



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA PARA LA ADMINISTRACIÓN,
ATENCIÓN Y CONTROL DE INCIDENTES DE
LA UNIDAD DE SERVICIOS DE CÓMPUTO
ACADÉMICO (SAACI)

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

ANTONIO CARRIZOSA
MARTÍNEZ

DIRECTORA DE TESIS:

ING. MARÍA DEL ROSARIO BARRAGÁN PAZ



CIUDAD UNIVERSITARIA, MÉXICO D. F.

2010

Agradecimientos

A mi padre Antonio, gracias por todo tu apoyo y por la confianza brindada durante mis estudios. Por tu infinito esfuerzo al procurar que no faltara nada que limitara mis estudios, por tus sabios consejos y por guiarme en los momentos difíciles.

A mi madre Gregoria, gracias por tu esfuerzo, confianza y apoyo, que al igual que papá, siempre estuviste pendiente de que tuviera todos los recursos necesarios para que terminara mi carrera. Desde pequeño fuiste tú la que me mostró el camino, contigo estudiaba y resolvías mis dudas, sin embargo, al llegar el momento en que ya no podías ayudarme nunca te diste por vencida y buscaste los medios o la persona que pudiera hacerlo.

A los dos, a pesar de tal vez desconocer sobre mi carrera, siempre estuvieron ahí para escucharme, mostrando interés. Gracias por estar conmigo durante este largo camino, que al igual que yo, saben que fue difícil realizar este sueño. Ustedes son mi mayor ejemplo de que nunca debo darme por vencido por muy complicado que parezca el problema. Sin duda todo lo que vivimos en familia es la mayor razón por la que me propusiera ser un profesional para así ayudarlos como ustedes me ayudaron. Gracias por todo, a los dos los llevo presentes todos los días de mi vida, los quiero mucho mamá y papá.

A mis hermanas Lulú y Eli, por cuidarme desde pequeño, también ustedes me apoyaron hasta donde les fue posible, compartimos muy bonitos recuerdos y momentos muy difíciles que nos hicieron fuertes. De verdad que hubiera deseado ser su hermano mayor para haberlas cuidado y apoyado como ustedes lo hicieron conmigo, tal vez así muchas cosas hubieran sido diferentes para ustedes. Gracias por todo, las quiero mucho.

A mi tío Valentín Cruz, sin duda tu ejemplo siempre me inspiró, por tu calidad como persona y profesional, me mostraste como salir adelante sin perder el piso. Gracias por tus consejos durante la carrera, por todo el conocimiento que me transmitiste, por todo tu apoyo y sobre todo por acercarme a lo que hoy es mi profesión y hacerme entender lo que es estudiar ingeniería. Muchísimas gracias por apoyar a mis padres cuando más lo necesitaban y además por no sólo estar pendiente de mis estudios, sino de la carrera de todos mis primos. No me cansaré de decirlo, muchas gracias por todo tío.

A mi primo Guillermo Cruz Molina, por ser como mi hermano, gracias por compartir tantos momentos desde la niñez; por compartir gustos en común, formas de pensar y de ser; por recordarme que la familia siempre será la familia. Los dos nos hemos apoyado y aconsejado para salir adelante para ser lo que hoy somos. Sabes que cuentas conmigo y recuerda: *“Un amigo es alguien que te da total libertad de ser tú mismo”*.

A mis compañeros de carrera, mejor conocidos como los *CAMPANUDOS: Víctor, Zamora, Joe, Haro, Neza, Niño y Nakamura*, gracias a todos por los momentos que pasamos, el estudiar juntos, el apoyarnos como personas y profesionales y por todos los ratos de diversión. Formamos un gran equipo en todos los aspectos, sin ustedes no hubiera sido lo mismo.

En especial gracias *Rolando* que no sólo apoyaste a un servidor sino además me brindaste tu compañerismo y amistad. Tu ejemplo me demostró que siempre se tiene que buscar más, por lo mismo es que decidí entrar a UNICA. Amigo mil Gracias.

A UNICA, por abrirme sus puertas, por todo el apoyo y la capacitación recibida, por ser ese aspecto que llegara a complementar mi formación dentro de la Universidad. Gracias a todos los compañeros que conforman UNICA, a los instructores que en algún momento me dieron curso. A mis compañeros de la Generación 48: Leticia, Lidya, Dulce, Erick "el Bofo", Juan Carlos, Verónica, Javier, Iliana y Faraday, fueron una muestra clara de amistad, compañerismo y apoyo.

A la Ing. María del Rosario Barragán por apoyarme en la tesis, gracias por todo su tiempo, consejos, paciencia y observaciones que realizó para que pudiera cumplir este objetivo.

A la Ing. Beatriz Barrera Hernández por todo el apoyo recibido en UNICA, por haberme dado de su tiempo para revisar este trabajo y hacer observaciones, por sus consejos, por permitirme pertenecer al DSA y darme muchas oportunidades de crecimiento profesional.

A las dos muchas gracias por su confianza, amistad y afecto demostrado desde que ingresé a UNICA.

A la Facultad de Ingeniería y profesores que me dieron clases, gracias por el tiempo, la dedicación y los consejos que contribuyeron a mi formación personal y profesional, todo lo que aprendí lo aplico día a día en mi vida profesional.

A un amigo que brindó 13 años de su cariño y compañía a la familia sin pedir nada a cambio, este es un pequeño homenaje. Gracias nunca te olvidaremos.

A Carol Nayeli Jiménez Pérez y a ti pequeño Fernando, gracias por llegar a mi vida, por todo su apoyo durante este trabajo, por su inmensa paciencia conmigo y porque sacrificaron tiempo para ayudarme.

Fernando, espero que este trabajo sea un ejemplo para ti y una motivación para que algún día tú también te llenes de deseos para salir adelante, tal vez aún no comprendas del todo lo que digo, pero yo daré lo mejor de mí para que sea un buen ejemplo para ti. Sobre todo gracias por recordarme lo maravilloso que es la niñez y lo divertido que es pasar tiempo contigo.

Carol, por ser siempre tan exigente en como hago las cosas y sobre todo por ser como eres. Este trabajo lleva mucho de ti, al leerlo es una forma de recordar cuánto nos amamos, por lo mismo quiero que por favor sientas este éxito también como tuyo.

Gracias por compartir buenos y malos momentos, por superar problemas que parecían alejarnos pero que a pesar de todo nos han unido más. Muchas gracias por todo, TE AMO y así será para siempre.

A ti, ya que al leer este trabajo le das vida...

INTRODUCCIÓN	1
CAPÍTULO I. MARCO TEÓRICO CONTEXTUAL.....	5
I.1. UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO (UNICA)	5
I.1.1. Misión	5
I.1.2. Visión.....	5
I.1.2. Visión	5
I.1.3. Valores	5
I.1.4. Objetivos	6
I.1.5. Estructura de UNICA	6
I.2. SITUACIÓN ACTUAL	8
I.2.1. Proceso de Control de Incidentes.....	8
CAPÍTULO II. MARCO TEÓRICO CONCEPTUAL.....	11
II.1.CONCEPTO DE SISTEMA DE INFORMACIÓN	11
II.1.1. Elementos de un Sistema de Información.....	11
II.2.PROCESO UNIFICADO.....	12
II.2.1. Estructura del Proceso Unificado	13
II.2.2. Explicación de las disciplinas	15
II.2.3. Desarrollo Iterativo	16
II.2.4. Administración de Requerimientos	17
II.2.5. Arquitectura de componentes	18
II.3.LENGUAJE UNIFICADO DE MODELADO (UML)	19
II.3.1. Arquitectura 4+1.....	21
II.3.3. Diagramas de actividades	29
II.3.4. Diagramas de secuencia	32
II.3.5. Diagramas de Clases	33
II.3.6. Diagrama de componentes.....	40
II.3.7. Diagrama de despliegue.....	41
II.3.8. Elementos comunes a todos los diagramas.....	42
II.4.INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS	45
II.4.1. Diseño Orientado a Objetos.....	45
II.4.2. Clases	46
II.4.3. Objetos	46
II.4.4. Métodos,	46
II.4.5. Abstracción	48
II.4.6. Encapsulamiento	50
II.4.7. Herencia	51
II.4.8. Polimorfismo	52
II.4.9. Ventajas del modelo de objetos	53
II.5.HERRAMIENTAS DE DESARROLLO	54
II.5.1. Java	54
II.5.2. iBatis	58
II.5.3. Apache Struts	60
II.5.4. PostgreSQL.....	61
II.6.PATRONES DE DISEÑO	63
II.6.1. Modelo Vista Controlador - MVC.....	64
II.7.SEGURIDAD DE LA INFORMACIÓN.....	65
II.7.1. Integridad	66

II.7.2. Confidencialidad	66
II.7.3. Disponibilidad.....	66
II.7.4. Irrefutabilidad.....	67
II.7.5. Autenticación	67
CAPÍTULO III. DESARROLLO DEL PROYECTO	71
III.1. POSICIONAMIENTO DEL PRODUCTO.....	71
III.1.1. Oportunidad de Negocio	71
III.1.2. Problemática.....	71
III.1.3. Declaración de posición del producto	72
III.1.4. Perfiles de usuario	73
III.1.5. Especificación de requerimientos	74
III.1.6. Objetivos del proyecto	75
III.1.7. Metas	75
III.1.8. Alcances	75
III.1.8. Alcances	75
III.1.9. Restricciones	76
III.1.10. Recursos tecnológicos y humanos	77
III.1.11. Costo	78
III.2. ESPECIFICACIONES DE CASOS DE USO	79
III.3. DEFINICIÓN DE ACTORES	107
III.4. DIAGRAMAS DEL SISTEMA CON UML	109
III.4.1. Diagrama de casos de uso	109
III.4.2. Diagrama de actividades	109
III.4.3. Diagramas de secuencia	114
III.4.4. Diagrama de clases	120
III.4.5. Diagrama de componentes	124
III.5. DIAGRAMA LÓGICO DE DATOS	125
III.6. DISEÑO DEL SISTEMA	126
III.6.1. Diagrama físico de datos (DER)	126
III.6.2. Diccionario de datos para el DER	127
III.7. PRUEBAS DEL SISTEMA	136
III.7.1. Introducción	136
III.7.2. Pruebas de Caja Blanca	137
III.7.3. Pruebas de Caja Negra	143
III.7.4. Porcentaje de cobertura adecuado al realizar pruebas	152
CAPÍTULO IV. MANUALES DEL SISTEMA.....	155
IV.1. PRUEBAS DEL SISTEMA	157
CONCLUSIONES	193
GLOSARIO DE TÉRMINOS.....	197
FUENTES CONSULTADAS.....	201

INTRODUCCIÓN

La Unidad de Servicios de Cómputo Académico (UNICA) tiene como función principal proporcionar a la Facultad de Ingeniería servicios de cómputo y apoyo en actividades relacionadas de una manera eficiente.

Para llevar a cabo lo anterior, los departamentos que conforman la estructura organizacional de UNICA se encargan de atender los asuntos que les competen de acuerdo a las funciones que tienen asignadas.

El Departamento de Seguridad en Cómputo (DSC) se encarga de proteger de amenazas y riesgos la infraestructura de red y los recursos informáticos; así, cuando detecta algún problema informa a la institución involucrada por medio de un oficio. Además, como medio de control realiza un registro de los incidentes (virus informáticos, ataques a servidores, entre otros) que se presentan a fin de obtener estadísticas que ayuden a detectar más fácilmente los puntos débiles y a tomar decisiones al respecto.

Por otro lado, los Departamentos de Investigación y Desarrollo (DID), Redes y Operación de Servidores (DROS) y Servicios Académicos (DSA) entre otras actividades, son encargados de atender incidentes como son: soporte técnico, respaldos de información, actividades de índole administrativa, entre otros.

En todos los casos, el registro de la atención a dichos incidentes y la creación de estadísticas se realizan de forma manual en Hojas de cálculo Excel; así mismo, los oficios que expide el DSC hacia otras dependencias para informar sobre algún incidente y recomendar las medidas correspondientes para el mismo se realiza en Word mediante una plantilla que contiene la información general del oficio, al cual sólo se le adaptan los datos conforme al incidente y a la persona a quien va dirigido.

Precisamente, el objetivo de este proyecto es analizar, diseñar, desarrollar e implantar un software que permita sistematizar los procesos actuales de forma fácil y segura, reduciendo los tiempos y costos invertidos en el proceso de control de incidentes. De acuerdo a las necesidades de información de cada uno de los departamentos se desarrolló un sistema que se compone de dos módulos: un Módulo para el Departamento de Seguridad en Cómputo y un Módulo General para el resto de los Departamentos de UNICA.

A lo largo de los capítulos expuestos se presenta el detalle de la problemática y el desarrollo de la solución propuesta. De esta forma el proyecto se estructura de la siguiente manera:

Capítulo I. Marco teórico contextual. Este capítulo está dedicado a enlistar la razón de ser, las funciones y la situación actual de UNICA con la finalidad de poder conocer el entorno de la problemática y así, realizar la mejor propuesta que cubra las necesidades y permita cumplir las metas y objetivos de la institución.

Capítulo II. Marco teórico conceptual. El objetivo de este capítulo es recopilar toda la información teórica necesaria sobre las metodologías y el software de programación utilizados, para que sirvan como base y fundamento de su aplicación a lo largo del trabajo. De esta manera, se proporciona una visión global de la metodología del Proceso Unificado (RUP) y del análisis y diseño Orientado a Objetos; una descripción de las características de las herramientas que conforman a la plataforma de desarrollo (Java, Apache Struts, iBatis, PostgreSQL), finalizando con el tema de la seguridad de la información en el sistema.

Capítulo III. Desarrollo del Proyecto. Se presentan los productos de trabajo requeridos por la metodología utilizada en el proyecto, tales como: definición del problema, planteamiento de la solución, estudio del perfil de los usuarios, diagramas en UML, diccionario de datos, modelado de datos tanto lógico como físico y pruebas realizadas al sistema.

Capítulo IV. Manuales del Sistema. Como parte importante de la documentación de un producto de software, este capítulo se conforma por el manual de usuario.

Conclusiones y lecciones aprendidas. Se provee un resumen de la experiencia personal durante la elaboración del presente trabajo.

Glosario de Términos: Se presentan los significados de las palabras técnicas utilizadas a lo largo del proyecto.

CAPÍTULO I

Marco Teórico Contextual

CAPÍTULO I. MARCO TEÓRICO CONTEXTUAL

I.1. UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO (UNICA)

I.1.1. MISIÓN

Proporcionar eficaz y eficientemente, en el ámbito institucional, los servicios de cómputo y el apoyo en actividades relacionadas que coadyuven al proceso integral de formación académica de la Facultad de Ingeniería.

I.1.2. VISIÓN

La proyección de la Unidad de Servicios de Cómputo Académico al año 2010 es continuar siendo una Unidad líder en la prestación de servicios de cómputo de vanguardia a la Facultad de Ingeniería, al entorno universitario y a la sociedad en general.

- Contando con la organización, administración y los recursos adecuados.
- Líderes en la formación, capacitación y difusión de la cultura informática.
- Contando con la infraestructura de red de cómputo moderna y tecnología de punta, brindando servicios de calidad y alta disponibilidad en tecnologías de información y comunicación.
- Contando con los servicios y procesos de atención sistematizados y actuales en apoyo a los eventos de seguridad informática.
- Contando con la infraestructura adecuada y mecanismos para la actualización continúa del equipo de cómputo.

I.1.3. VALORES

El ambiente de trabajo de los integrantes de la UNICA se basa en un clima de cordialidad, respeto, honestidad, responsabilidad, ética y compromiso.

I.1.4. OBJETIVOS

- Proporcionar a nivel institucional, los servicios de apoyo en cómputo que los alumnos de la Facultad requieren para la realización y cumplimiento eficaz de sus tareas sustantivas.
- Formar recursos humanos de calidad.
- Ofrecer a la comunidad de la Facultad capacitación en lo relativo a tópicos de cómputo. Esta capacitación se divide en cursos de capacitación para el personal académico, cursos complementarios para alumnos y cursos de superación para el personal administrativo.
- Mantenerse a la vanguardia en todo lo relativo al cómputo a través de la investigación, incorporar a la actividad de la Facultad nuevas tendencias de cómputo.

- Apoyar a la Secretaría General en las actividades que involucren institucionalmente a la Facultad de Ingeniería.

I.1.5. ESTRUCTURA DE UNICA

La estructura organizacional de UNICA es la siguiente:

Jefatura de UNICA

- Departamento de Servicios Académicos (DSA).
 - Coordinación de Salas de Cómputo (CSC).
- Departamento de Investigación y Desarrollo (DID).
- Departamento de Redes y Operación de Servidores (DROS).
- Departamento de Seguridad en Cómputo (DSC).

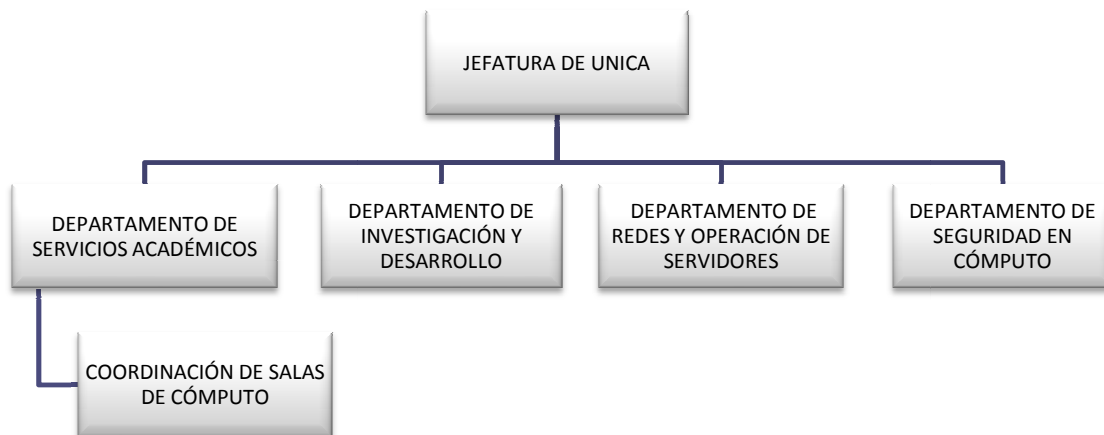


Figura 1.1 Organigrama de UNICA.

I.1.5.1. FUNCIONES DE LOS DEPARTAMENTOS

DEPARTAMENTO DE SERVICIOS ACADÉMICOS

Planear, organizar e impartir los cursos de cómputo que UNICA ofrece en sus modalidades: semestrales, intersemestrales, de fin de semana, especiales, externos e interno, así como los requeridos para la formación de los alumnos becarios de la Unidad.

Formar los recursos humanos que la Unidad necesita para el cumplimiento de sus funciones, mediante la coordinación del Programa de Formación de Becarios.

Proporcionar servicios de cómputo como: la plataforma educativa (EDUCAFI) y acceso a los alumnos a las salas de cómputo.

En promedio se imparten 120 cursos al año en los que se atienden alrededor de 760 alumnos, por otra parte, se capacitan aproximadamente 15 becarios al semestre.

DEPARTAMENTO DE INVESTIGACIÓN Y DESARROLLO

Estar al día en tecnología y avances en materia de cómputo, a efecto de estar en la posibilidad de incorporarla eficientemente a los servicios que la Unidad ofrece a sus alumnos, manteniéndoles oportunamente actualizados.

Desarrollar sistemas de información que apoyen a las actividades académico-administrativas de la Unidad y/o de la Facultad; y eventualmente a la solución de problemas específicos de clientes externos.

Administrar y dar mantenimiento al servicio de bases de datos que se ofrece a los alumnos.

Coordinación del área de multimedia.

Administración de la página Web de la Unidad.

DEPARTAMENTO DE REDES Y OPERACIÓN DE SERVIDORES

Administrar, operar, mantener y dar seguridad a la red de comunicación de la Facultad y a la intercomunicación con la red central de la UNAM; así como, desarrollar e implantar proyectos para la expansión del servicio.

Coordinar el Subcomité de Administradores y fungir como el área que administra la red de cómputo general ante DGSCA.

Dar soporte técnico a Secretarías y Divisiones de la FI.

Administrar y dar mantenimiento a las cuentas del servicio de correo electrónico y de Internet de la Facultad.

DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

Tiene como misión el fortalecer la seguridad informática de la institución salvaguardando la información electrónica contra daños, destrucción, uso no autorizado o robo; mantener la integridad de ésta, permitiendo que sea oportuna, precisa, confiable y completa; ayudar a alcanzar las metas de la institución y permitir el uso eficiente de los recursos tecnológicos destinados para el tratamiento de la información.

COORDINACIÓN DE SALAS DE CÓMPUTO

Proporcionar el servicio de cómputo (préstamo de equipo de cómputo con acceso a internet y paquetería de uso común) y de impresión a los alumnos de la Facultad para que estos puedan realizar sus tareas y trabajos de investigación.

Al efecto se cuenta con cuatro salas de atención a usuarios: una en el Conjunto Norte y tres en el Conjunto Sur, de estas últimas, dos se encuentran ubicadas en la División de Ciencias básicas y la otra en Posgrado.

En estas salas se cuenta con un total de 220 computadoras personales y 12 impresoras, así como equipo al servicio exclusivo de los alumnos, el cual se complementa con 9 servidores y computadoras necesarias para la administración y control del servicio.

Las salas prestan servicio en un horario de 9 de la mañana a 9 de la noche y se cuenta además con el servicio de asesoría especializada brindada por becarios instructores en línea y de forma presencial.

I.2. SITUACIÓN ACTUAL

I.2.1. PROCESO DE CONTROL DE INCIDENTES

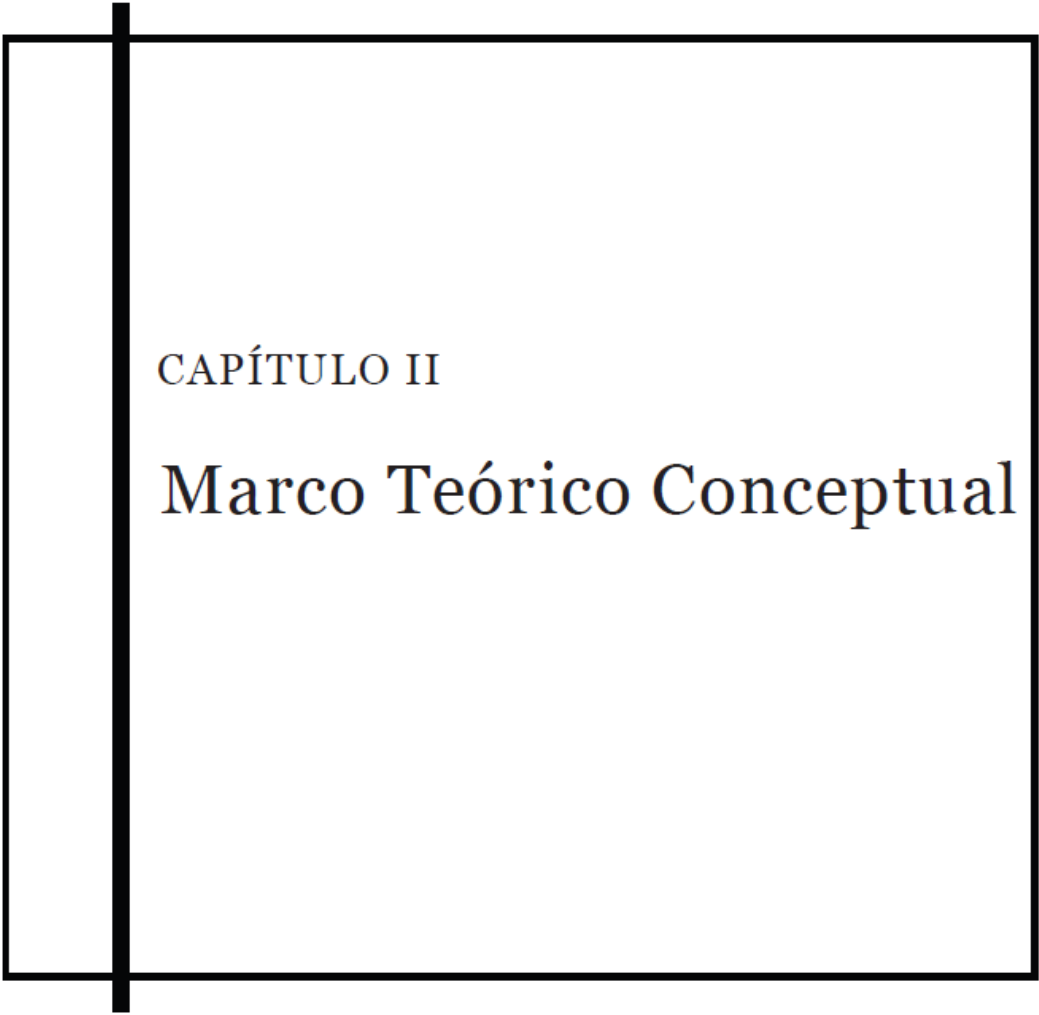
Los Jefes de cada uno de los Departamentos de UNICA llevan el control de los Incidentes que atiende el personal a su cargo a fin de conocer a detalle: las actividades que se realizan, el tiempo que tardan en dar solución y el status en que se encuentran los incidentes (atendidos, no atendidos). Dicho control les permite generar estadísticas para con ellas elegir caminos de acción y medir los índices de desempeño del sistema de gestión de calidad con el que cuenta UNICA.

Cabe mencionar que cada Departamento lleva el registro propio de los Incidentes que atiende de la siguiente manera:

En el Departamento de Seguridad en Cómputo (DSC), la captura de los datos de la atención y la generación de estadísticas de los incidentes se realiza en archivos de Excel. Asimismo, los oficios que expide el DSC hacia otras dependencias para informar sobre algún incidente y recomendar las medidas correspondientes para el mismo se efectúan en una plantilla de Word a la cual se le adaptan los datos conforme al incidente y a la persona a quien va dirigido.

En los Departamentos: de Investigación y Desarrollo (DID); de Servicios Académicos (DSA); y en el de Redes y Operación de Servidores (DROS), se lleva el control de las actividades del personal por medio del Sistema de Control y Evaluación de Becarios (SICEB), no obstante, dicho sistema no permite llevar un control detallado de las tareas que se realizan, como por ejemplo, el conocer el tiempo exacto que se lleva la atención de alguna labor. En algunos casos, estos datos se registran manualmente en archivos de Excel; situación que obliga a las áreas a invertir tiempo en actividades de control interno, lo que repercute en una mayor probabilidad de retrasos, omisiones y duplicidad de trabajo.

Por otra parte, la Jefatura de UNICA, principal interesado en conocer a detalle las actividades que se realizan en cada uno de los Departamentos, requiere reportes inmediatos de dichas actividades, sin embargo, con el control actual la información que se le proporciona es poco precisa y oportuna.



CAPÍTULO II

Marco Teórico Conceptual

CAPÍTULO II. MARCO TEÓRICO CONCEPTUAL

II.1. CONCEPTO DE SISTEMA DE INFORMACIÓN.

Actualmente cuando pensamos en sistemas de información, nace la imagen referente a computadoras, programas y tecnología; sin embargo, los sistemas de información existen desde el mismo día en que se creó la primera organización humana: “Estructura compuesta por un conjunto de personas distribuidas en departamentos o funciones con arreglo y con ciertos criterios de división de trabajo y coordinación.”¹

Antes de definir lo que es un sistema de información hay que aclarar que no es lo mismo datos que información. Es común confundir los datos con la información, pero realmente “los datos están constituidos por los registros de los hechos, acontecimientos, transacciones y más.”² Por otro lado, la información es el resultado de los datos procesados de tal forma que son útiles o significativos para el receptor de los mismos; por lo que podemos decir, que los datos son la materia prima para obtener información.

De lo anterior, podemos definir que un sistema de información es: “Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la entidad, recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha entidad y para las actividades de dirección y control correspondientes (decisiones) para desempeñar su actividad de acuerdo a su estrategia de negocio.”³

Los sistemas de información traen como beneficios la reducción de los costos y el tiempo en realizar las actividades; con esto se justifica la inversión en un sistema de información para sistematizar procesos.

II.1.1. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Los sistemas de información están integrados por 5 elementos que son: *los datos, los usuarios, los procedimientos, el hardware y el software.*

Independientemente de los elementos mencionados anteriormente, un sistema de información cuenta con funciones y un comportamiento. Las funciones son las acciones que realiza el sistema por petición de los usuarios. Y el comportamiento se observa en los sistemas de información que tienen interfaz gráfica, ya que el usuario decide mediante la interacción con el sistema el comportamiento que tomarán los datos a través de comandos o con un clic del mouse.

Los sistemas de información toman énfasis en los datos, ya que ese será el elemento clave para la funcionalidad de la entidad, si se cuenta con una información organizada será más fácil acceder a ella cuando se requiera.

¹ DE PABLO, I., *El reto informático. La gestión de la información en la empresa*, Editorial Pirámide, Madrid, 1989.

² FERRY Lucey, *Management Information Systems for Change*, DP Publicaciones, 1991.

³ ANDREU Rafael, Joan Ricart, et. al., *Estrategia y Sistemas de información*, Editorial Mc-Graw-Hill, Madrid, 1991.

II.2. PROCESO UNIFICADO

El Proceso Unificado de Rational (RUP) "es un proceso de desarrollo de software configurable que se adapta a través de los proyectos variados en tamaños y complejidad. Se basa en el uso de la tecnología orientada a objetos."⁴

El Proceso Unificado consiste en una serie de actividades con el fin de transformar los requerimientos en un sistema de software funcional.

Las características primordiales del Proceso Unificado son:

- *Iterativo e incremental*: Se dice que es iterativo porque se produce un artefacto en cada iteración; se revisa y se produce una segunda versión (Incremental) y así sucesivamente, lo que permite un equilibrio entre la arquitectura y los casos de uso durante todo el proceso de desarrollo.
- *Centrado en la arquitectura*: Lo que permite tener una abstracción del sistema desde un punto de vista en particular para tener una perspectiva de aquellas entidades que requieran en un determinado momento del proceso de desarrollo.
- *Basado en casos de uso*: Los casos de uso guían el diseño, la implementación y las pruebas. El que estén bien formulados garantizan el éxito del proyecto.

El Proceso Unificado de Rational se emplea para:

- Guiar la administración para el desarrollo en forma iterativa, para balancear los requerimientos, el tiempo y los riesgos del proyecto.
- Describir cada uno de los pasos involucrados en la captura de los requerimientos y en el establecimiento de una arquitectura para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.
- Describir en qué forma entregar el producto final, cómo desarrollarlo y proveer patrones. El proceso unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El Proceso Unificado de Rational se caracteriza por:

- Interacción continua con el usuario desde un inicio, para identificar los requerimientos y evitar malos entendidos, haciéndolo participe durante todo el proceso de desarrollo.
- Mitigación de riesgos antes de que ocurran ya que se tiene una planeación de todos los procesos y se contemplan posibles errores en cada versión de los artefactos a los que el sistema puede estar sujeto durante su operación.
- Liberaciones frecuentes, lo que significa la realización de artefactos cada vez más cercanos a cubrir los requerimientos.

⁴ RUMBAUGH, Jacobson Ivar and Booch Grady., *El Lenguaje Unificado de Modelado. Manual de Referencia*. Editorial Addison Wesley Rational Software Corporation. Año 2000. ISBN: 84-7829-037-0.

- Aseguramiento de la calidad en cada artefacto para cubrir las expectativas del proyecto.
- Involucrar al equipo en todas las decisiones del proyecto con lo cual se garantiza una plena actividad de los desarrolladores, lo cual permite un enriquecimiento en los artefactos y reduce los problemas de malos entendidos entre los miembros, gracias a que todos participan a lo largo del desarrollo.

II.2.1. ESTRUCTURA DEL PROCESO UNIFICADO

El proceso se maneja en dos dimensiones:

- *Eje Horizontal:* Representa el tiempo, muestra aspectos dinámicos del proceso, como ciclos, fases, iteraciones e hitos.
- *Eje Vertical:* Aspectos fijos del proceso, actividades, trabajadores y flujo de trabajo.

El Proceso Unificado de Rational (RUP) divide el proceso de desarrollo en ciclos teniendo como resultado final un producto, cada ciclo se divide en 4 fases, éstas a su vez se desarrollan en una o más iteraciones en donde se aplican todas las actividades de ingeniería, pero con diferente énfasis. Las fases son: inicio, elaboración, construcción y transición (Figura 2.1).

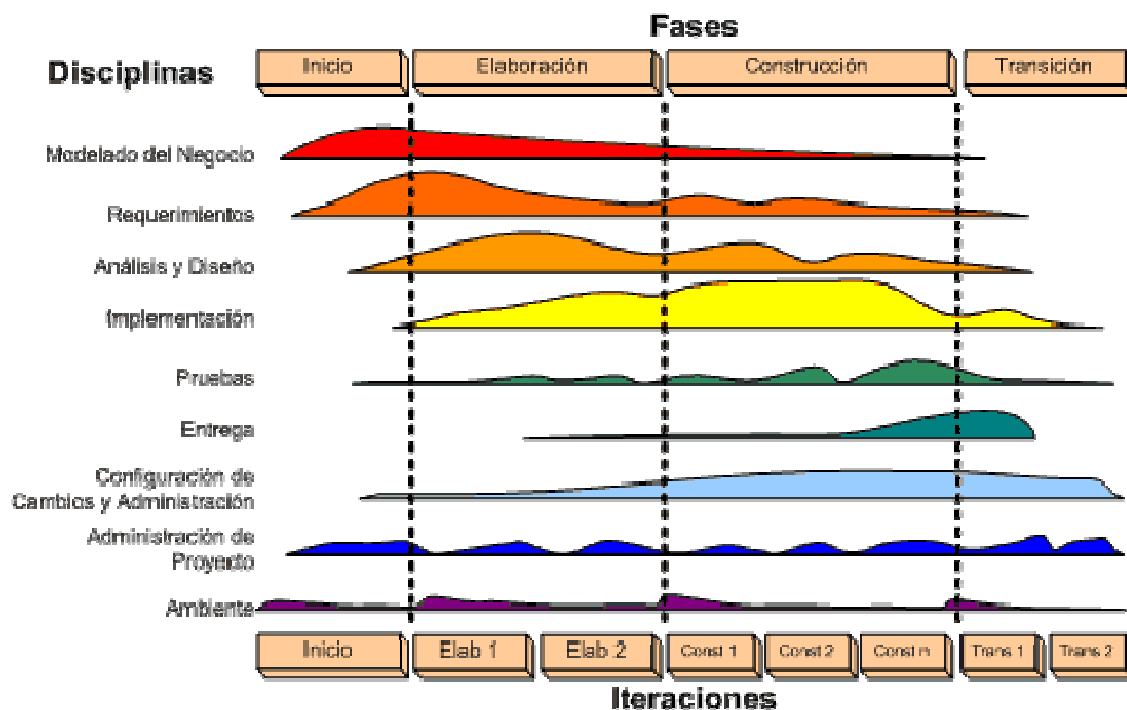


Figura 2.1 Fases del Proceso Unificado de Rational (RUP).

II.2.1.1. FASE DE INICIO

Esta fase tiene como propósito definir y acordar el alcance del proyecto, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

A partir del modelo de casos de uso y de la lista de riesgos, se puede determinar qué casos de uso deben implementarse primero para atacar los riesgos de mayor exposición. Con base en la información previa se realiza el proceso de planificación general y un plan de trabajo detallado para la siguiente fase, así como el plan para la siguiente iteración. El plan de pruebas debe planearse en esta fase, ejecutarse desde la primera iteración de la fase de elaboración y refinarse sucesivamente durante el ciclo de vida del proyecto.

II.2.1.2. FASE DE ELABORACIÓN

Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él. Al final de la fase se determina la viabilidad de continuar el proyecto y si se decide proseguir, dado que la mayor parte de los riesgos han sido mitigados, se escriben los planes de trabajo de las etapas de construcción y transición y se detalla el plan de trabajo de la primera iteración de la fase de construcción.

II.2.1.3. FASE DE CONSTRUCCIÓN

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto.

II.2.1.4. FASE DE TRANSICIÓN

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto al inicio del mismo.

II.2.2. EXPLICACIÓN DE LAS DISCIPLINAS

Las disciplinas proveen una vista dentro de los elementos del proceso fundamentales dentro de RUP desde la perspectiva de una disciplina general de habilidades.

Cada disciplina describe un conjunto de actividades asociadas y basada en artefactos (documentos). RUP describe disciplinas por niveles, un resumen de roles, artefactos y actividades y cada artefacto sirve en cada disciplina para el desarrollo de esta.

- **Modelo de negocio.** En esta disciplina se deben de entender los problemas actuales que enfrenta el negocio, así como la estructura de éste. Una vez entendido debemos de ir identificando los requisitos del sistema que se desarrollará para el negocio.
- **Requerimientos.** En esta disciplina se debe mantener un acuerdo entre los clientes y otros participantes sobre los alcances del sistema. Ya teniendo los requerimientos, estos deben de ser entendibles para los desarrolladores del sistema, se deben contemplar los costos y el tiempo de realización, así como definir interfaces que sean entendibles y de acuerdo al modelo de negocio para que el usuario se familiarice.
- **Análisis y Diseño.** En esta etapa es importante haber aterrizado los requerimientos del sistema para así transformar los requerimientos en un diseño del software a desarrollar. También se debe pensar en la arquitectura para el sistema y elaborar un buen diseño para que éste pueda adaptarse e implementarse a cualquier plataforma.
- **Implementación.** En esta disciplina el diseño se debe aplicar como archivos fuente, binarios, ejecutables y otros. Cada encargado de las unidades debe probar ésta y si se detecta algún error, debe notificarse.
- **Pruebas.** Esta disciplina se encarga de que el sistema a desarrollar sea de calidad. Al realizar las pruebas en cada unidad desarrollada del software se debe de documentar los defectos en la calidad de este. También se encarga de verificar que las funciones del sistema sean las diseñadas y que a su vez los requisitos estén bien implementados.
- **Despliegue.** Esta etapa asegura que el producto final esté disponible para ser usado por los usuarios finales. Incluye su distribución y soporte a los usuarios.
- **Configuración y control de cambios.** Esta disciplina revisa que los artefactos no sean modificados y asegura la integridad de éstos. También se encarga de verificar el proceso del sistema implementado.
- **Administración de proyectos.** Esta etapa administra los objetivos, riesgos y restricciones para el desarrollo del sistema y verifica que se esté desarrollando de acuerdo a los requerimientos del cliente y usuarios. Esta disciplina se basa en las mejores prácticas para que se lleve un buen desarrollo del sistema y así también visualizar los riesgos en cualquiera de las disciplinas y etapas.

- **Entorno o Ambiente.** Esta disciplina da soporte al proyecto a través de herramientas adecuadas para los procesos y métodos que conlleva el desarrollo del sistema. Las herramientas que se utilicen deben de ajustar al modelo de negocio.

II.2.3. DESARROLLO ITERATIVO

RUP describe cómo aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas.

RUP provee a cada miembro del equipo de las guías de proceso, plantillas y herramientas necesarias para que el equipo completo tome ventaja de las siguientes mejores prácticas (Figura 2.2):

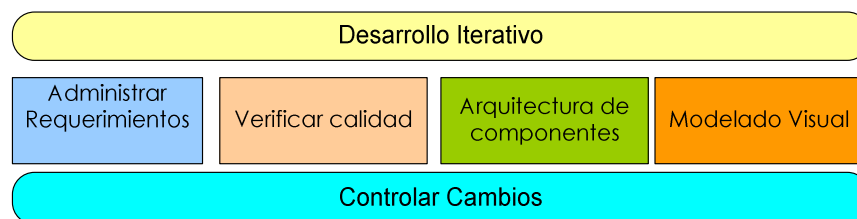


Figura 2.2. Mejores prácticas para el desarrollo de software.

RUP utiliza el modelo de espiral, que tiene como objetivo mejorar los tiempos de desarrollo y tener una correcta concepción del proyecto, la idea es partir produciendo una pequeña parte del sistema y una vez completada, se crea la segunda parte, que se integra a la primera, de manera de que en cada iteración, se obtiene una versión aumentada del sistema. El proceso termina cuando se considera que el sistema ha alcanzado un nivel de maduración que cumple los objetivos con los que fue realizado.

El modelo en espiral es actualmente el más indicado para el desarrollo de software ya que permite tomar en cuenta los riesgos posibles y contar con un plan de contingencia ante la presencia de alguno.

Como se muestra en la Figura 2.3 las fases de las que consta son:

- ✓ **Planificación:** Determinación de objetivos, alternativas y restricciones.
- ✓ **Análisis de riesgo:** Análisis de alternativas e identificación / resolución de riesgos.
- ✓ **Ingeniería:** Desarrollo del producto del "siguiente nivel".
- ✓ **Evaluación del cliente:** Valorización de los resultados de la ingeniería.

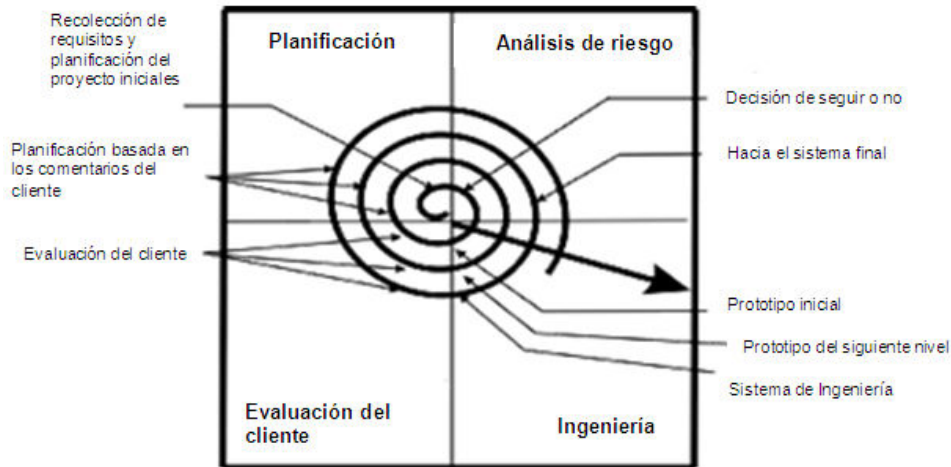


Figura 2.3 Metodología en espiral.

En la primera iteración se definen los objetivos, las alternativas y las restricciones; y se analizan e identifican los riesgos. En caso de que al hacer el análisis de riesgos se encuentre que los requerimientos no están bien identificados, se puede realizar un prototipo (ingeniería) para orientar tanto al desarrollador como al cliente.

El cliente evalúa el prototipo y sugiere las modificaciones pertinentes. Con base a las modificaciones realizadas y los comentarios del cliente se realiza la siguiente fase de planificación y de análisis de riesgo. En cada iteración debe existir una decisión de "seguir o no seguir".

Con cada iteración se construyen sucesivas versiones del software, cada vez más completas y, al final, el propio sistema operacional.

II.2.4. ADMINISTRACIÓN DE REQUERIMIENTOS

RUP proporciona una guía para obtener, organizar y documentar los requerimientos y las restricciones, cuenta con un documento llamado Visión, en el cual se describen los objetivos de la organización, las personas que intervienen con el sistema tanto directa como indirectamente, las necesidades de los usuarios, la descripción del producto y los alcances de éste.

Usa también el modelado de casos de uso en donde se representan las relaciones de los usuarios con los componentes del sistema. Todo esto con el fin de que el producto final cumpla en su totalidad con los objetivos para los que fue creado.

Los casos de uso permiten:

- Que las comunicaciones estén basadas en requerimientos claramente definidos.
- Que los requerimientos puedan ser priorizados, filtrados y monitoreados.
- Que sea posible realizar evaluaciones objetivas de funcionalidad y performance.
- Que las inconsistencias se detecten más fácilmente.

Los casos de uso y los escenarios utilizados en RUP han demostrado ser una manera excelente de capturar los requerimientos funcionales de guiar el diseño, la implementación y las pruebas del sistema, logrando así que el sistema satisfaga las necesidades del usuario.

II.2.5. ARQUITECTURA DE COMPONENTES

Este proceso se enfoca en el desarrollo basado en una arquitectura robusta que sea escalable. La creación de sistemas sofisticados de 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.

RUP provee un enfoque sistemático para definir una arquitectura utilizando componentes nuevos y preexistentes.

II.3. LENGUAJE UNIFICADO DE MODELADO (UML)

El Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) es considerado un lenguaje porque proporciona un vocabulario y reglas para utilizarlo; además es un lenguaje de modelado debido a que el vocabulario y reglas se utilizan para la representación conceptual y física para escribir planos del diseño de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software.

El esfuerzo para la definición de UML empezó cuando Grady Booch, James Rumbaugh e Ivar Jacobson diseñaron cada uno por separado su propia metodología para el análisis y diseño orientado a objetos, las cuales predominaban sobre las de sus competidores.

Para octubre de 1994, James Rumbaugh que creó la metodología OMT (Técnica de Modelado de Objetos) ingresó a Rational Software Corporation, donde ya trabajaba Grady Booch. Por su parte, Jacobson ingresó a Rational un año después con el objetivo principal de la unificación de los métodos de Booch y Rumbaugh.

Al final, UML es el fruto del trabajo de muchas personas y las ideas que hay en él provienen de un amplio espectro de trabajos anteriores como se aprecia en la Figura 2.4.

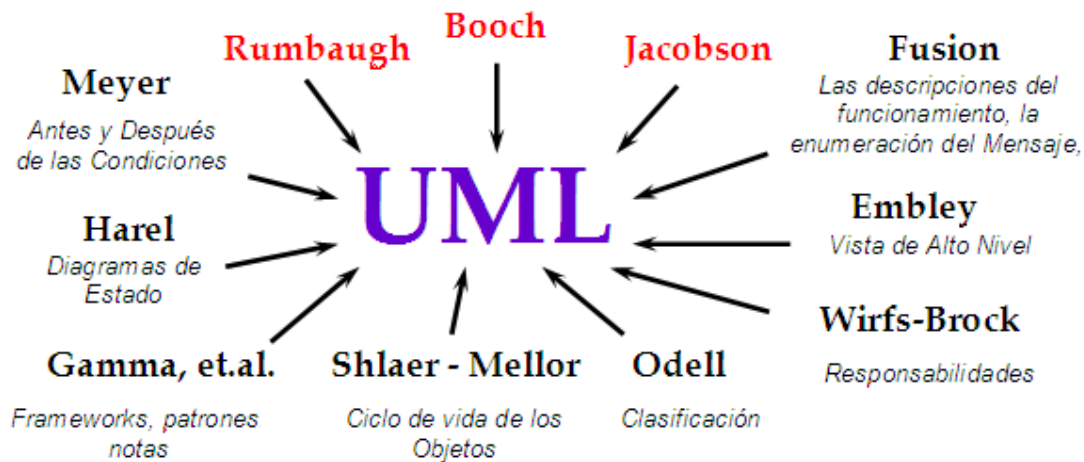


Figura 2.4 Creadores del UML.

Para 1997 se produjo la versión 1.0 del UML, poniéndolo a consideración del OMG (Grupo de Administración de Objetos responsable del cuidado y el establecimiento de los estándares de tecnologías orientadas a objetos) se propuso como un lenguaje de modelado estándar. Sin embargo, este no fue aceptado hasta que se generó la versión 1.1. para convertirse en un estándar en el modelado de software y con una evolución que continúa. Actualmente, la última versión de UML es la 2.0.

UML es:

- Un lenguaje que se centra en la representación conceptual y física de un sistema. Para que sea óptimo, es recomendable que esté dirigido por casos de uso, centrado en la arquitectura y que sea de forma iterativa e incremental.
- Es Unificado ya que integra lo mejor de varios autores, notaciones y técnicas.
- Facilita entender el modelado de los sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación entre el cliente y el analista de sistemas, lo cual es una gran ventaja durante el desarrollo, pues al ser un estándar, los modelos pueden ser interpretados por los clientes o por personas ajenas al diseño. Además de que permite detectar omisiones o inconsistencias durante el diseño.

UML es un lenguaje para visualizar

Un problema al que nos enfrentamos al modelar, es que nuestro modelo sea comprensible para todos. Al escribir modelos en UML se facilita la comunicación debido a que detrás de cada símbolo hay una semántica bien definida. De esta manera, un desarrollador puede escribir un modelo en UML y otro desarrollador puede interpretarlo sin problemas.

UML es un lenguaje para especificar

En particular UML construye modelos precisos, no ambiguos y complejos de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar un sistema.

UML es un lenguaje para construir

UML no es un lenguaje de programación visual, pero sus modelos pueden conectarse de forma directa a una gran cantidad de lenguajes de programación. Esto quiere decir que es posible establecer correspondencias desde un modelo UML a un lenguaje de programación como por ejemplo Java, o incluso a tablas en una base de datos.

UML es un lenguaje para documentar

Una organización de software que trabaja bien produce toda clase de información que es utilizada por un proceso de desarrollo de software o un sistema existente. Esto incluye:

- Requisitos
- Arquitectura
- Diseño
- Código fuente
- Planificación de proyectos
- Pruebas
- Prototipos
- Versiones

Así que UML cubre la documentación de la arquitectura de un sistema y todos sus detalles.

UML no está limitado al modelado de software, de hecho, es lo suficientemente expresivo para modelar sistemas que no son software, como flujos de trabajo en el sistema jurídico entre otros.

II.3.1. ARQUITECTURA 4+1

La visualización, especificación, construcción y documentación de un sistema con gran cantidad de software requiere que el sistema sea visto desde varias perspectivas. Cada usuario (analistas, desarrolladores, integradores de sistemas, encargados de las pruebas, encargados de la documentación técnica, jefes de proyecto y usuarios finales) ve al sistema de formas diferentes en diversos momentos a lo largo de la vida del proyecto.

Así es que la arquitectura de un sistema es quizás el artefacto más importante que puede emplearse para manejar estos diferentes puntos de vista y controlar el desarrollo iterativo e incremental de un sistema a lo largo de su ciclo de vida.

En general la arquitectura nos da una vista del conjunto de decisiones sobre:

- La organización del sistema de software.
- La selección de elementos estructurales y sus interfaces a través de los cuales se constituye el sistema.
- Su comportamiento, como se especifica en las colaboraciones entre esos elementos.
- La composición de estos elementos estructurales y de comportamiento en subsistemas progresivamente más grandes.
- El estilo arquitectónico que guía esta organización: los elementos estáticos y dinámicos y sus interfaces, sus colaboraciones y su composición.

Además, la arquitectura de software no tiene que ver solamente con la estructura y el comportamiento, sino también con el uso, la funcionalidad, el rendimiento, la capacidad de adaptación, la reutilización, la capacidad de ser comprendido, las restricciones económicas y tecnológicas y los compromisos entre alternativas así como los aspectos estéticos.

Para describir mejor la arquitectura de un sistema con gran cantidad de software, se propone el uso de cinco vistas interrelacionadas. Las vistas en UML consisten en un número de diagramas y todos esos diagramas juntos permiten obtener una perspectiva completa del sistema. En el siguiente diagrama (Figura 2.5) se describen cada una de las vistas que tiene UML:



Figura 2.5 En la arquitectura 4+1, cada vista es una proyección de la organización y de la estructura del sistema, centrada en un aspecto particular del mismo.

Vista de casos de uso

Describe la funcionalidad del sistema desde la perspectiva externa, tal y como es percibido por los usuarios finales. Con UML los aspectos estáticos se pueden concretar con los diagramas de caso de uso, mientras que los aspectos dinámicos con los diagramas de interacción (secuencia y colaboración), diagramas de estado y de actividades.

Vista Lógica

Describe la estructura interna del sistema. Comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y su solución. Esta vista soporta los requisitos funcionales y algunos no funcionales del sistema. Con UML, los aspectos estáticos de esta vista se pueden concretar con los diagramas de clases y de objetos; los aspectos dinámicos, con los diagramas de interacción (secuencia y colaboración), diagramas de estado y de actividades.

Vista de Componentes

Comprende los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. Esta vista se preocupa principalmente de la gestión de las distintas versiones de un sistema, a partir de componentes y archivos un tanto independientes y que pueden ensamblarse para producir un sistema en ejecución. Con UML, los aspectos estáticos de esta vista se pueden concretar con los diagramas de componentes; los aspectos dinámicos, con los diagramas de interacción (secuencia y colaboración), diagramas de estado y de actividades.

Vista de Procesos

Permite describir los procesos o algoritmos internos del sistema. También esta vista puede complementarse modelando los procesos de negocio en que operará el sistema. Con UML, los aspectos estáticos de esta vista se pueden concretar con los diagramas de clases y objetos; los aspectos dinámicos, con los diagramas de interacción (secuencia y colaboración), diagramas de estado y de actividades.

Vista de Distribución o despliegue

Contiene los nodos que forman la topología de hardware sobre la que se ejecuta el sistema. Se preocupa de la distribución, entrega e instalación de las partes que constituyen el sistema físico. Con UML, los aspectos estáticos de esta vista se pueden concretar con los diagramas de despliegue; los aspectos dinámicos con los diagramas de interacción (secuencia y colaboración).

II.3.2. DIAGRAMAS DE UML⁵

El UML está conformado por diversos diagramas gráficos y como UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de trabajar con diagramas es de representar diversas perspectivas de un sistema. Al modelar un sistema con UML se describe lo que hará el sistema, sin embargo, no dice como se implementará dicho sistema. A continuación se describen brevemente los diagramas más comunes del UML, cada diagrama es usado para un propósito específico.

Los diagramas de UML se dividen en tres grupos:

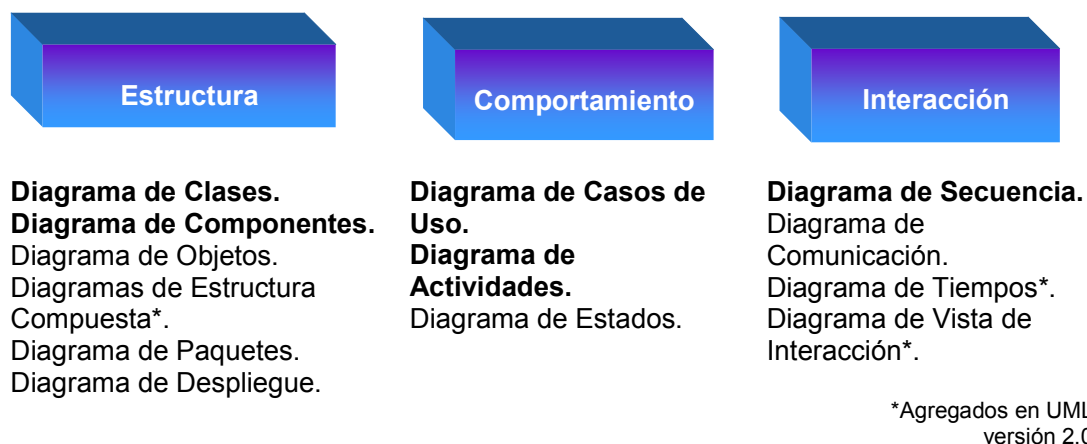


Figura 2.6 División de los Tipos de Diagramas en UML.

⁵ Los diagramas modelados en el presente trabajo, se resaltan con letras negritas en la Figura 2.6. Cabe mencionar que se utilizó una herramienta de software libre llamada StarUML para crear dichos diagramas debido a que se basa en los estándares de UML.

II.3.2.1. CASOS DE USO

Es una técnica utilizada para la captura del comportamiento deseado del sistema en desarrollo o de una parte de éste, sin la necesidad de especificar cómo se implementa dicho comportamiento.

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

En otras palabras, es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores⁶ en respuesta a un evento que inicia un actor principal sobre el propio sistema. Cabe mencionar que no son exclusivos de la metodología Orientado a Objetos, por lo que pueden ser utilizados en proyectos que sigan cualquier metodología de desarrollo.

Características de los casos de uso.

- Los casos de uso deben de llevar un nombre o en su caso describirse con el identificador que les corresponda, si es que los nombres son demasiados largos.
- Los nombres de los casos de uso deben ser únicos, comenzar con verbo y seguidos por un sustantivo.
- Los usuarios y analistas deben entender por igual los nombres de los casos de uso.
- Todos los casos deben de interactuar con un actor de forma directa o indirecta.
- Cada caso de uso representa una forma de usar el sistema.

II.3.2.2. DIAGRAMA DE CASOS DE USO

Es un diagrama que muestra un conjunto de casos de uso, actores y sus relaciones, con la finalidad de comprender y visualizar un sistema o subsistema, de forma que cualquier usuario entienda cómo utilizar ese elemento y permita a los desarrolladores implementarlo.

Notación

Gráficamente, los casos de uso se representan por una elipse, éstos se delimitan por un rectángulo que representa al sistema, mientras que los actores son monigotes o hombres de palo (*StickMan*) que se encuentran fuera del rectángulo, con un nombre debajo.

⁶ Actor: Toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad.

Las relaciones entre los casos de uso se dibujan dentro del rectángulo; y las relaciones entre los casos de uso y los actores también se deben especificar, como se muestra en la figura 2.7.

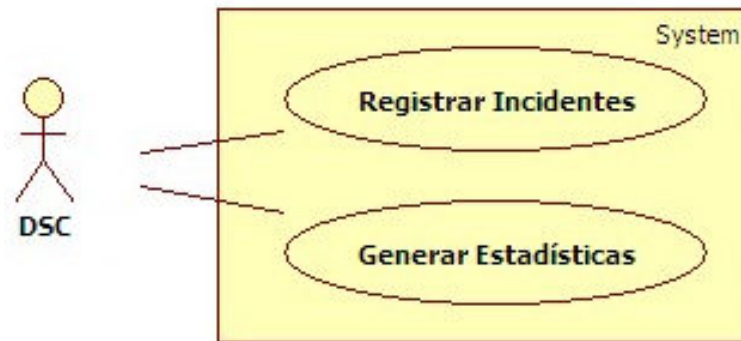


Figura 2.7 Elementos del Diagrama de Casos de Uso.

II.3.2.2.1. TIPOS DE RELACIONES

II.3.2.2.1.1. RELACIÓN <<INCLUDES>> (O <<USES>>)

Permite incluir la misma funcionalidad en dos o más casos de uso sin necesidad de repetir los detalles. El caso de uso incluido siempre forma parte de algún caso de uso base o principal mucho más amplio que lo incluye.

El caso de uso base conoce cuándo, dónde y por qué debe ejecutarse el caso de uso incluido. Cuando el caso de uso base ejecuta el comportamiento del caso de uso incluido, ejecuta todo el comportamiento descrito del incluido y luego continúa de acuerdo a su caso de uso original.

Gráficamente, la relación de inclusión se representa como una dependencia estereotipada, como se muestra en la figura 2.8:

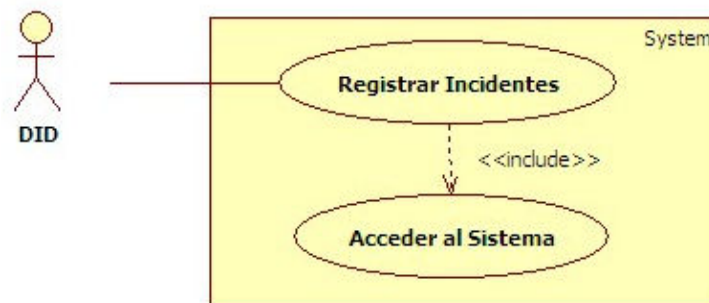


Figura 2.8 Ejemplo de cómo se representa una relación <<include>> en el Sistema. En este ejemplo, el caso de uso base "Registrar Incidentes" incluye el comportamiento descrito en el caso de uso "Acceder al Sistema". Esto significa que para poder registrar un incidente primero debe cumplir los pasos para poder acceder al Sistema.

II.3.2.2.1.2. RELACIÓN <<EXTENDS>>

Una relación de extensión entre casos de uso significa que un caso de uso base es similar a otro, pero que se extiende su funcionalidad y se describe una variación de la conducta normal.

Se utiliza para modelar la parte de un caso de uso que el usuario puede ver como comportamiento opcional dentro del sistema. También se puede modelar un subflujo separado que se ejecuta sólo bajo ciertas condiciones o para varios flujos que se pueden insertar en un punto dado controlados por la interacción con un actor.

Cuando se ejecuta el caso de uso base este sigue su curso, pero ante determinadas condiciones, su comportamiento se ve interrumpido con el del caso de uso que lo extiende.

El caso de uso extensión es el que conoce cuándo, dónde y por qué debe incorporar su comportamiento.

El caso de uso base no nombra al caso de uso que le extiende. El caso de uso extensión nombra al caso de uso base cuando se cumple la condición que lo hace ejecutarse.

La relación *extends* representa casos excepcionales o muy raramente invocados.

En algunas situaciones se puede crear nuevos casos de uso que extiendan a uno base, evitando así modificarlo directamente.

La idea principal de usar *extends* es con el propósito de no tocar requisitos ya finalizados. No obstante se recomienda crear casos de uso con *extends* cuando sea necesario, porque son más difíciles de comprender para la gente y también más difíciles de mantener.

La figura 2.8 representa una relación *extends*; donde los casos de uso “Recuperar Contraseña” y “Modificar Contraseña” son una extensión del caso de uso “Acceder al Sistema” debido a que se consideran como flujos alternos que se utilizan sólo cuando el usuario ha olvidado su contraseña o la desea modificar.

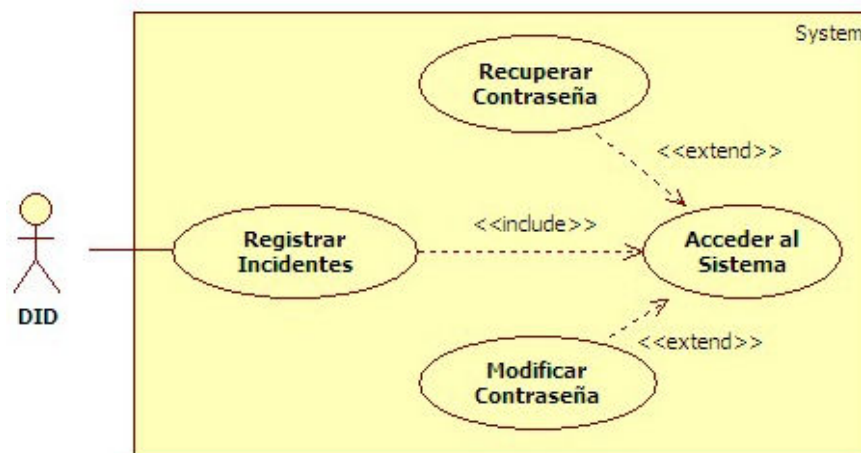


Figura 2.8 Ejemplo de una relación <<extends>>.

II.3.2.2.2. ESCENARIOS

Cada secuencia de acciones en particular, se le conoce como escenario de casos de uso. Un escenario es una instancia de un caso de uso que muestra una ruta con una combinación particular de condiciones.

Debido a que pueden existir cientos de escenarios en una aplicación, son útiles como casos de prueba ya que describen los datos de entrada y resultados esperados al probar un camino a través de un caso de uso.

Los escenarios se pueden clasificar en dos tipos:

1. Primarios o Flujo normal. Son los que describen lo que sucede al trabajar normalmente con el sistema.
2. Secundarios: Describen las excepciones del escenario primario o en los flujos alternos.

II.3.2.2.3. BENEFICIOS DE LOS DIAGRAMAS DE CASOS DE USO

Los diagramas de casos de uso se emplean para modelar la vista de casos de uso estática de un sistema. Esta vista cubre principalmente el comportamiento del sistema (los servicios visibles externamente que proporciona el sistema en el contexto de su entorno).

Cuando se modela la vista de casos de uso estática de un sistema, normalmente se emplearán los diagramas de casos de uso de una de las dos formas siguientes:

1. Para modelar el contexto de un sistema.

Modelar el contexto de un sistema implica dibujar una línea alrededor de todo el sistema y asegurar que actores quedan fuera del sistema e interactúan con él. Aquí, se emplearán los diagramas de casos de uso para especificar los actores y el significado de sus roles.

2. Para modelar los requisitos de un sistema.

El modelado de los requisitos de un sistema implica especificar qué debería hacer el sistema (desde un punto de vista externo), independientemente de cómo se haga. Aquí se emplearán los diagramas de casos de uso para especificar el comportamiento deseado del sistema. De esta forma, un diagrama de casos de uso permite ver el sistema entero como una caja negra; se puede ver qué hay fuera del sistema y cómo reacciona a los elementos externos, pero no se puede ver cómo funciona por dentro.

II.3.2.2.4. DOCUMENTACIÓN DE LOS CASOS DE USO

El comportamiento de un caso de uso se puede especificar describiendo un flujo de eventos de forma contextual lo suficientemente claro para que alguien ajeno al sistema lo entienda fácilmente. Al escribir el flujo de eventos se debe incluir cuándo y cómo empieza y acaba el caso de uso; cuándo interactúa con los actores, los flujos básicos y los flujos alternativos del comportamiento.

Al describir el flujo de eventos de un caso de uso es recomendable utilizar un lenguaje comprensible para el cliente y evitar tecnicismos.

Realmente, no existe alguna regla que indique qué información es la que debemos utilizar al documentar casos de uso, por lo tanto es posible definir el contenido de tal forma que se adapte a nuestras necesidades, siempre y cuando no sea muy detallada pero tampoco demasiado simple.

Para el desarrollo de este trabajo se contemplaron los elementos más importantes para la documentación de los casos de uso y se propone la siguiente plantilla para la documentación de los mismos.

Nombre	Nombre asignado al caso de uso.
Creado por	Nombre de la persona responsable de realizar el caso de uso.
Última actualización	Nombre de la persona que realiza la última actualización.
Fecha de creación	Fecha en que se realiza el caso de uso.
Fecha actualización	Fecha en que se realizó la última actualización al caso de uso.
Actores	Descripción de los sujetos involucrados en el caso de uso, así como de su participación e intereses.
Descripción	Breve explicación en pocas líneas de lo que hace el caso de uso y del objetivo que logra.
Precondiciones	Es una lista de actividades o condiciones que deben cumplirse para que el caso de uso pueda ser iniciado. Describe el estado que el sistema debe presentar antes de que el caso de uso sea ejecutado.
Flujo Principal y Alternativos	Secuencia de acciones o tareas que se dan en operación normal. Describe el flujo de eventos que ocurrirán en condiciones ideales, para que se obtenga algún resultado de valor para los actores.
Excepciones	Control de riesgos y errores por medio de variables controladas en el momento de interrupción del flujo principal o alterno.
Requisitos Especiales	Se describen las condiciones requeridas para que el caso de uso del sistema sea terminado.

II.3.3. DIAGRAMAS DE ACTIVIDADES

Los diagramas de actividades típicamente se utilizan para modelar los flujos de trabajo, operaciones internas y procesos de negocio. Muestran las actividades de un objeto y se representan con flechas que conectan a otros objetos.

A cada actividad se le representa por un rectángulo con las esquinas redondeadas. Cuando se procesa alguna actividad, continúa con la siguiente actividad. Las flechas representan la transición de una actividad a otra, como se muestra en la figura 2.9.



Figura 2.9 Cuando se completa la actividad 1, el flujo pasa inmediatamente a la actividad 2.

Este diagrama cuenta con un punto inicial (representado por un círculo relleno) y uno final que puede ser de dos tipos. El punto final de actividad se representa por una diana, mientras que el final de flujo se describe con un círculo con una cruz.



Final de Actividad Final de Flujo

.Figura 2.10 Diagramas para indicar el final de una actividad o de un flujo.

La diferencia entre los dos es que el final del flujo denota el final de un solo flujo de control, y el final de actividad denota el final de todos los flujos dentro de la actividad.

II.3.3.1. BIFURCACIÓN

Por lo general, la secuencia de actividades llega a situaciones donde tiene que realizar alguna decisión, donde una condición puede llevarla por un camino y otra por otro, aclarando que son mutuamente exclusivas (Figura 2.11).

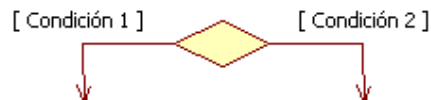


Figura 2.11 Diagrama para indicar una condición.

Las condiciones de guarda son los posibles resultados de una acción que sirven como condición para la realización de otra actividad. Esta condición se describe utilizando corchetes [].

II.3.3.2. DIVISIÓN Y UNIÓN

También es posible encontrar flujos concurrentes, especialmente cuando se modelan flujos de trabajo de procesos de negocio. En UML se utiliza una barra de sincronización para especificar la división y unión de los flujos de control paralelos. Gráficamente, una barra de sincronización se representa como una línea horizontal o vertical ancha.

Una división puede tener una transición de entrada y dos o más transiciones de salida, las cuales representan un flujo de control independiente. Después de la división, las actividades asociadas continúan su camino en paralelo (Figura 2.12).

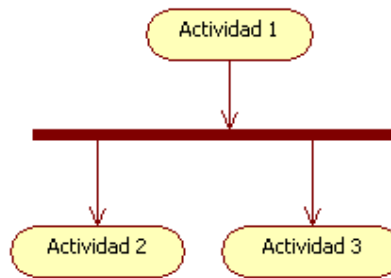


Figura 2.12 Diagrama para indicar una división de actividades en UML.

Una unión representa la sincronización de dos o más flujos de control concurrentes. La unión puede tener dos o más transiciones de entrada y una transición de salida. Antes de llegar a la unión, las actividades asociadas con cada uno de los caminos continúan en paralelo. Es decir, cada una de las actividades espera hasta que todos los flujos de entrada han alcanzado la unión y partir de ahí continúa con un único flujo de control que sale de dicha unión (Figura 2.13).

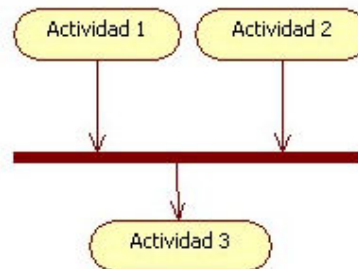


Figura 2.13 Diagrama para representar la sincronización de flujos en UML.

II.3.3.3. CARRILES

Los diagramas de actividades pueden dividirse en carriles, estos permiten determinar quién es responsable de las actividades que estén dentro de los carriles, cómo pasan las actividades y quién las hace. Gráficamente se representan con una línea vertical u horizontal, que divide el diagrama de actividades en zonas o grupos, dentro de cada zona se representa a la persona o sistema que participa en el proceso (Figura 2.14).

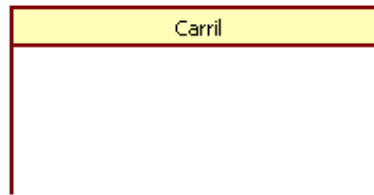


Figura 2.14 Diagrama para indicar un carril en UML.

Cada carril tiene un nombre único dentro del diagrama y cada actividad pertenece a una única calle, sin embargo, los flujos pueden cruzar los carriles.

Ejemplo: Cuando un usuario se autentica para ingresar en un sistema (Figura 2.15):

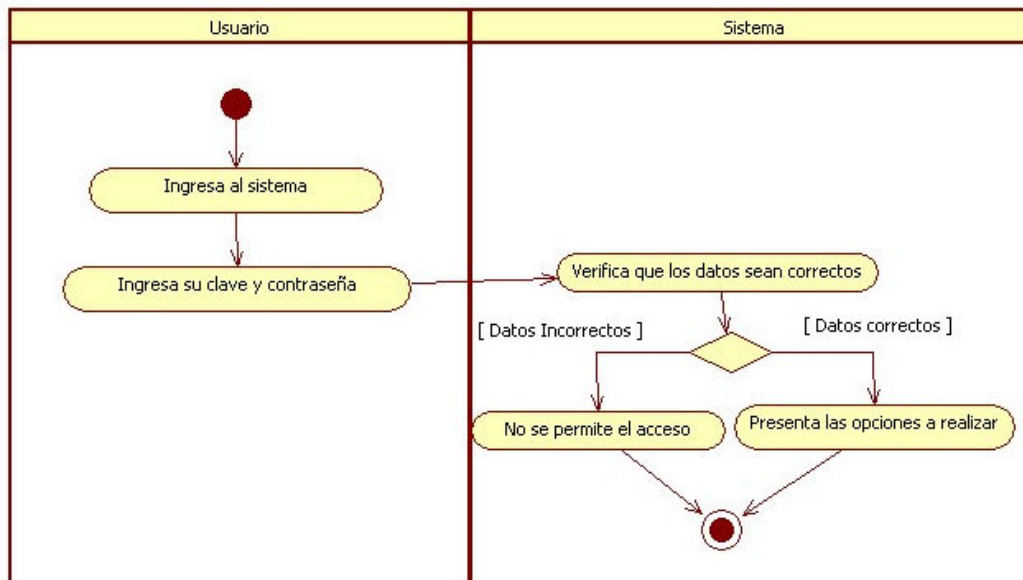


Figura 2.15 Ejemplo de un Diagrama de actividades.

Con el ejemplo anterior podemos decir que un diagrama de actividad es parecido a un diagrama de flujo; la diferencia entre estos es que los diagramas de actividad pueden representar actividades en paralelo.

Es buena práctica asociar los diagramas de actividades a los casos de usos detectados para esclarecer el comportamiento. Es posible hacer un diagrama de actividades por cada flujo de un caso de uso, pero basta con hacer el diagrama del flujo principal y en caso necesario, algunos flujos alternos.

Nota: No existe un número límite de carriles que se puedan utilizar, ya que estos pueden variar dependiendo del modelado, sin embargo, lo recomendable es utilizar los necesarios procurando no perder la claridad del modelo.

La gran ventaja de los diagramas de actividades reside en que manejan y promueven el comportamiento en paralelo, además de que estos no son propios de un análisis orientado a objetos. Se recomienda utilizarlos en los siguientes casos:

- Dentro del ciclo de desarrollo de software son usados en la etapa de requerimientos, ya que permiten tener la secuencia visual de las acciones dentro de un caso de uso.
- En aplicaciones multihilos o threads.
- Para la comprensión del flujo de trabajo a través de numerosos casos de uso.
- No se sugiere utilizarlos para comprender el comportamiento de los objetos o como colaboran entre sí los objetos.

II.3.4. DIAGRAMAS DE SECUENCIA

El diagrama de secuencia forma parte del modelado dinámico del sistema. En él se modelan las llamadas entre clases desde un punto concreto del sistema. Es útil para observar la vida de los objetos del sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

En el diagrama de secuencia se muestra el orden de las llamadas en el sistema y cómo los objetos se comunican entre sí; las actividades y acciones asociadas a los objetos al transcurrir determinado tiempo. Es imposible representar en un solo diagrama de secuencia todas las secuencias posibles del sistema, por ello es importante escoger un punto de partida.

Este diagrama consta de objetos que gráficamente se representan por rectángulos con un nombre (subrayado) con los mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

Los objetos se deben colocar en la parte superior del diagrama, a lo largo del eje X, de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. Debajo de cada objeto existirá una línea discontinua conocida como la línea de vida de un objeto (Figura 2.16):

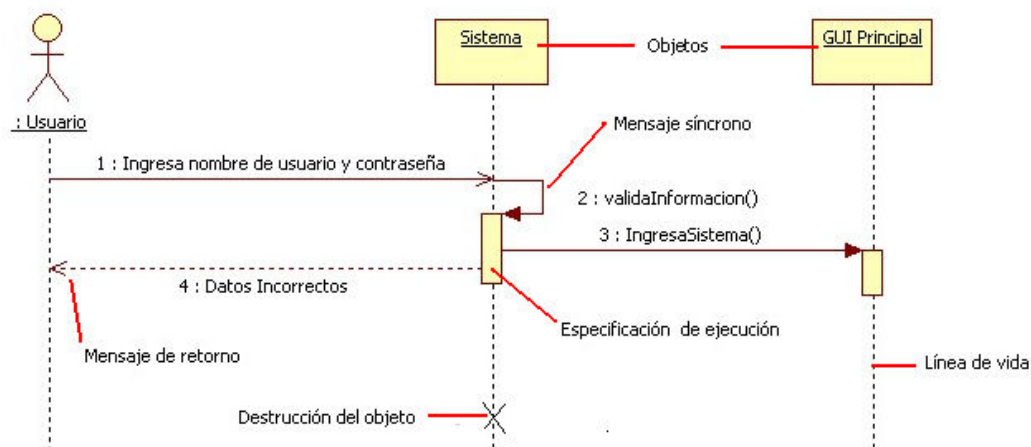


Figura 2.16 Ejemplo de un Diagrama de Secuencia.

Con la línea de vida de un objeto se encuentra un rectángulo conocido como activación, éste representa la ejecución de una operación que realiza el objeto. Su longitud representa la duración de la activación.

Cuando se termina de describir los mensajes, se cierra la línea de vida con una **X** indicando que ha llegado a su fin.

El orden del tiempo a lo largo de la línea de vida es significativo, por lo general, no importa la distancia exacta; simplemente muestran secuencias relativas, por lo tanto, no son diagramas del tiempo a escala.

Un mensaje que va de un objeto a otro pasa de la línea de vida de un objeto a la de otro. Además, un objeto puede enviarse un mensaje a sí mismo. Los mensajes pueden ser simples o de forma sincrónica o asincrónica.

Los mensajes simples son los que hacen la transferencia del control de un objeto a otro. En los mensajes sincrónicos los objetos esperan la respuesta al mensaje antes de que continúen con su trabajo. En cambio, si un objeto envía un mensaje asincrónico, los objetos no esperarán una respuesta antes de continuar.

II.3.5. DIAGRAMAS DE CLASES

Se utilizan para modelar la vista de diseño estática de un sistema. Esta vista soporta principalmente los requisitos funcionales de un sistema. Principalmente se utilizan para modelar:

El vocabulario del sistema: En este modelado implica tomar decisiones sobre qué abstracciones son parte del sistema y cuáles caen fuera de sus límites. Con los diagramas de clases se especifican estas abstracciones y sus responsabilidades.

Las colaboraciones simples: Una colaboración es un conjunto de clases, interfaces y otros elementos que colaboran entre sí, para proporcionar un comportamiento cooperativo mayor que la suma de todos los elementos. Con los diagramas de clases se visualiza y especifica este conjunto de clases y sus relaciones.

Modelar esquemas lógicos de bases de datos: Un esquema es un plano para el diseño conceptual de una base de datos. Mediante los diagramas de clases se puede modelar un esquema para una base de datos relacional o una base de datos orientada a objetos.

Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa o inversa.

Los diagramas de clases es un tipo especial de diagrama y comparte las propiedades comunes al resto de los diagramas (un nombre y un contenido gráfico que es una proyección de un modelo), gráficamente es una colección de nodos y arcos.

Normalmente contienen los siguientes elementos:

- Clases.
- Interfaces.
- Relaciones de dependencia, generalización, asociación, agregación y composición.

II.3.5.1. CLASES

Una *clase* es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica (Figura 2.17).

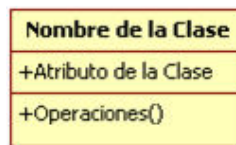


Figura 2.17 Diagrama de Clase en UML.

La notación en UML de un diagrama de clases es la siguiente:


Nombre: Cada clase debe tener un nombre que la distinga de otras clases, la clase puede dibujarse mostrando sólo su nombre. Para nombrar la clase se considera lo siguiente:


1. El nombre debe estar en singular y debe caracterizar de la mejor forma a la abstracción.
2. El nombre debe venir directamente del vocabulario del dominio.


Atributos: Un atributo es una abstracción de un tipo de dato o estado que puede incluir una instancia de una clase.

Operaciones: Una operación es una abstracción de algo que se puede hacer a un objeto y que es compartido por todos los objetos de la clase.


Los atributos y operaciones tienen *adornos*⁷ gráficos para indicar su visibilidad y grado de comunicación con el entorno. En UML se pueden especificar cuatro niveles de visibilidad:

 **Public:** Cualquier clase externa con visibilidad hacia la clase puede utilizar la característica.

 **Protected:** Cualquier descendiente de la clase puede utilizar la característica.

 **Private:** Sólo la propia clase puede utilizar la característica.

⁷ En UML un adorno es un elemento que hace visibles aspectos de la especificación de los elementos.

 **Package:** Sólo los clasificadores declarados en el mismo paquete pueden utilizar la característica.

Existen tres diferentes clasificaciones de clases:

Entity: Maneja la persistencia de los datos durante la vida del objeto y su comportamiento asociado. Este tipo de clase suele reflejar entidades del mundo real o elementos necesarios para realizar tareas internas al sistema. También se denominan clase dominio, ya que suelen tratar con abstracciones de entidades del mundo real.

Por ejemplo, en la Figura 2.18:

Se encuentra la clase **Personal**, la cual contiene la información necesaria para el registro del personal al que apoya o atiende UNICA, como es: el nombre, los apellidos y la responsabilidad (cargo).

El atributo *idPersonal* permite realizar tareas internas dentro del sistema, como el identificar de manera única un registro del resto de la información.

Además, la clase cuenta con los métodos de acceso (*get/set*) que permiten modificar y obtener la información.

Personal
-idPersonal: Integer -nombre: String -apellidos: String -responsable: String
+getIdPersonal(): Integer +setIdPersonal(idPersonal: Integer) +getNombre(): String +setNombre(nombre: String) +getApellidos(): String +setApellidos(apellidos: String) +getResponsable(): String +setResponsable(responsable: String)

Figura 2.18 Clase Personal

Boundary: Maneja comunicaciones entre el entorno del sistema y el sistema; suelen proporcionar la interfaz del sistema con un usuario o con otro sistema, en general, por tanto, modelan las interfaces del sistema. Cuando se trata de clases que definen la interfaz con otro sistema se refinarán, durante la fase de diseño, para tener en cuenta los protocolos de comunicación elegidos.

Por ejemplo, la Figura 2.19:

La clase **AltaDIDValidatorForm**, permite modelar la interfaz del sistema para el registro de los incidentes y apoyos que realiza UNICA.

Sin esta clase no sería posible mostrar en el sistema el listado de los diferentes tipos de apoyos y del personal que conforma UNICA.

Además no solamente deja mostrar información en el sistema, sino también se comunica con el mismo al momento de procesar la información para su validación o registro en la base de datos.

AltaDIDValidatorForm
-listaBecarios: List -apoyosList: List
<<create>>+AltaDIDValidatorForm() +getListBecarios(): List +setListaBecarios(listaBecarios: List) +getDidVO(): DID +setDidVO(didVO: DID) +getApoyosList(): List +setApoyosList(apoyosList: List) +getApoyoVO(): Apoyo +setApoyoVO(apoyoVO: Apoyo)

Figura 2.19 Clase AltaDIDValidatorForm.

Control: Maneja el comportamiento secuenciado específico de uno o varios casos de uso. Se trata de clases que coordinan los eventos necesarios para llevar a cabo el comportamiento que se especifica en el caso de uso, representan su dinámica.

Por ejemplo, la Figura 2.20:

Mediante la clase **ActionServlet**, se tiene el control que permite la comunicación entre la vista y el modelo del sistema, de tal forma que se ejecuten los flujos necesarios para realizar una actividad dentro del sistema, como por ejemplo: Registrar un oficio.

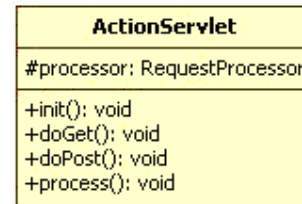


Figura 2.20 Clase ActionServlet

Las variantes de las clases son las clases abstractas, éstas a diferencia de una clase normal, no puede ser instanciada directamente pues posee métodos que no han sido definidos, es decir, en implementación. La única forma de utilizarlos es definiendo subclases, que implementan los métodos abstractos definidos.

II.3.5.2. INTERFACES

Una *interfaz* se refiere a una colección de operaciones que sirve para especificar un servicio de una clase o de un componente. Las interfaces permiten visualizar, construir, especificar y documentar las líneas de separación dentro de un sistema.

Gráficamente una interfaz se representa como una clase estereotipada o se dibuja como un pequeño círculo (Figura 2.21).

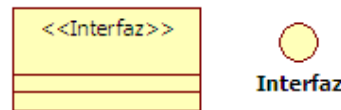


Figura 2.21 Diagrama de una interfaz en UML.

Muchos lenguajes de programación soportan el concepto de interfaces, incluyendo Java, pero las interfaces no sólo son importantes para separar la especificación y la implementación de una clase o un componente, además cuando se pasa a un sistema mucho más grande, se pueden usar para especificar la vista externa de un paquete o subsistema.

II.3.5.3. RELACIONES

Una relación es una conexión entre elementos que permite visualizar, especificar, construir y documentar redes de relaciones al nivel de detalle que se desee, incluso es necesario para soportar las ingenierías directa e inversa entre modelos y código.

La existencia de una relación denota una vía de comunicación (enlace) por la que se pueden enviar peticiones y mensajes entre ellas. Existen varios tipos de relaciones, las principales son:

- Dependencia.
- Asociación.
- Generalización.
- Agregación.
- Composición.

II.3.5.3.1. DEPENDENCIA

Es una relación “de uso” entre dos elementos, uno de ellos dependiente y el otro independiente. Lo cual indica que un cambio en la especificación de un elemento puede afectar a otro elemento que lo utiliza, pero no necesariamente a la inversa.

Gráficamente, una dependencia se representa como una línea discontinua, dirigida hacia el elemento del que se depende (Figura 2.22):

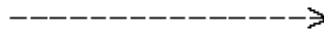


Figura 2.22 Línea para indicar dependencia.

Por ejemplo: Una relación de dependencia sería un capturista ya que hace uso de la computadora para poder trabajar (Figura 2.23).



Figura 2.23 Ejemplo de una relación de dependencia.

II.3.5.3.2. GENERALIZACIÓN

Es una relación entre un elemento general (llamado padre) y un tipo más específico de ese elemento (llamado hijo). El hijo puede añadir nueva estructura y comportamiento o modificar el comportamiento del padre. Las instancias del hijo pueden utilizarse donde quiera que se puedan usar las instancias del padre.

Un diagrama que tenga un padre utiliza la herencia simple, sin embargo, hay veces que se incorpora aspectos de varios objetos, este caso es conocido como herencia múltiple y permite modelar mejor las relaciones.

Gráficamente, se representa como una línea continua dirigida con una punta de flecha vacía apuntando al padre (Figura 2.24).



Figura 2.24 Línea para indicar generalización.

II.3.5.3.3. ASOCIACIÓN

Es una relación estructural que especifica que los objetos de un elemento están conectados con los objetos de otro. Las asociaciones representan que una instancia de A "tiene" instancias de B, además por cada instancia de A pueden existir "n" instancias del otro.

La asociación puede ser nombrada con algún *verbo* que describa de forma más precisa el tipo de relación, por ejemplo: vende, imparte, participa, realiza. Gráficamente una asociación se representa como una línea continua.

Por ejemplo, un músico puede participar en conciertos (Figura 2.25):

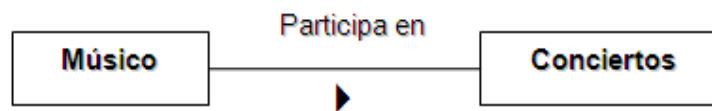


Figura 2.25 Ejemplo de una asociación.

II.3.5.3.4. MULTIPLICIDAD

La multiplicidad propiamente no es una relación, se utiliza para especificar en las relaciones el número de instancias de una clase con respecto a la otra.

La multiplicidad se escribe como una expresión con un valor mínimo y un valor máximo, que pueden ser iguales; se utilizan dos puntos consecutivos para separar ambos valores.

En la siguiente tabla se describen las diferentes opciones que se pueden utilizar para describir la multiplicidad en una relación:

Indica muchos	_____*
Indica exactamente uno	_____1
Indica cero o más	_____0..*
Indica uno o más	_____1..*
Indica cero o uno	_____0..1
Indica un rango específico	_____2..4

Por ejemplo: Cuando se indica una multiplicidad en un extremo de una asociación de clases, se está especificando cuántos objetos de la clase de ese extremo puede haber para cada objeto de la clase en el otro extremo (Figura 2.26).

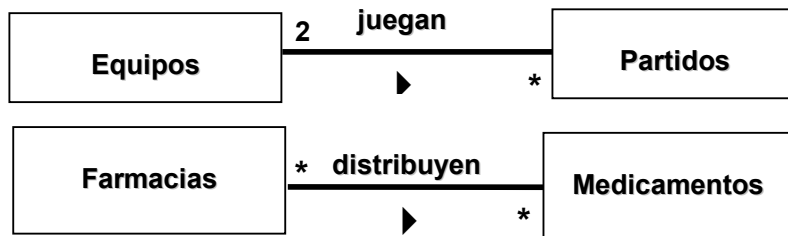


Figura 2.26 Dos equipos juegan muchos partidos y muchas farmacias distribuyen muchos medicamentos.

II.3.5.3.5. AGREGACIÓN

Como ya se mencionó, una asociación representa una relación estructural entre iguales, es decir, que ninguna clase es más importante que la otra. Sin embargo, si se desea modelar una relación “todo parte”, en la cual una clase representa todo y consta de elementos más pequeños (las partes), se les denomina agregación. Por lo tanto, es una relación donde una de las clases es parte de la otra. Si la clase contenedora es destruida, la clase contenida puede permanecer.

Gráficamente, se representa por una línea continua dirigida hacia el elemento contenedor con un rombo vacío (Figura 2.27):



Figura 2.27 “B es parte de A” y “A contiene B”.

Por ejemplo, en la figura 2.28; Un Sistema Solar **contiene** Planetas y un Estado **contiene** Municipios.

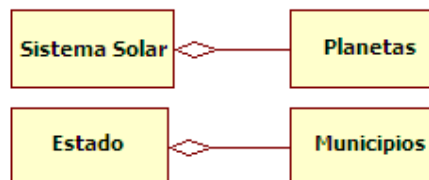


Figura 2.28 Relaciones de agregación.

II.3.5.3.6. COMPOSICIÓN

La composición es una forma de agregación con una fuerte relación de pertenencia y vidas coincidentes de la parte con el todo. Es decir, que el tiempo de vida de las partes depende directamente de la clase contenedora (Figura 2.29).



Figura 2.29 “B es parte de A” y “A contiene B”.

Por ejemplo, en la figura 2.30: El cerebro es parte del ser humano y un ser humano no puede existir sin cerebro.



Figura 2.30 Ejemplo de una composición.

II.3.6. DIAGRAMA DE COMPONENTES

Los diagramas de componentes, muestran una descripción de los elementos físicos del sistema y sus relaciones. Un diagrama de componentes representa las dependencias entre los componentes del Software los cuales pueden ser: archivos, paquetes, bibliotecas, módulos, código, etc. Esto nos permite observar la estructura e implementación del sistema.

Un componente es una parte conceptual y reemplazable de un sistema. Un componente es una unidad de código fuente que sirve como bloque constructor para la estructura física de un sistema. Gráficamente se representa como un rectángulo con pestañas (Figura 2.31).

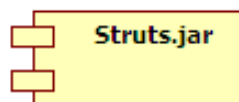


Figura 2.31 Diagrama de la librería de Struts como un Componente en UML.

Si el componente representa una librería o un ejecutable, se puede ilustrar su contenido en términos de subsistemas de implementación.

Si el componente representa un archivo, se puede indicar su contenido describiendo las clases de diseño que contiene (Figura 2.32).

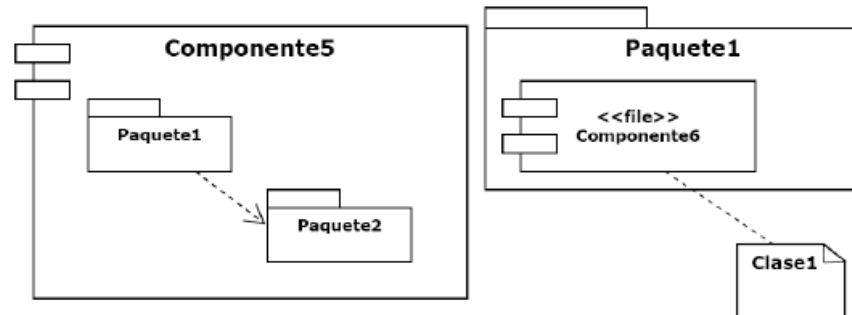


Figura 2.32 Diagrama de Componentes.

Básicamente existen tres tipos de componentes:

1. **Componentes de despliegue.** Son los componentes necesarios para formar un sistema ejecutable, tales como las bibliotecas dinámicas (DLL's y EXE's).
2. **Componentes producto del trabajo.** Son básicamente productos que quedan al final del proceso de desarrollo, y consisten en cosas tales como archivos de código fuente y archivos de datos a partir de los cuales se crean los componentes de despliegue.
3. **Componentes de ejecución.** Estos componentes se crean como consecuencia de un sistema en ejecución, tales que son una instancia a partir de una librería de clases.

II.3.7. DIAGRAMA DE DESPLIEGUE

Un diagrama de despliegue nos permite capturar la topología de los nodos del sistema y nos muestra los artefactos que residen en los nodos. Es decir, se sitúa el software en el hardware que lo contiene.

Un diagrama de despliegue o distribución muestra la ubicación de los componentes en nodos, de tal manera que se obtenga una vista de la distribución física del sistema en la organización.

Los elementos que contiene son:

- Nodos.
- Artefactos.
- Procesos.
- Relaciones de asociación.

Un nodo representa un recurso computacional (hardware) que generalmente, tiene memoria y a menudo capacidad de procesamiento (PC, Terminal, etc.).

Un nodo es un elemento donde se ejecutan los componentes. Gráficamente un nodo es un cubo que representa el despliegue físico de estos componentes (Figura 2.33).

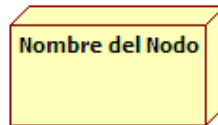


Figura 2.33 Representación de un Nodo en UML.

Un artefacto es una manifestación física de un componente, los artefactos se alojan dentro de los nodos. Gráficamente los artefactos son clases con un icono de documento o bien con el estereotipo <<artefact>> (Figura 2.34).

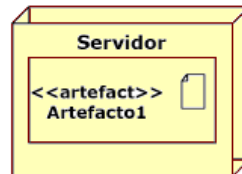


Figura 2.34 Diagrama de un artefacto en UML.

La asociación entre nodos es conocida como conexión. En este contexto la asociación representa una conexión física entre nodos, como puede ser una conexión Ethernet o un bus compartido (Figura 2.35).

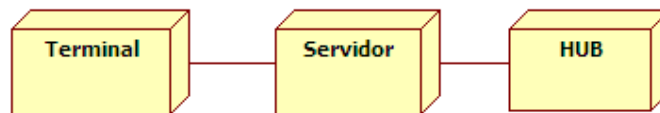


Figura 2.35 Diagrama de una conexión física entre nodos.

Los procesos deben de asignarse a un dispositivo de hardware para su ejecución. Un proceso es un hilo de control de la ejecución de un programa o sistema. Un sistema grande puede dividirse en procesos múltiples o hilos de control.

II.3.8. ELEMENTOS COMUNES A TODOS LOS DIAGRAMAS

UML proporciona una serie de mecanismos comunes que sirven para adaptar el lenguaje a necesidades particulares, pero dentro de un marco ordenado respetando las reglas.

Dentro de estos mecanismos están los adornos, las divisiones comunes y los mecanismos de extensibilidad. De esta manera UML es un lenguaje estándar "abierto-cerrado" con posibilidad de extenderse de manera controlada.

Los adornos son elementos que hacen visibles aspectos de la especificación del elemento, no obstante, hay que tener claro que no es necesario mostrarlo todo en un diagrama, es más importante que el diagrama sea claro (Figura 2.36):

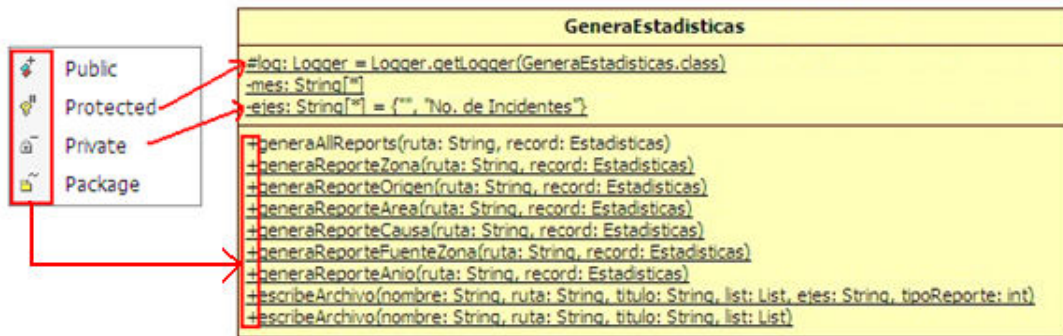


Figura 2.36 Clase “GeneraEstadisticas” con adornos que permiten indicar el nivel de visibilidad de los atributos y operaciones por medio de StarUML.

II.3.8.1. NOTAS

Son las partes explicativas de los modelos UML ya que son comentarios que se pueden aplicar para describir y hacer observaciones sobre cualquier elemento del modelo.

Gráficamente se representa como un rectángulo con una esquina doblada, junto con un comentario textual o gráfico como se muestra en la figura 2.37:

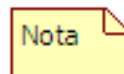


Figura 2.37 Nota en UML.

II.3.8.2. PAQUETES

Un paquete es el elemento de organización básica de un modelo de UML. Los diagramas y elementos pueden agruparse en un paquete y a la vez puede contener paquetes subordinados.

Gráficamente, un paquete se representa como una carpeta, que normalmente solo incluye su nombre y a veces su contenido (Figura 2.38).

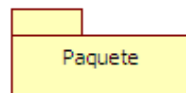


Figura 2.38 Paquete en UML

Existen dos divisiones comunes en UML:

- Clasificador/instancia.
- Interfaz/implementación.

El principio es separar *qué hace* algo (interfaz) de *cómo lo hace* (implementación). La interfaz define un contrato que garantiza seguir implementaciones específicas.

II.3.8.3. MECANISMOS DE EXTENSIBILIDAD

- Estereotipos.
- Valores etiquetados.
- Restricciones.

II.3.8.3.1. ESTEREOTIPOS

Amplía el vocabulario del UML, permitiendo crear nuevos tipos de bloques de construcción, derivados de los existentes y específicos para determinado problema. Gráficamente, se presentan con un nombre entre pico paréntesis y colocado sobre el nombre de otro elemento (Figura 2.39):

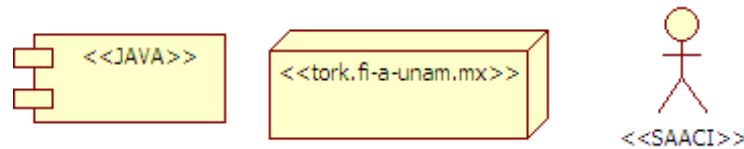


Figura 2.39 Estereotipos en UML.

Los estereotipos aplican a diversos elementos de UML como: Clases, Casos de Uso, Relaciones, Actores, etc.

II.3.8.3.2. VALORES ETIQUETADOS

Es una propiedad de un estereotipo que permite añadir nueva información en el elemento que lleva ese estereotipo. Gráficamente un valor etiquetado se representa como una cadena de caracteres de la forma: {etiqueta1 = valor1, etiqueta2 = valor2}.

II.3.8.3.3. RESTRICCIONES

Permite extender la semántica de un bloque de construcción de UML, permitiendo añadir nuevas reglas o modificar las existentes. Todas las restricciones irán siempre encerradas entre llaves.

II.4. INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

La manera de desarrollar software ha cambiado a lo largo de las décadas. La demanda por código reutilizable y el constante crecimiento de la exigencia a los programadores de software para mejorar los códigos han desembocado en un diseño de software mejorado y en técnicas de implementación que facilitan los actuales principios de ingeniería de software y sus requerimientos.

Podemos caracterizar el avance de la industria del software a través de tres “edades”:

- La edad orientada a procedimientos.
- La edad orientada a la estructura.
- La edad orientada a objetos.

En la programación orientada a procedimientos el código se escribe en cierto orden para resolver el problema, basándose en los conocimientos del proceso y de la programación. La desventaja de la aplicación resultante es que fuerza al usuario a seguir un camino predefinido desde un paso determinado a otro paso.

En la programación estructurada, el crecimiento de una aplicación hace que el mantenimiento de la misma se convierta en una tarea difícil, debido al gran número de procedimientos interrelacionados que podemos llegar a tener. El hecho de efectuar una pequeña modificación en un proceso, nos puede suponer el tener que recorrer un gran número de funciones del programa, no relacionadas por un nexo común.

La programación orientada a objetos (POO), se trata de una evolución de la programación estructurada basada en funciones, que permite agrupar elementos de código (rutinas y datos) con funcionalidades similares, bajo un sistema unificado de manipulación y acceso a dichos elementos.

II.4.1. DISEÑO ORIENTADO A OBJETOS

Es una estrategia de diseño en la cual los diseñadores del sistema piensan en términos de cosas reales en lugar de operaciones o funciones. El sistema se compone de objetos que interactúan entre ellos y que mantienen su propio estado y suministran operaciones de esa información. Ocultan la información de la representación del estado y después limitan su acceso.

Un proceso orientado a objetos comprende el diseño de clases y las relaciones entre éstas.

Para el diseño orientado a objetos se utilizan una serie de estrategias para el desarrollo:

- *El análisis orientado a objetos*: Al desarrollar el modelado de la aplicación, los objetos identificados reflejan las entidades y operaciones que se asocian con el problema a solucionar.
- *El diseño orientado a objetos*: Los objetos están relacionados con la solución del problema.

- *La programación orientada a objetos:* Se refiere a llevar a cabo el diseño de software utilizando un lenguaje de programación orientado a objetos, debido a que permite la implementación directa de los objetos y suministra recursos para definir las clases.

Existen una serie de principios fundamentales para comprender cómo se puede modelar la realidad al trabajar bajo el paradigma de la programación orientada a objetos. A continuación se hará una descripción general de los conceptos base de la POO.

II.4.2. CLASES

Una clase físicamente no es ni más ni menos que código. Cuando definimos una clase, realmente estamos definiendo dos cosas diferentes: los datos que puede manipular o contener y la forma de acceder a esos datos.

Por lo tanto, una clase es el conjunto de especificaciones o normas que definen cómo va a ser creado un objeto. Donde los atributos de una clase son las características externas y visibles de los objetos, por ejemplo, tamaño y color son atributos. Mientras que el comportamiento es el conjunto de acciones que un objeto de determinada clase puede realizar.

Por ejemplo, si partimos de cosas del mundo real, un Alumno puede ser considerado como clase. Por un lado tendremos los datos de dicho Alumno y por otro, el conjunto de acciones que puede realizar el Alumno: estudiar, reprobado, inscribirse, etc. En el primer caso, los datos del Alumno, el nombre, domicilio, edad, etc., son los que estarán representados por una serie de campos o atributos, mientras que la forma de modificar o acceder a esa información del Alumno se hará por medio de *métodos*. Así que los atributos o características y las acciones a realizar son las que definen a una clase.

El problema más frecuente al que nos enfrentamos al trabajar con este tipo de análisis es lograr identificar los atributos y características de la clase. Una técnica muy usada es realizar las siguientes preguntas para poder identificarlos:

1. Los atributos de la clase: Se identifican respondiendo a la pregunta: *¿Qué tiene?* .
2. Para identificar los métodos se responde a la pregunta: *¿Qué hace?*.

II.4.3. OBJETOS

Por un lado tenemos que una clase es la definición de los atributos, campos o propiedades y que la forma en que accederemos a ellos es por sus métodos. Para poder trabajar con la clase es necesario tener un objeto de dicha clase.

Un objeto es una entidad que tiene un estado y un conjunto de operaciones definidas que operan sobre ese estado. El estado se representa como un conjunto de atributos del objeto. Las operaciones que son asociadas con el objeto proveen servicios a otros objetos (clientes) que solicitan estos servicios cuando se requiere llevar a cabo algún cálculo.

La figura 2.40 muestra la representación común de los objetos:

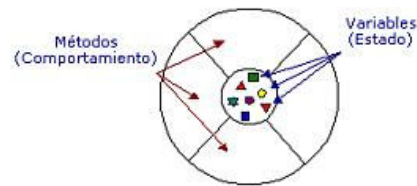


Figura 2.40 Vista lógica de un objeto.

Como se observa en la figura 2.40, un objeto puede visualizarse como una célula, de la cual hacemos una vista transversal y observamos que el núcleo es la parte privada (el estado del objeto) y lo que rodea al núcleo y lo protege del exterior es su parte pública (el comportamiento del objeto).

Las clases sirven como una *plantilla* para crear objetos, ya que incluyen las declaraciones de todos los atributos y las operaciones.

Para comprender la relación entre un objeto y su clase, pensemos en una Clase como un molde para hacer galletas, ya que el molde define las características de cada galleta, por ejemplo el tamaño y la forma.

La Clase se utiliza para crear Objetos, por lo tanto, los objetos son las galletas, y a pesar de que todas las galletas son hechas en el mismo molde, cada galleta va a tener características y atributos diferentes (Figura 2.41).



Figura 2.41 Una clase conceptualmente puede ser como un molde de galletas.

En conclusión, los objetos tienen las siguientes cualidades:

Identidad: Es la característica que distingue un objeto de otro.

Comportamiento: Los objetos proporcionan comportamiento y es la característica que hace útiles a los objetos. Por ejemplo, la mayor parte del tiempo podemos ignorar el funcionamiento interno de un coche y pensar en su comportamiento. El funcionamiento interno existe, pero normalmente es inaccesible. Lo que es accesible es el comportamiento de un objeto.

Estado: Los objetos tienen estado y es la característica que hace referencia al funcionamiento interno de un objeto y que le permite proporcionar el comportamiento que le define. Un objeto bien diseñado mantiene su estado inaccesible. No nos preocupa *cómo* realiza un objeto sus acciones, pero sí *que* las realice. Dos objetos podrían contener casualmente el mismo estado, pero seguirían siendo dos objetos distintos.

Finalmente, es posible resumir que una *clase* constituye la representación abstracta de algo, mientras que un *objeto* constituye la representación concreta de lo que una clase define. Al proceso por el cual obtenemos un objeto a partir de las especificaciones de una clase comúnmente se conoce como ***instanciar objetos***.

II.4.4. MÉTODOS

Es un mecanismo por medio del cual se comunican los objetos y normalmente son una solicitud de ejecución de algún comportamiento. Por lo tanto, es posible decir que un método es un mensaje que se le aplica a un objeto para que realice una determinada tarea.

Tras la recepción de un mensaje el objeto realizará una acción, ésta puede ser el envío de otros mensajes, el cambio de su estado, o la ejecución de cualquier otra tarea que se requiera que realice el objeto.

Esta comunicación puede comprenderse como la petición de un servicio. Para que los objetos trabajen en conjunto, un objeto A (*cliente*) envía a otro una llamada para realizar una operación y el objeto receptor B (*proveedor*) ejecutará la operación (Figura 2.42).

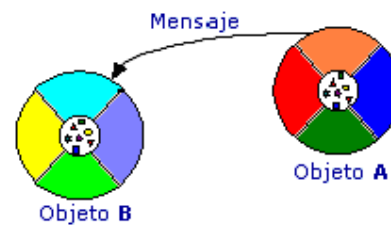


Figura 2.42 Mensajes entre objetos.

Un método se implementa dentro de la clase, Estos pueden ser subrutinas (que no tienen un valor de retorno) o funciones (que tienen un valor de retorno).

II.4.5. ABSTRACCIÓN

La abstracción es la capacidad de conceptualizar entidades genéricas de información a partir de cosas del mundo real, se enfatizan características comunes que interesan y se ignoran las características menos importantes.

Las propiedades de los objetos de una misma clase, pueden hacerlos tan distintos que sea difícil reconocer que pertenecen a una clase idéntica. No obstante, nosotros reconocemos a qué clase pertenecen, gracias a la abstracción.

Tomemos como ejemplo práctico un automóvil, como ya se mencionó, para identificar los atributos preguntamos: *¿Qué tiene el automóvil?* La respuesta es: color, marca, modelo, placas. Para identificar los métodos preguntamos: *¿Qué hace el automóvil?* Un automóvil puede hacer lo siguiente: arrancar, frenar, acelerar, transportar, contaminar.

Al realizar el proceso de abstracción del automóvil, obtenemos una lista con las características y comportamiento que tiene en común el automóvil. Esto no solo nos permite identificar los atributos y métodos, nos permite ver que aspectos no son importantes o relevantes para el modelado. Si hacemos una tabla con la información adquirida tenemos:

¿Qué tiene el automóvil?	¿Qué hace el automóvil?
Color	Arrancar
Volante	Frenar
Modelo	Acelerar
Placas	Transportar
Numero de llantas	Contaminar

En la tabla anterior se enumeran cada una de las características del automóvil, para este ejemplo, las características que están tachadas no son importantes, debido a que el análisis está enfocado a funciones *específicas* del automóvil: arrancar, frenar, acelerar.

Mientras que, de los atributos nos sirven como dato el número de llantas, como se enfoca únicamente en automóviles, es lógico que todos los autos tengan 4 llantas, por lo que es obsoleto este dato, también el volante, ya que todos los autos tienen un volante. De esta manera realizamos un filtro de la información que realmente nos interesa (Figura 2.43).



Figura 2.43 Representación del Proceso de Abstracción.

Del mismo modo que hacemos al identificar objetos reales, la abstracción nos ayuda a la hora de desarrollar una aplicación, permitiéndonos identificar los objetos que van a formar parte de nuestro programa, sin necesidad de disponer aún de su implementación; nos basta con reconocer los aspectos conceptuales que cada objeto debe resolver.

Debido a esto es posible dividir la abstracción en dos: abstracción funcional y abstracción de datos.

1. La *abstracción funcional* permite definir las operaciones que realiza un objeto, dichas operaciones son las que se relacionan con la operación general del sistema al que pertenecen. Pongamos como ejemplo el siguiente: De un reproductor de música, al enfocarse en la abstracción del funcionamiento encontramos: reproducir, avanzar, detener las melodías.

2. La *abstracción de datos* es similar a la abstracción funcional pero se relaciona con los tipos de datos. Es decir, nos permite ignorar los detalles de cómo es que los datos se representan internamente. Por ejemplo, en un lenguaje de programación de alto nivel, es posible representar a un número en términos de flotantes, dobles o enteros, cuando internamente se almacenan como números binarios.

Además de que la abstracción de datos involucra a la abstracción funcional, debido a que cuando se realizan operaciones sobre un dato abstracto la representación interna permanece oculta.

Por ejemplo, cuando abordamos el desarrollo de un programa de gestión orientado a objetos, realizamos una abstracción de los objetos que necesitaríamos para resolver los procesos del programa: un objeto Empleado, para gestionar al personal de la empresa; un objeto Factura, para gestionar las ventas realizadas de productos; un objeto Usuario, para verificar las personas que utilizan la aplicación, etc.

II.4.6. ENCAPSULAMIENTO

Es la característica de autonomía de la Orientación a Objetos, permite crear objetos con una funcionalidad y ocultando los detalles de la puesta en práctica de los aspectos no importantes para otros objetos.

Un objeto encapsulado puede verse como una caja negra a la cual se envían mensajes para que realice algo. Lo mismo pasa con muchos objetos de la vida cotidiana.

Un objeto encapsulado debe ofrecer una interfaz pública, es decir, un conjunto de comportamientos o métodos que permitan operar el objeto. Dicha interfaz pública lista los servicios proporcionados por el objeto.

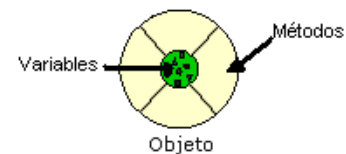


Figura 2.44 Vista de un objeto.

La interfaz actúa como un contrato con el mundo exterior que define lo que cualquier entidad externa puede hacer con el objeto (Figura 2.44).

Es posible mencionar los siguientes aspectos más importantes del encapsulamiento:

- Permite *cambios transparentes de implementación* para los demás objetos con los que se relacione en tanto no se altere la interfaz. Esto evita efectos secundarios indeseables en el resto del sistema.
- Así mismo, un objeto puede ser transferido alrededor del sistema sin alterar su estado y conducta.
- El encapsulamiento convierte a los objetos en componentes conectables.
- Promueve la división de la responsabilidad entre objetos del sistema.

Por lo tanto, los objetos proveen el beneficio de la modularidad y el ocultamiento de la información; mientras que las clases proveen el beneficio de la reutilización. Así que los

programadores de software utilizan la misma clase, y por lo tanto el mismo código una y otra vez para crear muchos objetos.

II.4.7. HERENCIA

La herencia es la característica más importante de la programación Orientada a Objetos, y es un mecanismo para definir una nueva clase en términos de una clase existente, de tal forma que se extienda su funcionalidad.

Gráficamente, la herencia suele representarse como un árbol. Descendiendo en el árbol se encuentran las clases especializadas (subclases o clases hijas) y ascendiendo por el árbol se encuentran clases generalizadas (clases base, superclase o padre). (Figura 2.45)

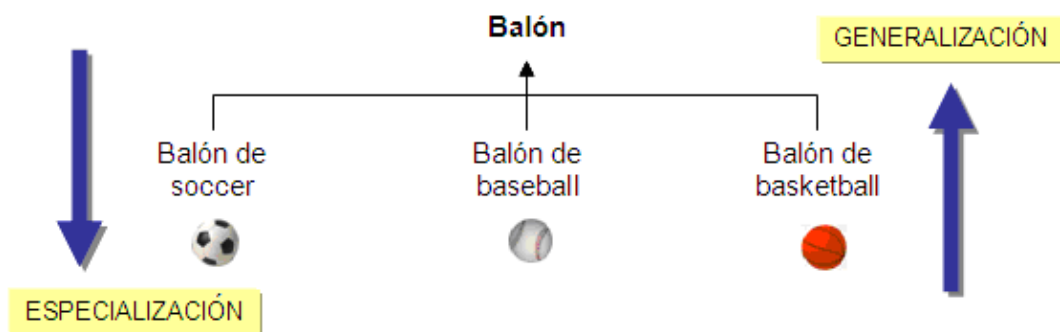


Figura 2.45 Ejemplo de una relación con Herencia.

En la Programación Orientada a Objetos, la herencia está totalmente ligada a la reutilización del código. Por lo tanto, podemos tomar una clase derivada y crear una nueva subclase a partir de ella, y así sucesivamente, componiendo lo que se denomina una jerarquía de clases. En una jerarquía de clases, las subclases tienen propiedades extras que la clase padre no tiene.

Por ejemplo en la Figura 2.46 se tiene lo siguiente:

- ✓ La clase A es la superclase de B y C.
- ✓ La clase C es la superclase de D.
- ✓ Las clases B y C son subclases de A.
- ✓ La clase D es una subclase de C.

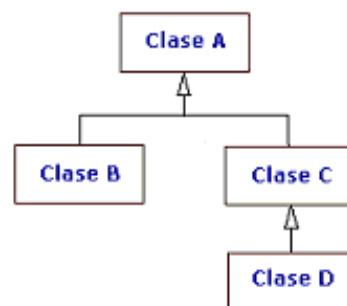


Figura 2.46 Jerarquía de Clases.

Existen dos tipos de herencia: Herencia Simple y Herencia Múltiple. La primera indica que se pueden definir nuevas clases a partir de una sola clase inicial, mientras que la segunda indica que se pueden definir nuevas clases a partir de dos o más clases iniciales. Java es un lenguaje que sólo soporta la herencia simple.

II.4.8. POLIMORFISMO

El polimorfismo significa tomar muchas formas. En términos de objetos significa que bajo el mismo nombre, un comportamiento o acción puede realizarse de diversas formas.

Por ejemplo, los objetos: puerta y ventana pueden presentar un comportamiento llamado “*abrir*” y que en cada caso se realiza de forma distinta. Entonces, el polimorfismo solamente funciona cuando una operación común tiene el mismo resultado semántico y cuando existe una relación de herencia.

La principal ventaja del polimorfismo es que alienta la reutilización de código.

El polimorfismo se puede clasificar en dos:

- **Polimorfismo dinámico** (o polimorfismo de inclusión): es aquél que hace posible tratar de forma genérica objetos relacionados que comparten el mismo comportamiento pero los implementan de forma distinta. En código se determina que un método pueda realizar diferentes tareas, a partir del objeto que damos como argumento, ya que una referencia de un tipo de clase hija puede considerarse o hacerse pasar por un tipo de clase padre.

Como ejemplo supongamos que del árbol de herencia de la clase “Balón” incluye un comportamiento que permite “lanzar” el balón. En cada subclase el comportamiento existe pero se implementa de forma distinta. También supongamos que existe una clase “jugador” y necesitamos que este “jugador” sea capaz de lanzar balones.

No es necesario crear un comportamiento para lanzar cada tipo de balón. Sólo un comportamiento que reciba un “Balón” y lo lance. Gracias a la herencia, es posible emplear la superclase como parámetro de este comportamiento y entonces aceptará cualquier objeto que esté en esa jerarquía de clases.

- **Polimorfismo estático** (o **polimorfismo *ad hoc***): Es el que permite utilizar el mismo comportamiento con diferentes parámetros. En cada comportamiento difiere el número y tipo de parámetros. Cuando se programa es mejor conocido como la *sobrecarga del método* y es útil cuando un comportamiento no está definido por sus argumentos, es decir, el comportamiento es independiente.

Por ejemplo, un comportamiento para obtener la división entre dos números:

division (entero a, entero b)

division (flotante a, flotante b)

division (doble a, doble b)

division (doble a, doble b, doble c)

Sin la sobrecarga sería necesario dar un nombre distinto a cada comportamiento. De esta manera se nos permite tener múltiples maneras de hacer algo.

II.4.9. VENTAJAS DEL MODELO DE OBJETOS

Ayuda a explotar la potencia expresiva de los lenguajes de programación basados en objetos y orientados a objetos. Aprovechando por ejemplo las jerarquías de clases en el proceso de diseño.

Promueve la reutilización de software, de diseños enteros conduciendo a la creación de marcos de desarrollo de aplicaciones reutilizables. Los sistemas orientados a objetos suelen ser más pequeños lo que significa escribir y mantener menos código, lo que se traduce en beneficios de costo y planificación.

El modelo de objetos produce sistemas que se construyen sobre formas intermedias estables, que son más flexibles al cambio. Tales sistemas evolucionan en el tiempo en lugar de ser abandonados en cuanto se produce el primer cambio en los requerimientos.

Reduce los riesgos inherentes al desarrollo de sistemas complejos pues distribuye la integración a lo largo del ciclo de vida en vez de suceder como un evento principal. La separación inteligente de intereses reduce los riesgos de desarrollo e incrementa la confianza en la corrección del diseño.

II.5. HERRAMIENTAS DE DESARROLLO

II.5.1. JAVA

Los inicios de Java se remontan a finales de los años 80's aunque no es hasta 1990 cuando el equipo de James Gosling (Jefe de equipo de desarrolladores de SUN MICROSYSTEMS) en su afán de crear un lenguaje tal que los programas desarrollados con él pudieran ejecutarse sobre cualquier arquitectura, desarrollan el **Oak**. Un lenguaje pequeño en cuanto a tamaño, fiable y sobre todo independiente de la plataforma, muy parecido a C y C++.

A Oak se le cambió el nombre por Java. Se dice que decidieron ponerle ese nombre mientras tomaban café (Java es también el nombre de un tipo de café, originario del este de Asia, de la isla del mismo nombre), aunque hay algunos que afirman que el nombre deriva de las siglas de **James Gosling**, **Arthur Van Hoff**, y **Andy Bechtolsheim**.

Debido a la importancia que tomó Java para SUN, desarrolló un navegador utilizando el lenguaje Java, al cual nombró HotJava, dicho navegador permitía trabajar con Applets Java (pequeños programas escritos en Java ejecutables/interpretables desde un navegador), además de que podía ser ejecutado sobre cualquier computadora independientemente del sistema operativo que tenga.

Es así como nace el primer paquete de desarrollo que, con la experiencia y práctica de programadores, se fue depurando poco a poco. El programa evolucionó y dio lugar al primer kit completo de programación: Java Developers Kit ver. 1.0, que después mejoró hasta la actual versión 1.6 ya disponible en internet.

Es tal el desarrollo esperado de Java que se está estudiando crear chips que ejecuten los bytecodes Java (código java) por hardware para mejorar los tiempos de respuesta, que por software dejan todavía mucho que desear. Actualmente IBM y SUN trabajan en un proyecto para desarrollar microprocesadores que ejecuten Java directamente.

II.5.1.2. VERSIONES DE JAVA

Java 1.0. (Conformado por 8 paquetes y 212 clases)

Primera versión pública. El diseño del lenguaje no es demasiado bueno y hay demasiados errores. Respecto a seguridad, es restrictivo por defecto, no dejando hacer demasiado al código no fiable.

Java 1.1 (Conformado por 23 paquetes y 504 clases)

Mejoras de rendimiento en la Máquina Virtual, nuevo modelo de eventos en AWT, clases anidadas, serialización de objetos, API de JavaBeans, archivos jar, internacionalización, API Reflection (Reflexión), JDBC (Java Data base Connectivity), RMI (Remote Method Invocation). Es la primera versión lo suficientemente estable y robusta.

Java 1.2 (Conformado por 59 paquetes y 1520 clases)

Se producen notables mejoras a todos los niveles, como es Swing, Java2D, Corba, API Collections. Para enfatizar ésto, Sun lo renombra como "Java 2". El JDK (Java Development Kit) se renombra como SDK (Software Development Kit). Se divide en J2SE, J2EE y J2ME.

Java 1.3 (Conformado por 77 paquetes y 1595 clases)

Orientada sobre todo a la resolución de errores y a la mejora del rendimiento; se producen algunos cambios menores como la inclusión de JNDI (Java Naming and Directory Interface) y la API Java Sound. También incluye un nuevo compilador de alto rendimiento JIT (Just In Time).

Java 1.4 (Conformado por 103 paquetes, 2175 clases)

Mejora notablemente el rendimiento y añade entre otros soporte de expresiones regulares, una nueva API de entrada/salida de bajo nivel (NIO, New I/O), clases para el trabajo con Collections, procesado de XML, mejoras de seguridad como el soporte para la criptografía mediante las Java Cryptography Extension (JCE), la inclusión de la Java Secure Socket Extension (JSSE) y el Java Authentication and Authorization Service (JAAS).

Java 1.5 (Conformado por 131 paquetes y 2656 clases)

También conocido como Tiger, renombrado por motivos de marketing como Java 5.0. Incluye como principales novedades: tipos genéricos (generics), autoboxing/unboxing (conversiones implícitas entre tipos primitivos y las clases que contienen un tipo primitivo como atributo, mejor conocidas como *Wrappers*), Enumerados; Bucles simplificados; la impresión de texto con formato "printf"; Funciones con número de parámetros variable; Metadatos en clases y métodos.

Java 1.6

También conocido como Mustang. En esta versión, Sun cambió el nombre "J2SE" por Java SE. Los cambios más importantes introducidos en esta versión son: Un nuevo marco de trabajo y APIs que hacen posible la combinación de Java con lenguajes dinámicos como PHP, Python, Ruby y JavaScript. Incluye el motor Rhino de Mozilla, una implementación de Javascript en Java. Incluye un cliente completo de Servicios Web y soporta las últimas especificaciones para Servicios Web, como JAX-WS 2.0, JAXB 2.0. Se implementaron otras ideas, como mejoras en la interfaz gráfica y en el rendimiento.

II.5.1.3. CARACTERÍSTICAS DEL LENGUAJE JAVA

Los siguientes puntos muestran las principales características del lenguaje de programación Java.

Orientado a objetos. Java es un lenguaje totalmente orientado a objetos: encapsulación, herencia, polimorfismo, etc. No soporta herencia múltiple, sin embargo, incorpora la implementación de interfaces. Además, cuenta con un amplio conjunto de bibliotecas estándar.

Portable. Java es un lenguaje multiplataforma, los programas escritos se compilan en *bytecodes* los cuales son independientes del sistema operativo, por lo tanto no requieren ser pre-compilados o modificados, siempre y cuando exista una máquina virtual de Java.

Seguro. El código generado por Java pasa por diversas pruebas, debido a que el código debe cumplir con las reglas de acceso y seguridad establecidas, ya que los programas en Java no acceden directamente al hardware de la máquina. Al cargar un programa en memoria, la máquina virtual de Java verifica los *bytecodes* de la aplicación.

Sencillo. Java simplifica algunos aspectos a la hora de programar, por ejemplo en el manejo de la memoria, la máquina virtual gestiona automáticamente y libera la memoria por medio del recolector de basura (*Garbage Collector*), reduciendo los errores de programación generados por la mala administración de la memoria. Además, simplifica la sintaxis con respecto al lenguaje C++ al eliminar el uso de apuntadores.

Distribuido. Java proporciona las herramientas y librerías necesarias para la ejecución de sistemas distribuidos mediante Java RMI, JNDI. Permitiendo acceder a la información a través de la red.

Dinámico. Java cuenta con un amplio conjunto de bibliotecas estándar, para trabajar con colecciones, archivos, acceso a bases de datos (JDBC), interfaces gráficas de usuario (awt/swing), redes, aplicaciones distribuidas (EJB), interfaces Web (Servlets/JSP), compresión de datos, criptografía, de forma dinámica debido a que la carga de las clases se realiza cuando estas son requeridas.

II.5.1.4. LA MÁQUINA VIRTUAL DE JAVA (JVM)

La Máquina Virtual Java es el núcleo del lenguaje de programación Java. Es la responsable de la interpretación y ejecución de los programas y es la clave de muchas de las características principales de Java, como la portabilidad, la eficiencia y la seguridad.

El compilador es el encargado de convertir el código fuente de un programa en un código intermedio llamado *bytecode* que es independiente de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java (Figura 2.47).

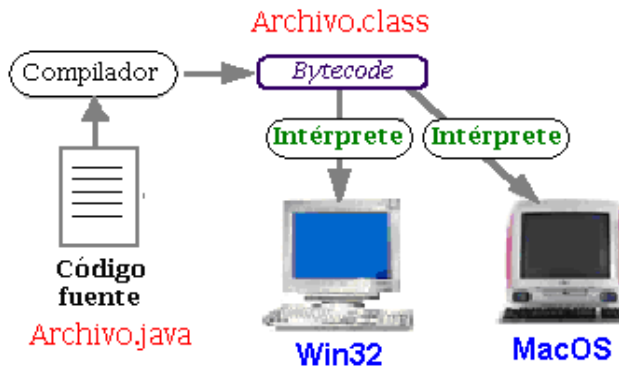


Figura 2.47 Ejecución de un Programa con Java.

Cuando se ejecuta un programa en Java, las instrucciones no son ejecutadas directamente por el hardware de la máquina, sino que son pasadas a un elemento de software intermedio que es el encargado de que dichas instrucciones sean ejecutadas por el hardware.

Precisamente Java fue diseñado pensando en que se implementaría sobre una amplia gama de sistemas operativos y de procesadores, así que se incluyeron dos capas de software para aumentar su portabilidad.

La primera dependiente de la plataforma conocida como *adaptador*, mientras que la segunda, que es independiente de la plataforma, se le llama *interfaz de portabilidad*. De esta forma, la única parte que se tiene que escribir para una plataforma nueva, es el adaptador.

El sistema operativo proporciona los servicios de manejo de ventanas, red, sistema de archivos, etcétera (Figura 2.48).

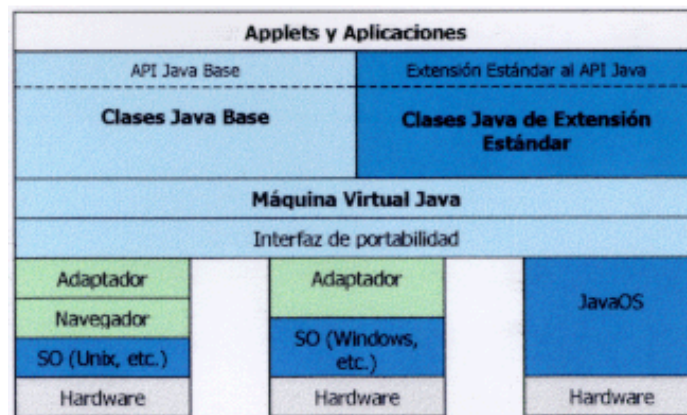


Figura 2.48 Capas de una Aplicación realizada con Java.

II.5.2. IBATIS

iBatis⁸ es un framework de código abierto basado en capas, desarrollado por Apache, que se ocupa de la capa de persistencia, dicha capa se encuentra entre la capa de negocio y la de base de datos. Simplifica la implementación del patrón de diseño Direct Access Objects (DAO) y la persistencia de objetos en bases de datos relacionales.

Con iBatis se pueden utilizar modelos de datos poco normalizados, por lo tanto no es completamente transparente, debido a que asocia objetos de modelo (JavaBeans) con sentencias SQL o procedimientos almacenados, esto mediante archivos XML, sin embargo, simplifica la utilización de la base de datos (Figura 2.49).

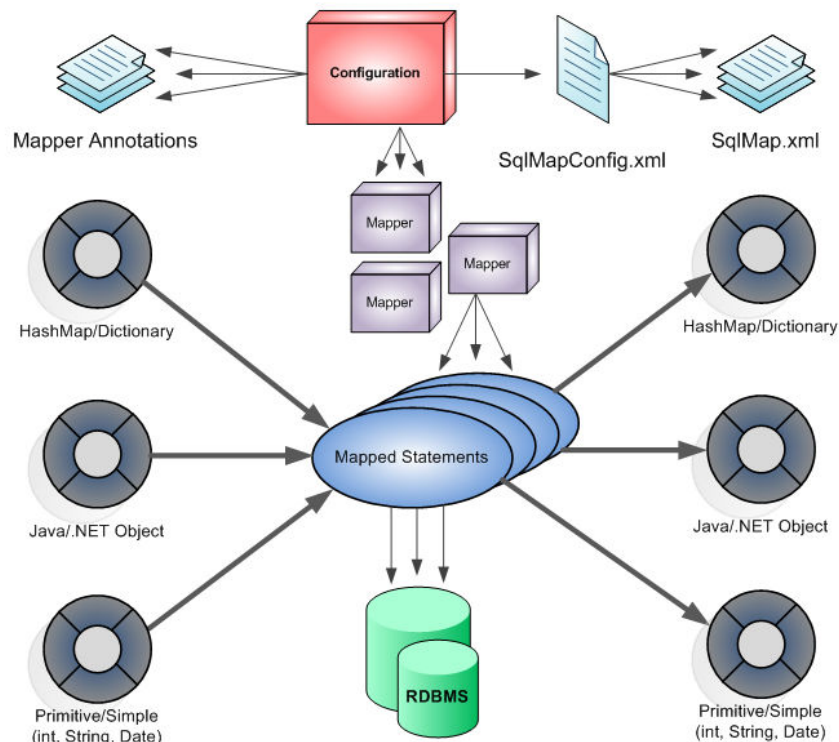


Figura 2.49 Estructura del funcionamiento de iBatis.

II.5.2.1. CARACTERÍSTICAS DE IBATIS

La capa de Persistencia se puede subdividir en tres capas:

- La capa de abstracción es la interfaz con la capa de la lógica de negocio, haciendo las veces de “fachada” entre la aplicación y la persistencia. Se implementa de forma general mediante el patrón Data Access Object (DAO) y particularmente se implementa utilizando su framework DAO (ibatis-dao.jar).

⁸ El 21 de Mayo de 2010, se anunció que iBatis pasó a formar parte del proyecto Google Code bajo el nombre de MyBatis. Consultado en: **ANNOUNCEMENT**, Clinton Begin, 21 de mayo de 2010. Disponible desde internet en: <http://ibatis.apache.org/> [Último acceso: 5 de junio de 2010].

- La capa de persistencia es la interfaz con la base de datos que se ocupa de la gestión de los datos mediante un API. iBatis utiliza su framework SQL-MAP (ibatis-sqlmap.jar). Además cada objeto del modelo, que representa al objeto en la aplicación, se relaciona con un archivo del tipo sqlMap.xml, que contiene sus sentencias sql.
- La capa de driver se ocupa de la comunicación con la propia base de datos utilizando un driver específico para la misma.

Toda implementación de iBATIS incluye los siguientes componentes:

- Data Mapper: Proporciona una forma sencilla de interacción de datos entre los objetos Java y bases de datos relacionales.
- Data Access Object: Abstracción que oculta la persistencia de objetos en la aplicación y proporciona un API de acceso a datos al resto de la aplicación.

iBatis implementa el patrón DAO. Este patrón ayuda a centralizar la responsabilidad del acceso a datos en un único punto, y oculta al resto de la aplicación la implementación específica. Gracias a este patrón, una posible migración del Sistema Gestor de Base de Datos sólo afectaría a la capa de persistencia, resultando transparente para el resto de las capas.

II.5.2.2. VENTAJAS E INCONVENIENTES DEL USO DE IBATIS

iBatis presenta grandes ventajas con respecto al resto de soluciones de su misma familia:

- iBATIS DAO facilita el manejo de la capa de persistencia mediante la utilización de interfaces.
- iBATIS SQL MAPS facilita la asignación de datos entre el modelo de base de datos y el modelo de clases de la aplicación.
- Simplicidad:
 - De aprender a utilizarlo.
 - Fácil de mantener.
 - Programación declarativa (configurando archivos XML).
 - Separación entre SQL y el lenguaje de programación de la aplicación.
- Portabilidad: entre Java y diferentes proveedores de bases de datos.
- Se trata de código abierto (Open Source).

No obstante, también existen un conjunto de inconvenientes que hacen que en ocasiones iBatis no sea la herramienta más idónea:

- Las bases de datos que gestiona iBATIS deben ser exclusivamente relacionales.
- No es totalmente transparente (hay que programar las consultas).
- Pierde funcionalidad si casi todas las sentencias SQL son construidas dinámicamente.

II.5.3. APACHE STRUTS

Struts es un marco de trabajo *open-source* distribuido con licencia Apache para desarrollo de aplicaciones Web sobre la Plataforma Java EE y está diseñado para seguir el patrón de diseño Modelo Vista Controlador (MVC). Hace uso y extiende la **API** Java Servlet, con el fin de separar las capas de presentación, modelo y control, por lo cual permite una buena organización de la aplicación.

En una aplicación Web estándar, el cliente generalmente envía información al servidor por medio de formularios HTML. Esta información es entonces pasada a un Servlet que la procesa, interactúa con una base de datos y produce una página HTML que envía como respuesta; o bien, la información es pasada a una página JSP que tiene código java incrustado junto con código HTML para obtener el mismo resultado.

Estas dos metodologías resultan inadecuadas para proyectos medianos o grandes porque mezclan lógica de aplicación con lógica de presentación y por eso muestran una gran dificultad al realizar mantenimiento.

El marco Struts actúa como una fachada para aplicaciones Java, proporcionando una clase controladora (un Servlet conocido como ActionServlet) y facilitando la creación de plantillas para la presentación en Web (generalmente Java Server Pages). El desarrollador es responsable de programar dicho controlador y de crear el archivo de configuración para el mismo, el cual une éste con el modelo y con las plantillas de presentación. (Figura 2.50)

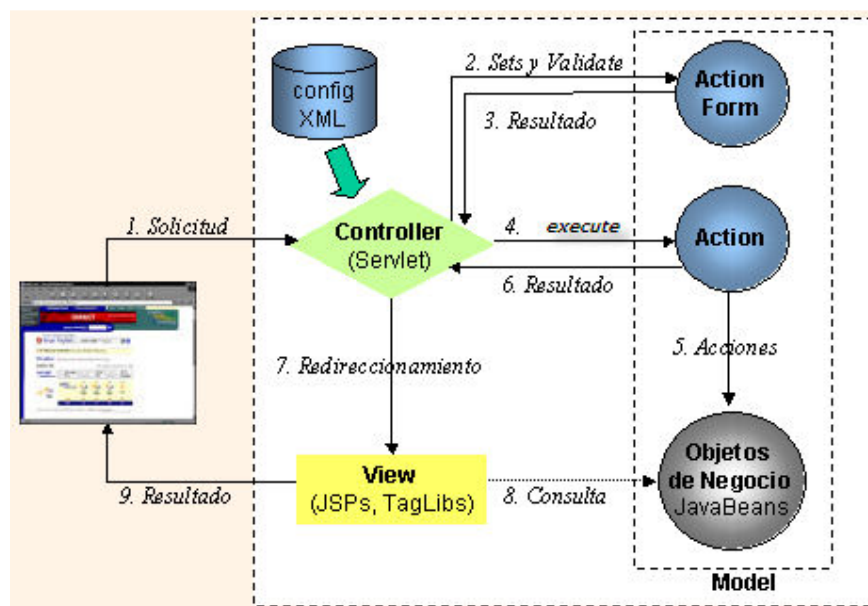


Figura 2.50 Funcionamiento de Struts.

Se decidió trabajar con este marco debido a que es uno de los más utilizados para el desarrollo de aplicaciones Web, es estable y maduro.

Struts proporciona los siguientes componentes para desarrollo:

- Un Servlet (ActionServlet) que actúa como controlador MVC totalmente configurable.
- Clases base que son extendidas para implementar la lógica de la aplicación Web: Action y ActionForm.
- Un diverso conjunto de etiquetas personalizadas JSP que cooperan con el controlador para su uso en componente de vistas (páginas JSP).
- Un marco de validación de datos de entrada desde formularios HTML. También pueden implementarse la validación manualmente dentro de las clases que extienden ActionForm.
- Mecanismos para el manejo y reporte de errores.
- Soporte para la internacionalización a través de archivos de recursos y Java Locales.

II.5.4. POSTGRESQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional (ORDBMS) basado en el proyecto POSTGRES, de la Universidad de Berkeley. PostgreSQL es una derivación libre (OpenSource⁹) de este proyecto.

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Sin embargo, no es un Sistema de Gestión de Bases de Datos puramente orientado a objetos.

II.5.4.1. CARACTERÍSTICAS PRINCIPALES DE POSTGRESQL

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, cadenas de bits, etc. Permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.

⁹ Un software Open Source no tiene una compañía, en lugar de eso reúne programadores que trabajan vía Internet: alguien escribe un programa y lo pone en un lugar donde cualquiera puede acceder; otros programadores lo modifican y cuando está listo lo ponen a disposición de los usuarios; éstos lo prueban y si encuentran errores los reportan a los programadores quienes nuevamente revisan el programa.

- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

II.6. PATRONES DE DISEÑO

Cuando se desarrolla software el diseño del sistema es muy importante, un buen diseño nos ayuda a dar una solución concreta al problema. Sin embargo, para un problema pueden existir muchas soluciones, incluso puede darse el caso de que la solución que se proponga, ya fuera desarrollada por alguien más.

Precisamente para evitar esto, lo mejor sería tener una documentación de los problemas en común con su respectiva solución. Para esto existen los patrones de diseño, que son esquemas genéricos probados que pretenden solucionar problemas de diseño generados en un contexto.

Cada patrón de diseño tiene la función de resolver algún conjunto individual de problemas de diseño y cada uno describe el conjunto de problemas que se encarga de resolver. De esta forma, a través de la descripción del problema es posible determinar si el patrón se aplica al problema específico que enfrenta.

Además el patrón de diseño identifica los objetos significativos de la solución desde el punto de vista de la arquitectura, así como las responsabilidades y relaciones que comparten con estos objetos.

Los patrones de diseño permiten identificar, nombrar y abstraer temas repetitivos y comunes en el diseño orientado a objetos, así como identificar clases, instancias, sus roles, colaboraciones y la distribución de responsabilidades.

Al ser soluciones de diseño probadas nos ayudan a que los sistemas sean:

- Adaptables.
- Extensibles.
- De fácil mantenimiento.
- Reutilizables.
- Alto desempeño.
- Escalables.
- Confiables.

Los patrones de diseño se pueden clasificar en:

1. *Patrones de creación*: Facilitan la creación de instancias de muchas clases mediante instanciación genérica, simplicidad y restricciones de creación.
2. *Patrones de comportamiento*: Se relacionan con la organización del flujo de control en un sistema.
3. *Patrones estructurales*: Describe formas efectivas de particionar y combinar elementos de una aplicación.
4. *Patrones de arquitectura*. Expresa un esquema organizativo estructural fundamental para sistemas software.

II.6.1. MODELO VISTA CONTROLADOR - MVC

Es un patrón de arquitectura, el cual proporciona un método para organizar y dividir el diseño en tres capas o partes.

1. **Modelo:** Es el que representa toda la lógica del negocio.
2. **Vista:** Es la responsable de desplegar la información proporcionada por el modelo.
3. **Controlador:** Es el encargado de comunicar y ser el puente de comunicación entre la Vista y el Modelo. Interpreta los eventos y llama a los métodos apropiados del modelo o la vista.

Antes cuando no se implementaba el MVC se agrupaban los tres objetos en uno solo, lo cual provocaba que los sistemas fueran de difícil mantenimiento y no reutilizables. De ahí el que se separarán para incrementar la flexibilidad y reutilización de los sistemas.

Esta es la razón principal por la cual se tomó el camino de trabajar con este patrón para el desarrollo del sistema. Debido a que desacopla las vistas de los modelos estableciendo entre ellos un protocolo de notificación, cada vista refleja el estado del modelo y cada vez que los datos cambian estos cambios se ven reflejados en las vistas que dependen de él.

MVC en una aplicación Web con Struts.

1. El cliente realiza una petición HTTP la cual es recibida por el controlador que es un servlet (ActionServlet).
2. El controlador procesa la petición. Es probable que realice una petición al modelo. El resultado de procesar la petición se estructura en la forma de un Value Object.
3. El controlador redirecciona el proceso a la vista, representada por una página Web (en este caso Java Server Pages).
4. La responsabilidad de la página es generar la vista con la información del modelo el cual obtiene del Value Object.
5. La vista regresa una página al navegador vía una respuesta HTTP.

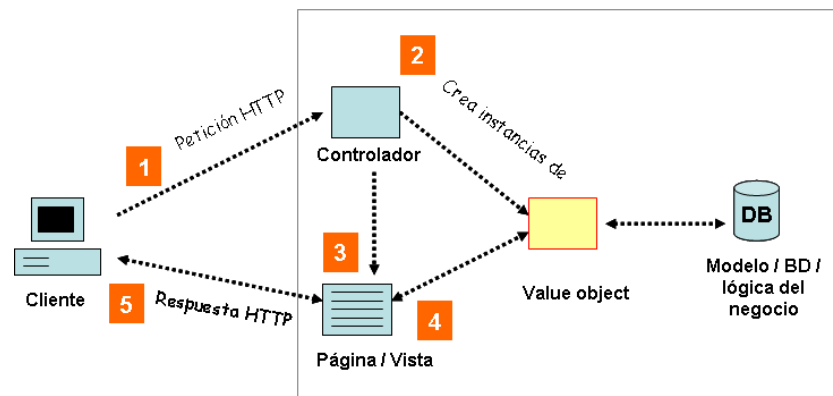


Figura 2.51 Modelo Vista Controlador aplicado en una aplicación Web.

II.7. SEGURIDAD DE LA INFORMACIÓN

Cuando se habla de seguridad en la información hablamos de diversas técnicas, aplicaciones y dispositivos encargados de asegurar la integridad y privacidad de la información de un sistema y sus usuarios.

La necesidad de la seguridad radica en proteger el recurso más valioso de las entidades: **la información**, y una institución educativa no es la excepción; por este motivo la seguridad de la información tiene como principal propósito resguardar la información independientemente de que ésta se encuentre impresa, dentro de un equipo de cómputo o incluso en el saber de las personas.

La seguridad en el área de cómputo, tiene su importancia en el valor que representa en conjunto la información; proporcionando una ventaja competitiva o un riesgo potencial que cualquier entidad posea:

- Una base de datos con información de cualquier tipo referente al giro de la entidad.
- Información confidencial.
- Desarrollo de Software (código fuente) etc.

Para garantizar la seguridad de la información, ésta debe de estar fuera del alcance de cualquier amenaza que pueda afectar su funcionalidad:

- Corrompiéndola.
- Accediéndola indebidamente o incluso
- Eliminándola o hurtándola.

Para que cualquier sistema que se pueda definir como seguro (aunque realmente no hay ningún sistema que sea 100% seguro), debe contar con las siguientes características:

- Integridad: Garantizar que la información sea exacta y completa al momento de ser sujeta al método de procesamiento.
- Confidencialidad: Asegurar que la información está disponible sólo para aquellos usuarios autorizados para su acceso.
- Disponibilidad: Debe Implementar medidas de seguridad necesarias para garantizar que los usuarios autorizados tengan acceso a la información cuando así lo requieran.
- Irrefutabilidad: (No-Rechazo o No Repudio) Que no se pueda negar la autoría.
- Autenticación: Asegurar que se es la persona que dice ser.

De acuerdo a las necesidades de cada entidad, tendrá más peso un objetivo que otro; pero lo anterior no significa que sólo se debe estar pendiente a uno de ellos, al contrario, se debe de mantener un equilibrio entre los cinco.

II.7.1. INTEGRIDAD

La integridad se refiere a la garantía de que la información sea y permanezca confiable, completa y exacta, ésta propiedad permite asegurar que no se ha falseado la información. Por ejemplo, que los datos recibidos o recuperados son exactamente los que fueron enviados o almacenados, sin que se haya producido ninguna modificación, adición o borrado.

El objetivo de la Integridad es mantener la información segura y utilizable sólo por los usuarios autorizados.

La relevancia de este término, radica en la necesidad de asegurar que la información refleja la realidad que la genera. Se trata de proporcionar los medios para verificar que el origen de los datos es el correcto, quién los envió y cuándo fueron enviados y recibidos.

II.7.2. CONFIDENCIALIDAD

La confidencialidad tiene relación con la protección de información frente a posibles accesos no autorizados, con independencia del lugar en que reside la información o la manera en que se almacena.

Es por eso, que la información corporativa y toda aquella información sensible o valiosa que una organización maneja, necesita ser protegida mediante estrictas medidas de control. En este caso, la verificación y la autorización son dos mecanismos que se emplean para asegurar la confidencialidad de la información.

La información únicamente puede ser conocida y debe estar accesible al personal autorizado, de acuerdo a su cargo y funciones que desempeñan en la organización. Además, las mejores prácticas de la gestión de la Seguridad de la Información plantea la necesidad de que el personal autorizado firme un acuerdo de confidencialidad con la empresa. Igualmente, para salvaguardar la confidencialidad, es necesario tener implantadas medidas de seguridad que eviten el acceso y utilización de la información por personas no autorizadas.

II.7.3. DISPONIBILIDAD

La información es uno de los principales activos de negocio de cualquier compañía, y como tal, tiene que ser accesible para los usuarios autorizados de acuerdo al cargo y funciones que desempeñan en la organización. Disponibilidad es un concepto amplio que abarca no solo a la información, sino a los procesos que sustenta su generación y uso. Estamos hablando de los procesos de negocios que, en la medida en que se integran entre sí, demandan y generan información, que debe cumplir con las características de oportunidad y accesibilidad descritas anteriormente.

Disponibilidad de la información, también refiere a la seguridad que la información pueda ser recuperada en el momento que se necesite, esto es, evitar su pérdida o bloqueo, bien sea por ataque doloso, mala operación accidental o causas fortuitas o de fuerza mayor.

II.7.4. IRREFUTABILIDAD

La irrefutabilidad o no repudio es una consecuencia de un proceso confiable de autenticación, y se refiere a la responsabilidad de las partes a cumplir con los compromisos adquiridos. Nuevamente, las claves para asegurar la no repudiación de transacciones, dependerá de la confianza que las partes depositen sobre el proceso de autenticación y la existencia de basamentos legales o contractuales previos a la ejecución de la transacción.

II.7.5. AUTENTICACIÓN

Al hablar de autenticación nos referimos al acto mediante el cual algo o alguien confirma ser auténtico, es decir, que establece que es la persona que efectivamente dice ser, en el caso de las personas confirman su identidad y en el caso de los objetos su procedencia.

Los métodos de autenticación están dados a partir de lo que se utiliza para la verificación y están basados en 3 aspectos que son:

- Algo conocido: como una contraseña o llave pública.
- Algo poseído: como una tarjeta inteligente, un dispositivo USB o una tarjeta de identidad.
- Alguna característica física del usuario o un acto involuntario del mismo: como la verificación de voz, escritura, huellas o la retina del ojo.
- Algo que determina su posición sobre la tierra, mediante dispositivos geoposicionadores (GPS) que reciben señales y las usan para calcular la latitud, longitud y altura del lugar en que se encuentran.

En la mayoría de los sistemas de información vía Web el proceso de autenticación es a través de una contraseña, mismo que le es asignado a cada usuario y con base en un rol le da autorización a determinados recursos.



CAPÍTULO III

Desarrollo del proyecto

CAPÍTULO III. DESARROLLO DEL PROYECTO

III.1. POSICIONAMIENTO DEL PRODUCTO

III.1.1. OPORTUNIDAD DE NEGOCIO

Al implantar el SAACI, la Unidad de Servicios de Cómputo Académico (UNICA) se beneficia al:

Contar con un sistema capaz de llevar un registro detallado de la atención brindada por el personal de UNICA a los incidentes o apoyos internos (dentro de UNICA) o externos (a todas las áreas o divisiones de la Facultad de Ingeniería).

Disponer de información veraz y oportuna para la generación de reportes y oficios, que permitan enterar a los jefes de los Departamentos de UNICA sobre las actividades del personal a su cargo.

Por otro lado, UNICA se mantiene a la vanguardia tecnológica, con un sistema eficiente, fácil de usar y desarrollado con software libre y herramientas que hoy en día son utilizadas por grandes empresas del país.

III.1.2. PROBLEMÁTICA

<p>Necesidad:</p>	<p>La atención de incidentes se realiza a través de una solicitud al área correspondiente, la cual es concentrada y validada.</p> <p>Actualmente, el control del flujo de la información, desde que se genera la solicitud hasta que ésta es atendida se realiza de forma manual en hojas de cálculo Excel; situación que obliga a las áreas a invertir tiempo en actividades de control interno lo que repercute en una mayor probabilidad de retrasos, omisiones, duplicidad de trabajo y uso excesivo de recursos materiales.</p> <p>Por otro lado, no se cuenta con estadísticas inmediatas de supervisión que permitan a las áreas de atención y a las ejecutivas reportes inmediatos sobre la generación, atención y control de los incidentes</p>
<p>Afecta a:</p>	<p>Usuarios encargados de realizar las tareas de atención a incidentes (Jefes del DSC, DROS, DSA y DID respectivamente).</p>
<p>El impacto asociado es:</p>	<p>El tiempo invertido en llevar el control interno de los incidentes atendidos, debido a la falta de uniformidad del proceso dentro de UNICA.</p> <p>Es necesario tener un control detallado de las actividades de atención a incidentes que se realizan en UNICA.</p>

<p>Una solución satisfactoria sería:</p>	<p>Desarrollar un sistema con acceso vía Web a personal autorizado, el cual controle y administre la información de los incidentes en una sola base de datos. Dicho sistema permitirá generar los reportes de manera automática para los departamentos: DSC: Por zona, por causa, por año, por área, por dirección IP, por quién reporta y por fecha; DID, DROS y DSA: Por becario, por periodo, por incidente atendido y por categoría.</p>
---	--

III.1.3. DECLARACIÓN DE POSICIÓN DEL PRODUCTO

<p>Para:</p>	<p>Usuarios encargados de realizar las tareas de atención a incidentes de cómputo (Jefes de los Departamentos: Seguridad en Cómputo (DSC); de Investigación y Desarrollo (DID); de Servicios Académicos (DSA); de Redes y Operación de Servidores (DROS) respectivamente).</p>
<p>Quienes:</p>	<p>Necesitan llevar un control detallado sobre la atención de incidentes de cómputo que realiza el personal de su respectivo departamento. Además de con ello, poder medir los índices de desempeño del sistema de gestión de calidad con el que cuenta UNICA.</p>
<p>SAACI:</p>	<p>Es un sistema Web que es accedido por personal autorizado, quienes administran y consultan datos sobre los incidentes atendidos por el DSC, DID, DROS y DSA.</p>
<p>Que:</p>	<p>Atenderá las necesidades particulares de captura de información sobre incidentes de cómputo para cada departamento, de manera que el personal encargado pueda concentrar sus esfuerzos en la toma de decisiones.</p>
<p>A diferencia de:</p>	<p>Llevar un control interno en hojas de cálculo Excel, que requieren de mayor tiempo en su actualización.</p>
<p>Nuestro producto:</p>	<p>Sistematizará el proceso de atención a incidentes de cómputo, permitiendo concentrar la información en una base de datos, de la cual se podrán realizar consultas. A su vez, permitirá generar de manera automática reportes de los incidentes requeridos tanto por los Jefes del DSC, del DID, del DROS, del DSA y Jefatura respectivamente; como por la Secretaría General de la Facultad de Ingeniería.</p>

III.1.4. PERFILES DE USUARIO

Usuario:	María del Rosario Barragán Paz
Puesto:	Jefe del Departamento de Investigación y Desarrollo de UNICA
Responsabilidades:	Coordinadora de desarrollo de proyectos informáticos de la FI.
Criterio de éxito o aceptación del sistema:	El sistema permite al personal de los Departamentos de UNICA (DID, DROS, DSA y Jefatura) capturar los datos de los incidentes que son atendidos de manera que se tenga centralizada la información en una sola base de datos y permita generar estadísticas en el momento deseado.
Grado de participación:	<p>Usuario:</p> <ul style="list-style-type: none"> > Responsable de listar los requerimientos del sistema para el módulo general. > Determinar si la funcionalidad es correcta. > Revisar las interfaces del sistema y su funcionamiento. > Verificar la integridad del sistema, vigencia y veracidad de la información. > Promover la utilización del sistema dentro de los Departamentos correspondientes.
Comentarios y observaciones:	Ninguno.

Usuario:	Rafael Sandoval Vázquez
Puesto:	Jefe del Departamento de Seguridad en Cómputo (DSC) de UNICA
Responsabilidades:	Implementar y hacer cumplir los reglas que la UNAM dictamine en cuanto al uso de la red dentro de la FI.
Criterio de Éxito o aceptación del sistema:	<p>El sistema permite al personal del DSC capturar los datos de los incidentes que son atendidos.</p> <p>El sistema permite generar reportes y estadísticas de los incidentes atendidos por zona, área, causa, año, por quién reporta el incidente.</p>
Grado de participación:	<p>Usuario:</p> <ul style="list-style-type: none"> > Responsable de listar los requerimientos del sistema para el módulo del DSC. > Determinar si la funcionalidad es correcta. > Verificar la integridad del sistema, vigencia y veracidad de la información.
Comentarios y observaciones:	Ninguno.

III.1.5. ESPECIFICACIÓN DE REQUERIMIENTOS

Requerimientos Funcionales
<p>1) El sistema debe ser accedido vía Web por cada usuario a través de un nombre de usuario y una contraseña.</p> <p>2) Dependiendo del usuario que ingrese al sistema, se habilitarán las diferentes opciones que realizará dentro del sistema.</p> <p>3) El administrador del sistema proporcionará a los usuarios autorizados un nombre de usuario y una contraseña temporal para acceder al sistema, éste deberá permitir al usuario modificar su contraseña así como proporcionar la respuesta a una pregunta secreta para futuras modificaciones u olvidos.</p> <p>4) El sistema será dividido en dos módulos, uno que cubra las necesidades de información del DSC y otra para la información de los incidentes generales para el DID, DROS, DSA y la Jefatura de UNICA.</p> <p>5) El sistema debe tener módulos para alta, modificación y consulta de los datos de los incidentes.</p> <p>6) El sistema permitirá generar reportes automáticos en hoja de cálculo Excel de los siguientes criterios:</p> <p>Para el DSC: Por zona, por causa, por año, por área, por dirección IP, por quién reporta y por fecha. Para el DID, DROS, DSA y Jefatura: Por becario, por periodo, por incidente atendido y por categoría.</p> <p>7) El sistema brindará al usuario datos previamente capturados en catálogos, sin embargo, en caso de no estar enlistado el dato requerido, el sistema permitirá realizar la captura en el mismo formulario.</p> <p>8) El sistema debe permitir dar a conocer lo siguiente:</p> <p>8.1) En el caso de incidentes de cómputo registrados por el DID, DROS, DSA y Jefatura: Fecha de solicitud; Fecha de Atención; Periodo; Becario que atendió; Categoría (Externo/Interno) si la categoría es interna el Nombre del Departamento de UNICA, pero si es externa el nombre del Solicitante, División o Secretaría a la que pertenece; apoyo realizado (Especial, Respaldos, Seguridad, Administrativo, etc.); tiempo estimado y real de respuesta (en días, horas o minutos); descripción del incidente; Status (Atendido/No Atendido).</p> <p>8.2) Para el caso de incidentes registrados por el DSC: persona que atendió; Zona (A, B, C, P, entre otras); Área a la que se atiende (Dirección, División de Ciencias Básicas, Secretaría General, etc.; Fecha de registro; Persona a quien va dirigido el asunto; Quién reporta (DGSCA, DSC); Causa; Respuesta. En caso de requerir la generación de un Oficio se agrega el dato No. de Oficio UNICA y una lista opcional de las personas que reciben copia del oficio.</p>
Requerimientos no funcionales
<p>Desarrollar la documentación del sistema: Manual de usuario, manual del programador.</p>

III.1.6. OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL:

Analizar, diseñar, desarrollar e implementar un Sistema para la Administración, Atención y Control de Incidentes de la Unidad de Servicios de Cómputo Académico.

OBJETIVOS PARTICULARES:

- Modernizar los procedimientos actuales.
- Contar con un sistema Web con el fin de que la aplicación sea independiente de la plataforma en la que se ejecute.
- Dividir el sistema en dos módulos: uno para el DSC y otro para el DID, DROS, DSA y Jefatura con el fin de cubrir las necesidades comunes del control de incidentes para dichos departamentos.
- Acceder al servicio de atención de incidentes de forma fácil, amigable y segura.

III.1.7. METAS

- Mejorar la atención y comunicación entre los usuarios y los diversos Departamentos de UNICA.
- Mejorar el control y registro de apoyos o incidentes en cómputo.
- Seguimiento y reducción del tiempo de atención del servicio.
- Obtener parámetros de evaluación que permitan establecer procesos de mejora continua.
- Visualización del control y el seguimiento a la atención de incidentes o apoyos.
- Disminución de costos.

III.1.8. ALCANCES

El sistema estará diseñado para cubrir las necesidades de control de información de los incidentes en cómputo o de los diversos apoyos que se presenten (Administrativos, Seguridad, etc.) de tal forma que permita darle un seguimiento a la atención de éstos.

Se agilizará la generación de los reportes, la cual será de manera automática a partir de la información que se tenga registrada en el sistema.

El sistema permitirá dar a conocer la relación de incidentes registrados por los siguientes criterios:

- Para incidentes en general: Por nombre del becario asignado; periodo de atención; status y categoría del incidente.
- Para incidentes de seguridad en cómputo y generación de oficios: periodo; año y fecha de atención; zona; dirección IP; causa; por quién reporta y área.

Para llevar a cabo lo anterior, el sistema contemplará dos módulos principales para administrar la información de los incidentes como son:

Un Módulo para el DSC y otro general para los tres departamentos restantes de UNICA.

Para el Módulo del DSC, el sistema permitirá:

- Generar un oficio automáticamente.
- Registrar el incidente sin generar un oficio.
- Llevar un control de cómo es la atención a los incidentes.
- Actualizar la información.
- Generar estadísticas.
- Realizar búsquedas de los incidentes por los siguiente criterios: zona, dirección IP, causa, por quién reporta, [periodo, año, fecha] y área.

Para el Módulo general, el sistema permitirá:

- Registrar incidentes o apoyos.
- Llevar un control de cómo es la atención a los incidentes o apoyos.
- Actualizar la información.
- Realizar búsquedas de los incidentes o apoyos por los siguiente criterios: por becario, por periodo, por status (atendido/no atendido) y por categoría (interno/externo).

III.1.9. RESTRICCIONES

- El sistema es estable en el navegador Web: FireFox versión 2.0 y superior.
- El sistema generará en un principio sólo reportes en formato RTF y Excel.

III.1.10. RECURSOS TECNOLÓGICOS Y HUMANOS

El Departamento de Investigación y Desarrollo (lugar donde será albergado el SAACI) cuenta con los siguientes recursos humanos y tecnológicos para darle alojamiento, seguimiento y mantenimiento al sistema.

<p>Recursos Tecnológicos</p>	<p>Para el desarrollo del producto, UNICA cuenta con equipos que tienen las siguientes características:</p> <p>Procesador: Pentium IV o superior. Memoria RAM: 512 Mb o superior. Disco Duro: 80 Gb o superior. Sistema Operativo: Windows XP Professional. Paquetería:</p> <ul style="list-style-type: none"> ▪ Lenguaje de Modelado Unificado (UML) con StartUML. ▪ NetBeans. ▪ Lenguaje Java (versión 1.5 o posterior). ▪ Manejador de base de datos POSTGRESQL 8.2. ▪ Servidor de Aplicaciones Web TOMCAT (versión 1.5 o posterior). <p>Red: Los equipos están en red conectados a través de una Interfaz Fast Ethernet y tienen salida a Internet.</p> <p>Para el alojamiento del sistema, se cuenta con Servidores Web y de Base de Datos en producción que tienen en general las siguientes características:</p> <p>Procesador: Pentium IV en adelante. Memoria RAM: 1 GB en adelante. Disco Duro: 160 GB en adelante. Sistema Operativo: Linux Fedora. Paquetería:</p> <ul style="list-style-type: none"> ▪ Manejador de base de datos POSTGRESQL (versión 8.2). ▪ Lenguaje Java (versión 1.5 o posterior). ▪ Servidor de Aplicaciones Web TOMCAT (versión 1.5 o posterior). <p>Red: Los equipos están en red conectados a través de una Interfaz Fast Ethernet y tienen salida a Internet.</p>
<p>Recursos Humanos</p>	<p>UNICA cuenta con personal capacitado para cumplir las tareas de desarrollo y mantenimiento de los sistemas que la Unidad requiere.</p> <p>El perfil básico de conocimiento que debe cubrir para el proyecto es:</p> <ul style="list-style-type: none"> ▪ Conocimientos de programación orientada a objetos y del patrón de diseño MVC (Modelo-Vista-Controlador). ▪ Lenguaje Java, Java Server Pages, Javascript, XML y HTML. ▪ Manejo de Frameworks de Java, principalmente iBatis y Apache Struts. ▪ Manejador de base de datos PotsgreSQL.

III.1.11. COSTO

Para obtener un estimado de los costos para el desarrollo del sistema, se considero el análisis, diseño, programación, pruebas e implementación. En la siguiente tabla se indica a detalle lo que se considera para el costo final del presente proyecto:

Recursos	Horas trabajadas	Costo por hora	Costo real
Desarrollador del Sistema*	1440	\$100.00	\$144,000.00
Consultores y asesores	360	\$120.00	\$43,200.00
Totales			\$187,200.00

Otros Recursos	Cantidad de equipos o servicios	Costo Unitario Referencial	Costo Total
Adquisición de Hardware**	1 servidor para base de datos.	\$9,000.00	\$18,000.00
	1 servidor de aplicaciones web.	\$9,000.00	
Licencias de Software***		\$0	\$0
Mantenimiento de la aplicación****		\$0	\$0
Energía Eléctrica*****		\$2,000.00	\$2,000.00
Totales			\$20,000.00

Costo Final			\$207,200.00
--------------------	--	--	---------------------

* El factor principal del costo del proyecto es el salario del Desarrollador del Sistema que asciende a \$144,000.00. Muchas veces cuando se desarrollan sistemas, es necesaria la inclusión de asesores externos o consultores para acelerar o apoyar el desarrollo del sistema. En este caso, se tuvo apoyo y asesoría de dos becarios de UNICA, lo cual involucró un costo de \$43,200.

** Se consideró el costo de los servidores para alojar tanto la aplicación como la base de datos (ambiente de producción), basándose en los requerimientos mínimos que se mencionaron en el punto III.1.10. (Recursos Tecnológicos y Humanos), obteniendo un costo aproximado de \$18,000.

*** No aplican pues para el desarrollo del sistema se utiliza software libre: Lenguaje de programación JAVA con Apache Struts; Framework iBatis; Manejador de Base de Datos PostgreSQL; y Apache Tomcat como servidor de aplicaciones Web.


**** Como plus del servicio, se consideran 6 meses de asesoría y soporte gratuito a la aplicación.

***** Para obtener este dato utilizamos el software llamado Microsoft Joulemeter y con base en ello se determinó el costo final del consumo de energía durante el desarrollo del proyecto que es de \$2,000.

En resumen, se obtiene un costo total de **\$207,200.00**.

III.2. ESPECIFICACIONES DE CASOS DE USO

RELACIÓN DE CASOS DE USO		
ID	NOMBRE	CLAVE
01	Acceder al Sistema	A-SIS
02	Recuperar Contraseña	R-CON
03	Modificar Contraseña	M-CON
04	Registrar Incidentes	R-INC
05	Generar Oficio	G-OFC
06	Consultar Incidentes	C-INC
07	Actualizar Información del Incidente	A-INC
08	Generar Estadísticas	G-EST


	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-SIS
	Institución:	UNICA	Página:	1/2

CASO DE USO

Id Caso de Uso:	01		
Nombre:	ACCEDER AL SISTEMA		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	13/Septiembre/2007	Fecha actualización:	23/Marzo/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los departamentos del DID, DSC, DSA, DROS y Jefatura de UNICA	X		Los usuarios autorizados para acceder al sistema podrán: <ul style="list-style-type: none"> ▪ Llevar un control de la atención que se le da a los diferentes incidentes (N-01) asignados a UNICA. ▪ Obtener reportes, estadísticas y consultas específicas acerca de dichos incidentes.

Descripción:
Permite ingresar al Sistema para la Administración, Atención y Control de Incidentes de la Unidad de Servicios de Cómputo Académico (UNICA).
Precondiciones:
<i>Del proceso</i> <ul style="list-style-type: none"> ▪ El actor debe contar con un nombre de usuario y contraseña válidos para poder hacer uso del sistema (N-02).
Flujo Principal y Alternativos
<p>I) El caso de uso inicia cuando alguno de los jefes de los departamentos de UNICA ingresan al sistema por las siguientes razones:</p> <ul style="list-style-type: none"> ▪ Registrar incidentes. ▪ Actualizar el status o los datos de los incidentes registrados. ▪ Consultar la información registrada sobre los incidentes. ▪ Generar estadísticas o reportes de incidentes registrados.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-SIS
	Institución:	UNICA	Página:	2/2

- II) El sistema pide que se introduzcan los siguientes datos: **(N-03)**.
- Nombre de usuario (login), Contraseña.
- III) El sistema provee una opción para que el usuario pueda recuperar su contraseña si no la recuerda (**Ver caso de uso: Recuperar contraseña**).
- IV) El usuario captura los datos y acepta el ingreso **(E-01) (E-02)**.

IV.1) El sistema verifica el tipo de usuario y sus privilegios:

IV.1.1) Si el usuario que ingresa es el Jefe del Departamento de Seguridad en Cómputo (DSC), el sistema le muestra las siguientes opciones:

- ✓ Registrar incidentes.
- ✓ Generar oficios.
- ✓ Consultar incidentes. **(N-04)**
- ✓ Consultar oficios. **(N-04)**
- ✓ Generar estadísticas de los incidentes.
- ✓ Administración de cuentas de usuarios del DSC.
- ✓ Salir del sistema.

IV.1.2) Si el usuario es alguno de los Jefes del Departamento de Investigación y Desarrollo (DID), del Departamento de Servicios Académicos (DSA), del Departamento de Redes y Operación de Servidores (DROS) o es de Jefatura el sistema le muestra las siguientes opciones:

- ✓ Registrar incidentes.
- ✓ Realizar búsquedas de incidentes: por becario, por periodo, por status (atendido/no atendido) y por categoría (interno/externo). **(N-04)**
- ✓ Salir del sistema.

V) El caso de uso termina.

Excepciones:

E-01 → Para permitir el ingreso, el sistema verifica que los campos de nombre de usuario y contraseña no se dejen en blanco y que estén registrados en la base de datos.

E-02 → El sistema vuelve a solicitar los datos hasta que sean válidos.


Notas:

N-01 → Para fines específicos de este documento, llamaremos *Incidente* a aquel suceso o evento que requiere de la atención por parte de algún departamento de UNICA. Por ejemplo: Incidentes de seguridad en cómputo, peticiones administrativas o de soporte (arreglo de impresoras, etc.), entre otros.

N-02 → El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema.

N-03 → Si el usuario ya tiene asignado su nombre de usuario y contraseña, puede modificar su esta última en cualquier momento (**Ver caso de uso: Modificar Contraseña**).

N-04 → El usuario podrá modificar los datos de los incidentes que consulta.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-CON
	Institución:	UNICA	Página:	1/2

CASO DE USO

Id Caso de Uso:	02		
Nombre:	RECUPERAR CONTRASEÑA		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	13/Septiembre/2007	Fecha actualización:	15/Mayo/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los departamentos del DID, DSC, DSA, DROS y Jefatura de UNICA	X		Usuarios autorizados para utilizar el sistema y hayan olvidado su nombre de usuario y/o contraseña.

Descripción:

Permite recordar el nombre de usuario y/o la contraseña para ingresar al sistema, dichos datos se enviarán al correo electrónico del usuario.

Precondiciones:

Del proceso

- El actor debe de seleccionar la opción “¿Olvidó su Contraseña?” ubicada en la página principal del sistema. (Ver caso de uso: Acceder al Sistema).
- El actor debe tener registrado en el sistema la siguiente información: [Nombre de usuario, Contraseña, Nombre, Apellidos, Departamento de UNICA al que pertenece, Correo Electrónico, Respuesta a una pregunta secreta]. (N-01)


Flujo Principal y Alternativos

I) El caso de uso inicia cuando alguno de los Jefes actores desea ingresar al sistema pero no recuerda su nombre de usuario y/o contraseña.

II) El usuario selecciona la opción: ¿Olvidó su contraseña? desde la página de Autenticación.

III) El sistema despliega un formulario solicitando:

- Nombres.
- Apellidos.
- Un listado para seleccionar una pregunta secreta.
- Respuesta a la pregunta secreta.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-CON
	Institución:	UNICA	Página:	2/2

III.1) El sistema provee además una opción para que el usuario pueda modificar su contraseña, para esto el usuario debe de conocer su nombre de usuario y contraseña actual. (Ver caso de uso Modificar contraseña).

IV) El usuario captura los datos y selecciona la opción Recordar **(E-01)**.

IV.1) Si los datos son correctos, el sistema envía el nombre de usuario y contraseña al correo del usuario que se tiene registrado y se muestra el mensaje: "Sus datos se han enviado con éxito a su correo electrónico". **(N-02)**.

IV.2) Si los datos son incorrectos el sistema muestra el siguiente mensaje: "El usuario no existe favor de verificar". **(E-02)**

V) El usuario selecciona la opción para regresar a la página principal del sistema.

VI) El caso de uso termina.

Excepciones

E-01 → Para poder modificar el nombre de usuario y la contraseña el sistema verifica lo siguiente:


- α) Que los campos de nombres, apellidos y respuesta a la pregunta secreta no se dejen en blanco.
- β) Que el nombre, los apellidos, la pregunta y respuesta secreta estén registrados en el sistema.

E-02 → El sistema vuelve a solicitar los datos hasta que sean válidos.

Notas:

N-01 → La información del usuario debe ser ingresada por el administrador principal del sistema.

N-02 → El remitente del correo será saaci@cancun.fi-a.unam.mx

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	M-CON
	Institución:	UNICA	Página:	1/2

CASO DE USO

Id Caso de Uso:	03		
Nombre:	MODIFICAR CONTRASEÑA		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	13/Septiembre/2007	Fecha actualización:	24/Agosto/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los departamentos del DID, DSC, DSA, DROS y Jefatura de UNICA	X		Usuarios autorizados para utilizar el sistema y que deseen modificar su contraseña para acceder al sistema.

Descripción:

Permite modificar la contraseña para ingresar al Sistema para la Administración, Atención y Control de Incidentes de UNICA. Además si el usuario lo desea puede cambiar la pregunta y respuesta secreta.

Precondiciones:


Del proceso

- El actor debe de seleccionar la opción “Modifica tu Contraseña” ubicada en la página ¿Olvidó su contraseña? (**Ver caso de uso: Recuperar Contraseña**).
- Para modificar la contraseña, pregunta y respuesta secreta, es necesario que el actor conozca su nombre de usuario y contraseña actual.
- El actor debe tener registrado en el sistema la siguiente información: [Nombre de usuario, Contraseña, Nombre, Apellidos, Departamento de UNICA al que pertenece, Correo Electrónico, Respuesta a una pregunta secreta]. (**N-01**)

Flujo Principal y Alternativos

I) El caso de uso inicia cuando alguno de los actores desea modificar su contraseña.

II) El usuario selecciona la opción: “Modifica tu Contraseña”.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	M-CON
	Institución:	UNICA	Página:	2/2

III) El sistema despliega un formulario solicitando:

- Nombre de Usuario (Login).
- Contraseña actual.
- Nueva contraseña.
- Confirmación de Nueva contraseña.
- Una opción para cambiar la pregunta y respuesta secreta.

IV) El usuario captura los datos y selecciona la opción Modificar **(E-01)**.

IV.1) Si los datos son correctos el sistema modifica la contraseña y muestra el mensaje: “Éxito al modificar la contraseña.”

IV.2) Si el actor selecciona también desea cambiar la respuesta y/o pregunta secreta y los datos son correctos, el sistema modifica la información y muestra el mensaje: “Éxito al modificar la contraseña, sus datos personales se han registrado satisfactoriamente”.

IV.3) Si los datos son incorrectos el sistema muestra el siguiente mensaje: “El usuario no existe favor de verificar.” **(E-02)**.

V) El usuario selecciona la opción para regresar a la página principal del sistema.

VI) El caso de uso termina.

Excepciones


E-01 → Para modificar la contraseña el sistema verifica lo siguiente:

- a) Que los campos de nombre de usuario, contraseña actual, nueva contraseña y confirmación de la contraseña no se dejen en blanco.
- b) Que la confirmación de la nueva contraseña coincida con el campo de “Nueva contraseña”.
- c) Que la contraseña actual y el nombre de usuario del usuario estén registrados en el sistema.
- d) Que la nueva contraseña tenga como mínimo 8 caracteres y máximo 12 caracteres.
- e) Si selecciona cambiar la pregunta secreta, la respuesta y pregunta no se dejen en blanco.

E-02 → El sistema vuelve a solicitar los datos hasta que sean válidos.

Notas:

N-01 → La información del usuario debe ser ingresada por el administrador principal del sistema.


	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	1/6

CASO DE USO

Id Caso de Uso:	04		
Nombre:	REGISTRAR INCIDENTES		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	23/Septiembre/2007	Fecha actualización:	27/Mayo/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los departamentos del DID, DSC, DSA y DROS.	X		Usuarios encargados de realizar el registro del Incidente.

Descripción:
Permite realizar el registro de los incidentes asignados a los becarios de UNICA en sus diferentes Departamentos.
Precondiciones:
<p><i>Del proceso</i></p> <ul style="list-style-type: none"> ▪ El actor debe contar con nombre de usuario y contraseña válidos para poder hacer uso del sistema (N-01). ▪ Para realizar el registro de un incidente del DID, DSA, DROS y Jefatura, es necesario contar con la siguiente información: <ul style="list-style-type: none"> • Nombre del becario al que se le asigna el incidente. • Categoría del incidente (Interno o Externo). <ul style="list-style-type: none"> • Interno: Nombre del Departamento en UNICA. • Externo: Persona a quien va dirigido y división o secretaría en que labora. • Tipo de incidente. • Tiempo estimado de respuesta. • Status del incidente (Atendido/No Atendido). <ul style="list-style-type: none"> • Si es Atendido: Registrar el tiempo real de respuesta y la fecha en que se atendió. • Fecha de registro. • Descripción. ▪ Para realizar el registro de un incidente del DSC es necesario contar con la siguiente información: <ul style="list-style-type: none"> • Asunto

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	2/6

- Nombre de la persona a quien va dirigida la atención, el cargo que ocupa en la Facultad y el área donde labora.
- Dirección IP.
- Quién reporta el incidente. (DGSCA o DSC).
- Magnitud del incidente.
- Causa del incidente.
- Nombre(s) de los becarios que atendieron el incidente.
- Tiempo de la atención.
- Observaciones.

Flujo principal y alternativos:

I. El caso de uso inicia cuando los actores ingresan al sistema para registrar un incidente.

II. El usuario selecciona la opción “Registrar Incidentes”.


II.1. Si los usuarios que ingresan pertenecen al DID, al DROS o al DSA, el sistema despliega un formulario solicitando:

- **Becario.**

- El sistema da una opción para consultar los nombres de becarios activos de UNICA.
- El usuario selecciona la opción.
- El sistema muestra en una sub-ventana el listado de todos los nombres de los becarios activos, así como el listado de los departamentos de UNICA.
- El usuario puede localizar al becario en el listado completo por su nombre **(R-01)** o bien, buscarlo por el departamento al que pertenece.
 - Si el usuario selecciona algún departamento del listado, el sistema despliega los nombres de los becarios activos que pertenecen a dicho departamento.
- El usuario selecciona el nombre del becario.
- El sistema pone en los campos correspondientes de la ventana principal el nombre y apellidos del becario seleccionado.

- **Categoría del apoyo: Interno o Externo (N-03).**

- Si el usuario selecciona **Externo**, el sistema despliega un listado con el nombre de las Divisiones o Secretarías a las que se puede dar apoyo.
- El usuario selecciona la División o Secretaría.
- El sistema muestra un listado de la División o Secretaría, asociado con otro listado que contiene los nombres de las personas que laboran en dicho lugar.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	3/6

- Si la División o Secretaría a quien va dirigido el apoyo no se encuentra enlistada, el sistema permite registrar los datos seleccionando la opción de “Agregar otro”.
 - El sistema despliega en el mismo formulario los campos para agregar una nueva División o Secretaría; así como los campos para el nombre y apellidos de la persona a quien va dirigido el apoyo.
- Si el nombre de la persona no se encuentra enlistado, el sistema permite registrar los datos seleccionando la opción de “Agregar otro”.
 - El sistema despliega en el mismo formulario los campos de nombre y apellidos.
- Si el usuario selecciona **Interno**, el sistema oculta los campos para la categoría Externo (*Ver categoría: Externo*) y muestra un listado con los nombres de los departamentos de UNICA.

- **Tipo de apoyo.**

- Si el tipo de apoyo no se encuentra enlistado, el sistema permite registrar los datos seleccionando la opción: “Agregar otro”.
 - El sistema despliega en el mismo formulario el campo para el registro del nuevo tipo de apoyo.

- **Tiempo estimado de respuesta en minutos, horas o días.**
- **Status del incidente (Atendido/No Atendido).**
- **Fecha en que se atendió el incidente (Si el estatus es Atendido).**
- **Fecha de registro (N-02).**
- **Descripción de la atención dada.**

II.2. Si los usuarios que ingresan pertenecen al DSC, el sistema despliega un formulario solicitando:


- **Asunto.**

- El sistema muestra un listado de los asuntos registrados anteriormente.
 - Si el asunto no está enlistado, el usuario selecciona la opción “Agregar Otro”.
 - El sistema despliega un campo para agregar el “Nuevo Asunto”.

Información del Solicitante:

- **Dirigido a.**


- El sistema muestra un listado de las personas que solicitan la atención del Incidente.
 - Si selecciona una persona se muestra en un listado el área asociada a la persona.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	4/6


- Es posible cambiar el área o agregar una nueva seleccionando la opción “Agregar Otro”.
 - El sistema solicita la siguiente información para la “Nueva Área”.
 - Nombre completo del área.
 - Siglas.
- Si el nombre de la persona no está enlistado, el usuario selecciona la opción “Agregar Otro”.
- El sistema solicita la siguiente información de la “Nueva persona”:
 - Nombre.
 - Apellidos.
 - Responsable.
 - Área.
 - Si el área no está enlistada, el usuario selecciona la opción “Agregar Otro”.
 - El sistema solicita la siguiente información para la “Nueva Área”.
 - Nombre completo del área.
 - Siglas.

Información del Incidente:


- **Fecha (DD/MM/AAAA).** El sistema pondrá la fecha actual.
- **Dirección IP.**
 - El sistema muestra un listado con los primeros 3 octetos de la dirección IP donde se presenta el incidente y alado un campo para indicar el cuarto octeto.
 - Si los tres primeros octetos no se encuentran enlistados, el usuario selecciona la opción “Agregar Otro”.
 - El sistema oculta el campo para el cuarto octeto y solicita la siguiente información para el Alta de la nueva IP:
 - IP.
 - Zona.
 - Si los primeros octetos si se encuentran enlistados, el usuario introduce el último octeto de la dirección IP.
- **Reporta.**
 - El sistema muestra un listado de las instituciones que pueden reportar los incidentes (DGSCA y DSC).
- **Magnitud del Incidente.**
 - El sistema muestra un listado con las opciones de criticidad del incidente: Moderado, Importante, Crítico.
 - El usuario selecciona entre las opciones mostradas.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	5/6

- **Causa.**
 - El sistema despliega un listado con las causas registradas.
 - Si la causa no se encuentra enlistada, el usuario selecciona la opción “Agregar Otro”.
 - El sistema solicita la siguiente información para el Alta de la nueva causa:
 - Nombre de la causa.
 - Descripción.
 - **Información de la Atención del Incidente:**
 - **Realizó.**
 - El sistema muestra un listado con los nombres de los becarios activos que pertenecen a la DSC.
 - El sistema tiene una opción para actualizar este listado de forma automática. (R-02)
 - **Tiempo.**
 - El sistema solicita que se introduzca el tiempo en minutos, horas o días.
 - El usuario introduce el tiempo en que tardó en atenderse el incidente.
 - **Observaciones.**
 - El sistema solicita que se introduzca algún comentario u observación de cómo fue la atención al incidente.
- V) El usuario captura la información y acepta el registro (E-01).
- VII) El sistema despliega un mensaje de que se realizó con éxito el registro.
- VIII) Si el usuario desea registrar otro incidente el caso de uso inicia en el punto II, de lo contrario, el caso de uso termina.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	R-INC
	Institución:	UNICA	Página:	6/6

Excepciones
<p>E-01 -> En caso de que no se realice el registro, el sistema informa sobre el problema o se indica si algún dato no es válido.</p>
Requisitos Especiales:
<p>R-01 -> El listado estará ordenado por nombres alfabéticamente.</p> <p>R-02 -> Los becarios ya deben estar registrados o dados de baja en la base de datos del SAPIUN (Sistema para la Administración de la Información Personal de UNICA).</p>
Notas:
<p>N-01 El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema (Ver caso de uso: Acceder al Sistema).</p> <p>N-02 La fecha de registro por omisión es la fecha actual del sistema.</p> <p>N-03 Pertenecen a la categoría externa todos aquellos incidentes que son presentados en instituciones o entidades de la FI que no pertenecen a UNICA.</p>

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	G-OFC
	Institución:	UNICA	Página:	1/3

CASO DE USO

Id Caso de Uso:	05		
Nombre:	GENERAR OFICIO		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	23/Septiembre/2007	Fecha actualización:	10/Junio/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefe del Departamento de Seguridad en Cómputo (DSC)	X		Usuario encargado de generar el oficio de un incidente.

Descripción:

Permite generar el oficio de un incidente de seguridad atendido por el DSC de UNICA. Además, en caso de que sea necesario es posible dar de alta nuevos asuntos, causas de Incidentes, direcciones IP, información de las personas relacionadas con el incidente y el área donde laboran en la Facultad de Ingeniería y las personas que van a tener copia del oficio.

Precondiciones:


Del proceso

- El actor debe contar con nombre de usuario y contraseña válidos para poder hacer uso del sistema **(N-01)**.
- Para realizar un registro de un incidente es necesario contar con la siguiente información:
 - Asunto.
 - Nombre de la persona a quien va dirigido la atención, el cargo que ocupa en la Facultad y área donde labora.
 - Dirección IP.
 - Quién reporta el incidente.
 - Causa del incidente.
 - Fecha
 - Nombre(s) de las personas a quien se envía copia del oficio.

Flujo Principal y Alternativos


I) El caso de uso inicia cuando el Jefe del Departamento de Seguridad en Cómputo ingresa al sistema para generar el oficio de un incidente.

II) El usuario selecciona la opción para Generar Oficios.


	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	G-OFC
	Institución:	UNICA	Página:	2/3

II) El sistema despliega un formulario solicitando:

- **Asunto.**
 - El sistema muestra un listado de los asuntos registrados anteriormente.
 - Si el asunto no está enlistado, el usuario selecciona la opción “Agregar Otro...”.
 - El sistema despliega un campo para agregar el “*Nuevo Asunto*”.
- **Dirección IP.**
 - El sistema muestra un listado con los primeros 3 octetos de la dirección IP donde se presenta el incidente y alado un campo para indicar el cuarto octeto.
 - Si los tres primeros octetos no se encuentran enlistados, el usuario selecciona la opción “Agregar Otro”.
 - El sistema oculta el campo para el cuarto octeto y solicita la siguiente información para el Alta de la nueva:
 - IP.
 - Zona.
 - Si los primeros octetos si se encuentran enlistados, el usuario introduce el último octeto de la dirección IP.
- **Dirigido a.**
 - El sistema muestra un listado de las personas que solicitan la atención del Incidente.
 - Si selecciona una persona se muestra en un listado el área asociada a la persona.
 - Es posible cambiar el área o agregar una nueva seleccionando la opción “Agregar Otro”.
 - El sistema solicita la siguiente información para la “Nueva Área”.
 - Nombre completo del área.
 - Siglas.
 - Si el nombre de la persona no está enlistado, el usuario selecciona la opción “Agregar Otro”.
 - El sistema solicita la siguiente información de la “Nueva persona”:
 - Nombre.
 - Apellidos.
 - Responsable.
 - Área.
 - Si el área no está enlistada, el usuario selecciona la opción “Agregar Otro”.
 - El sistema solicita la siguiente información para la “Nueva Área”.
 - Nombre completo del área.
 - Siglas.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	G-OFC
	Institución:	UNICA	Página:	3/3

<ul style="list-style-type: none"> • Reporta. <ul style="list-style-type: none"> ▪ El sistema muestra un listado de las instituciones que pueden reportar los incidentes (DGSCA, DSC). • Causa. <ul style="list-style-type: none"> ▪ El sistema despliega un listado con las causas registradas. ▪ Si la causa no se encuentra enlistada, el usuario selecciona la opción “Agrega Otro”. <ul style="list-style-type: none"> • El sistema solicita la siguiente información para el Alta de la nueva causa: <ul style="list-style-type: none"> ○ Nombre de la causa. ○ Descripción. • Fecha. <ul style="list-style-type: none"> ▪ El sistema despliega un calendario para que el usuario seleccione la fecha. • Con copia para (C.C.P.). <ul style="list-style-type: none"> ▪ El sistema despliega un listado con los nombres de las personas que pueden recibir copia del oficio. ▪ El usuario puede seleccionar a una o más personas. ▪ Si el nombre de la persona no se encuentra enlistada, el sistema permite registrar los datos seleccionando la opción “Agrega otro”. <ul style="list-style-type: none"> • El sistema despliega una sub-ventana con los campos para agregar el nombre, apellidos y responsabilidad de la nueva persona a registrar. <p>IV) El usuario captura la información y acepta el registro (E-01).</p> <p>V) El sistema despliega un mensaje de que se realizó con éxito el registro y una opción para abrir el oficio.</p> <p style="padding-left: 40px;">V.1) Si el usuario selecciona la opción para abrir el oficio, el sistema le permite guardar el oficio en formato RTF.</p> <p>VI) Si el usuario desea registrar otro incidente el caso de uso inicia en el punto II, de lo contrario el caso de uso termina.</p>
Excepciones
E-01 -> En caso de que no se realice el registro el sistema informa sobre el problema (indica si algún dato no es válido, error de conexión con la base de datos, etc.).
Notas:
N-01 → El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema (Ver caso de uso: Acceder al Sistema).

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	C-INC
	Institución:	UNICA	Página:	1/5

CASO DE USO

Id Caso de Uso:	06		
Nombre:	CONSULTAR INCIDENTES		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	04/Octubre/2007	Fecha actualización:	07/Julio/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los departamentos del DID, DSC, DSA y DROS.	X		Usuarios autorizados de consultar incidentes o apoyos registrados en el sistema.

<p>Descripción:</p> <p>Para el caso del DID, del DROS y del DSA, el sistema permite realizar la búsqueda del incidente por su status (Atendido/No atendido), por becario, por periodo o bien por categoría (Interno o Externo).</p> <p>Para el caso del DSC, el sistema permite realizar la búsqueda del incidente por los criterios: Zona, Dirección IP, Causa, Institución que reporta, Periodo, Año, Fecha, y Áreas. (N-02).</p>
<p>Precondiciones:</p> <p><i>Del proceso</i> El actor debe contar con nombre de usuario y contraseña válidos para poder hacer uso del sistema (N-01).</p>
<p>Flujo Principal y Alternativos</p> <p>I) El caso de uso inicia cuando los actores ingresan al sistema para consultar un incidente.</p> <p>II) Si los usuarios que ingresan al sistema pertenecen al DID, al DROS o al DSA, el sistema muestra un listado con los diferentes parámetros de consulta:</p> <ol style="list-style-type: none"> 1. Incidentes Atendidos 2. Por becario 3. Por periodo 4. Por categoría <p>II.1) El usuario selecciona el parámetro de consulta.</p>

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	C-INC
	Institución:	UNICA	Página:	2/5

II.1.1) Si la opción elegida es **Incidentes Atendidos**, el sistema despliega por omisión los incidentes *No atendidos* con los siguientes datos: **(E-01)**:

- Fecha de Solicitud.
- Fecha de Atención.
- Periodo.
- Atendió.
- Categoría.
- División o Secretaría.
- Dirigido a.
- Apoyo realizado.
- Tiempo estimado de respuesta.
- Tiempo de respuesta.
- Descripción.
- Status.


II.1.1.1) El sistema permite alternar la consulta entre los incidentes Atendidos y No atendidos.

II.1.2) Si la opción elegida es **Por becario** el sistema despliega los siguientes datos: **(E-01)**:

- Fecha de Solicitud.
- Fecha de Atención.
- Periodo.
- Atendió.
- Categoría.
- División o Secretaría.
- Dirigido a.
- Apoyo realizado.
- Tiempo estimado de respuesta.
- Tiempo de respuesta.
- Descripción.
- Status.

II.1.2.1) El sistema permite realizar la búsqueda de un becario en particular para lo cual solicita el(los) nombre(s) y apellidos del becario.

- El usuario selecciona la opción para buscar el nombre del becario.
- El sistema muestra en una sub-ventana el listado de todos los nombres de los becarios activos, así como el listado de los departamentos de UNICA.
- El usuario puede localizar al becario en el listado completo por su nombre **(R-01)** o bien, buscarlo por el Departamento al que pertenece.
 - Si el usuario selecciona algún departamento del listado, el sistema despliega los nombres de los becarios activos que pertenecen a dicho departamento.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	C-INC
	Institución:	UNICA	Página:	3/5

- El usuario selecciona el nombre del becario.
- El sistema coloca los datos de nombre y apellidos del becario en los campos respectivos.
- El usuario selecciona la opción Buscar.
- El sistema despliega los incidentes que atendió el becario con los siguientes datos:
 - Fecha de Solicitud.
 - Fecha de Atención.
 - Periodo.
 - Atendió.
 - Categoría.
 - División o Secretaría.
 - Dirigido a.
 - Apoyo realizado.
 - Tiempo estimado de respuesta.
 - Tiempo de respuesta.
 - Descripción.
 - Status.

II.1.3) Si la opción elegida es **Por categoría**, el sistema despliega por omisión los incidentes *No atendidos* con los siguientes datos: **(E-01)**:

- Fecha de Solicitud.
- Fecha de Atención.
- Periodo.
- Atendió.
- Categoría.
- División o Secretaría.
- Apoyo realizado.
- Tiempo estimado de respuesta.
- Tiempo de respuesta.
- Descripción.
- Status.


II.1.3.1) El sistema permite alternar la consulta seleccionando la categoría Externo o Interno.

II.1.3.2) Si el usuario selecciona la categoría **Externo**, el sistema despliega el listado con las Divisiones o Secretarías registradas asociado a un listado con los nombres de las personas a quien va dirigido el apoyo.

II.2) El sistema permite actualizar la información del registro de incidente habilitando el campo de **Status** del incidente. (**Ver caso de uso Actualizar Información del Incidente**).

II.3) El sistema despliega una opción para Generar el Reporte en Excel de los resultados obtenidos.

III) Si los usuarios que ingresan al sistema pertenecen al DSC:

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	C-INC
	Institución:	UNICA	Página:	4/5

III.1) El sistema muestra un listado con las opciones:

Búsqueda de Incidentes Atendidos.
Búsqueda de Oficios.

III.2) Si el usuario selecciona la opción “**Búsqueda de Incidentes Atendidos**”:

III.2.1) El sistema permite desplegar los siguientes datos:

- Oficio DSC.
- Atendió.
- Zona.
- Área.
- Fecha.
- Persona a quien va dirigido.
- Asunto.
- Quien reporta.
- Causa.
- Tiempo.

III.2.2) El sistema muestra además los diferentes parámetros de búsqueda:

- Zona.
- Dirección IP.
- Causa.
- Reporta.
- Periodo.
- Año.
- Fecha.
- Área.

III.2.3) El usuario selecciona los criterios dentro de cada uno de los parámetros de búsqueda.

III.2.4) Si el usuario desea una búsqueda general, deberá poner la opción *TODOS* para cada criterio.


III.3) Si el usuario selecciona la opción “**Búsqueda de Oficios**”:

III.3.1) El sistema despliega por omisión todos los Oficios registrados con los siguientes datos:

- Oficio DSC.
- Oficio UNICA.
- Zona.
- Área.
- Fecha.
- Persona a quien va dirigido.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	C-INC
	Institución:	UNICA	Página:	5/5

<ul style="list-style-type: none"> ▪ Asunto. ▪ Quién Reporta. ▪ Causa. ▪ Respuesta. <p>III.3.2) El usuario selecciona los criterios dentro de cada uno de los parámetros de búsqueda.</p> <p>III.3.3) Si el usuario desea una búsqueda general, deberá poner la opción TODOS para cada criterio.</p> <p>IV) El sistema despliega una opción para Generar el Reporte en Excel de los resultados obtenidos.</p> <p>V) El sistema permite actualizar la información del registro de incidente habilitando el campo de Oficio DSC. (<i>Ver caso de uso: Actualizar Información del Incidente</i>).</p> <p>VI) Si el usuario desea realizar otra consulta, el caso de uso inicia en el punto II, de lo contrario el caso de uso termina.</p>
<p>Excepciones</p> <p>E-01→ Si el sistema no encuentra resultados para la búsqueda, informa de esto al usuario mostrando un mensaje.</p>
<p>Requisitos Especiales:</p> <p>R-01 -> El listado estará ordenado por nombres alfabéticamente.</p>
<p>Notas:</p> <p>N-01-> El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema (<i>Ver caso de uso: Acceder al Sistema</i>).</p> <p>N-02-> Estos criterios de búsqueda pueden ser combinados para realizar una consulta más específica.</p>

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-INC
	Institución:	UNICA	Página:	1/5


CASO DE USO

Id Caso de Uso:	07		
Nombre:	ACTUALIZAR INFORMACIÓN DEL INCIDENTE		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	04/Octubre/2007	Fecha actualización:	27/Agosto/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefes de los Departamentos: DID, DSA, DROS, DSC y Jefatura de UNICA.	X		Usuarios encargados de actualizar la información del incidente.

Descripción:
Permite actualizar la información de los incidentes registrados.
Precondiciones:
<p><i>Del proceso</i></p> <ul style="list-style-type: none"> • El actor debe contar con nombre de usuario y contraseña válidos para poder hacer uso del sistema (N-01). • El actor debe de realizar la búsqueda del incidente. (Ver caso de uso: Consultar Incidente) y seleccionar el campo correspondiente del incidente que desea actualizar.


Flujo Principal y Alternativos
<p>I) El caso de uso inicia cuando los actores ingresan al sistema para actualizar la información del incidente seleccionado desde la búsqueda realizada.</p> <p>II) Si el usuario que ingresa son los Jefes del DID, DROS, DSA y Jefatura de UNICA.</p> <p>II.1) En cualquiera de las opciones de búsqueda a las que haya ingresado el usuario (Ver caso de uso: Consultar Incidentes), el usuario selecciona el campo de Status (Atendido/No Atendido) del incidente que desea modificar.</p> <p>II.2) El sistema presenta una nueva ventana con un formulario, donde se muestra toda la información del incidente seleccionado [Becario, Categoría de apoyo, Tipo de apoyo, Tiempo estimado de respuesta, Status del incidente, Fecha de registro, Fecha en que se atendió el incidente, Descripción].</p>

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-INC
	Institución:	UNICA	Página:	2/5

El sistema presenta las siguiente opciones para que el usuario pueda modificar dicha información:

- **Becario.**
 - El sistema da una opción para consultar los nombres de becarios activos de UNICA.
 - El usuario selecciona la opción.
 - El sistema muestra en una sub-ventana el listado de todos los nombres de los becarios activos, así como el listado de los departamentos de UNICA.
 - El usuario puede localizar al becario en el listado completo por su nombre (**R-01**) o bien, buscarlo por el departamento al que pertenece.
 - Si el usuario selecciona algún departamento del listado, el sistema despliega los nombres de los becarios activos que pertenecen a dicho departamento.
 - El usuario selecciona el nombre del becario.
 - El sistema pone en los campos correspondientes de la ventana principal el nombre y apellidos del becario seleccionado.

- **Categoría del apoyo: Interno o Externo. (N-02).**
 - Si el usuario selecciona **Externo**, el sistema despliega un listado con el nombre de las Divisiones o Secretarías a las que se puede dar apoyo.
 - El usuario selecciona la División o Secretaría.
 - El sistema muestra un listado de la División o Secretaría, asociado con otro listado que contiene los nombres de las personas que laboran en dicho lugar.
 - Si la División o Secretaría a quien va dirigido el apoyo no se encuentra en la lista, el sistema permite registrar los datos seleccionando la opción de “*Agregar otro*”.
 - El sistema despliega en el mismo formulario los campos para agregar una nueva División o Secretaría; así como los campos para el nombre y apellidos de la persona a quien va dirigido el apoyo.
 - Si el nombre de la persona no se encuentra enlistado, el sistema permite registrar los datos seleccionando la opción de “*Agregar otro*”.
 - El sistema despliega en el mismo formulario los campos de nombre y apellidos.
 - Si el usuario selecciona **Interno**, el sistema oculta los campos para la categoría Externo (Ver categoría: Externo) y muestra un listado con los nombres de los departamentos de UNICA.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-INC
	Institución:	UNICA	Página:	3/5

- **Tipo de apoyo.**

- Si el tipo de apoyo no se encuentra enlistado, el sistema permite registrar los datos seleccionando la opción: “*Agregar otro*”.
 - El sistema despliega en el mismo formulario el campo para el registro del nuevo tipo de apoyo.

- **Tiempo estimado de respuesta en minutos, horas o días.**
- **Status del incidente (Atendido/No Atendido).**
- **Fecha en que se atendió el incidente (Si el estatus es Atendido).**
- **Fecha de registro.**
- **Descripción de la atención dada.**

III) Si el usuario que ingresa es el Jefe del DSC.

III.1) En cualquiera de las opciones de búsqueda a las que haya ingresado el usuario (***Ver caso de uso: Consultar Incidentes***), el usuario selecciona el campo de Oficio DSC del incidente que desea modificar.

III.1.1) Si lo que desea modificar es un Incidente:

El sistema despliega toda la información del incidente seleccionado:


III.1.2) Si lo que desea modificar es un Oficio, el sistema despliega toda la información del oficio seleccionado [Asunto, Dirección IP, Dirigido a, Reporta, Causa; Fecha, C.C.P]. El sistema presenta las siguiente opciones para que el usuario pueda modificar dicha información:

- **Asunto.**


- El sistema muestra un listado de los asuntos registrados anteriormente.
 - Si el asunto no está enlistado, el usuario selecciona la opción “*Agregar Otro*”.
 - El sistema despliega un campo para agregar el “*Nuevo Asunto*”.

- **Dirección IP.**


- El sistema muestra un listado con los primeros 3 octetos de la dirección IP donde se presenta el incidente y alado un campo para indicar el cuarto octeto.
- Si los tres primeros octetos no se encuentran enlistados, el usuario selecciona la opción “*Agregar Otro*”.
 - El sistema oculta el campo para el cuarto octeto y solicita la siguiente información para el Alta de la nueva IP:
 - IP.
 - Zona.

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-INC
	Institución:	UNICA	Página:	4/5

<ul style="list-style-type: none"> <ul style="list-style-type: none"> <ul style="list-style-type: none"> ▪ Si los primeros octetos si se encuentran enlistados, el usuario introduce el último octeto de la dirección IP. • Dirigido a. <ul style="list-style-type: none"> ▪ El sistema muestra un listado de las personas que solicitan la atención del Incidente. <ul style="list-style-type: none"> ▪ Si selecciona una persona se muestra en un listado el área asociada a la persona. <ul style="list-style-type: none"> • Es posible cambiar el área o agregar una nueva seleccionando la opción “Agregar Otro”. <ul style="list-style-type: none"> ○ El sistema solicita la siguiente información para la “Nueva Área”. <ul style="list-style-type: none"> ▪ Nombre completo del área. ▪ Siglas. ▪ Si el nombre de la persona no está enlistado, el usuario selecciona la opción “Agregar Otro”. ▪ El sistema solicita la siguiente información de la “Nueva persona”: <ul style="list-style-type: none"> • Nombre. • Apellidos. • Responsable. • Área. <ul style="list-style-type: none"> ○ Si el área no está enlistada, el usuario selecciona la opción “Agregar Otro”. ○ El sistema solicita la siguiente información para la “Nueva Área”. <ul style="list-style-type: none"> ▪ Nombre completo del área. ▪ Siglas. • Reporta. <ul style="list-style-type: none"> ▪ El sistema muestra un listado de las instituciones que pueden reportar los incidentes (DGSCA, DSC). • Causa. <ul style="list-style-type: none"> ▪ El sistema despliega un listado con las causas registradas. ▪ Si la causa no se encuentra enlistada, el usuario selecciona la opción “Agregar Otro”. <ul style="list-style-type: none"> • El sistema solicita la siguiente información para el Alta de la nueva causa: <ul style="list-style-type: none"> ○ Nombre de la causa. ○ Descripción. • Fecha. <ul style="list-style-type: none"> ▪ El sistema despliega un calendario para que el usuario seleccione la fecha.
--

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	A-INC
	Institución:	UNICA	Página:	5/5

<ul style="list-style-type: none"> ● Con copia para (C.C.P.). <ul style="list-style-type: none"> ▪ El sistema despliega un listado con los nombres de las personas que pueden recibir copia del oficio. ▪ El usuario puede seleccionar a una o más personas. ▪ Si el nombre de la persona no se encuentra enlistada, el sistema permite registrar los datos seleccionando la opción “Agregar otro”. <ul style="list-style-type: none"> ● El sistema despliega una sub-ventana con los campos para agregar el nombre, apellidos y responsabilidad de la nueva persona a registrar. <p>IV) El usuario Modifica los datos deseados y acepta el registro (E-01).</p> <p>V) El sistema despliega un mensaje de que la información se actualizó satisfactoriamente.</p> <p>VI) Si el usuario desea actualizar otro incidente el caso de uso inicia en el punto I de lo contrario, el caso de uso termina.</p>
Excepciones
E-01 → En caso de que no se actualice el registro, el sistema informa sobre el problema o se indica si algún dato no es válido.
Requisitos Especiales:
R-01 → El listado estará ordenado por nombres alfabéticamente.
Notas:
N-01 → El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema (Ver caso de uso: Acceder al Sistema).
N-02 → Pertenecen a la categoría externa todos aquellos incidentes que son presentados en instituciones o entidades que no pertenecen a UNICA.


	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	G-EST
	Institución:	UNICA	Página:	1/2

CASO DE USO

Id Caso de Uso:	08		
Nombre:	GENERAR ESTADÍSTICAS		
Creado por:	Antonio Carrizosa Martínez	Última actualización por:	Antonio Carrizosa Martínez
Fecha de creación:	04/Diciembre/2007	Fecha actualización:	30/Septiembre/2009

Actores:			
Rol	Participación		Descripción Participación / Intereses
	Directa	Indirecta	
Jefe del Departamento de Seguridad en Cómputo (DSC).	X		Usuario autorizado de consultar y generar las estadísticas de la atención a los incidentes registrados en el sistema.

Descripción:
Permite realizar las estadísticas de los incidentes atendidos por área, año, causa, quién reporta, zona, por la combinación de estos dos últimos (Quién reporta y Zona) o por la combinación de Todos los parámetros anteriores.
Precondiciones:
<i>Del proceso</i> El actor debe contar con nombre de usuario y contraseña válidos para poder hacer uso del sistema (N-01) .
Flujo Principal y Alternativos
<p>I) El caso de uso inicia cuando el Jefe del Departamento de Seguridad en Cómputo ingresa al sistema para consultar las estadísticas de los incidentes de seguridad en cómputo.</p> <p>II) El usuario selecciona del menú principal la opción "Estadísticas".</p> <p>III) El sistema despliega una ventana para generar las estadísticas con las siguientes opciones:</p> <ul style="list-style-type: none"> ✓ Por Área ✓ Por Año ✓ Por Causa ✓ Por Quién reporta ✓ Por Quién reporta y Zona ✓ Por Zona ✓ Todos

	Nombre del proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	G-EST
	Institución:	UNICA	Página:	2/2

Además, el sistema muestra un campo para indicar el año en que se desea generar las estadísticas.

IV) El usuario selecciona el parámetro de búsqueda e ingresa el año al que desea aplicar el parámetro.

V) El sistema genera la(s) gráfica(s) con la información correspondiente **(E-01)**.

V.1) Si el usuario genera estadísticas por zona, área, y causa del reporte, el sistema despliega una gráfica de barras del número de incidentes que ocurrieron en cada una respectivamente.

V.2) Si el usuario genera estadísticas por quién reporta y zona, el sistema despliega una gráfica de barras, indicando el número de incidentes en cada zona reportados por DGSCA y DSC.

V.3) Si el usuario genera estadísticas por año, el sistema despliega una gráfica de pastel con el porcentaje de los incidentes en cada mes.

V.4) Si el usuario genera estadísticas por la opción “todos”, el sistema despliega todas las gráficas e información en la misma página.

VI) El sistema despliega una opción para Guardar las Estadísticas en formato RTF.

VII) Si el usuario desea generar otra estadística, el caso de uso inicia en el punto **II**, de lo contrario el caso de uso termina.

Excepciones

E-01 → Si no se encuentran registros, el sistema muestra un mensaje informando esto al usuario.

Notas:

N-01 → El nombre de usuario y contraseña debe de ser proporcionado por el administrador principal del sistema (**Ver caso de uso: Acceder al Sistema**).

III.3. DEFINICIÓN DE ACTORES

Actor:	JEFATURA DE LA UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO (UNICA)
Descripción:	Supervisar el desarrollo de proyectos y actividades analizando con los jefes de departamentos los planes de trabajo e informes correspondientes.
Casos de Uso donde participa:	Registrar Incidentes Actualizar información del Incidente Consultar Incidente

Actor:	JEFATURA DE LA DIRECCIÓN DE SEGURIDAD EN CÓMPUTO (DSC)
Descripción:	Usuario del sistema, encargado de brindar y dar seguimiento a todas las actividades de seguridad en cómputo dentro de la Facultad de Ingeniería.
Casos de Uso donde participa:	Registrar Incidentes Actualizar información del Incidente Consultar Incidente Generar Estadísticas Generar Oficio

Actor:	JEFATURA DE LA DIRECCIÓN DE SERVICIOS ACADÉMICOS (DSA)
Descripción	Usuario del sistema, se encarga de la formación de los recursos humanos que la Unidad necesita para el cumplimiento de sus funciones.
Casos de Uso donde participa:	Registrar Incidentes Actualizar información del Incidente Consultar Incidente

Actor:	JEFATURA DE LA DIRECCIÓN DE INVESTIGACIÓN Y DESARROLLO (DID)
Descripción	Usuario del sistema, responsable de las actividades de desarrollo de sistemas de información, administración y control de apoyo a las actividades académico-administrativas.
Casos de Uso donde participa:	Registrar Incidentes Actualizar información del Incidente Consultar Incidente

Actor:	JEFATURA DE LA DIRECCIÓN DE REDES Y OPERACIÓN DE SERVIDORES (DROS)
Descripción	Usuario del sistema, responsable de administrar, operar, mantener y dar seguridad a la red de la Facultad.
Casos de Uso donde participa:	Registrar Incidentes Actualizar información del Incidente Consultar Incidente

III.4. DIAGRAMAS DEL SISTEMA CON UML

III.4.1. DIAGRAMA DE CASOS DE USO

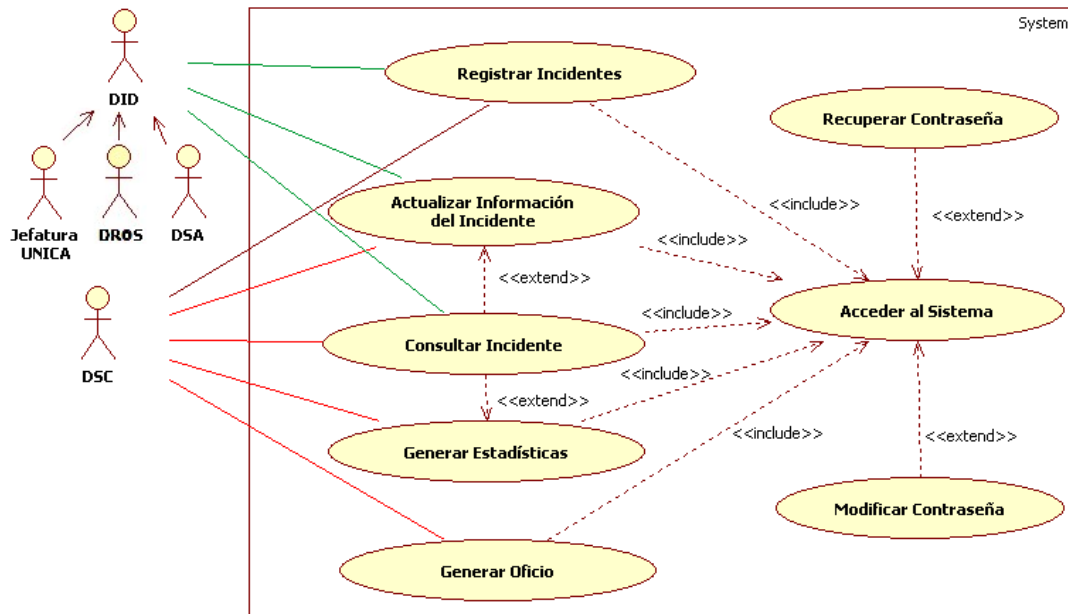


Figura 3.1. Diagrama de Casos de Uso del SAACI.

III.4.2. DIAGRAMA DE ACTIVIDADES

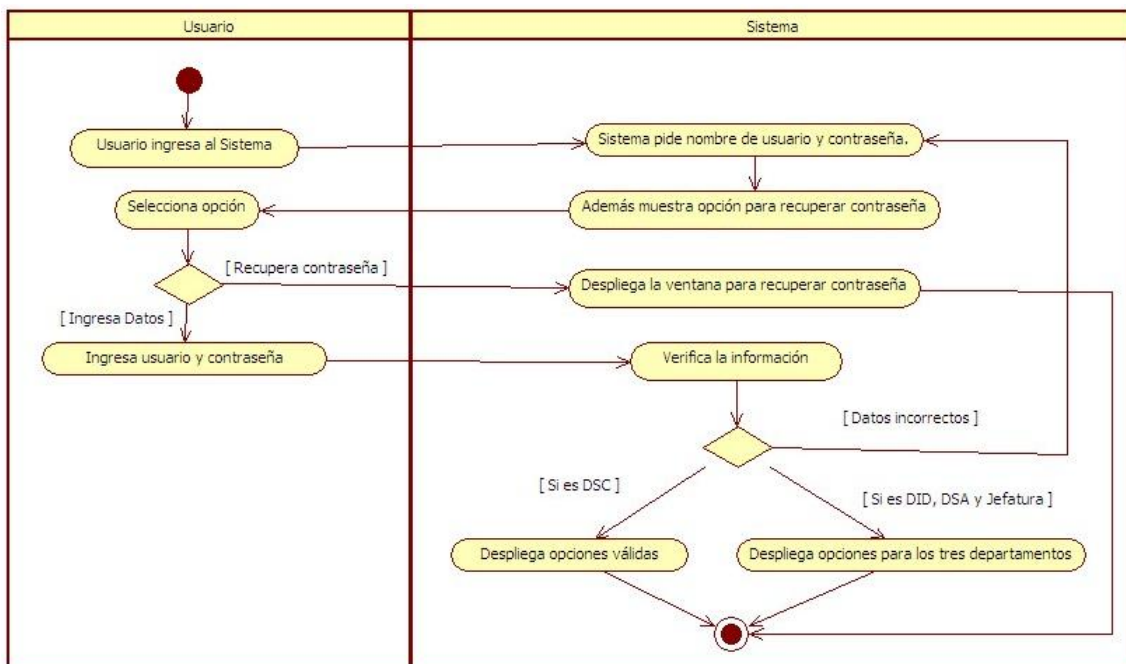


Figura 3.2. Diagrama de Actividades para Acceder al Sistema.

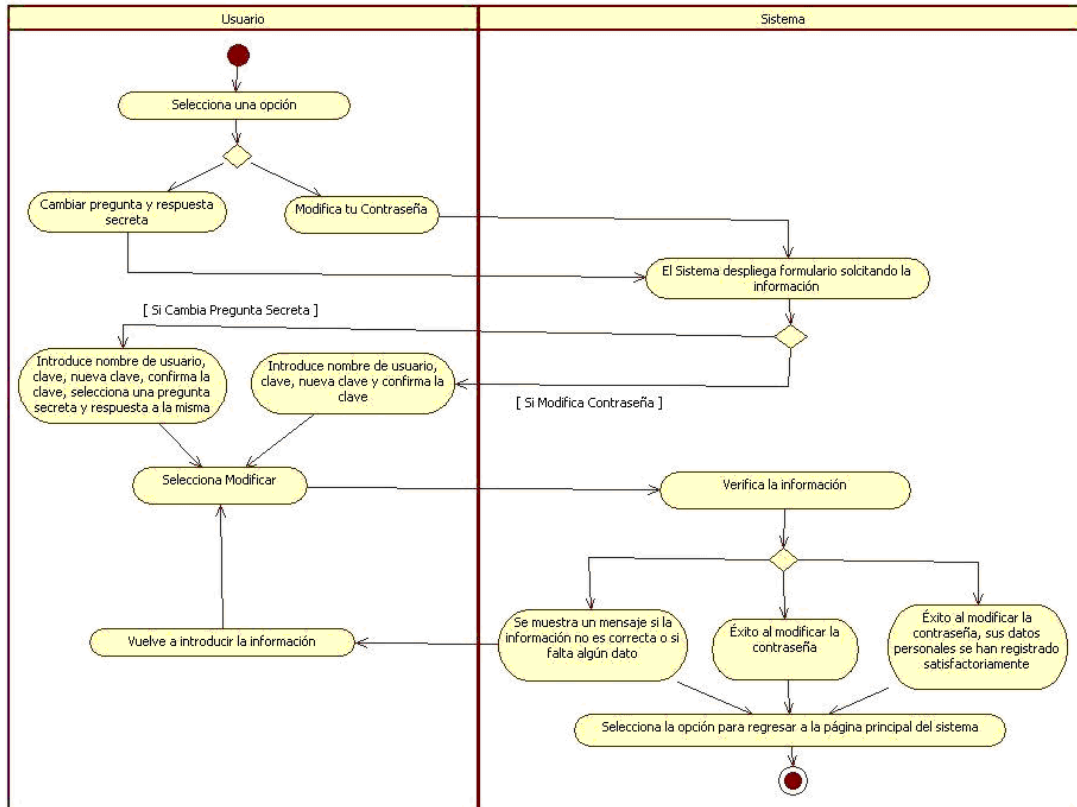


Figura 3.3. Diagrama de Actividades para Modificar la Contraseña del Sistema.

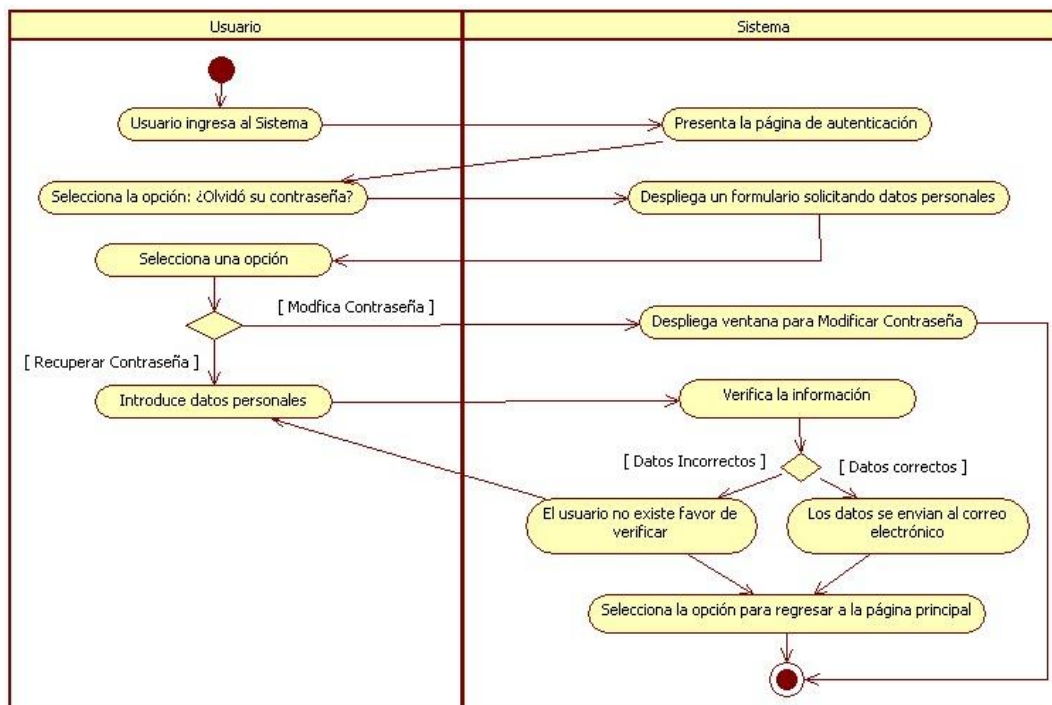


Figura 3.4. Diagrama de Actividades para Recuperar la Contraseña del Sistema.

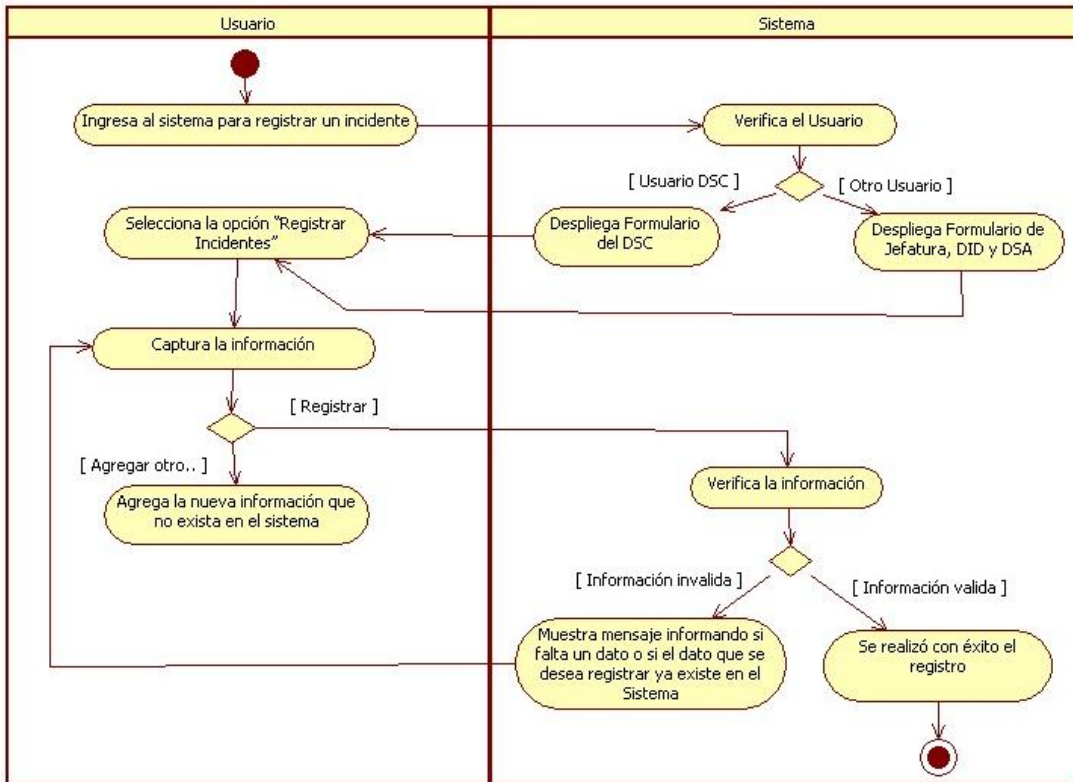


Figura 3.5. Diagrama de Actividades para Registrar Incidentes en el Sistema.

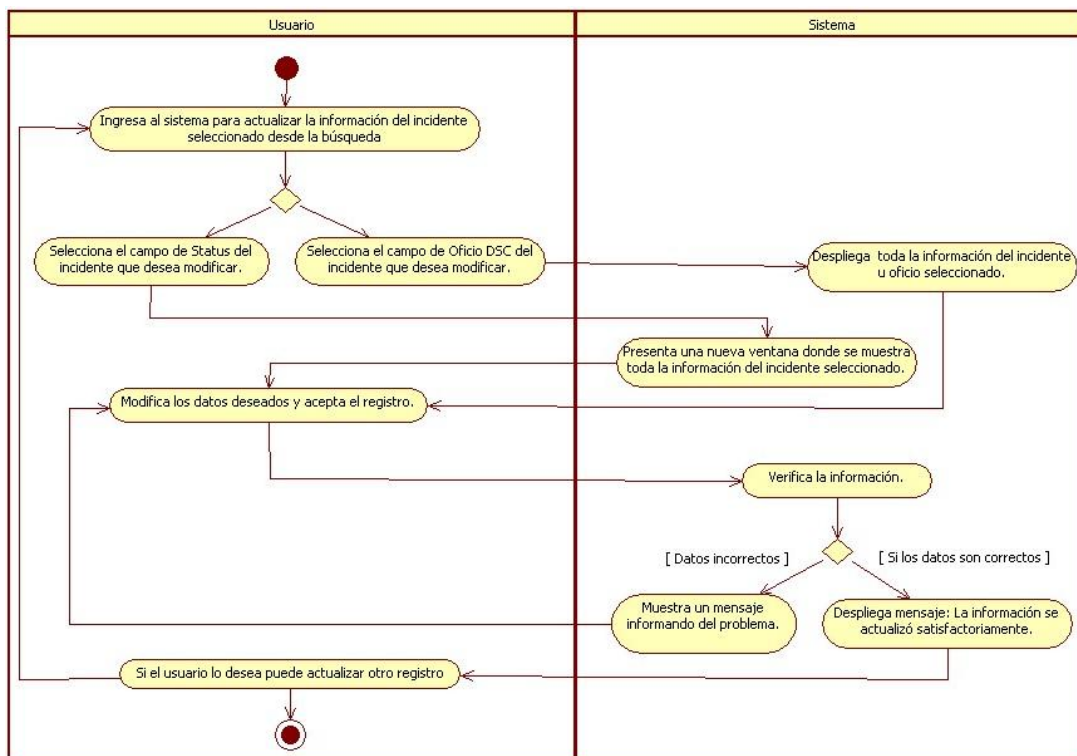


Figura 3.6. Diagrama de Actividades para Actualizar la información.

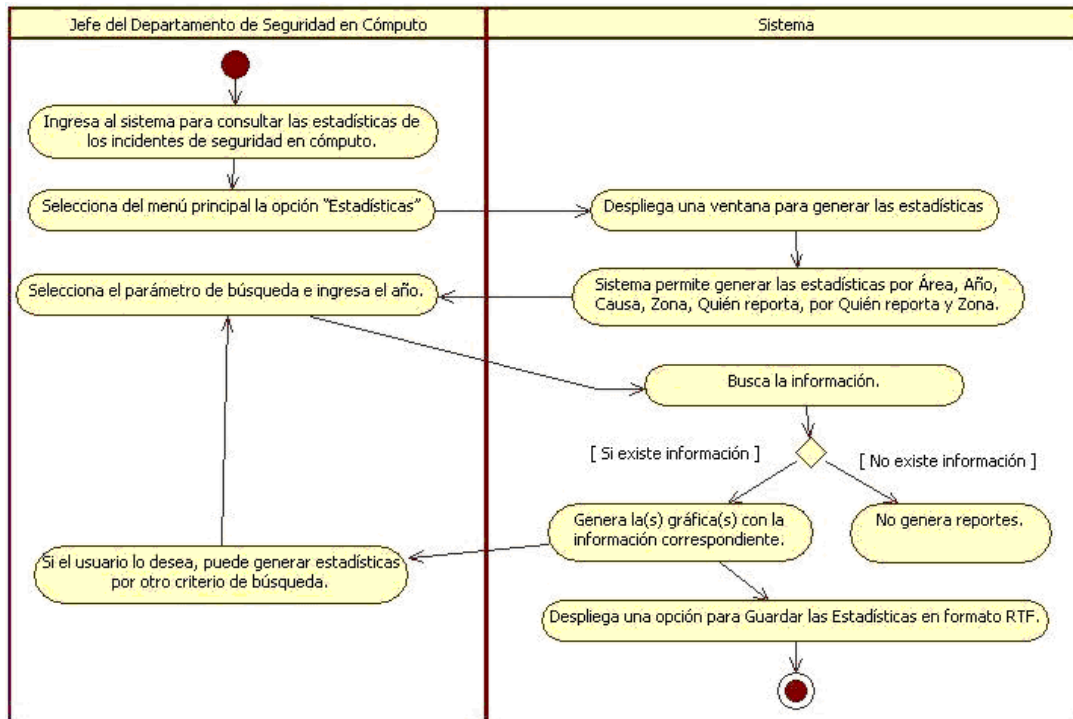


Figura 3.7. Diagrama de Actividades para Generar Estadísticas.

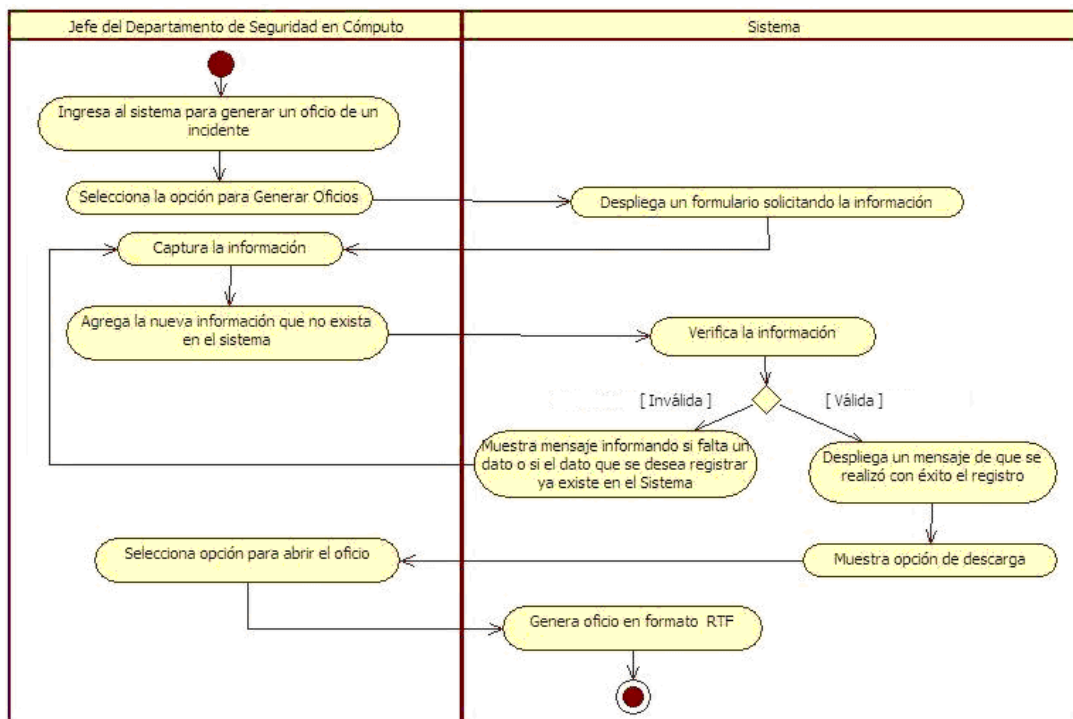


Figura 3.8. Diagrama de Actividades para Generar un Oficio.

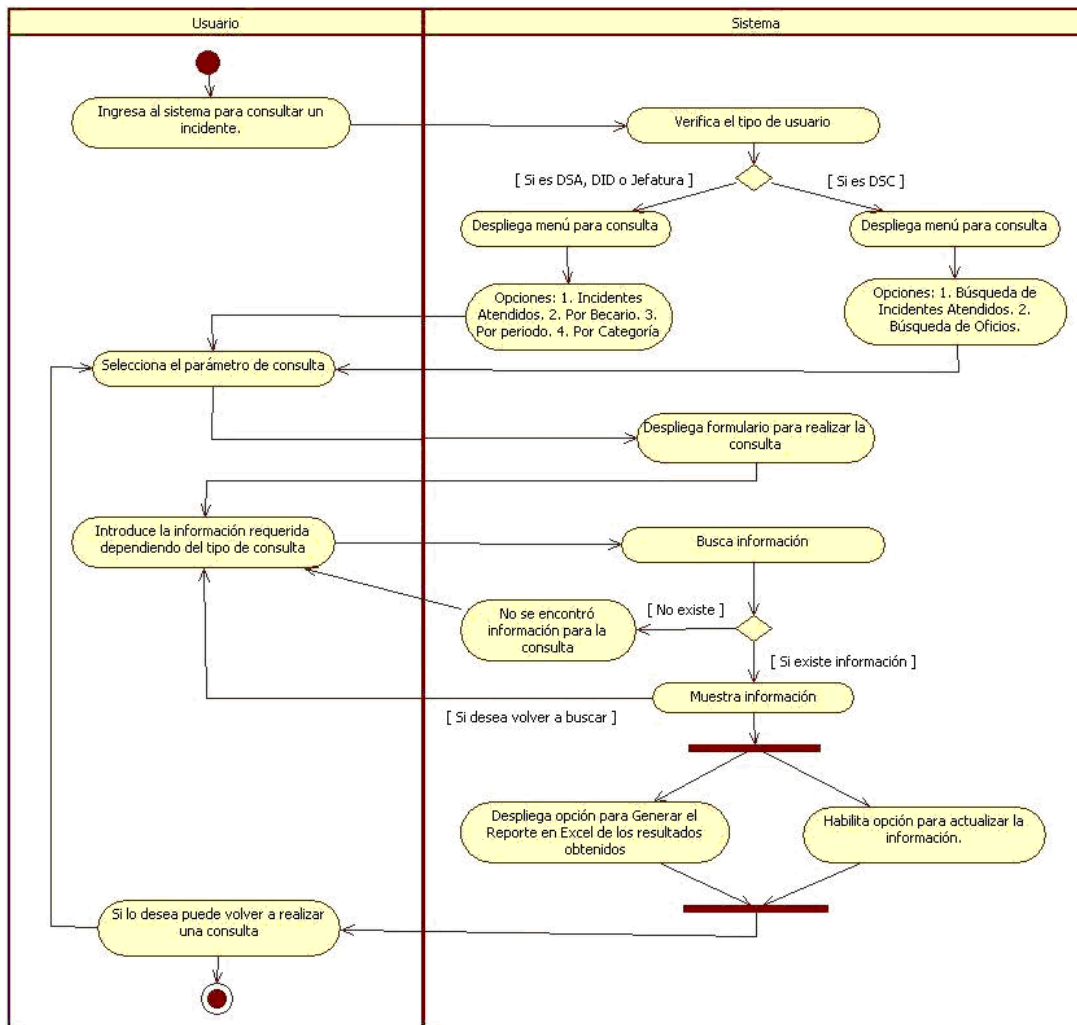


Figura 3.9. Diagrama de Actividades para Consultar la Información del Sistema.

III.4.3. DIAGRAMAS DE SECUENCIA

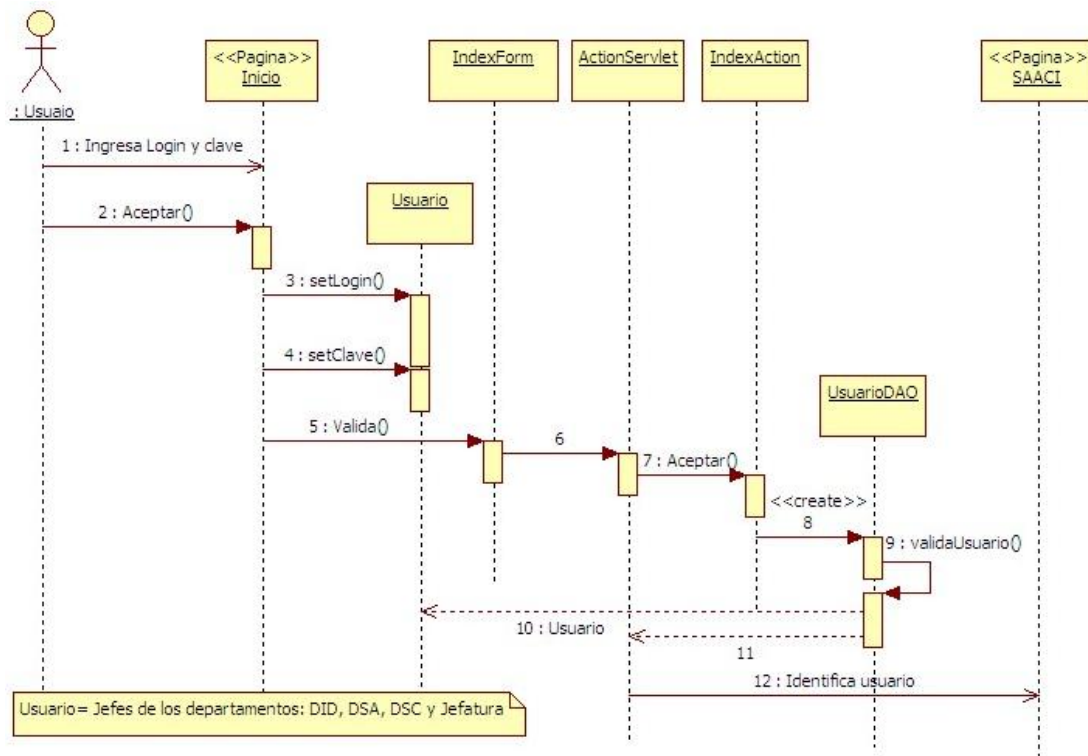


Figura 3.10. Diagrama de Secuencia para Ingresar al Sistema.

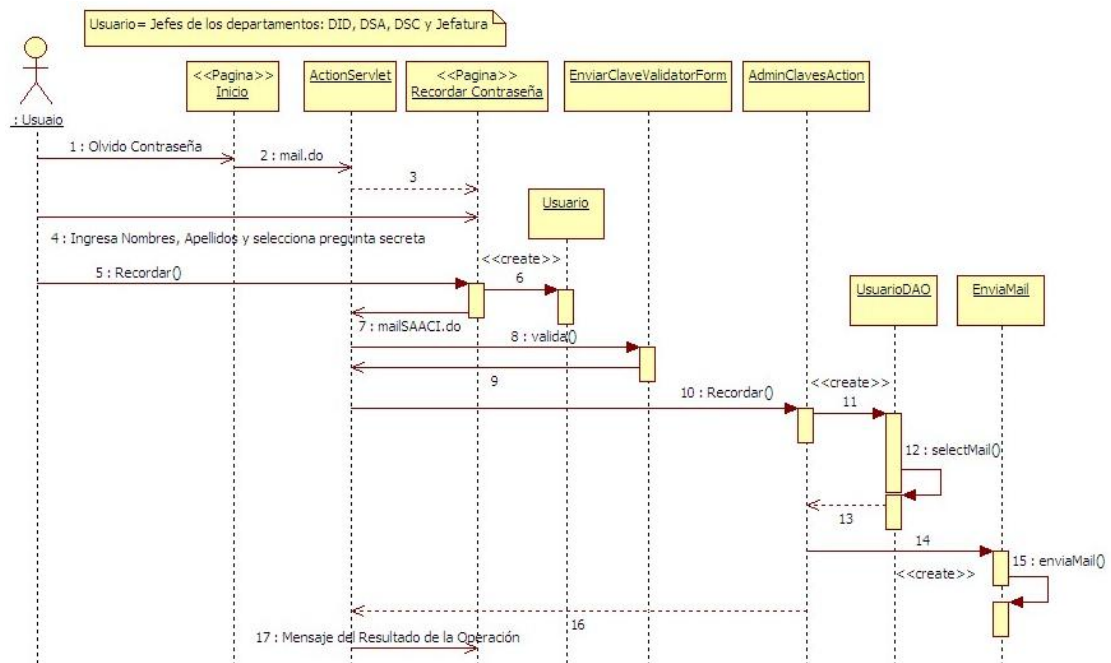


Figura 3.11. Diagrama de Secuencia para Ingresar al Sistema

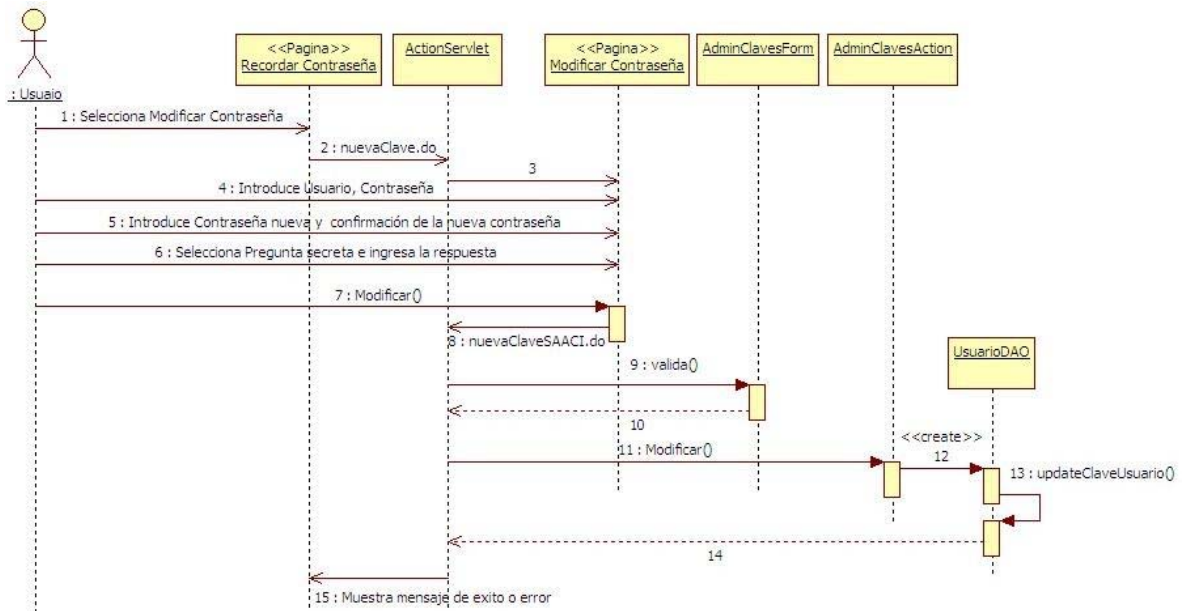


Figura 3.12. Diagrama de Secuencia para Modificar la Contraseña del Sistema.

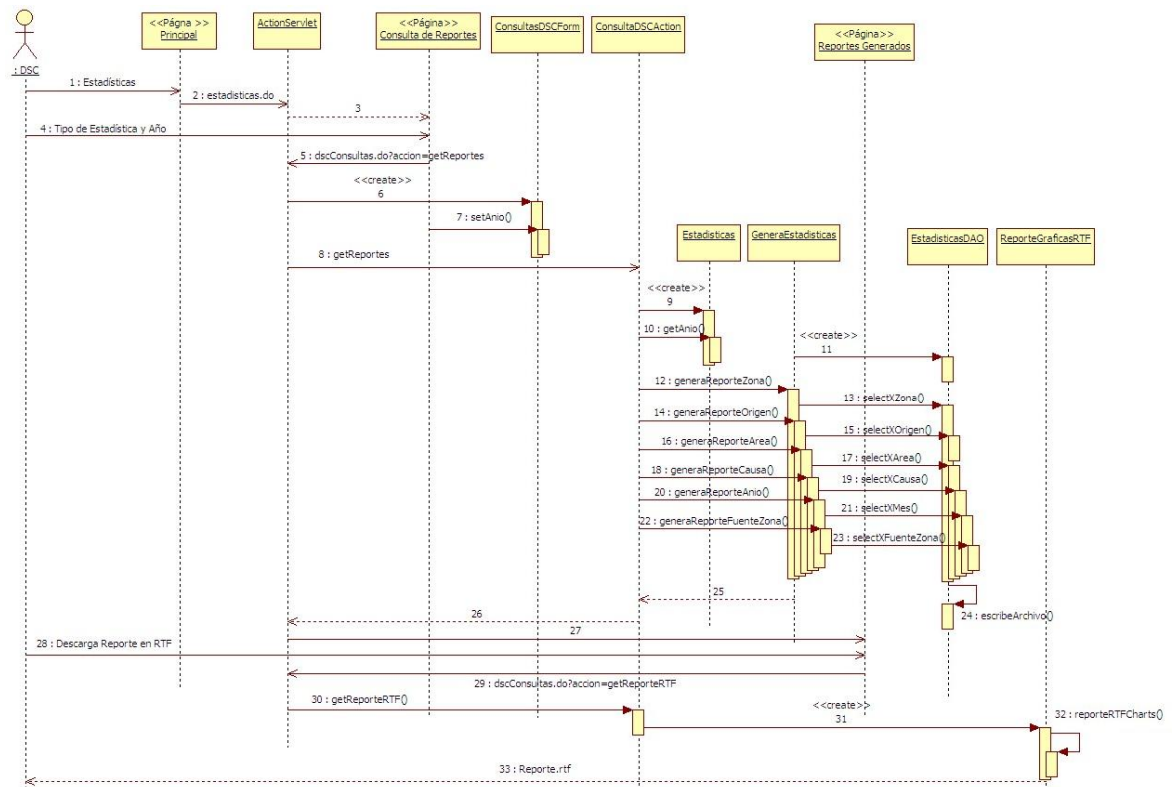


Figura 3.13. Diagrama de Secuencia para Generar Estadísticas.

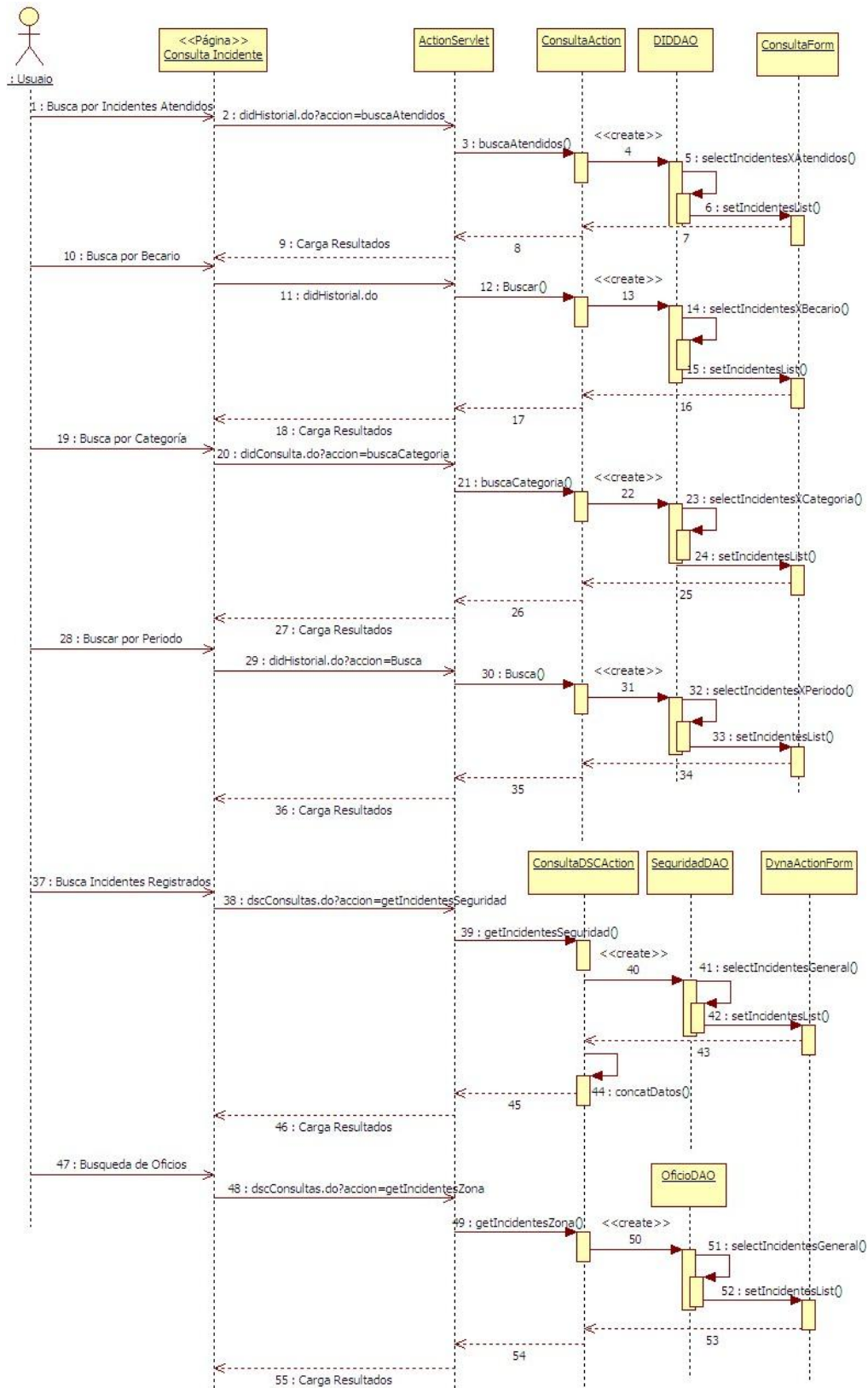


Figura 3.14. Diagrama de Secuencia para Consultar la Información del Sistema.

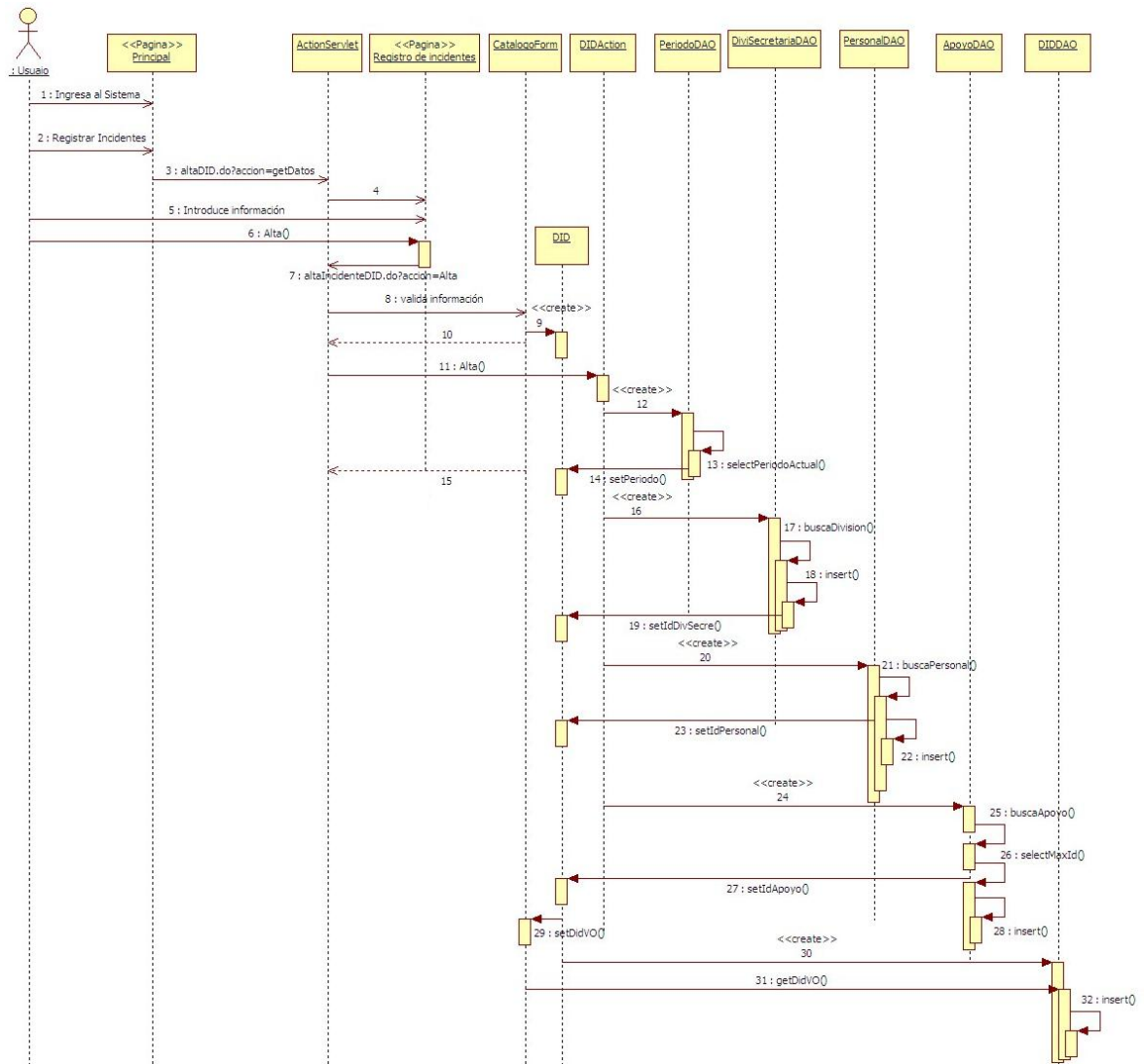


Figura 3.15. Diagrama de Secuencia para Registrar la Información de los incidentes.

III.4.4. DIAGRAMA DE CLASES

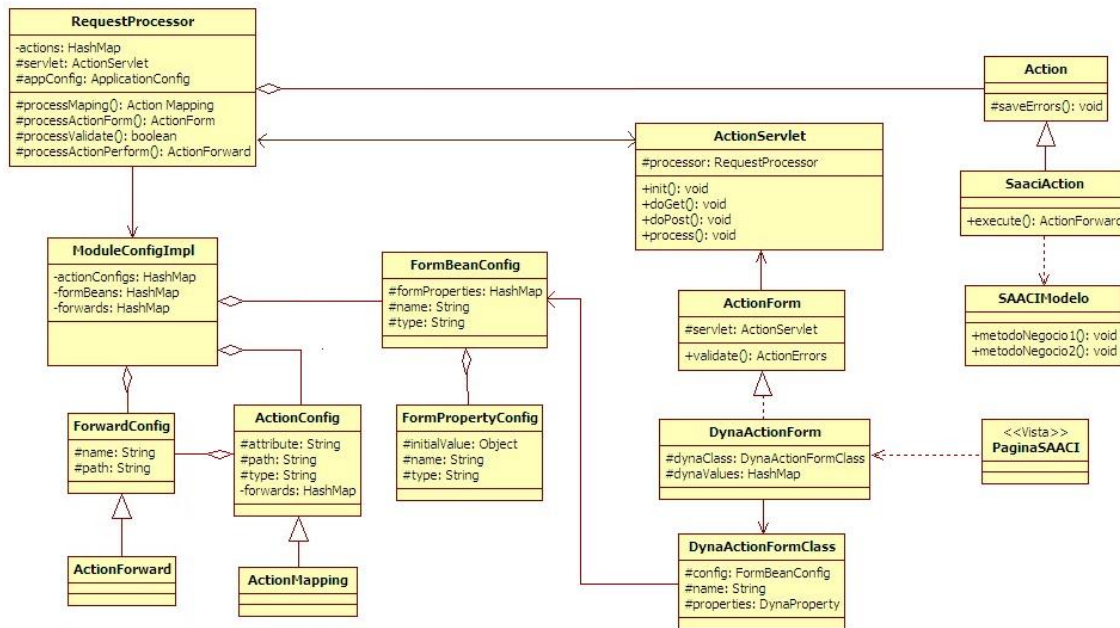


Figura 3.18. Diagrama de Clases del SAACI utilizando Struts.

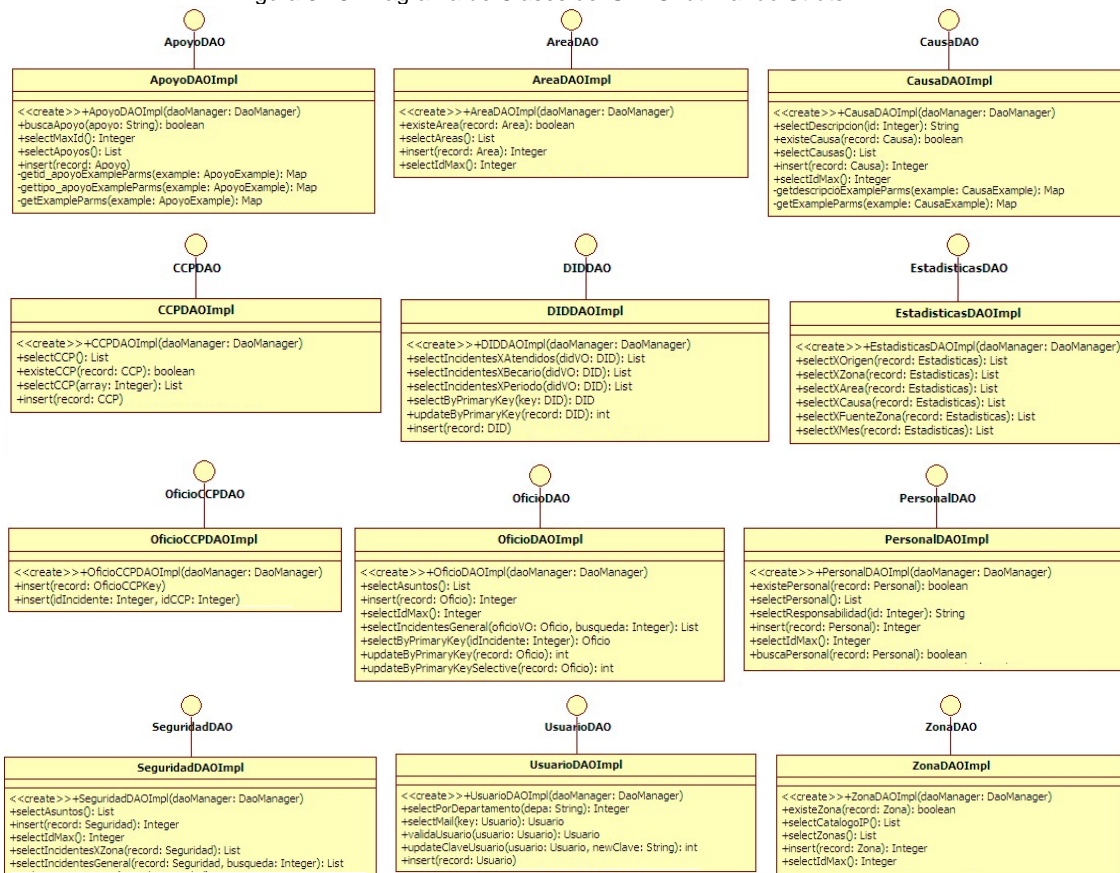


Figura 3.19. Principales Clases para el Manejo de la Persistencia.

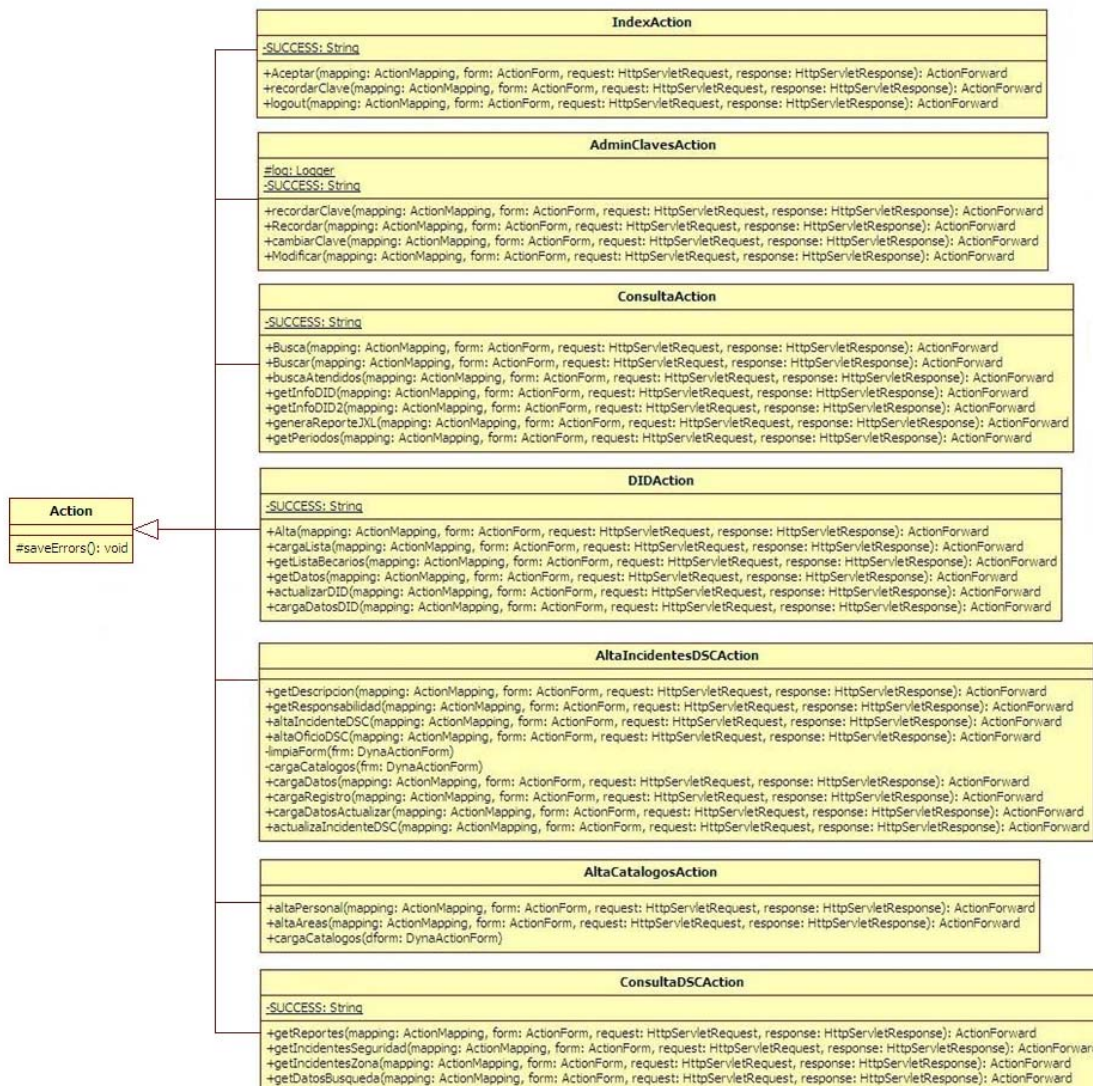


Figura 3.20. Principales Clases Action del Sistema.

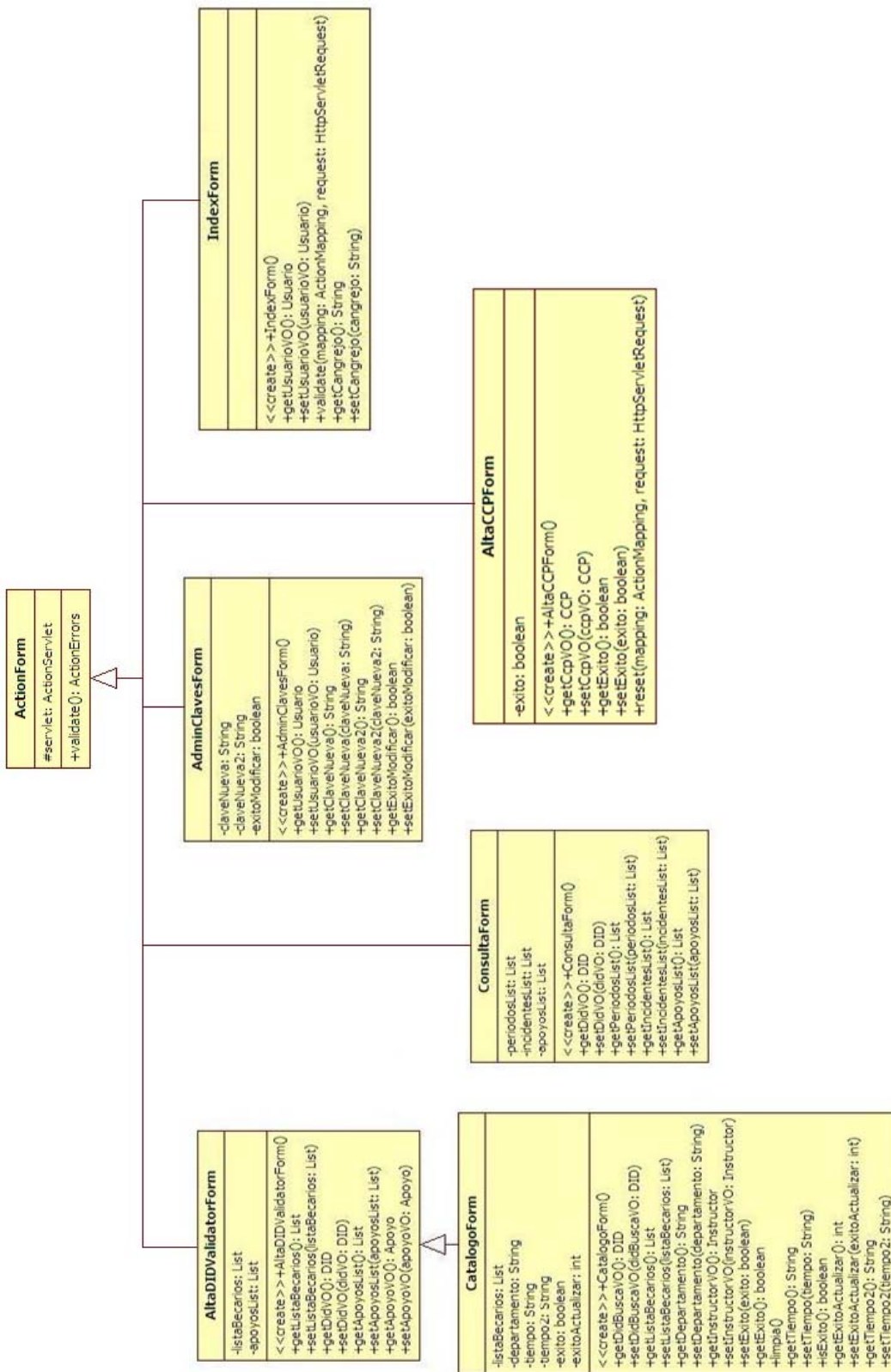


Figura 3.21. Principales Clases ActionForm del Sistema.

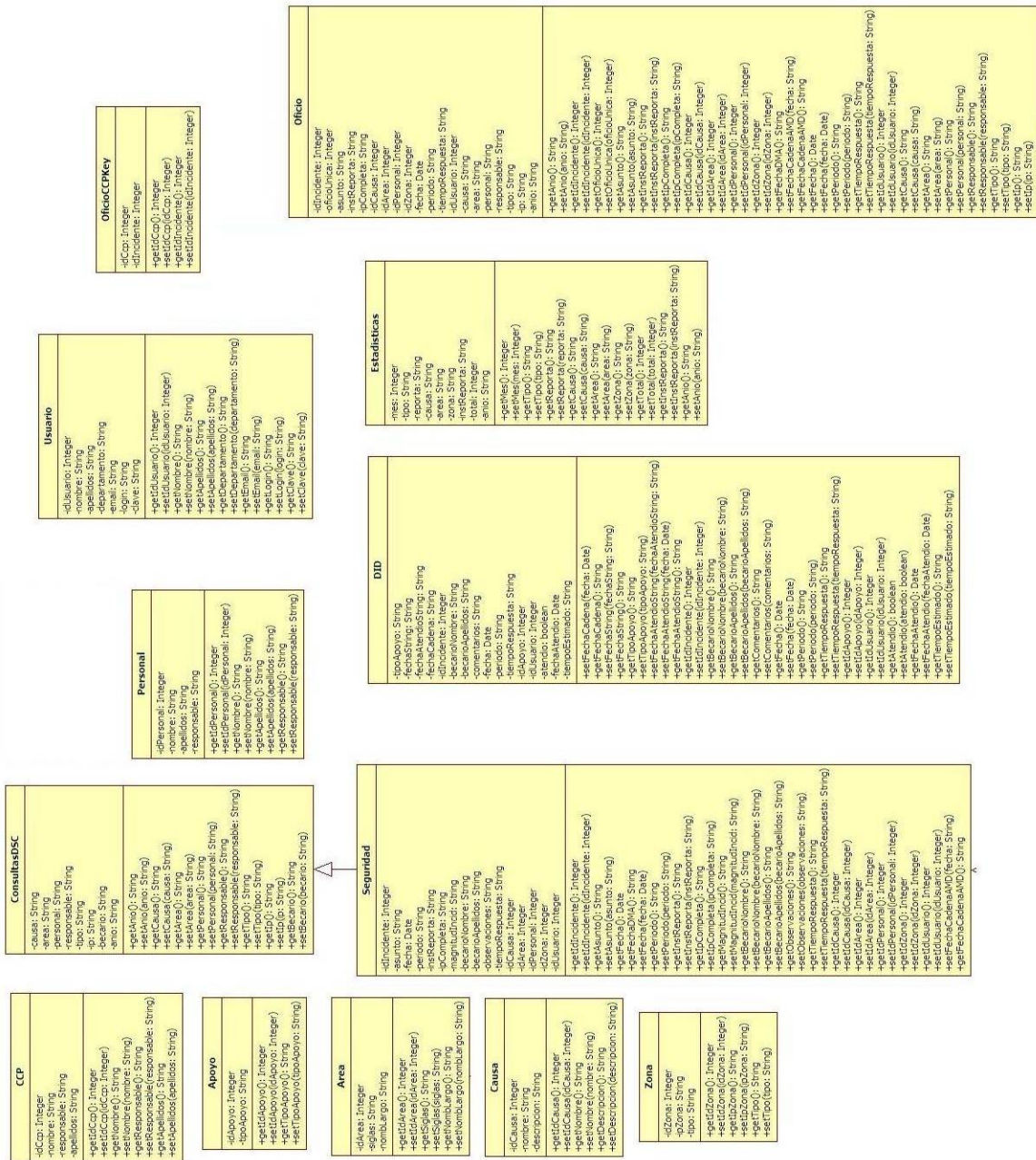


Figura 3.22. Principales Clases “Java Bean” del Sistema.

III.4.5. DIAGRAMA DE COMPONENTES

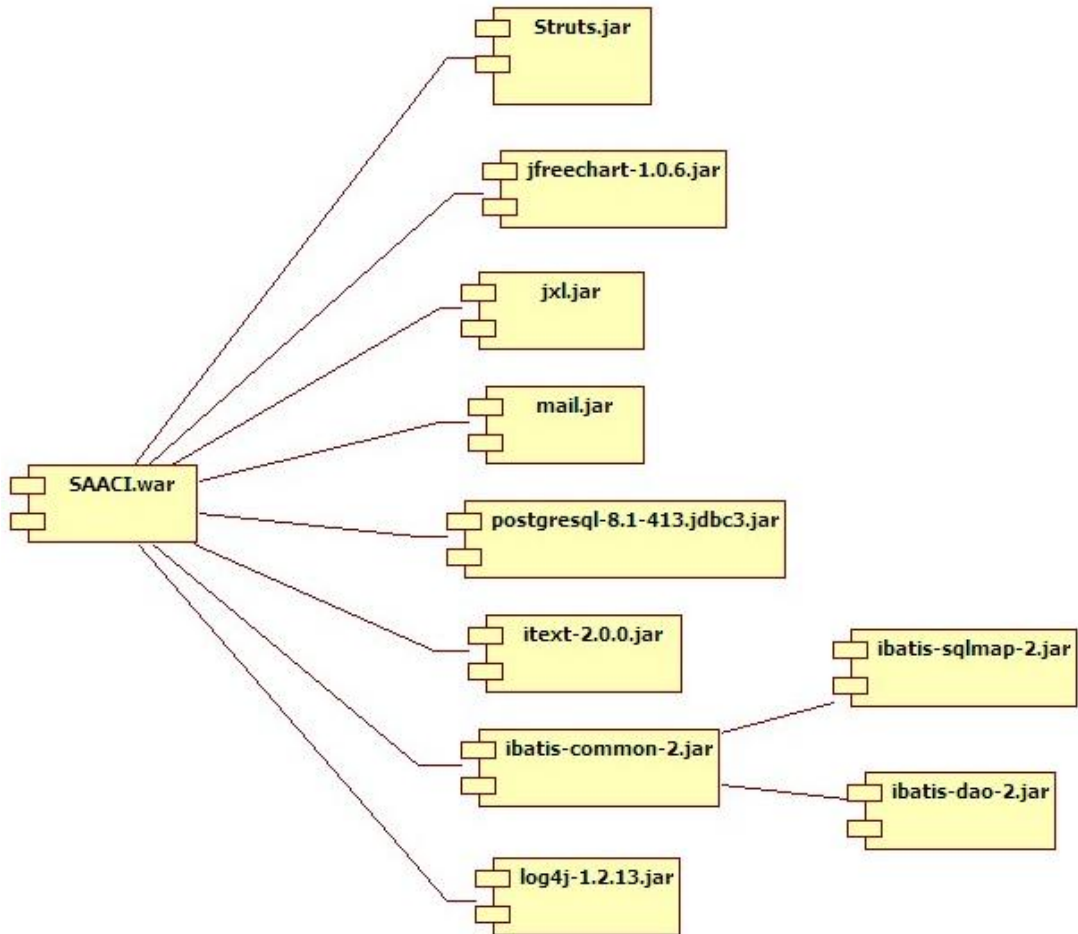
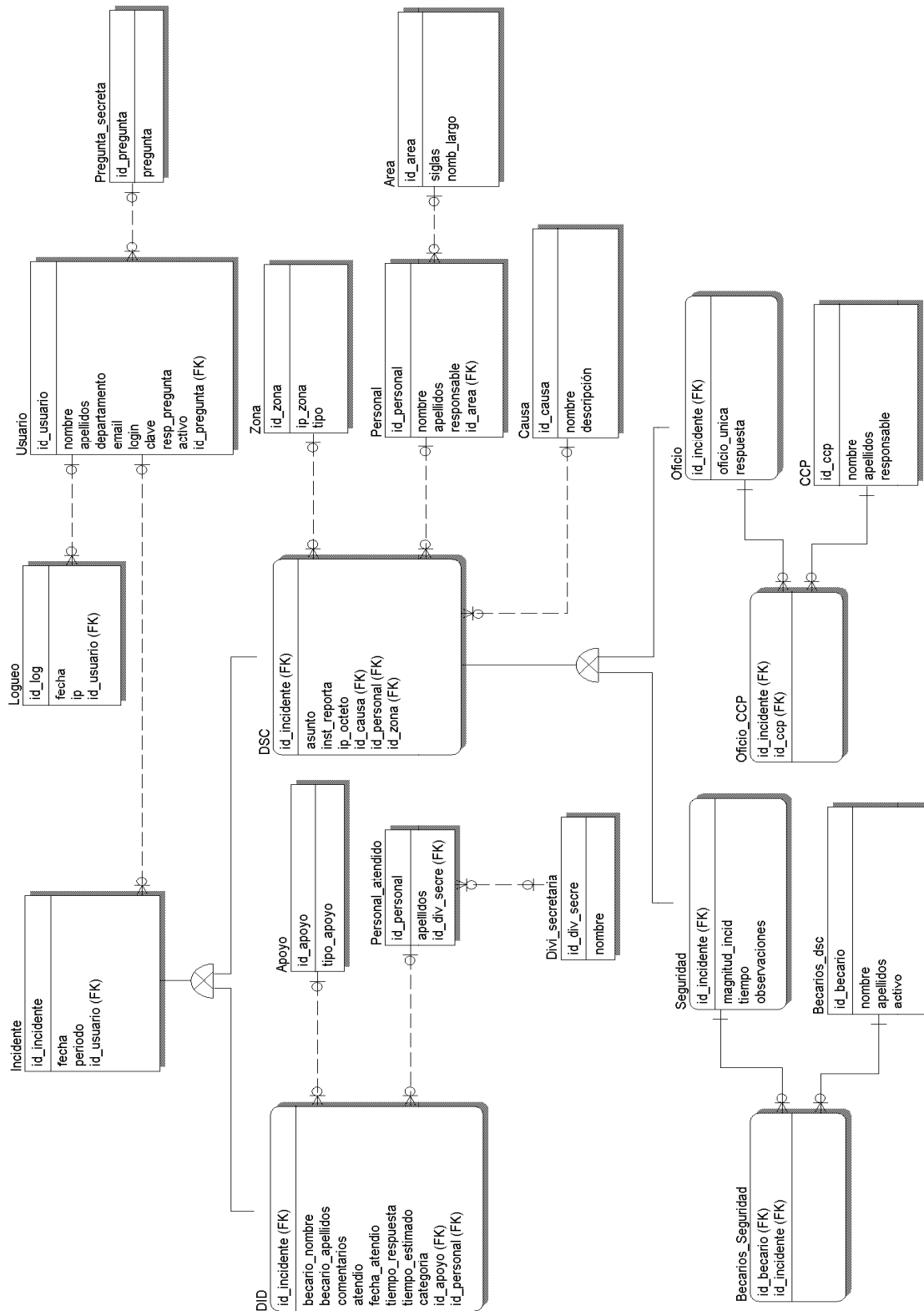


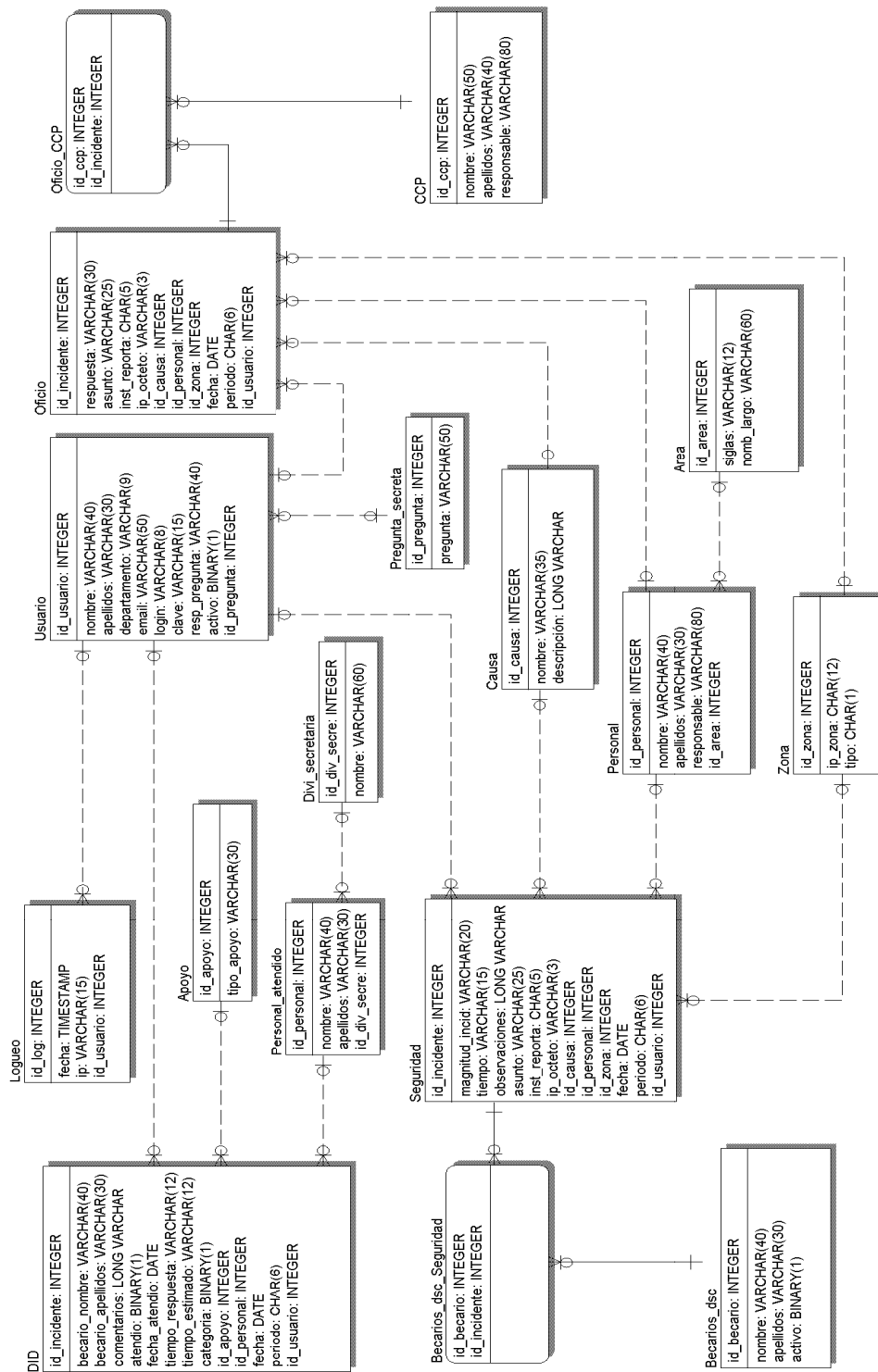
Figura 3.23. Principales Componentes que conforman al Sistema.

III.5. DIAGRAMA LÓGICO DE DATOS



III.6. DISEÑO DEL SISTEMA

III.6.1. DIAGRAMA FÍSICO DE DATOS (DER)



III.6.2. DICCIONARIO DE DATOS PARA EL DER

DICCIONARIO DE DATOS
 UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO
SISTEMA PARA LA ADMINISTRACIÓN, ATENCIÓN Y CONTROL DE INCIDENTES
 (SAACI)

NOMBRE DE LA TABLA: USUARIO

COLUMNA	DESCRIPCIÓN	TIPO DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_usuario	Número único identificador del usuario del sistema.	Numérico		X	X		Incremento automático
nombre	Nombre del usuario.	caracter variable	40	X			
apellidos	Apellido(s) del usuario.	caracter variable	40	X			
departamento	Departamento de UNICA en el que labora el usuario del sistema.	caracter variable	9	X			
email	Correo electrónico del usuario del sistema.	caracter variable	50	X			
login	Clave con la que el usuario se autentica ante el sistema.	caracter variable	8	X			
clave	Contraseña del usuario.	caracter variable	15	X			
id_pregunta	Número único identificador de la pregunta secreta que eligió el usuario al modificar su contraseña.	Numérico		X			
resp_pregunta	Respuesta del usuario a la pregunta secreta que eligió.	caracter variable	40	X			
activo	Estado de la cuenta del usuario: True = Activo, False = Inactivo	Boolean		X			True

NOMBRE DE LA TABLA: DID

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
Id_incidente	Número único identificador del incidente.	numérico (entero)		X	X		Incremento automático
becario_nombre	Nombre del becario a cargo del incidente.	caracter variable	40	X			
becario_apellidos	Apellido(s) del becario a cargo del incidente.	caracter variable	40	X			
comentarios	Comentarios /observaciones acerca de la atención del incidente.	Texto					
atendio	Estado en el que se encuentra el incidente: Atendido = True, No atendido = False.	Boolean					False
fecha_atendio	Fecha real en la que se atendió el incidente.	Fecha					
tiempo_respuesta	Tiempo real de atención al incidente.	caracter variable	12				
tiempo_estimado	Tiempo estimado de atención al incidente.	caracter variable	12				
categoria	Categoría del incidente (Interno= True, Externo= False).	Boolean					False
fecha	Fecha de registro del incidente en el sistema.	Fecha		X			
periodo	Periodo o semestre en el que se atendió el incidente.	caracter	6	X			
id_apoyo	Núm. único identificador del tipo de apoyo que se da (Seguridad, Administrativos, Especial, etc.)	numérico (entero)		X		APOYO (id_apoyo)	
id_personal	Núm. único identificador del personal que es atendido.	numérico (entero)				PERSONAL_ ATENDIDO (id_personal)	
id_usuario	Núm. identificador del usuario del sistema que realiza el registro del incidente.	numérico (entero)		X		USUARIO (id_usuario)	

NOMBRE DE LA TABLA: PERSONAL_ATENDIDO

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_personal	Número único identificador del personal al que se le atiende.	numérico		X	X		Incremento automático
nombre	Nombre del personal al que se le atiende.	caracter variable	40	X			
apellidos	Apellidos del personal al que se le atiende.	caracter variable	40	X			
id_div_secre	Número único identificador de la División o Secretaría a la que pertenece el personal atendido.	numérico		X			

NOMBRE DE LA TABLA: DIVI_SECRETARIA

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_div_secre	Número único identificador de la División o Secretaría a la que pertenece el personal atendido.	numérico (entero)		X	X		Incremento automático
nombre	Nombre de la División o Secretaría a la que pertenece el personal atendido.	caracter variable	60	X			

NOMBRE DE LA TABLA: APOYO

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_apoyo	Número único identificador del apoyo.	numérico (entero)		X	X		Incremento automático
tipo_apoyo	Tipo de apoyo que se puede dar a un incidente (Administrativo, Seguridad, Soporte, Especial, etc.)	caracter variable	30	X			

NOMBRE DE LA TABLA: SEGURIDAD

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_incidente	Número único identificador del incidente.	numérico (entero)		X	X		Incremento automático
magnitud_incid	Magnitud del incidente (Moderado, Importante o Crítico).	caracter variable	20	X			
tiempo	Tiempo real de atención del incidente.	caracter variable	15	X			
observaciones	Comentarios/ observaciones acerca de la atención del incidente.	Text		X			
asunto	Asunto por el cual se realiza la atención al incidente(Incidente en Seguridad).	caracter variable	25	X			
inst_reporta	Institución que reporta el incidente (DGSCA, UNICA).	caracter	5	X			
ip_octeto	Ultimo octeto de la IP de la máquina comprometida.	caracter variable	3	X			
fecha	Fecha en la que se registró el incidente.	Fecha		X			
id_causa	Número único identificador de la causa del incidente.	numérico (entero)		X		CAUSA (id_causa)	
id_personal	Número único identificador del personal atendido.	numérico (entero)		X		PERSONAL id_personal	
id_zona	Número único identificador de la zona donde se presentó el incidente.	numérico (entero)		X		ZONA (id_zona)	
id_usuario	Número único identificador del usuario que realiza el registro del incidente.	numérico (entero)		X		USUARIO (t_usuario)	

NOMBRE DE LA TABLA: CAUSA

COLUMNA	DESCRIPCIÓN	TIPO DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_causa	Número único identificador de la causa que ocasionó el incidente.	Numérico (entero)		X	X		Incremento automático
nombre	Nombre de la causa que ocasionó el incidente (spam, virus y gusanos, violación a la propiedad intelectual, phishing, actividad intrusiva, cambio de infraestructura de red).	caracter variable	35	X			
descripción	Descripción de la causa.	texto		X			

NOMBRE DE LA TABLA: ZONA

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_zona	Número único identificador de la zona donde se presenta el incidente.	numérico (entero)		X	X		
ip_zona	IP de la zona donde se presenta el incidente.	caracter variable	12	X			
tipo	Tipo de la zona donde se presenta el incidente de acuerdo a la IP:	caracter	1	X			

NOMBRE DE LA TABLA: AREA

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_area	Número único identificador del área donde labora el personal atendido.	numérico (entero)		X	X		Incremento automático
siglas	Siglas del área donde labora el personal atendido.	caracter variable	12	X			
nomb_largo	Nombre del área donde labora el personal atendido.	caracter variable	60	X			

NOMBRE DE LA TABLA: PERSONAL

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_personal	Número único identificador del personal al que se atiende.	numérico (entero)		X	X		Incremento automático
nombre	Nombre completo de la persona a la que se atiende.	caracter variable	40	X			
apellidos	Apellidos de la persona a la que se atiende.	caracter variable	40	X			
responsable	Cargo de la persona	caracter variable	80	X			
id_area	Número único identificador del área al que pertenece el personal atendido.	numérico (entero)		X			

NOMBRE DE LA TABLA: BECARIOS_DSC_SEGURIDAD

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_becario	Número único identificador del becario que atiende el incidente.	numérico (entero)		X	X	BECARIOS_DSC (id_becario)	Incremento automático
id_incidente	Número único identificador del incidente relacionado con el becario.	numérico (entero)		X	X	SEGURIDAD (id_incidente)	

NOMBRE DE LA TABLA: BECARIOS_DSC

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_becario	Número único identificador del becario.	Numérico (entero)		X	X		Incremento automático
nombre	Nombre del becario.	caracter variable	40	X			
apellidos	Apellidos del becario.	caracter variable	40	X			
activo	Estado del becario (Activo = True, Inactivo= False).	boolean		X			True

NOMBRE DE LA TABLA: OFICIO

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_incidente	Número único identificador del incidente.	numérico (entero)		X	X		Incremento automático
respuesta	Respuesta al oficio (reporte de algún incidente).	caracter variable	30	X			
asunto	Asunto del oficio.	caracter variable	25	X			
inst_reporta	Institución que reporta el incidente.	caracter	5	X			
ip_octeto	Asunto por el cual se realiza la atención al incidente(Incidente en Seguridad).	caracter	3	X			
fecha	Fecha en que se realizó el oficio.	fecha		X			
periodo	Periodo/semestre en el que se realiza el oficio.	caracter	6	X			
id_causa	Número único identificador de la causa del incidente.	numérico (entero)		X		CAUSA (id_causa)	
id_personal	Número único identificador del personal al que va dirigido el oficio.	numérico (entero)		X		PERSONAL (id_personal)	
id_zona	Número único identificador de la zona donde se presenta el incidente.	numérico (entero)		X		ZONA (id_zona)	
id_usuario	Número único identificador del usuario que utiliza el sistema y realiza el oficio.	numérico (entero)		X		USURIO (id_usuario)	

NOMBRE DE LA TABLA: CPP

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_cpp	Número único identificador del personal al que se le envía una copia del oficio.	numérico (entero)		X	X		Incremento automático
nombre	Nombre de la persona a la que se le envía una copia del oficio.	caracter variable	40	X			
apellidos	Apellidos de la persona a la que se le envía una copia del oficio.	caracter variable	40	X			
responsable	Puesto que ocupa dentro de la Facultad de Ingeniería.	caracter variable	80	X			

NOMBRE DE LA TABLA: LOGUEO

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_log	Número único identificador del logueo de los usuarios.	numérico (entero)		X	X		Incremento automático
fecha	Fecha y hora en la que accede el usuario al sistema.	fecha / hora		X			
ip	Dirección IP de la máquina desde donde accede el usuario.	caracter	15	X			
id_usuario	Número único identificador del usuario que accede al sistema.	numérico (entero)		X			

NOMBRE DE LA TABLA: PREGUNTA_SECRETA

COLUMNA	DESCRIPCIÓN	TIPO DE DATO	TAM	NOT NULL	PK	FK	DEFAULT
id_pregunta	Número único identificador de las preguntas secretas con las cuales se identifican los usuarios.	numérico (entero)		X	X		Incremento automático
pregunta	Pregunta secreta	caracter variable	50	X			

III.7. PRUEBAS DEL SISTEMA

III.7.1. INTRODUCCIÓN

Cuando se diseña o implementa un sistema de cómputo se debe tener en mente la facilidad de las pruebas y que al mismo tiempo las pruebas deben de mostrar un conjunto de características para alcanzar el objetivo de encontrar la mayor cantidad de errores con un mínimo de esfuerzo.

A continuación las siguientes características son las que propician la creación de software que tenga facilidad de prueba:

- **Operatividad:** Si un sistema está diseñado e implementado con la calidad en mente, serán relativamente escasos los errores que bloquearán la ejecución de las pruebas, lo cual permite el avance de éstas sin correcciones ni reinicios.
- **Observabilidad:** Las entradas proporcionadas como parte de la prueba producen salidas distintas. Los estados y las variables del sistema son visibles y pueden consultarse durante la ejecución. Por lo tanto, el código fuente es accesible.
- **Control:** Cuanto mejor se controle el software, mejor se automatizarán y mejorarán las pruebas. Por lo que las pruebas pueden ser especificadas, automatizadas y reproducidas.
- **Simplicidad:** El programa debe de mostrar simplicidad funcional, estructural y de código.
- **Estabilidad:** Cuantos menos cambios haya, menores alteraciones habrá en la prueba. Los cambios al software son poco frecuentes, se controlan cuando ocurre errores y no validan las pruebas existentes.

Existen dos maneras de probar cualquier producto construido:

1. Si se conoce la función específica para la que se diseñó el producto, se aplican pruebas que demuestren que cada función es plenamente operacional, mientras se buscan los errores de cada función.
2. Si se conoce el funcionamiento interno del producto, se aplican pruebas para asegurarse de que “todas las piezas encajan”, es decir, que las operaciones internas se realizan de acuerdo con las especificaciones y que se han probado todos los componentes internos de manera adecuada.

Al primer enfoque de prueba se le denomina prueba de caja negra; al segundo prueba de caja blanca.

III.7.2. PRUEBAS DE CAJA BLANCA

Las pruebas de caja blanca, en ocasiones llamada prueba de caja de cristal, verifican todos los posibles caminos del código. Para que una prueba de caja blanca se considere correcta, se debe de recorrer, al menos conceptualmente todos los caminos del programa.

Considerando no solamente las entradas correctas, sino también las Incorrectas. Lo más importante que hay que considerar para realizar pruebas de caja blanca es lo siguiente:

- Cómo se manejan los errores dentro de la caja.
- Que se cubran perfectamente todas las rutas del código (Se recorren todos los caminos de las condiciones IF o equivalente).
- Cómo se deben de modificar las entradas para que recorran todas las rutas del código.
- Qué dependencias existen con otros recursos.

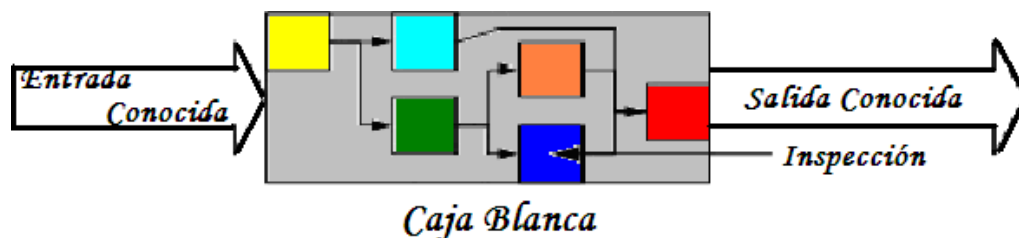


Figura 3.24. Diagrama de una Prueba de Caja Blanca.

III.7.2.1. PRUEBA DE LA RUTA BÁSICA

Este método permite que el diseñador de casos de prueba obtenga una medida de complejidad lógica de un diseño de rutas de ejecución. Los casos de prueba derivados para ejercitar el conjunto básico deben garantizar que se ejecuta cada instrucción del programa por lo menos una vez durante la prueba.

Al realizar las pruebas en el sistema se tomaron en cuenta todos los segmentos de código generados dentro de las clases de negocio, las cuales están conformados por bucles y condiciones.

Para esto, se realiza una representación del flujo de control por medio de una gráfica de flujo (o gráfica del programa)¹.

¹ En realidad, el método de la ruta básica se aplica sin el uso de las gráficas de flujo. Sin embargo, sirven como notación útil para comprender el flujo de control e ilustrar el enfoque.

Ejemplo de prueba de la ruta básica	
Módulo:	Generar Oficio
Flujo:	Insertar en la base de datos las personas que recibirán copia del oficio.
Código Fuente	
<pre> ArrayList copyIds = (ArrayList) dform.get("copyDatos"); (1) (2) (3) if (ccp.length >= 1 && idIncidente != null) { OficioCCPDAO daoOficioCCP = (OficioCCPDAO) DAOConfig.getDao(OficioCCPDAO.class); (4) for (int i = 0; i < ccp.length; i++) (5) { if (!copyIds.contains(ccp[i])) (6) { daoOficioCCP.insert(idIncidente, ccp[i]); (7) } } List original = Arrays.asList(ccp); (8) for (int j = 0; j < copyIds.size(); j++) (9) { if (!original.contains(copyIds.get(j))) (10) { daoOficioCCP.delete(Integer.valueOf(copyIds.get(j).toString()), idIncidente); (11) } } } dform.set("exitoOficio", "1"); (12) </pre>	
Grafo para el código anterior:	

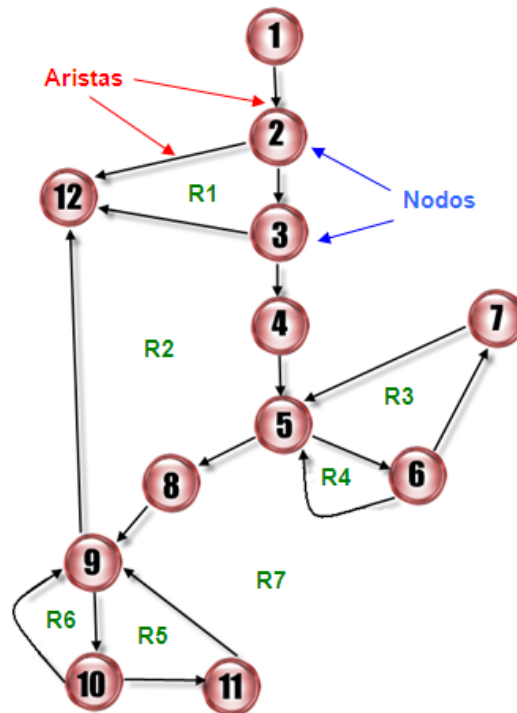


Figura 3.25. Grafo del código para insertar a las personas que recibirán copia del oficio.

Los círculos, llamados **nodos** representan una o más instrucciones del código. Las flechas en la gráfica de flujo, son las **aristas** o **enlaces** y representan el flujo de control.

Una **arista** debe de terminar en un nodo, aunque el nodo no represente ninguna instrucción.

Las áreas que limitan aristas y nodos se denominan **regiones**. Cuando se cuentan las regiones se incluyen las áreas ubicadas fuera de la gráfica.

Cada nodo que contiene una condición es un **nodo predicado** y se caracteriza porque de él emanan dos o más aristas.

Como se mencionó el objetivo principal de la prueba es que por lo menos una vez durante la prueba se ejecute una vez las instrucciones. Debe observarse que un conjunto básico no es único, y que es posible derivar varios conjuntos básicos.

Por lo tanto, para conocer cuántas rutas se deben de buscar, se realiza el cálculo de la complejidad ciclomática. La complejidad ciclomática es una métrica que proporciona una medida cuantitativa de la complejidad lógica del programa y resulta útil para predecir cuáles módulos tienen más probabilidad de contener errores.

La complejidad ciclomática se basa en la teoría gráfica y se calcula de la siguiente manera:

1. El número de regiones corresponde a la complejidad ciclomática.
2. La complejidad ciclomática, $V(G)$, de una gráfica de flujo G , se define como:
 $V(G) = E - N + 2$.
 Donde E es el número de aristas y N el número de nodos de la gráfica de flujo.
3. La complejidad ciclomática, $V(G)$, de una gráfica de flujo G , también se define como $V(G) = P + 1$.

Donde P es el número de nodos predicado incluidos en la gráfica de flujo G .

<p>Cálculo de la complejidad ciclomática del grafo anterior:</p> <p>Del grafo anterior:</p> <ol style="list-style-type: none"> 1. Tiene 7 regiones. 2. $V(G) = 17$ aristas – 12 nodos +2 = 7 3. $V(G) = 6$ nodos predicado +1= 7 <p>Lo más notable del valor de $V(G)$ es que proporciona un límite superior del número de pruebas que debe diseñarse y ejecutarse para garantizar la cobertura de todas las instrucciones del programa. En este caso el límite superior es de 7.</p> <p>Resultado de la prueba:</p> <p>Estos casos de prueba permitieron detectar errores en los flujos de los programas y así disminuir la complejidad del código. Cabe mencionar que debido al número de pruebas que se presentan en cada procedimiento, resultó complicado verificar la totalidad del código, por lo que se realizaron pruebas sólo en los flujos principales del sistema, obteniendo un 96% de éxito.</p>

III.7.2.2. PRUEBAS DE ESTRUCTURA DE CONTROL

La técnica de la prueba de la ruta básica es simple y efectiva, sin embargo, no es suficiente por sí misma. Siempre es necesario complementarla con otras variantes de las pruebas de estructura de control, permitiendo una mejor cobertura y calidad de la prueba de caja blanca.

III.7.2.2.1. PRUEBA DE CONDICIÓN

Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en un módulo del programa. Una condición simple es una variable booleana o una expresión relacional, la cual toma la forma:

$$E_1 <operador\ relacional> E_2$$

Donde E_1 y E_2 son expresiones aritméticas y *<operador relacional>* es uno de los siguientes: <, <=, !=, > o >=. Una condición compuesta la integra dos o más condiciones simples, operadores booleanos y paréntesis.

Si una condición es incorrecta, entonces por lo menos un componente de la condición es incorrecto. Entre los tipos de errores en una condición se incluye los presentes en el operador booleano (que sean incorrectos, que falten o que tenga adicionales), en la variable booleana, en los paréntesis booleanos, en los operadores relacionales y en la expresión aritmética. En conclusión, este método se concentra en la prueba de cada condición del programa para asegurar que no contiene errores.

III.7.2.2.2. PRUEBA DEL FLUJO DE DATOS

Este método selecciona rutas de prueba en un programa de acuerdo con las ubicaciones de las definiciones y los usos de las variables en el programa. Esta prueba se ilustra suponiendo que a cada instrucción de un programa se le asigna un número de instrucción, y que ninguna función modifica sus parámetros o variables globales.

Las estrategias de prueba de flujo de datos son útiles para seleccionar caminos de prueba de un programa que contenga sentencias if o de bucles anidados.

III.7.2.2.3. PRUEBAS DE BUCLES

Es una técnica que se concentra exclusivamente en la validez de la construcción de bucles. Es posible definir cuatro diferentes clases de bucles: bucles simples, concatenados, anidados y no estructurados.

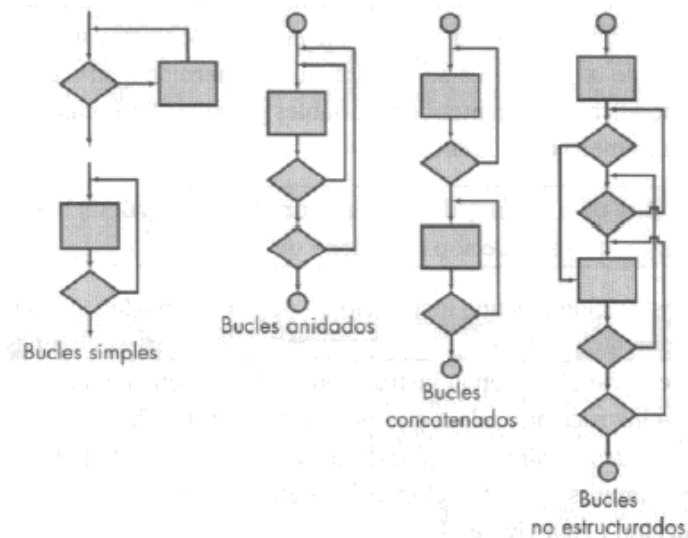


Figura 3.26. Existen cuatro tipos de bucles.

- Bucles simples: Se aplica donde n es el número máximo de pasos que permite el bucle.
 1. Omitir por completo el bucle.
 2. Sólo un paso por el bucle.
 3. Dos pasos por el bucle.
 4. m pasos por el bucle, donde $m < n$.
 5. $n = 1, n, n+1$ pasos por el bucle.

- Bucles anidados: A medida que crece el nivel de anidamiento en un bucle, el número de pruebas posibles crecería demasiado. Beizer sugiere un enfoque que ayuda a reducir el número de pruebas.
 1. Empezar en el bucle más interno y asignar todos los bucles los valores mínimos.
 2. Aplicar pruebas de bucle simple al más interno mientras se mantienen los externos en los valores mínimos del parámetro de iteración. Además, agregar otras pruebas para los valores fuera de rango o excluidos.
 3. Trabajar hacia fuera, conduciendo pruebas para el siguiente bucle, pero manteniendo todos los demás bucles externos en valores mínimos y otros bucles anidados en valores "típicos".
 4. Seguir mientras no se hayan probado todos los bucles.

- Bucles concatenados, Los bucles concatenados se prueban empleando el enfoque de los bucles simples, si cada uno de los bucles es independiente. Los bucles no son independientes cuando el contador del bucle 1 se emplea como valor inicial para el bucle 2. Si es así, se recomienda el enfoque aplicado a los bucles anidados.

- Bucles no estructurados. Siempre que sea posible, esta clase de bucles debe diseñarse nuevamente para reflejar el uso de bucles estructurados.

<p>Resultado de la prueba de Estructura de Control:</p> <p>Las pruebas de estructura de control permitieron encontrar inconsistencias en el flujo de los datos, principalmente errores tan pequeños como lo es un operador relacional incorrecto. La cobertura de pruebas no abarcó la totalidad de los flujos de los programas, pero sí, los principales del sistema. Incluso se detectaron variables que se utilizaban en condiciones IF-ELSE pero que ya no fueron utilizadas conforme fue cambiando el desarrollo.</p> <p>Como caso particular, en el módulo de estadísticas del Departamento de Seguridad en Cómputo se presentaron errores debido a una condición compuesta que estaba mal declarada.</p>
<p>Resultado de la prueba de Bucles:</p> <p>Problema: Al redireccionar una página, el explorador no mostraba nada.</p> <p>Solución: Al realizar pruebas en los bucles, se encontró un bucle que nunca salía del ciclo provocando que la siguiente página nunca se mostrara.</p> <p>Estas pruebas permitieron refinar la programación en los bucles para hacerlos más óptimos.</p> <p style="padding-left: 40px;">Las pruebas en los bucles anidados fueron satisfactorias. Cabe mencionar que fueron de gran ayuda las técnicas sugeridas por Beizer.</p> <p style="padding-left: 40px;">Las pruebas en los bucles concatenados, resultaron laboriosas debido a la gran cantidad que hay en el sistema, aún así esto no fue obstáculo para verificar que los bucles cumplieran con su funcionalidad.</p> <p>Conclusión de las pruebas de caja blanca:</p> <p>Como se puede apreciar las pruebas de caja blanca realizadas en el sistema fueron exitosas y ayudaron a corregir detalles de la programación.</p> <p>Los grafos obtenidos son la base para realizar las pruebas de estructura de control y permiten conocer el límite de pruebas a realizar.</p>

III.7.3. PRUEBAS DE CAJA NEGRA

Las pruebas de caja negra son llamadas pruebas de comportamiento en las cuales se alimentan las entradas y se verifica que las salidas sean las correctas sin verificar cómo está trabajando internamente el proceso. Se concentran en los requisitos funcionales del software.

Las pruebas de caja negra limitan la habilidad de pruebas del código, porque no se sabe si se está probando todo posible camino o ruta.

Las pruebas de caja negra tratan de encontrar errores en las siguientes categorías:

1. Funciones incorrectas o faltantes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en acceso a base de datos externas.
4. Errores de comportamiento o desempeño.
5. Errores de inicialización y término.

A diferencia de las pruebas de caja blanca, que se realizan al inicio del proceso de pruebas, las de caja negra tienden a aplicarse durante las últimas etapas de las pruebas.

Esto se debe a que desatienden la estructura de control y se concentran en el dominio de la información. Para esto se debe revisar lo siguiente:

- Para cada valor válido de entrada debe existir un valor apropiado de salida.
- Para cada valor inválido de entrada, solo un valor apropiado debe de ser regresado.
- Para cada estado válido de entrada, ocurre un estado válido de salida.
- Para cada estado inválido de entrada, ocurre un estado válido de salida.

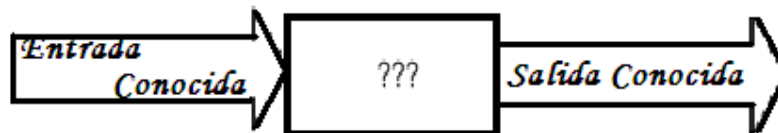


Figura 3.27. Pruebas de Caja Negra.

III.7.3.1. MÉTODO GRÁFICO DE PRUEBA

El primer paso en la prueba de caja negra es comprender los objetos modelados en el software y la relación entre ellos. Una vez que se logra esto, el siguiente paso consiste en definir la serie de pruebas que verifican que “todos los objetos tienen la relación esperada entre sí”.

Para ello se crea una gráfica de objetos importantes (representados como nodos) con sus relaciones y luego se realizan una serie de pruebas que cubran la gráfica de tal manera que se ejercite cada objeto y relación, para que se descubran los errores.

Los nodos se representan como círculos conectados por enlaces que toman un número diferente de formas.

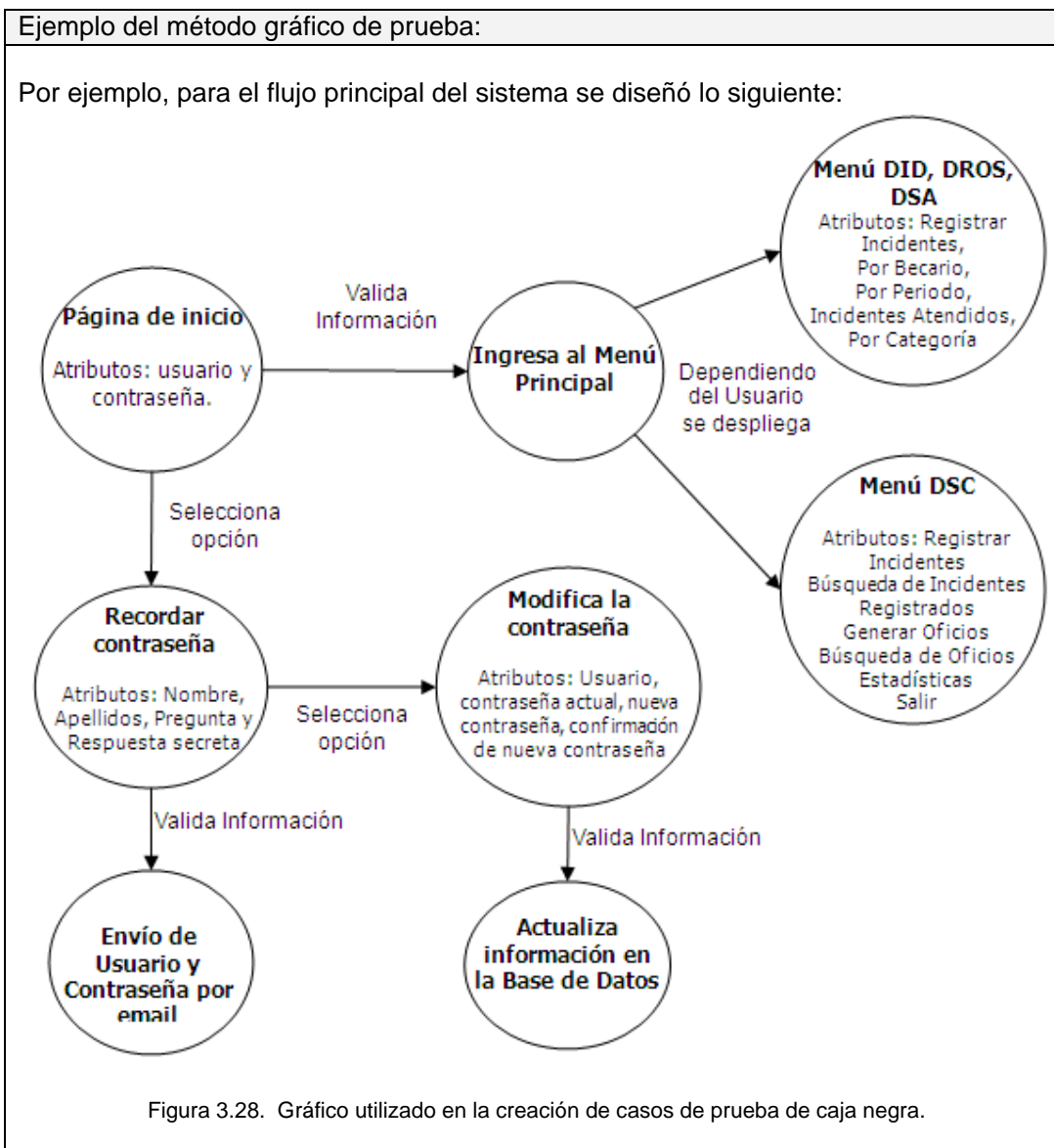
Un enlace directo (representado por una flecha) indica que una relación se mueve en una sola dirección.

Un enlace bidireccional, también denominado enlace simétrico, indica que la relación se aplica en ambas direcciones.

Los enlaces paralelos se emplean cuando se establece un número diferente de

relaciones entre los nodos de la gráfica.

El primer objetivo al realizar estas pruebas es para asegurarse que ningún nodo sea omitido y que los atributos de los objetos sean correctos. Además la cobertura de enlaces se prueba basándose en sus propiedades.



Resultado del método gráfico de prueba:

Al realizar las pruebas se logró abarcar cada uno de los métodos que se realizan para cumplir el flujo de la aplicación, cubriendo los objetos indicados en el grafo. Dado que se examina el comportamiento en cada nodo, se valida que en cada uno de ellos la salida sea la correcta o que se encuentre en un estado válido. Principalmente, los problemas que se encontraron fueron de validación o problemas en el flujo de la aplicación por un mal redireccionamiento, pero al encontrarlos se procedió a corregir estos detalles.

III.7.3.2. PARTICIÓN EQUIVALENTE

Es un método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de las cuales se puede dividir casos de prueba. Un caso de prueba ideal de manejo simple descubre una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos de caracteres) lo cual de otra manera requeriría la ejecución de muchos casos antes de que se observe el error general.

La idea principal de trabajar con partición equivalente es reducir el número total de casos de prueba que deben desarrollarse.

El diseño de casos de prueba para partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada.

Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleanas.

Las clases de equivalencia se definen de acuerdo con lo siguiente:

1. Si una condición de entrada especifica un rango, se definen una clase de equivalencia válida y dos no válidas.
2. Si una condición de entrada requiere un valor específico, se definen una clase de equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se definen una clase de equivalencia válida y otra no válida.
4. Si una condición de entrada es booleana, se define una clase de equivalencia válida y otra no válida.

Al aplicar estos puntos se desarrollan y ejecutan los casos de prueba para cada objeto de los datos del dominio de entrada. Los casos de prueba se seleccionan de modo que el mayor número de atributos de clases de equivalencia se ejercita una vez.

Aplicación del Método de partición equivalente:	
Ejemplo 1:	
Módulo:	Página de autenticación.
Prueba:	Se verificó el acceso de usuarios autorizados.
<p>Con el método de partición equivalente se aplicaron pruebas en el sistema determinando un conjunto de estados válidos e inválidos para los diferentes casos de prueba, como por ejemplo el flujo para determinar si un usuario tiene acceso o no al sistema.</p> <p>Para este caso dada la condición de entrada (usuario y contraseña), existe un estado válido cuando se confirman los datos del usuario que ingresa al sistema. El estado inválido es cuando los datos del usuario son incorrectos o cuando no se ingresa ningún dato.</p>	

En la figura 3.27 se muestra la pantalla de autenticación del sistema:



Figura 3.29. Cuando el usuario y/o la contraseña es incorrecta, el sistema despliega un mensaje informando sobre el problema.

Resultados de la prueba:

Se encontró un error al querer ingresar al sistema. que no era controlado por la aplicación, provocando que se desplegará la siguiente pantalla:

```

type Informe de Excepción
mensaje
descripción El servidor encontró un error interno () que hizo que no pudiera rellenar este requerimiento.
excepción
java.lang.NullPointerException
org.apache.catalina.loader.WebappClassLoader.findResourceInternal(WebappClassLoader.java:1965)
org.apache.catalina.loader.WebappClassLoader.findResource(WebappClassLoader.java:932)
org.apache.catalina.loader.WebappClassLoader.getResourceAsStream(WebappClassLoader.java:1153)
org.apache.struts.util.PropertyMessageResources.loadLocale(PropertyMessageResources.java:254)
org.apache.struts.util.PropertyMessageResources.getMessage(PropertyMessageResources.java:147)
org.apache.struts.util.MessageResources.getMessage(MessageResources.java:307)
org.apache.struts.util.MessageResources.getMessage(MessageResources.java:334)
org.apache.struts.util.MessageResources.getMessage(MessageResources.java:215)
org.apache.struts.action.RequestProcessor.processException(RequestProcessor.java:528)
org.apache.struts.action.RequestProcessor.processActionPerform(RequestProcessor.java:433)
org.apache.struts.action.RequestProcessor.process(RequestProcessor.java:236)
org.apache.struts.action.ActionServlet.process(ActionServlet.java:1196)
org.apache.struts.action.ActionServlet.doPost(ActionServlet.java:432)
javax.servlet.http.HttpServlet.service(HttpServlet.java:710)
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
mx.unica.saaci.session.SessionSAACI.doFilter(SessionSAACI.java:51)
org.netbeans.modules.web.monitor.server.MonitorFilter.doFilter(MonitorFilter.java:390)
nota La traza completa de la causa de este error se encuentra en los archivos de diario de Apache Tomcat/6.0.14.
    
```

Apache Tomcat/6.0.14

Figura 3.30. Pantalla de error dentro del SAACI al ingresar al Sistema.

Si se regresaba a la página anterior y nuevamente se introducía el usuario y la contraseña ya se podía ingresar al sistema sin ningún problema. Esto no era directamente un error de programación.

Se realizaron pruebas y se revisó a detalle el reporte del error, encontrando que el problema se debía a la versión del driver para la conexión a la base de datos.

Inicialmente se trabajó con el driver **pg74.1jdbc3.jar** para Postgres 7.x, pero como la versión de base de datos fue migrada a la 8.1.4, fue necesario cambiar el driver (**postgresql-8.1-413.jdbc3.jar**), debido a que provocaba conflictos para conectarse desde el servidor.

Por lo tanto, esta prueba permitió determinar un error que no era propio de la programación del sistema, sino mas bien de un conflicto de las versiones del driver con la base de datos.


Ejemplo 2:	
Módulo:	Módulo del Departamento de Seguridad en Cómputo.
Prueba:	Verificar la generación de Oficios.

Al igual que en el ejemplo anterior, se aplicó el método de partición equivalente para comprobar la generación de Oficios en el módulo del Departamento de Seguridad en Cómputo. Las pruebas se realizaron minuciosamente debido a que la información de entrada tenía que coincidir con la del oficio que se genera.

Definimos las condiciones de la prueba:

La condición de entrada es la captura de los datos del Oficio.
 La condición de salida válida es la confirmación de éxito del registro y la descarga del oficio.
 La condición de salida inválida es cuando algún dato falta o no es válido y el sistema notifica del problema por medio de un mensaje.

La figura 3.31 es un ejemplo del registro de un oficio, en el cual se capturan todos los datos y se selecciona la opción para generar el oficio.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA
SECRETARÍA GENERAL
UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO

OFICIO/UNICA/ /2007

ASUNTO: INCIDENTE EN SEGURIDAD ▾
Dirección IP: Agregar otra... ▾

Alta de una nueva IP:

IP: . . .
 Zona:

Dirigido a:
Responsable: **Área:**

Nueva Área

Nombre Completo del Área:
Siglas:

Presente

Me dirijo a usted para informarle que el (la) tienen informes, en el sentido de que la computadora con dirección IP, misma que tenemos registrada bajo su responsabilidad.

La vulnerabilidad que se marca en los reportes es que se encuentra como:

Alta de una nueva Causa:

Nombre de la Causa:
Descripción:

Le solicito sea tan amable de indicarle al administrador de esta computadora que el equipo sea revisado de inmediato, y envíe a este Departamento por oficio los problemas detectados y las acciones llevadas a cabo para resolver el incidente; de la misma forma es necesario incluya la siguiente información en el reporte:

- Marca del equipo
- Sistema Operativo
- Modelo
- Nombre del responsable directo

Es importante recordarle, que manifestamos nuestro apoyo y disposición para resolver los problemas relativos a seguridad que afecten sus sistemas de cómputo y redes.

Sin más por el momento reciba un cordial saludo.

Atentamente.

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Ciudad Universitaria, D. F., a

RESPONSABLE DEL DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

ING. RAFAEL SANDOVAL VÁZQUEZ

C.C.P.:

Agregar otro... ▾

Figura 3.31. Pantalla para introducir la información del Oficio.

Si el registro se hace correctamente el sistema nos muestra el siguiente mensaje:

El Oficio se registró satisfactoriamente.



Figura 3.32. Opción que permite descargar el oficio del Sistema.

Finalmente el sistema nos genera el siguiente Oficio, donde podemos verificar que efectivamente arroja los datos que fueron introducidos.



FACULTAD DE INGENIERÍA

SECRETARÍA GENERAL
 UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO
 OFICIO/UNICA/ 23 / 2009
 ASUNTO: INCIDENTE EN SEGURIDAD
 Dirección IP:10.51.117.100

Dirigido a: CAROL NAYELI JIMENEZ PEREZ
 LIDER DE PROYECTO
 DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO

Presente

Me dirijo a usted para informarle que el Departamento de Seguridad en Cómputo tienen informes, en el sentido de que la computadora con dirección 10.51.117.100, misma que tenemos registrada bajo su responsabilidad.

La vulnerabilidad que se marca en los reportes es que se encuentra como: PHISING
 Le solicito sea tan amable de indicarle al administrador de esta computadora que el equipo sea revisado de inmediato, y envíe a este Departamento por oficio los problemas detectados y las acciones llevadas a cabo para resolver el incidente; de la misma forma es necesario incluya la siguiente información en el reporte:

- Marca del equipo
- Modelo
- Sistema Operativo
- Nombre del responsable directo

Es importante recordarle, que manifestamos nuestro apoyo y disposición para resolver los problemas relativos a seguridad que afecten sus sistemas de cómputo y redes.

Sin más por el momento reciba un cordial saludo.

Atentamente.

POR MI RAZA HABLARÁ EL ESPÍRITU

Ciudad Universitaria, D. F., a 07 de Julio del 2009

RESPONSABLE DEL DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

ING. RAFAEL SANDOVAL VAZQUEZ

C.C.P.:
 ENRIQUE BARRANCO VITE
 ALEJANDRO LORA SERNA

Figura 3.33. Oficio generado por el Sistema.

Resultados de la prueba:

Con estas pruebas se aseguró que la generación de oficios se realizara correctamente y sobre todo se verificó que el sistema terminará en un estado válido tal como se muestra en la Figura 3.33.

III.7.3.3. PRUEBA DE LA TABLA ORTOGONAL

Esta se aplica en problemas en los cuales el dominio de entrada es relativamente pequeño, pero demasiado grande para una prueba exhaustiva. Este método resulta útil sobre todo para encontrar errores asociados con las fallas de región (una categoría de error asociada con los defectos de la lógica en un componente de software).

Para crear una prueba de tabla ortogonal se crea una tabla ortogonal de casos de prueba, la cual tiene una “propiedad de equilibrio”. Es decir, que los casos de prueba son representados con puntos y están uniformemente dispersos por todo el dominio de la prueba.

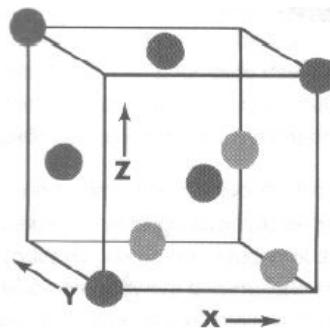


Figura 3.34. Ejemplo de una tabla ortogonal.

Estos casos de prueba sólo son útiles cuando se está seguro de que los parámetros de prueba no interactúan. Detectarán fallas de lógica donde un solo valor de parámetro hace que el software funcione mal. Sin embargo, este método no detecta fallas de lógica que provoquen un mal funcionamiento cuando dos o más parámetros toman ciertos valores simultáneamente.

La ventaja de realizar pruebas de la tabla ortogonal es que permiten proporcionar una buena cobertura de prueba con un número considerablemente menor de casos de prueba que la estrategia exhaustiva. También se puede utilizar para:

- ❖ Detectar y aislar todas las fallas de modalidad simple. Es un problema consistente con cualquier parámetro simple.
- ❖ Detectar todos los fallos de modalidad doble: Cuando se presenta un problema donde están afectados dos parámetros que intervienen conjuntamente.

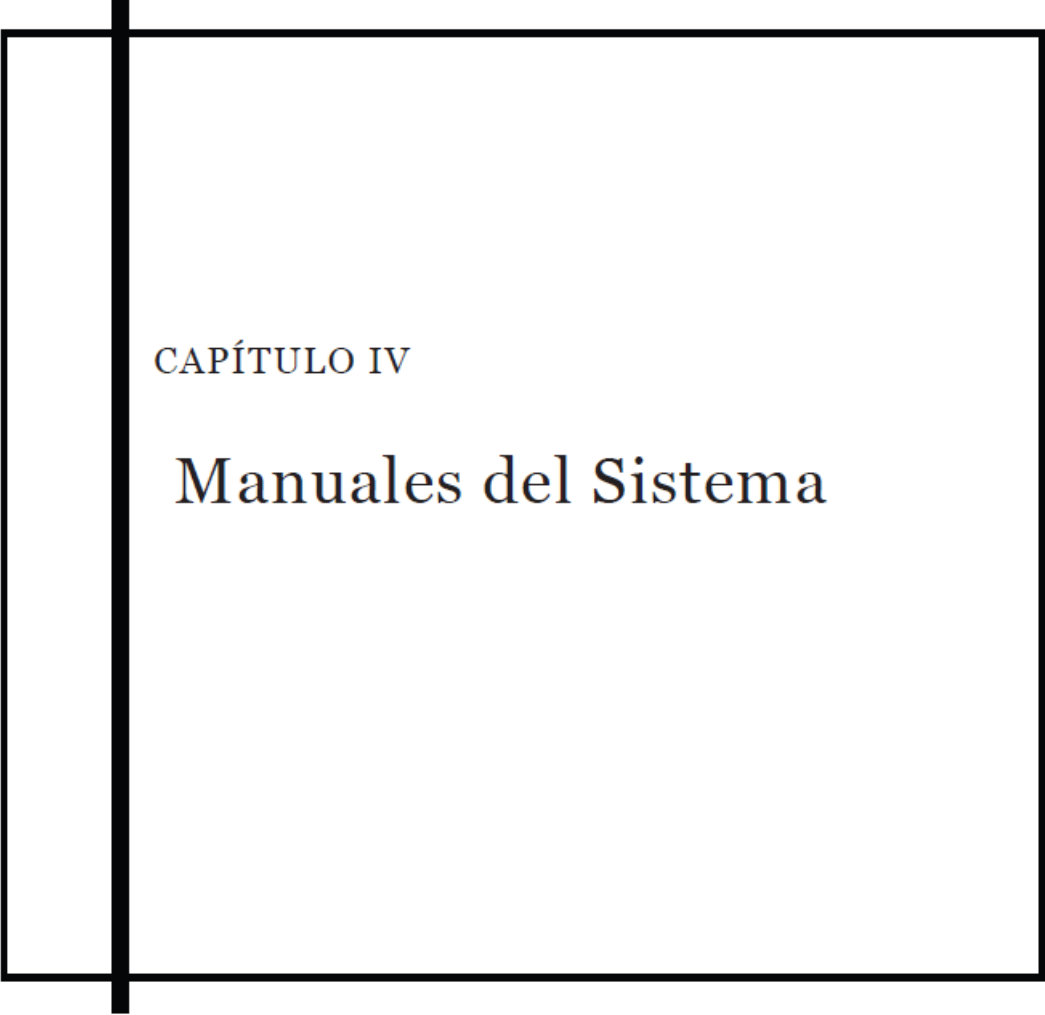
III.7.4. PORCENTAJE DE COBERTURA ADECUADO AL REALIZAR PRUEBAS

Como todas las métricas, el porcentaje de cobertura de código no es una medida absoluta de la calidad o la efectividad del código, sino una directriz auxiliar. Por lo tanto, no es necesario imponer un porcentaje definitivo como criterio de aceptación.

Sin embargo, es recomendable lograr un porcentaje de cobertura de código alrededor de 80 a 85%.

Algunos estudios han demostrado que el beneficio de subir a porcentajes por encima de 90% es marginal y muy complejo de lograr en componentes no triviales.²

² Para un análisis más detallado, leer el whitepaper “*How to misuse code coverage*”, de Brian Marick: <http://www.exampler.com/testing-com/writings/coverage.pdf> [Consultado el 11/10/2009]



CAPÍTULO IV

Manuales del Sistema

CAPÍTULO IV. MANUALES DEL SISTEMA

Para la implementación del sistema se realizaron los siguientes manuales:

- **Manual del usuario:** Documento técnico donde se describen las características y funciones del Sistema para la Administración, Atención y Control de Incidentes (SAACI) con la finalidad de servir como guía al usuario final.
- **Manual del programador:** Documento técnico donde se describe a detalle la arquitectura del sistema (dividido por medio del patrón de diseño Modelo Vista Controlador) y su configuración.

Lo anterior con la finalidad de servir como referencia para futuras modificaciones al sistema, o bien el desarrollador conozca las herramientas utilizadas.



Manual de Usuario



Sistema de Administración, Atención y
Control de Incidentes



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Tabla de contenido

Introducción 2

Requerimientos básicos 2

Autenticación 2

Recuperar contraseña 5

Modificar contraseña 5

Módulo de Administración de Incidentes para el DID, DSA, DROS Y JEFATURA..... 7

 Inicio 7

 Registrar Incidentes..... **¡Error! Marcador no definido.**

 Consulta por becario 10

 Consulta por periodo..... 11

 Consulta por Incidentes atendidos..... 13

 Consulta por categoría 14

 Generar reporte de Incidente del Módulo General 15

 Actualizar información de Incidente. 15

 Salir 16

Módulo de Administración de Incidentes para el DSC 17

 Inicio 17

 Registrar Incidentes 18

 Búsqueda de Incidentes registrados 20

 Generar reporte de Incidente del DSC 21

 Actualizar información de Incidente 21

 Generar Oficios..... 24

 Búsqueda de Oficios..... 27

 Actualizar información del Oficio 28

 Estadísticas 29

 Salir 33

Responsables del SAACI 33



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Introducción

SAACI es un sistema diseñado para atender las necesidades de control de los incidentes que se presentan en UNICA, los cuales pueden ser:

- Incidentes de red (spam, virus y gusanos, violación a la propiedad intelectual, phishing, actividad intrusiva, cambio de infraestructura de red, entre otros), o bien
- Incidentes que requieren de soporte técnico o labores administrativas dentro de la Facultad de Ingeniería (FI).

El sistema está dirigido principalmente al personal encargado de llevar la administración de dichos incidentes dentro de los Departamentos de UNICA: Seguridad en Cómputo (DSC), Investigación y Desarrollo (DID), Servicios Académicos (DSA), Redes y Operación de Servidores (DROS) y Jefatura. Está dividido en dos módulos principales:

1) Administración de Incidentes para los departamentos: DID, DSA, DROS y Jefatura (al cual llamaremos Módulo General).

2) Administración de Incidentes para el DSC (al cual llamaremos Módulo para el DSC).

Para los usuarios de cada uno de estos departamentos, el sistema muestra opciones específicas para el control de incidentes que serán explicadas a detalle en el presente.

Requerimientos básicos

Tener instalado un navegador Web: Mozilla FireFox versión 2 o posterior.


Autenticación

El sistema será accedido vía Web en la siguiente dirección:

<http://localhost>¹

Proporcione su nombre de usuario y su contraseña para autenticarse ante el sistema y presione el botón “**Aceptar**” (Figura 4.1).

¹ Por motivos de seguridad, las direcciones IP reales fueron omitidas.

	Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
	Institución:	UNICA	Fecha:	Abril 09
	Documento:	Manual de Usuario	Versión:	1



Viernes 3 de Abril de 2009 19:09:49

Figura 4.1. Pantalla de autenticación del usuario

Si se teclea un nombre de usuario y/o contraseña incorrecto(s) el sistema no permitirá su ingreso y le mostrará el mensaje: **“El usuario no existe, favor de verificar”** (Figura 4.2).



Domingo 5 de Abril de 2009 12:58:34


 El usuario no existe, favor de verificar

Figura 4.2. Pantalla de autenticación del usuario. Error de autenticación.

NOTA: Por cuestión de seguridad se recomienda que cambie los datos que le proporcionó el administrador al momento de entrar por primera vez al sistema (Ver Modificar Contraseña).

Si los datos introducidos son correctos, el sistema le mostrará la página principal del módulo correspondiente al departamento al que pertenece (Figura 4.3 y 4.4).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Figura 4.3. Página principal para el módulo de Administración de Incidentes para el DSC.



Figura 4.4 Página principal para el módulo de Administración de Incidentes para los Departamentos: DID, DSA, DROS y Jefatura.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Recuperar contraseña

En el caso de olvido de la contraseña, ingrese a la opción “**¿Olvidó su contraseña?**” dentro de la página principal y proporcione el(los) nombre(s) y apellidos con los que fue registrado en el sistema, así como la pregunta secreta y su respectiva respuesta. Una vez proporcionados los datos presione el botón “**Recordar**” (Figura 4.5).

Figura 4.5. Pantalla para recuperar la contraseña.

Si los datos proporcionados son incorrectos, el sistema mostrará la leyenda: “**El usuario no existe, favor de verificar**”.

Si los datos son correctos, la contraseña será enviada al correo electrónico que proporcionó al momento de su alta y se mostrará el mensaje: “**Sus datos se han enviado con éxito a su correo electrónico**”. El remitente del correo será **saaci**.

Modificar contraseña

Entre a la opción: “**¿Olvidó su contraseña?**” ubicada en la página principal del sistema y elija la opción “**Modifica tu contraseña**” (Figura 4.6).

Figura 4.6. Link para la pantalla: Modificar la contraseña.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Ingrese su nombre de usuario, la contraseña anterior y la nueva contraseña, ésta deberá consistir en un mínimo de ocho y un máximo de doce caracteres.

El sistema presentará una lista de opciones de preguntas secretas, elija una y proporcione la respuesta correspondiente.

Al ingresar los datos, presione el botón **“Modificar”** (Figura 4.7).

Domingo 15 de Noviembre de 2009 19:23:24

[MODIFICAR CONTRASEÑA](#)

<p>Nombre de Usuario: <input type="text"/></p> <p>Nueva Contraseña: <input type="text"/></p> <p>Pregunta Secreta: <input type="text" value="Selecciona una pregunta"/></p>	<p>Contraseña Actual: <input type="text"/></p> <p>Confirma Contraseña: <input type="text"/></p> <p>Respuesta a la Pregunta Secreta: <input type="text"/></p>
--	--

[INICIO](#) [REGRESAR](#)

Figura 4.7. Pantalla para modificar la contraseña.

Una vez que el sistema admitida la nueva contraseña podrá utilizarla para autenticarse.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Módulo de Administración de Incidentes para el DID, DSA, DROS y Jefatura

Si usted pertenece a los departamentos: DID, DSA, DROS o Jefatura, el sistema le mostrará una página con las siguientes opciones de menú:

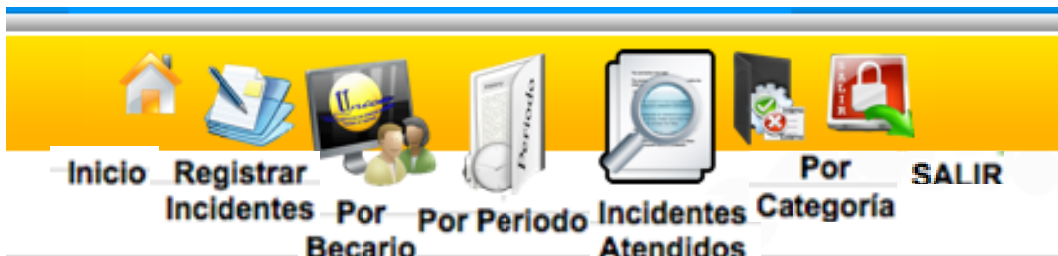


Figura 4.8. Menú del Módulo del DID, DSA, DROS y Jefatura.



Inicio

Permite ir a la página de bienvenida de este módulo.



Figura 4.9. Pantalla principal del Módulo del DID, DSA, DROS y Jefatura.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1




Registrar Incidentes

Permite realizar el registro del incidente. Al elegir esta opción el sistema muestra la siguiente pantalla (Figura 4.10):

Figura 4.10. Pantalla para el Alta de Incidentes.

Proporcione los siguientes datos:

Nombre del becario: Para desplegar el listado de los becarios registrados en el sistema, presione el icono . Se desplegará una sub-ventana con un combo que contiene los nombres de los departamentos de UNICA.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Figura 4.11. Pantalla para el Alta de Incidentes. Asignación del incidente a un becario.

Elija el **nombre del departamento** al que pertenece el becario. El sistema mostrará los nombres de los becarios activos de dicho departamento. Seleccione el **nombre del becario** a cargo del incidente, como consecuencia el sistema llenará los campos de nombre y apellidos en la página de registro.

Categoría: El incidente puede ser:

- 1) *Interno*, aquellos incidentes que se presentan dentro de UNICA,
- 2) *Externo*, aquellos incidentes que se presentan fuera de UNICA como por ejemplo, en la Secretaría o las Divisiones de la Facultad.

Por omisión, el incidente se considera como *Interno*, en caso de ser *Externo* seleccione dicha opción, para este último, el sistema desplegará un combo más con los nombres de las personas a la que se les brinda el servicio, elija el nombre correspondiente.

Tipo de apoyo: Seleccione el tipo de apoyo brindado (administrativo, seguridad, especial, respaldo, soporte técnico). En el caso de que no se encuentre enlistado, elija la opción “**Agregar otro...**”, aparecerá en el formulario un campo de texto en el que podrá introducir el nuevo tipo de apoyo.

Tiempo estimado de respuesta: Proporcione el tiempo estimado de solución del incidente, expresada en minutos, horas o días.

Status: El incidente puede tener el status *Atendido/No atendido*. Para elegir entre uno u otro seleccione el check-box

Fecha de registro: El sistema proporciona automáticamente la fecha actual del sistema en formato dd/mm/aaaa (día/mes/año).

Fecha en que se atendió: Sólo si el incidente tiene status de *Atendido*, el sistema despliega un calendario del que debe elegir la fecha en la que fue atendido (formato dd/mm/aaaa).

Descripción: Indique alguna descripción, nota u observación acerca del incidente en cuestión, este campo soporta un máximo de 250 caracteres.

Una vez capturados todos los datos, presione el botón “**Alta**” para registrar el Incidente. El sistema confirmará con el mensaje “**Su información se ha registrado en el sistema**”.




Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Consulta por becario

Permite realizar búsquedas por el nombre de becario que está a cargo de algún incidente.

Para poder consultar el nombre de los becarios registrados en el sistema presione el icono . Se desplegará una sub-ventana con un combo que contiene los nombres de los departamentos de UNICA (Figura 4.12).

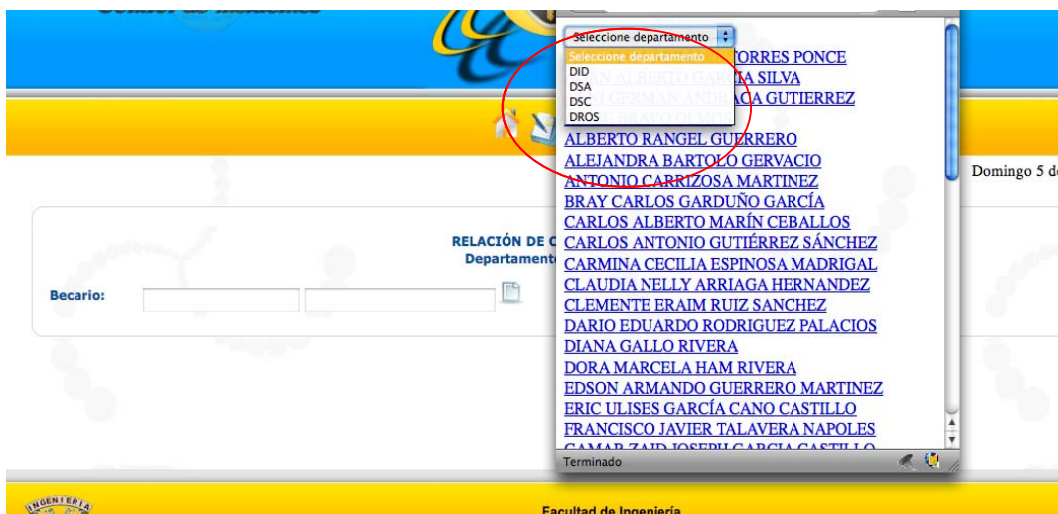


Figura 4.12. Pantalla de Consulta de Incidentes por becario. Sub-ventana con información de los departamentos y becarios de UNICA.

Elija el **nombre del departamento** al que pertenece el becario y enseguida el nombre del becario a cargo del incidente. Una vez seleccionado, el sistema llenará los campos de nombre y apellidos del becario en la página de consulta.

Presione el botón “**Buscar**” para realizar la consulta.

Si no se encuentran coincidencias para la búsqueda indicada, el sistema mostrará la leyenda: “**No se encontraron registros**”, en caso contrario, el sistema mostrará para cada incidente: *Fecha de solicitud, Fecha de atención, Periodo, Becario que Atendió, Categoría, Apoyo realizado, Tiempo Estimado de respuesta, Tiempo real de respuesta, Descripción y Status* (Figura 4.13).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

RELACIÓN DE CASOS ATENDIDOS POR BECARIO
Departamento de Investigación y Desarrollo

Becario:

Información Registrada en el Sistema

Fecha de Solicitud	Fecha de Atención	Periodo	Atendio	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009	2009-03-25	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ADMINISTRATIVO	10 minutos		nuevos datos	Atendido
09/03/2009	2009-03-09	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ESPECIALES	5 minutos		nuevos cambios	Atendido
30/08/0007	0031-01-15	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ESPECIALES	25 minutos	12 minutos	SE DEBEN DE ACTUALIZAR LAS FECHAS CAROL SECRE GENARL 25 Y 12 MIN Y ES ESPECIAL ATENDIDO Y EXTERNO	Atendido

Generar Reporte



Figura 4.13. Pantalla de Consulta de Incidentes por becario. Resultado de la consulta.

NOTA: Para generar el reporte de resultados de la consulta en formato Excel, vea **Generar reporte de Incidente del Módulo General** en este Manual.



Consulta por periodo

Permite realizar búsquedas por el periodo o semestre en el que se presentó el incidente. Al elegir esta opción, el sistema desplegará por omisión el listado de incidentes del periodo actual con los siguientes datos: *Fecha de solicitud, Fecha de Atención, Periodo, Atendió, Categoría, Apoyo realizado, Tiempo estimado de respuesta, Tiempo de respuesta, Descripción y Status (Atendido/No atendido)* (Figura 4.14).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

RELACIÓN DE CASOS ATENDIDOS POR PERIODO
Departamento de Investigación y Desarrollo

Periodo: 2009-2

Información Registrada en el Sistema

Fecha de Solicitud	Fecha de Atención	Periodo	Atendio	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	TECNICO	78 horas		NO SE	Atendido
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	SOPORTE	25 minutos		soporte ineterno ya esta atendido	Atendido
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	SOPORTE	25 minutos		soporte ineterno ya esta atendido	Atendido
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Externo	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Externo	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido
25/03/2009		2009-2	LUIS ARMANDO FLORES ALVAREZ	Externo	SEGURIDAD	7 horas		se mand AMI a consejo tecnico con romero mtz	No Atendido
25/03/2009		2009-2	LUIS ARMANDO FLORES ALVAREZ	Externo	SEGURIDAD	7 horas		se mand AMI a consejo tecnico con romero mtz	No Atendido
25/03/2009	2009-03-25	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ADMINISTRATIVO	10 minutos		nuevos datos	Atendido
24/03/2009		2009-2	DARIO EDUARDO RODRIGUEZ PALACIOS	Interno	TECNICO	45 minutos		esta es una prueba interna	No Atendido
13/03/2009	2009-03-16	2009-2	CLAUDIA NELLY ARRIAGA HERNANDEZ	Externo	ESPECIALES	10 minutos	12 minutos	Se actualizop las fechas	Atendido
13/03/2009	2009-03-13	2009-2	JORGE FLORES CRUZ	Externo	ADMINISTRATIVO	3 minutos	33 horas	Se cambio la segunda fecha y es externo direccion y chema. 33 horas	Atendido
13/03/2009		2009-2	OSCAR JOSAFAT GASCON BUSIO	Externo	TECNICO	12 horas		este debe ser una categoria INTERNA cjhacar informacion 12 dias	Atendido
09/03/2009	2009-03-09	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ESPECIALES	5 minutos		nuevos cambios	Atendido
02/03/2009	2009-03-06	2009-2	CARMINA CECILIA ESPINOSA MADRIGAL	Externo	TECNICO	12 días	5 días	se volvio externo tecnico jaramillo y secretaria admin	Atendido
23/02/2009		2009-2	VICTOR DUEÑAS TELLO	Externo	TECNICO	12 minutos		NUEVA PRUEBA ALTA EN DIRECCION	No Atendido
19/02/2009		2009-2	DARIO EDUARDO RODRIGUEZ PALACIOS	Externo	ESPECIALES	1 horas		ESTA ES UNA PRUEBA MAS..!! 19/02/09	Atendido
30/08/0007	0031-01-15	2009-2	ANTONIO CARRIZOSA MARTINEZ	Externo	ESPECIALES	25 minutos	12 minutos	SE DEBEN DE ACTUALIZAR LAS FECHAS CAROL SECRE GENARL 25 Y 12 MIN Y ES ESPECIAL ATENDIDO Y EXTERNO	Atendido

Generar Reporte

Figura 4.14. Pantalla de Consulta de Incidentes por periodo. Resultado de la consulta.

Puede consultar el listado de los incidentes registrados de un periodo específico, cambiando el periodo en el combo que se presenta, como se muestra en la siguiente imagen (Figura 4.15):

RELACIÓN DE CASOS ATENDIDOS POR PERIODO
Departamento de Investigación y Desarrollo

Periodo: 2009-2

Información Registrada en el Sistema

Fecha de Solicitud	Fecha de Atención	Periodo	Atendio	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	TECNICO	78 horas		NO SE	Atendido
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	SOPORTE	25 minutos		soporte ineterno ya esta atendido	Atendido
25/03/2009	2009-03-25	2009-2	VICTOR DUEÑAS TELLO	Interno	SOPORTE	25 minutos		soporte ineterno ya esta atendido	Atendido
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Externo	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Externo	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido

Figura 4.15. Pantalla de Consulta de Incidentes por periodo. Selección de periodos.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Una vez seleccionado el periodo, el sistema desplegará los incidentes capturados durante dicho periodo.

NOTA: Para generar el reporte de resultados de la consulta en formato Excel, vea **Generar reporte de Incidente del Módulo General** en este Manual.



Consulta por Incidentes atendidos

Permite realizar búsquedas por el status del incidente (**Atendido/No atendido**). Al elegir esta opción, el sistema desplegará por omisión el listado de incidentes con status “Atendido” con los siguientes datos: *Fecha de solicitud, Fecha de Atención, Periodo, Atendió, Categoría, Apoyo realizado, Tiempo estimado de respuesta, Tiempo de respuesta, Descripción y Status (Atendido/No atendido)* (Figura 4.16).

RELACIÓN DE CASOS ATENDIDOS
Departamento de Investigación y Desarrollo

Status:

Información Registrada en el Sistema

Fecha de Solicitud	Fecha de Atención	Periodo	Atendió	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Dirigido a: ALBERTO ESTRADA RUIZ en DIRECCIÓN	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Dirigido a: ALBERTO ESTRADA RUIZ en DIRECCIÓN	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido
25/03/2009		2009-2	LUIS ARMANDO FLORES ALVAREZ	Dirigido a: RAUL ROMERO MARTÍNEZ en CONSEJO TÉCNICO	SEGURIDAD	7 horas		se mand AMI a consejo tecnico con romero mtz	No Atendido

Figura 4.16. Pantalla de Consulta de Casos Atendidos.

Puede cambiar entre el criterio “Atendido/No Atendido” en el combo “status”, como se muestra en la siguiente imagen (Figura 4.17):

RELACIÓN DE CASOS ATENDIDOS
Departamento de Investigación y Desarrollo

Status:

Información Registrada en el Sistema

Fecha de Solicitud	Fecha de Atención	Periodo	Atendió	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Dirigido a: ALBERTO ESTRADA RUIZ en DIRECCIÓN	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido

Figura 4.17. Pantalla de Consulta de Casos Atendidos. Selección de status.

NOTA: Para generar el reporte de resultados en formato Excel, vea **Generar reporte de Incidente del Módulo General** en este Manual.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Consulta por categoría

Permite realizar búsquedas por la categoría del incidente: Interno/Externo. Al elegir esta opción, el sistema desplegará por omisión el listado de incidentes con categoría “Interno” con los siguientes datos: *Fecha de solicitud, Fecha de Atención, Periodo, Atendió, Categoría, Apoyo realizado, Tiempo estimado de respuesta, Tiempo de respuesta, Descripción y Status (Atendido/No atendido)* (Figura 4.18).

Fecha de Solicitud	Fecha de Atención	Periodo	Atendió	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
24/C3/2009		2009-2	DARIO EDUARDO RODRIGUEZ PALACIOS	Interno	TECNICO	45 minutos		esta es una prueba interna	No Atendido
25/C3/2009	2009-03-25	2009-2	VICTOR DUENAS TELLO	Interno	TECNICO	78 horas		NO SE	Atendido
25/C3/2009	2009-03-25	2009-2	VICTOR DUENAS TELLO	Interno	SOPORTE	25 minutos		soporte interno ya esta atendido	Atendido
25/C3/2009	2009-03-25	2009-2	VICTOR DUENAS TELLO	Interno	SOPORTE	25 minutos		soporte interno ya esta atendido	Atendido

Figura 4.18. Pantalla de Consulta de Casos Atendidos por categoría.

Para ver el listado de los incidentes Externos, seleccione la categoría en el combo “categoría”, como se muestra en la siguiente imagen (Figura 4.19). El sistema permite además, detallar la búsqueda indicando la División/Secretaría y persona que reportó el incidente.

Fecha de Solicitud	Fecha de Atención	Periodo	Atendió	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
24/03/2009		2009-2	DARIO EDUARDO RODRIGUEZ PALACIOS	Interno	TECNICO	45 minutos		esta es una prueba interna	No Atendido
25/03/2009	2009-03-25	2009-2	VICTOR DUENAS TELLO	Interno	TECNICO	78 horas		NO SE	Atendido

Figura 4.19. Pantalla de Consulta de Casos Atendidos por categoría. Selección de criterios de búsqueda.


Presione el botón “**Buscar**” una vez definidos los criterios de búsqueda.

NOTA: Para generar el reporte de resultados en formato Excel, vea **Generar reporte de Incidente del Módulo General** en este Manual.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Generar reporte de Incidente del Módulo General

Si desea generar el reporte en formato de hojas de cálculo Excel, presione el botón **Generar Reporte**  ubicado al final de los resultados de cada una de las Consultas: Por becario, Por periodo, Incidentes Atendidos o Por Categoría.

El sistema mostrará un cuadro de diálogo para **Guardar** el archivo, seleccione el botón **“Aceptar”**. (Figura 4.20).

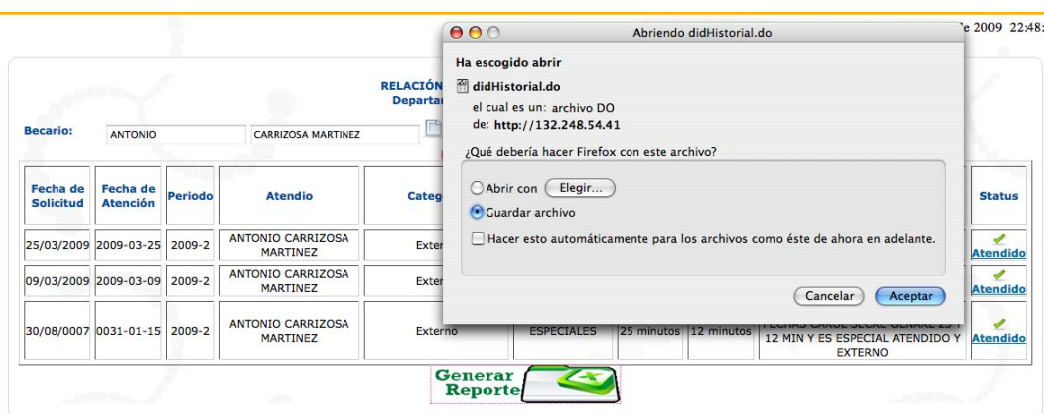


Figura 4.20. Pantalla de Generación de reporte en Excel.

NOTA IMPORTANTE: El archivo se guardará en su computadora bajo el nombre **didHistorial.do**, es necesario cambiar la extensión **.do** por **.xls** (o bien **.xlsx** para formato office 2007) para que sea reconocido por el programa Excel, así mismo se recomienda cambiar el nombre del archivo con la finalidad de llevar un mejor control de los reportes generados.

Actualizar información de Incidente

Si se desea corregir o actualizar la información de algún incidente, es necesario que primero localice el registro por medio de las opciones de Consulta (Por becario, Por periodo, Incidentes Atendidos o Por Categoría) que se presentan en el menú principal. Posteriormente, haga clic sobre la opción Atendido/No atendido del campo “Status” del registro a modificar (Figura 4.21).

Fecha de Solicitud	Fecha de Atención	Periodo	Atendió	Categoría	Apoyo Realizado	Tiempo Estimado de Respuesta	Tiempo de Respuesta	Descripción	Status
25/03/2009		2009-2	LEONARDO ALVAREZ CHIMAL	Dirigido a: ALBERTO ESTRADA RUIZ en DIRECCIÓN	SOPORTE	40 minutos		debe ser etxerno direc chema no atendido	No Atendido

Figura 4.21. Resultado de consulta. Para modificar el registro haga clic sobre la opción Atendido/No atendido del campo “Status” del registro a modificar.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

El sistema desplegará una página con los datos del incidente (Figura 4.22):

Figura 4.22. Pantalla de Actualización de Datos del Incidente.

Podrá modificar los siguientes datos:

- Becario:** Nombre del becario a cargo del incidente.
- Categoría:** Índole del incidente (Interno o Externo).
- Dirigido a:** Secretaría o División en la que se presenta el incidente (sólo para la categoría “Externo”).
- Persona:** Nombre de la persona de la Secretaría o División a la que se le atiende (en caso de que el accidente sea “Externo”).
- Tipo de apoyo:** Apoyo que se brinda en el incidente.
- Tiempo estimado de respuesta:** Tiempo estimado de atención al incidente.
- Tiempo de respuesta:** Tiempo real de atención al incidente.
- Status:** Status del incidente (Atendido/No atendido).
- Fecha de registro:** Fecha del sistema en que se registró el incidente.
- Fecha en que se atendió:** Fecha real en la que se atendió el incidente.
- Descripción:** Observación o nota del incidente.

Modifique los datos necesarios (excepto la fecha de registro del incidente) y confirme la actualización presionando el botón “**Actualizar**”.



Salir

Permite cerrar la sesión iniciada.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Módulo de Administración de Incidentes para el DSC

Si usted pertenece al Departamento de Seguridad en Cómputo (DSC), el sistema le mostrará las siguientes opciones de menú:



Figura 4.23. Menú del Módulo para el DSC.



Inicio

Permite ir a la página de bienvenida de este módulo (Figura 4.24).



Figura 4.24. Página principal del Módulo del DSC.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Registrar Incidentes

Permite dar de alta un incidente. El sistema mostrará la siguiente pantalla:



Departamento de Seguridad en Cómputo Relación de Incidentes

Asunto:

Información del Solicitante

Dirigido a:

Información del Incidente

Fecha (DD/MM/AAAA): 04/07/2009

Dirección IP:

Reporta:

Magnitud del Incidente:

Causa:

Información de la Atención al Incidente

Realizó: SANTA ROSA GARCIA CORRO
LILIANA LIZETTE TORRES SANDOVAL
ORLANDO ZALDÍVAR ESQUIVEL
JAIME ROMO GUTIERREZ
JOSE DE JESUS GALVAN ROSAS
EDSON ARMANDO GUERRERO MARTINEZ
FRANCISCO JAVIER TALAVERA NAPOLES
VICTOR GONZ EZ OLMOS
JUAN PATTI CORONA
ABRAHAM ABIMAEI TORRES PONCE
LUIS ARMANDO FLORES ALVAREZ
BRAY CARLOS GARDU? GARC?

Tiempo:

Observaciones:

Figura 4.25. Pantalla para Registrar Incidentes.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Introduzca los siguientes datos:

Asunto: El sistema desplegará los asuntos registrados anteriormente.

Seleccione dentro del listado el asunto del incidente en cuestión. Si no se encuentra, elija la opción “**Agregar Otro**” y capture el nuevo asunto.

Dirigido a: Seleccione el nombre de la persona a quien va dirigido el reporte del incidente. Si no está enlistado, elija la opción “**Agregar Otro**” y proporcione el nombre, apellidos, responsabilidad y área donde labora la persona.

Fecha: El sistema coloca la fecha actual del sistema y no podrá ser modificada.

Dirección IP: Elija la IP dentro del listado proporcionado e introduzca los últimos tres octetos de la dirección en donde se presentó el incidente. Si no se encuentra, seleccione la opción “**Agregar Otro**” y proporcione los datos requeridos.

Reporta: Seleccione la institución que reporta el incidente (DSC o DGSCA).

Magnitud del incidente: Escoja una opción para calificar este rubro: Moderado, Crítico e Importante.

Causa: Elija una causa de las que enlista el sistema, si no se encuentra seleccione la opción “**Agregar Otro**” y proporcione el nombre y la descripción de la nueva causa.

Realizó: El sistema muestra una lista con los nombres de los becarios que pueden atender los incidentes, para seleccionar a más de un becario, mantenga presionada la tecla *Ctrl.* y con el Mouse haga clic en el nombre.

Tiempo: Anote el tiempo invertido en la solución del incidente.

Observaciones: Especifique algún comentario o nota adicional al incidente.

Una vez que proporcione todos los datos requeridos, presione el botón “**Registrar**”.

Si el registro tuvo éxito, el sistema mostrará en pantalla el siguiente mensaje “**Su información se ha registrado en el sistema**”.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Búsqueda de Incidentes registrados

Al seleccionar esta opción el sistema mostrará los campos de Zona; Dirección IP; Causa; Reporta; Periodo; Año; Área; Fecha inicial y final, con los cuales puede hacer combinaciones para hacer una búsqueda más específica (Figura 4.26).



Sabado 4 de Julio de 2009 19:00:34

ATENCIÓN DE INCIDENTES DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

Zona: Causa: Periodo: Área:

Dirección IP: Reporta: Año: Fecha: al

NO TIENE NINGÚN CASO REGISTRADO

Figura 4.26. Pantalla para Consultar Incidentes. Criterios de búsqueda.

Si desea una búsqueda general en cualquiera de estos criterios elija la opción “**Todas**” (Figura 4.27) y presione el botón “**Buscar**”.



Domingo 10 de Enero de 2010 19:23:19

ATENCIÓN DE INCIDENTES DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

Zona: Causa: Periodo: Área:

Dirección IP: Reporta: Año: Fecha: al

Información Registrada en el Sistema

Oficio DSC	Atendio	Zona	Area	Fecha	Persona a quien va dirigido	Asunto	Quien Reporta	Causa	Tiempo
2	JUAN CARLOS HERNANDEZ VARGAS	A	SG	31/05/2009	CAROL NAYELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE 132.248.54.23	DGSCA	GUSANO	23 minutos

Generar Reporte


Figura 4.27. Pantalla para Consultar Incidentes. Resultado de búsqueda.

NOTA: Para generar el reporte de resultados en formato Excel, vea **Generar reporte de Incidente del DSC** en este Manual.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Generar reporte de Incidente del DSC

Si desea generar el reporte en formato de hojas de cálculo Excel, presione el botón **Generar Reporte**  ubicado al pie de los listados de cada una de las Consultas: Por incidentes registrados y Por Oficios proporcionadas en el menú principal.

El sistema mostrará un cuadro de diálogo para **Guardar** el archivo, seleccione el botón **“Aceptar”**. (Figura 4.28).

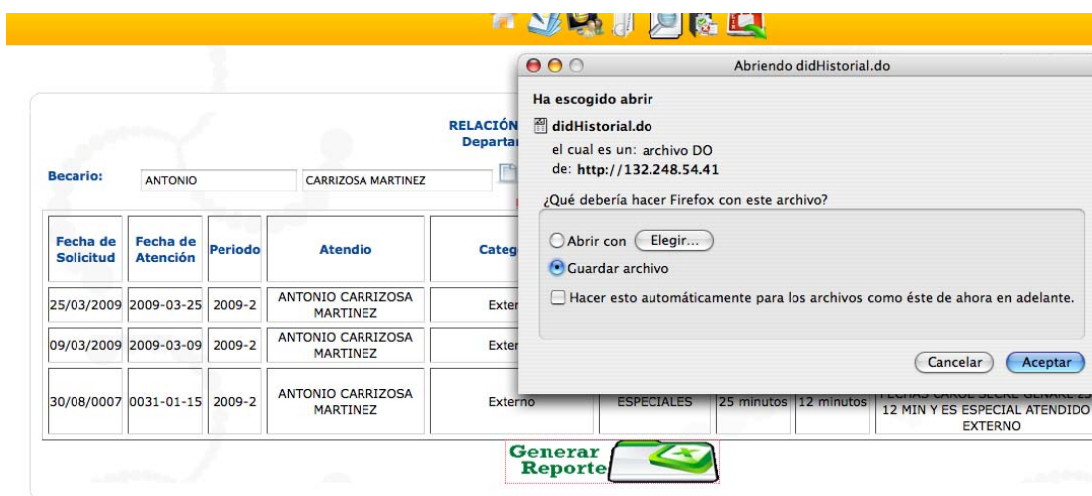


Figura 4.28. Pantalla de Generación de reporte en Excel.

NOTA IMPORTANTE: El archivo se guardará en su computadora bajo el nombre **didHistorial.do**, es necesario cambiar la extensión **.do** por **.xls** (o bien **.xlsx** para formato Office 2007) para que sea reconocido por el programa Microsoft Excel, así mismo se recomienda cambiar el nombre del archivo con la finalidad de llevar un mejor control de los reportes generados.

Actualizar información de Incidente

Si se desea corregir o actualizar la información de algún incidente, es necesario que primero localice el registro por medio de los criterios que brinda la opción "Búsqueda de Incidentes registrados" del menú principal. Posteriormente, haga clic sobre el número correspondiente al campo "Oficio DSC" del registro a modificar (Figura 4.29).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

ATENCIÓN DE INCIDENTES DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

Zona: Causa: Período: Área:

Dirección IP: Reporta: Año: Fecha: al

Información Registrada en el Sistema

Oficio DSC	Atendio	Zona	Area	Fecha	Persona a quien va dirigido	Asunto	Quien Reporta	Causa	Tiempo
4	LIDYA GARCIA LOPEZ	A	PR	10/01/2010	FERNANDO GARCIA JIMENEZ DE NADA	INCIDENTE 132.248.54.23	DSC	VIRUS	12 minutos
3	No hay becario asignado	A	PR	06/12/2009	CARLOS CARGAS CARGA JEFE DE HOUSE MASTER	INCIDENTE 132.248.54.12	DGSCA	VIRUS	12 minutos
2	JUAN CARLOS HERNANDEZ VARGAS	A	DGSCA	31/05/2009	CAROL NAYELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE 132.248.54.23	DGSCA	GUSANO	23 minutos
1	JUAN CARLOS HERNANDEZ VARGAS	A	DGSCA	31/05/2009	CAROL NAYELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE 132.248.54.78	DSC	GUSANO	12 minutos

Generar Reporte

Figura 4.29. Resultado de consulta. Para modificar el registro haga clic sobre el número del campo "Oficio DSC" del registro a modificar.

El sistema desplegará una página con los datos del incidente (Figura 4.30):

Departamento de Seguridad en Cómputo Relación de Incidentes

Asunto:

Información del Solicitante

Dirigido a:

Responsable: Área:

Nueva Área

Nombre Completo del Área:

Siglas:

Información del Incidente

Fecha (DD/MM/AAAA): 31/05/2009

Dirección IP:

Reporta:

Magnitud del Incidente:

Causa:

Información de la Atención al Incidente

Realizó:

Tiempo: minutos

Observaciones:

Figura 4.30. Pantalla para modificar los datos del Incidente.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

El sistema le permitirá realizar cambios a los siguientes campos:

Asunto: Seleccione dentro del listado el asunto del incidente en cuestión. Si no se encuentra, elija la opción “**Agregar Otro**” y capture el nuevo asunto.

Dirigido a: Seleccione el nombre de la persona a quien va dirigido el reporte del incidente. Si no está enlistado, elija la opción “**Agregar Otro**” y proporcione el nombre, apellidos, responsabilidad y área donde labora la persona.

Fecha: El sistema coloca la fecha actual del sistema y no podrá ser modificada.

Dirección IP: Elija la IP dentro del listado proporcionado e introduzca los últimos tres octetos de la dirección en donde se presentó el incidente. Si no se encuentra, seleccione la opción “**Agregar Otro**” y proporcione los datos requeridos.

Reporta: Seleccione la institución que reporta el incidente (DSC o DGSCA).

Magnitud del incidente: Escoja una opción para calificar este rubro: Moderado, Crítico e Importante.

Causa: Elija una causa de las que enlista el sistema, si no se encuentra seleccione la opción “**Agregar Otro**” y proporcione el nombre y la descripción de la nueva causa.

Realizó: El sistema muestra una lista con los nombres de los becarios que pueden atender los incidentes, para seleccionar a más de un becario, mantenga presionada la tecla *Ctrl.* y con el Mouse haga clic en el nombre.

Tiempo: Anote el tiempo invertido en la solución del incidente.

Observaciones: Especifique algún comentario o nota adicional al incidente.

Una vez que proporcione todos los datos requeridos, presione el botón “**Registrar**”.

Si el registro tuvo éxito, el sistema mostrará en pantalla el siguiente mensaje “**Su información se ha registrado en el sistema**”.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Introduzca los siguientes datos:

Asunto:

El sistema desplegará los asuntos registrados anteriormente.

Elija dentro del listado el asunto del incidente. Si no se encuentra, seleccione la opción **“Agregar Otro”** y proporcione el nuevo asunto.

Dirección IP: Elija la IP dentro del listado proporcionado e introduzca los últimos tres octetos de la dirección en donde se presentó el incidente. Si no se encuentra enlistada, elija la opción **“Agregar Otro”** y proporcione los datos requeridos.

Dirigido a: Seleccione dentro del listado proporcionado el nombre de la persona a quien va dirigido el reporte del incidente. Si no está enlistado, elija la opción **“Agregar Otro”** y proporcione el nombre, apellidos, responsabilidad y área donde labora la persona.

“Me dirijo a usted para informarle que el (la) :” Seleccione la institución que reporta el incidente (DSC o DGSCA).


“La vulnerabilidad que se marca en los reportes es que se encuentra como:” Elija una causa de las que enlista el sistema, al seleccionarla el sistema desplegará la descripción.

Si la causa no se encuentra, escoja la opción **“Agregar Otro”** y proporcione el nombre y la descripción de la nueva causa.

“Ciudad Universitaria, D.F., a” Elija en el calendario la fecha con la que se emitirá el oficio.

C.C.P.: Especifique la(s) persona(s) a la(s) cual(es) se le(s) envía copia del oficio. Si no está(n) enlistada(s), seleccione la opción **“Agregar Otro”** y proporcione el nombre, apellidos y la responsabilidad que tienen en el área donde laboran.

Una vez que proporcione todos los datos requeridos, presione el botón **“Generar Oficio”**. Si el registro tuvo éxito, el sistema mostrará en pantalla el mensaje **“El oficio se registró satisfactoriamente”**, además del enlace para la descarga del oficio generado (Figura 4.32).

	Nombre del Proyecto: Sistema para la Administración, Atención y Control de Incidentes	Clave: MAN-USU
	Institución: UNICA	Fecha: Abril 09
	Documento: Manual de Usuario	Versión: 1

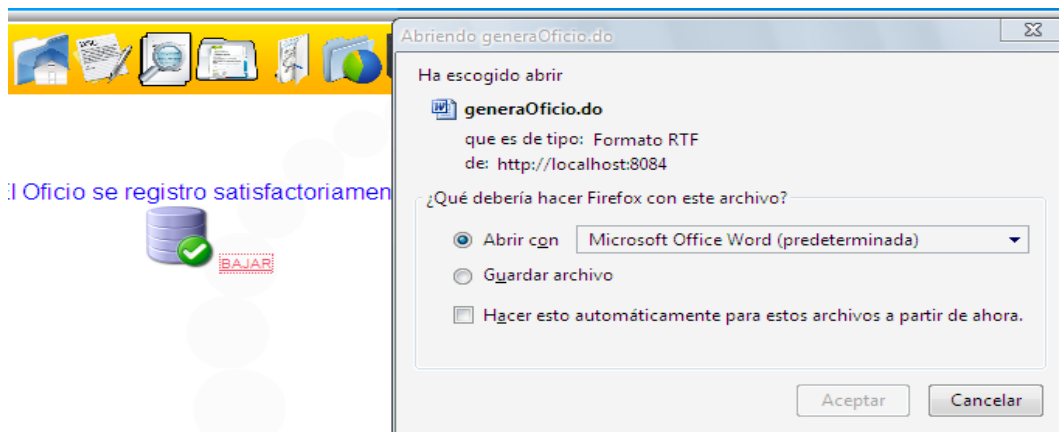


Figura 4.32. Pantalla para descargar el Oficio generado.

NOTA IMPORTANTE: El archivo se guardará en su computadora bajo el nombre **generarOficio.do**, es necesario cambiar la extensión **.do** por **.doc** (o **.docx** para Microsoft 2007) o bien **.rtf** para que sea reconocido por el programa Word, así mismo se recomienda cambiar el nombre del archivo con la finalidad de llevar un mejor control de los reportes generados (Figura 4.33).



Figura 4.33. Muestra de un Oficio generado en formato .doc.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1



Búsqueda de Oficios

Al seleccionar esta opción el sistema mostrará los campos de Zona; Dirección IP; Causa; Reporta; Periodo; Año; Área; Fecha inicial y final, con los cuales puede hacer combinaciones para hacer una búsqueda más específica (Figura 4.34).



Sábado 4 de Julio de 2009 19:00:34

ATENCIÓN DE INCIDENTES DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO

Zona: Todas | Causa: Todas | Periodo: Todos | Área: Todas

Dirección IP: Todas | Reporta: Todas | Año: | Fecha: al

[Buscar]

NO TIENE NINGÚN CASO REGISTRADO

Figura 4.34. Búsqueda de Oficios.

Si desea una búsqueda más general en cualquiera de estos criterios elija la opción **"Todas"** (Figura 4.35) que se encuentra el combo correspondiente.

ATENCIÓN DE INCIDENTES - BÚSQUEDA DE OFICIOS
Departamento de Seguridad en Cómputo

Zona: A | Causa: Todas | Periodo: Todos | Área: Todas

Dirección IP: Todas | Reporta: DGSCA | Año: | Fecha: 04/10/2009 al

[Buscar]

Información registrada en el sistema

Oficio DSC	Oficio UNICA	Zona	Area	Fecha	Persona a quien va dirigido	Asunto	Quien Reporta	Co
4	17	A	SECRETARIA GENERAL	21/11/2009	CAROL NAVELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE 132.248.34.34	DGSCA	VIS

[Generar Reporte]

Limpiar | Hoy | Cerrar

<Prev | Next>

Enero 2010

Do	Lu	Ma	Mi	Ju	Vi	Sa	Do
						1	2
3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26
27	28	29	30	31			

Domingo 10 de Enero de 2010 19:21:43

Figura 4.35. Pantalla para Consultar Oficios. Resultado de consulta.

NOTA: Para generar el reporte de resultados en formato Excel, vea **Generar reporte de Incidente del DSC** en este Manual.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Actualizar información del Oficio

Si se desea corregir o actualizar la información de algún oficio, es necesario que primero localice el registro por medio de los criterios que brinda la opción **"Búsqueda de Oficios"** del menú principal. Posteriormente, haga clic sobre la opción correspondiente al campo **"Respuesta"** del registro a modificar (Figura 4.36).

ATENCIÓN DE INCIDENTES - BÚSQUEDA DE OFICIOS
Departamento de Seguridad en Cómputo

Zona: Causa: Período: Área:

Dirección IP: Reporta: Año: Fecha: al

Información Registrada en el Sistema

Oficio DSC	Oficio UNICA	Zona	Area	Fecha	Persona a quien va dirigido	Asunto	Quien Reporta	Causa	Respuesta
5	18	A	PRUEBAME	20/02/2010	FERNANDO GARCIA JIMENEZ DE NADA	INCIDENTE 132.248.54.2	DSC	VIRUS	Sin respuesta
8	21	A	PRUEBAME	04/02/2010	FERNANDO GARCIA JIMENEZ DE NADA	INCIDENTE EN SEGURIDAD 132.248.54.255	DGSCA	VIRUS	Sin respuesta
9	22	B	DIRECCIÓN GENERAL DE SERVICIOS DE CÓMPUTO ACADÉMICO	01/02/2010	CAROL NAYELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE EN SEGURIDAD 143.666.66.23	DGSCA	VIRUS	Sin respuesta
4	17	A	DIRECCIÓN GENERAL DE SERVICIOS DE CÓMPUTO ACADÉMICO	21/11/2009	CAROL NAYELI JIMENEZ PEREZ LIDER DE PROYECTO	INCIDENTE 132.248.54.34	DGSCA	VIRUS	Sin respuesta

Figura 4.36. Resultado de consulta. Para modificar el registro haga clic sobre el contenido del campo "Respuesta" del registro a modificar.

El sistema desplegará una página con los datos del Oficio (Figura 4.37.):

Domingo 8 de Agosto de 2010 10:27:43

Departamento de Seguridad en Cómputo Relación de Oficio Generado

Asunto:

Información del Solicitante

Dirigido a:

Responsable: Área:

Información del Oficio

Fecha (DD/MM/AAAA):

Dirección IP:

Reporta:

Causa:

Descripción: muy denso virus

Personas que tienen dirigido una copia del Oficio

Agregar otro...

Respuesta al Oficio:

Figura 4.37. Página que permite la actualización de datos de un Oficio.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

El sistema le permitirá modificar los siguientes datos:

Asunto:

El sistema desplegará los asuntos registrados anteriormente.

Elija dentro del listado el asunto del incidente. Si no se encuentra, seleccione la opción **“Agregar Otro”** y proporcione el nuevo asunto.

Dirección IP: Elija la IP dentro del listado proporcionado e introduzca los últimos tres octetos de la dirección en donde se presentó el incidente. Si no se encuentra enlistada, elija la opción **“Agregar Otro”** y proporcione los datos requeridos.

Dirigido a: Seleccione dentro del listado proporcionado el nombre de la persona a quien va dirigido el reporte del incidente. Si no está enlistado, elija la opción **“Agregar Otro”** y proporcione el nombre, apellidos, responsabilidad y área donde labora la persona.

“Me dirijo a usted para informarle que el (la):” Seleccione la institución que reporta el incidente (DSC o DGSCA).

“La vulnerabilidad que se marca en los reportes es que se encuentra como:” Elija una causa de las que enlista el sistema, al seleccionarla el sistema desplegará la descripción.

Si la causa no se encuentra, escoja la opción **“Agregar Otro”** y proporcione el nombre y la descripción de la nueva causa.

“Ciudad Universitaria, D.F., a” Elija en el calendario la fecha con la que se emitirá el oficio.

C.C.P.: Especifique la(s) persona(s) a la(s) cual(es) se le(s) envía copia del oficio. Si no está(n) enlistada(s), seleccione la opción **“Agregar Otro”** y proporcione el nombre, apellidos y la responsabilidad que tienen en el área donde laboran.

Una vez que modifique los datos, presione el botón **“Actualizar Oficio”**. Si el registro tuvo éxito, el sistema mostrará en pantalla el mensaje **“El oficio se actualizó satisfactoriamente”**.



Estadísticas

Podrá generar estadísticas de los incidentes atendidos por los siguientes criterios: Área; Año; Causa; Fuente y Origen; Quien Reporta; Zona y por todos los anteriores, especificándolo en el combo *Estadísticas por*.

Es necesario que indique el año del cual desea el reporte (Figura 4.38).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

Bienvenido ING. RAFAEL SANDOVAL VAZQUEZ

UNICA

Sistema para la Administración, Atención y Control de Incidentes

SAACI

Sabado 4 de Julio de 2009 19:05:17

ATENCIÓN DE INCIDENTES
Departamento de Seguridad en Cómputo

Estadísticas por:

Año:

Facultad de Ingeniería
Unidad de Servicios de Cómputo Académico
UNICA - SAACI

Figura 4.38. Pantalla de la opción para generar Estadísticas. El sistema muestra en pantalla las siguientes gráficas dependiendo del criterio elegido (Figuras 4.39 a 4.43):

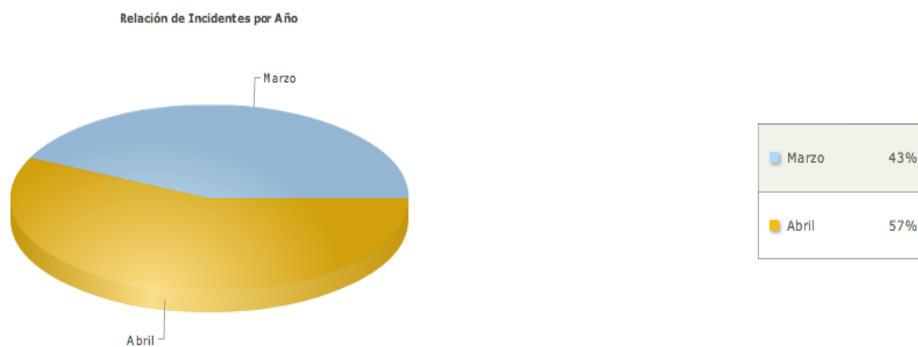


Figura 4.39. Gráfica de Porcentaje de Incidentes del Año seleccionado detallado por mes. (Opción: Estadísticas por Año).



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

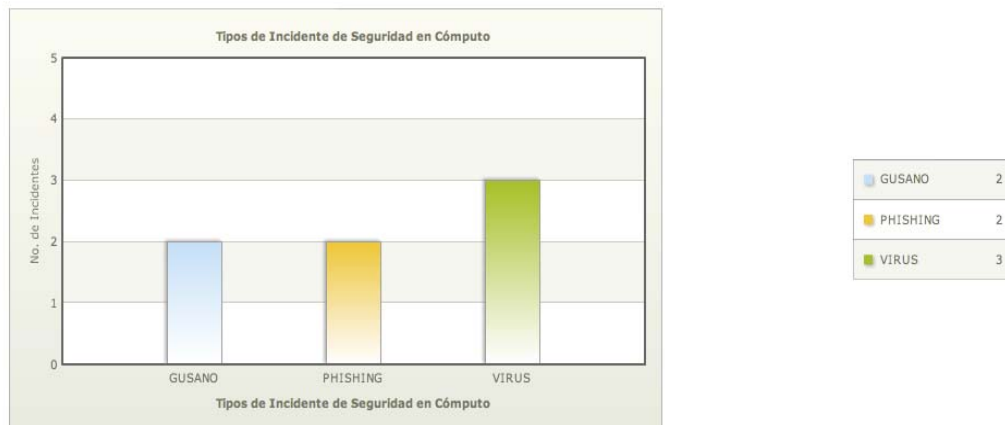


Figura 4.40. Gráfica de Estadísticas por causa del Incidente en el año seleccionado.



Figura 4.41. Gráfica de Estadísticas por origen del reporte (Quién reporta) en el año seleccionado.

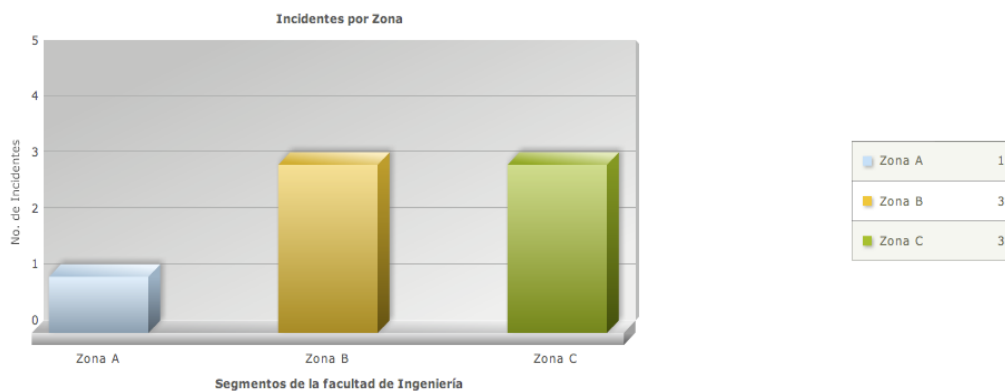


Figura 4.42. Gráfica de Estadísticas de Incidentes por Zona en el año seleccionado.



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

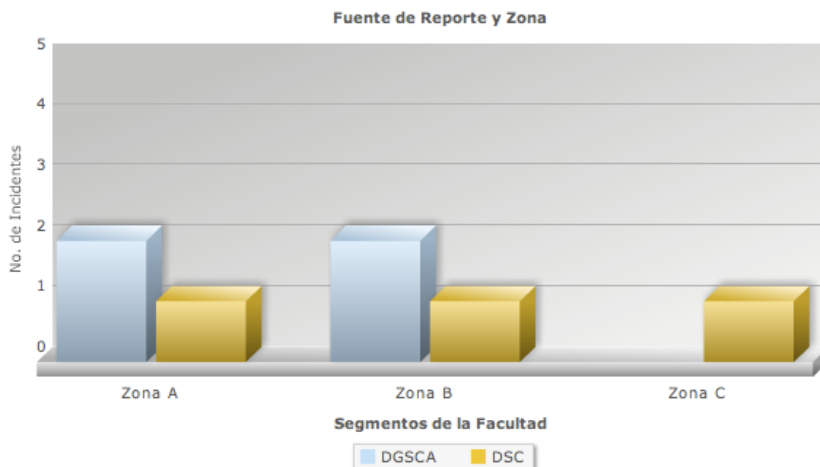



Figura 4.43. Gráfica de Estadísticas de Incidentes por Origen de reporte (quién reporta) y Zona en el año seleccionado.

Solamente cuando se realiza la búsqueda por la opción **TODOS** en el combo

Estadísticas por. El sistema muestra el final de las gráficas el icono  con el cual puede generar un documento Word con el resumen y gráficas resultado de la consulta del año especificado (Figura 4.44)


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO
DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO
ATENCIÓN A INCIDENTES:

Elaborado por:
Ing. Rafael Sandoval Vázquez

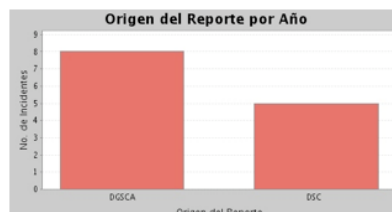
Atención a Incidentes de Seguridad en Cómputo:

El Departamento de Seguridad en Cómputo atendió y dio seguimiento hasta su finalización a:




Gráfica 1. Incidentes por zona.

De los cuales la fuente del reporte fue:



Gráfica 2. Origen del reporte.

Los reportes provenientes de DGSCA y del DSC se dividen de la siguiente forma por zona:



Gráfica 3. Fuente de reporte por zona



Nombre del Proyecto:	Sistema para la Administración, Atención y Control de Incidentes	Clave:	MAN-USU
Institución:	UNICA	Fecha:	Abril 09
Documento:	Manual de Usuario	Versión:	1

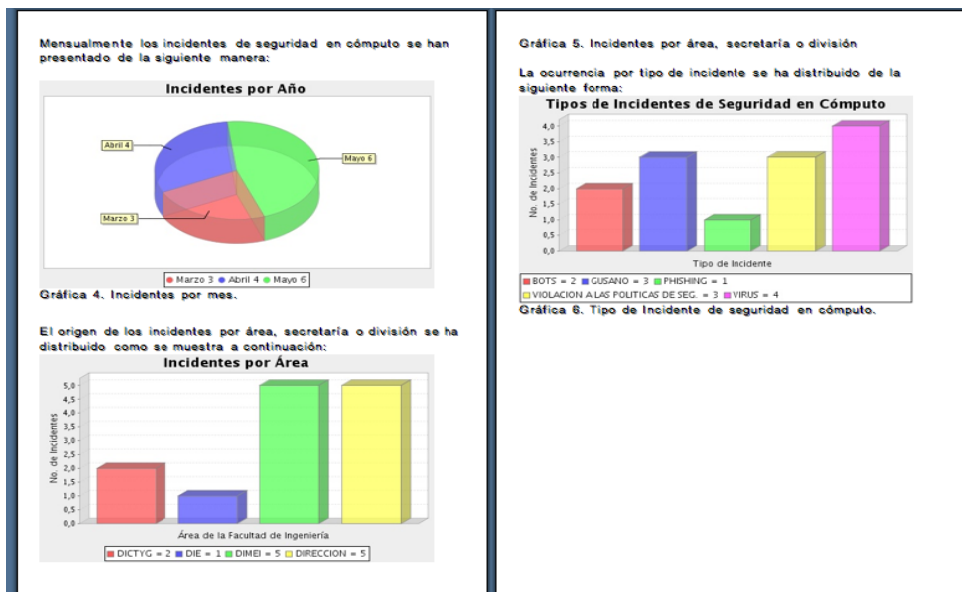


Figura 4.44. Ejemplo del Documento de Word con las Estadísticas de Incidentes que genera el sistema para el año indicado.

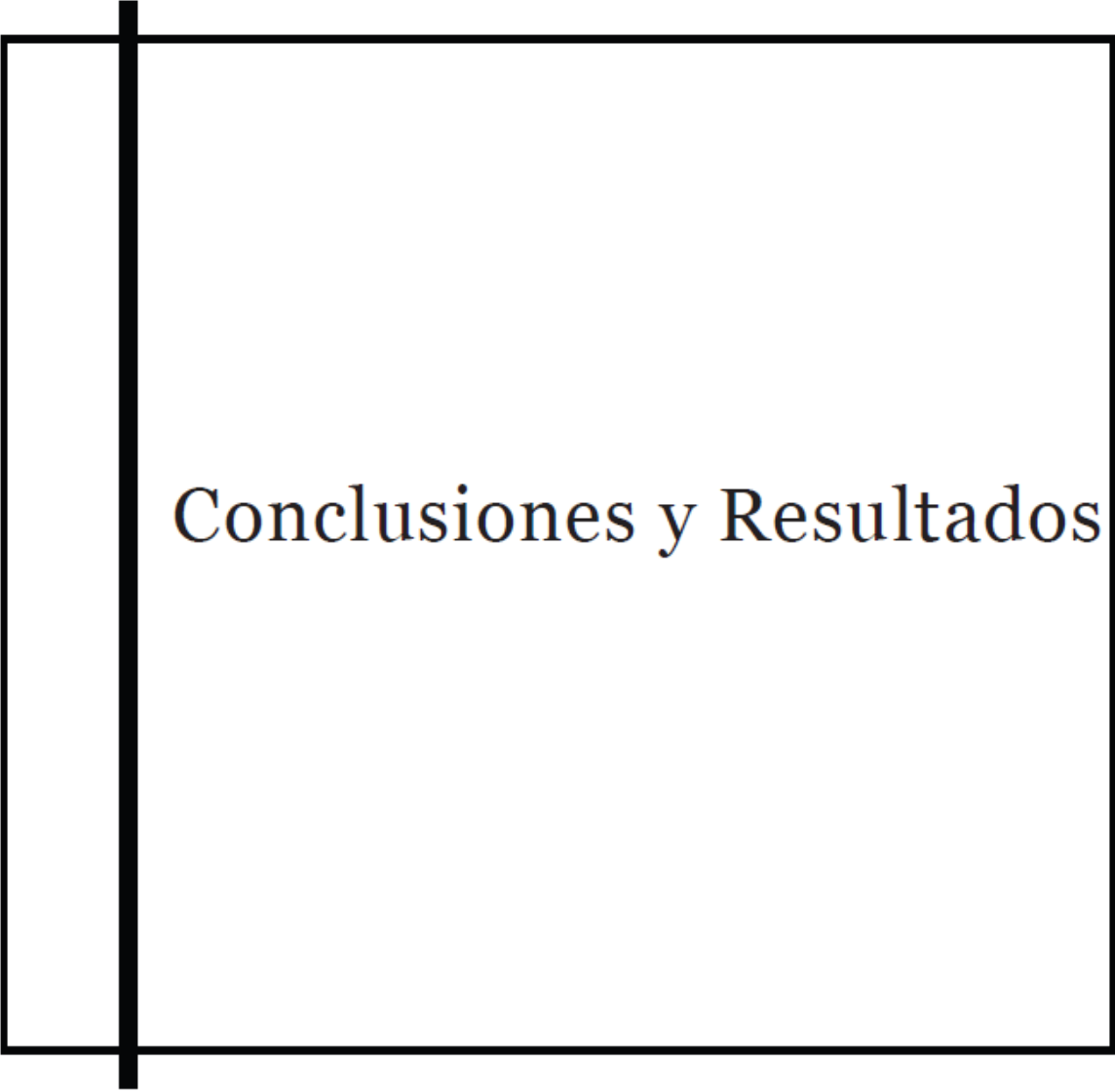


Salir

Permite cerrar la sesión iniciada.

Responsables del SAACI

Antonio Carrizosa Martínez



Conclusiones y Resultados

CONCLUSIONES

CONTRIBUCIONES DE LA TESIS

Se desarrolló un Sistema Web para la Administración, Atención y Control de Incidentes (SAACI) como solución a la necesidad de los Jefes de la Unidad de Servicios de Cómputo Académico (UNICA) de llevar un control detallado de las actividades que realiza el personal a su cargo, a fin de generar estadísticas que les permitan poder detectar puntos de mejora en la calidad del servicio.

Con la creación de este proyecto se puso en práctica la metodología RUP (*Rational Unified Process o bien Proceso Unificado de Racional*) la cual es la más utilizada para el análisis, implementación y documentación de sistemas Orientados a Objetos. Con ella, el modelado de requerimientos y la estructura del sistema se facilitaron gracias a la utilización de diagramas de UML (*Unified Modeling Language o Lenguaje Unificado de Modelado*).

Para el desarrollo del sistema se utilizaron herramientas de software libre, que no involucran pagos de licencias, como es el lenguaje de programación Java, junto con los frameworks de Apache Struts e iBatis, que permitieron separar el sistema en capas tomando como referencia el patrón de diseño Modelo-Vista-Controlador (MVC); entre los beneficios de la utilización de este modelo se pueden destacar la reducción de los tiempos en el desarrollo y la facilidad de adaptar el sistema a los posibles cambios en requerimientos posteriores.

Algunas de las ventajas que tiene SAACI sobre el proceso actual para el registro, control y seguimiento de los incidentes o apoyos son las siguientes:

- ✓ Al automatizar el proceso, se evita el traspapeleo de documentos; se agiliza la generación de oficios y se evade que la información se guarde en archivos de Excel.
- ✓ La información de los incidentes o apoyos realizados por parte de los becarios de UNICA se encuentra centralizada y disponible para su consulta, lo cual permite darle un mejor seguimiento al servicio y facilita la obtención de parámetros de evaluación para establecer procesos de mejora continua.
- ✓ El registro de la atención de incidentes es mucho más fácil, amigable y sobre todo seguro.

SAACI se divide en dos módulos: un Módulo para el Departamento de Seguridad en Cómputo y un Módulo General para el resto de los Departamentos de UNICA. Al sistema, sólo podrán acceder usuarios registrados y de acuerdo a sus privilegios podrán:

- Generar y consultar oficios; Consultar y generar estadísticas de los Incidentes de seguridad en cómputo por los siguientes criterios: periodo o año y fecha de atención; zona donde se presentó; dirección IP de la máquina donde se detectó; causa del incidente (virus, actividad intrusiva, entre otras); e Institución que lo reportó.

- Consultar los Incidentes de categoría general por los siguientes criterios: nombre del becario asignado a la actividad; periodo de atención (semestre en el que se presenta el incidente); status (atendido/no atendido); o categoría del incidente (interno, externo).

PERSPECTIVAS A FUTURO

- Que la información recopilada por SAACI, sirva como complemento al Sistema de Control y Evaluación de Becarios (SICEB) que se utiliza actualmente en UNICA para medir el desempeño del personal a su cargo.
- Brindar capacitación del software empleado para el desarrollo del sistema al personal de UNICA para que con ello puedan dar mantenimiento a SAACI en futuros requerimientos y asimismo les sirva como herramienta para el ámbito laboral.
- Construir un módulo adicional que atienda las necesidades de control de los incidentes de la Coordinación de Salas de Cómputo (CSC).



Glosario de Términos

GLOSARIO DE TÉRMINOS

ADORNO.- Detalle de la especificación de un elemento que se añade a su notación gráfica básica.

ARTEFACTO.- Pieza discreta de información que es utilizada o producida por un proceso de desarrollo de software o un sistema existente.

AJAX.- (Asynchronous JavaScript And XML). Técnica de desarrollo Web para crear aplicaciones interactivas que son ejecutadas en el cliente mientras mantienen comunicación asíncrona con el servidor en segundo plano. Gracias a esto último es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo cual repercute en el aumento de la interactividad, velocidad y usabilidad de las aplicaciones.

CARRIL.- Participación sobre un Diagrama de Actividades para organizar las responsabilidades de las acciones.

CLASES WRAPPERS.- Clases que modelan los tipos de datos primitivos tales como enteros y flotantes, precisamente estos tipos primitivos son los únicos elementos en Java que no son clases. Estas clases se encuentran en **java.lang**.

DID.- Siglas del Departamento de Investigación y Desarrollo de UNICA.

DSA.- Siglas del Departamento de Servicios Académicos de UNICA.

DSC.- Siglas del Departamento de Seguridad en Cómputo de UNICA.

FRAMEWORK.- Es un patrón arquitectónico que proporciona un plantilla extensible para aplicaciones dentro de un dominio. Es posible ver a un framework como una micro arquitectura que incluye un conjunto de mecanismos que colaboran entre sí para resolver un problema en un dominio en común. Cuando se especifica el framework, se especifica el esqueleto de una arquitectura, junto a los elementos variables, que se muestran a los usuarios que quieren adaptar el framework a su propio contexto.

HTML.- Siglas de **HyperText Markup Language** (*Lenguaje de Mercado de Hipertexto*). Es un lenguaje para codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. Es usado para crear páginas Web.

IBATIS (APACHE).- Framework de código abierto basado en capas que se ocupa de la capa de Persistencia (situada entre la lógica de Negocio y la capa de la Base de Datos). Asocia objetos de modelo (JavaBeans) con sentencias SQL o procedimientos almacenados mediante archivos descriptores XML, simplificando la utilización de bases de datos). Puede ser implementado en Java y .NET.

INCIDENTE.- Para fines específicos de este documento, llamaremos Incidente a aquel suceso o evento que requiere de la atención y solución por parte de algún departamento de UNICA. Ej. Incidentes de seguridad en cómputo, peticiones

administrativas o de soporte (arreglo de impresoras), desarrollo de pequeñas aplicaciones, entre otras.

ITERACIÓN.- Es un mini proyecto que tiene como resultado una versión interna de cada uno de los artefactos que pueden ser generados en un proceso de desarrollo de software.

JAVASCRIPT.- Lenguaje de programación script interpretado por el navegador Web. Se utiliza principalmente embebido en HTML u otros lenguajes similares, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas. Fue creado por Brendan Eich en Netscape, actualmente es soportado por todos los navegadores Web.

JAVA SERVLETS.- Es un programa que se ejecuta en un servidor, su uso más común es generar páginas Web de forma dinámica a partir de los parámetros de la petición que envíe el navegador Web. Se usan como alternativas a los programas CGI por ser persistentes, es decir, que una vez que han sido iniciados, se mantienen en memoria y pueden satisfacer múltiples solicitudes, en contraste, los programas CGI desaparecen una vez que han satisfecho una solicitud.

JSP.- (JavaServer Pages). Se trata de una tecnología Java desarrollada por Sun Microsystems, que permite generar contenido dinámico para Web, en forma de documentos HTML , XML o de otro tipo. Esta tecnología permite que códigos Java y ciertas otras acciones predefinidas, sean integrados en contenido estático.

MÉTODO.- Implementación de una operación.

MVC (MODELO-VISTA-CONTROLADOR).- Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la *vista* es la página HTML y el código que provee de datos dinámicos a la página. El *modelo* es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el *controlador* es el responsable de recibir los eventos de entrada desde la vista.

OBJETO.- Entidad con unos límites bien definidos e identidad que encapsula estado y comportamiento; instancia de una clase.

OFICIO.- Documento que es generado por el Departamento de Seguridad en Cómpu para informar a alguna Entidad que se detectó un problema de seguridad dentro de su infraestructura de red.

PRODUCTO.- Artefacto de desarrollo, como los modelos, el código, la documentación y los planes de trabajo.

REQUISITO.- Característica, propiedad o comportamiento deseado de un sistema.

RTF.- (Rich Text Format). Formato para documentos desarrollado por Microsoft desde 1987 para el intercambio de documentos en múltiples plataformas. La mayoría de los procesadores de texto pueden leer y escribir documentos RTF. La extensión de los archivos de este formato suele ser ".rtf".

UNICA.- Siglas de la **Unidad de Servicios de Cómputo Académico**.

UML.- Lenguaje Unificado de Modelado, un lenguaje para visualizar, especificar, construir y documentar artefactos de un sistema con gran cantidad de software.

VISTA.- Proyección de un modelo, que se ve desde una perspectiva o un punto de vista dado, y que omite entidades que no son relevantes desde esa perspectiva.

VISTA DINÁMICA.- Aspecto de un sistema que destaca su comportamiento.

VISTA ESTÁTICA.- Aspecto de un sistema que destaca su estructura.

XML.- Acrónimo del inglés eXtensible Markup Language (Lenguaje de marcado extensible) desarrollado por el World Wide Web Consortium (W3C). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. XML se propone además como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo entre otros.

FUENTES CONSULTADAS

BIBLIOGRAFÍA

- ANDREU**, Rafael, Joan Ricart, et. al., *Estrategia y Sistemas de información*, Editorial Mc-Graw-Hill, Madrid, 1991.
- BOOCH**, Grady, *El Lenguaje Unificado de Modelado*, 2da. ed., Editorial Pearson Educación, Madrid, 2006.
- DEITEL**, Paul J., *Cómo programar en Java*, 7ma. ed., Editorial Pearson Educación, México, 2008.
- DE PABLO**, I., *El reto informático. La gestión de la información en la empresa*, Editorial Pirámide, Madrid, 1989.
- DORAY**, Arnold, *Programación con Apache Struts*. Editorial Anaya Multimedia, México, 2007.
- FERRY**, Lucey, *Management Information Systems for Change*, DP Publicaciones, 1991.
- PRESSMAN**, Roger S., *Ingeniería del Software. Un enfoque práctico*, 6ta. ed., Editorial McGraw Hill, 2005.
- RUMBAUGH**, Jacobson I and Booch G., *El Lenguaje Unificado de Modelado. Manual de Referencia.*, Editorial Addison Wesley, Rational Software Corporation, 2000.
- SCHMULLER**, Joshep, *Aprendiendo UML en 24 horas*, Editorial McGraw-Hill, México, 2000.
- SOMMERVILLE**, Ian, *Ingeniería del Software*, 6ta. ed., Editorial Addison Wesley, México, 2002.

MESOGRAFÍA

- Página oficial del proyecto Apache Tomcat.** Disponible desde internet en: <http://tomcat.apache.org/> [acceso 15 Febrero de 2010].
- Página oficial de iBatis.** Disponible desde internet en: <http://ibatis.apache.org/> [acceso 02 Octubre de 2009].
- Página oficial del proyecto de iBatis, actualmente llamado MyBatis.** Disponible desde internet en: <http://www.mybatis.org>, <http://code.google.com/p/mybatis/> [acceso 28 Mayo de 2010].
- Página oficial de JAVA.** Disponible desde internet en: <http://www.java.com/es/> [acceso 02 Febrero de 2010].
- Página oficial del proyecto PostgreSQL.** Disponible desde Internet en: <http://www.postgresql.org/> [acceso 23 Marzo de 2010].

iBatis. Disponible desde internet en:
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJAR/iBatis>
[acceso 02 Octubre de 2009].

Introducción a UML. Disponible desde Internet en:
<http://gidis.ing.unlpam.edu.ar/downloads/pdfs/IntroduccionUML.PDF>
E, [acceso 12 Enero de 2010].

Desarrollo Ágil de Software. Disponible desde internet en:
http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software
[acceso 12 Agosto de 2009].

Desarrollo iterativo e incremental, Disponible desde Internet en:
<http://www.proyectosagiles.org/desarrollo-iterativoincremental>>,
[acceso 16 de Diciembre de 2009].

Proyectos ágiles, desarrollo iterativo e incremental. Disponible desde internet en: <http://www.proyectosagiles.org/desarrollo-iterativo-incremental>, [acceso 12 Agosto de 2009].

Manual de Umbrello UML Modeller. Disponible desde Internet en:
<http://docs.kde.org/stable/es/kdesdk/umbrello/index.html>,
[acceso 07 Julio de 2009].