



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE  
PUEBLA**

*FACULTAD DE CIENCIAS DE LA COMPUTACIÓN*

**TEMA:**

**SISTEMA WEB PARA LA ADMINISTRACIÓN DE SERVICIOS  
INFORMÁTICOS DE LOS LABORATORIOS DE CÓMPUTO DE LA FCC**

**TESIS PROFESIONAL**

**PARA OBTENER EL TÍTULO DE  
INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA:**

**JOSÉ ABRAHAM BAEZ BAGATELLA**

**DIRIGIDA POR:**

**M.C. MIGUEL RODRÍGUEZ HERNÁNDEZ**

**PUEBLA, PUEBLA OCTUBRE DE 2014**

# ÍNDICE

<b>INDICE</b>	2
<b>INTRODUCCIÓN</b>	5
<b>CAPÍTULO 1. MARCO TEÓRICO.</b>	7
<b>1.1. Ingeniería de Software.</b>	8
<b>1.2. Metodología RUP.</b>	8
1.2.1. <i>El proceso unificado de Rational.</i>	8
1.2.2. <i>Características esenciales.</i>	9
1.2.3. <i>Ciclo de Vida.</i>	12
1.2.4. <i>¿Que es UML?</i>	14
1.2.5. <i>Modelos de UML.</i>	15
<b>1.3. Modelos de proceso de software.</b>	20
1.3.1. <i>Modelo lineal.</i>	20
1.3.2. <i>Modelo en espiral.</i>	22
1.3.3. <i>Modelo de prototipos.</i>	23
1.3.4. <i>Proceso unificado de desarrollo de software.</i>	25
1.3.5. <i>Modelo incremental.</i>	26
<b>1.4. Bases de datos.</b>	28
1.4.1. <i>Introducción y definición de Bases de Datos.</i>	28
1.4.2. <i>DBMS.</i>	28
1.4.3. <i>Usuarios de la base de datos.</i>	31
<b>1.5. Modelos de bases de datos.</b>	32
1.5.1. <i>Modelos de datos.</i>	32
1.5.2. <i>El modelo Entidad-Relación (E-R).</i>	33
1.5.3. <i>Modelo relacional.</i>	35
1.5.4. <i>Llaves primarias y foráneas.</i>	36
<b>1.6. Normalización.</b>	37
1.6.1 <i>Definición de normalización.</i>	37
1.6.2. <i>Formas normales.</i>	38
1.6.2.1. <i>Primera forma normal (1FN).</i>	38
1.6.2.2. <i>Segunda forma normal (2FN).</i>	39
1.6.2.3. <i>Tercera forma normal (3FN).</i>	41

<b>1.7. Arquitectura cliente-servidor.</b>	42
1.7.1. <i>Código abierto.</i>	42
1.7.2. <i>Apache.</i>	44
1.7.3. <i>PHP.</i>	45
1.7.4. <i>MySQL.</i>	46
1.7.5. <i>AJAX.</i>	47
1.7.6. <i>RIA.</i>	48
<b>CAPITULO 2. ANÁLISIS Y DISEÑO (ORIENTADO A OBJETOS).</b>	50
<b>2.1. Introducción.</b>	51
<b>2.2. Planteamiento del problema.</b>	51
<b>2.3. Objetivo general.</b>	52
<b>2.4. Requerimientos del problema.</b>	52
2.4.1. <i>Requerimientos funcionales.</i>	52
2.4.2. <i>Requerimientos no funcionales.</i>	53
<b>2.5. Alcances.</b>	54
<b>2.6. Metodología.</b>	55
<b>2.7. Análisis del sistema.</b>	56
2.7.1. <i>Glosario de términos.</i>	56
2.7.2. <i>Diagrama de casos de uso.</i>	56
2.7.3. <i>Especificación de casos de uso.</i>	60
2.7.4. <i>Escenarios.</i>	80
2.7.5. <i>Diagrama de clases.</i>	94
<b>2.8. Diseño del sistema.</b>	95
2.8.1. <i>Diagramas de realización de casos de uso.</i>	95
2.8.2. <i>Diagramas de colaboración.</i>	105
2.8.3. <i>Diagrama de clases de diseño.</i>	115
2.8.4. <i>Diagramas de secuencia.</i>	118
<b>CAPÍTULO 3. DISEÑO DE LA BASE DE DATOS.</b>	131
<b>3.1. Diseño conceptual de la base de datos.</b>	132
3.1.1. <i>Modelo entidad-relación.</i>	132
3.1.2. <i>Descripción de entidades y relaciones.</i>	133

3.1.3.	<i>Normalización.</i>	135
3.1.4.	<i>Diseño lógico.</i>	142
<b>CAPITULO 4. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA.</b>		151
<b>4.1.</b>	<b>Implementación.</b>	152
<b>4.2.</b>	<b>Implementación de la interfaz.</b>	158
4.2.1.	<i>Página de inicio.</i>	158
4.2.2.	<i>Interfaces de usuario.</i>	161
4.2.3.	<i>Interfaz del colaborador.</i>	161
4.2.4.	<i>Interfaz del coordinador.</i>	163
4.2.5.	<i>Interfaz del administrador.</i>	163
<b>4.3.</b>	<b>Pruebas del sistema.</b>	166
4.3.1.	<i>Prueba 1. Instalación del sistema.</i>	166
4.3.2.	<i>Prueba 2. Agregar usuarios al sistema.</i>	168
4.3.3.	<i>Prueba 3. Registrar nuevo módulo.</i>	170
4.3.4.	<i>Prueba 4. Autenticación y acceso seguro.</i>	172
4.3.5.	<i>Prueba 5. Manipulación del inventario de hardware.</i>	173
4.3.6.	<i>Prueba 6. Trabajando con "Autoacceso".</i>	176
4.3.7.	<i>Prueba 7. Generar reportes.</i>	178
4.3.8.	<i>Prueba 8. Reportar incidentes.</i>	181
4.3.9.	<i>Prueba 9. Crear respaldos, eliminar registros y restaurar.</i>	184
4.3.10.	<i>Prueba 10. Manipular información personal.</i>	187
<b>CONCLUSIONES.</b>		189
<b>TRABAJOS A FUTURO.</b>		191
<b>BIBLIOGRAFÍA.</b>		192

# INTRODUCCIÓN

La administración de recursos en una organización significa la asignación eficiente de este medio a cada miembro o elemento del organismo. Las tareas de administración a menudo son muy extenuantes y es necesaria una larga cantidad de tiempo, continua reorganización y personal suficiente, incluido un líder o experto, para que pueda hacerse de manera eficaz.

Un sistema de gestión de recursos facilita esta tarea mediante la tecnología informática, pudiendo organizar datos de manera eficiente y además ahorrando tiempo en la reorganización y visualización de la información. Es decir el ingreso y expulsión de datos están controlados por un sistema que es necesario organizar una sola vez por un experto en el área, de manera que la información sea constante y entendible. Por medio de estos gestores los líderes o administradores pueden visualizar estadísticas y tomar decisiones respecto a estas.

Mediante la investigación realizada al campo donde este sistema será implementado, los laboratorios de cómputo de la “Facultad de Ciencias de la Computación” en la Benemérita Universidad Autónoma de Puebla, se ha podido crear una visión homogénea, combinando el sistema de administración que se utiliza actualmente y las necesidades de automatización y estadísticas, dando como resultado el planteamiento de un sistema capaz de realizar la administración eficiente de los recursos manejados en estas instalaciones.

Actualmente el sistema ha completado su fase de desarrollo y cumple con todas las características propuestas.

## ***I. Organización de inventarios***

La primera de las seis partes cruciales que componen el sistema, el registro de cada pieza física y lógica es acumulado en inventarios digitales, estos son organizados y separados según su función.

## ***II. Acceso libre***

También llamado “autoacceso”, esta sección del sistema permite la regulación del acceso a las instalaciones y crear el registro de ellas de manera digital y limpia.

### ***III. Incidentes***

Es necesario conservar una bitácora de los desperfectos ocurridos, con el fin de que estos puedan grabarse de manera durable y tengan una base solida para sustentar el acontecimiento el sistema ofrece una manera homogénea de crear reportes de estos fallos.

### ***IV. Reportes***

Los reportes son realmente estadísticas de utilización de los recursos del laboratorio, el administrador en jefe y el personal que labora en estos podrá saber como se ha estado utilizando, trabajando y corrigiendo la situación de las instalaciones.

### ***V. Personal***

El personal que labora es reclutado por los coordinadores y estos deben crear un registro en papel que corrobore su asignación. Además para poder tener acceso al sistema estos nuevos colaboradores deberán ser dados de alta por su coordinador.

### ***VI. Sistema***

El administrador en jefe del sistema puede correr tareas simples de respaldo, eliminación y restauración de datos sin necesidad de contactar directamente al administrador de datos del servidor y exponer la seguridad e integridad del sistema.

Con estos seis componentes se crea un sistema para la administración de los recursos informáticos de los laboratorios de cómputo de la FCC el cual ha sido nombrado como "SALC-FCC", acrónimo de "Sistema de Administración de los Laboratorios de Computo de la FCC", un sistema a la medida y con crecimiento deseable según la necesidad de los tiempos. Esta tesis explica la creación, desarrollo e implementación del sistema.

**CAPÍTULO 1**  
**MARCO TEÓRICO**

## **1.1 INGENIERÍA DE SOFTWARE.**

Un proyecto de software surge siempre por alguna necesidad, tales como negocios, servicios, producción, sistematización, etc. En los últimos años se ha vuelto una industria que se dirige a una gran variedad de tecnologías y áreas de aplicación. Para estandarizar una forma de producir software con alta calidad es que surge la ingeniería de software. Abran, Moore, Bourque, Dupis (2004) en su publicación Guide to the Software Engineering Body of Knowledge definen la ingeniería de software como: “La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software” (p. 1-1).

Un ingeniero de software debe tener un enfoque metódico y jerárquico en su trabajo, como disciplinado de la ingeniería deberá usar todas las teorías, métodos y herramientas convenientes de manera que siempre busque descubrir la solución a los problemas, así como las técnicas de gestión, producción y desarrollo de proyectos de software, con el objetivo de que el producto final sea confiable, eficiente y de calidad.

## **1.2 METODOLOGÍA RUP**

### ***1.2.1 El proceso unificado de Rational***

Jacobson, Booch, Rumbaugh (1999) establecen que: “El proceso unificado es un proceso de desarrollo de software y a su vez, un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en software” (p. 4).

Este proceso utiliza como base el lenguaje unificado de modelado (Unified Modeling Language, UML) y está basado en componentes, es decir la conexión precisa de elementos de software con la interfaz de usuario. El proceso unificado es una generalización de recomendaciones de trabajo para generar una amplia variedad de software, el cual se rige por 3 constantes, conducido por casos de uso, centrado en la arquitectura, iterativo e incremental.

## **1.2.2 Características esenciales**

### *Conducido por casos de uso*

El concepto de usuario puede referirse a una persona o grupo de personas, así como un sistema o sistemas que logren una acción con el software desarrollado y esta sea mutua. Este tipo de interacción se le nombra caso de uso. La definición de Jacobson et al. (1999) es: “Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema” (p. 5).

Un caso de uso es la respuesta a la pregunta ¿Qué debe hacer el sistema? Y también ¿Cada persona puede hacer lo mismo en el sistema? Es aquí donde cada usuario que va a tener dicho sistema será clasificado y recibirá el nombre de actor. Un actor puede tener los mismos o diferentes casos de uso en un mismo sistema. La buena elección de actores como la buena definición de los casos de uso es de igual importancia en el desarrollo.

### *Centrado en la arquitectura*

El termino arquitectura de software es muy similar al utilizado en la construcción, características como la estructura, estética, servicios, seguridad y mejoras son análogas y en ambos casos un buen plano auxilia al constructor a tener una imagen de lo que se desea fabricar. La arquitectura de software contiene en su estructura dos aspectos clave, los dinámicos y los estáticos para darle una base sólida pero adaptable a las necesidades del cliente, y debe considerar factores tales como la plataforma en la que será utilizado y los elementos reutilizables.

La estética del producto es también un elemento clave en el entendimiento y eficacia, la cual es bien definido por parte de los clientes y los desarrolladores. Así como en una edificación se ofrecen servicios de agua, gas, o electricidad, en un software también se ofrecen servicios varios, como fiabilidad, seguridad, eficiencia y más.

Un buen diseño de arquitectura para el software es fundamental para el desarrollo solido de este, pero ¿Qué relación tiene la arquitectura con los casos de uso? La respuesta a esta interrogante es interactiva y evolutiva, ninguno de los dos puede ser suficiente por sí misma. El planteamiento del modelo de casos de uso debe guiar a la arquitectura por qué camino ir, sin embargo en el momento del diseño arquitectónico es posible que

algunos aspectos del modelo de casos de uso cambie, ambos irán evolucionando. La única diferencia podría ser que el diseño arquitectónico debe cumplir con todos los casos de uso planteados.

El arquitecto de software debe poder crear un sistema capaz de mejorar y evolucionar, no solo en su desarrollo, sino en posteriores generaciones. Jacobson et al. (1999) sugiere tres puntos para guiar al arquitecto durante el proceso.

- Crea un esquema en borrador de la arquitectura, comenzando por la parte de la arquitectura que no es específica de los casos de uso (por ejemplo, la plataforma). Aunque esta parte de la arquitectura es independiente de los casos de uso, el arquitecto debe poseer una comprensión general de los casos de uso antes de comenzar la creación del esquema arquitectónico.
- A continuación, el arquitecto trabaja con un subconjunto de los casos de uso especificados, con aquellos que representen las funciones clave del sistema en desarrollo. Cada caso de uso seleccionado se especifica en detalle y se realiza en términos de subsistemas, clases, y componentes.
- A medida que los casos de uso se especifican y maduran, se descubre más de la arquitectura. Esto, a su vez, lleva a la maduración de más casos de uso.

Este proceso continúa hasta que se considere que la arquitectura es estable (p. 6).

### *Iterativo e incremental*

Cuando se lleva a cabo un proyecto de software de un mediano o gran alcance el esfuerzo de trabajo por parte del equipo ya sea trabajando en secuencia o paralelamente la tarea, puede llegar a ser muy grande y tardar incluso años antes de llegar al producto final. Para trabajos de este tipo varios expertos recomiendan dividir el proyecto en tareas más pequeñas con equipos reducidos, cumpliendo metas cortas para alcances largos.

Sin embargo el hecho de dividir un proyecto en muchos mini-proyectos no soluciona el problema, pues aunque puedan ahora trabajarse más cosas y presentar un avance rápido, es posible que al ensamblar todas las partes, estas no combinen. Para esto se necesita una organización y un control de trabajo. La iteratividad se fundamentara en dos factores, primeramente identificar los casos de uso que logran una mayor utilidad del proyecto y segundo, localizar cuales son los riesgos más graves del proyectos y tomar las medidas de prevención y contención. Jacobson et al. (1999) recomiendan:

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos —como suele suceder— el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque (p. 7).

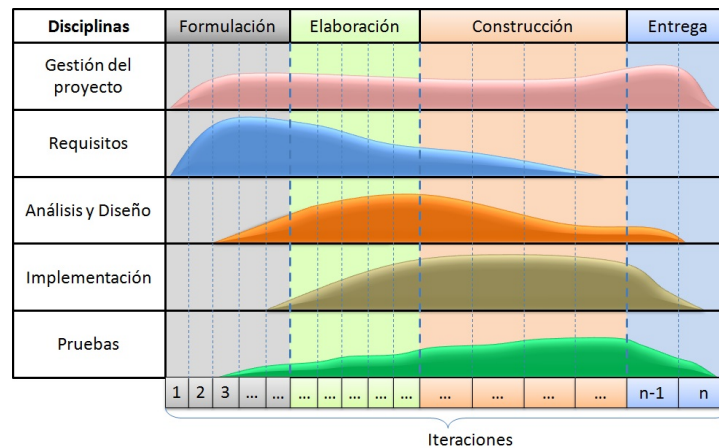
Un proyecto de software puede generar muchos costos, una forma de economizar es solo realizar las iteraciones necesarias para el objetivo del proyecto, pero siempre debe contarse con que pueden ocurrir errores no previstos o falta de elementos de software no planeadas, lo cual incrementará las iteraciones y por supuesto el costo. Un proceso iterativo bien planeado y controlado ofrece muchos beneficios, según Jacobson et al. (1999) cuatro de estos son:

- La iteración controlada reduce el coste del riesgo a los costes de un solo incremento. Si los desarrolladores tienen que repetir la iteración, la organización sólo pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.
- La iteración controlada reduce el riesgo de no sacar al mercado el producto en el calendario previsto. Mediante la identificación de riesgos en fases tempranas del desarrollo, el tiempo que se gasta en resolverlos se emplea al principio de la planificación, cuando la gente está menos presionada por cumplir los plazos. En el método “tradicional”, en el cual los problemas complicados se revelan por primera vez en la prueba del sistema, el tiempo necesario para resolverlos normalmente es mayor que el tiempo que queda en la planificación, y casi siempre obliga a retrasar la entrega.
- La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente.
- La iteración controlada reconoce una realidad que a menudo se ignora —que las necesidades del usuario y sus correspondientes requisitos no pueden definirse completamente al principio. Típicamente, se retinan en iteraciones sucesivas. Esta forma de operar hace más fácil la adaptación a los requisitos cambiantes (p. 7-8).

### 1.2.3 Ciclo de vida

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

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto (Wikipedia, 2013).



**Figura 1.2.1. Flujo de trabajo e iteraciones del proceso unificado.**

Fuente: Gestion de proyectos de Software. (2014). Proyecto Latin. Recuperado de <http://escritura.proyectolatin.org/gestion-de-proyectos-de-software/ejemplos-de-procesos/>

Cada fase termina con un hito, el cual tiene muchos objetivos, y uno de los más importantes es cuando los directores deben tomar decisiones para que el trabajo pueda continuar con la siguiente fase. Los hitos permiten a los desarrolladores controlar el progreso. La figura 1.2.1 muestra una aproximación de cómo fluye el trabajo durante cada iteración y a su vez durante cada fase.

Durante la fase de inicio se analiza el negocio y se propone la idea del producto final. Jacobson et al. (1999) sugiere que esta fase responde a las preguntas: “¿Cuáles son las principales funciones del sistema para sus usuarios más importantes? ¿Cómo podría ser la arquitectura del sistema? ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?”(p. 11). Estas preguntas pueden contestarse con los primeros casos de uso, un arquitectura improvisada y con esto empezar estimar el tiempo y esfuerzo para desarrollar el proyecto.

En la fase de diseño se trabajarán paralelamente los detalles de los casos de uso y la arquitectura del sistema. Esto dará la estructura arquitectónica del modelo de casos de uso y la implementación. En este punto el director del proyecto puede planificar el esfuerzo y el tiempo para terminar el proyecto.

Durante la fase de implementación, como su nombre lo dice, es momento de crear el producto en base a la arquitectura. Con esto se obtiene la primera versión del producto terminado, pero hay que preguntarse ¿cubre el producto las necesidades de algún usuario en este punto?

La fase de transición o pruebas es aquel periodo en el que el beta producto es probado para encontrar deficiencias y defectos. El equipo de desarrollo debe tomar en cuenta todos los errores e ineficiencias que las pruebas revelen.

Cada ciclo es capaz de generar una nueva versión final del software, la cual debe ser preparada y lista para su entrega. Dicho producto debe ser capaz de satisfacer las necesidades del cliente y de los futuros usuarios. Para esto, el software debe incluir todos los requisitos, casos de uso, arquitectura sólida, modelo de análisis, modelo de diseño, modelo de despliegue, implementación, casos de prueba, requerimientos no funcionales (como seguridad y eficacia) y los artefactos ejecutables. Todos estos modelos mencionados, Jacobson et al. (1999) los describen de la siguiente forma:

- Un modelo de casos de uso, con todos los casos de uso y su relación con los usuarios.
- Un modelo de análisis, con dos propósitos: Refinar los casos de uso con más detalle y establecer la asignación inicial de funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento.
- Un modelo de diseño que define: (a) la estructura estática del sistema en la forma de subsistemas, clases e interfaces y (b) los casos de uso reflejados como colaboraciones entre los subsistemas, clases, (c) interfaces.

- Un modelo de implementación, que incluye componentes (que representan al código fuente) y la correspondencia de las clases con los componentes.
- Un modelo de despliegue, que define los nodos físicos (ordenadores) y la correspondencia de los componentes con esos nodos.
- Un modelo de prueba, que describe los casos de prueba que verifican los casos de uso.
- Y, por supuesto, una representación de la arquitectura(p.10).

Dichos modelos representan al sistema completamente, y cada uno de estos representa el funcionamiento del sistema detalladamente.

#### **1.2.4 ¿Qué es UML?**

UML o lenguaje unificado de modelado es un Lenguaje estándar para la visualización especificación construcción y documentación de los sistemas de software. Permite visualizar los resultados por medio de esquemas están avisados. Este lenguaje contiene un vocabulario de tres categorías: elementos relaciones y diagramas, cada uno de ellos con su propia complejidad, Jacobson et al. (1999) los describen:

Hay cuatro tipos de elementos: estructurales, de comportamiento, de agrupación y de anotación. Hay siete tipos principales de elementos estructurales: casos de uso, clases, clases activas, interfaces, componentes, colaboraciones y nodos. Hay dos tipos de elementos de comportamiento: interacciones y máquinas de estados. Hay cuatro tipos de agrupaciones: paquetes, modelos, subsistemas y marcos de trabajo. Y hay un tipo principal de elementos de anotación: notas.

Dentro de la segunda categoría, la de relaciones, encontramos de tres tipos: de dependencia, de asociación y de generalización.

Y en la tercera categoría, la de diagramas, UML proporciona nueve tipos: diagramas de caso de uso, de clases, de objetos, de secuencia, de colaboración, de estados, de actividad, de componentes y de despliegue (p. 408).

También contiene una notación gráfica para cada uno de los elementos de las tres categorías. Donde en su tercera categoría, es una conjunción de todos los posibles elementos estructurales y relacionales.

### 1.2.5 Modelos de uml

En el momento de la planeación del proyecto de desarrollo, tanto el equipo desarrollador como el cliente pueden tener visiones diferentes de lo que se necesita con el próximo software. Es por eso que debe haber una línea intermedia, una representación de las ideas físicamente, de manera que ambas partes puedan opinar y acordar las bases y construcción del producto. Se propone el uso de artefactos llamados “modelos”. Un modelo es una abstracción de un objeto o idea que existe en el mundo real o que es posible pensar, es decir que es posible llevar la idea de la necesidad de software a un modelo tangible.

Existen varios modelos propuestos para esta tarea, los mas utilizados son el modelo de dominio, negocio, casos de uso, análisis, diseño, implementación y pruebas. Cuando todos estos se implementan se tiene una vision bien detallada del proyecto que permitirán futuras mejoras ahorrando tiempo al equipo de desarrollo.

*Modelo de dominio.* Un modelo hablado a alto nivel, que nombra los objetos más importantes en el contexto del sistema, estos objetos son obtenidos por medio de entrevistas con los clientes capturando requisitos. Siguiendo la definición de modelo, estos objetos pueden ser tangibles es decir, por ejemplo tanto consumibles almacenados en una bodega como las notas y facturas que se expiden cuando son comerciados.

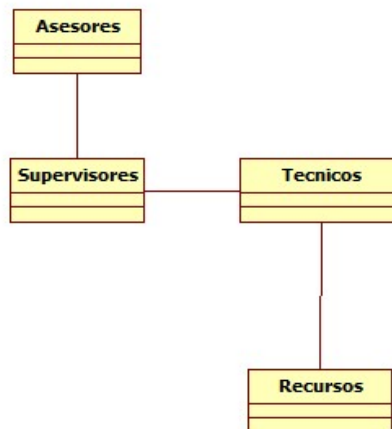
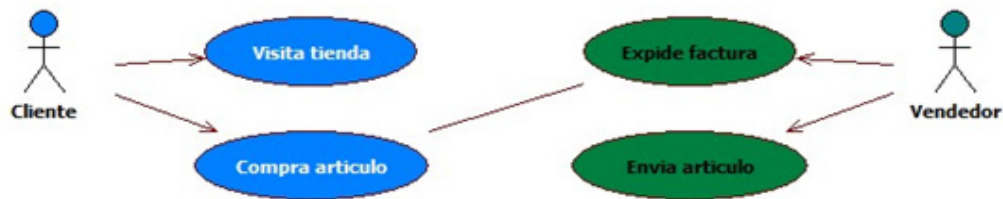


Figura 1.2.2. Modelo de dominio como diagrama de clases en UML

Este modelo puede ser descrito mediante el diagrama de clases de UML, como lo muestra la figura 1.2.2. Cada objeto en este diagrama denota cada objeto que el equipo de desarrollo considera debe desarrollarse y si es posible, como se jerarquizan entre ellos.

*Modelo de negocio.* Por el lado del cliente, esta es una técnica sugerida para comprender el funcionamiento del negocio, en este modelo se conocen los actores del negocio y se describe la interacción entre los actores del proceso y los clientes, un acercamiento a como serán los futuros casos de uso. Por ejemplo vease la figura 1.2.3:



**Figura 1.2.3. Modelo del negocio como diagrama de casos de uso en UML**

El cliente visita la tienda del vendedor y se interesa por un artículo.

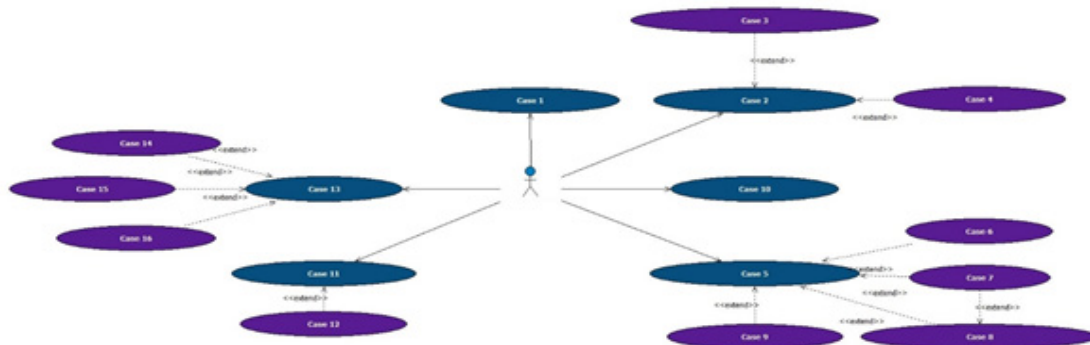
El vendedor expide una factura al cliente por dicho recurso en moneda internacional.

El cliente paga la factura al vendedor el valor correspondiente.

El vendedor envía o entrega el artículo al cliente.

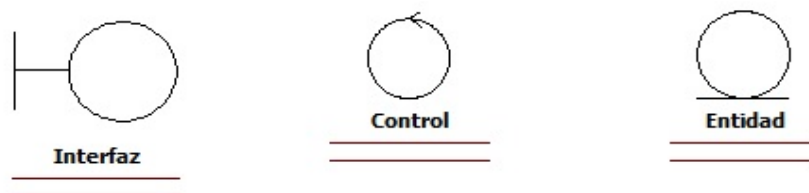
Se pueden utilizar dos diagramas de UML para describir estas interacciones, el diagrama de casos de uso y el de objetos.

*Modelo de casos de uso.* Este modelo es el que permite a los desarrolladores y a los clientes llegar a un acuerdo en lo que el cliente necesita del software y lo que el equipo puede desarrollar. Este modelo es la visión fundamental del análisis y desarrollo del sistema. La mejor descripción de este modelo la definen Jacobson et al. (1999): “Un modelo de casos de uso es un modelo del sistema que contiene actores, casos de uso y sus relaciones” (p.127). Mediante el diagrama de casos de uso de UML puede generarse este modelo de forma gráfica y amigable, vease la figura 1.2.4.



**Figura 1.2.4. Modelo de casos de uso como diagrama de casos de uso en UML.**

*Modelo de análisis.* El modelo de análisis se compone por paquetes de análisis, esto es abstracciones de los objetos que interactúan y sus relaciones, una vez abstraídos son llamados “clases de análisis”. Los actores del modelo de casos de uso interactúan con estas clases de análisis para cumplir todas las tareas especificadas como casos de uso, es posible visualizar los pasos que debe seguir cada tarea de manera singular y todos los objetos que la componen. Las clases de análisis son estereotipadas de tres maneras como muestra la figura 1.2.5: entidad, control e interfaz.



**Figura 1.2.5. Estereotipos de clases de analisis en UML.**

La clase interfaz como su nombre lo plantea, es la comunicación física entre el actor y el sistema mediante la limitante de la pantalla, solo lo que se muestra en pantalla es lo que el usuario podrá hacer. Estas interfaces pueden ser formularios, paneles, interfaces de comunicación, sensores, etc.

La clase entidad es una representación del componente del sistema dedicado a guardar información de forma duradera, por ejemplo una base de datos o un fichero de almacenamiento en disco, y a menudo estas clases son directamente tomadas de los objetos propuestos por el diagrama de clases, en el modelo de dominio o bien por el diagrama de objetos en el modelo de negocio.

La clase control representan la secuencia y coordinación de los mensajes entre clases de análisis, un control puede conectar la comunicación entre una clase interfaz y una clase entidad por ejemplo. Es recomendable utilizar estos controles para el paso de mensajes entre cualquier clase de análisis pues así habrá una sincronización entre ellas y se evitan los colapsos de información.

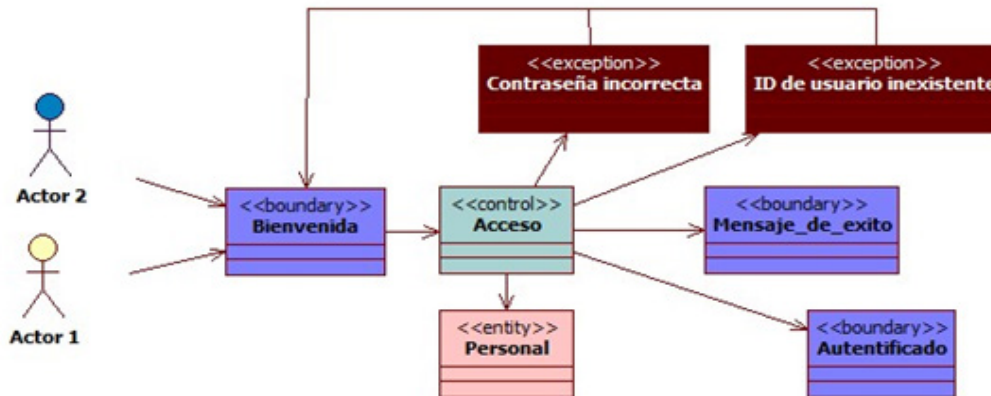


Figura 1.2.6. Modelo de análisis como diagrama de realización de casos de uso en UML

En el modelo de análisis, se pueden usar los diagramas propuestos en UML, el diagrama de realización de casos de uso (figura 1.2.6) y colaboración.

*Modelo de diseño.* Siguiendo los pasos de los diagramas de realización de casos de uso, este modelo describe todos los pasos de mensajes, secuencias y realización de cada caso de uso, tomando en cuenta los requisitos funcionales y no funcionales que intervienen en este así como las restricciones del entorno. Este modelo impacta directamente en la implementación del sistema. Se compone de distintos artefactos como las clases de diseño, realización de caso de uso-diseño, subsistema de diseño, interfaz, etc.

Una clase de diseño es una abstracción completa de un objeto, preferentemente ya considerado en los diagramas de realización de casos de uso. Deben detallarse todos sus atributos y operaciones, los parámetros que recibirá de construcción, los tipos de datos, etc.

La realización de caso de uso-diseño es la descripción específica de cómo se realiza un caso de uso, utilizando las clases de diseño y sus operaciones, a pesar de que no se mantendrá tal y como está durante el desarrollo del software, es una gran base para la implementación. Este puede reflejarse en un diagrama de UML conocido como diagrama de secuencia (figura 1.2.7).

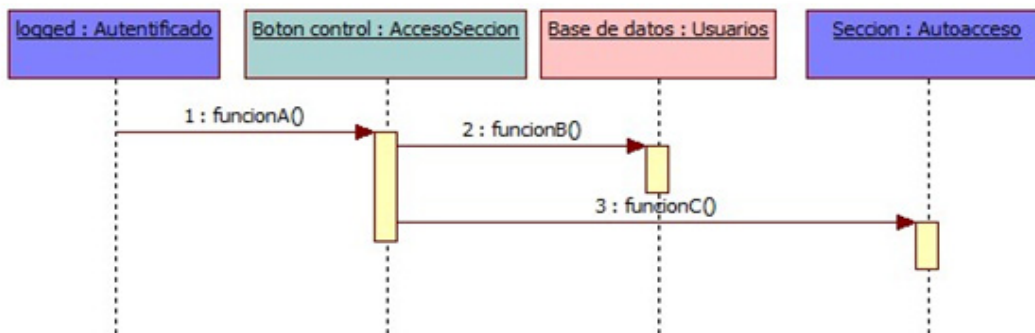


Figura 1.2.7. Modelo de diseño como diagrama de secuencia en UML

*El modelo de implementación.* Una transformación de cada modelo de diseño pasando de medio a bajo nivel, Jacobson et al. (1999) lo definen como: “El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, etc.”(p. 257).

Un componente es un encapsulamiento de los elementos de un modelo, el cual se estereotipa de 5 maneras: ejecutable, archivo, biblioteca, tabla de datos y documento. Esta interacción puede entenderse mejor con el diagrama propuesto mediante UML

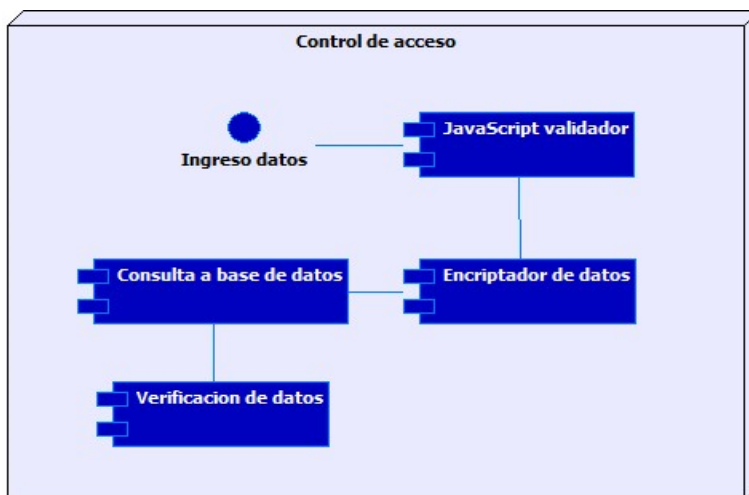


Figura 1.2.8. Modelo de implementación como diagrama de componentes en UML

llamado diagrama de componentes (figura 1.2.8).

*Modelo de pruebas.* Este describe las pruebas hechas a cada componente, como fue hecha cada prueba y los resultados obtenidos. En este modelo se utiliza como base el artefacto caso de prueba que especifica cómo debe ser probado el sistema y los componentes. Estas especificaciones fueron tratadas con anterioridad como los requisitos del sistema. Posiblemente el modelo más extenso, Jacobson et al. (1999) lo

definen como:

- Un caso de prueba que especifica cómo probar un caso de uso o un escenario específico de un caso de uso. Un caso de prueba de este tipo incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y pos condiciones especificadas por el caso de uso y que se sigue la secuencia de acciones especificadas por el caso de uso. Obsérvese que un caso de prueba basado en un caso de uso especifica típicamente una prueba del sistema como “caja negra”, es decir, una prueba del comportamiento observable externamente del sistema.
- Un caso de prueba que especifica cómo probar una realización de caso de uso-diseño o un escenario específico de la realización. Un caso de prueba de este tipo puede incluir la verificación de la interacción entre los componentes que implementan dicho caso de uso. Obsérvese que los casos de prueba basados en una realización de caso de uso típicamente especifican una prueba del sistema como “caja blanca”, es decir, una prueba de la interacción interna entre los componentes del sistema (p. 284).

## 1.3 MODELOS DE PROCESO DE SOFTWARE

### 1.3.1 Modelo Lineal

El modelo lineal o en cascada sugiere el desarrollo de manera secuencial y sistematizada mediante cuatro fases: análisis, diseño, código, prueba (véase la figura 1.3.1).



Figura 1.3.1. Fases del modelo de desarrollo lineal.

Esta forma básica de desarrollo cuenta con las siguientes actividades:

- Ingeniería y modelado de sistemas/información.
- Análisis de los requisitos del software.
- Diseño.
- Generación de código.
- Pruebas.
- Mantenimiento.

Siendo éste un modelo secuencial tiene la ventaja de estar organizado desde un principio y no mezclar las fases. Sin embargo, este modelo por sí solo carece del potencial para abarcar grandes proyectos pues es difícil que los clientes en una sola reunión expongan todos los requisitos, y también es muy posible que el producto final contenga muchos errores.

*Ingeniería y modelo de sistemas/información.* Como el software siempre forma parte de un sistema más grande (O empresa), el trabajo comienza estableciendo requisitos de los elementos del sistema y asignando al software algún subgrupo de estos requisitos. Esta visión del sistema es esencial cuán de software se debe interconectar con otros elementos como hardware, personas y bases de datos. La ingeniería y el análisis de sistemas comprenden los requisitos que se recogen en el nivel del sistema con una pequeña parte de análisis y de diseño. La ingeniería de información abarca los requisitos que se recogen en el nivel de empresa estratégico y en el nivel del área de negocio.

*Análisis de los requisitos del software.* El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del (los) programa (s) a construirse, el ingeniero («analista») del software debe comprender el dominio de información del software (descrito en el Capítulo II), así como la función requerida, comportamiento, rendimiento e interconexión.

*Diseño.* El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos del programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso del diseño traduce requisitos en una representación del software donde se puede evaluar su calidad antes de que comience la codificación.

*Generación de código.* El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada la generación de código se realiza mecánicamente.

*Pruebas.* Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

*Mantenimiento.* El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (por ejemplo: se requiere un cambio debido a un sistema operativo o dispositivo periférico nuevo), o porque el cliente requiere mejoras funcionales o de rendimiento. El soporte y mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo. (Pressman, 2002).

### 1.3.2 Modelo en espiral

El modelo en espiral fue propuesto en 1988 por el ingeniero informático Barry Boehm. La propuesta era evolucionar el modelo de desarrollo de software del solo una secuencia a un ciclo de ellas, Sommerville (2005) detalla cuatro actividades por ciclo:

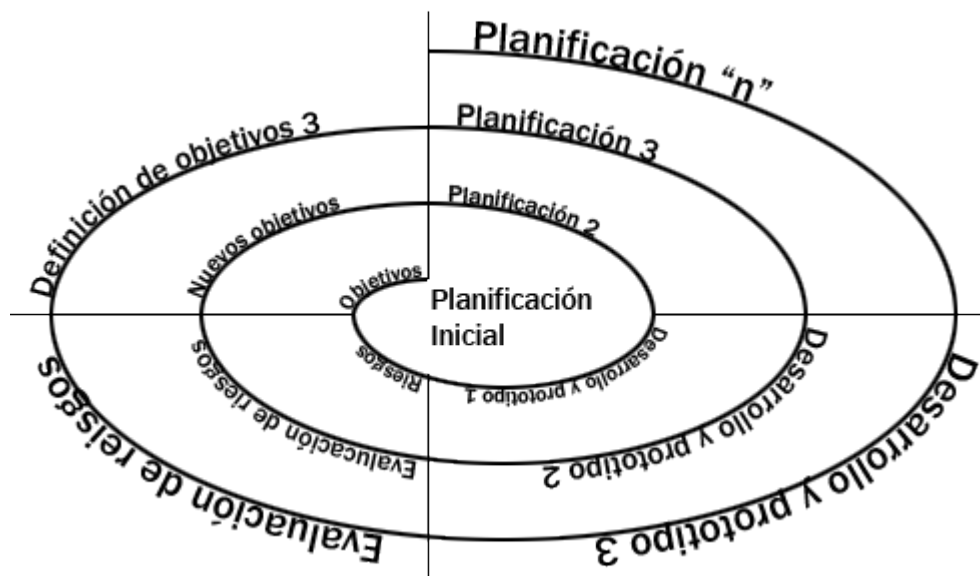


Figura 1.3.2. Fases y visualización del modelo de desarrollo en espiral.

1. *Definición de objetivos.* Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso y el producto, y se traza un plan detallado de gestión. Se identifican los riesgos del proyecto. Dependiendo de estos riesgos, se planean estrategias alternativas.

2. *Evaluación y reducción de riesgos.* Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto identificados. Se define los pasos para reducir dichos riesgos. Por ejemplo, si existe el riesgo de tener requerimientos inapropiados, se puede desarrollar un prototipo del sistema.

*3. Desarrollo y validación.* Después de la evaluación de riesgos, se eligió un modelo para el desarrollo del sistema. Por ejemplo, si los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de seguridad son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado, y así sucesivamente. El modelo en cascada puede ser el más apropiado para el desarrollo si el mayor riesgo identificado es integración de los subsistemas.

*4. Planificación.* El proyecto se revisa y se toman la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se desarrollan los planes para la siguiente fase del proyecto (p. 68-69).

El ciclo de esta espiral comienza con la definición de objetivos y detallar la funcionalidad. Se propone las formas de alcanzar estos objetivos y se evalúan los riesgos que cada uno de estos objetivos contiene, así como el costo y el número de iteraciones requeridas para completar el software. Después de revisado el plan por el cliente y el equipo de desarrollo se procede a generar un prototipo, este es probado y validado, esto ayudará al equipo de desarrollo a encontrar los errores de fabricación y a analizar qué requisitos del software se han cumplido.

A diferencia del modelo en cascada, el modelo en espiral es capaz de identificar riesgos y reunir mejor los requisitos por lo cual puede utilizarse para proyectos de mayor tamaño produciendo software de calidad. Las desventajas es que es un método muy tardado por lo cual lo hace muy costoso, además debe contar con personal muy experimentado en identificación de riesgos.

### ***1.3.3 Modelo de prototipos***

El paradigma de construcción de prototipos surge de la necesidad de un cliente que está desorientado sobre los detalles de lo que realmente necesita. Se debe comenzar con la recolección de requisitos y definición de objetivos, en este punto el equipo de desarrollo puede crear un “diseño rápido” para el cliente, con esto empezarán a verse detalles del software para el cliente y así empezar la construcción de un prototipo.

El prototipo es puesto a prueba con el cliente y así refinar los requisitos, de esta manera el desarrollador comprenderá mejor que se necesita hacer. En este punto ocurre de la siguiente iteración y se seguirán produciendo prototipos hasta que alguno sirva para el propósito descrito. Este último servirá como piloto para el diseño del sistema real.

En la mayoría de los proyectos, el primer sistema construido apenas se puede utilizar. Puede ser demasiado lento, demasiado grande o torpe en su uso, o las tres a la vez. No hay otra alternativa que comenzar de nuevo, aunque nos duela pero es más inteligente, y construir una versión rediseñada en la que se resuelvan estos problemas... Cuando se utiliza un concepto nuevo de sistema o una tecnología nueva, se tiene que construir un sistema que no sirva y se tenga que tirar, porque incluso la mejor planificación no es omnisciente como para que esté perfecta la primera vez. Por lo tanto la pregunta de la gestión no es si construir un sistema piloto y tirarlo. Tendremos que hacerlo. La Única pregunta es si planificar de antemano construir un desechable, o prometer entregárselo a los clientes (Brooks, 1995).



**Figura 1.3.3. Fases en el ciclo del modelo de desarrollo de prototipos**

Este modelo de construcción es muy cómodo para el equipo de desarrollo y el cliente, ideal para clientes que no tiene experiencia para detallar necesidades de software. Sin embargo la construcción de prototipos tiene dos grandes problemas:

- La cantidad de requisitos puede crecer indefinidamente pues en cada iteración el cliente puede pedir ajustes conforme a lo que en ese momento necesita, esto puede hacer lento el desarrollo de un producto final.
- El equipo de desarrollo suele hacer más compromisos de los que puede cumplir, y el trabajar de prisa le llevará a cometer malas elecciones de las características de desarrollo, las cuales podrían ser inadecuadas o ineficaces, y terminaría aplicándolas al sistema.

### **1.3.4 Proceso unificado de desarrollo de software**

El proceso unificado de desarrollo de software es un modelo publicado en 1999 por Ivar Jacobson, Grady Booch, James Rumbaugh quienes también desarrollan el lenguaje unificado de modelado (UML). Este modelo se puede clasificar como un híbrido, ya que es posible utilizar técnicas genéricas de desarrollo y agrupándolas en varias iteraciones, con la agilidad de poder trabajar todos los procesos propuestos en el mismo modelo paralelamente, produciendo software de calidad y evolutivo con cada iteración.

Pressman (2002) se refiere a este proceso como: “El proceso unificado de desarrollo de software representa un número de modelos de desarrollo basados en componentes que han sido propuestos en la industria. Utilizando el Lenguaje de Modelado Unificado (UML), el proceso unificado define los componentes que se utilizarán para construir el sistema y las interfaces que conectarán los componentes” (p. 28-29).

Para entender el funcionamiento de este modelo iterativo, se deben tomar en cuenta 3 características, las fases, los flujos y las iteraciones.

- Los flujos pueden ser el de cualquier proceso de desarrollo genérico, por ejemplo requerimientos, análisis, diseño, código y prueba.
- Las iteraciones es el paso completo de todos los flujos una vez, lo cual produce un avance resaltable en el proyecto.
- Las fases pertenecen a la teoría del modelo de proceso unificado de desarrollo de software, y se compone de varias iteraciones, el líder de proyecto debe planear los tiempos en las que cada fase debe empezar y cada evento con la que debe terminar.

Para comenzar el proyecto, no se necesita un experto para obtener los requisitos de software, pues se pueden recolectar en varias iteraciones en las primeras fases del desarrollo, pero se debe tener cuidado de documentar correctamente cada uno de estos y planear su desarrollo a su debido tiempo, así como con la aparición de nuevos requisitos, se debe analizar su impacto antes de aceptarlos.

Con el objetivo de crear una buena arquitectura con cada elemento que formara el producto final, UML ofrece muchos diagramas y representaciones gráficas para ayudarse en el proceso de análisis y diseño. Se debe ser consciente que el producto final de cada iteración de desarrollo debe cumplir con todo lo especificado en los casos de uso, es decir, asegurarse de la calidad del software.

### ***1.3.5 Modelo incremental***

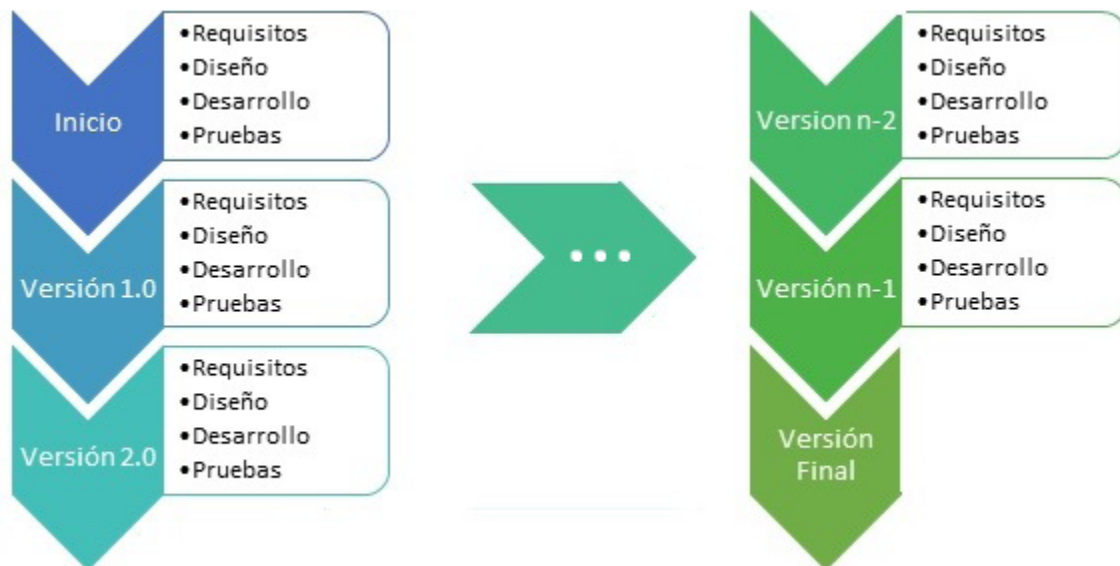
El desarrollo incremental es un modelo que implementa dos técnicas genéricas de desarrollo para producir software que cumpla con los requisitos solicitados por el cliente. La técnica de desarrollo lineal o en cascada es muy útil cuando se trata de producir software elemental, esta técnica combinada con el desarrollo por prototipos hace ciclos del modelo en cascada, ambas técnicas gozan de todas sus ventajas al implementarlas juntas. Esto es, se vuelve un modelo iterativo, donde cada iteración crea un producto final incompleto, pero operable para el cliente, este producto se prueba y se analiza nuevamente con lo cual se crea el incremento de funciones en el software.

El modelo de proceso incremental, como la construcción de prototipos y otros enfoques evolutivos, es iterativo por naturaleza. Pero a diferencia de la construcción de prototipos, el modelo incremental se centra en la entrega de un producto operacional con cada incremento. Este modelo tiene como ventajas:

- No requiere una gran cantidad de personal.
- Se pueden cumplir todos los requisitos acordados.
- El cliente no debe esperar hasta un producto final que tomara largo tiempo desarrollar, sino que podrá utilizar desde el primer sistema las funciones que se hayan acordado primero y empezar a sacarle provecho al software.
- El cliente puede instruirse fácilmente en el funcionamiento de sistemas enormes, dado que lo puede aprender gradualmente.

- La comunicación constante con el cliente lo pone al tanto de las cosas que aún podrían faltarle al sistema y llegar a un acuerdo con el equipo de desarrollo.
- La aceptación del producto final es casi siempre satisfactoria.
- Dado que lo que más importa del sistema es lo que se desarrollará primero, el cliente tomara más tiempo en obtener el primer prototipo, pero los incrementos de este serán por seguro más rápidos.

Este modelo acepta las fases del modelo de desarrollo lineal tal y como son, aunque también pueden ser con metas y tiempos de planeación y desarrollo ligeramente más reducidos, no es indispensable hacer todos los componentes del producto final en la primera fase.



**Figura 1.3.4. Iteraciones y fases del modelo de desarrollo incremental.**

Sin embargo el modelo puede llegar a fallar por razones como la adaptabilidad del sistema a los incrementos, pues en cada desarrollo puede llegar a ser confusa o compleja la adaptación de nuevos módulos de software. También el equipo de desarrollo debe concentrarse en hacer incrementos rápidos, lo cual puede hacer que cada entrega este con muchas funciones incompletas o bien tengan que reescribirse. Y aunque la práctica continua de los requisitos de software es una ayuda, el cliente puede excederse en lo que desea, esto puede llegar a dañar los tiempos y costos en la entrega del producto final.

## **1.4 BASES DE DATOS**

### ***1.4.1 Introducción y definición de Bases de Datos***

Un base de datos es una colección de datos organizados y relacionados entre sí, han sido refinados en múltiples ocasiones y son completamente abstractos, funcionan como una fuente central de datos que servirán a uno o muchos usuarios y sistemas, se dividen en 2 ramas según sus características, estáticas y dinámicas.

*Bases de datos estáticas.* Son de tipo no modificables, esta base suele utilizarse para solo leer, por ejemplo un blog de información diaria, un alimentador de noticias o también un conjunto de datos para estadística posterior.

*Bases de datos dinámicas.* Les es posible modificar sus datos, y son en las que se pueden realizar todas las operaciones básicas: insertar, eliminar, actualizar y consultar. El tipo de base de datos utilizada para fines empresariales o de negocios.

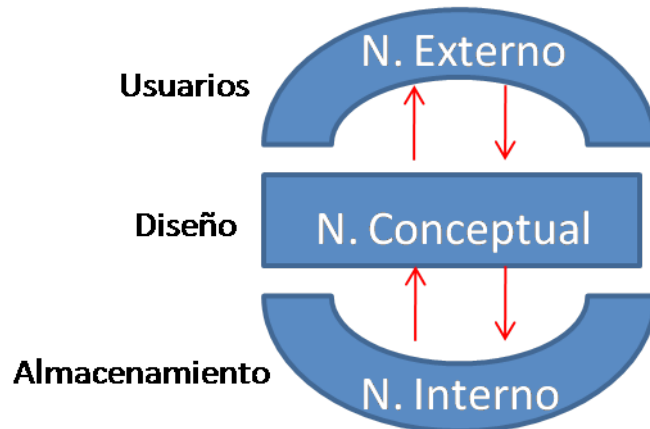
Actualmente las bases de datos usan un motor más complejo que en sus inicios, un sistema gestor de bases de datos (SGBD) que le brinda todas las funcionalidades de manera estandarizada. Estos gestores pueden soportar modelos de datos complejos.

### ***1.4.2 DBMS***

Data Base Management System o “Sistema Gestor de Bases de Datos” es el centro de funcionamiento de una base de datos. Le permiten el almacenamiento, modificación y la extracción de datos, y contiene herramientas para el funcionamiento de sus operaciones principales: añadir, modificar, borrar y consultar. Los gestores son creados para controlar una gran cantidad de información de manera práctica y eficiente, además debe ser fiable, íntegro, seguro y sin anomalías.

La gestión de los datos, según Silberschatz, Korth, Sudarshan (2002) “implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información” (p. 1).

El comité ANSI-SPARC en 1975 propone una arquitectura de tres capas para los SGBD, de forma que separe la actividad física de la base de datos, las aplicaciones que se ejecutan en esta y a los usuarios que la interactuarán. Esta arquitectura (figura 1.4.1) abstrae tres niveles:



**Figura 1.4.1 Arquitectura ANSI-SPARC**

Fuente: Niveles teóricos BD. (2014). Seguridad Base de Datos. Recuperado de <http://segbasedatos.wordpress.com/niveles-teoricos-bd/>

*Nivel físico o interno.* Un esquema de cómo están almacenados los datos en el disco duro. En este esquema se describe su organización, su flujo de entrada y salida, los tamaños y tipos de los registros entre muchas otras características.

*Nivel conceptual.* Se detalla la estructura de la base de datos para el equipo de programación y desarrollo mediante un esquema. Este esquema contiene entidades, atributos, relaciones, métodos y restricciones, sin mostrar físicamente el almacenamiento de cada uno.

*Nivel de aplicación o externo.* Un esquema para el cliente o usuario final, se describen solo partes de la base de datos de mayor interés en una visión individual sin mostrar las relaciones y estructuras de la base de datos conceptual.

Mediante esta arquitectura, los SGBD proporcionan independencia lógica y física de los datos. La independencia lógica brinda la capacidad de modificar el esquema conceptual sin tener que alterar el esquema de usuario o las aplicaciones que usan la base. Y la independencia física permite alterar el esquema físico sin tener que alterar el conceptual o externo. Los SGBD son complejos, contienen muchos paquetes que permiten almacenar y explotar datos. Estos paquetes son: lenguaje, diccionario de datos, seguridad e integridad de datos.

### *Lenguaje del SGBD.*

Cada gestor es manejado con un lenguaje apropiado el cual involucra tipos de usuario, los más comunes son administrador, diseñador del modelo para la base de datos, programador y el usuario final. Estos lenguajes permiten especificar los datos que compondrán la base de datos, la estructura de la misma, las relaciones, las reglas de integridad, seguridad, etc. Estos lenguajes se dividen en dos tipos:

- *Lenguaje de definición de datos.* Abreviado comúnmente por sus siglas en ingles DDL, es el lenguaje utilizado para especificar el esquema conceptual e interno. Se definirán las entidades, relaciones, atributos, restricciones y operaciones.
- *Lenguaje de manipulación de datos.* O bien conocido por sus siglas en ingles como DML, se utiliza para modificar y obtener el contenido de la base de datos. Sus funciones principales son realizar consultas, crear, actualizar y eliminar datos. Este lenguaje es muy adaptable a ser utilizado por lenguajes de alto nivel.

### *El diccionario de datos.*

Este diccionario es donde se deposita toda la información de los datos que se almacenaran en la base de datos. Es decir que contiene las características lógicas del lugar donde se almacenan los datos del sistema, la estructura física y lógica de la base de datos, el espacio en disco que utilizaran los datos, las restricciones de integridad, los privilegios de los usuarios, los valores por defecto de las tablas, todos los objetos del esquema conceptual y las maneras de acceso a los objetos.

También el diccionario debe ser capaz de soportar los tres niveles de abstracción de la arquitectura y estar integrado al gestor. Así puede agilizar la transferencia de información desde el nivel de aplicación al nivel físico.

### *Seguridad e integridad de datos.*

Los sistemas gestores de bases de datos ofrecen mecanismos para garantizar la integridad y seguridad de los datos. Uno de estos es controlar que solo los usuarios autorizados puedan entrar a la base de datos, también es necesario tener en cuenta que eventualmente habrá daños accidentales o intencionales y se debe asegurar la creación de restricciones para que la base de datos no pierda su solidez e integridad, estos procedimientos de prevención programados por un administrador, deben ser soportados por el sistema gestor.

Algunos gestores poseen herramientas para un uso concurrente asegurando la solidez de la base, también disponen de métodos de respaldo y restauración, y muchos de los nuevos gestores han desarrollado módulos de recuperación o “rollback” en caso de que algo se dañe y sea necesario llevar a la base de datos nuevamente a un estado consistente.

### **1.4.3 Usuarios de la base de datos**

Los usuarios de una base de datos pueden administrar, crear y manipular la estructura y contenido de la misma, pero no es necesariamente un solo usuario el que hará esto. Cada gestor es construido de manera distinta y ofrecen distintos tipos de clasificación de usuarios, con privilegios para realizar distintas tareas. Oracle propone tres tipos de usuario:

*Administrador de la base de datos.* Como su nombre lo dice, la función de este usuario es administrar, es decir asegurarse que la base de datos está cumpliendo con los objetivos para el cliente. Entre sus tareas están:

- Seleccionar e instalar el sistema gestor de bases de datos.
- Crear la base de datos de acuerdo al esquema conceptual propuesto y las cuentas de usuario.
- Planeamiento y control de las dimensiones en espacio físico que ocupará el sistema gestor.
- Determinar las formas de uso y acceso a la base, e instruir a los programadores de aplicaciones acerca de ello.
- Asesorar en todo momento al equipo de desarrollo.
- Vigilar minuciosamente el acceso, uso, irregularidades, carga del servidor, etc.
- Efectuar procedimientos de recuperación de la base de datos en caso de ser necesario.
- Realizar auditorías al sistema que hace uso de la base de datos de manera frecuente, así se detectarán fallas estructurales y de seguridad.
- Optimizar siempre el rendimiento de la base de datos cuando sea posible.

*Programador de aplicaciones.* Este usuario solo le es permitido el lenguaje de manipulación de datos, es decir no tiene permisos para alterar la estructura física y conceptual de la base de datos. Su función es crear aplicaciones para el nivel usuario y valerse de muchas herramientas para lograr su objetivo, tales como las herramientas de desarrollo rápido de aplicaciones (DRA) o lenguajes de programación que soporten el uso de módulos para DML. Asesorado por un administrador de base de datos se establecerán los accesos y flujo de datos en estas aplicaciones.

*Usuario final.* Una persona o sistema que accede a visualizar información procesada de la base de datos mediante la aplicación desarrollada por el equipo de programación, el cual no utilizara ningún lenguaje del gestor y sus permisos son limitados. El administrador de base de datos especifica que permisos tendrán los usuarios finales.

## **1.5 MODELOS DE BASES DE DATOS**

### ***1.5.1 Modelos de datos.***

Los modelos de bases de datos surgen a partir de la estandarización ANSI-SPARC y sus tres niveles de abstracción, estos modelos son el grupo de ideas y herramientas que describirán la estructura de una base de datos, o bien el esquema conceptual de la base de datos. Estos modelos son el idioma en el que se comunicarán el usuario y el diseñador. Estos esquemas se clasifican y pueden dividirse en dos modelos lógicos y un modelo físico.

El modelo lógico basado en objetos. Son los que mejor describen los datos a nivel conceptual y usuario, lo más cercano al mundo real. Es posible especificar restricciones de datos para agregar rigidez al modelo, mejorar su semántica, optimizar el rendimiento, etc. Los principales modelos de esta clase son el “modelo entidad-relación” (ER) y el “modelo orientado a objetos” (OO). Hoy en día el más utilizado es el modelo ER.

El modelo lógico basado en registros. Utilizados para mejorar el entendimiento desde el nivel conceptual hacia el nivel físico. Estos modelos se basan más en la forma de almacenamiento que la base de datos tendrá, contemplando los valores fijos y variables que cada registro almacenará en disco. Aunque estos modelos no son capaces de representar el código de la base de datos, existen lenguajes y aplicaciones que pueden realizar consultas y modificaciones a los datos mediante este modelo.

Los representantes más aceptados de este modelo son el “relacional”, “jerárquico” y “de red”, entre los cuales el modelo relacional es más aceptado por su simplicidad y atomicidad.

El modelo físico. Describe como se almacenan los datos en disco y todos los artefactos que se utilizarán en el proceso. Este ayuda a calcular un tamaño estimado de almacenamiento a determinado tiempo. No hay muchos modelos de este tipo y los más conocidos son el modelo “unificador” y “de memoria de elementos”.

### **1.5.2 El modelo Entidad-Relación (E-R)**

Este modelo se clasifica como un modelo de tipo lógico, estos modelos se caracterizan por estar en la línea media entre los niveles de aplicación y físico. Esta línea media permite que los clientes puedan confirmar que sus necesidades son cubiertas y a su vez ya es una forma cercana a la manera de almacenamiento que tendrá en los servidores.

El modelo entidad-relación es propuesto en 1976 por Peter Chen como una representación de los problemas del mundo real como interacciones o relaciones entre entidades. Este modelo podría decirse que es un híbrido entre el modelo relacional y el orientado a objetos, hereda todos sus buenos rasgos y es uno de los más utilizados hoy en día para el diseño y creación de bases de datos. Existen cuatro nociones básicas que deben analizarse para el entendimiento del modelo: entidades, relaciones, atributos y dominio.

*Entidad.* Es un objeto real o abstracto, pero que existe en el mundo real y es distinguible de otros objetos y es nuestro objeto de análisis para abstraer la información que puede poseer tal objeto. Se puede tomar como ejemplo un objeto “Libro” el cual existe en el mundo real y es distinguible de otros medios impresos, este objeto posee muchas características tales como título, autor, editorial, etc., pero este único objeto no es la entidad, es un ejemplar de una generalización de objetos “Libros”. Las entidades suelen nombrarse siempre en plural.

*Relación.* El lazo que asocia o vincula las entidades. Una relación es la interacción que tiene una entidad con otra. Por ejemplo las entidades “libros” y “socios”, las tres son entidades que existen en el mundo real, la relación puede ser “Un libro es prestado a un socio”, donde el verbo “es prestado” funciona como la relación que une a ambos objetos pertenecientes a entidades.

*Atributo.* Las características que posee una entidad. Como una entidad pertenece al mundo real, cada objeto que se conoce en el mundo real tiene muchas características que los hacen únicos. Por ejemplo si de la entidad “Libros” tomamos un ejemplar podremos distinguir que estos poseen un título, autor, editorial, edición, fecha de producción, número de ejemplares, etc., y la entidad “Socios” que se refiere a personas, como se sabe tienen información única como su nombre, apellidos, fecha de nacimiento, sexo, CURP, RFC, etc.

*Dominio.* Cuando se trata de clasificar los distintos valores que puede tomar un atributo, se puede observar que en algunas ocasiones los valores serán limitados y en algunas otras no, a esto se le llama dominio. Hay dos clases de dominios; dominios por intensión y dominios por extensión.

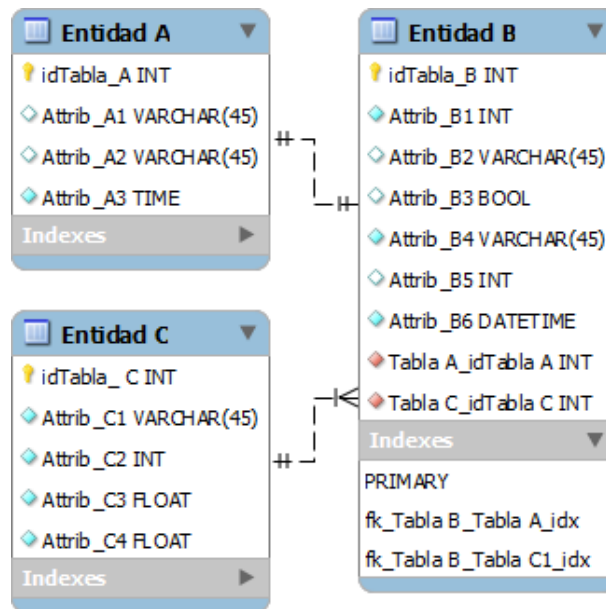


Figura 1.5.1. Representación gráfica del modelo entidad-relación.

- Un dominio por intensión es aquel en el que no se conoce un límite finito para el valor que pueda tomar el atributo, por ejemplo una fecha, nombre de persona, etc., en este caso solo se asigna el tipo de dato que guardará el atributo.
- Un dominio por extensión es lo contrario, cuando se conoce claramente el límite de opciones que puede tomar un atributo, por ejemplo los días de la semana, idiomas, países, etc., para esto no solo basta con especificar la naturaleza del dato del atributo, sino además se debe hacer una restricción a nivel de aplicación de forma que solo pueda almacenar estos valores definidos.

La representación gráfica de este modelo es con tablas y grafos, observese la figura 1.5.1, las tablas representan las entidades y sus atributos van contenidos en estas tablas. Las líneas que unen a las tablas son la representación de la relación y siempre son etiquetadas por los verbos correspondientes entre las entidades.

### **1.5.3 Modelo relacional**

El modelo actualmente es el principal modelo de datos para el desarrollo, debido a su simplicidad se posiciona encima del modelo jerárquico y de red. Propuesto en 1970 por el doctor Edgar Frank Codd como solución a los problemas de redundancia de datos y mejoras a las estructuras de datos que no poseían los modelos de datos de ese entonces. Los avances más importantes que el modelo de datos relacional incorpora respecto a los modelos de datos anteriores son:

- *Sencillez y uniformidad.* Los usuarios ven la base de datos relacional como una colección de tablas, y al ser la tabla la estructura fundamental del modelo, este goza de una gran uniformidad, lo que unido a unos lenguajes no navegacionales y muy orientados al usuario final, da como resultado la sencillez de los sistemas relacionales.
- *Solida fundamentación teórica.* Al estar el modelo definido con rigor matemático, el diseño y la evaluación del mismo puede realizarse por métodos sistemáticos basados en abstracciones.
- *Independencia de la interfaz de usuario.* Los lenguajes relacionales, al manipular conjuntos de registros, proporcionan una gran independencia respecto a la forma en la que los datos están almacenados.

(De Miguel Castaño, Piattini Velthuis, Marcos Martinez, 2000).

La estructura relacional puede representarse como una tabla, llamada relación. Las columnas representan los atributos de un objeto y las filas contienen registros o valores contenidos en el dominio del atributo, estas son llamadas relaciones pues todos los atributos y sus dominios pertenecen al mismo concepto. Estas estructuras relacionales pueden esquematizarse y virtualmente enlazarse logrando fácil obtención de datos. El enlace se logra mediante un campo en común que compartan las relaciones, este campo en común se le denomina campo llave y es capaz de identificar inequívocamente a una tupla de la relación.

### **1.5.4 Llaves primarias y foráneas**

Es importante en una base de datos que cada colección, llámese tabla, entidad o relación dependiendo del modelo de datos que se utilice, sus atributos posean una manera de distinguirse unívocamente, de modo que ninguna tupla de la entidad tenga duplicados.

Las llaves son capaces de identificar a la entidad de otras entidades y ayudan a reconocer y distinguir las relaciones.

La pregunta es: ¿Cómo decir cuando se tiene una llave, o como crear una? Una llave puede ser un atributo o un conjunto de estos en la misma entidad pero se debe cumplir con ciertas características para obtener esa distinción:

- Única. Es decir dentro del dominio de datos que se le dio, nunca debe repetirse para cada registro. Por ejemplo si tenemos una entidad “Ciudadanos” y esta entidad posee entre sus atributos: nombre, apellidos, vivienda, CURP, sexo, fecha de nacimiento, etc., el único atributo que posee la característica de ser unívoco es el CURP, pues ningún ciudadano posee el mismo que otro.
- Mínima. Se debe buscar que esta llave sea con la menor cantidad de atributos posible, pero siempre cumpla la unicidad. Por ejemplo en la entidad “Libros” que contiene los atributos: autor, título, editorial, edición, fecha de publicación, país de publicación, número de páginas, etc., no podemos encontrar un atributo que por sí solo sea único, entonces se debe buscar más de un atributo que en conjunto logre el objetivo, por ejemplo “autor, título, edición” son el mínimo de columnas que pueden identificar unívocamente a un libro de otros que se le parezcan.

Estas llaves convencionalmente son llamadas candidatas, y pueden existir más de una de estas en cada entidad. De estas llaves surgen dos definiciones en su uso, las primarias y las foráneas.

- Una llave primaria denota una llave candidata que ha sido elegida por el diseñador de la base de datos como el elemento principal para identificar a la entidad.
- Una llave foránea es una llave primaria que pertenece a una entidad pero que ha sido citada desde otra entidad. Es decir un campo o campos en común que tiene una entidad con otra segunda entidad y ese es la llave de esa segunda entidad. Las llaves foráneas denotan las relaciones entre entidades.

La buena elección de llaves depende de la experiencia del diseñador y es vital para los procesos de normalización de la base de datos, lo cual impacta en la eficacia y eficiencia de la base de datos.

## **1.6 NORMALIZACIÓN**

### ***1.6.1 Definición de normalización***

La normalización en una base de datos entidad-relación es un proceso en el cual se transforma la visión del diseñador de entidades complejas a estructuras más pequeñas, estables, simples y fáciles de mantener, dicho proceso cuenta con una serie de reglas a seguir para lograr una base de datos normalizada.

Con la normalización se obtienen las ventajas de integridad de datos, eliminación de redundancias, disminución de problemas de actualización de datos y se compacta el peso de la base de datos en el disco.

Antes de empezar con el procedimiento se necesita analizar cada entidad propuesta para identificar las dependencias funcionales de los atributos en cada una de estas. Una dependencia funcional es una conexión entre uno o más atributos, para ejemplificarlo se puede decir que en una entidad “Estudiantes” el atributo “número de estudiante” es capaz por si solo de identificar el nombre y los apellidos de un estudiante determinado, entonces se dice que el nombre y los apellidos son funcionalmente dependientes del número de estudiante.

Los tres tipos de dependencia funcional más importantes a seguir antes de empezar el proceso son: dependencia funcional total, dependencia funcional multivalente y dependencia funcional transitiva. Existen otros dos tipos de dependencia, sin embargo el proceso de normalización las tratará más adelante.

- *Dependencia funcional total.* Cada atributo es funcionalmente dependiente de uno o varios atributos. Es decir que dado un atributo o un conjunto de ellos, es posible obtener unívocamente a otro de los atributos en la entidad, un concepto parecido al de la llave candidata.

- *Dependencia funcional multivalente.* Ocurre cuando dado un atributo o conjunto de ellos, es posible deducir varias veces el mismo atributo en diferentes registros. Por ejemplo dado el nombre y apellidos de una persona, es posible deducir sus teléfonos, sin embargo este último campo es capaz de contener más de un valor para la misma persona.
- *Dependencia funcional transitiva.* Considerando los elementos X, Y y Z que son atributos o grupos de estos y pertenecen a una misma entidad, es posible deducir Y a partir de X, o sea Y es funcionalmente dependiente de X, pero también es posible deducir Z a partir de Y, lo cual quiere decir que a partir de X se puede deducir Z. Este tipo de transitividad hace que Z sea funcionalmente dependiente de dos llaves potenciales.

Esta identificación de dependencias funcionales lleva al diseñador a tener una vista más clara de que atributo o atributos son llaves candidatas y dado ese paso puede empezar a encontrar las llaves primarias. De esta manera se puede empezar con los procedimientos de normalización.

### **1.6.2 Formas normales**

El objetivo de la normalización es que el esquema relacional alcance un mejor nivel de consistencia, lo que significa eliminar las redundancias y los errores al momento de una actualización. La manera de aplicar una normalización efectiva es transformando el esquema en una forma normal. Un esquema está en cierta forma normal si cumple ciertas características, entre mayor sea su nivel de normalización, es decir una forma normal más alta, los problemas serán menores y mucho menos frecuentes. Hoy existen 6 formas normales bien definidas y 3 formas normales derivadas de estas debido a casos muy selectos de anomalías. Una base de datos consistente basta con que le sea transformada hasta la tercera forma normal, la gran mayoría de errores de esquema y redundancias serán eliminadas en ese punto.

#### **1.6.2.1 Primera forma normal (1FN)**

La primera forma normal definida por De Miguel Castaño et al. (2000) dice: “Se dice que una relación está en 1FN cuando cada atributo solo toma un valor del dominio simple subyacente” (p. 157)

Para facilitar el entendimiento de la 1FN recuérdese la dependencia funcional multivalente, el objetivo de esta forma normal es separar esa multivalencia de un atributo, para dejarlo únicamente con un solo valor. Para ejemplificar esta forma normal, véase la tabla 1 que representa la entidad “Empleados”.

**Tabla 1.6.1: Entidad empleados, sus atributos y registros.**

no_empleado	nombre	apellidos	departamento	proyectos
3250014238	Pedro	Morales García	Construcción	Bacheo sección 1390 Pavimentación local Puente carretera
3250057316	Andrés	Vásquez Torres	Administración	Puente carretera Pavimentación local
3261455637	Liliana	Hernández Castillo	Supervisión	Bacheo sección 1390

El atributo proyectos en esta entidad tiene una dependencia funcional directamente del número de empleado, sin embargo aloja más de un valor de su dominio. La primera forma normal establece que estos valores deben separarse, lo cual lograría una entidad como la que se muestra a continuación.

**Tabla 1.6.2: Primera forma normal de la entidad empleados.**

no_empleado	nombre	apellidos	departamento	proyectos
3250014238	Pedro	Morales García	Construcción	Bacheo sección 1390
3250014238	Pedro	Morales García	Construcción	Pavimentación local
3250014238	Pedro	Morales García	Construcción	Puente carretera
3250057316	Andrés	Vásquez Torres	Administración	Puente carretera
3250057316	Andrés	Vásquez Torres	Administración	Pavimentación local
3261455637	Liliana	Hernández Castillo	Supervisión	Bacheo sección 1390

Se ha separado el campo proyectos que contenía múltiples registros para que cada proyecto pertenezca a un registro únicamente y cumpla la primera forma normal.

Se pueden apreciar datos repetidos después de efectuar la transformación, pero aun con estas redundancias, ya se considera que esta entidad está en primera forma normal, pues cada atributo toma un único valor de su dominio.

### **1.6.2.2 Segunda forma normal (2FN)**

De Miguel Castaño et al. (2000) definen la segunda forma normal así:

Se dice que una relación esta en 2FN si:

- Está en 1FN.
- Cada atributo no principal tiene dependencia funcional completa respecto de cada una de las claves (p. 157).

Para aplicar la segunda forma normal, el diseñador debe estar ya al tanto de que atributos o conjunto de ellos son claves candidatas. Por ejemplo supóngase la entidad “Trabajadores” como se muestra en la tabla.

**Tabla 1.6.3: Entidad trabajadores, sus atributos y registros.**

Atributo	Registros		
no_trabajador	45514563	45514012	45514801
nombre	Alberto Pérez	Josué Ramírez	Liliana Vera
puesto	Supervisor	Asesor	Asesor
horario	0800-1600	0800-1400	0800-1400
salario	8000	5600	5600
teléfonos	4456311,2221456610	2224559124,2221331547	2221914547
torre	1	2	1
área	15	40	39

El primer paso es verificar que la entidad se encuentre en primera forma normal, pero el campo teléfonos posee más de un valor en su dominio, entonces debe separarse este atributo para cada registro.

**Tabla 1.6.4: Primera forma normal de la entidad trabajadores.**

Atributo	Registros				
no_trabajador	45514563	45514563	45514012	45514012	45514801
nombre	Alberto Pérez	Alberto Pérez	Josué Ramírez	Josué Ramírez	Liliana Vera
puesto	Supervisor	Supervisor	Asesor	Asesor	Asesor
horario	0800-1600	0800-1600	0800-1400	0800-1400	0800-1400
salario	8000	8000	5600	5600	5600
teléfono	4456311	2221456610	4456311	2221331547	2221914547
torre	1	1	2	2	1
área	15	15	40	40	39

Ahora que la entidad está en primera forma normal, se cumple con el primer requisito que define la segunda forma normal. El siguiente paso es identificar las dependencias funcionales. Obsérvese la siguiente tabla.

**Tabla 5: Analisis de atributos llave en la entidad trabajadores.**

Atributo(s) llave	Atributo(s) no llave.
no_trabajador	nombre, puesto, horario, salario, teléfono
torre	área

Cada fila en esta tabla representa una relación con su atributo llave y sus dependientes, es decir ahora se tienen 2 entidades y estas se encuentran en 2FN.



**Figura 1.6.1. Entidades trabajadores y torres en segunda forma normal.**

### 1.6.2.3 Tercera forma normal (3FN)

La tercera forma normal que está a punto de analizarse, es definida por De Miguel Castaño et al. (2000) como:

Un esquema de relación R esta en tercera forma normal si, y sólo si:

- Está en 2FN.
- No existe ningún atributo no principal que dependa transitivamente de alguna clave de R (p.159).

En la primera y segunda forma normal ya se han tratado las dependencias funcionales multivalente y total, esta se refiere a la última dependencia funcional mencionada que es la de transitividad.

Se debe separar de una entidad aquellos atributos que posean una dependencia transitiva y crear una nueva entidad con ellos. Consideremos el ejemplo anterior con las entidades: trabajadores y torres.

**Tabla 1.6.6: Analisis de dependencias transitivas en la entidad trabajadores**

Atributo	Registros	
puesto	Supervisor	Asesor
salario	8000	5600

Hay una dependencia entre los atributos puesto y salario, es decir de puesto se puede deducir salario y de no\_trabajador se deduce puesto. La tercera forma normal requiere que las entidades se encuentren en 2FN y se cumple con eso, el último paso es separar esa dependencia transitiva en dos entidades.

Trabajadores	Torres	Empleos
no_trabajador	no_torre	puesto
nombre	área	salario
horario		
teléfono		

**Figura 1.6.2. Entidades trabajadores, torres y empleos en tercera forma normal.**

Estas tres entidades cumplen con la tercera forma normal, pero se puede notar que el atributo teléfono de la entidad Trabajadores sigue siendo un problema pues causa múltiples filas de datos repetidos y es muy posible que el atributo original teléfonos sea una entidad y no un atributo. Las siguientes formas normales resuelven este y otros problemas que el ejemplo presente.

Existe un aspecto de la normalización que debe tenerse muy presente, esta es una serie de reglas y técnicas de cómo llevar un diseño de base de datos a su forma más eficaz, sin embargo no es capaz de interpretar la naturaleza y semántica del problema, es decir, no puede determinar que entidades deben existir y las relaciones entre ellas.

## **1.7 ARQUITECTURA CLIENTE – SERVIDOR**

### ***1.7.1 Código abierto (Open source)***

En pocas palabras, los programas de software libre son programas cuyas licencias les dan a los usuarios la libertad de ejecutar el programa para cualquier propósito, estudiar y modificar el programa, y redistribuir copias ya sea del original o el modificado (sin tener que pagar regalías a los desarrolladores previos).

(David A. Wheeler, 2014)

Esta definición de David Wheeler es la más cercana a lo que es el verdadero significado del concepto software libre y como funciona en el mundo real. Este software promueve el ideal del crecimiento de la sabiduría de manera que en conjunto haya mucha más gente preparada para cualquier reto.

La historia del software libre se remonta a los primeros años en los que la computadora se convertía en ciencia e industria, el compartimiento de programas y algoritmos era algo común entre adeptos de la computación, sin embargo con el paso de los años y las revolucionarias ideas de William H. Gates este movimiento se vio opacado creando así el gran mercado de licencias de software y programas de código cerrado. En vista de este proceso la Fundación de Software Libre (Free Software Fundación) encabezada por el investigador Richard Stallman y el proyecto GNU crean el concepto “copyleft” con el objetivo de otorgar el permiso de libre uso, copia, modificación y redistribución con la condición que el usuario que modifique y redistribuya su trabajo lo siga haciendo bajo la misma licencia copyleft.

Con la iniciativa de la fundación de software libre, surge también la iniciativa del código abierto (OSI) la cual se basa en crear software poderoso y confiable de uso personal y también con intenciones de negocios. OSI define los derechos del software de código abierto:

- No se pagan regalías o alguna otra tarifa impuesta al momento de la redistribución.
- Disponibilidad del código fuente.
- Se tiene derecho a crear modificaciones y trabajos derivados.
- Puede requerir versiones modificadas para ser distribuidos como la versión original más actualizaciones.
- No hay discriminación de grupos o personas.
- No hay discriminación en función de su uso.
- Todos los derechos fluyen hacia las versiones redistribuidas.
- La licencia aplica al programa completo y a cada uno de sus componentes.
- La licencia no debe restringir otro software, permitiendo así la distribución de software de código abierto y de código cerrado juntos.

(Van Reijswoud, de Jager, 2008).

Un software que le es posible redistribuirse y reestudiarse, para ser modificado tiende a evolucionar, a mejorar y posteriormente se convertirá en distintas soluciones mas alla de su propósito original.

### **1.7.2 Apache**

Basado en el servidor web número uno en 1995 del Centro Nacional para Aplicaciones de Supercomputo y como respuesta al abandono de ese proyecto nace el servidor Apache. La primera versión se distribuye al público en abril de 1995 con la primera versión libreada oficial el 1 de diciembre de ese mismo año.

El grupo Apache que se encarga de gestionar los avances y actualizaciones del software motivan a otros programadores para participar en el desarrollo de nuevos módulos componentes, detección de fallos y soluciones. El grupo evalúa el trabajo y decide si utilizar el código o no en la siguiente versión. Apache es un software sin costo y pertenece al proyecto OSI (Open Source Initiative) lo cual concede al usuario todos los derechos expresados con anterioridad. Apache es un servidor web basado en Unix, y hoy en día es capaz de funcionar en muchas sistemas operativos operados mediante GUI, como Windows y Mac, además de funcionar en sistemas tipo servidor tales como OS/2.

Apache presenta muchas otras características, entre ellas un elaborado Índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online (Mohammed J. Kabir, 2003).

Se cuenta con diversos módulos que componen básicamente al servidor, tales como:

- SSL: Comunicación vía "Secure Shell".
- Rewrite: "Basado en reglas (basado en un analizador de expresiones regulares) para reescribir URLs solicitados sobre la marcha. Soporta un número ilimitado de reglas y un número ilimitado de condiciones de reglas adjuntas para cada regla, para proporcionar un mecanismo de manipulación URL muy flexible y potente" (The apache software fundation, 2014).
- WebDav: Protocolo de distribución, autorización y control de versiones.
- Deflate: compresión de ficheros por parte del servidor a modo que lleguen de forma segura al cliente.

- Proxys y Apache Java services: Soporte para el uso de proxy y servicios de Java como los servlets y Páginas dinámicas HTML y XML manejada por un servidor Java.
- CFML: “es un conjunto impulsado por la comunidad de programas que automáticamente configura clientes Tomcat para que coincida con los clientes creados en Apache o IIS. Mod\_cfml elimina la necesidad de configurar los sitios web dos veces - una vez en su servidor web y de nuevo en Tomcat - y lleva a cabo esta tarea de forma automática para usted” (ModCFML, 2014).

Además de que incluye componentes para manejar las nuevas tecnologías como limitadores de ancho de banda, PHP, Perl, Python, Ruby, REXX entre otros.

### **1.7.3 PHP**

Creado en 1994 por Rasmus Lerdof, y nombrado originalmente como “Personal Home Page Tools” (PHP Tools), hoy es un lenguaje de programación para páginas web basado en el lenguaje de programación C.

“PHP (acrónimo recursivo de “PHP: Hypertext Preprocessor”) es un lenguaje ampliamente usado de código abierto y para propósito general que esta especialmente acondicionado para el desarrollo de páginas web y puede ser embebido en páginas HTML” (The PHP group, 2014)

PHP es un lenguaje de programación y no debe ser confundido con HTML el cual es un lenguaje de interpretación. Este lenguaje se ejecuta del lado del servidor que a diferencia de JavaScript, este no puede ser alterado por el cliente. Es posible crear una multitud de ideas en PHP como contenidos dinámicos y paso de información por medio de formularios, pero generalizando los scripts de php son usados para:

*Scripts del lado del servidor.* Que es el uso más común. Con esto se pueden programar las acciones que tomará una página web al momento de cargarse y posteriormente mediante otras técnicas como AJAX. Para que esta característica funcione es necesario contar con un analizador de php, servidor web y navegador web. El servidor se encarga de ejecutar las instrucciones mediante el analizador y reinterpretarlo en HTML, el resultado se puede observar en el navegador.

*Línea de comandos.* Esta es la opción para cuando no se cuenta con un navegador y tampoco un servidor web, pero se cuenta con un analizador de PHP como el programador de tareas en Windows, en el cual se puede ejecutar un script.

*Aplicaciones gráficas para cliente.* El componente PHP-GTK otorga al lenguaje utilizar secciones de la librería de gráficos GTK+ para escribir scripts que generen aplicaciones gráficas.

Esta tecnología es capaz de funcionar en la mayoría de los sistemas operativos más comunes de hoy en día y en otros más, así como en distintos servidores lo cual incluye a Apache y a Microsoft Internet Information Server (Microsoft IIS) entre otros. También soporta la mayoría de los gestores de bases de datos de la actualidad, mencionando a Oracle, MySQL, PostgreSQL, ODBC, Unix dbm y muchos más.

Otra característica de PHP es que no se limita solo a la programación de páginas web, sino que además es posible programar ficheros XML, crear documentos PDF o películas flash. Los protocolos que soporta son múltiples empezando por el más usado HTTP, y otros como POP3, LDAP, IMAP y más.

#### **1.7.4 MySQL**

MySQL es un sistema gestor de bases de datos relacionales (SGBD) que permite la múltiple ejecución de usuarios e hilos. Creado en 1995 por la compañía MySQL AB y después parte de Sun Microsystems posteriormente adquirido por Oracle, se puede adquirir bajo la licencia GNU GPL. Este gestor puede funcionar en más de 20 plataformas distintas incluyendo a los sistemas operativos más populares como Mac OS X, Windows (todas sus versiones), Windows Server (2000 a 2012) y Linux. Adicionalmente posee: kernel que permite la ejecución de multihilos, sistema de contraseñas y privilegios, tablas con hasta 50 millones de registros, 64 índices por tabla de hasta 16 columnas cada uno y con peso máximo de 1000 Bytes.

Es posible el soporte de bases de datos relacionales y posee la característica del manejo de llaves primarias y foráneas, también puede realizar operaciones tipo transacción ACID y tiene un gran conjunto de instrucciones SQL, etc. Pero lo que hace único a MySQL es:

- *Múltiples motores de almacenamiento para distintas tablas.* Los motores MyIsam, InnoDB, BDB, CSV son ejecutados de forma nativa.
- *Múltiples transacciones.* Haciendo un grupo de transacciones es posible ejecutarlas todas secuencialmente incrementando la cantidad que se pueden hacer por segundo.

### **1.7.5 AJAX**

AJAX es diferente de ser un lenguaje de programación, en realidad es una serie de técnicas que usan como base el lenguaje de programación JavaScript. El 18 de febrero de 2005 Jesse James Garret define AJAX como lo siguiente:

Ajax no es una tecnología. En realidad son muchas tecnologías, con cada una demostrando sus propias características, y en conjunto para mostrar nuevas y poderosas acciones. Ajax incorpora:

- Presentaciones basadas en estándares usando XHTML y CSS.
- Despliegues e interacciones dinámicas usando los Modelos de objetos del documento.
- Intercambio y manipulación de datos usando XML y XSLT.
- Recuperación asíncrona de datos usando XMLHttpRequest.
- Y JavaScript para enlazar a todos juntos.

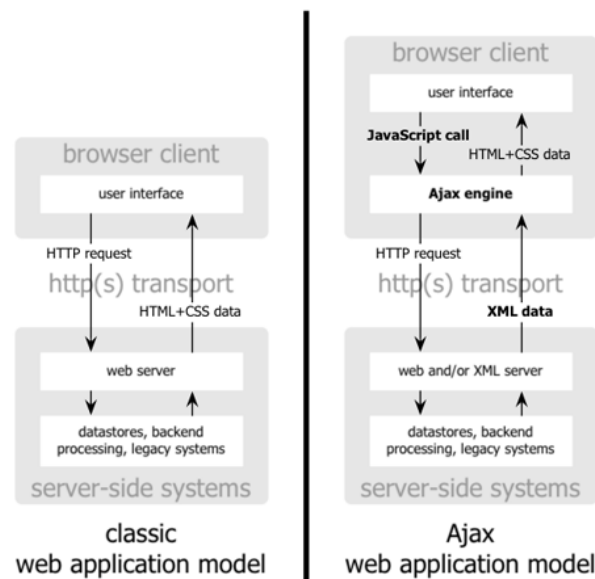
(Garret J.J., 2005)

Revoluciona la forma en la que se construyen sitios web de manera clásica, es decir solicitar una respuesta HTTP a un servidor web, el servidor procesa la información, recupera datos, crea sesiones, etc., y al terminar traduce un documento HTML para el cliente. Ajax tiene la característica de usar la técnica de programación estándar una y otra vez en el momento que se requiera sin tener que hacer un refresco de página en el navegador.

Ajax funciona desde el cliente (client-based) escrita en JavaScript y es capaz de procesar y recuperar información del servidor mediante una solicitud que no es HTTP, más bien puede ser una solicitud XMLHttpRequest.

En el lado del servidor puede utilizarse cualquier lenguaje de programación que soporte el servidor web, la mejor compatibilidad de respuesta la tiene PHP, esta respuesta la informa el servidor y puede ser recuperada en los formatos JSON o XML. También puede emplear esta información del servidor en cualquier DOM, CSS o XHTML.

La tecnología fue tan bien aceptada en la comunidad que hoy existen múltiples Frameworks que facilitan la ejecución de técnicas Ajax. Los más usados son jQuery, Moontools, Dojo toolkit, ExtJS, qooxdoo y Prototype.



**Figura 1.7.1. Las diferencias entre el modelo de programación web clásica y con las técnicas Ajax.**

Fuente: Garret, J., J. (2005). Ajax: A New Approach to Web Applications. Recuperado de <http://www.adaptive-path.com/publications/essays/archives/000385print.php>

### 1.7.6 RIA

Siglas para el término Rich Internet Applications, aplicaciones que tratan de imitar un escritorio de un sistema operativo y su GUI, comúnmente corren en un navegador personalizado o en una máquina virtual, las plataformas más comunes son Adobe Flash, Microsoft Silverlight y JavaFX. Un ejemplo sencillo de una RIA son los juegos basados en los navegadores que muchos sitios y redes sociales ofrecen.

Estas aplicaciones no son nuevas, pueden rastrearse en el pasado, como por ejemplo Remote Scripting de Microsoft en 1999, Macromedia Shockwave en 2002 y los Applets de Java.

Sin embargo con la entrada de las tecnologías Ajax y las mejoras de los navegadores de internet tales como HTML5, JavaScript, GWT, ExtJS o Vaadin han disparado las capacidades de desarrollo y aplicación de las RIA.

La característica de estas aplicaciones es que se ejecutan en un ambiente limitado llamado "sandbox" lo cual le da seguridad a cambio de estar limitado al cliente que puede ser cualquiera de los sistemas operativos Windows, MAC OS o Linux. Su operación consiste en envío y recepción de datos de manera síncrona o asíncrona haciendo la aplicación muy dinámica, pero a su vez es posible que cause mucha carga en el tráfico del servidor. Además pueden ser actualizadas automáticamente y la instalación que se requiere es la de la máquina virtual en la que opera, que está disponible por defecto en la mayoría de los navegadores web, o se consigue fácilmente.

**CAPITULO 2**

**ANALISIS Y DISEÑO  
(ORIENTADO A OBJETOS)**

## **2.1 INTRODUCCIÓN**

Se requiere un sistema de apoyo en línea para administradores y personal que gestiona en los laboratorios de cómputo de la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla; con el propósito de controlar y lograr una buena administración de los recursos y servicios que estos ofrecen.

Se realizará una investigación que comprenderá una serie de entrevistas con los miembros que administran el funcionamiento de los laboratorios, con el propósito de conocer la situación actual y las necesidades que se tienen. De esta manera será posible explicar los requisitos del software a desarrollarse.

## **2.2 PLANTEAMIENTO DEL PROBLEMA**

Actualmente, no se cuenta con un método estandarizado ni automatizado del control de acceso, administración de servicios y gestión de problemas de los laboratorios de cómputo de la Facultad de Ciencias de la Computación.

Hoy los laboratorios operan bajo un reglamento con ciertas normativas para el personal responsable de estos, así como para los profesores y alumnos que hacen uso de sus instalaciones. Cada uno cuenta con más de 30 computadoras para uso del alumnado; y se habilita un proyector de video como apoyo para el profesor. En la zona de administración, los colaboradores cuentan con un mínimo de 4 PCs, disponibles para el control del módulo que les corresponde.

Se propone la creación de un sistema en línea que pueda homogeneizar el modo de administración de estos, y a su vez, dando como resultado el mejor control sobre ellos.

## **2.3 OBJETIVO GENERAL**

El sistema de administración de los laboratorios de cómputo será un sistema en una plataforma web que proporcione plantillas para el registro de recursos, el personal responsable de estas áreas y los problemas que acontecen; así como la elaboración de un control de acceso de usuarios, tanto alumnos como profesores. Por último, la realización de un generador de reportes para dichos aspectos.

Específicamente se desea:

- Administrar el flujo de personas que acceden a las instalaciones.
- Registrar los acontecimientos que ocurran dentro de los laboratorios.
- Inventariar los recursos que disponen.
- Generar reportes de cada acción registrada dentro de los laboratorios.

## **2.4 REQUERIMIENTOS DEL PROBLEMA**

### ***2.4.1 Requerimientos funcionales***

Después de las entrevistas con el cliente se han identificado 3 actores para el sistema, cada uno tiene labores similares en el sistema. También se han logrado identificar 29 requisitos funcionales, los cuales se listan a continuación.

- Autenticar personal.
- Inventariar software.
- Registrar instalación de software.
- Actualizar software.
- Inventariar hardware.
- Agregar equipo.
- Montar equipo.
- Visualizar información detallada.
- Actualizar hardware.
- Administrar acceso.
- Consultar incidentes.
- Registrar incidente.

- Generar reportes.
- Reportar hardware.
- Reportar software.
- Reportar accesos.
- Reportar incidentes.
- Administrar personal.
- Registrar usuario.
- Eliminar usuario.
- Modificar usuario.
- Visualizar información de usuario.
- Administrar sistema.
- Restaurar registros.
- Eliminar registros.
- Respalidar registros.
- Administrar edificio.
- Administrar información.
- Cerrar sesión.
- Instalar sistema.

Posteriormente se detallará que interacción tienen los tres actores con estos requisitos.

#### ***2.4.2 Requerimientos no funcionales***

El sistema contará con características que lo harán eficiente y contar con él en cualquier momento.

- *Rendimiento.* Interfaces ligeras y fáciles de cargar en memoria.
- *Disponibilidad.* Siendo un sistema web, este estará en línea garantizada por el contrato de servicios web con Hostinger México.
- *Seguridad.* El sistema solo funciona para el personal autorizado, por eso se han incluido componentes de encriptación en el cliente e identificación segura por parte del servidor, de manera que solo el personal registrado y con contraseña pueda acceder.
- *Usabilidad.* Se ha realizado una investigación a aspectos de memoria e interacción humana, dando como resultado que el sistema no fuerce la memoria a corto plazo del usuario y de esa manera pueda hacer uso del sistema siempre al 100%.

- *Portabilidad.* El sistema es utilizable en cualquier navegador web en cualquier dispositivo, por ejemplo PC, Smartphone, Mac, iPhone, etc.
- *Escalabilidad.* El sistema es capaz de adaptarse a futuros requerimientos y mejoras.
- *Concurrencia.* El sistema soporta múltiples usuarios conectados a la vez e interactuando continuamente.
- *Mantenición.* El sistema cuenta con componentes sencillos para efectuar mantenimientos a través de la interfaz.
- *Interfaz.* Se realizó un análisis de diseño y se creó una plantilla amigable y funcional para los usuarios.

## 2.5 ALCANCES

El sistema web de administración de servicios informáticos de los laboratorios de cómputo de la FCC, ayudará al:

*Colaborador.* Ahora puede mantener actualizada la información de los recursos físicos con los que cuenta el laboratorio, también será capaz de administrar el acceso y salida de los estudiantes que deseen hacer uso de las instalaciones, además le será posible hacer llegar reportes de incidentes dentro del laboratorio de forma rápida y segura al coordinador del mismo, y por ultimo podrá obtener estadísticas de uso del laboratorio.

*Coordinador.* El trabajo del coordinador se verá apoyado con un catálogo digital de todos los recursos físicos del laboratorio al que está a cargo y su condición de funcionamiento, también le será posible monitorear el trabajo de cada uno de los colaboradores, además podrá crear reportes con las condiciones del laboratorio y los incidentes que ocurran en él, por ultimo tiene la responsabilidad de dar de alta y de baja el registro de los colaboradores en el sistema.

*Administrador.* Usuario de tipo único, solo se admite un usuario administrador en el sistema y es el responsable de mantener el sistema funcionando. El administrador es el que instalará el sistema en el servidor web desplegando todos sus componentes, es capaz de crear los almacenes de información necesaria para los laboratorios de computo, también es responsable de hacer el registro y baja de los coordinadores en el sistema así como asignarlos a las instalaciones correspondientes.

Además le es posible realizar tareas de mantenimiento a la base de datos mediante la interfaz, y por ultimo este usuario no tiene acceso a la información que se registre en los módulos, es decir no podrá visualizar los datos de los recursos físicos que dispone cada uno.

## **2.6 METODOLOGÍA**

El proceso de la creación del sistema se basara en la metodología de desarrollo de software “Proceso Unificado” el cual es de dominio público. También tiene la conveniencia de soportar técnicas orientadas a objetos usando UML como notación común.

### *Análisis*

Se definirán los objetivos cruciales del sistema para los usuarios, las posibilidades de desarrollo, las restricciones de seguridad y la visión del proyecto final. Se modelaran los diagramas de casos de uso y dependiendo de estos también se realizaran los diagramas de realización de casos de uso con lo que se explicará con detalles de alto nivel el funcionamiento. Estos diagramas se modificarán como el proyecto lo requiera en cada iteración.

### *Diseño*

Se plantearan las vistas del sistema a nivel de componentes, su interacción y continuo funcionamiento, se empezarán a laborar los diagramas de secuencia y de colaboración con el objetivo de poseer una vista de la interacción de los componentes y que restricciones deben ser impuestas. También se creara el diseño gráfico del sistema considerando factores humanos y computacionales, para asegurar que el sistema pueda ser entendido y utilizado a su máximo.

### *Implementación*

En esta etapa se codificará lo propuesto en el modelo de diseño respecto a la iteración correspondiente, y debe considerarse que en el módulo componente habrá agregados y modificaciones posteriores, por lo que cada uno deberá ser documentado correctamente y de manera enriquecida.

## *Pruebas y transición*

El plan decidido para la actual iteración será probada exhaustivamente y se procederán a corregir los errores lógicos y de seguridad que implique. Una vez corregidos todos se procederá con el próximo plan, o bien la próxima iteración.

## **2.7 ANÁLISIS DEL SISTEMA**

### ***2.7.1 Glosario de términos***

**Módulo.** Palabra utilizada para referenciar un laboratorio de cómputo de la Facultad de Ciencias de la Computación de la BUAP.

**Instalaciones.** Referencia a todo el equipo físico que posee el laboratorio.

**Software.** Término general que se aplica a un componente no físico de la computadora.

**Programa.** Término para referenciar un único elemento de software.

**Hardware.** Conjunto de objetos físicos que constituyen una computadora o bien operan en base a una.

**Pieza.** Se hace enfoque a una sola unidad de hardware.

**Personal.** Término utilizado para identificar a las personas que operan el laboratorio y mantienen su buen funcionamiento, p. ej. coordinador, colaborador.

**Usuarios.** Se utilizará para nombrar al personal que operará el sistema.

**Alumnado.** Los miembros de la universidad que poseen una matrícula válida y vigente.

### ***2.7.2 Diagrama de casos de uso***

Un caso de uso es una funcionalidad del sistema, que puede ser una interacción entre el usuario y el sistema, comunicaciones entre los componentes que realizan un cambio visible al sistema o bien una interacción de un sistema con otro, el conjunto de todas las funcionalidades que el sistema tendrá es visualizado en el diagrama de casos de uso.

El diagrama de casos de uso para este proyecto se ha dividido en 3 partes, debido a la extensión. Cada diagrama corresponde a los casos de uso que podrá utilizar cada miembro del personal, y aunque algunos parezcan repetidos, estos varían dependiendo el privilegio del personal. La intención de dividirlo en 3 es con el objetivo de entender la función de cada actor en el sistema. Los actores son: colaborador, coordinador y administrador.

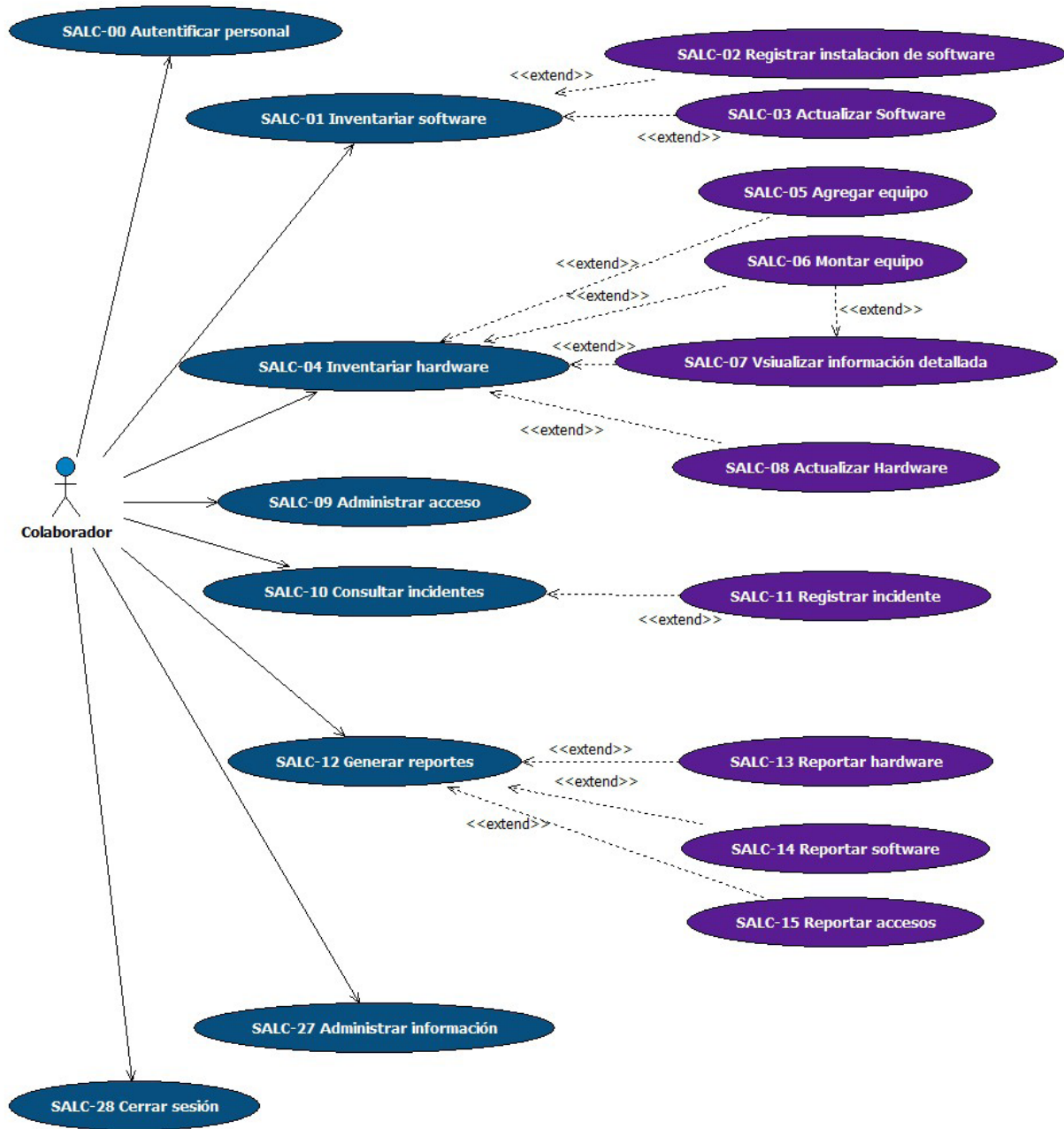


Figura 2.7.1. Diagrama de casos de uso del actor colaborador.

El colaborador posee un total de 18 casos de uso, este actor tiene un nivel de usuario final, será capaz de introducir mucha variedad de datos y le será posible modificar una corta cantidad de ellos, mas no le es permitido borrar registros o incluir datos en partes definitivas del sistema. Este usuario es personal del laboratorio, tiene un tiempo limitado de operación en el sistema dependiendo lo que su coordinador decida. El diagrama de casos de uso dedicado a este actor puede visualizarse en la figura 2.7.1.



Figura 2.7.2. Diagrama de casos de uso del actor coordinador.

El actor coordinador posee 17 casos de uso con un nivel de usuario final con privilegios mas elevados, este usuario puede insertar distintos tipos de datos al sistema, es capaz de hacer modificaciones en todos los datos que inserte y posee capacidad de ingresar registros en algunos puntos clave del sistema, sin embargo tampoco le es posible eliminar información del sistema asegurando un historial de datos firme. Este usuario es personal del laboratorio mas no labora de manera interna en el, es un profesor de la facultad de ciencias de la computación, y aunque su tiempo de operación sea limitado, es de mayor extensión que la del colaborador. La figura 2.7.2 ilustra los casos de uso que se requisitaron.

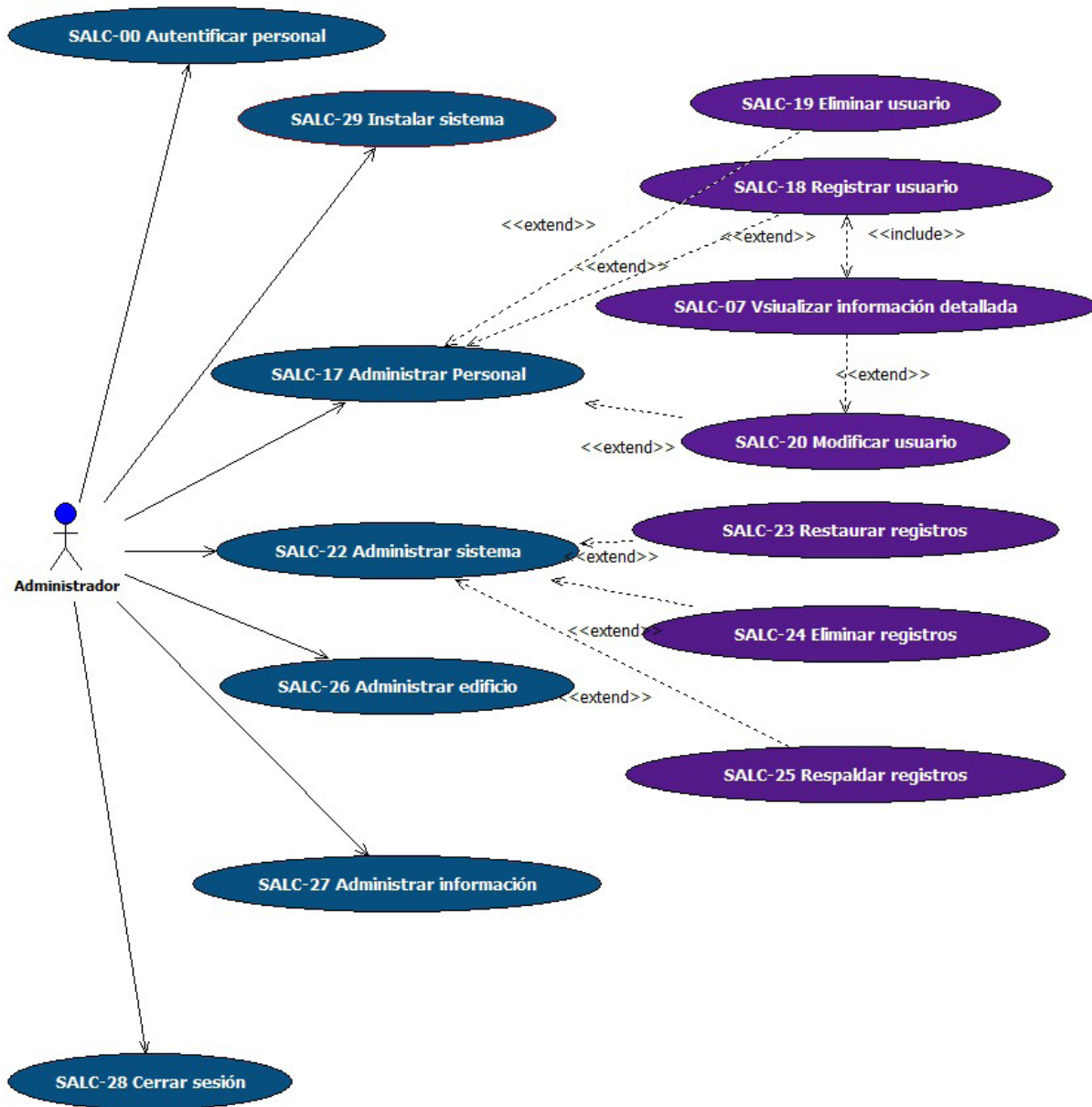


Figura 2.7.3. Diagrama de casos de uso del actor administrador

El actor administrador posee 14 casos de uso con nivel de usuario de programador de aplicaciones, le es posible realizar inserciones de datos, modificaciones y eliminación de información, es decir posee permisos totales de manipulación de datos, mas no le es permitido alterar la estructura de la base de datos del sistema, le es posible crear respaldos y restauraciones del sistema en cualquier momento y es el encargado de instalar el sistema y de registrar a cualquier tipo de usuario, pero preferentemente a los coordinadores. El administrador del sistema no precisamente debe laborar en los laboratorios de computo, puede ser una persona externa designada por la academia, sin embargo su sesión en el sistema es definitivo y no tiene tiempo de expiración.

Debido a que el administrador del sistema no necesariamente es alguien involucrado con los laboratorios de computo no tendrá acceso a visualizar la información del laboratorio que sea ingresada en el sistema. Las funciones de este actor las expresa la figura 2.7.3.

### ***2.7.3 Especificación de casos de uso***

Un caso de uso como primera herramienta para entender una funcionalidad en el sistema en ciertas ocasiones puede ser interpretado por el cliente y por el equipo de desarrollo de manera distinta. La especificación de un caso de uso es un artefacto propuesto para que ambas partes les sea posible acordar que es lo que esta parte del sistema hará. Estas especificaciones no poseen una regla estándar, pero si se proponen diferentes formas y características en las cuales este puede cumplir el objetivo, las características básicas que una especificación debe cumplir son:

- Participantes e interacción. Se aclara que actor o actores estan involucrados en este caso de uso.
- Eventos. Describir el flujo de eventos durante la interacción del actor con el sistema, la secuencia debe estar bien detallada y especificar si el caso de uso puede desencadenar un flujo alterno o bien otro caso de uso.
- Eventos alternos y de excepción. También es conveniente describir que otras tareas puede realizar este caso de uso de manera detallada y como controlar los posibles errores contemplados de modo que el sistema sea estable.

La especificación de casos de uso ha sido organizada en forma de tabla con el siguiente formato:

**Tabla 7. Formato de especificación de casos de uso.**

<b>ID – CU:</b>	El identificador debe ser “SALC-” numero de caso de uso en dos digitos.		
<b>Nombre de CU:</b>	El nombre del caso de uso.		
<b>Creado por:</b>	Nombre del recolector de los requisitos que concluyeron con este caso de uso.		
<b>Fecha de creación:</b>	Se intuye	Fecha de modificación:	Historial de fechas de modificaciones.
<b>(6,79,127) Relevancia primaria.</b>			
<b>(90,28,147) relevancia secundaria.</b>			
<b>Actor(es):</b>	Los actores que involucra este caso de uso.		
<b>Descripción:</b>	Describe en lenguaje español la funcion del caso de uso.		
<b>Precondiciones:</b>	Condicion del sistema para poder realizar este caso de uso.		
<b>Disparador:</b>	Ultimo eslabon de la ruta para poder acceder a este caso de uso.		
<b>Flujo normal:</b>	Secuencia de pasos que el sistema y el usuario realizaran en esta funcionalidad.		
<b>Flujo alternativo:</b>	Secuencia de opciones alternas que el mismo caso de uso contempla.		
<b>Excepciones:</b>	Posibles errores contemplados que pueden ser manejados por la interfaz.		
<b>Notas:</b>	Modificaciones, relevancia y requisitos no funcionales incluidos.		

Todos los casos de uso propuestos tienen una función crucial en el sistema, pero solo se presentará un conjunto de ellos que toman mas importancia entre todos. El criterio para determinar esta importancia es medida en los casos de uso que mas frecuentemente se utilizarán por los usuarios.

<b>ID – CU:</b>	SALC-01			
<b>Nombre de CU:</b>	Inventariar software			
<b>Creado por:</b>	José Abraham Baez Bagatella			
<b>Fecha de creación:</b>	12 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013	
<b>Actor(es):</b>	Colaborador Coordinador			
<b>Descripción:</b>	A través de este caso de uso el usuario podrá visualizar y administrar todo el software disponible en el módulo.			
<b>Precondiciones:</b>	El usuario debió haber completado exitosamente el caso de uso SALC-00 “Autenticar personal”.			
<b>Disparador(es):</b>	El usuario entra a la sección “MI LABORATORIO” en pantalla, luego elegir la opción “INVENTARIO DE SOFTWARE”. Haber concluido exitosamente (2) del flujo alterno A-3 del caso de uso SALC-01 “Inventariar software”.			
<b>Flujo normal:</b>	1	El sistema reconoce el privilegio del usuario que acceso a la sección (E-6). Colaborador, se ejecuta A-1. Coordinador, se ejecuta A-2.		
<b>Flujo alternativo:</b>	A-1	(1)	El sistema muestra al colaborador instrucciones de manejo de la sección, una lista del software registrado previamente con casillas de verificación a la izquierda de cada miembro. También las opciones “Registrar nuevo” y “Registrar Instalación” y el botón “Excluir o cambiar”. REGISTRAR NUEVO, se ejecuta A-3. REGISTRAR INSTALACION, se ejecuta el caso de uso SALC-02 “Registrar instalación de software”. EXCLUIR O CAMBIAR, se ejecuta A-4.	
		(2)	Termina el caso de uso.	
	A-2		El sistema muestra al coordinador, una lista del software registrado previamente.	
		(2)	Termina el caso de uso.	
	A-3	(1)	El sistema muestra un formulario con 4 campos: “Fabricante”, “Nombre”, “Versión”, “Serie”.	
		(2)	El usuario proporciona los datos en los campos mostrados.	
		(3)	El sistema recolecta los datos proporcionados por el usuario y los valida (E-1) (E-2) (E-3). Luego realiza una consulta a la base de datos creando un nuevo registro con los datos proporcionados.	
		(4)	Termina el caso de uso.	
	A-4	(1)	El sistema revisa que casillas de verificación han sido seleccionadas (E-4) (E-5).	
		(2)	El sistema ejecuta el caso de uso SALC-03 “Actualizar Software”.	
		(3)	Termina el caso de uso.	
	<b>Excepciones:</b>	E-1	No se ha proporcionado un nombre de software. a) El sistema informa que ningún nombre de software se ha introducido y le invita al usuario a proporcionar uno. b) El caso de uso continúa.	

	E-2	No se ha proporcionado la versión. a) El sistema informa que ningún número de versión ha sido introducido y le invita al usuario a proporcionar uno. b) El caso de uso continúa.
	E-3	No se ha proporcionado el Fabricante. a) El sistema informa que ningún fabricante ha sido introducido y le invita al usuario a proporcionar uno. b) El caso de uso continúa.
	E-4	No se ha seleccionado ninguna casilla. a) El sistema informa que no hay casillas de verificación activadas y le invita al usuario a seleccionar una. b) El caso de uso continúa.
	E-5	Se han seleccionado más de una casilla. a) El sistema informa que hay más de una casilla de verificación activada y le invita al usuario a seleccionar solo una. b) El caso de uso continúa.
<b>Notas:</b>		El inventario de software es parte esencial del sistema y cuenta con restrictivas desde el cliente para que no se altere su funcionamiento así como control de seguridad de envío de datos.

<b>ID – CU:</b>	SALC-04		
<b>Nombre de CU:</b>	Inventariar hardware		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	13 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Actor(es):</b>	Colaborador Coordinador		
<b>Descripción:</b>	Este caso de uso proporcionará al colaborador la información del hardware disponible en el módulo.		
<b>Precondiciones:</b>	Haber completado el caso de uso SALC-00 “Autenticar personal”.		
<b>Disparador:</b>	El usuario entra a la sección “MI LABORATORIO” en pantalla, luego elegir la opción “INVENTARIO DE HARDWARE”. Haber completado el paso 2 del flujo alterno A-2 del caso de uso SALC-04 Inventariar hardware. Haber completado el caso de uso SALC-08 Actualizar hardware.		
<b>Flujo normal:</b>	1	El sistema muestra una lista con el hardware registrado previamente con casillas de verificación a la izquierda de cada elemento, el botón “Más información”, las opciones REGISTRAR NUEVO, AGREGAR EQUIPO, VER ARMADO DE EQUIPOS.	
	2	El colaborador puede seleccionar alguna casilla de verificación y/o seleccionar una opción. REGISTRAR NUEVO se ejecuta A-2. VER ARMADO DE EQUIPOS se ejecuta el caso de uso SALC-06 Montar equipo. AGREGAR EQUIPO, se ejecuta el caso de uso SALC-05 Agregar equipo. Más información, se ejecuta el caso de uso SALC-07 Visualizar información detallada. Hacer clic sobre el nombre de una pieza, se ejecuta A-1	
	3	Termina el caso de uso.	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema valida que casillas de verificación han sido marcadas (E-4) (E-5) y ejecuta el caso de uso SALC-07 Visualizar información detallada.
		(2)	Termina el caso de uso.
	A-2	(1)	El sistema muestra un formulario con los campos “Fabricante”, “Categoría”, “Capacidad”, “Color”, “No. Serie”, “Etiqueta BUAP”, “Condición” y el botón “Registrar”.
		(2)	El usuario puede rellenar el formulario y elegir el botón “Registrar”.
		(3)	El sistema verifica los datos proporcionados (E-1) (E-2) (E-3).
		(4)	El sistema realiza una consulta a la base de datos y agrega la información del nuevo hardware a registrar.
(5)	Termina el caso de uso.		
<b>Excepciones:</b>	E-1	No se ha proporcionado un Fabricante. a) El sistema muestra un mensaje informando que no se ha introducido un fabricante e invita al colaborador a proporcionar uno. b) Continúa el caso de uso.	

	E-2	<p>No se ha especificado una categoría de hardware.</p> <p>a) El sistema muestra un mensaje informando que no se ha especificado una categoría de hardware e invita al colaborador a indicar uno.</p> <p>b) Continúa el caso de uso.</p>
	E-3	<p>No se ha especificado la condición de funcionalidad del hardware registrado.</p> <p>a) El sistema muestra un mensaje informando que no se ha introducido una condición e invita al usuario a indicar una.</p> <p>b) Continúa el caso de uso.</p>
	E-4	<p>No se ha seleccionado alguna casilla de verificación.</p> <p>a) El sistema muestra un mensaje informando al usuario que no ha seleccionado una casilla de verificación y lo invita a hacerlo.</p> <p>b) Continúa el caso de uso.</p>
	E-5	<p>Se ha seleccionado más de una casilla de verificación.</p> <p>a) El sistema muestra un mensaje informando al usuario que ha seleccionado múltiples casillas de verificación y lo invita a solo marcar una.</p> <p>b) Continúa el caso de uso.</p>
<b>Notas:</b>	<p>La sección de inventario de hardware es parte esencial del sistema, cuenta con control de seguridad de datos para evitar inconsistencias y ataques.</p>	

<b>ID – CU:</b>	SALC-09		
<b>Nombre de CU:</b>	Administrar acceso		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	13 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Actor(es):</b>	Colaborador		
<b>Descripción:</b>	Este caso de uso le permitirá al colaborador obtener una lista de las personas que actualmente usan el laboratorio.		
<b>Precondiciones:</b>	Haber completado el caso de uso SALC-00 “Autenticar personal”.		
<b>Disparador:</b>	Haber seleccionado “AUTOACCESO” en la pantalla. Haber completado (1) del flujo alternativo A-1 del caso de uso SALC-08 Administrar acceso. Haber completado (1) del flujo alternativo A-2 del caso de uso SALC-08 Administrar acceso.		
<b>Flujo normal:</b>	1	El sistema muestra una lista de instrucciones de uso, así como un formulario con los campos “Matricula”, “Nombre(s)”, “Apellido paterno”, “Apellido Materno”, “Asignar equipo”, una lista de las personas que se han registrado para usar las instalaciones con una casilla de verificación a la izquierda de cada miembro y las opciones ENTRADA y SALIDA.	
	2	El colaborador puede proporcionar la información de la persona que desea utilizar las instalaciones (E-1) o puede seleccionar la casilla de verificación de algún miembro de la lista de personas.	
	3	El colaborador selecciona una opción: ENTRADA, el sistema ejecuta A-1. SALIDA, el sistema ejecuta A-2.	
	4	Termina el caso de uso.	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema registra los datos de la persona especificada en el formulario (E-1) y muestra un mensaje de éxito en el registro.
		(2)	Termina el caso de uso.
	A-2	(1)	El sistema elimina de la lista los miembros con la casilla de verificación correspondiente (E-2), registra la hora actual como hora de salida y muestra un mensaje de éxito en la operación.
		(2)	Termina el caso de uso.
<b>Excepciones:</b>	E-1	Falta algún dato. a) El sistema muestra un mensaje de fallo. b) El caso de uso continúa.	
		E-2	No hay alguna casilla de verificación activada. a) La consulta a la base de datos no es mandada y se muestra un mensaje especificando que se seleccione alguna casilla de verificación. b) El caso de uso continúa.
<b>Notas:</b>	El control de auto acceso es otro aspecto fundamental del sistema.		

<b>ID – CU:</b>	SALC-11		
<b>Nombre de CU:</b>	Registrar incidente.		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	14 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Actor(es):</b>	Colaborador Coordinador		
<b>Descripción:</b>	Este caso de uso le permitirá al usuario registrar un incidente ocurrido en las instalaciones del laboratorio.		
<b>Precondiciones:</b>	Haber completado exitosamente el caso de uso SALC-00 Autenticar personal		
<b>Disparador:</b>	Haber seleccionado la opción REPORTAR INCIDENTE en el caso de uso SALC-10 Consultar incidentes. Haber oprimido el botón “Reportar pieza” en el caso de uso SALC-07 Visualizar información detallada.		
<b>Flujo normal:</b>	1	El sistema ofrece al usuario un formulario para registrar el incidente, instrucciones y el botón “Completar reporte”.	
	2	El sistema reconoce el medio de acceso a esta sección: Si el disparador fue haber oprimido el botón “Reportar pieza” en el caso de uso SALC-07 Visualizar información detallada, se ejecuta A-1. Si el disparador fue haber seleccionado la opción REPORTAR INCIDENTE en el caso de uso SALC-10 Consultar incidentes, se ejecuta A-2.	
	3	El sistema muestra un mensaje de operación exitosa.	
	4	Termina el caso de uso.	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema muestra además una leyenda de que pieza está reportando y no mostrará los campos de selección para piezas de hardware.
		(2)	El colaborador proporcionara un título del reporte del fallo, un resumen del reporte, una descripción de máximo 500 caracteres. Una fecha y hora en la que sucedió el evento, luego elige la opción ENVIAR (E-1) (E-2) (E-3) (E-4) (E-5) (E-6) (E-7).
		(3)	Continúa el caso de uso.
	A-2	(1)	El colaborador proporcionara un título del reporte del fallo, un resumen del reporte, una descripción de máximo 500 caracteres. Puede proporcionar diversos datos del hardware afectado, una “fecha” y “hora” en la que sucedió el evento, luego elige la opción ENVIAR (E-1) (E-2) (E-3) (E-4) (E-5) (E-6) (E-7).
		(2)	Continúa el caso de uso.
<b>Excepciones:</b>	E-1	No se ha especificado un título del reporte. a) El sistema muestra un mensaje en el que especifica que el título es obligatorio. b) El caso de uso continúa.	

	E-2	No se ha especificado una descripción. a) El sistema muestra un mensaje en el que se especifica que la descripción es inválida e invita al colaborador a escribir una descripción de máximo 500 caracteres. b) El caso de uso continúa.
	E-3	No se ha especificado un resumen. a) El sistema muestra un mensaje en el que se especifica que el resumen es un campo obligatorio. b) El caso de uso continúa.
	E-4	No se ha especificado la hora del suceso. a) El sistema muestra un mensaje informando que no se ha indicado una hora del suceso. b) El caso de uso continúa.
	E-5	No se ha especificado los minutos de la hora del suceso. a) El sistema muestra un mensaje informando que no se ha indicado los minutos de la hora del suceso. b) El caso de uso continúa.
	E-6	No se ha especificado la condición del meridiano. a) El sistema muestra un mensaje informando que no se ha indicado la condición AM o PM de la hora. b) El caso de uso continúa.
	E-7	No se ha especificado la fecha del suceso. a) El sistema muestra un mensaje informando que no se ha indicado la fecha del suceso. b) El caso de uso continúa.
<b>Notas:</b>		El registro de reportes de incidentes cuenta con múltiples formas de reportar el cual deberá ser instruido posteriormente.

<b>ID – CU:</b>	SALC-13		
<b>Nombre de CU:</b>	Reportar hardware		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	14 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Actor(es):</b>	Colaborador Coordinador		
<b>Descripción:</b>	Este caso de uso permitirá al usuario levantar diferentes reportes de hardware y exportarlos a un archivo PDF.		
<b>Precondiciones:</b>	Haber completado exitosamente el caso de uso SALC-00 “Autenticar personal”		
<b>Disparador:</b>	Haber seleccionado la opción HARDWARE en el caso de uso SALC-12 “Generar reportes”		
<b>Flujo normal:</b>	1	El sistema ofrece al usuario instrucciones y las opciones “Reporte general de hardware”, “Reporte de refacciones”, “Reporte de piezas dañadas”, “Reporte de armado de equipos”, “Reporte de equipo”.	
	2	El usuario elige una opción: Reporte de armado de equipos, el sistema ejecuta A-1. Reporte de equipo, el sistema ejecuta A-2. Reporte de refacciones, el sistema ejecuta A-3. Reporte de piezas dañadas, el sistema ejecuta A-4. Reporte general de hardware, el sistema ejecuta A-5.	
	4	Termina el caso de uso.	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema ejecuta una consulta a la base de datos, exporta un reporte de piezas ordenado por identificador de equipo de manera ascendente a un archivo PDF.
		(2)	El usuario obtiene el reporte generado por medio del gestor de descargas de su navegador.
		(3)	Continúa el caso de uso.
	A-2	(1)	El sistema muestra un formulario para elegir un número de equipo.
		(2)	El usuario selecciona un número de equipo y oprimir el botón “Generar”.
		(3)	El sistema ejecuta una consulta a la base de datos, exporta un reporte de piezas asignadas al equipo especificado en la caja de texto, a un archivo PDF.
		(4)	El usuario obtiene el reporte generado por medio del gestor de descargas de su navegador.
		(5)	Continúa el caso de uso.
	A-3	(1)	El sistema ejecuta una consulta a la base de datos, exporta un reporte de piezas ordenado por identificador de pieza los cuales no estén asignados a algún equipo, a un archivo PDF.
		(2)	El usuario obtiene el reporte generado por medio del gestor de descargas de su navegador.
		(3)	Continúa el caso de uso.

	A-4	(1)	El sistema ejecuta una consulta a la base de datos, exporta un reporte de piezas ordenado por fecha que haya sido registrada como defectuosa, a un archivo PDF.
		(2)	El usuario obtiene el reporte generado por medio del gestor de descargas de su navegador.
		(3)	Continúa el caso de uso.
	A-5	(1)	El sistema muestra un formulario para elegir un rango de fechas.
		(2)	El usuario puede seleccionar un rango de fechas y oprimir el botón "Generar" (E-1).
		(3)	El sistema ejecuta una consulta a la base de datos, exporta un reporte de piezas ordenado por fecha de registro de manera ascendente, a un archivo PDF.
		(4)	El usuario obtiene el reporte generado por medio del gestor de descargas de su navegador.
		(5)	Continúa el caso de uso.
<b>Excepciones:</b>	E-1	<p>La fecha límite es inferior a la fecha inicio.</p> <p>a) El sistema muestra un mensaje de error especificando que la fecha límite es inferior a la fecha de inicio y ofrece volver a introducirla.</p> <p>b) Continúa el caso de uso.</p>	
<b>Notas:</b>	El generador de reportes de hardware puede generar múltiples reportes de varias formas, se necesitará instrucción posterior.		

<b>ID – CU:</b>	SALC-18		
<b>Nombre de CU:</b>	Registrar usuario		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	20 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 12 de Octubre de 2013 27 de abril de 2014
<b>Actor(es):</b>	Coordinador Administrador		
<b>Descripción:</b>	Este caso de uso permitirá al coordinador registrar a un nuevo colaborador para el modulo. También le permitirá al administrador registrar un nuevo coordinador.		
<b>Precondiciones:</b>	Haber completado exitosamente el caso de uso SALC-00 “Autenticar personal”.		
<b>Disparador:</b>	Haber seleccionado la opción “Nuevo colaborador” en el caso de uso SALC-17 Administrar personal. Haber seleccionado la opción “Nuevo coordinador” en el caso de uso SALC-17 Administrar personal.		
<b>Flujo normal:</b>	1	El sistema muestra un formulario para capturar “Nombre de usuario”, “Contraseña”, “Confirmación de contraseña”, “Nombre”, “Apellido paterno”, “Apellido Materno”, “Numero de trabajador”, “Correo electrónico”, “Teléfono casa”, “Teléfono personal”, e indica que casillas son obligatorias para rellenar. El sistema reconoce el privilegio del usuario: Coordinador, se ejecuta A-1. Administrador, se ejecuta A-2.	
	2	El coordinador puede proporcionar los datos en el formulario.	
	3	El coordinador elige la opción REGISTRAR.	
	4	El sistema lee los datos proporcionados (E-1) (E-2) y realiza una consulta a la base de datos.	
	5	El sistema registra los cambios efectuados y manda un mensaje de éxito al coordinador.	
	6	Termina el caso de uso.	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema también muestra en pantalla la caja de texto para “Matricula”, “Carrera”, “Horario”, “Inicio actividad”. También el botón “Registrar”.
		(2)	El coordinador puede llenar el formulario y oprimir el botón “Registrar”. (E-1)(E-2)(E-3)(E-4)(E-5)(E-6)(E-7)(E-8)(E-9)(E-10).
		(3)	El sistema hace una consulta a la base de datos y registra al nuevo colaborador.
		(4)	Termina el caso de uso.

	A-2	(1)	El sistema también muestra en pantalla la caja de texto para “Numero de trabajador”, “Horario”, “Módulo a coordinar”. También el botón “Registrar”.
		(2)	El administrador puede llenar el formulario y oprimir el botón “Registrar”. E-1)(E-2)(E-3)(E-4)(E-5)(E-6)(E-7)(E-8)(E-9)
		(3)	El sistema hace una consulta a la base de datos y registra al nuevo coordinador.
		(4)	Termina el caso de uso.
<b>Excepciones:</b>	E-1		No se ha escrito nombre de usuario. a) El sistema envía un mensaje al usuario invitándole a que proporcione un nombre de usuario. b) El caso de uso continúa.
	E-2		No ha proporcionado una contraseña. a) El sistema informa al coordinador que falta proporcionar la contraseña. b) El caso de uso continúa.
	E-3		No ha confirmado la contraseña. a) El sistema indica al usuario que confirme la contraseña. b) El caso de uso continúa.
	E-4		No ha proporcionado el nombre. a) El sistema le indica al usuario que proporcione el nombre del nuevo personal a ser registrado. b) El caso de uso continúa.
	E-5		No ha proporcionado el apellido paterno. a) El sistema le indica al usuario que proporcione el apellido paterno del nuevo personal a ser registrado. b) El caso de uso continúa.
	E-6		No ha proporcionado el apellido materno. a) El sistema le indica al usuario que proporcione el apellido materno del nuevo personal a ser registrado. b) El caso de uso continúa.
	E-7		No ha proporcionado la matricula. a) El sistema le indica al usuario que proporcione la matricula o número de trabajador del nuevo personal a ser registrado. b) El caso de uso continúa.
	E-8		No ha proporcionado el correo electrónico. a) El sistema le indica al usuario que proporcione el correo electrónico del nuevo personal a ser registrado. b) El caso de uso continúa.
	E-9		No ha proporcionado el teléfono personal. a) El sistema le indica al usuario que proporcione el teléfono personal del nuevo usuario a ser registrado. b) El caso de uso continúa.

	E-10	No ha indicado el inicio de actividad. a) El sistema le indica al usuario establecer la fecha de inicio de actividad para el nuevo personal que será registrado. b) El caso de uso continúa.
<b>Notas:</b>		El módulo de registro de usuarios cuenta con validación de información inmediata y asincrónica para que los datos proporcionados sean válidos y conservar la solidez de la base de datos.

<b>ID – CU:</b>	SALC-25			
<b>Nombre de CU:</b>	Respaldar registros			
<b>Creado por:</b>	José Abraham Baez Bagatella			
<b>Fecha de creación:</b>	1 de Febrero de 2014	<b>Fecha de modificación:</b>	21 de Febrero de 2014	
<b>Actor(es):</b>	Administrador			
<b>Descripción:</b>	Mediante este caso de uso el administrador podrá crear archivos de respaldo de registros de la base de datos.			
<b>Precondiciones:</b>	Haber completado el caso de uso SALC-00 Autenticar personal.			
<b>Disparador:</b>	Haber seleccionado “Respaldar registros” en el caso de uso SALC-22 Administrar sistema.			
<b>Flujo normal:</b>	1	El sistema muestra instrucciones de uso y las opciones “Respaldo de la base de datos completa”, “Respaldo de registros de incidentes”, “Respaldo de registros de Autoacceso”, “Respaldo de recursos de hardware”, “Respaldo de recursos de software”, “Respaldo de personal registrado”.		
	2	El administrador elige una opción: Respaldo de la base de datos completa, se ejecuta A-1. Respaldo de registros de incidentes, se ejecuta A-2. Respaldo de registros de Autoacceso, se ejecuta A-2. Respaldo de recursos de hardware, se ejecuta A-2. Respaldo de recursos de software, se ejecuta A-2. Respaldo de personal registrado, se ejecuta A-3.		
	3	Termina el caso de uso.		
<b>Flujo alternativo:</b>	A-1	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la base de datos completa en un archivo que se le ofrece al administrador para descargar.	
		(2)	El administrador descarga el archivo.	
		(3)	Termina el caso de uso	
	A-2	(1)	El sistema ofrece un formulario para especificar un rango de fechas y el botón “Respaldar”.	
		(2)	El administrador puede especificar un rango de fechas y oprimir el botón “Respaldar” (E-1).	
		(3)	Si la opción elegida fue Respaldo de registros de incidentes, se ejecuta A-4. Respaldo de registros de Autoacceso, se ejecuta A-5. Respaldo de recursos de hardware, se ejecuta A-6. Respaldo de recursos de software, se ejecuta A-7	
		(4)	Continúa el caso de uso.	
	A-3	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la información del personal que opera el sistema en un archivo que se le ofrece al administrador para descargar.	
		(2)	El administrador descarga el archivo.	
(3)		Termina el caso de uso		

	A-4	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la información de los incidentes registrados en un archivo que se le ofrece al administrador para descargar.
		(2)	El administrador descarga el archivo.
		(3)	Termina el caso de uso
	A-5	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la información de los registros de auto acceso en un archivo que se le ofrece al administrador para descargar.
		(2)	El administrador descarga el archivo.
		(3)	Termina el caso de uso
	A-6	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la información de los recursos de hardware en un archivo que se le ofrece al administrador para descargar.
		(2)	El administrador descarga el archivo.
		(3)	Termina el caso de uso
	A-7	(1)	El sistema realiza una consulta a la base de datos y extrae una copia en modo SQL de la información de los recursos de software en un archivo que se le ofrece al administrador para descargar.
		(2)	El administrador descarga el archivo.
		(3)	Termina el caso de uso
<b>Excepciones:</b>	E-1	La fecha inicial es posterior a la fecha límite. a) El sistema le indica al administrador que la fecha inicial es posterior a la fecha límite y le invita a ofrecer una fecha límite posterior a la fecha inicial. b) Continúa el caso de uso.	
<b>Notas:</b>			

<b>ID – CU:</b>	SALC-26		
<b>Nombre de CU:</b>	Administrar edificio		
<b>Creado por:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	3 de Febrero de 2014	<b>Fecha de modificación:</b>	
<b>Actor(es):</b>	Administrador		
<b>Descripción:</b>	Con este caso de uso el administrador puede gestionar el agregado de nuevos laboratorios.		
<b>Precondiciones:</b>	Haber completado el caso de uso SALC-00 Autenticar personal.		
<b>Disparador:</b>	<p>Haber elegido la opción “Mi edificio” en la pantalla.</p> <p>Haber terminado con éxito el flujo A-1 del caso de uso SALC-26 Administrar edificio.</p> <p>Haber oprimido el botón “Cancelar” en el flujo A-2 del caso de uso SALC-26 Administrar edificio.</p> <p>Haber terminado con éxito el flujo A-2 del caso de uso SALC-26 Administrar edificio.</p>		
<b>Flujo normal:</b>	1	El sistema muestra instrucciones de uso y el botón “Nuevo módulo” así como una lista de los módulos previamente registrados y los botones “Eliminar Modulo”, “Asigna Coordinador”, “Remover asignación”, “Cambiar Nombre”.	
	2	<p>El administrador elige una opción:</p> <p>“Nuevo módulo”, se ejecuta A-1.</p> <p>“Eliminar Modulo”, el sistema pedirá confirmación al usuario, sí la hay se ejecuta A-2.</p> <p>“Asigna Coordinador”, se ejecuta A-3.</p> <p>“Remover asignación”, el sistema pedirá confirmación al usuario, sí la hay se ejecuta A-4</p> <p>“Cambiar Nombre”, se ejecuta A-5</p>	
	3	Termina el caso de uso	
<b>Flujo alternativo:</b>	A-1	(1)	El sistema muestra un formulario para establecerle un nombre al nuevo módulo y el botón “Agregar modulo”.
		(2)	El administrador puede escribir un nombre en la caja de texto y oprimir el botón “Agregar modulo”.
		(3)	El sistema registra el nuevo módulo en la base de datos y le asigna el nombre proporcionado.
		(4)	Termina el caso de uso.
	A-2	(1)	El sistema hace una consulta a la base de datos y marca la fecha de exclusión del módulo seleccionado, así como desasigna el coordinador y remueve la pertenencia de las piezas de hardware y programas de software de los módulos.
		(2)	Termina el caso de uso.

	A-3	(1)	El sistema ofrece en pantalla la lista de coordinadores disponibles con casillas de verificación a un lado de ellos, también los botones "Aceptar" y "Cancelar".
		(2)	El administrador elige una casilla de verificación y oprime el botón "Aceptar" (E-1) (E-2).
		(3)	El sistema realiza una consulta a la base de datos y asigna la responsabilidad del módulo al coordinador seleccionado.
		(4)	Termina el caso de uso.
	A-4	(1)	El sistema realiza una consulta a la base de datos y remueve la responsabilidad del coordinador actual del módulo, quedando sin responsable dicho modulo.
		(2)	Termina el caso de uso.
	A-5	(1)	El sistema ofrece una caja de texto y el botón "Cambiar".
		(2)	EL administrador puede escribir un nuevo nombre y oprimir el botón "Cambiar".
		(3)	El sistema realiza una consulta a la base de datos y cambia el nombre asignado al módulo.
		(4)	Termina el caso de uso.
	<b>Excepciones:</b>	E-1	No hay casillas de verificación activadas. a) El sistema muestra al usuario que no se han activado ninguna de casillas de verificación y lo invita a seleccionar una. b) Continúa el caso de uso.
		E-2	Hay más de una casilla de verificación activa. a) El sistema advierte al usuario que hay más de una casilla de verificación activa y lo invita a solo selecciona una. b) Continúa el caso de uso
<b>Notas:</b>			

<b>ID – CU:</b>	SALC-29	
<b>Nombre de CU:</b>	Instalar sistema	
<b>Creado por:</b>	José Abraham Baez Bagatella	
<b>Fecha de creación:</b>	1 de Marzo de 2014	<b>Fecha de modificación:</b>
<b>Actor(es):</b>	Administrador	
<b>Descripción:</b>	Con este caso de uso se puede instalar el sistema y registrar al administrador.	
<b>Precondiciones:</b>	Ninguna	
<b>Disparador:</b>	Acceder al fichero “install.php”.	
<b>Flujo normal:</b>	1	El sistema muestra un formulario de registro para el administrador, donde se le pedirá que proporcione su nombre de usuario, contraseña, confirmar contraseña, nombre, apellido paterno, apellido materno, número de identificación (IFE, numero de trabajador, cartilla de servicio militar), correo electrónico, teléfono de casa, teléfono personal, horario de atención, carrera cursada.
	2	El administrador llena el formulario con sus datos y oprime el botón “Instalar”(E-1) (E-2) (E-3) (E-4) (E-5) (E-6) (E-7) (E-8).
	3	El sistema empieza a realizar la instalación de la base de datos, procedimientos almacenados y registro del administrador, también es eliminado el archivo “install.php” del servidor. Luego se le presenta la página índice al administrador.
	4	Termina el caso de uso.
<b>Flujo alternativo:</b>	Ninguno	
<b>Excepciones:</b>	E-1	No ha proporcionado el nombre. a) El sistema le indica al usuario que proporcione su nombre. b) El caso de uso continúa.
	E-2	No ha proporcionado el apellido paterno. a) El sistema le indica al usuario que proporcione su apellido paterno. b) El caso de uso continúa.
	E-3	No ha proporcionado el apellido materno. a) El sistema le indica al usuario que proporcione su apellido materno. b) El caso de uso continúa.
	E-4	No ha proporcionado la matricula. a) El sistema le indica al usuario que proporcione la matricula o número de trabajador. b) El caso de uso continúa.
	E-5	No ha proporcionado el correo electrónico. a) El sistema le indica al usuario que proporcione el correo electrónico donde se le pueda contactar. b) El caso de uso continúa.
	E-6	No ha proporcionado el teléfono personal. a) El sistema le indica al usuario que proporcione su teléfono personal. b) El caso de uso continúa.

	E-7	<p>No se ha proporcionado el horario de atención.</p> <ul style="list-style-type: none"> <li>a) El sistema le indica al usuario que la información del horario de atención es necesaria.</li> <li>b) El caso de uso continúa.</li> </ul>
	E-8	<p>No ha proporcionado el nivel de estudios.</p> <ul style="list-style-type: none"> <li>a) El sistema le indica al usuario que es necesario escribir la carrera profesional que curso.</li> <li>b) El caso de uso continúa.</li> </ul>
<b>Notas:</b>		

### 2.7.4 Escenarios

Un escenario es una instancia de un caso de uso, es decir una ejemplificación de la interacción entre un usuario y el sistema. Cada caso de uso cuenta con 3 tipos diferentes de escenario: básico, alterno y excepción.

Se expondrán los escenarios de los casos de uso seleccionados anteriormente mediante el formato de una tabla.

**Tabla 8. Formato de escenarios de casos de uso.**

<b>Identificador de caso de uso:</b>			
<b>Tipo de flujo:</b>			
<b>Autor:</b>			
<b>Fecha de creación:</b>		<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1			
2			
3			

Un escenario es redactado simulando un diálogo entre el sistema y el usuario, los usuarios poseen un nombre real y son parte del personal de los laboratorios de cómputo.

*Escenarios para el caso de uso SALC-01 - Inventariar software*

<b>Identificador de caso de uso:</b>	SALC-01		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	21 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Martín Valencia ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Software"		
2	El sistema muestra una lista con el software previamente registrado, casillas de verificación a la izquierda de cada programa, el botón "Excluir o cambiar", y las opciones "Registrar nuevo, Registrar instalación de software".		
3	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-01		
<b>Tipo de flujo:</b>	Alternativo		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	29 de julio de 2013	<b>Fecha de modificación:</b>	10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Martín Valencia ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Software"		
2	El sistema muestra una lista con el software previamente registrado, casillas de verificación a la izquierda de cada programa, el botón "Excluir o cambiar", y las opciones "Registrar nuevo, Registrar instalación de software".		
3	Martín selecciona la opción "Registrar nuevo".		
4	El sistema muestra un formulario con instrucciones, los campos "Fabricante, Nombre, Versión y Serie" y el botón "Registrar nuevo".		
5	Martín llena el formulario asignando al campo compañía desarrolladora con "Maplesoft", el campo "nombre" con "MAPLE" y versión "17". Luego elige la opción "Registrar nuevo".		
6	El sistema valida que el nombre y la versión sean válidos, luego realiza una consulta a la base de datos para crear un nuevo registro con los datos proporcionados.		
7	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-01		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	29 de julio de 2013	<b>Fecha de modificación:</b>	10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Martín Valencia ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Software"		
2	El sistema muestra una lista con el software previamente registrado, casillas de verificación a la izquierda de cada programa, el botón "Excluir o cambiar", y las opciones "Registrar nuevo, Registrar instalación de software".		
3	Martín selecciona la opción "Registrar nuevo".		
4	El sistema muestra un formulario con instrucciones, los campos "Fabricante, Nombre, Versión y Serie" y el botón "Registrar nuevo".		
5	Martín llena el formulario asignando al campo compañía desarrolladora con "Maplesoft" y el campo "nombre" con "MAPLE". Luego elige la opción "Registrar nuevo".		
6	El sistema causa una excepción.		
7	El sistema informa con un mensaje que la versión del software no ha sido proporcionada.		
8	El caso de uso continúa.		

*Escenarios para el caso de uso SALC-04 - Inventariar hardware*

<b>Identificador de caso de uso:</b>	SALC-04		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	29 de julio de 2013	<b>Fecha de modificación:</b>	10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Enrique Ramos ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Hardware"		
2	El sistema muestra instrucciones, una lista del hardware previamente registrado, el botón "más información" y las opciones "Registrar nuevo, Agregar equipo y Ver armado de equipos".		
3	Enrique elige alguna opción.		
4	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-04		
<b>Tipo de flujo:</b>	Alterno		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	29 de julio de 2013	<b>Fecha de modificación:</b>	10 de noviembre de 2013
Paso			
Paso	Detalle		
1	Enrique Ramos ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Hardware"		
2	El sistema muestra instrucciones, una lista del hardware previamente registrado, el botón mas información y las opciones "Registrar nuevo, Agregar equipo y Ver armado de equipos".		
3	Enrique elige la opción "Registrar nuevo".		
4	El sistema muestra un formulario con las opciones "Fabricante", "Categoría", "Capacidad", "Color", "No. Serie", "Etiqueta BUAP", "Condición" y el botón "Registrar".		
5	Enrique rellena el formulario con los datos: Fabricante: Hewlett Packard; Categoría: Monitor; Color: Negro; No. Serie: 009456; Etiqueta BUAP: 804556478; Condición: Funcional. Y luego oprime el botón "Registrar"		
6	El sistema valida que los datos tengan un formato correcto, luego realiza una consulta a la base de datos para corroborar que la pieza no haya sido registrada con anterioridad, por ultimo agrega el nuevo registro con los datos proporcionados.		
7	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-04		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	29 de julio de 2013	<b>Fecha de modificación:</b>	10 de noviembre de 2013
Paso			
Paso	Detalle		
1	Enrique Ramos ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Mi laboratorio", después elige el apartado "Inventario de Hardware"		
2	El sistema muestra instrucciones, una lista del hardware previamente registrado, el botón mas información y las opciones "Registrar nuevo, Agregar equipo y Ver armado de equipos".		
3	Enrique elige la opción "Registrar nuevo".		
4	El sistema muestra un formulario con las opciones "Fabricante", "Categoría", "Capacidad", "Color", "No. Serie", "Etiqueta BUAP", "Condición" y el botón "Registrar".		
5	Enrique rellena el formulario con los datos: Fabricante: Hewlett Packard; Categoría: Monitor; Color: Negro; No. Serie: 009456; Etiqueta BUAP: 804556478. Y luego oprime el botón "Registrar"		
6	El sistema valida que los datos tengan un formato correcto, verifica que el campo de condición no ha sido rellenado y causa una excepción.		
7	El sistema le informa a Enrique que no ha especificado una condición de funcionamiento y le invita a que la indique.		
8	Termina el caso de uso.		

## Escenarios para el caso de uso SALC-09 - Administrar acceso

<b>Identificador de caso de uso:</b>	SALC-09		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Eduardo Olivera ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Autoacceso".		
2	El sistema muestra instrucciones, una lista con el nombre de las personas que actualmente están utilizando las instalaciones, un formulario con campos de información para registrar el acceso a una nueva persona, casillas de verificación a la izquierda de cada miembro de la lista y los botones "Entrada" y "Salida".		
3	Eduardo rellena el formulario con los datos de un alumno que desea entrar a hacer uso de las instalaciones y oprime el botón "Entrada".		
4	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-09		
<b>Tipo de flujo:</b>	Alternativo		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Eduardo Olivera ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Autoacceso".		
2	El sistema muestra instrucciones, una lista con el nombre de las personas que actualmente están utilizando las instalaciones, un formulario con campos de información para registrar el acceso a una nueva persona, casillas de verificación a la izquierda de cada miembro de la lista y los botones "Entrada" y "Salida".		
3	Eduardo quiere registrar la salida de "Abraham Baez" con matrícula "200514799", equipo asignado "6" y hora de entrada "14:00" horas. Selecciona la casilla de verificación donde se han guardado los datos de Abraham Baez. Luego selecciona la opción SALIDA.		
4	El sistema hace una consulta a la base de datos, modifica el campo "hora de salida" asignando la hora actual como contenido. El sistema elimina de la lista en pantalla el registro de Abraham Baez y muestra un mensaje de éxito.		
5	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-09		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 10 de noviembre de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Eduardo Olivera ha realizado con éxito el caso de uso "Autenticar Personal" y da clic en la sección "Autoacceso".		
2	El sistema muestra instrucciones, una lista con el nombre de las personas que actualmente están utilizando las instalaciones, un formulario con campos de información para registrar el acceso a una nueva persona, casillas de verificación a la izquierda de cada miembro de la lista y los botones "Entrada" y "Salida".		
3	Eduardo desea registrar a Abraham Baez para que acceda a las instalaciones. Llena el formulario con su nombre "Abraham Baez Bagatella", su matrícula "2000514799". Luego oprime el botón "Entrada".		
4	El sistema informa que no se ha asignado un equipo e invita a Eduardo a asignarlo.		
5	Continúa el caso de uso.		

### *Escenarios para el caso de uso SALC-11 - Registrar incidente*

<b>Identificador de caso de uso:</b>	SALC-11		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Paso</b>	<b>Detalle</b>		
1	David ha accedido a la opción "Registrar incidente".		
2	El sistema muestra instrucciones para escribir el reporte, un formulario con campos para escribir el reporte y el botón "Completar reporte".		
3	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-11		
<b>Tipo de flujo:</b>	Alterno		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Paso</b>	<b>Detalle</b>		
1	David ha accedido a la opción "Registrar incidente" desde el caso de uso SALC-07 Visualizar información detallada oprimiendo el botón "Reportar Pieza".		
2	El sistema ofrece un formulario con los campos "Titulo", "Resumen", "Descripción", "Condición de funcionamiento", "Hora del incidente", "Fecha del incidente" y el botón "Completar Reporte".		
3	David desea registrar que se ha dañado un cable de red. David coloca como título del reporte "Cable averiado", como resumen "Se ha roto el cable de red de la maquina 20", la descripción es "Una mesa ha aplastado el cable de red del equipo 20 y se han roto algunos cables internos, se especifica la condición de funcionamiento como "No funcional" la hora del incidente es 5:00 pm con fecha del 14 de noviembre de 2013 y luego oprime el botón "Completar reporte".		
4	El sistema valida los datos y realiza una consulta a la base de datos creando un nuevo registro con los datos del incidente, luego informa que el reporte ha sido realizado con éxito.		
5	Termina el caso de uso		

<b>Identificador de caso de uso:</b>	SALC-11		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	22 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Paso</b>	<b>Detalle</b>		
1	David ha accedido a la opción "Registrar incidente".		
2	El sistema ofrece un formulario con los campos "Titulo", "Resumen", "Descripción", "Equipo", "Parte", "Fabricante", "No. Serie", "No. BUAP", "Condición de funcionamiento", "Hora del incidente", "Fecha del incidente" y el botón "Completar Reporte".		
3	David desea registrar que se ha extraviado una silla. David coloca como título del reporte "Silla extraviada", como resumen "Falta una silla en la fila 3", la descripción es "Hoy a las 19:30 horas en la última visita que tuve al laboratorio antes de cerrarlo hice conteo de material de las instalaciones y note que falta una silla en la fila 3", el fabricante es "Ninguno" y el número de serie es 3, se especifica la condición de funcionamiento como "No funcional" y luego oprime el botón "Completar reporte".		
4	El sistema valida los datos e informa que no se ha especificado una hora del incidente e invita a David a indicarla.		
5	El caso de uso continúa.		

## Escenarios para el caso de uso SALC-13 - Reportar hardware

<b>Identificador de caso de uso:</b>	SALC-13		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	14 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Zulema Castillo ha accedido a la sección "Generar reportes de hardware".		
2	El sistema muestra las opciones "Reporte general de hardware", "Reporte de refacciones", "Reporte de piezas dañadas", "Reporte de armado de equipos", "Reporte de equipo".		
3	Termina el caso de uso.		

Identificador de caso de uso:	SALC-13		
Tipo de flujo:	Alterno		
Autor:	José Abraham Baez Bagatella		
Fecha de creación:	14 de julio de 2013	Fecha de modificación:	29 de julio de 2013
Paso	Detalle		
1	Zulema Castillo ha accedido a la sección "Generar reportes de hardware".		
2	El sistema muestra las opciones "Reporte general de hardware", "Reporte de refacciones", "Reporte de piezas dañadas", "Reporte de armado de equipos", "Reporte de equipo".		
3	Zulema da clic en la opción "Reporte general de hardware".		
4	El sistema ofrece instrucciones, un formulario para indicar un rango de fechas para el reporte, botones de borrado para las fechas y el botón "Generar".		
5	Zulema no rellena ningún campo y oprime el botón "Generar"		
6	El sistema valida las fechas proporcionadas, pueden estar en blanco para crear una colección de registros sin restricción de fechas. Realiza una consulta a la base de datos y extrae la información del hardware registrado, luego genera un archivo PDF con esta información y la presenta en el navegador.		
7	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-13		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	14 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013
<b>Paso</b>	<b>Detalle</b>		
1	Zulema Castillo ha accedido a la sección "Generar reportes de hardware".		
2	El sistema muestra las opciones "Reporte general de hardware", "Reporte de refacciones", "Reporte de piezas dañadas", "Reporte de armado de equipos", "Reporte de equipo".		
3	Zulema da clic en la opción "Reporte general de hardware".		
4	El sistema ofrece instrucciones, un formulario para indicar un rango de fechas para el reporte, botones de borrado para las fechas y el botón "Generar".		
5	Zulema indica como primera fecha el 10 de enero de 2014 y como segunda fecha el 1 de enero de 2014, después oprime el botón "Generar".		
6	El sistema valida las fechas proporcionadas, indica que la fecha de inicio es posterior a la fecha límite y le invita a revisar las fechas una vez más.		
7	El caso de uso continúa.		

### *Escenarios para el caso de uso SALC-18 - Registrar usuario*

<b>Identificador de caso de uso:</b>	SALC-18		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	20 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 12 de Octubre de 2013 27 de abril de 2014
<b>Paso</b>	<b>Detalle</b>		
1	Rosa García ha accedido a la sección para registrar un nuevo colaborador.		
2	El sistema muestra un formulario para capturar "Nombre de usuario", "Contraseña", "Confirmación de contraseña", "Nombre", "Apellido paterno", "Apellido Materno", "Matricula", "Carrera", "Horario", "Inicio actividad", "Correo electrónico", "Teléfono casa", "Teléfono personal", e indica que casillas son obligatorias para rellenar. También el botón "Registrar"		
3	Rosa desea registrar a Zulema Castillo Castillo como colaboradora del laboratorio e ingresa sus datos. Luego oprime el botón "Registrar".		
4	El sistema valida los datos proporcionados y realiza una consulta a la base de datos validando que la información no sea repetida y luego realiza un nuevo registro con los datos proporcionados.		
5	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-18		
<b>Tipo de flujo:</b>	Alterno		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	20 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 12 de Octubre de 2013 27 de abril de 2014
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez realizará el registro de un nuevo coordinador y accesa a la sección registrar nuevo coordinador.		
2	El sistema muestra un formulario para capturar “Nombre de usuario”, “Contraseña”, “Confirmación de contraseña”, “Nombre”, “Apellido paterno”, “Apellido Materno”, “Numero de trabajador”, “Horario”, “Módulo a coordinar” “Correo electrónico”, “Teléfono casa”, “Teléfono personal”, e indica que casillas son obligatorias para rellenar.		
3	Abraham proporciona los datos de un nuevo coordinador y luego oprime el botón “Registrar”.		
4	El sistema valida los datos proporcionados y realiza una consulta a la base de datos validando que la información no sea repetida y luego realiza un nuevo registro con los datos proporcionados.		
5	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-18		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	20 de julio de 2013	<b>Fecha de modificación:</b>	29 de julio de 2013 12 de Octubre de 2013 27 de abril de 2014
<b>Paso</b>	<b>Detalle</b>		
1	Rosa García ha accedido a la sección para registrar un nuevo colaborador.		
2	El sistema muestra un formulario para capturar “Nombre de usuario”, “Contraseña”, “Confirmación de contraseña”, “Nombre”, “Apellido paterno”, “Apellido Materno”, “Matricula”, “Carrera”, “Horario”, “Inicio actividad”, “Correo electrónico”, “Teléfono casa”, “Teléfono personal”, e indica que casillas son obligatorias para rellenar. También el botón “Registrar”		
3	Rosa desea registrar a Zulema Castillo Castillo como colaboradora del laboratorio e ingresa sus datos. Luego oprime el botón “Registrar”.		
4	El sistema valida los datos proporcionados y realiza una consulta a la base de datos, luego el sistema verifica que el nombre de usuario que se eligió ya está en uso y le indica a Rosa que debe proporcionar otro.		
5	El caso de uso continúa.		

*Escenarios para el caso de uso SALC-25 - Respaldo registros*

<b>Identificador de caso de uso:</b>	SALC-25		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	1 de Febrero de 2014	<b>Fecha de modificación:</b>	21 de Febrero de 2014
<b>Paso</b>			
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez accede a la sección "Respaldo registros".		
2	El sistema muestra instrucciones y las opciones "Respaldo de la base de datos completa", "Respaldo de registros de incidentes", "Respaldo de registros de Autoacceso", "Respaldo de recursos de hardware", "Respaldo de recursos de software", "Respaldo de personal registrado".		
3	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-25		
<b>Tipo de flujo:</b>	Alternativo		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	1 de Febrero de 2014	<b>Fecha de modificación:</b>	21 de Febrero de 2014
<b>Paso</b>			
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez accede a la sección "Respaldo registros".		
2	El sistema muestra instrucciones y las opciones "Respaldo de la base de datos completa", "Respaldo de registros de incidentes", "Respaldo de registros de Autoacceso", "Respaldo de recursos de hardware", "Respaldo de recursos de software", "Respaldo de personal registrado".		
3	Abraham oprime la opción "Respaldo de la base de datos completa".		
4	El sistema realiza una consulta a la base de datos y crea una copia de todos los registros en un archivo de texto con formato y extensión SQL, luego lo presenta a Abraham como archivo listo para descargarse.		
5	Abraham acepta o rechaza la descarga del archivo.		
6	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-25		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	1 de Febrero de 2014	<b>Fecha de modificación:</b>	21 de Febrero de 2014
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez accede a la sección "Respaldo registros".		
2	El sistema muestra instrucciones y las opciones "Respaldo de la base de datos completa", "Respaldo de registros de incidentes", "Respaldo de registros de Autoacceso", "Respaldo de recursos de hardware", "Respaldo de recursos de software", "Respaldo de personal registrado".		
3	Abraham oprime la opción "Respaldo de registros de incidentes".		
4	El sistema muestra instrucciones para este tipo de respaldo y dos campos para un rango de fechas, botones para borrar los campos y el botón "Respaldo".		
5	Abraham rellena el primer campo de fecha con 1 de marzo de 2014 y el segundo campo con el 1 de febrero de 2014.		
6	El sistema valida los datos de los campos de fechas e indica que la fecha límite es posterior a la fecha inicial e invita a Abraham a corregir el rango.		
7	Continúa el caso de uso.		

*Escenarios para el caso de uso SALC-26 - Administrar edificio*

<b>Identificador de caso de uso:</b>	SALC-26		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	3 de febrero de 2014	<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez da clic en la sección "Mi edificio".		
2	El sistema muestra el botón "Nuevo módulo" y una lista con los módulos ya registrados, también los botones "Eliminar módulo", "Asigna coordinador", "Remover asignación", "Cambiar nombre".		
3	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-26		
<b>Tipo de flujo:</b>	Alterno		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	3 de febrero de 2014	<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez da clic en la sección "Mi edificio".		
2	El sistema muestra el botón "Nuevo módulo" y una lista con los módulos ya registrados, también los botones "Eliminar módulo", "Asigna coordinador", "Remover asignación", "Cambiar nombre".		
3	Abraham desea agregar un nuevo módulo y oprime el botón "Nuevo módulo".		
4	El sistema muestra un campo para nombrar el módulo y el botón "Agregar módulo".		
5	Abraham escribe el nombre del nuevo módulo como "104C-101" y oprime el botón "Agregar módulo".		
6	El sistema realiza una consulta a la base de datos y agrega la serie de registros para el funcionamiento de un nuevo módulo y le informa a Abraham que el registro fue exitoso.		
7	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-26		
<b>Tipo de flujo:</b>	Excepción.		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	3 de febrero de 2014	<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez da clic en la sección "Mi edificio".		
2	El sistema muestra el botón "Nuevo módulo" y una lista con los módulos ya registrados, también los botones "Eliminar módulo", "Asigna coordinador", "Remover asignación", "Cambiar nombre".		
3	Abraham desea asignar un coordinador a un módulo y oprime el botón correspondiente "Asigna coordinador".		
4	El sistema muestra una lista con los coordinadores registrados con casillas de verificación a la izquierda de sus nombres, los botones "Aceptar" y "Cancelar".		
5	Abraham no selecciona alguna casilla de verificación y oprime el botón "Aceptar"		
6	El sistema le indica a Abraham que debe al menos seleccionar una casilla de verificación.		
7	Continúa el caso de uso.		

## Escenarios para el caso de uso SALC-29 - Instalar sistema

<b>Identificador de caso de uso:</b>	SALC-29		
<b>Tipo de flujo:</b>	Básico		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	1 de Marzo de 2014	<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez desea instalar el sistema "SALC-FCC" en el servidor web, así que accede al archivo "install.php" en el servidor.		
2	La pantalla muestra instrucciones y un formulario para recabar el "Nombre de usuario", "Contraseña", "Confirmación de contraseña", "Nombre", "Apellido paterno", "Apellido materno", "Número de identificación", "Dirección de correo electrónico", "Teléfono de casa", "teléfono personal", "horario de atención", "carrera cursada" del administrador a registrar y el botón "Instalar".		
3	Abraham rellena los datos que se le piden y luego oprime el botón "Instalar".		
4	El sistema valida los datos, siendo estos correctos crea la base de datos en el servidor, genera los procedimientos almacenados de la base de datos y realiza el registro de los datos del administrador. Una vez terminado eso elimina el fichero "install.php" del servidor y re direcciona el navegador al índice del sistema.		
5	Termina el caso de uso.		

<b>Identificador de caso de uso:</b>	SALC-29		
<b>Tipo de flujo:</b>	Excepción		
<b>Autor:</b>	José Abraham Baez Bagatella		
<b>Fecha de creación:</b>	1 de Marzo de 2014	<b>Fecha de modificación:</b>	
<b>Paso</b>	<b>Detalle</b>		
1	Abraham Baez desea instalar el sistema "SALC-FCC" en el servidor web, así que accede al archivo "install.php" en el servidor.		
2	La pantalla muestra instrucciones y un formulario para recabar el "Nombre de usuario", "Contraseña", "Confirmación de contraseña", "Nombre", "Apellido paterno", "Apellido materno", "Número de identificación", "Dirección de correo electrónico", "Teléfono de casa", "teléfono personal", "horario de atención", "carrera cursada" del administrador a registrar y el botón "Instalar".		
3	Abraham rellena los datos, pero no especifica un horario de atención y luego oprime el botón "Instalar".		
4	El sistema valida los datos y advierte que el campo "Horario de atención" esta vacío, le informa a Abraham que ese campo es obligatorio y lo invita a especificarlo.		
5	Continúa el caso de uso.		

### 2.7.5 Diagrama de clases

El diagrama de clases utilizado como modelo de dominio refleja el conexto del sistema de administración de los laboratorios de cómputo (figura 2.7.4), es decir, los objetos tangibles y pensables que se han analizado en el uso de los laboratorios. Estas clases son acercamientos al modelo de base de datos posterior, donde cada clase podría ser una entidad en un modelo entidad-relación.

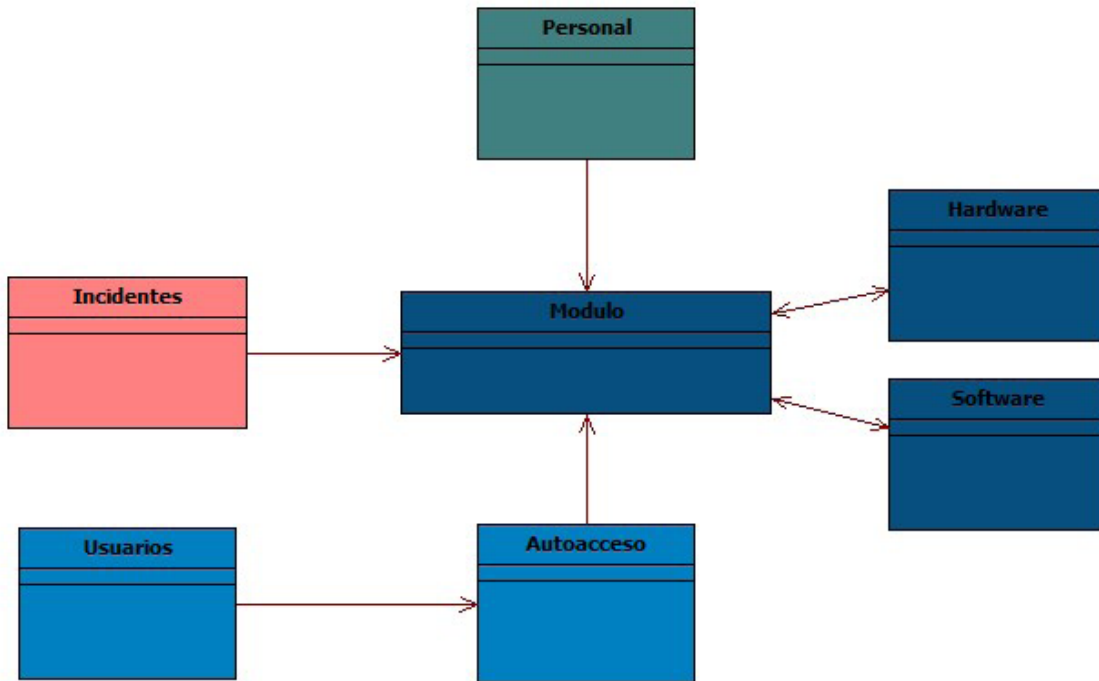


Figura 2.7.4. Diagrama de clases del sistema de administración de los laboratorios de cómputo

## **2.8 DISEÑO DEL SISTEMA**

### ***2.8.1 Diagramas de realización de casos de uso.***

Los diagramas de realización de casos de uso junto con los diagramas de colaboración, pertenecen al modelo de análisis del sistema. Los diagramas de realización de casos de uso o VOPC por sus siglas en inglés (View of Participating Classes) están conformados por los actores, clases de análisis y las asociaciones entre ellas.

Un diagrama de realización de casos de uso expresa la funcionalidad total del caso de uso, todos sus flujos y la interacción entre las clases.

El sistema de administración de los laboratorios de cómputo, cuenta con 29 casos de uso, de los cuales se ha abstraído un total de 60 clases de análisis conformadas por 36 clases interfaz, 19 clases de control y 5 clases entidad. Se expondrán los diagramas de los casos de uso anteriormente presentados.

Diagrama de realización de casos de uso para SALC-01 Inventariar software

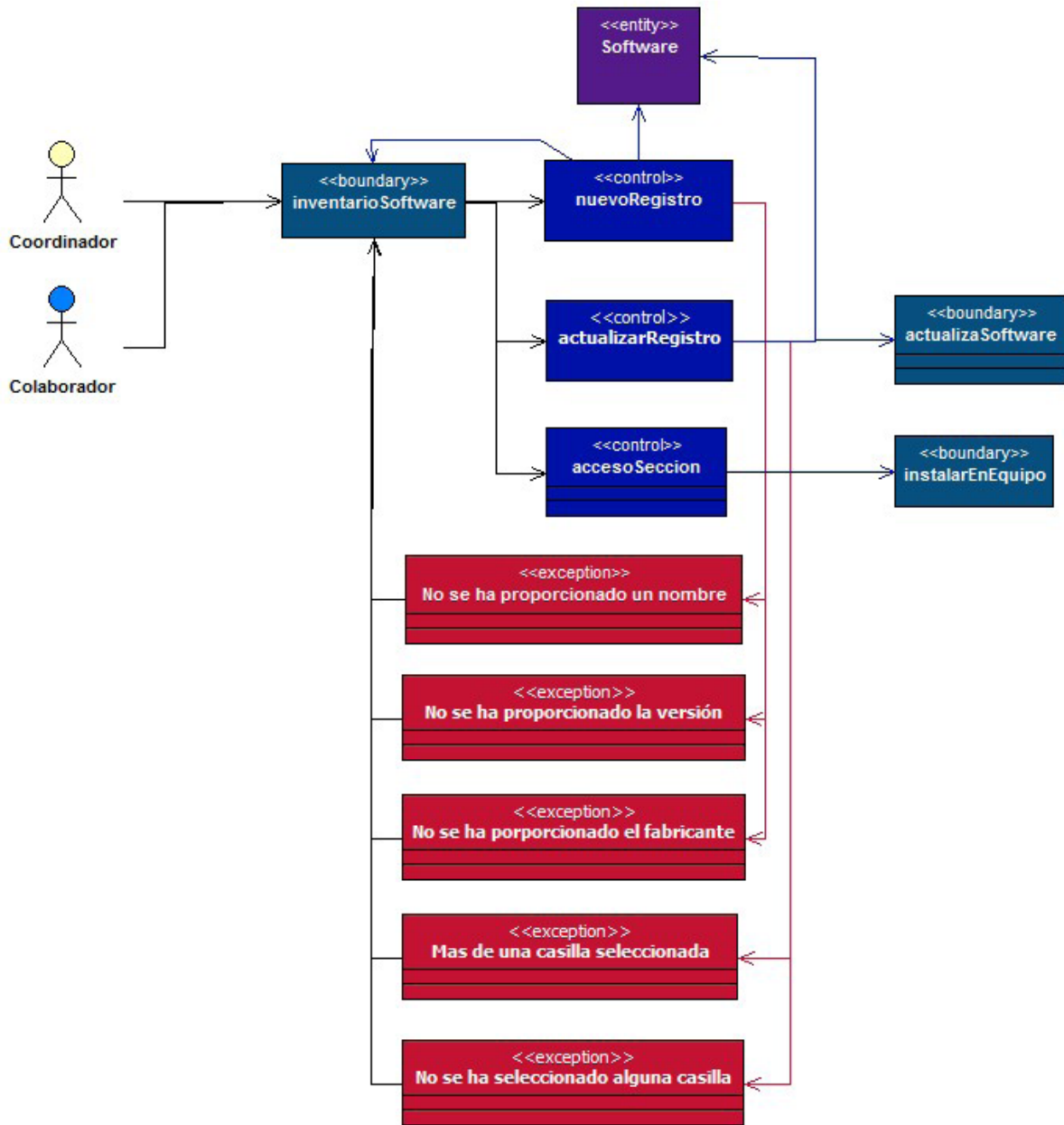


Figura 2.8.1. Diagrama de realización del caso de uso inventariar software

Diagrama de realización de casos de uso para SALC-04 Inventariar hardware

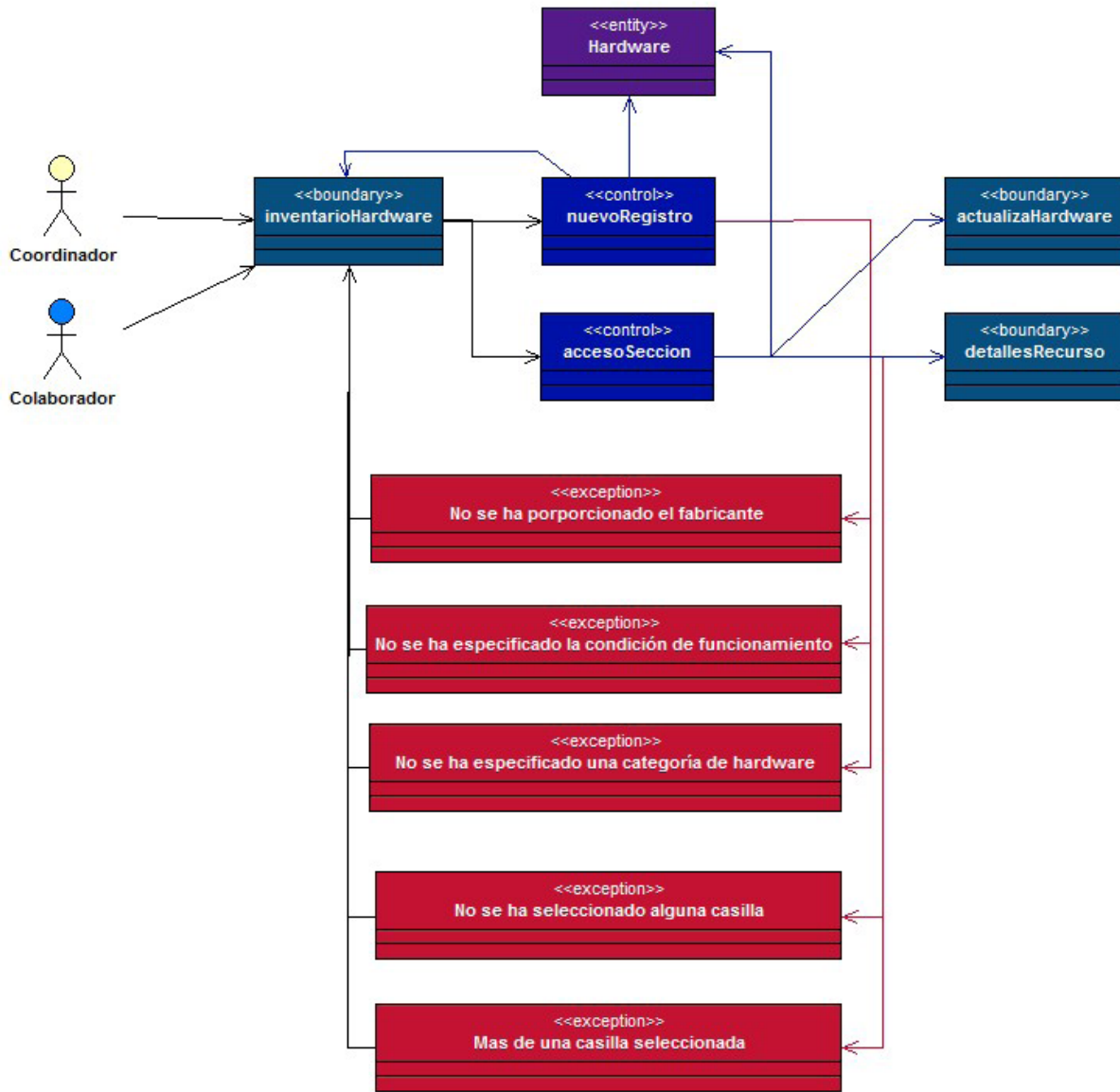


Figura 2.8.2. Diagrama de realización del caso de uso inventariar hardware

Diagrama de realización de casos de uso para SALC-09 Administrar acceso

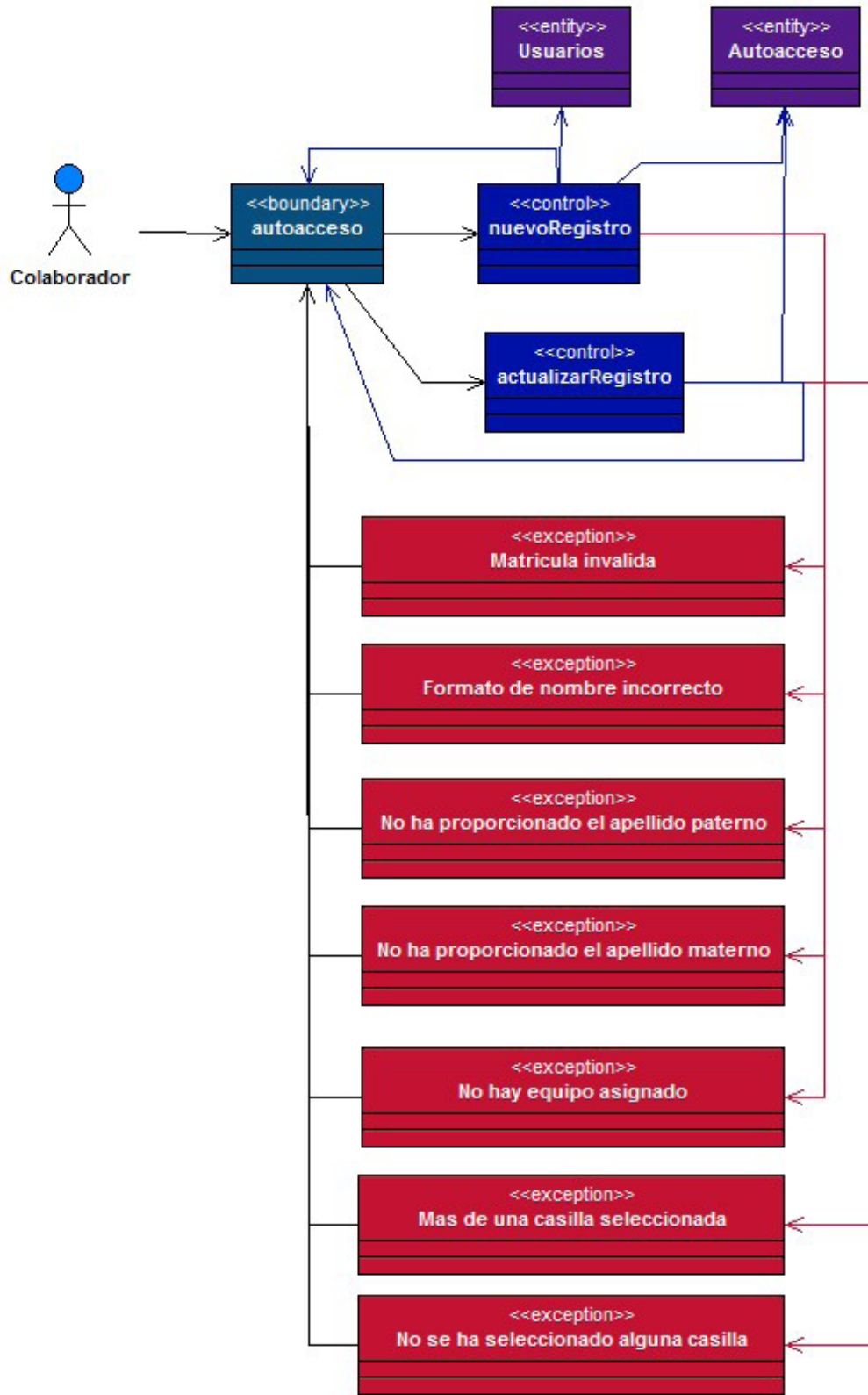


Figura 2.8.3. Diagrama de realización del caso de uso administrar acceso

Diagrama de realización de casos de uso para SALC-11 Registrar incidente

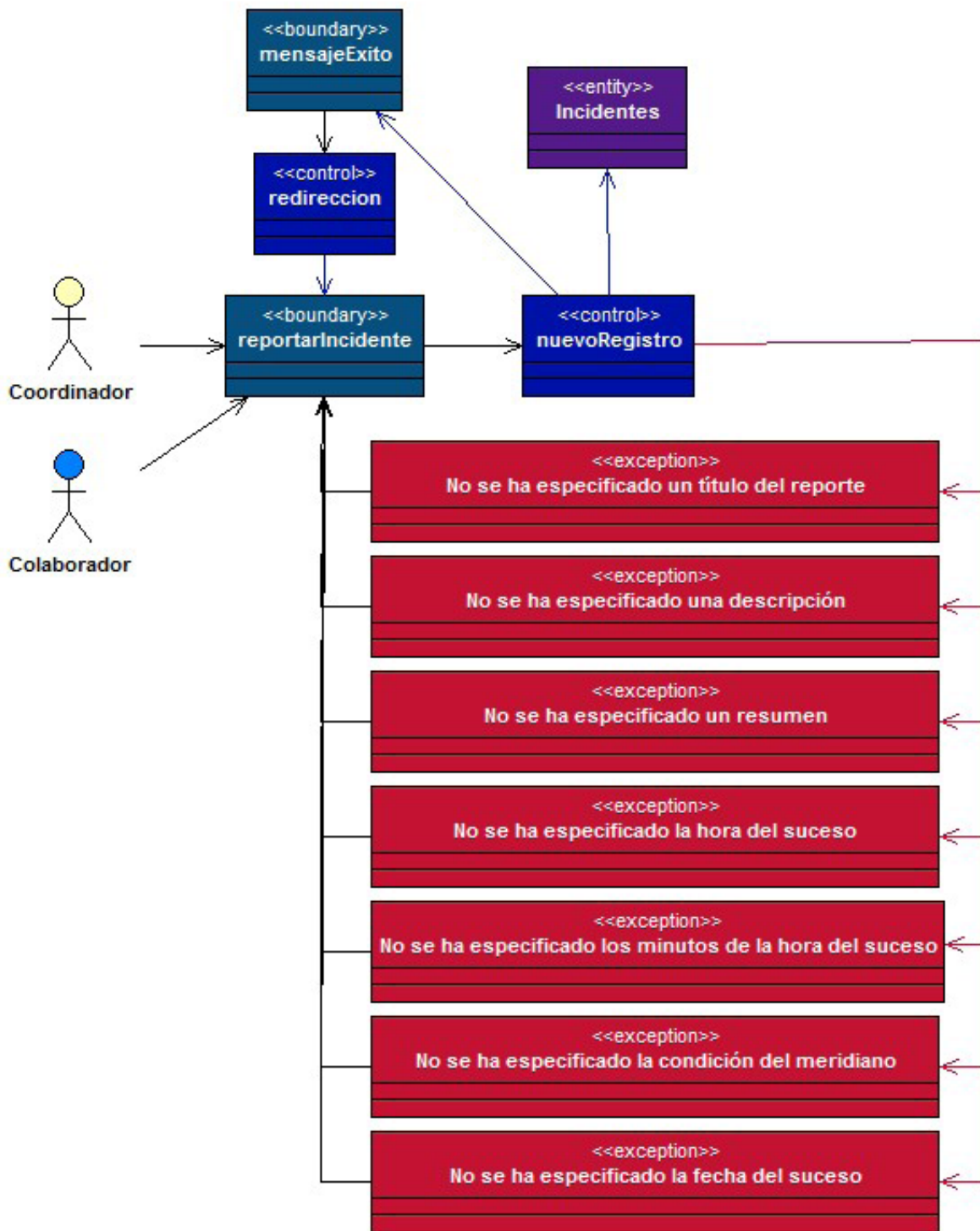


Figura 2.8.4. Diagrama de realización del caso de uso registrar incidente

Diagrama de realización de casos de uso para SALC-13 Reportar hardware

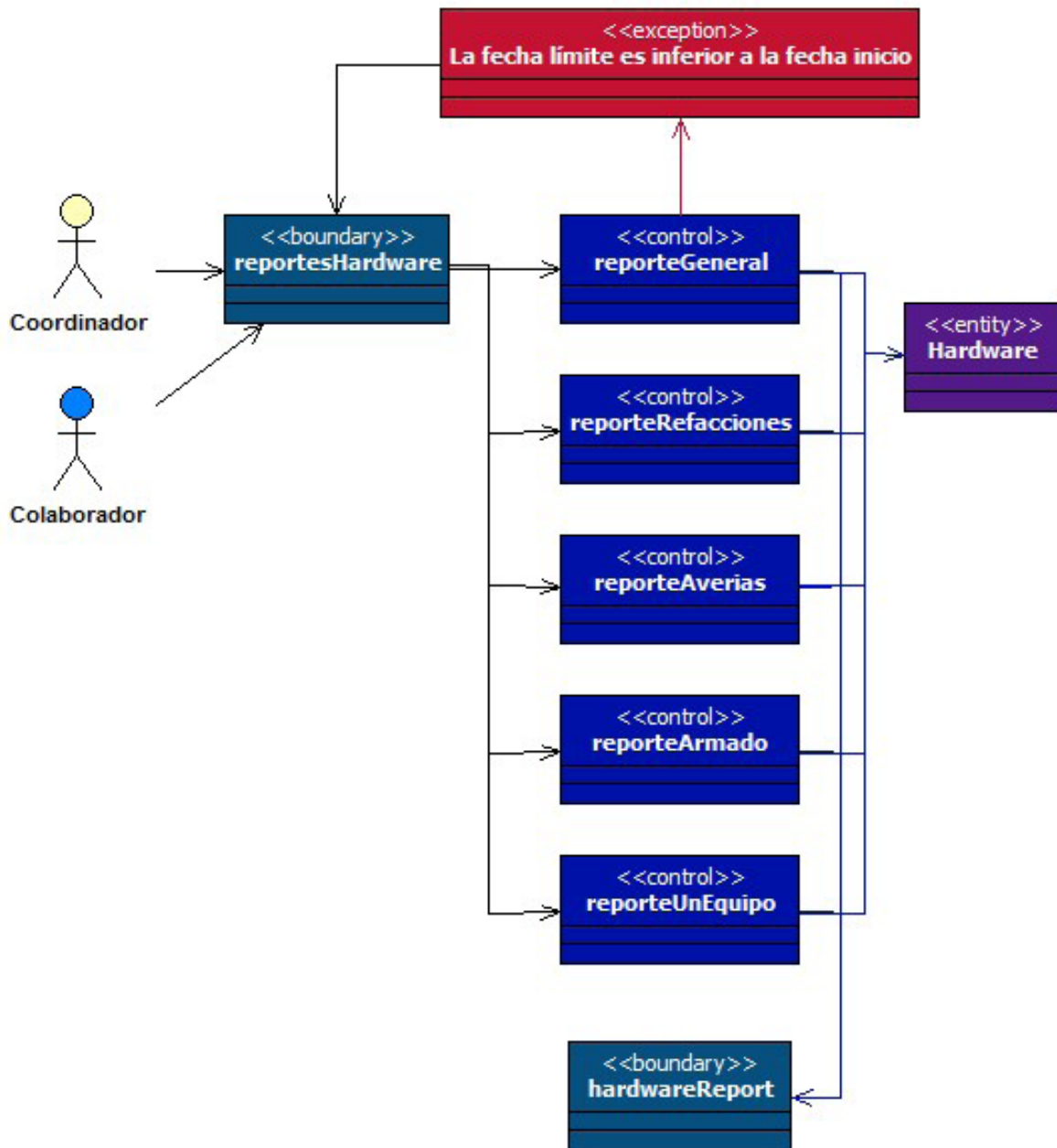


Figura 2.8.5. Diagrama de realización del caso de uso reportar hardware

Diagrama de realización de casos de uso para SALC-18 Registrar usuario

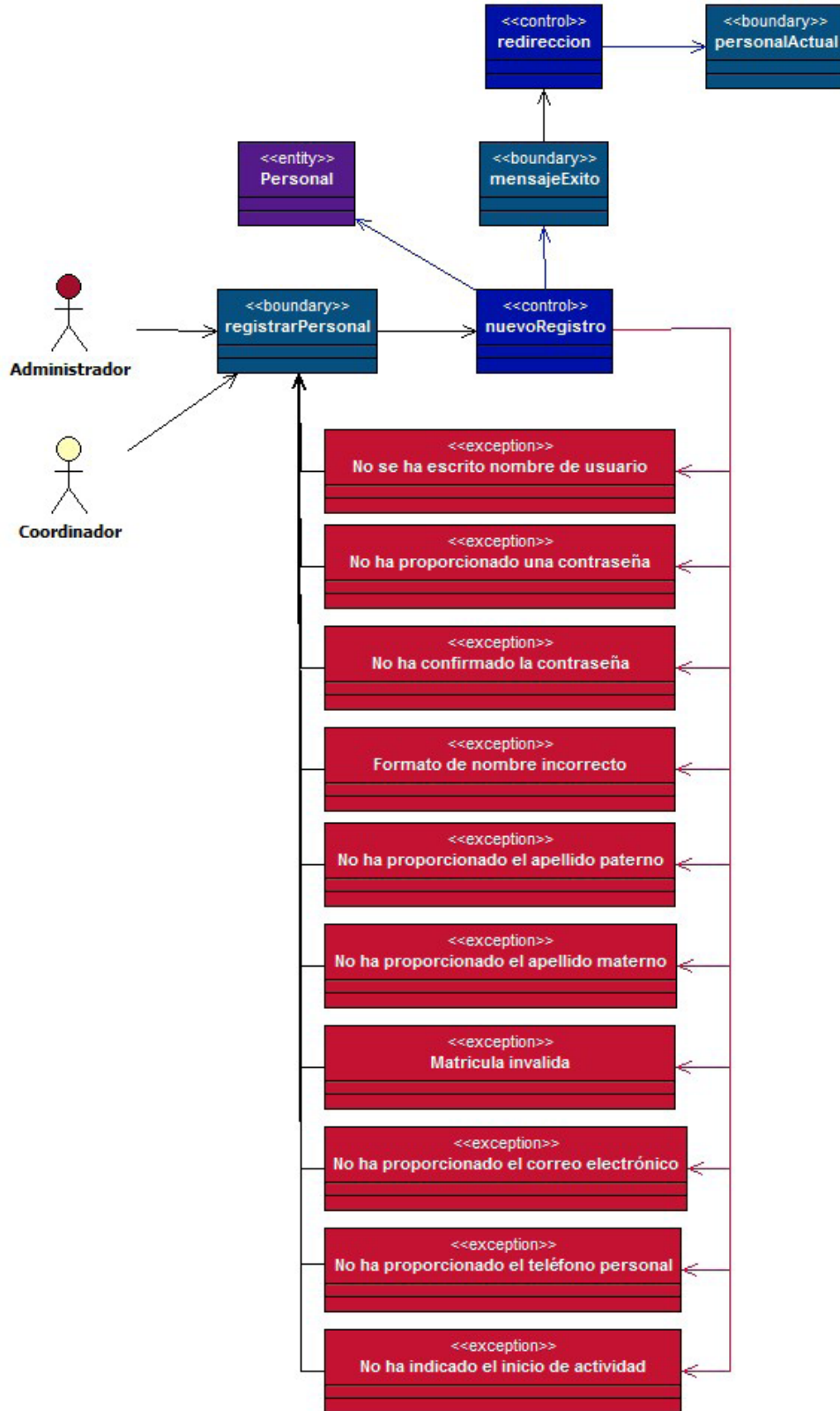


Figura 2.8.6. Diagrama de realización del caso de uso registrar usuario

Diagrama de realización de casos de uso para SALC-25 Respalda registros

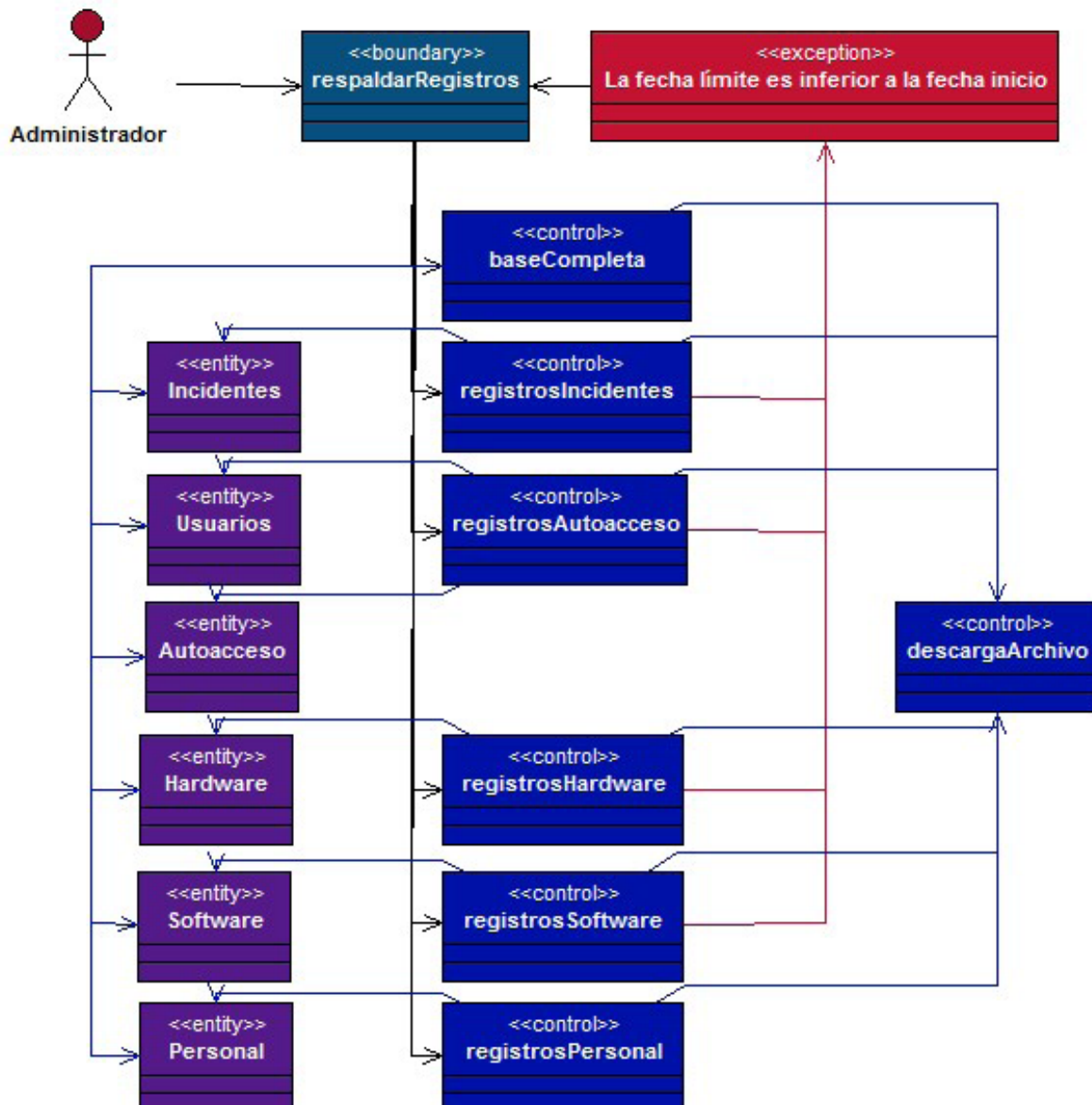


Figura 2.8.7. Diagrama de realización del caso de uso respaldar registros

Diagrama de realización de casos de uso para SALC-26 Administrar edificio

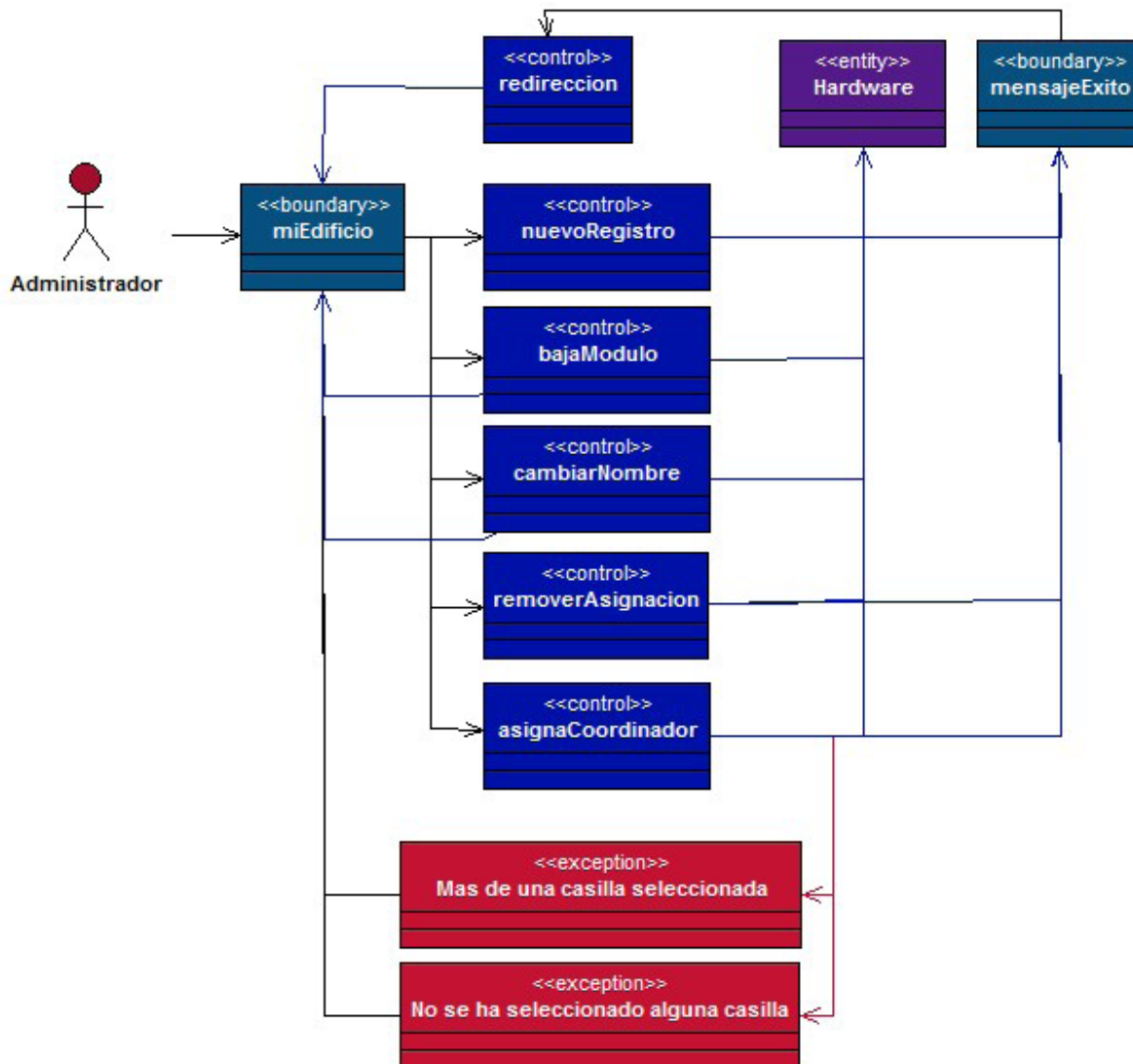


Figura 2.8.8. Diagrama de realización del caso de uso administrar edificio

Diagrama de realización de casos de uso para SALC-29 Instalar sistema

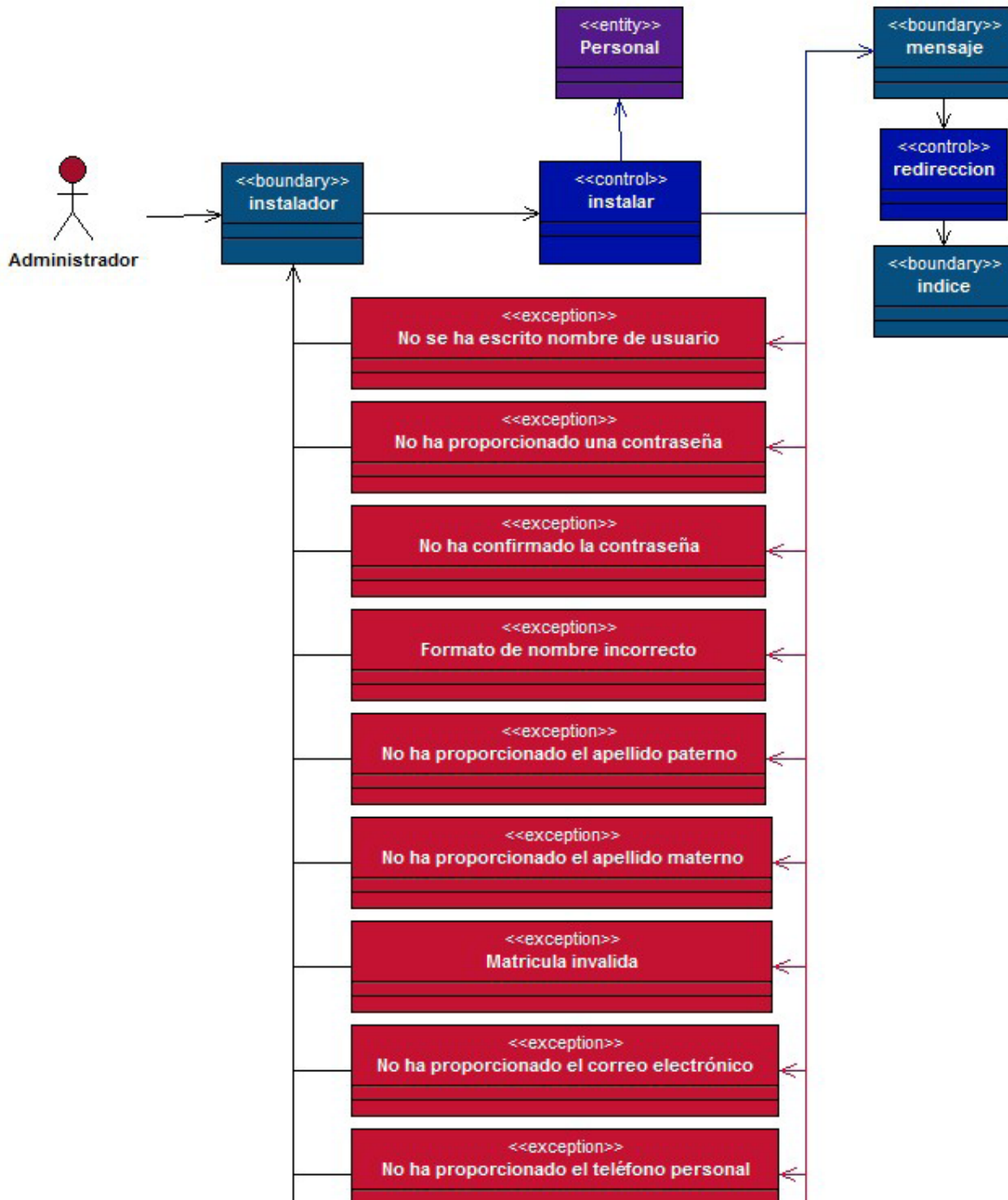


Figura 2.8.9. Diagrama de realización del caso de uso instalar sistema

### ***2.8.2 Diagramas de colaboración***

Continuando con el modelo de análisis de la metodología del proceso unificado para el desarrollo de software, se presentan los diagramas de colaboración. Estos diagramas descienden un nivel mas acercandose a las rutinas de programación.

En estos diagramas se visualizan las clases y como interactuan sus objetos, es decir ya puede observarse el paso de mensajes, esencial en la programación orientada a objetos.

Se ha seleccionado de cada caso de uso citado anteriormente, describir el flujo básico el cual se puede apreciar en las especificaciones de casos de uso y si bien el flujo básico es muy corto, se describirá la función más importante de cada caso de uso. También en algunos casos la cantidad de flujos alternos que un caso de uso puede tener es muy grande, así que se expondrá solo un flujo alternativo. Por último no todas las excepciones ocurren en un mismo flujo, estas pueden tener varias combinaciones, se expondrán las más relevantes y las que tendrán más ocurrencia.

Diagramas para SALC-01 Inventariar software

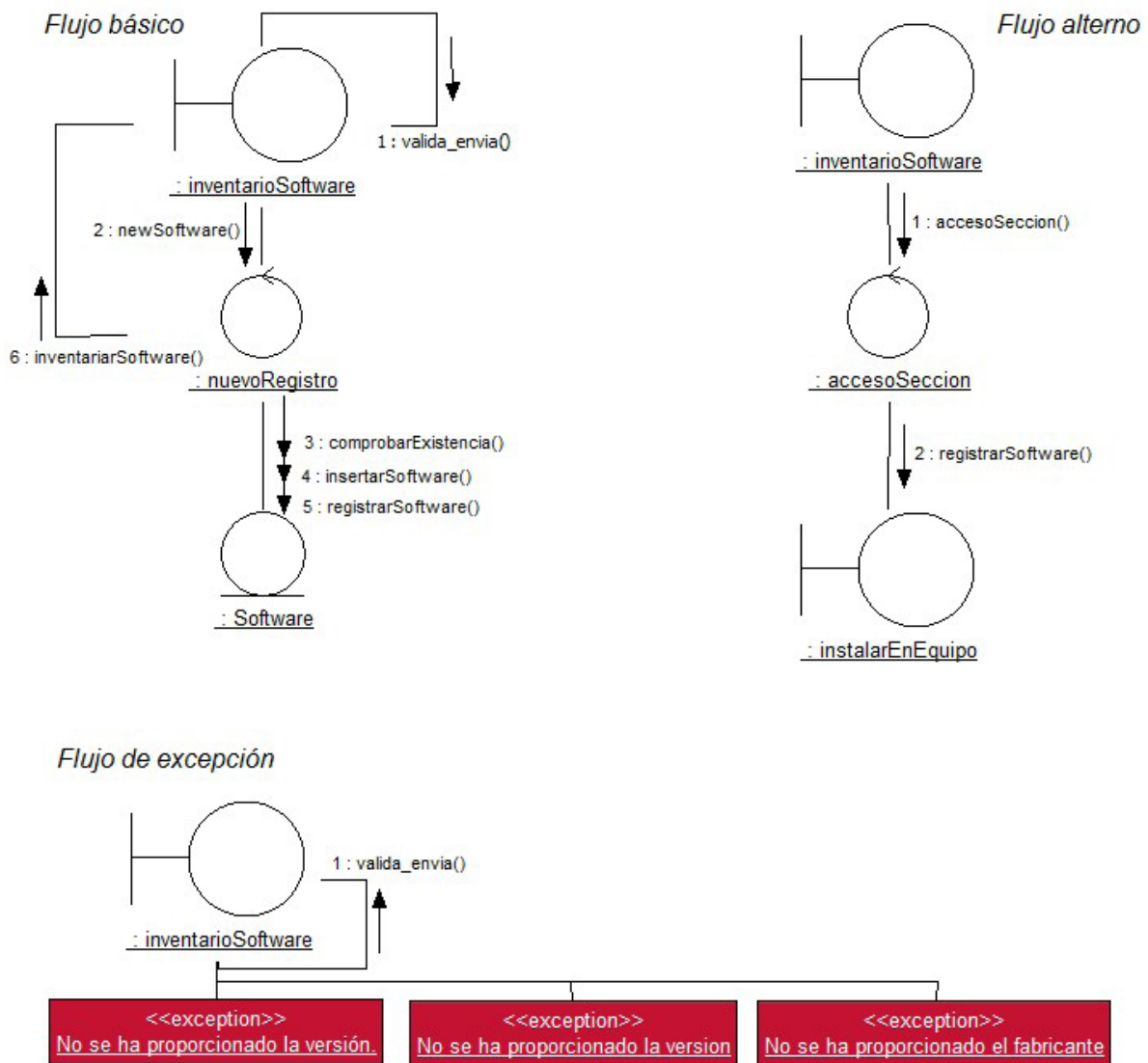


Figura 2.8.10. Diagramas de colaboración para el caso de uso inventariar software

Diagramas para SALC-04 Inventariar hardware

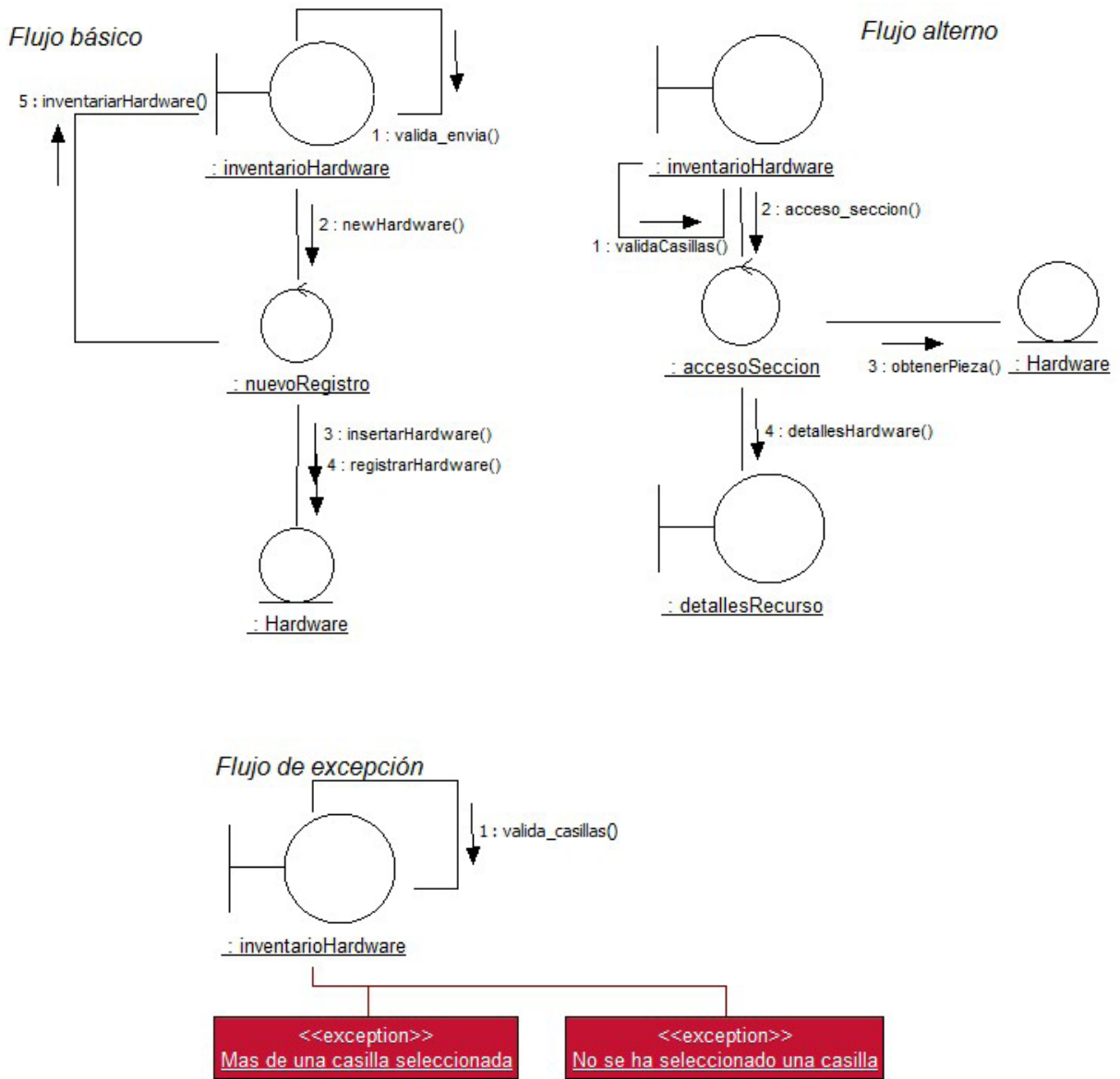


Figura 2.8.11. Diagramas de colaboración para el caso de uso inventariar hardware

Diagramas para SALC-09 Administrar acceso

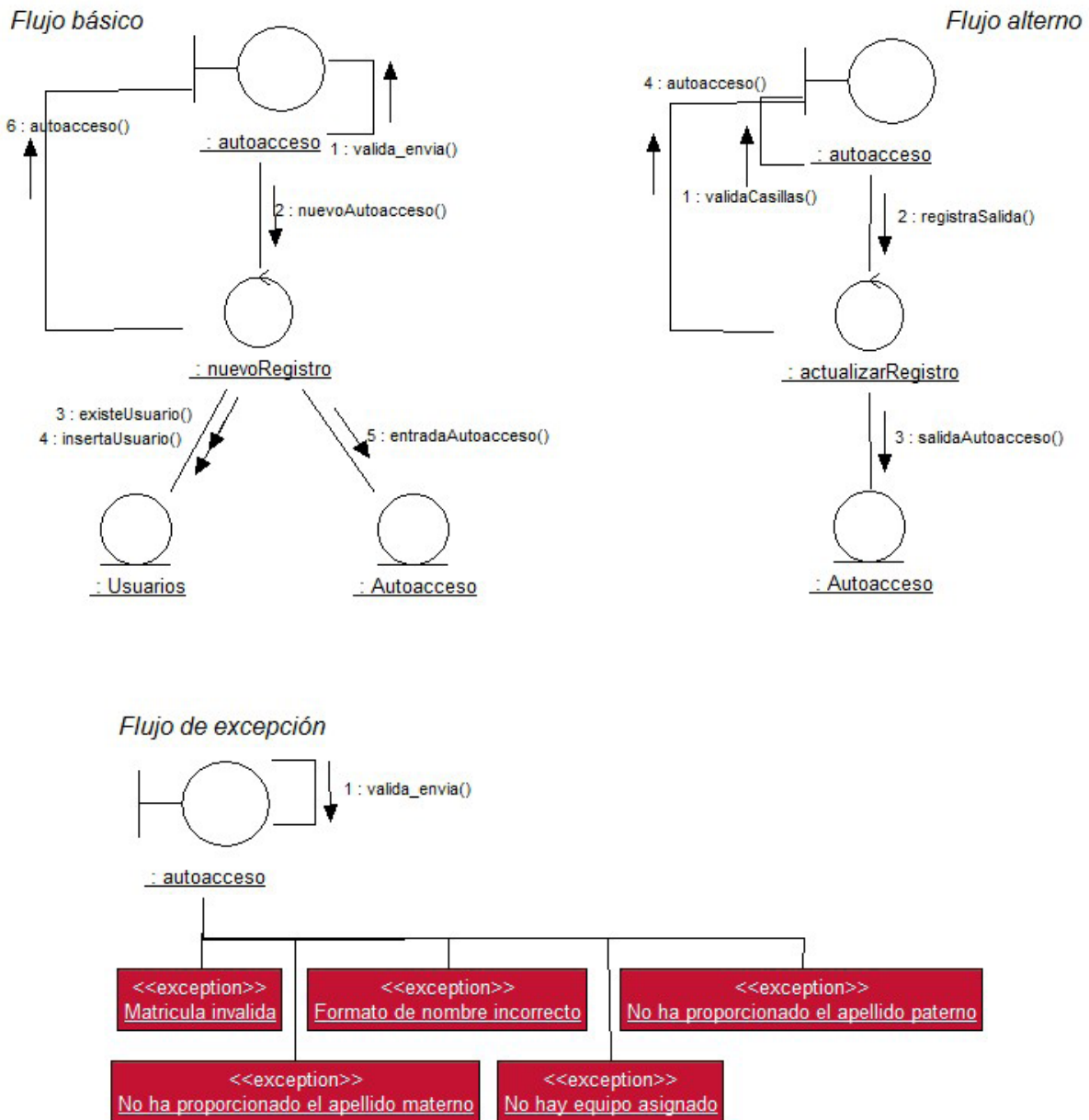
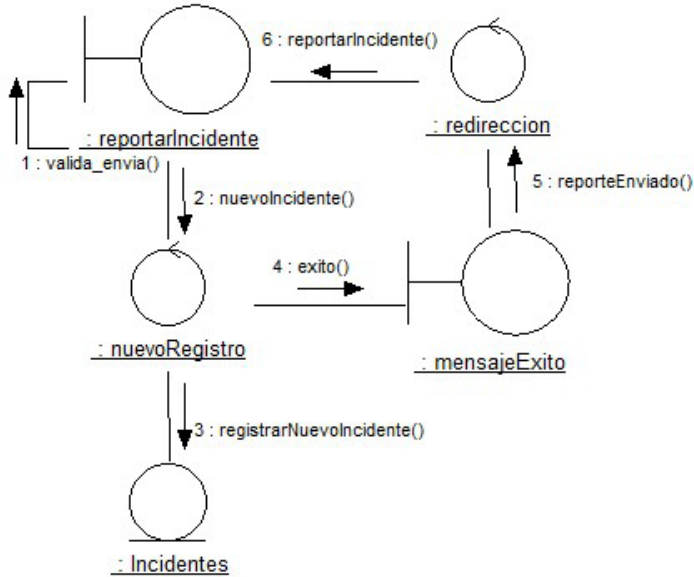


Figura 2.8.12. Diagramas de colaboración para el caso de uso administrar acceso

## Diagramas para SALC-11 Registrar incidente

### Flujo básico



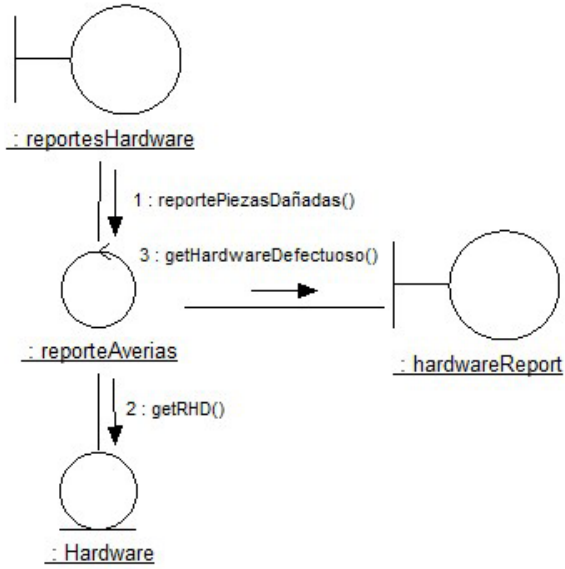
### Flujo de excepción



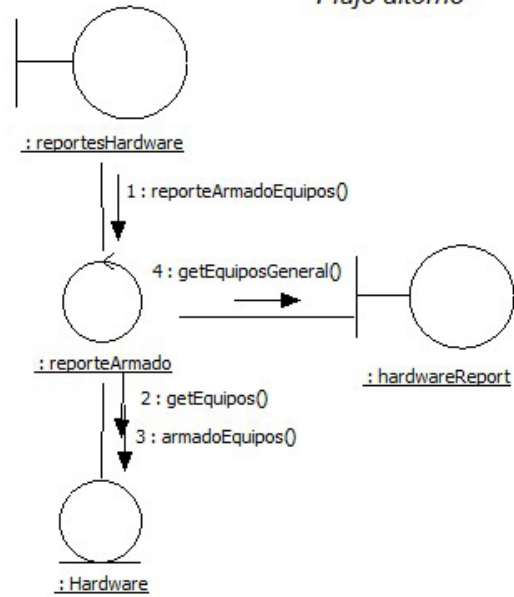
Figura 2.8.13. Diagramas de colaboración para el caso de uso registrar incidente

Diagramas para SALC-13 Reportar hardware

Flujo básico



Flujo alterno

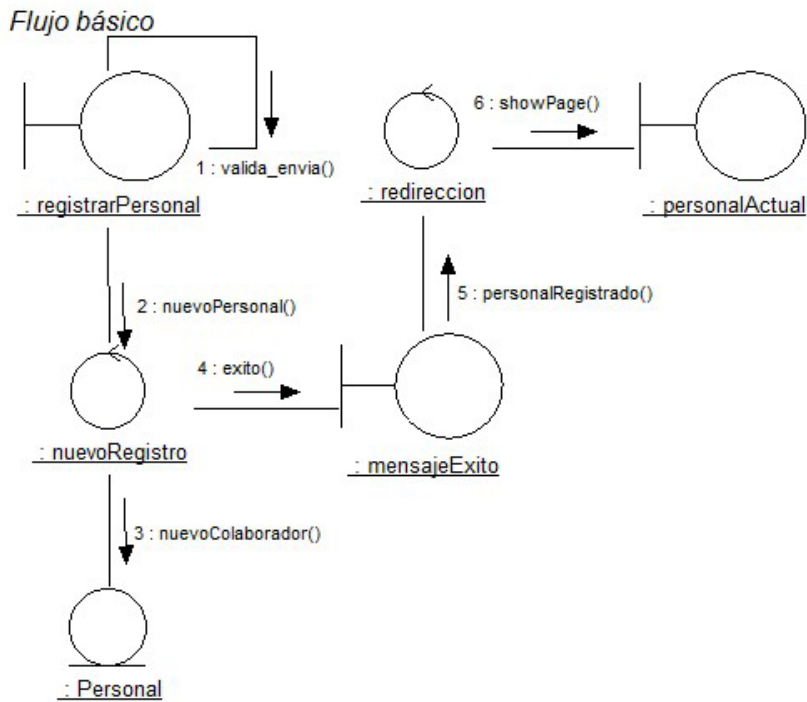


Flujo de excepción



Figura 2.8.14. Diagramas de colaboración para el caso de uso reportar hardware

Diagramas para SALC-18 Registrar usuario



*Flujo de excepción*

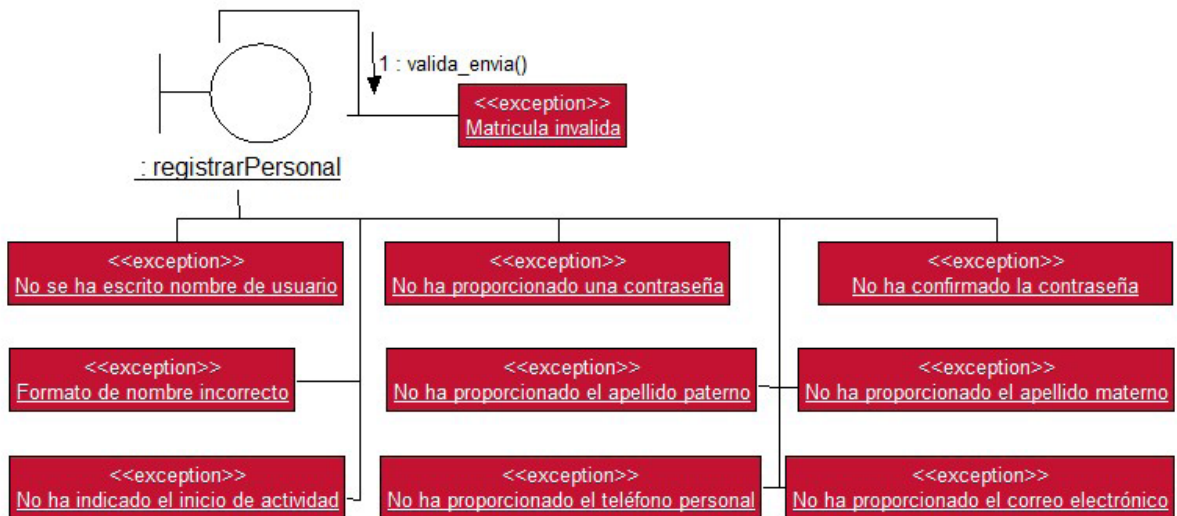
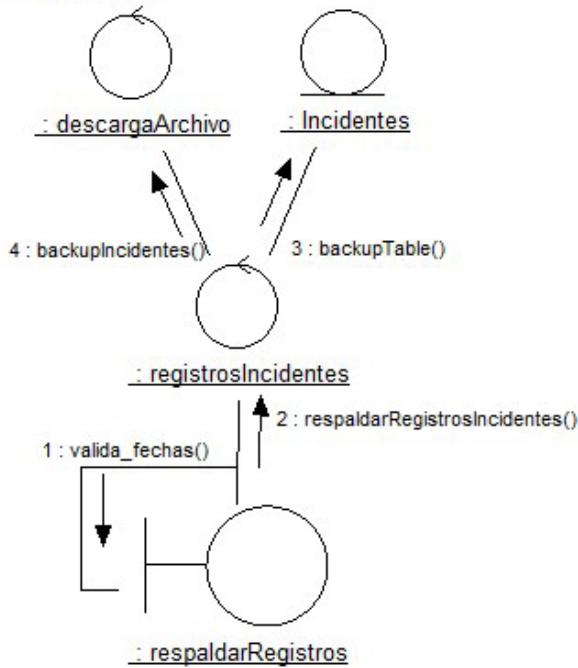


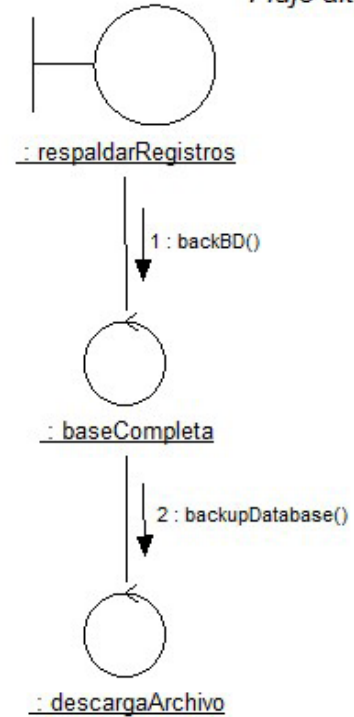
Figura 2.8.15. Diagramas de colaboración para el caso de uso registrar usuario

Diagramas para SALC-25 Respaldo registros

Flujo básico



Flujo alterno



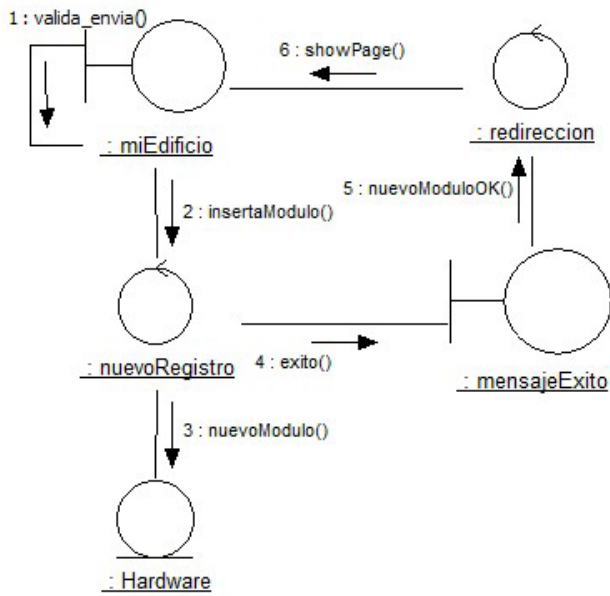
Flujo de excepción



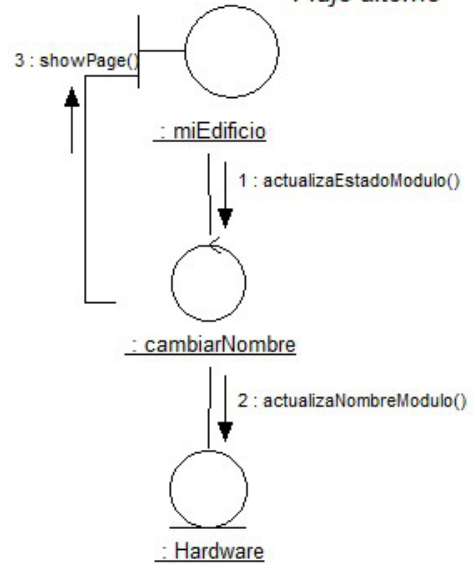
Figura 2.8.16. Diagramas de colaboración para el caso de uso respaldar registros

Diagramas para SALC-26 Administrar edificio

Flujo básico



Flujo alterno

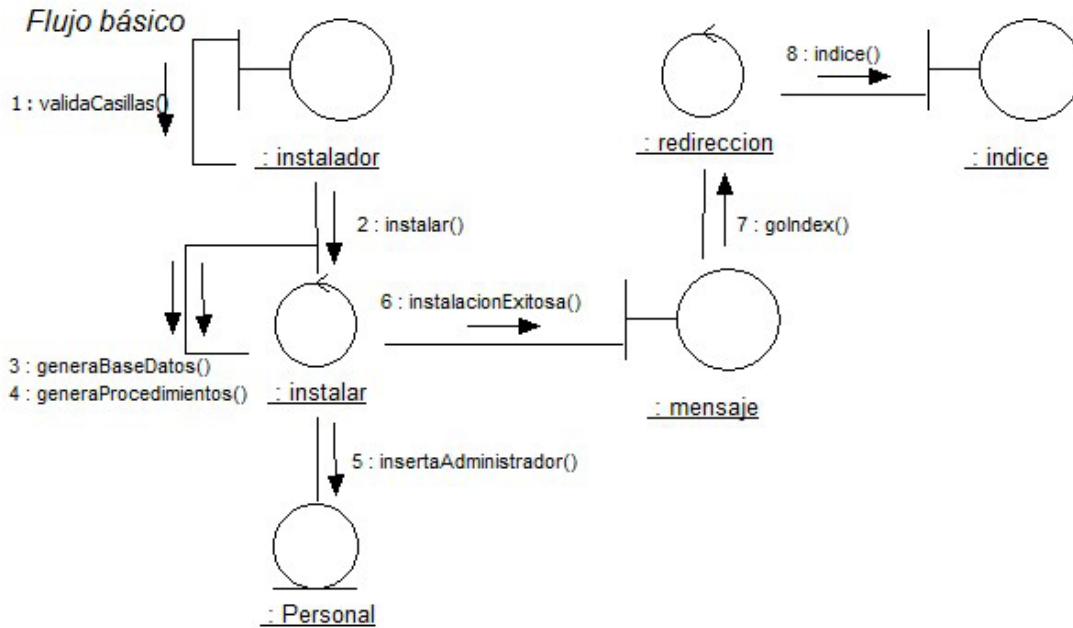


Flujo de excepción



Figura 2.8.17. Diagramas de colaboración para el caso de uso administrar edificio

## Diagramas para SALC-29 Instalar sistema



## Flujo de excepción

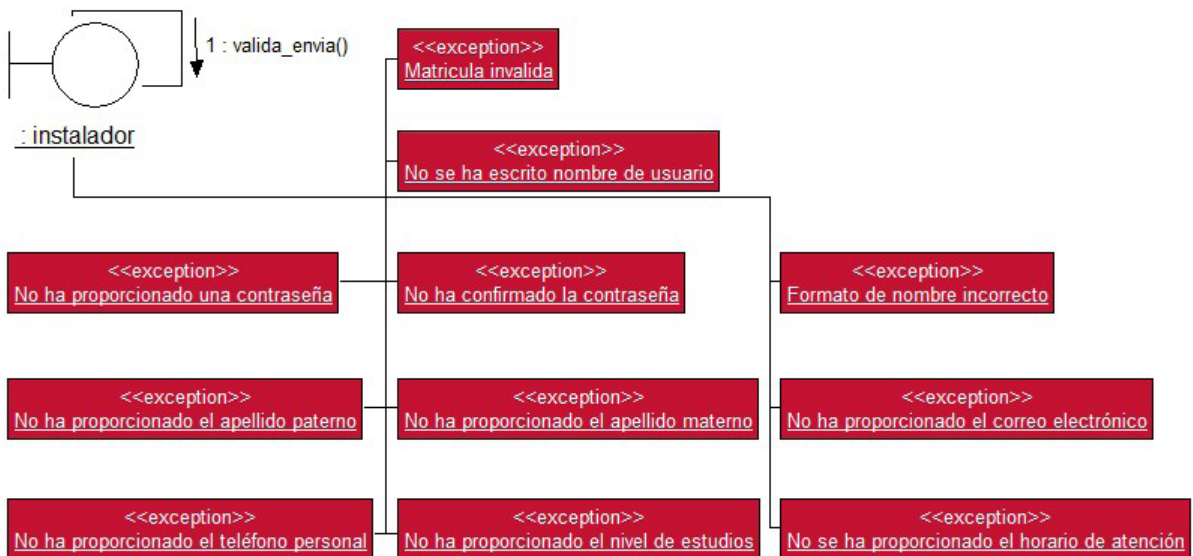


Figura 2.8.18. Diagramas de colaboración para el caso de uso instalar sistema

### 2.8.3 Diagrama de clases de diseño

El modelo de diseño propuesto en el proceso unificado de desarrollo de software plantea la completa definición de cada clase de análisis. El sistema a desarrollar posee 60 clases de análisis, al poseer un alto número de clases, se decidió estandarizar el comportamiento de las clases de análisis de tipo interfaz, estableciendo que todas estas posean los mismos métodos para mostrar el contenido en el explorador, así como los mismos atributos para la recolección de datos, página de contenido, y módulo que lo solicita. El diagrama de estas ahora clases de diseño puede observarse en la figura 2.8.19.

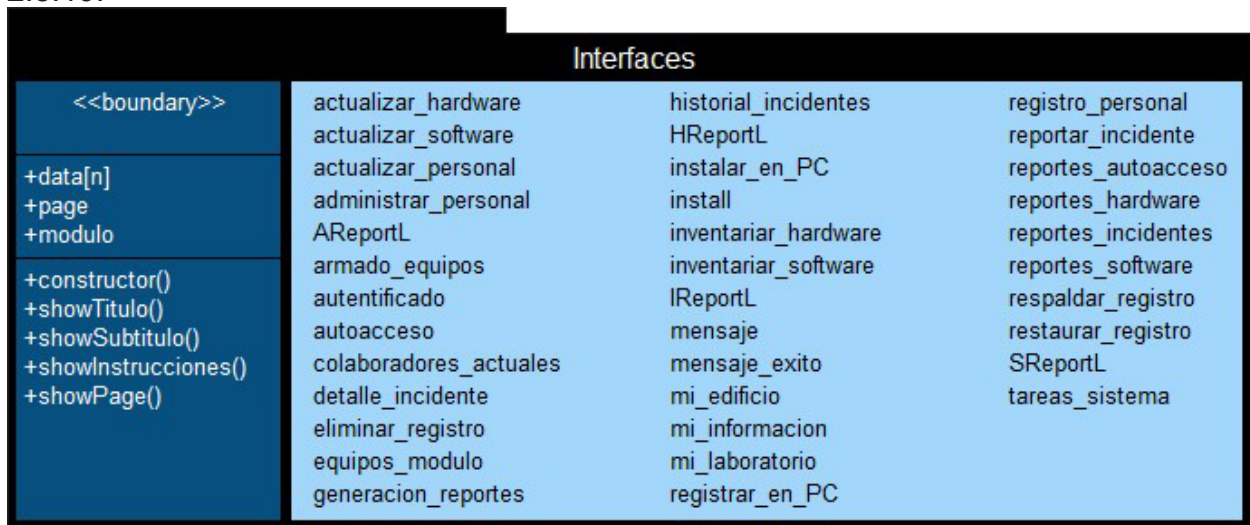


Figura 2.8.19. Clases de diseño para las interfaces del sistema

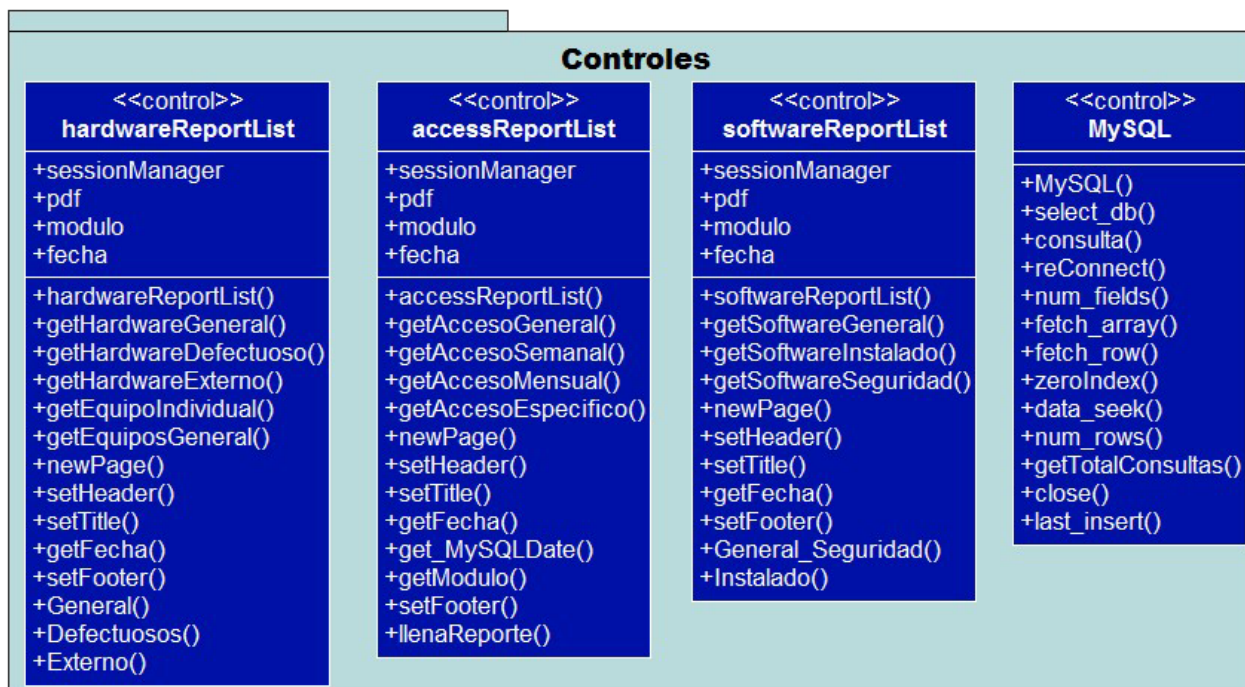


Figura 2.8.20. Clases de diseño para los controles del sistema

Las clases de analisis tipo control no pueden ser estandarizadas por sus distintas tareas, las figuras 2.8.20 y 2.8.21 muestran las clases de tipo control abstraídas de las clases de diseño.



Figura 2.8.21. Clases de diseño para los controles del sistema

Y por ultimo las clases de analisis tipo entidad, siendo estas el conjunto de operaciones que una entidad en la base de datos estrictamente permitirá realizar solo las operaciones sugeridas por los métodos. La figura 2.8.22 expone las clases de diseño para las entidades del sistema.

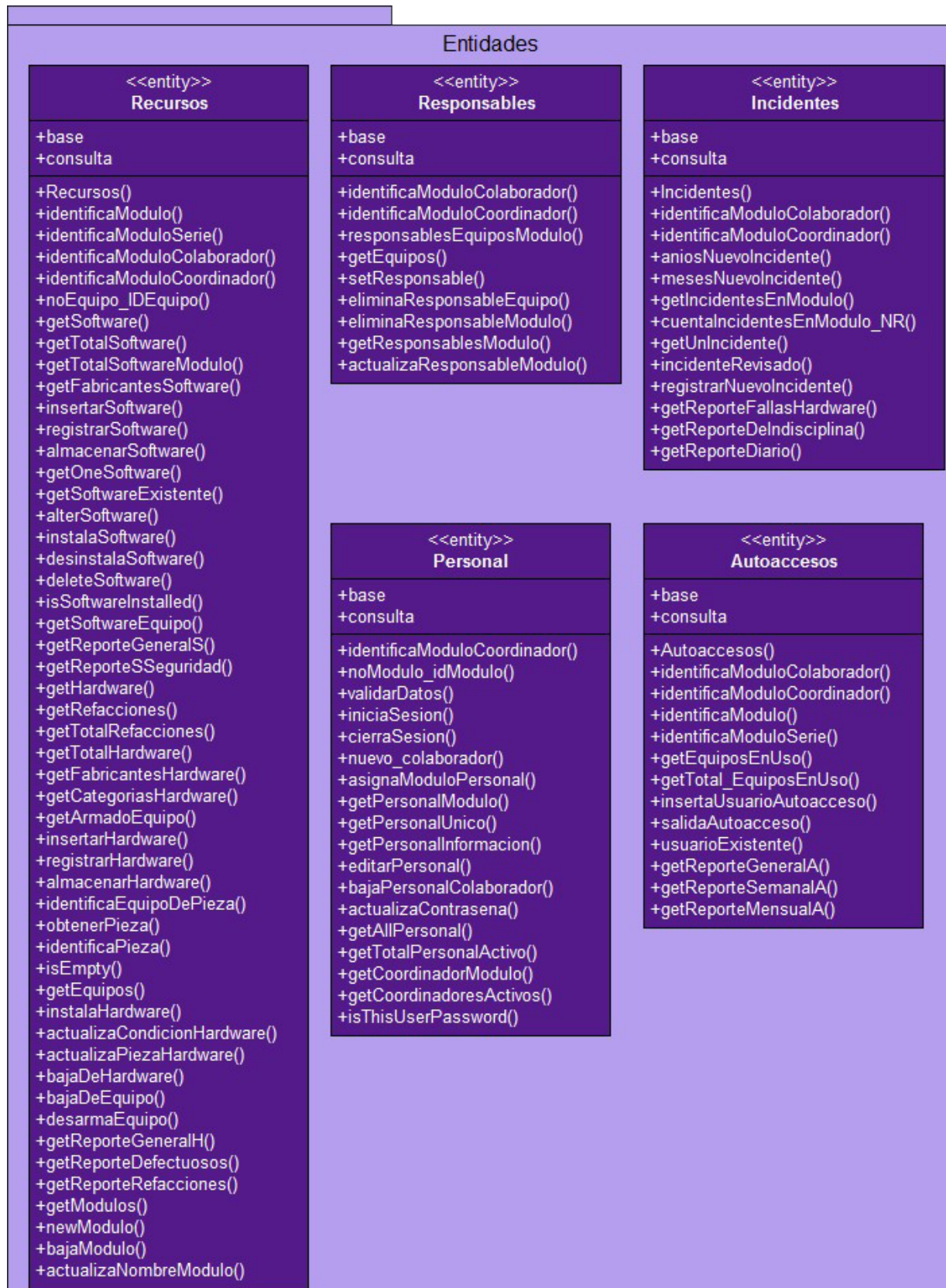


Figura 2.8.22. Clases de diseño para las entidades del sistema

## 2.8.4 Diagramas de secuencia

Los diagramas de secuencia explican paso a paso como se comunicarán los objetos, y que métodos serán utilizados para esta comunicación, además puede observarse el tiempo de control de cada objeto. Son muy parecidos a los diagramas de colaboración, pero con dos diferencias, la primera es que estos al pertenecer al modelo de diseño poseen un nivel más alto de abstracción y las clases ya se han definido con sus atributos y métodos principales.

De la misma manera que en los diagramas de colaboración, se mostrarán las funciones más utilizadas de los casos de uso más relevantes del sistema.

### Diagramas de secuencia para SALC-01 Inventariar Software

#### Flujo básico

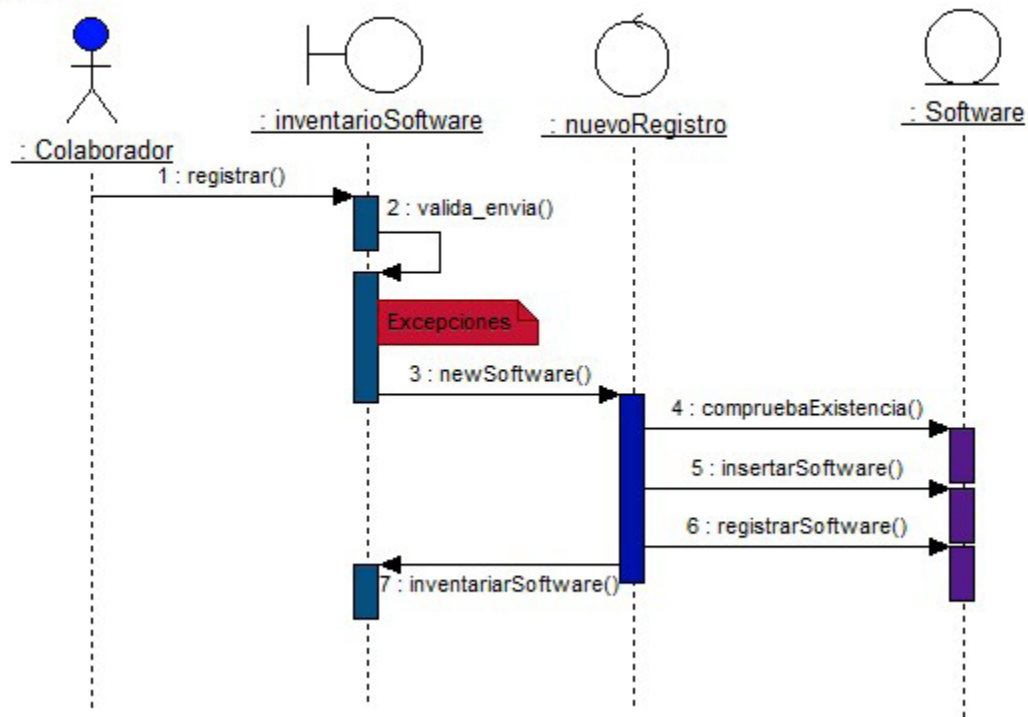


Figura 2.8.4.1. Diagrama de secuencia de la función principal del caso de uso inventariar software.

*Flujo alterno*

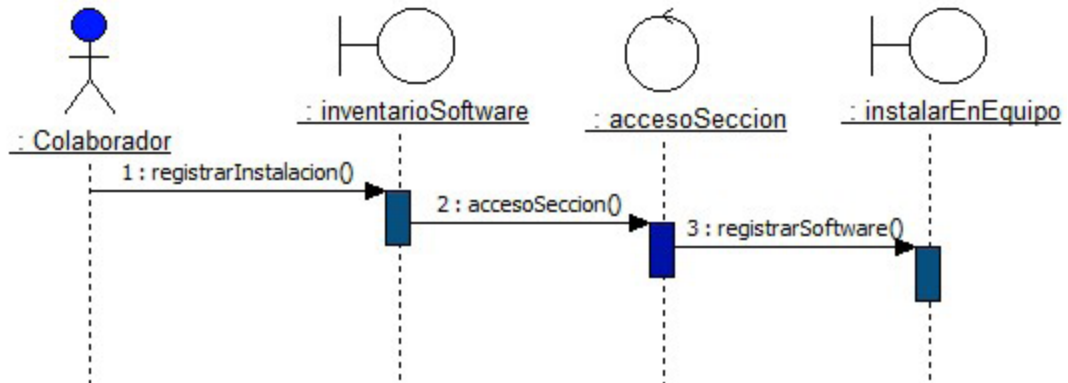


Figura 2.8.4.2. Diagrama de secuencia de la función de registro de instalación del caso de uso inventariar software.

*Flujo de excepción*

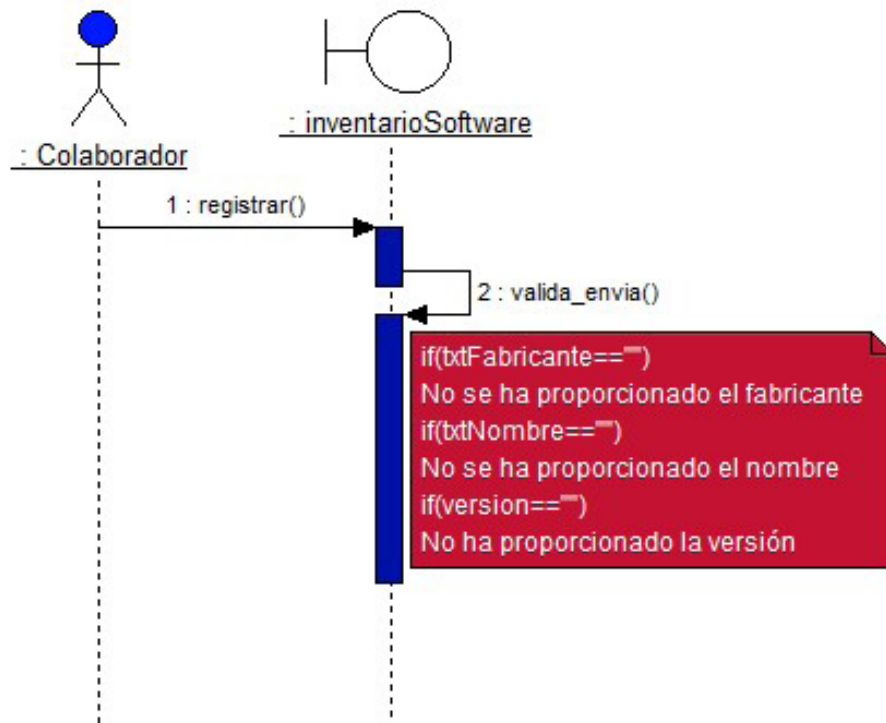


Figura 2.8.4.3. Diagrama de secuencia de las excepciones que pueden ocurrir del caso de uso inventariar software.

Diagramas de secuencia para el caso de uso SALC-04 Inventariar hardware

Flujo básico

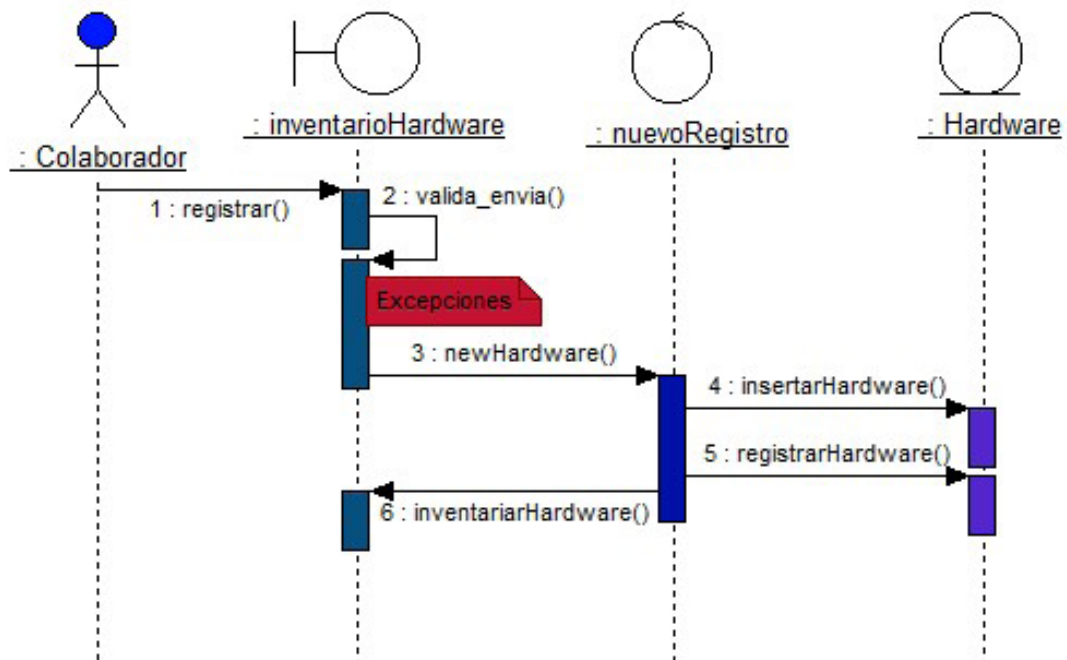


Figura 2.8.4.4. Diagrama de secuencia de la función principal del caso de uso inventariar hardware.

Flujo alternativo

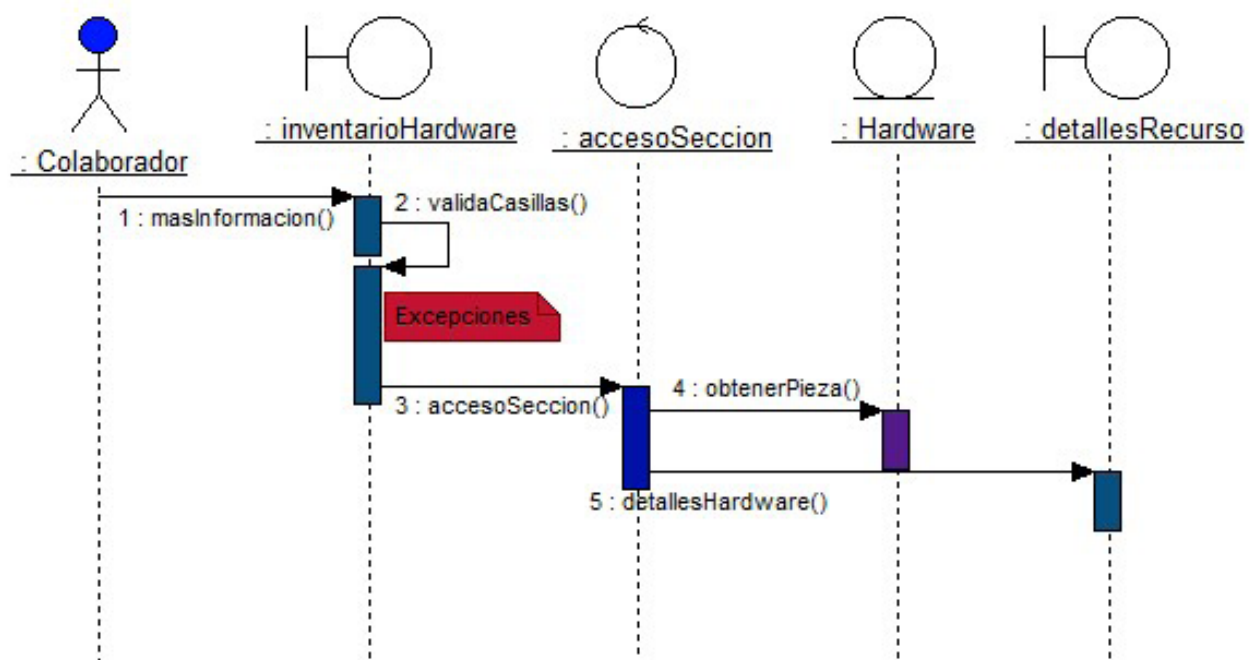


Figura 2.8.4.5. Diagrama de secuencia de la función de información detallada del caso de uso inventariar hardware.

*Flujo de excepción*

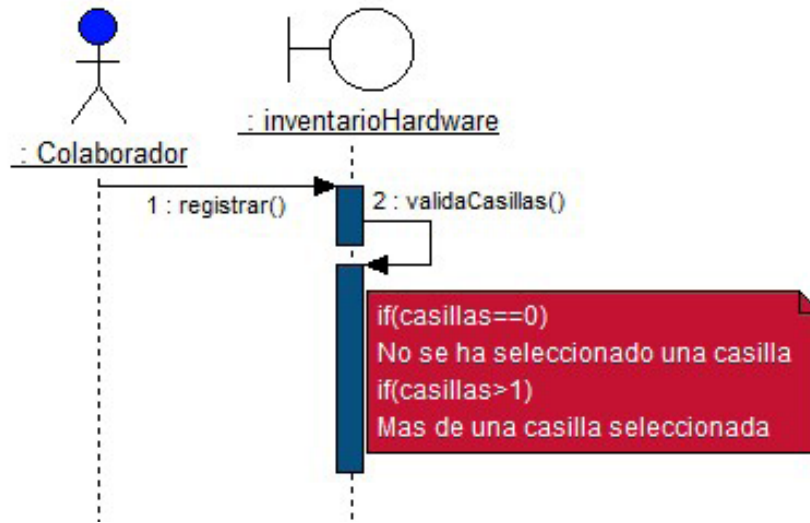


Figura 2.8.4.6. Diagrama de secuencia de algunas excepciones que pueden ocurrir del caso de uso inventariar hardware.

*Diagramas de secuencia para el caso de uso SALC-09 Administrar acceso*

*Flujo básico*

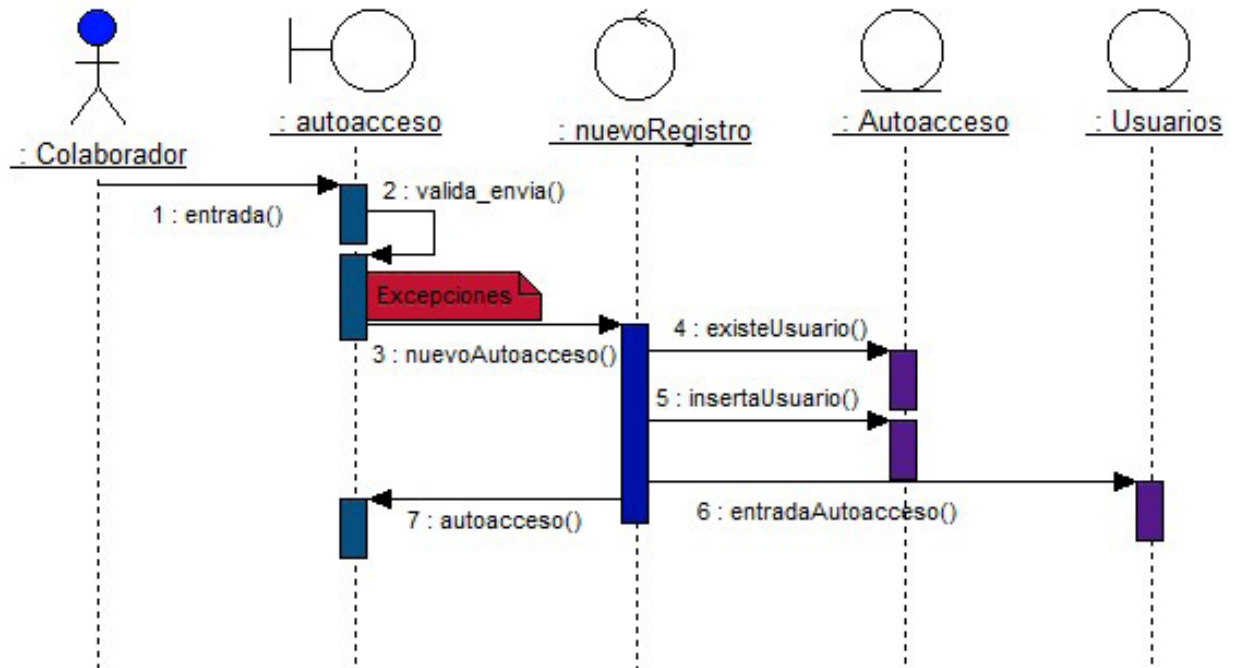


Figura 2.8.4.7. Diagrama de secuencia de la función principal del caso de uso administrar acceso.

### Flujo alterno

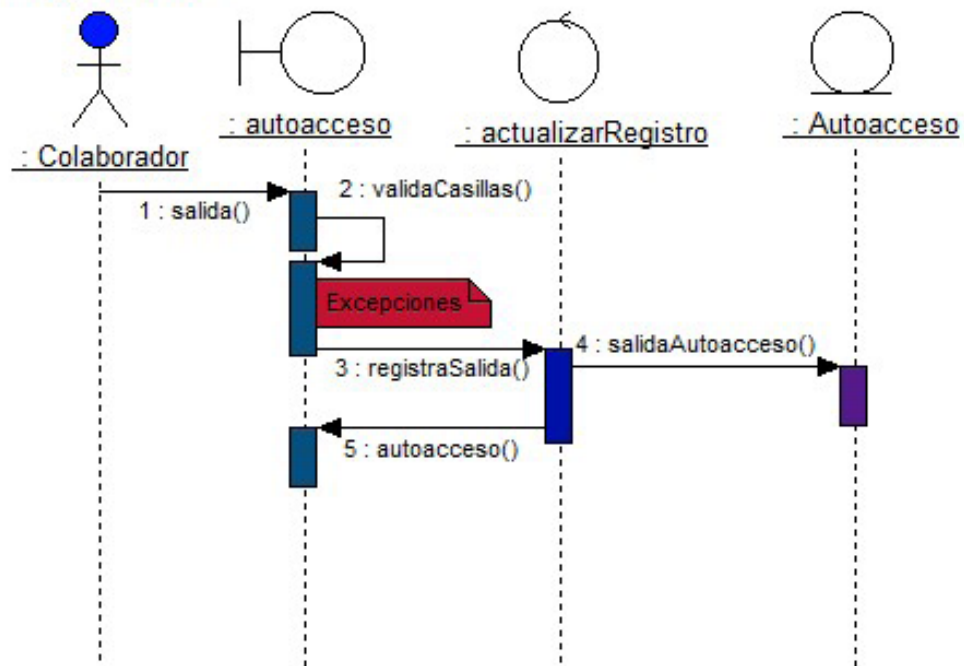


Figura 2.8.4.8. Diagrama de secuencia de la función de salida del módulo del caso de uso administrar acceso.

### Flujo de excepción

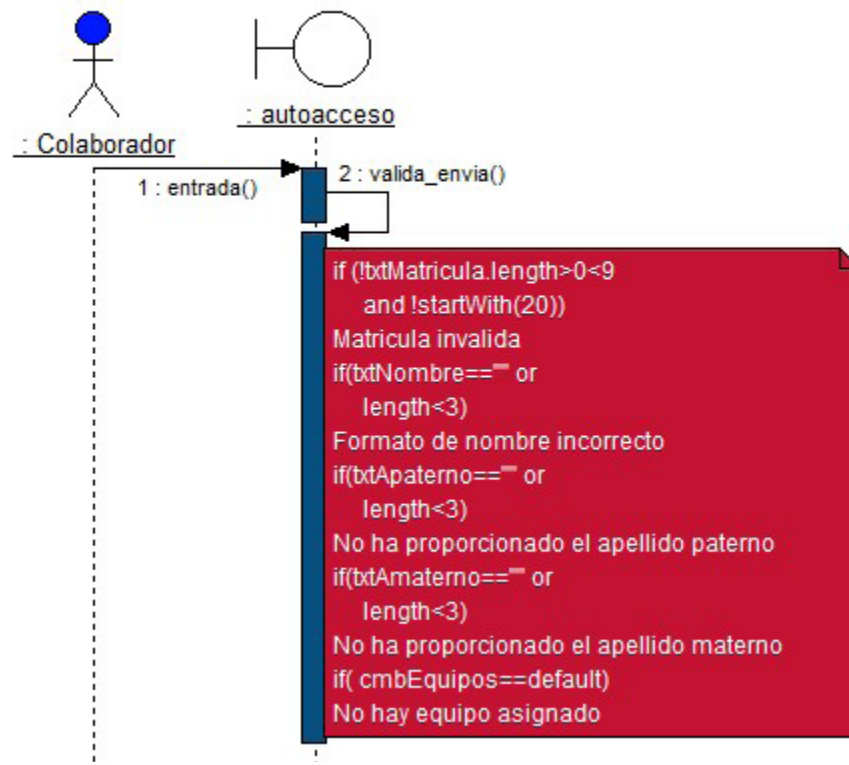


Figura 2.8.4.9. Diagrama de secuencia de las excepciones que pueden ocurrir al registrar una entrada al módulo.

Diagramas de secuencia para el caso de uso SALC-11 Registrar incidente

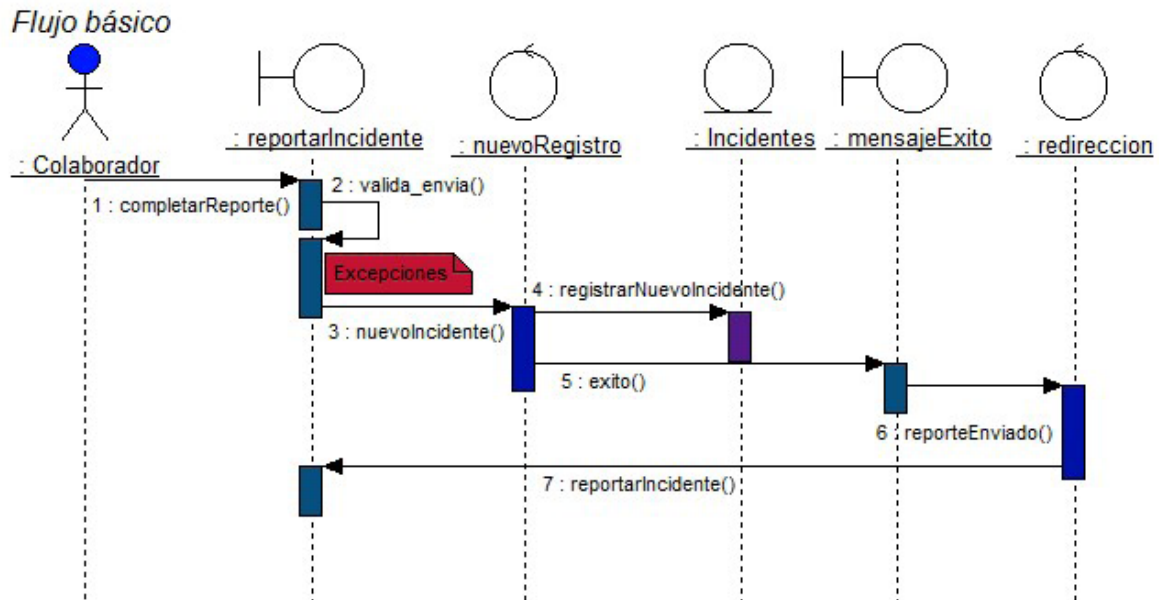


Figura 2.8.4.10. Diagrama de secuencia de la función principal del caso de uso registrar incidente.

*Flujo de excepción*

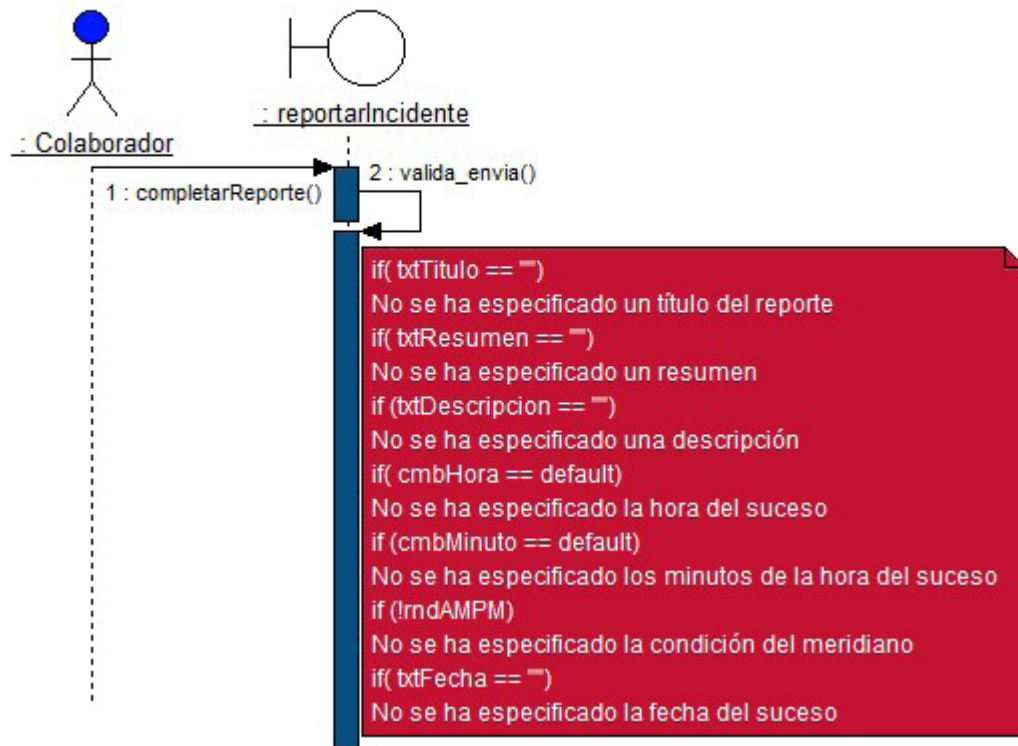


Figura 2.8.4.11. Diagrama de secuencia de las excepciones que pueden ocurrir mientras se escribe un informe de incidente.

Diagramas de secuencia para el caso de uso SALC-13 Reportar hardware

Flujo básico

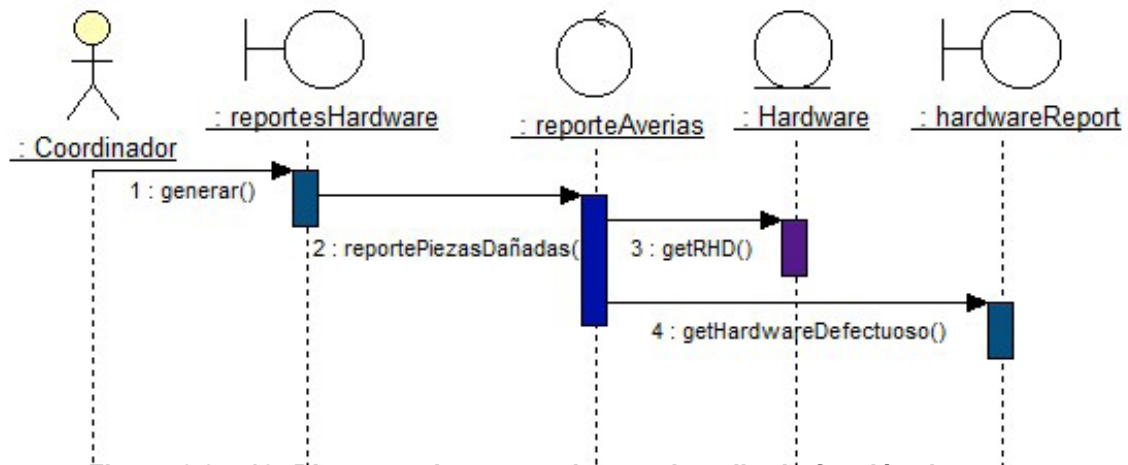


Figura 2.8.4.12. Diagrama de secuencia que describe la función de mas relevancia del caso de uso reportar hardware.

Flujo alternativo

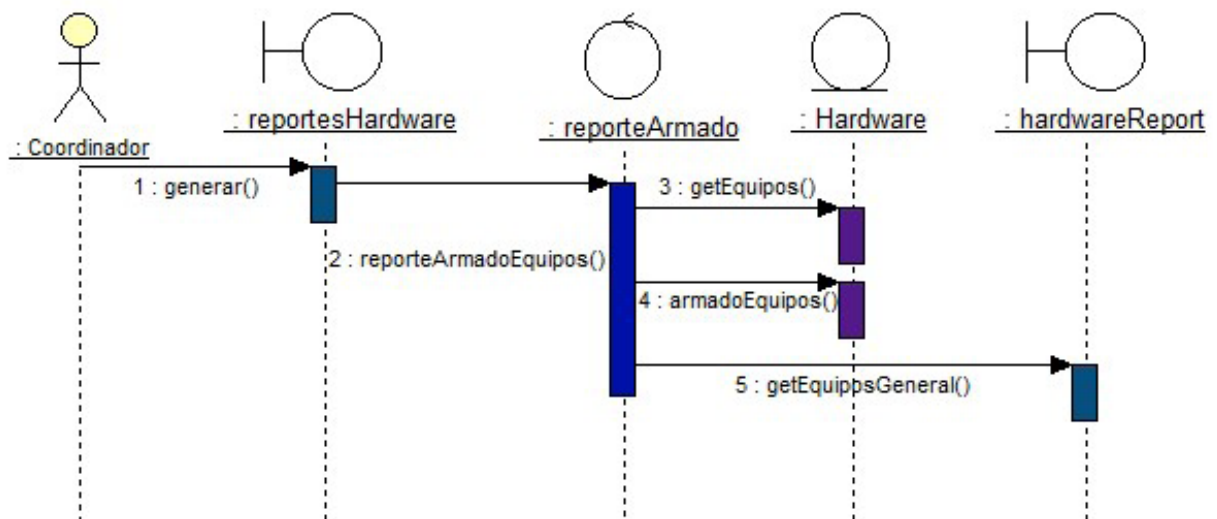


Figura 2.8.4.13. Diagrama de secuencia de una funcion alternativa del caso de uso reportar hardware.

*Flujo de excepción*

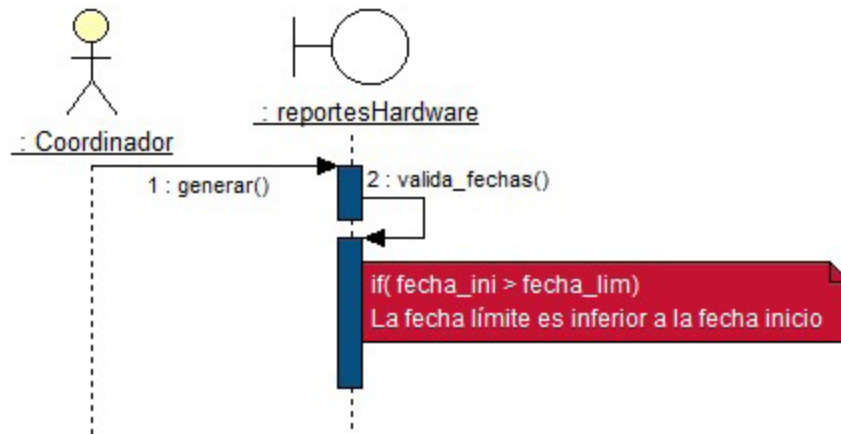


Figura 2.8.4.14. Diagrama de secuencia de el unico tipo de excepcion que causará el caso de uso reportar hardware.

*Diagramas de secuencia para el caso de uso SALC-18 Registrar usuario*

*Flujo básico*

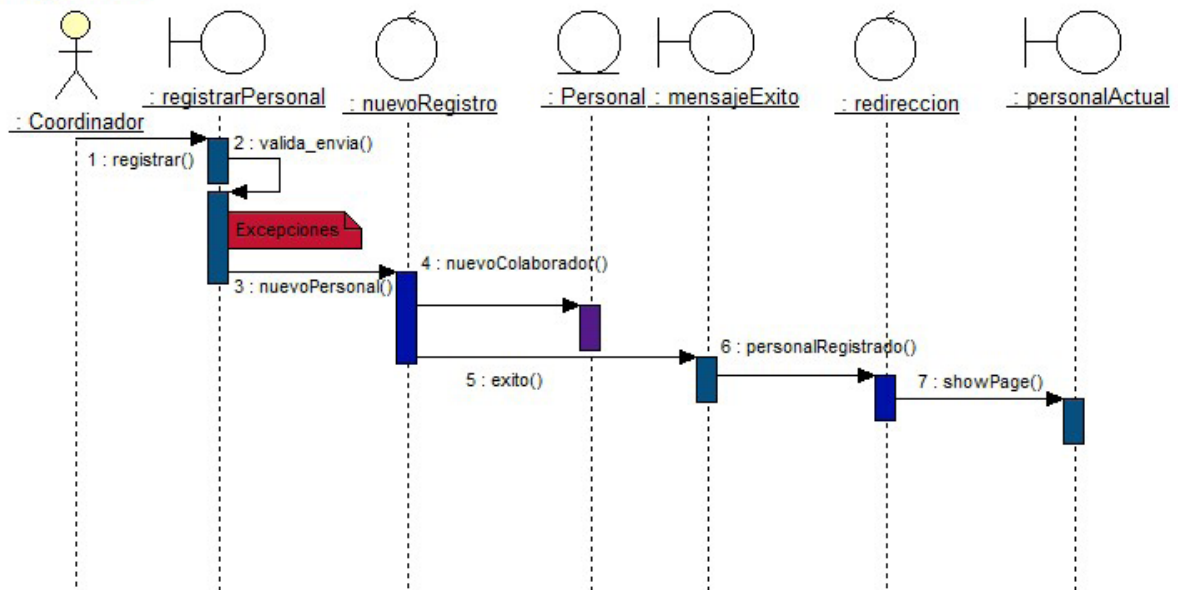


Figura 2.8.4.15. Diagrama de secuencia de la función principal del caso de uso registrar usuario

### Flujo de excepción

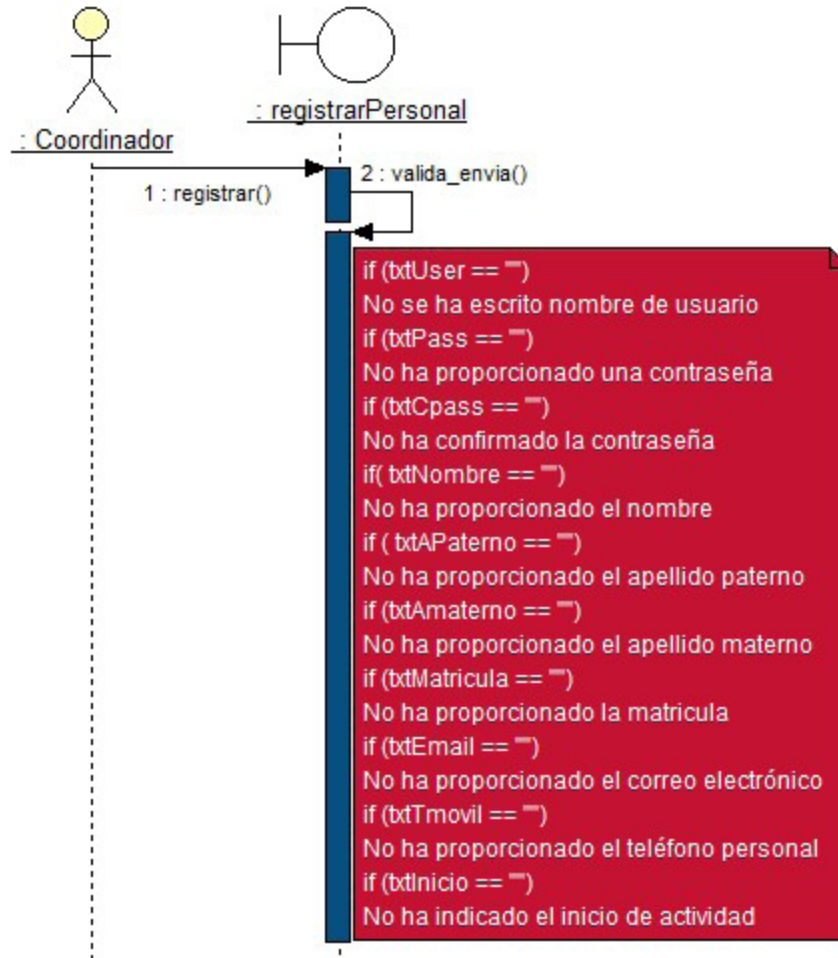


Figura 2.8.4.16. Diagrama de secuencia que expresa todas las excepciones que se pueden causar al registrar un nuevo usuario.

Diagramas de secuencia para el caso de uso SALC-25 Respaldar registros

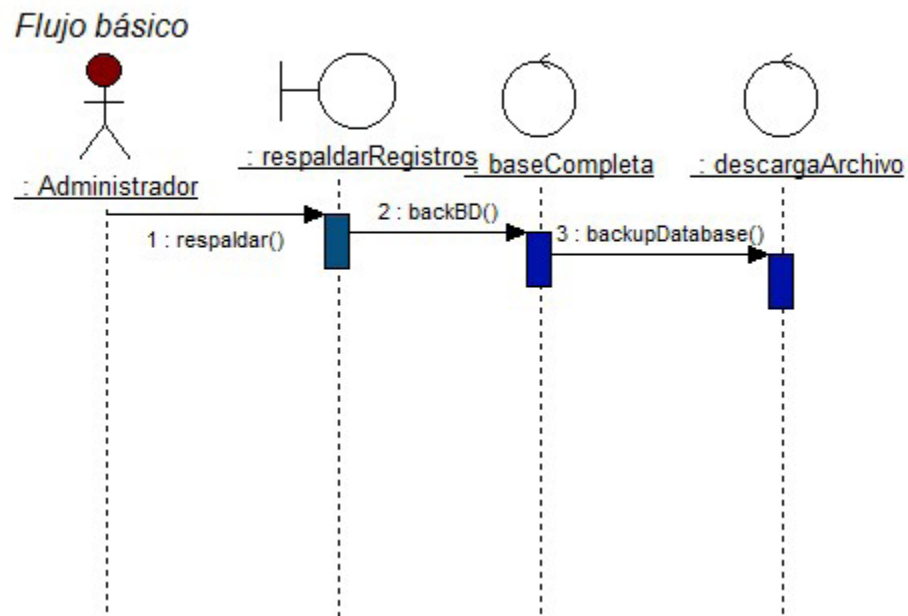


Figura 2.8.4.17. Diagrama de secuencia de la función principal del caso de uso respaldar registros.

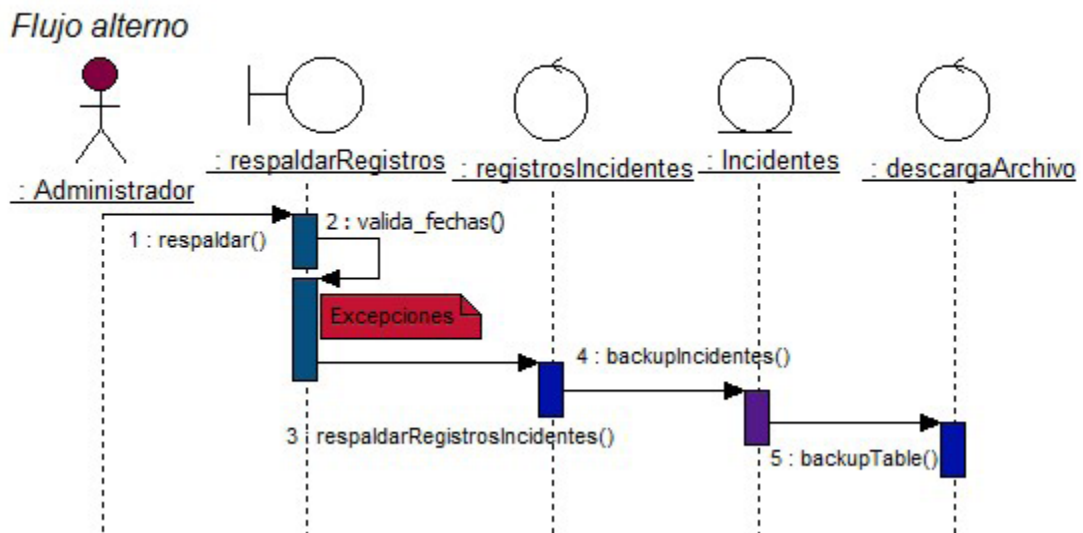


Figura 2.8.4.18. Diagrama de secuencia para una de las funciones alternas del caso de uso respaldar registros

### Flujo de excepción

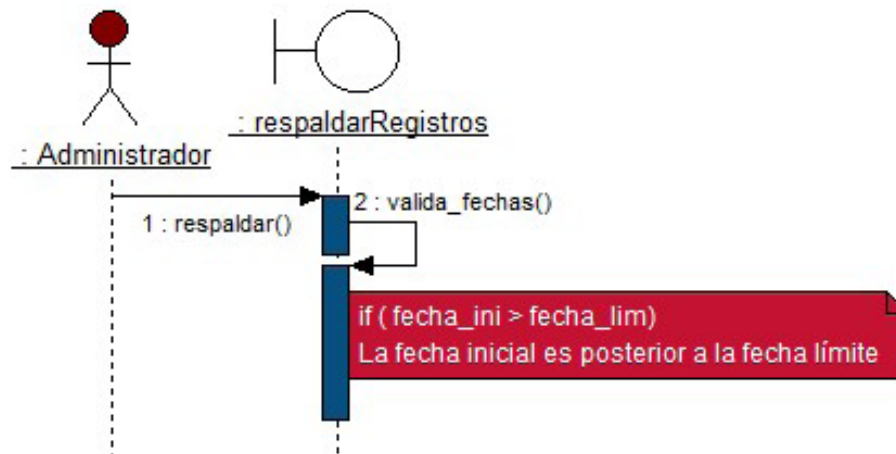


Figura 2.8.4.19. Diagrama de secuencia de la unica posible excepcion para el caso de uso respaldar registros.

### Diagramas de secuencia para el caso de uso SALC-26 Respalda registros

#### Flujo básico

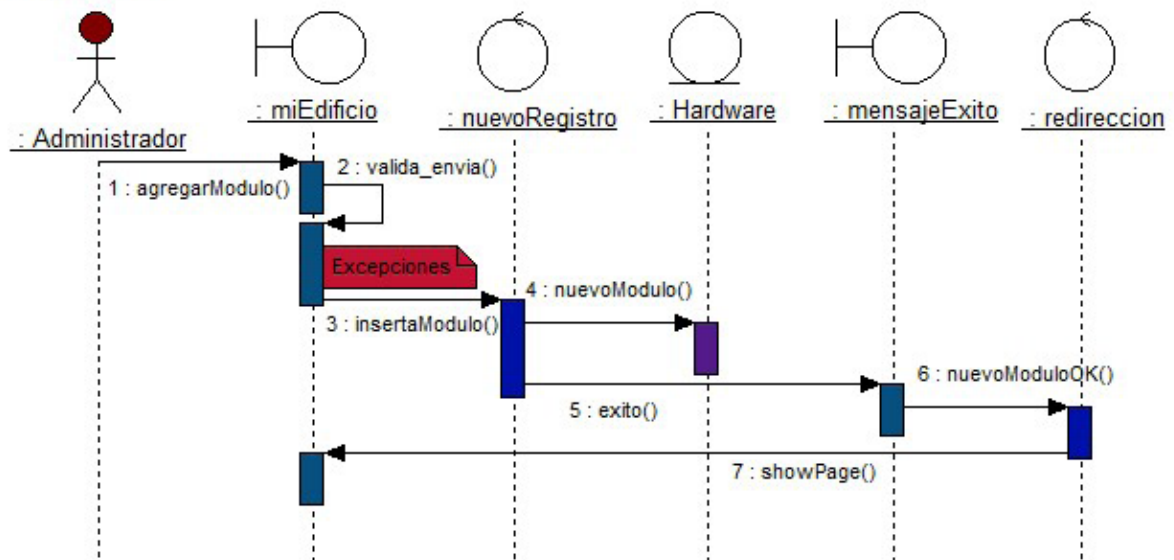


Figura 2.8.4.20. Diagrama de secuencia de la función principal del caso de uso respaldar registros.

*Flujo alterno*

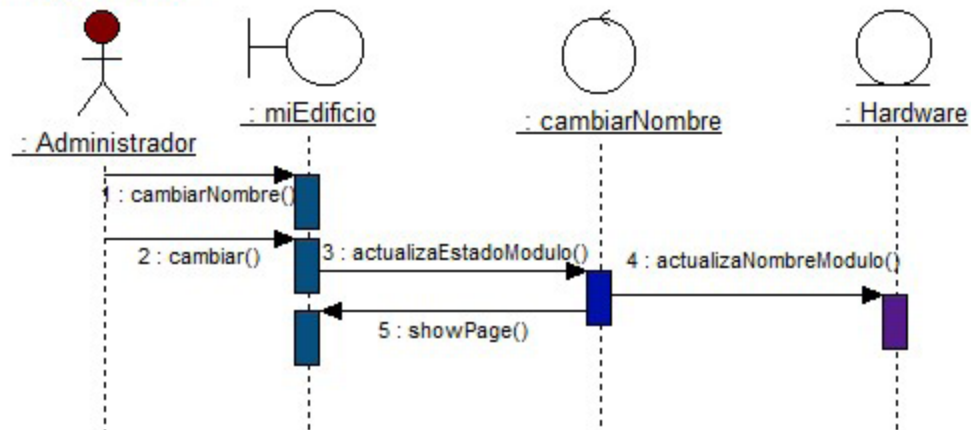


Figura 2.8.4.21. Diagrama de secuencia para alguna de las varias funciones del caso de uso respaldar registros.

*Flujo de excepción*

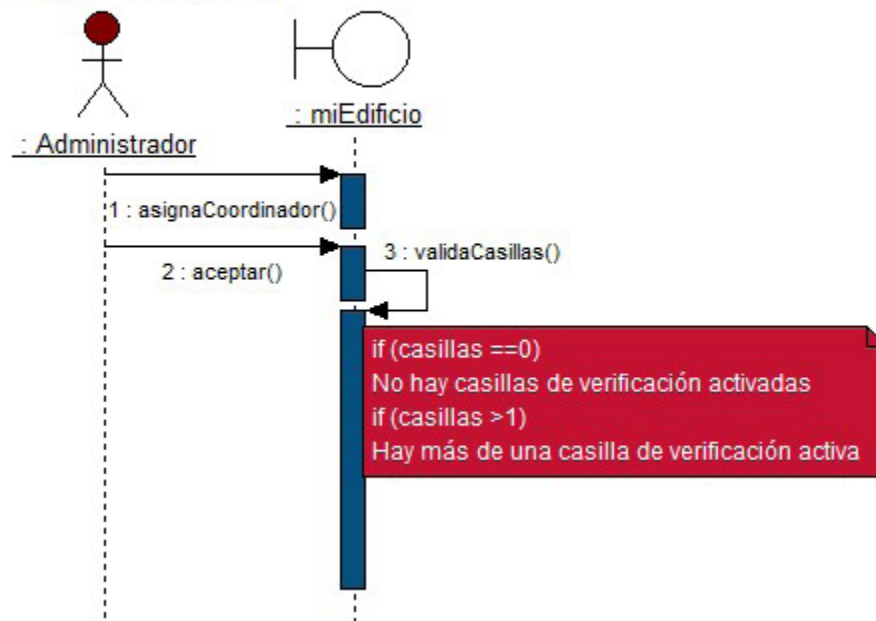


Figura 2.8.4.22. Diagrama de secuencia que muestra las excepciones que pueden ocurrir mientras se asigna un coordinador.

Diagramas de secuencia para el caso de uso SALC-29 Instalar sistema

Flujo básico

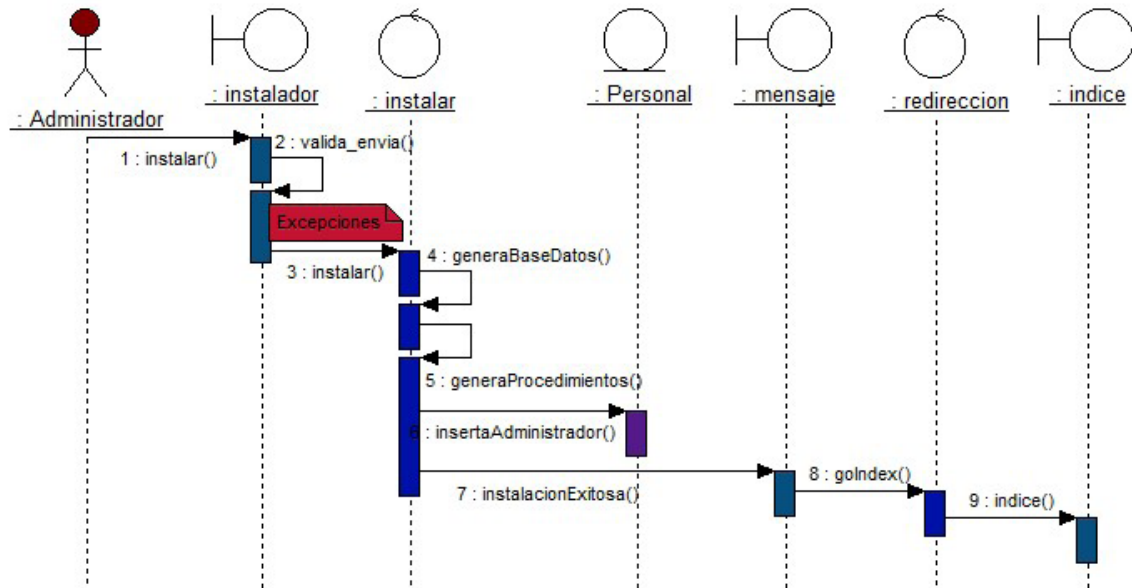


Figura 2.8.4.23. Diagrama de secuencia que explica los pasos para instalar el sistema.

Flujo de excepción

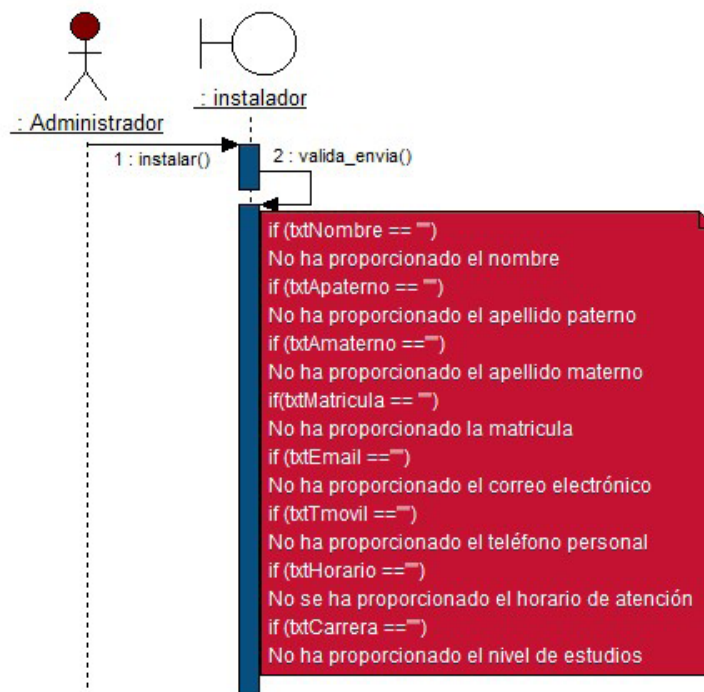


Figura 2.8.4.24. Diagrama de secuencia que expresa las multiples excepciones que pueden ocurrir al instalar el sistema

**CAPITULO 3**

**DISEÑO DE LA BASE DE DATOS**

## 3.1 DISEÑO CONCEPTUAL DE LA BASE DE DATOS

### 3.1.1 Modelo entidad-relación

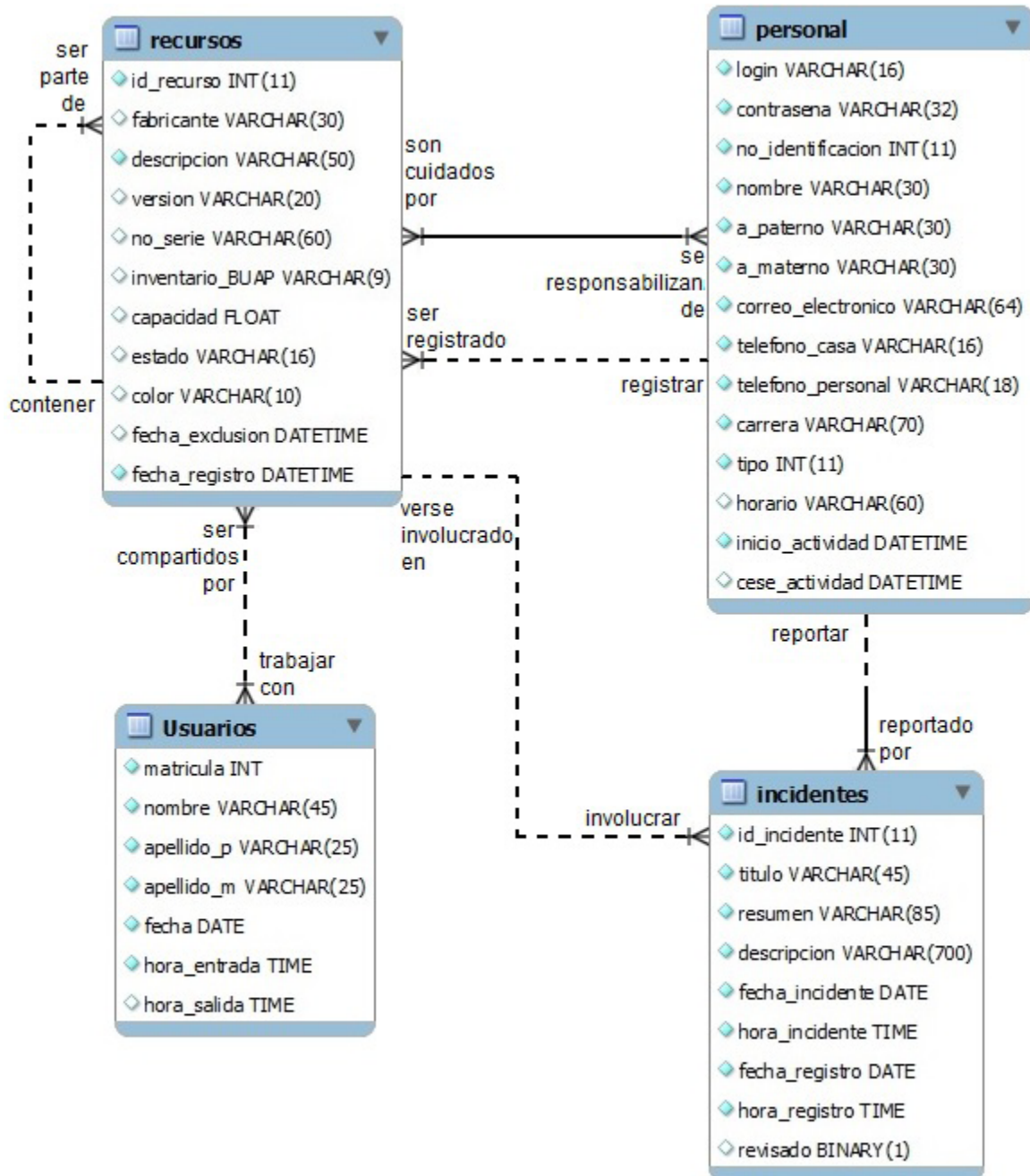


Figura 3.1.1. Modelo entidad-relación del sistema

### 3.1.2 Descripción de entidades y relaciones

Mediante entrevistas con el personal que atiende los laboratorios de cómputo se han logrado identificar 4 entidades primordiales: recursos, personal, incidentes y usuarios.

*Entidades.*

*Recursos:* Entidad que describe todos los valores físicos y lógicos que posee el laboratorio, el cual en sí mismo es un recurso. Esto significa que todas las instalaciones físicas como mesas, sillas, piezas de computadora, cables eléctricos, etc., pertenecen a este apartado. También incluye a los programas de computadora con los que cuenta el módulo, y debe ser capaz de contener el registro de las cosas que ya no se usan más en el laboratorio. La figura 3.1.2.1 describe la entidad recursos.



Column Name	Data Type
id_recurso	INT(11)
fabricante	VARCHAR(30)
descripcion	VARCHAR(50)
version	VARCHAR(20)
no_serie	VARCHAR(60)
inventario_BUAP	VARCHAR(9)
capacidad	FLOAT
estado	VARCHAR(16)
color	VARCHAR(10)
fecha_exclusion	DATETIME
fecha_registro	DATETIME

Figura 3.1.2.1. Entidad recursos

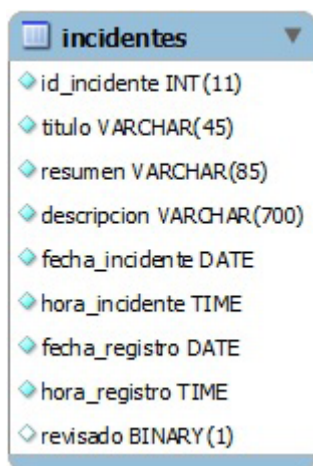


Column Name	Data Type
login	VARCHAR(16)
contrasena	VARCHAR(32)
no_identificacion	INT(11)
nombre	VARCHAR(30)
a_paterno	VARCHAR(30)
a_materno	VARCHAR(30)
correo_electronico	VARCHAR(64)
telefono_casa	VARCHAR(16)
telefono_personal	VARCHAR(18)
carrera	VARCHAR(70)
tipo	INT(11)
horario	VARCHAR(60)
inicio_actividad	DATETIME
cese_actividad	DATETIME

Figura 3.1.2.2. Entidad personal

*Personal:* Entidad que contiene la información de las personas que trabajan en los laboratorios, se dividen en tres cargos, administrador, coordinador y colaborador cada uno con distintos trabajos y privilegios en el sistema. La figura 3.1.2.2 describe los datos necesarios de los miembros del personal.

*Incidentes*: Entidad que guardará la información de accidentes, fallas e indisciplinas dentro de las instalaciones. Puede hacerse un reporte de manera muy variada dependiendo de lo que se desee informar. También debe poder almacenarse un historial de estos incidentes para futuros estudios. Los campos que se requieren pueden observarse en la figura 3.1.2.3.



incidentes	
id_incidente	INT(11)
titulo	VARCHAR(45)
resumen	VARCHAR(85)
descripcion	VARCHAR(700)
fecha_incidente	DATE
hora_incidente	TIME
fecha_registro	DATE
hora_registro	TIME
revisado	BINARY(1)

**Figura 3.1.2.3. Entidad incidentes**



Usuarios	
matricula	INT
nombre	VARCHAR(45)
apellido_p	VARCHAR(25)
apellido_m	VARCHAR(25)
fecha	DATE
hora_entrada	TIME
hora_salida	TIME

**Figura 3.1.2.4. Entidad usuarios**

*Usuarios*: Entidad que registrará a cada persona que utilice el laboratorio en los momentos que no haya una materia asignada a esa hora o bien la sesión se haya suspendido. Estas personas son alumnos de la universidad que deben presentar su credencial de estudiante para poder ofrecer el uso de un equipo de cómputo. Los datos que almacenará pueden apreciarse en la figura 3.1.2.4.

### Relaciones

Mediante la investigación previa se ha determinado que existen las siguientes relaciones.

- Un miembro del personal es responsable de uno o más recursos, así como cada recurso tiene más de un responsable.
- Cada recurso e incidente es registrado por un miembro del personal, pero también es posible que un miembro del personal nunca registre un recurso o incidente.
- Existen recursos de características especiales que pueden no ser registrados por el personal, como los módulos.
- Un recurso puede ser parte de otro recurso, por ejemplo, un monitor y un teclado son partes de otro recurso llamado equipo de cómputo, que a su vez es parte del módulo.

- Un recurso puede verse envuelto en más de un incidente
- Los usuarios pueden trabajar con uno o más recursos y un recurso puede ser compartido por muchos usuarios.
- Un incidente puede involucrar una pieza de software y es reportado por alguien del personal.

### 3.1.3 Normalización

#### Primera forma normal (1FN)

Se dice que una relación está en primera forma normal si cada atributo toma solo un valor de su dominio.

Observando las relaciones que hay entre los recursos y el personal puede decirse que el campo en común para que ambas relaciones se efectúen será login de la tabla personal. Este campo en común podría tomar más de un valor de su dominio.

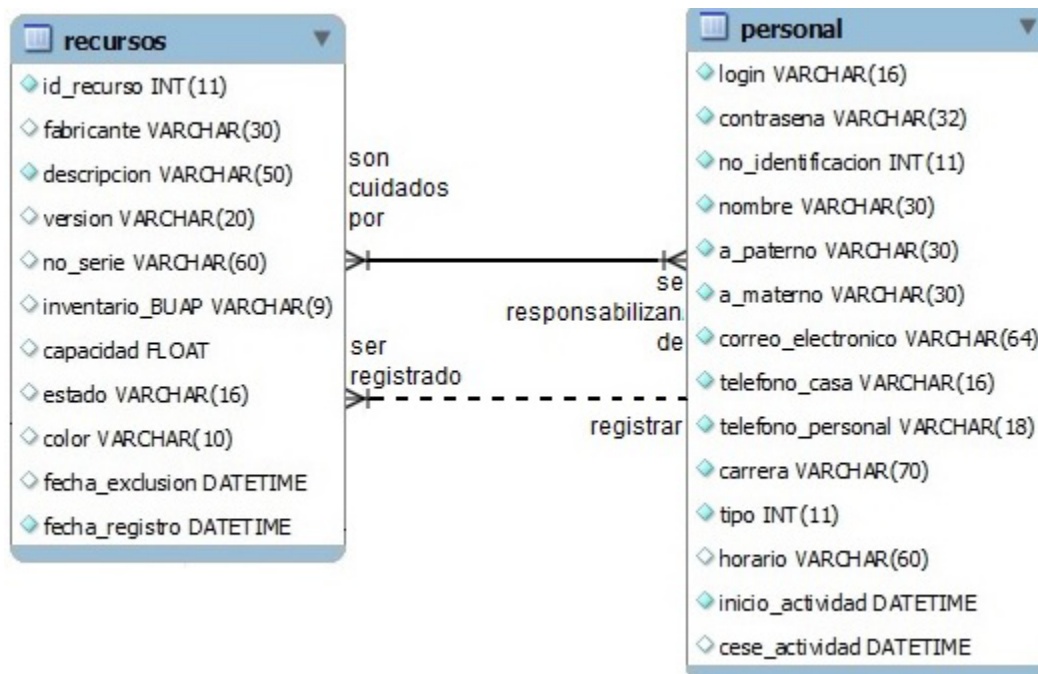


Figura 3.1.3.1. Relaciones entre recursos y personal

Siguiendo la semántica del problema se procederá a rediseñar las relaciones entre las entidades recursos y personal.

La relación “Un recurso es cuidado por una o más personas y a su vez una persona se responsabiliza de uno o más recursos”, se transformará en:

- Un recurso posee uno o más responsables.
- Alguien del personal puede tener muchas responsabilidades.

Surgiendo así la entidad “Responsable” siendo un intermediario que no solo romperá la relación “muchos a muchos” sino que también hará que todas las entidades estén en primera forma normal.

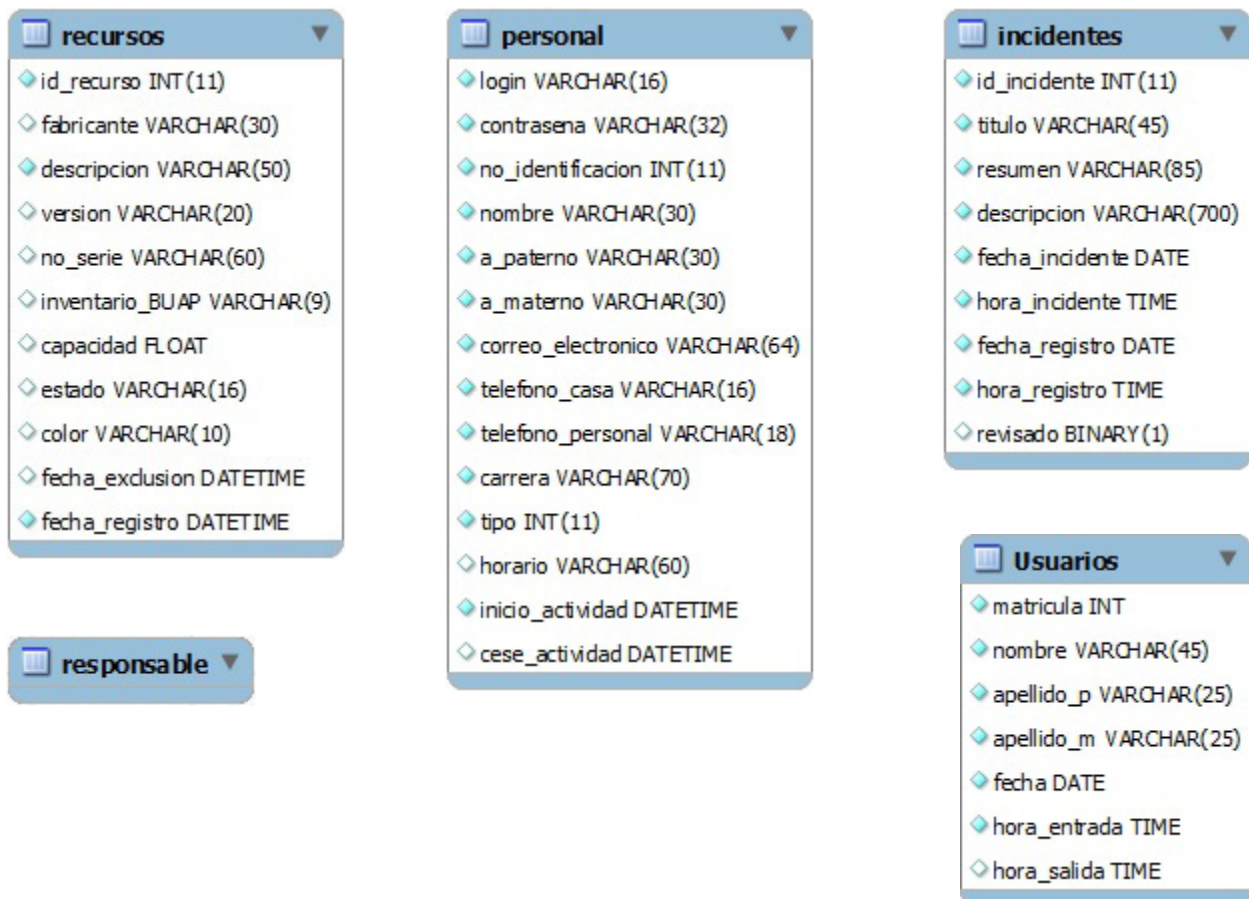
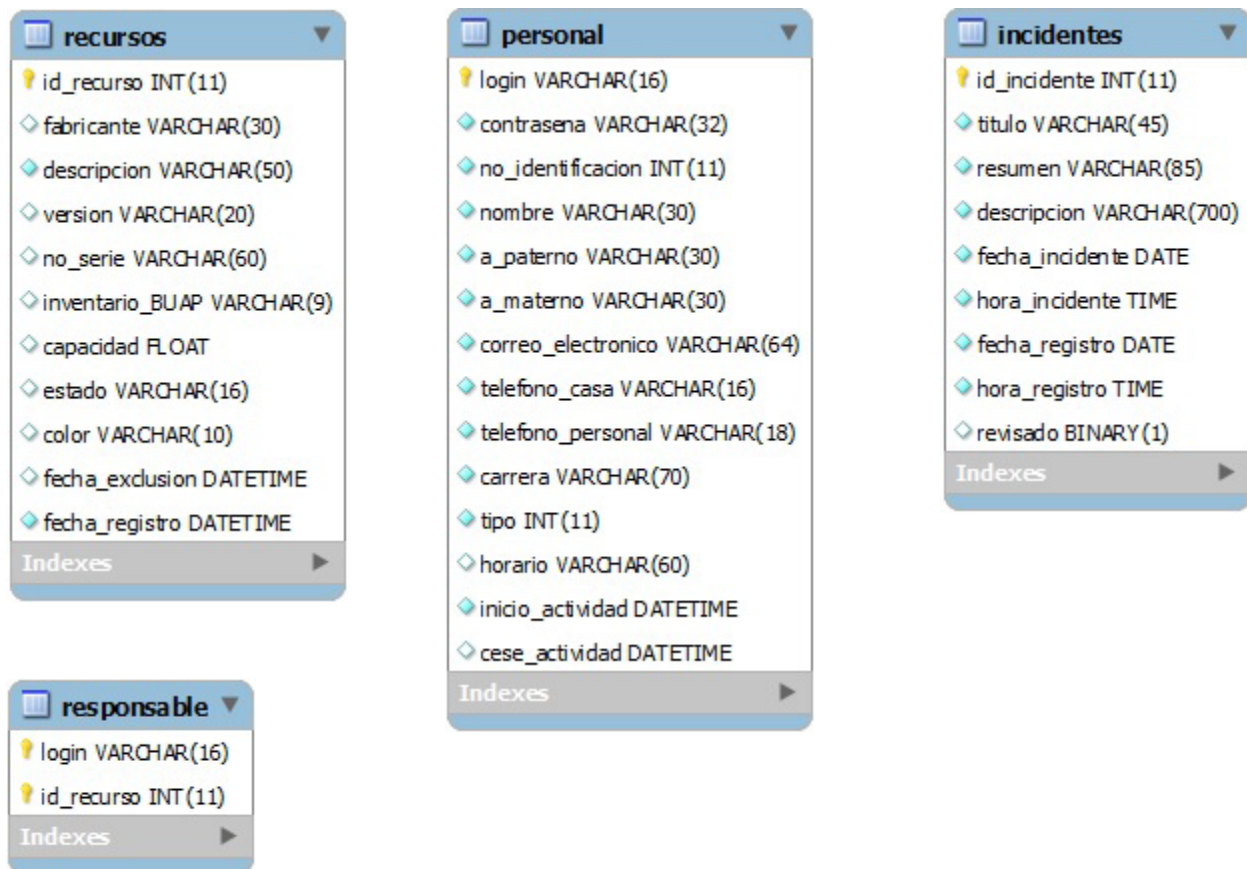


Figura 3.1.3.2. Primera forma normal de las entidades del sistema

### Segunda forma normal (2FN)

Una relación está en segunda forma normal si se encuentra en primera forma normal y además cada atributo que no sea llave dependa funcionalmente de alguna llave candidata.

La relación recursos, personal e incidentes cumplen la segunda forma normal, se han elegido los atributos que serán llaves primarias para estas relaciones vease la figura 3.1.3.3.



**Figura 3.1.3.3. Segunda forma normal de las entidades recursos, personal, incidentes y responsable.**

Mientras que para la entidad “Responsable” que referenciará a la llave primaria de las entidades recursos y personal, su llave primaria será la combinación de estas, y se encuentra en segunda forma normal.

La relación usuarios no está en segunda forma normal, pues los atributos `fecha`, `hora_entrada` y `hora_salida` no pueden ser unívocamente identificados por el campo `matricula`. Entonces se procede a separar estos atributos y crear una nueva entidad llamada `autoacceso`, contendrá los campos `fecha`, `hora_entrada`, `hora_salida` y un nuevo atributo que se usará como llave llamado `id_autoacceso` (observe la figura 3.1.3.4). Así la entidad `autoacceso` se define como:

*Autoacceso*: Entidad que registrará la entrada y salida de alumnos que deseen utilizar las instalaciones.

The image shows two database entity definitions side-by-side. The left one is for 'autoacceso' and the right one is for 'usuarios'. Both have a primary key icon (yellow lightning bolt) next to their first attribute.

Entity	Attribute	Data Type
autoacceso	id_autoacceso	INT(11)
	fecha_registro	DATE
	hora_entrada	TIME
	hora_salida	TIME
usuarios	matricula	INT(11)
	nombre	VARCHAR(45)
	apellido_p	VARCHAR(25)
	apellido_m	VARCHAR(25)

**Figura 3.1.3.4. Segunda forma normal de las entidades usuarios y autoacceso**

Todas las entidades ahora están en segunda forma normal.

### *Tercera forma normal (3FN) y forma normal de Boyce-Codd*

Una relación se encuentra en tercera forma normal si está en segunda forma normal y además ningún atributo dependa funcionalmente de otro excepto la llave primaria, es decir no exista una dependencia transitiva. Las seis entidades que ahora posee el modelo no tienen dependencias transitivas, todos y cada uno de los atributos no primarios dependen únicamente de su llave primaria.

Mientras que la forma normal de Boyce-Codd es afirmada por De Miguel Castaño et al. (2000) como: “Se dice que una relación se encuentra en FNBC si, y sólo si, todo determinante es una clave candidata” (p. 160).

- En la entidad recursos su único atributo determinante es id\_recurso, el cual es clave candidata.
- En la entidad personal los atributos determinantes son login, no\_identificacion, telefono\_casa y telefono\_personal, todos son claves candidatas y se eligió login como llave primaria.
- La entidad incidentes solo tiene un campo determinante que es id\_incidente, el cual es clave candidata.
- La entidad usuarios posee como campo determinante solo el atributo matricula, el cual es clave candidata.
- La entidad autoacceso tiene el campo determinante id\_autoacceso que es clave candidata.

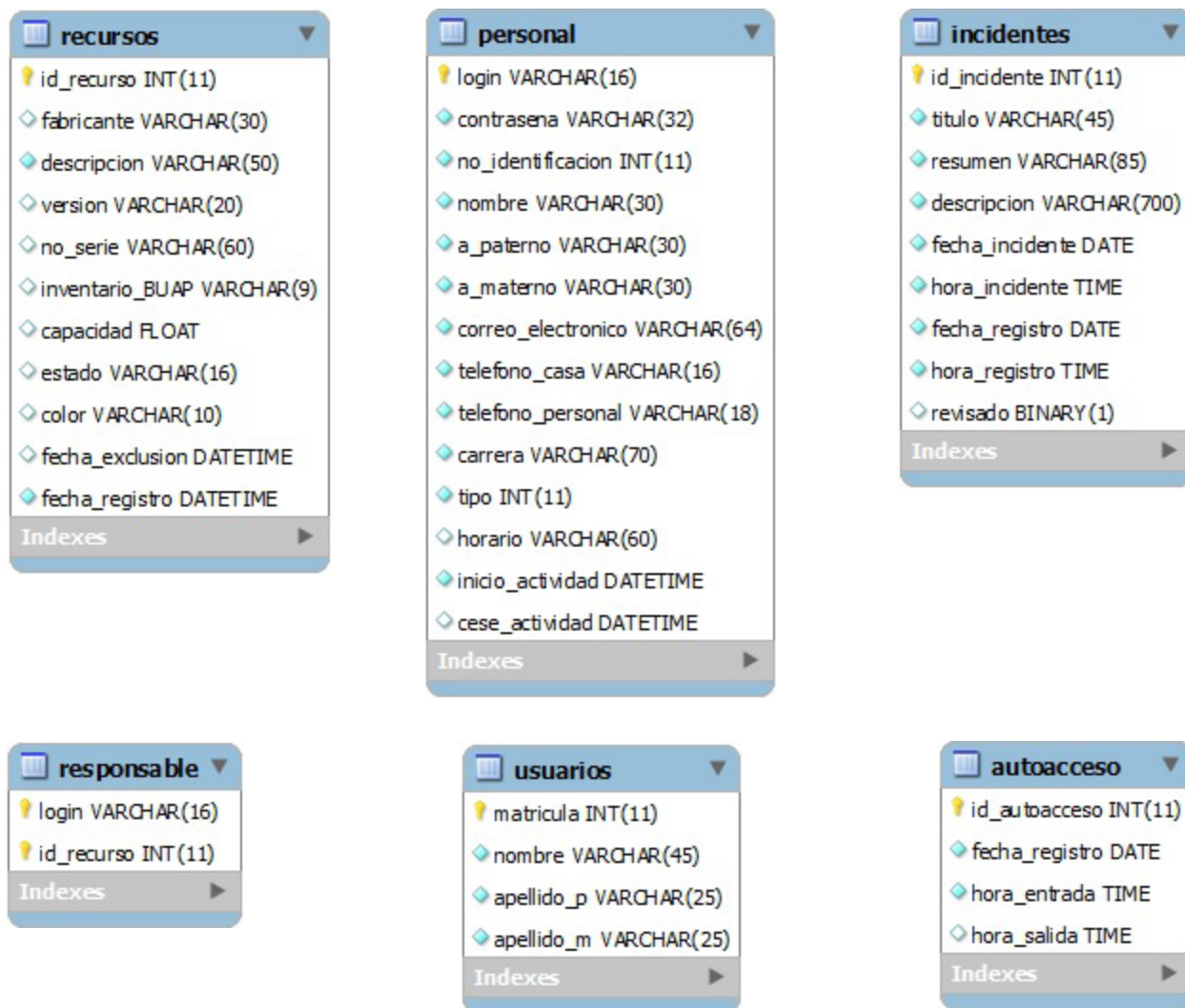


Figura 3.1.3.5. Tercera forma normal y forma normal de Boyce-Codd de las entidades del sistema.

Las entidades están en 3FN y forma normal de Boyce-Codd

#### Cuarta forma normal (4FN)

De Miguel Castaño et al. (2000) describe la forma normal: "Diremos que una relación se encuentra en 4FN si, y solo si, las únicas dependencias multivaluadas no triviales son aquellas en las cuales una clave multi-determina un atributo, es decir, toda dependencia multivaluada viene determinada por una clave candidata" (p. 185).

El análisis de los requisitos para la construcción de la base de datos dice que cada módulo posee su propia lista de software y además se debe saber que software tiene instalado cada equipo de cómputo.

Supongamos que se tienen los siguientes registros.

**Tabla 3.1.1. Ejemplo de registros de la entidad recursos**

<b>Recursos</b>			
<b>id_recurso</b>	1	2	19
<b>fabricante</b>	null	null	null
<b>descripción</b>	Módulo	Módulo	Equipo de computo
<b>versión</b>	null	null	null
<b>numero_serie</b>	1	2	27
<b>inventario_BUAP</b>	null	null	null
<b>capacidad</b>	null	null	null
<b>estado</b>	Funcional	Funcional	Funcional
<b>color</b>	null	null	null
<b>fecha_exclusión</b>	null	null	null
<b>fecha_registro</b>	2014-06-01 16:56:12	2014-06-01 16:56:52	2014-06-20 14:56:30
<b>instalado_en</b>	null	null	1
<b>id_recurso</b>	20	21	84
<b>fabricante</b>	null	null	Microsoft
<b>descripción</b>	Equipo de computo	Equipo de computo	Office 2013 Professional
<b>versión</b>	null	null	15.0.4420
<b>numero_serie</b>	29	30	null
<b>inventario_BUAP</b>	null	null	null
<b>capacidad</b>	null	null	null
<b>estado</b>	Funcional	Funcional	null
<b>color</b>	null	null	null
<b>fecha_exclusión</b>	null	null	null
<b>fecha_registro</b>	2014-06-20 15:00:30	2014-06-20 15:00:40	2014-06-20 15:30:00
<b>instalado_en</b>	1	1	1
<b>id_recurso</b>	85	86	87
<b>fabricante</b>	Microsoft	Microsoft	Microsoft
<b>descripción</b>	Windows 7 Professional	Windows 7 Professional	Windows 7 Professional
<b>versión</b>	6.1009	6.1009	6.1009
<b>numero_serie</b>	null	null	null
<b>inventario_BUAP</b>	null	null	null
<b>capacidad</b>	null	null	null
<b>estado</b>	null	null	null
<b>color</b>	null	null	null
<b>fecha_exclusión</b>	null	null	null
<b>fecha_registro</b>	2014-06-20 15:31:01	2014-06-20 15:31:41	2014-06-20 15:33:20
<b>instalado_en</b>	1	2	22

Se pueden observar que un registro como el de Windows 7 Professional, está siendo multivaluado en su último atributo. La cuarta forma normal dicta que debe haber una separación de estos atributos multivaluados en otra entidad, sin embargo la separación de este campo debe ser cuidadosa de manera que no se pierda la semántica del problema.

Cada módulo posee su propia lista de software, es posible entonces crear una entidad que liste los módulos y el software que poseen referenciándolo en la entidad recursos, a esta se le llamará *almacen\_software*.

Se debe saber que software tiene instalado cada equipo de cómputo, también debe crearse una entidad que referencie un equipo de cómputo y especifique un recurso de software. Esta entidad se llamará *software\_equipo*.

Así las nuevas entidades se definen como (vease la figura 3.1.3.6) .

*Almacen\_Software*: Entidad que referenciará un recurso de software e indicará que módulo tiene en su almacén ese recurso.

*Software\_Equipo*: Entidad que guarda una relación entre el equipo de cómputo y el software que tiene instalado.



**Figura 3.1.3.6. Entidades almacen de software y software en equipo.**

Se tienen ahora 8 entidades las cuales están en 4FN.

### 3.1.4 Diseño lógico

Una vez normalizadas las entidades hasta su cuarta forma normal y siguiendo la semántica del problema junto con las relaciones previamente mencionadas, se puede obtener el modelo conceptual de la base de datos (figura 3.1.4.1).

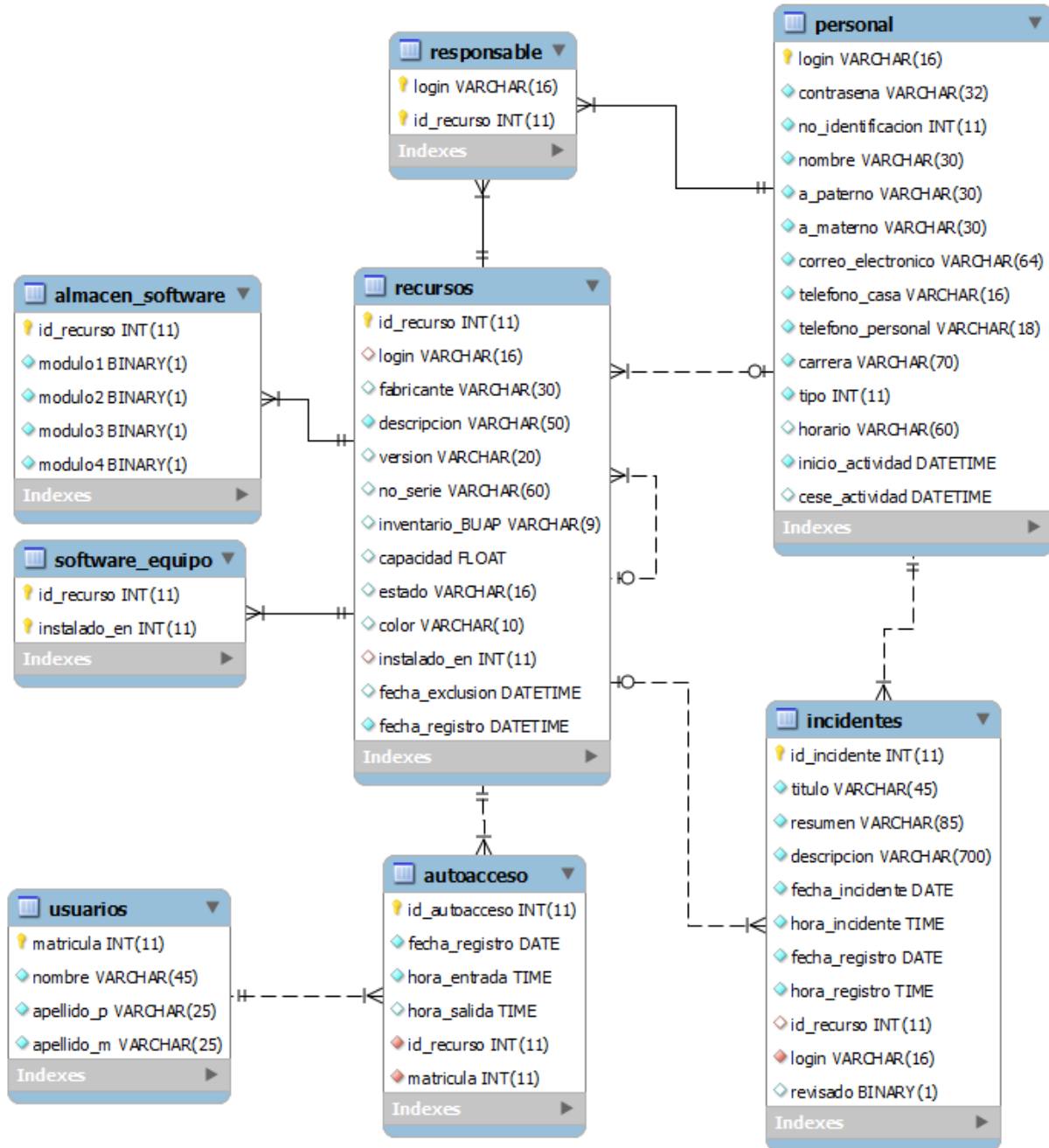


Figura 3.1.4.1. Modelo conceptual de la base de datos.

El diseño lógico se caracteriza por ser simple, ligero para el procesador y sin redundancias, un diseño lógico estándar se basa en el modelo relacional y el objetivo es transformar el modelo conceptual E/R en un modelo relacional de manera que cualquier SGBD pueda soportarlo. La transformación del modelo entidad-relación a un relacional debe seguir tres reglas básicas propuestas por De Miguel Castaño et al. (2000):

1. Todo tipo de entidad se convierte en una relación.
2. Todo tipo de interrelación N:M se transforma en una relación.
3. Para todo tipo de interrelación 1:N se realiza lo que se domina *propagación de clave* (regla general), o bien se crea una nueva relación (p. 345).

Siguiendo estos pasos el modelo conceptual se vuelven datos más simples.

**Personal** (**login\***, **contrasena\***, **no\_identificacion\***, **nombre\***, **a\_paterno\***, **a\_materno\***, **correo\_electronico\***, **telefono\_casa\***, **telefono\_personal\***, **carrera\***, **tipo\***, **horario**, **inicio\_actividad\***, **cese\_actividad**)

**Recursos** (**id\_recurso\***, **fabricante**, **descripcion\***, **version**, **no\_serie**, **inventario\_BUAP**, **capacidad**, **estado**, **color**, **fecha\_exclusion**, **fecha\_registro\***, **login**, **instalado\_en**)

**Responsable** (**login\***, **id\_recurso\***)

**Almacen\_software** (**id\_recurso\***, **modulo1\***, **modulo2\***, **modulo3\***, **modulo4\***)

**Software\_equipo** (**id\_recurso\***, **instalado\_en\***)

**Incidentes** (**id\_incidente\***, **título\***, **resumen\***, **descripcion\***, **fecha\_incidente\***, **hora\_incidente\***, **fecha\_registro\***, **hora\_registro\***, **revisado**, **id\_recurso**, **login\***)

**Usuarios** (**matricula\***, **nombre\***, **apellido\_p\***, **apellido\_m\***)

**Autoacceso** (**id\_autoacceso\***, **fecha\_registro\***, **hora\_entrada\***, **hora\_salida**, **id\_recurso\***, **matricula\***)

A continuación se deben crear las sentencias en lenguaje de definición de datos con la información del modelo relacional sin olvidar su cardinalidad que aún se puede apreciar en el modelo conceptual.

*Paso 1.* Identificar los dominios de cada atributo clave y no clave, si se observa el modelo conceptual, este paso ya está hecho.

*Paso 2.* Crear las tablas a partir de las relaciones. En un lenguaje de definición de datos estandar debería poderse crear la siguiente sintaxis.

**CREATE TABLE [IF NOT EXISTS] nombre\_de\_la\_tabla**

Las entidades se transforman en tablas:

```
CREATE TABLE Personal
CREATE TABLE Recursos
CREATE TABLE Responsable
CREATE TABLE Almacen_software
CREATE TABLE Software_equipo
CREATE TABLE Incidentes
CREATE TABLE Usuarios
CREATE TABLE Autoacceso
```

*Paso 3.* Transformar atributos en columnas de la tabla de manera que se puedan establecer las llaves primarias para cada relación y también identificar atributos únicos alternativos. Mediante SQL en el lenguaje de definición de datos puede agregar este paso al momento de crear una tabla:

**CREATE TABLE [IF NOT EXISTS] nombre\_de\_la\_tabla (definición\_de\_columnas)**

La definición de columnas se divide en definicion de campos y definicion de llaves , la definicion de campos puede ser muy enriquecido dependiendo de la necesidad, una sintaxis en SQL debería poder aceptar los siguientes parametros.

(definición de campo)

**(Nombre\_de\_campo tipo [NULL o NOT NULL] [DEFAULT valor\_por\_defecto] [AUTO\_INCREMENT] )**

Definir una llave primaria requiere definir primero todas las columnas y agregar la sintaxis PRIMARY KEY (nombre de la columna), es decir:

**(definicion de columnas)** = (definicion de campo 1, definicion de campo 2, ..., definicion de campo n, definicion de llaves)

```
CREATE TABLE IF NOT EXISTS Personal (login VARCHAR (16) NOT NULL,
contrasena VARCHAR(32) NOT NULL , no_identificacion INT(11) NOT NULL , nombre
VARCHAR(30) NOT NULL , a_paterno VARCHAR(30) NOT NULL , a_materno
VARCHAR(30) NOT NULL , correo_electronico VARCHAR(64) NOT NULL DEFAULT " ,
telefono_casa VARCHAR(16) NOT NULL DEFAULT " , telefono_personal VARCHAR(18)
NOT NULL DEFAULT " , carrera VARCHAR(70) NOT NULL DEFAULT " , tipo INT(11)
NOT NULL , horario VARCHAR(60) NULL DEFAULT NULL , inicio_actividad DATETIME
NOT NULL , cese_actividad DATETIME NULL DEFAULT NULL , PRIMARY KEY (login) )
```

```
CREATE TABLE IF NOT EXISTS Recursos ( id_recurso INT(11) NOT NULL
AUTO_INCREMENT , login VARCHAR(16), fabricante VARCHAR(30) NULL DEFAULT
NULL , descripcion VARCHAR(50) NOT NULL , version VARCHAR(20) NULL DEFAULT
NULL , no_serie VARCHAR(60) NULL DEFAULT " , inventario_BUAP VARCHAR(9) NULL
DEFAULT NULL , capacidad FLOAT NULL DEFAULT NULL , estado VARCHAR(16) NULL
DEFAULT NULL , color VARCHAR(10) NULL DEFAULT NULL , instalado_en INT(11),
fecha_exclusion DATETIME NULL DEFAULT NULL , fecha_registro DATETIME NOT
NULL , PRIMARY KEY (id_recurso) )
```

```
CREATE TABLE IF NOT EXISTS Almacen_software ( id_recurso INT(11) NOT
NULL , modulo1 BINARY(1) NOT NULL , modulo2 BINARY(1) NOT NULL , modulo3
BINARY(1) NOT NULL , modulo4 BINARY(1) NOT NULL , PRIMARY KEY (id_recurso) )
```

```
CREATE TABLE IF NOT EXISTS Usuarios ( matricula INT(11) NOT NULL ,
nombre VARCHAR(45) NOT NULL , apellido_p VARCHAR(25) NOT NULL , apellido_m
VARCHAR(25) NOT NULL , PRIMARY KEY (matricula) )
```

```
CREATE TABLE IF NOT EXISTS Autoacceso ( id_autoacceso INT(11) NOT NULL
AUTO_INCREMENT , fecha_registro DATE NOT NULL , hora_entrada TIME NOT NULL
, hora_salida TIME NULL DEFAULT NULL , id_recurso INT(11) NOT NULL , matricula
INT(11) NOT NULL , PRIMARY KEY (id_autoacceso) )
```

```
CREATE TABLE IF NOT EXISTS Incidentes ( id_incidente INT(11) NOT NULL
AUTO_INCREMENT , titulo VARCHAR(45) NOT NULL , resumen VARCHAR(85) NOT
NULL , descripcion VARCHAR(700) NOT NULL , fecha_incidente DATE NOT NULL ,
hora_incidente TIME NOT NULL , fecha_registro DATE NOT NULL , hora_registro TIME
NOT NULL , id_recurso INT(11), login VARCHAR(16) NOT NULL , revisado BINARY(1)
NULL DEFAULT NULL , PRIMARY KEY (id_incidente) )
```

```
CREATE TABLE IF NOT EXISTS Responsable ( login VARCHAR(16) NOT NULL
, id_recurso INT(11) NOT NULL DEFAULT '0' , PRIMARY KEY (login, id_recurso) )
```

```
CREATE TABLE IF NOT EXISTS Software_equipo ( id_recurso INT(11) NOT
NULL , instalado_en INT(11) NOT NULL , PRIMARY KEY (instalado_en, id_recurso) )
```

*Paso 4.* Transformar las relaciones en llaves foráneas y referencias. Si la interrelación es 1: N se propagará la llave primaria de la relación 1 hacia la N como llave foránea. Si la interrelación es de tipo 1:1 la propagación de la clave es bidireccional. Hay que hacer énfasis en la cardinalidad de estas propagaciones colocando la opción NOT NULL para las interrelaciones 1: N y 1:1, y la opción NULL para interrelaciones 0: N y la combinatoria para las 1:0 y 0:1. Hasta este punto las entidades no están relacionadas, la opción FOREIGN KEY y REFERENCES de SQL crea este enlace:

**Definición de llaves** = PRIMARY KEY (**campo\_llave\_de\_esta\_tabla**), FOREIGN KEY [**columna\_de\_esta\_tabla**] REFERENCES otra\_tabla [(**campo\_llave\_otra\_tabla**)].

Así la especificación de tablas, sus columnas, llaves y la lógica de sus relaciones se convierten en el diseño lógico estándar que debe ser capaz de tener compatibilidad con muchos SGBD basados en lenguaje de consulta estructurado (por sus siglas en inglés Structured Query Language). La transformación del modelo conceptual a su diseño lógico es:

```

CREATE TABLE IF NOT EXISTS Personal (
  login VARCHAR(16) NOT NULL ,
  contraseña VARCHAR(32) NOT NULL ,
  no_identificacion INT(11) NOT NULL ,
  nombre VARCHAR(30) NOT NULL ,
  a_paterno VARCHAR(30) NOT NULL ,
  a_materno VARCHAR(30) NOT NULL ,
  correo_electronico VARCHAR(64) NOT NULL DEFAULT " ,
  telefono_casa VARCHAR(16) NOT NULL DEFAULT " ,
  telefono_personal VARCHAR(18) NOT NULL DEFAULT " ,
  carrera VARCHAR(70) NOT NULL DEFAULT " ,
  tipo INT(11) NOT NULL ,
  horario VARCHAR(60) NULL DEFAULT NULL ,
  inicio_actividad DATETIME NOT NULL ,
  cese_actividad DATETIME NULL DEFAULT NULL ,
  PRIMARY KEY (login) )

```

```

CREATE TABLE IF NOT EXISTS Almacen_software (
  id_recurso INT(11) NOT NULL ,
  modulo1 BINARY(1) NOT NULL ,
  modulo2 BINARY(1) NOT NULL ,
  modulo3 BINARY(1) NOT NULL ,
  modulo4 BINARY(1) NOT NULL ,
  PRIMARY KEY (id_recurso) ,
  FOREIGN KEY (id_recurso )
  REFERENCES Recursos (id_recurso )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS Usuarios (
  matricula INT(11) NOT NULL ,
  nombre VARCHAR(45) NOT NULL ,
  apellido_p VARCHAR(25) NOT NULL ,
  apellido_m VARCHAR(25) NOT NULL ,
  PRIMARY KEY (matricula) )

```

```

CREATE TABLE IF NOT EXISTS Recursos (
  id_recurso INT(11) NOT NULL AUTO_INCREMENT ,
  login VARCHAR(16) NULL DEFAULT NULL ,
  fabricante VARCHAR(30) NULL DEFAULT NULL ,
  descripcion VARCHAR(50) NOT NULL ,
  version VARCHAR(20) NULL DEFAULT NULL ,
  no_serie VARCHAR(60) NULL DEFAULT '' ,
  inventario_BUAP VARCHAR(9) NULL DEFAULT NULL ,
  capacidad FLOAT NULL DEFAULT NULL ,
  estado VARCHAR(16) NULL DEFAULT NULL ,
  color VARCHAR(10) NULL DEFAULT NULL ,
  instalado_en INT(11) NULL DEFAULT NULL ,
  fecha_exclusion DATETIME NULL DEFAULT NULL ,
  fecha_registro DATETIME NOT NULL ,
  PRIMARY KEY (id_recurso) ,
  FOREIGN KEY (login )
  REFERENCES Personal (login )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  FOREIGN KEY (instalado_en )
  REFERENCES Recursos (id_recurso )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS Responsable (
  login VARCHAR(16) NOT NULL ,
  id_recurso INT(11) NOT NULL DEFAULT '0' ,
  PRIMARY KEY (login, id_recurso) ,
  FOREIGN KEY (login )
  REFERENCES Personal (login )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  FOREIGN KEY (id_recurso )
  REFERENCES Recursos (id_recurso )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS Autoacceso (
  id_autoacceso INT(11) NOT NULL AUTO_INCREMENT ,
  fecha_registro DATE NOT NULL ,
  hora_entrada TIME NOT NULL ,
  hora_salida TIME NULL DEFAULT NULL ,
  id_recurso INT(11) NOT NULL ,
  matricula INT(11) NOT NULL ,
  PRIMARY KEY (id_autoacceso) ,
  FOREIGN KEY (id_recurso )
  REFERENCES Recursos (id_recurso )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  FOREIGN KEY (matricula )
  REFERENCES Usuarios (matricula )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```

CREATE TABLE IF NOT EXISTS Incidentes (
  id_incidente INT(11) NOT NULL AUTO_INCREMENT ,
  titulo VARCHAR(45) NOT NULL ,
  resumen VARCHAR(85) NOT NULL ,
  descripcion VARCHAR(700) NOT NULL ,
  fecha_incidente DATE NOT NULL ,
  hora_incidente TIME NOT NULL ,
  fecha_registro DATE NOT NULL ,
  hora_registro TIME NOT NULL ,
  id_recurso INT(11) NULL DEFAULT NULL ,
  login VARCHAR(16) NOT NULL ,
  revisado BINARY(1) NULL DEFAULT NULL ,
  PRIMARY KEY (id_incidente) ,
  FOREIGN KEY (login )
  REFERENCES Personal (login )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  FOREIGN KEY (id_recurso )
  REFERENCES Recursos (id_recurso )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

```
CREATE TABLE IF NOT EXISTS Software_equipo (  
  id_recurso INT(11) NOT NULL ,  
  instalado_en INT(11) NOT NULL ,  
  PRIMARY KEY (instalado_en, id_recurso) ,  
  FOREIGN KEY (id_recurso )  
  REFERENCES Recursos (id_recurso )  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

# **CAPITULO 4**

## **IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA**

## 4.1 IMPLEMENTACIÓN

El “Sistema de Administración de los Laboratorios de Computo” de la facultad de ciencias de la computación, será llamado por su acrónimo “SALC-FCC”. Es un sistema en la web para uso de los profesores y alumnos que son responsables del funcionamiento y administración de los laboratorios.

Este sistema está construido en base a conocimientos de tecnologías de la información e ingeniería de software siguiendo las recomendaciones del proceso unificado de desarrollo de software. Se ha estructurado bajo el modelo cliente-servidor, utilizando servicios web proporcionados por el software Apache, programación cliente con JavaScript y JQuery, programación servidor mediante el lenguaje PHP, gestión de base de datos por MySQL y técnicas de recuperación de información dinámicamente AJAX.

Se ha construido con el apoyo de programas de software de modelado de base de datos MySQL Workbench y MySQL Query Browser, también para la codificación de PHP, JavaScript y HTML se ha utilizado Eclipse. La información de las interfaces es manejada como clases en el lenguaje PHP siguiendo el paradigma orientado a objetos.

Además se ha construido un instalador del sistema que en caso de cumplir con los requisitos recomendados (véase la tabla 4.1) podrá instalarse de manera automática en el servidor y listo para usarse.

Por último el modulo incluido de respaldo del sistema permitirá una fácil migración entre servidores, asegurando así su portabilidad.

Requerimientos del sistema:

Minimos

**Tabla 4.1.1. Requisitos minimos para funcionamiento del sistema.**

<b>Servidor web</b>	<b>SGBD</b>	<b>Explorador</b>
Apache ver. 2.2.14	MySQL 5.1.61	Microsoft Internet Explorer 8
PHP 5.3	phpMyAdmin 3.5.1	Mozilla Firefox 5.0
100MB disco	Tablas MyISAM	Google Chrome
10MB transferencia	Procedimientos almacenados	Opera

## Recomendados

**Tabla 4.1.2. Requisitos recomendados para funcionamiento del sistema.**

Servidor web	SGBD	Explorador
Apache ver. 2.4.9	MySQL 5.6.17	Microsoft Internet Explorer 10
PHP 5.5.12	phpMyAdmin 4.1.14	Mozilla Firefox 30.0
2GB disco	Tablas innoDB	Google Chrome
100MB transferencia	Procedimientos almacenados	Opera

### Base de datos.

La base de datos está implementada para funcionar idealmente con el motor innoDB, pero pueden también funcionar correctamente con el motor MyISAM. Se cuentan con 8 tablas que pueden observarse en el modelo conceptual de la figura 3.1.4.1.

También para obtener un grado de seguridad para evitar la inyección de SQL, solo se manejan procedimientos almacenados para cualquier interacción con la base de datos, un total de 121 procedimientos almacenados fueron programados. Además por motivos de integridad y manejo del sistema, se ha desactivado el borrado en cascada de datos.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<b>login</b>	varchar(16)	utf8_general_ci		No	Ninguna
<b>contrasena</b>	varchar(32)	utf8_general_ci		No	Ninguna
<b>no_identificacion</b>	int(11)			No	Ninguna
<b>nombre</b>	varchar(30)	utf8_general_ci		No	Ninguna
<b>a_paterno</b>	varchar(30)	utf8_general_ci		No	Ninguna
<b>a_materno</b>	varchar(30)	utf8_general_ci		No	Ninguna
<b>correo_electronico</b>	varchar(64)	utf8_general_ci		No	
<b>telefono_casa</b>	varchar(16)	utf8_general_ci		No	
<b>telefono_personal</b>	varchar(18)	utf8_general_ci		No	
<b>carrera</b>	varchar(70)	utf8_general_ci		No	
<b>tipo</b>	int(11)			No	Ninguna
<b>horario</b>	varchar(60)	utf8_general_ci		Sí	NULL
<b>inicio_actividad</b>	datetime			No	Ninguna
<b>cese_actividad</b>	datetime			Sí	NULL

**Figura 4.1.1. Tabla de personal**

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>id_recurso</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>login</b>	varchar(16)	utf8_general_ci		Sí	<i>NULL</i>	
<b>fabricante</b>	varchar(30)	utf8_general_ci		Sí	<i>NULL</i>	
<b>descripcion</b>	varchar(50)	utf8_general_ci		No	<i>Ninguna</i>	
<b>version</b>	varchar(20)	utf8_general_ci		Sí	<i>NULL</i>	
<b>no_serie</b>	varchar(60)	utf8_general_ci		Sí		
<b>inventario_BUAP</b>	varchar(9)	utf8_general_ci		Sí	<i>NULL</i>	
<b>capacidad</b>	float			Sí	<i>NULL</i>	
<b>estado</b>	varchar(16)	utf8_general_ci		Sí	<i>NULL</i>	
<b>color</b>	varchar(10)	utf8_general_ci		Sí	<i>NULL</i>	
<b>instalado_en</b>	int(11)			Sí	<i>NULL</i>	
<b>fecha_exclusion</b>	datetime			Sí	<i>NULL</i>	
<b>fecha_registro</b>	datetime			No	<i>Ninguna</i>	

Figura 4.1.2. Tabla de recursos

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>id_incidente</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>titulo</b>	varchar(45)	utf8_general_ci		No	<i>Ninguna</i>	
<b>resumen</b>	varchar(85)	utf8_general_ci		No	<i>Ninguna</i>	
<b>descripcion</b>	varchar(700)	utf8_general_ci		No	<i>Ninguna</i>	
<b>fecha_incidente</b>	date			No	<i>Ninguna</i>	
<b>hora_incidente</b>	time			No	<i>Ninguna</i>	
<b>fecha_registro</b>	date			No	<i>Ninguna</i>	
<b>hora_registro</b>	time			No	<i>Ninguna</i>	
<b>id_recurso</b>	int(11)			Sí	<i>NULL</i>	
<b>login</b>	varchar(16)	utf8_general_ci		No	<i>Ninguna</i>	
<b>revisado</b>	binary(1)			Sí	<i>NULL</i>	

Figura 4.1.3. Tabla de Incidentes

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>id_autoacceso</u>	int(11)			No	Ninguna	AUTO_INCREMENT
fecha_registro	date			No	Ninguna	
hora_entrada	time			No	Ninguna	
hora_salida	time			Sí	NULL	
id_recurso	int(11)			No	Ninguna	
matricula	int(11)			No	Ninguna	

Figura 4.1.4. Tabla de autoacceso

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<u>matricula</u>	int(11)			No	Ninguna
nombre	varchar(45)	utf8_general_ci		No	Ninguna
apellido_p	varchar(25)	utf8_general_ci		No	Ninguna
apellido_m	varchar(25)	utf8_general_ci		No	Ninguna

Figura 4.1.5. Tabla de usuarios

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>login</u>	varchar(16)	utf8_general_ci		No	Ninguna	
<u>id_recurso</u>	int(11)			No	0	

Figura 4.1.6. Tabla de responsables

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>id_recurso</u>	int(11)			No	Ninguna	
modulo1	binary(1)			No	Ninguna	
modulo2	binary(1)			No	Ninguna	
modulo3	binary(1)			No	Ninguna	
modulo4	binary(1)			No	Ninguna	

Figura 4.1.7. Tabla de almacén de software.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<u>id_recurso</u>	int(11)			No	Ninguna	
<u>instalado_en</u>	int(11)			No	Ninguna	

Figura 4.1.8. Tabla de software en equipo

USP_ACTUALIZA_ALMACEN	USP_ACTUALIZA_PERSONAL_PASS
USP_ACTUALIZA_BAJA_EQUIPO	USP_ACTUALIZA_RECURSO_ESTADO
USP_ACTUALIZA_BAJA_HARDWARE	USP_ACTUALIZA_SALIDA_AUTOACCESO
USP_ACTUALIZA_BAJA_PERSONAL	USP_ACTUALIZA_SOFTWARE
USP_ACTUALIZA_DESARMA_EQUIPO	USP_ADMIN_ELIMINA_AUTOACCESO
USP_ACTUALIZA_HARDWARE_PIEZA	USP_ADMIN_ELIMINA_HARDWARE_DESECHADO
USP_ACTUALIZA_INCIDENTE_REVISADO	USP_ADMIN_ELIMINA_INCIDENTES
USP_ACTUALIZA_INSTALACION_PIEZA	USP_ADMIN_ELIMINA_SOFTWARE_OBSOLETO
USP_ACTUALIZA_NOMBRE_MODULO	USP_AUTENTIFICA_PERSONAL
USP_ACTUALIZA_PERSONAL	USP_AUTOACCESO_EN_MODULO_N
USP_GET_SOFTWARE_OBSOLETO	USP_INSERTA_RESPONSABLE
USP_GET_SOFTWARE_SEGURIDAD	USP_INSERTA_SOFTWARE
USP_GET_SOFTWARE_SEGURIDAD_M1	USP_INSERTA_USUARIO
USP_GET_SOFTWARE_SEGURIDAD_M2	USP_INSTALLA_SOFTWARE_EN_EQUIPO
USP_GET_SOFTWARE_SEGURIDAD_M3	USP_PIEZA_DENTRO_DE_EQUIPO
USP_GET_SOFTWARE_SEGURIDAD_M4	USP_REGISTRAR_ALMACEN
USP_INSERTA_HARDWARE	USP_REGISTRAR_HARDWARE_MODULO
USP_INSERTA_INCIDENTE	USP_SOFTWARE_EXISTE_EN_ALMACEN
USP_INSERTA_NUEVO_MODULO	USP_TOTAL_HARDWARE_MODULO
USP_INSERTA_PERSONAL	

Figura 4.1.9. Breve vista a los procedimientos almacenados

## Diseño de la interfaz

El sistema posee una plantilla base programada en HTML y CSS, y todas las páginas son basadas en una plantilla (Figura 4.1.10) .

Se utilizan tres tipografías base: “Franklin Gothic Medium”, “Verdana” y “Courier”, con el objetivo de que el sistema pueda visualizarse en cualquier dispositivo desde el que se quiera acceder y la información se presente con respecto al diseño.



**Figura 4.1.10. Plantilla de diseño base**

Mientras que solo se manejan 5 colores:



R: 6 G: 79 B: 127

Color Base para los contenedores de elementos de control y el logotipo.



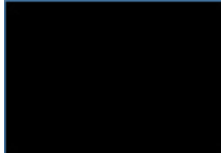
R: 0 G: 17 B: 167

Color Base para los hipervínculos no visitados.

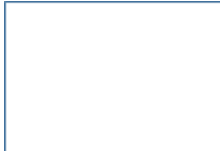


R: 85 G: 26 B: 138

Color Base para los hipervínculos visitados.



R: 0 G: 0 B: 0  
Color Base para el texto en general.



R: 255 G: 255 B: 255  
Color Base para la sección de encabezado y fondo de los contenidos.

## 4.2 IMPLEMENTACIÓN DE LA INTERFAZ

### 4.2.1 *Página de inicio*

Cualquier usuario del sistema será recibido por la página de autenticación en el cual deberá iniciar una sesión escribiendo su nombre de usuario y contraseña.

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación  
SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO

ID usuario:   
Contraseña:

Figura 4.2.1.1. Página de inicio del sistema.

Una vez autenticados, la pantalla inicial los recibirá con un breve mensaje y los guiará hacia las secciones que pueden utilizar. Para cada usuario la interfaz de inicio es diferente, la figura 4.2.1.2 es para el colaborador, 4.2.1.3 para el coordinador y 4.2.1.4 para el administrador.



**Figura 4.2.1.2. Autenticación y bienvenida del usuario colaborador**

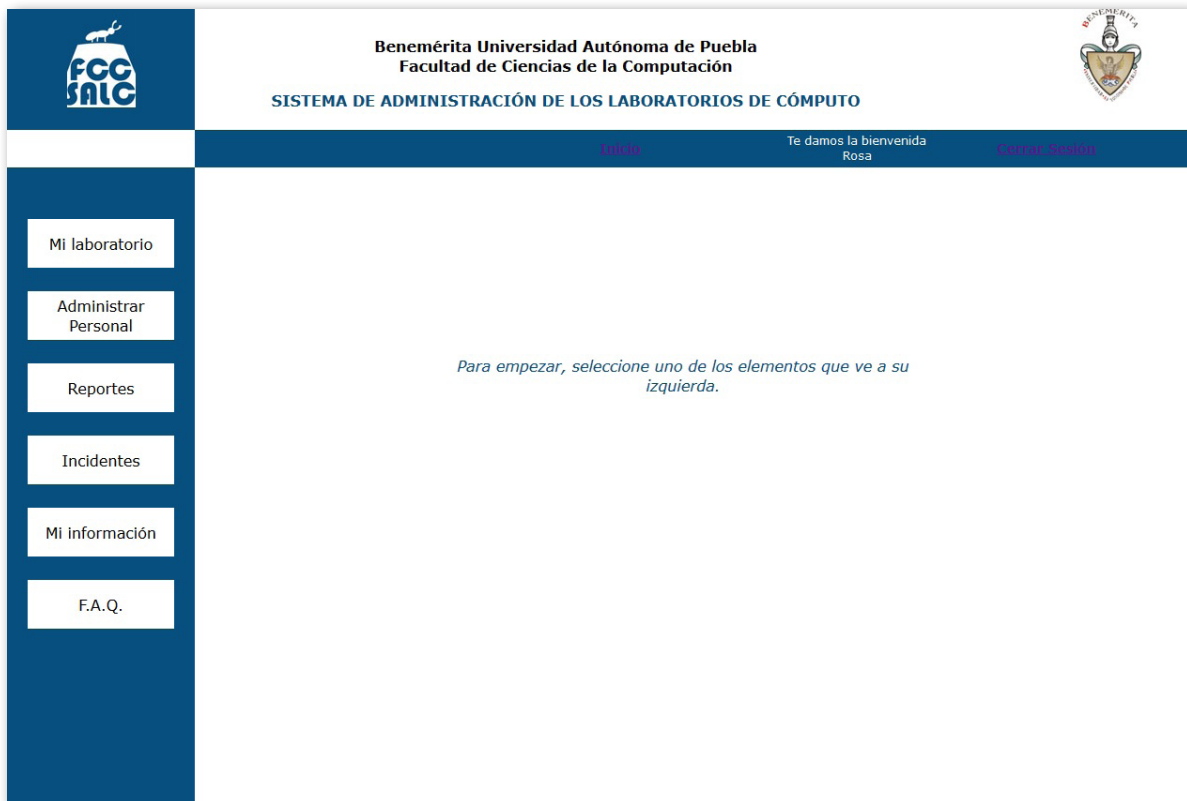


Figura 4.2.1.3. Autenticación y bienvenida al usuario coordinador

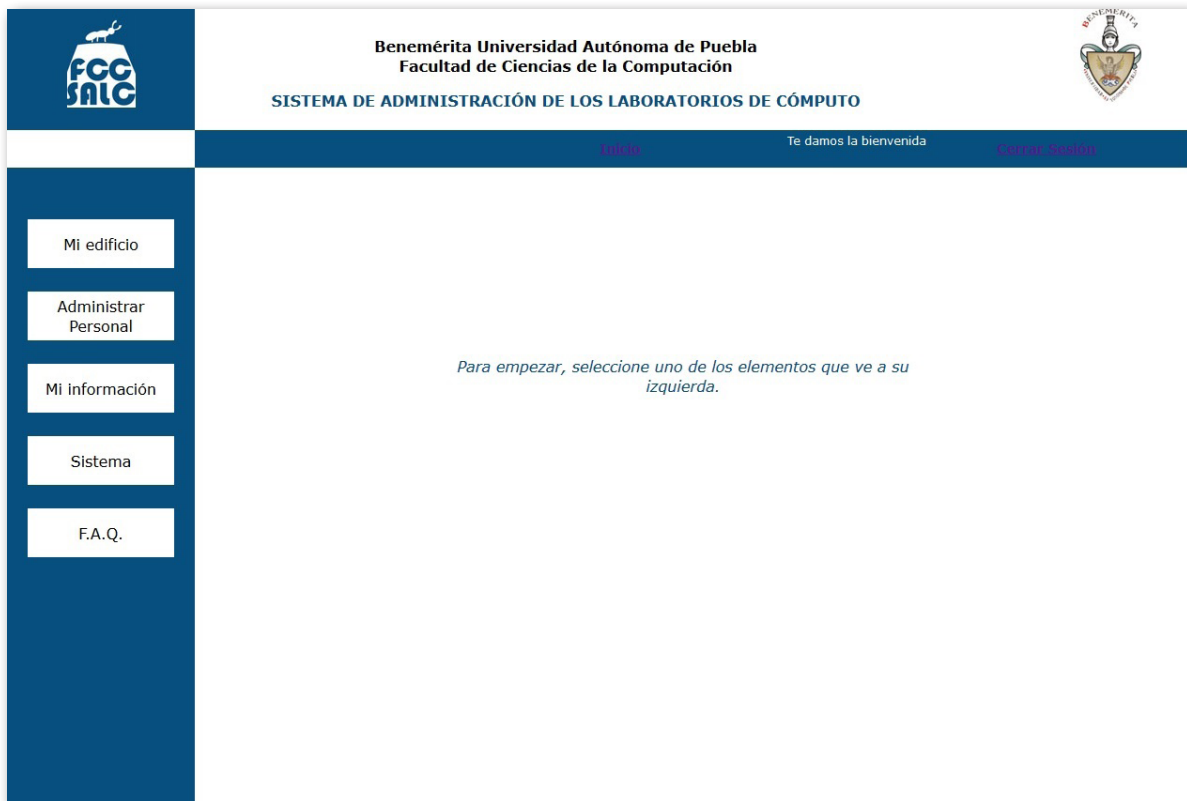


Figura 4.2.1.4. Autenticación y bienvenida al usuario administrador

### **4.2.2 Interfaces de usuario**

Un usuario del sistema debe existir en la base de datos y debe ser dado de alta por algún usuario con el privilegio de hacerlo. Este usuario solo podrá hacer uso del sistema iniciando una sesión con los datos que se le proporcionen.

El sistema se divide en 3 tipos de usuario: colaborador, coordinador y administrador, cada uno con privilegios y acciones diferentes dentro del sistema. Se cuenta un total de 36 interfaces y muchas acciones para cada usuario.

Como medida de seguridad para robo de contraseñas se implementó un módulo de encriptación MD5, de manera que estas viajen seguras y además cada entrada de datos a la interfaz es revisado con métodos de PHP para intentar evitar la inyección de SQL.

Además con el objetivo de conservar la integridad y eficiencia de la base de datos, cada información que es recibida, se decodifica UTF-8 transformándolo en ISO-8859-1 y así conservar los signos de acentuación y la letra "ñ". La salida de datos es recodificada a UTF-8 para su interpretación por el navegador.

### **4.2.3 Interfaz del colaborador**

*Colaborador.* Puede acceder a 21 interfaces y cuenta con permisos para ingresar nuevos datos y modificar algunos.

Las funciones principales del colaborador es:

- Registrar nuevo hardware y software.
- Responsabilizarse de equipos de computo.
- Modificar datos de software.
- Registrar instalaciones de software y hardware en los equipos.
- Crear reportes de incidentes.
- Crear entradas y salidas de autoacceso.
- Generar reportes de hardware, software y autoacceso.
- Cambiar la contraseña de su cuenta.

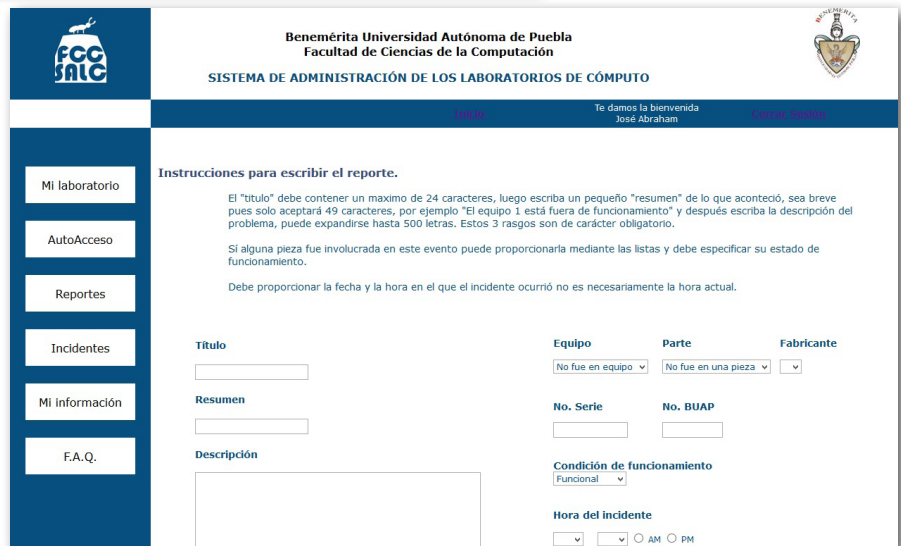


Figura 4.2.3.1. Interfaces principales del usuario colaborador.

#### **4.2.4 Interfaz del coordinador**

*Coordinador.* Puede acceder a 25 interfaces, algunas las comparte con los colaboradores y puede ingresar nuevos datos, modificar y eliminar algunos, tiene permisos de registro de usuarios tipo colaborador (vease la figura 4.2.4.1).

Sus funciones son más de tipo supervisión con las tareas de:

- Registrar, actualizar y excluir hardware.
- Registrar, actualizar y dar de baja colaboradores para su laboratorio.
- Generar reportes de software, hardware e incidentes.
- Visualizar incidentes y eliminarlos a voluntad.
- Actualizar su contraseña.

#### **4.2.5 Interfaz del administrador**

*Administrador.* Tiene acceso a 16 interfaces independientes de las que tiene acceso un colaborador o un coordinador, dentro del sistema. Los permisos de este usuario son de ingreso, modificación y eliminación de datos, además extracción y restauración de información y por ultimo puede hacer registro de usuarios tipo coordinador (vease la figura 4.2.4.2).

Sus tareas principales son:

- Incluir nuevos modulos y eliminarlos.
- Registrar, actualizar y excluir coordinadores.
- Asignar coordinadores a los módulos.
- Hacer respaldos, eliminaciones y restauraciones en la base de datos.
- Modificar completamente toda la informacion propia.

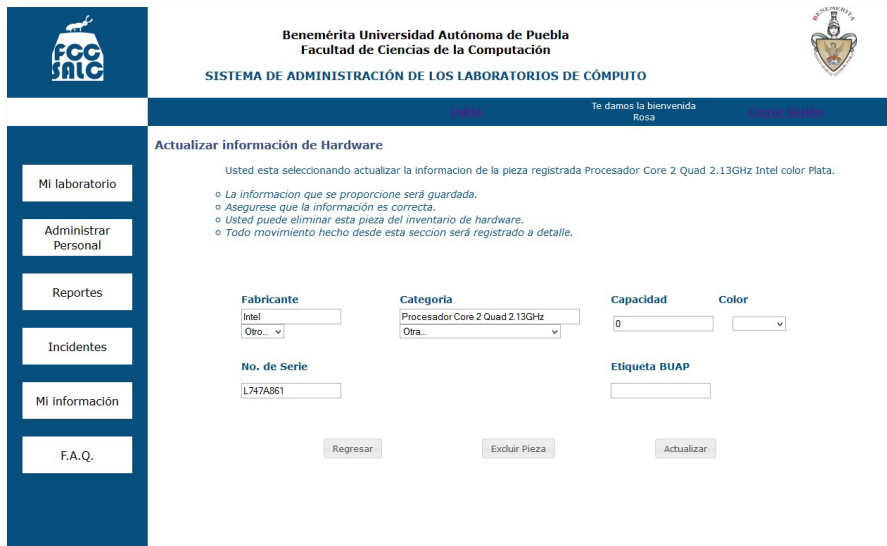


Figura 4.2.4.1. Principales interfaces del usuario coordinador



Figura 4.2.4.2. Principales interfaces del administrador

### 4.3 Pruebas del sistema

En este punto se colocará el sistema en su base de operación y se debe asegurar que el sistema cumpla con los requisitos, es decir lo que el cliente quiere y necesita. Durante las pruebas del sistema es posible encontrar fallos, nuevos problemas que aun no se han resuelto o son consecuencia de la implementación de nuevos componenetes.

Las pruebas se llevarán a cabo con las funciones de mas interaccion que posee el sistema, se usarán de base los escenarios de casos de uso que se presentaron en la sección 2.7.4.

Un caso de prueba que especifica como probar un caso de uso o un escenario especifico de un caso de uso. Un caso de prueba de este tipo incluye la verificación del resultado de la interaccion entre los actores y el sistema, que se satisfacen las precondiciones y postcondiciones especificadas por el caso de uso y que sigue la secuencia de acciones especificadas por el caso de uso (Jacobson et al., 2000) (p. 281).

#### 4.3.1. Prueba 1. Instalación del sistema

The screenshot shows a web-based installation form for the 'SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO' at the Benemérita Universidad Autónoma de Puebla. The form is titled 'Bienvenido a SALC-FCC.' and provides instructions for configuration. It is divided into two main sections:

- 1. Elija su nombre de administrador y contraseña.** This section contains three input fields: 'Nombre de usuario:' with the value 'Administrador', 'Contraseña:' with masked characters, and 'Repita su contraseña:' also with masked characters. Each field has an 'OK' button to its right.
- 2. Proporcione su información de contacto.** This section includes several input fields, some marked with a red asterisk to indicate they are mandatory:
  - 'Nombre' (José Abraham)
  - 'Apellido paterno' (Baez)
  - 'Apellido materno' (Bagatella)
  - 'Número de identificación' (200514799)
  - 'Direccion de correo electronico' (icc\_bagatella@outlook.com)
  - 'Teléfono de casa' (2222224133)
  - 'Teléfono Personal' (2223410522)
  - 'Horario de atención.(Max 60 caracteres)' (Lunes a Viernes de 13:00 a 19:00)
  - 'Nivel de estudios.(Carrera Profesional)' (Ingeniería en Ciencias de la Computación)

At the bottom of the form is a button labeled 'Instalar'.

Figura 4.3.1.1. Recopilando datos del administrador.

Antes de la instalación del sistema en el servidor puede o no existir la base de datos creada, en esta prueba la base de datos existe, más está vacía.

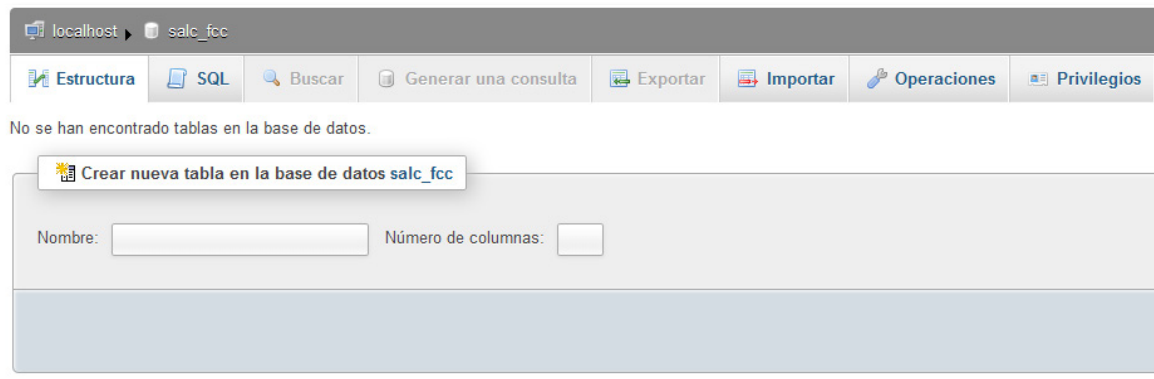


Figura 4.3.1.2. La base de datos esta vacia.

Una vez presionando el botón Instalar luego de agregados los datos que se piden, el sistema validará que todos los datos entren en forma consistente y luego procederá a hacer un “query” en el sistema lo cual creará las tablas de la base de datos, importará los procedimientos almacenados e insertará el registro del administrador mediante una procedimiento almacenado.

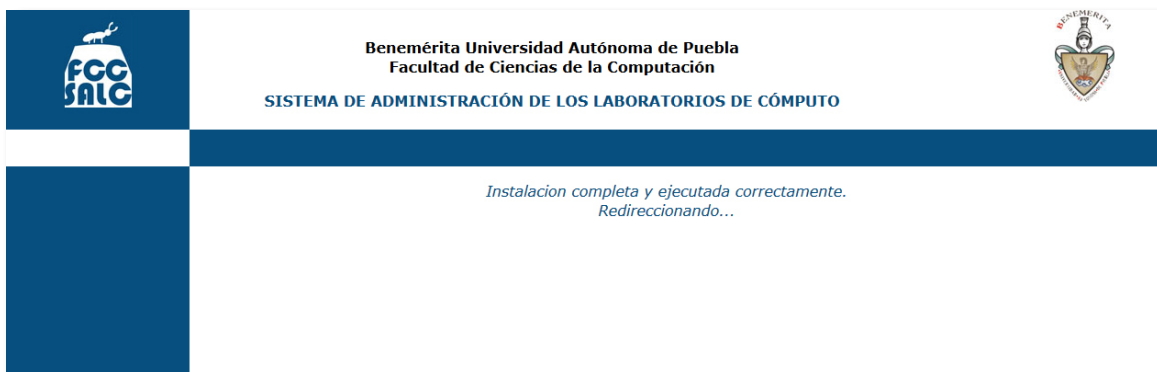


Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño
almacen_software	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32.0 KB
autoacceso	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	48.0 KB
incidentes	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	48.0 KB
personal	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8_general_ci	16.0 KB
recursos	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	48.0 KB
responsable	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32.0 KB
software_equipo	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	32.0 KB
usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	16.0 KB
<b>8 tablas</b>	<b>Número de filas</b>	<b>1</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>272.0 KB</b>

Figura 4.3.1.3. Se crea la base de datos sí no existe y se crean las tablas.

Nombre			Tipo	Muestre el tipo
USP_ACTUALIZA_ALMACEN			PROCEDURE	
USP_ACTUALIZA_BAJA_EQUIPO			PROCEDURE	
USP_ACTUALIZA_BAJA_HARDWARE			PROCEDURE	
USP_ACTUALIZA_BAJA_PERSONAL			PROCEDURE	
USP_ACTUALIZA_DESARMA_EQUIPO			PROCEDURE	
USP_ACTUALIZA_HARDWARE_PIEZA			PROCEDURE	
USP_ACTUALIZA_INCIDENTE_REVISADO			PROCEDURE	
USP_ACTUALIZA_INSTALACION_PIEZA			PROCEDURE	
USP_ACTUALIZA_NOMBRE_MODULO			PROCEDURE	
USP_ACTUALIZA_PERSONAL			PROCEDURE	
USP_ACTUALIZA_PERSONAL_PASS			PROCEDURE	
USP_ACTUALIZA_RECURSO_ESTADO			PROCEDURE	
USP_ACTUALIZA_SALIDA_AUTOACCESO			PROCEDURE	
USP_ACTUALIZA_SOFTWARE			PROCEDURE	
USP_ADMIN_ELIMINA_AUTOACCESO			PROCEDURE	
USP_ADMIN_ELIMINA_HARDWARE_DESECHADO			PROCEDURE	
USP_ADMIN_ELIMINA_INCIDENTES			PROCEDURE	
USP_ADMIN_ELIMINA_SOFTWARE_OBSOLETO			PROCEDURE	
USP_AUTENTIFICA_PERSONAL			PROCEDURE	
USP_AUTOACCESO_EN_MODULO_N			PROCEDURE	
USP_BUSCA_SOFTWARE			PROCEDURE	
USP_CHECK_SOFTWARE_INSTALADO			PROCEDURE	
USP_DESINSTALA_SOFTWARE_EQUIPO			PROCEDURE	
USP_ELIMINA_RESPONSABLE_EQUIPO			PROCEDURE	
USP_ELIMINA_RESPONSABLE_MODULO			PROCEDURE	
USP_ELIMINA_RESPONSABLE_RECURSO			PROCEDURE	
USP_ELIMINA_SOFTWARE			PROCEDURE	
USP_ENTRADA_AUTOACCESO			PROCEDURE	

login	contrasena	no_identificacion	nombre	a_paterno	a_materno	correo_electronico	telefono_casa	telefono_persona
Administrador	ff7a0e56e67ead89bf4f0728f20b8faa	200514799	José Abraham	Baez	Bagatella	icc_bagatella@outlook.com	(222)2-22-4133	(044)2223-41-0522

**Figura 4.3.1.4 Creación de los procedimientos almacenados e inserción del administrador.**

### **4.3.2 Prueba 2. Agregar usuarios al sistema**

Administrar personal permite el agregado de nuevos coordinadores y poder consultar o eliminar la información de todo el personal registrado en el sistema. Se procederá a agregar un nuevo coordinador, el formulario se llena como se aprecia en la imagen 4.3.2.1

### Proporcione los datos del nuevo Coordinador

Los datos incluidos en esta sección se utilizarán para que el coordinador pueda ingresar al sistema.  
Los campos marcados con \* son obligatorios.

**Nombre de Usuario:**  \* El nombre de usuario esta disponible

**Contraseña:**  \* OK

**Confirmar contraseña:**  \* OK  
 El coordinador podrá cambiar su contraseña posteriormente

Ingrese los datos personales del coordinador , los cuales serviran con el proposito de datos de contacto.

**Nombre**  \*    
 **Apellido Paterno**  \*    
 **Apellido Materno**  \*    
 **Número de trabajador**  \*

Correcto

**Correo electronico**  \* OK    
 **Telefono Casa**     
 **Telefono Personal**  \*

OK

**Horario**     
**Módulo a Coordinar**

**Figura 4.3.2.1. Formulario de registro para la nueva coordinadora.**

Al momento de presionar el botón de registro, se realiza una nueva entrada en la base de datos.

login	contrasena	no_identificacion	nombre	a_paterno	a_materno	correo_electronico	telefono_cas
Administrador	ff7a0e56e67ead89bf4f0728f20b8faa	200514799	José Abraham	Baez	Bagatella	icc_bagatella@outlook.com	(222)2-22-413
RGTamayo	cb860a88452cf25c1ca5b7e96d3a73bc	200000001	Rosa	García	Tamayo	rgtamayo@cs.buap.mx	



**Benemérita Universidad Autónoma de Puebla**  
Facultad de Ciencias de la Computación

**SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO**



Inicio     Te damos la bienvenida José Abraham     Central Auxiliar

*El usuario RGTamayo ha sido registrado exitosamente. Usted será redirigido en 3 segundos.*

**Figura 4.3.2.2. El nuevo coordinador se ha registrado.**

### 4.3.3 Prueba 3. Registrar nuevo módulo

El sistema debe hacer la inserción de los nuevos módulos como un recurso físico de la base de datos mediante un procedimiento almacenado.

The screenshot shows the Benemérita Universidad Autónoma de Puebla Facultad de Ciencias de la Computación SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO. The interface includes a header with the university logo and a navigation bar with links for Inicio, Te damos la bienvenida José Abraham, and Contrar Sesión. The main content area contains instructions for registering a new module and a form with a text input field labeled 'Nombre:' containing '104C - 101' and an 'Agregar Módulo' button.

Figura 4.3.3.1. Asignar un nombre al nuevo módulo.

Se agrega el módulo y se le pone un nombre adecuado como el número de edificio y salón, después se el botón “Agregar módulo” agregará esta información a la base de datos y luego la mostrara en pantalla.

```
CALL USP_INSERTA_NUEVO_MODULO();
```

id_recurso	login	fabricante	descripcion	version	no_serie	inventario_BUAP	capacidad	estado	color	instalado_en
1	Administrador	104C - 101	Modulo	NULL	01	NULL	NULL	NULL	NULL	NUL

The screenshot shows the Benemérita Universidad Autónoma de Puebla Facultad de Ciencias de la Computación SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO. The interface includes a header with the university logo and a navigation bar with links for Inicio, Te damos la bienvenida José Abraham, and Contrar Sesión. The main content area displays a message: 'El nuevo modulo ha sido registrado. Será redirigido en 3 segundos.'

Figura 4.3.3.2. El módulo se registra en la base de datos.

Se pueden realizar distintas operaciones en cada uno de los módulos agregados (vease la figura 4.3.3.3).

<b>Módulo 01</b> <b>104C - 101</b>
Coordinador
Registrado el Viernes 15 de Agosto de 2014
Acciones:
Eliminar Módulo
Asigna Coordinador
Remover asignación
Cambiar Nombre

Figura 4.3.3.3. El módulo posee distintas operaciones.

Ahora se asignara un coordinador al módulo 1, previamente ya se ha registrado a uno. Oprimiendo el botón “Asigna coordinador” se podrá elegir un coordinador activo para el modulo.

<b>Elija un Coordinador:</b>	
<input checked="" type="checkbox"/>	Rosa García Tamayo
Aceptar	Cancelar

Figura 4.3.3.3. El módulo posee distintas operaciones.

Al marcar y aceptar al coordinador, se crea un nuevo registro en la entidad “Responsable” para señalar la relación entre el coordinador y el modulo.

	login	id_recurso
<input type="checkbox"/> Editar	RGTamayo	1

Figura 4.3.3.4. Se escribe un registro para crear la relación entre el recurso y el personal

#### 4.3.4. Prueba 4. Autenticación y acceso seguro

La autenticación de los usuarios es la pieza de arranque del sistema, la página índice preguntará a un usuario su ID y su contraseña, se probará ingresar con el coordinador que se acaba de registrar.

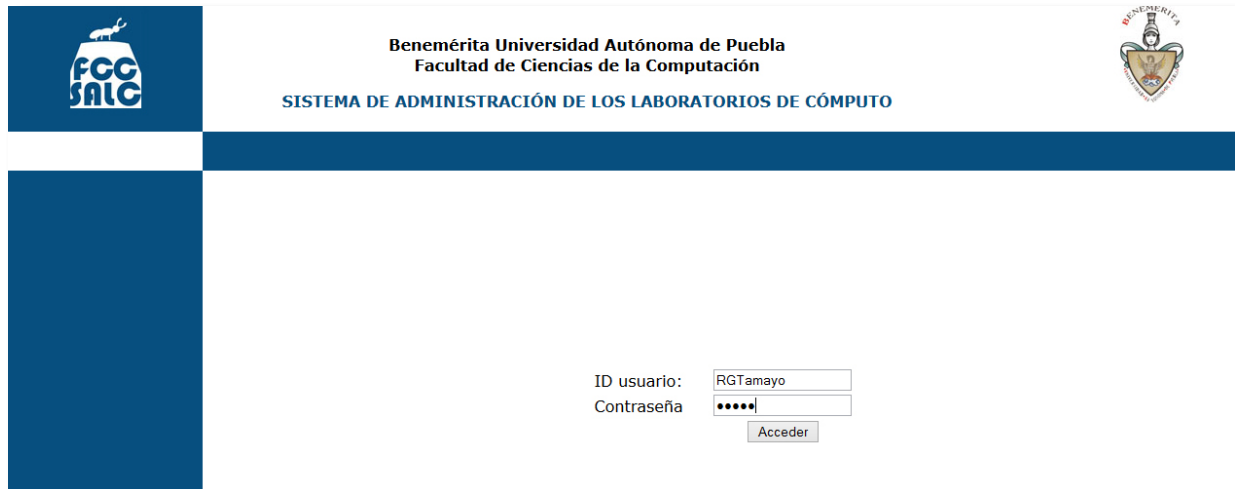


Figura 4.3.4.1. Pantalla de inicio de sesión.

El sistema por medio de un procedimiento almacenado obtiene algunos datos de la base para comparar la información y en caso de que sea correcta, se guarden algunos en variables de sesión.

```
CALL USP_AUTENTIFICA_PERSONAL ('RGTamayo');
```

login	contrasena	tipo	nombre
RGTamayo	cb860a88452cf25c1ca5b7e96d3a73bc	2	Rosa

Figura 4.3.4.2. Procedimiento y datos recuperados de la base.

El sistema comprueba que el usuario exista, luego comprueba que la contraseña coincida la cual ha sido encriptada mediante MD5. Si ambas pruebas son satisfactorias se inicia la sesión y se toman los datos de login, tipo y nombre para las variables de sesión que persistirán hasta cerrarla.

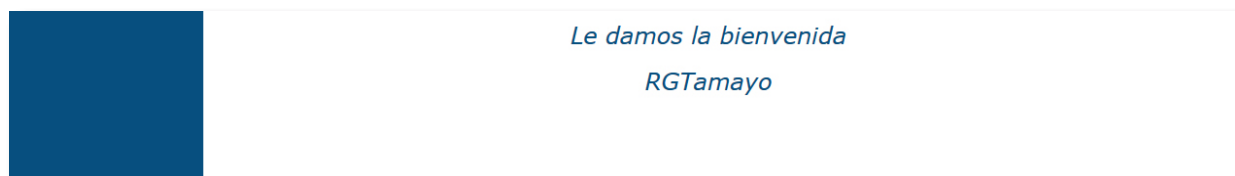


Figura 4.3.4.3. Mensaje de bienvenida.

En caso contrario, se desechará el intento de inicio de sesión y se informará que algún dato es incorrecto.

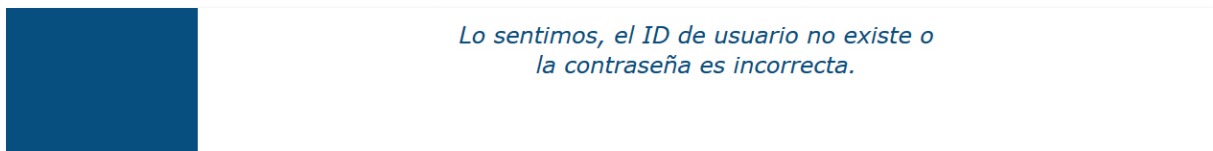


Figura 4.3.4.4. Mensaje de fallo al iniciar sesión.

### 4.3.5. Prueba 5. Manipulación del inventario de hardware

El inventario de hardware muestra una lista del hardware registrado previamente en el laboratorio mediante un procedimiento almacenado.

Mi laboratorio

---

AutoAcceso

---

Reportes

---

Incidentes

---

Mi información

---

F.A.Q.

**Inventario de hardware**

Se muestra una lista con todas las piezas de hardware individuales que ha registrado hasta el momento y que no han sido dadas de baja por el coordinador.

- o Si desea editar la información de una pieza listada, haga un clic en el nombre de la pieza que desea editar su información.
- o Si desea conocer mas acerca de una pieza, marque una casilla de verificación y haga clic en el botón "Mas información".
- o Si desea registrar una nueva pieza de hardware, haga un clic sobre la imagen con la leyenda "Registrar Nuevo".
- o Si desea registrar un nuevo equipo de cómputo en el laboratorio o tomar la responsabilidad sobre alguno, haga clic en la imagen con la leyenda "Agregar Equipo".
- o Si desea registrar el armado de un equipo pieza a pieza, haga clic en la imagen con la leyenda "Ver armado de equipos".

Instalar	Pieza	No. Serie	BUAP	Condición
<input type="checkbox"/>	Procesador Core i5-4670k Intel Plata	46214EA631DF45		Funcional
<input type="checkbox"/>	Procesador Core i5-4440k Intel Plata	1000ACB410299		Funcional
<input type="checkbox"/>	Procesador Core 2 Quad Intel Plata	100322AE22440		Funcional
<input type="checkbox"/>	Procesador Core i3-3220 Intel Plata	66554E10478A2		Funcional
<input type="checkbox"/>	Procesador Core 2 Quad Intel Plata	L8458724		Funcional
<input type="checkbox"/>	Procesador Core i5-3330 Intel Plata	7A324651E340		Funcional
<input type="checkbox"/>	Procesador Core 2 Duo E8400 Intel Plata	Q98JA251		Funcional
<input type="checkbox"/>	Procesador Core i5-3330 Intel Plata	L229B897		Funcional
<input type="checkbox"/>	Memoria RAM DDR2 667MHz Kingston Technology Verde 2	KVR667D2N5	400146564	Funcional
<input type="checkbox"/>	Memoria RAM DDR2 533MHz Kingston Technology Verde 2	KVR533D4N4/2G	400165764	Funcional

Página 1

Registrar Nuevo

Agregar Equipo

Ver armado de equipos

Figura 4.3.5.1. Inventario de hardware.

Mediante la acción "Registrar nuevo" se mostrará un formulario para ingresar datos acerca de una nueva pieza de hardware presente en el laboratorio, los cuales se muestran en la figura 4.3.5.2.

**Registrar un nuevo hardware.**

Para registrar una nueva pieza de hardware debe proporcionar al menos 3 datos obligatorios los cuales estan marcados con un asterisco (\*).

- o Visualice el fabricante de la pieza que va a registrar, si esta en la lista seleccíonelo, si no lo esta, seleccione la opción "Otro..." y escríbalo.
- o La categoría se refiere a el tipo de pieza que quiere registrar, por ejemplo teclado, mouse, monitor, etc.
- o Seleccione también la condición de funcionamiento que tiene esta pieza en este momento.
- o El campo capacidad solo uselo cuando registre piezas que funcionen como almacenamiento de datos. Por ejemplo discos duros, el numero se interpretará como Gigabytes (GB)

**Fabricante\*** 
**Categoría\*** 
**Capacidad** 
**Color**

**No. Serie** 
**Etiqueta BUAP** 
**Condición\***

**Figura 4.3.5.1. Registro de una nueva pieza.**

Al presionar "Registrar" estos datos crean un registro en la tabla de recursos del sistema para almacenar la presencia y condición de esta pieza, también se verá listada en las piezas que pertenecen al laboratorio.

ID	Fabricante	Categoría	Modelo	No. Serie	Capacidad	Color	Condición	Plata	ID
24	Seagate	Disco Duro SATA 3.5	471S9700014	320141042		320	Funcional	Plata	20
25	Western Digital	Disco Duro SATA 3.5	WD1600AAJS	320141084		320	Funcional	Plata	20
26	Western Digital	Disco Duro SATA 3.5	WD3200AAJS	320142000		320	Funcional	Plata	20
27	Seagate	Disco Duro SATA 3.5	ST3320418AS	320142001		320	Funcional	Plata	20
28	Western Digital	Disco Duro SATA 3.5	WD3200AAJS	320144015		320	Funcional	Plata	20
29	Western Digital	Disco Duro SATA 3.5	WD3200AAJS	320142015		320	Funcional	Plata	20
30	Western Digital	Disco Duro SATA 3.5	WD3200AAJS	320141050		320	Funcional	Plata	20
31	Kingston Technology	Memoria RAM DDR2 667MHz	KVR533D4N4/4G	400451271		4	Funcional	Verde	20

**Inventario de hardware**

Se muestra una lista con todas las piezas de hardware individuales que ha registrado hasta el momento y que no han sido dadas de baja por el coordinador.

- o Si desea editar la información de una pieza listada, haga un clic en el nombre de la pieza que desee editar su información.
- o Si desea conocer mas acerca de una pieza, marque una casilla de verificación y haga clic en el botón "Mas información".
- o Si desea registrar una nueva pieza de hardware, haga un clic sobre la imagen con la leyenda "Registrar Nuevo".
- o Si desea registrar un nuevo equipo de cómputo en el laboratorio o tomar la responsabilidad sobre alguno, haga clic en la imagen con la leyenda "Agregar Equipo".
- o Si desea registrar el armado de un equipo pieza a pieza, haga clic en la imagen con la leyenda "Ver armado de equipos".

Instalar	Pieza	No. Serie	BUAP	Condición
<input type="checkbox"/>	Disco Duro SATA 3.5 Western Digital Plata 320 GB	WD3200AAJS	320142015	Funcional
<input type="checkbox"/>	Disco Duro SATA 3.5 Western Digital Plata 320 GB	WD3200AAJS	320141050	Funcional
<input checked="" type="checkbox"/>	Memoria RAM DDR2 667MHz Kingston Technology Verde 4 GB	KVR533D4N4/4G	400451271	Funcional

Mas información Página 3

**Figura 4.3.5.2. La nueva pieza se almacena en la base de datos y se refleja en la interfaz.**

También es posible visualizar que equipos están dados de alta en el módulo. Presionando el enlace a "Agregar equipo" se listarán los equipos actualmente registrados mediante un procedimiento almacenado. Y es posible agregar nuevos equipos presionando algún botón "Agregar equipo".

```
CALL USP GET EQUIPOS MODULO(1);
```

id_recurso	no_serie	estado
2	01	funcional
3	02	funcional
4	03	funcional

Figura 4.3.5.3. Listado de equipos disponibles

Después de que se confirme la entrada de este, el sistema hará un registro en la base de datos con la información del nuevo equipo de cómputo (figura 4.3.5.4).

id_recurso	no_serie	estado
2	01	funcional
3	02	funcional
4	03	funcional
5	04	funcional
6	05	funcional
7	06	funcional
8	07	funcional
32	10	funcional

Figura 4.3.5.4. El nuevo equipo y su registro en la base de datos

Y por último también es posible revisar en que equipo están colocadas las piezas de hardware registradas. En la sección llamada “Armado de equipos” se listan los equipos, las piezas instaladas y las refacciones, es decir las piezas que no están montadas en alguna computadora. Mediante la combinación de 3 procedimientos almacenados se logra una vista completa de esta sección.

**Figura 4.3.5.5. Sección “Armado de Equipos” y los procedimientos que logran la visualización.**

La instalación o desinstalación de las piezas son reflejadas en la base de datos, la figura 4.3.5.6 muestra los componentes del equipo 1 mediante el campo “instalado\_en” (puede verse la figura 3.1.4.1 para entender la función de este campo).

id_recurso	fabricante	descripcion	capacidad	instalado_en
15	Intel	Procesador Core 2 Duo E8400	NULL	2
18	Kingston Technology	Memoria RAM DDR2 533MHz	2	2
30	Western Digital	Disco Duro SATA 3.5	320	2

**Figura 4.3.5.6. Vista en la base de datos de las piezas instaladas en el equipo 1**

### 4.3.6. Prueba 6. Trabajando con “Autoacceso”

La sección de autoacceso permite registrar una entrada al laboratorio por parte de un alumno mientras no hay sesión de clases. Se captura la matricula, nombre y apellidos de alumno, y para ofrecer más control sobre los equipos, se asigna que equipo de cómputo utilizará esa persona, para probar esto se ingresarán los datos que se aprecian en la figura 4.3.6.1.

**Benemérita Universidad Autónoma de Puebla**  
Facultad de Ciencias de la Computación

**SISTEMA DE ADMINISTRACIÓN DE LOS LABORATORIOS DE CÓMPUTO**

Te damos la bienvenida  
José Abraham

**Registro de Autoacceso**

Para realizar el registro de entrada solicite al alumno su credencial de estudiante vigente y llene los datos que se piden, todos son obligatorios.

- o Escriba la matrícula del alumno, luego presione la tecla TAB o haga clic en el campo "Nombre", si el alumno ha utilizado las instalaciones anteriormente, automáticamente aparecerán sus datos, si no lo ha hecho, rellene cuidadosamente sus datos.
- o Escriba el nombre o nombres del alumno.
- o Escriba su apellido paterno.
- o Escriba su apellido materno.
- o Seleccione el equipo que vaya a asignar su uso al alumno e informeselo.
- o Finalmente presione el botón "Entrada", si los datos se han escrito de manera válida se creará el registro. Devuelva su credencial al alumno.

**Matrícula** **Nombre** **Apellido Paterno** **Apellido Materno** **Equipo**  
 201004998 Carolina Rosas Cruz 01

**Entrada**

**Consulta y salida de Autoacceso**

Figura 4.3.6.1. Datos de prueba en la interfaz de autoacceso.

Una vez pulsado el botón “Entrada” el sistema creará el registro de usuario y el registro de autoacceso en la base de datos, además mostrará en pantalla que el equipo está siendo utilizado.

matricula	nombre	apellido_p	apellido_m	id_autoacceso	fecha_registro	hora_entrada	hora_salida	id_recurso	matricula
200841147	Paulina	Rodrigo	Bolaños	28	2014-08-18	12:59:08	15:01:21	32	201107941
200913320	Israel	Badillo	Ballinas	29	2014-08-18	13:37:00	15:01:33	3	201287644
200914544	Rodolfo	Gutierrez	Perez	30	2014-08-18	13:41:38	15:01:34	5	201014785
200925613	Alejandro	Sandoval	Herrera	31	2014-08-20	23:02:30	23:11:47	2	201004998
200931554	Magali	Uceda	Sanchez	32	2014-08-20	23:11:34	23:11:45	3	200841147
201004998	Carolina	Rosas	Cruz	33	2014-08-20	19:11:59	NULL	2	201004998

**Benemérita Universidad Autónoma de Puebla**  
Facultad de Ciencias de la Computación

Te damos la bienvenida  
José Abraham

**Consulta y salida de Autoacceso**

Se muestra una lista con los datos de los alumnos que se han registrado para utilizar las instalaciones. Si lo desea puede registrar la salida del alumno seleccionando la casilla de verificación a la izquierda de cada nombre y pulsando el botón "Salida".

Equipo asignado	Matrícula	Nombre	Hora de entrada
01	201004998	Carolina Rosas Cruz	19:11:59

Salida

Pagina 1

Figura 4.3.6.2. Se registra el acceso en la base de datos y se refleja al usuario en la interfaz.

Es posible registrar tantos accesos como equipos de cómputo posea el laboratorio, para anunciar una salida el alumno la reporta con el colaborador encargado, selecciona su nombre y presiona el botón de salida.

**Consulta y salida de Autoacceso**

Se muestra una lista con los datos de los alumnos que se han registrado para utilizar las instalaciones.  
Si lo desea registrar la salida del alumno seleccionando la casilla de verificación a la izquierda de cada nombre y pulsando el botón "Salida".

	Equipo asignado	Matricula	Nombre	Hora de entrada
<input checked="" type="checkbox"/>	01	201004998	Carolina Rosas Cruz	19:11:59
<input type="checkbox"/>	02	200841147	Paulina Rodrigo Bolaños	19:12:58
<input type="checkbox"/>	03	200913320	Israel Badillo Ballinas	19:13:04
<input type="checkbox"/>	04	201241005	Rocío Herrera Valdez	19:19:24
<input type="checkbox"/>	05	200914544	Rodolfo Gutierrez Perez	19:19:39
<input type="checkbox"/>	06	200931554	Magali Uceda Sanchez	19:30:55
<input type="checkbox"/>	07	201014410	Marcos Hernandez Mendez	19:43:12
<input type="checkbox"/>	10	201044551	Hector Leal Castillo	19:48:30

Salida Pagina 1

**Figura 4.3.6.3. Seleccionando una casilla y oprimiendo el boton "Salida" marcará la salida del usuario.**

El sistema actualizará el registro de autoacceso guardando la hora de salida, no son borrados de la base de datos los registros de autoacceso y de usuario con el motivo de guardar un historial que puede verse reflejado más tarde en un reporte.

id_autoacceso	fecha_registro	hora_entrada	hora_salida	id_recurso	matricula
28	2014-08-18	12:59:08	15:01:21	32	201107941
29	2014-08-18	13:37:00	15:01:33	3	201287644
30	2014-08-18	13:41:38	15:01:34	5	201014785
31	2014-08-20	23:02:30	23:11:47	2	201004998
32	2014-08-20	23:11:34	23:11:45	3	200841147
33	2014-08-20	19:11:59	20:45:59	2	201004998

**Figura 4.3.6.3. Los registros de autoacceso no se eliminan, solo actualizan su estado.**

### 4.3.7. Prueba 7. Generar reportes

El generador de reportes puede crear 16 diferentes tipos de reportes en 4 categorías (figura 4.3.7.1).

Para crear un reporte del material del laboratorio se accede al apartado "Hardware". Crear un reporte implica una o varias consultas a la base de datos con tal de obtener la información necesaria. El reporte general obtiene un historial de todas las piezas que se han registrado, ya sea que estén instaladas, como refacción o estén averiadas.



Figura 4.3.7.1. Generador de reportes del sistema.

Se pueden especificar rangos de fechas para solo saber que piezas fueron registradas entre el lapso dado. Al presionar el botón generar se obtiene el reporte general solicitado, para el cual se utiliza un solo procedimiento almacenado y la ayuda de la clase de php FPDF.



Figura 4.3.7.2. Generador de reportes de hardware.

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La F de FPDF significa Free (gratis y libre): puede usted usarla para cualquier propósito y modificarla a su gusto para satisfacer sus necesidades. (FPDF, 2014)

Pieza	Fecha de registro y condición
Procesador Core i5-4670k Intel Plata Serie 46214EA631DF45	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core i5-4440k Intel Plata Serie 1000ACB410299	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core 2 Quad Intel Plata Serie 100322AE22440	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core i3-3220 Intel Plata Serie 66554E10478A2	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core 2 Quad Intel Plata Serie L8458724	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core i5-3330 Intel Plata Serie 7A324651E340	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core 2 Duo E8400 Intel Plata Serie Q98JA251	Miercoles 20 de Agosto de 2014 Funcional
Procesador Core i5-3330 Intel Plata Serie L229B897	Miercoles 20 de Agosto de 2014 Funcional
Memoria RAM DDR2 667MHz Kingston Technology 2 GB Verde Serie KVR667D2N5 No. BUAP 400146564	Miercoles 20 de Agosto de 2014 Funcional
Memoria RAM DDR2 533MHz Kingston Technology 2 GB Verde Serie KVR533D4N4/2G No. BUAP 400165764	Miercoles 20 de Agosto de 2014 Funcional

Figura 4.3.7.3. Un reporte general de Hardware.

```
CALL USP_GET_RGH(1,0,'2014-01-01',NOW());
```

descripcion	fabricante	color	
Procesador Core i5-4670k	Intel	Plata	NULL
Procesador Core i5-4440k	Intel	Plata	NULL
Procesador Core 2 Quad	Intel	Plata	NULL
Procesador Core i3-3220	Intel	Plata	NULL
Procesador Core 2 Quad	Intel	Plata	NULL

Figura 4.3.7.4. Procedimiento almacenado generador del reporte

Los reportes más específicos como “Reporte de armado de equipos” utilizan repetidamente un procedimiento almacenado para obtener el reporte.

```
CALL USP_GET_ARMADO_EQUIPO_MODULO('01',1,0,20);
CALL USP_GET_ARMADO_EQUIPO_MODULO('02',1,0,20);
CALL USP_GET_ARMADO_EQUIPO_MODULO('03',1,0,20);
CALL USP_GET_ARMADO_EQUIPO_MODULO('04',1,0,20);
```

Figura 4.3.7.5. El procedimiento almacenado se ejecuta en un ciclo para obtener el reporte.



Fecha: Viernes 22 de Agosto de 2014

Modulo 01

**Armado del equipo 01**

Pieza	Condición
Procesador Core 2 Duo E8400 Intel Plata Serie Q98JA251	Funcional
Memoria RAM DDR2 533MHz Kingston Technology 2 GB Verde Serie KVR533D4N4/2G No. BUAP 400165764	Funcional
Disco Duro SATA 3.5 Western Digital 320 GB Plata Serie WD3200AAJS No. BUAP 320141050	Funcional

**Armado del equipo 02**

Pieza	Condición
Procesador Core i5-4670k Intel Plata Serie 46214EA631DF45	Funcional
Memoria RAM DDR2 667MHz Kingston Technology 2 GB Verde Serie KVR667D2N5 No. BUAP 400146564	Funcional
Disco Duro SATA 3.5 Western Digital 320 GB Plata Serie WD3200AAJS No. BUAP 320142015	Funcional

**Figura 4.3.7.6. Un reporte de armado de equipos.**

### 4.3.8. Prueba 8. Reportar incidentes

Para reportar incidentes existen dos métodos, el reporte general y el referenciado de hardware. Un reporte general consiste en ingresar a la sección “Incidentes” y oprimir el vínculo a “Reportar incidente”. Se carga un formulario con múltiples cajas de texto para ingresar información correspondiente a las posibilidades de un reporte.

Mi laboratorio

---

AutoAcceso

**Historial de incidentes**

Desde esta seccion puede consultar los incidentes que hayan ocurrido y sido reportados en el módulo.

- o Si desea ver el reporte completo, presione una casilla de verificación.
- o Si desea realizar un reporte haga clic en el vinculo "[Reportar Incidente](#)".

**Figura 4.3.8.1. Acceso a un reporte general por medio del historial de incidentes**

**Instrucciones para escribir el reporte.**

El "titulo" debe contener un maximo de 24 caracteres, luego escriba un pequeño "resumen" de lo que aconteció, sea breve pues solo aceptará 49 caracteres, por ejemplo "El equipo 1 está fuera de funcionamiento" y después escriba la descripción del problema, puede expandirse hasta 500 letras. Estos 3 rasgos son de carácter obligatorio.

Si alguna pieza fue involucrada en este evento puede proporcionarla mediante las listas y debe especificar su estado de funcionamiento.

Debe proporcionar la fecha y la hora en el que el incidente ocurrió no es necesariamente la hora actual.

**Título**

**Resumen**

**Descripción**

**Equipo**  **Parte**  **Fabricante**

**No. Serie**  **No. BUAP**

**Condición de funcionamiento**

**Hora del incidente**  
   AM  PM

**Fecha del incidente**

Figura 4.3.8.2 Formulario para la creacion de un reporte de incidente.

Un reporte referenciado de hardware se refiere a especificar una pieza en el inventario y a partir de ella generar el reporte. Se debe acceder al apartado de inventario de hardware, seleccionar una pieza y oprimir el botón “Más información”, luego deberá oprimir el botón “Reportar pieza”.

Excluir	Pieza	No. Serie	BUAP	Condición
<input type="checkbox"/>	Procesador Core i5-4670k Intel Plata	46214EA631DF45		Funcional
<input type="checkbox"/>	Procesador Core i5-4440k Intel Plata	1000ACB410299		Funcional
<input type="checkbox"/>	Procesador Core 2 Quad Intel Plata	100322AE22440		Funcional
<input type="checkbox"/>	Procesador Core i3-3220 Intel Plata	66554E10478A2		Funcional
<input type="checkbox"/>	Procesador Core 2 Quad Intel Plata	L8458724		Funcional
<input type="checkbox"/>	Procesador Core i5-3330 Intel Plata	7A324651E340		Funcional
<input type="checkbox"/>	Procesador Core 2 Duo E8400 Intel Plata	Q98JA251		Funcional
<input type="checkbox"/>	Procesador Core i5-3330 Intel Plata	L229B897		Funcional
<input checked="" type="checkbox"/>	Memoria RAM DDR2 667MHz Kingston Technology Verde 2 GB	KVR667D2N5	400146564	Funcional
<input type="checkbox"/>	Memoria RAM DDR2 533MHz Kingston Technology Verde 2 GB	KVR533D4N4/2G	400165764	Funcional

Mas información Página 1

**Estado actual de la pieza:**

**Esta pieza se encuentra instalada en el equipo 02**

Nombre de la pieza      Memoria RAM DDR2 667MHz Kingston Technology 2 Verde  
 No. de Serie              KVR667D2N5  
 Etiqueta BUAP            400146564  
 Condición                  Funcional

Figura 4.3.8.3. Procedimiento para crear un reporte referenciado de hardware

Un reporte general puede utilizarse como medio de comunicación, reportar una indisciplina o cualquier cosa que no involucre precisamente una pieza registrada en el sistema. Puede introducirse información como la de la figura 4.3.8.4 y presionar el botón “Completar Reporte”.

Mi laboratorio

AutoAcceso

Reportes

Incidentes

Mi información

F.A.Q.

**Instrucciones para escribir el reporte.**

El "titulo" debe contener un maximo de 24 caracteres, luego escriba un pequeño "resumen" de lo que aconteció, sea breve pues solo aceptará 49 caracteres, por ejemplo "El equipo 1 está fuera de funcionamiento" y después escriba la descripción del problema, puede expandirse hasta 500 letras. Estos 3 rasgos son de carácter obligatorio.

Sí alguna pieza fue involucrada en este evento puede proporcionarla mediante las listas y debe especificar su estado de funcionamiento.

Debe proporcionar la fecha y la hora en el que el Incidente ocurrió no es necesariamente la hora actual.

**Título**

**Resumen**

**Descripción**

Me he percatado de que en la tercera fila de computadoras, hay un charco de agua y resulta que en el plafón existe una gotera.

**Equipo**

**Parte**

**Fabricante**

**No. Serie**

**No. BUAP**

**Condición de funcionamiento**

**Hora del incidente**

AM  PM

**Fecha del incidente**

id_incidente	titulo	resumen	descripcion	fecha_incidente	hora_incidente	fecha_registro	hora_registro	id_recurso	login	revisado
1	Gotera	Se ha percibido una gotera en el plafón	Me he percatado de que en la tercera fila de compu...	2014-08-19	17:30:00	2014-08-22	17:35:18	NULL	JAbraham	0

**Figura 4.3.8.4. Creacion, completación y reflejo en la base de datos de un reporte de incidente.**

El sistema validará que los campos cumplan con las restricciones de integridad de la base de datos y luego creará el registro en la tabla de incidentes. Este registro estará presente en la página principal de la sección de incidentes hasta que un coordinador lo actualice como revisado, el cual en dicho caso cambiara un dato del registro que servirá para no visualizarse más, sin embargo este no será eliminado.

**Historial de incidentes**

Desde esta sección puede consultar los incidentes que hayan ocurrido y sido reportados en el módulo.

- o Si desea ver el reporte completo, presione una casilla de verificación.
- o Si desea eliminar un reporte que ya haya revisado, seleccione la casilla de verificación a la derecha del elemento y confírmelo.
- o Si desea realizar un reporte haga clic en el vínculo "[Reportar Incidente](#)".

Reporte	Resumen	Ocurrido	Ver completo	Eliminar
Gotera	Se ha percibido una gotera en el plafón	Martes 19 de Agosto de 2014 a las 05:30:00 PM	<input type="checkbox"/> Ver reporte	<input type="checkbox"/>

Página 1

**Figura 4.3.8.5. Vista de la lista de reportes sin revisar.**

### 4.3.9. Prueba 9. Crear respaldos, eliminar registros y restaurar.

La sección “Sistema” permite crear respaldos de registros, eliminación y restauración. Para crear un respaldo se debe acceder al apartado “Respaldo Registros”. Se probará respaldar, eliminar y recuperar los registros de autoacceso.

**Respaldo Registros del Sistema.**

En esta sección podrá crear respaldos de los registros clave del sistema en un archivo de tipo SQL, se recomienda altamente que no se altere el archivo que se le proporcionará, dado que le servirá para reestablecer dichos registros posteriormente. Elija una opción:

- o Respaldo de la base de datos completa.
- o Respaldo de registros de incidentes.
- o Respaldo de registros de autoacceso.
- o Respaldo de recursos de hardware.
- o Respaldo de recursos de software.
- o Respaldo de personal registrado.

Si solo le interesa respaldar registros a partir de cierto rango de fechas, puede especificarlo abajo. En caso de dejarlos vacíos, se hará un respaldo completo de la tabla.

Desde:  Hasta:

**Figura 4.3.9.1. Interfaz de respaldos de registros de la base de datos.**

Se presionará la opción “Respaldo de registros de autoacceso”, dejando los campos de fecha vacíos y presionando el botón “Respaldo”, el sistema respaldará todos los registros que existan en la tabla de autoacceso y los enviará en un archivo a través del navegador.

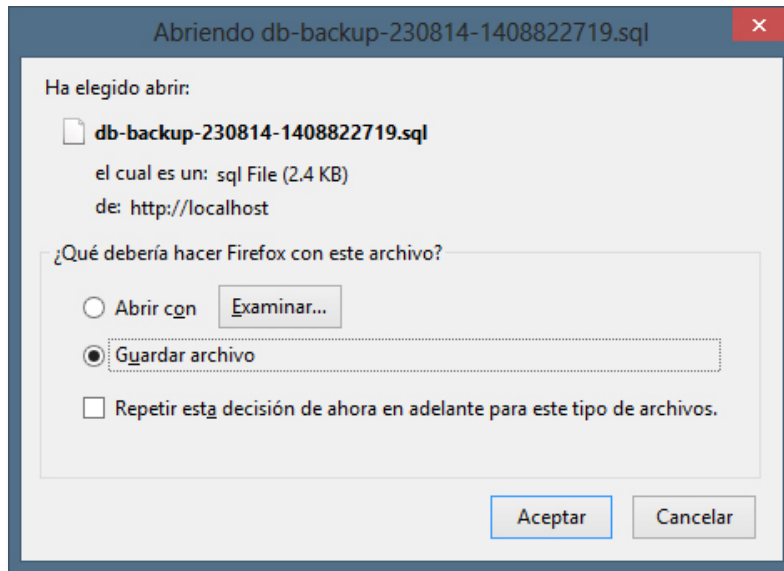


Figura 4.3.9.2. El explorador ofrece la descarga del archivo de respaldo.

Ahora se procederán a eliminar los datos de la tabla de autoacceso, en el apartado “Eliminar registros”, pulsando la opción “Eliminar registros de autoacceso” y dejando los campos de fechas vacíos se eliminarán todas las entradas de autoacceso.

Eliminar registros del sistema

Precaución: Las siguientes funciones eliminarán los registros del sistema, la única forma de restaurar los registros eliminados será mediante la creación de [respaldos de registros](#). Se recomienda ampliamente hacer un respaldo de los registros antes de eliminarlos.

Elija una opción:

- Eliminar registros de incidentes.
- Eliminar registros de autoacceso.
- Eliminar registros de hardware desechado.
- Eliminar registros de software obsoleto.

Si solo le interesa eliminar registros a partir de cierto rango de fechas, puede especificarlo abajo. En caso de dejarlos vacíos, se eliminará todo el contenido de la tabla.

Desde:   Hasta:

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). ( La consulta tardó 0.0004 seg )

```
SELECT *
FROM `autoacceso`
WHERE 1
LIMIT 0 , 30
```

Figura 4.3.9.3. Los registros de la tabla de incidentes se eliminan mediante la interfaz, y se puede comprobar que la tabla no posee entradas.

La restauración de los registros se efectúa desde el apartado “Restaurar registros”, se selecciona la opción “Restaurar registros de autoacceso” el cual muestra un selector de archivos para elegir el archivo que contiene la información previamente respaldada, luego presionar el botón “Restaurar”.

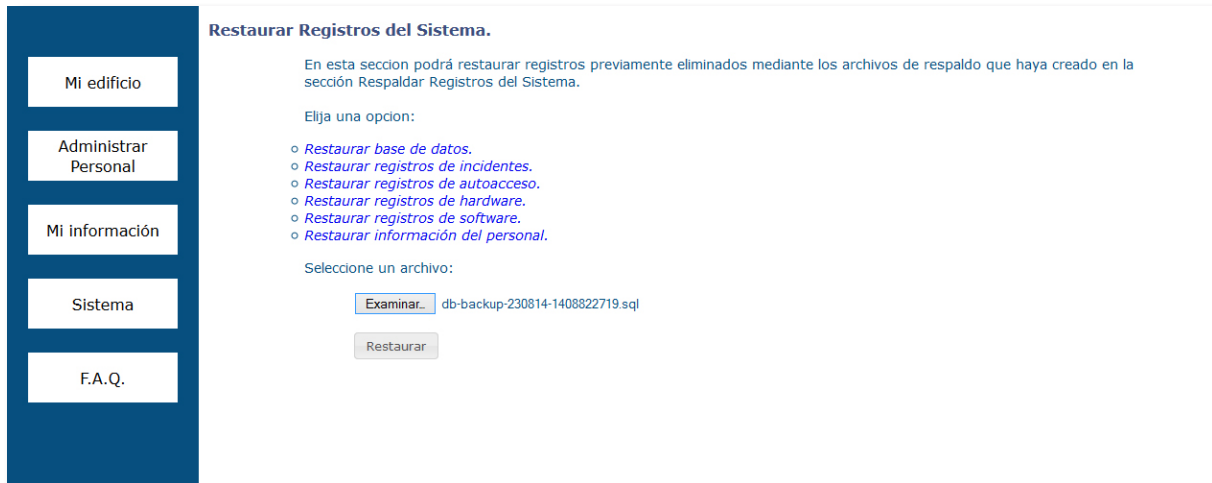


Figura 4.3.9.4. Se selecciona un archivo con datos respaldados.

Mostrando registros 0 - 29 (~40 total) La consulta tardó 0.0009 seg

```
SELECT *
FROM 'autoacceso'
LIMIT 0, 30
```

id_autoacceso	fecha_registro	hora_entrada	hora_salida	id_recurso	matricula
1	2014-08-20	22:24:46	23:28:15	2	201014410
2	2014-08-20	22:25:31	23:28:17	3	201004998
3	2014-08-20	22:27:45	23:28:14	4	200913320
4	2014-08-20	13:29:29	14:53:49	7	201004998
5	2014-08-20	13:30:19	14:59:12	2	201107941
6	2014-08-20	13:31:12	15:10:34	8	201241005
7	2014-08-20	13:32:08	14:33:09	5	201222143
8	2014-08-20	13:32:36	14:57:59	3	200913320
9	2014-08-20	14:58:31	15:10:31	3	201044551
10	2014-08-20	14:59:06	15:10:33	4	201014785
11	2014-08-19	10:39:50	11:10:36	2	200914544
12	2014-08-19	10:41:39	11:23:15	8	200925613
13	2014-08-19	10:42:26	13:01:03	3	200931554
14	2014-08-19	10:43:44	13:01:07	32	201214977
15	2014-08-19	10:54:39	13:01:05	4	201287644
16	2014-08-19	11:10:30	11:23:21	5	200841147

Figura 4.3.9.5. La base de datos contiene nuevamente los registros eliminados

El sistema lee el archivo y realiza la inserción de datos a la base.

### 4.3.10. Prueba 10. Manipular información personal

Un usuario es capaz de cambiar la contraseña de su sesión en el momento que desee, la sección “Mi información” muestra una ficha de datos del usuario.



The screenshot shows the user profile page. At the top, there is a header with the logo of FCG SALC on the left, the text "Benemérita Universidad Autónoma de Puebla Facultad de Ciencias de la Computación" in the center, and the university crest on the right. Below the header, a blue navigation bar contains the text "Inicio", "Te damos la bienvenida Rosa", and "Cerrar Sesión". On the left side, there is a vertical menu with buttons for "Mi laboratorio", "Administrar Personal", "Reportes", "Incidentes", and "Mi información". The main content area displays the user's information: "Rosa García Tamayo", "Número de trabajador: 200000001", "Correo electrónico: rgtamayo@cs.buap.mx", "Teléfono personal: (044)2224-37-9026", and "Horario: Lunes a Viernes de 13:00 a 19:00". At the bottom of this information box are two buttons: "Regresar" and "Cambiar contraseña".

Figura 4.3.10.1. Ficha de datos de registro de un usuario.

Presionando el botón “Cambiar contraseña” se presentará un formulario para cambio de datos, los usuarios tipo colaborador y coordinador obtendrán un formulario como el de la figura 4.3.10.2, y el administrador puede alterar más datos, los cuales se ven en la figura 4.3.10.3.



The screenshot shows the password change form. The header and navigation bar are identical to the previous figure. The left menu is also present. The main content area contains the following text: "Los campos marcados con \* son obligatorios." and "Si desea cambiar su contraseña escriba su contraseña anterior." Below this, there are three input fields: "Contraseña anterior:", "Contraseña:", and "Confirmar contraseña:", each followed by a red asterisk. At the bottom of the form is a "Registrar" button.

Figura 4.3.10.2. Vista de cambio de contraseña para los usuarios Colaborador y Coordinador

**Figura 4.3.10.3. Vista de cambio de multiples datos para el usuario Administrador.**

Pueden alterarse los datos y establecer una nueva contraseña si se desea, al momento de oprimir el botón “Registrar”, el sistema validará cada dato que proporcione y luego mandará la solicitud de actualización.

login	contrasena	no_identificacion	nombre	a_paterno	a_materno	correo_electronico	telefono
Administrador	f77a0e56e67ead89b4f0728f20b8faa	200514799	José Abraham	Baez	Bagatella	icc_bagatella@outlook.com	
JAbraham	f588a3f1e5147a7c8d257339c4ba7a6f	200514799	José Abraham	Baez	Bagatella	zdm_ck@outlook.com	
RGTamayo	cb860a88452cf25c1ca5b7e96d3a73bc	200000001	Rosa	García	Tamayo	rgtamayo@cs.buap.mx	

login	contrasena	no_identificacion	nombre	a_paterno	a_materno	correo_electronico	telefono
Administrador	a1755be35d978fa87d941260daa7a0e1	200514799	José Abraham	Baez	Bagatella	icc_bagatella@outlook.com	
JAbraham	f588a3f1e5147a7c8d257339c4ba7a6f	200514799	José Abraham	Baez	Bagatella	zdm_ck@outlook.com	[222]2
RGTamayo	cb860a88452cf25c1ca5b7e96d3a73bc	200000001	Rosa	García	Tamayo	rgtamayo@cs.buap.mx	

**Figura 4.3.10.4. Los datos que cambian actualizan un los registros en la base de datos.**

Se hace una actualización a la base de datos y la información nueva es guardada.

## **CONCLUSIONES**

La formulación del proyecto constó de varias entrevistas con colaboradores y coordinadores de los módulos y de esta manera pudieron determinarse los aspectos básicos, comunes y fundamentales del cuidado y atención al laboratorio, se empezó a trabajar un prototipo de interfaz y a plantear todos los casos de uso fundamentales para el futuro sistema.

Durante la elaboración primeramente se trabajaron los “planos” de la estructura del sistema, es decir los diagramas derivados de los casos de uso sugeridos por el proceso unificado de desarrollo de software obteniendo así una visión completa y el alcance definitivo del proyecto. También se planeó la imagen, interfaz y los aspectos interactivos entre los usuarios y el sistema.

La construcción fue una de las etapas más largas del proyecto pues constó de muchos componentes e interfaces, además se implementaron todos los componentes requeridos en los diagramas de realización de casos de uso, estos trabajados en los lenguajes HTML, JavaScript, PHP, CSS2. Se trabajó y evolucionó el diseño conceptual de la base de datos, su modelo final cumple con los requisitos planteados por la problemática, se eligió el gestor de bases de datos MySQL, se construyó el modelo lógico de la base de datos en base a este lenguaje de definición y manipulación y además se crearon todos los procedimientos almacenados con los que funcionará el sistema. También se implementaron componentes de seguridad como es el caso del módulo de encriptación, y de integridad de datos, que interpretarán lo capturado al cotejamiento de la base de datos y viceversa. Estos no aparecen en los diagramas de casos de uso dado que son requisitos no funcionales. Por último se utilizaron algunas técnicas de contenido dinámico por medio de AJAX, JQuery y CSS3.

Para la fase de entrega se realizaron múltiples pruebas del sistema completado y se detectaron errores en las primeras 2 sesiones de pruebas, estos fueron corregidos inmediatamente. Luego se les pidió a algunos colaboradores del módulo 1 que probarán el sistema, siendo esta prueba exitosa durante el tiempo de uso que se le dio.

El sistema de administración de los laboratorios de cómputo es estable. Ha sido probado contra 11 técnicas de inyección de SQL, ninguna teniendo éxito. También se ha intentado saturar los registros de la base de datos y la cantidad de datos que carga cada componente y su rendimiento sigue siendo bueno. Se ha expuesto el sistema a varios tipos de datos erróneos que pueden ser capturados por los componentes de integridad, tales como fechas o signos de puntuación no son un problema. Los módulos componentes de migración han funcionado correctamente siempre y cuando se cumplan los requisitos recomendados planteados en la sección de implementación (véase la tabla 4.1.2).

El uso del sistema es sencillo e inteligible para cada tipo de usuario, se trabajó con técnicas que estimulan la memoria a corto plazo, de reconocimiento y semántica, también con diseño gráfico profesional para que la interfaz sea apacible para el usuario. Los registros digitales pueden perdurar más tiempo y son más confiables que los registros a papel, y la comunicación por medio de este canal entre los colaboradores y coordinadores es específicamente para asuntos del laboratorio. Con este sistema se pretende facilitar la tarea de administración y coordinación del personal que trabaja en los laboratorios de cómputo.

## TRABAJOS A FUTURO

SALC-FCC ya es un sistema que cubre las necesidades fundamentales del trabajo del personal de los laboratorios, y se estructuró de manera que pudieran agregarse más componentes y características que cumplan con deseos y otras necesidades que puedan ir surgiendo con el paso del tiempo. Todas las características pueden apreciarse en las secciones 2.7 y 2.8 los casos de uso, especificación de casos de uso y el modelo de diseño, así como las necesidades en la base de datos mediante el modelo conceptual de la sección 3.1.

Las recomendaciones de los trabajos al futuro en el sistema son:

- *Buscador.* Las secciones de inventarios pueden llegar a ser muy extensas y las páginas de elementos también puede ser un número muy grande, la inclusión de un buscador por ciertas características clave podría hacer más ágil una búsqueda determinada.
- *Información detallada.* En la sección de inventario de hardware existe la sección de más información de las piezas, podría realizarse un intérprete o una conexión con un motor de búsqueda que encuentre detalles profundos de las piezas.
- *Técnicas AJAX superiores.* El sistema cuenta con dinamismo mediante AJAX en varias secciones, pero este puede aplicarse a todo el proyecto de manera bien planeada, sobre todo en los listados o ingresos de datos.
- *HTML5.* SALC-FCC está construido con los estándares propuestos para XHTML el cual está basado en HTML4 y XML, la nueva tecnología en interpretación de hipertexto puede hacer muchas mejoras en las interfaces.
- *Variedad de reportes.* Pueden crearse más variedad de reportes de datos y estadísticas conforme el trabajo en el laboratorio lo requiera.

Se recomienda que las mejoras y agregados al sistema se describan de la misma manera que el proyecto fue creado o bien se utilice otra técnica formal de ingeniería de software con el objetivo de preservar su portabilidad y su capacidad de mejoras por otros desarrolladores.

## BIBLIOGRAFÍA

1. Abran, A., Moore, J.W., Bourque, P. & Dupis, R. (2004). *Guide to the Software Engineering Body of Knowledge*. Estados Unidos: The Institute of Electrical and Electronics Engineers. Jacobson, I., Booch, G. & Rumbaugh, J. (1999). *El proceso unificado de desarrollo de software*. Madrid, España: Pearson educación.
2. Pressman, R.S. (2002). *Ingeniería del software, un enfoque práctico*. Quinta edición. Madrid, España: McGraw Hill.
3. Sommerville, I. (2005). *Ingeniería del software*. Séptima edición. Madrid, España: Pearson educación.
4. Brooks, F.P. (1995) *The Mythical Man-Month*. Estados Unidos: Addison-Wesley.
5. Silberschatz, A., Korth, H.F. & Sudarshan, S. (2002). *Fundamentos de bases de datos, cuarta edición*. Madrid, España: McGraw Hill.
6. De Miguel Castaño, A., Piattini Velthuis, M. & Marcos Martínez, E. (2000). *Diseño de bases de datos relacionales*. D.F., México: Alfaomega grupo editor.
7. Kabir, M.J. (2003). *La biblia de Servidor Apache 2*. Madrid, España: Ediciones Anaya multimedia.
8. Van Reijswoud, V. & De Jager, A. (2008). *Free and open source software for development*. Milan, Italia: Polimetrica.
9. Wheeler, D.A. (2014) *Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*. Recuperado de [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)
10. The apache software foundation (2014). *mod\_rewrite - Apache HTTP Server Version 2.2*. Recuperado de [http://httpd.apache.org/docs/2.2/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html)
11. ModCFML (2014). *Home – ModCFML*. Recuperado de <http://www.modcfml.org/>

12. Garret J.J. (2005). *Ajax: A new Approach to Web Applications*. Recuperado el 14 de junio de 2014 en <http://www.adaptivepath.com/publications/essays/archives/000385print.php>

13. The PHP group (2014) . *What is PHP?* . Recuperado de <http://www.php.net/manual/en/intro-what-is.php> .

14. Wikipedia (2013). *Proceso unificado*. Recuperado de [http://es.wikipedia.org/wiki/Proceso\\_Unificado](http://es.wikipedia.org/wiki/Proceso_Unificado).

15. FPDF. (2014). *What is FPDF?*. Recuperado de <http://www.fpdf.org/?lang=en>.