



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Desarrollo de una Aplicación Web para el
Centro de Cómputo de una Procesadora
de Medios de Pago que automatice
la Gestión de Incidencias.
Trabajo Especial de Grado

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
Por los Bachilleres
Jennifer Carolina Lugo Moreno
Nelwin Alfredo Rodríguez Mavarez
para optar al título de
Licenciado en Computación

Tutor: Prof. Johnny Sepúlveda

Caracas, Febrero de 2010

**Asunto: “Desarrollo de una Aplicación Web para el Centro de Cómputo de
una Procesadora de Medios de Pago que automatice la Gestión de
Incidencias”**

Autores: Jennifer Lugo
y Nelwin Rodriguez
Tutor: Prof. Johnny Sepúlveda
Fecha: 08de Febrero de 2010

RESUMEN

El presente Trabajo Especial de Grado expone el desarrollo de una aplicación Web capaz de gestionar eficientemente, las incidencias ocurridas en el Centro de Cómputo. La Aplicación desarrollada deberá facilitar, el registro actualización, consulta y seguimiento de las incidencias. Adicionalmente, deberá mejorar el rendimiento, productividad y capacidad de respuesta al personal del Centro de Cómputo.

La Aplicación se desarrolló siguiendo una metodología denominada Proceso Unificado, la cual utiliza como una de sus herramientas principales los diagramas de casos de uso para representar las funcionalidades y requerimientos de la aplicación, se caracteriza por ser capaz de adaptarse a cambios de requisitos en cualquier punto del ciclo de desarrollo de la aplicación y estimula el continuo contacto que debe mantenerse entre el desarrollador y el cliente.

Para la implementación se utiliza como herramientas de desarrollo, PHP (Hypertext Pre-processor), Sistema Manejador de Bases de Datos MySQL y JavaScrip, estas herramientas se caracterizan por ser Software Libre y juntas proveen las funcionalidades necesarias para la construcción de la aplicación.

El resultado obtenido es una aplicación sencilla de manejar, que logra optimizar y facilitar el proceso de gestión de incidencias, a través de una infraestructura tecnológica que abre las puertas hacia un crecimiento competitivo y permanente.

Palabras Claves: Medios de pago, Procesadora de medios de pago, Centro de Cómputo, Incidencias, Sistema de Registro de Incidencias.

ACTA

Quienes suscriben, miembros del Jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado “Desarrollo de una Aplicación Web para el Centro de Computo de una Procesadora de Medios de Pago que Automate la Gestión de Incidencias” y presentado por la Br. Jennifer Lugo (C.I. 16.007.936) y el Br. Nelwin Rodriguez (16.117.818), a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del Jurado, se fijó el día 08 de Febrero de 2010, a la 10 horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala 1 de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 18 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 8 de Febrero de 2010.

Prof. Johnny Sepúlveda
Tutor

Prof. Adrián Bottini
Jurado

Prof. Silvia Díaz
Jurado

Agradecimientos y Dedicatorias

Dedico este trabajo en primer lugar a Dios Todopoderoso, por guiarme, iluminarme y acompañarme, a el agradezco todo lo que tengo y lo que soy.

A mi querida madre, Thaiz por acompañarme en cada momento de mi vida. Por su apoyo incondicional, por confiar en mí y por ese espíritu de lucha que solo tú tienes, con el que me enseñaste a vivir, a cumplir sueños y que hoy me trae hasta aquí. Gracias por ser quien eres, sin ti nunca lo hubiera logrado TQM.

A mi Padre, por estar siempre a mi lado y brindarme su cariño y apoyo.

A mi Hermano, por ser mi amigo incondicional, mi vida sin ti no sería la misma, gracias por ser como eres TQM.

A Todos los que llevan el apellido "MORENO", porque han sido mi gran familia, ejemplo digno de unión, gracias por que si volviera a nacer definitivamente estaría de nuevo entre ustedes.

A mi Esposo, por darme valor cuando lo necesitaba, por apoyarme y confiar siempre en mí.

A mis Abuelos (Kanon y Carlota), gracias por todos los años de paciencia y ternura que recibí, a ustedes debo gran parte de lo que soy. Kanon, gracias por seguir siendo el mismo abuelo cariñoso y consentidor es una dicha el poder tenerte y a ti abuela que no estas físicamente hoy conmigo gracias porque nunca has dejado de acompañarme, gracias por todo lo momentos que viví a tu lado, por tus consejos, y ese cariño especial que recibí de ti. Fuiste y serás siempre parte de este corazón.

A mis principales compañeros de estudio, con quienes compartí mis mejores momentos en la Universidad, desde hacer los proyectos más difíciles, hasta salir a divertirnos y olvidarnos de ellos por un momento.

A mi compañero de tesis, por ayudarme y compartir juntos este arduo trabajo.

A todas aquellas personas que de una manera u otra estuvieron presentes a lo larga de esta etapá que hoy culmina llena éxito, satisfacción, alegrías y recuerdos inolvidables.

Gracias a todos

Jennifer

Dedico este trabajo en primer lugar a Dios, por acompañarme en cada etapa de mi vida.

También lo dedico muy especialmente a mis padres, quienes siempre fueron una guía y siempre me dieron fuerzas para avanzar en este arduo camino que llaman vida. A ellos les ofrezco esta meta cumplida y sus frutos.

A toda mi familia, los que están cerca y los que están lejos, por su cariño y apoyo que he recibido.

A todos mis amigos, porque siempre fueron quienes me otorgaron ratos de felicidad en mi vida.

A todas aquellas personas, algunas pasajeras, otras aun en espera de un destino que nos cruce de nuevo, que en algún momento me dieron algo (palabras, risas, experiencias, sentimientos) que realmente provocaron un crecimiento en mí, y me ayudaron a trascender, evolucionar y vivir muy buenos momentos.

Gracias a todos

Nelwin

A el Profesor Johnny Sepúlveda, por aceptar ser nuestro tutor y apoyarnos en el desarrollo de este trabajo y darnos su orientación y confianza, por el respaldo que nos brindó y todos sus asesoramientos metodológicos en la realización de este Trabajo de Grado, aportándonos siempre sus orientaciones académicas, sin dejarnos a un lado al estar siempre presente y al tanto de nuestro trabajo día a día y por su rápida respuesta a las constantes peticiones hechas, lo cual le agradeceremos enormemente, sobre todo gracias por el compromiso y la responsabilidad.

A los profesores que fueron parte del jurado (Profa. Silvia Díaz y Prof. Adrián Bottini) por guiarnos en la culminación de este trabajo especial de grado gracias a su efectiva colaboración para lograrlo.

A todos aquellos profesores que de una manera u otra nos dieron y dejaron su gran aporte al ejercer su papel de docentes preocupándose realmente por enseñar para que ese aporte lo logremos utilizar al salir al verdadero y complicado mundo como lo es el “laboral” donde finalmente y realmente aplicaremos nuestros conocimientos adquiridos gracias a ellos.

A todas aquellas personas que de una manera u otra forma estuvieron presentes aportando su granito de arena, para que este proyecto llegara a feliz término.

Jennifer Lugo
Nelwin Rodríguez

Índice de Figuras

Índice de Tablas

Introducción

La tecnología está hecha para ser utilizada y para aprovechar sus potencialidades en aras de mejorar y simplificar las tareas del día a día. Los profesionales en el área de computación se encuentran estrechamente relacionados con los avances tecnológicos y deben crecer a la par con ellos, fomentando su uso para beneficiar a cualquier persona u organización que requiera mejorar y actualizar los procesos que llevan a cabo.

El apoyo a las nuevas tecnologías y el fomentar la construcción de aplicaciones que automaticen y faciliten los procesos dentro de las organizaciones, contribuyen al ahorro significativo de tiempo, dinero y esfuerzo; y constituye la principal motivación para el desarrollo de este trabajo, siendo el Centro de Cómputo (CECOMP) un ejemplo de éste tipo de organizaciones que requieren de una reingeniería de procesos y están abiertas a las propuestas de soluciones innovadoras que maximicen el nivel de eficiencia de sus procesos, junto con la satisfacción de todo el personal que labora dentro del ente tecnológico.

En tal sentido, el objetivo fundamental de este Trabajo Especial de Grado consiste en implementar una aplicación Web que automatice y al mismo tiempo optimice los procesos de Gestión de Incidencias que se llevan a cabo durante la labor diaria del Centro de Cómputo.

Para la mejor comprensión del lector este documento se ha estructurado en los siguientes capítulos:

Capítulo 1: Planteamiento del Problema y Situación Actual. Se describe el mundo de las procesadoras de medios de pagos, mostrando conceptos básicos, el caso de negocio, y la situación actual de la misma y se presenta el problema a atacar en este Trabajo Especial de Grado.

Capítulo 2: Solución Propuesta. Se presenta una solución para el problema planteado, su alcance, importancia y justificación, evaluando los beneficios que se generarán, al igual que se especifican las distintas herramientas tecnológicas que se utilizan para el desarrollo de la solución.

Capítulo 3: Metodología. Se presenta la metodología de desarrollo que se escogió para el desarrollo de la solución planteada, explicando cada una de las fases que la conforman.

Capítulo 4: Especificación de Requerimientos. Se especifica como fueron obtenidos los requerimientos, se muestran los casos de uso y su respectiva descripción.

Capítulo 5: Análisis del Sistema. En este capítulo se describen las actividades que transformaran los diagramas de casos de uso definidos durante la recopilación de requerimientos en los diferentes modelos de análisis, así como el diagrama de secuencia, mostrando la iteración de los objetos de entidad, frontera y control.

Capítulo 6: Diseño del Sistema. Aquí se presentan los patrones de diseño que rigieron la creación y el diseño de la interfaz, y el modelo de la Base de Datos del sistema.

Capítulo 7: Construcción del Sistema. En este capítulo se presenta el proceso de realización del sistema, describiendo los métodos que se utilizaron, las pruebas realizadas al sistema con sus respectivos resultados, y el proceso de implantación del mismo.

Capítulo I: Planteamiento del Problema y Situación Actual.

Hoy en día, debido a las diversas actividades que se desarrollan tanto en el ámbito laboral como personal, es necesario tener procesos que permitan administrar de forma adecuada el tiempo y los recursos con los que se cuentan, sobretodo en los ambientes donde se implementen soluciones tecnológicas, debido a que lamentablemente dichas soluciones no siempre resultan ser infalibles, sino que por distintos motivos pueden llegar a fallar y así ocasionar que dichas actividades no se logren cumplir de la manera deseada.

Tal es el caso de la procesadora de medios de pago, ésta en sí es una solución tecnológica, que ofrece distintos servicios a instituciones bancarias con el fin de que dichas instituciones se enfoquen más a lo que a finanzas se refiere y la procesadora se ocupe de la parte de infraestructura tecnológica, manteniendo la correcta operatividad de los sistemas involucrados, ya que por diversas razones los mismos pueden sufrir fallas, presentar retardos o simplemente no funcionar según los acuerdos estipulados entre las entidades bancarias y la procesadora de medios de pago. Dichas fallas las llamaremos incidencias, aquí es donde entra en juego el Centro de Cómputo, ya que esta unidad es la encargada de velar por el correcto funcionamiento de los sistemas en su rol de *Servicio de Ayuda*.

El *Servicio de Ayuda* presenta un rezago en cuanto a la utilización de recursos tecnológicos que faciliten la ejecución de todos los procesos adscritos a su función, tales como: recopilación de los datos necesarios para registrar la incidencia, escalarla eficientemente para su rápida resolución y administrar las experiencias acumuladas en la solución de las fallas. A continuación, se presenta la cadena de procesos y situaciones que se originan cuando se presenta una incidencia.

Registro de la incidencia: Después que el operador verifica la ocurrencia de una incidencia debe llenar la planilla de control, la cual está creada en un documento de Microsoft Excel. En esta planilla se colocan todos los datos de la persona contacto que reporta la incidencia, sea interna o externa (nombre, apellido, empresa para la que trabaja, cargo que ocupa, medio de comunicación a través del cual reporta), hora en que se inició la incidencia, hora de comunicación, datos del operador que da inicio a la incidencia y la descripción detallada del problema. Posteriormente, clasifica la incidencia según el servicio que afecta y la prioridad que debe tener su resolución. Después que el operador escribe todos estos datos, genera un número de ticket y se lo comunica al contacto que reporta la incidencia para poder realizar el seguimiento de la misma, recién en este momento el operador comienza a buscar la solución del problema, ya que el operador, a su vez, es el primer nivel de soporte.

Problema de tiempo al registrar y clasificar las incidencias: El tiempo que conlleva la realización de toda la actividad de registro y clasificación de la incidencia, depende exclusivamente de la velocidad con que el operador pueda tomar y escribir estos datos y en la medida en que tarde más o menos tiempo se postergará proporcionalmente el inicio de la etapa de solución del problema. Adicionalmente, es común conseguir en la “planilla de control” tales como: incidencias que no han sido correctamente clasificadas, incidencias con datos errados de las personas que comunican la falla, incidencias que han sido reportadas por personal del cliente que no están autorizados por la procesadora de medios de

pago, etc. Muchas de estas fallas son reportadas por desconocimiento del software o hardware que manejan, y no porque sean fallas técnicas de la plataforma.

Asignación y escalado de la incidencia: Todas las incidencias cerradas se encuentran en carpetas archivadas para que puedan ser consultadas por el operador en su función de soporte de primer nivel y así, en caso de que vuelvan a ocurrir, tener la información disponible para solucionar la falla lo más pronto posible. Cuando el operador no puede solucionar la falla en un tiempo prudencial, la remite al personal de soporte de segundo nivel. El operador, en este caso, debe llenar los datos de la persona a quien escala el caso y estar constantemente comunicado con ella para que le de una nueva descripción detallada y actualizada del problema. En caso de que este personal tarde mucho tiempo y no resuelva la falla debe promover el escalado a un tercer nivel con la finalidad de no incumplir los niveles de servicios acordados con el cliente. Al realizar el escalado, el operador debe solicitar nuevamente una descripción actualizada del problema y escribir en la planilla de control de incidencias todos los datos del soporte de tercer nivel e igualmente estar constantemente en contacto con dicho soporte para poder mantener al cliente informado, en caso de que este lo requiera utilizando el número de ticket que se le entregó al momento de abrir la incidencia.

Problema de Almacenamiento y Búsqueda : El tiempo que el operador debe invertir en consultar el archivo de incidencias cerradas, la incertidumbre de no saber si encontrará allí la solución que aplique a la falla actual y lo poco amigable visualmente que resultan estas ingentes cantidades de papel hacen que la consulta realizada a estos archivos sea casi inexistente, por tanto, cada vez que ocurre una incidencia, que había ocurrido anteriormente, se pueden presentar dos casos que dificultan la solución del problema: 1.- En las oportunidades anteriores en que se presentó la incidencia, fue atendida por un operador diferente al que la está atendiendo actualmente (motivado a la rotación de turnos). 2.- Puede que el operador, aunque haya atendido la incidencia, no recuerde la solución, por tanto comienza nuevamente la búsqueda de la solución, incluyendo el escalado de la incidencia, traduciendo todo esto en demoras para restablecer el servicio afectado.

Adicionalmente, el operador invierte grandes cantidades de tiempo en el llenado de las planillas, inclusive, al tener que escribir siempre los datos de las personas que figuren como soporte de segundo y tercer nivel que, por lo general, son las mismas, esto trae como consecuencia que se le acumulen ejecuciones de procesos y omisiones en la inspección de los sistemas de monitoreo con el consiguiente resultado negativo para el CECOMP y para la procesadora de medios de pago en general.

Cierre de Incidencias: El *Servicio de Ayuda* es el único encargado de cerrar las incidencias y esto lo realiza después de haber requerido la conformidad del cliente y recibir una respuesta satisfactoria, después de lo cual llena en la “planilla de control” todos los datos de la persona que autoriza el cierre de la incidencia por parte del cliente y posteriormente imprime y archiva la planilla en las carpetas de incidencias cerradas.

Problema de Actualización y Localización: En muchas ocasiones, el operador, después de haber recibido la conformidad del cliente posterga el cierre del incidente para ejecutar algún proceso o realizar labores de monitoreo y luego,

por omisión o falta de tiempo, no la cierra definitivamente o no la archiva, perdiéndose esta información para los archivos.

Gestión de Incidencias: Este subproceso tiene como objetivo realizar un seguimiento de las incidencias periódicamente, para comprobar que se están cumpliendo en todo momento los niveles de servicio acordados.

El éxito de la Gestión de Incidencias se demuestra a través de:

- La reducción de los tiempos necesarios para la resolución de las incidencias.
- La disminución de los costes asociados a la resolución.

Para conseguir estos objetivos es imprescindible:

- Capturar indicadores que permitan medir el proceso.
- Verificar el grado de cumplimiento del proceso establecido mediante la realización de auditorías periódicas.

Problema de Tiempo de Respuesta: Todos los intentos que se han hecho para activar este proceso han fallado, por la falta de un método eficiente y perdurable que permita la extracción de la data de las planillas de control de incidencia y de la data de los archivos con una inversión mínima de tiempo, por tanto, no se está logrando reducir los tiempos de atención de las incidencias ni los costos asociados a estas (tiempo del personal en la resolución de la incidencia, tiempo que se deja de invertir en otros procesos fundamentales, las multas impuestas por incumplimientos en los niveles de servicio acordado y desgaste de la buena imagen de la empresa).

I.1. Situación Actual

En esta sección se describe el marco teórico referido a la industria de los medios de pago, en el cuál se analiza el modelo básico de este sistema, las herramientas involucradas: tarjetas de crédito o tarjetas de débito, para así llegar a mostrar el papel que juegan las procesadoras de medios de pago.

I.1.1. MEDIOS DE PAGOS.

Desde tiempos muy remotos los seres humanos han utilizado diferentes medios para poder adquirir bienes y servicios, y así poder satisfacer sus necesidades elementales. Al pasar de los años las formas de pago se han diversificado y especializado de la mano con la globalización. A tal punto que ya no se hace imperativo que el cliente este presente al realizar una transacción, y mucho menos

que él necesite cancelar con papel moneda, lográndose así un beneficio sustancial de tiempo, seguridad y una capacidad de penetración de mercado sin límite.

De otra forma surge un modelo de pago el cual se basa en el soporte de una entidad financiera básica, la cual será la encargada de la intermediación entre los cobros y los pagos, apoyando así las relaciones comerciales entre vendedores y consumidores de bienes y servicios.

Este modelo será el garante de que un vendedor o comerciante pueda realizar una venta a un comprador o cliente final que como forma de pago empleará a una entidad financiera y esta última garantizará el pago al banco o entidad de ahorro en la que el comercio vendedor mantiene sus depósitos, realizando estas dos entidades financieras todo un intercambio, compensación y liquidación de flujos monetarios para poder realizar el cobro/pago derivado de la venta/contrato comercial realizado entre vendedor y comprador.



Figura Nº 1. Ejemplo de uso de un Medio de Pago

Los elementos que conforman el modelo son las siguientes:

- El cliente final, comprador.
- El comercio, vendedor (*merchant*).
- La entidad financiera emisora, entidad financiera del cliente final que emite alguna forma de título garantizando y respaldando el pago de su cliente.
- La entidad financiera adquirente, entidad financiera del comercio, que adquiere los títulos anteriores y los deposita a su cliente, el comercio.
- Un sistema para intercambiar, compensar y liquidar estos títulos para que se produzcan de forma segura los flujos monetarios entre la entidad emisora y la adquirente.

El esquema tuvo un desarrollo fundamental a partir de la aplicación de tecnología para poder realizar de forma electrónica todo el proceso de transferencia de fondos. Precisamente con este fin surgió el concepto de tarjeta de crédito con

datos identificativos del cliente final almacenados en la banda magnética de la tarjeta y un terminal de captura electrónica de estos datos (punto de venta) que sirve al comercio para iniciar todo el proceso que desemboca finalmente en la transferencia de fondos desde la entidad del cliente final a la entidad del comercio, cerrando el proceso de venta comercial en el plano económico.

Este es el modelo básico de la industria de los medios de pago, denominado modelo cuatripartito (por los 4 actores que intervienen: cliente final, comercio, entidad emisora y entidad adquirente). Este sistema produce un 'efecto red', es decir, cuantas más tarjetas se emiten más comercios aceptan su uso lo que a su vez potencia de nuevo una mayor emisión de tarjetas. [Cas-2008]

I.1.2. CONCEPTOS EMPLEADOS EN EL MODELO: MEDIOS DE PAGOS Y CENTROS DE PROCESOS.

Para el desarrollo óptimo de este modelo surgen diferentes esquemas, por un lado unas marcas comerciales -marcas de los productos del sistema - y por otro unas reglas de operación para sistematizar todo el proceso de transacción, compensación y liquidación. Adicionalmente también surgen entidades de apoyo tecnológico/operativo para realizar los procesos de autorización de operaciones, intercambio/compensación y liquidación, se trata de los centros procesadores.

Se dice así que un medio de pago es la suma de una marca comercial y de la tecnología de proceso y redes (TPV, Cajeros).

I.1.3. BASE ECONÓMICA DEL MODELO.

Gracias a este modelo se inicia el desarrollo de los medios de pago electrónicos basados en tarjetas y redes terminales, las franquicias, los centros de procesos que intercambian, compensan y liquidan operaciones entre entidades financieras, las entidades emisoras de tarjetas y las entidades adquirentes de operaciones de puntos de venta.

Sin embargo, para que el modelo sea sostenible en el plano económico, se definió un concepto económico fundamental: la tasa de intercambio, que constituye un pago de la entidad financiera adquirente a la entidad financiera emisora de la tarjeta. Además, también supone una remuneración a la entidad emisora por su esfuerzo de implantar en sus clientes finales esas tarjetas que permiten realizar de forma electrónica la garantía y el pago, así como iniciar el proceso de transferencia electrónica de fondos. De esta forma, se constituía el factor económico fundamental del caso de negocios de la industria.

El comercio, que llega a un libre acuerdo con su entidad adquirente por el cual al recoger esta entidad las transacciones en su punto de venta realiza el abono al comercio del importe de la venta, admite una comisión de la entidad adquirente. Por otra parte, la entidad adquirente realiza un pago al emisor de la tarjeta conforme la tasa de intercambio, usando para ello la comisión cobrada al comercio, y la entidad emisora recibe dicha tasa de intercambio remunerando así su esfuerzo de implantación y garantía de pago del importe de la venta. Por último, el cliente final puede realizar su compra apoyándose en este sistema para realizar el pago de forma cómoda, rápida y segura.

En algunos casos la entidad emisora añade a estas transacciones un crédito que concede al cliente final para financiar las operaciones, bajo riesgo de la propia entidad emisora y al margen de la operativa de cobro-pago, surgiendo así las tarjetas de crédito. Frente a los casos en que no existe crédito y la compensación del pago se realiza en un periodo tan corto como si fuera en tiempo real, se incorporan lo que conocemos como tarjetas de débito.

Sumando a este modelo un concepto de tasas que las franquicias establecen al acordar con las entidades financieras el uso de sus marcas y las tarifas que los centros de proceso establecen por realizar los distintos servicios operativos necesarios, se tiene ya todo el conjunto de relaciones económicas que hace sostenible el caso de negocio de la industria. [Cas-2008]

I.1.4. LOS INSTRUMENTOS DEL MODELO: TARJETA DE CRÉDITO Y DE DÉBITO.

Las tarjetas de crédito se han convertido en una parte esencial de la vida cotidiana de la mayoría de personas y ha cambiado por completo la gestión de las finanzas personales.

El primer paso para entender cómo funciona una tarjeta de crédito es saber en qué consiste exactamente, y que hay diferentes tipos de 'plásticos' en circulación. Los más comunes son las tarjetas de débito y crédito. La diferencia principal es que en la primera, los pagos se cargan directamente y sobre todo de forma más o menos inmediata a la cuenta bancaria del titular, mientras que en la segunda el pago se suele aplazar en el tiempo (un mes en la mayoría de los casos). De esta forma, las tarjetas de crédito, como su propio nombre indica, son una fórmula de obtener un dinero que todavía no está en la cuenta corriente del usuario pero que se supone que estará en el momento del cobro. Es como si el banco o la entidad emisora adelantasen al titular un dinero que se supone que obtendrá en el corto plazo.

En resumen, mientras que con las tarjetas de débito sólo se puede disponer del dinero que hay en la cuenta (a veces permiten un pequeño sobregiro), con las tarjetas de crédito se puede conseguir más capital del que realmente se dispone. De ahí el componente de crédito. La cantidad dependerá de la situación financiera de cada persona y sobre todo de su solvencia. [Cas-2008]

I.1.5. PROCESO OPERATIVO DEL MODELO.

Para llegar a feliz término la operación, se debe realizar el siguiente proceso operativo:

- **Autentificar al cliente final** que desea realizar el pago a efectos de identificarlo ante la entidad emisora del medio de pago/tarjeta con la que desea pagar.
- **Autorizar por dicho emisor la operación de pago**, lo que implica compromiso por el emisor respecto la garantía de pago al comercio, factor clave en la operatividad de las tarjetas por cuanto esta garantía

de pago conlleva la confianza de base de la aceptación de la tarjeta como una divisa más para el pago por el comerciante vendedor.

- **Registrar la operación para poder realizar la gestión de compensación entre las entidades que intervienen:** entidad emisora de la tarjeta, entidad adquiriente que da servicio al comercio.
- **Compensar las operaciones,** es decir, para cada uno de los que intervienen en el proceso, se debe calcular el importe a pagar (entidad emisora) y a cobrar (entidad adquiriente) con sus correspondientes comisiones y tasas en particular la tasa de intercambio entre adquiriente y emisor.
- **Liquidar contablemente,** es decir, proceder al cargo y abono de los importes correspondientes, por el procesador, en las cuentas de la entidad emisora y la entidad adquiriente.

Este proceso se completa cuando la entidad emisora descuenta de la cuenta personal del cliente los importes que la entidad adquiriente abona al comercio en su cuenta personal, cerrando así el flujo de cobros-pagos entre ellas.

Las etapas descritas en la cadena de valor anterior son realizadas por empresas conocidas como procesadores de medios de pago, que son los facilitadores de que se realice el proceso necesario sobre todo cuando el emisor y el adquiriente de una operación son entidades diferentes.

Usualmente los procesadores añaden a los servicios anteriores, que configuran el núcleo de la actividad de proceso, otros servicios relacionados con la administración y gestión de las operaciones, servicios de valor añadido y que se centran principalmente en las actividades de:

- Gestión de las redes de terminales en puntos de ventas necesarios para realizar la captura de las operaciones.
- Gestión del fraude tanto del emisor como del adquiriente.
- Gestión de disputas del proceso de garantía de pago.
- Gestión de la información capturada en las operaciones, información base sobre los titulares de tarjetas y los comercios y su uso de los medios de pago.

Las procesadoras desarrollan unos servicios básicos de la actividad soporte del día a día. En realidad, se puede afirmar que un medio de pago es la suma de tecnología y mercadeo de marca/producto. En esa tecnología, el procesador es el icono de base, facilita el desarrollo del negocio y la implementación de los servicios al titular de las tarjetas y al comerciante que las acepta, pero además se configura como clave en el control de la actividad desarrollada con los medios de pago. Por tanto, los procesadores se configuran como estratégicos para las entidades

financieras en la medida que afectan tanto en el desarrollo de la relación comercial con los clientes titulares y comercios (implementan la oferta de servicios comprometida en la relación entre la entidad y estos clientes) como gestionan una información y procesos básicos en el dominio de la relación con dichos clientes. [Cas-2008]

I.1.6. OBJETIVO DE LA PROCESADORA DE MEDIOS DE PAGO.

La compañía tiene por objeto suministrar servicios de procesamiento automático y transaccional para medios de pago; emisión y procesamiento de pagos y demás transacciones que se efectúen a través de puntos de ventas o dispositivos remotos colocados en establecimientos comerciales y que operen mediante el uso de tarjetas de débito, crédito, consumo y cualquier otro medio de pago de similar naturaleza creado, emitido u operado por Bancos; y en general prestar servicios de asesoría y consultoría en sistemas de información.

Para poder cumplir con todos los servicios anteriormente expuestos se necesita contar con una infraestructura tecnológica, que va desde los puntos de venta hasta las entidades bancarias pasando por los enrutadores, servidores y demás dispositivos que componen la red de la procesadora de medios de pago (Ver figura 2), lo suficientemente estable de forma que se logre mantener la operatividad de la misma los 365 días del año sin que esta labor se vea interrumpida.

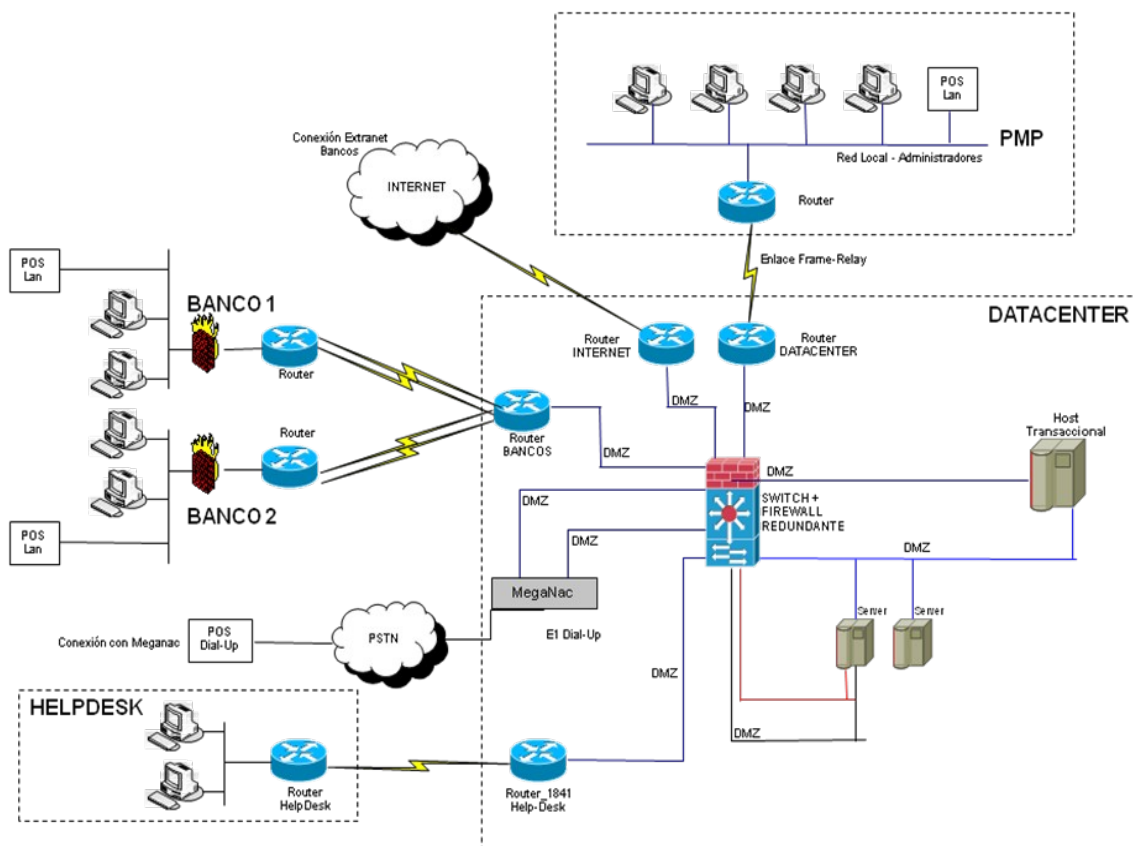


Figura Nº 2. Infraestructura Tecnológica de una PMP.

Adicionalmente, debemos tener presente que no todos los servicios prestados por la procesadora dependen en su totalidad de un computador o un sistema

automático, y que no solo se trata de aprobar o rechazar transacciones a través de un punto de venta, sino que también se manejan procesos para los cuales se necesita de recursos humanos que lleven a cabo ciertas tareas, como por ejemplo: gestionar instalaciones de puntos de ventas en los comercios, gestionar la compensación de los pagos a los comercios, etc.; para la realización de dichas tareas se cuenta con diversas aplicaciones informáticas que ayudan a brindar los servicios ofrecidos. Estas aplicaciones son tan importantes como la infraestructura en sí para llevar a cabo el objetivo de la procesadora según los niveles de servicios acordados con las entidades bancarias.

Todo lo antes expuesto necesita de un control exhaustivo y un continuo monitoreo, esto debido a que tanto las distintas aplicaciones como la infraestructura tecnológica en sí pueden sufrir de fallas, retardos, desperfectos que ocasionen un mal desempeño y por tanto afecten el negocio de la procesadora de medios de pago, todos estos imprevistos que logren dicho efecto son llamados incidencias.

Para atender todas estas incidencias se necesita de una unidad dentro de la procesadora de medios de pago que se encargue prever que se presenten dichas incidencias mediante el continuo monitoreo. Dicha unidad lleva por nombre Centro de Cómputo; sin embargo el Centro de Cómputo no siempre podrá, mediante su control darse cuenta de dichos eventos, es por ello que las demás unidades de las procesadora, al verse directamente afectadas por alguna incidencia se convierten en clientes del Centro de Cómputo y notifican al mismo para que se dé pie al proceso de Gestión de Incidencias.

I.1.7. ESQUEMA ORGANIZACIONAL.

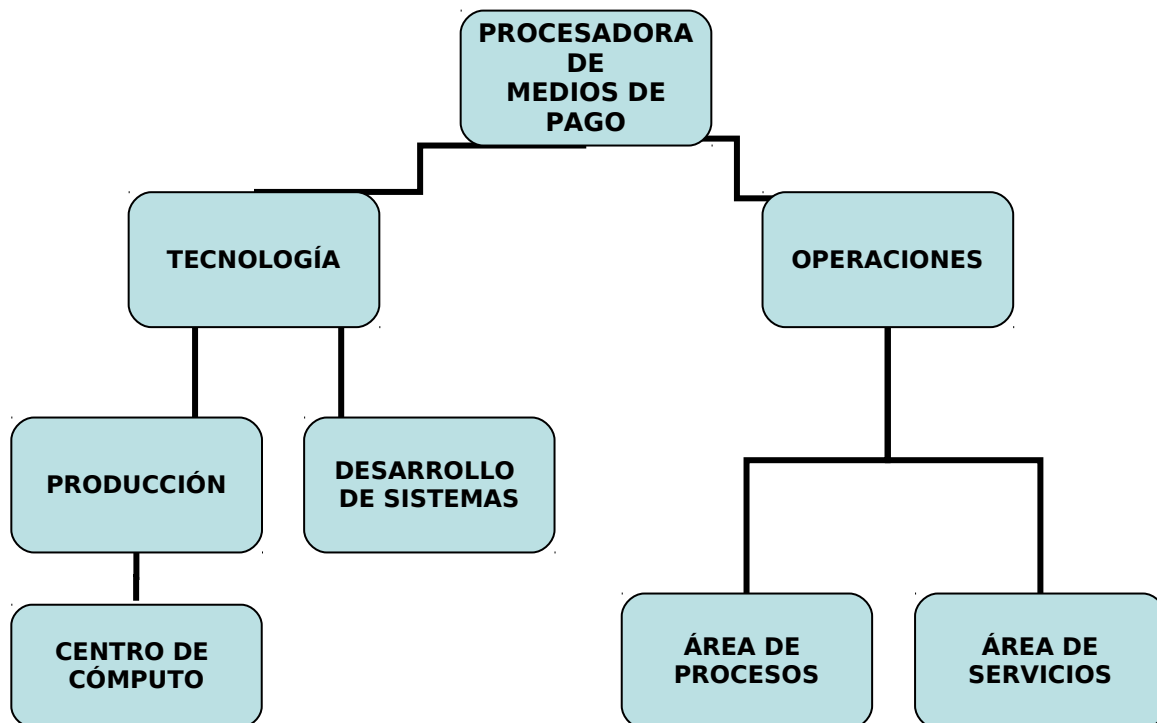


Figura Nº 3. Estructura Organizacional de una Procesadora de Medios de Pago.

La estructura organizacional es el conjunto de unidades (Gerencia General, Gerencias, Jefaturas, y Unidades) debidamente interrelacionadas entre sí, que representan el flujo de autoridad y responsabilidad existente dentro de la organización; elementos definidos en procura de lograr los objetivos señalados para cumplir con la misión y visión de este tipo de empresas. Es importante aclarar que desde cualquier unidad reflejada en este esquema se pueden detectar incidencias que afecten la misión y visión de la organización y que cada una de ellas está en la obligación de informar al Centro de Cómputo la presencia de cualquier eventualidad para que así estos puedan llevar el control y ayudar a solventar cada una de esas incidencias reportadas.

I.1.8. EL CENTRO DE CÓMPUTO.

El Centro de Cómputo, (en lo sucesivo denominado CECOMP) tiene como finalidad garantizar la operatividad de la infraestructura IT (Tecnología de la Información) en conjunto, velando por su correcto funcionamiento y operatividad las 24 horas del día y los 365 días del año, todo esto con un nivel de respuesta óptimo y de acuerdo a las exigencias del cliente.

El operador del CECOMP, administra, monitorea y gestiona todos los sistemas instalados con el fin de garantizar los más altos niveles de disponibilidad y confiabilidad de las redes, sistemas, equipos y datos, bajo estrictas medidas de calidad y servicio, brindando además, un soporte especializado y personalizado de mesa de ayuda para los clientes externos, así como, soporte y apoyo permanente a las áreas internas para la elaboración de proyectos que tienen que ver con el CECOMP.

En cumplimiento con todo lo expuesto, el CECOMP tiene designadas tres tareas fundamentales: ejecución de procesos, monitoreo de la infraestructura IT y Servicio de Ayuda de la infraestructura IT.

Ejecución de procesos internos: Engloba un conjunto de procesos, de cuya ejecución depende el cumplimiento del servicio ofrecido a los clientes, como por ejemplo, cierres bancarios o carga de archivos recibidos de las marcas Visa o Mastercard.

Monitoreo de la Infraestructura IT: Este proceso comprende la constante inspección de los sistemas de alerta que detectan anomalías en los sistemas de hardware y software indispensables para la ejecución de los procesos internos y otros servicios necesarios para el cumplimiento de los compromisos adquiridos.

Servicio de Ayuda: Es la función bajo la cual CECOMP registra, escala y hace seguimiento a cualquier suceso o problema, originado por hardware o software, que pueda afectar o esté afectando el servicio. Estos problemas son denominados incidencias y deben ser detectados por el operador durante el monitoreo y ejecución de los procesos internos o en última instancia por algún usuario interno o externo que informe de la ocurrencia de una anomalía.

Capítulo II: Planteamiento de la Solución.

En éste capítulo se propone una solución adecuada al problema relacionado con la Gestión de Incidencias, que se adapta a las necesidades de la organización.

Luego se detallan el objetivo general y los objetivos específicos del Trabajo Especial de Grado, así como su alcance y la importancia y justificación del mismo, enfocando las mejoras que aportan al sistema actual, los beneficios que genera a la comunidad involucrada en el problema y las herramientas tecnológicas utilizadas para el desarrollo e implementación de la solución planteada.

II.1. Solución Propuesta

Se propone desarrollar un sistema automatizado que permita la gestión, administración y control de las incidencias reportadas al *Servicio de Ayuda* del CECOMP de una manera efectiva, teniendo como principales características:

1. *Uso de una Base de Datos de la Infraestructura IT y los contactos que reportan y solucionan:* Esto permitirá registrar incidencias con sólo seleccionar unos pocos datos de una lista, dependiendo de la selección se llenaran todos los campos necesarios que requiere el registro de la incidencia, con esto se incrementa la eficiencia del operador en el registro, clasificación y escalado de la incidencia y

adicionalmente limita el reporte de fallas sólo a las personas con autorización.

2. *Uso de una Base de Datos de Conocimientos:* Esto permitirá registrar de forma automática la solución de las incidencias para que posteriormente puedan ser consultadas de una manera eficiente y amigable, con lo cual, se tendría siempre disponible la solución encontrada a cualquier falla que se repita, disminuyendo el tiempo para la solución de las reincidencias y el costo asociado a la restitución del servicio.
3. *Creación de un Monitor de Incidencias:* Esto permitirá visualizar el estado actual de las incidencias reportadas, para controlar que todas cumplan el ciclo completo hasta ser cerradas.
4. *Generación de Reportes e Indicadores:* Permitirá la recopilación de toda la información disponible para hacerle un seguimiento a las cantidades y tipos de incidencias reportadas y establecer planes de acción que conlleven a soluciones definitivas que eviten y/o disminuyan a la mínima expresión la ocurrencia de las mismas.

II.2. Objetivos

En esta sección se muestran tanto el objetivo general como los específicos que se desean cubrir con la solución propuesta.

II.2.1. OBJETIVO GENERAL

Desarrollar una Aplicación Web que lleve a cabo el proceso de Gestión de Incidencias minimizando los errores y aumentando la satisfacción de los usuarios.

II.2.2. OBJETIVOS ESPECÍFICOS

Los objetivos específicos para el desarrollo e implementación del Sistema Automatizado de Gestión de Incidencias son los siguientes:

- Realizar el levantamiento de información de la situación actual.
- Determinar y documentar los procesos.
- Determinar las necesidades de información con los diferentes usuarios: Operadores, personal de soporte de segundo y tercer nivel y personal administrativo.

- Establecer el modelo propuesto. Diseño del sistema (aprobación por parte del cliente).
- Desarrollar el sistema de acuerdo al modelo certificado por el cliente.
- Pruebas de control de calidad técnica.
- Pruebas de control de calidad funcional.
- Implantar el Sistema.
- Puesta en marcha.

II.3. Importancia y Justificación

Día a día en el Centro de Cómputo se presentan incidencias con los sistemas, incidencias que deben pasar por un proceso de registro para llegar a su solución, ese proceso de registro puede volverse engorroso porque la plantilla de incidencias no está centralizada ni automatizada, por lo cual se complica el proceso de búsqueda y de actualización. Adicionalmente el personal trabaja en horarios rotativos, lo cual es un factor clave en la pérdida de continuidad de la búsqueda de la solución de cada una de las incidencias.

La importancia de automatizar estas actividades para facilitarlas, permite el ahorro de tiempo y esfuerzo humano. Contribuye también, a incrementar la satisfacción del cliente brindando soluciones a las incidencias de forma rápida y eficaz. Permitiendo llevar la continuidad de la búsqueda de la solución y disminuyendo considerablemente el tiempo de respuesta en el caso de que se presenten solicitudes que ya han sido solventadas anteriormente.

El desarrollo de esta nueva aplicación permitirá disponer de la información en digital agilizando la búsqueda para las incidencias ya resueltas. Notificar de forma inmediata mediante un correo electrónico al ente encargado de dar solución cada nueva incidencia reportada, basándose en criterios propios de las reglas del negocio.

II.4. Estudio de Factibilidad.

Una vez identificada la problemática existente y establecidas las causas que justifican el uso de una aplicación Web, es pertinente realizar un estudio de factibilidad que permita determinar la viabilidad de implantar una nueva aplicación. Los aspectos tomados en cuenta para este estudio fueron clasificados en: factibilidad técnica, beneficios tangibles e intangibles y factibilidad operativa, los cuales se describen a continuación.

II.4.1. FACTIBILIDAD TÉCNICA Y ECONÓMICA.

En la factibilidad técnica se realizó una evaluación de la plataforma tecnológica existente en la procesadora de medios de pago, esto con la finalidad de

verificar que la misma sea compatible con el desarrollo e implantación de la aplicación Web propuesta.

Para ello se recaudó información bajo los enfoques de hardware y software, generando los siguientes cuadros:

Hardware Disponible en la Procesadora de Medios de Pago.

Equipo IBM Modelo 8215-EB4 Pentium D 2,8 GHz, 1 Disco Duro de 70 Gb, 1Gb de RAM.
Red Ethernet.
Topología de Red: Estrellar.
Cableado estructurado categoría 5E.
Router: Cisco Modelo: 2811 Tipo de Interfaz: FastEthernet.
Switch: Cisco Modelo: 3750 Tipo de Interfaz: FastEthernet.

Tabla N° 1. Hardware Disponible en la Procesadora de Medios de Pago.

Software Disponible en la Procesadora de Medios de Pago.

Sistema Operativo Windows Server 2003 servipack 2.0
Navegador Internet Explorer 6.0 y Mozilla Firefox
Herramientas de escritorio Office 2003
Herramientas de Diseño y/o Desarrollo: Suite Macromedia 8 (Dreamweaver 8, Flash 8).

Tabla N° 2. Software Disponible en la Procesadora de Medios de Pago.

Con el hardware existente en la procesadora de medios de pago se determinó que no es necesario la adquisición de nuevos equipos, ni actualizar los existentes ya que estos satisfacen los requerimientos para el funcionamiento de la aplicación Web para la gestión de incidencias.

En cuanto al software, las herramientas existentes en la organización son insuficientes para el desarrollo e implementación de la aplicación Web, siendo necesario adquirir un sistema manejador de base de datos y un servidor Web.

Finalmente, como resultado del estudio de factibilidad técnica y económica se determinó que en los actuales momentos, la procesadora de medios de pago posee la infraestructura tecnológica necesaria a nivel de hardware, pero a nivel de software carece de herramientas necesarias para el desarrollo y la implementación del sistema. Afortunadamente estas herramientas faltantes son software libre (como lo son el servidor Apache con el módulo PHP y el manejador de base de datos MySQL), lo que no amerita hacer ningún desembolso para adquirirlas, por tanto la puesta en marcha del sistema planteado en este trabajo no se verá afectada.

II.4.2. BENEFICIOS TANGIBLES E INTANGIBLES.

- Reducción de costos operativos (papelería, archivos, etc.)

- Disponer de información en tiempo real de las incidencias reportadas.
- Facilitar el control y seguimiento de los incidentes presentados en las distintas áreas de la procesadora de medios de pago.
- Contar con la información de forma eficaz y precisa.
- Mantener al tanto a todos los involucrados en la resolución de una incidencia.
- Generar estadísticas e indicadores en tiempo real.

II.4.3. FACTIBILIDAD OPERACIONAL.

El sistema va a garantizar rapidez a la hora de reportar un nuevo incidente por parte de cualquier departamento perteneciente a la procesadora de medios de pago, esto traerá al personal del Centro de Cómputo mayor rapidez y veracidad en la información que se le solicite para escalar los incidentes a los entes que se encargan de resolverlos.

II.5. Herramientas Tecnológicas.

En esta sección se describe la tecnología utilizada, mencionando arquitecturas y tecnologías que fueron consideradas para el desarrollo del Trabajo Especial de Grado.

II.5.1. DEFINICIÓN DE APLICACIONES WEB.

Una aplicación Web es un programa que corre en un servidor Web y que utiliza las páginas Web como la interfaz de usuario[*Informaticamilenium-2009*]. Estas páginas pueden ser accedidas por un cliente Web, usualmente un navegador Web que se encuentra corriendo en un PC, un teléfono móvil, un PDA, entre otros. Los usuarios utilizan estas aplicaciones accediendo al servidor Web que las contiene, a través de Internet o de una Intranet.

Este tipo de aplicaciones están constituidas por un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada página principal, con un nombre de dominio y dirección en Internet específicos. Dicho conjunto de páginas conforman un sitio Web, y éstos son utilizados por las instituciones públicas y privadas, organizaciones e individuos, para comunicarse con el mundo entero. [*Masternewmedia-2009*]

Arquitectura Cliente - Servidor.

Las aplicaciones Web son aplicaciones basadas en la arquitectura “Cliente-Servidor”. Esta arquitectura consiste en un servidor que proporciona un servicio y un cliente que desea acceder al servicio.

La arquitectura Cliente-Servidor es un modelo para el desarrollo de aplicaciones, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos. [Mitecnologico-2008]

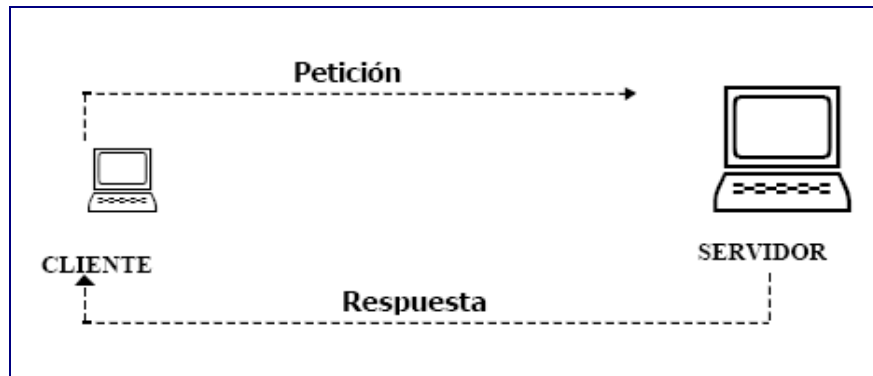


Figura Nº 4. Modelo Cliente/Servidor

Como se puede apreciar en la Figura Nº 4, en esta arquitectura la computadora de cada uno de los usuarios, llamada **Cliente**, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como **Servidor**. Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes, y de procesarla según le convenga. Los distintos servidores también pueden intercambiar información dentro de ésta arquitectura.

Los clientes o servidores pueden estar conectados en una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es Internet. Cliente-Servidor es el modelo de interacción más común entre aplicaciones en una red. No forma parte de conceptos de Internet como los protocolos IP, TCP o UDP, sin embargo, todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo.

Se puede decir que la arquitectura Cliente-Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información; estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente-Servidor, las empresas de cómputo enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen. [mipagina.cantv.net-2009]

II.5.2. HERRAMIENTAS TECNOLÓGICAS PARA EL DESARROLLO DE APLICACIONES WEB.

El desarrollo de una aplicación Web requiere el uso de un conjunto de tecnologías que permiten diseñar y estructurar el contenido de las páginas, implementar las funcionalidades y el dinamismo de las mismas, alojarlas y ponerlas en funcionamiento para su utilización por parte de los usuarios.

Entre éstas tecnologías tenemos:

- **Tecnologías del Lado del Cliente:** están insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el navegador. Estas tecnologías son utilizadas fundamentalmente para mostrar la información y dar estética al sitio Web. En particular usaremos **HTML, XHTML, CSS y JavaScript**.
- **Tecnologías del Lado del Servidor:** permiten construir código que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Existen diversas tecnologías del lado del servidor, ampliamente utilizadas en el desarrollo de aplicaciones Web. Será descrito el Servidor Apache por ser uno de los servidores más utilizados para el despliegue de aplicaciones Web desarrolladas en PHP.
- **Sistemas Manejadores de Bases de Datos:** consisten en un conjunto de datos relacionados entre sí y un conjunto de herramientas de software (y/o hardware) para tener acceso a dichos datos, y a su vez procesarlos y administrarlos. Se estudiará MySQL como sistema manejador de base de datos de software libre fácilmente integrable con aplicaciones escritas en PHP.

HTML (Lenguaje de Marcado de Hipertexto).

HTML es el acrónimo en inglés de *Hyper Text Markup Language* (lenguaje de marcado de hipertexto), actualmente en su versión 4.01, es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. HTML es el estándar usado en la World Wide Web, y se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

Las etiquetas o marcas de HTML, consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en el navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del computador. Por ejemplo: <P> es una etiqueta que indica inicio de un párrafo y su correspondiente marca de cierre es </P>.

La estructura básica de un documento HTML esta constituida por las etiquetas <html>, <head>, <title> y <body>, las cuales conforman el cuerpo principal del documento.

El lenguaje HTML puede ser editado con cualquier editor de textos básico, como puede ser el Bloc de Notas de Windows (o Notepad), o cualquier otro editor que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, etc. [adelat-2006]

CSS 2.0 (Hojas de Estilo en Cascada).

Hojas de Estilo en Cascada (*Cascading Style Sheets*), es una tecnología simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza generalmente para dar estilo a documentos HTML, separando el contenido de la presentación. Los estilos definen la forma de mostrar los elementos, permitiendo a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en el CSS afectará a todas las páginas vinculadas a dicho CSS en las que aparezca ese elemento.

Las hojas de estilos también son utilizadas para ubicar los elementos HTML en la página, permitiendo ordenar los elementos que tengan cierto tamaño y posición, dando una mejor organización del código HTML o XHTML.

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne. En la Figura Nº 5 se puede observar un ejemplo de una regla escrita en el lenguaje de estilos CSS.

<pre>h1 {color: red;}</pre>	- h1 es el selector. - {color: red;} es la declaración.
-----------------------------	---

Figura Nº 5. Reglas para definir estilos CSS

El selector funciona como enlace entre el documento y el estilo, especificando los elementos que se van a ver afectados por esa declaración. La declaración es la parte de la regla que establece cuál será el efecto. En el ejemplo anterior, el selector h1 indica que todos los elementos h1 se verán afectados por la declaración donde se establece que la propiedad color va a tener el valor red (rojo) para todos los elementos h1 del documento o documentos que estén vinculados a esa hoja de estilos.

Las tres formas más conocidas de dar estilo a un documento son las siguientes [w3c-2005]:

- Utilizando una hoja de estilo externa que estará vinculada a un documento a través del elemento <link>, el cual debe ir situado en la sección <head> del documento HTML.
- Utilizando el elemento <style>, en el interior del documento al que se le quiere dar estilo, y que generalmente se situaría en la sección <head>. De esta forma los estilos serán reconocidos antes de que la página se cargue por completo.
- Utilizando estilos directamente sobre aquellos elementos que lo permiten a través del atributo <style> dentro de <body>. Pero este tipo de estilo pierde las ventajas que ofrecen las hojas de estilo al mezclarse el contenido con la presentación.

JavaScript 1.3.

Es un Lenguaje de programación, introducido por Netscape. Usado para la creación de acciones y procesos de automatización para páginas Web. Puede ser insertado entre el código HTML e interpretado y ejecutado por el navegador, actualmente todos los navegadores lo soportan.

JavaScript es un lenguaje interpretado (no requiere compilación) orientado a las páginas Web, con una sintaxis semejante a la del lenguaje Java y el Lenguaje C.

Las dos principales características de JavaScript son, por un lado que es un lenguaje basado en objetos (es decir, el paradigma de programación es básicamente el de la programación dirigida a objetos, pero con menos restricciones), y por otro JavaScript es además un lenguaje orientado a eventos, debido por supuesto al tipo de entornos en los que se utiliza (Windows y sistemas X-Window). Esto implica que gran parte de la programación en JavaScript se centra en describir objetos (con sus variables de instancia y métodos de "clase") y escribir funciones que respondan a movimientos del ratón, pulsación de teclas, apertura y cerrado de ventanas o carga de una página, entre otros eventos.

JavaScript es un lenguaje que permite aumentar la interactividad y la personalización de los sitios Web. [Riv-2009]

PHP (Hypertext Pre-processor).

PHP es un acrónimo que significa Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdof en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en

PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, aunque el número de sitios en PHP ha declinado desde agosto de 2005. La más reciente versión principal del PHP fue la versión 5.2.6 de 1 de mayo de 2008. [ProgramasYorbin-2009]

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de páginas web, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de comandos, de la misma manera como Perl o Python pueden hacerlo, a esta versión de PHP se la llama PHP CLI (*Command Line Interface*).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el *script* solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

PHP permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgre, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

Ventajas de PHP.

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre las cual es se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite el desarrollo de un código más corto, simple, flexible y mantenible.

- Es un software no propietario, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aún estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. [tutorialgratistutorial-2008]

Apache.

Apache es un servidor de páginas Web de código abierto, sólido y para uso comercial desarrollado por la *Apache Software Foundation*. Actualmente en su versión 2.2.3.

Apache está diseñado para ser un servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y los diferentes entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades, o que una misma característica o funcionalidad sea implementada de diferente manera para obtener una mayor eficiencia. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular (Ver Figura Nº 6). Este diseño permite a los administradores de sitios Web elegir que funcionalidades van a ser incluidas en el servidor seleccionando que módulos se van a usar, ya sea al compilar o al ejecutar el servidor. [Apache-2009]

La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos de Apache se pueden clasificar en tres categorías:

- **Módulos Base:** Módulo con las funciones básicas de Apache.
- **Módulos Multiproceso:** son los responsables de conectar con los puertos de red de la máquina, aceptar las peticiones, y generar los procesos hijo que se encargan de servir dichas peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

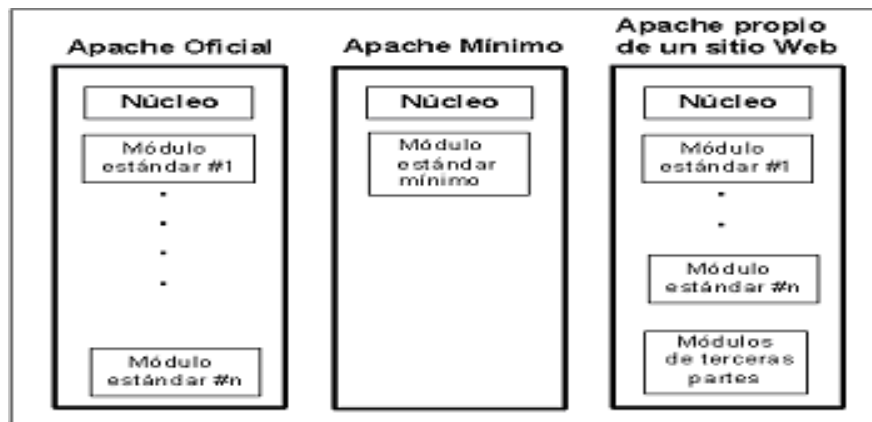


Figura Nº 6. Arquitectura Modular de Apache

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta Apache, optimizando el rendimiento y rapidez del código.

La arquitectura modular de Apache le permite a éste servidor interactuar y funcionar en conjunto con otros servidores.

Sistema Manejador de Bases de Datos MySQL 5.0.

MySQL es un Sistema Manejador de Bases de Datos (SMBD) desarrollado por la empresa MySQL AB siendo hoy en día uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación se debe, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características.

Las características más importantes de MySQL se describen a continuación:

- Multihilo: núcleo de aplicación completamente multihilo. Puede usar múltiples CPUs, si están disponibles.
- Disponibilidad en diversas plataformas y sistemas operativos (Linux, Windows, Solaris, MacOS X, OS/2, AIX, entre otros).
- Provee motores de almacenamiento que tratan con distintos tipos de tablas, tanto transaccionales como no transaccionales. Entre ellos tenemos MyISAM, Memory, Merge, InnoDB, BDB.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Ruby and TCL.
- El código de MySQL esta comprobado por Purify (herramienta para verificación de problemas de memoria).
- Soporta un ambiente Cliente/Servidor.

[MYSQL-2009]

Capítulo III: Marco Metodológico.

A lo largo de este capítulo se presenta la metodología que se utilizó para el desarrollo del Sistema de Gestión de Incidencias, justificando su uso y describiendo cada uno de las fases que la componen.

En este caso particular se utilizó el marco de desarrollo conocido como “Proceso Unificado”, en este capítulo se abarcará una breve descripción del mismo,

para luego ir mostrando su adaptación al caso particular del desarrollo del Sistema de Gestión de Incidencias.

III.1. Proceso Unificado.

El Proceso Unificado de Desarrollo de Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el *RUP* (Rational Unified Process).

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el *RUP*, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del *RUP*. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El nombre Proceso Unificado se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes.

III.2. Características.

Iterativo e Incremental

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones. Estas iteraciones ofrecen como resultado un *incremento* del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

Dirigido por los casos de uso

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Centrado en la arquitectura

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, plomería, etc.

Enfocado en los riesgos

El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

III.3. Principios de Desarrollo.

El proceso unificado se basa en 6 principios claves, con los que define una forma efectiva de trabajar para los equipos de desarrollo de software.

Gestión de Requisitos.

RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de Caso de Uso y escenarios para representar los requisitos.

Desarrollo de Software Iterativo.

Desarrollo del producto mediante iteraciones con ciclos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.

Desarrollo Basado en Componentes.

La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.

Modelo Visual (usando UML).

UML es un lenguaje para visualizar, especificar, construir y documentar los componentes de un sistema. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelado visual también ayuda a mantener la consistencia entre los componentes del sistema: requisitos, diseños e implementaciones. En resumen, el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.

Verificación Continua de la Calidad.

Es importante que la calidad de todos los componentes se evalúe en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de todo el proceso.

Gestión de Cambio.

El cambio es un factor de riesgo crítico en los proyectos de software. Los componentes de un software pudieran cambiar debido a acciones de mantenimiento posteriores a la entrega del producto y/o durante el proceso de desarrollo, por posibles cambios en los requisitos. La Gestión de Cambios y de Configuración es la disciplina de RUP encargada de este aspecto.

III.4. Ciclo de Vida.

El ciclo de vida RUP divide el proceso en cuatro fases: fase de inicio, fase de elaboración, fase de construcción y fase de transición; dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura N° 7 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura.

Durante la **fase de inicio** las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

En la **fase de elaboración**, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la **fase de construcción**, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refina su análisis y diseño luego se procede a su implementación y pruebas. De esta forma se ejecuta una pequeña cascada para cada ciclo. Realizando tantas iteraciones como sea necesario, hasta que se termine la implementación de la nueva versión del producto.

En la **fase de transición** se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

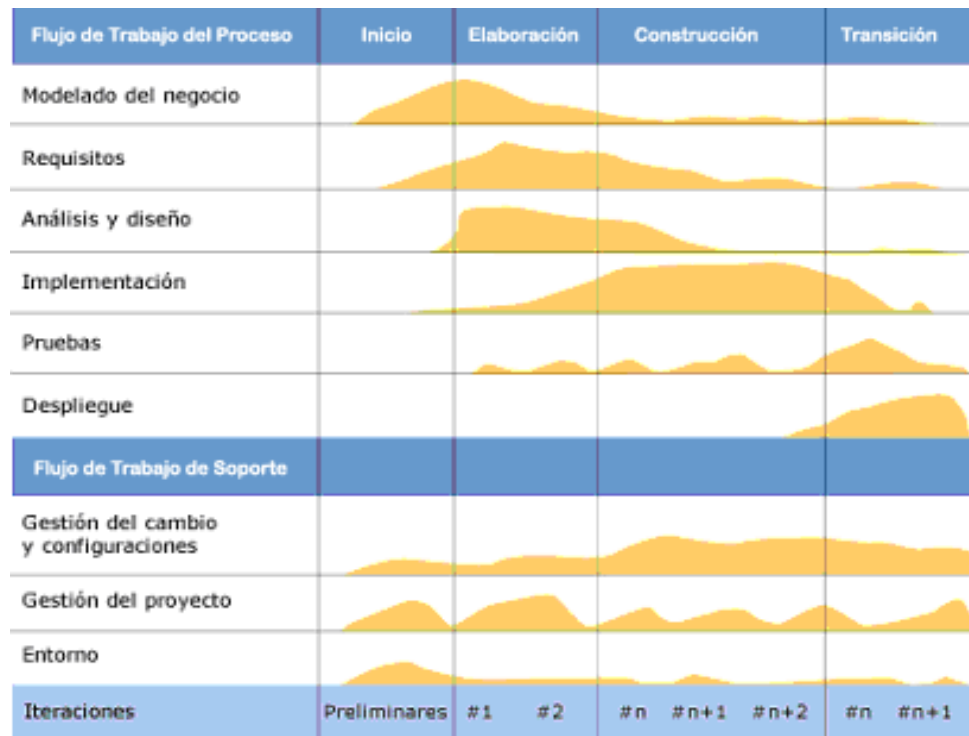


Figura N° 7. Estructura RUP

III.4.1. FASE DE INICIO.

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

Los objetivos de esta fase son:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.
- Estimar los riesgos, las fuentes de incertidumbre.

III.4.2. FASE DE ELABORACIÓN.

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Los objetivos de esta fase son:

- Definir y validar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costos si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un costo razonable y en un tiempo razonable.

III.4.3. FASE DE CONSTRUCCIÓN.

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos incluyen:

- Minimizar los costos de desarrollo mediante la optimización de recursos y evitando tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea posible.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea posible.

III.4.4. FASE DE TRANSICIÓN.

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere completar la documentación, entrenar al

usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

Los principales objetivos de esta fase son:

- Conseguir que el usuario se valga por si mismo al momento de utilizar el producto final, y en caso de necesitar algún tipo de ayuda la misma se encuentre a su disposición a través de la documentación del producto.
- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

[Let-2009]

Capítulo IV: Especificación de Requerimientos.

El enfoque proporcionado por el Proceso Unificado plantea que el sistema debe ser construido desde el punto de vista del usuario, como un mapa del problema descrito en términos de actores, escenarios y casos de uso, es por ello que a lo largo del capítulo se hablará de los métodos utilizados para la obtención de los requerimientos, para luego poder describir las funcionalidades que el sistema deberá brindar.

IV.1. Técnica de Extracción y Especificación de requerimientos.

La técnica seleccionada para la extracción de los requerimientos fue el uso de entrevistas, las cuales se emplearon para recopilar información proveniente de personas o grupos involucrados en el proceso.

Se realizó una primera entrevista con los usuarios, efectuando una conversación abierta y general, tratando de entender el dominio del problema.

Sucesivamente, por medio de otras entrevistas más detalladas se logró definir paso a paso el proceso que se sigue actualmente para el registro de incidencias. La ejecución de esta técnica permitió obtener la información necesaria para comenzar a construir las especificaciones de los requerimientos a través de UML, específicamente los diagramas de Casos de Uso. A continuación se describen las actividades para llegar al modelo de especificación de requerimientos.

IV.2. Identificación de los Actores.

Los tipos de usuarios que el sistema soporta son los descritos a continuación:

- Usuario Cliente: Es quien podrá reportar nuevas incidencias dentro del sistema, y consultar las que se han reportado anteriormente, son usuarios que pertenecen a otras unidades de la procesadora, y que dependiendo de su función dentro de la misma pueden identificar incidencias y reportarlas al Centro de Cómputo.
- Usuario Operador: Es quien podrá reportar nuevas incidencias y modificar las ya existentes, actualizándolas con datos importantes, tales como, a quién fue escalada o el estado actual de dichas incidencias, son los usuarios que pertenecen al Centro de Cómputo de la procesadoras, los cuales se encargan de gestionar las incidencias.
- Usuario Analista: Es quien recibirá del sistema los reportes estadísticos generados por el sistema.
- Usuario Administrador: Es el encargado de administrar parámetros del sistema, tales como creación de nuevos usuarios.

IV.3. Identificación de los Escenarios.

A continuación se describen los distintos escenarios que se contemplan en el sistema.

Nombre del escenario:	Administración de Perfiles de Usuario
Instancias de actores Participantes:	Administrador: Usuario Administrador Cliente: Usuario Cliente Operador: Usuario Operador Analista: Usuario Analista
Flujo de Eventos:	1. Administrador procede a crear el perfil de: Cliente, Operador y Analista. 2. Cada uno de los perfiles creados puede ingresar al sistema. 3. Cliente y Operador pueden ingresar al sistema para reportar y consultar incidencias. 4. Analista puede ingresar al sistema e ingresar los datos necesarios para generar reportes estadísticos.

Tabla N° 3. Escenario Usuario Administrativo

Nombre del escenario:	Administración de Incidencias
Instancias de actores Participantes:	<i>Cliente: Usuario Cliente</i> <i>Operador: Usuario Operador</i> <i>Administrador: Usuario Administrador</i>
Flujo de Eventos:	<i>1. Se presenta una incidencia en los procesos diarios.</i> <i>2. Cliente u Operador proceden a crear el reporte de la incidencia que se presentó.</i> <i>3. Operador escala la incidencia al área resolutoria.</i> <i>4. Operador recibe respuesta del área resolutoria.</i> <i>5. Operador actualiza el reporte de incidencia con la información recibida por parte del área resolutoria.</i> <i>6. Cliente u Operador proceden a consultar el avance de la resolución de la incidencia.</i> <i>7. Cliente indica estado de satisfacción con respecto a la solución encontrada.</i> <i>8. Operador procede a cerrar el reporte de incidencia.</i> <i>9. Administrador procede a eliminar reportes de incidencias creados con errores.</i>

Tabla N°4. Escenario Usuario Cliente

Nombre del escenario:	Consulta de Incidencias
-----------------------	-------------------------

Instancias de actores Participantes:	<i>Cliente: Usuario Cliente Operador: Usuario Operador</i>
Flujo de Eventos:	1. <i>Cliente u Operador acceden al sistema.</i> 2. <i>Selecciona la opción de consulta de reportes de incidencias.</i> 3. <i>Realiza la búsqueda según el criterio seleccionado.</i> 4. <i>Observa y analizan los datos con opción a imprimirlos.</i>

Tabla N°5. Escenarios Usuario Cliente

Nombre del escenario:	<i>Generar Reportes Estadísticos</i>
Instancias de actores Participantes:	<i>Analista: Usuario Analista</i>
Flujo de Eventos:	1. <i>Analista accede al sistema.</i> 2. <i>Analista escoge el mes del cual desea generar el reporte.</i> 3. <i>Observa y analiza los datos con opción a imprimirlos.</i>

Tabla N°6. Escenario Usuario Analista.

IV.4. Identificación de los Casos de Uso y Relaciones.

A continuación se describen los casos de uso identificados y sus relaciones:

Nombre del caso de uso:	<i>Iniciar Sesión</i>
Actores participantes:	<i>Iniciado por: Usuario Administrador, Usuario Operador, Usuario Cliente, Usuario Analista: "Usuario".</i>
Condición de Entrada:	<i>Que el "Usuario" tenga acceso al sistema de registro de incidencias mediante un explorador Web.</i>
Flujo de eventos:	1. <i>El "usuario" ingresa al navegador Web y coloca la dirección donde se encuentra alojado el sistema de registro de incidencias.</i> 2. <i>Al ingresar el sistema presenta el formulario para el ingreso, el cual contiene los campos: nombre de usuario y contraseña.</i> 3. <i>Una vez agregados los datos de acceso, el "Usuario" presiona el botón "Ingresar".</i>

	4. El sistema toma estos datos y verifica que el "Usuario" ingresado exista en la base de datos. - En caso de existir el sistema remite al "Usuario" al menú de opciones que pueda ejecutar. - En caso de no existir o no ser correctos los datos ingresados, se muestra un mensaje de error, le indica al usuario que intente nuevamente y lo remite nuevamente al formulario de ingreso.
Condición de salida:	En caso de error, el sistema muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la operación.
Requerimientos especiales:	Tener creado un perfil de "usuario" en el sistema.

Tabla N°7. Casos de Uso: Iniciar Sesión

Nombre del caso de uso:	Crear Perfil de Usuario
Actores participantes:	Iniciado por: Usuario Administrador: "Usuario". Se comunica con: Usuario Operador, Usuario Analista, Usuario Cliente.
Condición de Entrada:	El usuario debe haber iniciado sesión.
Flujo de eventos:	- El "Usuario" accede a la opción de "Crear perfil de usuario". - El sistema le presenta el formulario para el perfil del usuario, el cual incluye los siguientes campos: cédula del usuario, nombre, apellido, número de empleado, tipo de usuario (donde se da la opción de los diferentes tipos de usuarios que pueden ser: Administrador, Operador, Cliente y Analista), cargo, correo electrónico, nombre de usuario y contraseña. - Una vez completado el formulario, el "Usuario" presiona el botón "agregar". - En caso de que alguno de los campos no este lleno el sistema le notificará al usuario que llene todos los campos. - El sistema verifica que el número de de cédula y nombre de usuario del perfil ingresado no existan en la base de datos. - En caso de existir, le muestra un mensaje de error al "Usuario" y solicita que ingrese otro valor para los datos repetidos.

	En caso de no existir almacena los datos y envía un correo electrónico al “Usuario” con copia al nuevo usuario registrado en el sistema.
Condición de salida:	El sistema muestra un mensaje indicando que la operación fue realizada con éxito. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Haber solicitado al “Usuario” la creación de un nuevo perfil usuario vía correo electrónico.

Tabla N°8. Casos de Uso: Crear Perfil Usuario.

Nombre del caso de uso:	Listar los Usuarios
Actores participantes:	Iniciado por: Usuario Administrador.
Condición de Entrada:	El “Usuario” debe iniciar sesión. Extiende los casos de uso: “Modificar Perfil” y “Eliminar Perfil”.
Flujo de eventos:	- El “Usuario” accede a la opción de “Listar Usuarios”. - El sistema muestra el listado de todos los usuarios registrados en el sistema con los datos mas relevantes de los mismos. Cada entrada de la lista presenta la opción “modificar” y “eliminar”. - El “Usuario” podrá imprimir el listado presionando el botón “imprimir”. - El “Usuario” podrá ordenar el listado por cualquiera de los campos mostrados en el listado (por Ej: cédula, nombre, apellido, número de empleado, etc.)
Condición de salida:	Si no hay usuarios en la base de datos, el sistema muestra un mensaje indicando que no existen usuarios que listar. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Este caso de uso no tiene requerimientos especiales.

Tabla N°9. Casos de Uso: Listar los Usuarios.

Nombre del caso de uso:	Listar Histórico de Accesos y Operaciones
Actores participantes:	Iniciado por: Usuario Administrador.
Condición de Entrada:	El “Usuario” debe iniciar sesión.
Flujo de eventos:	- El “Usuario” accede a la opción “Histórico de Accesos y Operaciones”. - El sistema le solicita que seleccione una fecha de inicio y fecha fin para definir el rango de búsqueda,

	en caso de que no sean llenados ninguno de los campos, el sistema traerá toda la información almacenada en la base de datos. 3. El sistema mostrará el listado de todos los accesos que se han hecho al sistema y las operaciones realizadas en el mismo, ordenándolos por fecha y hora. Por cada entrada se mostrará: Hora en la cual se realizó el acceso o la operación, Acción que fue realizada, que afectó dicha acción y qué usuario la realizó.
Condición de salida:	Si no hay accesos ni operaciones en la base de datos, el sistema muestra un mensaje indicando que no existen. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Este caso de uso no tiene requerimientos especiales.

Tabla N° 10. Casos de Uso: Listar Histórico de Acceso y Operaciones.

Nombre del caso de uso:	<i>Modificar Perfil</i>
Actores participantes:	Iniciado por: Usuario Administrador. Se comunica con Listar los Usuarios
Condición de Entrada:	El "Usuario" debe iniciar sesión.
Flujo de eventos:	1. El "Usuario" accede a la opción de "Modificar perfil de usuario". 2. El sistema presenta el formulario para modificar los campos del perfil del usuario deshabilitándolos campos: número de cédula, nombre, apellido y número de empleado. 3. El "Usuario" modifica los datos y presiona el botón "modificar".
Condición de salida:	El sistema muestra un mensaje de operación realizada con éxito y un botón para retornar al Listado de Usuarios. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Que existan usuarios previamente creados en el sistema y que se haya solicitado al Administrador la modificación del perfil de usuario vía correo electrónico.

Tabla N° 11. Casos de Uso: Modificar Perfil.

Nombre del caso de uso:	<i>Eliminar Perfil</i>
Actores participantes:	Iniciado por: Usuario Administrador. Se comunica con Listar los Usuarios
Condición de Entrada:	El "Usuario" debe iniciar sesión.

Flujo de eventos:	1. El "Usuario" selecciona del listado de usuarios los usuarios que desea eliminar y presiona el botón "Eliminar Usuarios". 2. El sistema le solicita confirmación antes de realizar la operación. 3. Si el "Usuario" confirma la operación, se eliminan los perfiles seleccionados de la base de datos.
Condición de salida:	El sistema muestra un mensaje de operación realizada con éxito y un botón para retornar al Listado de Usuarios. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Que existan usuarios previamente creados en el sistema y que se haya solicitado al Administrador la eliminación del perfil de usuario vía correo electrónico.

Tabla N°12. Casos de Uso: Eliminar Perfil.

Nombre del caso de uso:	<i>Crear Incidencia</i>
Actores participantes:	Iniciado por: <i>Usuario Operador, Usuario Cliente: "Usuario".</i>
Condición de Entrada:	<i>El "Usuario" debe iniciar sesión.</i>
Flujo de eventos:	1. El "Usuario" accede a la opción de "Reportar Incidencia". 2. El sistema valida los datos del "Usuario" y presenta el formulario de creación de incidencias correspondiente a su perfil de usuario. 3. Una vez completado el formulario, el "Usuario" presiona el botón "Guardar". 4. El sistema almacena los datos ingresados por el "Usuario" y envía un correo electrónico al área de atención de incidencias notificando que se ha reportado una nueva incidencia.
Condición de salida:	El sistema muestra un mensaje de operación realizada con éxito, indicando número de reporte para que se pueda llevar el respectivo seguimiento. En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.
Requerimientos especiales:	Se debe haber detectado una incidencia para poder reportarla.

Tabla N°13. Casos de Uso: Crear Incidencia.

Nombre del caso de uso:	<i>Listar Incidencias</i>
Actores participantes:	Iniciado por: <i>Usuario Operador, Usuario Cliente: "Usuario".</i>

Condición de Entrada:	<i>El "Usuario" debe iniciar sesión. Extiende el caso de uso: "Consultar Detalle de Reportes de Incidencias".</i>
Flujo de eventos:	1. El "Usuario" accede a la opción "Consulta de Reportes de Incidencia". 2. Según el tipo de usuario el sistema le presenta el listado de todos los reportes de incidencias que han sido reportados por su área, en caso de que el usuario sea de tipo operador el sistema muestra el listado de todos los reportes almacenados en la base de datos, ordenados por fecha y hora de creación. 3. El "Usuario" podrá cambiar el orden según el campo de su preferencia (por ejemplo: número de incidencia, número de ticket externo, aperturado por, estado de incidencia, etc.). 4. Si el "Usuario" es un usuario operador, el sistema le presenta en cada entrada de la lista la opción "ver detalle". 5. El "Usuario" podrá imprimir el listado presionando el botón "imprimir".
Condición de salida:	<i>En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.</i>
Requerimientos especiales:	<i>Que existan reportes de incidencias previamente creados en el sistema.</i>

Tabla N° 14. Casos de Uso: Listar Incidencias.

Nombre del caso de uso:	<i>Consultar Detalle de Incidencias</i>
Actores participantes:	<i>Iniciado por: Usuario Operador: "Usuario".</i>
Condición de Entrada:	<i>El "Usuario" debe iniciar sesión. Extiende el caso de uso: "Actualizar Reportes de Incidencias".</i>
Flujo de eventos:	1. El "Usuario" accede a la opción "ver detalle". 2. El sistema le presenta el formulario de detalles de incidencias con todos los campos que han sido llenados al momento de registrar la incidencia, deshabilitando los campos principales, tales como, numero de incidencia, fecha y hora de registro, descripción principal, etc. 3. El "Usuario" consulta los todos los campos llenos, y podrá imprimir la información presionando el botón "imprimir".
Condición de salida:	<i>En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.</i>

Requerimientos especiales:	<i>Que existan reportes de incidencias previamente creados en el sistema.</i>
----------------------------	---

Tabla N° 15. Casos de Uso: Consultar Detalle de Incidencias.

Nombre del caso de uso:	<i>Actualizar Incidencias</i>
Actores participantes:	<i>Iniciado por: Usuario Operador: "Usuario".</i>
Condición de Entrada:	<i>El "Usuario" debe iniciar sesión.</i>
Flujo de eventos:	1. El "Usuario" accede a la opción "ver detalle". 2. El sistema le presenta el formulario de incidencias con todos los campos que han sido llenados al momento de registrar la incidencia, deshabilitando los campos principales, tales como, numero de incidencia, fecha y hora de registro, descripción principal, etc. 3. El "Usuario" ingresa en los campos habilitados los datos que desea llenar, y presiona el botón "Actualizar". 4. El sistema solicita confirmación de la actualización. 5. El "Usuario" confirma. 6. El sistema almacena los cambios en la Base de Datos.
Condición de salida:	<i>El sistema muestra un mensaje de operación realizada con éxito, indicando número de reporte para que se pueda llevar el respectivo seguimiento.</i> <i>En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.</i>
Requerimientos especiales:	<i>Que existan reportes de incidencias previamente creados en el sistema.</i>

Tabla N° 16. Casos de Uso: Actualizar Incidencias.

Nombre del caso de uso:	<i>Eliminar Incidencias</i>
Actores participantes:	<i>Iniciado por: Usuario Administrador.</i>
Condición de Entrada:	<i>El "Usuario" debe iniciar sesión.</i>
Flujo de eventos:	1. El "Usuario" accede a la opción "Listar e Incidencias". 2. El sistema muestra el listado de todas las incidencias almacenadas en la base de datos, para cada entrada presenta una casilla de selección. 3. El "Usuario" marca la(s) casilla(s) correspondiente(s) a la(s) incidencias que desee eliminar y presiona el botón "Eliminar".

	4. <i>El sistema pide confirmación.</i> 5. <i>El usuario confirma.</i> 6. <i>Se realizan los cambios en la base de datos.</i>
Condición de salida:	<i>En caso de error, se muestra un mensaje informando el tipo y descripción de la falla ocurrida durante la transacción.</i>
Requerimientos especiales:	<i>Que existan incidencias reportadas previamente en el sistema.</i> <i>Que se haya solicitado al administrador la eliminación de dichas incidencias mediante las vías formales, como correo electrónico, por razones de equivocación por parte del usuario al guardar un reporte de incidencia.</i>

Tabla N° 17. Casos de Uso: Eliminar Incidencias.

IV.5. Identificación de Requerimientos No Funcionales

- El diseño debe ayudar a prevenir posibles errores.
- Se debe facilitar y optimizar el acceso a los usuarios.
- El sistema debe proveer al usuario ayuda en línea.
- El sistema debe ser compatible con diferentes navegadores Web.
- El sistema debe permitir el acceso concurrente al mismo.
- El sistema debe proveer elementos de seguridad para controlar y monitorear el acceso al mismo.
-

IV.6. Diagrama de Casos de Uso.

A continuación se presenta el diagrama principal de casos de uso del Sistema de Gestión de Incidencias:

- Nivel 0.

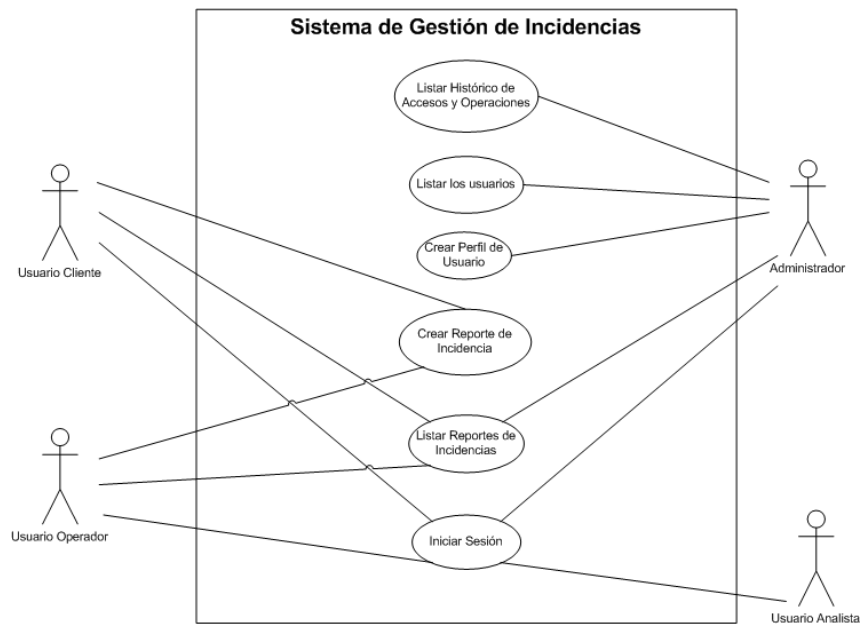


Figura N°8. Diagrama Casos de Uso Nivel 0: Sistema de Gestión de Incidencias.

- Nivel 1.

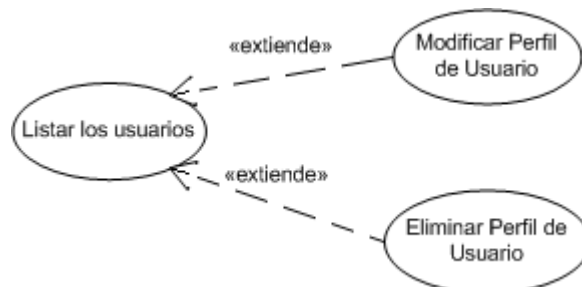


Figura N° 9. Diagrama Casos de Uso Nivel 1: Listar los Usuarios

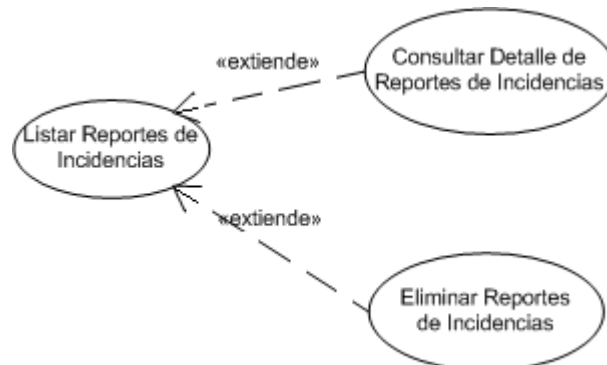


Figura N° 10. Diagrama Casos de Uso Nivel 1: Listar Reporte de Incidencias.

- Nivel 2.



Figura N°11. Diagrama Casos de Uso Nivel 1: Consultar Detalle de Reporte de Incidencias.

Capítulo V: Análisis del Sistema.

En este capítulo se describen las actividades que transformaran los diagramas de Casos de Uso definidos durante la recopilación de requerimientos en los diferentes modelos de análisis, así como el diagrama de secuencia, mostrando la iteración de los objetos de entidad, frontera y control.

V.1. Modelo Funcional.

Consiste en refinar y validar con los usuarios los Casos de Uso anteriormente contruidos. Utilizando los recursos previamente reunidos en la fase de recopilación de requerimientos.

Para llegar a la validación se revisaron las entrevistas efectuadas varias veces, observamos las anotaciones realizadas y cualquier otro artefacto obtenido de las entrevistas.

Finalmente, se presentó el diagrama de Casos de Uso principal el cual es mostrado en el capítulo anterior con sus respectivas validaciones (véase Capítulo IV sección IV.6).

V.2. Modelo de Objeto de Análisis.

A continuación los objetos de entidad, frontera y control identificados en el dominio del problema.

V.2.1. IDENTIFICACIÓN DE LOS OBJETOS DE ENTIDAD.

Examinando los casos de uso, se detectaron los siguientes objetos entidad:

Nombre	Descripción
Usuario	Persona que ingresa al sistema
Perfil	Perfil de usuario asociado a la persona que ingresa al sistema.
Operación	Actividad (agregar, modificar y eliminar)
Acceso	Registro de fecha y hora de ingreso del usuario.

Tabla N° 18. Objetos de Entidad.

V.2.2. IDENTIFICACIÓN DE LOS OBJETOS DE FRONTERA.

A continuación se muestran los diferentes objetos fronteras identificados:

BotonIniciarSesion	MensajeDeExito
MensajeDeError	FormularioInicioSesion
BotonSeleccionarOperacion	FormularioOpcionesSistema
FormularioReporteIncidencia	BotonRegistrarIncidencia
FormularioBusqueda	BotonBuscar
BotonVerDetalle	FormularioDetalle
BotonEliminar	BotonModificar

MenuPrincipal	BtonListarUsuario
BotonCrearperfil	BotonListarHistorico
FormularioNuevoUsuario	

Tabla N° 19. Objetos de Frontera.

V.2.3. IDENTIFICACIÓN DE LOS OBJETOS DE CONTROL.

Estos objetos son los encargados de coordinar los objetos de entidad y los objetos de frontera. A continuación una lista de los objetos de control identificados:

ControlUsuario	ControlIncidencia
ControlInicio	ControlListado
ControlCambioClave	ControlReportesEstadisticos

Tabla N° 20. Objetos de Control.

V.2.4. DIAGRAMA DE SECUENCIA.

La idea básica de un diagrama de secuencia consiste en representar el flujo de eventos de un Caso de Uso a través de los objetos definidos en el modelo de análisis funcional y el intercambio de mensajes entre ellos. A continuación se muestran los diagramas de secuencia de las principales funciones que realizará el Sistema de Gestión de Incidencias.

En las siguientes páginas mostraremos el diagrama de secuencia de las funciones principales del sistema.

Consultar Incidencias.

A continuación se muestra el diagrama que representa el cómo se realiza la consulta de incidencias reportadas dentro del Sistema de Gestión de Incidencias.

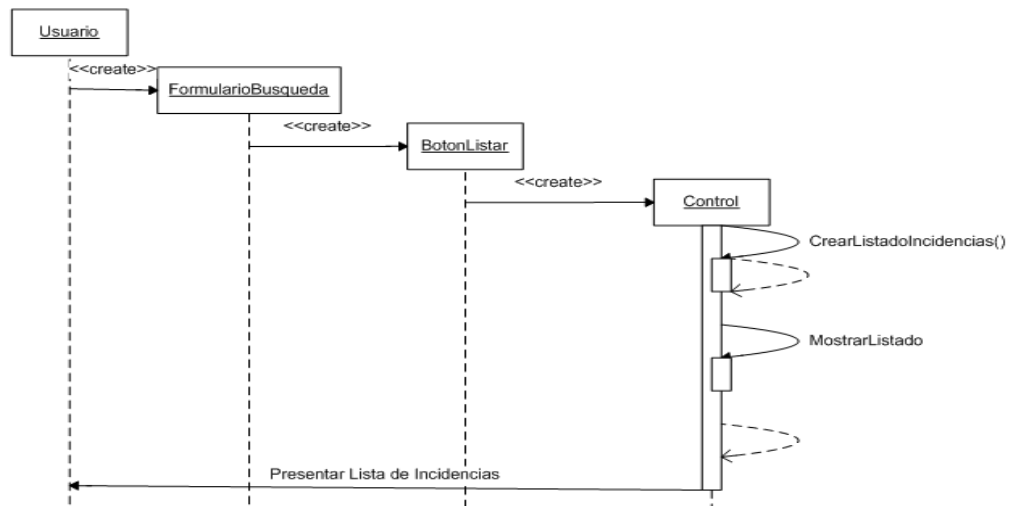


Figura N°12. Diagrama de Secuencia: Consultar Incidencias.

Registrar Incidencia.

A continuación se muestra el diagrama que representa el cómo se registra una nueva incidencia dentro del Sistema de Gestión de Incidencias.

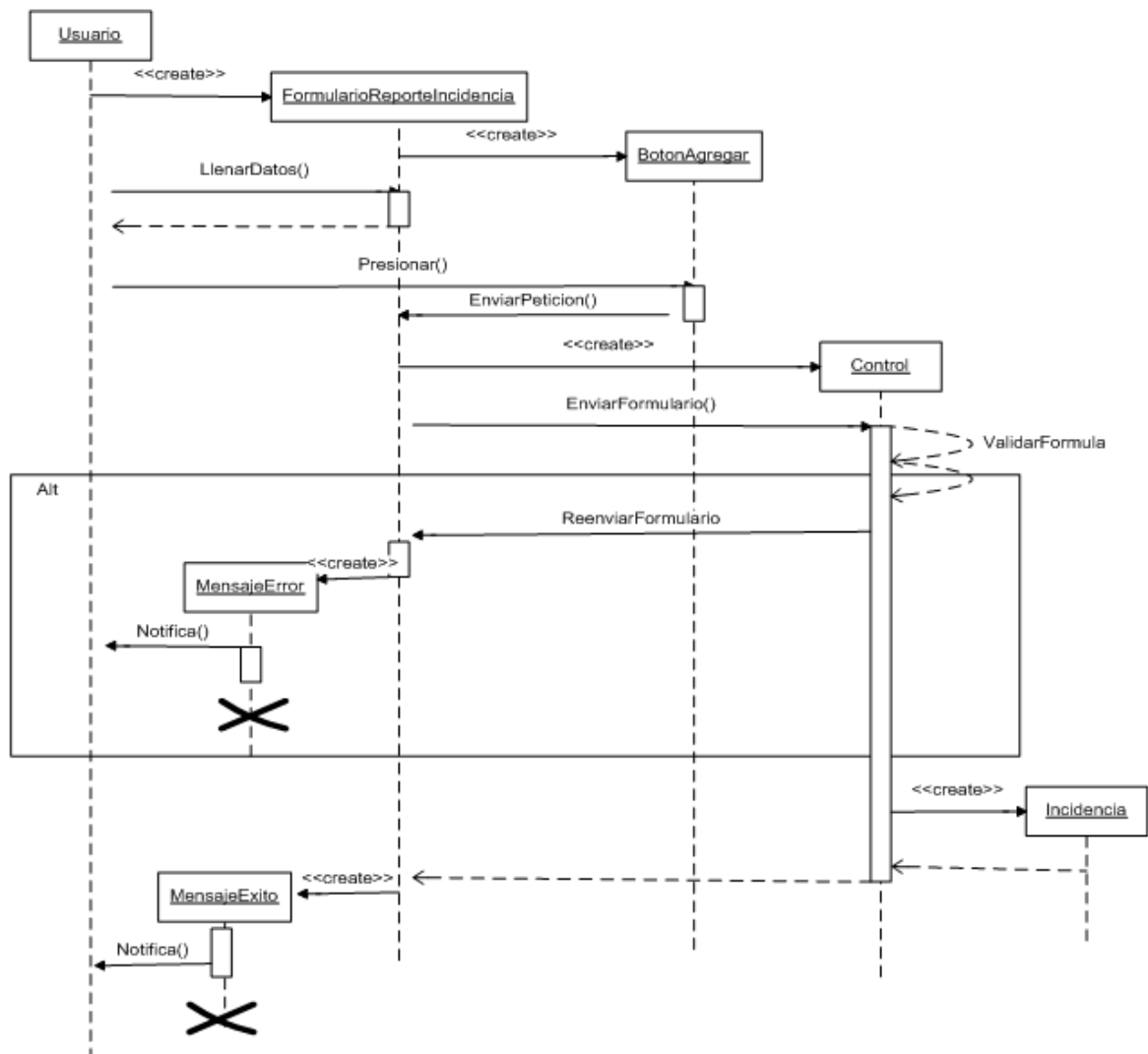


Figura N°13. Diagrama de Secuencia: Registrar de Incidencias.

Inicio de Sesión.

A continuación se muestra el diagrama que representa el inicio de sesión dentro del Sistema de Gestión de Incidencias.

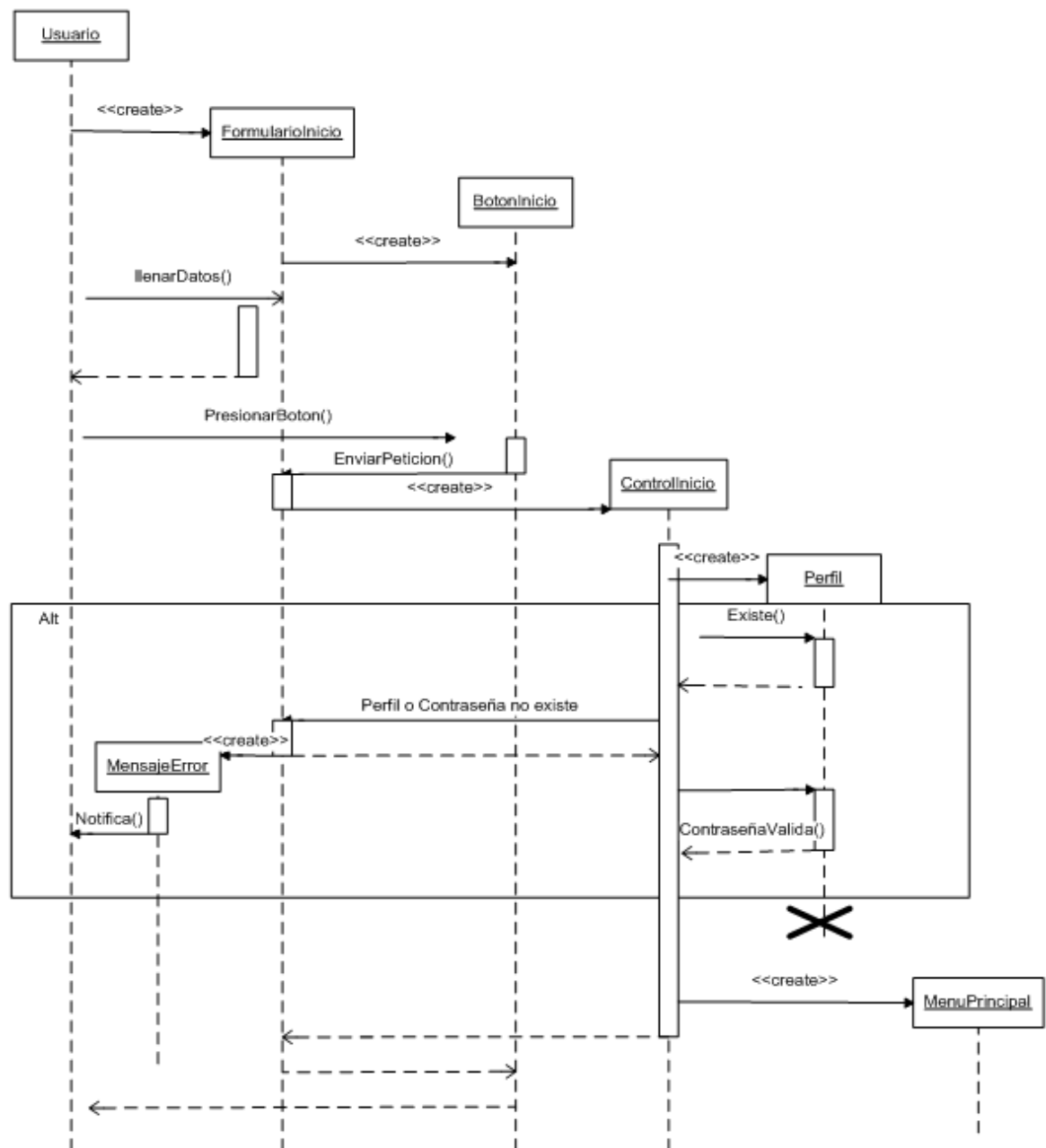


Figura N°14. Diagrama de Secuencia: Inicio de Sesión.

Capítulo VI: Diseño del Sistema.

En este capítulo se describe la transformación del modelo de análisis al modelo de diseño de sistema. Se presenta el Modelo de la Base de Datos que soportará el Sistema de Gestión de Incidencias, al igual que se mostrará el diseño de la interfaz que se utilizará en dicho sistema.

VI.1. Base de Datos

En esta etapa se describe la estrategia de almacenamiento que se seleccionó para aplicar a los datos del Sistema de Gestión de Incidencias, al igual que se describe el proceso de diseño de la Base de Datos, para así poder mostrar el diagrama relacionado a dicho diseño.

VI.1.1. SELECCIÓN DE LA ESTRATEGIA PARA EL ALMACENAMIENTO DE LOS DATOS.

De los modelos existentes para realizar la tarea de almacenamiento de datos, se seleccionó el Modelo Relacional de Base de Datos; ya que en este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que se almacenan no tiene mayor relevancia, además la información puede ser recuperada o almacenada por medio de consultas a dichas relaciones, lo que ofrece una amplia flexibilidad a la hora de administrar la información.

VI.1.2. DISEÑO DE LA BASE DE DATOS.

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así el diseño de una base de datos relacional se descompone en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual forma parte de las especificaciones de requisitos de los usuarios y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del Sistema Manejador de Base de Datos que se vaya a utilizar para manipularla.

Modelo Entidad-Relación.

En este capítulo reflejamos el producto final del diseño conceptual.

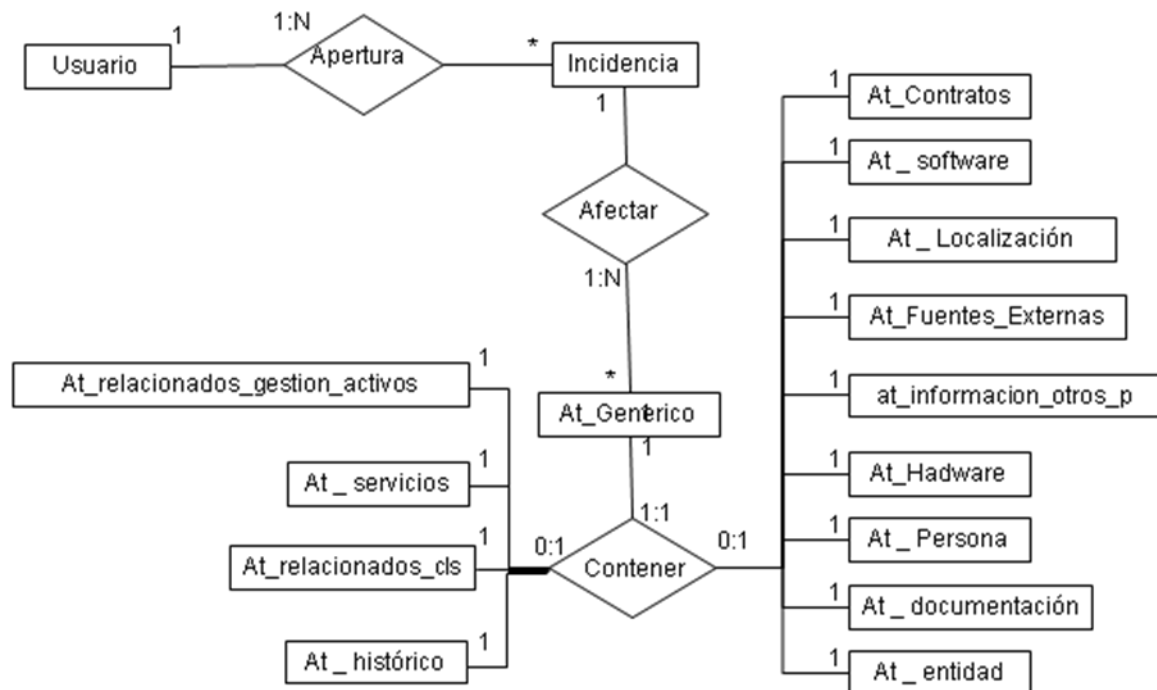


Figura N°15. Modelo Entidad-Relación.

VI.2. Diseño del Sistema

El diseño es un modelo del sistema, realizado con una serie de principios y técnicas, que permite describir el sistema con el suficiente detalle como para ser implementado. Pero los principios y reglas no son suficientes, en el contexto de diseño podemos observar que los buenos ingenieros tienen esquemas y estructuras de solución que usan numerosas veces en función del contexto del problema. Estos esquemas y estructuras son conceptos reusables y nos permiten no reinventar la rueda. Un buen ingeniero reutiliza un esquema de solución ante problemas similares [Lag-2007]. Es por ello que para la realización de este trabajo se decide utilizar un patrón de diseño de software para poder simplificar el diseño del mismo.

VI.2.1. PATRÓN “MODELO-VISTA-CONTROLADOR” (MVC).

En esta etapa se toma en consideración el uso de un patrón de diseño de software orientado a desarrollo de aplicaciones Web como lo es el **Patrón “Modelo- Vista-Controlador”** ya que el mismo nace con la premisa de que: “La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio”. [Lag-2007], como lo es en el caso de aplicaciones Web, entonces si trabajamos mezclando los componentes de interfaz de usuario con los del negocio, tendremos como consecuencia que al momento de querer cambiar de interfaz tendremos que modificar también los componentes de negocio, esto ocasiona más trabajo y más riesgo de error. Por tanto al usar este patrón buscamos desacoplar la vista del modelo y así poder mejorar la reusabilidad, de esta manera las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Dicho modelo está compuesto por tres elementos, los cuales son:

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo clásico es el de la información de una base de datos, que se puede presentar de diversas formas: diagrama de torta, de barras, tabular, etc.

En el diseño del Sistema de Gestión de Incidencias se utilizó este esquema de la siguiente manera:

1. Se le presentará la interfaz de ingreso al sistema (la vista) donde se le solicita nombre de usuario y contraseña.
2. El usuario ingresa los datos solicitados y presiona el botón “ingresar”, de estos datos son enviados al servidor (el controlador) quien consultará en la base de datos (el modelo) si el usuario existe en el sistema.
3. Dependiendo del valor de la consulta el servidor (el controlador) enviará una respuesta, la cual será mostrada en la interfaz (la vista).

Al igual que el caso expuesto anteriormente, el uso del Patrón “Modelo-Vista-Controlador”, se puede evidenciar en el resto del sistema como cuando se registra una nueva incidencia, se registra un nuevo usuario, se realiza una consulta de algún reporte estadístico, etc.

Frameworks.

Actualmente para la implementación de este patrón se ha vuelto muy popular el uso de software llamado framework, el cual ofrece facilidades a los programadores, como por ejemplo que les permite crear controladores que puedan ser probados completamente, acciones, modelos y vistas que puedan integrarse con diversas plantillas.

En particular, para el diseño del Sistema de Gestión de Incidencias se decidió utilizar un framework muy popular llamado CodeIgniter, el cual está compuesto por un conjunto de herramientas para personas que construyen su aplicación Web usando PHP. Su objetivo es permitirle a los programadores desarrollar proyectos mucho más rápido de lo que podrían si lo escribiesen desde cero, proveyéndoles un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a esas librerías.

Es importante hacer notar que CodeIgniter tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. Si no se necesita agregar

separación, o se descubre que mantener los modelos requiere más complejidad de la deseada, pueden ignorarse y construir la aplicación mínimamente usando Controladores y Vista. CodeIgniter también permite incorporar código existente, o incluso desarrollar librerías de núcleo para el sistema, habilitando el trabajar en una forma que hace que tenga más sentido para cada programador [CodeIgniter-2009].

pChart.

Basándose en los escenarios que debe cubrir el Sistema de Gestión de Incidencias, los cuales fueron definidos en el Capítulo IV: Especificación de Requerimientos, específicamente en el identificado como: “*Generar Reportes Estadísticos*” se decide utilizar pChart para lograr este cometido.

pChart es una librería escrita en PHP que permite dibujar gráficos estadísticos., cuyas características principales son:

- Recibe datos desde consultas SQL, archivos CSV o de datos suministrados de forma manual.
- Los gráficos utilizan algoritmos que logran que los gráficos sean de alta definición.
- Permite crear muchos tipos de gráficos, entre las que se encuentran: torta, barras, lineales, 3D, planos, etc.

[pChart-2009].

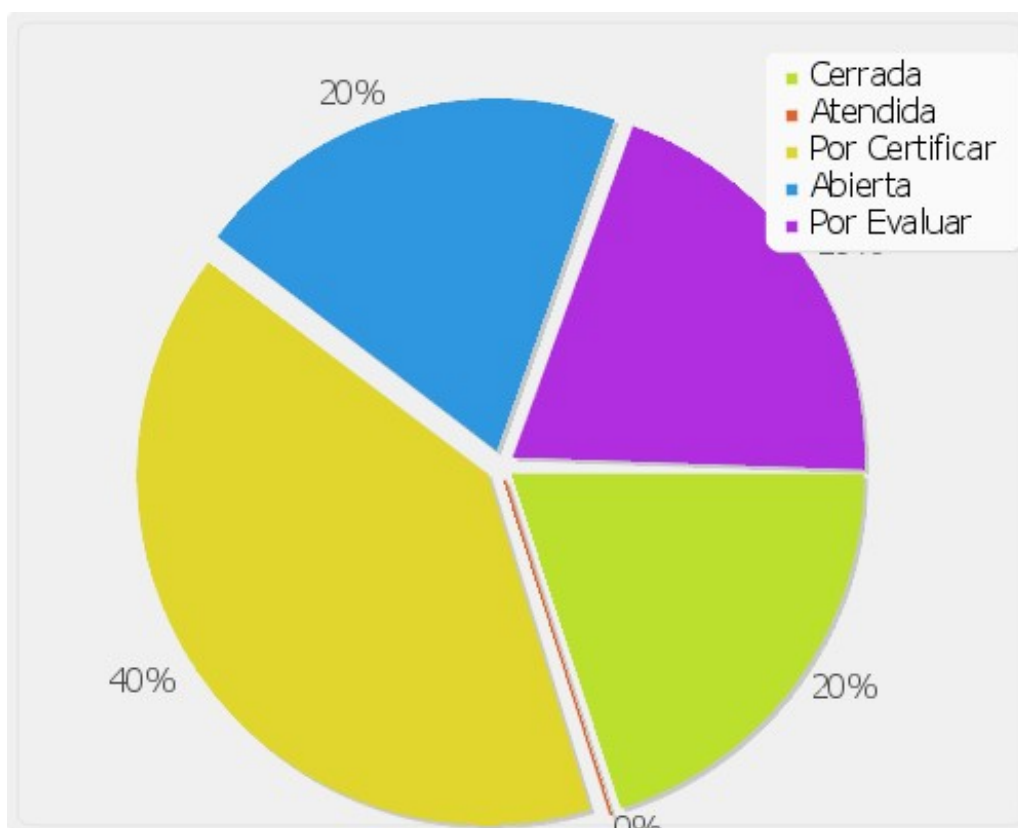


Figura N°16. Ejemplo Reporte Estadístico generado con pChart.

Con ayuda de esta librería se logró de forma sencilla realizar los reportes estadísticos, y así permitir a los usuarios de tipo “Analista” poder verificar mensualmente el porcentaje de distribución de las incidencias en los distintos estados, como se muestra en la figura 16 (recuérdese que los estados por los que puede pasar una incidencia son: Abierta, Atendida, Por Certificar, Por Evaluar, Cerrada y que sólo los usuarios de tipo “Operador” y “Administrador” pueden modificar dichos estados).

VI.3. Diseño de Interfaz

En esta etapa se crean los patrones de diseño como herramientas que ayudan a la producción de interfaces usables, junto con los lineamientos para el diseño de la interfaz de usuario.

VI.3.1. PATRONES DE INTERACCIÓN.

La aprobación final de una aplicación de software por parte del usuario depende en gran medida de la percepción que éste tenga del sistema, por lo que la interfaz juega un papel importante. Cuando se habla del diseño de interfaces de usuario se manifiesta la importancia de diseñar interfaces usables (fáciles de usar, robustas, flexibles, etc.), sin embargo la forma de incorporar esta característica es poco clara. De aquí que se procura el uso de los **Patrones de Interacción**.

Un patrón de interacción está orientado a describir todo lo relacionado con la interfaz de usuario o la interacción entre el usuario y la aplicación; está dedicado a presentar soluciones exitosas a problemas repetidos o recurrentes que se les presentan a los usuarios cuando utilizan las aplicaciones computacionales, al mismo tiempo que promueve la reutilización de los buenos diseños ya existentes, minimizando así los tiempos de desarrollo de la interfaz de usuario. Los patrones de interacción deben estar siempre enfocados en los usuarios, buscando la producción de interfaces realmente usables; por ende, deben facilitar soluciones que garanticen su satisfacción durante la interacción que éste mantenga con las aplicaciones a desarrollar, debido a que el encargado de medir qué tan funcional es el sistema y cuán exitoso es, será el usuario.

En definitiva se puede decir que un patrón de interacción sirve para describir un problema y plantear así la mejor forma para solventarlo, facilita la comunicación entre las personas que conforman el equipo de desarrollo, facilita el diseño del prototipo de la interfaz de usuario, tomando en cuenta la activa participación de éste en la construcción de la aplicación y así lograr que la misma sea realmente usable, pensando siempre en que el usuario debe cumplir una tarea con el sistema y la interfaz debe estar orientada a darle al usuario todas las facilidades, confort y el poder para realizar esa tarea.

La mayoría de los patrones deben centrarse en soluciones que mejoren la usabilidad, es decir, que cada patrón debe mostrar el impacto en los indicadores de usabilidad. Estos indicadores de usabilidad, tal como lo propone Nielsen (1993), se centran en que los sistemas sean fáciles de memorizar, deben evitar lo más posible que el usuario cometa errores y en caso de ocurrir, proveer las vías para

recuperarse de la situación, que el usuario se sienta agrado de utilizar la aplicación, entre otros.

Según Acosta (2004), un patrón tiene un formato bien definido, y su estructura en términos generales es la siguiente:

- El nombre del patrón.
- El problema que resuelve.
- El contexto en el que se encuentra.
- La solución al problema planteado, con una breve explicación del mismo.
- El ejemplo gráfico de la solución.
- La usabilidad que describe el impacto del uso del patrón.
- La consecuencia que describe el resultado de aplicar el patrón.

Ahora bien, en la literatura es posible encontrar diferencias en la estructura de los patrones, de aquí que no todos los elementos mencionados anteriormente son esenciales; sólo se conservan como elementos indispensables los cuatro primeros.

Seguidamente se expone la lista de patrones propuestos por Welie (2007), algunos de los cuales fueron utilizados en la Intranet corporativa desarrollada:

- Patrón de enlace a página de inicio
- Patrón de página de inicio
- Patrón de color alternativo de filas
- Patrón de registro de usuario
- Patrón de acceso al sistema
- Patrón de casilla de búsqueda
- Patrón de búsqueda avanzada
- Patrón de resultados de búsqueda
- Patrón de paneles desplegables

A continuación se describen y muestran ejemplos de algunos de estos patrones:

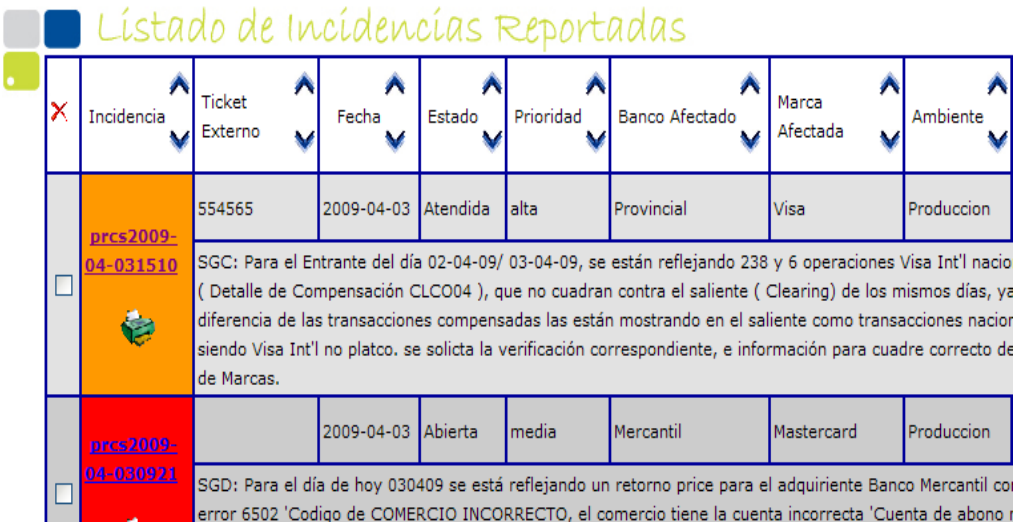
Nombre de Patrón:	Aplicación basada en Web
Problema:	El usuario necesita realizar tareas complejas en un sitio Web.
Contexto:	El sitio es para "hacer" cosas mas que para encontrar información. Los usuarios pueden realizar tareas complejas tales como leer y escribir correos electrónicos, colocar órdenes, manejar cuentas de banco, etc. En la mayoría de los casos hay "objetos" involucrados que pertenecen a los usuarios, los cuales necesitan crear, cambiar, eliminar, o actualizar. Una aplicación basada en la Web es una aplicacion que bien podria ser una aplicacion normal, sólo que ésta corre en un navegador Web.
Solución:	Se crean paneles que pueden ser ocultados o mostrados independientes unos de los otros. En la que en la mayoría de los casos hay "objetos" que los usuarios pueden manipular y así poder crear, cambiar, eliminar o actualizar dichos "objetos".
Ejemplo:	

Tabla N° 21. Patrones de Iteración: Aplicación basada en Web.

Nombre de Patrón:	Color alternativo de filas
Problema:	El usuario necesita leer o chequear una tabla en busca de una información en particular
Contexto:	Tabla de resumen de elementos
Solución:	Alternar el color de las filas para hacer la tabla más legible.
Ejemplo:	

	Nombre	Apellido	Nombre de Usuario	Cedula de Identidad	Código de Empleado	Tipo de Usuario
	Ruben	Pacheco	rpacheco	6376442	e07105	Operador
	Renny	Leon	rleon	123456	e07106	Analista
	Nelwin	Rodriguez	nrodriguez	16117818	e07328	Administrador
	Manuel	Mejias	mmejias	123456	e03567	Analista

Tabla N° 22. Patrones de Iteración: Color Alternativo de Filas.

Nombre de Patrón:	Registro de Usuario
Problema:	El usuario necesita registrarse para tener acceso al Sistema de Registro de Incidencias
Contexto:	Formulario de registro del sistema
Solución:	Se muestra el formulario que permite a los usuario obtener una sesión compuesta por un nombre de usuario y una contraseña con la cual puede acceder
Ejemplo:	

Tabla N°23. Patrones de Iteración: Registro de Usuario.

Nombre de Patrón:	Acceso al Sistema
Problema:	El usuario necesita identificarse para tener acceso al contenido del Sistema de Registro de Incidencias
Contexto:	Área de acceso al usuario
Solución:	Se presenta el área de acceso que permite al usuario registrado acceder al sistema a partir del nombre de usuario y la contraseña obtenidos en el proceso de registro
Ejemplo:	

	 Inicio de Sesión: Usuario: Contraseña: ¿Olvidó su clave? Haga click aquí
--	--

Tabla N° 24. Patrones de Iteración: Acceso al Sistema.

Nombre de Patrón:	Casilla de Búsqueda
Problema:	El usuario necesita encontrar información específica
Contexto:	Sistemas con área de búsqueda
Solución:	Se le presenta una casilla al usuario donde este pueda colocar el identificar del ítem del cual desea obtener información
Ejemplo:	

Tabla N° 25. Patrones de Iteración: Casilla de Búsqueda.

Nombre de Patrón:	Búsqueda Avanzada
Problema:	El usuario necesita encontrar un reporte específico de un gran conjunto de reportes registrados en el Sistema de Registro de Incidencias
Contexto:	Sistemas con área de búsqueda
Solución:	Se ofrece una función de búsqueda avanzada especificando el límite de la búsqueda y el orden en que aparecen los resultados
Ejemplo:	

--	--

Tabla N° 26. Patrones de Iteración: Búsqueda Avanzada.

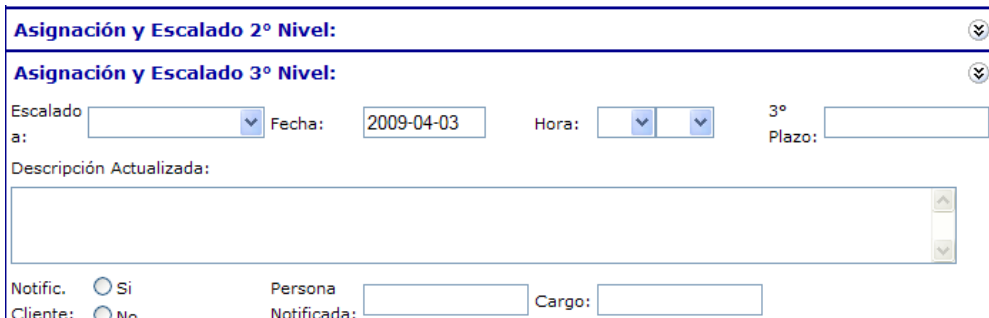
Nombre de Patrón:	Paneles Plegables
Problema:	El usuario necesita acceder a la información o las funcionalidades del sistema pero sólo bajo ciertas circunstancias, o de manera temporal
Contexto:	Sistemas donde se puedan obviar detalles o información dependiendo de las circunstancias
Solución:	Se crean paneles que pueden ser ocultos o mostrados independientes unos de los otros
Ejemplo:	

Tabla N° 27. Patrones de Iteración: Paneles Plegables.

VI.3.2. LINEAMIENTOS PARA EL DISEÑO DE LA INTERFAZ

Los lineamientos son reglas que se establecen en una organización para la construcción de interfaces de usuarios, a fin de:

- Promover la consistencia visual y funcional dentro de la aplicación, y con respecto a otras aplicaciones.
- Promover un buen diseño.
- Mantener la usabilidad de la aplicación.

A continuación se presentan tres aspectos fundamentales utilizados en el Sistema de Registro de Incidencias, los cuales surgieron tanto por demanda del usuario final como por la implementación de los patrones de interacción antes descritos:

Diseño

- Contraste de color entre el fondo, el texto y las imágenes, de manera que no perturben la legibilidad de la página.
- Uso apropiado y consistente del color, utilizando la paleta de colores seleccionada por el usuario.

- Consistencia en el uso de colores, imágenes y fondos dentro del sistema, con la finalidad de conservar una homogeneidad en las páginas de la misma.
- Adecuada organización de los elementos de cada uno de los formularios dentro de las páginas del sistema, a fin de facilitar la identificación de los mismos.

A continuación en la figura N° 17, se muestra un ejemplo donde se ve reflejado el cumplimiento del lineamiento de diseño: “Contraste de color entre el fondo, el texto y las imágenes, de manera que no perturben la legibilidad de la página”.



Figura N° 17.

lineamiento de Diseño.

Ejemplo de

Navegabilidad

- Esquema de navegación consistente, para permitir que el usuario se familiarice con las páginas visitadas y reconozca fácilmente todos los elementos mostrados.
- Consistencia en la forma y color de todos los enlaces incorporados en cada una de las páginas, a fin de mantener un mismo estándar en cada vínculo publicado facilitando su visualización.
- Títulos representativos que faciliten al usuario conocer dónde está ubicado dentro del sistema.

La siguiente imagen (Figura 18) muestra un ejemplo donde se cumple el lineamiento de navegabilidad: “Consistencia en la forma y color de todos los enlaces incorporados en cada una de las páginas, a fin de mantener un mismo estándar en cada vínculo publicado facilitando su visualización”.

Figura N°

Listado de Incidencias Reportadas							
☐	oprs2008-11-072200	448809448909	2008-11-07	Cerrada	alta		
		STRATUS: Se estan recibiendo mensajes de control con error CL9200 CLR011 1043 Create I					
☐	oprs2008-11-071309	448620	2008-11-07	Cerrada	alta		
		Copio texto del correo: "El siguiente es para informar que entre las 13:09 y 13:17 se presen					
		abortadas en el ambiente de Producción del Stratus, en este lapso de tiempo llegaron a ser 12					
		tareas siendo la hora critica 13:11 a 13:15. Es importante mencionar que este hecho teni					
		de hecho no sucedía desde que el Provincial dejó de informar el campo 40 en las transaccio					
☐	srvs2008-11-061000	448031	2008-11-06	Cerrada	baja	Provincial,Mercantil	
		El sistema está gnerando Ordenes de servicio de Mantenimiento Preventivo que no aplican y					
		especificaciones bajo la que fue crada la funcionalidad de generación automática de o/s. Se					
		V000000775, V000000778 V000000784 y V000000790 para el mismo comercio y para ser at					
		consecutivos. Deben generarse cada 30 días. Esta funcionalidad fue probada y certificada.					
☐	prcs2008-10-111023	447807	2008-10-11	Cerrada	alta	Mercantil	
		Para los días 11-12/10/08 el Reporte de Liquidación indica los siguientes estatus: NO definido					
		Otros. El mismo sigue manteniendo a la fecha estos estatus.					
☐	oprs2008-11-060700	0	2008-11-06	Cerrada	media		
		El día de hoy Franklin Moreno indica que a las 7:15am no se habían recibido los archivos de					

de lineamiento de Navegabilidad.

18. Ejemplo

Usabilidad

- Balance entre el contenido y la apariencia, con la finalidad de no sobrecargar la pantalla y dificultar la visibilidad de sus elementos.
- Uso correcto de terminologías y encabezado, para asegurar que el encabezado tenga consistencia con el contenido mostrado.
- Constante visibilidad en cuanto a la identificación de las páginas, para garantizar que el usuario esté al tanto que se encuentra en el Sistema de Registro de Incidencias.
- Empleo de una interfaz atractiva, intuitiva y familiar para el usuario, de manera de atraer la atención y promover el uso de la aplicación para la gestión de incidencias.
- Facilidad de aprendizaje y memorización, a fin de ofrecer una interacción sencilla y fácil de recordar cada vez que el usuario necesite utilizar el Sistema de Registro de Incidencias.

Para entender de manera visual el empleo del lineamiento de usabilidad se presenta a continuación la Figura N° 19, donde se ejemplifica el lineamiento: "Constante visibilidad en cuanto a la identificación de las páginas, para garantizar que el usuario esté al tanto que se encuentra en el Sistema de Registro de Incidencias".

Sistema de Registro de Incidencias

Detalle de la Incidencia

prcs2009-04-031510

Informante:

Sandra Ortega

Fecha:

2009-04-03

Cargo:

Analista Administra

Entidad:

TST

Modo Comunicación:

E-mail

Hora Inicio:

15:10:00

Hora Comunicación:

15:30:00

Aperturado por:

Area de Procesos

Banco Afectado:

☒ Provincial
 ☐ Mercantil

Marca Afectada:

☐ Mastercard
 ☒ Visa

Ambiente Afectado:

☒ Produccion
 ☐ Calidad

Descripción Detallada

SGC: Para el Entrante del día 02-04-09/ 03-04-09, se están reflejando 238 y 6
 operaciones Visa Int'l nacional no platco (Detalle de Compensación CLCO04),
 que no cuadran contra el saliente (Clearing) de los mismos días, ya que la

Clasificación y Primer Soporte:

Servicio Afectado:

Gestion de Comer

Clasificación:

Software

Afecta Negocio:

☒ Si
 ☐ No

Prioridad:

P2

Urgencia:

Alta

1º Plazo:

Figura N° 19. Ejemplo de lineamiento de Usabilidad.

Capítulo VII: Construcción del Sistema.

En este capítulo se abarca la adaptación de la fase de Construcción del Proceso Unificado al caso particular de la aplicación Web a desarrollar, especificando cada una de las tareas a realizar dentro de dicha fase.

Inicialmente se detallan dos módulos en los cuales fue dividida la implementación, cada uno agrupa distintos Casos de Usos, los cuales fueron descritos en la fase de Análisis del Sistema.

Para cada uno de estos módulos se describen las actividades efectuadas durante la fase de Construcción y finalmente, se muestra su respectivo prototipo de interfaz.

VII.1. Identificación de Módulos del Sistema

Durante la fase de construcción, se tomó la iniciativa de separar el sistema en dos módulos para facilitar su desarrollo, dichos módulos abarcan los Casos de Uso descritos en el Capítulo de Análisis del Sistema; a continuación se definen los módulos y su correspondencia con los Casos de Uso.

El primer módulo llamado Gestión de Usuarios, incluye todos los Casos de Uso referente al manejo de usuarios, los cuales son: “Crear Perfil de Usuario”, “Listar los Usuarios”, “Listar Historico de Acceso y Operaciones” y “Iniciar Sesión”.

El segundo módulo llamado Gestión de Incidencias, incluye los Casos de Uso donde se refleja interacción con los reportes de Incidencias, los cuales son: “Crear Reporte de Incidencia” y “Listar Reportes de Incidencias”.

VII.2. Construcción de los Módulos del Sistema

En este apartado se detallan las tareas realizadas para el desarrollo e implementación de cada uno de los módulos.

VII.2.1. MÓDULO: GESTIÓN DE USUARIOS.

El módulo Gestión de Usuarios, como su nombre lo indica incluye todo lo referente al manejo de usuarios dentro del Sistema de Registro de Incidencias, desde la creación hasta la eliminación, pasando por modificaciones como cambio de contraseñas, datos y perfiles.

VII.2.1.1. ANÁLISIS Y DISEÑO

Durante esta etapa se identifican las clases que involucran el desarrollo del módulo de gestión de usuarios, y se muestra la relación entre las mismas.

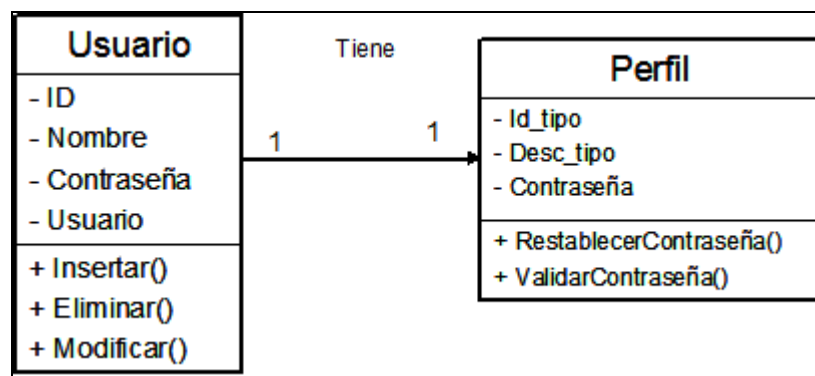


Figura N° 20.

Clase.

Diagrama de

Para la implementación se diseñó una función que permite insertar un nuevo usuario al sistema y asociarle un perfil, la aplicación maneja cuatro tipos de perfil cada uno de ellos se encuentra identificado por un código único y estos definen los permisos sobre la aplicación que puede tener cada usuario.

También se desarrolló una función que permite eliminar usuarios del sistema, siempre y cuando el usuario no haya tenido actividad en el mismo, en caso contrario, su cuenta quedará en estatus inactiva.

Cada usuario debe poseer una contraseña válida para tener acceso a la aplicación, para ello se diseñó una función que se encarga de corroborar la validez del usuario y contraseña y otra que permite verificar la existencia única de cada nombre de usuario.

VII.2.1.2. IMPLEMENTACIÓN

Las funciones antes descritas forman parte del Modulo de Gestión de Usuarios , cuya interfaz se muestra en las siguientes figuras:

Agregar nuevo usuario:

Figura N° 21.
registro de

Ejemplo de
usuario.

Eliminar usuarios:

Figura
Ejemplo de

usuarios.

N° 22.
eliminar

Ingreso al sistema:

Figura N° 23.
sistema.

Ejemplo de ingreso al

VII.2.1.3. PRUEBAS

Para constatar el correcto funcionamiento de la implementación del sistema, se realizaron pruebas piloto con cuatro usuarios cada uno con perfiles distintos. Estos realizaron pruebas al autenticarse en el sistema, verificando los permisos otorgados a cada usuario. Se tomaron en cuenta las observaciones de los usuarios para agregar mejoras al sistema.

VII.2.2. MÓDULO: GESTIÓN DE INCIDENCIAS.

El modulo de Gestión de Incidencias, es el módulo de mayor prioridad dentro del sistema, ya que es dónde se ve el ciclo de vida de las incidencias, desde que las mismas son reportadas dentro del sistema hasta que son solventadas, dejando registro de todos los avances que se realizan para lograr su solución.

VII.2.2.1. ANÁLISIS Y DISEÑO

Igual que en el módulo anterior en esta etapa se identifican las clases que involucran el desarrollo del mismo y su respectiva relación.

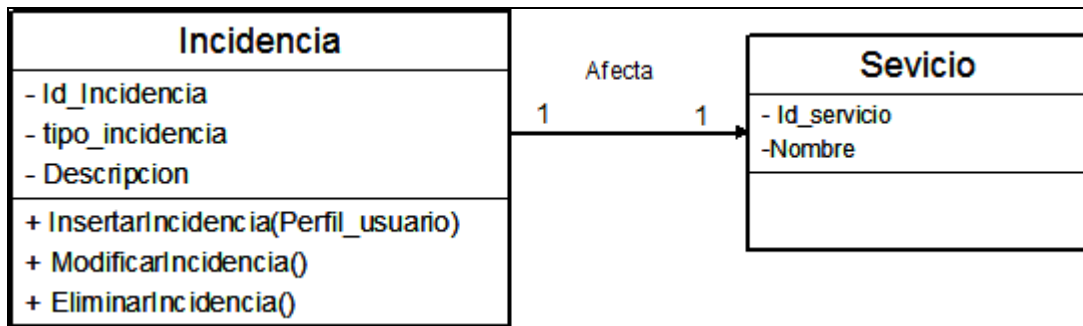


Figura N° 24. Diagrama de Clase.

En el caso de la implementación de este módulo en particular, la función para guardar un nuevo reporte de incidencia, recibe como parametro el perfil del usuario, ya que dependiendo del valor del mismo se guardarán ciertos datos por defecto, por ejemplo en el caso de que un reporte de incidencia sea abierto por un usuario de tipo cliente, la función no guardará datos sobre escalado de segundo o tercer nivel, ni mucho menos podrá escoger el estado de la incidencia (en este caso se guarda en estado "Abierto" por defecto), sólo guardará los datos básicos que se necesitan para que posteriormente los operadores puedan realizar el respectivo seguimiento del reporte hasta llegar a su solución. En cambio, un usuario de tipo Operador o Administrador podrá colocar información de a quien se escaló un reporte de incidencia, que estado tiene, desde "Abierto" hasta "Cerrado" pasando por distintos estados como "Atendido", "Por certificar", etc. Las figuras 25 y 26 muestran el formulario para un usuario de tipo Cliente y otro para un usuario de tipo Administrador, respectivamente, para que se pueda visualizar la diferencia entre ambas.

The screenshot shows a web form titled "Registro de Incidencias". It contains several input fields and checkboxes for recording an incident. The form is organized into sections for user information, incident details, and classification.

Form Fields:

- Informante:** A dropdown menu.
- Fecha:** A date input field.
- Hora Inicio:** A time selection widget with hour and minute dropdowns.
- Banco Afectado:** Two checkboxes: ☐ Provincial and ☐ Mercantil.
- Marca Afectada:** Two checkboxes: ☐ Mastercard and ☐ Visa.
- Ambiente Afectado:** Two radio buttons: ☐ Produccion and ☐ Calidad.
- Descripción Detallada:** A large text area for detailed description.
- Clasificación y Primer Soporte:**
 - Servicio Afectado:** A dropdown menu.
 - Afecta Negocio:** Two radio buttons: ☐ Si and ☐ No.
 - Prioridad:** A dropdown menu with a question mark icon.
 - Urgencia:** A dropdown menu.

Buttons: At the bottom, there are three buttons: "Limpiar", "Imprimir", and "Guardar".

Figura N° 25. Ejemplo de registro de incidencias para usuarios Clientes.

Registro de Incidencias

Informante:

Cargo:

Entidad:

Modo Comunicación:

Fecha: 2009-09-03

Hora Inicio: 09

Hora Comunicación: 09:41:21

Banco: ☐ Provincial

Marca: ☐ Mastercard

Ambiente: ☐ Produccion

Afectado: ☐ Mercantil

Afectada: ☐ Visa

Afectado: ☐ Calidad

Descripción Detallada

Clasificación y Primer Soporte:

Servicio:

Afectado:

Clasificación:

Afecta Negocio: ☐ Si ☐ No

Prioridad:

Urgencia:

1º Plazo:

Asignación y Escalado 2º Nivel:

Escalado a:

Fecha: 2009-09-03

Hora:

2º Plazo:

Descripción Actualizada:

Ticket Externo:

Asignación y Escalado 3º Nivel:

Resolución de Incidencia:

Comprobación de Usuario:

Confirmación: ☐ Si ☐ No

Cliente: ☐ Si ☐ No

Persona Confir.:

Cargo:

Cliente Conforme: ☐ Si ☐ No

Estado de la Incidencia:

Descripción Detallada

**Figura
Ejemplo
registro**

**N° 26.
de
de**

incidencias para usuarios Operadores.

Los reportes de incidencias no sólo son para ser creados y luego visualizarlos sino que luego de crearlos el usuario va a querer conocer: ¿Qué ha pasado con el caso reportado?, ¿Quién lo atiende?, saber si ya fue solventado, etc., para poder responder a todas esas interrogantes el sistema permite hacer actualizaciones a dichos reportes y así ir llevando una bitácora de avance, donde se puede observar el estado en el que se encuentra, las causas, la solución y los correctivos que fueron tomados para que no se vuelva a presentar dicho error, todo esto se maneja con la función modificar que posee el sistema, esta opción sólo podrá ser ejecutada por un usuario de tipo Operador o de tipo Administrador.

Como su nombre lo indica este es un módulo de Gestión de Incidencias por tanto si existe una función para guardar y otra para modificar debe existir una función para eliminar; dicha función sólo puede ser utilizada por el usuario de tipo "Administrador", ya que el hecho de que un reporte de incidencia sea eliminado no es algo que ocurra con mucha frecuencia (se debe recordar que este sistema debe mantener el registro de todos los incidentes presentados en la labor diaria de la empresa y así poder evitar que estos se repitan y en caso de que eso ocurra tener registrada la solución del mismo), es por ello que no todos los usuarios pueden tener el perfil para utilizar esta opción, la misma sólo se utiliza en caso de que haya duplicidad o errores en los reportes y se le sea notificado al usuario "Administrador"

sobre estos casos, eliminando estos tipos de reportes errados se podrá mantener unas estadísticas más veraces sobre la operatividad de la procesadora de medios de pagos, en la figura 27 se muestra como se le presenta la opción de eliminar las incidencias al usuario “Administrador”.

Ejemplo de reportes de

Listado de Incidencias Reportadas

☐ Incidencia
 Ticket Externo
 Fecha
 Estado
 Prioridad
 Banco Afectado
 Marca Afectada

	2009-09-02	media	Provincial,Mercantil
☐ srsvs2009-09-021034	El sistema IVR esta indicando que los operadores se encuentran ocupados aun cuando se intenta por segunda vez cae la llamada sin ningun problema. Si los operadores se primer intento el sistema no debe indicar entonces "que se encuentran ocupados". Es comercios que llaman e incide en el abandono de llamadas innecesariamente		
☐ qsvs2009-09-010901	0	2009-09-01	alta Provincial,Mercantil
☐ qsvs2009-09-010900	613964	2009-09-01	alta Provincial,Mercantil

Eliminar

**Figura N° 27.
eliminar
incidencias.**

VII.2.2.2. PRUEBAS

Las pruebas realizadas a este módulo consistieron en validar la correcta funcionalidad, verificando que en efecto se pudiera guardar, modificar, y eliminar las incidencias registradas en la aplicación:

- Validando que se guardaran en base de datos los mismos datos ingresados por el usuario sin alteraciones.
- Validando lo anterior pero en el caso de actualizaciones a los reportes.
- Validando que en efecto al momento de eliminar un reporte el mismo fuese borrado de la base de datos.

Conclusiones

Por la naturaleza iterativa del proceso de desarrollo RUP, se pudo analizar la opinión de los involucrados, la estabilidad y la calidad del producto en cada una de las iteraciones, de tal forma que el proyecto se pudiera ir refinando a medida que se avanzaba en su desarrollo. Así mismo, el proceso se adaptó a las características propias del proyecto, tanto en tamaño como en las condiciones que lo regularían.

Por consiguiente, se alcanzó un alto grado de satisfacción por parte de los usuarios, quienes manifestaron su conformidad con el desempeño de la aplicación y con los resultados obtenidos. De esta manera se avala que los objetivos planteados se han alcanzado, ya que el diseño de la aplicación Web desarrollada cumple a cabalidad con el propósito para el que fue concebida, generando así una serie de ventajas importantes sobre el proceso de gestión de incidencias, entre las que se encuentran:

- Manejo Automatizado del Proceso.
- Ahorro de tiempo y esfuerzo, se centralizó y automatizó la plantilla de registro de incidencia, facilitando el proceso de registro, actualización y consulta.
- Mayor rendimiento y productividad por parte del personal del Centro de Cómputo, ya que permite llevar la continuidad de los casos de incidencia al notificar de forma inmediata mediante correo electrónico, al ente encargado de dar solución a los mismos.
- Incrementa la satisfacción del personal brindando soluciones a las incidencias de forma rápida y eficaz.
- Disminuye considerablemente el tiempo de respuesta en el caso de que se presenten solicitudes que hayan sido solventadas con anterioridad.
- Permite disponer de la información en digital agilizando la búsqueda.
- Permite el monitoreo eficaz de las incidencias, a través de la generación de reportes e indicadores.

De esta manera la aplicación logra gestionar las incidencias de manera eficiente al facilitar la ejecución óptima de los procesos operativos entre los que se destacan el contacto que reporta (operador que atiende de la falla o incidencia), el contacto que soluciona (operador o asesor que resuelve o finaliza la incidencia), registro (el registro de manuales o acciones realizadas para solventar la incidencia), además de hacer posible la generación de reportes, indicadores y un monitoreo eficaz de las incidencias.

Referencias Bibliográficas

- [*Informaticamilenium-2009*] Definiciones de Aplicaciones Web [En línea]
<http://www.informaticamilenium.com.mx/Paginas/espanol/sitioweb.htm>
- [*Masternewmedia-2009*] Beneficios de las Aplicaciones Basadas en Web [En línea] http://www.masternewmedia.org/es/aplicaciones_web

- [Mitecnologico-2008] Componentes de Aplicaciones Clientes/Servidor
<http://www.mitecnologico.com/Main/ComponentesDeAplicacionesClienteServidor>
- [mipagina.cantv.net-2009] Características Cliente/Servidor [En línea]
<http://www.mipagina.cantv.net/continuacion/grupo3.htm>
- [adelat-2006] Lenguajes del Lado del Servidor o Cliente [En línea]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
- [w3c-2005] W3C. Hojas de Estilo [En línea]
<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>
- [Riv-2009] Víctor Rivas Santos (2009), JavaScript [En línea]
http://geneura.ugr.es/~victor/cursillos/javascript/js_intro.html
- [ProgramasYorbin-2009] Creaciones de Páginas Web Dinámicas [En línea]
<http://www.programas-yorbin.blogspot.com/2009/02/manual-de-php-creacion-de-paginas-web.html>
- [tutorialgratistutorial-2008] Manual PHP [En línea]
<http://tutorialgratistutorial.blogspot.com/2008/03/tutorial-php-manual-php-curso-php.html>
- [Apache-2009] Apache (2009). Módulos de Multiprocesamiento. [En línea]
<http://httpd.apache.org/docs/2.0/es/mpm.html>
- [Apache-2009] Apache Software Foundation
<http://httpd.apache.org/docs/2.2/es/mpm.html>
- [MYSQL-2009] Manual de Referencia MySQL
<http://dev.mysql.com/>

- [Let-2009] P. Letelier (2009) [En línea]

<https://pid.dsic.upv.es/C1/Material/.../introducción%20a%20RUP.doc>
- [Cas-2008] Castells Delgado Francisco José (2008), Ahorro e Inversión BlogActiBva [En Línea]

<http://www.actibva.com/blog/2008/04/16-es-la-industria-de-medios-de-pago-estrategica-para-las-entidades-financieras>

<http://www.actibva.com/blog/2008/05/29-es-la-industria-de-los-medios-de-pago-estrategica-para-las-entidades-financieras-y-ii>

<http://www.actibva.com/blog/2008/09/10-es-la-industria-de-medios-de-pago-estrategica-para-las-entidades-financieras-2>
- [Lag-2007] Lago, Ramiro (2007), Patrones de Diseño de Software [En Línea]

http://www.proactiva-calidad.com/java/patrones/index.html#algunos_patrones
- [Welie-2007] Welie.com (2009), A Pattern Library for Interaction Design [En Línea]

<http://www.welie.com/>
- [CodeIgniter-2009] CodeIgniter (2009), Manual de Code Igniter [En Línea]

www.conocimientovirtual.edu.co
- [pChart-2009] pChart (2009), pChart a PHP Class to build Charts

<http://pchart.sourceforge.net/>