

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención del título de:
INGENIERA DE SISTEMAS**

**TEMA:
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO
PARA GESTIONAR LOS TRABAJOS DE HELP DESK DEL ÁREA DE
TECNOLOGÍA DE LA COOPERATIVA CODESARROLLO BASADO EN ITIL
Y SILVERLIGHT**

**AUTORA:
XIMENA ALEXANDRA CATOTA TOCA**

**DIRECTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

Quito, abril de 2015

DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL TRABAJO DE TITULACIÓN

Yo, autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de la autora.

Ximena Alexandra Catota Toca
C.C. 172022186-8

DEDICATORIA

Dedico el presente trabajo principalmente a Dios por ser mi soporte en todo momento y haberme permitido llegar hasta este momento tan importante en mi formación profesional.

A mis padres por la dedicación, ejemplo de lucha, consejos, comprensión, amor en los momentos difíciles, por ayudarme con los recursos necesarios para estudiar, por la formación en valores y por su apoyo incondicional.

A mis hermanos y amigos que de alguna manera u otra me brindaron su apoyo.

Ximena Alexandra Catota Toca

AGRADECIMIENTO

A mis profesores de la Universidad Politécnica Salesiana que aportaron en mi formación profesional y humana especialmente a mi tutor de tesis el ingeniero Franklin Hurtado por haberme guiado con gran excelencia en el desarrollo de mi proyecto de titulación.

Ximena Alexandra Catota Toca

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
PRESENTACIÓN Y FORMULACIÓN DEL PROBLEMA	2
1.1 Título del tema	2
1.2 Planteamiento del problema.....	2
1.3 Formulación de objetivos.....	3
1.4 Justificación del proyecto	3
1.5 Alcance	4
CAPÍTULO 2	6
MARCO TEÓRICO	6
2.1 Fundamentos de ITIL.....	6
2.1.1 Historia y concepto de ITIL	6
2.1.2 Características de ITIL	7
2.2 Ciclo de vida de ITIL.....	7
2.2.1 Estrategia de servicio (SE)	7
2.2.2 Diseño del servicio	7
2.2.3 Transición de servicio	8
2.2.4 Operación del servicio.....	12
2.2.5 Mejora continua.....	19
2.3 Definición de help desk o mesa de ayuda.....	19
2.4 Metodología XP (Programación Extrema)	20
2.4.1 Fundamentos	20
2.4.2 Principios básicos.....	21
2.4.3 Fases programación extrema XP.....	22
2.4.4 Ventajas y desventajas	25
2.4.5 Diseño de software	25
2.5 Tecnología a emplearse	26
2.5.1 Programación en capas.....	26
2.5.2 Windows Communication Foundation (WCF)	27

2.5.3	Language Integrated Query (LINQ).....	28
2.5.4	Lenguaje de marcado XAML.....	29
2.5.5	Silverlight.....	30
2.5.6	Expression blend.....	32
2.6	Especificación de requerimientos.....	32
2.6.1	Historias de usuario.....	32
2.7	Listado de entregables.....	47
CAPÍTULO 3.....		48
ANÁLISIS Y DISEÑO.....		48
3.1	Análisis.....	48
3.1.1	Situación actual de Bancodesarrollo.....	48
3.1.2	Análisis de los procesos actuales.....	50
3.2	Diseño del sistema.....	57
3.2.1	Diseño de datos.....	57
3.2.2	Diseño arquitectónico.....	59
3.2.3	Diseño de interfaces.....	61
3.3	Arquitectura del sistema.....	68
3.3.1	Diagrama de navegación.....	69
CAPÍTULO 4.....		71
CONSTRUCCIÓN Y PRUEBAS.....		71
4.1	Construcción.....	71
4.1.1	Estándares de programación.....	71
4.1.2	Código relevante.....	72
4.1.3	Pruebas.....	75
CONCLUSIONES.....		91
RECOMENDACIONES.....		93
LISTA DE REFERENCIAS.....		94
ANEXOS.....		96

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Procesos de la gestión de incidentes.....	14
<i>Figura 2:</i> Gestión de peticiones.....	17
<i>Figura 3:</i> Fases de la metodología XP.	22
<i>Figura 4.</i> Fases de XP.	24
<i>Figura 5:</i> Diseño de software.....	26
<i>Figura 6.</i> Programación por capas.	27
<i>Figura 7.</i> Diseño de Silverlight.	31
<i>Figura 8.</i> Organigrama de Bancodesarrollo.	49
<i>Figura 9:</i> Diagrama del proceso actual de la institución	54
<i>Figura 10.</i> Diagrama del sistema web mesa de ayuda.	56
<i>Figura 11.</i> Diseño de base de datos.....	58
<i>Figura 12.</i> Diagrama de clases	59
<i>Figura 13.</i> Diagrama de procesos.....	60
<i>Figura 14.</i> Pantalla que permite al usuario ingresar al sistema.....	61
<i>Figura 15.</i> Interfaz de inicio.....	61
<i>Figura 16.</i> Interfaz de registro de usuarios	62
<i>Figura 17.</i> Interfaz de administrador de roles	63
<i>Figura 18.</i> Interfaz de ingreso de solicitud.....	63
<i>Figura 19.</i> Interfaz de aprobación de requerimientos.	64
<i>Figura 20.</i> Interfaz de asignación de solicitud	64
<i>Figura 21.</i> Interfaz de ingreso de incidente.....	65
<i>Figura 22.</i> Interfaz para la asignación de incidentes.....	65
<i>Figura 23.</i> Interfaz de seguimiento del incidente	66
<i>Figura 24.</i> Interfaz de evaluación del incidente	66
<i>Figura 25.</i> Interfaz de la base del conocimiento	67
<i>Figura 26.</i> Interfaz para la consulta de solicitudes.....	67
<i>Figura 27.</i> Diseño de la arquitectura por capas.....	68
<i>Figura 28.</i> Diagrama de navegación	70
<i>Figura 29.</i> DataContext: archivo de configuración para conexión de datos.	72
<i>Figura 30.</i> Contrato de servicios - WCF	73
<i>Figura 31.</i> Código para búsqueda de coincidencias	74

Figura 32. Código que encuentra posibles coincidencias en la base del conocimiento. 75

ÍNDICE DE TABLAS

Tabla 1. <i>Comparativa entre casos de uso e historias de usuario</i>	23
Tabla 2. <i>Ventajas y desventajas de XP</i>	25
Tabla 3. <i>Listado de entregables</i>	47
Tabla 4. <i>Listado de hardware y software de Bancodesarrollo</i>	49
Tabla 5. <i>Actores del sistema mesa de ayuda</i>	55
Tabla 6. <i>Métodos DataContext</i>	73
Tabla 7. <i>Matriz de resultados pruebas funcionales</i>	76
Tabla 8. <i>Resumen de pruebas de stress</i>	90

ÍNDICE DE ANEXOS

Anexo 1. Pruebas de stress	94
Anexo 2. Implementación del sistema.....	100

RESUMEN

Considerando la importancia de brindar un buen soporte técnico que ayude a mantener la calidad de los servicios de la institución Codesarrollo. El presente trabajo tiene como objetivo el diseño y desarrollo de una aplicación web para la gestión de los trabajos de help desk del departamento de tecnología de dicha institución usando como guía ITIL V3 y para el diseño de la interfaz gráfica Silverlight. Esta gestión se ha clasificado en dos tipos de peticiones: requerimientos nuevos e incidentes cuyas especificaciones están definidas por el usuario basándose en lineamientos de ITIL V3, así como también el uso de la base del conocimiento que ayuda al usuario a resolver sus problemas por si solo.

Para el desarrollo de este proyecto se emplea la metodología ágil XP, una metodología que se centra principalmente en cumplir con las necesidades del usuario ya que este es parte del equipo por lo tanto está involucrado en el proceso de construcción del sistema.

ABSTRACT

Considering the importance of providing good support to help maintain the quality of services Codesarrollo institution. This paper aims at designing and developing a web application for managing help desk jobs technology department of that institution using as a guide ITIL V3 and design Silverlight GUI. This management is classified into two types of requests: new requirements and incidents whose specifications are user based on guidelines of ITIL V3 defined, as well as the use of the knowledge base that helps users solve their problems by itself.

For the development of this project Agile XP, a methodology that focuses primarily on meeting user needs as this is part of the team thus is involved in the process of building the system is used.

INTRODUCCIÓN

Actualmente las instituciones consideran, que el éxito está directamente relacionado con la calidad y agilidad en la entrega de sus servicios por lo cual dichas organizaciones frecuentemente hacen uso de tecnologías, mismas que permiten lograr su meta.

Así también ante lo mencionado, se observa cotidianamente el incremento de inquietudes relacionadas con la utilización de estas tecnológicas por lo que las organizaciones han preparado un equipo llamado “*soporte técnico*” quienes son encargados de responder a las novedades y quejas en los servicios.

Por tanto para dar respuesta eficiente a cada solicitud es necesario disponer de una adecuada gestión de eventos y para llevar a cabo esta tarea a más de contar únicamente con el personal capacitado, es preciso el apoyo tecnológico para cumplir con esta misión.

En tal sentido, el objetivo del presente proyecto de titulación es entregar al departamento de tecnología de la información de Bancodesarrollo, un sistema informático basado en las necesidades específicas del banco y en los lineamientos que sugiere ITIL (Librería de Infraestructura de Tecnologías de Información.), en cuanto a la entrega de servicios, gestión de eventos, gestión de incidentes, y gestión del conocimiento ya que la organización cuenta con el equipo de soporte técnico.

Este sistema informático, permitirá al usuario en muchas ocasiones resolver sus problemas por sus propios medios y que la entidad logre el éxito en la calidad y agilidad de sus servicios.

CAPÍTULO 1

PRESENTACIÓN Y FORMULACIÓN DEL PROBLEMA

1.1 Título del tema

Análisis, diseño e implementación de un sistema informático para gestionar los trabajos de help desk del área de tecnología de la Cooperativa Codesarrollo basado en ITIL y Silverlight.

1.2 Planteamiento del problema

En la actualidad la Cooperativa de Ahorro y Crédito “CODESARROLLO” se ha expandido notablemente por todo el país, muestra de ello es la apertura de varias agencias en diferentes provincias y la reciente conversión a banco en febrero del 2014, tomando el nombre de “Bancodesarrollo”. (bancodesarrollo.fin.ec, 2015)

Ante dicha expansión, el Banco se ha visto en la necesidad de adquirir e implementar nuevos Hardware y Software, con el objetivo de cumplir con las metas propuestas ante este nuevo reto de brindar un servicio de calidad como Banco.

El aumento hardware y software implica la presencia de un número mayor de alertas sobre requerimientos, tareas o incidentes, mismos que son observados por los clientes internos quienes reportan al departamento de tecnología para recibir una atención oportuna a sus necesidades, sobre todo para que el ritmo y la calidad de atención del banco no se vea afectada. Lamentablemente se han dado situaciones que no se reportan o que no reciben la respuesta oportuna en el tiempo adecuado y el desarrollo normal de la actividad del banco se ha visto alterada. Cabe recalcar que no existe una documentación formal sobre los *requerimientos*, protocolos exigidos por la Superintendencia de bancos poniendo en riesgo a la organización por posibles llamados de atención.

Ante tal problema existe una propuesta de solución que es la implementación de un sistema informático que se encargue de receptar todas las alertas, de que estas, reciban una atención oportuna, generar en el cliente interno la capacidad de responder a ciertos problemas por sí mismos y crear documentos formales de los requerimientos; logrando así un desempeño de los servicios del Bancodesarrollo con agilidad, calidad y excelencia.

1.3 Formulación de objetivos

Objetivo General

Desarrollar un sistema informático para el departamento de tecnología de la Cooperativa Codesarrollo que permita gestionar y evaluar los procesos de solución de incidentes, tareas, y/o requerimientos basado en ITIL y usando Silverlight.

Objetivos Específicos

1. Determinar las posibles debilidades en los procesos llevados a cabo por el área de tecnología y proponer mejoras basadas en ITIL V3.
2. Construir el sistema informático propuesto basado en ITIL V3 y aplicaciones Silverlight.
3. Implementar un mecanismo de control automatizado del ingreso del usuario al manejo de información relacionado a las diferentes asignaciones de incidentes, tareas y/o requerimientos.
4. Realizar un seguimiento automatizado de los diferentes incidentes, tareas y/o requerimientos asignados al personal del departamento de tecnología.
5. Llevar un registro permanente y actualizado, sobre el estado en que se encuentran los incidentes, tareas y/o requerimientos gestionados por el departamento de tecnología.
6. Optimizar el tiempo que se utiliza en atender los procesos de solución de incidentes, tareas y/o requerimientos.

1.4 Justificación del proyecto

Actualmente se pueden encontrar algunos sistemas de código libre que prestan el servicio de mesa de ayuda, pero no siempre se adaptan del todo a las reglas de negocio que posee la organización. Por lo que el sistema planteado se adecua totalmente a las necesidades de la institución y a sus reglas de negocio tales como: el manejo de etapas en las que se ha dividido el ciclo de vida de un incidente, requerimiento, tarea o nuevo proyecto, las alertas por incumplimiento del tiempo asignado, el registro de un historial, la generación de documentos al finalizar cada etapa, detección de cambios sobre la base de datos (en caso de existir alguno), evaluación por los usuarios internos a la calidad de servicio que brinda el área de

tecnología, reporte de datos estadísticos acerca de la calidad de servicio y reportes en general.

Además el manejo de la herramienta debe ser amigable de acuerdo con las preferencias de los usuarios internos.

Por las características mencionadas anteriormente, la facilidad de ingresar la solicitud para la solución de un incidente, tarea, requerimiento o nuevo proyecto al departamento de tecnología, la facilidad de consultar los estados posibles de dicha solicitud, la optimización del tiempo, y la fácil medición de la calidad de servicio del área de tecnología se puede afirmar que este sistema beneficiará tanto al personal operativo como al personal del área de tecnología puesto que se podrá obtener información rápida y oportuna.

1.5 Alcance

Dentro del procedimiento es importante tomar en cuenta el alcance del sistema informático que se va a construir.

El presente proyecto de tesis se enfoca en el diseño, la construcción e implementación de un sistema que permita una mejor gestión sobre los siguientes temas:

- **Control automatizado del personal autorizado**, al manejo de la información relacionada con las diferentes asignaciones de tareas y proyectos a los diferentes recursos del área de tecnología, haciendo uso de claves de acceso a través de un nombre de usuario y contraseña.
- **Se realizará un seguimiento automatizado** de los diferentes incidentes, tareas, requerimientos, y/o nuevos proyectos de la siguiente manera:
 - El cliente interno registrará los diferentes incidentes, tareas, requerimientos, o nuevos proyectos con el estado de “ingresado”.
 - El usuario responsable asignará la tarea al personal que labora en el área de tecnología con el estado “asignado”.
 - El responsable deberá diariamente registrar los avances de la misma en caso de ser un requerimiento.
 - Seguidamente se realizarán las pruebas correspondientes, las mismas que se registraran en el sistema.

- **Optimizar el tiempo** que se otorga en atender los procesos de solución de incidentes, tareas y/o requerimientos, mediante envío de correos a las personas responsables de gestionarlos.
- **Mediante el registro de estados** de los incidentes, tareas, requerimientos, o nuevos proyectos los usuarios responsables podrán hacer consultas permanentes sobre el estado y avances de los mismos.
- **Respaldo de la información** de todos los incidentes y requerimientos que ingresan al departamento de tecnología usando el almacenamiento de la información correspondiente en una base de datos mediante la interfaz gráfica del sistema.
- **Uso de firmas electrónicas en imagen** para la documentación generada en los diferentes procesos.
- **Uso de reportes** referentes al proceso de solución de requerimientos y estadísticas de los avances de cada tarea.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Fundamentos de ITIL

2.1.1 Historia y concepto de ITIL

ITIL (Biblioteca de Infraestructura de Tecnologías de la Información).

ITIL nació en la década de 1980, a través de la agencia central de telecomunicaciones y computación del gobierno británico – CCTA, quien creó y desarrolló una guía de buenas prácticas para que las oficinas del sector público del gobierno británico fueran más eficientes en su trabajo y por lo tanto redujeran los costos derivados de los procesos y recursos de TI, sin embargo estas guías de buenas prácticas demostraron ser útiles para ser aplicadas en cualquier organización ya que se adaptan a las necesidades y circunstancias. Fue tan útil que en la actualidad ITIL resume la gestión de servicios TI como uno de sus apartados, habiéndose ampliado el conjunto de guías de buenas prácticas referentes a la gestión de la seguridad de la información, gestión de niveles de servicio, perspectiva del negocio, gestión de activos software, gestión de aplicaciones, entre otros.

(B-able, 2013)

Es necesario exponer que actualmente las tecnologías de la información juegan un papel muy importante dentro de una organización debido a la automatización de funciones que disponen sobre el proceso administrativo y de infraestructura, permitiendo el control de gestión, diseño de la organización de sus actividades y mejora de la competitividad.

ITIL permite a una organización obtener ventaja competitiva, innovar, perfeccionar y rediseñar procesos, facilitar procesos administrativos, mejorar la calidad y funcionalidad de sus productos, pero sobretodo ayuda a mejorar el servicio al cliente con una administración eficiente de la tecnología de la información. (Marlon, 2006)

Actualmente ITIL pasó a pertenecer oficialmente a “**Axelos Global Best Practice**” desde el 1 de enero del 2014, ésta empresa le da mayor importancia a la certificación de estas buenas prácticas.

2.1.2 Características de ITIL

- Desarrollada sin derechos de autor.
- De dominio público.
- Compendio de mejores prácticas.
- Estándar Internacional.

2.2 Ciclo de vida de ITIL

2.2.1 Estrategia de servicio (SE)

Tiene como principal objetivo desarrollar una estrategia en la organización en cuanto a las tecnologías de la información.

2.2.2 Diseño del servicio

Es el segundo aspecto del ciclo de vida de ITIL V3 puesto que tiene como misión, diseñar nuevos o modificar los servicios ya existentes para su inclusión en el catálogo y su paso a producción.

2.2.2.1 Gestión del catálogo de servicios

Deriva de la cartera de servicios y es un documento que tiene valor, tanto dentro como fuera de la organización, ya que en este se delimitan los servicios que se pueden ofrecer a los clientes y hasta que nivel pueden ser desarrollados.

Esta gestión implica los siguientes apartados:

- Definición de procesos

Tiene como objetivo definir el catálogo de servicios, discriminando a los que no están activos y agruparlos por familiaridad ya que por lo general están relacionados por el área de funcionalidad. Los datos mínimos que debe contener el catálogo son: nombre y descripción, propietario del servicio, cliente, otras partes implicadas (proveedores, instituciones, etc.), fechas de versión y revisión, etc.

- Mantenimiento y actualización del Catálogo de Servicios

Tiene como objetivo realizar un seguimiento del desempeño, utilización y aprovechamiento de los servicios que se encuentra en el catálogo.

2.2.2.2 Gestión de los niveles de servicio.

El objetivo de la gestión de niveles de servicio es: llegar a un acuerdo con el cliente en un marco de referencia en el que se registren todos los acontecimientos del proyecto, de manera que se pueda llevar a cabo un servicio TI de calidad y a un coste aceptable.

2.2.2.3 Gestión de la disponibilidad

Su objetivo es ofrecer una base para la satisfacción del cliente, es decir todos los servicios deben estar alineados con los SLAs y la infraestructura TI.

2.2.2.4 Gestión de la seguridad de la información

Es la encargada de garantizar que desastres naturales u otras fuerzas externas tengan consecuencias terribles sobre el negocio.

2.2.2.5 Gestión de la capacidad

Tiene como objetivo procurar que el servicio disponga la capacidad de recursos (almacenamiento, rendimiento, eficiencia) necesaria y en el momento que lo requiera.

2.2.2.6 Gestión de la continuidad

Su objetivo es garantizar la infraestructura y los servicios más importantes de la organización, de modo que puedan superar la ocurrencia de un desastre en el menor tiempo posible.

Es importante recalcar, que la Gestión de la Continuidad del Servicio está destinado al fracaso sino se asigna una cantidad de recursos suficientes, tanto en el plano humano como de equipamiento (software y hardware).

2.2.3 Transición de servicio

Se encarga de mejorar las prácticas comunes de la organización de TI, en cuanto a la liberación o puesta en marcha de software y hardware, así como también el proceso que implica dicho cambio. De esta manera ITIL ayuda a que no se realicen gastos innecesarios por pérdidas de tiempo.

2.2.3.1 Gestión del cambio

El objetivo principal de la gestión de cambios es planificar, analizar, y evaluar los cambios que ha de efectuarse asegurando procesos eficaces. Además gestiona el

proceso de cambio que incluye: Hardware, equipo de comunicación y software, documentación y procedimientos asociados a la infraestructura.

Los cambios pueden proceder principalmente de las siguientes fuentes: legislación, problemas de la infraestructura, innovaciones, nuevos mercados/ servicios, equilibrarse o superarse y gestión de proveedores.

2.2.3.1.1 Procesos de la gestión de cambios

Registro

El primer paso del proceso de cambio es registrar adecuadamente las peticiones de cambio (**RFCs**).

El origen de una RFC puede ser de muy distinta índole:

- **Gestión de Problemas:** se encarga de proponer soluciones a errores conocidos. Esto puede acarrear un cambio en la infraestructura TI.
- **Nuevos Servicios:** el desarrollo de nuevos servicios usualmente requiere cambios de la infraestructura TI.
- **Estrategia empresarial:** la dirección puede decidir una redirección estratégica que por regla general requieren de cambios de hardware, software y/o procedimientos.
- **Actualizaciones de software de terceros:** los proveedores pueden dejar de soportar versiones anteriores de paquetes de software o introducir nuevas versiones con grandes mejoras que recomienden la actualización.
- **Imperativo legal:** un cambio de legislación puede exigir cambios en la infraestructura TI.
- **Otro:** en principio cualquier empleado, cliente o proveedor puede sugerir mejoras en los servicios que pueden requerir cambios en la infraestructura.

Aceptación y Clasificación del cambio

Su principal objetivo es evaluar preliminarmente la pertinencia de los registros del RFC, ya que puede ser simplemente rechazada si se considera que el cambio no está justificado o se puede solicitar su modificación, si se considera una mejor definición o aspectos susceptibles.

Clasificación

La categoría dependiendo de la urgencia y el impacto de la misma.

- **Baja:** puede ser conveniente realizar este cambio junto a otros.
- **Normal:** Es conveniente realizar el cambio pero siempre que ello no entorpezca algún otro cambio de más alta prioridad.
- **Alta:** es un cambio que debe realizarse sin demora pues esto provoca una deterioración apreciable del servicio.
- **Urgente:** este implica la solución de problemas que están provocando una interrupción o deterioro grave del servicio.

La determinación de la prioridad de una petición depende del impacto que cause sobre la organización o el deterioro del servicio.

Aprobación y Planificación del cambio

Precautela una eficiente planificación de los cambios ya que un cambio menor puede provocar una reacción en cadena con resultados catastróficos.

Por lo tanto, debe existir una planificación y análisis para seguidamente aprobar los RFCs pendientes y calendarizar los cambios correspondientes, de lo cual está encargado el comité asesor del cambio.

Implementación del cambio

Su objetivo es el de supervisar y coordinar todo el proceso de la implementación de cambios para asegurar que:

- Tanto el software desarrollado como el hardware adquirido se ajustan a las especificaciones predeterminadas.
- Se cumplen los calendarios previstos y la asignación de recursos es la adecuada.
- El entorno de pruebas es realista y simula adecuadamente el entorno de producción.
- Los planes de back-out permitirán la rápida recuperación de la última configuración estable.

Evaluación del cambio

Antes de cerrar el cambio es necesario verificar que ha sido satisfactorio para el servicio. Para ello se debe tomar en cuenta las siguientes preguntas:

- ¿Se cumplieron los objetivos previstos?
- ¿En qué medida se apartó el proceso de las previsiones realizadas por la Gestión de Cambios?
- ¿Provocó el cambio problemas o interrupciones del servicio imprevistas?
- ¿Cuál ha sido la percepción de los usuarios respecto al cambio?

Si los resultados han sido satisfactorios se procederá al cierre de las RFC.

Cambios de emergencia

Este proceso es el encargado de encontrar una respuesta inmediata en casos de interrupciones de alto impacto sobre el sistema.

El objetivo prioritario en estos casos es restaurar el servicio, sin embargo es necesario que en el cierre del cambio se obtenga la misma información que se dispone de un cambio normal.

2.2.3.2 Gestión de entregas y despliegues

Es la encargada de la implementación y control de calidad de todo el software y hardware instalado en el ambiente de producción. Además se encarga de mantener actualizada la biblioteca de medios definitivos, donde se almacenan piezas de repuesto y documentación para la rápida reparación de problemas de hardware.

2.2.3.3 Gestión del conocimiento

La Gestión del Conocimiento se encarga de establecer unos criterios de registro y de acometer labores periódicas de clasificación, evaluación y mejora de los datos disponibles.

Es la encargada de centralizar toda esta información en un repositorio denominado Sistema de Gestión del Conocimiento del Servicio

Las principales actividades de la Gestión del Conocimiento se resumen en:

- Definir una **estrategia** y difundirla a toda la organización TI, usando políticas generales referentes a los datos: qué registrar, cuándo hacerlo, cómo

estructurar los datos, qué clase de información es susceptible de ser corregida o eliminada, quién registra la información, quién la revisa, quién la valida, quienes la pueden consultar libremente.

- Ayudar a la **transferencia** de conocimiento entre personas, equipos y departamentos.
- **Gestionar la información y los datos** para garantizar su calidad y utilidad.

Para documentar y analizar este proceso, además de colaborar con otras gestiones de ITIL se debe considerar:

- **A partir de los errores detectados y las soluciones aportadas** en cada caso desde la Gestión de Incidencias y Errores, puede confeccionarse un registro que recibe el nombre de **Base de Errores Conocidos (KEDB)** y que ayuda a minimizar el tiempo de catalogación y solución de los mismos en el futuro. Lo mismo sucede con la Gestión de Problemas.
- **La Gestión de Cambios aportará documentación** sobre las propuestas de cambio tanto las previamente aprobadas como las que han sido rechazadas. Esta información relativa a las posibles consecuencias de un error, le proporciona al Centro de Servicios la posibilidad de anticiparse al cliente.

2.2.4 Operación del servicio

Esta es la fase más crítica de todas, porque debe permitir a la organización asegurar la prestación de servicios de una manera fácil y eficiente.

Los principales objetivos de la fase de Operación del servicio son:

- Coordinar e implementar todos los procesos, actividades y funciones necesarias para la prestación de los servicios acordados con los niveles de calidad aprobados.
- **Dar soporte a todos los usuarios** del servicio.
- Gestionar la infraestructura tecnológica necesaria para la prestación del servicio.
- Buscar un equilibrio entre estabilidad y capacidad de respuesta.

Los principales procesos asociados a la Fase de Operación del Servicio son:

2.2.4.1 Gestión de Eventos

Es el responsable de monitorizar todos los eventos que acontezcan en la infraestructura TI, con el objetivo de asegurar su correcto funcionamiento y ayudar a prever incidencias futuras.

Se denomina **evento** a todo suceso que se pueda detectar y tenga importancia para la estructura de la organización.

Las actividades de la gestión de eventos son:

- **Aparición de eventos:** el proceso se inicia cuando ocurre el suceso.
- **Notificación de eventos:** el evento es notificado al equipo o responsable de gestión.
- **Detección y filtrado de eventos:** la notificación llega a una herramienta de gestión que lee e interpreta el evento con el fin de determinar si merece mayor atención o no.
- **Clasificación de eventos:** se le asigna una categoría y un nivel de prioridad.
- **Correlación:** se analiza si existen eventos similares, así como la importancia del evento y se decide si es necesario tomar medidas.
- **Disparadores:** se ponen en marcha los mecanismos necesarios para dar respuesta al evento.
- **Opciones de respuesta:** se eligen las soluciones a adoptar.
- **Revisión de acciones y cierre:** se revisan los eventos importantes para determinar si se han tratado correctamente si es el caso se cierra el proceso.

2.2.4.2 Gestión de incidencias

Su responsabilidad es la de registrar todas las incidencias que afecten a la calidad del servicio y restaurarlo a los niveles acordados de calidad en el más breve plazo posible. No se debe confundir la **gestión de incidencias** con la **gestión de problemas** ya que la gestión de problemas se preocupa de encontrar y analizar las causas de un determinado incidente, por el contrario la gestión de incidencias se encarga exclusivamente de la restauración del servicio según como este definido en el documento de **Acuerdos de nivel de servicio (SLA)** correspondiente.

Pro
ceso de la
gestión de
incidentes.

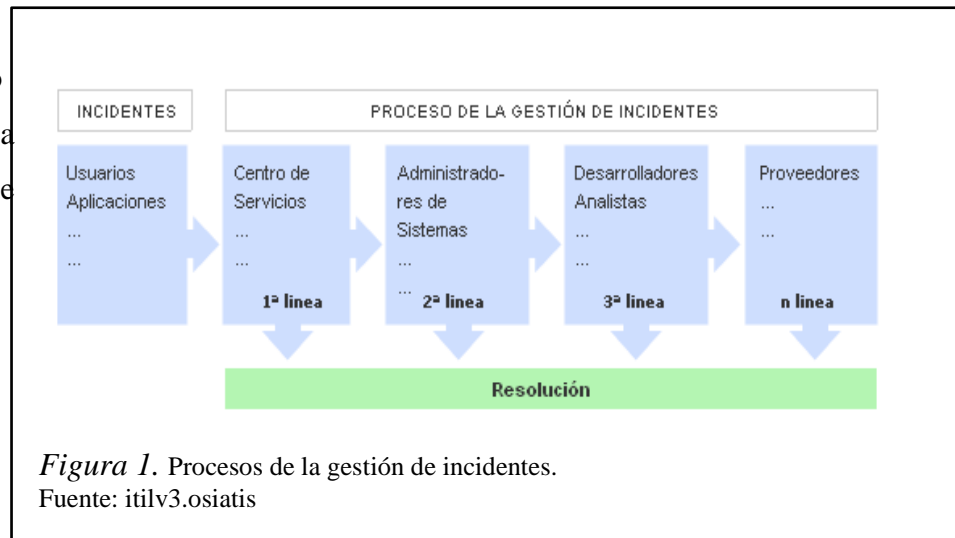


Figura 1. Procesos de la gestión de incidentes.
Fuente: itilv3.osiatis

En una organización frecuentemente existen concurrencias de incidencias por lo cual es importante que estas sean priorizadas.

Priorización

Se basa en dos puntos importantes:

- **Impacto:** determina la importancia de la incidencia dependiendo de los procesos y el número de usuarios afectados.
- **Urgencia:** depende principalmente del nivel de servicio acordado en el SLA.

Además se deben tener en cuenta factores tales como los recursos necesarios, el tiempo de solución esperado, etc.

Dependiendo de la prioridad del incidente los recursos deberán ser asignados.

La prioridad del incidente puede cambiar ya que se pueden dar soluciones momentáneas con el objetivo de restablecer aceptablemente los niveles del servicio

Escalado y soporte

Frecuentemente el centro de servicios no se ve capaz de resolver la incidencia, por lo cual deba recurrir a un especialista o a un superior que pueda tomar decisiones, que estén fuera de su alcance. Este proceso se llama **escalado**.

Tipos de escalado:

- **Funcional:** se requiere de un especialista.
- **Jerárquico:** se debe acudir a una persona con mayor autoridad para tomar decisiones.

2.2.4.2.1 Clasificación y registro

Registro:

El primer paso es registrar el incidente para una correcta gestión.

Las incidencias pueden tener origen de diversas fuentes tales como: técnicas de usuario, aplicaciones, etc. Y es importante que estas sean rápidamente registradas pues se corre el riesgo de que la aparición de nuevas incidencias demore indefinidamente por lo cual el proceso resultaría más costoso.

- **La admisión a trámite del incidente:** el Centro de Servicios debe de ser capaz de evaluar si el servicio requerido se incluye en el SLA, si no es así debe enviarlo a la autoridad correspondiente.
- **Comprobación de que ese incidente** no haya sido registrado para evitar duplicados.
- **Asignación de referencia:** al incidente se le asigna una identificación única.
- **Registro inicial:** se ha de introducir la información básica necesaria para el procesamiento del incidente: fecha, hora, descripción del incidente, etc.
- **Información de apoyo:** se incluirá cualquier información relevante para la solución del incidente.
- **Notificación del incidente:** se debe notificar a los usuarios involucrados sobre la incidencia para evitar afectar en su flujo habitual de trabajo.

Clasificación

Tiene como objetivo mediante la información del incidente, agilizar eficientemente el proceso de solución, este proceso debe cumplir con al menos los siguientes pasos:

- **Categorización:** se asigna una categoría dependiendo del origen, servicio que afecta y tipo de incidencia.
- **Establecimiento del nivel de prioridad:** dependiendo del impacto y la urgencia.
- **Asignación de recursos:** designación del personal indicado.
- **Monitorización del estado y tiempo de respuesta esperado:** seguimiento de la solución del incidente.

2.2.4.2.2 Análisis, resolución y cierre

Como primer paso se examina el incidente con la ayuda de KB (Base del conocimiento) para determinar si se puede identificar con alguna incidencia ya resuelta.

Si el incidente escapa de las posibilidades del centro de servicios, éste se re direcciona a la persona indicada es decir especialista o autoridad superior.

Durante el ciclo de vida del incidente, se debe actualizar la información del mismo, de modo que los involucrados dispongan de información sobre su estado.

Después de haber solucionado el incidente se debe:

- Confirmar a los usuarios.
- Incorporar el proceso de solución al **Sistema de gestión del conocimiento del servicio (SKMS)**.
- Reclasificar el incidente si fuera necesario.
- Actualizar la información en la CMDB (Base de gestión de configuraciones) sobre los elementos de configuración implicados en el incidente.
- Cerrar el incidente.

2.2.4.3 Gestión de peticiones

Gestiona las solicitudes planteadas por los usuarios al departamento de **Tecnologías de la información** TI proporcionándoles información y acceso a los servicios de la organización.

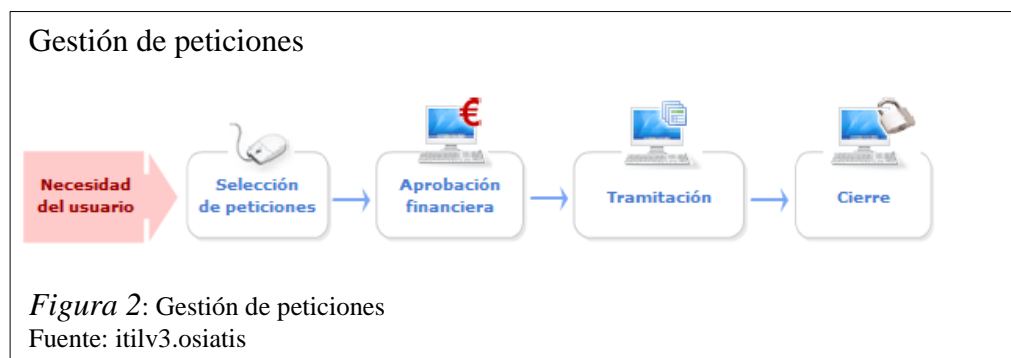
La Gestión de Peticiones recibe la siguiente información para iniciar su trabajo:

- Peticiones de servicio, planteadas por los usuarios.
- Peticiones de cambio (RFCs) por los usuarios.

- Descripción detallada del servicio.
- Políticas de Seguridad.

Las actividades incluidas en el proceso de **gestión de peticiones** son:

- **Selección de peticiones:** permite que los usuarios realicen sus peticiones mediante una interfaz. En ella el cliente podrá escoger entre los tipos de peticiones que se ajusten a su caso.
- **Aprobación financiera de la petición:** Es la encargada de analizar el impacto económico que genera una petición de modo que sea fácil decidir si es o no tramitada.
- **Tramitación:** La petición es resuelta por la o las personas encargadas, para que seguidamente sea notificada dicha solución al centro de servicios y a los usuarios implicados, de modo que estos puedan comprobar dicha solución.



2.2.4.4 Gestión de problemas

Tiene como objetivo investigar todas las causas que generan alguna alteración en el servicio y sus posibles soluciones para restablecer la calidad del servicio y finalmente asegurando que los cambios han surtido los efectos esperados.

La Gestión de Problemas puede ser:

- 1) **Reactiva:** analiza los incidentes para revelar su causa y propone soluciones a los mismos.
- 2) **Proactiva:** monitoriza la calidad de la infraestructura TI para prevenir incidentes.

2.2.4.4.1 Control de problemas

Tiene como objetivo principal conseguir, que los problemas se conviertan en errores conocidos para que el control de errores pueda proponer las soluciones correspondientes.

Fases

Identificación y registro

Tiene como objetivo identificar los problemas mediante varias fuentes de información tales como: Base de datos de incidencias, análisis de infraestructura TI, deterioro de los niveles de servicio.

El registro de problemas es muy similar al registro de incidentes pero prestando mayor atención a su naturaleza y posible impacto.

El registro debe incorporar al menos la siguiente información:

- Los **Elementos de configuración** (CIs) implicados.
- Causas del problema.
- Síntomas asociados.
- Soluciones temporales.
- Servicios involucrados.
- Niveles de prioridad, urgencia e impacto.
- Estado: activo, error conocido, cerrado.

Clasificación y asignación de recursos

Engloba las características del problema así como también las áreas funcionales que se ven afectadas. Otro factor determinante es la prioridad del problema, que al igual que en el caso de los incidentes, se determina a partir de la urgencia.

Una vez clasificado el problema y determinada su prioridad, se deben asignar los recursos necesarios para su solución.

Análisis y diagnóstico

Determinar las causas del problema proporcionando soluciones temporales hasta implementar los cambios que resuelvan definitivamente el problema.

Una vez determinadas las causas del problema, éste se convierte en un Error Conocido y se remite al Control de Errores para su posterior procesamiento.

2.2.4.4.2 Control de errores

Una vez que se ha determinado las causas de un problema, es responsabilidad del Control de errores el registro del mismo para que este se convierta en un error conocido.

2.2.5 Mejora continua

Tiene como principal objetivo ofrecer mejores servicios, adaptados a las cambiantes necesidades de los usuarios, mediante procesos internos optimizados y la continua monitorización y medición de todas las actividades y procesos involucrados en la prestación de servicios TI.

Las actividades que realiza:

- **Recomendar mejoras** para todos los procesos involucrados en la prestación de servicios TI.
- **Monitorizar los parámetros** de seguimiento de niveles de calidad contrastándolos con los **Acuerdos de nivel de servicio (SLA)**.
- **Proponer mejoras** que aumenten el Retorno de la inversión (ROI) y Valor de la inversión (VOI) asociados a los servicios.
- **Dar soporte** para la definición de nuevos servicios.

2.3 Definición de help desk o mesa de ayuda

Es un conjunto de recursos tecnológicos y humanos que prestan ayuda a la gestión y solución de todas las posibles incidencias relacionadas a las tecnologías de la información y computación.

El personal o recurso humano encargado de mesa de ayuda debe poseer conocimientos de software, hardware y telecomunicaciones para estar en la capacidad de proporcionar respuestas, soluciones y otorgar asesoramiento a los usuarios finales.

Las instituciones generalmente proporcionan soporte de mesa de ayuda, usando varios canales como números telefónicos, mensajería instantánea, correo electrónico, sitios web y sistemas informáticos.

El servicio de mesa de ayuda debe proveer a los usuarios de un solo recurso a donde van a ser remitidas todas sus dudas, inquietudes y necesidades relacionadas con las tecnologías de la información y la comunicación. (slideshare.net, 2015)

2.4 Metodología XP (Programación Extrema)

Forma parte de las metodologías ágiles para el desarrollo de software que se basan en la adaptabilidad de cualquier cambio con el propósito de aumentar las posibilidades de éxito en un proyecto.

En el año de 1999 la programación extrema nace de la mano de **Kent Beck** como un proceso poco convencional en el cual enfatiza algunos puntos: la comunicación dentro del equipo, la facilidad y sencillez de la implementación, la implicación del usuario final dentro del proceso y toma de decisiones, y la rapidez y efectividad.

XP es “una forma de desarrollar software: liviana, eficiente, de bajo riesgo, flexible, predecible, científica y divertida” (Beck K. , 1999)

Esta metodología de programación da como prioridad a los trabajos que dan un resultado directo y que reducen el papeleo que existe alrededor de la programación.

Es una metodología adecuada para proyectos medianos y pequeños en los cuales los requerimientos pueden cambiar durante la etapa de desarrollo.

2.4.1 Fundamentos

XP se basa en cuatro fundamentos los cuales son:

- **Comunicación** es una palabra clave en este modelo ya que el equipo debe estar continuamente en conversaciones de los avances y los cambios requeridos por los usuarios finales.
- **Simplicidad** por la manera de desarrollar el software evitando la burocracia en el proceso.

- **Retroalimentación** inclina al programador a obtener el estado del software constantemente a través de las pruebas unitarias y la integración continua así como también el usuario final está informado del estado del proyecto a través de los test de aceptación.
- **Valentía** XP requiere de este valor en los programadores ya que deben estar dispuestos a realizar cambios sobre las líneas de código sin importar la cantidad de código que se tenga con el fin de realizar un trabajo de calidad y conforme al usuario final.(Beck K. A., 2004)

2.4.2 Principios básicos

1. Retroalimentación a escala fina

- **Principio de pruebas:** debe establecer un periodo de pruebas de aceptación del programa donde se determinan las entradas del sistema y resultados esperados.
- **Procesos de planificación:** en esta fase el usuario tendrá que escribir sus necesidades determinadas en historias de usuario dependiendo de la complejidad del problema.
- **El cliente en el sitio:** se le dará poder para definir los requerimientos, señalar las prioridades y responder las preguntas de los programadores.
- **Programación en parejas:** programación en parejas en una sola máquina con el objetivo de producir mejores aplicaciones, aunque este principio no puede ser para todo el mundo. Además según Martin Fowler en 2006 escribió un artículo en el que explica que no es necesario programar en parejas para estar practicando la agilidad. (Fowler, 2006)

2. Proceso continuo.

- **Integración continua:** el equipo hace un rápido progreso implementando las nuevas características del software.
- **Refactorización:** permite mejorar el diseño del sistema a través de todo el desarrollo mediante el diseño y recodificación.
- **Entregas pequeñas:** rápidamente colocan un sistema sencillo en producción que se actualiza de forma rápida y constante.

3. Entendimiento compartido

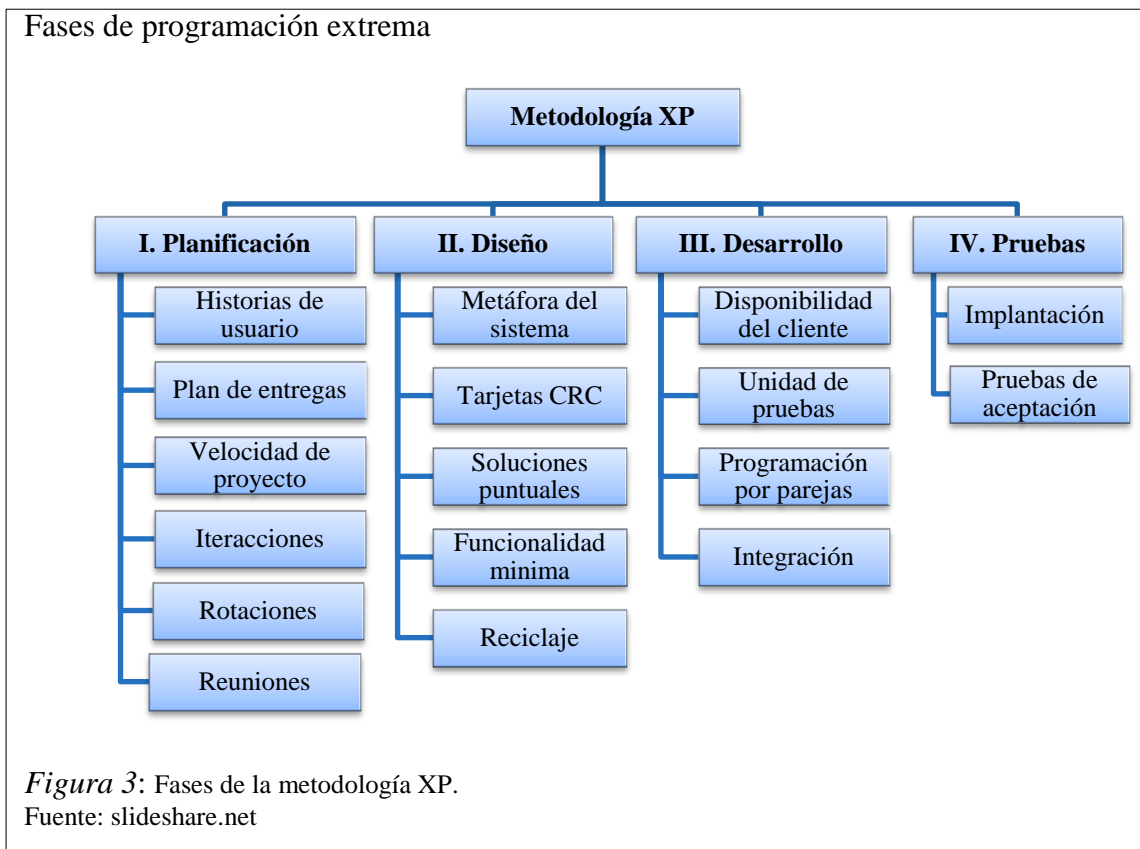
- **Diseño simple:** se enfoca en entregar un sistema que cumpla las necesidades inmediatas del cliente.
- **Propiedad colectiva del código:** mientras haya más gente trabajando en una pieza menos errores aparecerán, todos son los propietarios de todo.
- **Estándar de codificación:** define la propiedad del código compartido así como las reglas para escribirlo y documentarlo.

4. Bienestar del programador

- **La semana de 40 horas:** minimizar las horas extras y mantener a los programadores frescos genera código de mayor calidad.

(Beck K. A., 2004)

2.4.3 Fases programación extrema XP



2.4.3.1 Fase de planificación del proyecto

En esta fase el cliente plantea en forma general los requerimientos del proyecto detallados en las historias de usuario, a su vez los desarrolladores se familiarizan con las herramientas, reglas del negocio que van a usar y hacen una estimación de

esfuerzos llegando de esta manera a acuerdos sobre el contenido de la primera entrega.

- a) **Historias de usuario:** su propósito es análogo al de casos de uso. En estas se describen las especificaciones de requisitos donde los desarrolladores y clientes establecen tiempos de implementación, y prioridades.
- b) **Rotaciones:** evita que las personas compriman el flujo normal del proceso permitiendo que todos conozcan como funcional el sistema.
- c) **Reuniones:** seguimiento diario.
- d) **Correcciones:** se debe corregir las diferentes funcionalidades cuando estas fallen y plasmar los cambios en las iteraciones de las historias de usuario.

2.4.3.1.1 Historias de usuario vs casos de uso

Tabla 1. *Comparativa entre casos de uso e historias de usuario*

Concepto	Casos de uso	Historias de usuario
Objetivo	Modelar la interacción entre un actor y el sistema.	Redactar una descripción breve de una funcionalidad tal y como la percibe el cliente.
Estructura	Texto detallado donde se sugiere una plantilla predefinida a completar con conceptos técnicos.	Corta y consistente en una o dos frases escritas en el lenguaje del usuario.
Agilidad	Requieren tiempo para el análisis y la redacción de plantillas predefinidas	Se pueden escribir en poco tiempo.
Comunicación	Modelo textual asociado a diagramas: todo tiene que estar escrito	Basada en la comunicación verbal y orientada a la colaboración y discusión para clarificar detalles.
Compresión	Suelen ser de difícil comprensión, incluso para personal técnico	Fácil de leer y comprender

Nota. Comparativa de casos de uso vs historias de usuarios. Fuente: (**agile-ux.com, 2014**)
Elaborado por: Ximena Catota

2.4.3.2 Fase de diseño

Está directamente relacionada con la fase de planificación en la que se realizó el análisis de las historias de usuario con el fin de diseñar el software.

- a) **Metáfora del sistema:** es el uso de insumos como glosarios, una correcta especificación de los nombres de métodos y clases, esto estimula las historias

de usuario en lugar de los diagramas de lenguaje unificado del modelado (UML) que expresa la visión evolutiva del proyecto.

- b) **Tarjetas Clase-responsabilidad-colaborador (CRC):** representan los objetos que contiene una clase y con las que esta interactúa.
- c) **Funcionalidad mínima:** no se debe añadir una funcionalidad extra al programa a pesar de que este pensada a un futuro para evitar un desperdicio de tiempo y recursos.

2.4.3.3 Fase de desarrollo

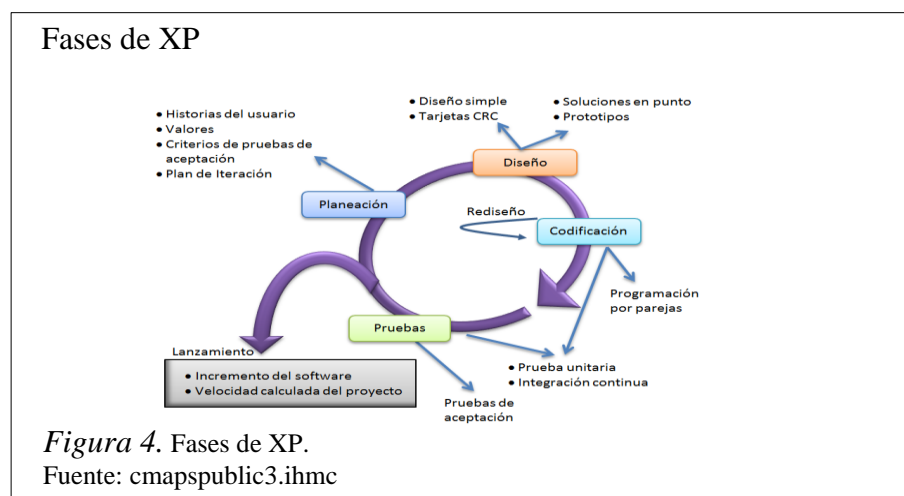
Antes del desarrollo de cada historia de usuario el cliente debe especificar lo más detalladamente posible lo que ésta hará y también tendrá que estar presente cuando se realicen las pruebas que verifiquen que la historia implementada cumple la funcionalidad especificada.

- Frecuencia en la integración del código.
- Optimizaciones al final.

2.4.3.4 Fase de pruebas

Uno de los pilares fundamentales de XP es el uso de pruebas para comprobar el funcionamiento del código que se va implementando en la construcción del proyecto. En esta fase se realizan pruebas de aceptación de cada una de las historias de usuario.

- a) **Implementación:** implementación de las historias de usuario.
- b) **Pruebas de aceptación:** evaluación del cliente de cada implementación que se hace en el software.



2.4.4 Ventajas y desventajas

Tabla 2. *Ventajas y desventajas de XP*

Ventajas	Desventajas
Programación organizada	No es recomendable para proyectos grandes
Proceso flexible.	Es necesario la presencia del cliente todo el tiempo
Menor porcentaje de errores	Todo el proceso se basa en la comunicación
El cliente está al tanto del desarrollo.	No tiene costo o tiempo definido

Nota. Ventajas y desventajas de la programación extrema. Fuente: slideshare.net
Elaborado por: Ximena Catota

2.4.5 Diseño de software

El diseño de software es una tarea muy importante porque es una representación significativa del sistema que se va a construir. Esta tarea produce un diseño de datos, un diseño arquitectónico, un diseño de interfaz y un diseño de componentes los cuales juegan un papel muy importante para conseguir el éxito del proyecto ya que un planteamiento correcto de estos puntos permitira obtener resultados optimos y satisfactorios para el cliente. (S.Pressman, 2002)

Diseño de datos

El diseño de datos transforma el modelo de dominio de información que se crea durante el analisis en las estructuras de datos que se necesitaran para implementar el software. Los objetos de datos y las relaciones definidas en el diagrama de relación entidad proporcionan la base pde un buen diseño de datos. (S.Pressman, 2002)

Diseño arquitectónico

Define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos. (Shaw & Garlan , 1996)

Diseño de interfaces

Describe la manera de comunicarse el software dentro de si mismo, con sistemas que interoperan dentro de el y con las personas que lo utilizan. Una interfaz implica un

flujo de información y un tipo de comportamiento. Por lo tanto los diagramas de flujo

y de datos proporcionan gran parte de la información que se requiere para el diseño de la interfaz. (S.Pressman, 2002)

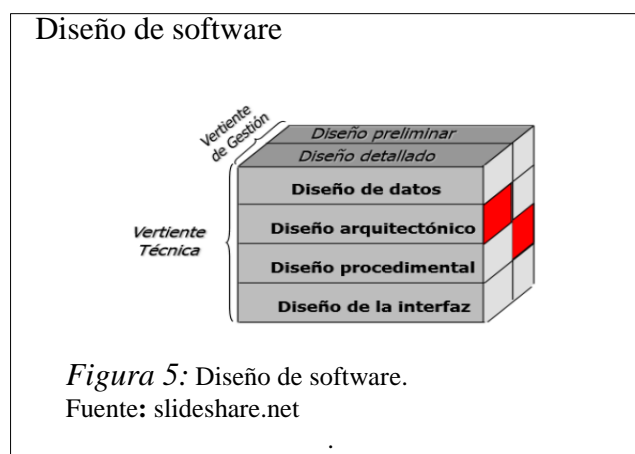


Figura 5: Diseño de software.

Fuente: slideshare.net

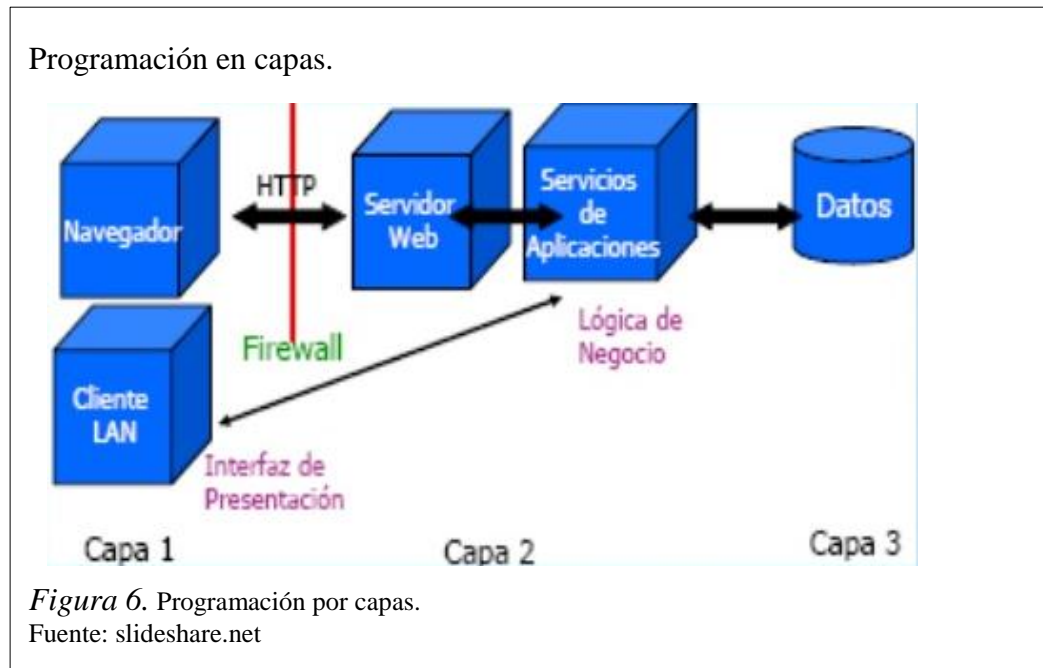
2.5 Tecnología a emplearse

2.5.1 Programación en capas

Es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios, datos y diseño.

- **Capa de presentación.** “Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso.” (Calle, 2013).
- **Capa de negocio:** “Aquí es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio e incluso de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse.” (Calle, 2013).

- **Capa de datos:** “Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.” (Calle, 2013).



2.5.2 Windows Communication Foundation (WCF)

Es un marco de trabajo para la creación de aplicaciones orientadas a servicios.

WCF cuenta con la posibilidad de enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Estos mensajes pueden ser tan simples o tan complejos.

Un extremo de servicio puede formar parte de un servicio, ser un servicio hospedado en una aplicación, o puede ser un cliente de un servicio que solicita datos de un extremo de servicio. (msdn.microsoft.com, 2015)

2.5.2.1 Características de WCF

- **Orientada a servicios:** permite la creación de aplicaciones orientadas a servicios.
- **Interoperabilidad:** WCF implementa los estándares del sector modernos para la interoperabilidad de servicios web.

- **Metadatos de servicios:** admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares.
- **Seguridad:** cifra los mensajes para proteger la privacidad, y obligar a los usuarios a que se autenticuen antes de permitirles recibir mensajes.
- **Varios transportes y codificaciones:** los mensajes pueden enviarse con cualquiera de los protocolos y codificaciones integrados.
(msdn.microsoft.com, 2015)

2.5.3 Language Integrated Query (LINQ)

Como su nombre lo indica es un lenguaje de consultas integrado que expone operadores de consulta sin importar cual fuese la fuente de datos y con la consistencia de utilizar el mismo patrón para todas las consultas.

Es una tecnología integrada en .NET embebido en el framework que usa la sintaxis en forma nativa de cualquier lenguaje de programación soportando por .NET. Lo cual proporciona el soporte del compilador y permite concentrarse únicamente en las búsquedas en lugar de hacer la rutina para cada búsqueda

LINQ extiende el lenguaje a través de sentencias parecidas a SQL y pueden ser usadas para extraer y procesar convenientemente los datos. (Pialorsi & Russo, 2009)

2.5.3.1 Ventajas de LINQ

- Proporciona una solución para el problema de mapeo objeto-relacional así como también para simplificar la interacción entre los objetos y fuentes de datos.
- LINQ integra consultas directamente dentro de .NET, como C # y Visual Basic a través de un conjunto de extensiones a estos idiomas.
- Disminuye el tiempo de compilación para las consultas así como buenos consejos de característica IntelliSense de Visual Studio. LINQ cambiará significativamente algunos aspectos con respecto a la forma de manejar y manipular datos con sus aplicaciones y componentes.

2.5.3.2 Componentes de LINQ

Dependiendo de la fuente de datos que se va a usar se escoge el componente de LINQ a tales componentes se agrupan en:

LINQ to SQL

Es una implementación de un conjunto de clases, estructuras, interfaces y enumeraciones utilizadas para escribir consultas a bases de datos relacionales como PostgreSQL, SQL Server o MySQL.

LINQ to SQL permite modelar la capa de datos de las aplicaciones de una manera más simple y limpia.

Ejemplo:

Consultando Productos de la base de datos

El siguiente código usa una consulta LINQ para obtener una secuencia IEnumerable de objetos productos de la categoría " Cuadernos".

```
DataContext db = new DataContext();  
var productos= from p in db.Productos  
               where p.categoria.categoriaNombre == "Cuadernos"  
               select p;
```

Una vez que se tenga el modelo de datos a través del server Explorer se hace la conexión a la base de datos y se arrastra las tablas al diagrama DataContext.

“DataContext: es una clase que nos permite a través de ella realizar nuestras consultas a las entidades de nuestra base de datos” (blogams.com, 2010)

LINQ to Objects: Es la API predeterminada de LINQ que permite escribir consultas para arreglos, estructuras y colecciones de objetos en memoria.

LINQ to XML: Es la API de LINQ que proporciona la habilidad de escribir consultas para procesar fuentes de datos XML.

LINQ to Data Set: Es la API dedicada a trabajar con clases Data set y DataTables, ya que aún existen aplicaciones y desarrolladores que utilizan esta solución.

2.5.4 Lenguaje de marcado XAML

Su estructura está basada en el **lenguaje de marcas extensible** (XML). Es utilizado para crear la **Interfaz Gráfica de Usuario** (GUI), con herramientas comunes como ventanas, cuadros de diálogo, controles de usuario, etc.

Etiquetas descriptivas dentro de caracteres < >. Cada elemento es iniciado con un nombre específico y siempre se debe señalar su cierre usando la etiqueta </>. Por ejemplo:

```
<StackPanel>  
  <Botón contenido = "Click Me" />  
</ StackPanel>
```

XAML forma parte de Microsoft Windows Presentation Foundation que son características de Microsoft .NET Framework 3.5 relacionadas con la presentación visual de aplicaciones basadas en Windows y de aplicaciones cliente basadas en exploradores web. (msdn.microsoft, 2015).

2.5.5 Silverlight

Es framework para el desarrollo de aplicaciones web impulsadas por el framework .NET compatibles con varios navegadores, dispositivos y sistemas operativos ofreciendo un nuevo nivel de interactividad. El entorno de tiempo de ejecución de Silverlight está disponible como un plug-in para navegadores web que se ejecutan en Microsoft Windows y Mac OS X. (msdn.microsoft, Silverlight, 2015)

Silverlight ofrece un modelo programación flexible compatible con lenguajes .NET y además se integra con las aplicaciones web existentes.

Su primera versión fue lanzada en septiembre del 2007 y actualmente se distribuye de forma gratuita, también cuenta con Moonlight que es una versión de código abierto para los sistemas operativos como Linux. En el 2008 se incluye el **Common Language Runtime** (CLR) y el subconjunto de .NET y finalmente en el 2010 fortalece a **Rich Internet Applications** (RIA). (slideshare.net, Introduccion de Silverlight, 2015)

La base de programación de Silverlight es el **lenguaje extensible de formato para aplicaciones** (XAML) para crear las interfaces de usuario y se puede emplear una amplia variedad de idiomas para la funcionalidad de back-end. Y el acceso a los objetos está dado por C#.

2.5.5.1 Características de Silverlight

- Las aplicaciones Silverlight se pueden escribir en cualquier lenguaje de programación. Cualquier herramienta de desarrollo que se pueda utilizar con .NET puede trabajar con Silverlight. Además de que Microsoft ha proporcionado a Microsoft expression blend una herramienta complementaria a visual studio para el diseño de aplicaciones de Silverlight.
- Silverlight agrega funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones, etc. en forma similar a lo que hace Adobe Flash. Compite con adobe flex, JavaFX y algún as prestaciones de AJAX.
- Cuenta con un excelente rendimiento cuando se ejecuta, ya que utiliza pocos recursos.
- Genera una mejor experiencia de usuario por los controles y componentes versátiles, la portabilidad entre plataformas y la rapidez de la actualización de aplicaciones a miles de usuarios al mismo tiempo. (Vidales & Falagán, 2012)

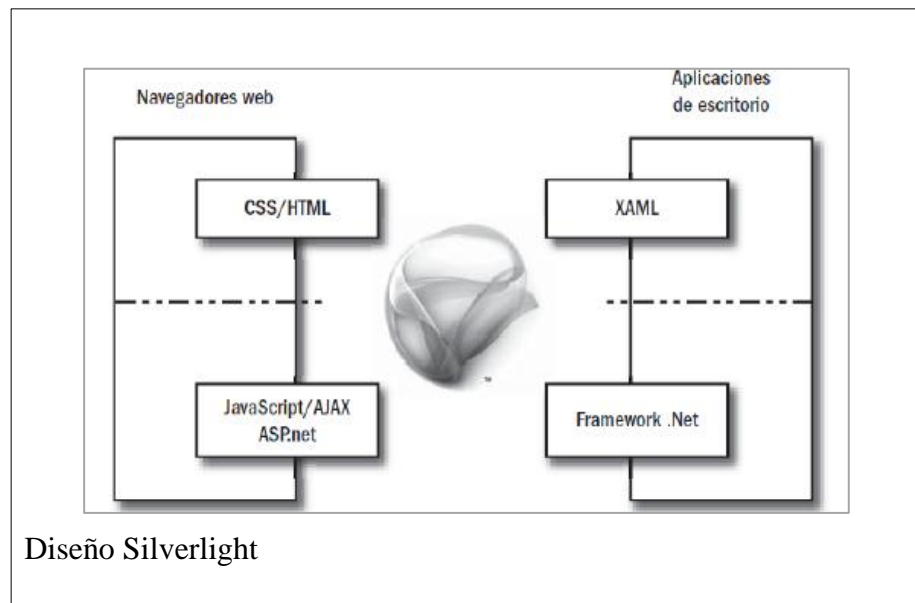


Figura 7. Diseño de Silverlight.
Fuente: www.silverlight.net

2.5.6 Expression blend

Es una herramienta que viene incluida en Silverlight tools, es útil para diseñadores y programadores ya que permite crear gráficos, diseñar animaciones y generar experiencias de usuario.

Expression blend utiliza XAML y el mismo sistema de proyectos que visual studio lo que permite que programadores y diseñadores compartan los mismos archivos.”

2.6 Especificación de requerimientos

Como se lo había documentado en la sección de marco teórico la metodología XP usa historias de usuario para especificar recolectar los requerimientos del sistema.

Las historias de usuario que se recopilieron para llevar a cabo el desarrollo del sistema se detallan a continuación:

2.6.1 Historias de usuario

Historia de Usuario			
Numero:	RF01	Nombre:	Pantalla de autenticación de usuario.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	3	Iteración asignada:	2
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción:	<ul style="list-style-type: none">•Los usuarios deberán identificarse para acceder a cualquier parte del sistema.•El sistema desplegará funcionalidades por usuario de acuerdo a los roles asignados los mismos que garantizan seguridad sobre la información.•Esta pantalla debe tener la opción "olvido su contraseña" para que mediante otra pantalla sea recuperada por envío de correo electrónico.•Debe poseer las validaciones respectivas de usuario y contraseña, tales como: usuario activo, usuario existente, contraseña válida.		
Observaciones:			

Historia de Usuario			
Numero:	RF02	Nombre:	Carga de usuarios.
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción:			
<ul style="list-style-type: none"> Se debe realizar una carga manual de los usuarios existentes del banco desde un listado obtenido del Core Financiamiento debido a que estos datos no cambian frecuentemente. Pero se requiere de una pantalla para el mantenimiento de los usuarios (edición y creación de nuevos usuarios). 			
Observaciones:			
*Como desarrollador se propuso la creación de un web services para la migración automática de los usuarios del banco existentes en el Core Financiamiento, no obstante la institución prefiere carga manual ya que esta data no cambia con frecuencia.			

Historia de Usuario			
Numero:	RF03	Nombre:	Pantalla de mantenimiento de un nuevo usuario.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción:			
Se requiere una pantalla para el registro y edición de un nuevo usuario en el sistema, para ello se debe crear en primera instancia a la persona con los siguientes datos: Identificación, nombres, teléfono, extensión, email, y cargo en la institución.			
<ul style="list-style-type: none"> Campos de texto: identificación, nombres, teléfono, extensión, email. Combo box: con el listado de los cargos existentes en la institución (catálogo). Campos obligatorios: identificación, nombres, extensión, email. Datagrid: listado de usuarios con la información de cada persona. 			
Se sugiere añadir en la misma pantalla la creación del Usuario para la persona con los siguientes campos:			
<ul style="list-style-type: none"> Campos de texto: código usuario. (el código de usuario debe ser el mismo que se usa en el Core) Combo box: agencia, rol, departamento, área. (Catálogos). 			
Botón guardar: almacena información; Nuevo: ingreso de nuevos datos; Editar: modificar información.			
Estos campos a excepción de la identificación y el código de usuario deben ser modificables.			
Observaciones:			

Pantalla y funcionalidad aprobada

Historia de Usuario			
Numero:	RF04	Nombre:	Cambiar contraseña
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	3	Iteración asignada:	2
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción:	<p>*Se requiere una pantalla para el cambio de contraseñas de los usuarios. Esta pantalla debe tener los siguientes campos.</p> <ul style="list-style-type: none">• Campos de password: contraseña anterior y contraseña actual. De manera que valide esta información.• Validar la información: Indica si la contraseña es válida• Botón guardar: graba la información, y emite un mensaje de cambio exitoso.		
Observaciones:	Funcionalidades y pantalla aprobada.		
Historia de Usuario			
Numero:	RF05	Nombre:	Olvido su contraseña.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción:	<p>Se requiere una pantalla para la restauración de contraseñas en caso de que el usuario la haya olvidado. Esta pantalla debe tener la siguiente funcionalidad:</p> <ul style="list-style-type: none">• Campo de ingreso: código de usuario.• Botón de enviar: después de ingresar el usuario al presionar este botón debe enviar al correo del usuario una contraseña con la que pueda acceder al sistema.• Para acceder a esta opción debe existir un link en la pantalla de login del sistema con la leyenda: olvido su contraseña?• Validaciones de existencia de usuario y campo obligatorio.		
Observaciones:	Pantalla y funcionalidad aprobada		

Historia de Usuario			
Numero:	RF06	Nombre:	Pantalla de administración de roles.
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1

Prioridad: (alta/media/baja)	baja
Desarrollador encargado:	Ximena Catota
Descripción: Se requiere una pantalla para la administración de roles: creación, edición, y asignación de accesos al sistema. Esta pantalla debe tener la siguientes características: Para la creación de un nuevo rol: <ul style="list-style-type: none"> • Campos de texto: código de usuario (solo de lectura), nombre del rol. • Check: de activación de usuario para la creación este debe estar en activo. • Botón de permisos de administración: al presionar este botón debe desplegar el listado del menú disponible en el sistema en el lado izquierdo (menú no asignado). • Listado derecho menú asignado. Botones para agregar de un lado al otro los menús dependiendo del acceso que se le quiera otorgar a cada rol.	
Observaciones: Funcionalidades y pantalla aprobada.	

Historia de Usuario			
Numero:	RF07	Nombre:	Tipos de solicitud
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción: Los procesos de una solicitud dependen de su problemática por lo tanto el sistema deberá contar con dos diferentes tipos de solicitud tales como: - Incidente: solicitud que debe ser atendida y resuelta de inmediato ya que afecta al servicio. - Requerimiento: es una funcionalidad nueva o modificación de funcionalidades anteriores en cualquier aplicativo existente en la institución. *En el sistema se debe manejar las áreas existentes en el departamento de TI ya que las solicitudes se han de dirigir según al área de trabajo a la que corresponda. -Para una fácil selección del usuario se solicita el uso de combo box tanto como para el área, aplicativos del área y módulos de los aplicativos. *Se solicita tener un campo en el que se pueda tener una breve descripción de la solicitud (incidente, requerimiento).			
Observaciones: Funcionalidades y cabecera de la pantalla del ingreso de solicitud aprobada			

Historia de Usuario			
Numero:	RF08	Nombre:	Detalle de la solicitud -

			requerimiento
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	3	Iteración asignada:	2
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:			
Para el detalle del tipo de solicitud requerimiento se solicita los siguientes campos:			
* Descripción del requerimiento o caso en cuyo campo se puedan agregar varios puntos mediante una pantalla alterna.			
* Anexos: campo en el que se pueda describir los documentos que avale la solicitud.			
* Comentarios: campo en el que se puede hacer un corto comentario si el usuario así lo requiere.			
* Sugerencias: campo en el que se puede hacer una breve sugerencia si así el usuario lo desea.			
* Firmas para la aprobación de la solicitud: estas firmas serán cargadas en combo box de acuerdo a los roles de jefaturas, las firmas necesarias son: jefe responsable, jefe de tecnología y jefe de operaciones.			
Observaciones:			
*Pantalla aprobada			
*Se sugirió establecer por defecto las firmas de aprobación pero el usuario necesita escoger las firmas por los suplentes en caso de vacaciones o de tener uno o más suplentes.			

Historia de Usuario			
Numero:	RF09	Nombre:	Detalle solicitud por afectación a BDD
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	4	Iteración asignada:	3
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:			
Para el detalle del tipo de solicitud requerimiento se solicita los siguientes campos:			
* Descripción del requerimiento o caso en cuyo campo se puedan agregar varios puntos mediante una pantalla alterna.			
* Anexos: campo en el que se pueda describir los documentos que avale la afectación			
* Justificativos: campo para describir la justificación por la cual se requiere afectar a la base de datos			
* Firmas para la aprobación de la solicitud: estas firmas serán cargadas en combo box, de las siguientes jefaturas: Auditor, Jefe de tecnología, Jefe de operaciones y Gerente general			
Observaciones:			
Se sugirió establecer por defecto las firmas de aprobación pero el usuario necesita escoger las firmas por los suplentes en caso de vacaciones o de tener uno o más suplentes.			

Historia de Usuario			
Numero:	RF10	Nombre:	Aprobación solicitud requerimiento

Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:			
<p>La solicitud deberá tener la aprobación después de que todos los usuarios hayan firmado mediante la activación de un Chek presionado firmar.</p> <p>Esta pantalla deberá contar con:</p> <ul style="list-style-type: none"> *Un combo box con la información de tipo de requerimiento. *Un combo box que contiene los estados de cada tipo de requerimiento 			
Observaciones:			
Añadir un botón para negar la solicitud que envíe al usuario solicitante un correo con la razón y el usuario que negó la solicitud.			

Historia de Usuario			
Numero:	RF11	Nombre:	Gestión de incidente - ingreso
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:			
<p>Mediante la pantalla de ingreso de solicitud al escoger la opción incidente se deberá solicitar al usuario el ingreso de los siguientes campos:</p> <ul style="list-style-type: none"> • Descripción breve: campo obligatorio. • Detalle: campo obligatorio. • Envío de correo electrónico para notificar el ingreso del incidente al jefe de área correspondiente para su pronta asignación. • Validaciones de campos obligatorios. 			
Observaciones:			
Campo faltante descripción.			

Historia de Usuario			
Numero:	RF12	Nombre:	Gestión de incidente - asignación
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	3	Iteración asignada:	2

Prioridad: (alta/media/baja)	alta
Desarrollador encargado:	Ximena Catota
Descripción: Se requiere una pantalla para la asignación de incidentes que contenga lo siguiente: <ul style="list-style-type: none"> • Combo box: que contenga el listado de códigos de incidentes pendientes de asignar de manera que al seleccionar alguno se despliegue información general del incidente. • Campos de lectura con la siguiente información: Aplicativo, modulo, descripción breve del incidente, fecha ingreso, tipo requerimiento, usuario reporta. • En esta pantalla el usuario que asignara el incidente debe establecer mediante combo box el nivel de impacto y prioridad al incidente. • Para la asignación del recurso se requiere un combo box con el listado de los recursos humanos pertenecientes al área. • Un Botón guardar que al presionar almacene los datos y envíe un correo electrónico al usuario asignado para la solución del incidente. • Un botón devolver que al presionar devuelva el incidente al usuario solicitante con una razón de la devolución y así como también el envío del correo comunicando esta devolución. 	
Observaciones: Añadir campo de comentario para la devolución del incidente.	

Historia de Usuario			
Numero:	RF13	Nombre:	Gestión de incidente – cierre - devolución
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requiere una pantalla para el cierre de incidentes que contenga lo siguiente: <ul style="list-style-type: none"> • Combo box que contenga el listado de códigos de incidentes pendientes de cerrar por el usuario asignado de manera que al seleccionar alguno se despliegue información general del incidente. • Campos de lectura con la siguiente información: Aplicativo, modulo, descripción breve del incidente, fecha ingreso, tipo requerimiento, usuario reporta, nivel de impacto, prioridad. • Campos de ingreso por el usuario (recurso asignado): <ul style="list-style-type: none"> • Solución: campo obligatorio que describe la solución al incidente. • Observaciones: campo opcional no obligatorio. • Combo box listado de opciones (cerrar, devolver) si se devuelve termina el ciclo. • Un botón buscar solución que al presionar busque en la base de conocimientos soluciones posibles del caso y en caso de hallar coincidencias listarlas en una pantalla alterna que permita usar la solución. • Un botón guardar que al presionar almacene la información y envíe un correo al usuario solicitante para que realice la evaluación correspondiente. 			
Observaciones: Pantalla y funcionalidad aprobada.			

Historia de Usuario			
Numero:	RF14	Nombre:	Gestión de incidente - calificación
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota.		
Descripción:			
<p>Se requiere una pantalla para la asignación de incidentes que contenga lo siguiente:</p> <ul style="list-style-type: none"> • Combo box que contenga el listado de códigos de incidentes pendientes de calificar por el usuario. • Campos de lectura con la siguiente información: Aplicativo, modulo, descripción breve del incidente, fecha ingreso, tipo requerimiento, usuario reporta. • Sección de encuesta con las siguientes preguntas cerradas: <ul style="list-style-type: none"> -Se cumplieron con los objetivos? Si, no. -Selección de calificación: buena, muy buena, excelente, pésima, regular (catálogo). • Validaciones de respuestas coherentes no puede tener calificaciones buenas y no haber cumplido con el o los objetivos. • Un Botón guardar que al presionar almacene los datos y envíe un correo electrónico al usuario asignado la calificación en caso de ser negativa debe enviar el correo con copia al gerente de TI y al jefe de área. Si es calificación negativa el incidente debe ser reabierto y mediante un correo se envía la información al usuario solicitante. 			
Observaciones:			
Pantalla y funcionalidad aprobada basada en las sugerencias de ITIL. (fuera del alcance)			

Historia de Usuario			
Numero:	R15	Nombre:	Gestión de incidente - reasignación
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	4	Iteración asignada:	3
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:			
<p>Se requiere una pantalla para la reasignación de incidentes que contenga lo siguiente:</p> <ul style="list-style-type: none"> • Combo box que contenga el listado de códigos de incidentes asignados pendientes de manera que al seleccionar alguno se despliegue información general del incidente. • Campos de lectura con la siguiente información: Aplicativo, modulo, descripción breve del incidente, fecha ingreso, tipo requerimiento, usuario reporta, nivel de impacto, prioridad, recurso actual asignado, Detalle del incidente. • Campos de ingreso: <ul style="list-style-type: none"> Comentario: campo obligatorio que describe un comentario del porque es reasignado. • Combo box listado de recursos: técnicos del departamento de tecnología. • Un botón reasignar que al presionar almacene la información y envíe un correo al usuario solicitante sobre el cambio de recurso asignado. <p>Esta opción debe estar habilitada para todos los recursos del departamento de TI ya que</p>			

cualquier recurso debe ser capaz de reasignar a otro recurso.

Observaciones:

Cambio sobre detalle de incidente realizado fuera del alcance.

Historia de Usuario			
Numero:	RF16	Nombre:	Gestión del conocimiento – alimentación automática de la base del conocimiento.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	3	Iteración asignada:	2
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:	Se requiere el almacenamiento automático de las soluciones de cada incidente en el proceso de cierre de incidente. De modo que la base del conocimiento contenga posibles soluciones a errores que se puedan repetir.		
Observaciones:	Funcionalidad aprobada.		
Historia de Usuario			
Numero:	RF17	Nombre:	Gestión del conocimiento – bitácora de errores.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad: (alta/media/baja)	alta		
Desarrollador encargado:	Ximena Catota		
Descripción:	Se requiere una pantalla para la bitácora de soluciones con las siguiente descripción: <ul style="list-style-type: none"> • Campo de ingreso: caja de texto para el ingreso del error o palabras relacionadas con el error. • Un botón buscar: que al presionar busque en la base de conocimientos soluciones posibles relacionadas con el error o las palabras ingresadas en caso de hallar coincidencias listarlas en un datagrid con la siguiente información: fecha de reporte del incidente, descripción corta del problema, solución, usuario que soluciono el incidente. 		
Observaciones:	Pantalla y funcionalidad aprobada.		

Historia de Usuario			
Numero:	RF18	Nombre:	Consulta de solicitudes
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad:	alta		

(alta/media/baja)	
Desarrollador encargado:	Ximena Catota
Descripción: Se requiere una pantalla para la consulta de solicitudes que cada usuario ingresa, esta pantalla debe desplegar únicamente las solicitudes reportadas por el usuario conectado. Debe contener los siguientes datos: <ul style="list-style-type: none"> - Numero de solicitud - Estado - Detalle 	
Observaciones:	

Historia de Usuario			
Numero:	RF19	Nombre:	Ingresar solicitud
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requiere una pantalla con las siguientes características para el ingreso de las solicitudes.			
Datos generales de lectura: <ul style="list-style-type: none"> • Agencia, cargo y nombre el usuario conectado. • Número de solicitud: campo de lectura auto generable. • Fecha de solicitud: campo de lectura que contenga la fecha de la conexión. 			
Detalle de la funcionalidad: <ul style="list-style-type: none"> • Se describe en las historias de usuario RF08 y RF09 			
Observaciones: Adición del campo detalle para solicitudes de tipo incidente. (Petición hecha fuera del levantamiento de requerimientos).			

Historia de Usuario			
Numero:	RF20	Nombre:	Asignar solicitud. (todas las áreas)
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1

Prioridad: (alta/media/baja)	baja
Desarrollador encargado:	Ximena Catota
Descripción:	
<p>Después de que la solicitud haya sido aprobada se requiere contar con una pantalla para la asignación de recursos, esta pantalla debe contar con la siguiente descripción:</p> <ul style="list-style-type: none"> • Combo box: listado de solicitudes pendientes para asignar, de modo que al seleccionar una se desplieguen los siguientes campos: • Campos de lectura: aplicativo, modulo, descripción corta de la solicitud, fecha de ingreso, tipo de requerimiento, usuario que reporta. • Combo box: para la asignación del nivel de impacto que tiene la solicitud, y prioridad de la misma. (Catálogos). • Detalle de las peticiones que conforman la solicitud: cada petición se puede asignar a diferentes recursos. • Botón guardar: al presionar este botón la información se almacena y se debe enviar un correo de aviso a los usuarios involucrados. 	
Observaciones:	

Historia de Usuario			
Numero:	RF21	Nombre:	Proceso de solución - levantamiento de especificaciones. (todas las áreas)
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	5	Iteración asignada:	4
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción:			
<p>Se requiere una pantalla con las siguientes características:</p> <ul style="list-style-type: none"> • Combo box: listado de solicitudes pendientes de solución con el detalle correspondiente, de modo que al seleccionar una se desplieguen los siguientes campos: <ul style="list-style-type: none"> ○ Campos de lectura: aplicativo, modulo, descripción corta de la solicitud, fecha de ingreso, tipo de requerimiento, usuario que reporta, nivel de impacto, prioridad, detalle de la petición. • Campos de ingreso: nombre y detalle de la especificación. • Botón guardar: almacenamiento de información. . (finalizada la etapa emisión del documento de especificaciones). • Se requiere que mediante esta pantalla el recurso pueda almacenar la información en uno o varios días, para lo cual se usa el campo finalizar etapa para que al estar activo se pueda pasar a la etapa de análisis. 			
Observaciones:			
*Aprobación de pantalla y funcionalidad.			

Historia de Usuario			
Numero:	RF22	Nombre:	Proceso de solución – análisis. (área de desarrollo)
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota.		
Descripción: Se requiere una pantalla para el ingreso del análisis de la solicitud que contenga los siguientes datos:			
<ul style="list-style-type: none"> • Combo box: listado de solicitudes pendientes. • Campos de lectura: aplicativo, modulo, descripción corta de la solicitud, fecha de ingreso, tipo de requerimiento, usuario que reporta, nivel de impacto, prioridad, detalle de la petición. • Campos de ingreso: descripción del análisis. • Campo finalizar etapa para que al estar activo se pueda pasar a la etapa de avances. • Botón guardar: almacenamiento de información. . (finalizada la etapa emisión del documento de análisis). • Si hay tareas compartidas en una solicitud no se debe poder avanzar de etapa sin que todos los recursos hayan ingresado su análisis de la tarea asignada. 			
Observaciones: *Pantalla y funcionalidad aprobada. *En definición del formato del documento de análisis (institución aún no entrega formatos).			

Historia de Usuario			
Numero:	RF23	Nombre:	Proceso de solución avances.(desarrollo)
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	3	Iteración asignada:	2
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requiere una pantalla para el registro de avances del desarrollo que dará solución a la solicitud, dicha pantalla contendrá los siguiente:			
<ul style="list-style-type: none"> • Combo box: listado de solicitudes pendientes. • Campos de lectura: aplicativo, modulo, descripción corta de la solicitud, fecha de ingreso, tipo de requerimiento, usuario que reporta, nivel de impacto, prioridad, detalle de la petición. • Campos de ingreso: nombre y descripción del avance (campos obligatorios) y observaciones, Combo box que contenga las 24 horas, Chek de si es para base de datos si el avance se trata de una afectación a la base de datos. • Campo finalizar etapa para que al estar activo se pueda pasar de etapa.(pruebas) • Botón guardar: almacenamiento de información. (finalizada la etapa emisión del documento de avances) 			
Si hay tareas compartidas en una solicitud no se debe poder avanzar de etapa sin que todos los recursos hayan finalizado los avances de la tarea asignada.			

Observaciones: *Pantalla y funcionalidad aprobada. *En definición del formato del documento de avances (institución aún no entrega formatos).
--

Historia de Usuario			
Numero:	RF24	Nombre:	Proceso de solución pruebas(desarrollo)
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción:			
Se requiere una pantalla para el registro de pruebas del desarrollo de la solicitud. <ul style="list-style-type: none"> • Combo box: listado de solicitudes pendientes. • Campos de lectura: aplicativo, modulo, descripción corta de la solicitud, fecha de ingreso, tipo de requerimiento, usuario que reporta, nivel de impacto, prioridad, detalle de la petición. • Campos de ingreso: descripción de las pruebas, anexos (campos obligatorios) y recomendaciones, Campo finalizar etapa fin del ciclo. • Botón guardar: almacenamiento de información y emisión del documento de pruebas y certificación. 			
Observaciones:			
*En definición del formato del documento de pruebas y certificación (institución aún no entrega formatos).			

Historia de Usuario			
Numero:	RF25	Nombre:	Parametrización general.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	4	Iteración asignada:	3
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción:			
Se solicitan pantallas para el mantenimiento de cierta información parametrizable: <ul style="list-style-type: none"> • Pantalla para la información de agencias (creación, edición y almacenamiento de la información). • Pantalla para la información de departamentos de la institución (creación, edición y almacenamiento de la información). • Pantalla para la información de aplicativos de cada área (creación, edición y almacenamiento de la información). • Pantalla para la información de módulos de cada aplicativo (creación, edición y almacenamiento de la información). 			
Observaciones:			

*Pantallas y funcionalidad aprobada

Historia de Usuario			
Numero:	RF26	Nombre:	Parametrización base del conocimiento.
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requieren pantallas para la parametrización de la base del conocimiento: <ul style="list-style-type: none">• Pantalla para parametrizar el número de palabras y los rangos de coincidencia permitidos para la búsqueda de soluciones en la bitácora de soluciones. (creación, adición y almacenamiento)• Pantalla para la adición de palabras que no se deban contemplar en la búsqueda de soluciones en la base del conocimiento. (creación y almacenamiento)			
Observaciones: *Pantallas y funcionalidad aprobada.			

Historia de Usuario			
Numero:	RF27	Nombre:	Reportes generales.
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	media		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requiere una pantalla para el acceso a reportes generales tales como: <ul style="list-style-type: none">• Solicitudes en gestión: este reporte debe ser exportable a excel, pdf o word y debe contener los siguientes campos: área, solicitante, recurso asignado, código de solicitud, prioridad, descripción corta, fecha de ingreso, fecha de asignación, fecha de finalización filtrado por fechas de ingreso.• Solicitudes por usuario: este reporte debe ser exportable a excel, pdf o word y debe contener los siguientes campos: tipo de solicitud, código de solicitud, prioridad, fecha de solicitud, fecha de asignación, fecha de finalización parámetro de entrada: código usuario.• Reporte de calificaciones de los incidentes: este reporte debe ser exportable a excel, pdf o word y debe contener los siguientes campos: código de incidente, prioridad, impacto, usuario solicitante, recurso asignado, fecha de incidente, fecha de finalización, calificación y comentario de la calificación.• Reporte del catálogo de aplicaciones.• Reporte del catálogo de módulos por aplicaciones.			
Observaciones:			

*Reportes aprobados.

Historia de Usuario			
Numero:	RF28	Nombre:	Reportes estadísticos.
Usuario:	Coordinador del área de desarrollo.		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción: Se requiere una pantalla para el acceso a reportes estadísticos tales como: <ul style="list-style-type: none">• Gráficos estadísticos del promedio de calificaciones obtenidas en un rango de fechas.• Gráficos estadísticos del promedio de incidentes resueltos en un rango de fechas.• Gráficos estadísticos del promedio de solicitudes gestionadas en un rango de fechas. Estos reportes deben ser exportables a pdf.			
Observaciones:			
Historia de Usuario			
Numero:	RF29	Nombre:	Eliminación de firmas
Usuario:	Coordinador del área de desarrollo		
Modificación historia No	2	Iteración asignada:	1
Prioridad: (alta/media/baja)	baja		
Desarrollador encargado:	Ximena Catota		
Descripción: En un inicio se solicitó la adición de firmas en imágenes de los usuarios pero se realizó un ajuste en ese requerimiento ya que para la institución son necesarias las firmas plasmadas en documentos físicos para la entrega de documentación al departamento de auditoria. Por lo que el área considera poco usable esta funcionalidad y una carga innecesaria de data en la base de datos con las imágenes de las firmas de cada uno de los usuarios del sistema mesa de ayuda.			
Observaciones: *Eliminación de alcance.			

2.7 Listado de entregables

De acuerdo con lo establecido en las especificaciones del sistema se tiene la siguiente lista de entregables.

Tabla 3. *Listado de entregables.*

Entregable	Descripción	Historias de usuario
Diagrama de procesos tentativo.	Consisten en establecer un diagrama del proceso tentativo que se llevara a cabo en cuanto a la gestión de trabajos de help desk.	--
Diagrama de entidad relación	Consisten en el diagrama correspondiente al diseño de datos.	--
Entregable 1 de software	Pantallas y funcionalidad para la parametrización de catálogos.	RF01, RF25, RF26
Entregable 2 de software	Pantallas y funcionalidad para la autenticación de usuarios, gestión de claves de acceso y roles que se asignaran a cada uno.	RF01, RF03, RF05, RF06
Entregable 3 de software	Pantallas y funcionalidad sobre el proceso de una solicitud de tipo requerimiento: ingreso, aprobación, asignación de solicitudes y envío de correo electrónico. (Todas las áreas: desarrollo, producción y BDD).	RF07, RF08, RF09, RF10, RF19, RF20
Entregable 4 de software	Pantallas y funcionalidad sobre el proceso de solución de solicitudes de tipo requerimiento: etapas de levantamiento de especificaciones, análisis, avances y pruebas.	RF21, RF22, RF23, RF24

Entregable 5 de software	<ul style="list-style-type: none"> - Pantallas y funcionalidad sobre la gestión de incidentes: ingreso, asignación, solución y devolución de incidentes. - Pantallas y funcionalidad de la gestión del conocimiento: bitácora de errores conocidos, funcionalidad como sugerencia de solución a incidentes nuevos. 	RF11,RF12, RF13,RF16, RF17, RF18
Entregable 6 de software	<ul style="list-style-type: none"> - Consulta de estados de las solicitudes ingresadas por el usuario. - Reportes funcionales, reportes estadísticos y generación de documentación al finalizar cada etapa. - Pantalla de ejecución de reportes. 	RF27, RF28, RF30.
Diagrama de navegación	Entrega del diagrama navegación de la página web.	--
Pruebas de aceptación.	Consiste en la ejecución de pruebas integrales del sistema completo.	--

Nota. Lista de entregables del proyecto. Fuente: autoría
Elaborado por: Ximena Catota.

CAPÍTULO 3

ANÁLISIS Y DISEÑO

3.1 Análisis

3.1.1 Situación actual de Bancodesarrollo

Bancodesarrollo cuenta con quince agencias en el país cuya operatividad se administra en la agencia matriz ubicada en Quito, sector la Floresta, en esta dependencia funciona las siguientes áreas:

Organigrama de Bancodesarrollo

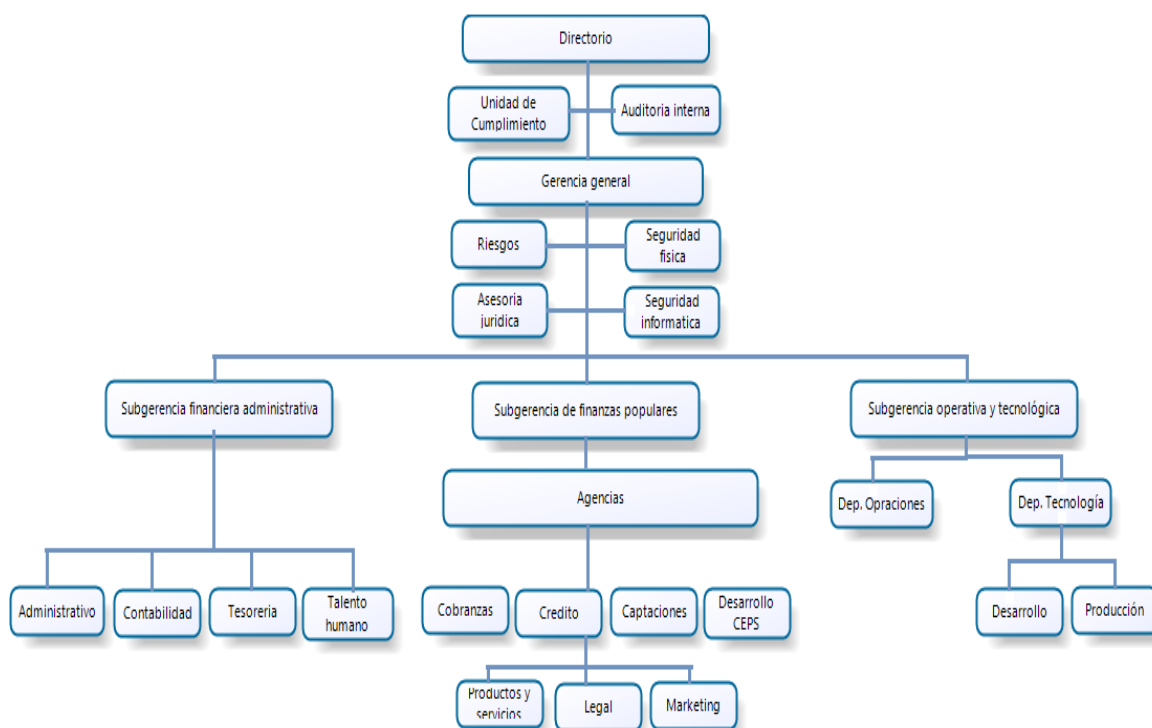


Figura 8. Organigrama de Bancodesarrollo.

Fuente: Bancodesarrollo.

Elaborado por: Ximena Catota

Hoy en día Bancodesarrollo cuenta con una variedad de hardware y software necesarios en la operatividad del mismo. Todas estas tecnologías son de gran importancia pero a continuación se mencionaran las que generan mayor atención para el banco.

Tabla 4. Listado de hardware y software de Bancodesarrollo.

Nombre	Descripción
Core financiero Financiamiento	Es el sistema informático principal del banco ya que mediante este se manejan todas las reglas del negocio del banco.
Risk control	Sistema informático usado para la prevención de lavado de activos.
Scoring	Sistema informático usado para el análisis de riesgos en los posibles créditos entregados.
Cuentas corrientes	Sistema informático para la administración de cuentas corrientes.
Correo electrónico institucional	Servicio de red que permite a los usuarios enviar y recibir mensajes
Internet	Red de comunicación
Redes	Conexión de los equipos informáticos a la red del banco.

Equipos de computación	Computadoras, impresoras, servidores, etc.
Base de datos del Core financiero	Entidad en la cual se pueden almacenar los datos generados en el Core financiero Financial.

Nota. Lista de algunos programas y herramientas de Bancodesarrollo. Fuente: Bancodesarrollo.
Elaborado por: Ximena Catota.

El departamento de tecnología es el responsable de la administración funcional de hardware y software de la institución (tabla 4), los mismos que son usados por el personal que labora en las diferentes agencias.

Diariamente el departamento de tecnología recibe reportes de inquietudes, incidentes, peticiones y solicitudes de cambio relacionadas con el funcionamiento de estas herramientas.

Actualmente se manejan tres tipos de peticiones que hace el cliente interno al departamento de tecnología estas son:

- **Incidente:** problema que afecta a la operatividad de los servicios que brinda el banco a sus clientes externos cuya solución debe ser inmediata.
- **Requerimiento:** solicitud de cambio o nueva funcionalidad sobre hardware o software del banco cuya solución se da a largo plazo.
- **Afectación a la base de datos:** solicitud de actualización a la base de datos sobre la información del banco que esta contiene cuya solución puede ser inmediata o a largo plazo dependiendo el problema.

Estas peticiones han sido atendidas de una manera informal y hasta hace poco este tipo de atención fue sostenible pero como se mencionó en el apartado 1.3 es necesario mejorar la calidad en este servicio.

3.1.2 Análisis de los procesos actuales.

Se realizó un análisis sobre la gestión de trabajos de help desk que actualmente realiza el departamento de tecnología de Bancodesarrollo y se pudo obtener la siguiente información:

1. En el momento que un cliente interno tiene algún incidente sobre el funcionamiento de algún servicio (hardware o software) que usa solicita soporte a cualquier persona que labore en el departamento de tecnología sin importar al

área a la que pertenezca, esto lo hace a través de diferentes medios: correo electrónico, telefónicamente, chat institucional, verbalmente, etc. Y casi siempre usando la prioridad de urgente.

Mejora con ITIL

- a) Mediante el concepto que maneja ITIL sobre la **definición de catálogos de servicios** que discrimina a los activos de los no activos y los agrupa por familiaridad que comúnmente esto determina el área al que pertenece dicho servicio poner a conocimiento del cliente este catálogo para obtener una correcta dirección del incidente o requerimiento.
- b) Usando el aparato de **gestión de incidencias** de ITIL realizar la priorización de las mismas basándose en los dos puntos importantes que este estándar recomienda:
 - **Impacto:** número de usuarios afectados
 - **Urgencia:** tiempo que se necesita

2. El incidente reportado no recibe alguna identificación para su tratamiento de manera que es tratado por la problemática.

Mejora con ITIL

Con el propósito de mejorar la administración de un incidente se propone usar una identificación única para el incidente tal y como lo recomienda ITIL en la **gestión de incidencias**.

3. El técnico al que fue reportado el incidente verifica el área al que pertenece y si no es su área devuelve el incidente al cliente.

Mejora con ITIL

ITIL recomienda recurrir a un especialista o a un superior que pueda tomar decisiones. Por lo cual se propone usar este escalamiento para no devolver el incidente al cliente y minimizar el tiempo de solución.

4. Después de que el técnico dio solución al incidente le notifica al usuario que el incidente ha sido superado y con esto finaliza el proceso.

Mejora con ITIL

- a) Con el propósito de tranquilizar al cliente sobre la incertidumbre que genera el estado de los incidentes que reporta. Se propone el uso de lo que recomienda ITIL una monitorización del estado del incidente.
 - b) Como lo recomienda ITIL se propone que después de confirmar con el usuario la solución del incidente se proceda con el cierre del mismo.
5. En el proceso de solución del incidente no hay documentación de la misma y en muchas ocasiones las incidencias se repiten. Lo que provoca una pérdida de tiempo tanto para el técnico como para el cliente.

Mejora con ITIL

Se propone el uso de la **base de errores conocidos** que recomienda ITIL en la que se almacena las incidencias ocurridas y la solución que se dio a cada una. Con esta mejora se evita la pérdida de tiempo en atender errores ya conocidos y adicionalmente se le da al cliente la posibilidad de solucionar las incidencias por sí mismo.

6. En cuanto a lo que se refiere a requerimientos se cuenta con un proceso mejor establecido y documentación únicamente para los que corresponden al área de desarrollo y en casos de afectación a la base de datos.

En el caso de afectación a la base de datos del Core financiero el proceso es más cuidadoso y menos frecuente ya que requiere la aprobación de las siguientes áreas: operaciones, tecnología, auditoría, y gerencia general.

Mejora con ITIL

- a) Con el objetivo de mejorar el proceso que llevan los requerimientos se propone el uso de lo que recomienda ITIL en la **gestión de cambios** sobre la aprobación, análisis, planificación e implementación del mismo. Ya que actualmente en el banco existen los procesos de aprobación, análisis y pruebas de los cambios realizados.
- b) Adicionalmente se recomienda el uso de esta mejora para todas las áreas.

3.1.2.1 Diagrama de procesos actuales.

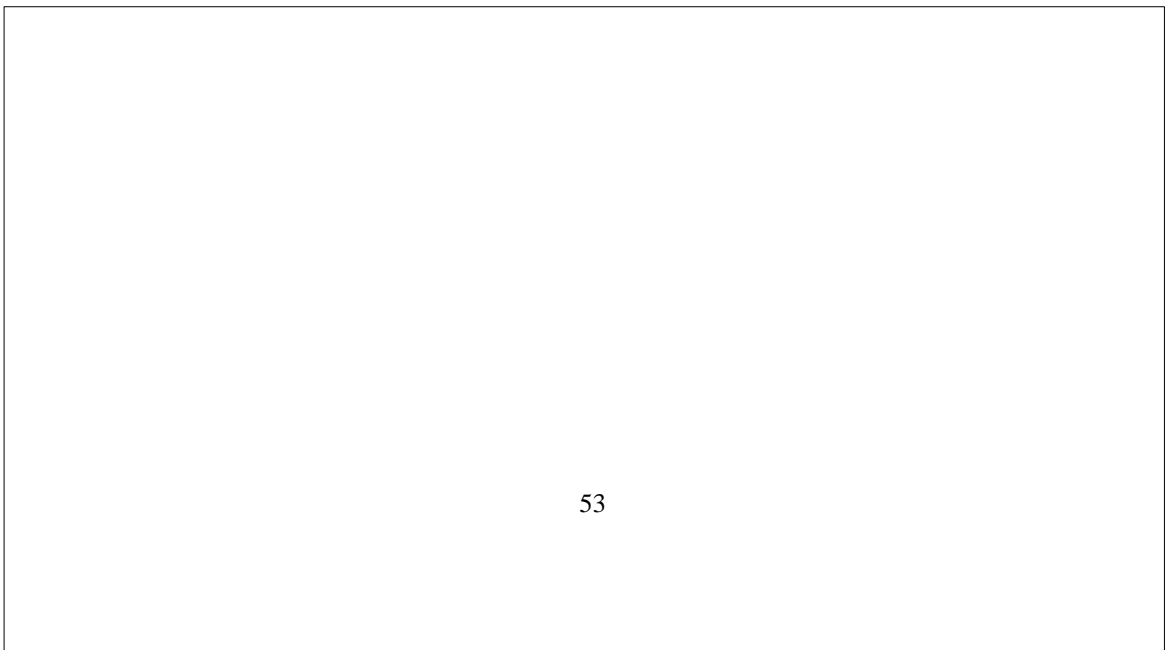


Diagram de procesos previo al sistema de mesa de ayuda

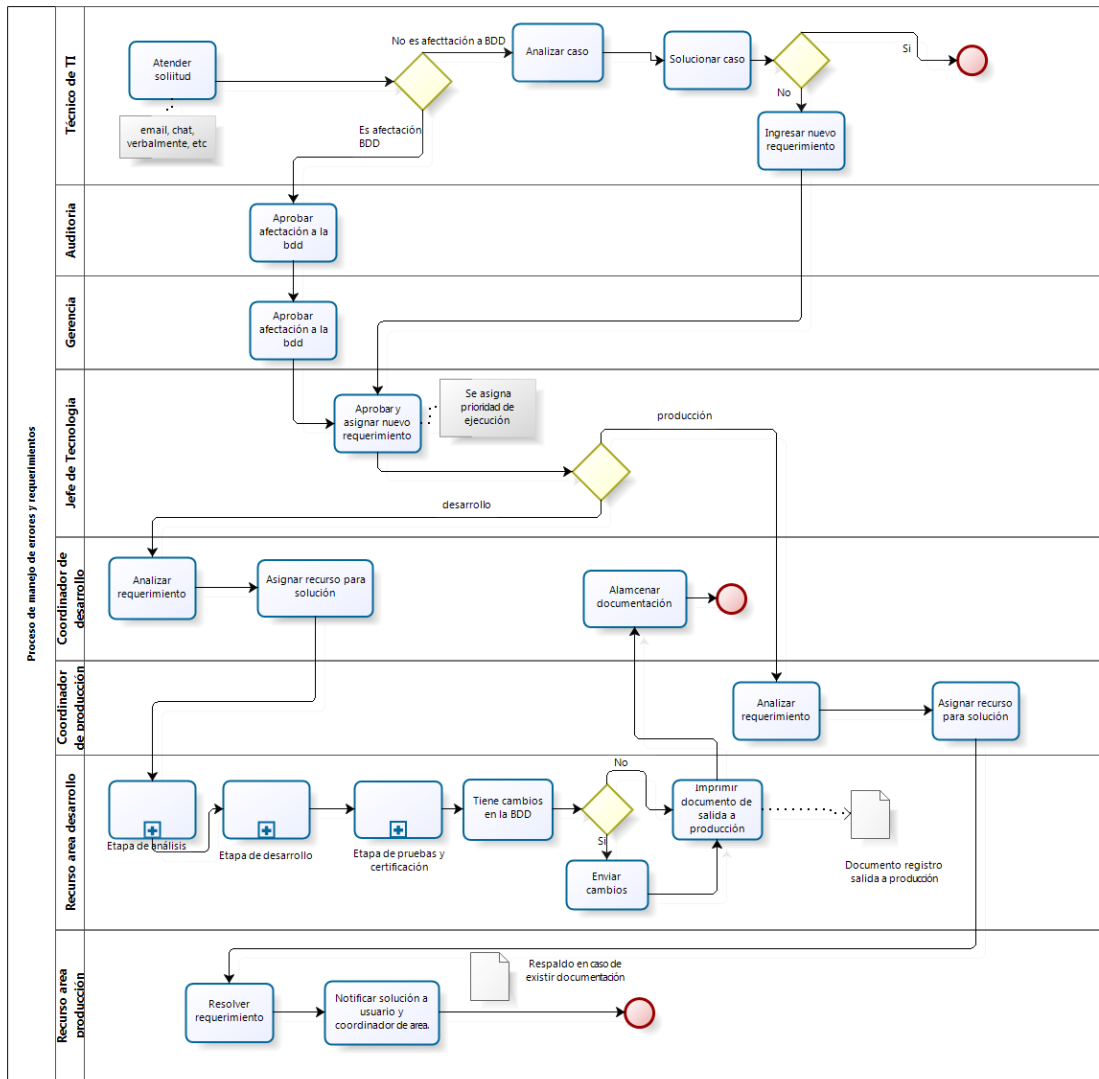


Figura 9: Diagrama del proceso actual de la institución

Elaborado por: Ximena Catota.

3.1.2.2 Diagrama de casos de uso

Los roles que interactúan con el sistema de mesa de ayuda son:

Tabla 5. *Actores del sistema mesa de ayuda*

Núm.	Actor	Descripción
1	Administrador	Este usuario se encarga de gestionar los datos del sistema, como la configuración de parámetros y asignación de permisos.
2	Jefe de tecnología	Este usuario se encarga de aprobar los requerimientos o afectación a base de datos.
	Jefe de operaciones	
3	Gerente	Estos usuarios se encarga de aprobar únicamente los requerimientos que corresponden a una afectación a la BDD del Core financiero.
4	Auditor	
5	Coordinador de área	Este usuario se encarga de la asignación de requerimientos, tareas e incidentes.
6	Técnico especializado	Este usuario se encarga de solucionar los incidentes, tareas y requerimientos. Se encarga de alimentar la base del conocimiento con el registro de soluciones que se otorgan a los incidentes.
7	Cliente interno	Este usuario se encarga de ingresar al sistema los incidentes y requerimientos. Y realizar su seguimiento. También evalúa la solución de incidentes.

Nota. Actores involucrados en el sistema. Fuente: Autoría.
Elaborado por: Ximena Catota.

Diagrama general del sistema web mesa de ayuda.

CASOS DE USO

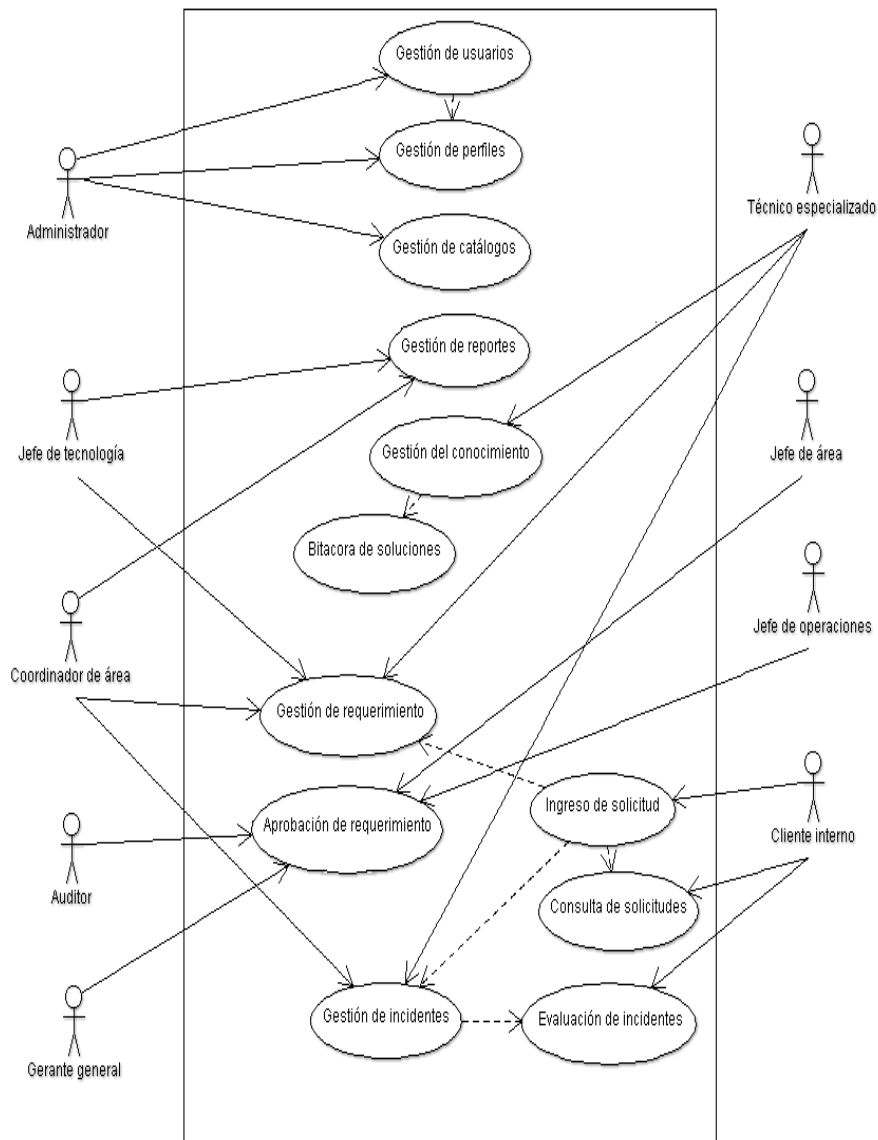


Figura 10. Diagrama del sistema web mesa de ayuda.

Fuente: Bancodesarrollo

Elaborado por: Ximena Catota

3.2 Diseño del sistema

En este apartado se describen los puntos que se analizaron y especificaron en los requerimientos del software.

3.2.1 Diseño de datos

Durante el desarrollo del software se realizó el modelado de los datos empleados por el sistema cuya representación gráfica se plasma en el diagrama de datos.

A continuación el diseño de datos del sistema mesa de ayuda:

3.2.2 Diseño arquitectónico

Como lo menciona Shaw en esta sección se definen los elementos estructurales del software para lo cual se usan los siguientes diagramas:

3.2.2.1 Diagrama de clases

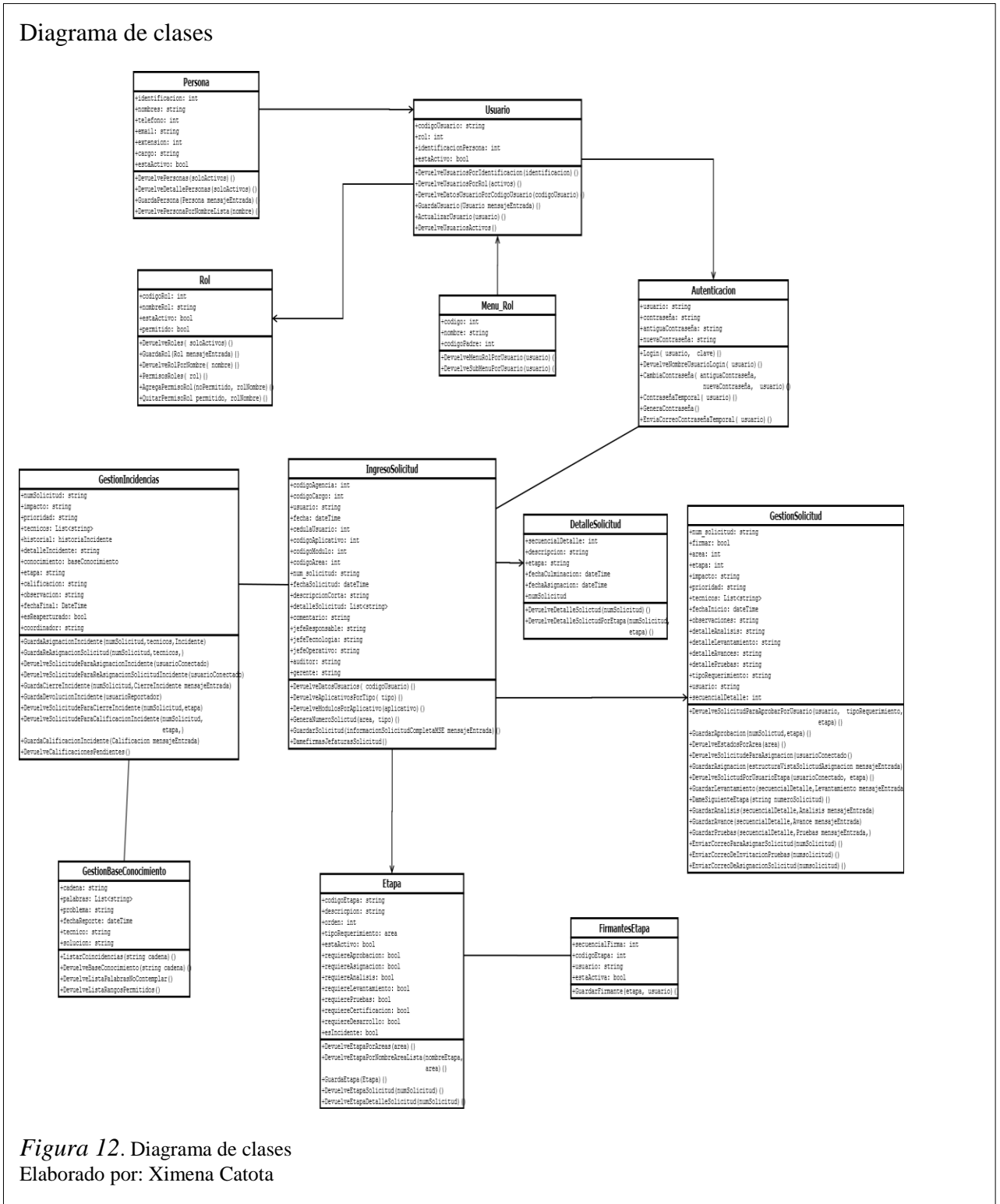
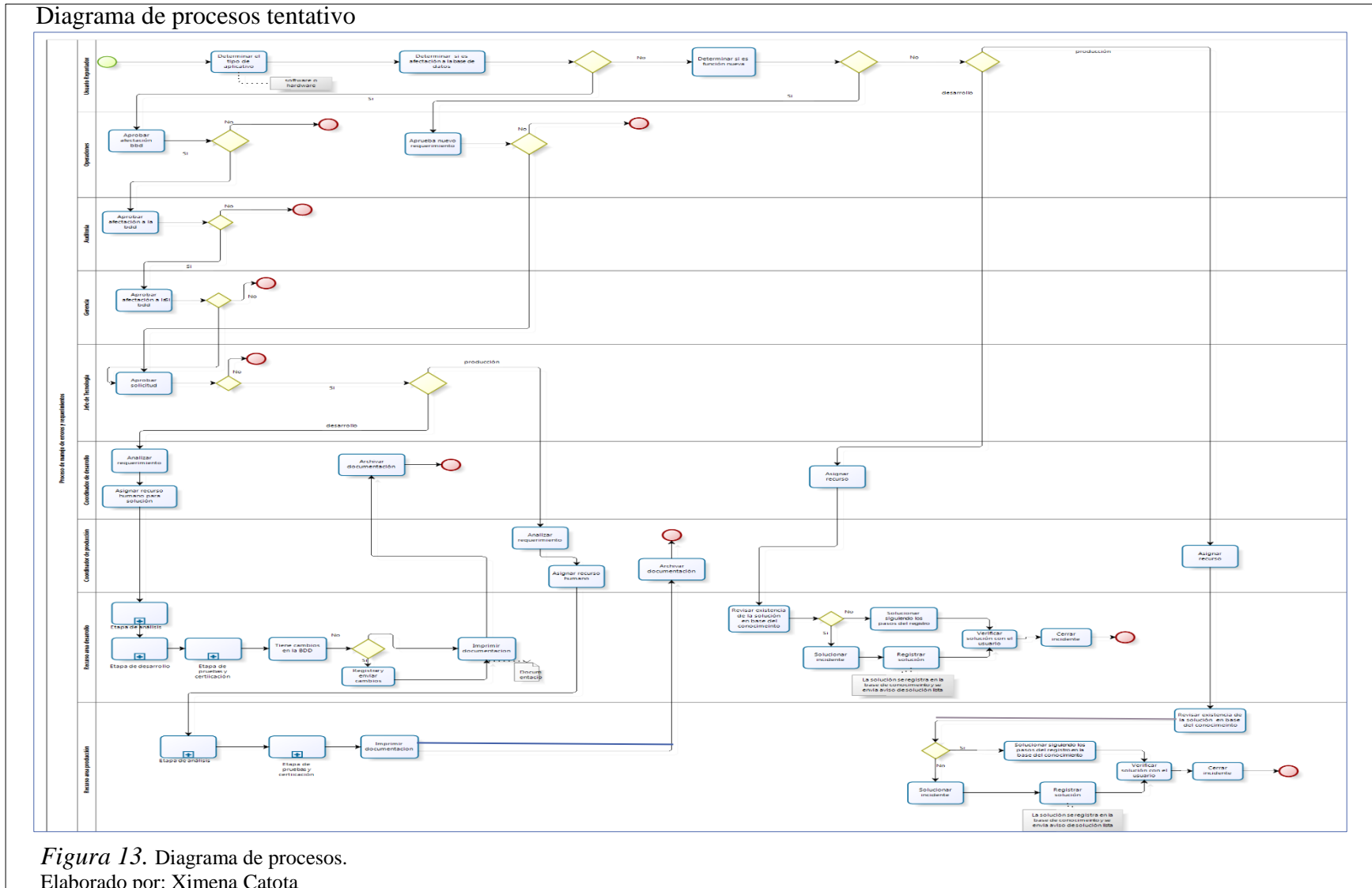


Figura 12. Diagrama de clases
Elaborado por: Ximena Catota

3.2.2.2 Diagrama de proceso tentativo.



3.2.3 Diseño de interfaces

En esta sección se trata de dar al usuario un bosquejo de las principales pantallas que podrá tener en el sistema.

A continuación las interfaces principales que conforman el sistema.

1. En este diseño se le proporciona al cliente una idea de la interfaz que tendrá el acceso al sistema.

usuario

Ingreso del

Usuario:		<input type="text"/>
Contraseña:		<input type="text"/>
		<input type="button" value="Ingresar"/>

Figura 14. Pantalla que permite al usuario ingresar al sistema
Elaborado por: Ximena Catota.

2. Después de que el usuario ingrese al sistema tiene la siguiente pantalla de inicio, en este bosquejo se le da al cliente la idea de la estructura que tendrá la interfaz de inicio.

Menú principal

BANER		
Inicio	Salir	Acerca de
M E N Ú	FORMULARIOS	

Figura 15. Interfaz de inicio

Administrador de roles

Código:

Nombre:

Esta Activo:

Lista de roles

Permisos

No permitidos

⇒

⇐

Permitidos

Figura 17. Interfaz de administrador de roles

Elaborado por: Ximena Catota

5. Como ya se lo había registrado en las historias de usuario el sistema maneja dos tipos de solicitudes, a continuación el bosquejo de la interfaz de ingreso solicitud tipo requerimiento.

Ingreso de solicitud

Agencia:

Cargo:

Usuario:

Número solicitud:

Fecha:

Requerimiento: Incidente: Área: ▾
 Aplicativo: ▾
 Modulos: ▾

Descripción corta:

Descripción detallada:

Anexos y fuentes

Comentario:

Sugerencia:

▾

Jefe de área

▾

Jefe de tecnología

▾

Jefe de operaciones

■ Campos de lectura

Figura 18. Interfaz de ingreso de solicitud

Elaborado por: Ximena Catota

6. El siguiente bosquejo presenta la interfaz donde se podrá realizar el proceso de aprobación en el caso de requerimientos.

Aprobación de requerimientos

63

Tipo de requerimiento: ▾

Etapas: ▾

Firmar

Aplicativo	Descripción	Num de solicitud	Firmar
			<input type="checkbox"/>

■ campos de lectura

Figura 19. Interfaz de aprobación de requerimientos.
Elaborado por: Ximena Catota

7. En el siguiente bosquejo se le proporciona al cliente una idea de la interfaz que será utilizada para la asignación de requerimientos.

Asignación de requerimiento

Asignar requerimiento

Número de incidente: <input type="text"/> ▾	Fecha de ingreso: <input type="text"/>
Aplicativo: <input type="text"/>	Tipo de requerimiento: <input type="text"/>
Módulo: <input type="text"/>	Nivel de impacto: <input type="text"/> ▾
	Prioridad: <input type="text"/> ▾
	Usuario que reporta: <input type="text"/>

Id detalle	Descripción	Técnico	Fecha inicio
01	Tarea de e empleo		

Recurso: ▾

Fecha inicio: ▾ 15

Seleccionar

Guardar

⏏ Pantalla alterna ■ campos de lectura

Figura 20. Interfaz de asignación de solicitud
Elaborado por: Ximena Catota.

8. En el siguiente bosquejo se presenta al cliente la interfaz que corresponderá al ingreso de incidentes.

Ingreso de incidente

64

Figura 21. Interfaz de ingreso de incidente

Elaborado por: Ximena Catota

9. En el siguiente bosquejo se proporciona al cliente una idea de lo que será la interfaz para la asignación de incidentes.

Figura 22. Interfaz para la asignación de incidentes

Elaborado por: Ximena Catota

10. En el siguiente bosquejo se proporciona al cliente una idea de la interfaz que se tendrá para el seguimiento de incidentes, en esta pantalla el usuario podrá cerrar el incidente dándole solución o devolviendo el incidente en el caso de que sea necesario. Así como también tendrá la posibilidad de buscar una posible solución en la base del conocimiento.

Seguimiento de incidentes

Seguimiento del incidente

Número de incidente: Fecha de ingreso:

Aplicativo: Tipo de requerimiento:

Módulo: Nivel de impacto:

Descripción corta: Prioridad:

Usuario que reporta:

Detalle del incidente:

Posibles soluciones

Fecha	Problema	Solución	Técnico

Buscar solución

Solución:

Observación:

Estado:

Pantalla alterna
 Campos de lectura

Figura 23. Interfaz de seguimiento del incidente

Elaborado por: Ximena Catota

11. El siguiente bosquejo es sobre la interfaz de evaluación cuya funcionalidad se construyó fuera del alcance pero tomando en cuenta las recomendaciones de ITIL.

Evaluación de incidentes

Evaluación del incidente

Número de incidente: Fecha de ingreso:

Aplicativo: Tipo de requerimiento:

Módulo: Nivel de impacto:

Descripción corta: Prioridad:

Usuario que reporta:

Conteste las siguientes preguntas:

¿Se cumplieron los o el objetivo previsto?
 Si: No:

¿Qué calificación usted asignaría a la solución del incidente?
 Calificación:

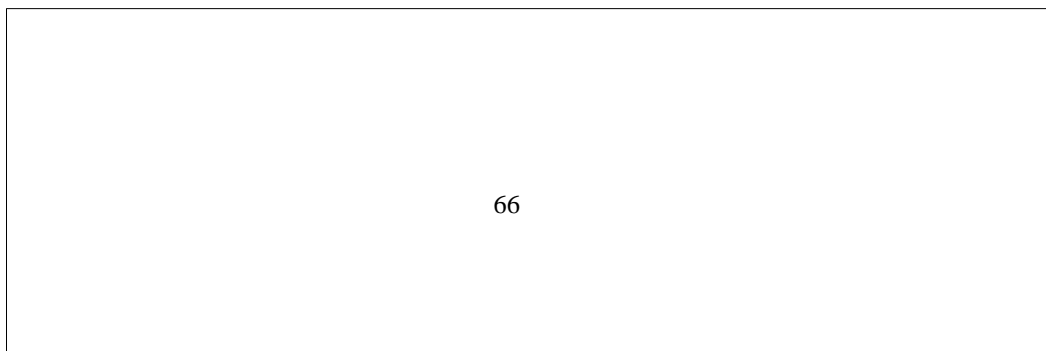
Comentario:

campos de lectura

Figura 24. Interfaz de evaluación del incidente

Elaborado por: Ximena Catota

12. En el siguiente bosquejo se proporciona al cliente una idea de la interfaz que será usada para consultar la base del conocimiento.



Base del conocimiento

Base del conocimiento

Ingresar palabras claves:

Número solicitud	Problema	Solución	Técnico

Figura 25.

Interfaz de la base del conocimiento
Elaborado por: Ximena Catota.

13. En el siguiente bosquejo se proporciona al cliente una idea de la interfaz que será usada para el monitoreo del proceso de solución de todas las solicitudes ingresadas.

Consulta de solicitudes

CONSULTA DE SOLICITUDES

Solicitudes ingresadas

Aplicativo	Código Solicitud	Descripción de solicitud	Estado

Figura 26. Interfaz para la consulta de solicitudes
Elaborado por: Ximena Catota

3.3 Arquitectura del sistema

En el caso del sistema informático para la gestión de trabajos de help desk para el Bancodesarrollo. Se usó la arquitectura cliente servidor de 3 capas.

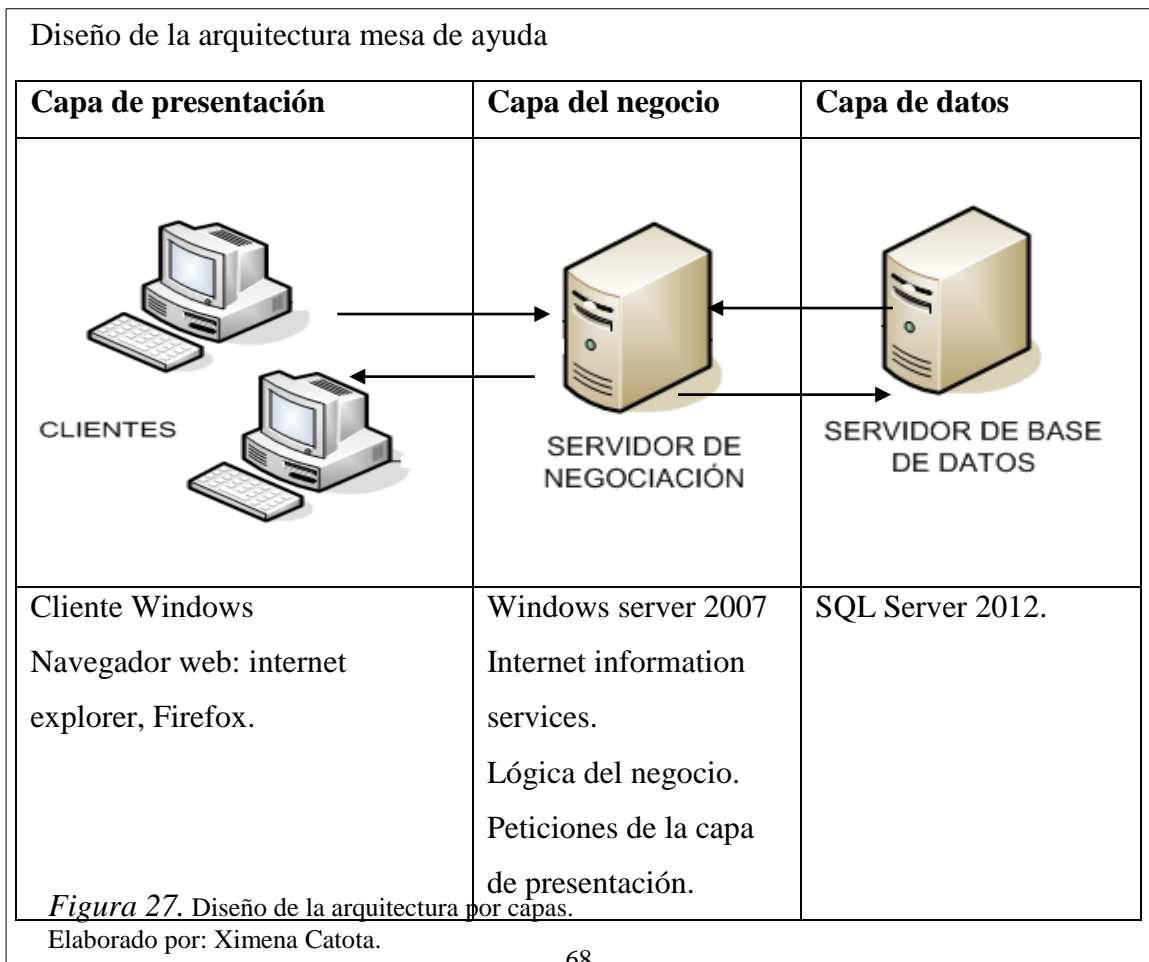
Cuya infraestructura consta de tres nodos procesadores que son:

- Estación de trabajo (cliente).
- Servidor web y
- Base de datos.

Capa de presentación o de usuario: interfaz gráfica que debe tener la característica de ser amigable y fácil de usar. Página web que visualiza el usuario desde su estación de trabajo.

Capa de negocio: servidor donde se ejecutan las peticiones hechas desde la capa de presentación.

Capa de datos: Base de datos.



3.3.1 Diagrama de navegación

Diagrama de navegación

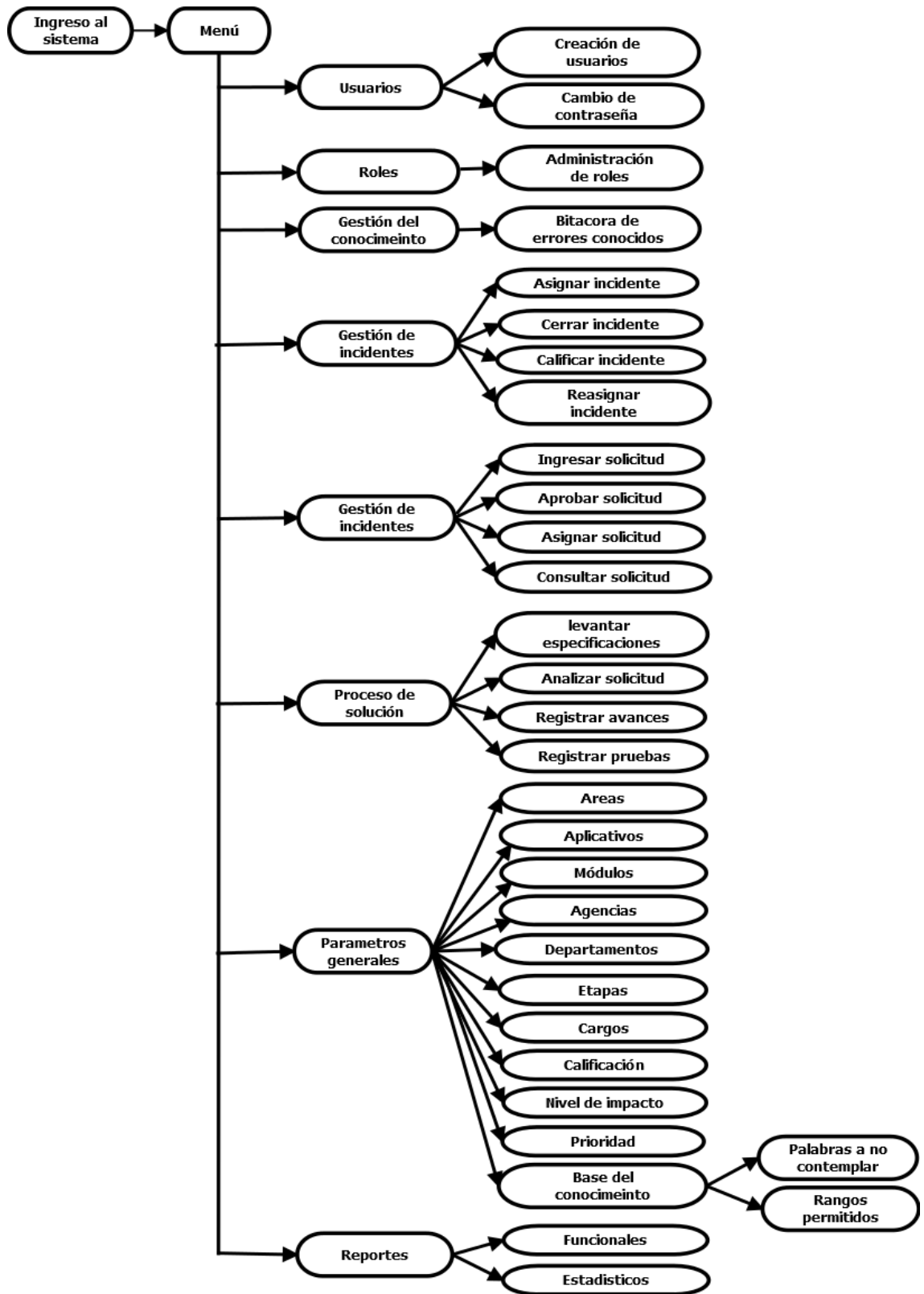


Figura 28. Diagrama de navegación
Elaborado por: Ximena Catota

CAPÍTULO 4

CONSTRUCCIÓN Y PRUEBAS

4.1 Construcción

4.1.1 Estándares de programación

Programación del servidor

- Se omitirán las tildes en todos los casos.
- Se usaran abreviaturas en casos necesarios.
- Se usara la primera letra en mayúscula para nombrar los métodos.

Atributos

- Si el nombre del atributo tiene más de una palabra la primera letra de la primera palabra está en minúscula y la primera letra de las demás palabras estarán en mayúscula.
- Comienza con un guión bajo (“_”), ejemplo: `_estaActivo`
- Variables y parámetros
- Uso de convenciones de nombres tipo CAMEL

Ejemplo: `etapa`, `numeroSolicitud`

Métodos

- Uso de convenciones de nombres tipo PASCAL

Ejemplo: `GuardaSolicitud`, `AsignaIncidente`

Clases

- Usar nombres en singular
- Usar convenciones tipo PASCAL.

Base de datos.

- Uso del formato en cada tabla `TBL_Nombre_de_la_tabla`

Desarrollo de interfaces

- Se usa la abreviatura `frmNombre` para llamar a los frame usados.

- Se usa abreviaturas para los componentes ejemplo: txt_nombre, cbox_nombre, dtg_nombre, btnNombre, etc.

4.1.2 Código relevante

En un sistema todo el código fuente que lo conforman es importante pero para proyectar una idea de cómo está codificado el sistema se presenta a continuación el código más relevante.

4.1.2.1 DataContext

Para el desarrollo de este sistema informático se utilizó LINQ para una más fácil manipulación de datos, para ello se debe hacer una asignación de un DataContext a la base de datos.

“El DataContext permite a LINQ el uso del mismo patrón operador de consulta estándar tales como select, y where” (msdn.microsoft.com, LINQ, 2015)

```
[global::System.Data.Linq.Mapping.DatabaseAttribute(Name="BDMesaAyuda")]
public partial class DCMesaAyudaDataContext : System.Data.Linq.DataContext
{
    private static System.Data.Linq.Mapping.MappingSource mappingSource = new AttributeMappingSource();

    #region Definiciones de métodos de extensibilidad
    partial void OnCreated();
    partial void InsertTBL_AGENCIA(TBL_AGENCIA instance);
    partial void UpdateTBL_AGENCIA(TBL_AGENCIA instance);
    partial void DeleteTBL_AGENCIA(TBL_AGENCIA instance);
    partial void InsertTBL_USUARIO(TBL_USUARIO instance);
    partial void UpdateTBL_USUARIO(TBL_USUARIO instance);
    partial void DeleteTBL_USUARIO(TBL_USUARIO instance);
    #endregion

    partial void DeleteTBL_FIRMANTES_ETAPA(TBL_FIRMANTES_ETAPA instance);
    #endregion

    public DCMesaAyudaDataContext() :
        base(global::System.Configuration.ConfigurationManager.ConnectionStrings["BDMesaAyudaConnectionString"].ConnectionString, mappingSource)
    {
        OnCreated();
    }
}
```

DataContext

Figura 29. DataContext: archivo de configuración para conexión de datos.

Elaborado por: Ximena Catota

Tabla 6. *Métodos DataContext*

Nombre	Versión
DataContext(IDbConnection)	Inicializa una nueva instancia de la clase DataContext haciendo referencia a la conexión utilizada por .NET Framework.
DataContext(String)	Inicializa una nueva instancia de la clase DataContext haciendo referencia a un origen de archivo.
DataContext(IDbConnection, MappingSource)	Inicializa una nueva instancia de la clase DataContext haciendo referencia a una conexión y un origen de asignación.
DataContext(String, MappingSource)	Inicializa una nueva instancia de la clase DataContext haciendo referencia a un origen de archivo y un origen de asignación.

Nota. Algunos métodos que usa el DataContext. Fuente: .microsoft.com

Elaborado por: Ximena Catota

4.1.2.2 Contrato de servicios

Para la comunicación entre la capa de presentación y la capa de negocio se usa la tecnología **Windows Communication Foundation (WCF)** con la que es posible enviar datos como mensajes asincrónicos de un extremo de servicio a otro.

En este caso un extremo es el cliente del servicio que solicita datos de un extremo de servicio.

Contratos de servicios WCF

```

[ServiceContract]
public interface IsvcMesaAyuda
{
    [OperationContract]
    List<string> DevuelveCatalogoImpacto(bool soloActivos);

    [OperationContract]
    List<Tbl_CatalogoImpacto> devuelveDetalleCatalogoImpacto(bool soloActivos);

    [OperationContract]
    string GuardaCatalogoImpacto(Tbl_CatalogoImpacto mensajeEntrada, bool esEdicion, bool validaDuplicado);

    [OperationContract]
    Tbl_CatalogoImpacto devuelveImpactoPorNombreLista(string nombreLista);

```

Figura 30. Contrato de servicios - WCF
Elaborado por: Ximena Catota

[ServiceContract] “Indica que una interfaz o una clase definen un contrato de servicio en una aplicación WCF” (.microsoft.com, 2015)”.

[OperationContract] “Indica que un método define una operación que forma parte de un contrato de servicio en una aplicación de WCF” (.microsoft.com, 2015)”.

4.1.2.3 Código de búsqueda

En el presente proyecto se quiere destacar una parte del código que es importante y se trata del código de programación que permite al cliente encontrar una o varias coincidencias de palabras contenidas en una o muchas frases.

Este código es importante para el uso de la base del conocimiento que subraya ITIL y que en el sistema se destaca por permitir al usuario la posibilidad de encontrar solución a incidentes antes presentados.

- e) Se forma un conjunto de palabras de la cadena ingresada por la interfaz gráfica.
 - i) Se busca en la base de datos el listado de las palabras que no se deben contemplar para la búsqueda de coincidencias.
 - ii) Se arma un arreglo de palabras válidas para la búsqueda de coincidencias.

Código para búsqueda de coincidencias

```
#region frases en un conjunto de palabras
string palabra = "";
List<string> palabras = new List<string>();
var palabrasNoContemplar = dc.TBL_NO_CONTEMPLARS.Select(a => a.PALABRA);

foreach (var item in cadena){
    if (item.ToString() == " "){
        if (palabra != "" & palabra != " "){
            bool existe = false;
            foreach (var itemNoContempla in palabrasNoContemplar){
                if (itemNoContempla == palabra)
                    existe = true;
            }
            if (!existe)
                { palabras.Add(palabra); }
            palabra = "";
        }
    }
    else
        palabra = palabra + item.ToString();
}
if (palabra != "" && palabra != " "){
    bool existe = false;
    foreach (var itemNoContempla in palabrasNoContemplar){
        if (itemNoContempla == palabra)
            existe = true;
    }
    if (!existe)
        palabras.Add(palabra);
    palabra = "";
}

string[] comparacion = palabras.ToArray();
#endregion
```

Figura 31. Código para búsqueda de coincidencias
Elaborado por: Ximena Catota

- f) Comparación para encontrar las posibles coincidencias en la base del conocimiento.
- i) En la base de datos se busca los rangos máximos y mínimos permitidos de acuerdo al número de palabras que conforman la cadena de entrada.
 - ii) Omitiendo signos de puntuación se hace una comparación entre el conjunto de palabras (a) y lo que se tiene en la base del conocimiento.
 - iii) Se obtiene el listado de coincidencias.

Código que encuentra posibles coincidencias en la base del conocimiento.

```
#region Posibles coincidencias
//Encuentra el minimo recomendado dentro de una frase
int minimoRecomendado = 0;

minimoRecomendado = dc.TBL_RANGOS.Single(a => a.RAM_MAX >= palabras.Count() && a.RAN_MIN <= palabras.Count()).RAN_REQUERIDO.Value;

var posiblesSoluciones = from posibleSolucion in dc.TBL_BASE_CONOCIMIENTO.Select(a => a.BAS_SECUENCIAL + " " + a.BAS_DESCRIPCION).ToArray()
    let w = posibleSolucion.Split(new char[] { '.', '?', '!', ' ', ';', ':', ',' },
        StringSplitOptions.RemoveEmptyEntries)
    where w.Distinct().Intersect(comparacion).Count() >= minimoRecomendado
    select posibleSolucion;

#endregion
```

Figura 32. Código que encuentra posibles coincidencias en la base del conocimiento.
Elaborado por: Ximena Catota

4.1.3 Pruebas

Efectuar varios tipos de pruebas sobre el software es de gran importancia para evitar pérdidas económicas y sociales.

Poner en marcha un sistema si haberlo probado antes puede poner en riesgo la reputación de la institución, por esta razón en este apartado se detallara los diferentes tipos de pruebas que se realizaron sobre el software.

4.1.3.1 Pruebas funcionales

Estas pruebas consisten en evaluar el desempeño de las funcionalidades del software, con el objetivo de determinar la correcta funcionalidad de la aplicación.

A continuación se presenta un resumen de las pruebas que se ejecutaron.

Matriz de resultados

Tabla 7. *Matriz de resultados pruebas funcionales*

Código de prueba	Satisfactoria		Observaciones
	Si	No	
PF01	x		Aprobado
PF02	x		Aprobado
PF03	x		Aprobado
PF04	x		Aprobado
PF05	x		Aprobado
PF06	x		Aprobado
PF07	x		Aprobado
PF08	x		Aprobado
PF09	x		Aprobado
PF10	x		Aprobado
PF11	x		Aprobado
PF12	x		Aprobado
PF13	x		La solución se guarda siempre con mayúsculas.
PF14	x		La prueba fue ejecutada únicamente para el área de producción y afectación a la base de datos. Para el tipo de solicitud de incidente añadir un espacio para el ingreso detallado del mismo.
PF15	x		Solo se debería seleccionar el área el estado debe ser únicamente “ingresado”.
PF16	x		Aprobado
PF17	x		Aprobado
PF18	x		Aprobado
PF19	x		Aprobado
PF20	x		Se solicita añadir el envío de correo electrónico para la invitación a pruebas
PF21	x		Aprobado
PF22	x		Se solicita pantallas para el resto de catálogos.
PF23	x		Adición de nuevos reportes en una nueva versión.

Nota. Resumen de pruebas funcionales. Fuente: Autoría.

Elaborado por: Ximena Catota

Como se puede observar en la matriz de resultados se puede decir que la aprobación por parte del cliente sobre la funcionalidad del sistema es de un 75%.

A continuación se detallan las pruebas funcionales que se realizaron de acuerdo a las historias de usuario que se detallaron anteriormente.

Cabe decir que estas pruebas funcionales que se presentan son la última versión puesto que en la metodología de desarrollo XP las pruebas son continuas.

Prueba funcional			
Historia de usuario:	Autenticación de usuario	N° historia:	RF01
ID caso de prueba:	PF01		
Precondiciones:			
<ul style="list-style-type: none"> • Código de usuario: correcto, e incorrecto. • Contraseña: correcta e incorrecta. 			
Descripción:			
<ul style="list-style-type: none"> • Ingreso de datos erróneos para comprobar validación. • Ingreso de datos correctos acceso al sistema según perfil asignado al usuario. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Validación correcta. • Autenticación correcta 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Carga inicial de datos	N° historia:	RF02
ID caso de prueba:	PF02		
Precondiciones:			
<ul style="list-style-type: none"> • No tener data en el sistema de ayuda. • Archivos de carga con la data inicial de usuarios. 			
Descripción:			
Se realizó la migración de datos de los usuarios de la institución llamados también clientes internos.			
Resultado obtenido:			
Data cargada correctamente.			
Estado de la prueba:	satisfactoria		
Observaciones:	Ninguno.		
Responsable ejecución:			

Prueba funcional			
Historia de usuario:	Mantenimiento de personas y usuarios	N° historia:	RF03
ID caso de prueba:	PF03		
Precondiciones:			
<ul style="list-style-type: none"> • Nuevo personal. • Antiguo personal. 			
Descripción:			
<ul style="list-style-type: none"> • Creación de una nueva persona en el sistema; almacenamiento de datos. • Modificación de usuarios existentes en el sistema: actualización de datos. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Creación correcta del usuario. • Modificación correcta del usuario antiguo. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Édison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Cambio de contraseña.	N° historia:	RF04
ID caso de prueba:	PF04		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario existente y activo en el sistema. 			
Descripción:			
Se ejecuta el cambio de contraseña para un usuario específico.			
Resultado obtenido:			
Cambio de contraseña exitoso.			
Estado de la prueba:	satisfactoria		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Olvido de contraseña	N° historia:	RF05
ID caso de prueba:	PF05		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario existente y activo en el sistema. 			
Descripción:			
<ul style="list-style-type: none"> • Ingreso de usuario. • Envío de contraseña al correo del usuario. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Validación correcta de usuario activo. • Envío de contraseña correcto. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Administración de roles	N° historia:	RF06
ID caso de prueba:	PF06		
Precondiciones:			
<ul style="list-style-type: none"> • Listado del menú existente. • Rol nuevo y antiguo. 			
Descripción:			
<ul style="list-style-type: none"> • Creación de un nuevo rol. • Edición de un rol antiguo. • Asignación de accesos al menú para el nuevo rol. • Asignar nuevo rol a un usuario activo. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Datos creados y actualizados correctamente. • Comprobación de accesos correcta. 			
Estado de la prueba:	satisfactoria		
Observaciones o errores asociados:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Categoría de solicitud – detalle solicitud (todas las áreas).	N° historia:	RF07, RF08, RF09, RF11
ID caso de prueba:	PF07		
Precondiciones:			
<ul style="list-style-type: none"> • Código de usuario activo con rol de cliente interno. • Caso de incidente. • Caso de solicitud. • Caso afectación a la base de datos. 			
Descripción:			
<ul style="list-style-type: none"> • Ingreso al sistema con el usuario con rol de cliente interno. • Ingreso a la opción proceso de solicitud > ingresar solicitud. • Caso uno incidente: selección de radio button opción incidente: ingreso y almacenamiento de datos. • Caso dos solicitud: selección de radio button opción requerimiento: ingreso y almacenamiento de datos. • Caso tres solicitudes con área afectación a la base de datos: ingreso y almacenamiento de datos. • Envío de correo electrónico para aprobación de jefaturas. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Ingreso y almacenamiento correcto de los tres tipo de solicitud. • Se comprueba que hay diferencias solicitadas en cada tipo de solicitud. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Aprobación de solicitud requerimiento.	N° historia:	RF10
ID caso de prueba:	PF08		
Precondiciones:			
<ul style="list-style-type: none"> • Solicitudes ingresadas: (todas las áreas: desarrollo, producción, bdd) • Usuarios con rol de jefaturas ingresados en las solicitudes. 			
Descripción:			
<ul style="list-style-type: none"> • Con los usuarios encargados de la aprobación de solicitudes se ingresó a la opción proceso solicitud > aprobar solicitud. • Filtro de área (desarrollo, producción, bdd). • Clic en activar el campo firmar. 			

<ul style="list-style-type: none"> • Almacenar información. • Envío de correo al coordinador correspondiente del área 	
Resultado obtenido: <ul style="list-style-type: none"> • Filtrado de requerimientos según su estado por área correcto. • Firma mediante la activación del campo firmar correcto. • Envío de correo correcto. 	
Estado de la prueba:	satisfactoria
Observaciones:	Ajuste del estado solo ingresado.
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).

Prueba funcional			
Historia de usuario:	Asignar incidente	N° historia:	RF11
ID caso de prueba:	PF09		
Precondiciones: <ul style="list-style-type: none"> • Incidente ingresado, casos para cada área (producción, desarrollo, afectación bdd). • Usuario coordinador de área. 			
Descripción: <ul style="list-style-type: none"> • Se selecciona el incidente para ser asignado. (evidencia de despliegue de información general del incidente). • Se selecciona del listado de recursos un técnico para asignar el incidente. • Se asigna prioridad e impacto. 			
Resultado obtenido: <ul style="list-style-type: none"> • Asignación de recurso, nivel de impacto y prioridad correcta. • Validación de campos obligatorios. • Envío de correo correcto. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Gestión de incidente – cierre - devolución	N° historia:	RF12
ID caso de prueba:	PF10		
Precondiciones:			
<ul style="list-style-type: none"> • Incidentes en estado asignado. • Usuarios asignados para la solución. (caso de cada área). 			
Descripción:			
<ul style="list-style-type: none"> • Se selecciona el incidente pendiente de cierre. (evidencia de despliegue de información general del incidente). • Buscar solución no existe coincidencias • Ingresar solución. Evidencia de alimentación automática de la base del conocimiento. <p>Caso dos devoluciones del incidente. Se ingresó la solución como error operativo y se devolvió el incidente.</p>			
Resultado obtenido:			
<ul style="list-style-type: none"> • Cierre correcto del incidente. • Alimentación de la base del conocimiento. • Validación de campos obligatorios. • Envío de correo electrónico alertando calificar la solución al usuario solicitante. • En el caso dos se devolvió fin del ciclo satisfactorio. Con envío de correo al usuario solicitante. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Evaluación de incidente	N° historia:	RF14
ID caso de prueba:	PF11		
Precondiciones:			
<ul style="list-style-type: none"> • Incidentes en estado cerrado. • Usuarios solicitantes. (caso de cada área). 			
Descripción:			
<ul style="list-style-type: none"> • Se selecciona el incidente pendiente de calificar en la opción gestión de incidentes calificar incidente. • Asignar calificación, validaciones correctas. • La calificación fue negativa por lo tanto el incidente fue reabierto. • Se verifico que si el usuario no califica los incidentes pendientes no puede ingresar ningún tipo de solicitudes. 			
Resultado obtenido:			

<ul style="list-style-type: none"> • Almacenamiento de la calificación correcta. • Validaciones de coherencia de respuestas correcta. • Envío de correcto de correo electrónico al usuario correcto • Verificación de calificación obligatoria correcta. 	
Estado de la prueba:	Satisfactoria.
Observaciones:	Ninguno.
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).

Prueba funcional			
Historia de usuario:	Reasignación de incidentes	N° historia:	RF15
ID caso de prueba:	PF12		
Precondiciones:			
<ul style="list-style-type: none"> • Incidentes en estado asignado. • Usuarios con acceso a la opción reasignar (todos los técnicos). 			
Descripción:			
<ul style="list-style-type: none"> • Se selecciona el incidente ya asignado. (evidencia de despliegue de información general del incidente). • Se asigna a otro técnico y se almacena la información. • Se verifico que todos los técnicos tengan acceso a esta opción. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Reasignación correcta • Envío de correcto de correo electrónico alertando calificar la solución al usuario solicitante. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguno.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Gestión del conocimiento	N° historia:	RF15 RF16 RF17
ID caso de prueba:	PF13		
Precondiciones:			
<ul style="list-style-type: none"> • Incidentes en estado asignado. • Usuarios con acceso a la pantalla seguimiento de incidente. 			
Descripción:			
<ul style="list-style-type: none"> • Se selecciona el incidente ya asignado. • Se ingresa la solución del incidente. • Clic en guardar. • Verificar en la pantalla de consulta a la base del conocimiento nuevas coincidencias relacionadas al problema que se dio solución 			
Resultado obtenido:			

<ul style="list-style-type: none"> • La solución fue almacenada automáticamente en la base del conocimiento. • Se verifico el funcionamiento de la pantalla de consultas a la base del conocimiento. 	
Estado de la prueba:	Satisfactoria.
Observaciones:	La solución se guarda siempre con mayúsculas.
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).

Prueba funcional			
Historia de usuario:	Ingresar solicitud	N° historia:	RF19,RF08,RF09,RF07
ID caso de prueba:	PF14		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con acceso a la funcionalidad ingreso de solicitud. • Datos para el ingreso de solicitud. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la opción requerimiento (área producción). • Ingreso de la información solicitada. • Ingreso de varias tareas. • Guardar la solicitud. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Almacenamiento exitoso. • Generación del número de solicitud de acuerdo al área que corresponde. • Se verifico el correcto funcionamiento de los tipos de solicitud. • Se verifico el funcionamiento de adición de varias tareas en el tipo de solicitud requerimiento. • Se verifico la información inicial que contiene la pantalla de ingreso de solicitud (cabecera). 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	La prueba fue ejecutada únicamente para el área de producción y afectación a la base de datos. Para el tipo de solicitud de incidente añadir un espacio para el ingreso detallado del mismo.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Aprobación de solicitud - requerimiento	N° historia:	RF10
ID caso de prueba:	PF15		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con acceso a la funcionalidad aprobación de solicitud. • Solicitudes ingresadas por aprobar. 			
Descripción:			
<ul style="list-style-type: none"> • Selección del área correspondiente. • Selección de la etapa. • Despliegue del listado pendiente de solicitudes por aprobar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Aprobación exitosa. • Envío exitoso de correo al coordinador del área correspondiente después de la aprobación de las jefaturas. • Cambio de etapa correctamente. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Solo se debería seleccionar el área el estado debe ser únicamente “ingresado”.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Asignar solicitud - requerimiento	N° historia:	RF20
ID caso de prueba:	PF16		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con acceso a la funcionalidad asignación de solicitud (coordinador del área correspondiente). • Solicitudes en estado aprobado. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser asignada. • Asignación de recursos por tarea (detalle de solicitud). • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Asignación exitosa. • Envío exitoso de correo al recurso asignado • Cambio de etapa correctamente. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguna.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Asignar solicitud - requerimiento	N° historia:	RF20
ID caso de prueba:	PF17		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con acceso a la funcionalidad asignación de solicitud (coordinador del área correspondiente). • Solicitudes en estado aprobado. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser asignada. • Asignación de recursos por tarea (detalle de solicitud). • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Asignación exitosa. • Visualización de información detallada de la solicitud. • Envío exitoso de correo al recurso asignado • Cambio de etapa correctamente. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguna.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Levantamiento de especificaciones.	N° historia:	RF21
ID caso de prueba:	PF18		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con rol de técnico cualquier área en este caso (técnico de producción). • Solicitudes en estado “asignado”. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser gestionada. • Ingreso del nombre y detalle de la especificación correspondiente. • Finalizar etapa. • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Proceso de levantamiento de n especificaciones exitoso. • Visualización de información detallada de la solicitud. • Cambio de etapa correctamente. • Se verifico que mientras no se finaliza de etapa el estado de la solicitud no cambia. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguna.		

Responsable ejecución:	Edison Logaña (coordinador de desarrollo).
-------------------------------	--

Prueba funcional			
Historia de usuario:	Análisis de solicitud.	N° historia:	RF22
ID caso de prueba:	PF19		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con rol de técnico cualquier área en este caso (técnico de producción). • Solicitudes en estado “levantamiento de especificaciones”. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser gestionada. • Ingreso del análisis realizado. • Finalizar etapa. • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Proceso de análisis exitoso. • Visualización de información detallada de la solicitud. • Cambio de etapa correctamente. • Se verifico que mientras no se finaliza de etapa el estado de la solicitud no cambia. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguna.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Desarrollo de solicitud.	N° historia:	RF23
ID caso de prueba:	PF20		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con rol de técnico cualquier área en este caso (técnico de producción). • Solicitudes en estado “análisis”. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser gestionada. • Ingreso del avance realizado y el tiempo que duro (varios ingresos). • Finalizar etapa. • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Proceso de análisis exitoso. • Visualización de información detallada de la solicitud. • Cambio de etapa correctamente. • Se verifico que mientras no se finaliza de etapa el estado de la solicitud no cambia. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Se solicita añadir el envío de correo electrónico		

	para la invitación a pruebas
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).

Prueba funcional			
Historia de usuario:	Pruebas de verificación.	N° historia:	RF24
ID caso de prueba:	PF21		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con rol de técnico cualquier área en este caso (técnico de producción). • Solicitudes en estado “desarrollo”. 			
Descripción:			
<ul style="list-style-type: none"> • Selección de la solicitud a ser gestionada. • Registrar detalle de las pruebas realizadas (varios ingresos). • Finalizar etapa. • Añadir firmantes (asistentes). • Clic en guardar. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Proceso de análisis exitoso. • Visualización de información detallada de la solicitud. • Cambio de etapa correctamente. • Se verifico que mientras no se finaliza de etapa el estado de la solicitud no cambia. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Ninguna.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

Prueba funcional			
Historia de usuario:	Pruebas de parametrización.	N° historia:	RF25, RF26
ID caso de prueba:	PF22		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con rol de administrador. 			
Descripción:			
<ul style="list-style-type: none"> • Se verifico el correcto funcionamiento para el mantenimiento de los siguientes datos que están como catálogos: <ul style="list-style-type: none"> - Agencias. - Departamentos. - Aplicativos. - Módulos. - Base del conocimiento. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Parametrización correcta 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Se solicita pantallas para el resto de catálogos.		

Responsable ejecución:	Edison Logaña (coordinador de desarrollo).
-------------------------------	--

Prueba funcional			
Historia de usuario:	Pruebas de parametrización.	N° historia:	RF27, RF28
ID caso de prueba:	PF23		
Precondiciones:			
<ul style="list-style-type: none"> • Usuario con accesos a reportes. • Información sobre la gestión de incidentes y gestión de requerimientos. 			
Descripción:			
<p>Se revisaron los siguientes reportes:</p> <ul style="list-style-type: none"> - Solicitudes en gestión todos los usuarios. - Solicitudes por usuario. - Reporte de calificaciones en un periodo de tiempo. - Reporte de catálogos de aplicaciones. - Reporte de catálogos de módulos por aplicación. - Reportes estadísticos del promedio de calificaciones obtenidas en un periodo de tiempo. - Reporte estadístico de un promedio de incidentes resueltos en un periodo de tiempo. - Reporte estadístico del promedio de solicitudes gestionadas en un rango de fechas. 			
Resultado obtenido:			
<ul style="list-style-type: none"> • Generación de reportes correcta. • Los reportes tienen la facilidad de ser exportados. 			
Estado de la prueba:	Satisfactoria.		
Observaciones:	Adición de nuevos reportes en una nueva versión.		
Responsable ejecución:	Edison Logaña (coordinador de desarrollo).		

4.1.3.2 Pruebas de stress

Una prueba de stress obliga a un sistema informático al máximo punto, para medir el rendimiento, la capacidad y las condiciones bajo las cuales responde de manera óptima a las solicitudes de los usuarios. Es importante mencionar que el rendimiento está íntimamente ligado al hardware de la computadora sobre la cual se está ejecutando.

Software utilizado para las pruebas.

Para las pruebas de stress se utilizó el software especializado “Webserver Load Performance Stress Test”, aplicación diseñada para probar el desempeño de una aplicación. Esta aplicación así como también el detalle de las pruebas realizadas sobre el sistema se describen en el **Anexo 1**.

Para las pruebas se tomó en consideración 3 parámetros, el número de usuarios concurrentes que en este caso se lo hizo con 200 que soportará el sistema, el tiempo promedio de respuesta y el porcentaje de error de cada uno de los casos.

Resultados

A continuación se presenta una tabla de los resultados aproximados que se obtuvieron en las pruebas de stress realizadas con el software anteriormente mencionado.

Tabla 8. *Resumen de pruebas de stress*

Número de usuarios	Tiempo de respuesta [ms]		Taza de error
	Mínimo	Máximo	
200	19 ms	40ms	0%

Nota. Ms=milisegundos. Fuente: Autoría.

Elaborado por: Ximena Catota.

En la tabla anterior se puede evidenciar que el software soporta satisfactoriamente la concurrencia de hasta 200 usuarios con un tiempo de respuesta mínimo de 19ms y

máximo de 40ms con una tasa de error del 0%, por lo cual se puede decir que el software está dentro del rango aceptable.

CONCLUSIONES

- El sistema web de mesa de ayuda desarrollado con Silverlight permite gestionar con una interfaz agradable al usuario, los trabajos de help desk mediante una mejor administración de incidentes, tareas y requerimientos gracias a las recomendaciones que ITIL propone en los apartados que fueron útiles para la construcción de este sistema.
- Mediante el sistema informático construido se contrarrestaron las debilidades que tenía el departamento de tecnología de Bancodesarrollo en cuanto al proceso de los trabajos de help desk, proponiendo la adecuación en el sistema de algunos lineamientos que ITIL V3 propone en sus apartados.
- El sistema informático lleva un control de acceso de acuerdo al rol de cada usuario para el manejo de la información relacionada con el proceso de incidencias, tareas y/o requerimientos.
- El sistema informático provee al usuario de una interfaz en la que pueda monitorear el estado de las incidencias, tareas y/o requerimientos siempre y cuando tenga acceso a esta información es decir de acuerdo al rol que posea.
- En el análisis de los procesos que se llevaban a cabo antes de la construcción del sistema se pudo detectar que muchos tardaban demasiado en su ejecución, tales como: la discriminación de las incidencias, priorización de solicitudes, asignación de los recursos. Y en otras ocasiones no tener una adecuada organización de la información extendía el tiempo en la atención que se brindaba al proceso de solución de los incidentes, tarea y o/requerimientos. Gracias a la implementación hecha en el sistema basada en ITIL este tiempo

disminuyo considerablemente puesto que al existir un solo medio de concentración de las solicitudes y un mecanismo informático que toma en cuenta las buenas practicas no se pierde el tiempo en tareas poco productivas o mal administradas.

- El uso de la metodología XP para el desarrollo de una aplicación es adecuada cuando el cliente no tiene claro lo que necesita pero se corre el riesgo de alargar el tiempo de finalización del proyecto por la naturaleza de la metodología que se basa en los cambios continuos hasta la conformidad del usuario final.
- ITIL es un conjunto de buenas prácticas que pueden ahorrar a las organizaciones tiempo y dinero. Además el uso adecuado de estas da a las instituciones un nivel de calidad superior en sus servicios.

RECOMENDACIONES

- Se recomienda incluir en una nueva versión del sistema la posibilidad de evaluar el desempeño de los técnicos que atienden los requerimientos ya que actualmente el sistema cuenta con esta opción únicamente para las incidencias.
- Como en todo sistema nuevo existe una resistencia al cambio por parte de los usuarios por lo cual se recomienda una capacitación del sistema para que los usuarios lo puedan usar sin ningún inconveniente.
- Se sugiere el uso de las recomendaciones que da ITIL en todos los procesos que lleva a cabo el departamento de tecnología para la mejora de sus servicios.
- Es recomendable realizar varios tipos de pruebas a las aplicaciones de software para prever a tiempo cualquier deficiencia en el sistema y entregar un producto de calidad al cliente.
- Para el desarrollo de software se recomienda el uso de herramientas que intentan unir el diseño y el desarrollo. Tal como es el caso de aplicaciones Silverlight en las que se recomienda el uso de expression blend para facilitar el trabajo del desarrollador en cuanto a la capa de presentación.
- Si no se tiene experiencia con el desarrollo en Silverlight se recomienda instalar visual studio 2010 e instalar Silverlight 4 tools para visual studio 2010 de manera que al crear un nuevo proyecto Silverlight se podrá visualizar herramientas y ventanas familiares a las que usa visual studio. Además Silverlight tools incluye mensajes de error útiles para depurar.

LISTA DE REFERENCIAS

- .microsoft.com. (2015). microsoft. Recuperado el 02 de 01 de 2015, de <https://msdn.microsoft.com/en-us/library/system.servicemodel.servicecontractattribute%28v=vs.110%29.aspx>
- agile-ux.com. (2014).
- Albaladejo, X. (01 de enero de 2012). proyectosagiles. Recuperado el 02 de febrero de 2013, de proyectosagiles: <http://www.proyectosagiles.org/lista-requisitos-priorizada-product-backlog>
- B-able. (2013). <http://es.slideshare.net/Biable/manual-itil-integro>.
- bancodesarrollo.fin.ec. (2015). <http://www.bancodesarrollo.fin.ec/>.
- Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley.
- Beck, K. A. (2004). Extreme Programming Explained. Addison Wesley.
- blogams.com. (2010). Linq to Sql parte 1.
- Calle, F. (2013). slideshare. Recuperado el 20 de junio de 2013, de <http://www.slideshare.net/Decimo/arquitectura-3-capas>
- Fowler, M. (2006). Pair Programming Misconceptions. martinowler.com.
- itilv3.osiatis.es. (2015). ITIL Foundation.
- Marlon, M. (2006). Fundamentos de ITIL. New Horizons.

- msdn.microsoft. (2015).
- msdn.microsoft. (2015). Silverlight.
- msdn.microsoft.com. (2015). <https://msdn.microsoft.com/es-es/library/ms731082%28v=vs.110%29.aspx>.
- msdn.microsoft.com. (2015). LINQ.
- paessler.com. (2015). <http://www.paessler.com/tools/webstress>.
- Pialorsi, P., & Russo, M. (2009). Programación LINQ. Anaya Multimedia.
- procesosdesoftware. (2013). Recuperado el 20 de junio de 2013, de <http://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>
- S.Pressman, R. (2002). Ingeniería del software un enfoque práctico. McGraw-hill/interamericana de españa s.a.
- Shaw, M., & Garlan , D. (1996). Arquitectura de software. Prentice-Hall.
- slideshare.net. (2015).
- slideshare.net. (2015). Introduccion de Silverlight.
- Vidales, & Falagán, M. (2012). Microsoft Silverlight en acción. Madrid: Grupo RC, Madrid - España.
- Vidales, M. F. (2012). Microsoft Silverlight en acción. Grupo RC, Madrid - España.

ANEXOS

Anexo 1. Pruebas de stress

Para realizar las pruebas de carga del sistema mesa de ayuda se utilizó la herramienta **Websver Stress Tool**.

Websver Stress Tool: es una aplicación de prueba HTTP y cliente / servidor de gran alcance diseñado para identificar problemas críticos de rendimiento en su sitio web o servidor web que puede impedir la experiencia óptima para los visitantes del sitio. Mediante la simulación de las peticiones HTTP generadas por cientos o incluso miles de usuarios simultáneos, puede probar el rendimiento de su servidor web bajo cargas normales y excesivos para garantizar que la información y los servicios críticos están disponibles en velocidades de sus usuarios finales esperan. (paessler.com, 2015)

Uso:

Seleccionar el tipo de prueba y el número de usuarios con el que se va a ejecutar la simulación.

Para comprobar la carga del servidor escogemos la opción RAMP.

Selección del tipo de prueba Webserver stress tool.

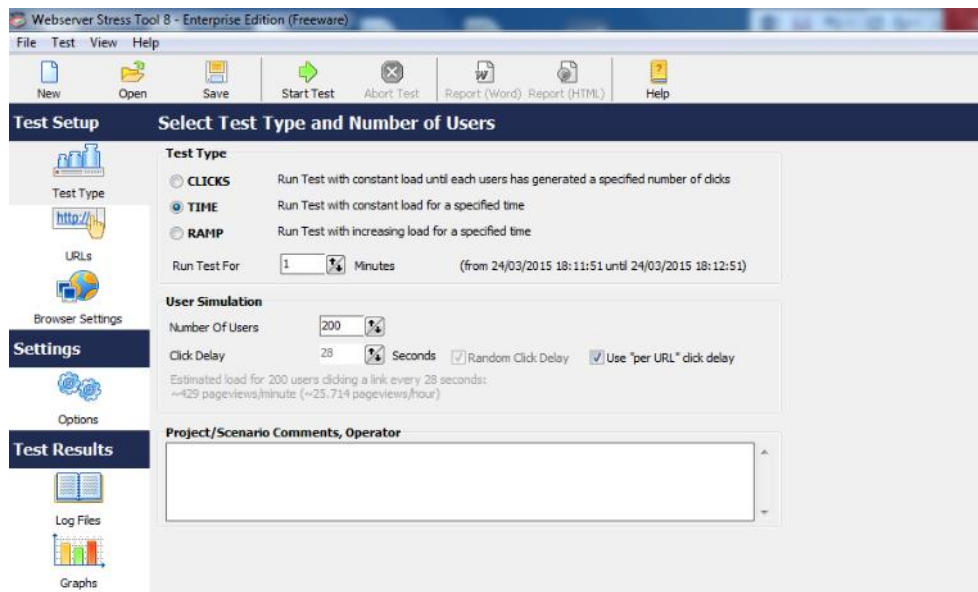


Figura. Selección del tipo de prueba Webserver stress tool.
Elaborado por: Ximena Catota.

Dar clic sobre la opción URL para ingresar la dirección del sistema.

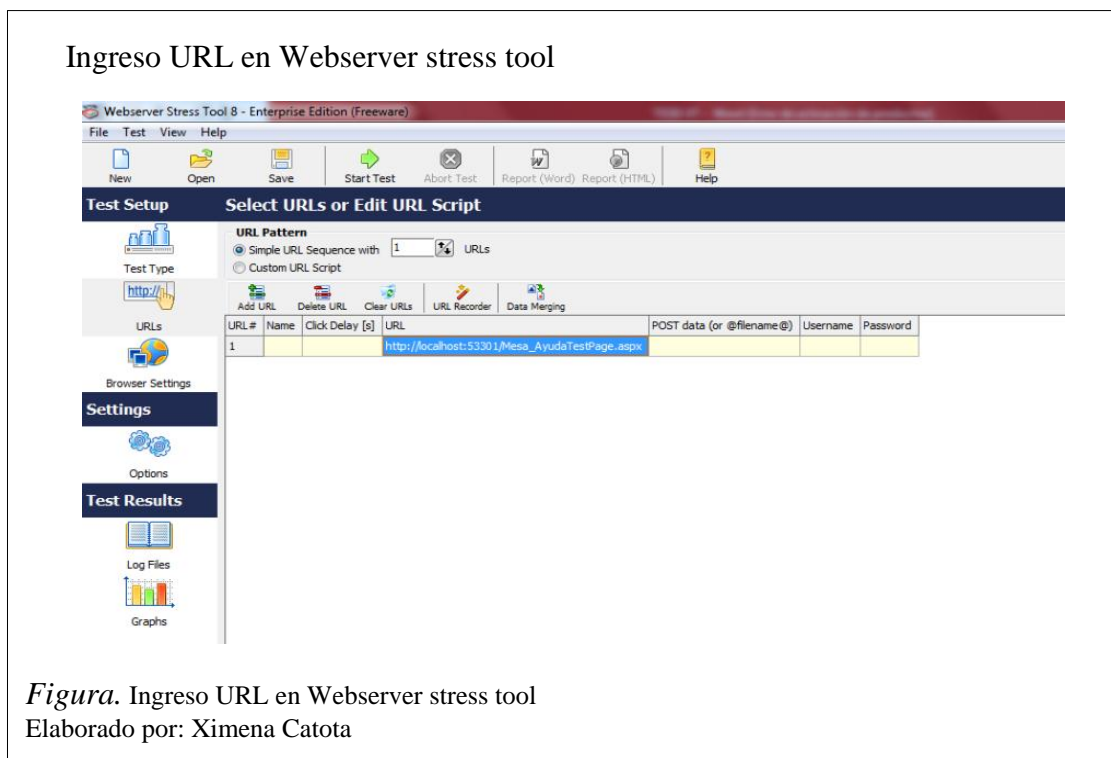


Figura. Ingreso URL en Webserver stress tool
Elaborado por: Ximena Catota

Clic en **start test** para iniciar con la prueba.

Prueba de carga del sistema

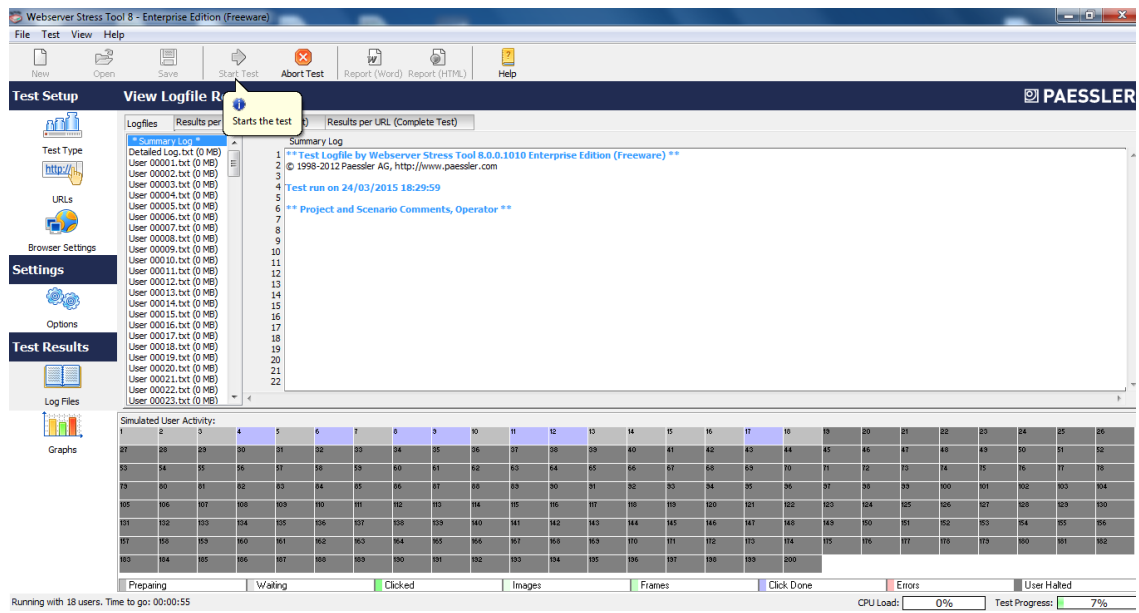
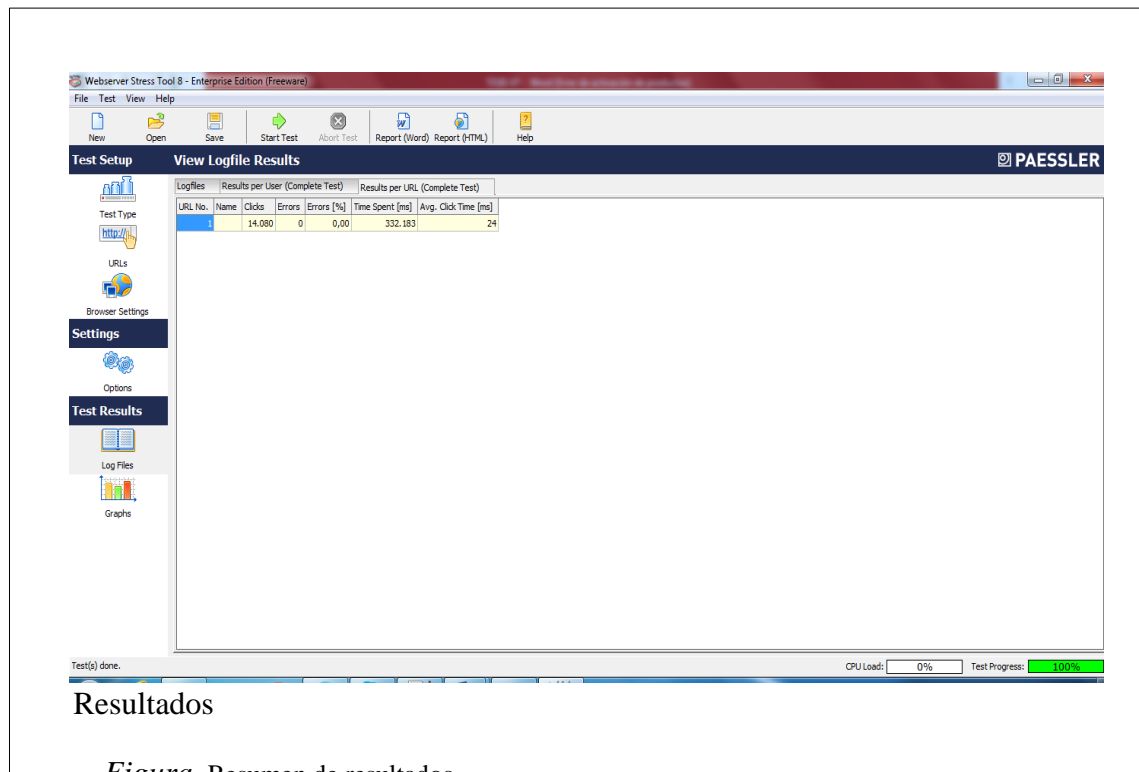


Figura. Prueba de carga del sistema
Elaborado por: Ximena Catota

Resumen de resultados



Resultados

Figura. Resumen de resultados
Elaborado por: Ximena Catota

Resultados por usuario

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	183	183	0		22	626.958	1.255,58
2	172	172	0		23	589.272	1.204,32
3	185	185	0		21	633.810	1.280,70
4	178	178	0		22	609.828	1.263,44
5	176	176	0		23	602.976	1.202,60
6	167	167	0		23	572.142	1.188,82
7	176	176	0		25	602.976	1.114,71
8	174	174	0		23	596.124	1.184,11
9	172	172	0		22	589.272	1.240,58
10	157	157	0		23	537.882	1.189,74
11	177	177	0		21	606.402	1.306,00
12	164	164	0		22	561.864	1.230,54
13	163	163	0		22	558.438	1.223,98
14	147	147	0		21	503.622	1.308,73
15	164	164	0		23	561.864	1.195,04
16	154	154	0		23	527.604	1.213,55
17	152	152	0		22	520.752	1.260,55
18	145	145	0		21	496.770	1.298,93
19	155	155	0		22	531.030	1.228,73
20	147	147	0		21	503.622	1.311,68
21	144	144	0		25	493.344	1.081,39
22	136	136	0		23	465.936	1.173,35
23	146	146	0		21	500.196	1.276,40
24	138	138	0		24	472.788	1.139,70
25	141	141	0		24	483.066	1.155,94
26	137	137	0		23	469.362	1.168,54
27	139	139	0		21	476.214	1.322,57
28	140	140	0		19	479.640	1.419,74
29	132	132	0		22	452.232	1.265,39
30	129	129	0		22	441.954	1.271,78
31	138	138	0		23	472.788	1.207,52
32	126	126	0		18	431.676	1.495,57
33	131	131	0		22	448.806	1.264,65
34	116	116	0		26	397.416	1.071,31
35	130	130	0		24	445.380	1.150,74
36	120	120	0		20	411.120	1.404,57
37	117	117	0		20	400.842	1.353,22
38	110	110	0		27	376.860	1.001,55
39	123	123	0		22	421.398	1.244,20
40	116	116	0		24	397.416	1.135,93
41	108	108	0		24	370.008	1.161,12
42	109	109	0		22	373.434	1.237,10
43	116	116	0		22	397.416	1.220,76
44	110	110	0		24	376.860	1.139,67
45	111	111	0		25	380.286	1.084,28
46	103	103	0		22	352.878	1.262,38
47	115	115	0		21	393.990	1.333,60
48	102	102	0		21	349.452	1.283,81
49	103	103	0		23	352.878	1.202,51
50	96	96	0		24	328.896	1.153,62
51	106	106	0		22	363.156	1.221,02
52	104	104	0		22	356.304	1.265,50
53	104	104	0		21	356.304	1.304,61
54	92	92	0		24	315.192	1.150,15
55	106	106	0		21	363.156	1.286,59
56	94	94	0		23	322.044	1.184,54

57	94	94	0	25	322.044	1.091,81
58	85	85	0	26	291.210	1.066,90
59	97	97	0	25	332.322	1.088,97
60	86	86	0	27	294.636	1.020,54
61	92	92	0	21	315.192	1.310,80
62	86	86	0	27	294.636	1.014,60
63	94	94	0	26	322.044	1.064,89
64	80	80	0	29	274.080	932,11
65	85	85	0	27	291.210	1.024,83
66	76	76	0	31	260.376	877,14
67	88	88	0	24	301.488	1.148,60
68	76	76	0	29	260.376	946,74
69	81	81	0	29	277.506	948,58
70	79	79	0	25	270.654	1.098,57
71	85	85	0	20	291.210	1.352,88
72	72	72	0	26	246.672	1.066,36
73	79	79	0	26	270.654	1.068,52
74	71	71	0	28	243.246	975,12
75	79	79	0	26	270.654	1.068,61
76	66	66	0	30	226.116	903,75
77	75	75	0	24	256.950	1.128,00
78	66	66	0	26	226.116	1.043,67
79	78	78	0	26	267.228	1.041,99
80	65	65	0	33	222.690	827,52
81	71	71	0	27	243.246	1.033,65
82	65	65	0	27	222.690	1.022,56
83	75	75	0	26	256.950	1.063,53
84	66	66	0	26	226.116	1.059,70
85	65	65	0	24	222.690	1.145,87
86	61	61	0	33	208.986	836,06
87	73	73	0	24	250.098	1.130,48
88	64	64	0	27	219.264	1.024,91
89	64	64	0	26	219.264	1.069,58
90	59	59	0	33	202.134	825,16
91	70	70	0	23	239.820	1.176,12
92	57	57	0	31	195.282	892,62
93	61	61	0	26	208.986	1.041,33
94	59	59	0	26	202.134	1.051,53
95	68	68	0	29	232.968	954,49
96	58	58	0	23	198.708	1.209,48
97	62	62	0	27	212.412	1.028,62
98	55	55	0	33	188.430	838,38
99	66	66	0	21	226.116	1.278,72
100	57	57	0	24	195.282	1.158,75
101	60	60	0	28	205.560	965,66
102	52	52	0	31	178.152	873,80
103	60	60	0	28	205.560	976,46
104	52	52	0	29	178.152	949,70
105	61	61	0	26	208.986	1.052,25
106	51	51	0	25	174.726	1.076,74
107	62	62	0	24	212.412	1.118,81
108	48	48	0	27	164.448	1.019,39
109	57	57	0	28	195.282	989,46
110	51	51	0	28	174.726	977,04
111	60	60	0	27	205.560	1.019,18
112	49	49	0	33	167.874	823,62
113	57	57	0	30	195.282	913,47
114	50	50	0	28	171.300	993,15
115	60	60	0	21	205.560	1.336,27

116	47	47	0	29	161.022	958,44
117	54	54	0	25	185.004	1.097,91
118	50	50	0	28	171.300	974,53
119	54	54	0	22	185.004	1.244,41
120	44	44	0	20	150.744	1.394,45
121	53	53	0	21	181.578	1.290,89
122	47	47	0	24	161.022	1.142,42
123	55	55	0	23	188.430	1.212,91
124	44	44	0	30	150.744	901,01
125	55	55	0	25	188.430	1.098,53
126	45	45	0	26	154.170	1.056,70
127	52	52	0	22	178.152	1.221,21
128	44	44	0	28	150.744	962,51
129	48	48	0	30	164.448	902,38
130	43	43	0	32	147.318	848,52
131	45	45	0	22	154.170	1.258,71
132	43	43	0	28	147.318	974,61
133	46	46	0	32	157.596	857,45
134	42	42	0	40	143.892	679,86
135	46	46	0	27	157.596	1.029,98
136	38	38	0	33	130.188	833,84
137	44	44	0	26	150.744	1.053,52
138	39	39	0	28	133.614	971,22
139	48	48	0	25	164.448	1.079,65
140	41	41	0	25	140.466	1.105,88
141	41	41	0	38	140.466	727,53
142	41	41	0	22	140.466	1.225,65
143	46	46	0	24	157.596	1.144,07
144	38	38	0	21	130.188	1.291,95
145	41	41	0	27	140.466	1.031,32
146	38	38	0	28	130.188	976,76
147	44	44	0	24	150.744	1.159,03
148	33	33	0	25	113.058	1.101,34
149	37	37	0	32	126.762	851,38
150	36	36	0	33	123.336	834,17
151	40	40	0	26	137.040	1.034,36
152	33	33	0	23	113.058	1.185,00
153	37	37	0	17	126.762	1.590,43
154	36	36	0	31	123.336	892,29
155	36	36	0	27	123.336	1.020,34
156	32	32	0	26	109.632	1.050,66
157	34	34	0	20	116.484	1.368,10
158	34	34	0	31	116.484	885,50
159	34	34	0	17	116.484	1.660,36
160	29	29	0	30	99.354	904,12
161	32	32	0	20	109.632	1.387,60
162	33	33	0	31	113.058	892,16
163	35	35	0	25	119.910	1.101,72
164	28	28	0	27	95.928	1.017,14
165	30	30	0	20	102.780	1.393,17
166	32	32	0	25	109.632	1.104,60
167	31	31	0	18	106.206	1.517,12
168	28	28	0	21	95.928	1.335,17
169	27	27	0	19	92.502	1.452,13
170	30	30	0	20	102.780	1.380,97
171	29	29	0	20	99.354	1.369,29
172	27	27	0	26	92.502	1.062,93
173	26	26	0	21	89.076	1.335,33
174	29	29	0	16	99.354	1.668,02

175	26	26	0	26	89.076	1.040,89
176	24	24	0	26	82.224	1.074,61
177	24	24	0	24	82.224	1.139,48
178	25	25	0	18	85.650	1.503,27
179	22	22	0	19	75.372	1.472,72
180	24	24	0	35	82.224	794,37
181	23	23	0	36	78.798	766,31
182	23	23	0	32	78.798	862,03
183	24	24	0	21	82.224	1.318,76
184	22	22	0	19	75.372	1.441,03
185	20	20	0	19	68.520	1.452,55
186	21	21	0	20	71.946	1.372,63
187	20	20	0	22	68.520	1.239,78
188	20	20	0	32	68.520	844,79
189	20	20	0	21	68.520	1.314,17
190	19	19	0	28	65.094	963,64
191	19	19	0	50	65.094	552,09
192	18	18	0	13	61.668	2.032,73
193	19	19	0	28	65.094	971,89
194	17	17	0	16	58.242	1.673,60
195	17	17	0	22	58.242	1.259,38
196	17	17	0	12	58.242	2.203,25
197	15	15	0	25	51.390	1.084,96
198	16	16	0	44	54.816	623,05
199	14	14	0	39	47.964	703,45
200	15	15	0	21	51.390	1.279,92



Gráfica de resultados de las pruebas de stress

Figura. Gráfica de resultados de las pruebas de stress
Elaborado por: Ximena Catota.

Anexo 2. Implementación del sistema

Requerimientos:

- Instalación de framework 4.5. (Para el uso de WCF y Silverlight).
- Internet Information Services de Windows correctamente configurado.
- Permisos de administración de Internet Information Services IIS.
- Microsoft SQL Server 2012.

Procedimiento:

1. Abrir el programa SQL Server Management Studio.
2. Conexión al motor de base de datos.

SQL Server 2012



Figura. SQL Server 2012
Elaborado por: Ximena Catota

3. Crear una base de datos con el nombre “MesaAyuda”.

Creación de la base de datos para el sistema

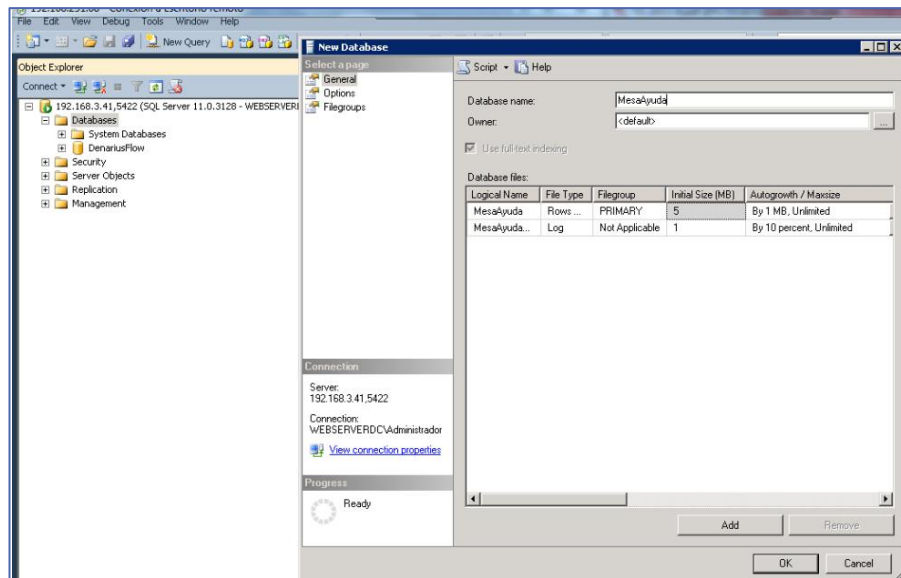


Figura. Creación de la base de datos para el sistema

Elaborado por: Ximena Catota

4. Restaurar la base de datos que se creó para la construcción del sistema.
5. Copiar al en el directorio `c:\inetpub\wwwroot` el proyecto tanto la parte del cliente y la parte del servidor.

Directorio inetpub

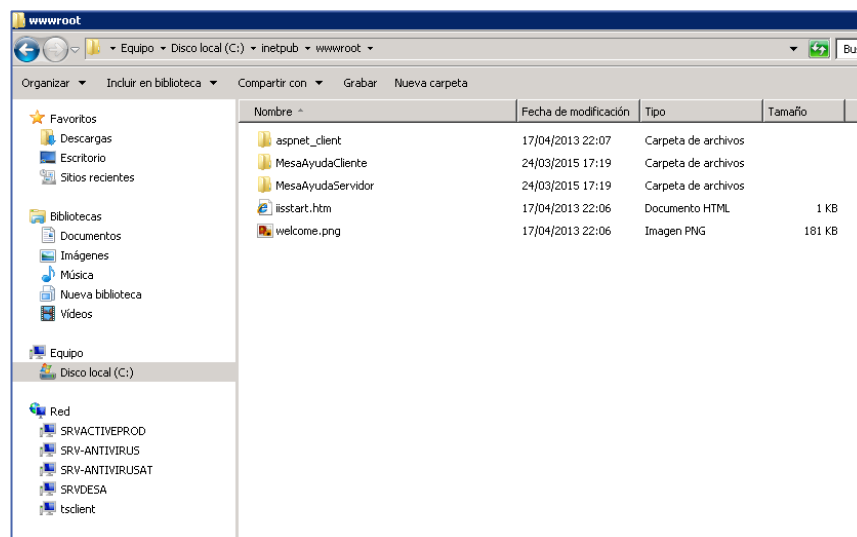
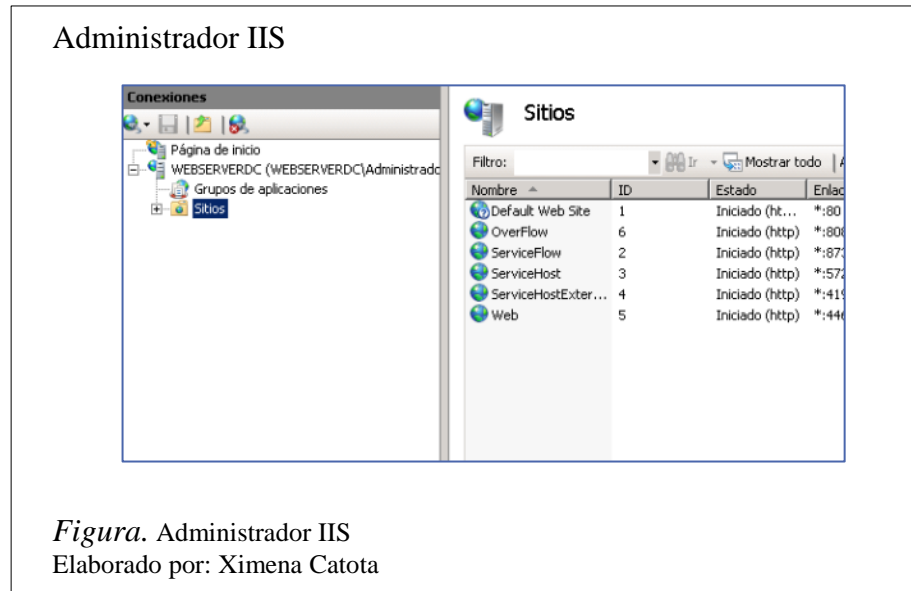


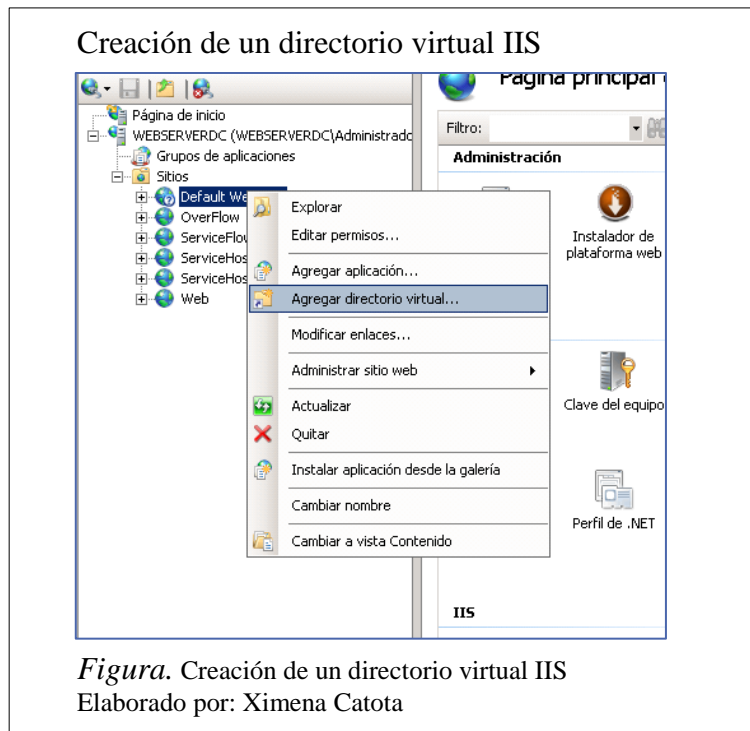
Figura. Directorio inetpub

Elaborado por: Ximena Catota

6. Crear un directorio virtual (IIS) mediante la interfaz de usuario que proporciona el administrador de IIS.
 - i) Abrir el administrador IIS.
 - ii) En la opción de conexiones se expande el nodo llamado sitios.



- iii) Clic en el sitio en el que se desea crear el directorio virtual.
- iv) En el sitio escogido clic derecho agregar directorio virtual.



- v) Asignar un alias para tener acceso a la dirección Uniform Resource Locator (URL).
- vi) En la opción ruta de acceso física, escribir la ruta de acceso física de la carpeta de contenido.



vii) Clic en aceptar.

7. Listo ya se puede usar el sistema mesa de ayuda.

Ingreso al sistema de mesa de ayuda



Figura. Ingreso al sistema de mesa de ayuda
Elaborado por: Ximena Catota