

**DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE
REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS**



**FACULTAD DE INGENIERIA
ESCUELA DE SISTEMAS**

**DISERTACION PREVIA A LA OBTENCION DEL TÍTULO DE
*INGENIERO EN SISTEMAS Y COMPUTACIÓN***

TEMA:

**“DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE
REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS EN UNA
EMPRESA DE DESARROLLO DE SOFTWARE”**

NOMBRE

GÁLVEZ POZO SANTIAGO JAVIER

DIRECTOR:

DE LA CRUZ DOMINGUEZ FABIAN

QUITO – 2017

DEDICATORIA Y AGRADECIMIENTO

RESUMEN

ÍNDICE DE CONTENIDOS

Dedicatoria y Agradecimiento.....	2
Resumen.....	2
Índice de Figuras	4
Índice de Tablas	5
1. CAPÍTULO 1: INTRODUCCIÓN	6
1.1. Objetivos	6
1.1.1. Objetivo general.....	6
1.1.2. Objetivos específicos	6
1.2. Funciones del Área de Soporte al Cliente de la empresa Soft Warehouse S.A 6	
1.3. Características del área.....	7
2. CAPÍTULO 2: FUNDAMENTO TEÓRICO	12
2.1. Técnicas de desarrollo de aplicaciones	12
2.1.1. Diseño ágil de aplicaciones (Extreme Programming).....	12
2.1.2. RAD (Diseño rápido de aplicaciones)	15
2.1.3. JAD (Diseño de aplicaciones por juntas)	18
2.2. Análisis de herramientas de desarrollo	23
2.2.1. Bases de datos	23
2.2.2. Lenguajes de programación	27
2.2.3. Herramientas case	30
2.3. Arquitectura de aplicaciones Web	35
2.3.1. Cliente servidor	36
2.3.2. Por 3 capas.....	37
2.4. Servicios de tecnología.....	38
2.4.1. Manejo de requerimientos.....	38
2.4.2. Manejo de incidentes	40
3. CAPÍTULO 3: CASO DE ESTUDIO	43
3.1. Introducción	43

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

3.1.1.	Objetivos	43
3.1.2.	General.....	43
3.1.3.	Específicos	43
3.2.	Principales Actividades.....	43
3.3.	Justificación de las herramientas seleccionadas para la construcción de la aplicación.....	44
4.	CAPÍTULO 4: LEVANTAMIENTO DE PROCESO.....	50
4.1.	Mapa de procesos.....	50
4.1.1	Cadena de Valor.....	50
4.2.	SIPOC	52
4.2.1.	Solicitar	52
4.2.1.1.	Definir Incidencia o Requerimiento.....	53
4.2.1.2.	Notificar Incidencia	54
4.2.2.	Procesar	55
4.2.2.1.	Analizar incidencia o requerimiento.....	55
4.2.2.2.	Presupuestar	56
4.2.3.	Solventar	57
4.2.3.1.	Solucionar Problema.....	57
4.2.4.	Evaluar	58
4.2.4.1.	Verificación de resolución	58
4.2.4.2.	Análisis de resultado.....	59
5.	CAPÍTULO 5: CONSTRUCCIÓN DEL PRODUCTO	60
5.1.	Inicio	60
5.1.1.	Análisis de requerimientos	60
5.1.2.	Especificaciones de funcionalidad	64
5.1.3.	Diagrama de Secuencia	86
5.2.	Elaboración.....	102
5.2.1.	Sistema ejecutable	102
5.2.2.	Lista de riesgos.....	110
5.3.	Construcción	110
5.3.1.	Elaboración.....	110
5.3.2.	Pruebas.....	111
5.4.	Transición	113
5.4.1.	Seguridades	113
5.4.2.	Costos	116
5.4.3.	Despliegue.....	116

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

5.4.4. Mantenimiento.....	117
6. CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES	118
6.1. Conclusiones.....	118
6.2. Recomendaciones.....	119
Anexos	120
Bibliografía.....	121

Índice de Figuras

FIGURA 1 ESTRUCTURA FIT-BANK	8
FIGURA 2 LOGO EXTREME PROGRAMMING.....	12
FIGURA 3 CICLO DE EVOLUCIÓN DEL MODELO ÁGIL XP	13
FIGURA 4 FASES DE LA METODOLOGÍA XP.....	15
FIGURA 5 RAD VS. METODOLOGÍA TRADICIONAL.....	16
FIGURA 6 FASES DE LA METODOLOGÍA RAD.....	17
FIGURA 7 JAD.....	19
FIGURA 8 UN JAD EXITOSO.....	20
FIGURA 9 SESIONES JAD	21
FIGURA 10 LOGO BASE DE DATOS.....	24
FIGURA 11 GESTORES DE BASES DE DATOS.....	25
FIGURA 12 LENGUAJES DE PROGRAMACIÓN.....	27
FIGURA 13 PRINCIPALES LENGUAJES DE PROGRAMACIÓN	28
FIGURA 14 VARIACIONES EN EL SIGNIFICADO DE CASE	31
FIGURA 15 RANGO HERRAMIENTAS CASE.....	34
FIGURA 16 ARQUITECTURA CLIENTE-SERVIDOR	35
FIGURA 17 ARQUITECTURA DE 3 CAPAS.....	37
FIGURA 18 MANEJO DE REQUERIMIENTOS.....	39
FIGURA 19 MANEJO DE INCIDENCIAS	40
FIGURA 20 GESTIÓN DE INCIDENCIAS	41
FIGURA 21 MAPA DE PROCESOS.....	50
FIGURA 22 DIAGRAMA SIPOC	52
FIGURA 23 DIAGRAMA GENERAL DE CASOS DE USO	63
FIGURA 24 DIAGRAMA F0. INGRESO AL SISTEMA.....	64
FIGURA 25 DIAGRAMA F1. ADMINISTRACIÓN DE INCIDENTES.....	65
FIGURA 26 F1.1 INGRESO DE INCIDENCIAS.....	66
FIGURA 27 F1.2 CONSULTAR INCIDENCIA.....	67
FIGURA 28 F1.3 MODIFICAR INCIDENCIA	68
FIGURA 29 F1.4 ACEPTAR INCIDENCIA	69
FIGURA 30 F1.5 ACTUALIZAR ESTATUS	70
FIGURA 31 F1.6 ASIGNAR CONSULTOR	71
FIGURA 32 F1.7 INGRESAR NOTAS ADICIONALES	72
FIGURA 33 F2. ADMINISTRACIÓN DE REQUERIMIENTOS.....	74
FIGURA 34 F2.1 INGRESO DE REQUERIMIENTO.....	75
FIGURA 35 F2.2 CONSULTAR REQUERIMIENTOS.....	76
FIGURA 36 F2.3 MODIFICAR REQUERIMIENTOS	77
FIGURA 37 F2.4 ACEPTAR REQUERIMIENTOS	78
FIGURA 38 F2.5 ACTUALIZAR ESTATUS	79
FIGURA 39 F2.6 ASIGNAR CONSULTOR	80
FIGURA 40 F2.7 INGRESAR NOTAS ADICIONALES	81
FIGURA 41 F3. ADMINISTRACIÓN DE TICKETS RESUELTOS.....	83

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

FIGURA 42 F3.1 CONSULTA	83
FIGURA 43 F4. ADMINISTRACIÓN DE REPORTES	84
FIGURA 44 F4.1 GENERAR REPORTE	85
FIGURA 45 DIAGRAMA DE SECUENCIA F0. INGRESO AL SISTEMA	87
FIGURA 46 DIAGRAMA DE SECUENCIA F1.1 INGRESO DE INCIDENCIA	88
FIGURA 47 DIAGRAMA DE SECUENCIA F1.2 CONSULTAR INCIDENCIA	89
FIGURA 48 F1.3 DIAGRAMA DE SECUENCIA MODIFICAR INCIDENCIA.....	89
FIGURA 49 DIAGRAMA DE SECUENCIA F1.4 ACEPTAR INCIDENCIA.....	90
FIGURA 50 DIAGRAMA DE SECUENCIA F1.5 ACTUALIZAR ESTATUS.....	91
FIGURA 51 DIAGRAMA DE SECUENCIA F1.6 ASIGNAR CONSULTOR	92
FIGURA 52 DIAGRAMA DE SECUENCIA F1.7 INGRESAR NOTAS ADICIONALES.....	92
FIGURA 53 DIAGRAMA DE SECUENCIA F2.1 INGRESO DE REQUERIMIENTO.....	93
FIGURA 54 DIAGRAMA DE SECUENCIA F2.2 CONSULTAR REQUERIMIENTO.....	94
FIGURA 55 DIAGRAMA DE SECUENCIA F2.3 MODIFICAR REQUERIMIENTO	95
FIGURA 56 DIAGRAMA DE SECUENCIA F2.4 ACEPTAR REQUERIMIENTO	96
FIGURA 57 DIAGRAMA DE SECUENCIA F2.5 ACTUALIZAR ESTATUS.....	97
FIGURA 58 DIAGRAMA DE SECUENCIA F2.6 ASIGNAR CONSULTOR	98
FIGURA 59 DIAGRAMA DE SECUENCIA F2.7 INGRESAR NOTAS ADICIONALES.....	99
FIGURA 60 DIAGRAMA DE SECUENCIA F3.1 CONSULTA TICKETS.....	100
FIGURA 61 DIAGRAMA DE SECUENCIA F4.1 GENERAR REPORTE.....	100
FIGURA 62 ESQUEMA DE DATOS	101
FIGURA 63 DIAGRAMA DE DESPLIEGUE.....	102
FIGURA 64 ESQUEMA DE SEGURIDAD	114
FIGURA 65 FLUJO DE UN MENSAJE PROCESADO POR EL FILTRO DE SEGURIDAD DESDE UCI	115

Índice de Tablas

TABLA 1 CLASIFICACIÓN HERRAMIENTAS CASE	33
TABLA 2 CADENA DE VALOR	51
TABLA 3 DIAGRAMA SIPOC DEFINIR INCIDENCIA O REQUERIMIENTO AUTOR: SANTIAGO GÁLVEZ.....	53
TABLA 4 DIAGRAMA SIPOC NOTIFICAR INCIDENCIA AUTOR: SANTIAGO GÁLVEZ	54
TABLA 5 DIAGRAMA SIPOC ANALIZAR INCIDENCIA O REQUERIMIENTO AUTOR: SANTIAGO GÁLVEZ.....	55
TABLA 6 DIAGRAMA SIPOC PRESUPUESTAR AUTOR: SANTIAGO GÁLVEZ.....	56
TABLA 7 DIAGRAMA SIPOC SOLUCIONAR PROBLEMA AUTOR: SANTIAGO GÁLVEZ.....	57
TABLA 8 DIAGRAMA SIPOC VERIFICACIÓN DE RESOLUCIÓN AUTOR: SANTIAGO GÁLVEZ.....	58
TABLA 9 DIAGRAMA SIPOC ANÁLISIS DE RESULTADO AUTOR: SANTIAGO GÁLVEZ.....	59
TABLA 10 ESPECIFICACIÓN DE COSTOS DEL SISTEMA HELPDESK.....	116

1. CAPÍTULO 1: INTRODUCCIÓN

Esta disertación de grado va a tratar sobre la necesidad de implementar un sistema que administre las incidencias y los requerimientos de los clientes de la empresa Soft Warehouse S.A, de tal manera que sean atendidas de forma rápida y eficiente las 24 horas del día, los 7 días de la semana, lo cual añadirá más valor al negocio.

Este capítulo menciona los objetivos que va a tener el proyecto tanto en general como en específico, y hace una pequeña aclaración de las funciones, características y problemática del área de Soporte al Cliente de la empresa.

1.1. Objetivos

1.1.1. Objetivo general

Analizar, diseñar y desarrollar un sistema para el manejo del proceso de incidencias y nuevos requerimientos que puedan tener clientes mediante tickets para la compañía Soft Warehouse.

1.1.2. Objetivos específicos

- Analizar la situación actual del proceso a automatizar.
- Diseñar la mejor alternativa de solución para la automatización del proceso.
- Desarrollar un sistema que automatice el proceso crítico de la empresa, utilizando software libre.
- Implementar la aplicación.
- Capacitar el manejo del software.
- Revisar las mejores prácticas en procesos de manejo de requerimientos e incidentes.

1.2. Funciones del Área de Soporte al Cliente de la empresa Soft Warehouse S.A

El área de soporte al cliente de la empresa es prácticamente toda la empresa si se excluye el área de administración, recursos humanos y contabilidad. Es por eso que este proyecto es de gran importancia para poder dar más valor al negocio y aumentar la satisfacción del cliente, concretamente la empresa ofrece productos para la sistematización de la gestión de las instituciones financieras, especializadas en bancos, cooperativas y empresas comerciales, específicamente enfocado a manejar y optimizar

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

las ganancias del negocio, improvisar el servicio al cliente y hacer más sencillo la producción y lanzamiento de nuevos productos, y servicios de tecnología como:

Asesoría para dimensionamiento de Hardware:

Partiendo de los volúmenes actuales y las proyecciones de crecimiento los técnicos le asesoran en las características y capacidades de los servidores y demás elementos tecnológicos para una operación eficiente.

Instalación de Software:

Instalación en los servidores del cliente, desde el software ambiental básico, la base de datos, los sistemas aplicativos y demás elementos de software requeridos para la eficiente operación de los sistemas

Acompañamiento en la implantación de sistemas:

Consultores funcionales y técnicos acompañan todo el proceso de implantación de los sistemas.

Capacitación técnica y a usuarios:

Cuentan con instructores de gran experiencia que permite garantizar la transferencia de conocimiento tanto a los técnicos de los clientes como a sus usuarios.

Mantenimiento:

Están especialmente comprometidos con el permanente mejoramiento y actualización de los sistemas, como parte de este servicio los clientes reciben periódicamente las nuevas versiones que incluyen mejoras y correcciones así como todas las adecuaciones de tipo legal y normativo.

1.3. Características del área

Para poder entender de una mejor manera la siguiente sección de la disertación se explicara más detalladamente que es FIT-BANK, y sus creadores nos aseguran que este software es:

“Es un core¹ para instituciones financieras creado para la sistematización de la gestión bancaria, y como ya se dijo anterior mente está especialmente orientado a controlar y mejorar la rentabilidad del negocio, elevar los niveles de servicio al cliente para aumentar el valor del negocio, facilitar el lanzamiento de nuevos productos y fortalecer dando impulso a la empresa.

Mediante estas observaciones minucioso de la información tanto como en la parte financiera como en la no financiera, este nos deja establecer cuáles clientes, productos, sucursales y ejecutivos generan ganancias y quienes no, de esta manera y a la par nos deja examinar él porque no están dando rentabilidad y con este análisis la institución o compañía pueda corregir insuficiencias para poder diseñar y crear productos y servicios ajustados a las necesidades particulares de los clientes con el propósito de volverlos rentables”. (FITBANK, 2013)

La apropiada gestión de la información de los clientes y una buena y completa orientación del servicio, darán a los ejecutivos comerciales manejar de una manera

¹ Núcleo del negocio, es la capa donde se encuentra toda la lógica del sistema y donde se hacen sus operaciones utilizando el concepto de Intercambio en tiempo real centralizado.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

eficiente la relación con los clientes logrando de esta manera clientes satisfechos y por lo tanto fieles a la compañía, con ganas de seguir confiando y comprando nuevos productos o servicios que esta proporcione.

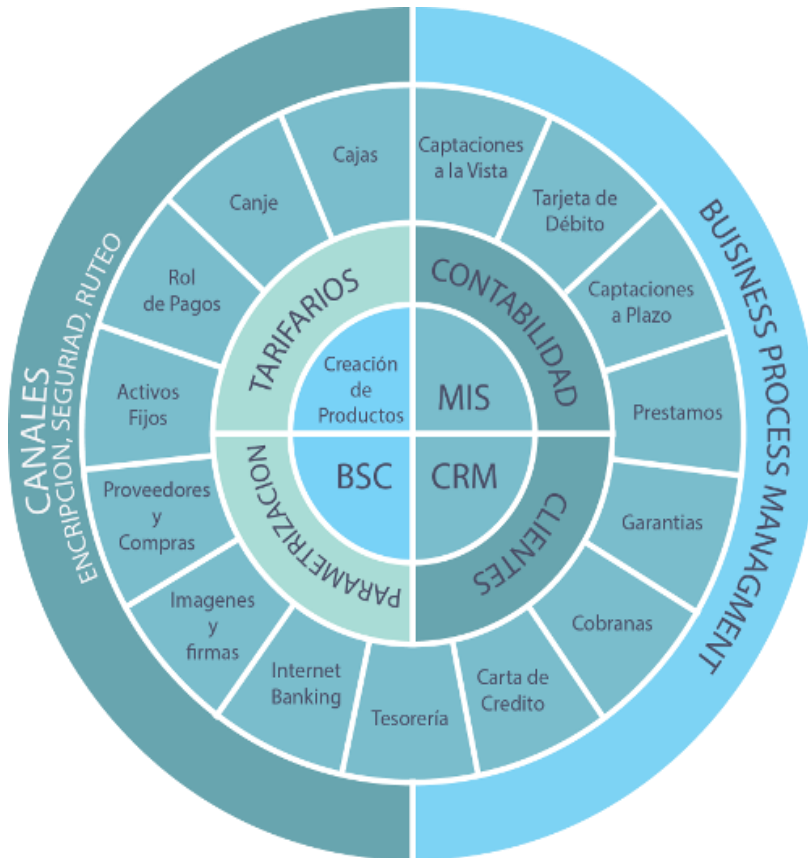


Figura 1 Estructura FIT-BANK

Fuentes: <https://fit-bank.com/fitbank.html>

Como podemos ver en la Figura 1, la estructura del core ha sido diseñado y construido en capas:

- La capa externa, llamada UCI por sus siglas en inglés, Universal Channel Interface, es según la empresa FitBank: “Una capa media independiente del núcleo, encargada al manejo de canales de acceso, se encarga de la interacción con el mundo externo (usuarios, clientes, prospectos, otras aplicaciones, etc.)” FIT-BANK. (n.d.). Retrieved December 08, 2016, from <https://fit-bank.com/fitbank.html>.

Además de manejar la seguridad y normalizar el acceso al Core, maneja canales como Cajeros Automáticos, Internet, Banca por teléfono, Kioscos, Mensajes de Texto con Celulares, Asistente digital personal o PDAs por sus ciclos en inglés etc.

- La capa de abajo es la de los varios módulos, estos son accedidos a través de transacciones ingresadas por cualquiera de los canales soportados por la capa del UCI, estas a su vez se relacionan con la base de datos de clientes y alimentan también las diferentes áreas.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- La capa hacia el centro del sistema que sigue es la de información acerca de clientes y contabilidad, estos subsistemas son en donde se basa la estructura del sistema. Los datos de los clientes está organizada de tal manera que pueden ser parametrizados si es necesario, su propósito es tener la información necesaria para poder crear productos o servicios conformes con su realidad y así obtener el mayor provecho de él.
- Por otra parte el subsistema contable fue elaborado de forma totalmente independiente de cualquier plan de cuentas y está dirigido a brindar oportuna información para el manejo del negocio.
- “En el mismo centro de todo el core bancario se encuentra el módulo de información gerencial o por sus siglas MIS, todos los subsistemas alimentan y almacenan información de alto interés para que la gerencia, la empresa asegure que sea de una institución financiera como cooperativas o Bancos.” FIT-BANK. (n.d.). Retrieved December 08, 2016, from <https://fit-bank.com/fitbank.html>

Algunas características tecnológicas que se debe tener en cuenta para la aplicación de este nuevo servicio que este proyecto implementará, y especialmente que el área de soporte al cliente tiene, son:

Internet nativo

Creado totalmente orientado para ambiente Internet, (3 capas) se funcionalidad le permite ser instalado en un Intranet para los usuarios y al mismo tiempo en Internet para que lo puedan usar usuarios tanto como clientes y prospectos.

Plataforma universal

Fue desarrollada en su totalidad en Java, XML, por lo tanto por las propiedades de este puede ser instalada sobre los principales equipos del mercado, también soporta bases de datos como Oracle, DB/2, SQL Server, Sysbase, PostgreSQL, MySQL.

Multidioma

Los usuarios pueden elegir a su gusto que idioma quieren usar gracias al diseño de FIT-BANK

Multicompañía

FIT-BANK puede soportar de una forma natural varias compañías, es por eso que no existe límite en la cantidad de empresas que el sistema puede administrar, cada empresa posee su propia contabilidad, su propios productos, por eso se manejan los procesos batch de forma independiente, los clientes son propios a nivel de todas las empresas pero cada una de estas solo podrá visualizar los clientes que posea.

Las transacciones entre compañías son muy sencillas de implementar, pues tiene seguridad y controles del mismo sistema.

“Non stop”

Todo el sistema fue creado con la capacidad de estar dispuesto todo el día todos los días de la semana, no cuentan con alguna clase de proceso que exija la suspensión del servicio a los usuarios. Todos los procesos batch se ejecutan basados en una agenda automática.

Soporte total a canales

Algunos de los canales como Internet de banca, Banca telefónica, Kioscos, celulares, Asistentes digitales (PDAs) y en resumen prácticamente cualquier dispositivo que aparezca en el futuro que soporte el nuevo estándar para cambio de información XML.

Bussines Process Management

FIT-BANK integra jBPM uno de las máquinas de Workflow más usadas mundialmente, jBPM permite definir los Procesos de Negocios de la Institución de forma sencilla y gráfica, se soporta BPEL y también JPDL.

1.4. Problemas presentados en el área

El tratamiento de la información de cada cliente con respecto a sus incidencias o requerimientos, así como los detalles de cada una de estas, al no existir un sistema exclusivo para su manejo, la información se maneja a través de mails, chats u otros medios donde la información no siempre es fácilmente accesible, además de que la información exclusiva de las empresas debe estar manejada solo entre el cliente y la empresa que ofrece el servicio.

Actualmente, el personal de la empresa Soft Warehouse recibe las novedades de los clientes mediante hojas electrónicas o en algunos casos mediante mails o mensajes por el chat de sus mails, o mensajes de texto a sus teléfonos celulares personales, lo cual hace que la información sea manejada de una manera un poco informal y dificulta el acceso a ella cuando se necesita tener constancia de que esa novedad ya fue solventada o de cómo se lo hizo.

Esta manera de llevar la información no es la más óptima cuando se requiere una constancia del trabajo realizado al momento de dar soporte al sistema, además de que no se lleva un historial de cómo resolver los problemas que es realmente importante.

La presente disertación contiene los siguientes capítulos:

- Capítulo I, una breve introducción de los objetivos del proyecto del área donde afectara y de la problemática que debe resolver.
- Capítulo II, cubre los aspectos teóricos que se revisarán para poder desarrollar este proyecto de la mejor manera, desde técnicas y herramientas hasta servicios tecnológicos que el sistema debe ofrecer.
- Capítulo III, nos da más amplia y profunda información sobre el impacto de este proyecto así como también la justificación o él por qué elegimos las herramientas que elegimos para el desarrollo.
- Capítulo IV, se genera el levantamiento de procesos, se describirán los procesos y subprocesos que integran la cadena de valor.
- Capítulo V, nos permite ver como toda la teoría se aplica en la práctica para poder construir el proyecto, aquí se mostrará el sistema ejecutándose.
- Capítulo VI, son las conclusiones y recomendaciones que podemos dar después de todo la experiencia generada en la elaboración de esta disertación

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

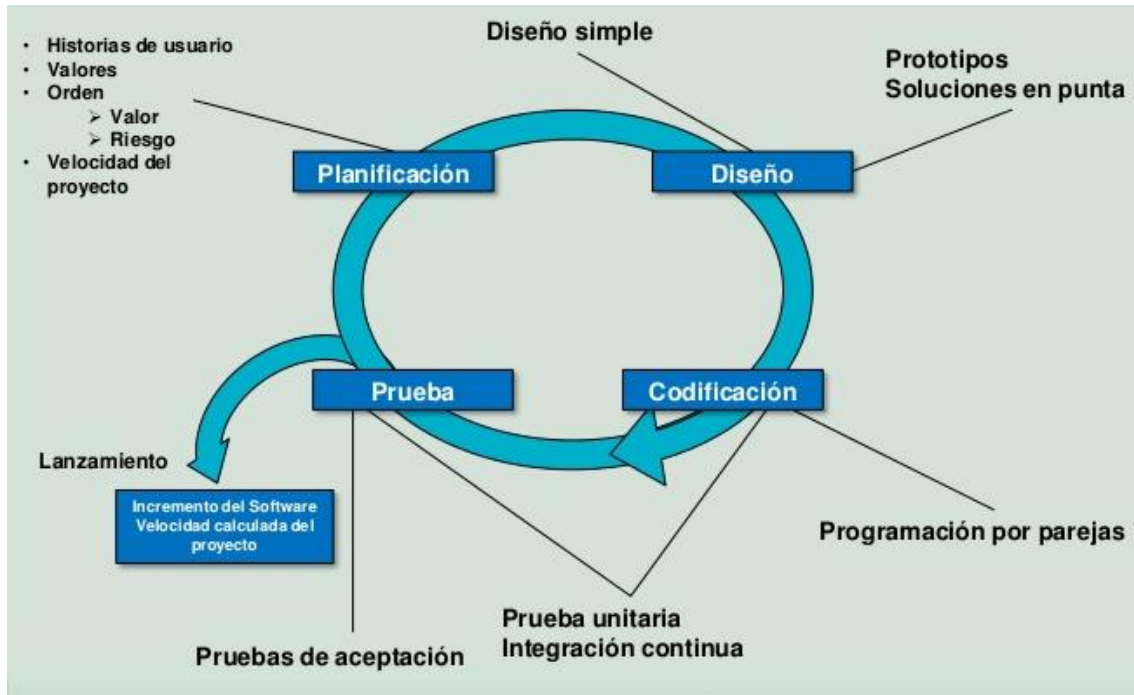


Figura 3 Ciclo de evolución del modelo ágil XP
Autor: Santiago Gálvez

Fases de la metodología Extreme Programming

Fase I - Planificación del proyecto

Historias de usuario:

Cuando se inicia un proyecto, que este enfocado a la metodología ya mencionada el primer paso a seguir es concretar las historias de usuario con él o los clientes. Estas tienen la finalidad compartida de los casos de uso pero con ciertas diferencias que se deben tener en cuenta: es importante que sean cortas, que consten de 3 o 4 líneas escritas por el cliente, debe ser un lenguaje no técnico y no debe hacer mucho hincapié en los detalles.

Las historias de usuarios son usadas para poder evaluar tiempos de desarrollo de la parte del desarrollo que narran. También son muy implementadas en la fase de pruebas.

Planificación de Entregas: Una vez elaboradas las historias de usuario es indispensable crear un plan de publicaciones, mucho más conocido y usado generalmente con su nombre en inglés "Release plan", donde se puedan visualizar las historias de usuario que se elaboraran para las repeticiones de versión o sector del software y los tiempos en las que se anunciarán. Un plan de publicaciones es una proyección para que los desarrolladores y clientes puedan establecer los tiempos de implementación perfectos para las historias de usuario, la prioridad que se dará a cada una de ellas y las historias que pertenecen a cada una de las secciones del programa.

Iteraciones: Cualquier proyecto que siga esta metodología se ha de dividir en iteraciones de normalmente pero variables 3 semanas de duración. Al iniciar cada una de las iteraciones los clientes deben seleccionar las historias de usuario previamente dadas en el plan de publicaciones para que serán implementadas.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

La Velocidad del Proyecto: Es una medida que representa que tan rápido se va a desarrollar el proyecto.

Programación en Parejas: Esta metodología sugiere que la programación se la desarrolle en pares, esto, según estudios, permite que crezca la productividad como también la calidad del software. Esta forma de programar requiere que dos programadores trabajen en un mismo equipo, uno estará codificando mientras el otro estará analizando si ese código está bien realizado y es adecuado para el programa.

Reuniones Diarias: También se recomienda que los que estén elaborando el programa programen reunirse diariamente y presenten las dificultades, como las solucionaron como también ideas en conjunto. Estas deben ser claras y sin complicaciones innecesarias y todos deberían participar abiertamente.

Fase II - Diseño

Diseños Simples: Esta metodología nos indica que se debe crear diseños fáciles de entender. Hay que hacerlo todo intuitivo y fácil de usar, si se logra esto a la larga el costo del proyecto, tiempo y el esfuerzo en la etapa de desarrollar disminuirán.

Glosarios de Términos: Al implementar un glosario de términos y una correcta especificación de las etiquetas de métodos y clases ayudará a comprender el diseño y será más fácil sus posteriores ampliaciones y como también reutilizar el código.

Riesgos: Si se presentan muchos inconvenientes potenciales en la fase de diseño, la metodología recomienda utilizar una pareja de desarrolladores para reducir el riesgo que supone esos inconvenientes.

Funcionabilidad extra: Jamás se debe crear una funcionalidad de más al programa si se piensa que esta será útil en un futuro. Tan solo un aproximado de un 10% de las funciones que se crean para que sean utilizadas en el futuro sirve, así que esta práctica es un desperdicio de recursos valiosos.

Refactorizar: esto es mejorar los códigos o estructuras ya creadas sin cambiar para lo que son hechos es decir sin alterar su funcionalidad. Para esto es necesario entender los códigos y una vez entendidos modificarlos para su optimización, esta práctica es bastante común como también desechar funcionalidades y diseños que ya no se van a usar.

Fase III - Codificación

Es importante mencionar de nuevo que el cliente es parte importante en todos los equipos, como también su presencia en las fases de la metodología.

A la hora de codificar una historia de usuario la presencia del cliente es necesaria. Los clientes son los que realizan y proveen las historias de usuario y tratan de modificar el tiempo en que van a implementarse. Por eso se debe ser muy específico y dar gran cantidad de detalle en la función que esta sección del programa hará, para constatar esto también es necesario que estén en las pruebas.

Fase IV - Pruebas

Una de las partes importante de la metodología XP es la comprobación de la funcionalidad por medio de las pruebas. Como se ilustraba en la Figura 4, podemos ver como las fases de esta metodología interaccionan entre sí y con la fase de pruebas.

El uso de las pruebas en XP es el siguiente:

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Las secciones del sistema se deben codificar para el entorno de pruebas antes de poder implementarlas en el sistema.
- Se deben crear las pruebas que pasarán los códigos antes de implementarlos.
- Tener en cuenta que las pruebas deben ser independientes al código que se evaluará.
- Como se comentó anteriormente las distintas pruebas deben estar subidas al repositorio común.
- Prueba de aceptación. Las pruebas mencionadas anteriormente no analizaran partes de las funcionalidades implementadas en las historias de usuarios sino verificaran las funciones generales.

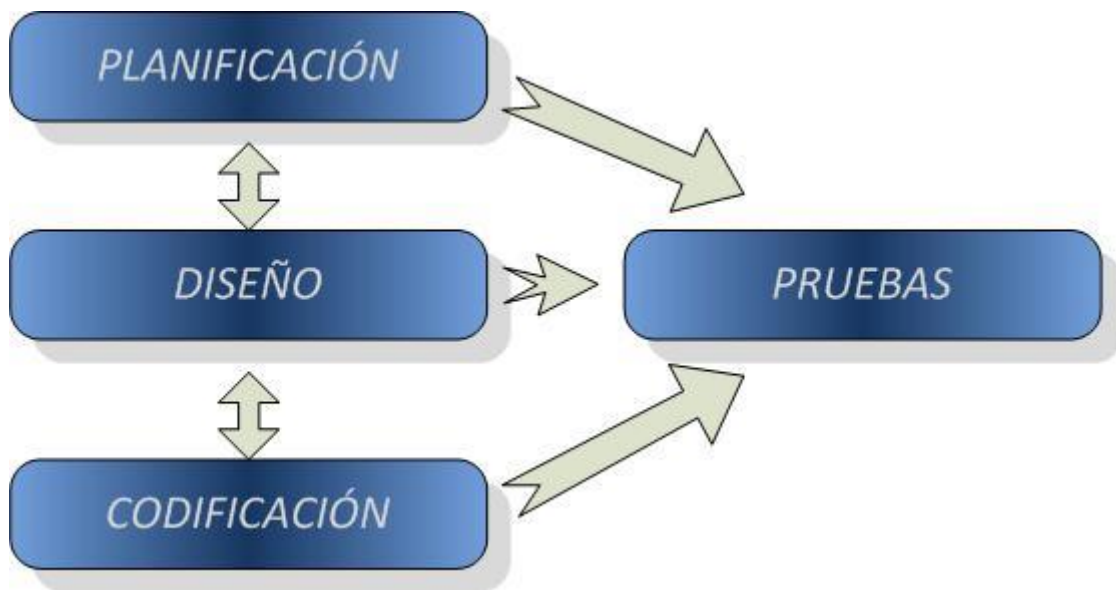


Figura 4 Fases de la metodología XP
Autor: Santiago Gálvez

Ventajas y Desventajas de la metodología XP

Ventajas:

- La codificación de las funciones es mucho mejor administrada.
- Existen menos errores.
- Se genera una sensación de satisfacción en los programadores.

Desventajas:

- Todas sus ventajas se verán n proyectos de plazo corto.
- Por la cercana participación del cliente, en caso de fallar se generan compromisos personales.

2.1.2. RAD (Diseño rápido de aplicaciones)

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Para poder saber de mejor manera lo que es RAD nos basaremos en lo que dice en autor del siguiente párrafo citado:

“El desarrollo rápido de aplicaciones o RAD (Rapid Application Development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980.

El método se basa en codificar por iteraciones, la elaboración de modelos y el uso de utilidades CASE.²

El RAD es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. RAD es una adaptación a "Alta velocidad" en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso RAD permite al equipo de desarrollo crear un "sistema completamente funcional" dentro de períodos cortos de tiempo.” (Curiosisimos, 2009) , PG 1.

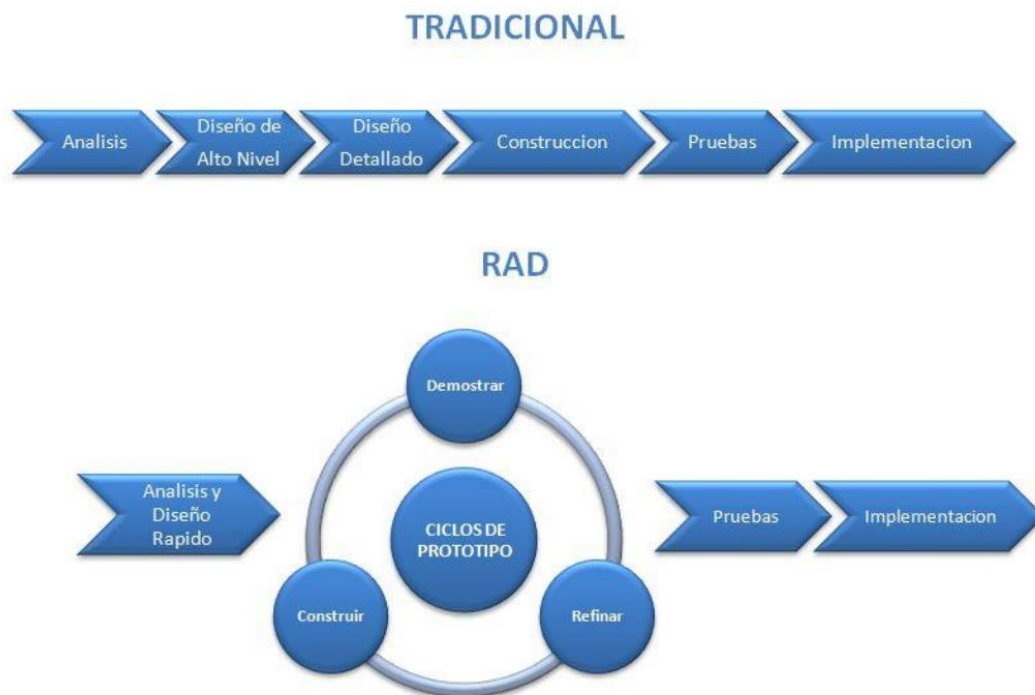


Figura 5 RAD vs. Metodología Tradicional.

Fuente: Rapid Application Development, <http://www.etondigital.com/services/>

Como podemos ver en la Figura 5 y basados en el autor ya antes referido podemos decir que:

“El tiempo es un limitante en un proyecto RAD demanda "ámbito en escalas". Si una aplicación de gestión puede modularse de forma que permita completarse cada una de las funciones principales en menos de tres meses es un candidato del RAD. Cada una

² Las herramientas CASE por sus siglas en inglés Computer Aided Software Engineering, o Ingeniería de Software Asistida por Computadora son programas creados y destinados a aumentar la productividad en el desarrollo reduciendo el costo como también el tiempo en que son creados. Estos programas también ayudaran a especificar una metodología de desarrollo de software.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

de las funciones puede ser afrontada por un equipo RAD diferente y ser integradas en un solo conjunto.

Al igual que todos los modelos de proceso, el enfoque RAD tiene inconvenientes:

- Para proyectos grandes aunque por escalas, el RAD requiere recursos humanos suficientes como para crear el número correcto de equipos RAD.
- RAD requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado.” (Curiosisimos, 2009)

Fases de Rapid Application Development

Si es que RAD está bien comprendido y es aplicado correctamente permite crear sistemas funcionales en periodos de 60 y 90 días, las fases se muestra en la Figura 6.

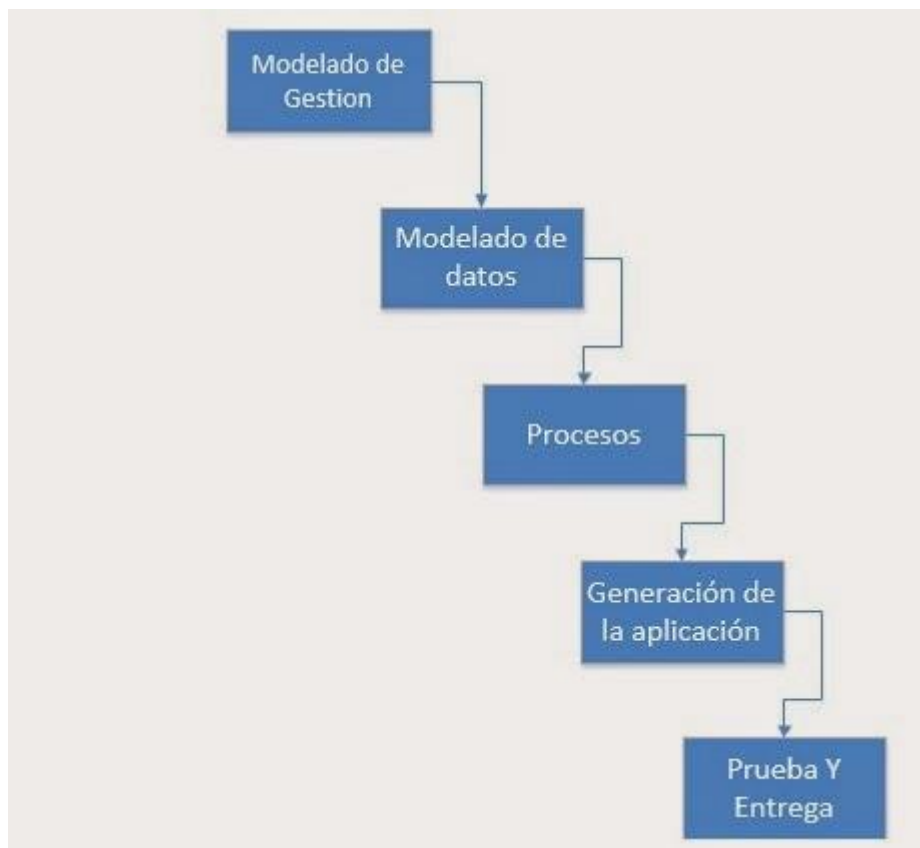


Figura 6 Fases de la metodología RAD
Autor: Santiago Gálvez

Modelado de gestión: la información que tenga cada una de las funciones debe ser suficiente para poder responder:

- ¿Qué información que maneja el proceso?
- ¿Cuál es la información que resulta?
- ¿Cuál es el origen y destino de la información?
- ¿Quién la proceso?

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Modelado de datos: En este proceso se crean los conjuntos de objetos de datos, estos deben ser creados para apoyar a la compañía.

Proceso: Las descripciones del proceso se crean para crear el CRUDE del proceso.

Generación de aplicaciones: este reutiliza programas ya hechos en vez de crear programas de cero para mejorar tiempos.

Pruebas de entrega: los tiempos de pruebas son reducidos gracias a que programas reutilizados ya fueron probados y aceptados.

Al aplicar esta metodología se tiene que tener muy en cuenta que:

“La limitación de tiempo impuesto en un proyecto RAD demanda "ámbito en escalas". Si una aplicación de gestión puede modularse de forma que permita completarse cada una de las funciones principales en menos de tres meses es un candidato del RAD. Cada una de las funciones puede ser afrontada por un equipo RAD diferente y ser integradas en un solo conjunto.” (Curiosisimos, 2009)

Ventajas y desventajas de la metodología RAD

Ventajas de RAD:

- Los avances del programa se pueden montar en cualquier plataforma
- Se usa una abstracción elevada.
- Se puede visibilizar en un periodo temprano.
- Gran flexibilidad.
- No se codifica a mano.
- Los usuarios están muy involucrados.
- Es probable que existan menos fallas.
- Reducción de costos.
- Ciclos de desarrollo cortos.
- Se estandariza la interfaz.

Desventajas de RAD

- No siempre es bueno construir.
- Costo alto de equipos y herramientas.
- Sus prototipos no siempre escalan.
- No existen demasiadas funciones (por "timeboxing"³).
- Se depende de terceros en componentes.

2.1.3. JAD (Diseño de aplicaciones por juntas)

³ El timeboxing consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que podamos en ese intervalo.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS



Figura 7 JAD
Fuente: www.matera.com

Como podemos visualizar en la Figura 7, JAD es una metodología de definición de requisitos y de diseño de la interfaz de usuario, que está fundamentada en reuniones participativas entre clientes, directiva y desarrolladores. En estas reuniones el tema a tratar debe ser el negocio.

Se debe ser cuidadoso porque en las reuniones pueden hacer ver a los clientes algo falso en cuanto al progreso del proyecto. Además, hay que prestar mucha atención con las estimaciones hechas sin un proceso, estas son las que llevan un mayor riesgo por el desconocimiento del sistema, estos deben ofrecer un rango mayor, uno que embarque una muy buena estimación y una muy mala estimación.

La técnica saca todo su potencial en modelos incrementales, esto es porque el sistema se mejora poco a poco adaptándose a las necesidades del cliente. Para su mejor funcionamiento es necesario que todos en el grupo se involucren al máximo. Si se utiliza de forma adecuada esta técnica deja ver requisitos no utilizados o poco útiles.

Estructura de la técnica

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

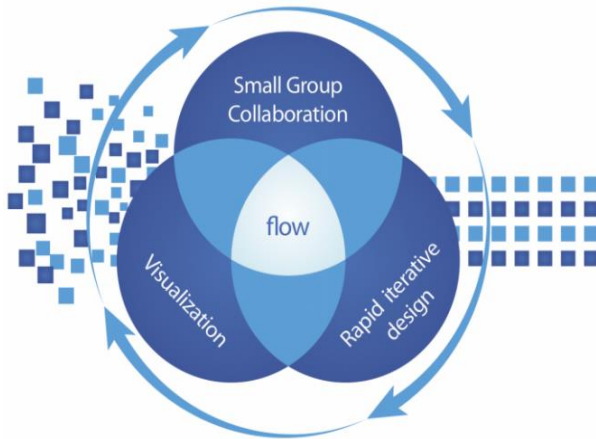


Figura 8 Un JAD exitoso
Autor: www.OneSprng.org

JAD consta de dos fases: planificación y diseño. Las dos hablan de los requisitos, pero sus niveles de abstracción son diferentes. Se debe tener en cuenta que en la planificación se tratan los requerimientos se estudia las potenciales utilidades y viabilidades de estas, en la fase de diseño se procede a probar el sistema por medio de prototipo y se elabora la interfaz de usuario, se habla del presupuesto, calendarización y la base de datos, obviamente si esta última es necesaria.

Cada una de estas fases llevaría entre 1 y 10 días. Es importante no confundir

la fase de diseño JAD con la del proyecto ya que JAD se aplica en la fase de planificación y análisis.

Como podemos ver en la Figura 8, después de colaboración entre los grupos o planificación y del diseño se debe visualizar para verificar si se está llegando al objetivo establecido, esto es un flujo que se sigue hasta la culminación del proyecto.

Todas las fases se dividen en tres secciones: preparación que es simplemente decidir quién va a cada reunión y de lo que se va a tratar, la reunión y la conclusión, donde se eligen los puntos clave expuestos durante la sesión y se guardan en algún medio de forma permanente. Tenemos que tener cuidado de no caer en la tecnofilia, el trabajo está bien elaborado cuando siempre tenga un soporte, sea accesible por todos y sea la imagen de lo consensuado por las partes.

Una vez terminado el proceso de JAD, se debe continuar con el método de desarrollo elegido. Es decir, JAD es independiente del modelo de desarrollo, por lo que se puede aplicar siempre.

Las sesiones

El equipo se reunirá en un lugar neutral, y este lugar deberá tener todo lo que el equipo necesite para enfocarse en el trabajo.

Como podemos ver en la Figura 9 los roles son los siguientes:

- Moderador
- Ejecutivo de parte de los clientes
- Usuario final
- Programador
- Secretario
- Especialistas(competentes para el trabajo que se esté realizando)

Se debe tener en cuenta que las reuniones deben ser realizadas solo por gente clave, se recomienda que el número de personas no sea mayor a 8 y por esta razón los especialistas no es necesario que asistan a todas las reuniones.

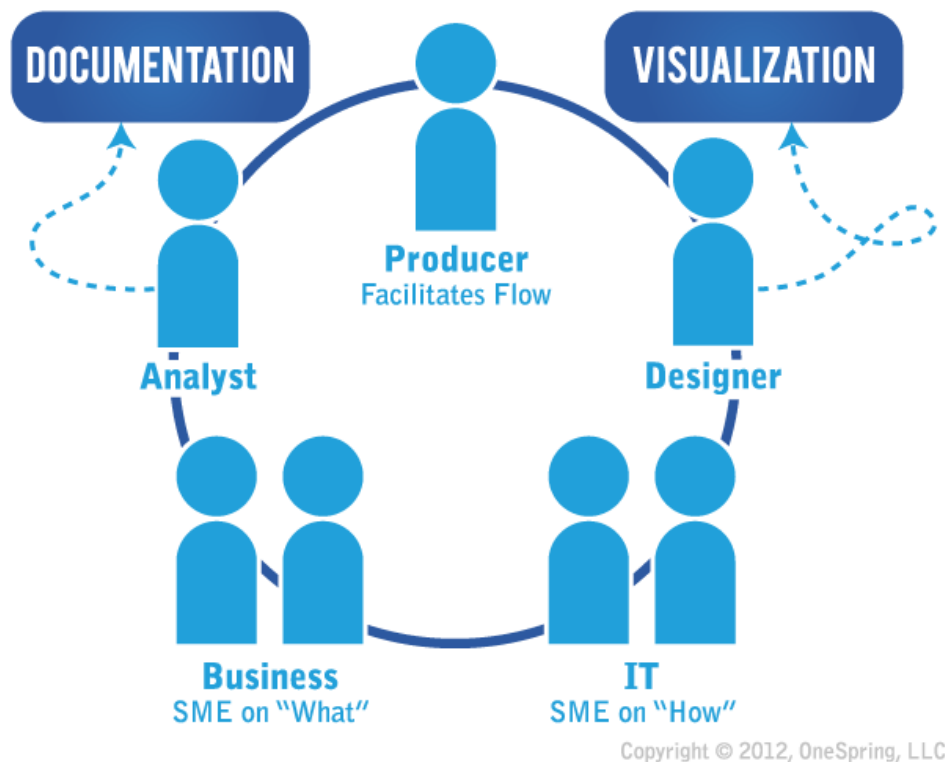


Figura 9 Sesiones JAD
Fuente: www.OneSpring.org

Es indispensable que las reuniones sean concisas y cortas para esto se necesita tener una planeación previa de los temas que se van a hablar en la reunión, McConnell, S. (1996) nos dice que:

Temas a tratar en las sesiones

En el JAD de planificación, que dura entre 1 y 5 días, se trata lo siguiente:

- 1.- Conducir la orientación. Introducción.
- 2.- Definir los requisitos de alto nivel.
- 3.- Limitar el alcance del sistema.
- 4.- Identificar y estimar las fases del diseño JAD.
- 5.- Identificar los participantes del diseño JAD.
- 6.- Planificar la sesión de diseño JAD.
- 7.- Documentar las decisiones tomadas.
- 8.- Conclusión.

En la reunión llevada a cabo durante la fase de diseño JAD se tratarán los siguientes puntos:

- 1.- Conducir la orientación.
- 2.- Refinar y limitar los requisitos de alto nivel identificados en la fase de plan JAD.
- 3.- Desarrollar un flujo de trabajo o workflow.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- 4.- Desarrollar la descripción de dicho workflow.
- 5.- Diseñar la interfaz de usuario.
- 6.- Especificar requisitos de procesamiento.
- 7.- Definir interfaces.
- 8.- Identificar grupos de datos y funciones.
- 9.- Documentar las decisiones consensuadas.
- 10.- Conclusión.

(Rapid development: Taming wild software schedules. Redmond, WA: Microsoft Press.)

Roles del JAD

El patrocinador del proyecto

Es quien presupuesta el proyecto, el dueño del sistema. Tienen el lugar más alto en la organización, de modo que ellos pueden tomar las decisiones y proporcionar los recursos necesarios y apoyar para el proyecto. Las responsabilidades de Patrocinador de proyecto son:

- Asegurar que los clientes correctos sean parte del grupo
- Asegurar que hay suficiente personal de soporte técnico en el proyecto
- Ayudar en la selección de casos de prueba
- Ayudar en la definición del alcance y funcionalidad
- Ayudar en el benchmarking⁴ contra los sistemas actuales y los sistemas externos
- Evaluar si el sistema es eficaz o no

Líder del Proyecto

Tiene que estar comprometido al proyecto, tener un conocimiento de fondo del área comercial y sistemas de información actuales relacionados. Necesitan ser entusiastas, objetivos y no permitirle que nadie domine el grupo. Responsabilidades del Líder de proyecto:

- Asegurar que todos los roles de su equipo estén asignados.
- Asegurar que las reuniones se planifiquen y publiquen respaldadas por agenda.
- Asegurar que la agenda se planifique y se siga.
- Asegurar que las tareas se asignen y se cumplan, y que el listado de tareas se ejecute en la secuencia prevista.
- Coordinar el esfuerzo de los analistas del equipo.

⁴ El benchmarking es un anglicismo que, en las ciencias de la administración de empresas, puede definirse como un proceso sistemático y continuo para evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Registrador

Toma los apuntes durante la sesión, y los revisa en un resumen conciso de discusiones y decisiones. Es importante que las notas resultantes no sean una transcripción de lo que se dijo. Este papel puede compartirse entre varios miembros del equipo según la necesidad. Estas notas sirven como una referencia al grupo al retomar las discusiones, y como referencia para retomar los puntos complejos. Responsabilidades:

- Tomar notas durante las reuniones.
- Resumir y compactar notas después de la reunión.
- Asegura que el líder del proyecto así como el patrocinador revisen las notas y las corrijan antes de publicarlas.
- Guardar un historial de notas para que los nuevos miembros del equipo que van a entrar en las próximas fases del proyecto estén al tanto de lo que ha pasado.

Encargado del tiempo

Es el responsable de asegurar que se cumpla la agenda establecida a fin de optimizar el tiempo.

Clientes

Son los que conocen cómo funcionará el sistema y cómo se usa. Ellos ayudarán al equipo a comprender las tareas que va a realizar el sistema. Responsabilidades:

- Definir la información con la que va a tratar el proceso.
- Crear casos de uso para su prueba.
- Analizar los obstáculos para el éxito en el ambiente actual.

Las ventajas y las desventajas

- Con métodos tradicionales, JAD es muy costoso y puede ser incómodo si el grupo es demasiado grande.
- JAD desarrolla un sentido de propiedad en los usuarios solicitantes del producto gracias a que están involucrados en el procesos de desarrollo.
- JAD es bastante bueno en dejar los requisitos claros, una comprensión mejor de metas comunes, y a un compromiso fuerte con el éxito del nuevo sistema.

2.2. Análisis de herramientas de desarrollo

2.2.1. Bases de datos



Figura 10 Logo Base de Datos
Autor: Santiago Gálvez

Una base de datos o banco de datos (en ocasiones abreviada con la sigla BD o expresada normalmente como se muestra en la Figura 10) es un compuesto de información que pertenece a un mismo tema en particular y guardada metódicamente para usarlo cuando convenga. Analizándolo desde este punto de vista una biblioteca es una base de datos conformada por documentos y textos impresos en papel e indexados para su consulta.

Sistema de Gestión de Bases de Datos (SGBD)

Según el autor de la siguiente referencia asegura es: “Un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas”. Que Es Una Base De Datos pdfs. (n.d.). Tomado Diciembre 08, 2016, de <http://ebooks-kings.com/pdf/que-es-una-base-de-datos>

A continuación en la Figura 11 se presentan algunas de los principales gestores de bases de datos.



Figura 11 Gestores de Bases de Datos
Autor: Santiago Gálvez

Principales tipos de Bases de Datos

Los dos siguientes tipos de bases de datos, aunque no los únicos, más son los importantes según el criterio del autor.

“Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones”. (Herrera, 2011)

Ejemplos de gestores de bases de datos más utilizadas.

PostgreSQL

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS



Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*).

Oracle Database

ORACLE® Es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera a Oracle Database como uno de los sistemas de bases de datos más completos destacando:

- Soporte de transacciones
- Estabilidad
- Escalabilidad
- Soporte multiplataforma.

Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia del Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

MySQL



Es un sistema para la de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Lenguaje SQL

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

En la Figura 13 se van a presentar los porcentajes del mercado que posee cada lenguaje de programación, este estudio fue realizado en el 2015.

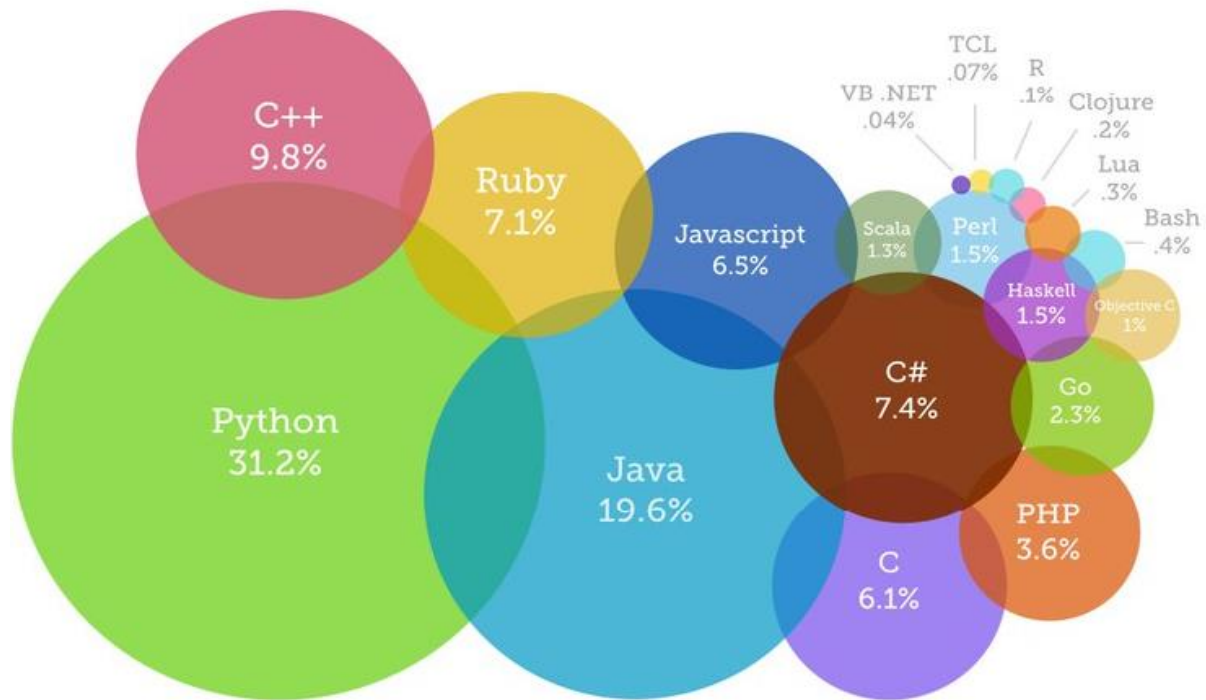


Figura 13 Principales Lenguajes de Programación
Fuente: www.codeeval.com

Clasificación

De la gran variedad de lenguajes de programación con características y con fines tan diferentes, se pueden clasificar en dos grupos lenguajes máquina y los lenguajes simbólicos.

Lenguajes de máquina

Son los lenguajes que son totalmente legibles para la máquina (computadora), ya que sus instrucciones son cadenas binarias (cadenas o series de caracteres de 0 y 1) que especifican una operación. El código máquina es el conocido código o sistema binario.

Lenguajes simbólicos

A los lenguajes de máquina les sucedieron aquellos que permiten programar con instrucciones muy parecidas al lenguaje que usamos los humanos (casi siempre en inglés). Los lenguajes simbólicos se ramifican a su vez en lenguajes de bajo nivel, también conocidos como ensambladores y lenguajes de alto nivel.

Lenguajes de bajo nivel o ensambladores

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Son más fáciles de utilizar que los lenguajes máquina, pero al igual que ellos, dependen de la máquina en particular. El lenguaje de bajo nivel por excelencia es el ensamblador. Las instrucciones en lenguaje ensamblador son instrucciones conocidas como nemotécnicos.

Un programa escrito en lenguaje ensamblador, requiere de una fase de traducción al lenguaje máquina para poder ser ejecutado directamente por la computadora, esto es en lo que difiere esencialmente con el lenguaje de máquina.

El programa original escrito en lenguaje ensamblador se denomina programa fuente y el programa traducido en lenguaje máquina se conoce como programa objeto, el cual ya es directamente entendible por la computadora. Los programas traducidos los elaboran el ensamblador, compiladores o intérpretes se les denomina procesadores de lenguajes.

Hoy en día, los lenguajes ensambladores tienen sus aplicaciones muy reducidas, se centran básicamente en aplicaciones de tiempo real, control de procesos y de dispositivos electrónicos.

Lenguajes de alto nivel

Estos lenguajes son los más utilizados por los programadores. Fueron creados para que las personas puedan leer o crear programas mucho más sencillos que los lenguajes de máquina y ensambladores. Un programa escrito en lenguaje de alto nivel es independiente de la máquina (las instrucciones no dependen del diseño del hardware o de una computadora en particular), por lo que estos programas son portables o transportables. Los programas escritos en lenguaje de alto nivel pueden ser ejecutados con poca o ninguna modificación en diferentes tipos de computadoras.

Los lenguajes de alto nivel se dividen en lenguajes procedurales y relacionales.

- Lenguaje procedural, se debe indicar lo que se quiere hacer y el modo de hacerlo
- Lenguaje relacional, solo se debe indicar lo que se quiere hacer, no hay necesidad de especificar el modo de hacerlo.

Las generaciones de los lenguajes son las siguientes:

- 1era Generación: lenguajes máquina
- 2da Generación: lenguajes ensambladores
- 3era Generación: lenguajes simbólicos de alto nivel procedurales.
- 4ta Generación: lenguajes relacionales.

Paradigmas

Estos existen para poder facilitar la difícil tarea de la programación en diferentes ámbitos. Hay 4 clases de programación que describen casi todos los lenguajes de programación actuales: el imperativo, el aplicativo, el lenguaje con base en reglas y el orientado a objetos.

- Lenguajes imperativos

Los lenguajes imperativos o de procedimiento controlados por instrucciones. Se componen de una serie de enunciados, que cambian el estado de la memoria en sesiones pequeñas o grandes. Esta es la más primitiva y la que muchos se imaginan al hablar de programación pues lo siguen usando programas muy conocidos y de uso amplio como C, C++, FORTRAN, ALGOL, P1. /I, Pascal, Ada, Smalltalk, COBOL.

- Lenguajes aplicativos

Para los lenguajes aplicativos o funcionales se debe prestar atención al cargo que el programa tiene, no solo a las variaciones en memoria que se dan conforme el programa se ejecuta.

En otras palabras, en vez de examinar la serie de estados a través de los cuales debe pasar la máquina para obtener una respuesta, debemos fijarnos en los resultados que surgieron. Los lenguajes que hacen énfasis en este punto de vista se conocen como lenguajes aplicativos o funcionales.

- Lenguajes con base en reglas

Se ejecutan solo si existe una condición que los habilita y cuando se logra satisfacer, ejecutan una acción ya definida. Un gran ejemplo de un lenguaje de esta clase es Prolog, también es conocida por programación lógica por sus condiciones habilitadoras que son expresiones lógicas escritas en predicados. Es muy similar a los lenguajes imperativos, pero con la diferencia que los enunciados no son secuenciales.

- Programación orientada a objetos

Para este tipo de lenguaje se crean objetos o abstracciones de la realidad y a estos se les da funciones.

Estos objetos son extensiones de otros más simples que tienen la cualidad de heredar propiedades.

Esto tiene muchas ventajas porque al construir objetos de datos estos ganan la eficiencia de lenguajes imperativos y al utilizar clases de funciones restringidas se construye la flexibilidad y confiabilidad del modelo aplicativo.

2.2.3. Herramientas case



Se las define como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, a lo largo de todos los pasos del Ciclo de Vida de desarrollo de un Software.

CASE se define también como:

- Conjunto de métodos, utilidades y técnicas que proporcionan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.
- La sigla genérica para una serie de programas y una filosofía de desarrollo de software que ayuda a automatizar el ciclo de vida de desarrollo de los sistemas.
- Una innovación en la organización, un concepto avanzado en la evolución de tecnología con un potencial efecto profundo en la organización. Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

La Figura 14 nos muestra las variaciones que puede tener la palabra CASE pero guardando siempre su significado.

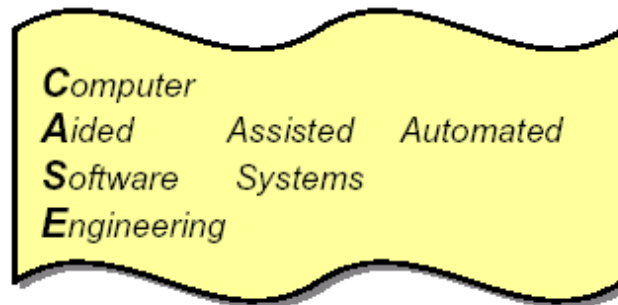


Figura 14 Variaciones en el significado de CASE
Fuente: <http://ceds.nauta.es/Catal/Products/caselist2.htm>

La realización de un nuevo software requiere que las tareas sean organizadas y completadas en forma correcta y eficiente. Las Herramientas CASE fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software.

La mejor razón para la creación de estas herramientas fue el incremento en la velocidad de desarrollo de los sistemas. Por esto, las compañías pudieron desarrollar sistemas sin encarar el problema de tener cambios en las necesidades del negocio, antes de finalizar el proceso de desarrollo.



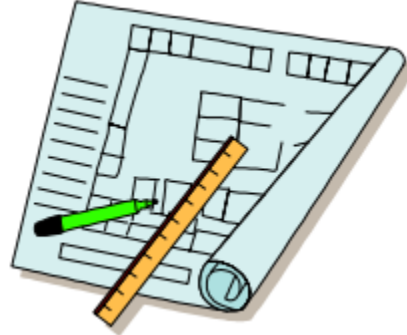
También permite a las compañías competir más efectivamente usando estos sistemas desarrollados para compararlos con sus necesidades de negocio actuales. En un mercado altamente competitivo, esto puede hacer la diferencia entre el éxito y el fracaso.

Las herramientas CASE permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Cuando CASE es integrado a las organizaciones estas pueden desarrollar rápidamente sistemas de mejor calidad para soportar procesos críticos del negocio y asistir en el desarrollo y promoción de la información de productos y servicios, algunas de ellas son:

- Verificar el uso de todos los elementos en el sistema diseñado.
- Computarizar el dibujo de diagramas.
- Generar la documentación del sistema.
- Creación de relaciones e la base de datos automática.
- Estructurar el código.



La principal ventaja de la utilización de una herramienta CASE, es aumentar de la calidad del avance en el desarrollo realizado y, en segundo, el aumento de la productividad. Para conseguir estos dos objetivos es conveniente contar con una metodología de trabajo, además de la propia herramienta. La mejora de calidad se consigue reduciendo sustancialmente muchos de los problemas de análisis y diseño, inherentes a los proyectos de mediano y gran tamaño (lógica del diseño, coherencia, consolidación, etc.). La mejora de productividad se consigue a través de la automatización de determinadas tareas, como la generación de código y la reutilización de objetos o módulos.

Clasificación de las Herramientas Case

No existe una única clasificación de herramientas CASE y, en ocasiones, es difícil hacer que encajen en una única clase.

Sin embargo se podrían clasificar con respecto a:

- Plataformas soportadas.
- Según las fases o fase del desarrollo del software.
- La arquitectura para la que fueron diseñadas.
- Que funciones cumplen.

Como podemos ver en la Tabla 1 las herramientas CASE, en función las fases o fase del desarrollo del software, se pueden agrupar de la siguiente forma:

1. Herramientas integradas, I-CASE (Integrated CASE, CASE integrado): comprenden todas las fases del ciclo de vida del desarrollo del producto. Son también llamadas CASE banco de trabajo.

2. Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior) o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

3. Herramientas de bajo nivel, L-CASE (Lower CASE - CASE inferior) o back-end, están destinadas a las últimas fases del desarrollo de software normalmente cuando el producto se implementa.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4. Juegos de herramientas o Tools-Case, es un tipo de Herramienta muy fácil de usar, sirven para computarizar por completo una fase dentro del ciclo de vida. Dentro de este grupo se encontrarían las herramientas de reingeniería, orientadas a la fase de mantenimiento.

Tipo de Case	Ventajas	Desventajas
I – Case	<ul style="list-style-type: none"> ➤ Integra el ciclo de vida. ➤ Permite lograr importantes mejoras de productividad a mediano plazo. ➤ Permite un eficiente soporte al mantenimiento de sistemas. ➤ Mantiene la consistencia de los sistemas a nivel corporativo. 	<ul style="list-style-type: none"> ➤ No es tan eficiente para soluciones simples, sino para soluciones complejas. ➤ Depende del Hardware y del Software. ➤ Es costoso.
Upper Case	<ul style="list-style-type: none"> ➤ Se utiliza en plataforma PC, es aplicable a diferentes entornos, ➤ Menor costo 	<ul style="list-style-type: none"> ➤ Permite mejorar la calidad de los sistemas, pero no mejora la productividad. ➤ No permite la integración del ciclo de vida.
Lower Case	<ul style="list-style-type: none"> ➤ Permite lograr importantes mejoras de productividad a corto plazo. ➤ Permite un eficiente soporte al mantenimiento de sistemas. 	<ul style="list-style-type: none"> ➤ No garantiza la consistencia de los resultados a nivel corporativo. ➤ No garantiza la eficiencia del Análisis y Diseño. ➤ No permite la integración del ciclo de vida.

Tabla 1 Clasificación Herramientas CASE
Fuente: Análisis Y Diseño De Sistemas 3ª. Edición Kendall & Kendall

En la Figura 15 podemos ver cuáles son los rangos que nos ofrecen las herramientas CASE, ya que no todas abarcan todas las fases o pueden ayudar con los análisis.

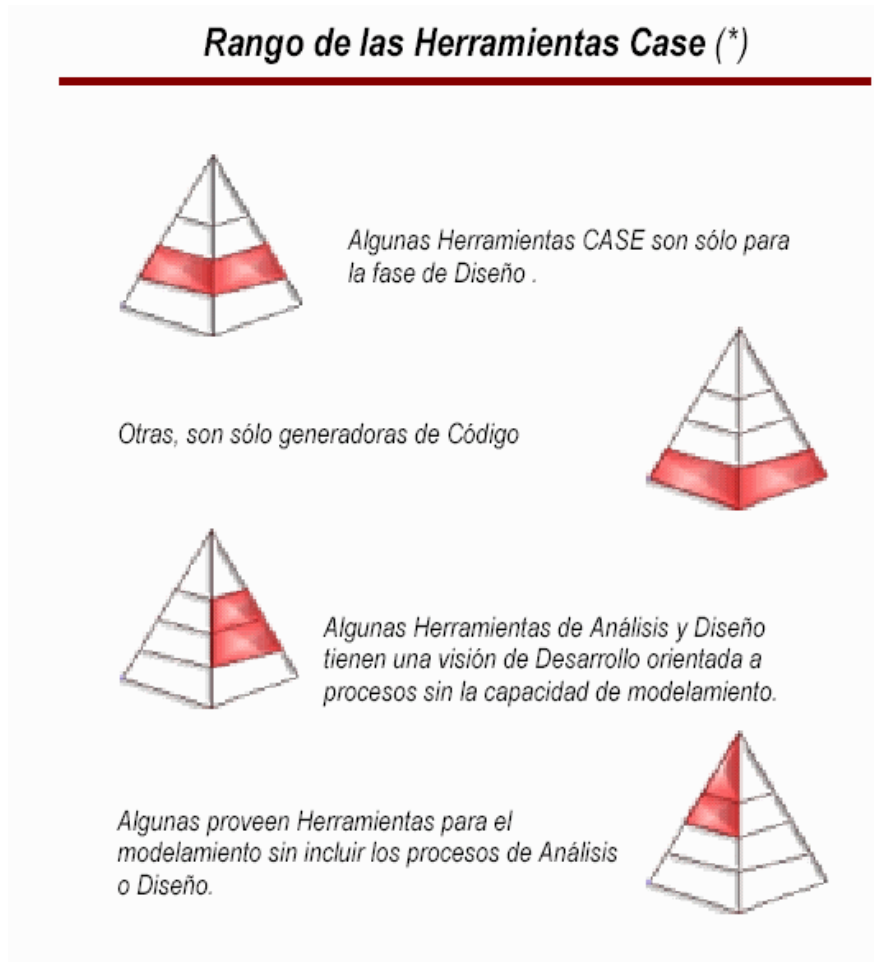
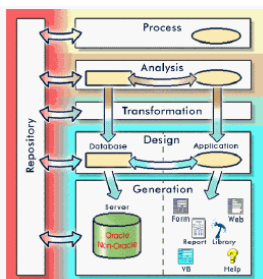


Figura 15 Rango Herramientas CASE
Fuente: Revista, Tecnología de Punta

Ejemplos de Herramientas Case más utilizadas.

Oracle Designer



Oracle Designer es un conjunto de herramientas para desarrollar las definiciones que necesita el usuario y computarizar la construcción de aplicaciones cliente/servidor flexibles y gráficas. Integrado con Oracle Developer, Oracle Designer provee una solución para desarrollar sistemas empresariales cliente/servidor de segunda generación.

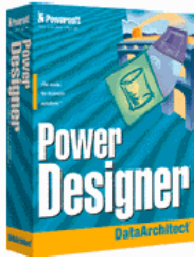
Provee sofisticadas aplicaciones cliente/servidor que pueden ser 100% generadas usando la lógica de la aplicación y el módulo de componente.

Esta herramienta habilita la capa de diseño que ya tiene el proyecto, protegiendo la versión actual. Cualquier cambio realizado por cualquier herramienta de Oracle Designer, en cualquier fase del proyecto, se almacena en un centro de cambios central,

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

haciendo que sea más fácil para el equipo acceder a esto y así también facilitando la administración del proyecto.

PowerDesigner



PowerDesigner es un conjunto de aplicaciones de la marca Powersoft para el diseño, modelado y construcción de datos por medio de diversas aplicaciones.

Es una base sólida de aplicaciones de alto rendimiento para los desarrolladores cliente/servidor que sirve para el análisis, diseño inteligente y construcción sólida de una base de datos orientado a modelos de datos a nivel físico y conceptual.

SNAP



SNAP es un CASE para el desarrollo de aplicaciones en Sistemas AS/400 de IBM. Proporciona el ambiente integral de trabajo, brindando la posibilidad de construir sistemas de inmejorable calidad, adheridos a los estándares S.A.A de IBM., totalmente documentados y ajustados a los requerimientos específicos de la organización, en una fracción del tiempo y costo del que se invertiría, si se utilizaran herramientas tradicionales. SNAP se ha consolidado como el CASE más poderoso y con mejor historial de resultados, disponible para desarrollo de sistemas en el AS/400 de IBM. Genera los programas nativos de mejor rendimiento en AS/400. Así mismo, proporciona dos ambientes de trabajo y genera aplicaciones nativas y/o Cliente/Servidor con el mismo esfuerzo de desarrollo.

2.3. Arquitectura de aplicaciones Web

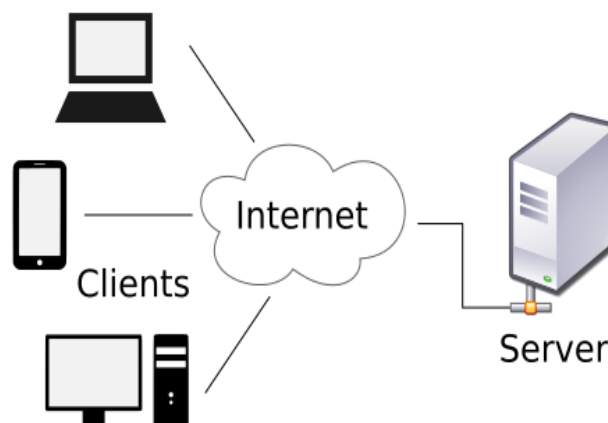


Figura 16 Arquitectura Cliente-Servidor
Autor: David Vignoni

2.3.1. Cliente servidor

Como podemos ver en la Figura 16 es un modelo de aplicación distribuida, y el autor nos asegura que:

“En el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta“. (Anonimo, 2014)

Características

En esta arquitectura el agente de una solicitud se lo denomina como cliente.

A continuación alguna de sus características:

- Es quien inicia solicitudes o peticiones.
- Es quien va a ser enviada la respuesta del servidor.
- Generalmente está conectado a varios servidores al mismo tiempo.
- Los usuarios pueden interactuar con el mediante la interface gráfica.

Al otro lado, recibiendo las solicitudes del cliente se encuentra el servidor.

A continuación algunas de sus características:



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS



- Esperan pasivamente a recibir y procesar las solicitudes de los usuarios.
- Su función es procesar las solicitudes que llegaron hasta él y enviar una respuesta.
- Normalmente, como ya lo mencionamos antes, los servidores aceptan un número considerable de clientes, se puede limitar el número de conexiones, como no.

2.3.2. Por 3 capas

Arquitectura 3 Capas

La arquitectura 3 capas o programación 3 capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos, tal como podemos ver en la Figura 17.

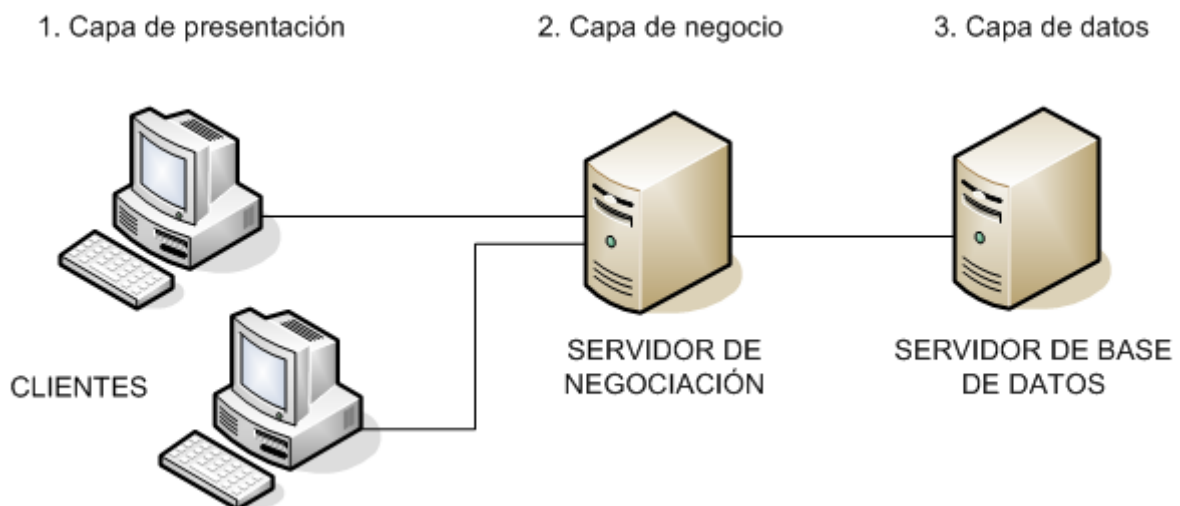


Figura 17 Arquitectura de 3 Capas
Autor: Santiago Gálvez

Ventajas de esta Arquitectura

- El desarrollo se puede llevar a cabo en varios niveles
- Desarrollos paralelos (en cada capa)
- Aplicaciones más robustas debido al encapsulamiento
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica)
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad)

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

Capas y niveles

- Capa de Presentación: esta es la parte que ve el usuario, las pantallas que se le muestra para que él interactúe con el programa (también se le conoce como “capa de usuario”), comunicándole la información y recolectando la información. Esta capa se comunica únicamente con la capa de negocio llevando y trayendo los datos o registros necesarios, es la interfaz gráfica del programa y debe ser lo más amena posible para una mejor comunicación con el usuario.
- Capa de negocio: Donde está toda la lógica del negocio, todos sus productos servicios serán procesados en esta capa, todos los procesos se realizan aquí.
- Capa de datos: Donde están guardados todos los datos, también se encarga de la comunicación de programa a la base de datos mediante herramientas.

2.4. Servicios de tecnología

2.4.1. Manejo de requerimientos

Como ITIL es el exponente en concepto y más desarrollado y aceptado método para la Gestión de Peticiones o manejo de requerimientos, me basaré en estos en la parte teórica, ITIL asegura que:

“Es la encargada de atender las peticiones de los usuarios proporcionándoles información y acceso rápido a los servicios estándar de la organización TI.

Es importante aclarar qué entendemos por petición de servicio, un concepto que engloba las solicitudes que los usuarios pueden plantear al departamento de TI:

- Solicitudes de información o consejo.
- Peticiones de cambios estándar (por ejemplo cuando el usuario olvida su contraseña y solicita una nueva)
- Peticiones de acceso a servicios IT.

La Gestión de Peticiones recibe las siguientes entradas para poder iniciar su labor:

- Peticiones de servicio, planteadas por los usuarios.
- Solicitud de cambios o RFCs, también de la misma fuente.
- Descripción detallada del servicio, proporcionada por el Portafolio de Servicios⁵.

⁵ Gestión del Portafolio de Servicios consiste en definir una estrategia de servicio que sirva para generar el máximo valor controlando riesgos y costos.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Políticas de Seguridad, de la Gestión de Seguridad⁶.

Las principales razones que respaldan la implementación del proceso de Gestión de Peticiones en la organización TI son:

- Proporciona al departamento comercial un acceso rápido y efectivo a servicios estándar. Esto mejora su productividad, la calidad de los servicios comerciales y los propios productos.
- Reduce la burocracia asociada al proceso de petición de acceso a servicios nuevos o ya existentes, reduciendo asimismo los costos.
- Incrementa el nivel de control sobre los servicios al centralizar la concesión de acceso a los mismos.
- Reduce costos al centralizar la negociación con proveedores respecto al acceso a los servicios, y también al reducir el costo del soporte". (ITIL®Foundation, 2011)

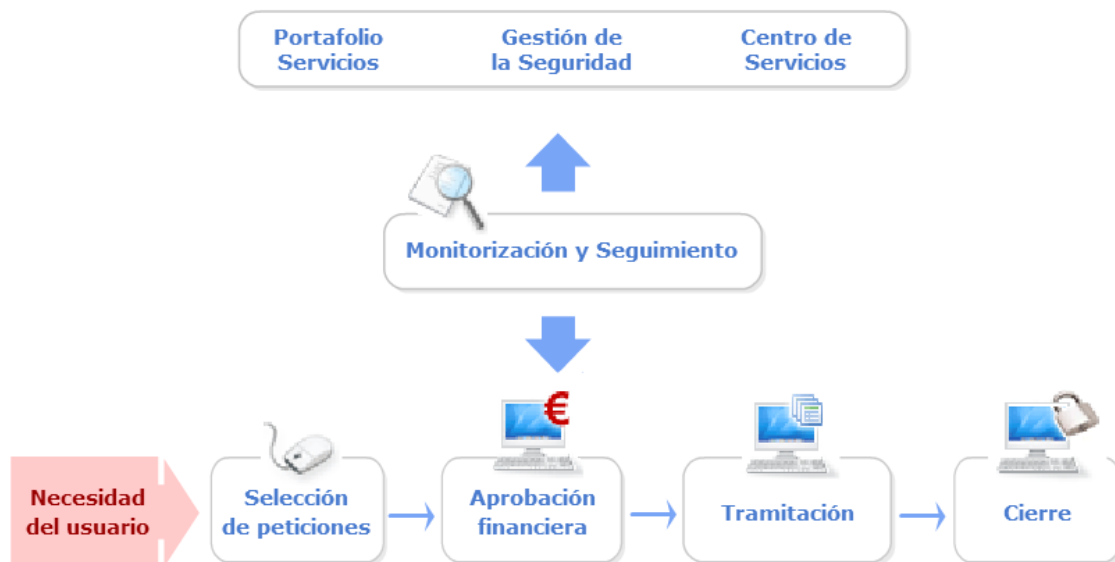


Figura 18 Manejo de Requerimientos

Fuente: http://itilv3.osiatis.es/operacion_servicios_TI/peticion_servicios_ti.php

Como podemos ver en la Figura 18 las actividades incluidas en el proceso de Gestión de Peticiones son:

- Selección de peticiones. Los usuarios, a través de las herramientas destinadas a tal fin por la Gestión de Peticiones, emiten sus peticiones conforme a una serie de tipologías predefinidas.
- Aprobación financiera de la petición. Dado que la mayoría de peticiones tienen implicaciones financieras, se considera su costo y se decide si tramitar la petición o no.
- Tramitación. La petición es referida por la persona o personas adecuadas según cada caso.

⁶ La Gestión de la Seguridad debe velar por que la información sea correcta y completa, esté siempre a disposición del negocio y sea utilizada sólo por aquellos que tienen autorización para hacerlo.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Cierre. Después de haber notificado al centro de servicio y que el cliente se encuentre conforme con la gestión se procede a cerrarla.

2.4.2. Manejo de incidentes

En esta sección nos basaremos en las normas ITIL para Gestión de Incidencias, y su autor no asegura que:

“Tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidente que cause una interrupción en el servicio.

La Gestión de Incidencias no debe confundirse con la Gestión de Problemas⁷, pues a diferencia de esta última, no se preocupa en encontrar y analizar las causas anteriores a un determinado incidente sino exclusivamente a restaurar el servicio. Sin embargo, es evidente, que existe una fuerte relación entre ambas.

Por otro lado, también es importante diferenciar la Gestión de Incidencias de la Gestión de Peticiones, que se ocupa de las diversas solicitudes que los usuarios plantean para mejorar el servicio, no cuando éste falla”. (ITIL®Foundation, 2011)

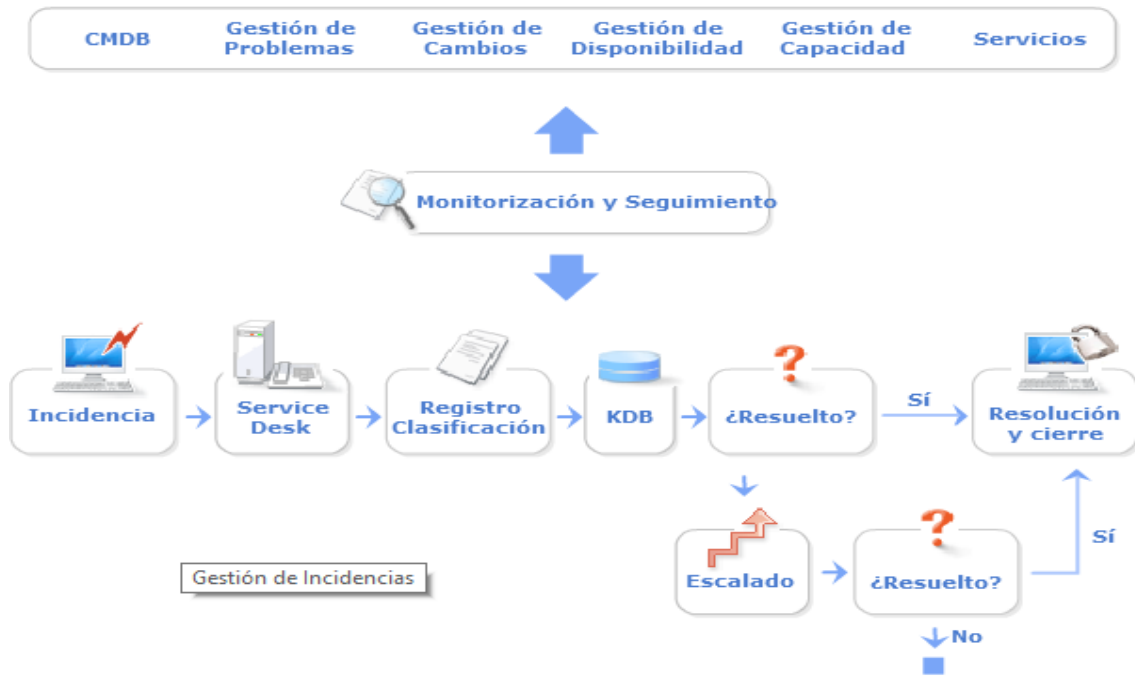


Figura 19 Manejo de Incidencias

Fuente: http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidencias.php

⁷ Algunas de las principales funciones de la Gestión de Incidentes son:

- Obtener total conocimiento de las causas reales como potenciales de los servicios de tecnología.
- Encontrar las soluciones más adecuadas a estas
- Solicitar los requerimientos de cambios necesarios para poder volver a tener una alta calidad en el servicio.
- La calidad debe ser indispensable y como medio para medirlo se harán revisiones post implementación para asegurar que los problemas se hayan solucionado sin crear más problemas.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

En la Figura 19 podemos ver muy resumidamente como se manejan los incidentes y los departamentos encargados en cada paso de este proceso.

Los objetivos principales de la Gestión de Incidencias son:

- Detectar cualquier alteración en los servicios TI.
- Registrar y clasificar estas alteraciones.
- Asignar el personal encargado de restaurar el servicio según se define en el SLA correspondiente.

Esta actividad requiere un estrecho contacto con los usuarios, por lo que el Centro de Servicios debe jugar un papel esencial en el mismo.

La Figura 20 resume el proceso de Gestión de Incidencias:

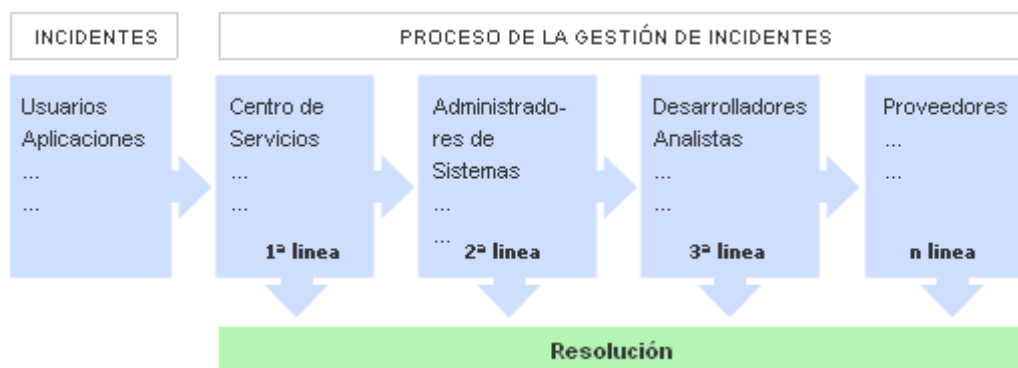


Figura 20 Gestión de Incidencias

Fuente: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/gestion_de_incidentes

Si bien el concepto de incidencia se la ve por cualquier persona como cualquier funcionamiento fuera de lo normal de los sistemas de hardware y software, según la guía de Soporte del Servicio de ITIL® una incidencia es:

“Cualquier cosa que no sea estandarizada en un servicio y que este causando, o potencialmente pueda causar, una ruptura o reducción de calidad del servicio”.

Por este motivo es bastante común que casi todo lo que pase por el centro de servicios puede ser considerado un incidente, sin embargo existen excepciones, como nuevas licencias, olvido de claves o usuarios, entre otras.

Es bueno tener en mente que si las modificaciones que se van a aplicar comprometen la infraestructura de algún sistema, no se consideran como un servicio regular o estándar y necesita ser tratado con un RFC, según la gestión de cambios.

Cuando se aplica una buena gestión de incidentes algunos beneficios pueden ser:

- Debido a la gestión de incidentes se mejora la productividad de parte del usuario.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Se cumplen los niveles que se especificaron previamente en los documentos de SLA⁸.
- Los procesos ya no se hacen a la ligera, si no que ya se controlan y los servicios tienen una monitorización.
- Se van a utilizar de una forma óptima los recursos con los que se cuenta.
- La satisfacción de los clientes mejora exponencialmente.

La gestión de incidencias mal llevada puede tener consecuencias negativas en el negocio, estas pueden ser:

- Los niveles de servicio pierden calidad.
- Se malgastan recursos, por ejemplo se asigna personal mucho más capacitado para la resolución de un caso o muchas personas.
- Se desaprovecha información extremadamente valiosa sobre las causas y efectos de las incidencias para futuros trabajos y avances.
- Debido a la lenta o mala atención de sus incidentes los clientes o usuarios quedan insatisfechos.

Los principales conflictos al momento de implementar la Gestión de Incidencias se puede resumir en:

- Los implicados en resolver no siguen los procedimientos previstos y se resuelven las incidencias sin ningún proceso o registro también podrían se escalan innecesariamente u omitiendo los protocolos ya establecidos.
- Al momento de existir una gran cantidad de incidencias entrantes, por la falta de un margen operativo, estas no se registran o lo hacen de una forma incorrecta provocando una mala operación de protocolos de clasificación y escalado.

⁸ Un acuerdo de nivel de servicio o ANS (en inglés Service Level Agreement o SLA), es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio.

3. CAPÍTULO 3: CASO DE ESTUDIO

En este capítulo se va a hablar del impacto que este proyecto va a tener en la empresa mediante sus objetivos, también se dará la justificación de las herramientas, técnicas y metodologías que se utilizaron para desarrollar el sistema.

3.1. Introducción

El tratamiento de la información de cada cliente con respecto a sus incidencias o requerimientos, así como los detalles de cada una de estos, al no existir un sistema exclusivo para su manejo, la información se maneja por medio de mails, chats u otros medios donde la información no siempre es fácilmente accesible, además de que la información exclusiva de las empresas debe estar manejada solo entre el cliente y la empresa que ofrece el servicio.

Actualmente, el personal de la empresa Soft Warehouse recibe las novedades de los clientes mediante hojas electrónicas o en algunos casos mediante mails o mensajes por el chat de sus mails, o mensajes de texto a sus teléfonos celulares, lo cual hace que la información sea manejada de una manera un poco informal y dificulta el acceso a ella cuando se necesita tener constancia de que esa novedad ya fue solventada o de como se lo hizo.

Esta manera de llevar la información no es la más óptima cuando se requiere una constancia del trabajo realizado al momento de dar soporte al sistema, además de que no se lleva un historial de cómo resolver los problemas que es realmente importante.

3.1.1. Objetivos

3.1.1.1. General

Analizar, diseñar y desarrollar un sistema para el manejo de requerimientos e incidencias mediante tickets para la compañía Soft Warehouse, y con esto lograr que el servicio de HelpDesk sea fácilmente accesible, Ofrezca un servicio de calidad consistente y homogénea, mantenga puntualmente informados a los usuarios y lleve un registro de toda la interacción con los mismos y sirva de soporte al negocio.

3.1.1.2. Específicos

- Analizar la situación actual del proceso a automatizar.
- Diseñar la mejor alternativa de solución para la automatización del proceso.
- Desarrollar un sistema que automatice el proceso crítico de la empresa, utilizando software libre.
- Implementar la aplicación.
- Capacitar el manejo del software.
- Revisar las mejores prácticas en procesos de manejo de requerimientos e incidentes.

3.2. Principales Actividades

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Las principales actividades relacionadas con la administración y ejecución adecuada del sistema los siguientes:

- Administración de entidades usadas:
 - Usuarios
 - Clientes
 - Sistema
 - Subsistemas
 - Unidad negocio
 - Módulo
 - Transacción

- Manejo de procesos críticos:
 - Gestión de requerimientos
 - Gestión de incidentes

- Manejo de procesos no críticos:
 - Reporte de Incidentes y Requerimientos en diferentes formatos
 - Consulta de tickets aprobados
 - Generación de historial de incidencias y como se resolvieron

3.3. Justificación de las herramientas seleccionadas para la construcción de la aplicación

Herramientas de Desarrollo

Hibernate

Hibernate⁹ es una herramienta de Mapeo objeto-relacional (ORM)¹⁰ para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML)¹¹ o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL¹².

⁹ Definición tomada de <http://hibernate.org/orm/>

¹⁰ Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

¹¹ eXtensible Markup Language (lenguaje de marcas extensible)

¹² GNU Lesser General Public License / Licencia de Software en la que cualquier persona tiene permisos de copia y distribución

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

La característica principal de Hibernate es su mapeo desde clases Java hacia tablas de bases de datos (y desde tipos de dato Java a tipos de dato SQL). Hibernate facilita también un mecanismo de consulta y manejo de datos. Hibernate genera llamadas SQL, y ahorra el trabajo de conversión de resultados a objetos.

Java

Para este proyecto se escogió el lenguaje Java, ya que con el podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cobra cada vez más importancia tanto en el ámbito de Internet como en de la informática en general. Fue desarrollado por la compañía Sun Microsystems actualmente adquirida por ORACLE, con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más avanzadas.

Es un lenguaje independiente de la plataforma, eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier sistema operativo del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que estos se entiendan perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas van a tener acceso con ordenadores distintos. También Java se caracteriza por estar desarrollándose incluso para distintos tipos de dispositivos además de los ordenadores como móviles, agendas electrónicas y en general para cualquier cosa que se le ocurra a la industria.

JavaScript

JavaScript ¹³ es un lenguaje de programación interpretado, fundado en prototipos, imperativo, dinámico y orientado a objetos.

Se utiliza principalmente en el lado del cliente, elaborado como parte de un navegador web haciendo posible mejoras en la interfaz de usuario y páginas dinámicas. Existe también una forma de JavaScript que se encuentra en el servidor o SSJS por sus siglas en inglés, Server-side JavaScript o SSJS. Su uso es también bastante significativo en aplicaciones que no involucran al internet como documentos PDF y aplicaciones de escritorio.

JavaScript se construyó con una sintaxis bastante parecida al lenguaje C, aunque incorpora nombres y formas de estresar del lenguaje Java. Sin embargo Java y JavaScript no están relacionados y tienen tanto forma de escribir y propósitos diferentes.

Todos los navegadores de la actualidad usan un intérprete del código JavaScript que está en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM)¹⁴.

¹³ Definición tomada de "JavaScript" - Wikipedia

¹⁴ Interfaz que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases

PostgreSQL

En un inicio se pensaba comenzar el proyecto con MySQL ya que por mucho tiempo fue el motor más popular; pero hoy es propiedad de Oracle y esto limita su evolución, pero se terminó decidiendo que PostgreSQL era mucho mejor por varios motivos entre estos, PostgreSQL es gratuito y libre, además de que hoy nos ofrece una gran cantidad de opciones avanzadas. De hecho, es considerado el motor de base de datos más avanzado en la actualidad.

Otra característica que gusta en la comunidad de desarrolladores sobre PostgreSQL es el Hot-Standby. Este permite que los clientes hagan búsquedas (sólo de lectura) en los servidores mientras están en modo de recuperación o espera. De esta manera, podemos hacer tareas de mantenimiento o recuperación sin bloquear completamente el sistema.

PostgreSQL aporta mucha flexibilidad a los proyectos. Por ejemplo, nos permite definir funciones personalizadas por medio de varios lenguajes. Algunos son:

1. PL/pgSQL
2. PL/Tcl
3. PL/Perl
4. PL/Python
5. PL/PHP
6. PL/Ruby
7. PL/Java

Otra ventaja de PostgreSQL es que está disponible para muchas plataformas y ofrece el código fuente desde el sitio oficial. Algunas de las construcciones oficiales son:

- Mac OS X
- Windows
- Solaris
- Red Hat
- Debian
- Ubuntu

La herramienta oficial para administrar las bases de datos de PostgreSQL es pgAdmin este nos permite desde hacer búsquedas SQL hasta desarrollar toda la base de datos de forma muy fácil e intuitiva, directamente desde la interfaz gráfica.

Herramientas de Diseño

Vertabelo

Es una herramienta que puede ser usada para diseñar el modelo de la base de datos, una de las mejores soluciones es que puede dar el poder del diseño de una base de datos en línea y de forma visual.

Se tomó la decisión de usar esta herramienta por varias características ofrecidas por el mismo entre algunas están:

Características de colaboración.

Basta con proporcionar un correo electrónico del miembro del equipo para compartir modelos y trabajar juntos en el diseño de la base de datos. También puede crear un enlace público al modelo de base de datos.

Equipos de apoyo distribuidos

Que los equipos estén repartidos por todo el mundo es una realidad cada vez más creciente. Usando Vertabelo todos pueden colaborar con su equipo en todas partes en cualquier momento.

Tiene para ser utilizada una interfaz de usuario muy buena para crear su esquema de base de forma rápida y sencilla. Puede editar su modelo en una sola ventana y ahorrar tiempo con atajos de teclado.

Las bases de datos que soporta son:

- PostgreSQL 9.x
- MySQL 5.5
- Microsoft SQL Server 2012
- SQLite 3.7.x
- Oracle 11g
- IBM DB2 9.7
- HSQLDB 2.3

PowerDesigner

Para este proyecto también se va a usar esta herramienta ya que se necesita no solo modelar la base de datos sino también para poder lograr un modelamiento de los procesos de negocios. PowerDesigner es la herramienta líder en la industria de tecnología, en el campo del modelamiento y en varios campos más, como gestión de metadatos para la arquitectura de datos, arquitectura de la información y la arquitectura de la empresa.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

También nos trae técnicas de análisis de impacto de gran alcance, la gestión del cambio en la etapa de diseño y la gestión de metadatos de la empresa.

El repositorio de metadatos de PowerDesigner mejora la colaboración y la comunicación entre todas las partes interesadas del proyecto, lo que facilita una respuesta más rápida al realizar cambios al proyecto y mejora la agilidad del negocio.

En resumen este producto nos permite:

- Aumenta la productividad de la empresa: Alinea el negocio y TI para ayudar a mejorar la productividad general.
- Es compatible con entornos abiertos.
- Incluye funciones de personalización: Es altamente personalizable para ayudar a hacer cumplir las normas.
- Diseñado para la complejidad de las empresas: Facilita las implementaciones de arquitectura empresarial mediante la captura de forma intuitiva las intersecciones entre todas las capas de arquitectura y perspectivas de la Empresa.
- Proporciona herramientas de análisis: Aumenta la agilidad del negocios con su funcionalidad de conectar y sincronizar

Técnicas de Desarrollo

Se tomará la técnica de desarrollo RAD (Diseño Rápido de Aplicaciones) ya que el tiempo para entregar del producto es corto, y los requisitos y fronteras del software son limitados. Esta técnica es buena, pues permite la interacción directa con el cliente final y la utilización de prototipos para agilizar el desarrollo. La necesidad de un grupo de desarrolladores es dependiente de la complejidad del producto, por lo cual, para este caso es necesario un solo desarrollador.

La técnica de desarrollo RAD se basa en el ciclo de vida general del desarrollo de software, tomando en cuenta que se trabaja con prototipos y con la interacción cara a cara con el usuario, se define el ciclo de la siguiente manera:

Inicio

- Análisis de requerimientos
- Especificaciones de funcionalidad
- Arquitectura
- Diseño

Elaboración

- Prototipo ejecutable
- Lista de riesgos

Construcción

- Implementación
- Pruebas

Transición

- Seguridades

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- Despliegue
- Mantenimiento

4. CAPÍTULO 4: LEVANTAMIENTO DE PROCESO

En este capítulo se realizará el modelamiento del mapa de procesos y de la cadena de valor. Además se describirán los procesos y subprocesos que integran la cadena de valor, por último también se desarrollara la tabla de proveedor, recursos, proceso y cliente o SIPOC¹⁵ por sus siglas en ingles de los procesos considerados clave para la administración de incidentes y requerimientos

4.1. Mapa de procesos

El área de desarrollo de la empresa Soft Warehouse, se encarga entre otras cosas de la administración de incidentes y requerimientos que pueden tener los clientes en forma totalmente ininterrumpida, las 24 horas del día los 7 días de la semana. Los procesos que intervienen en la administración de incidentes y requerimientos se describen a continuación:

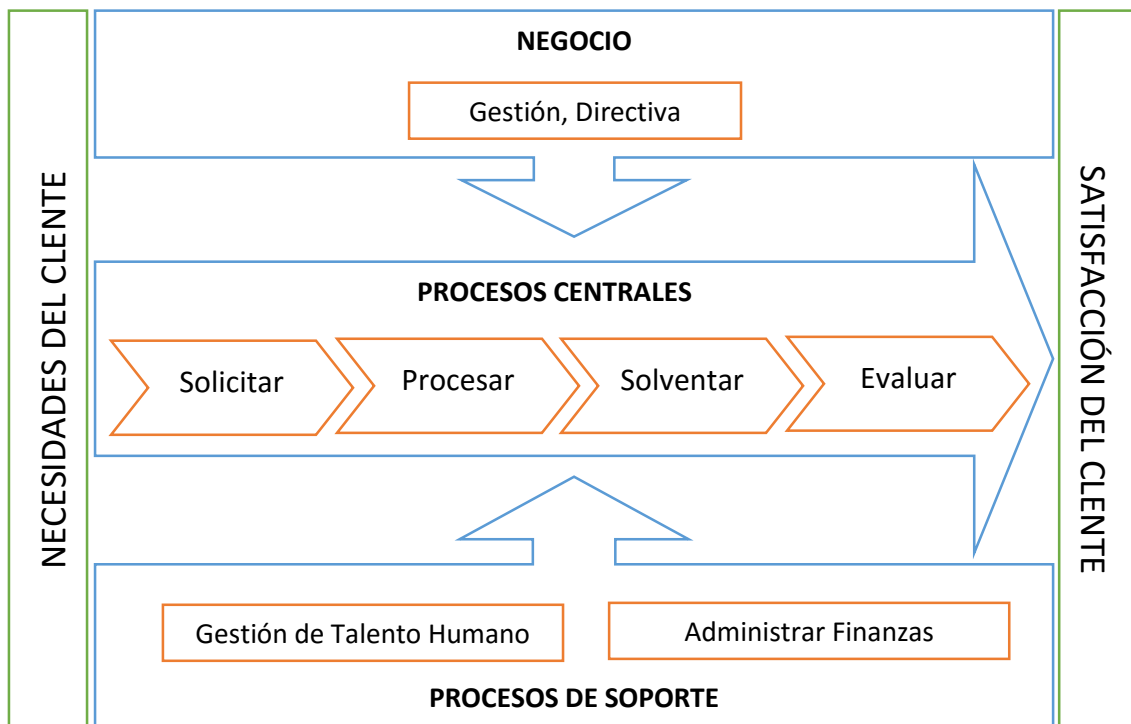


Figura 21 Mapa de Procesos
Autor: Santiago Gálvez

4.1.1 Cadena de Valor

La cadena de valor, es un modelo teórico que permite describir el desarrollo de las actividades de una organización empresarial generando valor al cliente final, descrito y popularizado por Michael Porter, nuestra cadena consta de 4 macro procesos los cuales son solicitar, procesar, solventar y evaluar.

¹⁵SIPOC, por sus siglas en inglés Supplier Inputs Process Outputs Customers es un diagrama que representa gráficamente un proceso de gestión.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS



En la siguiente tabla se presentara las principales actividades de los macro procesos ya antes mencionados en la cadena de valor.

Solicitar	<p>Definir incidencia o requerimiento</p> <ul style="list-style-type: none"> • Identificar el problema que presenta el software. • Intentar resolver la incidencia • Registrar en Gerencia de Aseguramiento de Calidad (GAC¹⁶) <p>Notificar Incidencia</p> <ul style="list-style-type: none"> • Asignar la prioridad • Detallar sistema, módulo y transacción • Generar evidencias del incidente • Definir necesidades del requerimiento • Crear ticket • Notificar al responsable
Procesar	<p>Analizar incidencia o requerimiento</p> <ul style="list-style-type: none"> • Verifica la competencia del incidente o la viabilidad del requerimiento • Establece un estatus • Incluir recomendaciones • Asignación de responsable • Notificar al consultor asignado <p>Presupuestar</p> <ul style="list-style-type: none"> • Verifica la competencia del incidente o la viabilidad del requerimeinto • Definir tiempo y encargados de desarrollo • Realizar un presupuesto • Aprobar solución
Solventar	<p>Solucionar Problema</p> <ul style="list-style-type: none"> • Investigar posibles soluciones • Efectúa la solución • Registrar cambios • Cambia el estado del ticket • Notificar al cliente
Evaluar	<p>Verificación de resolución</p> <ul style="list-style-type: none"> • Prueba • Rechazo o aprobación <p>Análisis de resultados</p> <ul style="list-style-type: none"> • Generar reportes • Evaluar • Toma de decisiones operativas • Recibe notificación

Tabla 2 Cadena de Valor
Autor: Santiago Gálvez

¹⁶ GAC es un documento donde se registra toda novedad con respecto a la calidad de un producto o servicio en un negocio, esta ayuda a que el grupo o gerente de calidad pueda garantizar la calidad de los ya mencionados.

4.2. SIPOC

Esta se puede resumir como una visión muy resumida y explicativa de todo el proceso de una empresa bajo una perspectiva particular, la cual viene dada por el significado de sus siglas Supplier-Inputs-Process-Output-Customers (Proveedor-Entrada-Proceso-Salida-Cliente).

- Proveedor (Supplier): persona que aporta recursos al proceso
- Recursos (inputs): todo lo que se requiere para llevar a cabo el proceso. Se considera recursos a la información, materiales e incluso, personas.
- Proceso (Process): conjunto de actividades que transforman las entradas en salidas, dándoles un valor añadido.
- Cliente (Customers): la persona que recibe el resultado del proceso. El objetivo es obtener la satisfacción de este cliente.

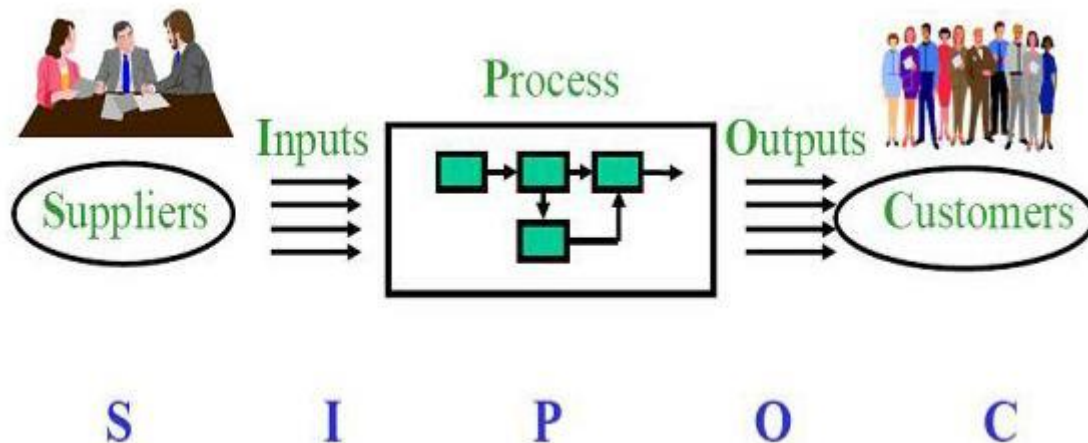


Figura 22 Diagrama SIPOC
Fuente: <http://community.asq.org>

De manera resumida los pasos a realizar para elaborar un Diagrama SIPOC pueden ser:

- Identificar los procesos de gestión
- Establecer las entradas del proceso, los recursos necesarios
- Establecer los proveedores de estas entradas al proceso
- Definir las salidas del proceso
- Establecer quién es el cliente de cada una de las salidas obtenidas

A continuación se diagramaran cada uno de los subprocesos mencionados dentro de los macro procesos en la tabla 2, con el modelo SIPOC el cual nos ayudara a analizar el proceso de una manera más detallada reconociendo al respectivo proveedor, así como también identificando todas las entradas y salidas del proceso, además nos permite determinar los clientes vinculados a cada paso del proceso.

4.2.1. Solicitar

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4.2.1.1. Definir Incidencia o Requerimiento

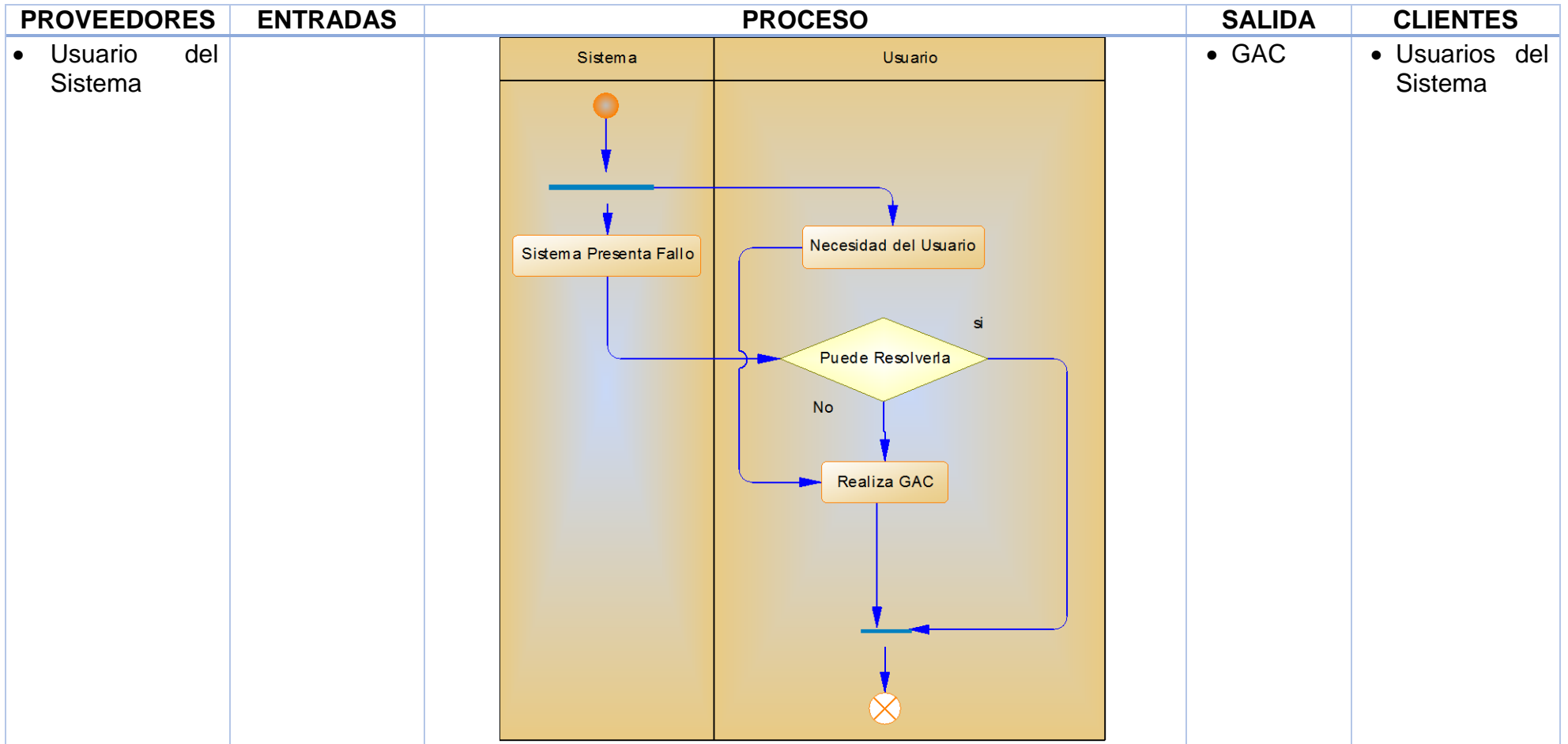


Tabla 3 Diagrama SIPOC Definir Incidencia o Requerimiento
 Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4.2.1.2. Notificar Incidencia

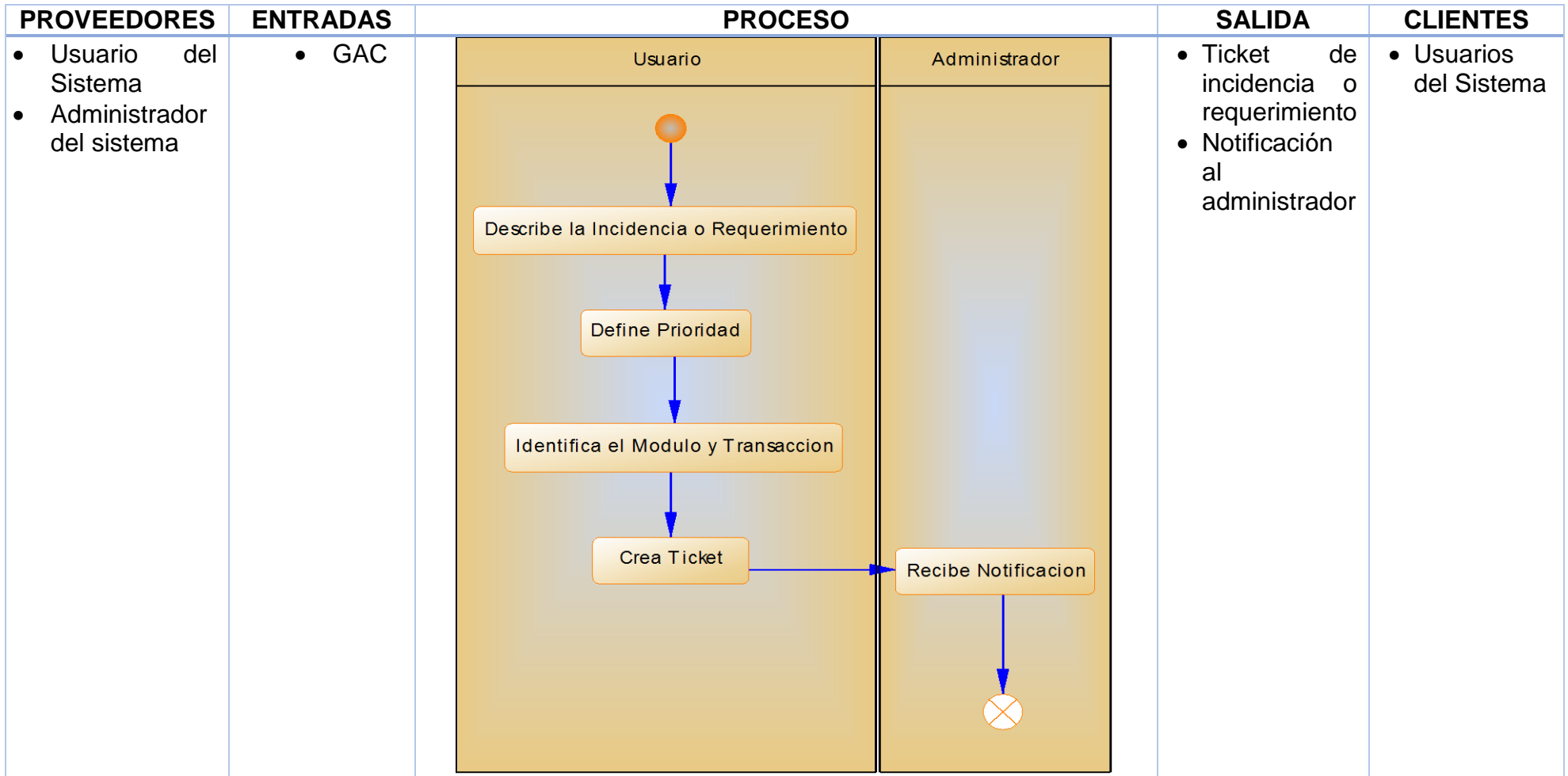


Tabla 4 Diagrama SIPOC Notificar Incidencia
 Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4.2.2. Procesar

4.2.2.1. Analizar incidencia o requerimiento

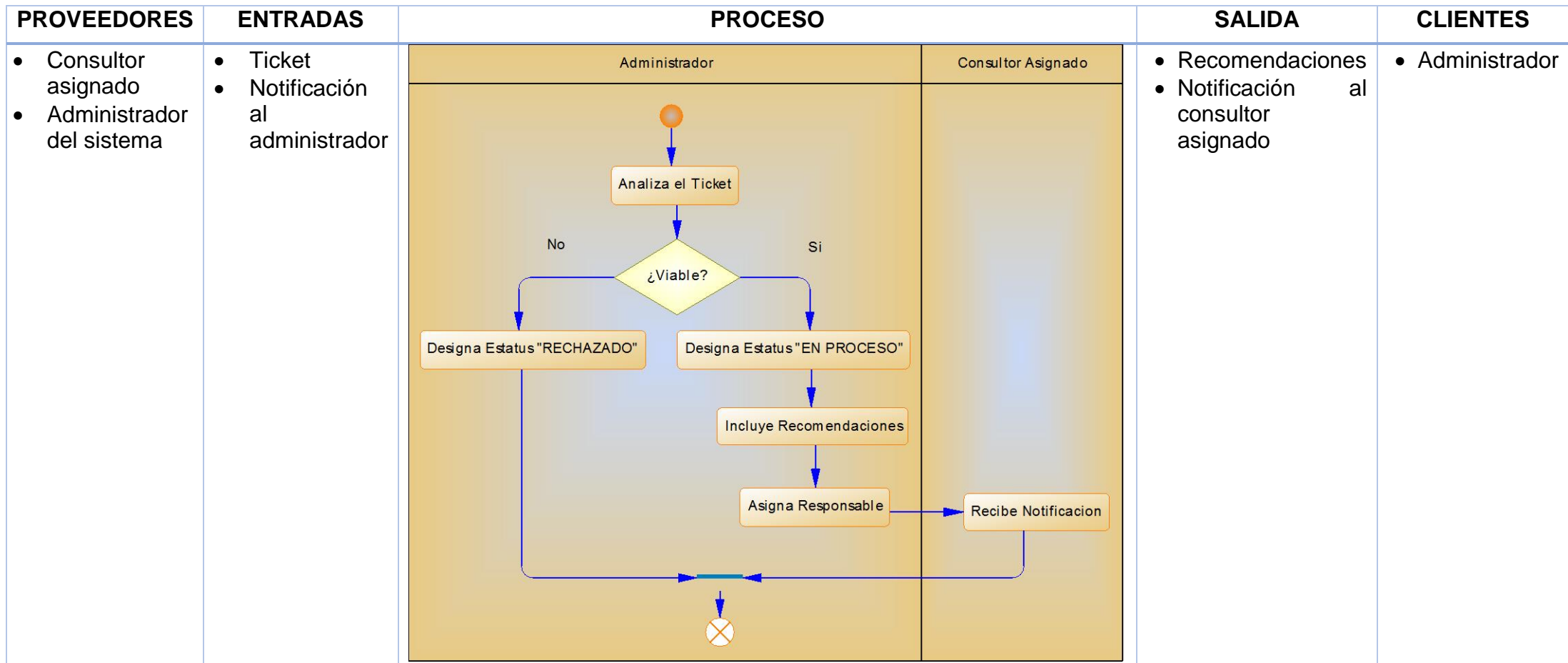


Tabla 5 Diagrama SIPOC Analizar incidencia o requerimiento
 Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4.2.2.2. Presupuestar

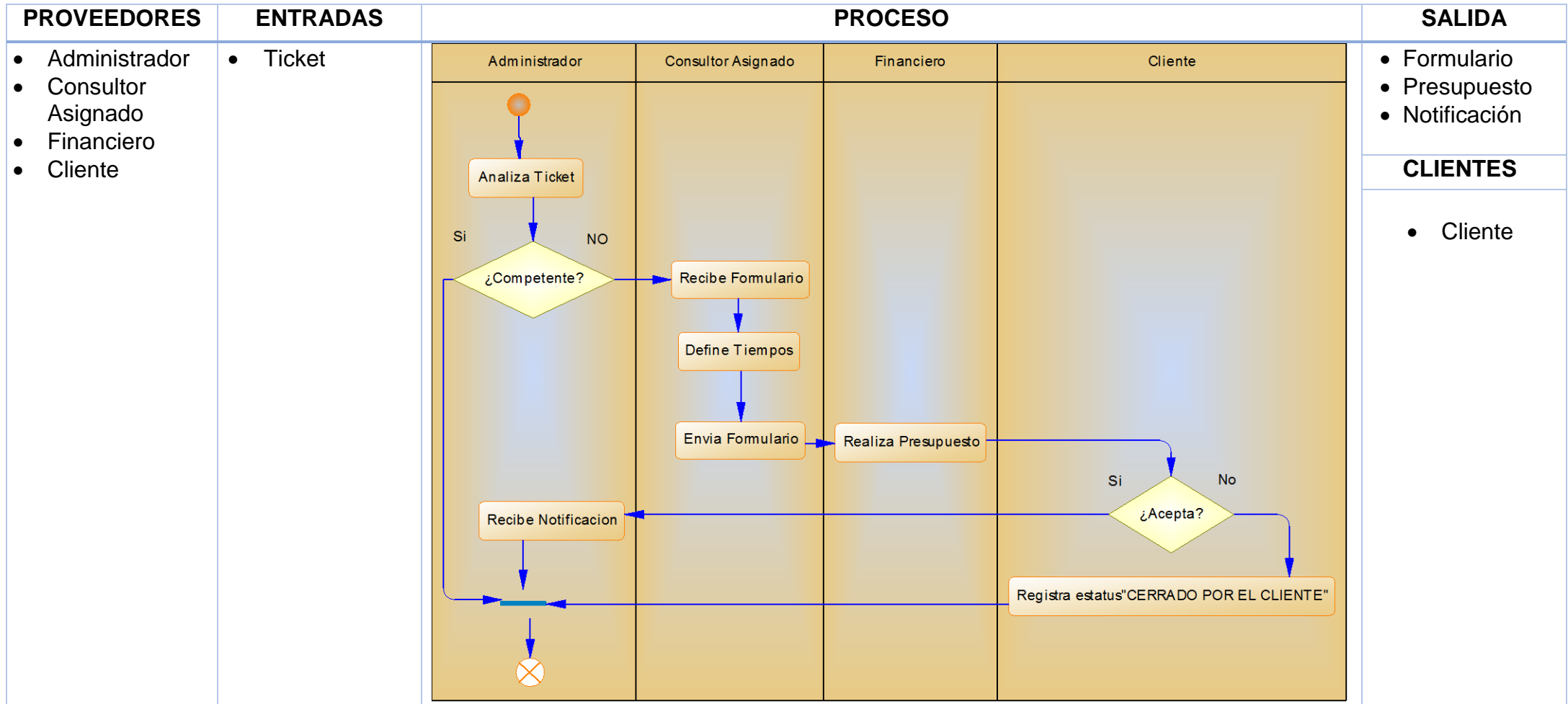


Tabla 6 Diagrama SIPOC Presupuestar
Autor: Santiago Gálvez

4.2.3. Solventar

4.2.3.1. Solucionar Problema

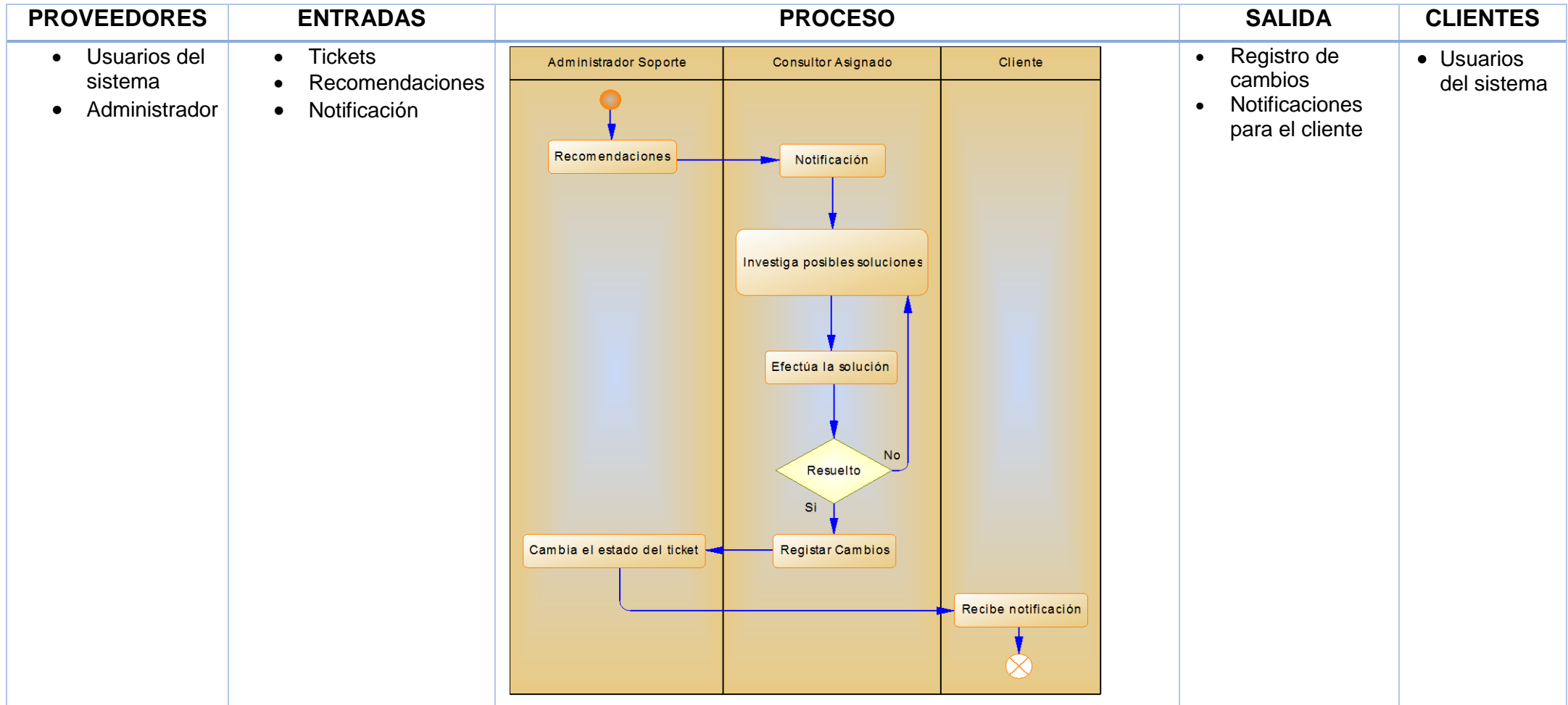


Tabla 7 Diagrama SIPOC Solucionar Problema
 Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

4.2.4. Evaluar

4.2.4.1. Verificación de resolución

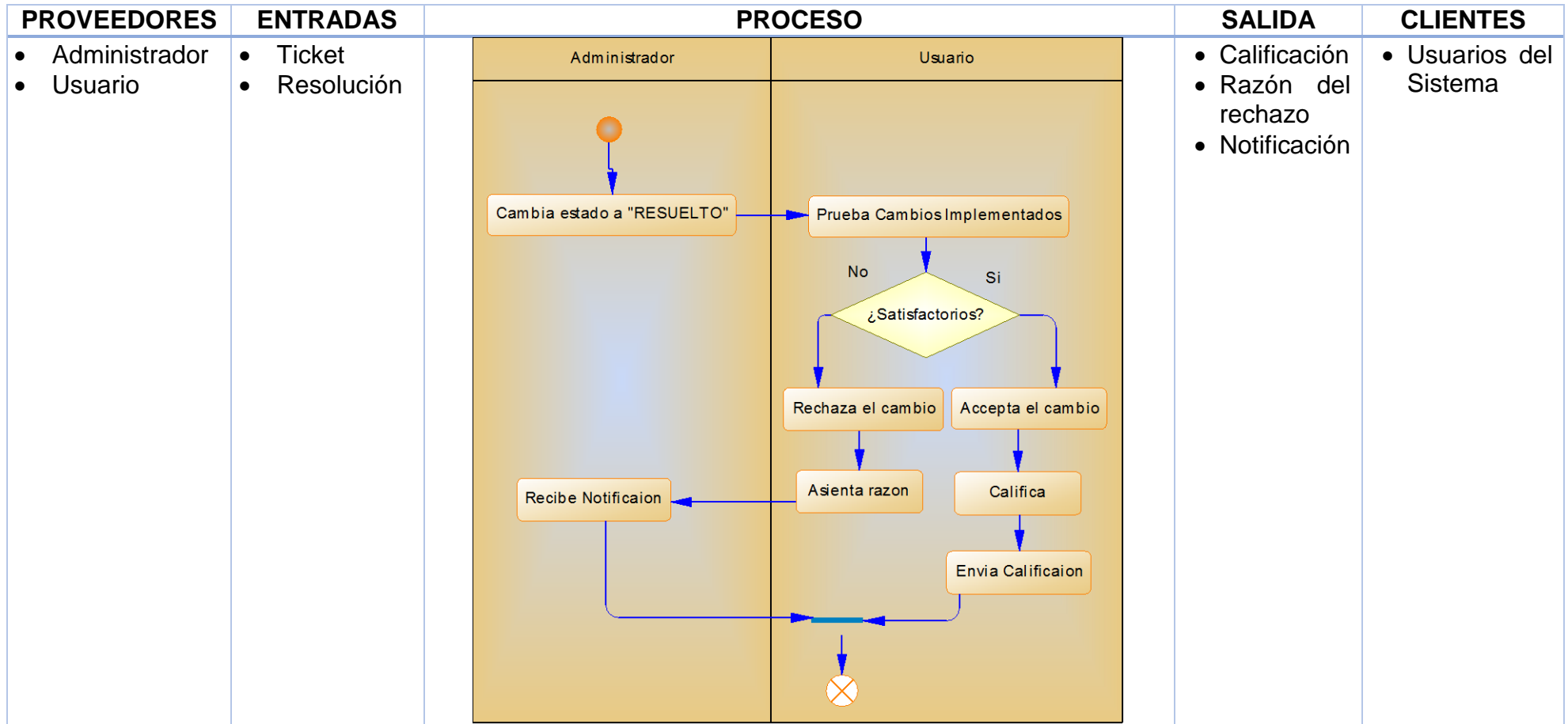


Tabla 8 Diagrama SIPOC Verificación de resolución
Autor: Santiago Gálvez

4.2.4.2. Análisis de resultado

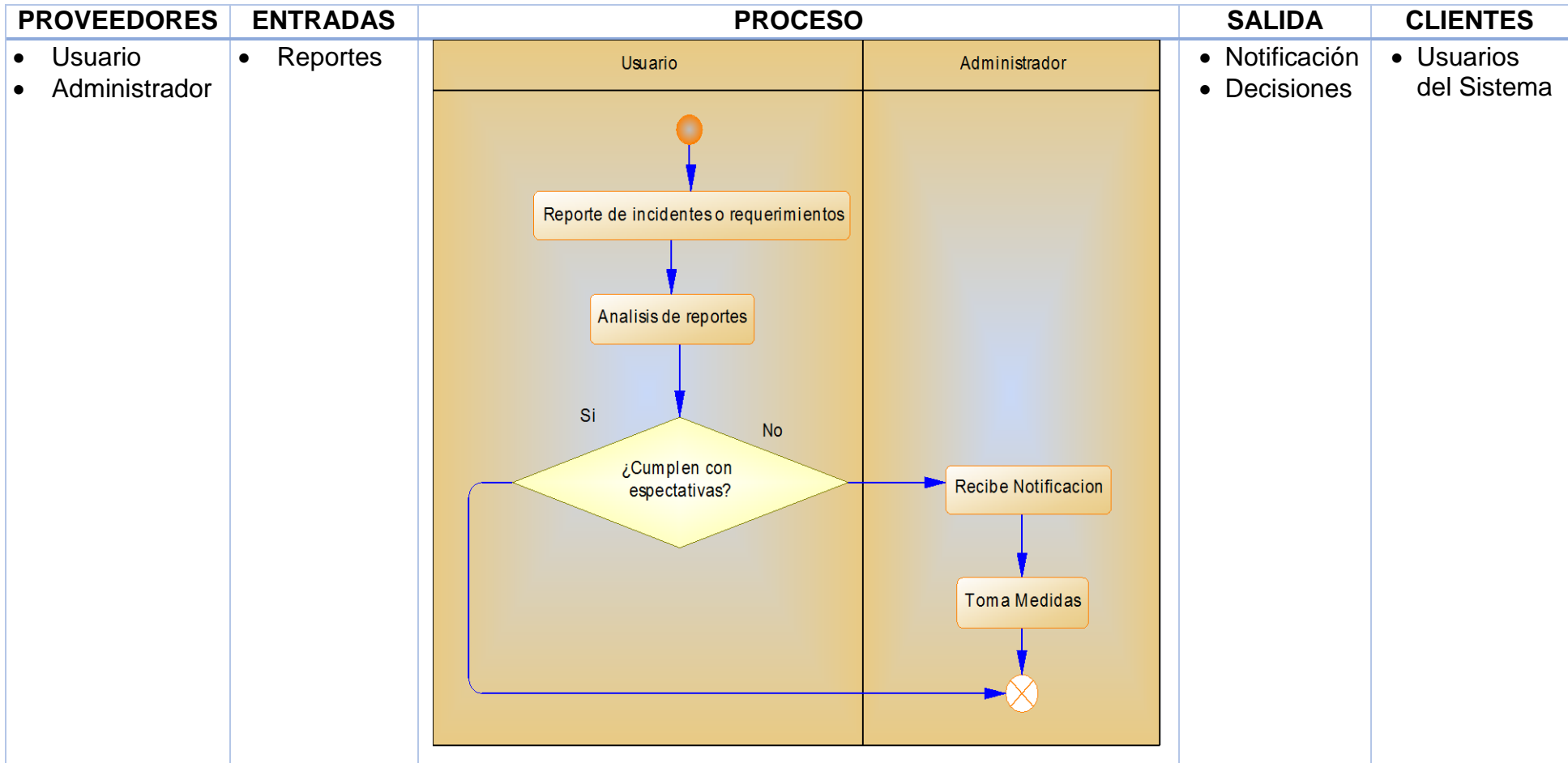


Tabla 9 Diagrama SIPOC Análisis de resultado
 Autor: Santiago Gálvez

5. CAPÍTULO 5: CONSTRUCCIÓN DEL PRODUCTO

Para la realización del sistema se usará el ciclo de vida RAD, el cual consta de las 4 fases básicas de desarrollo de aplicaciones.

- Inicio
- Elaboración
- Construcción
- Transición

En la fase **inicial**, se recolectan los requerimientos de funcionalidad de la aplicación. Luego, con la suficiente información obtenida, se generan los diagramas UML que muestran la arquitectura inicial de la aplicación antes de elaborarla. En esta fase también se muestran las imágenes (diseño) de cada ventana mostrada en la aplicación.

En la fase de **elaboración** se presentará un prototipo básico ejecutable que mostrará las funcionalidades que tendrá el producto luego de su construcción, además se filtrará la lista de posibles riesgos en el transcurso del ciclo de vida.

En la fase de **construcción** se hará uso de las técnicas de desarrollo de software aprendidas durante la carrera, implementando interfaces fácilmente reproducibles y herencias. Bajo cada módulo construido se procederá a hacer las respectivas pruebas de funcionalidad y diseño. Aquí se almacenarán los requerimientos faltantes del ciclo y se sumarán al siguiente ciclo.

Con esto se acaba el primer ciclo de construcción, preparándose para el siguiente análisis de requerimientos y elaboración de los diagramas UML respectivos al caso.

En la fase de **transición**, se asume que los ciclos de vida de RAD se han completado exitosamente, para luego preocuparnos por detalles finales durante la entrega del producto, como la seguridad de los datos y accesos, costos de construcción, instalación y mantenimiento, y finalmente, despliegue y entrega del producto al cliente final.

5.1. Inicio

5.1.1. Análisis de requerimientos

Es importante que al iniciar se consiga contar con un acuerdo preciso de lo que va a ser el producto y así tener un mecanismo que proteja los requerimientos originales de los desacuerdos que le cliente pueda encontrar.

Eso será posible con el documento de requerimientos claro y preciso, con el cual se podrán administrar los cambios y proteger tal documento contra un cliente que insista en interpretar de manera diferente una función que ya ha sido diseñado.

En el caso de este proyecto el sistema que se elaboró es un servicio extra que la empresa implemento para obtener una mayor satisfacción de sus clientes, por esta razón este documento se realizó entre los encargados del área de desarrollo y el gerente de la empresa.

Objetivo

Con frecuencia se suele hacer cambios a los requerimientos, generalmente, los usuarios no pueden saber lo que necesitan en forma precisa hasta que tratan de usar el producto terminado. Sin embargo, cuando se introduce un sistema nuevo se cambia la manera

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

en que ellos trabajan. Conforme el trabajo de desarrollo progresa empiezan a apreciar la manera en cómo afectará su ambiente, mientras más claro es eso para los usuarios, pensarán en nuevas funciones y características, por ésto, los requerimientos constantemente cambian hasta que se congelan en un producto.

La parte difícil del proceso de requerimientos de software es entender que es lo que los usuarios creen que necesitan y ayudarlos a definir esas necesidades en términos de funciones que generen un producto útil. El siguiente paso es realizar las especificaciones de requerimientos de software que represente el acuerdo común entre los usuarios y el equipo de trabajo acerca de lo que necesitan.

Es importante iniciar con un acuerdo preciso de lo que es el producto y así tener un mecanismo que proteja los requerimientos originales de los desacuerdos encontrados; eso será posible con un documento de requerimientos claro y preciso, con el cual se podrán administrar los cambios y proteger tal documento contra un cliente que insista en interpretar de manera diferente una función que ya ha sido diseñada.

Definición General del Sistema

Antecedentes

Tema: Desarrollo de un sistema para la administración de requerimientos e incidencias mediante tickets

La empresa se ha visto en la necesidad de desarrollar este sistema para mejorar la atención de los problemas o nuevas funcionalidades que los clientes puedan tener en el software que poseen, también para poder constar de un historial de incidencias y un manejo más confidencial de la información sensible de los clientes.

Requerimientos funcionales

- **F0. Ingreso al Sistema**

Esta opción permitirá a los usuarios poder ingresar al sistema con su usuario y contraseña previamente solicitada, si el usuario ingresa por primera vez, el sistema pedirá que ponga una nueva clave después de ingresar la primera.

- **F1. Administración de Incidentes**

La opción de Administración de Incidentes permitirá llevar un control de los tickets de incidencias que puedan surgir en los servicios que ofrece la empresa, administrando la información necesaria para poder saber dónde se originó como sistema, módulo, transacción, mensaje de error, descripción y la prioridad del ticket dado por el orden del mismo.

En la administración de incidentes, se podrán realizar las siguientes actividades:

- **F1.1** Ingreso de incidencia
- **F1.2** Consultar Incidencia
- **F1.3** Modificar Incidencia
- **F1.4** Aceptar Incidencia
- **F1.5** Actualizar estatus

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

- **F1.6** Asignar Consultor
- **F1.7** Ingresar Notas Adicionales

- **F2. Administración de Requerimiento**

La opción de Administración de Requerimientos permitirá llevar un control de los tickets de requerimientos que puedan surgir como nuevas necesidades de los clientes para personalizar más su sistema, administrando la información necesaria para poder saber dónde se necesita el cambio como sistema, módulo, transacción, descripción y la prioridad del ticket dado por el orden del mismo.

En la administración de requerimientos, se podrán realizar las siguientes actividades:

- **F2.1** Ingreso de requerimiento
- **F2.2** Consultar requerimiento
- **F2.3** Modificar requerimiento
- **F2.4** Aceptar requerimiento
- **F2.5** Actualizar estatus
- **F2.6** Asignar Consultor
- **F2.7** Ingresar Notas Adicionales

- **F3. Administración de Tickets Resueltos**

La opción de Administración de Tickets Resueltos permitirá a los usuarios hacer consultas por parámetros de los tickets que ya han sido aplicados por la empresa y aceptados por el cliente.

En la administración de tickets resueltos, se podrán realizar las siguientes actividades:

- **F3.1** Consulta Tickets

- **F4. Administración de Reportes**

La Administración de Reportes permitirá a los usuarios generar reportes de los tickets de incidencias o requerimientos según los parámetros ingresados, para generar un reporte lo único que se necesita obligatoriamente es un rango de fechas.

En la administración de reportes, se podrán realizar las siguientes actividades:

- **F4.1** Generar Reporte

En la Figura 23 se presentará el Diagrama General de Casos de Uso del sistema:

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

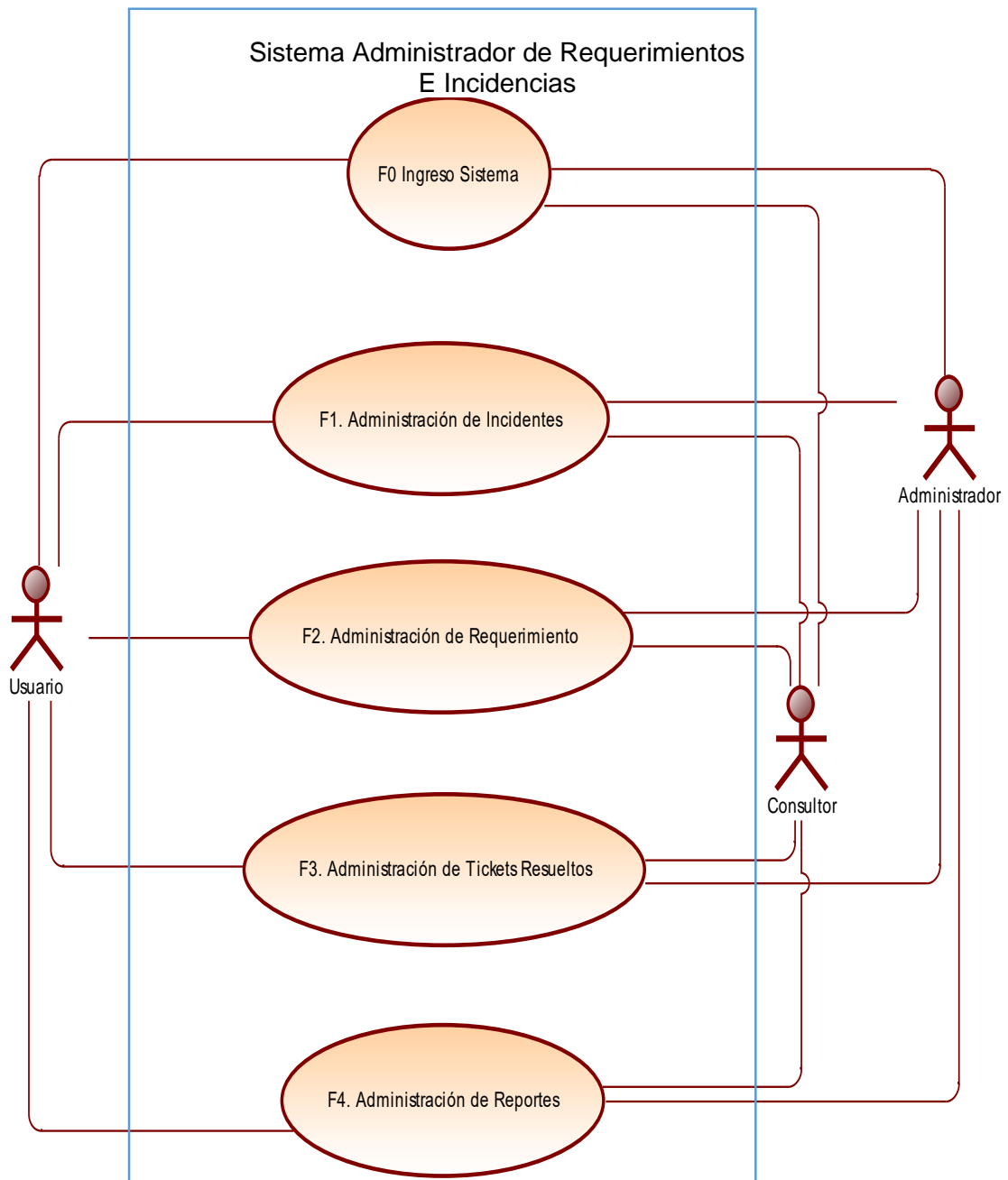


Figura 23 Diagrama General de Casos de Uso
Autor: Santiago Gálvez

5.1.2. Especificaciones de funcionalidad

En esta sección se pasa a detallar cada una de las funcionalidades anteriormente expresadas en la Figura 23:

F0. Ingreso al Sistema



Figura 24 Diagrama F0. Ingreso al sistema
Autor: Santiago Gálvez

Descripción:

Los usuarios podrán realizar su ingreso al sistema

Actores:

- Usuario
- Administrador
- Consultor

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta una ventana donde los usuarios podrán ingresar sus credenciales para su verificación	
2	El usuario ingresa su nombre de usuario	
3	El usuario ingresa su contraseña	E1
4	El usuario presiona el botón Ingresar	E2

Excepciones

Código	Descripción	Alternativa
E1	Datos incorrectos	Reintentar ingresar el usuario y la contraseña del usuario
E2	Error de conexión	Verificar la conexión a internet

F1. Administración de incidentes

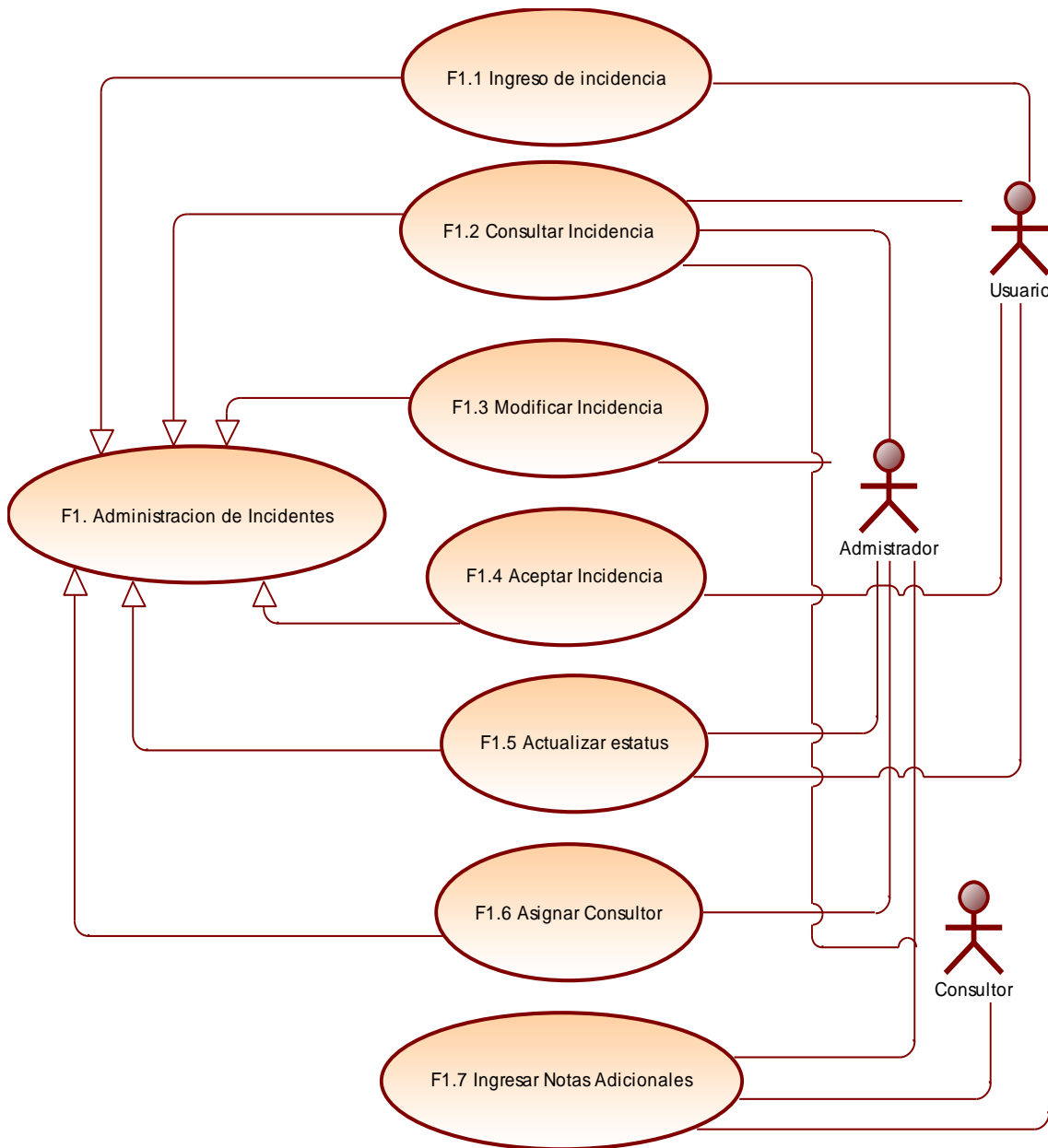


Figura 25 Diagrama F1. Administración de incidentes
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F1.1 Ingreso de Incidencia

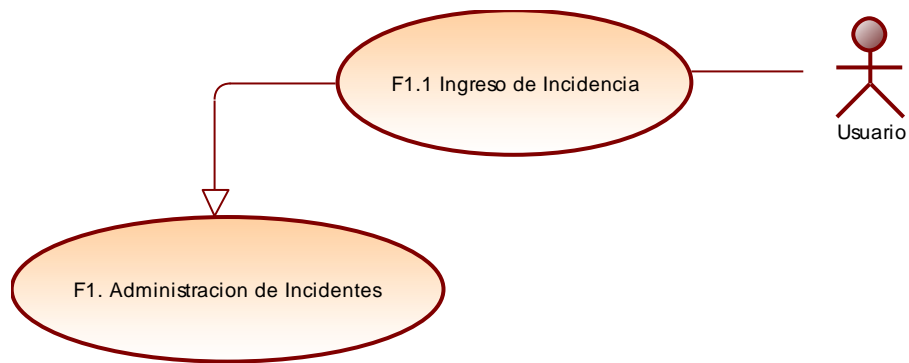


Figura 26 F1.1 Ingreso de Incidencias
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso los usuarios podrán realizar el ingreso de nuevos tickets de incidentes en el sistema

Actores:

- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El usuario selecciona el botón de NUEVO TICKET	
3	El sistema presenta el formulario para ingresar una incidencia	
4	El usuario ingresa los datos del formulario	E1
5	El usuario presiona GUARDAR	E2
6	El sistema almacena los datos	

Excepciones

Código	Descripción	Alternativa
E1	Campos obligatorios vacíos	Llenar los campos obligatorios.
E2	Error de conexión	Verificar la conexión a internet

F1.2 Consultar Incidencia

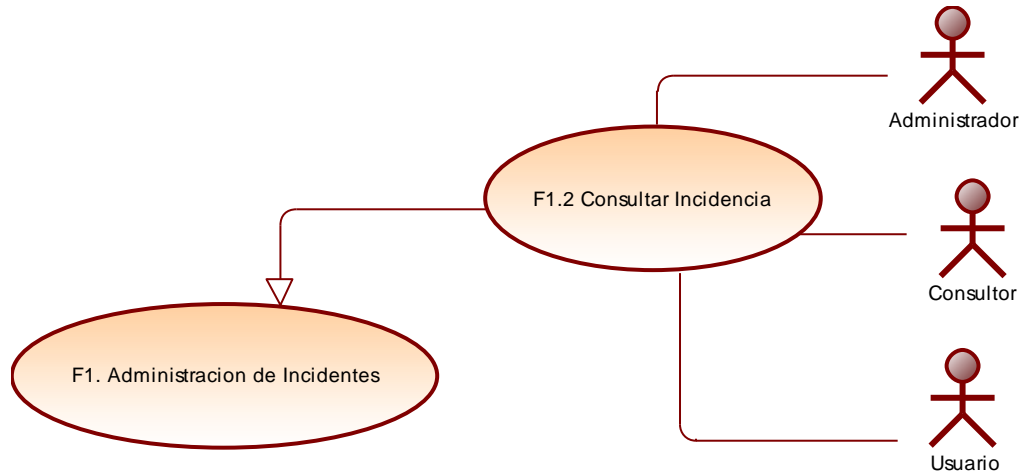


Figura 27 F1.2 Consultar Incidencia
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso los administradores, consultores y los usuarios podrán realizar la consulta de las incidencias

Actores:

- Usuario
- Administrador
- Consultor

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El usuario selecciona los parámetros de búsqueda	
3	El sistema presenta una lista de incidencias que cumple con los parámetros solicitados	E1

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet

F1.3 Modificar Incidencia

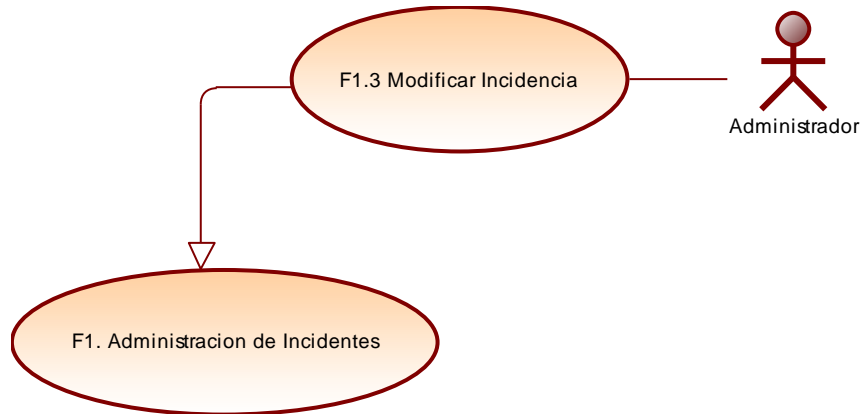


Figura 28 F1.3 Modificar Incidencia
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el administrador cuando sea necesario podrá cambiar o aumentar los datos que desea del ticket de incidencia para completar o actualizar su información.

Actores:

- Administrador

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El administrador modifica los datos necesarios	
3	El administrador pulsa el botón GUARDAR	E1
4	El sistema almacena los datos ingresados.	E2

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet
E2	Error al grabar	Contactarse con el soporte técnico

F1.4 Aceptar Incidencia

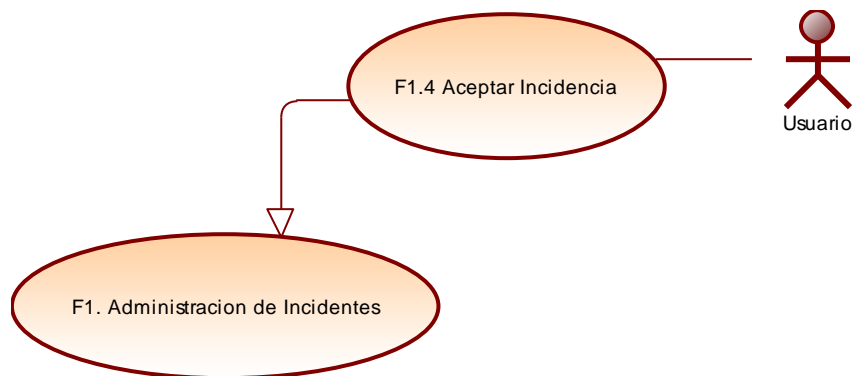


Figura 29 F1.4 Aceptar Incidencia
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el usuario podrá aceptar los cambios o soluciones aplicadas para solventar la incidencia dando por finalizado el seguimiento de la misma en el proceso

Actores:

- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El usuario pulsa el botón Aceptado	E1
3	El sistema presenta la pantalla encuesta de satisfacción	
4	El usuario selecciona un nivel de satisfacción	
5	El sistema habilita el botón enviar	
6	El usuario pulsa el botón enviar	E2
7	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet
E2	Error al grabar	Contactarse con el soporte técnico

F1.5 Actualizar estatus

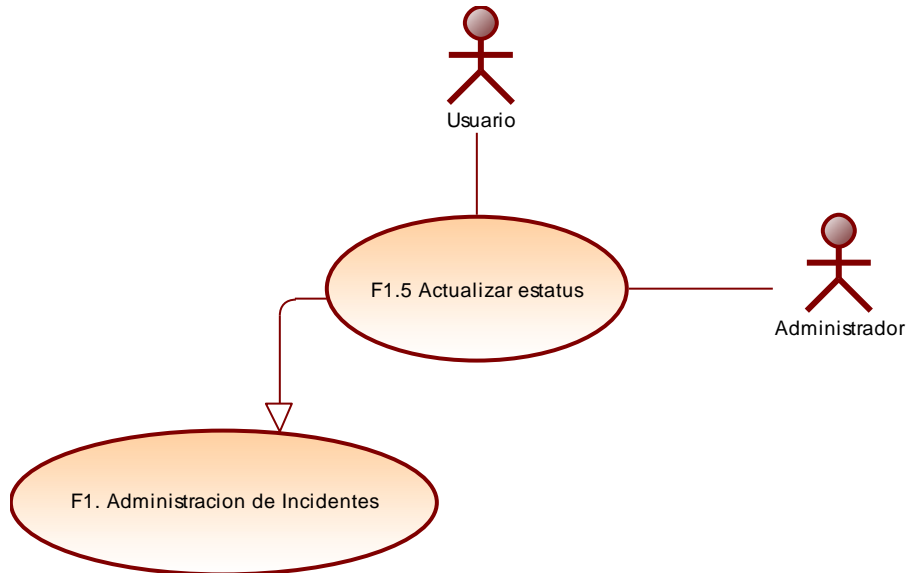


Figura 30 F1.5 Actualizar estatus
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso tanto el usuario como el administrador podrán modificar el estatus de la incidencia para completar o actualizar su información

Actores:

- Administrador
- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El actor pulsa la lista de valores de la columna Estatus	E1
3	El sistema presenta la lista de opciones para seleccionar un estatus	
4	El usuario selecciona un nuevo estatus	
5	El usuario selecciona el botón GUARDAR	E2

6	El sistema almacena los datos ingresados	
---	--	--

Excepciones

Código	Descripción	Alternativa
E1	Error en la lista de valores no devuelve resultados	Contactarse con el soporte técnico
E2	Error al grabar	Contactarse con el soporte técnico

F1.6 Asignar Consultor

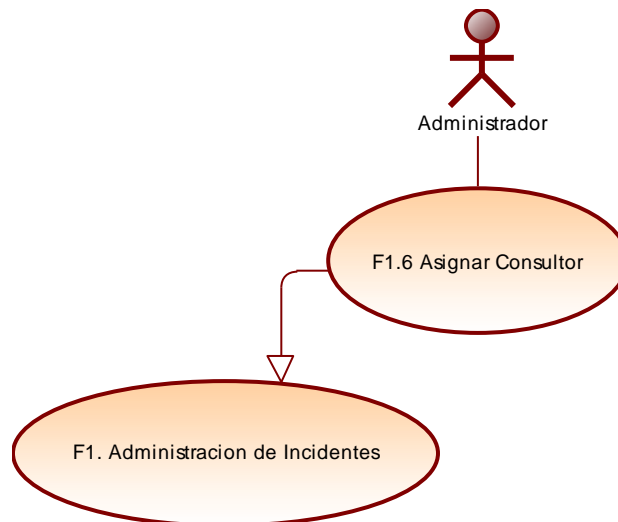


Figura 31 F1.6 Asignar Consultor
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el administrador podrá asignar a cada ticket de incidencia un consultor para su seguimiento y hasta su resolución

Actores:

- Administrador

Flujo Principal:

Paso	Actor	Excepción
------	-------	-----------

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

1	El sistema presenta la pantalla de los tickets de incidencias	
2	El administrador pulsa en la lista de valores de la columna consultor asignado	E1
3	El sistema presenta la lista de opciones para seleccionar un consultor	
4	El administrador selecciona un nuevo consultor	
5	El administrador selecciona el botón GUARDAR	E2
6	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Error en la lista de valores no devuelve resultados	Contactarse con el soporte técnico
E2	Error al grabar	Contactarse con el soporte técnico

F1.7 Ingresar Notas Adicionales

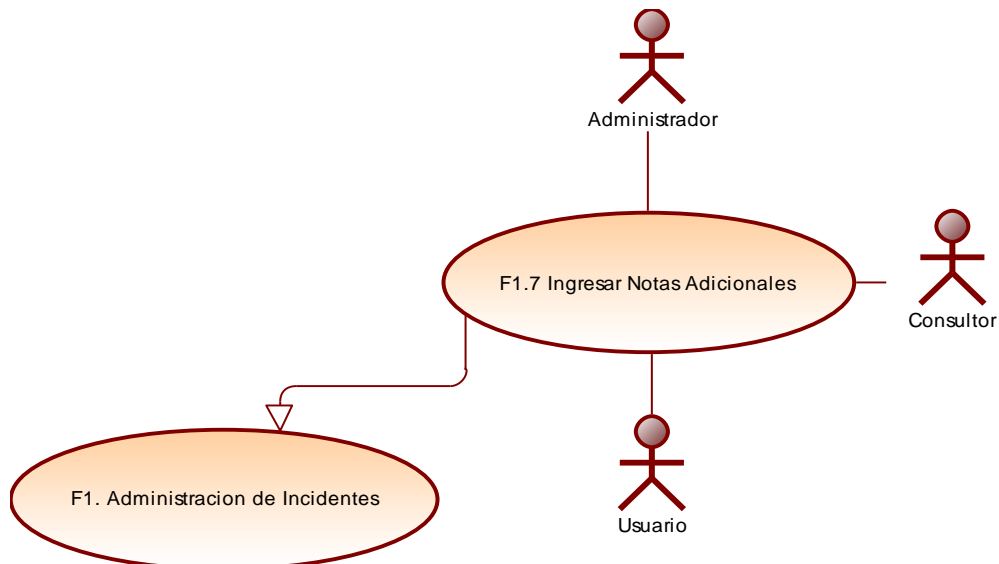


Figura 32 F1.7 Ingresar Notas Adicionales
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Descripción:

A través de este caso de uso el administrador, usuario y consultor podrán agregar notas adicionales para una incidencia

Actores:

- Administrador
- Consultor
- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El actor pulsa el botón (+) de la columna descripción	
3	El sistema presenta el formulario para agregar un nota adicional	
4	El actor ingresa los datos del formulario	E1
5	El actor presiona GUARDAR	E2
6	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Campos obligatorios vacíos	Llenar los campos obligatorios.
E2	Error de conexión	Verificar la conexión a internet

F2. Administración de Requerimientos

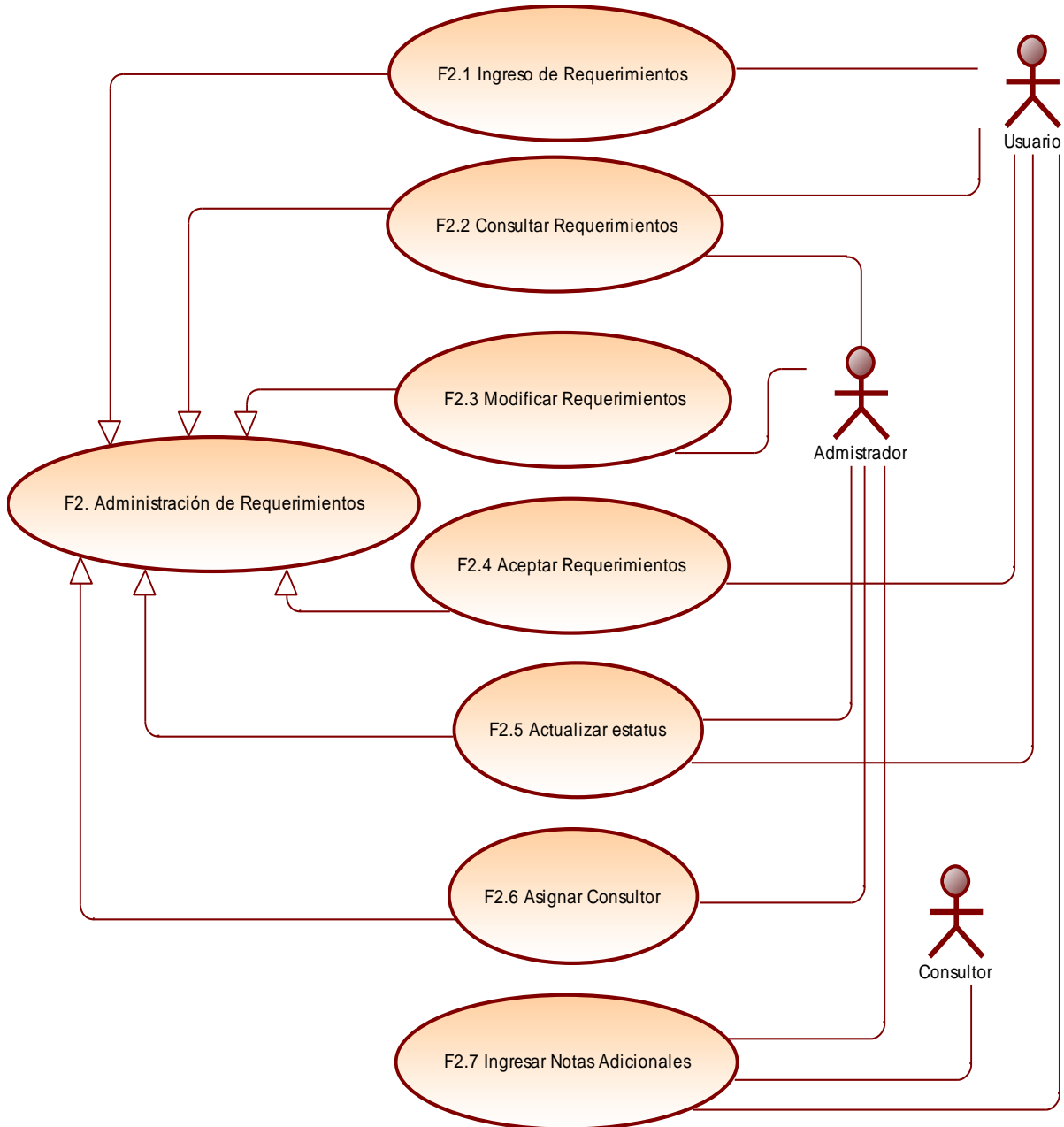


Figura 33 F2. Administración de Requerimientos
Autor: Santiago Gálvez

F2.1 Ingreso de Requerimientos

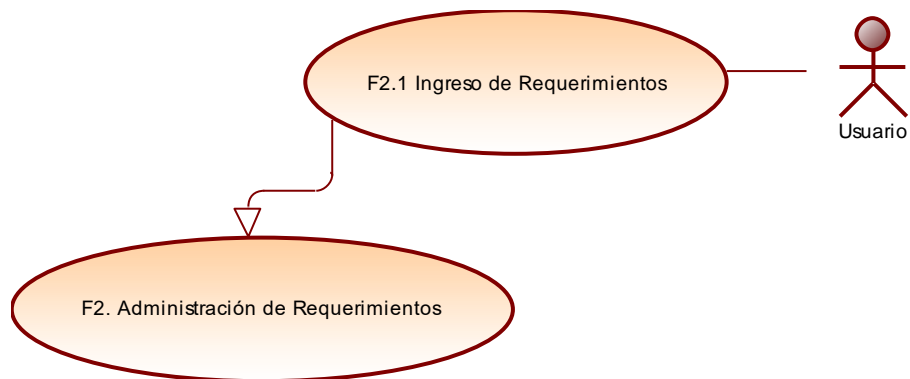


Figura 34 F2.1 Ingreso de Requerimiento
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso los usuarios podrán realizar el ingreso de nuevos tickets de requerimientos en el sistema

Actores:

- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El usuario selecciona el botón de NUEVO TICKET	
3	El sistema presenta el formulario para ingresar una incidencia	
4	El usuario ingresa los datos del formulario	E1
5	El usuario presiona GUARDAR	E2
6	El sistema almacena los datos	

Excepciones

Código	Descripción	Alternativa
E1	Campos obligatorios vacíos	Llenar los campos obligatorios.
E2	Error de conexión	Verificar la conexión a internet

F2.2 Consultar Requerimientos

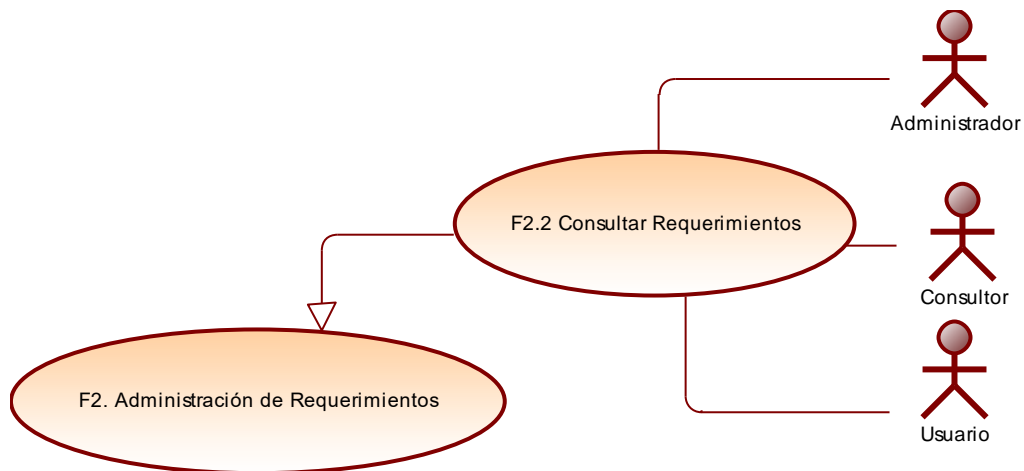


Figura 35 F2.2 Consultar Requerimientos
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso los administradores, consultores y los usuarios podrán realizar la consulta de las incidencias

Actores:

- Usuario
- Administrador
- Consultor

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El usuario selecciona los parámetros de búsqueda	
3	El sistema presenta una lista de requerimientos que cumple con los parámetros solicitados	E1

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet

F2.3 Modificar Requerimientos

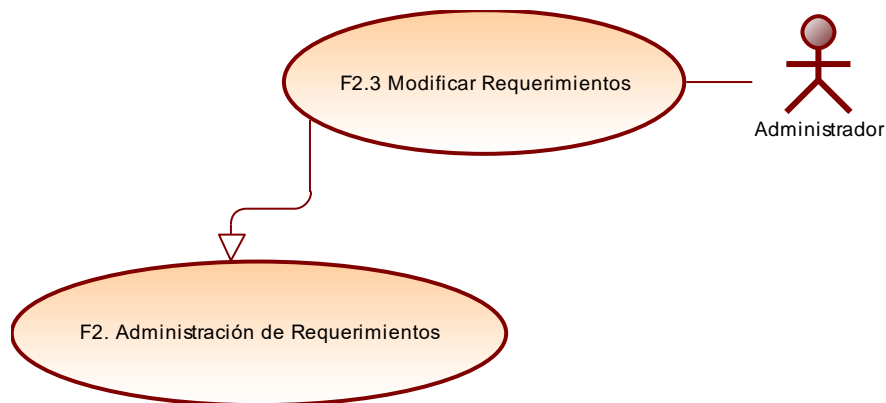


Figura 36 F2.3 Modificar Requerimientos
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el administrador cuando sea necesario podrá cambiar o aumentar los datos que desea del ticket de requerimientos para completar o actualizar su información.

Actores:

- Administrador

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El administrador modifica los datos necesarios	
3	El administrador pulsa el botón GUARDAR	E1
4	El sistema almacena los datos ingresados.	E2

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet
E2	Error al grabar	Contactarse con el soporte técnico

F2.4 Aceptar Requerimientos

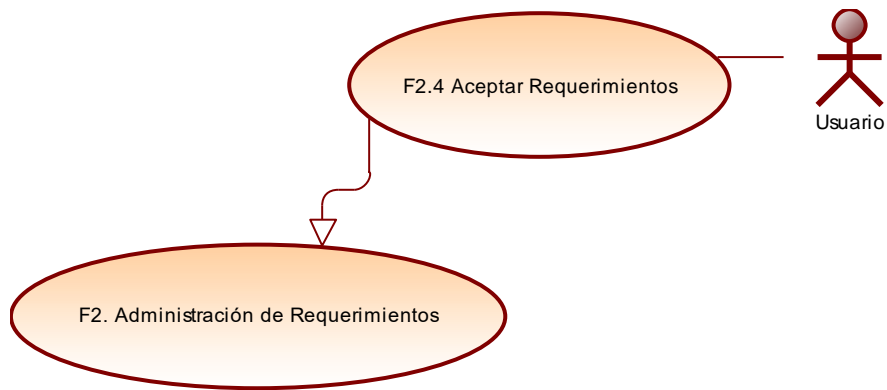


Figura 37 F2.4 Aceptar Requerimientos
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el usuario podrá aceptar los cambios o soluciones aplicadas para solventar el requerimiento dando por finalizado el seguimiento del mismo en el proceso

Actores:

- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El usuario pulsa el botón Aceptado	E1
3	El sistema presenta la pantalla encuesta de satisfacción	
4	El usuario selecciona un nivel de satisfacción	
5	El sistema habilita el botón enviar	
6	El usuario pulsa el botón enviar	E2
7	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet
E2	Error al grabar	Contactarse con el soporte técnico

F2.5 Actualizar estatus

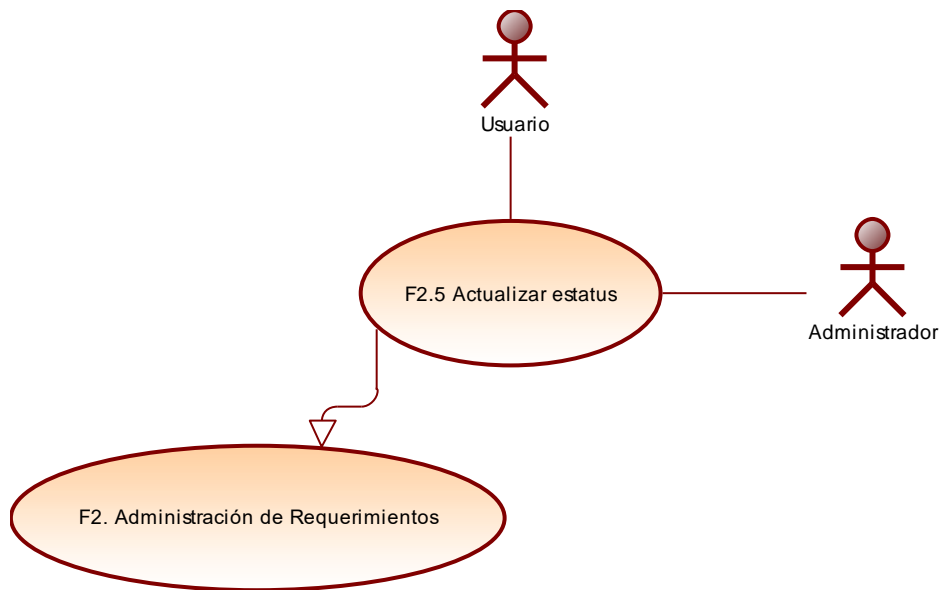


Figura 38 F2.5 Actualizar estatus
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso tanto el usuario como el administrador podrán modificar el estatus de la incidencia para completar o actualizar su información

Actores:

- Administrador
- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El actor pulsa la lista de valores de la columna Estatus	E1
3	El sistema presenta la lista de opciones para seleccionar un estatus	
4	El usuario selecciona un nuevo estatus	
5	El usuario selecciona el botón GUARDAR	E2

6	El sistema almacena los datos ingresados	
----------	--	--

Excepciones

Código	Descripción	Alternativa
E1	Error en la lista de valores no devuelve resultados	Contactarse con el soporte técnico
E2	Error al grabar	Contactarse con el soporte técnico

F2.6 Asignar Consultor

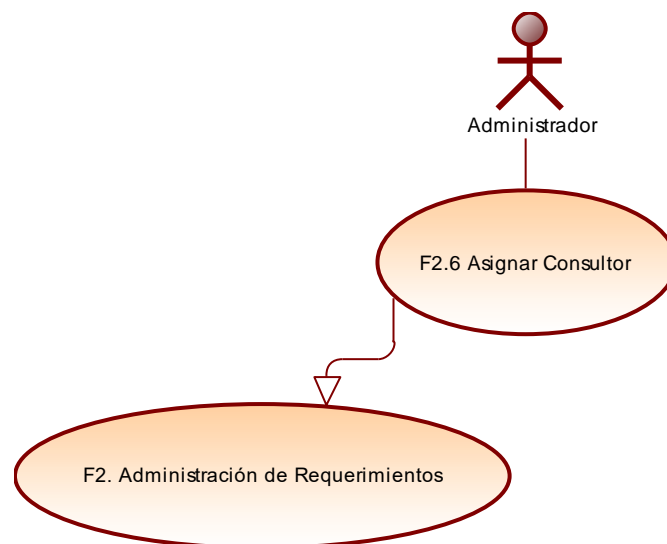


Figura 39 F2.6 Asignar Consultor
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el administrador podrá asignar a cada ticket de requerimientos un consultor para su seguimiento y hasta su resolución

Actores:

- Administrador

Flujo Principal:

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El administrador pulsa en la lista de valores de la columna consultor asignado	E1
3	El sistema presenta la lista de opciones para seleccionar un consultor	
4	El administrador selecciona un nuevo consultor	
5	El administrador selecciona el botón GUARDAR	E2
6	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Error en la lista de valores no devuelve resultados	Contactarse con el soporte técnico
E2	Error al grabar	Contactarse con el soporte técnico

F2.7 Ingresar Notas Adicionales

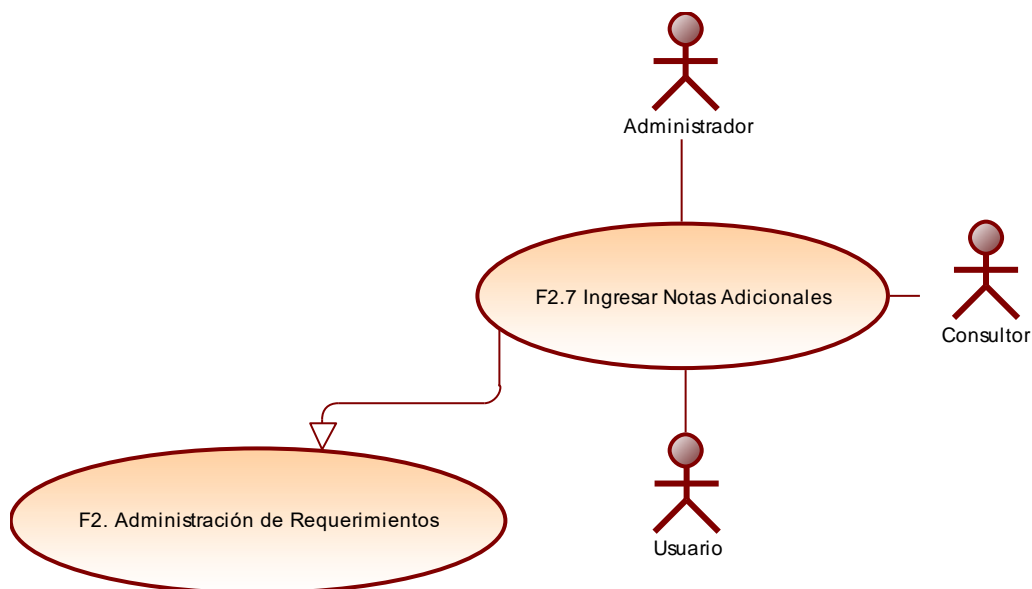


Figura 40 F2.7 Ingresar Notas Adicionales
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Descripción:

A través de este caso de uso el administrador, usuario y consultor podrán agregar notas adicionales para una incidencia

Actores:

- Administrador
- Consultor
- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de requerimientos	
2	El actor pulsa el botón (+) de la columna descripción	
3	El sistema presenta el formulario para agregar un nota adicional	
4	El actor ingresa los datos del formulario	E1
5	El actor presiona GUARDAR	E2
6	El sistema almacena los datos ingresados	

Excepciones

Código	Descripción	Alternativa
E1	Campos obligatorios vacíos	Llenar los campos obligatorios.
E2	Error de conexión	Verificar la conexión a internet

F3. Administración de Tickets Resueltos

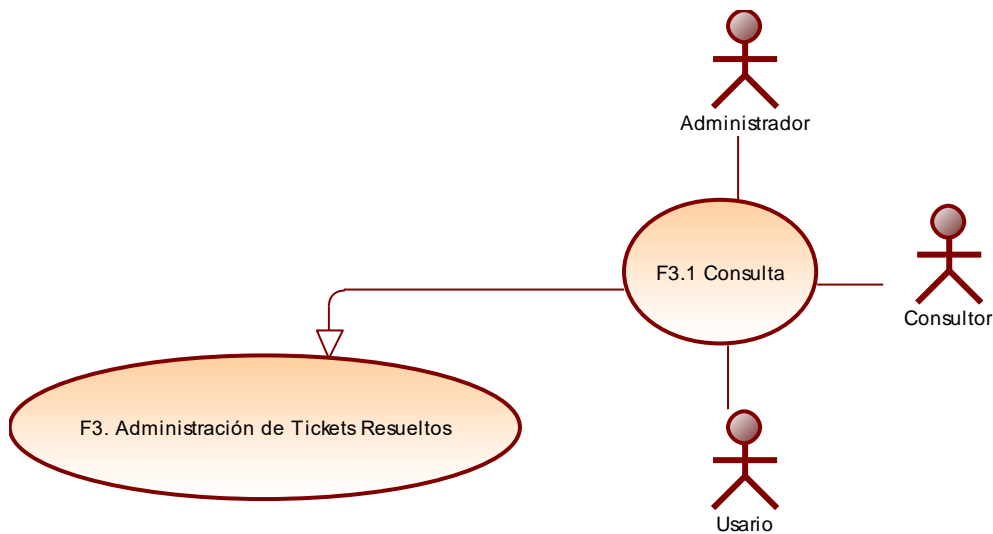


Figura 41 F3. Administración de Tickets Resueltos
Autor: Santiago Gálvez

F3.1 Consulta Tickets

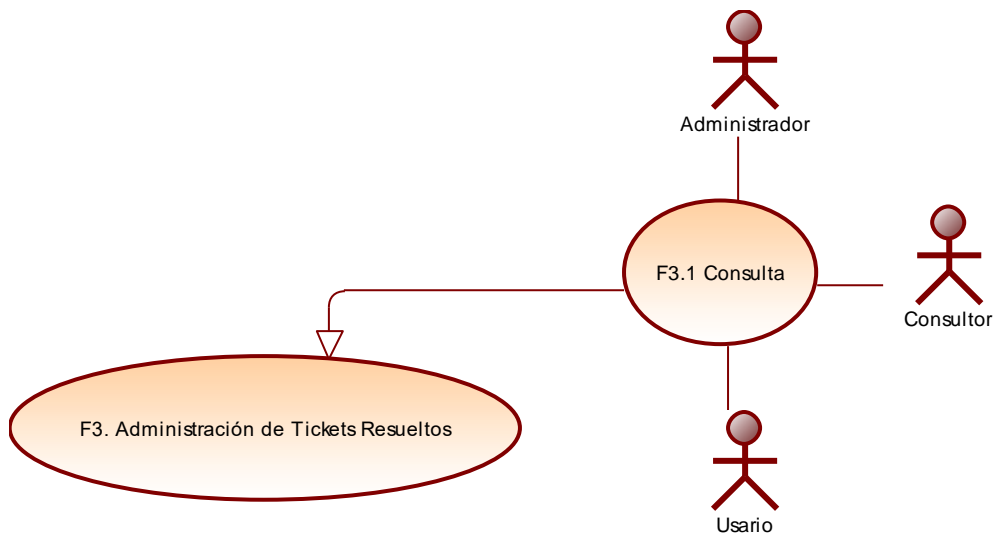


Figura 42 F3.1 Consulta
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el administrador, consultor y usuario podrán realizar la consulta de los tickets en general que se han resultado hasta el momento

Actores:

- Administrador
- Consultor
- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El usuario selecciona la pestaña de Tickets Aprobados	
3	El sistema presenta la pantalla de tickets aprobados	
4	El usuario ingresa parámetros de búsqueda	
5	El sistema presenta una lista de requerimientos que cumple con los parámetros solicitados	E1

Excepciones

Código	Descripción	Alternativa
E1	Error de conexión	Verificar la conexión a internet

F4. Administración de Reportes

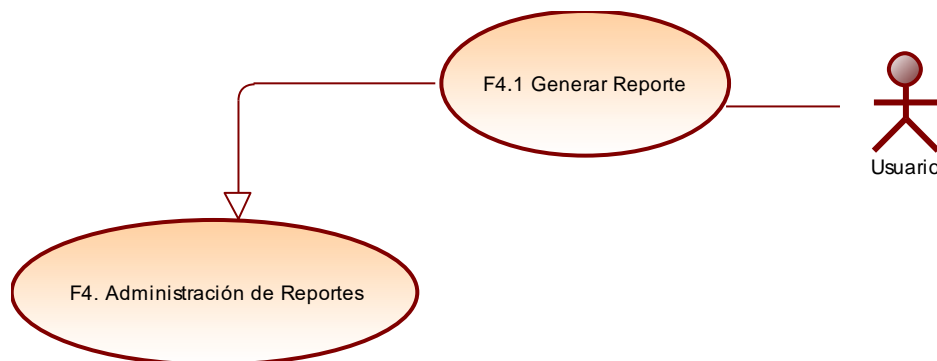


Figura 43 F4. Administración de Reportes
Autor: Santiago Gálvez

F4.1 Generar Reporte

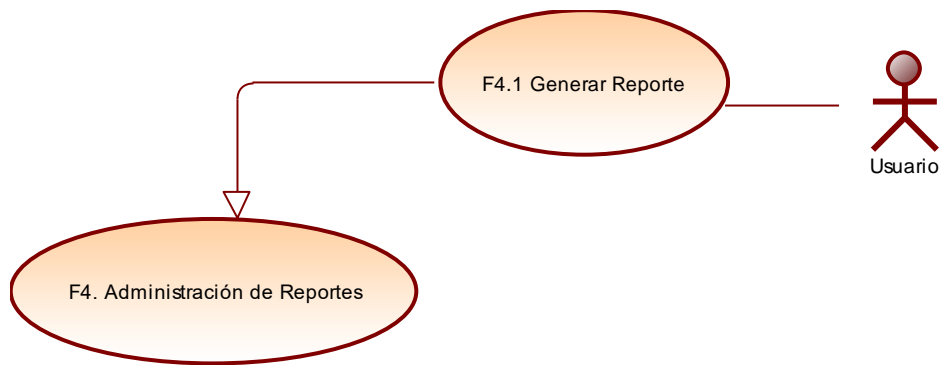


Figura 44 F4.1 Generar Reporte
Autor: Santiago Gálvez

Descripción:

A través de este caso de uso el usuario podrá generar reportes en un rango de fechas

Actores:

- Usuario

Flujo Principal:

Paso	Actor	Excepción
1	El sistema presenta la pantalla de los tickets de incidencias	
2	El usuario selecciona la pestaña de Reportes	
3	El sistema presenta la pantalla de reportes	
4	El usuario llena el formulario para generar el reporte	E1
5	El usuario presiona el botón (PDF) o (EXCEL)	E2
6	El sistema genera el reporte	

Excepciones

Código	Descripción	Alternativa
E1	Llenar los campos obligatorios.	Campos obligatorios vacíos
E2	Error de conexión	Verificar la conexión a internet

Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.

De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Muchos requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo. Esto significa que a menudo son más críticos que los requerimientos funcionales particulares. Mientras que el incumplimiento de este último degradará el sistema, una falla en un requerimiento no funcional del sistema lo inutiliza.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, etc.

El sistema dependerá de los siguientes requerimientos no funcionales:

- N1.** El sistema debe ser elaborado en herramientas libres
- N2.** El sistema debe acoplarse a los requerimientos mínimos de hardware de los clientes
- N3.** El sistema debe cumplir normas de desarrollo de la empresa Soft Warehouse
- N4.** El sistema debe manejar la base de datos clientes, sistemas, módulos, transacciones de la empresa.

5.1.3. Diagrama de Secuencia

A continuación se presentarán los diagramas de secuencia del sistema, estos muestran las interacciones que indican la secuencia de comunicaciones entre instancias de clases, componentes, subsistemas o actores.

F0. Ingreso al Sistema

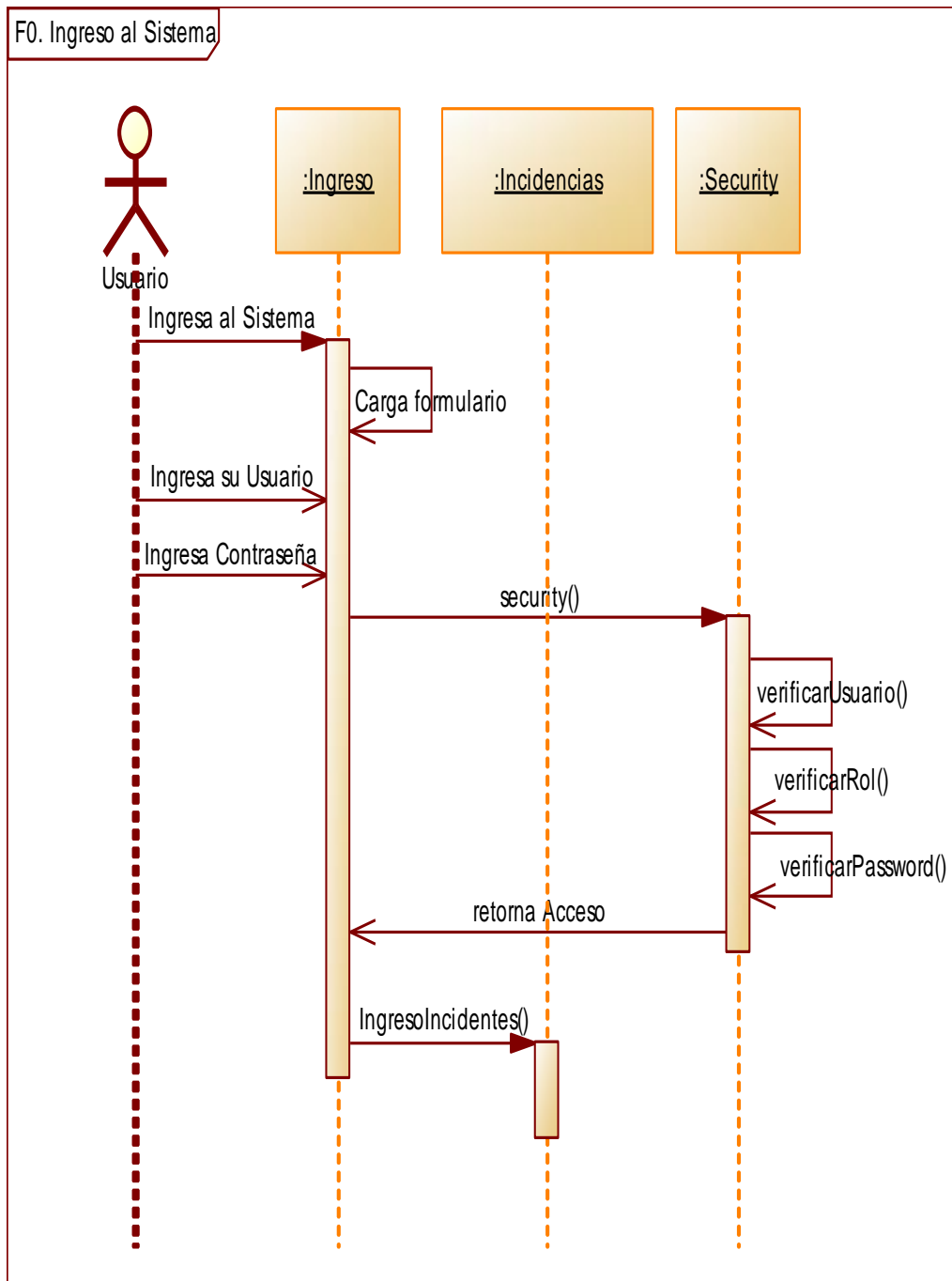


Figura 45 Diagrama de Secuencia F0. Ingreso al Sistema
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F1.1 Ingreso de incidencia

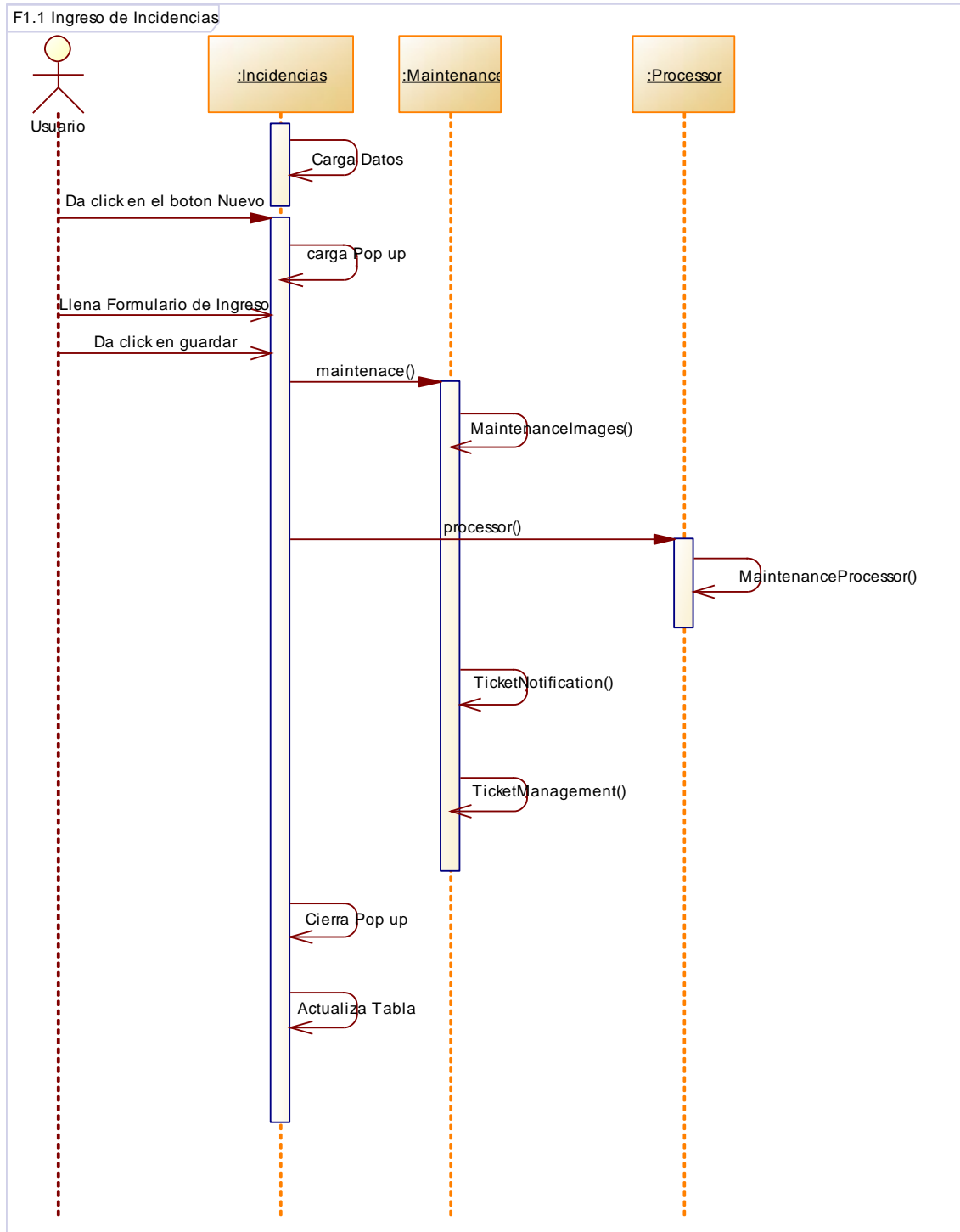


Figura 46 Diagrama de Secuencia F1.1 Ingreso de incidencia
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F1.2 Consultar Incidencia

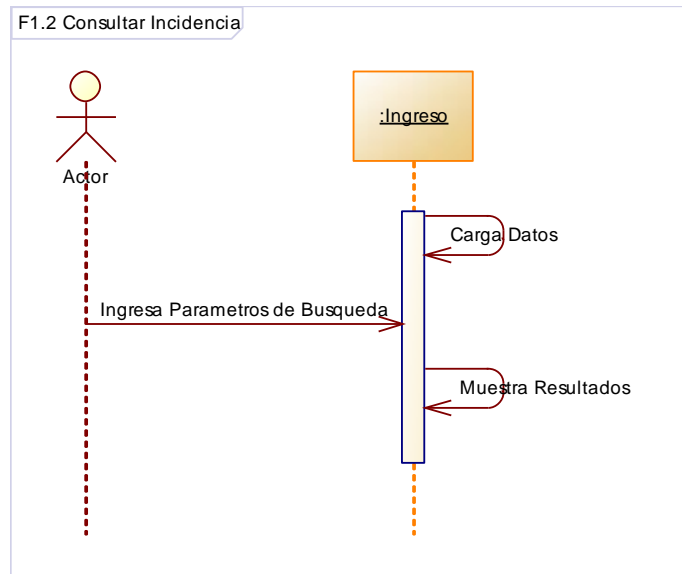


Figura 47 Diagrama de Secuencia F1.2 Consultar Incidencia
Autor: Santiago Gálvez

F1.3 Modificar Incidencia

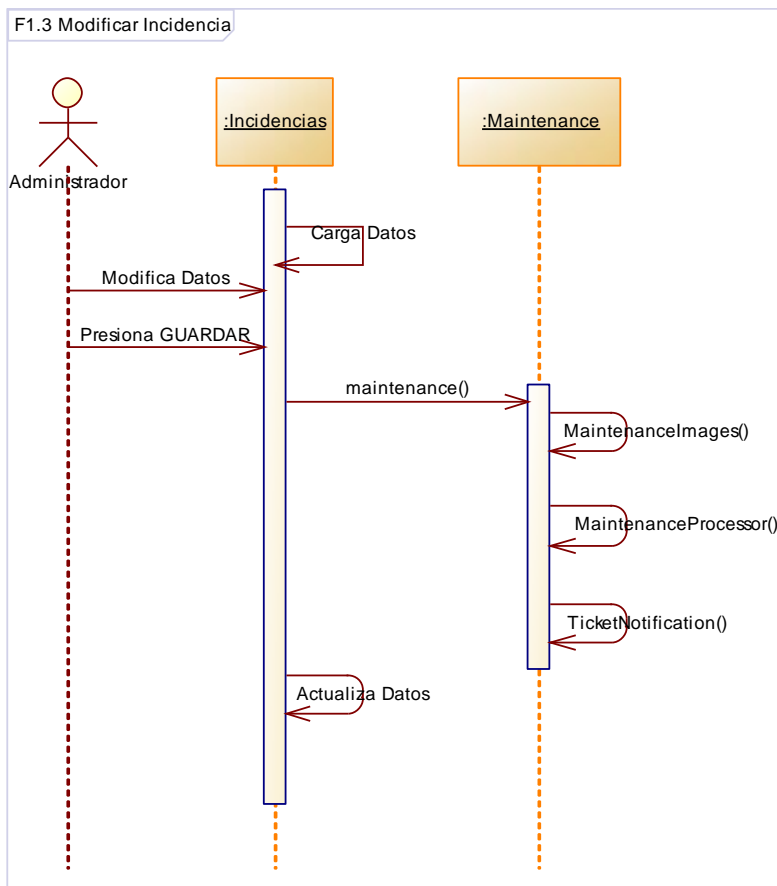


Figura 48 F1.3 Diagrama de Secuencia Modificar Incidencia
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F1.4 Aceptar Incidencia

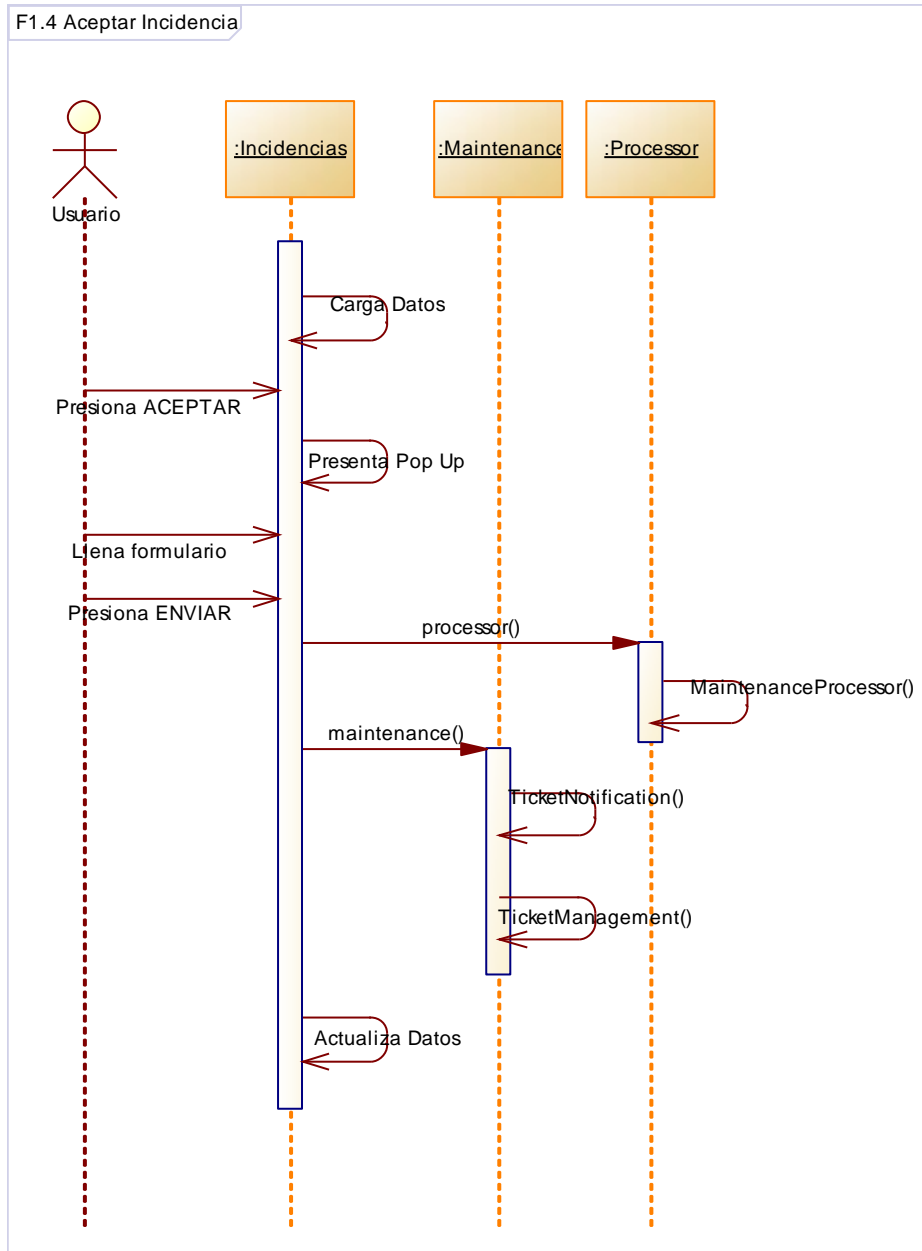


Figura 49 Diagrama de Secuencia F1.4 Aceptar Incidencia
Autor: Santiago Gálvez

F1.5 Actualizar estatus

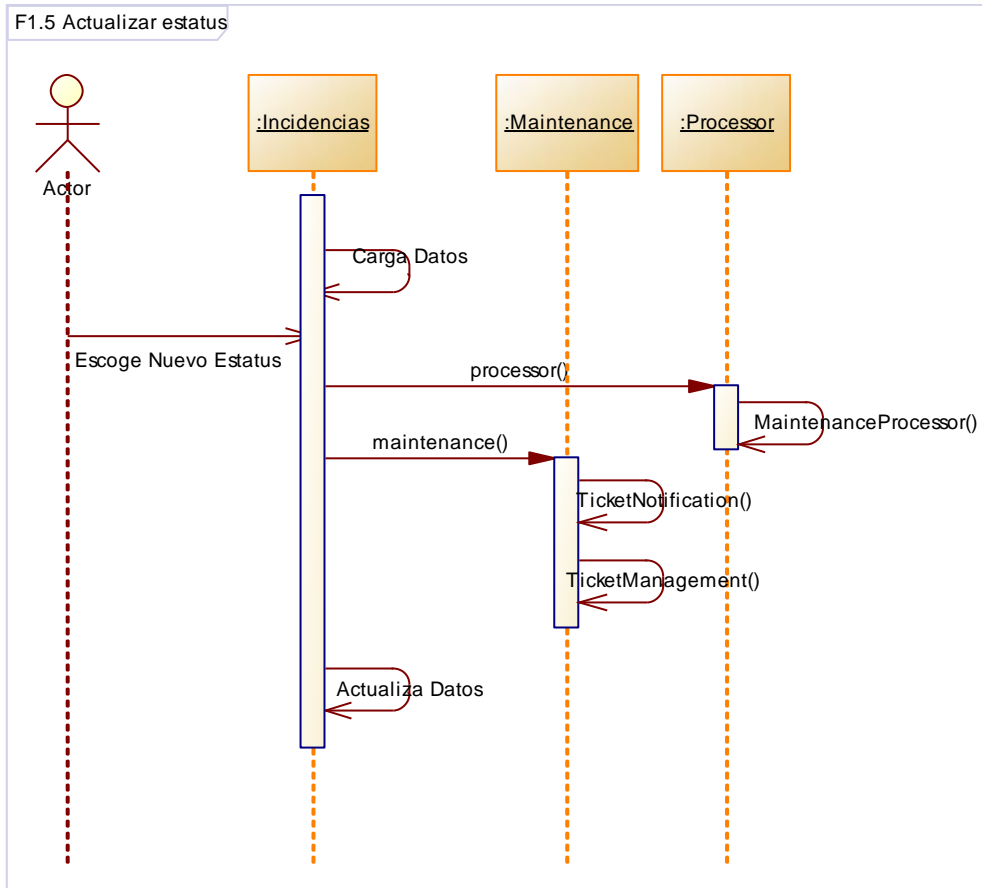


Figura 50 Diagrama de Secuencia F1.5 Actualizar estatus
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F1.6 Asignar Consultor

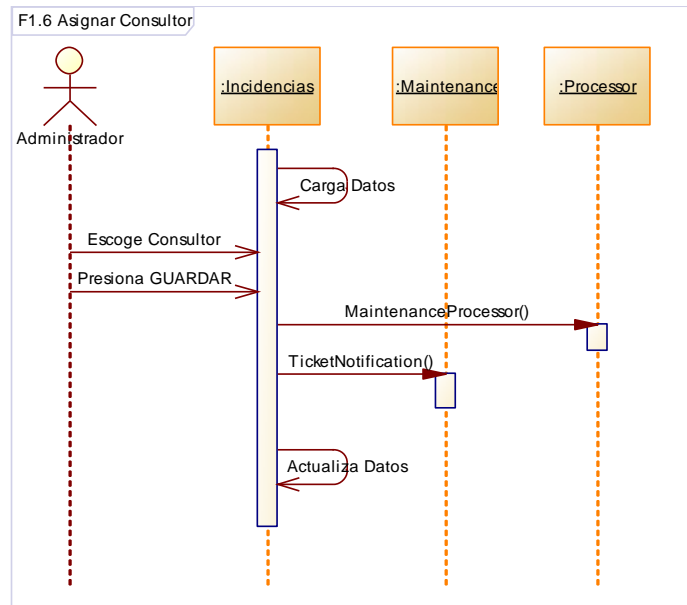


Figura 51 Diagrama de Secuencia F1.6 Asignar Consultor
Autor: Santiago Gálvez

F1.7 Ingresar Notas Adicionales

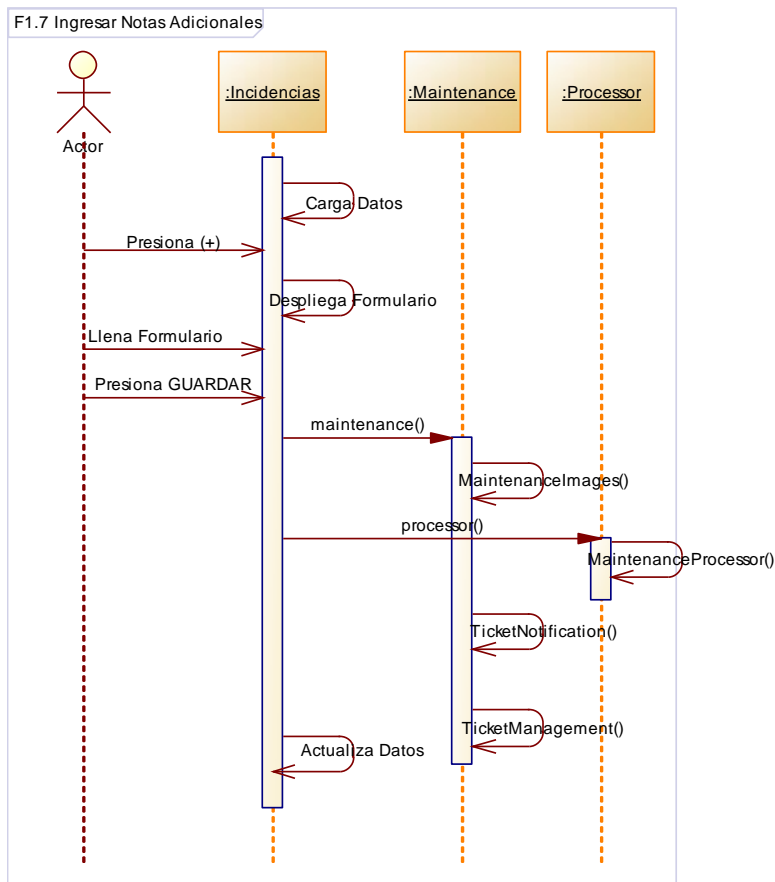


Figura 52 Diagrama de Secuencia F1.7 Ingresar Notas Adicionales
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.1 Ingreso de requerimiento

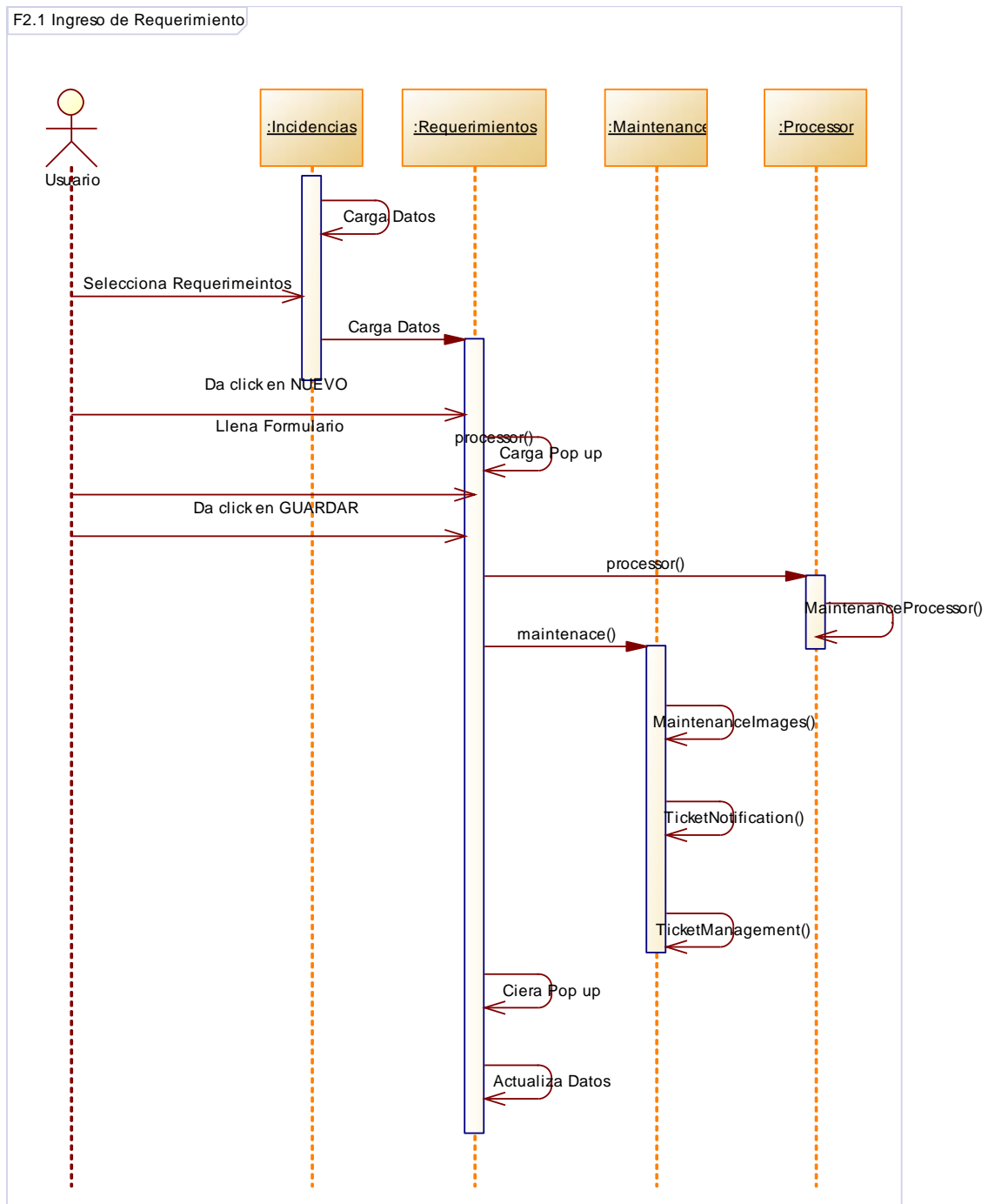


Figura 53 Diagrama de Secuencia F2.1 Ingreso de requerimiento
Autor: Santiago Gálvez

F2.2 Consultar requerimiento

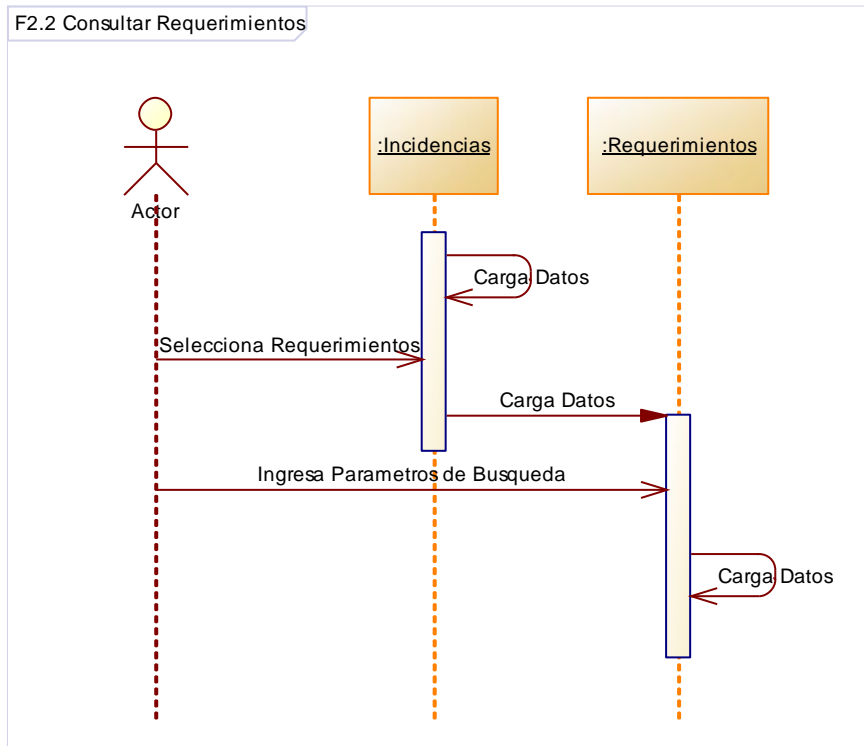


Figura 54 Diagrama de Secuencia F2.2 Consultar requerimiento
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.3 Modificar requerimiento

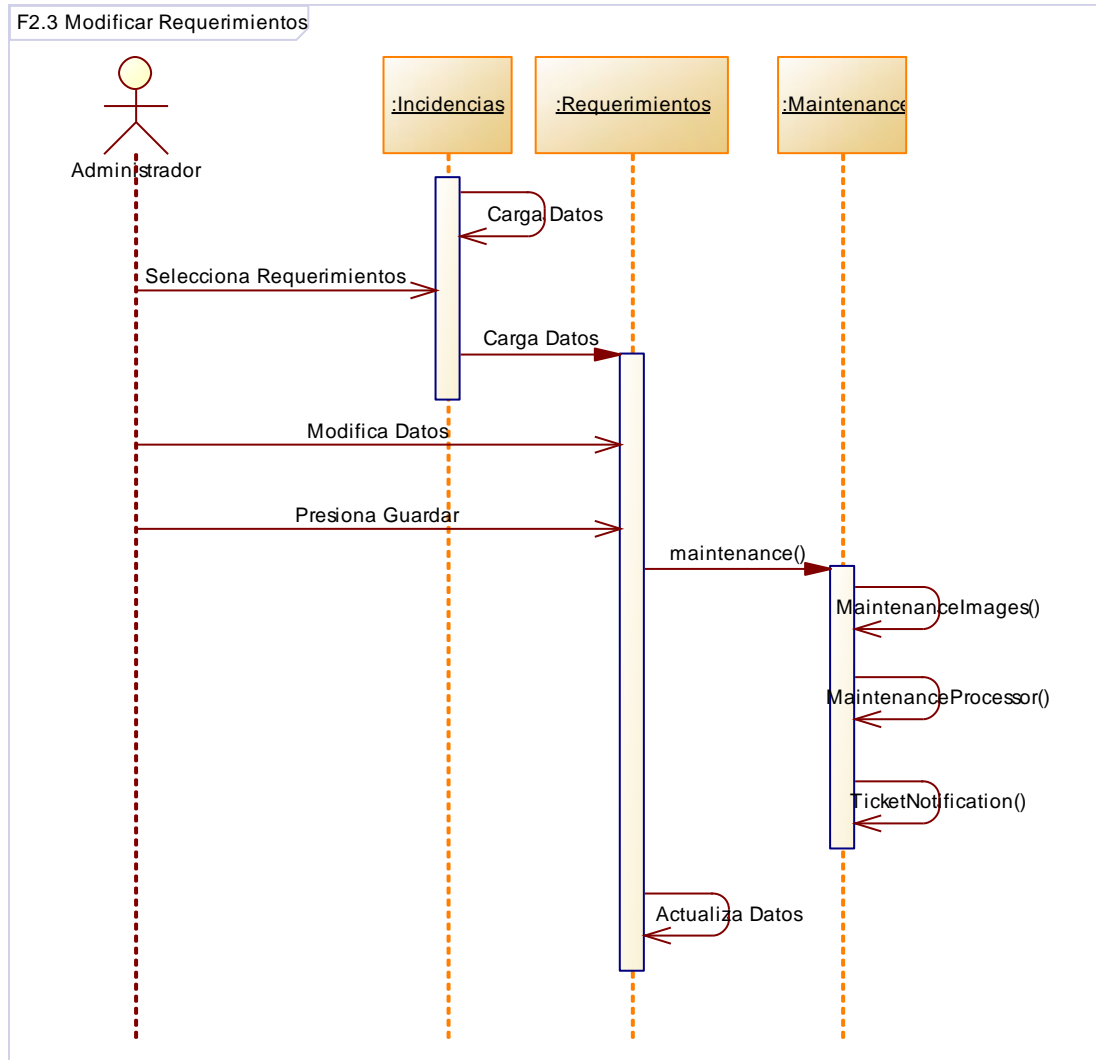


Figura 55 Diagrama de Secuencia F2.3 Modificar requerimiento
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.4 Aceptar requerimiento

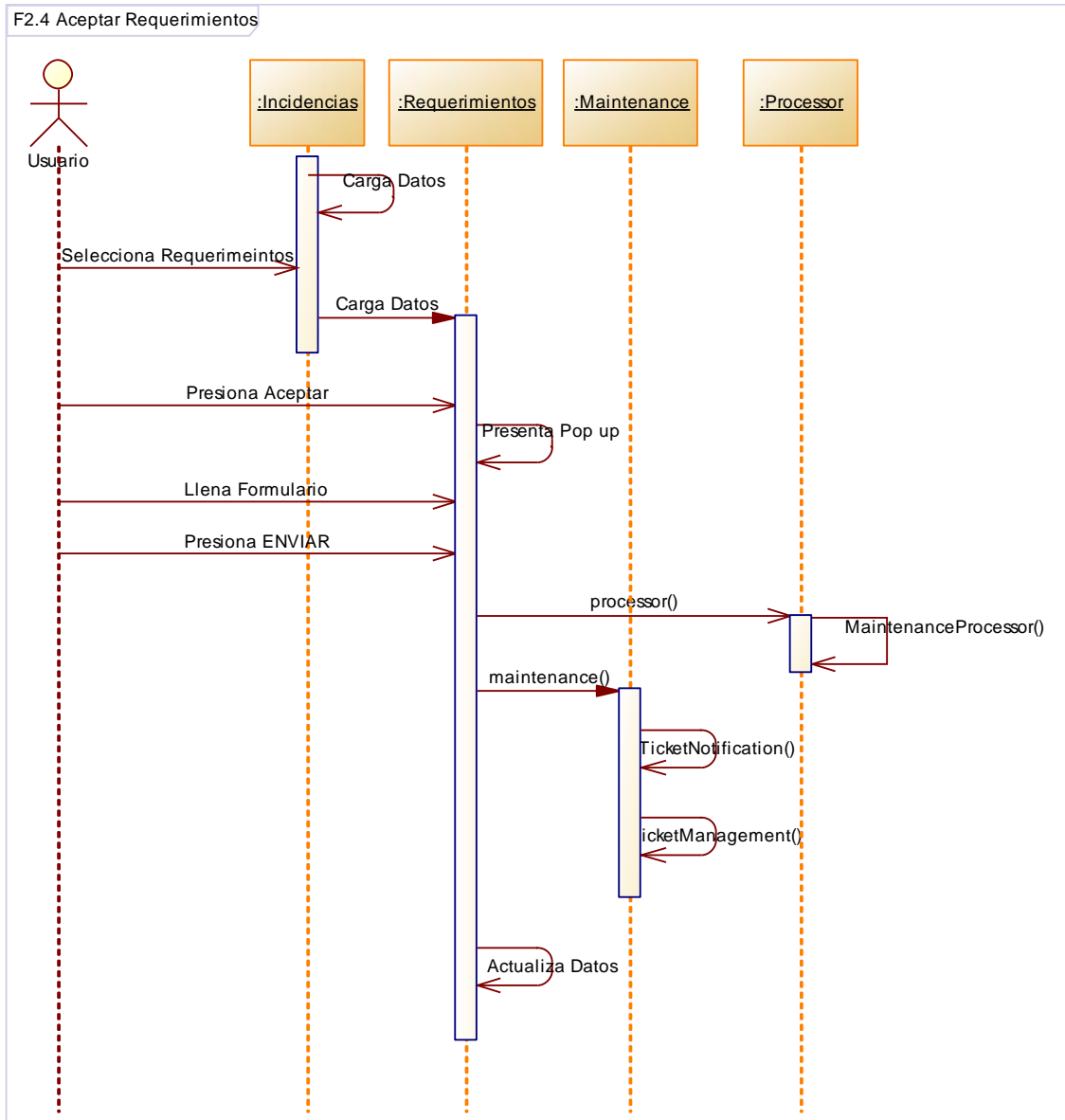


Figura 56 Diagrama de Secuencia F2.4 Aceptar requerimiento
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.5 Actualizar estatus

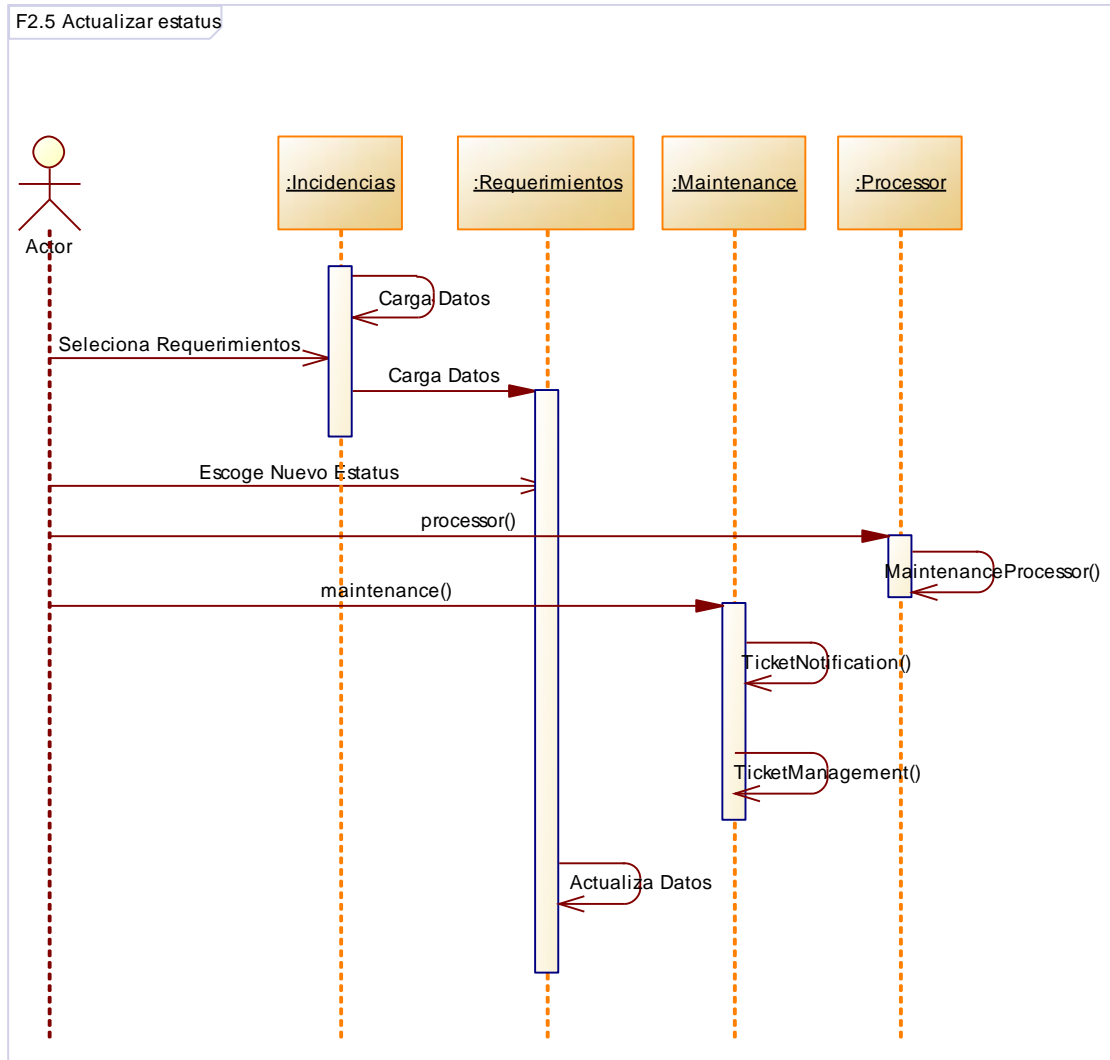


Figura 57 Diagrama de Secuencia F2.5 Actualizar estatus
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.6 Asignar Consultor

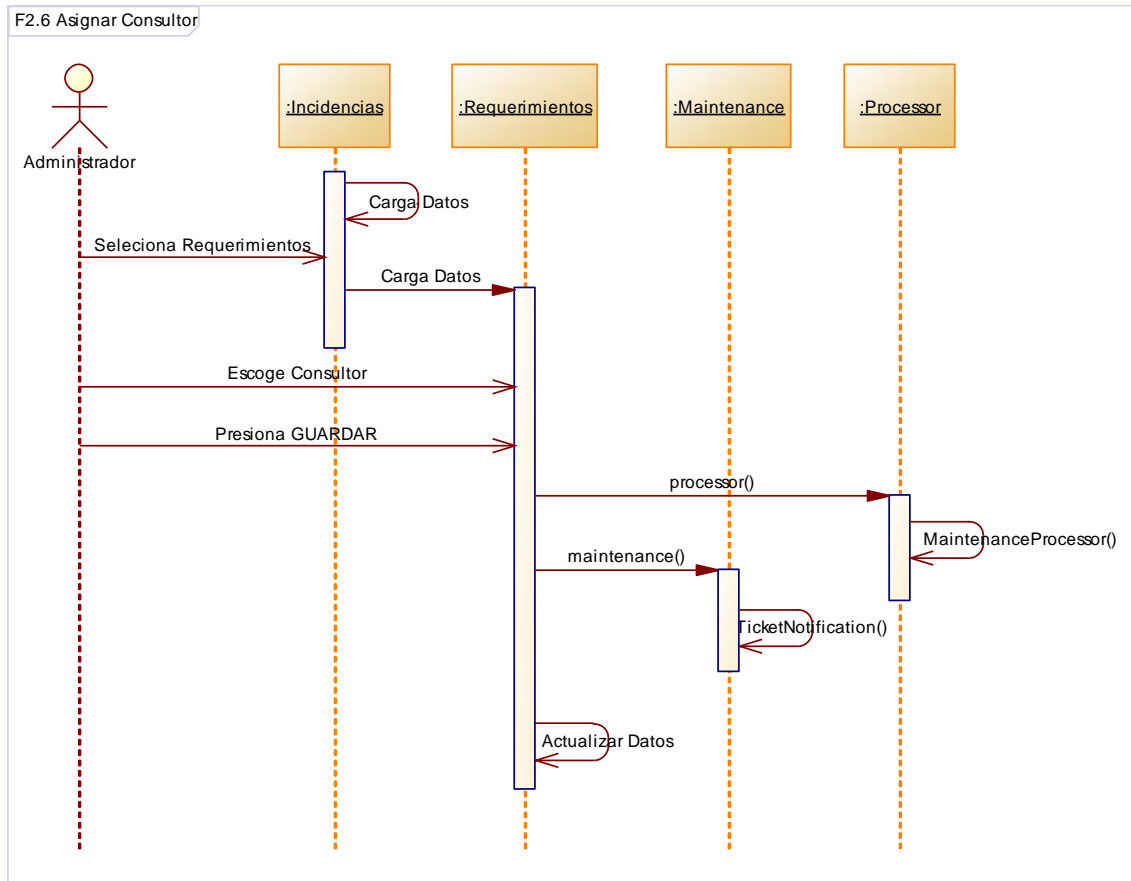


Figura 58 Diagrama de Secuencia F2.6 Asignar Consultor
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F2.7 Ingresar Notas Adicionales

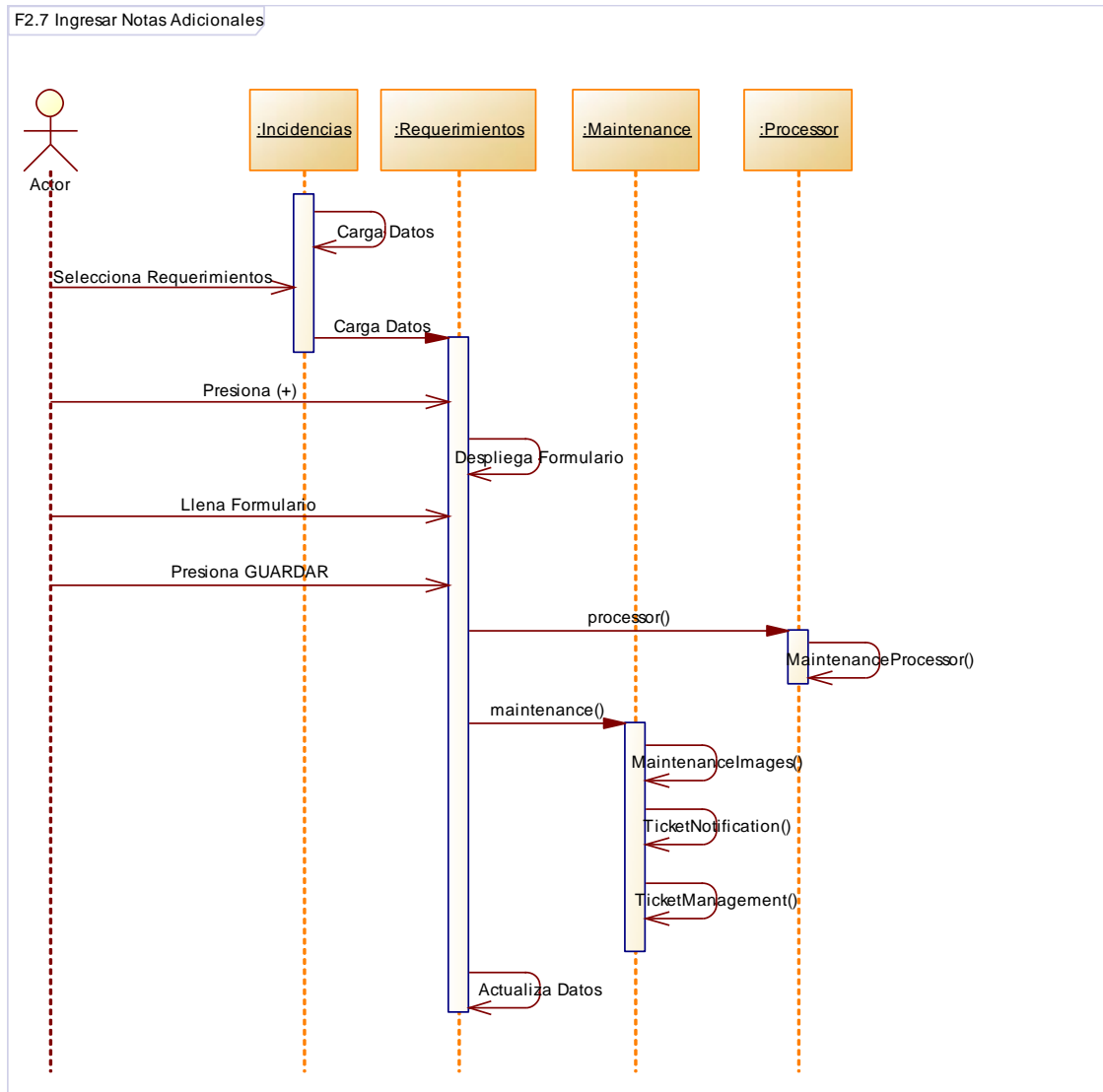


Figura 59 Diagrama de Secuencia F2.7 Ingresar Notas Adicionales
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

F3.1 Consulta Tickets

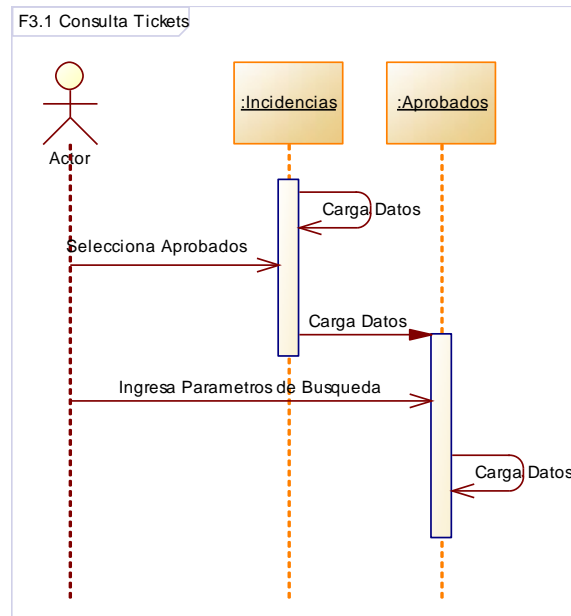


Figura 60 Diagrama de Secuencia F3.1 Consulta Tickets
Autor: Santiago Gálvez

F4.1 Generar Reporte

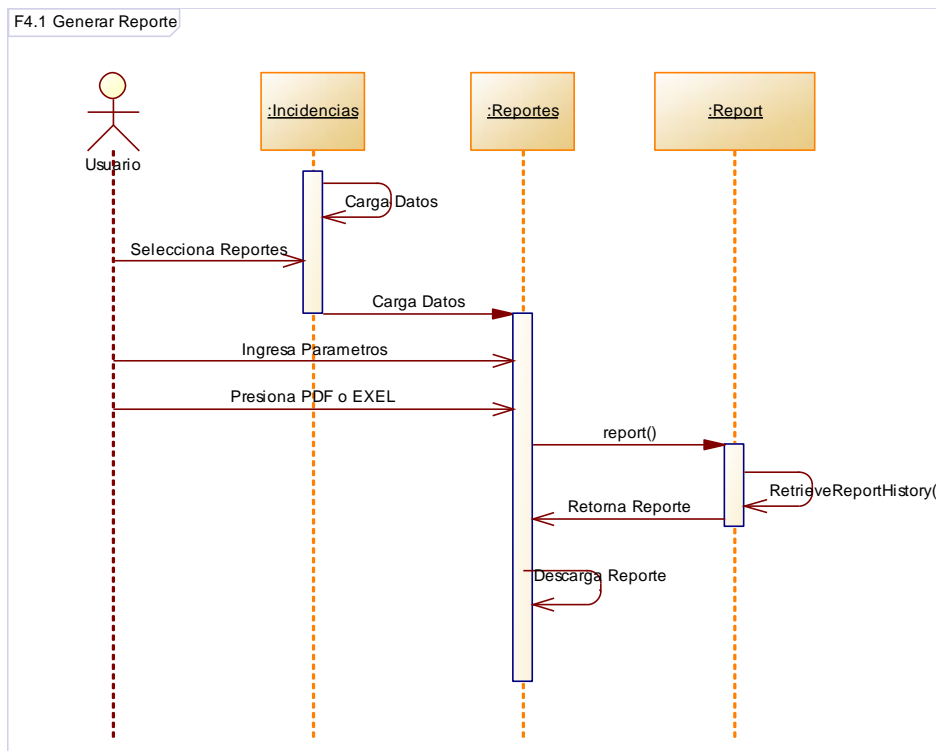


Figura 61 Diagrama de Secuencia F4.1 Generar Reporte
Autor: Santiago Gálvez

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Diagrama de Despliegue

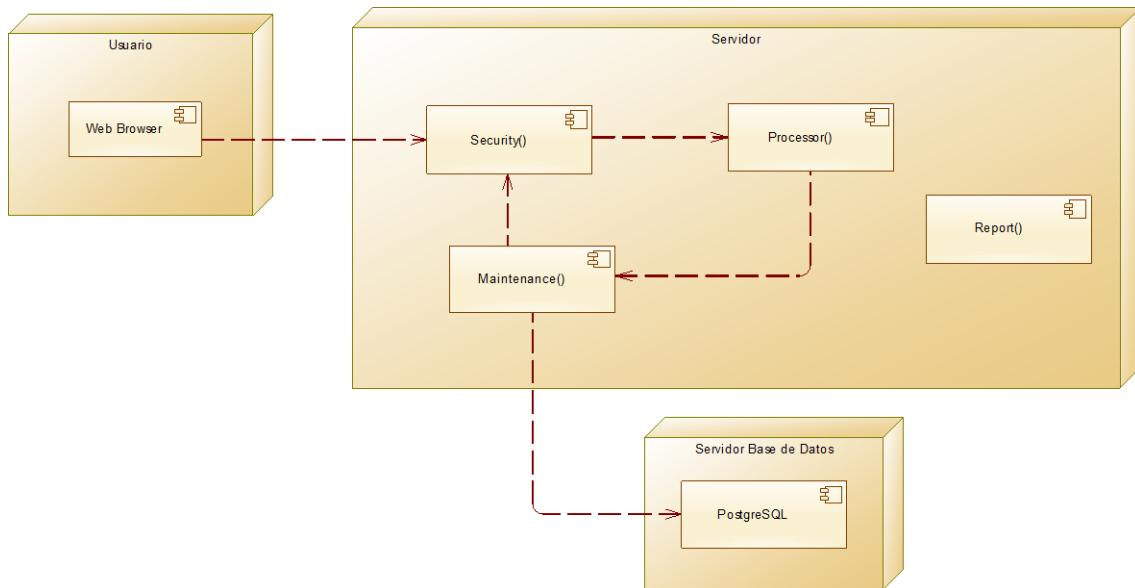


Figura 63 Diagrama de Despliegue
Autor: Santiago Gálvez

5.2. Elaboración

En este capítulo vamos a poder evidenciar mediante capturas de pantalla el sistema ya en funcionamiento, en este prototipo ejecutable vamos a seguir el proceso que normalmente se seguiría si este fuera un caso en la vida real.

Se comenzará por la ventana de ingreso, que es la misma para clientes como para usuarios de la empresa y continuaremos con las ventanas y elementos que tendría a disposición el cliente tanto para incidencias como requerimientos.

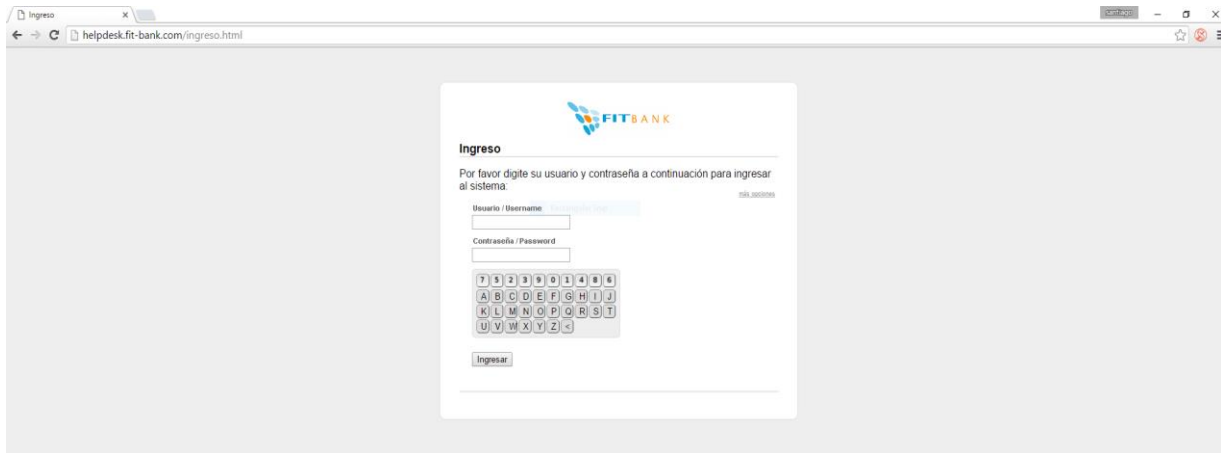
Esta sección concluirá con las ventanas de consultas y reportes, como las ventanas de consultas de incidentes y requerimientos ya aprobados y la ventana de generación de reportes exhibidos a continuación.

5.2.1. Sistema ejecutable

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

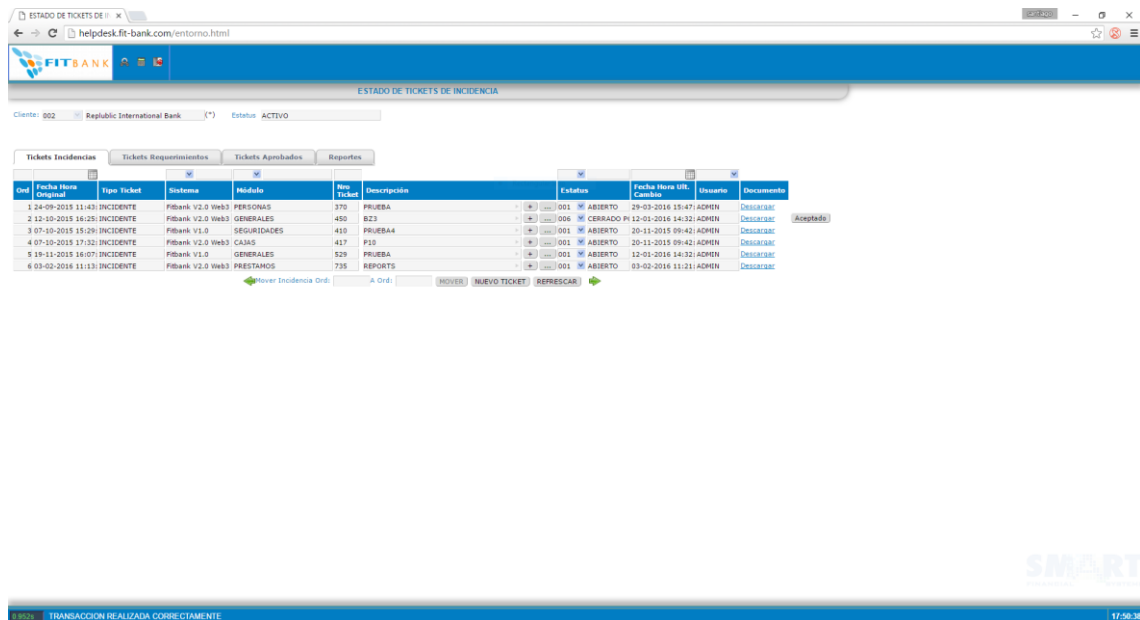
Ventana de Ingreso

La Venta de Ingreso sirve para permitir las funcionalidades de F0. Ingreso Sistema, algunas de sus funcionalidades que caracteriza esta ventana es que siempre pedirá una nueva clave la primera vez que se ingresa al sistema y tiene un número limitado de intentos antes de bloquear al usuario.



Ventana de Incidentes

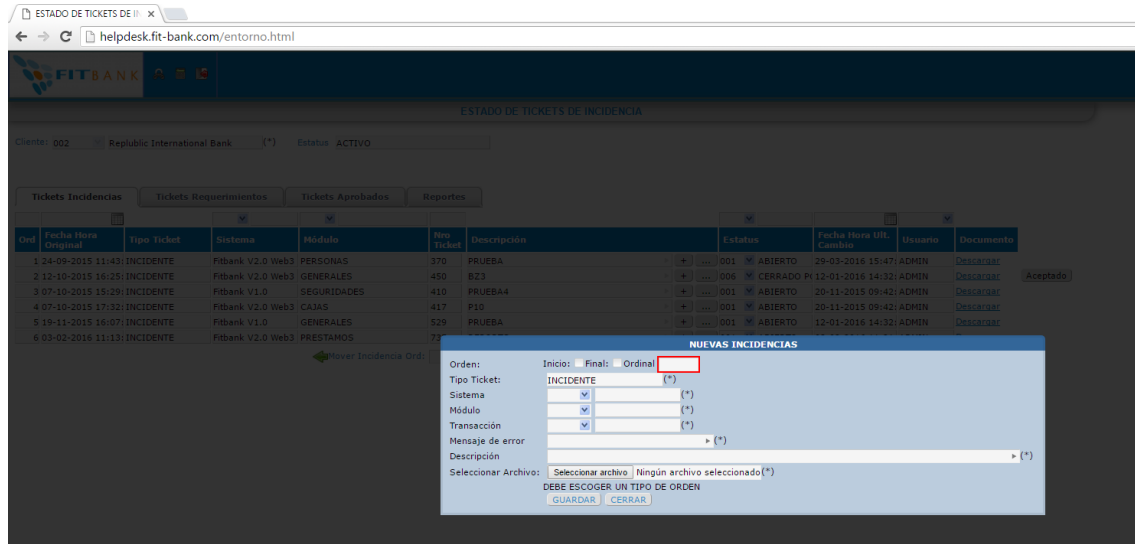
La Venta de Incidentes sirve para permitir las funcionalidades de F1. Administración de Incidentes, esta es la ventana principal de los incidentes, aquí se ejecuta la función F1.2 Consultar Incidencia y F1.5 Actualizar Estatus



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

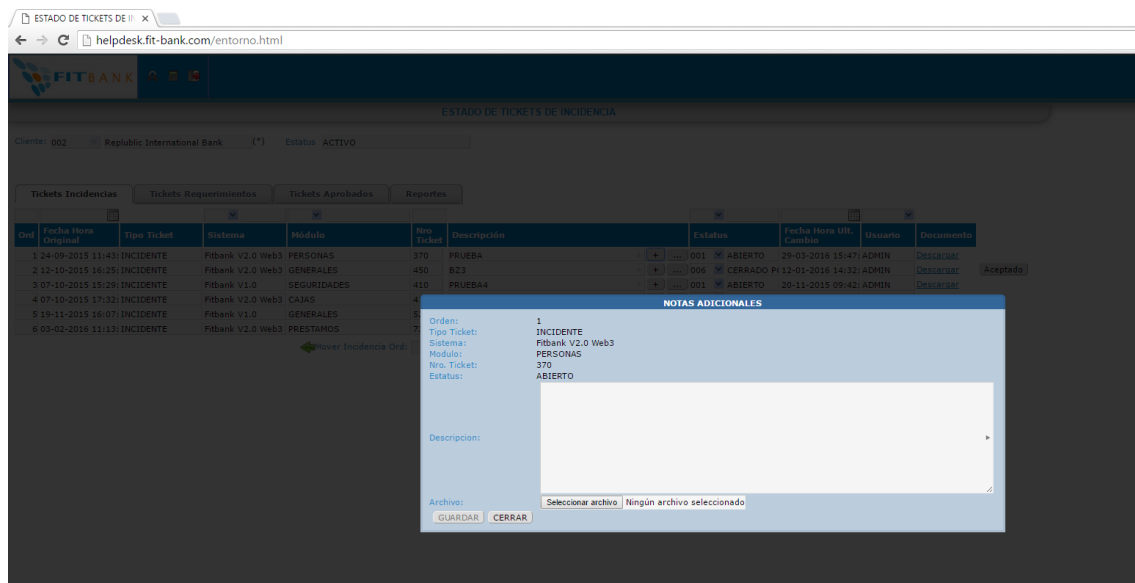
Nuevo Ticket

La Ventana de Nuevo Ticket sirve para permitir las funcionalidades de F1.1 Ingreso de Incidencia, esta ventana es donde se asigna la prioridad, modulo y transacción de las incidencias además de toda la información que describe el incidente.



Notas Adicionales

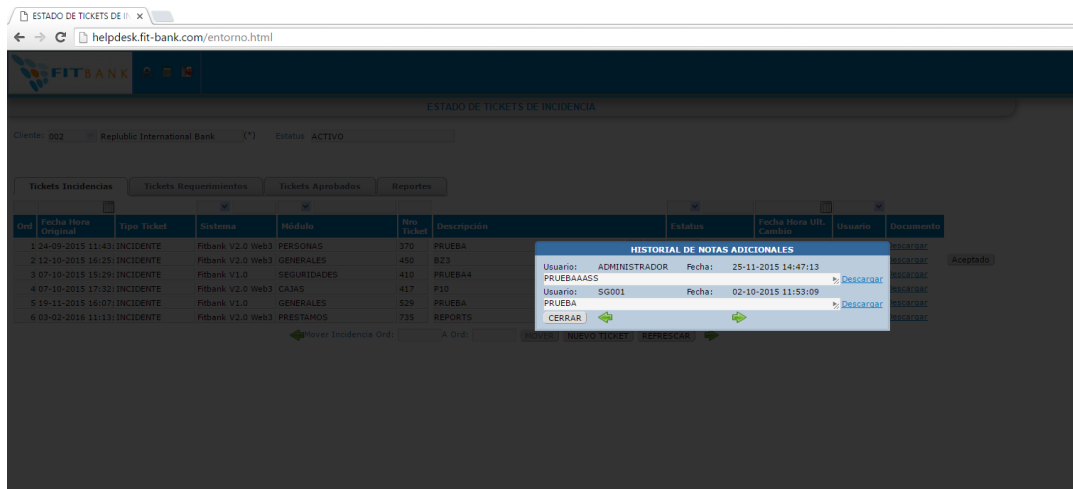
La Ventana de Notas Adicionales sirve para permitir las funcionalidades de F1.7 Ingresar Notas Adicionales, donde se podrá establecer una comunicación directa entre el consultor y el cliente.



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

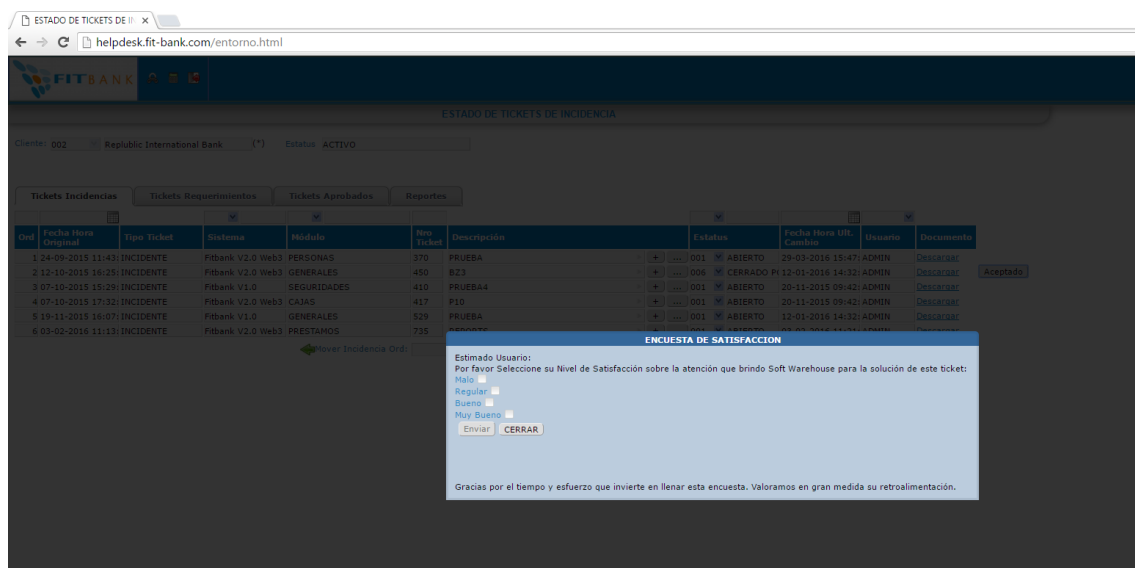
Consulta de Notas Adicionales

La Ventana de Consulta de Notas Adicionales sirve para permitir la visualización de la comunicación entre el consultor asignado y el cliente.



Aceptar Incidente

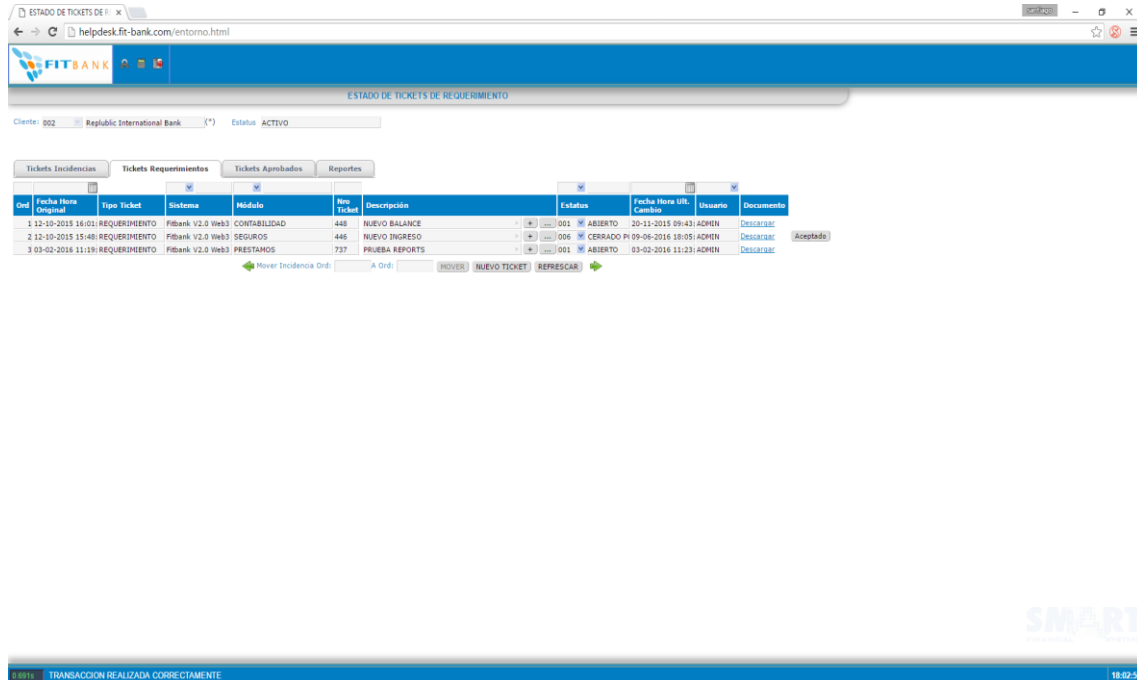
La Ventana de Aceptar Incidente sirve para permitir las funcionalidades F1.4 Aceptar Incidencia donde el cliente puede cerrar definitivamente el seguimiento del mismo.



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

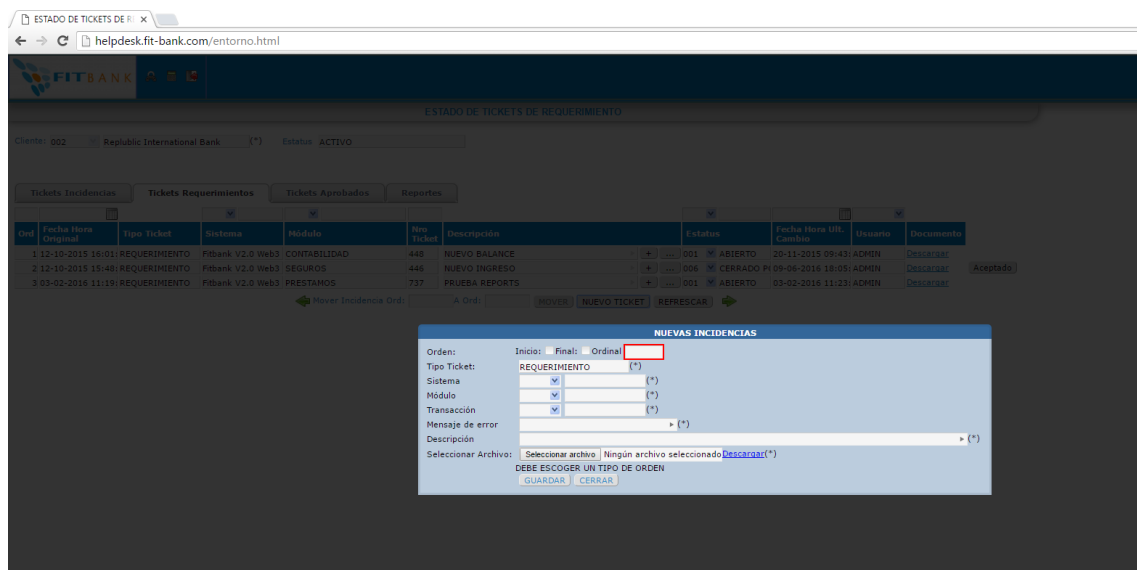
Ventana de Requerimientos

La Venta de Requerimientos sirve para permitir las funcionalidades de F2. Administración de Requerimientos, esta es la ventana principal de los requerimientos, aquí se ejecuta la función F2.2 Consultar Requerimientos y F2.5 Actualizar Estatus



Nuevo Requerimiento

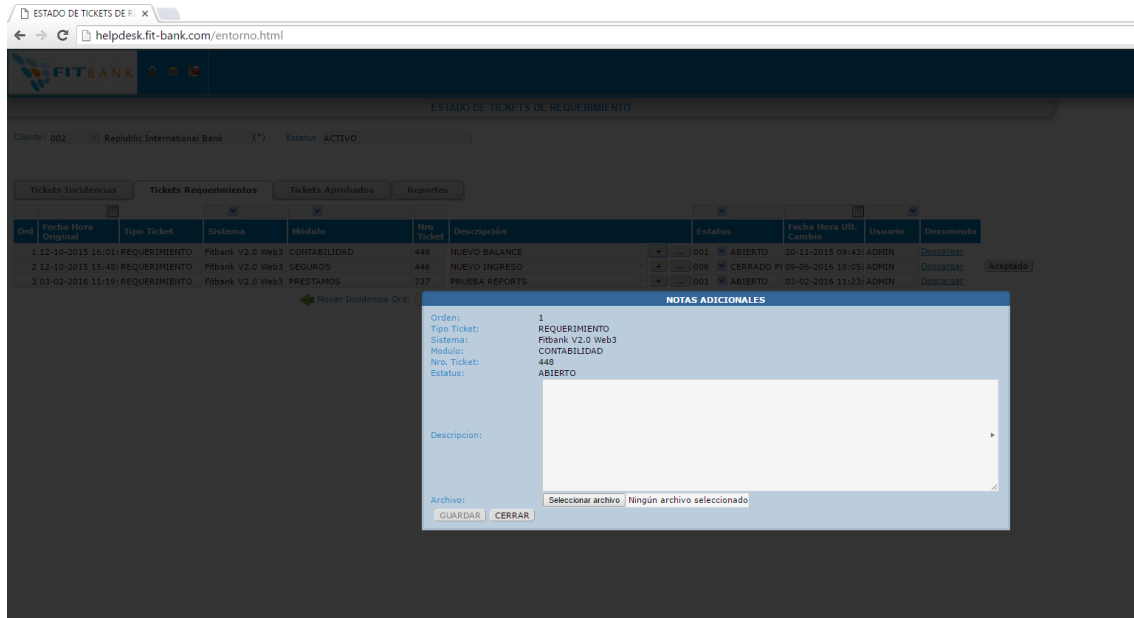
La Ventana de Nuevo Requerimiento sirve para permitir las funcionalidades de F2.1 Ingreso de Requerimientos, esta ventana es donde se asigna la prioridad, modulo y transacción de los requerimientos además de toda la información que lo describe.



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

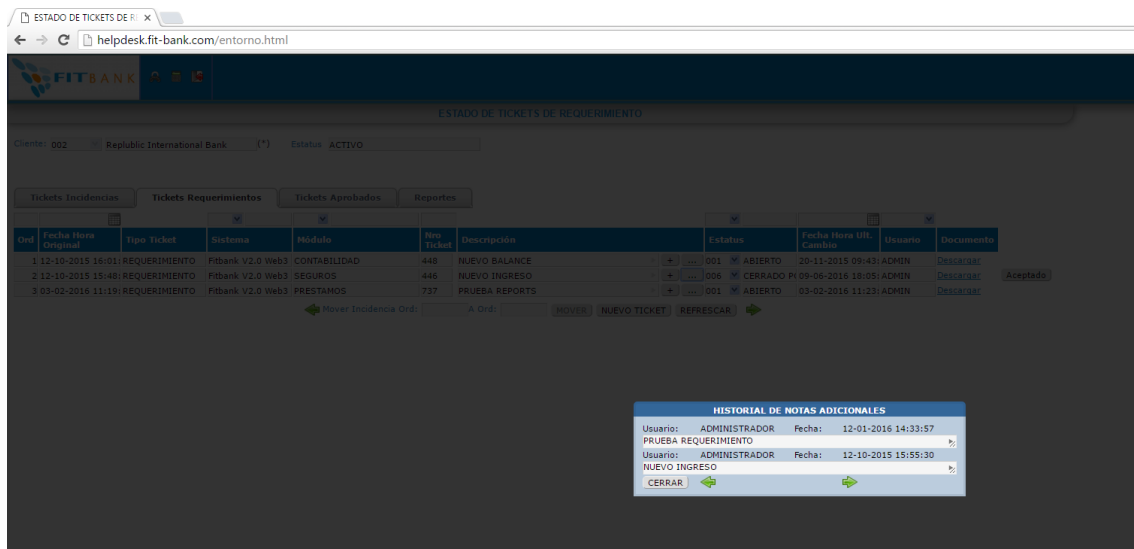
Notas Adicionales

La Ventana de Notas Adicionales sirve para permitir las funcionalidades de F2.7 Ingresar Notas Adicionales, donde se podrá establecer una comunicación directa entre el consultor y el cliente.



Consulta de notas adicionales

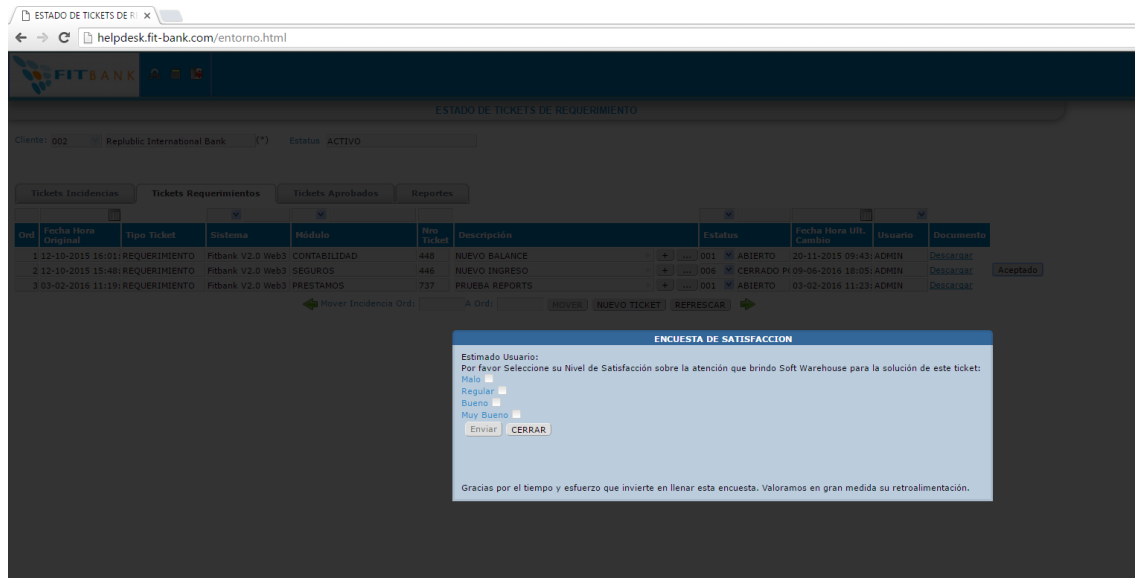
La Ventana de Consulta de Notas Adicionales sirve para permitir la visualización de la comunicación entre el consultor asignado y el cliente.



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

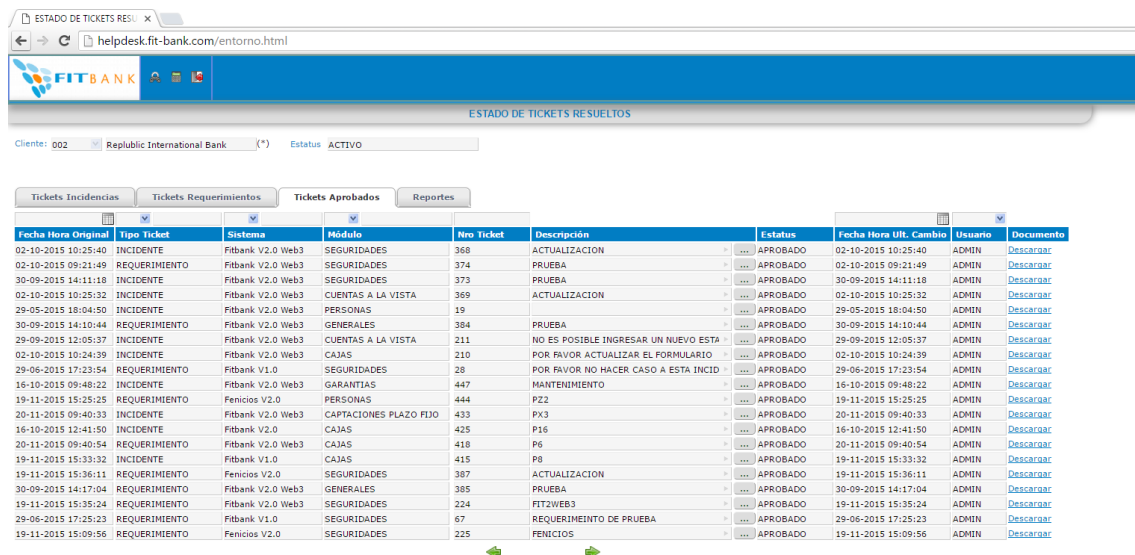
Aceptar Requerimiento

La Ventana de Aceptar Requerimientos sirve para permitir las funcionalidades F2.4 Aceptar Requerimientos donde el cliente puede cerrar definitivamente el seguimiento del mismo.



Tickets Resueltos

La Ventana de Tickets Aprobados sirve para permitir las funcionalidades de F3.Administración de Tickets Resueltos donde el cliente podrá visualizar de forma general todos los tickets ya aprobados.



DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Consulta con Parámetros

La Ventana de Consulta con Parámetros sirve para permitir las funcionalidades de F3.1 Consulta donde el cliente podrá visualizar de forma más específica según los parámetros ingresados los tickets.

Fecha Hora Original	Núm ticket	Sistema	Módulo	Núm ticket	Descripción	Estatus	Fecha Hora Ult. Cambio	Usuario	Documento
02-10-2015 10:25:40	368	Fitbank V2.0 Web3	SEGURIDADES		ACTUALIZACION	APROBADO	02-10-2015 10:25:40	ADMEN	Descargar
30-09-2015 14:11:18	373	Fitbank V2.0 Web3	SEGURIDADES		PRUEBA	APROBADO	30-09-2015 14:11:18	ADMEN	Descargar
02-10-2015 10:25:32	369	Fitbank V2.0 Web3	CUENTAS A LA VISTA		ACTUALIZACION	APROBADO	02-10-2015 10:25:32	ADMEN	Descargar
29-09-2015 10:04:00	19	Fitbank V2.0 Web3	PERSONAS			APROBADO	29-09-2015 10:04:00	ADMEN	Descargar
02-10-2015 10:24:39	210	Fitbank V2.0 Web3	CAJAS		POR FAVOR ACTUALIZAR EL FORMULARIO	APROBADO	02-10-2015 10:24:39	ADMEN	Descargar
29-09-2015 12:05:37	211	Fitbank V2.0 Web3	CUENTAS A LA VISTA		NO ES POSIBLE INGRESAR UN NUEVO ESTA	APROBADO	29-09-2015 12:05:37	ADMEN	Descargar
19-11-2015 15:33:32	415	Fitbank V1.0	CAJAS		P8	APROBADO	19-11-2015 15:33:32	ADMEN	Descargar
16-10-2015 09:48:22	447	Fitbank V2.0 Web3	GARANTIAS		MANTENIMIENTO	APROBADO	16-10-2015 09:48:22	ADMEN	Descargar
20-11-2015 09:48:33	433	Fitbank V2.0 Web3	CAPTACIONES PLAZO FIJO		FX3	APROBADO	20-11-2015 09:48:33	ADMEN	Descargar
16-10-2015 12:41:50	425	Fitbank V2.0	CAJAS		P16	APROBADO	16-10-2015 12:41:50	ADMEN	Descargar
09-10-2015 14:22:04	403	Fitbank V1.0	SEGURIDADES		PRUEBA	APROBADO	09-10-2015 14:22:04	ADMEN	Descargar
19-11-2015 15:55:40	442	Fitbank V1.0	CUENTAS A LA VISTA		PZ1	APROBADO	19-11-2015 15:55:40	ADMEN	Descargar
17-11-2015 11:16:00	431	Fitbank V2.0 Web3	PERSONAS		FX1	APROBADO	17-11-2015 11:16:00	ADMEN	Descargar
16-10-2015 10:25:33	424	Fitbank V2.0 Web3	CAJAS		P15	APROBADO	16-10-2015 10:25:33	ADMEN	Descargar
26-10-2015 16:17:55	422	Fitbank V2.0 Web3	PERSONAS		P13	APROBADO	26-10-2015 16:17:55	ADMEN	Descargar
20-10-2015 11:48:44	421	Fitbank V2.0	CAJAS		P12	APROBADO	20-10-2015 11:48:44	ADMEN	Descargar
20-11-2015 09:42:30	405	Fitbank V1.0	GENERALES		PRUEBA3	APROBADO	20-11-2015 09:42:30	ADMEN	Descargar
19-11-2015 15:28:09	434	Fitbank V1.0	SEGURIDADES		FX3	APROBADO	19-11-2015 15:28:09	ADMEN	Descargar
26-10-2015 16:17:55	423	Fitbank V2.0 Web3	CAJAS		P14	APROBADO	26-10-2015 16:17:55	ADMEN	Descargar
16-10-2015 12:44:17	420	Fitbank V2.0	SEGURIDADES		P11	APROBADO	16-10-2015 12:44:17	ADMEN	Descargar

Generación Reportes de Actividades Help Desk

La Ventana de Generación Reportes de Actividades Help Desk sirve para permitir las funcionalidades de F4.1 Generar Reporte donde el cliente podrá crear reportes según los parámetros que ingrese.

GENERACIÓN REPORTES DE ACTIVIDAD HELP DESK

Cliente: 002 Republic International Bank (*) Estatus: ACTIVO

Tickets Incidencias Tickets Requerimientos Tickets Aprobados Reportes

Fecha Desde: (*)

Fecha Hasta: (*)

Tipo Ticket: [v]

Módulo: [v]

Estatus: [v]

Usuario: [v]

PDF EXCEL

5.2.2. Lista de riesgos

En cualquier modelo de desarrollo de software, existen circunstancias que implican riesgos al proyecto, estos afectan los tiempos de entrega, los costos o hasta las capacidades del sistema.

Por eso es importante tener claro cuáles podrían ser los posibles riesgos que se podrían presentar cuando se está desarrollando la aplicación, los beneficios de saber esto es que los posibles riesgos que afecten al proyecto no provocarán cambios drásticos o demorarán las entregas del proyecto.

En el desarrollo de aplicaciones siempre existen los siguientes riesgos:

- Establecer un tiempo de desarrollo corto, cuando lo comparamos al tamaño del sistema

Si este fuera el caso, se debería llevar un control diario con el cliente para que pueda constatar el avance diario del proyecto, tenerle al tanto de problemas que se van presentando y de posibles cambios en las fechas.

- Limitaciones que se puedan dar por las herramientas usadas

Tener muy en cuenta todas las características de las herramientas que se van a usar y saber qué ventajas y desventajas pueden traer al proyecto. Si se encontrara una desventaja en una herramienta la cual pueda demorar las fechas de entrega del proyecto tener siempre una alternativa de la misma.

- Pérdida de información

Tener la buena práctica de crear respaldos de los módulos que se van desarrollando o usar herramientas como un servidor de repositorios.

5.3. Construcción

5.3.1. Elaboración

Gracias a las herramientas de estructuración, diseño y modelado, la construcción del sistema se facilita en gran proporción y también debido a la guía y previa planeación del sistema por parte de la empresa.

Se partió con un prototipo genérico y probado por un cliente específico, que consta de 4 administraciones cada una con sus consultas generales y por parámetros.

En base a este primer prototipo se hicieron los cambios necesarios para que el sistema se adapte a cualquier cliente de la empresa, y también las clases y entidades para que la delicada manipulación de la información de la empresa y del cliente se sincronice, todo el código implementado para que el sistema esté en funcionamiento está en el CD que se entregara con la disertación de grado.

5.3.2. Pruebas

- Caso de prueba: **F0. Ingreso al Sistema**

Pre-Condiciones: no hay pre-condiciones

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingreso al sistema			
<ul style="list-style-type: none"> - Datos Correctos - Datos incorrectos 	<ul style="list-style-type: none"> - Ingresa al sistema - Mensaje de error 	F0.	Correcto

Post-Condiciones: no hay post-condiciones

- Caso de prueba: **F1. Administración de Incidentes**

Pre-Condiciones: Ejecutar 10 veces F0.

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingreso de incidencia			
<ul style="list-style-type: none"> - Datos Correcto - Datos incorrectos 	<ul style="list-style-type: none"> - Almacena datos - Mensaje de error 	F1.1	Correcto
Consultar Incidencia	<ul style="list-style-type: none"> - Despliega Datos 	F1.2	Correcto
Modificar Incidencia			
<ul style="list-style-type: none"> - Datos Correctos - Datos incorrectos 	<ul style="list-style-type: none"> - Modifica datos - Mensaje de error 	F1.3	Correcto
Aceptar Incidencia	<ul style="list-style-type: none"> - Almacena datos 	F1.4	Correcto
Actualizar estatus	<ul style="list-style-type: none"> - Almacena datos 	F1.5	Correcto
Asignar Consultor	<ul style="list-style-type: none"> - Almacena Datos 	F1.6	Correcto
Ingresar Notas Adicionales			
<ul style="list-style-type: none"> - Datos Correctos - Datos incorrectos 	<ul style="list-style-type: none"> - Almacena datos - Mensaje de error 	F1.7	Correcto

Post-Condiciones: Ejecutar 10 veces F1 como requisito para poder ejecutar F3 y F4

- Caso de prueba: **F2. Administración de Requerimiento**

**DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE
REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS**

Pre-Condiciones: Ejecutar 10 veces F0.

Entradas	Resultado Esperado	Caso de Uso	Estado
Ingreso de requerimiento			
- Datos Correcto - Datos incorrectos	- Almacena datos - Mensaje de error	F2.1	Correcto
Consultar Incidencia	- Despliega Datos	F2.2	Correcto
Modificar requerimiento			
- Datos Correctos - Datos incorrectos	- Modifica datos - Mensaje de error	F2.3	Correcto
Aceptar requerimiento	- Almacena datos	F2.4	Correcto
Actualizar estatus	- Almacena datos	F2.5	Correcto
Asignar Consultor	- Almacena Datos	F2.6	Correcto
Ingresar Notas Adicionales			
- Datos Correctos - Datos incorrectos	- Almacena datos - Mensaje de error	F2.7	Correcto

Post-Condiciones: Ejecutar 10 veces F2 como requisito para poder ejecutar F3 y F4

- Caso de prueba: **F3. Administración de Tickets Resueltos**

Pre-Condiciones: Existencia de tickets de incidencias y requerimientos

Entradas	Resultado Esperado	Caso de Uso	Estado
Consulta Tickets			
- Datos Correcto - Datos incorrectos	- Despliega datos - Mensaje de error	F3.1	Correcto

Post-Condiciones: no hay post-condiciones

- Caso de prueba: **F4. Administración de Reportes**

Pre-Condiciones: Existencia de tickets de incidencias y requerimientos

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Entradas	Resultado Esperado	Caso de Uso	Estado
Generar Reporte <ul style="list-style-type: none">- Datos Correcto- Datos incorrectos	<ul style="list-style-type: none">- Despliega datos- Mensaje de error	F4.1	Correcto

Post-Condiciones: no hay post-condiciones

5.4. Transición

5.4.1. Seguridades

Anteriormente el software de la empresa Soft Warehouse S.A FitBank manejaba el esquema de seguridad embebido en el núcleo del negocio o CORE¹⁷, actualmente se ha realizado los cambios para que se lo pueda manejar igualmente en el CORE o también en la interface de comunicaciones universal o por sus siglas en ingles UCI¹⁸. Esto permite al cliente, configurar sus estrategias de seguridad según el nivel de acceso.

De este modo, se pueden recibir mensajes desde cualquier canal y validar su contenido en la capa de seguridad antes que sea procesada por el UCI para luego enviarlo al núcleo del negocio o CORE financiero.

De igual manera, si manejamos un sistema con múltiples CORE financieros, podemos configurar la estrategia de seguridad para que se dispare en el/los CORE financieros de destino, independientemente de cuál es el origen (capa de comunicación de UCI).

¹⁷ CORE es la palabra que se emplea para referirse al núcleo del negocio, como es un software dedicado a instituciones financieras se puede referir a CORE financiero.

¹⁸ Universal Communication Interface (UCI) es parte de las seguridades de FitBank con el cual los usuarios acceden a los servicios que ofrece la empresa.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

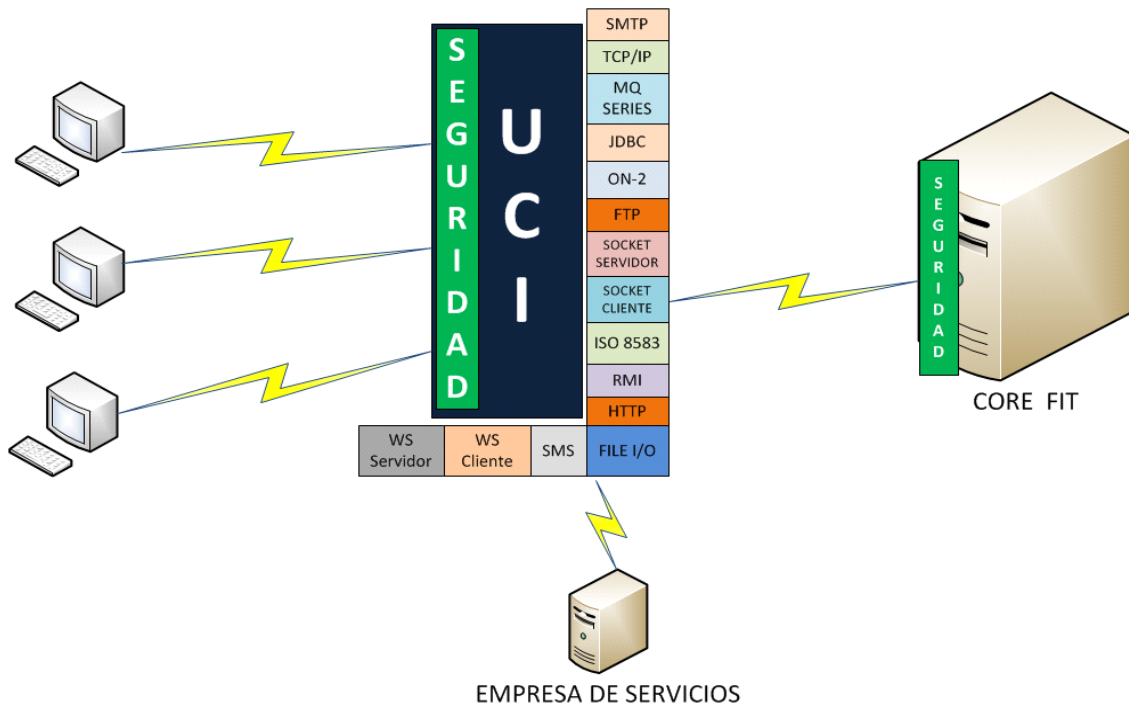


Figura 64 Esquema de Seguridad
Autor: BANTEC inc.

Como se puede apreciar en la Figura 64, el esquema de seguridad está presente tanto en el CORE como en el UCI, la aplicación lee la configuración de la estrategia de seguridad en los archivos de propiedades de la instancia, de este modo puede identificar en que instancia de la aplicación debe realizar las validaciones del caso.

Las clases que se ha modificado para estos cambios son DetailSender.java en el lado del UCI y RequestProcessorBySocket.java por el CORE.

CLASES DE SEGURIDAD

La lógica de clase de seguridad de FitBank se divide en 3 proyectos diferenciados por su procesamiento:

Clases de control de interfaces para procesar comandos de seguridad.

Entre las clases de control de interfaces, está el Procesador de mensajes de tipo SIG (seguridad) llamado SecurityProcessor.java, que es una implementación de Processor.java, que permite lanzar comandos parametrizados bajo una transacción, mediante TransactionScript.

SecurityProcessor.java es el encargado de cargar los comandos del tipo SecurityCommand.java para validaciones de seguridad.

Un comando del tipo SecurityCommand.java pertenece a los comandos de seguridad de la lógica del negocio, descritos en la siguiente sección.

Tomando este concepto inicial, FitBank propone interfaces para definir una lógica específica de Filtros de Seguridad. Estas lógicas pueden ser TransactionScript, BPMFlows, o clases especiales del cliente.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

Una clase del tipo SecurityFilter (filtros de seguridad), debe implementar la interfaz Filter.java, que propone métodos para ejecución de comandos cuando es Login o cuando es un proceso Transaccional.

FitBank propone usar DefaultSecurityFilter.java como estrategia para ejecutar una serie de comandos de validación de seguridad mediante TransactionScript. Estos comandos son parametrizados en la base de datos, bajo un código de transacción.

Además, en caso que el cliente desee disparar autorizaciones para permisos especiales en transacciones, FitBank propone BPMSecurityFilter.java que carga un flujo bpm (jpd) con la lógica de validación y autorización para el proceso en específico.

Estos filtros de seguridad pueden ser ejecutados desde UCI o CORE, según la lógica de negocio del cliente.

Clases de control transaccional que definen la lógica de seguridad del negocio

Estas son las clases (comandos) que manejan toda la lógica de seguridad transaccional del negocio. Están almacenadas en un proyecto especial para que el cliente pueda agregar o modificar cualquier validación según sus requerimientos.

Todos estos comandos extienden de la clase abstracta SecurityCommand.java, que propone métodos de ejecución específicos para ser cargados por SecurityProcessor.java.

Clases de utilitarios para validaciones paramétricas.

Estas son las clases que manejan la configuración de la capa de seguridad del sistema. Entre ellas, esta SecurityHelper.java, que implementa métodos para obtener el origen de ejecución del Filtro de Seguridad (uci, core, otros).

También se encuentra la clase Filter.java, que se ha ubicado en este proyecto específico para poder ser implementada desde cualquier otra parte del sistema, ya que las clases de utilitarios son cargadas desde cualquier lugar, sin ligar dependencias.

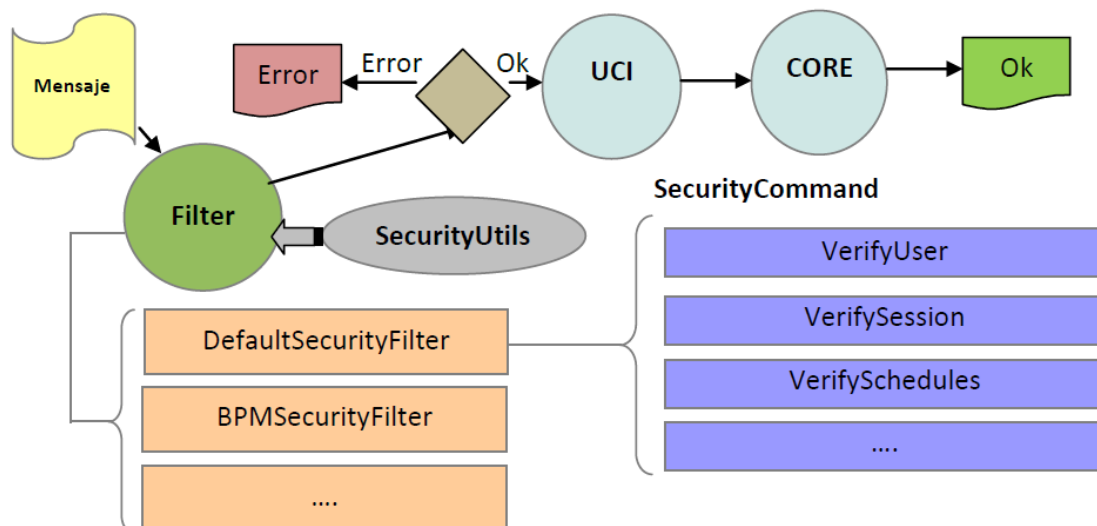


Figura 65 Flujo de un mensaje procesado por el Filtro de Seguridad desde UCI
Autor: BANTEC inc.

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

En la Figura 65 podemos ver el flujo que debe seguir un mensaje, o lo que es lo mismo un evento o acción que se genera en la capa del cliente del sistema.

Como podemos ver al principio siempre pasara por el filtro donde las clases de seguridad verificaran que todo esté funcionando bien y después mandarlo al UCI y después al CORE.

5.4.2. Costos

Los costos del proyecto están especificados en la Tabla 10, se toman en cuenta costos que la empresa ya tenía asumido pero que sin estos no se podría implementar el sistema tales como servidores o infraestructura de red.

Los períodos en esta tabla van a ser considerados como períodos mensuales.

Costos Directos	Periodo 1 Piloto	Periodo 2	Periodo 3	Total Ciclo de vida
Hardware				
Servidor	\$600	\$ 560	\$616	\$ 1.776
Periféricos	\$100	\$ 140	\$ 120	\$ 360
Componentes de Red	\$ 40	\$ 30	\$ 40	\$ 110
Total Costo de Hardware	\$ 740	\$ 730	\$ 776	\$ 2.246
Software				
Licencias	\$ -	\$ -	\$ -	\$ -
Mantenimiento SW	\$ 375	\$ 750	\$ 750	\$ 1.875
Desarrollo SW	\$700	\$ 700	\$ 700	\$ 2.100
Administración SW	\$1.500	\$ 1.500	\$1.500	\$ 4.500
Total Costos de Software	\$2.575	\$ 2.950	\$ 2.950	\$ 8.475
Administración				
Hosting	\$ 300	\$ 300	\$ 300	\$ 900
Informes	\$ -	\$ -	\$ -	\$ -
Total Costo de Administración (Sueldos del equipo de trabajo: administrador y consultores)	\$39.000	\$ 39.000	\$ 39.000	\$ 117.000

Tabla 10 Especificación de Costos del sistema HelpDesk
Autor: Santiago Gálvez

5.4.3. Despliegue

El sistema, se instalará bajo las especificaciones dadas en los requerimientos no funcionales.

En el caso de que existieran problemas con las especificaciones antes mencionadas, se deberá proceder a realizar nuevos requerimientos no funcionales que abarquen

DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS

software y hardware por igual, también se fijaría una nueva fecha de entrega considerando el impacto de los cambios realizados.

5.4.4. Mantenimiento

El mantenimiento del sistema se lo realizará una vez por semana por parte de cualquier miembro del equipo de desarrollo de la compañía o cuando exista un requerimiento de cambio, en este caso el mantenimiento del sistema será de forma inmediata, previa aprobación de parte del encargado del área de desarrollo.

El mantenimiento en la base de datos es fundamental y muy importante ya que una de las principales funciones de este sistema es crear históricos, por eso la base de datos deberá contar siempre con espacio suficiente así como también tendrá accesos limitados para mantener la información a salvo.

Un protocolo de respaldos también se implementará semanalmente para conservar respaldos de los datos de la base.

6. CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

Al finalizar el presente trabajo, es necesario recoger en este capítulo las principales observaciones a través de las siguientes secciones.

6.1. Conclusiones

- La metodología de desarrollo ágil “Rapid Application Development” (RAD) permite desarrollar aplicaciones de software siguiendo lineamientos bastante ágiles, los cuales son apropiados también para escalar desarrollo ágil a grandes empresas.
- Diseñar un sistema estructurado por capas es bastante complejo de estructurarlo especialmente al inicio del proyecto, por eso es muy importante tener una idea clara de lo que va a ser el producto.
- La selección de las herramientas y técnicas adecuadas, según el tamaño o condición del proyecto, es de mucha ayuda para reducir tiempos de entrega y en caso de ser necesario, reducir costos de producción.
- Las historias de usuario, diagramas UML y los casos de uso nos permiten apreciar visualmente las potenciales funcionalidades del sistema.
- El desarrollo mediante la utilización de definición de historias de usuario, tareas y metas, facilita mucho el trabajo del desarrollador, esto se logra gracias a que se construye un entendimiento previo a la programación.
- Al momento de la implementación de las capas del proyecto esos van a depender mucho de los diagramas realizados en un principio, con éstos, se logra entender la vinculación entre objetos y las dependencias necesarias en cada uno.
- Al momento de escoger las herramientas y técnicas para el desarrollo del sistema se debe ser muy cuidadoso ya que deben ser de mucha ayuda para reducir tiempos y en algunos casos también reducir costos.
- Es muy importante tener en cuenta los procesos que existen en la organización para la que se está elaborando el sistema, así se podrá definir los requerimientos y satisfacer las necesidades de los usuarios finales.
- Nunca perder el objetivo en la automatización de un proceso para las organizaciones o empresas, esto es ahorrar tiempo.
- Al tener al cliente prácticamente probando el sistema a diario fue realmente fácil la resolución de problemas, tanto en la fase de pruebas como al dar soporte al sistema.
- Es muy conveniente si se está haciendo un proyecto para una empresa grande ayudarse del conocimiento de ingenieros que entienden muy bien el sistema y más aún si estas familiarizados con el proceso a automatizar.

6.2. Recomendaciones

- Queda muy recomendado utilizar el acercamiento ofrecido por RAD para la implementación de marcos de trabajo de desarrollo de software ágiles en grandes empresas.
- Facilita mucho el trabajo del desarrollador si se profundiza el conocimiento sobre los procesos del negocio, así se puede tener buenas bases previas a cualquier especificación de requerimientos de diseño o de software.
- Administrar el tiempo de manera efectiva si se maneja un entorno de alta presión y stress, para lograr cubrir todas las actividades y responsabilidades de múltiples roles en una empresa.
- Una vez que se escoge una metodología seguir los pasos como se estableció en el diseño, de esta forma se cumplirán los tiempos entregas del sistema y así mantener la confianza del cliente que es algo sumamente importante.
- Se recomienda en lo posible mantener un ambiente de confianza con el cliente, de este modo podremos tratar problemas que se puedan presentar en el desarrollo del sistema como temas para mejorar el sistema y no debilidades.
- Se recomienda fomentar más la enseñanza de nuevas maneras y herramientas para desarrollar en vez de enfocarse en las antiguas, que a pesar de que son por donde todo desarrollador debe comenzar, no son las que se usan en la vida profesional.

**DESARROLLO DE UN SISTEMA PARA LA ADMINISTRACIÓN DE
REQUERIMIENTOS E INCIDENCIAS MEDIANTE TICKETS**

ANEXOS

Abreviaturas	Definiciones
XP	Extreme Programming, metodología de desarrollo de aplicaciones.
RAD	Este término nos provee el autor a continuación mencionado: “El desarrollo rápido de aplicaciones o RAD (Rapid Application Development) es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980”. (Curiosisimos, 2009)
CASE	Son programas informáticos destinados a incrementar la productividad en el desarrollo de programas reduciendo el tiempo y costo de las mismas.
JAD	JAD es el acrónimo de Joint Application Development, y es una práctica de definición de requisitos y de diseño de la interfaz de usuario, basada en reuniones entre los principales involucrados.
BD	Este término viene de: “Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso”. (Orueta, 2008)
SGBD	Esta abreviación, según el autor es: Es una abreviación de Sistemas gestores de bases de datos, estos nos dan la ventaja de poder acceder y manejar los datos de una manera ordenada y sencilla.
P2P	Es una red de ordenadores en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.
RFC	De las siglas Request for change o Peticiones de cambio
PIR	De las siglas Post- Implementation Reviews o Revisiones Post-Implementación
ORM	Mapeo de objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.
XML	Lenguaje de marcas extensible.
GNU LGPL	GNU Lesser General Public License / Licencia de Software en la que cualquier persona tiene permisos de copia y distribución.
SIPOC	Estas son las siglas de Supplier Inputs Process Outputs Customers.
GAC	Registrar en Gerencia de Aseguramiento de Calidad.
DML	lenguaje proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos.

BIBLIOGRAFÍA

Capítulo 1:

- FITBANK. (n.d.). Retrieved October 06, 2016, from <https://fit-bank.com/>

Capítulo 2:

2.1

Extreme Programming

- UNIVERSIDAD NACIONAL EXPERIMENTAL... - blogs.unellez.edu.ve. (n.d.). Retrieved March 10, 2016, from <http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>
- Programación Extrema | Lugar de coincidencia en Internet... (n.d.). Retrieved March 10, 2016, from <http://www.cyclopaedia.es/wiki/Programacion-Extrema-1>
- XP - Extreme Programing Ingenieria de Software. (n.d.). Retrieved March 10, 2016, from http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html

RAD

- MODELO RAPIDO DE DESARROLLO DE APLICACIONES. (2011). Retrieved March 10, 2016, from <https://lhija.wordpress.com/2011/10/28/modelo-rapido-de-desarrollo-de-aplicaciones/>

JAD

- McConnell, S. (1996). *Rapid development: Taming wild software schedules*. Redmond, WA: Microsoft Press.

2.2

Bases de Datos

- Produce: ¿Quieres aprender más sobre el tema Base de Datos? (n.d.). Retrieved March 10, 2016, from <http://inducccion.educatic.unam.mx/mod/page/view.php?id=979>

Herramientas CASE

- Kendall & Kendall *Análisis Y Diseño De Sistemas 3ª. Edición*
Páginas 15.16.17.18
- Herramientas CASE. (n.d.). Retrieved March 10, 2016, from <http://sistemas324.blogspot.com/2008/09/bibliografia.html>
- HERRAMIENTAS CASE. (n.d.). Retrieved March 10, 2016, from <http://herramientas-case.blogspot.com/2008/09/7bibliografia-sitios-web.html>
- Overview. (n.d.). Retrieved April 21, 2016, from <http://powerdesigner.de/en/overview/>

2.4 Servicios de Tecnología

- ITIL®FoundationGestión de servicios TI. (n.d.). Retrieved March 10, 2016, from <http://itilv3.osiatis.es/>