver anexo 10) sección única al usuario <i>Asesor Técnico</i> .
Administrar acciones de incidencia	
Administrar sistema Help Desk	Mostrar el resultado obtenido de la aplicación de una encuesta (ver anexo 10) sección única al usuario <i>Administrador</i> .

Tabla 44. Características evaluadas con la prueba de interfaz de usuario.

PRUEBA DE CARGA	
Característica	Resultado Esperado
URL de Inicio	Determinar la continuidad del funcionamiento del Sistema
URL lista de incidencias	Help Desk bajo la carga de 100, 500 y 1000 request.

Tabla 45. Características evaluadas con la prueba de carga.

6.4.1.3. ESTRATEGIA DE VERIFICACIÓN

En la Tabla 46 se presenta el enfoque recomendado para determinar cómo validar y verificar las funcionalidades del sistema. Se indica el objetivo, técnica y criterio de aceptación, para determinar cuándo una prueba se completó.

	Objetivo	Técnica	Criterio de aceptación
PRUEBA DE FUNCIONALIDAD	Asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, y proceso.	Ejecutar los casos de uso usando datos válidos y no válidos, para verificar que se obtienen los resultados y se despliegan los mensajes de error o advertencia apropiados.	Todas las pruebas establecidas en éste plan se realizaron y los defectos encontrados han sido debidamente identificados y corregidos.
PRUEBA INTERFAZ DE USUARIO	Verificar que la navegación a través de los elementos que se están probando refleje las funciones del negocio y los requerimientos.	Ejecutar el sistema Help Desk para verificar la navegabilidad y los objetos dentro de cada ventana, y realizar una encuesta a los principales usuarios del sistema.	Cada ventana ha sido verificada exitosamente y los defectos encontrados han sido debidamente identificados y corregidos.
PRUEBA DE CARGA	Verificar que el sistema mantendrá la continuidad de funcionamiento bajo condiciones de carga importantes.	Levantar el servicio del sistema Help Desk y enviar 100, 500, 1000 solicitudes de request mediante el software Apache Bench	El sistema no haya dejado de funcionar bajo las condiciones de carga, y visualizar el resumen del tiempo de respuesta de los casos respectivamente.

Tabla 46. Estrategia de verificación.

6.4.1.4. RECURSOS

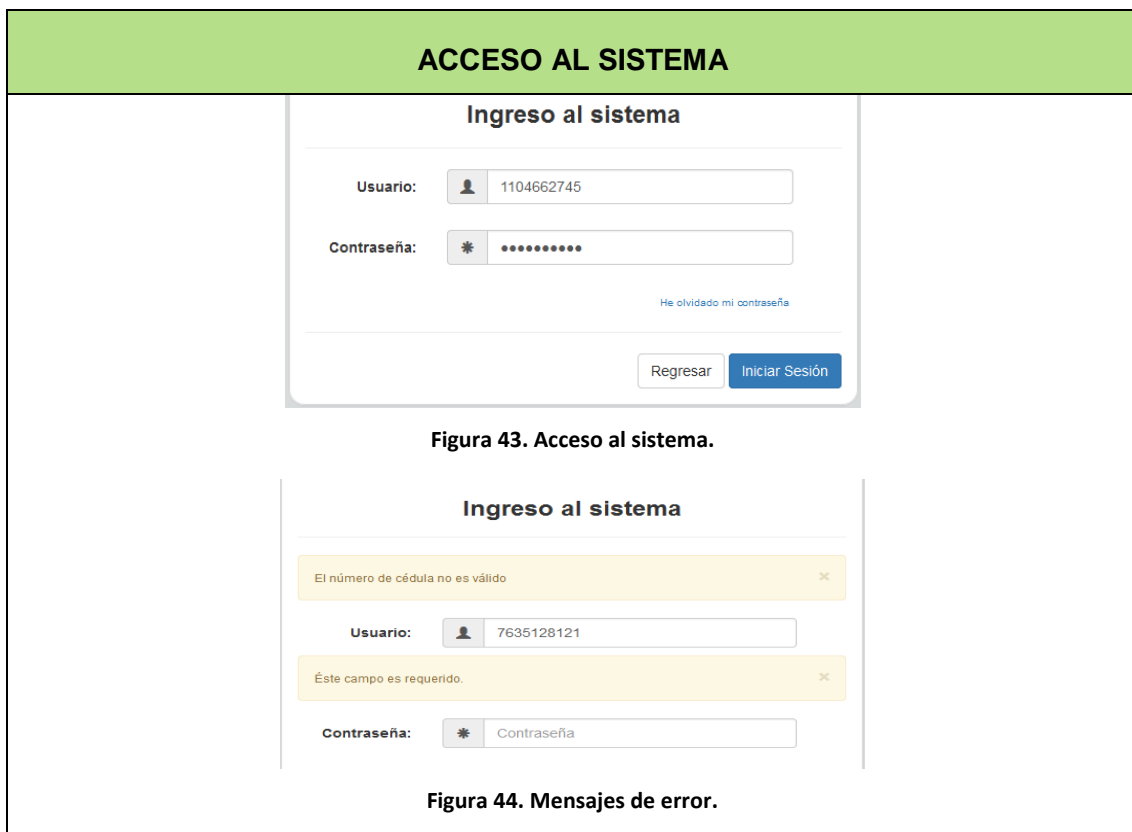
Para la ejecución de todas las pruebas se requiere esencialmente del sistema Help Desk desarrollada en Python y su base de datos Postgresql.

- **Para la prueba de funcionalidad:** Requerimientos Funcionales, Diagrama de Casos de Uso e Iteraciones, Diagrama de Procesos.
- **Para la prueba de interfaz de usuario:** Encuesta (Técnica de investigación), Diagrama de Procesos.
- **Para la prueba de carga:** Apache Brench.

6.4.2. EJECUCIÓN DEL PLAN DE VERIFICACIÓN Y VALIDACIÓN

En el apartado que se describe a continuación se muestra la ejecución de todas las pruebas establecidas en el plan de validación y verificación. La nomenclatura utilizada para representar el resultado de las pruebas reflejó los principales aspectos que se requiere conocer de la ejecución de la prueba.

6.4.2.1. EJECUCIÓN DE PRUEBAS DE FUNCIONALIDAD



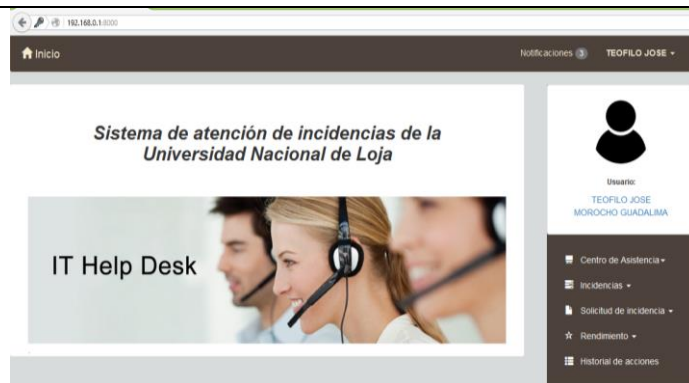


Figura 45. Panel del Usuario Jefe Departamento.

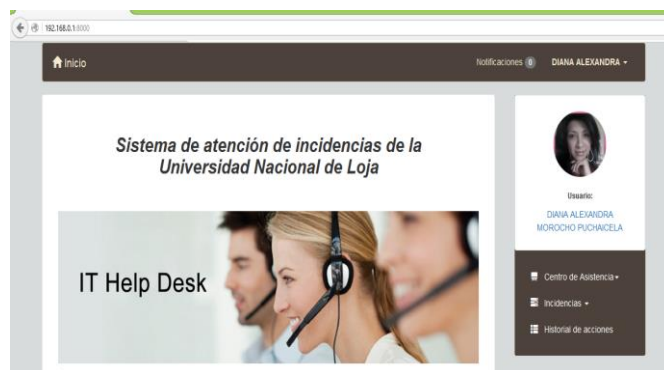


Figura 46. Panel del Usuario Administrador.

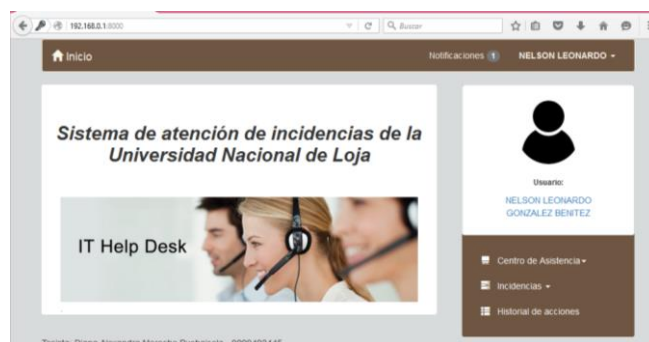


Figura 47. Panel del usuario Asesor Técnico.

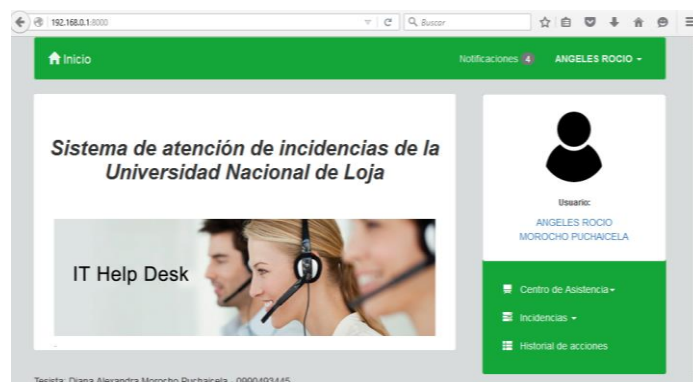


Figura 48. Panel del Usuario Final.

Fecha: 02/10/2015	Tipo: Prueba de Funcionalidad	Código: PF-001
Roles de usuario:	Administrador, Jefe Departamento, Asesor Técnico, Usuario Final.	
Requerimientos:	Diagrama General de Casos de Uso, Iteración PE-004.	
Descripción:	Se ingresó el número de cédula del usuario, y dependiendo del rol asignado, ingresa al sistema mostrando su panel de administración. Para el usuario Administrador y Jefe Departamento, el color del panel es Café, el usuario Asesor Técnico, Marrón, y para el Usuario Final es verde.	
Datos previos:	Registro de datos previos como: roles de usuario, usuarios, y configuración del centro de asistencia.	
Errores encontrados:	<ul style="list-style-type: none"> - No se valida el número de cédula. - Al iniciar sesión no se redirige a la página del index. 	
Errores corregidos:	<ul style="list-style-type: none"> - Cédula válida - Verificación del usuario según grupos determinados, y se renderiza hacia el index correspondiente. 	
Fortaleza:	<ul style="list-style-type: none"> - Dato de contraseña encriptado en la base de datos. 	
Verificado: SI	Corregido: SI	

Tabla 47. Prueba de Funcionalidad PF-001



Figura 49. Centro de Asistencia

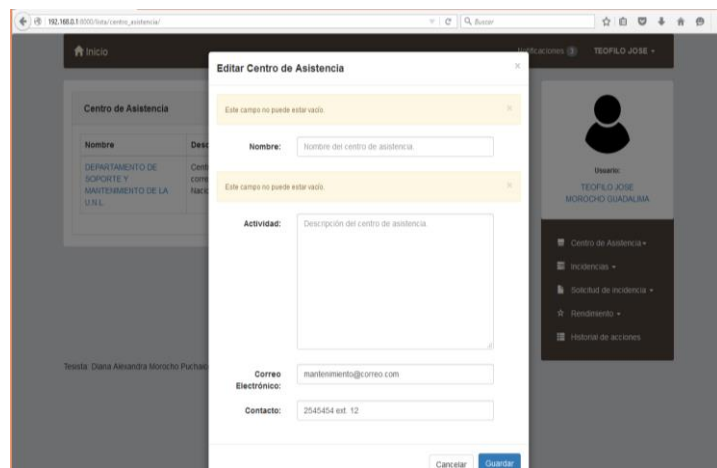


Figura 50. Mensajes de error.

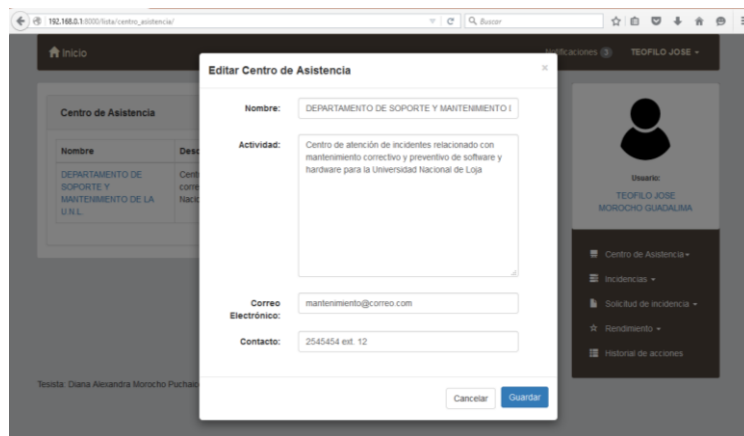


Figura 51. Actualizar Registro.

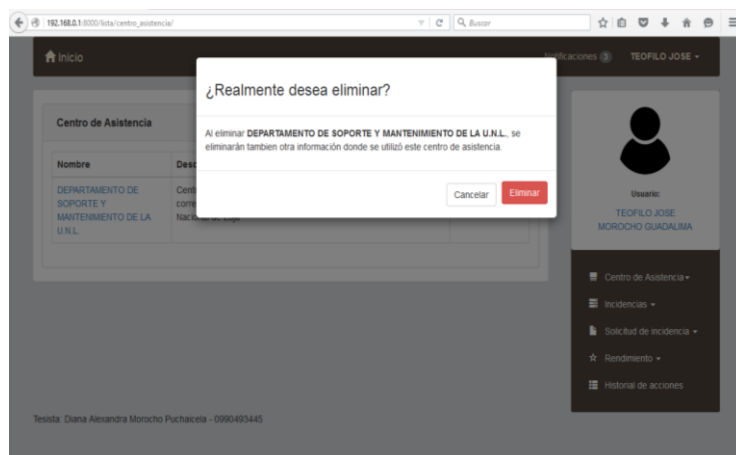


Figura 52. Eliminar Registro.

Fecha: 16/10/2015	Tipo: Prueba de Funcionalidad	Código: PF-002
Roles de usuario:	Administrador o Jefe Departamento.	

Requerimientos:	Diagrama General de Casos de Uso, Iteración PE-002.
Descripción:	Se asigna al usuario los permisos del centro de asistencia, para agregar, actualizar y eliminar registros.
Datos previos:	Registro de datos previos como: roles de usuario, usuarios.
Errores encontrados:	<ul style="list-style-type: none"> - Cuando se elimina un centro de asistencia, se realiza el borrado físico de datos en cascada. - Incidencias reportadas para ese centro, quedan huérfanas.
Errores corregidos:	<ul style="list-style-type: none"> - Solamente cuando no se hayan registrado incidencias para el centro será posible su eliminación lógica. - Al no poder eliminar centros de asistencia que tengan incidencias reportadas, las incidencias se mantienen.
Fortaleza	Al crear un centro de asistencia, permite que el sistema no solo sea aplicado para el objeto del proyecto, sino también para reportar otro tipo de incidentes.
Verificado: SI	Corregido: SI

Tabla 48. Prueba de funcionalidad PF-002.

ASIGNAR USUARIOS A CENTRO DE ASISTENCIA

Figura 53. Panel del Centro de Asistencia.

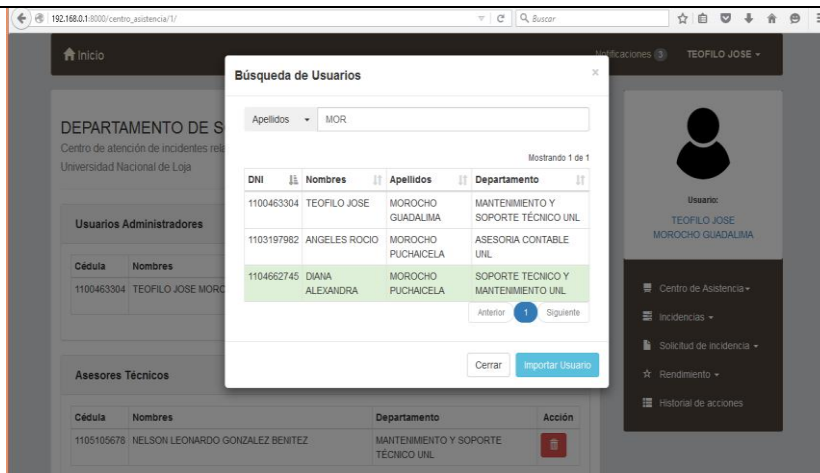


Figura 54. Búsqueda de usuarios.

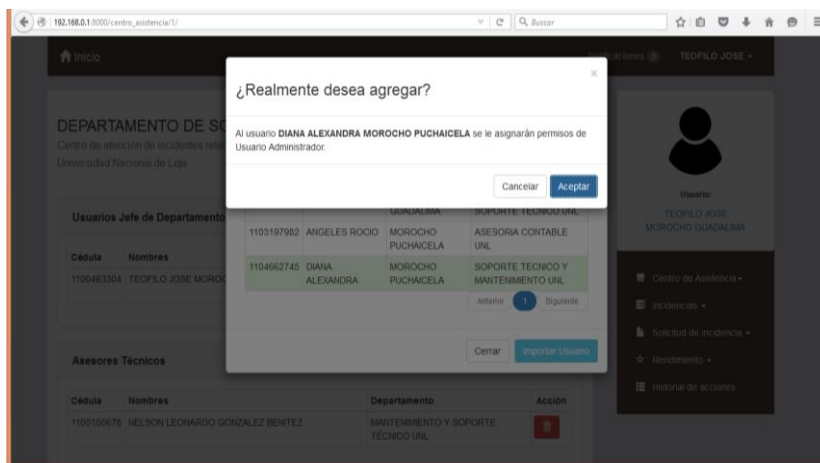


Figura 55. Asignar Jefe de Departamento.

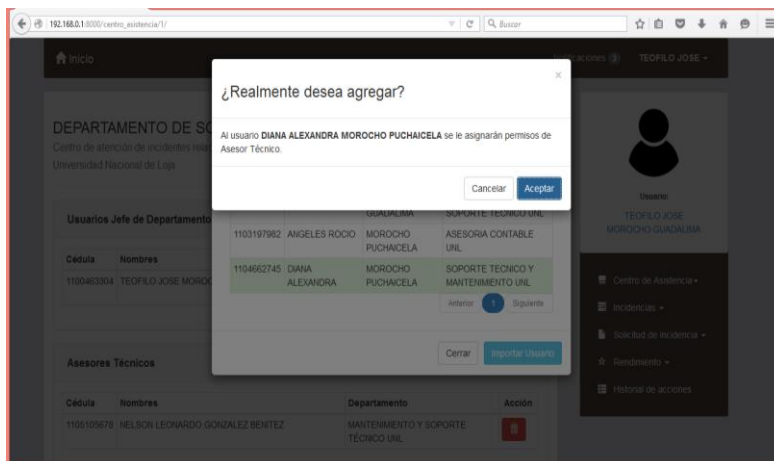


Figura 56. Asignar Asesor Técnico.

Fecha: 19/10/2015

Tipo: Prueba de Funcionalidad

Código: PF-003

Roles de usuario:

Administrador o Jefe Departamento.

Requerimientos:	Diagrama General de Casos de Uso, Iteración PE-002.
Descripción:	Permite agregar al centro de asistencia los usuarios Jefe Departamento (función de conmutador) y Asesor Técnico (función atender la incidencia)
Datos previos	Registro de datos previos como: roles de usuario, usuarios y centro de asistencia.
Errores encontrados:	Ninguno.
Errores corregidos:	Ninguno.
Fortaleza:	Asignar a los usuarios Jefe Departamento y Asesor Técnico los permisos necesarios para cumplir las funciones.
Verificado: SI	Corregido: NO

Tabla 49. Prueba de funcionalidad PF-003

ADMINISTRAR ACUERDOS DE NIVEL DE SERVICIO



The screenshot displays a web application interface for managing Service Level Agreements (SLAs). The main content area is divided into two sections:

- Asesores Técnicos:** A table with columns for 'Cédula', 'Nombres', 'Departamento', and 'Acción'. It lists Nelson Leonardo Gonzalez Benitez, associated with 'MANTENIMIENTO Y SOPORTE TÉCNICO UNL'.
- S.L.A. Acuerdos de nivel de servicio:** A table with columns for 'Servicio', 'T. Mínimo', 'T. Normal', 'T. Máximo', and 'Acción'. It lists three services: 'PRUEBA DE SERVICIOS CORTOS', 'MANTENIMIENTO CORRECTIVO DE COMPUTADORES CAMPUS', and 'REPARACION BASICA DE IMPRESORAS CAMPUS', each with associated time intervals and an 'Acción' icon.

The browser address bar shows the URL '192.168.0.1:8000/centro_asistencia/1/'. A sidebar on the right contains navigation options: 'Incidencias', 'Solicitud de incidencia', 'Rendimiento', and 'Historial de acciones'.

Figura 57. Lista de Acuerdos de Nivel de Servicio.

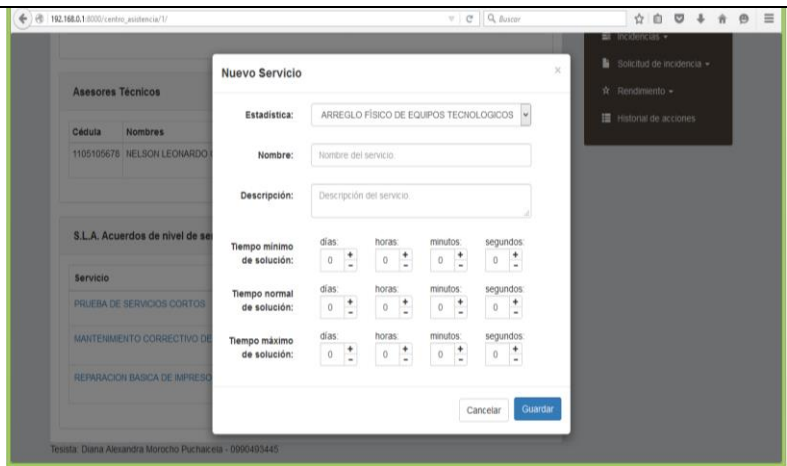


Figura 58. Nuevo SLA.

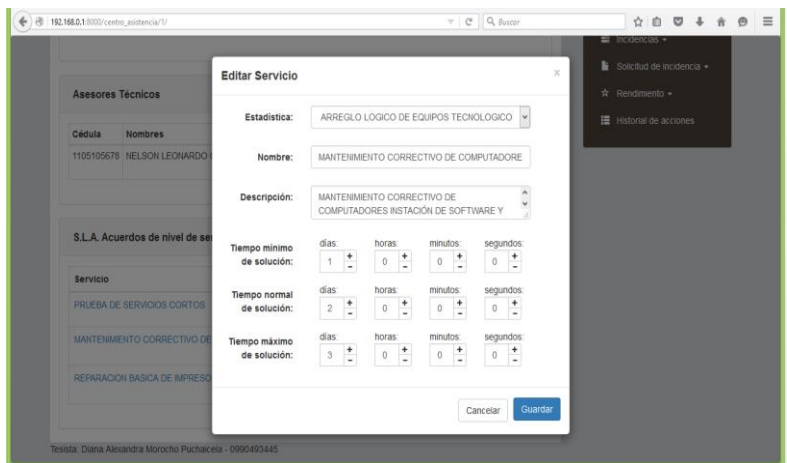


Figura 59. Actualizar SLA.

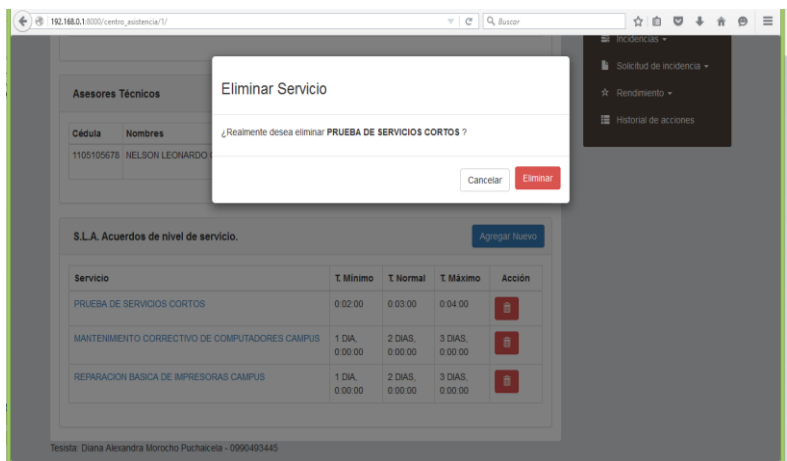


Figura 60. Eliminar SLA.

Fecha: 26/10/2015	Tipo: Prueba de Funcionalidad	Código: PF-004
Roles de usuario:	Administrador o Jefe Departamento.	

Requerimientos:	Diagrama General de Casos de Uso, Iteración PE-003.
Descripción:	Permite agregar al centro de asistencia los acuerdos de nivel de servicio para la atención de incidencias.
Datos previos	Registro de datos previos como: roles de usuario, usuarios y centro de asistencia.
Errores encontrados:	<ul style="list-style-type: none"> - Para agregar un acuerdo de nivel de servicio no existe un rango que limite el tiempo mínimo seleccionado, por lo que los SLA podrían registrarse a conveniencia del Asesor Técnico. - Validación entre tiempo mínimo, normal y máximo.
Errores corregidos:	<ul style="list-style-type: none"> - Para limitar el acuerdo de nivel de servicio, se creó una clase denominada Estadística_SLA y hace referencia a una actividad específica, donde el Administrador limita el tiempo mínimo y máximo. - Se corrigió bajo el siguiente control: El tiempo mínimo debe ser mayor o igual a la Estadística del SLA, el tiempo normal debe ser mayor o igual que el tiempo mínimo y el tiempo máximo debe ser mayor o igual que el tiempo normal. - El formulario presenta mensajes de error para datos no válidos.
Fortaleza:	<ul style="list-style-type: none"> - Crear Acuerdos de Nivel de Servicio, influye en la efectividad de la atención a incidentes, lo que significa optimizar el tiempo de respuesta en atención a los usuarios y reducir significativamente la utilización de recursos.
Verificado: SI	Corregido: SI

Tabla 50. Prueba de Funcionalidad PF-004

6.4.2.2. EJECUCIÓN DE PRUEBAS DE INTERFAZ DE USUARIO

Luego de la primera socialización del sistema realizada en la Unidad de Telecomunicaciones e Información, se autorizó la segunda socialización en la Sección de Soporte Técnico y Mantenimiento con el personal administrativo de ese departamento y, para validar las pruebas de interfaz de usuario planteadas en este Plan de Verificación y Validación se aplicó la encuesta del Anexo 10, que contiene preguntas específicas que permiten aceptar de forma cierta la navegabilidad y funcionalidades del sistema, además de verificar si éste cumple con el principal proceso de negocio de ésta

dependencia. De la premisa anterior, a continuación en la Figura 61 se muestra el principal proceso de negocio con la utilización del sistema.

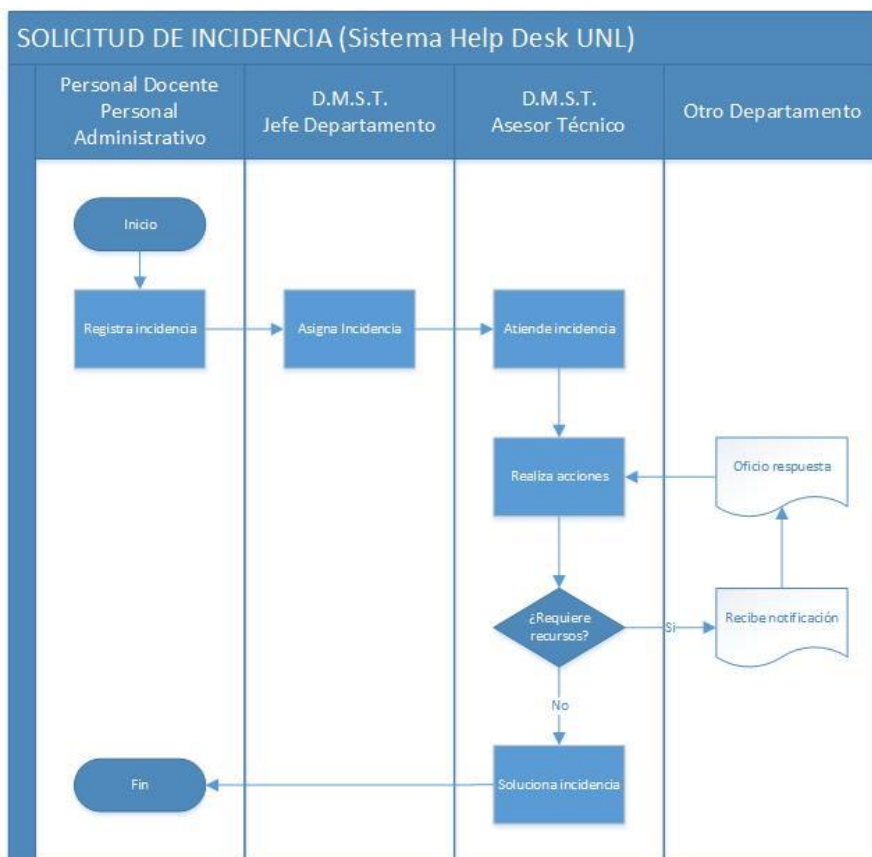


Figura 61. Diagrama del proceso solicitud de incidencia con el sistema Help Desk.

Si se compara el diagrama de procesos de la Figura 2 con el diagrama de procesos de la Figura 61, en un mismo contexto (Ejem. Atasco de papel en una impresora) se puede deducir dos entornos de ejecución del proceso de solicitud de incidencia que se muestra en la Tabla 51.

SOLICITUD DE INCIDENCIA		
	Con el sistema	Sin el Sistema
Tiempo de realización.	El tiempo de realización se vió considerablemente reducido gracias a que los acuerdos de nivel de servicio establecen el tiempo de ejecución mínima, normal y máxima.	El tiempo estimado de solución a una incidencia, está abierto a lo que el asesor técnico considere necesario, por lo que algunas incidencias podrían extender su tiempo de solución.

Recursos	Los recursos necesarios para reportar una incidencia únicamente involucran a los relacionados con el acceso al sistema.	Los recursos necesarios para reportar una incidencia van desde: - Llamada telefónica. - Utilización de papel para registrar la incidencia reportada.
Tiempo del personal	El sistema permite visualizar las incidencias actualmente atendidas por los asesores técnicos, lo que permite distribuir equitativamente el trabajo, así como el de influir al asesor a atender las incidencias en el menor tiempo estimado.	No existe un documento formal para determinar el tiempo estimado en la solución a incidentes. En alguna ocasión la atención a incidentes puede generar algún excedente de tiempo que puede ser utilizado en otra actividad.
Calidad de servicio	En el módulo de evaluación de la calidad del servicio, se puede establecer mediante una calificación especializada el resultado de cuan efectivo es el Asesor Técnico frente a la solución de un problema.	La medición de la calidad del servicio se obtiene registrando un apartado de la ficha de reporte ver Ficha de observación 004 del Anexo 6, sin embargo actualmente está en desuso, no se contabilizan ni evalúan los resultados.

Tabla 51. Comparativa de procesos.

De lo que se puede deducir que el sistema aporta considerablemente en la solución a incidentes.

Seguidamente se procede a realizar las pruebas establecidas en el plan de validación y verificación. De manera general se ha aplicado 4 encuestas, a los 4 tipos de usuarios del sistema, de las que se pudo obtener los siguientes resultados.

Valoración del Sistema Help Desk		
1. Marque con una X, si considera a la premisa como Verdadera (SI) o Falsa (NO).	SI	NO
• La utilización del sistema Help Desk es sencilla. Cada componente es de fácil localización.		
• La aplicación es de gran utilidad para la automatización de atención a incidencias.		
• La velocidad del ingreso de datos y notificaciones es rápida e inmediata.		
• El sistema ayuda al usuario a atender incidencias de forma rápida, reduciendo los errores.		

Figura 62. Preguntas de la encuesta.

Para obtener el resultado de la usabilidad del sistema, es decir la facilidad con que las personas pueden utilizar la aplicación para lograr la consecución de los objetivos, se planteó la siguiente pregunta.

- **La utilización del sistema Help Desk es sencilla. Cada componente es de fácil localización. Si () No ()**

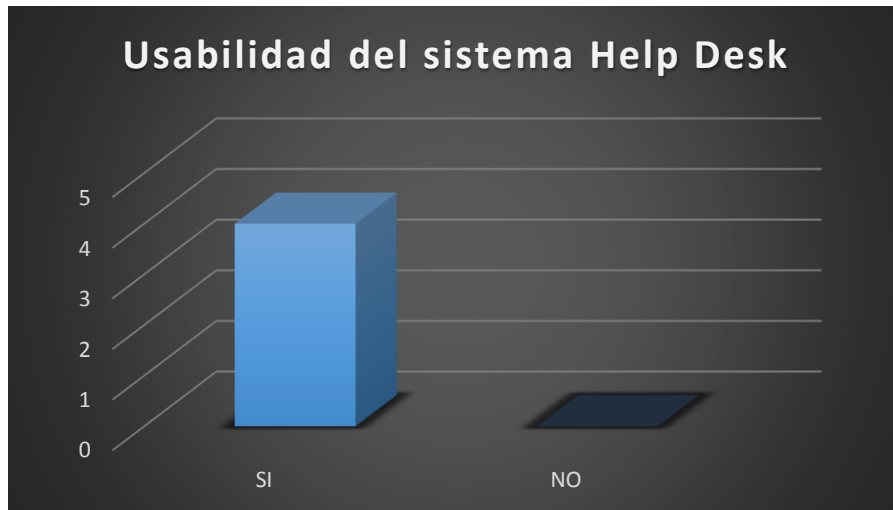


Figura 63. Respuesta de usabilidad del sistema Help Desk.

INTERPRETACIÓN: Cuatro de cuatro encuestados respondió que la afirmación era verdadera, es decir que no tuvieron inconveniente para usar la aplicación, por tanto se deduce que el sistema es de fácil uso.

Para definir el grado de aceptación de la aplicación, se aplicó la siguiente pregunta

- **La aplicación es de gran utilidad para la automatización de atención a incidencias. Si () No ()**



Figura 64. Respuesta de aceptación al sistema Help Desk.

INTERPRETACIÓN. Cuatro de cuatro encuestados respondió que la afirmación era verdadera, es decir que la aplicación es aceptada como solución a la atención de incidencias.

Para definir el grado de efectividad de la aplicación, se aplicó la siguiente pregunta:

- **La velocidad de ingreso de datos y notificaciones es rápida e inmediata.**

Si () No ()

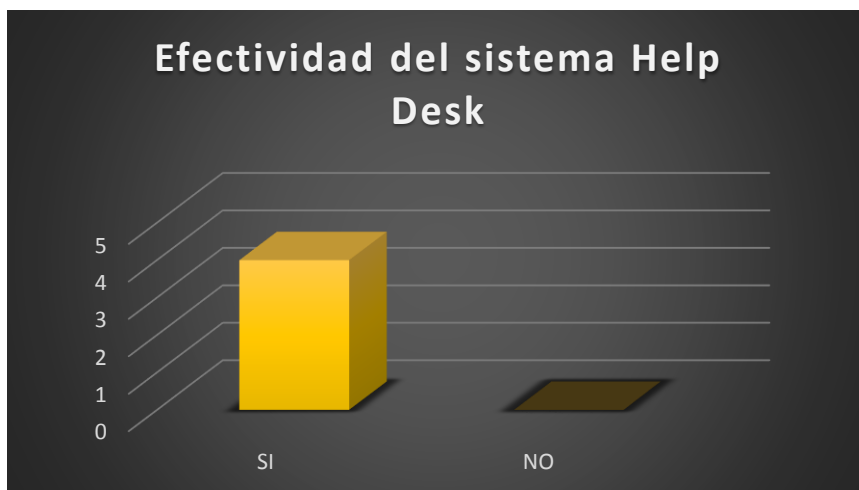


Figura 65. Efectividad del sistema Help Desk.

INTERPRETACIÓN: Cuatro de cuatro encuestados respondió que la afirmación era verdadera, es decir que la aplicación es efectiva en cuando a ingreso de datos y envío de notificaciones.

Para definir, el grado de utilidad del sistema, se aplicó la siguiente pregunta:

- **El sistema ayuda al usuario a atender incidencias de forma rápida, reduciendo los errores.** Si () No ()

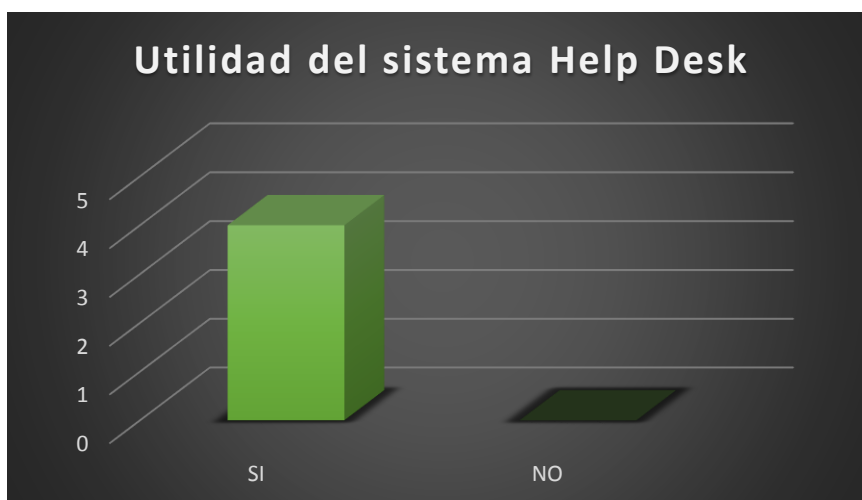


Figura 66. Utilidad del sistema Help Desk.

INTERPRETACIÓN: Cuatro de cuatro encuestados respondió que la afirmación era verdadera, es decir que la aplicación es útil.

Posteriormente, al aplicar la sección específica para cada rol de usuario de la encuesta, se obtuvo los siguientes resultados.

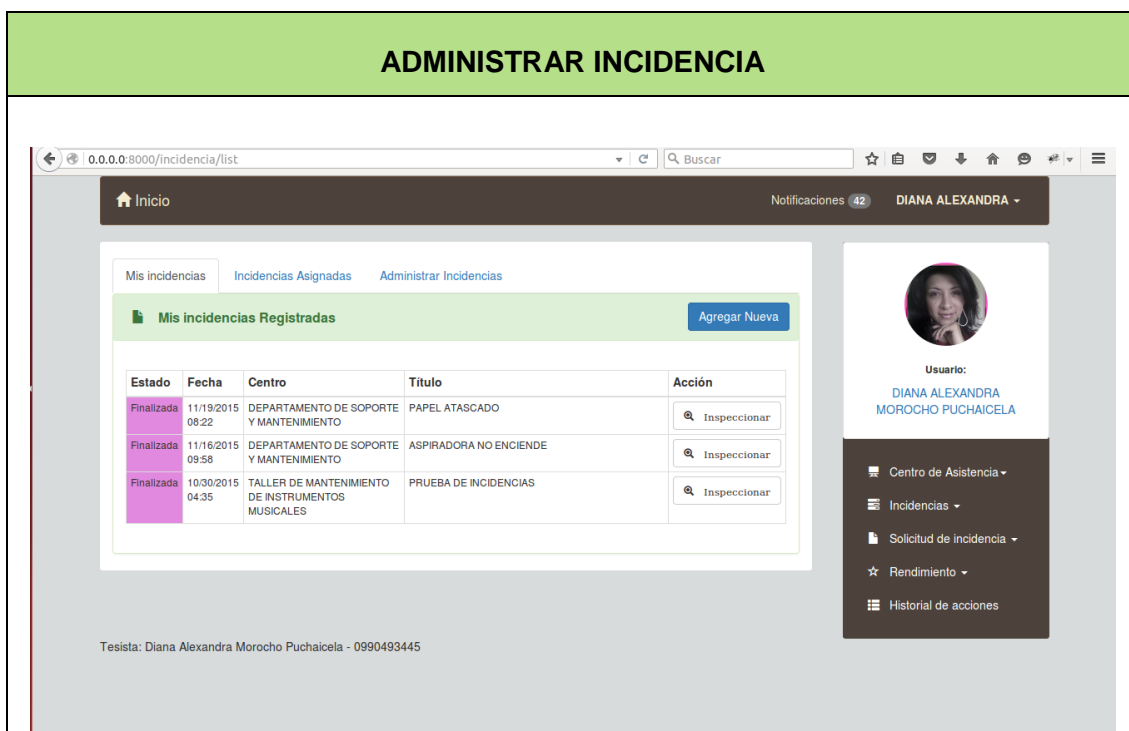


Figura 67. Administrar incidencia.

Fecha: 30/11/2015	Tipo: Prueba de Interfaz de Usuario	Código: PIU-001
Roles de usuario:	Usuario Final.	
Requerimientos:	Diagrama del proceso solicitud de incidencia, diagrama general de casos de uso, iteración PE-005.	
Descripción:	Esta funcionalidad le permite a los usuarios con rol de <i>Usuario Final</i> , reportar incidencias seleccionando el centro de asistencia a ser atendido, permite visualizar el estado actual y al finalizar realizar una evaluación de atención.	
Datos previos:	Registro de datos previos como: roles de usuario, usuarios, centro de asistencia y acuerdos de nivel de servicio.	
Errores encontrados:	<ul style="list-style-type: none"> - Algunas interfaces de tipo modal no se visualizan. - La interfaz para reportar bienes, incompleta. 	

Errores corregidos:	<ul style="list-style-type: none"> - Las interfaces de tipo modal, son desarrolladas con el framework Bootstrap, por lo que, al no encontrar el permiso de usuario para realizar la función, ésta no se renderiza. Además, para corregir éste problema se ha recurrido a peticiones AJAX en determinadas ventanas. - Interfaz de usuario para reportar bienes es corregida, la función no se visualizaba completamente debido a que la administración del bien no se realiza en ésta iteración.
----------------------------	---

Resultados de Encuesta:

Se aplicó la encuesta al Ing. Luis Malla, utilizando en el sistema el rol de Usuario Final, esto fue lo que respondió al plantearle el siguiente bloque de preguntas:

Espacio único para el Usuario Final.

- **¿Qué tan complejo es reportar una incidencia?**
 Nada complejo Medianamente complejo Muy complejo
- **Reportar bienes en una incidencia es una tarea**
 Fácil Compleja Difícil
- **Calificar la atención brindada al finalizar la incidencia resulta**
 Beneficioso Útil Inadecuado

Figura 68. Bloque de preguntas al Usuario Final.

- **¿Qué tan complejo es reportar una incidencia?: Nada complejo () Medianamente complejo () Muy complejo ()**



Figura 69. Complejidad de reporte de incidencia.

- **Reportar bienes en una incidencia es una tarea: Fácil () Compleja () Difícil ()**



Figura 70. Complejidad de reporte de bienes.

- Calificar la atención brindada al finalizar la incidencia resulta: **Beneficioso** () **Útil** () **Inadecuado** ()

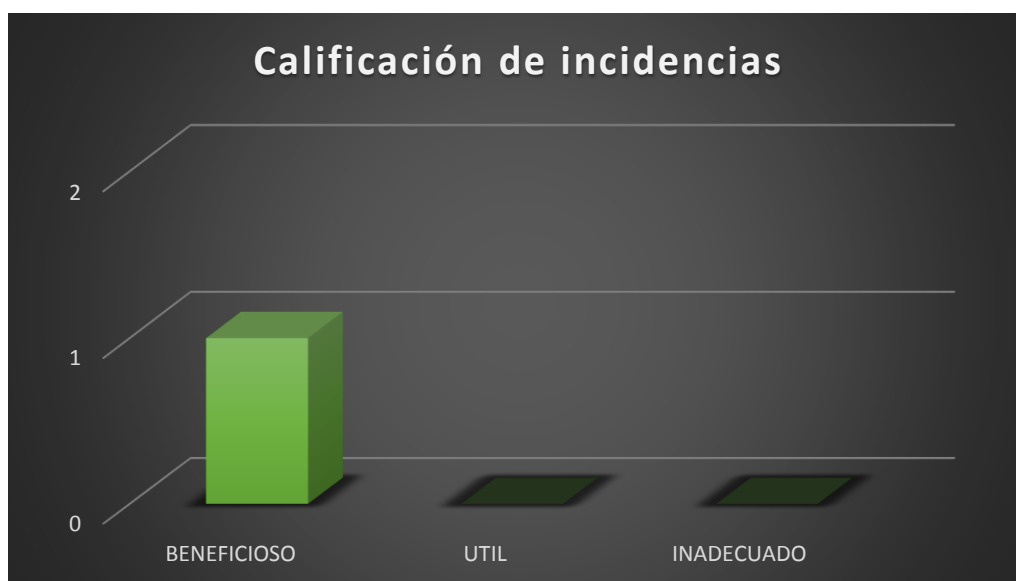


Figura 71. Calificación de incidencias.

INTERPRETACIÓN: Claramente en la figura 69, 70, y 71 el usuario manifiesta no tener inconvenientes al reportar incidencias.

Verificado: SI

Cumple: SI

Tabla 52. Prueba de Interfaz de usuario - Administrar Incidencia – PIU-002.

ASIGNACIÓN DE INCIDENCIA

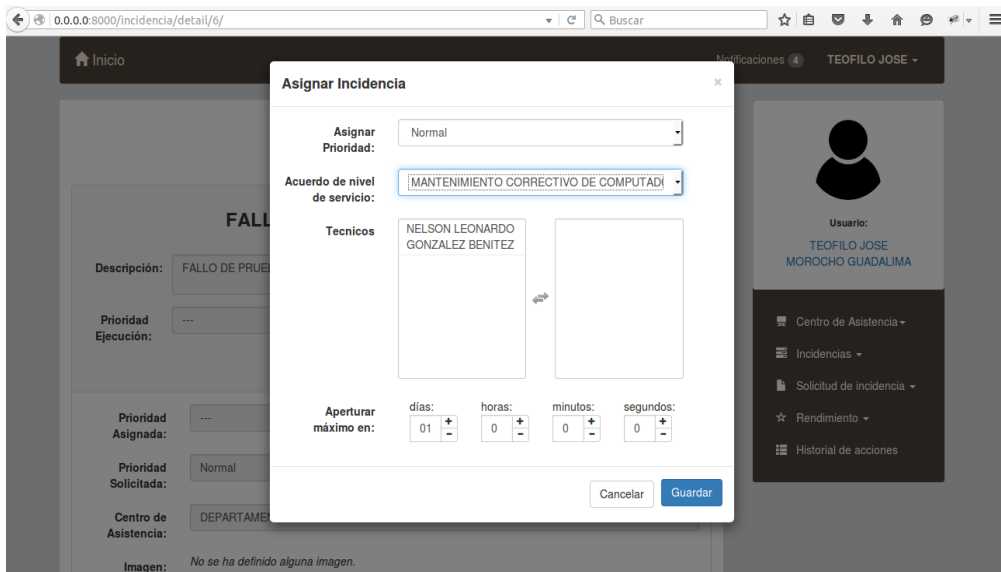


Figura 72. Asignación de incidencia.

Fecha: 30/11/2015	Tipo: Prueba de Interfaz de Usuario	Código: PIU-002
Roles de usuario:	Jefe de departamento, asesor técnico	
Requerimientos:	Diagrama del proceso solicitud de incidencia, diagrama general de casos de uso, iteración PE-007, iteración PE-008.	
Descripción:	Esta funcionalidad le permite al usuario <i>Jefe Departamento</i> asignar la incidencia para ser atendida hacia el usuario <i>Asesor Técnico</i> .	
Datos previos:	Registro de datos previos como: roles de usuario, usuarios, centro de asistencia, acuerdos de nivel de servicio, incidencia.	

Resultados de Encuesta:

Se aplicó las siguientes preguntas a la Ing. Mabel Rodríguez que inicialmente fue quien proporcionó las historias de usuario, en el rol de Jefe Departamento y se obtuvo:

Espacio único para el Jefe Departamento.		
• ¿Qué tan complejo es asignar una incidencia al Asesor Técnico?	Nada complejo <input type="checkbox"/>	Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>
• Verificar la calidad de servicio que brinda el usuario Asesor Técnico es una tarea	Fácil <input type="checkbox"/>	Compleja <input type="checkbox"/> Difícil <input type="checkbox"/>
• Asignar un tiempo máximo de apertura sobre las incidencias resulta	Beneficioso <input type="checkbox"/>	Útil <input type="checkbox"/> Inadecuado <input type="checkbox"/>

Figura 73. Preguntas dirigidas al Jefe de Departamento.

- ¿Qué tan complejo es asignar una incidencia al Asesor Técnico? Nada Complejo () Medianamente complejo () Muy complejo ()

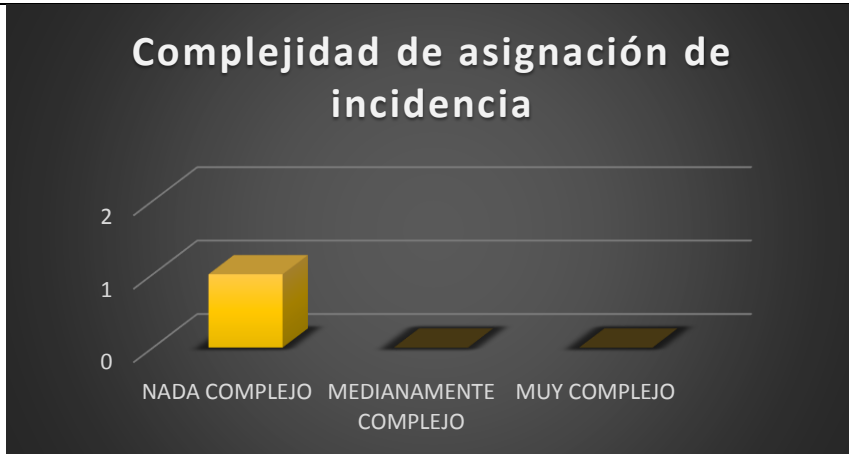


Figura 74. Complejidad de asignación de incidencia.

- Verificar la calidad de servicio que brinda el usuario Asesor Técnico es una tarea: Fácil() Compleja () Difícil ()

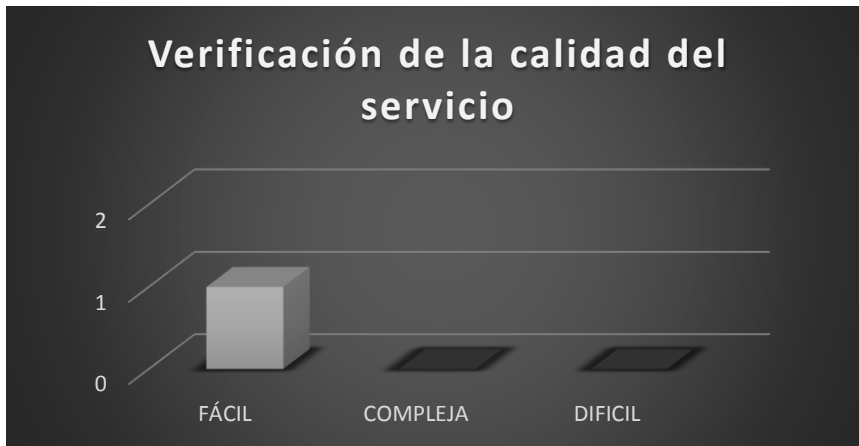


Figura 75. Verificación de la calidad de servicio.

- Asignar un tiempo máximo de apertura sobre las incidencias resulta: Beneficioso () Útil() Inadecuado ()



Figura 76. Tiempo de apertura máxima.

INTERPRETACIÓN: Claramente en la figura 74, 75, y 76 el usuario manifiesta no tener inconvenientes al asignar incidencias.

Verificado: SI	Cumple: SI
-----------------------	-------------------

Tabla 53. Prueba de Interfaz de usuario - Asignar Incidencia – PIU-002.



Figura 77. Atención de incidencia.

Fecha: 30/11/2015	Tipo: Prueba de Interfaz de Usuario	Código: PIU-003
Roles de usuario:	Asesor Técnico	
Requerimientos:	Diagrama del proceso solicitud de incidencia, diagrama general de casos de uso, iteración PE-009.	
Descripción:	Esta funcionalidad le permite al usuario <i>Asesor Técnico</i> iniciar la atención del incidente.	
Datos previos:	Registro de datos previos como: roles de usuario, usuarios, centro de asistencia, acuerdos de nivel de servicio, incidencia, asignación de la incidencia.	

Resultados de Encuesta:

Se aplicó la encuesta al Ing. Juan Pablo González en el rol de *Asesor Técnico*; de esto, se obtuvo la siguiente información:

Espacio único para el Asesor Técnico.		
• ¿Qué tan complejo es atender una incidencia que ha sido asignada?	Nada complejo <input type="checkbox"/>	Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>
• Registrar acciones sobre una incidencia es una tarea	Fácil <input type="checkbox"/> Compleja <input type="checkbox"/>	Difícil <input type="checkbox"/>
• Realizar el diagnóstico inicial de la incidencia y de los bienes reportados es	Muy útil <input type="checkbox"/> Poco útil <input type="checkbox"/>	Nada útil <input type="checkbox"/>

Figura 78. Preguntas dirigidas al Asesor Técnico.

- ¿Qué tan complejo es atender una incidencia que ha sido asignada?: Nada complejo() Medianamente complejo() Muy complejo()



Figura 79. Atención de incidencias asignadas.

- Registrar acciones sobre una incidencia es una tarea: Fácil() Compleja () Difícil ()



Figura 80. Registro de acciones de incidencia.

- Realizar el diagnóstico inicial de la incidencia y de los bienes reportados es: Muy útil() Poco útil() Nada útil ()

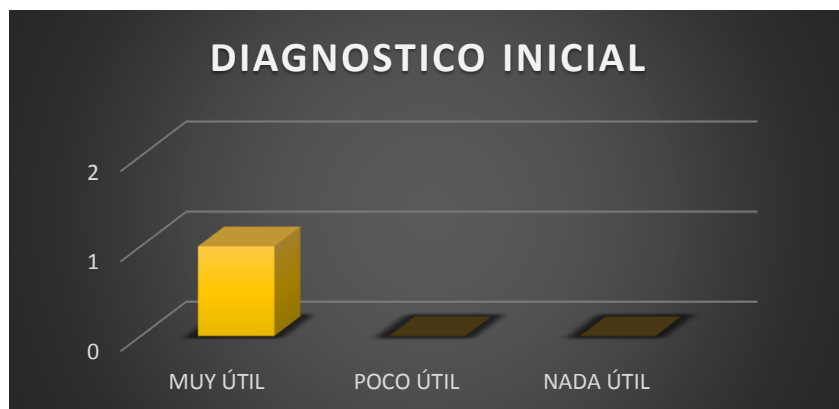


Figura 81. Diagnóstico inicial.

INTERPRETACIÓN: Claramente en la figura 79, 80, y 81 el usuario manifiesta no tener inconvenientes al atender incidencias.

Verificado: SI

Cumple: SI

Tabla 54. Prueba de Interfaz de usuario – Atender Incidencia – PIU-003

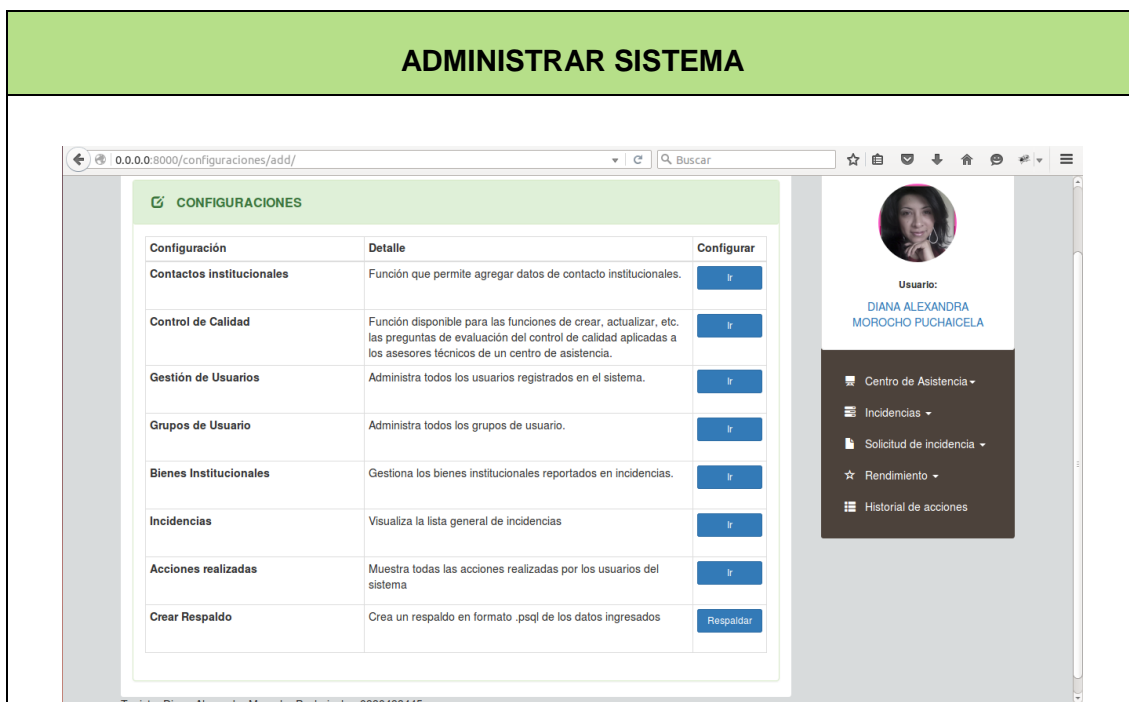


Figura 82. Administrar sistema

Fecha: 30/11/2015	Tipo: Prueba de Interfaz de Usuario	Código: PIU-004
Roles de usuario:	Administrador	
Requerimientos:	Ninguno.	
Descripción:	Éste módulo permite al usuario <i>Administrador</i> , gestionar todas las configuraciones iniciales del centro de asistencia y del sistema en general.	
Datos previos:	Registro de datos previos como: roles de usuario, usuarios.	

Resultados de Encuesta:

Se aplicó la encuesta al Ing. Edison Coronel, en el rol de *Administrador*, de esto, se obtuvo la siguiente información:

Espacio único para el Administrador.		
• ¿Qué tan complejo es administrar el panel de configuraciones?	Nada complejo <input type="checkbox"/>	Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>
• Agregar usuarios Jefe de Departamento y Asesor Técnico al centro de asistencia es una tarea	Fácil <input type="checkbox"/>	Complejo <input type="checkbox"/> Difícil <input type="checkbox"/>
• Registrar varios centros de asistencia para la atención de incidencias resulta	Beneficioso <input type="checkbox"/>	Útil <input type="checkbox"/> Inadecuado <input type="checkbox"/>

Figura 83. Preguntas dirigidas al administrador.

- ¿Qué tan complejo es administrar el panel de configuraciones? Nada complejo() Medianamente complejo () Muy complejo ()

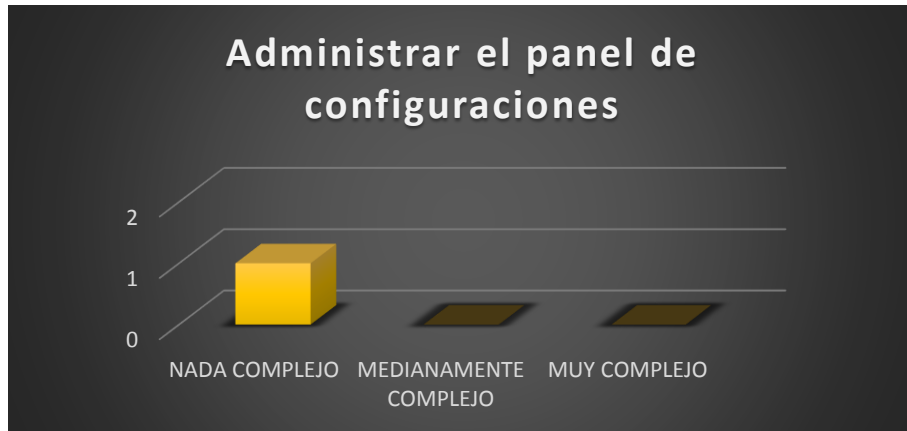


Figura 84. Administrar panel de configuraciones

- Agregar usuarios Jefe Departamento y Asesor Técnico al centro de asistencia es una tarea: Fácil() Complejo () Difícil ()

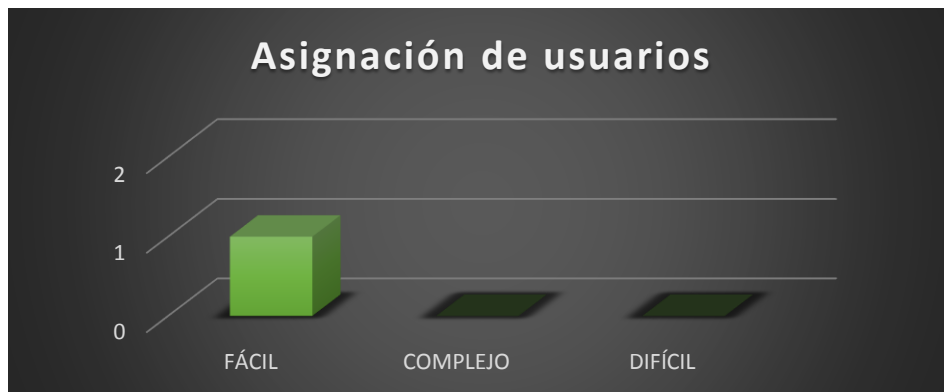


Figura 85. Asignación de usuarios.

- Registrar varios centros de asistencia para la atención de incidencias resulta: Beneficioso() Útil() Inadecuado()

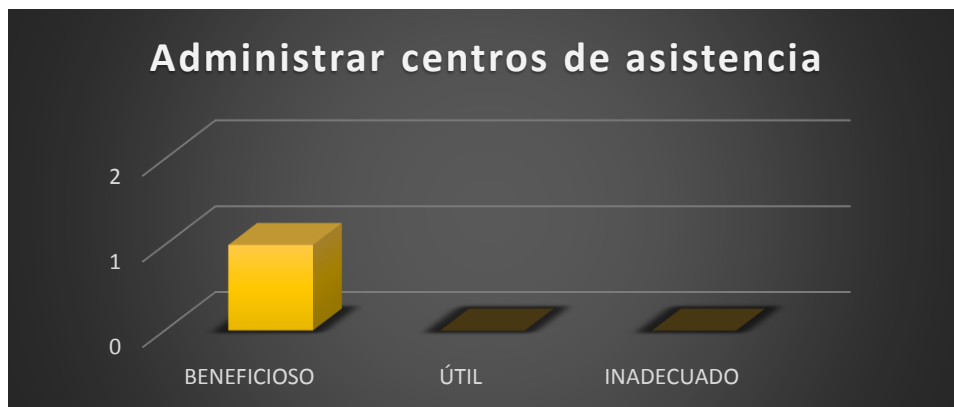


Figura 86. Administrar centros de asistencia.

INTERPRETACIÓN: Claramente en la figura 84, 85, y 86 el usuario manifiesta no tener inconvenientes al administrar el sistema.

Verificado: SI

Cumple: SI

Tabla 55. Prueba de Interfaz de usuario – Administrar Sistema – PIU-004

6.4.2.3. EJECUCIÓN DE PRUEBAS DE CARGA

URL DE INICIO

- Escenario 1. Prueba de carga con 100 peticiones.

```
dianita@dianita-Inspiron-5437:~$ ab -g Documentos/resultados.csv -n 100 http://
0.0.0.0:8000/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 0.0.0.0 (be patient).....done

Server Software:      WSGIServer/0.1
Server Hostname:     0.0.0.0
Server Port:         8000

Document Path:       /
Document Length:     4115 bytes

Concurrency Level:   1
Time taken for tests: 3.258 seconds
Complete requests:   100
Failed requests:     0
Total transferred:   429100 bytes
HTML transferred:   411500 bytes
Requests per second: 30.70 [#/sec] (mean)
Time per request:    32.576 [ms] (mean)
Time per request:    32.576 [ms] (mean, across all concurrent requests)
Transfer rate:       128.64 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    0  0.0    0    0
Processing: 16   32  7.4   33   46
Waiting:    16   32  7.4   33   45
Total:      16   33  7.4   33   46

Percentage of the requests served within a certain time (ms)
 50%    33
 66%    37
 75%    38
 80%    39
 90%    41
 95%    44
 98%    46
 99%    46
100%    46 (longest request)
dianita@dianita-Inspiron-5437:~$ █
```

Figura 87. Prueba de carga – escenario 1.

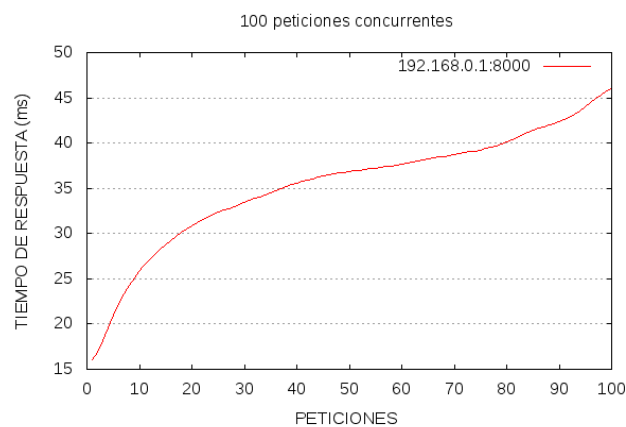


Figura 88. Gráfico de prueba de carga.

- **Escenario 2. Prueba de carga con 500 peticiones.**

```
dianita@dianita-Inspiron-5437:~$ ab -g Documentos/resultados.csv -n 500 http://0.0.0.0:8000/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 0.0.0.0 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests

Server Software:      WSGIServer/0.1
Server Hostname:     0.0.0.0
Server Port:         8000

Document Path:       /
Document Length:     4115 bytes

Concurrency Level:   1
Time taken for tests: 16.578 seconds
Complete requests:   500
Failed requests:     0
Total transferred:   2145500 bytes
HTML transferred:   2057500 bytes
Requests per second: 30.16 [#./sec] (mean)
Time per request:    33.157 [ms] (mean)
Time per request:    33.157 [ms] (mean, across all concurrent requests)
Transfer rate:       126.38 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:      0    0  0.0    0    0
Processing:   16   33  8.1   36   48
Waiting:      16   33  8.1   35   48
Total:        16   33  8.1   36   48

Percentage of the requests served within a certain time (ms)
 50%    36
 66%    38
 75%    39
 80%    39
 90%    42
 95%    44
 98%    45
 99%    46
100%    48 (longest request)
dianita@dianita-Inspiron-5437:~$ █
```

Figura 89. Prueba de carga – escenario 2.

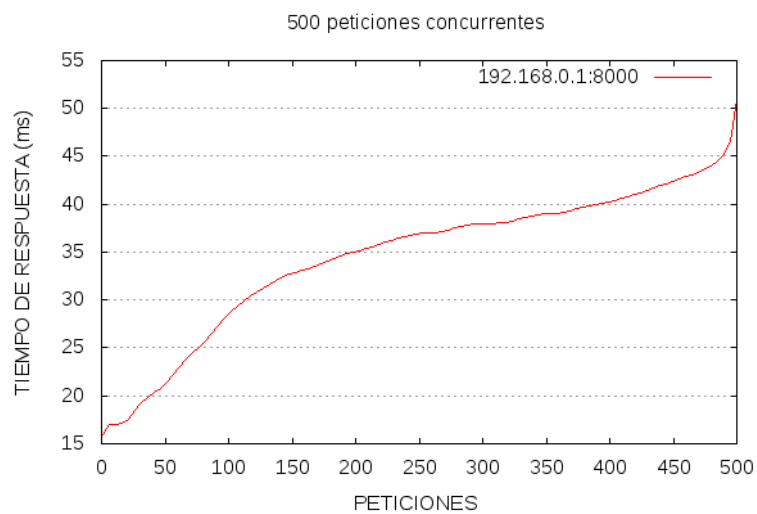


Figura 90. Gráfico de prueba de carga.

- **Escenario 3.**

```
dianita@dianita-Inspiron-5437:~$ ab -g Documentos/resultados.csv -n 1000 http://0.0.0.0:8000/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 0.0.0.0 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      WSGIServer/0.1
Server Hostname:     0.0.0.0
Server Port:         8000

Document Path:       /
Document Length:     4115 bytes

Concurrency Level:   1
Time taken for tests: 33.983 seconds
Complete requests:   1000
Failed requests:     0
Total transferred:   4291000 bytes
HTML transferred:    4115000 bytes
Requests per second: 29.43 [#/sec] (mean)
Time per request:    33.983 [ms] (mean)
Time per request:    33.983 [ms] (mean, across all concurrent requests)
Transfer rate:       123.31 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0    0  0.0    0    0
Processing: 16   34  8.1   36   53
Waiting:    16   33  8.0   36   52
Total:      16   34  8.1   36   53

Percentage of the requests served within a certain time (ms)
 50%    36
 66%    39
 75%    40
 80%    40
 90%    42
 95%    44
 98%    45
 99%    47
100%    53 (longest request)
dianita@dianita-Inspiron-5437:~$ █
```

Figura 91. Prueba de carga – escenario 3.

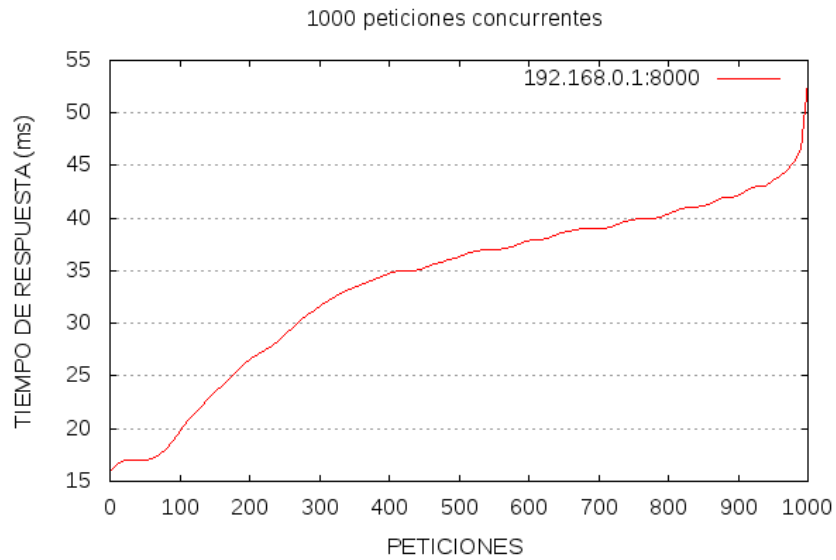
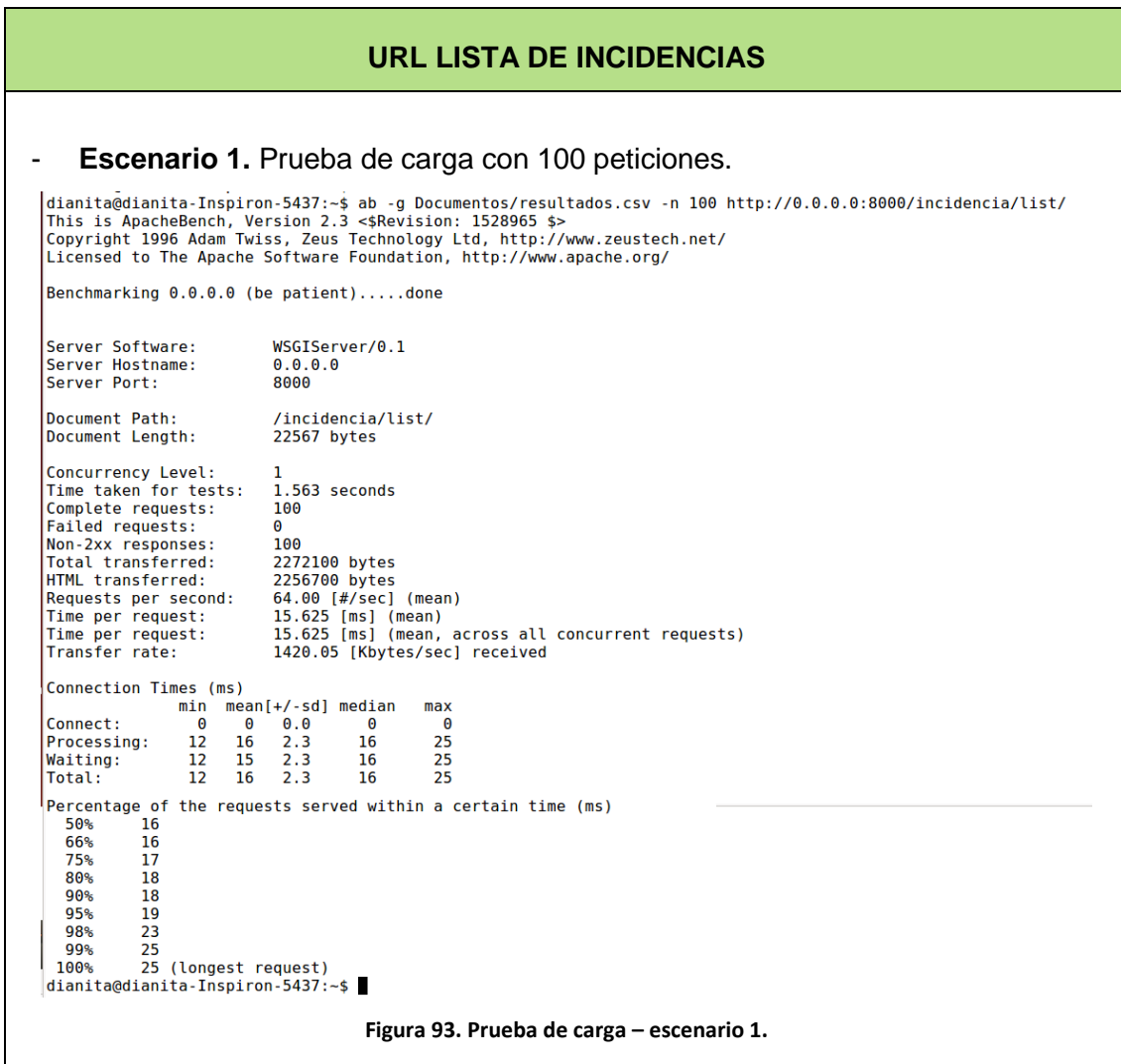


Figura 92. Gráfico de prueba de carga.

Fecha: 30/11/2015	Tipo: Prueba de Carga	Código: PC-001
Roles de usuario:	Cualquier usuario.	
Descripción:	Permite determinar si después de la carga de peticiones, el sistema continúa en función.	
Software previo:	Apache Brench, Plot.	
Resultados: Luego de ejecutar el sistema con diferente carga, éste mantiene su funcionalidad.		
Verificado: SI	Cumple: SI	

Tabla 56. Prueba de Carga – URL de Inicio.



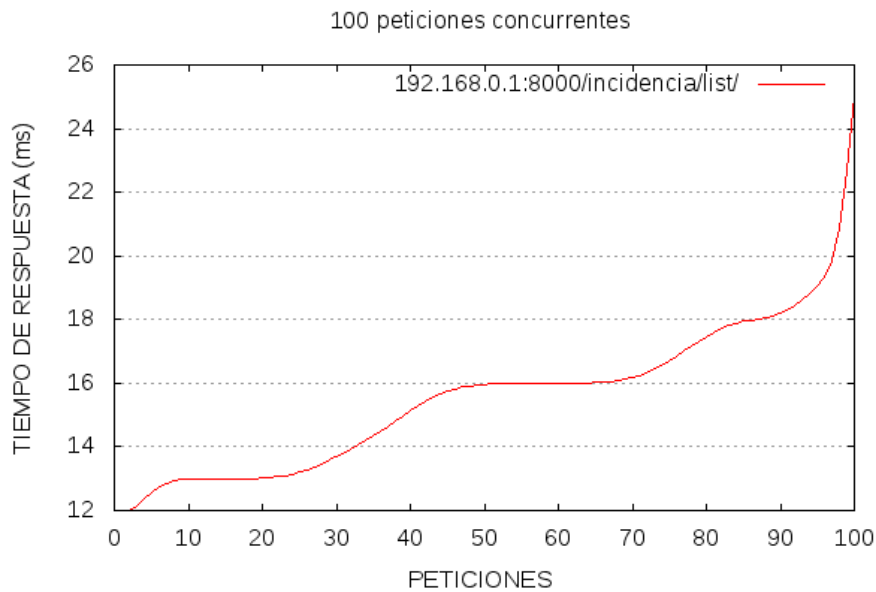


Figura 94. Gráfico de la prueba de carga.

- **Escenario 2. Prueba de carga con 500 peticiones.**

```
dianita@dianita-Inspiron-5437:~$ ab -g Documentos/resultados.csv -n 500 http://0.0.0.0:8000/incidencia/list/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 0.0.0.0 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Finished 500 requests

Server Software:      WSGIServer/0.1
Server Hostname:     0.0.0.0
Server Port:         8000

Document Path:       /incidencia/list/
Document Length:     22567 bytes

Concurrency Level:   1
Time taken for tests: 7.812 seconds
Complete requests:   500
Failed requests:     0
Non-2xx responses:   500
Total transferred:   11360500 bytes
HTML transferred:   11283500 bytes
Requests per second: 64.00 [#/sec] (mean)
Time per request:    15.624 [ms] (mean)
Time per request:    15.624 [ms] (mean, across all concurrent requests)
Transfer rate:       1420.12 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:    0    0  0.0    0
Processing: 12   16  2.4   16
Waiting:    12   15  2.4   15
Total:      12   16  2.4   16

Percentage of the requests served within a certain time (ms)
 50%    16
 66%    16
 75%    16
 80%    17
 90%    18
 95%    19
 98%    23
 99%    25
100%    35 (longest request)
dianita@dianita-Inspiron-5437:~$ █
```

Figura 95. Prueba de carga escenario 2.

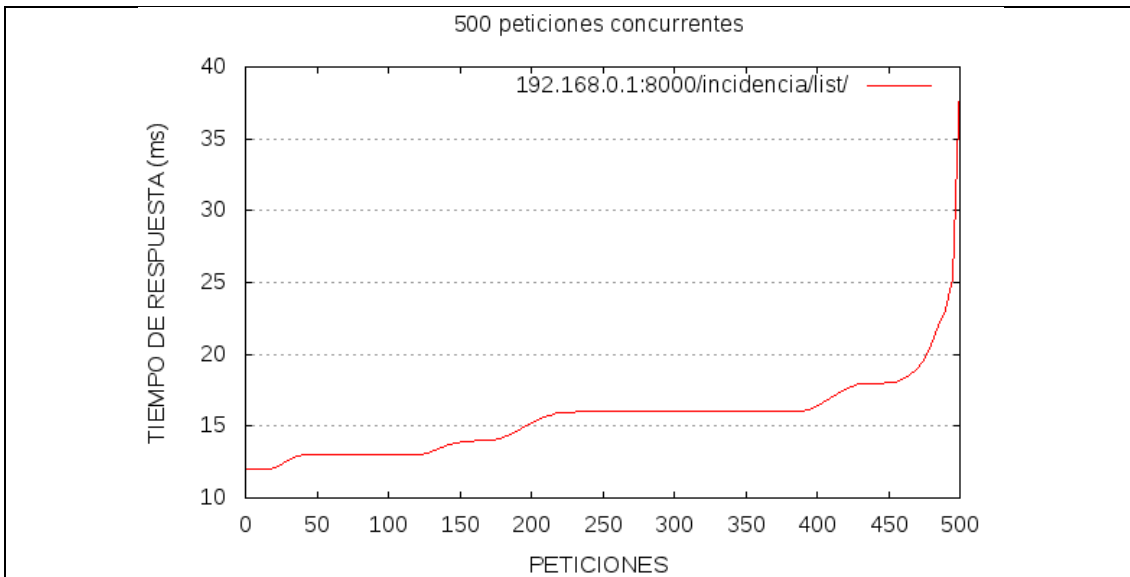


Figura 96. Gráfico de la prueba de carga.

- **Escenario 3. Prueba de carga con 1000 peticiones.**

```
dianita@dianita-Inspiron-5437:~$ ab -g Documentos/resultados.csv -n 1000 http://0.0.0.0:8000/incidencia/list/
This is ApacheBench, Version 2.3 <$Revision: 1528965 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
```

```
Benchmarking 0.0.0.0 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests
```

```
Server Software:      WSGIServer/0.1
Server Hostname:     0.0.0.0
Server Port:         8000

Document Path:       /incidencia/list/
Document Length:     22567 bytes

Concurrency Level:   1
Time taken for tests: 15.747 seconds
Complete requests:   1000
Failed requests:     0
Non-2xx responses:  1000
Total transferred:   22721000 bytes
HTML transferred:   22567000 bytes
Requests per second: 63.50 [#/sec] (mean)
Time per request:    15.747 [ms] (mean)
Time per request:    15.747 [ms] (mean, across all concurrent requests)
Transfer rate:       1409.05 [Kbytes/sec] received
```

```
Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:     0    0   0.0      0    0
Processing:  12   16   2.7     16   37
Waiting:     12   15   2.6     15   37
Total:       12   16   2.7     16   37
```

```
Percentage of the requests served within a certain time (ms)
 50%    16
 66%    16
 75%    17
 80%    18
 90%    18
 95%    19
 98%    23
 99%    26
100%    37 (longest request)
dianita@dianita-Inspiron-5437:~$ █
```

Figura 97. Prueba de carga – escenario 3.

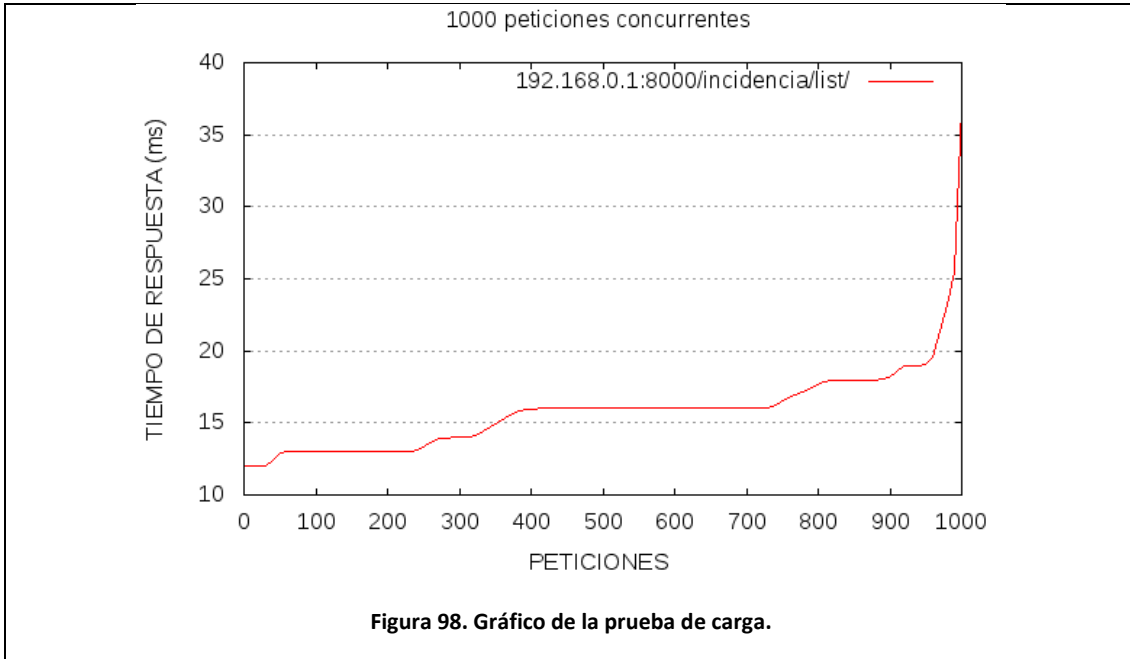


Figura 98. Gráfico de la prueba de carga.

Fecha: 30/11/2015	Tipo: Prueba de Carga	Código: PC-002
Roles de usuario:	Jefe Departamento o Asesor Técnico.	
Descripción:	Permite determinar si después de la carga de peticiones, el sistema continúa en función.	
Software previo:	Apache Brenc, Plot.	
Resultados:		
Luego de ejecutar el sistema con diferente carga, éste mantiene su correcto funcionamiento.		
Verificado: SI	Cumple: SI	

Tabla 57. Prueba de Carga – URL Lista de Incidencias.

6.4.2.4. RESUMEN DE CARGA DE DATOS

URL	REQUEST	TIEMPO POR TEST (SEC)	REQUEST POR SEGUNDO	TIEMPO POR PETICION (MS)	VALORACIÓN
INICIO	100	3.258	30.70	32.576	Criterio Aceptable
INICIO	500	16.578	30.16	33.157	
INICIO	1000	29.43	29.43	33.983	
LISTA DE INCIDENCIA	100	1.563	64.00	15.625	Criterio Aceptable
LISTA DE INCIDENCIA	500	7.812	64.00	15.624	
LISTA DE INCIDENCIA	1000	15.747	63.50	15.747	

Tabla 58. Resumen de carga de datos.

7. DISCUSIÓN

7.1. DESARROLLO DE LA PROPUESTA

El desarrollo del presente proyecto denominado “**SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA**”, permitió implementar un sistema web de entorno cliente servidor, utilizando el lenguaje de programación Python aplicando el patrón de diseño Modelo Vista Controlador a través del framework Django.

El objetivo general y los objetivos específicos se cumplieron en su totalidad, gracias a la aplicación de la metodología de desarrollo de software XP y la correcta utilización de métodos y técnicas de investigación científica. A continuación se detalla el cumplimiento de cada uno de ellos.

Objetivo Específico 1: *“Elaborar el análisis de requerimientos de los procesos administrativos referentes a la atención y solución de iteraciones de incidencias del departamento de Mantenimiento y Soporte Técnico.”*

Para cumplir con éste objetivo, se aplicó las técnicas de investigación científica para la recopilación de información, como son la observación directa y la entrevista, de aquí se obtuvo las historias de usuario, velocidad del proyecto, roles de XP, roles de usuario, requerimientos funcionales y no funcionales y definición del plan de iteraciones, para mayor detalle Ver el apartado 6.1. Fase de Planeación, 6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.1.6, 6.1.7.

Objetivo Específico 2: *“Diseñar el sistema utilizando como base el análisis de requerimientos funcionales bajo el uso de la metodología de desarrollo de software XP”*

Para dar cumplimiento a éste objetivo, se toma como punto de partida a los artefactos definidos en el Objetivo Específico 1 y el estudio realizado sobre la metodología ITIL (Ver el apartado 4.2.3. Metodología ITIL), para desarrollar la metáfora del sistema, el diagrama general de casos de uso, modelos conceptuales y tarjetas CRC (Ver el apartado 6.2. Fase de Diseño, 6.2.1, 6.2.2, 6.2.3, 6.2.4)

Objetivo Específico 3: *“Implementar los módulos de administración de procesos definidos en el análisis de requerimientos y diseño de software”*

Para cumplir éste objetivo, previamente se realizó la configuración de la tecnología utilizada (Ver el apartado 4.2.1. Arquitectura del sistema y 4.2.2. Herramientas y tecnología). Se ha codificado los módulos establecidos en el análisis de requerimientos y en el diseño, aplicando el patrón Model-View-Template, utilizando el lenguaje de programación Python con el framework Django y Postgresql en la capa de datos y aplicando el plan de iteraciones que propone el uso de la metodología XP (Ver el apartado 6.3. Fase de Codificación, 6.3.1, 6.3.2).

Objetivo Específico 4: *“Ejecutar un plan de validación al sistema Help Desk”*

Para cumplir con éste objetivo, se definió un plan de validación y verificación (Ver el apartado 6.4 Fase de pruebas, Literal 6.4.1, 6.4.2), donde se estableció las técnicas, requerimientos a validar y los tipos de pruebas a aplicar para finalmente ejecutar el plan con las pruebas de funcionalidad, de interfaz de usuario, y carga.

8. CONCLUSIONES

- Aplicar métodos y técnicas de investigación científica de forma correcta y oportuna permitieron recolectar información relevante, generar nuevos conocimientos y obtener la verdad sobre los hechos, para optimizar y automatizar el proceso de atención de incidencias del departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja.
- La utilización de convenciones que ITIL propone sobre el soporte al servicio efectivo, permitió en el sistema Help Desk establecer con norma la atención de incidencias y su solución ágil de modo que el proceso sufra el menor impacto posible y mantenga la continuidad del servicio.
- El sistema Help Desk integra acuerdos de nivel de servicio y, esto facilita que el tiempo de atención a incidencias se optimice, garantizando con métricas el rendimiento eficiente del personal durante la atención.
- La medición de la calidad del servicio del sistema Help Desk influencia a los usuarios que atienden incidencias para mantener el compromiso de cumplir con el objetivo de la calidad.
- Mediante la aplicación del plan de validación y verificación al sistema Help Desk, se pudo establecer el estado favorable de aceptabilidad, usabilidad, efectividad, y utilidad del sistema frente a los usuarios.
- La metodología de desarrollo de software XP facilitó la construcción del sistema Help Desk gracias a, su alto grado de adaptabilidad frente a los cambios realizados en la definición de requerimientos y, la producción de más desarrollo y menos documentación.
- La tecnología utilizada en el desarrollo fue la adecuada, puesto que estas herramientas poseen importantes características para adaptarse e integrarse unas con otras, y así construir un software que cumplió con los requerimientos funcionales y las necesidades el usuario.

9. RECOMENDACIONES

- Aplicar todas las convenciones del estándar ITIL para sistemas robustos y de mayor alcance o amplitud, debido a que la atención a incidencias no es el principal objetivo sino también el de emplear planes de contingencia, diagnóstico rápido en la toma de decisiones, el análisis cuantitativo y cualitativo de gestión de problemas y cambios.
- Ampliar el alcance de las funcionalidades de los sistemas comerciales de atención a incidencias, en cuanto a la valoración de la calidad de servicio, centros de asistencia múltiple, FAQ, acceso remoto y conexiones LDAP para acoplarse con sistemas a la medida.
- Integrar al sistema de atención a incidentes, como parte de los servicios tecnológicos de la Universidad Nacional de Loja, ya que permite la continuidad de los demás procesos operativos.
- Aplicar un plan de contingencia y seguridad de la información al Sistema Help Desk con la finalidad de analizar los posibles riesgos y afrontar desastres de diversos tipos.
- Utilizar el formato de presentación de las pruebas de éste documento, como un artefacto para la documentación de ejecución de pruebas de la metodología XP, puesto que presenta campos de información útil para determinar los errores encontrados y corregidos en cada iteración.
- Utilizar la tecnología aplicada en la construcción del software de éste proyecto, de manera especial el framework Django, debido a que, provee estructuras de codificación rápida como las “clases basadas en vistas” y la curva de aprendizaje es baja gracias a su extensa documentación disponible.
- Tomar como base éste proyecto, para desarrollar sistemas de gestión de problemas, que toma como base el reporte de incidencias para determinar cualitativamente y cuantitativamente las causas o factores que generan la problemática con la finalidad de proveer soluciones útiles y efectivas.

10. BIBLIOGRAFÍA

- [1] Universidad Nacional de Loja, «Universidad Nacional de Loja,» Unidad de Telecomunicaciones e Información, [En línea]. Available: <http://unl.edu.ec/>. [Último acceso: 27 Noviembre 2015].
- [2] CEINMER, «Herramientas Empresariales,» 2015. [En línea]. Available: <http://herramientasempresariales.com.mx/>. [Último acceso: 2015 Mayo 05].
- [3] J. v. Bon, Arjen de Jong, A. Kolthof y M. Pie, Fundamentos de ITIL V3, Van Haren, 2009.
- [4] Universidad de Girona, «Manual introductorio de ICONIX,» 2015. [En línea]. Available: <http://www.udg.edu/>. [Último acceso: 2015 Junio 15].
- [5] O. Amavizca, A. García y E. López, «Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software,» 2015. [En línea]. Available: <http://www.laccei.org/RefereedPapers/RP246.pdf>. [Último acceso: 14 Abril 2015].
- [6] P. Leteiler y C. Penadéz, «Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),» *Universidad Politécnica de Valencia*, p. 26, 2006.
- [7] C. Anay, Multimedia de Apoyo a la Enseñanza de la Metodología Rup, EAE, 2011.
- [8] M. Díaz, M. Pérez y L. Mendoza, «Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica,» 2015. [En línea]. Available: <http://www.academia-interactiva.com/doc/ise.pdf>. [Último acceso: 11 Junio 2015].
- [9] L. Echeverry y L. Delgado, «Caso práctico de la metodología ágil XP al desarrollo de software,» 2007. [En línea]. Available: <http://recursosbiblioteca.utp.edu.co/dspace/bitstream/11059/794/1/0053E18cp.pdf>. [Último acceso: 24 Noviembre 2015].
- [10] G. Quintanilla, «Verificación y validación,» 2008. [En línea]. Available: http://www.tidap.gob.mx/Presentaciones/talleres/p_quintanilla.ppt. [Último acceso: 02 Septiembre 2015].

- [11] C. J. Acuña, «Arquitecturas de software,» [En línea]. Available: <http://pegasus.javeriana.edu.co/~mad/Arquitecturas%20de%20SW.pdf>. [Último acceso: 04 Enero 2016].
- [12] A. Shekhar, *Django*, 2014.
- [13] Libros Web, El libro de Django, S/E, Agosto 2013.
- [14] W3Schools, «W3Schools,» 2016. [En línea]. Available: http://www.w3schools.com/css/css3_intro.asp. [Último acceso: 2016 Enero 02].
- [15] N. Downie, «Chartjs ORG,» 2015. [En línea]. Available: <http://www.chartjs.org/>. [Último acceso: 17 Agosto 2015].
- [16] Nodejs Foundation, «Node.js,» 2015. [En línea]. Available: <https://nodejs.org/en/>. [Último acceso: 22 Octubre 2015].
- [17] Leninmhs, «Redis, base de datos libre en memoria,» 2015. [En línea]. Available: <https://leninmhs.com>. [Último acceso: 28 Noviembre 2015].
- [18] Anishmenon, «Django-Realtime,» 2015. [En línea]. Available: <https://github.com/anishmenon/django-realtime>. [Último acceso: 22 Octubre 2015].
- [19] D. Jean-Mathieu y G. Frédéric, «Proyecto GLPI,» 2015. [En línea]. Available: <http://glpi-project.org/spip.php?rubrique25>. [Último acceso: 01 Diciembre 2015].
- [20] KMKey es Earcon, «Help Desk,» 2015. [En línea]. Available: http://www.kmkey.com/software_help_desk. [Último acceso: 02 Noviembre 2015].
- [21] M. Azorín, Métodos híbridos de investigación y dirección de empresas: ventajas e implicaciones, J.F: CuadEconDir, 2012.
- [22] S. Ceria, Ingeniería de software I. Casos de uso, un método práctico para explorar requerimientos, 2012.
- [23] J. S. S. Garreta, Ingeniería de proyectos informáticos, Graphic Group, 2003.
- [24] J. Yuni y C. Urbano, Técnicas para investigar 2, Córdoba, Argentina: Editorial Brujas, 2006.
- [25] A. Holovaty y K. M. Jacob , La guía definitiva de Django, Grupo Anaya Comercial, 2009, 2009.
- [26] D. Usama, J. Mlodgenski y K. Roibal, PostgreSQL Server Programming, Mumbai: Packt Birmingham, 2015.

- [27] G. J. Diego, El gran libro de HTML5, CSS3 y Javascript, Barcelona: Marcombo, 2012.
- [28] M. Otto y J. Thornton, Bootstrap 3 el manual oficial, Libros Web, 2015.
- [29] P. Bell, Introducing Github, Graveinstein. USA: O'Reilly Media Inc., 2015.
- [30] J. Joskowicz, «Reglas y prácticas en Extreme Programming,» 10 Febrero 2008. [En línea]. Available: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>. [Último acceso: 28 Enero 2015].
- [31] N. Bagnasco y N. Maurizio, «Plan de validación y verificación,» España, 2014.

11. ANEXOS

ANEXO 1. CERTIFICADO DE ACEPTACIÓN DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN.



UNL
UNIVERSIDAD
NACIONAL
DE LOJA

*Unidad de
Telecomunicaciones e
Información*

Certifica

Que el proyecto "SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA", desarrollado por la egresada Diana Alexandra Morocho Puchaicela, ha sido socializado, revisado y evaluado en detalle por el personal de la Unidad de Telecomunicaciones e Información, determinando el cumplimiento de requerimientos, especificaciones técnicas y normativas para su aceptación.

Es cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento.

Loja, 30 de Noviembre del 2015.


Milton Labanda, Mg.
DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN



Ciudad Universitaria "Guillermo Falconi Espinosa", La Argelia, Loja - Ecuador
Teléfonos: 07 2547252 Ext.: 126, Email: direccion.uti@unl.edu.ec, Web: <http://www.unl.edu.ec>

Figura 99. Certificado de aceptación UTI-1.

ANEXO 2. CERTIFICADO DE ACEPTACIÓN RESPONSABLE DE VERIFICACIÓN DE REQUERIMIENTOS.



UNL
UNIVERSIDAD
NACIONAL
DE LOJA

*Unidad de
Telecomunicaciones e
Información*

Certifica

Que el proyecto "SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA", desarrollado por la egresada Diana Alexandra Morocho Puchaicela, ha sido socializado, revisado y evaluado en detalle, determinando el cumplimiento de requerimientos, especificaciones, técnicas y normativas para su aceptación. Por tal motivo certifico que la aplicación es funcional y lista para su implantación.

Es cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento.

Loja, 30 de Noviembre del 2015.



Mabel Rodríguez, Ing.
**PERSONAL ADMINISTRATIVO DE LA UNIDAD DE
TELECOMUNICACIONES E INFORMACIÓN**



Ciudad Universitaria "Guillermo Falconí Espinosa", La Argelia, Loja - Ecuador
Teléfonos: 07 2547252 Ext.: 126, Email: direccion.uti@unl.edu.ec, Web: <http://www.unl.edu.ec>

Figura 100. Certificado de aceptación UTI-2.

ANEXO 3. CERTIFICADO DE TRADUCCIÓN.



THE CANADIAN HOUSE CENTER

THE CANADIAN HOUSE CENTER


El que suscribe, en representación de **THE CANADIAN HOUSE CENTER CIA. LTDA**, con RUC N° 1191723364001, el cual está aprobado por el **Ministerio de Educación del Ecuador** según resolución Ministerial N° 320 - 15.

CERTIFICA.-

Que el resumen de tesis titulada **“SISTEMA HELP DESK PARA EL DEPARTAMENTO DE MANTENIMIENTO Y SOPORTE TÉCNICO DE LA UNIVERSIDAD NACIONAL DE LOJA”** realizado por **DIANA ALEXANDRA MOROCHO PUCHAICELA** con cedula de identidad 1104662745 estudiante de la **CARRERA DE INGENIERIA EN SISTEMAS** de la Universidad Nacional de Loja, ha sido debidamente traducido por el Lic. Ross Sampayo docente coordinador de nuestra prestigiosa entidad especializada en la buena enseñanza del idioma inglés.

Se expide el presente documento, de acuerdo a la Ley, para los fines necesarios.

Loja, 02 de Diciembre de 2015



Lic. Ross Sampayo
COORDINADOR GENERAL
THE CANADIAN HOUSE CENTER

www.thecanadianhousecenter.com
Loja Matriz: Venezuela 19-77 Entre José María Peña y Av. Pío Jaramillo Alvarado • Loja-Ecuador • Telefonos: 2584334/2584450
Loja Centro: Miguel Bolfrio 14-35 Entre Bolívar y Sucre • Loja-Ecuador • Teléfono: 2571800



 /CHCLoja
 @CHCLoja

Figura 101. Certificado de traducción.

ANEXO 4. ORGANIGRAMA ESTRUCTURAL DE LA UNIVERSIDAD NACIONAL DE LOJA.

Organigrama estructural de la Universidad Nacional de Loja

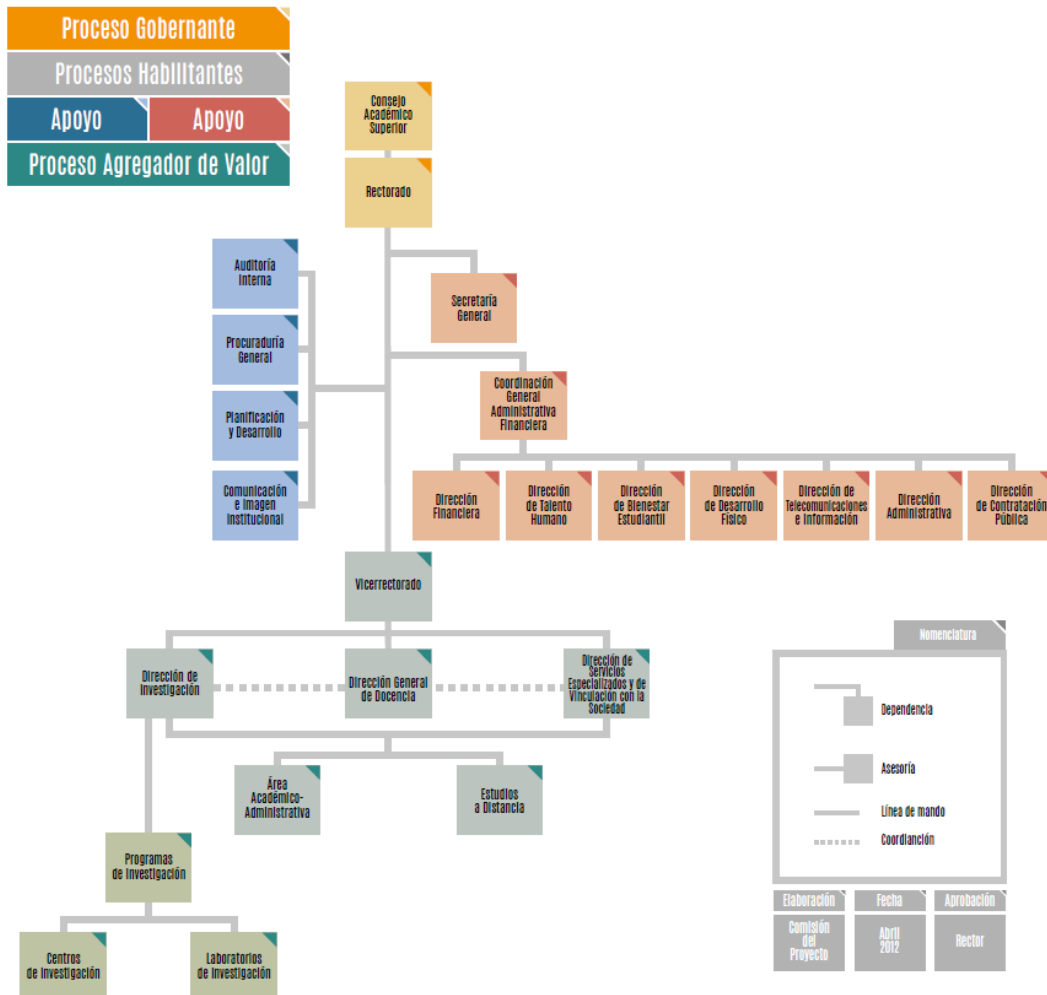



Figura 102. Orgánico estructural de la Universidad Nacional de Loja

ANEXO 5. ENTREVISTA DIRIGIDA A LA ING. MABEL RODRÍGUEZ



UNIVERSIDAD NACIONAL DE LOJA
Carrera de Ingeniería en Sistemas
"Sistema Help Desk para el departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja"

ENTREVISTA INFORMATIVA DIRIGIDA

La presente entrevista tiene como objetivo obtener información de la situación actual y de los procesos administrativos del Departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja, con la finalidad de desarrollar un sistema Help Desk para su gestión desde la Unidad de Telecomunicaciones e Información y como requisito previo a la obtención del título de Ingeniería en Sistemas.

1. ¿Qué función desempeña en el Departamento de Mantenimiento y Soporte Técnico?

La función que se desempeña aquí, es realizar el mantenimiento preventivo y correctivo de todos los equipos informáticos de la institución.

2. ¿Cómo está distribuido el personal que labora en esta instancia?

En este momento como estamos con un número tan reducido, es decir tres técnicos para toda la universidad, la persona que está disponible, o con menos de trabajo, va a solucionarlo en cuanto toca ir a cada una de las dependencias y sino más bien pedimos que los equipos los traigan a la oficina.

3. ¿Cuáles son los servicios que ofrecen en éste departamento?

El mantenimiento de todos los equipos informáticos, en cuanto a software nos referimos a la instalación de los programas básicos: el sistema operativo, ofimática y todos los utilitarios y también ayudamos a los sistemas informáticos que fueron realizados en el departamento como es el de la gestión administrativa, bodega general, esos sistemas también damos mantenimiento aquí.

4. ¿Cómo se reciben las solicitudes que requieren de asistencia técnica?

Sea por llamada telefónica o por oficio.

5. ¿Cuál es el proceso que debe seguir el usuario que requiera de asistencia técnica?

Cuando es por oficio le envían al director Milton Palacios y él a su vez determina qué tipo de asistencia es: redes u otra, y él lo sumilla y lo envía al departamento, y si es por teléfono, también a veces lo llaman a él directamente, o nos llaman a nosotros.

6. ¿Qué acción toma usted cuando las solicitudes de asistencia técnica sobrepasan el plan de trabajo diario?

Les pedimos paciencia, tienen que esperar es que no hay otra manera de solucionar, aunque nosotros al principio del año realizamos un cronograma de actividades que se van a desarrollar durante el año sobre todo para laboratorios y bibliotecas entonces en la fecha que nos corresponde hacemos el mantenimiento estemos como estemos.

Figura 103. Entrevista dirigida 1 de 2

7. ¿Cómo se establece el nivel de prioridad de las solicitudes de asistencia técnica?

Para nosotros todo es muy importante, pero si hay preferencias, por ejemplo en contabilidad el proceso no puede pararse, o un docente en clase un equipo no puede pararse entonces pedimos que nos digan de donde es, que trabajo está haciendo, aunque nosotros ya los conocemos, entonces ya sabemos la prioridad que se requiere.

8. ¿A qué tipo de problemas se brinda solución?

Por lo regular tenemos la instalación eléctrica en la universidad un poco mal, que eso ya no nos corresponde a nosotros pero igual hemos pedido por oficio que se arreglen las tomas eléctricas, esto ocasiona que se quemen muchas fuentes de poder entonces ese es un problema gravísimo, las máquinas no encienden y sino los virus que ese es otro problema principal.

9. ¿Qué funciones les asignan a las personas que laboran aquí como pasantes?, ¿tienen que ir a la ubicación de los equipos? o solamente están encargados de la parte técnica.

A los pasantes se los capacita rápidamente, pero se les da indicaciones. El usuario que solicita menciona los problemas sobre el equipo, entonces al pasante se le da una probabilidad del problema o un diagnóstico preliminar, por ejemplo cuando el diagnóstico es sobre virus, se le da las indicaciones al pasante sobre qué acciones tomar sobre el equipo, y si aún el equipo sigue mal entonces se les solicita que traigan el equipo. Y así, se les da más o menos el diagnóstico que puede ser del equipo para que ellos puedan hacer, porque tampoco se les puede asignar una responsabilidad muy grande.

10. ¿Cómo identifica la dependencia a la que corresponde el usuario que solicita el servicio de soporte?

Primero el nombre del usuario, el responsable del equipo, que dependencia trabaja y que área trabaja, y como aquí se trabaja varios años nos dan el nombre y rápidamente los identificamos.

11. ¿Qué datos solicitan para registrar el equipo?

Se les pide el nombre del custodio o responsable, el nombre de la persona que trae, la dependencia donde trabaja, el área, y el problema que más o menos tiene el equipo.

MABEL RODRIGUEZ / E.

Ing. Mabel Rodríguez

Encargada del Departamento de Mantenimiento y Soporte Técnico
de la Universidad Nacional de Loja



Figura 104. Entrevista dirigida 2 de 2

ANEXO 6. REGISTRO ANECDÓTICO.

Registro anecdótico [24], herramienta de aplicación de la técnica de la observación.


FICHA DE OBSERVACIÓN	
OBSERVADO:	Departamento de Soporte Técnico NRO.: 001
OBSERVADOR:	Morocho Puchaicela Diana Alexandra FECHA: 09-Dic-2014
LUGAR:	Universidad Nacional de Loja
CONTEXTO:	Personas y dependencias relacionadas al departamento de Soporte Técnico.
	<ul style="list-style-type: none"> - En el departamento, circulan varias personas: <ul style="list-style-type: none"> o Personal administrativo que labora en ese departamento. o Estudiantes que realizan las actividades de pasantías. o Personas que dejan sus equipos solicitando asistencia técnica. o Personal administrativo que labora en la Dirección de Telecomunicaciones e Información. - El departamento de Soporte Técnico brinda asistencia a todas las dependencias de la Universidad Nacional de Loja
ANEXO:	
	
<p>Figura 105. Ficha de observación 001.</p>	

Tabla 59. Registro anecdótico de las personas y dependencias relacionadas al departamento de Soporte Técnico.

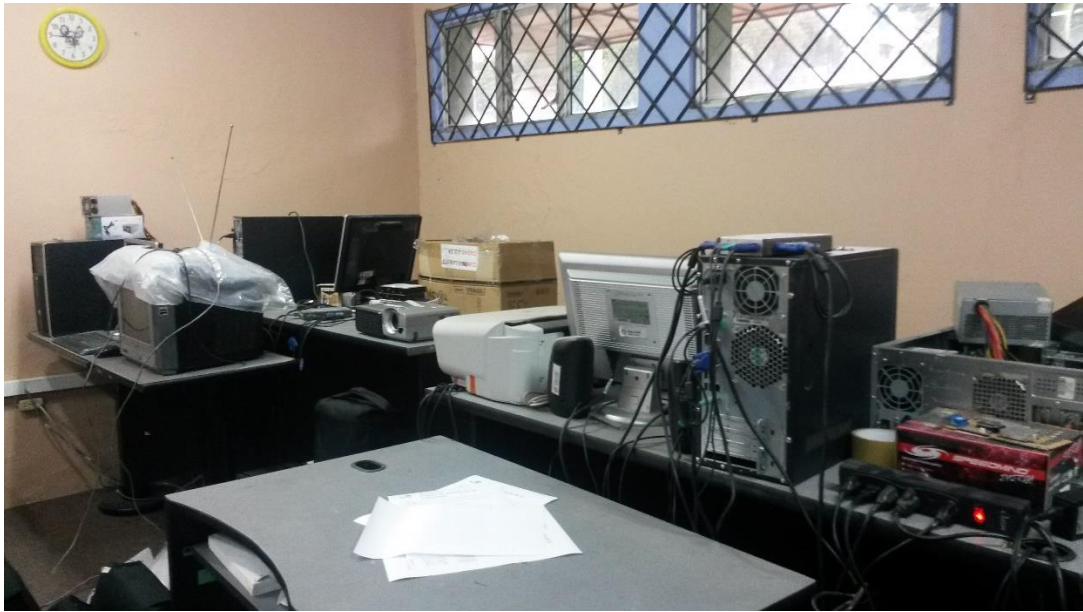
FICHA DE OBSERVACIÓN		
OBSERVADO:	Departamento de Soporte Técnico	FECHA: 10-Dic-2014
OBSERVADOR:	Morocho Puchaicela Diana Alexandra	NRO.: 002
LUGAR:	Universidad Nacional de Loja	
CONTEXTO:	Tipo de asistencia técnica.	
	<ul style="list-style-type: none"> - En el departamento de Soporte Técnico se realizan dos tipos de asistencia: en Software y en Hardware. - En asistencia de software se realizan configuraciones sobre el sistema operativo del equipo además de la instalación de programas utilitarios y complementarios. - En asistencia de hardware se ejecuta la limpieza y sustitución de piezas defectuosas, sobre computadores portátiles y de escritorio, monitores, teclados, mouses, impresoras y otros artefactos. 	
ANEXO:		
		
	Figura 106. Ficha de observación 002.	

Tabla 60. Registro anecdótico del tipo de asistencia que ofrecen en el departamento de Soporte Técnico.

FICHA DE OBSERVACIÓN		
OBSERVADO:	Departamento de Soporte Técnico	FECHA: 11-Dic-2014
OBSERVADOR:	Morocho Puchaicela Diana Alexandra	NRO.: 003

LUGAR:	Universidad Nacional de Loja
CONTEXTO:	Solicitud de asistencia técnica.
	<ul style="list-style-type: none"> - El personal administrativo que labora en el departamento de Soporte Técnico recepta llamadas telefónicas de personas que laboran en otra dependencia de la Universidad Nacional de Loja. El flujo de la llamada telefónica consiste en recibir los datos suficientes para identificar a la persona solicitante, y los problemas del equipo defectuoso. - Dependiendo del nivel de urgencia con el que se requiera reparar el equipo, se traslada el personal o se pide que se traslade el equipo hasta el departamento de Soporte Técnico. - El personal se turna para recibir las llamadas y dar solución al problema. - También se reciben solicitudes de asistencia por medio de oficios generalmente dirigidos por el Ing. Milton Palacios que es el coordinador de la Dirección de Tecnología e Información. - Algunas personas no realizan llamada ni envían oficios sino más bien se trasladan directamente al departamento para ser atendidas. - En este día, la carga de trabajo ha llegado a su punto máximo por lo que a las solicitudes son aplazadas para otro día. - Se ha recibido una solicitud del departamento de Contabilidad donde se requiere asistencia de carácter urgente, por lo que se ha tenido que dejar de lado las solicitudes en cola pendientes de ser atendidas.

Tabla 61. Registro anecdótico de solicitud de asistencia técnica.

FICHA DE OBSERVACIÓN	
OBSERVADO:	Departamento de Soporte Técnico FECHA: 13-Dic-2014
OBSERVADOR:	Morocho Puchaicela Diana Alexandra NRO.: 004
LUGAR:	Universidad Nacional de Loja
CONTEXTO:	Recepción de equipos.
	<ul style="list-style-type: none"> - La persona encargada de recibir los equipos, registra los datos requeridos en un documento denominado Reporte Técnico y revisa el estado de ingreso del equipo defectuoso.
ANEXO:	

UNIVERSIDAD NACIONAL DE LOJA
SECCIÓN DE MANTENIMIENTO ELECTRÓNICO E INFORMÁTICO
Telf: 072547252 ext. 129

REPORTE TÉCNICO

ÁREA: _____ FECHA REPORTE: _____ TELÉFONO: _____
 DEPENDENCIA: _____
 RESPONSABLE EQUIPO: _____
 CONTACTO: _____
 EQUIPO: _____ FECHA ATENCIÓN: _____
 MARCA: _____ REG. BODEGA: _____ SELLO GARANTIA: SI NO
 PROBLEMA: _____
 ACCIONES REAIZADAS: _____

 Hora de llegada: _____
 Hora de salida: _____ Ing. de Servicio _____ Firma del cliente _____
 C.I. : _____

Satisfacción del cliente:	Excelente <input type="checkbox"/>	Bueno <input type="checkbox"/>	Malo <input type="checkbox"/>	Deficiente <input type="checkbox"/>
---------------------------	---------------------------------------	-----------------------------------	----------------------------------	--

Figura 107. Ficha de observación 004.

Tabla 62. Registro anecdótico de recepción de equipos.

ANEXO 7. MODELO DEL DOMINIO

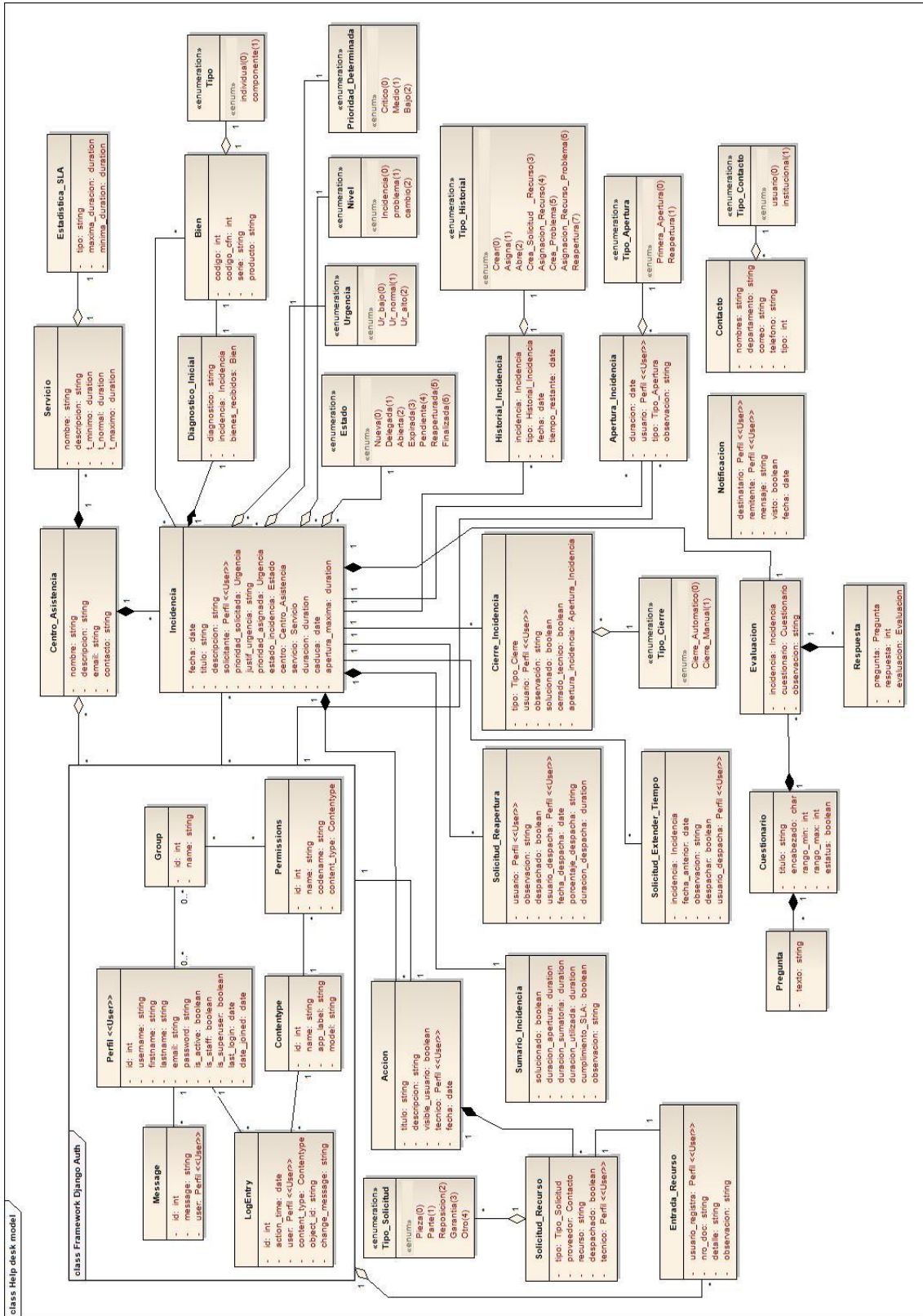


Figura 108. Modelo del dominio.

ANEXO 8. MODELO DE CLASES

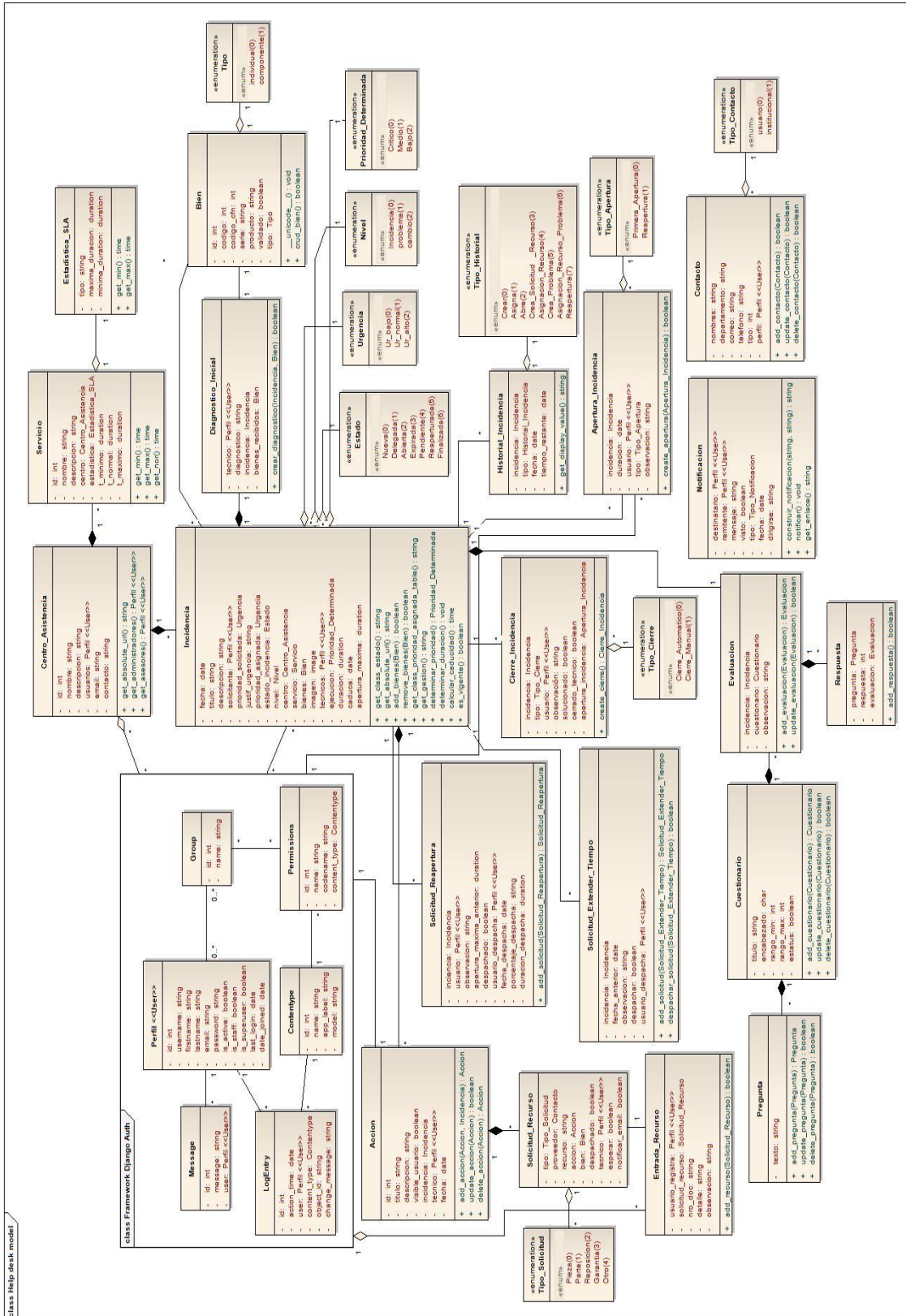


Figura 109. Modelo de clases.

ANEXO 10. MODELO DE ENCUESTA APLICADA.



Universidad Nacional de Loja

Carrera de Ingeniería en Sistemas

"Sistema Help Desk para el departamento de Mantenimiento y Soporte Técnico de la Universidad Nacional de Loja"

ENCUESTA DIRIGIDA

Encuestador: Diana Alexandra Morocho Puchaicela.	Fecha:	
Encuestado:	Firma:	
Presentación.		
La respuesta a la presente encuesta tiene como finalidad la comprobación de la información registrada durante la validación y verificación de las pruebas de interfaz de usuario aplicadas al Sistema Help Desk, y determinar el cumplimiento del proceso de atención a incidentes, en la presentación de resultados del proyecto de titulación previo a la obtención del título de Ingeniera en Sistemas		
Valoración del Sistema Help Desk		
1. Marque con una X, si considera a la premisa como Verdadera (SI) o Falsa (NO).		
	SI	NO
• La utilización del sistema Help Desk es sencilla. Cada componente es de fácil localización.		
• La aplicación es de gran utilidad para la automatización de atención a incidencias.		
• La velocidad del ingreso de datos y notificaciones es rápida e inmediata.		
• El sistema ayuda al usuario a atender incidencias de forma rápida, reduciendo los errores.		
2. Marque con una X según su criterio:		
Espacio único para el Usuario Final.		
• ¿Qué tan complejo es reportar una incidencia? Nada complejo <input type="checkbox"/> Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>		
• Reportar bienes en una incidencia es una tarea Fácil <input type="checkbox"/> Compleja <input type="checkbox"/> Difícil <input type="checkbox"/>		
• Calificar la atención brindada al finalizar la incidencia resulta Beneficioso <input type="checkbox"/> Útil <input type="checkbox"/> Inadecuado <input type="checkbox"/>		
Espacio único para el Asesor Técnico.		
• ¿Qué tan complejo es atender una incidencia que ha sido asignada? Nada complejo <input type="checkbox"/> Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>		
• Registrar acciones sobre una incidencia es una tarea Fácil <input type="checkbox"/> Compleja <input type="checkbox"/> Difícil <input type="checkbox"/>		
• Realizar el diagnóstico inicial de la incidencia y de los bienes reportados es Muy útil <input type="checkbox"/> Poco útil <input type="checkbox"/> Nada útil <input type="checkbox"/>		
Espacio único para el Jefe Departamento.		
• ¿Qué tan complejo es asignar una incidencia al Asesor Técnico? Nada complejo <input type="checkbox"/> Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>		
• Verificar la calidad de servicio que brinda el usuario Asesor Técnico es una tarea Fácil <input type="checkbox"/> Compleja <input type="checkbox"/> Difícil <input type="checkbox"/>		
• Asignar un tiempo máximo de apertura sobre las incidencias resulta Beneficioso <input type="checkbox"/> Útil <input type="checkbox"/> Inadecuado <input type="checkbox"/>		
Espacio único para el Administrador.		
• ¿Qué tan complejo es administrar el panel de configuraciones? Nada complejo <input type="checkbox"/> Medianamente complejo <input type="checkbox"/> Muy complejo <input type="checkbox"/>		
• Agregar usuarios Jefe de Departamento y Asesor Técnico al centro de asistencia es una tarea Fácil <input type="checkbox"/> Complejo <input type="checkbox"/> Difícil <input type="checkbox"/>		
• Registrar varios centros de asistencia para la atención de incidencias resulta Beneficioso <input type="checkbox"/> Útil <input type="checkbox"/> Inadecuado <input type="checkbox"/>		

Muchas gracias por su amabilidad y por el tiempo dedicado a contestar esta encuesta.

Figura 111. Formato de la encuesta aplicada