

# UACM

**Universidad Autónoma  
de la Ciudad de México**

---

*Nada humano me es ajeno*

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS  
ELECTRÓNICOS Y DE TELECOMUNICACIONES

**Desarrollo e implementación de un sistema de  
administración para el laboratorio de electrónica de  
la UACM San Lorenzo Tezonco**

TRABAJO RECEPCIONAL  
PARA OBTENER EL TÍTULO DE LICENCIADOS EN  
INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE  
TELECOMUNICACIONES

PRESENTAN

**EFRAÍN URINCHO ARMENTA  
VIOLETA ADGLAE PÉREZ ALCALÁ**

Director del trabajo recepcional:  
**M. en C. Joel Yazbek Buendía Gómez**

Ciudad de México, enero 2017

## SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



## UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

### RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

#### DERECHOS RESERVADOS ©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

## Dedicatorias

### ***Efraín:***

*Agradezco la educación, valores y principios inculcados por parte de mi familia, amigos y conocidos ya que apoyaron a desarrollarme durante el trayecto de mi educación. Agradezco la motivación por parte de mis profesores y el tiempo dedicado a mi persona.*

### ***Violeta:***

*A mis padres los seres mas sublimes que me dio la vida, a mis hermanos, a mis tios y a mis abuelos a quienes agradezco infinitamente su dedicación, tiempo y recursos que me regalaron para lograr un crecimiento profesional para mi persona. También agradezco a la UACM la cual a través de sus profesores que con paciencia y sabiduria me inculcaron mi vocación al servicio de los demás. Agradezco a mis amigos por haber compartido su tiempo y haberme brindado su amistad incondicional. Así mismo agradezco a todas las personas que han coincidido conmigo en este andar por la vida, y principalmente a aquellas que llegaron para quedarse.*





# Agradecimientos

*Agradecemos a la UACM por el apoyo  
recibido para la impresión y empastado  
de este trabajo.*



# Resumen

La Universidad Autónoma de la Ciudad de México (UACM), es una institución educativa de nivel superior para estudiantes a nivel licenciatura y posgrado en la Ciudad de México desde el año 2001. La demanda de servicios rápidos y correctos va creciendo desde su fundación hasta ahora, por esta razón los procesos de gestión y administración interna deben ser ejecutados por algún tipo de software, de tal forma que puedan responder a las necesidades de estudiantes, administrativos y docentes.

Por otro lado, la UACM posee equipo electrónico como osciloscopios, generadores de funciones, fuentes de poder, multímetros entre otros; dicho equipo es utilizado por los estudiantes de las carreras de ingeniería para realizar prácticas educativas para complementar y fortalecer el conocimiento aprendido en las clases. Los procesos de administración del equipo, en los laboratorios de electrónica de la UACM se realizan de manera manual hoy en día, las operaciones como préstamos y devoluciones son registrados en papel y se almacenan físicamente a través del tiempo. En este trabajo, se presenta el desarrollo e implementación de un sistema informático llamado SAMLAB UACM-SLT v 1.0 (Sistema de Administración del Laboratorio UACM-SLT v 1.0), el cual está conformado por los siguientes módulos de desarrollo: adquisición de datos, interfáz gráfica de usuario y procesamiento y almacenamiento de información. El acceso al sistema será de forma remota, desde una computadora cliente hacia una computadora servidor a través de una red establecida. El desarrollo SAMLAB UACM-SLT v1.0 estará instalado sobre el sistema operativo Centos 6.0, el servidor web utilizará los servicios proporcionados por LAMP (Linux-Apache-MySQL y PHP). Para el desarrollo, implementación y modelado de software se utilizó el Proceso Unificado de Rational (RUP) el cual sirvió para organizar las fases del proyecto, apoyados del lenguaje de modelado unificado de software UML (Unified Modeling Language). Al contar con el sistema SAMLAB UACM-SLT v1.0, el administrador del laboratorio tendrá las mismas posibilidades que tenía con la administración manual, con la diferencia que la información relacionada con estudiantes, equipo, préstamos y devoluciones estará organizada en una base de datos, la cual albergará la información y podrá ser consultada en cualquier momento.



# Índice

<b>Agradecimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vii</b>
<b>1. Planteamiento del problema</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Definición del problema . . . . .	2
1.2.1. Estado de administración antes de SAMLAB UACM-SLT v1.0 . . . . .	2
1.2.2. El manejo de información antes de SAMLAB UACM-SLT v1.0 . . . . .	4
1.3. Objetivos . . . . .	5
1.4. Justificación . . . . .	6
1.5. Alcances del proyecto . . . . .	6
<b>2. Fundamentación teórica</b>	<b>9</b>
2.1. Antecedentes . . . . .	9
2.1.1. Clasificación general del software . . . . .	9
2.1.2. Software para la administración . . . . .	11
2.2. Enfoque de desarrollo . . . . .	12
2.2.1. Análisis general de requerimientos . . . . .	13
2.2.2. Módulos que conforman al sistema SAMLAB . . . . .	15
2.2.3. Módulo de adquisición de datos . . . . .	16
2.2.4. Módulo de interfaz con el usuario . . . . .	18
2.2.5. Almacenamiento y procesamiento de información . . . . .	19
2.3. RUP como metodología de diseño para el software SAMLAB . . . . .	21
2.4. UML como lenguaje de diseño de software . . . . .	24
2.5. Los casos de uso para el desarrollo del sistema . . . . .	26
2.6. La red de computadoras . . . . .	27
2.6.1. Provisión de seguridad de red . . . . .	27
2.6.2. IPCOP como Firewall . . . . .	28
2.7. S.O. Centos 6 como servidor web . . . . .	29

---

2.7.1. LAMP . . . . .	30
<b>3. Implementación de la red</b>	<b>31</b>
3.1. La UACM como proveedor de red . . . . .	31
3.2. Instalación de servidor web HTTP . . . . .	34
3.3. Instalación MySQL . . . . .	36
3.4. Definición y creación de tablas con phpMyAdmin . . . . .	37
<b>4. Diseño e implementación de Software SAMLAB</b>	<b>43</b>
4.1. Caso identificar y autenticar administrador . . . . .	43
4.2. Caso estudiantes . . . . .	47
4.3. Caso equipo . . . . .	57
4.4. Caso Realizar Préstamo . . . . .	62
4.5. Caso Realizar Devolución . . . . .	68
4.6. Caso Realizar Cambios . . . . .	72
<b>5. Análisis de resultados</b>	<b>77</b>
5.1. Pruebas de funcionamiento . . . . .	78
5.1.1. Prueba de funcionamiento del sistema fase 1: . . . . .	79
5.1.2. Prueba de funcionamiento del sistema fase 2: . . . . .	80
5.2. Prueba de operación . . . . .	82
<b>Conclusiones</b>	<b>85</b>
<b>Bibliografía</b>	<b>89</b>

# Índice de figuras

1.1. Diagrama de flujo que representa el proceso manual préstamo-devolución . . . . .	3
1.2. Papeleta de préstamo y devolución manual . . . . .	4
1.3. Pequeña parte del conjunto de papeletas recolectadas a través del tiempo en el laboratorio de electrónica de la UACM-SLT . . . . .	5
2.1. Representación gráfica de acceso al sistema de estudiantes de la UACM . . . . .	11
2.2. Sistema bibliotecario de la UACM con el software Ex Libris . . . . .	12
2.3. Módulos que conforman el sistema SAMLAB UACM-SLT v1.0 . . . . .	15
2.4. Código que representa la matrícula en la credencial de estudiante de la UACM . . . . .	16
2.5. Creación de códigos de equipo . . . . .	18
2.6. Interfaz gráfica del sistema SAMLAB UACM-SLT v1.0 . . . . .	19
2.7. Diagrama de negocios RUP . . . . .	21
2.8. Esquema de diagramas disponibles en UML . . . . .	25
2.9. Diagrama de casos de uso en su nivel operacional básico . . . . .	26
2.10. Servidor SAMLAB UACM-SLT v 1.0 . . . . .	30
3.1. Esquema básico de abastecimiento de red a los laboratorios B-402 y B-404 . . . . .	32
3.2. Topología física de última milla . . . . .	32
3.3. Topología física de la red SAMLAB . . . . .	33
3.4. Accesos permitidos a la red SAMLAB . . . . .	34
3.5. Servicio web apache . . . . .	35
3.6. Diseño de las tablas en la base de datos SAMLAB . . . . .	37
3.7. Definición de la tabla admin . . . . .	38
3.8. Definición de tabla alumnos, la cual cuenta con 4 campos de tipo varchar . . . . .	39
3.9. Definición de la tabla material y los campos que la componen . . . . .	40
3.10. Tabla préstamo . . . . .	41
3.11. Tabla registro . . . . .	42

4.1. Diagrama de actividad de acceso del administrador . . . . .	44
4.2. Diagrama de secuencia del caso identificar y autenticar administrador . . . . .	45
4.3. Menú principal de acceso al sistema SAMLAB . . . . .	46
4.4. Submenú de la sección estudiantes . . . . .	47
4.5. Diagrama de casos de uso registro de estudiantes . . . . .	48
4.6. Diagrama de secuencia de caso registrar estudiante . . . . .	49
4.7. Diagrama de actividad del caso registrar estudiante . . . . .	50
4.8. Diagrama de secuencia caso de uso eliminar estudiante . . . . .	52
4.9. Diagrama de actividad del caso eliminar estudiante . . . . .	53
4.10. Diagrama de secuencia del caso editar estudiante . . . . .	55
4.11. Diagrama de secuencia del caso consultar estudiantes . . . . .	56
4.12. Diagrama de actividad del caso de uso consultar estudiantes . . . . .	57
4.13. Impresión de pantalla del submenú equipo . . . . .	59
4.14. Diagrama de casos de uso registro de equipo . . . . .	59
4.15. Diagrama de actividad eliminar equipo . . . . .	61
4.16. Diagrama de actividad editar registro de equipo . . . . .	61
4.17. Estudiante activo para realizar un préstamo . . . . .	63
4.18. Ciclo iterativo para agregar equipo a un préstamo . . . . .	64
4.19. Diagrama de secuencia para realizar un préstamo, considerando un caso ideal . . . . .	65
4.20. Diagrama de actividad del caso préstamo con validación de estudiante, condiciones normales . . . . .	66
4.21. Continuación del diagrama de actividad del caso préstamo, bajo las condiciones de validación de equipo, considerando la existencia de equipo disponible en el sistema . . . . .	67
4.22. Diagrama de actividad cancelar Préstamo, muestra la opción de cancelar durante el préstamo . . . . .	68
4.23. Diagrama de secuencia de la devolución de un préstamo secuencia normal . . . . .	70
4.24. Diagrama de actividad del caso realizar devolución secuencia normal . . . . .	71
4.25. Diagrama de secuencia del caso realizar cambios en un préstamo . . . . .	73
4.26. Diagrama de iteración del caso Devolución Parcial . . . . .	74
4.27. Diagrama de secuencia del caso consultas . . . . .	75
4.28. Consulta de la tabla préstamos activos . . . . .	76
4.29. Consulta de tabla historial . . . . .	76

# Índice de Tablas

2.1.	Proceso RUP 1 fase 1 . . . . .	23
2.2.	Proceso RUP 2 fase 2 . . . . .	24
2.3.	Tipos de seguridad en una red de datos, tomada de [10] . . . . .	27
3.1.	Comandos para instalación de apache . . . . .	35
3.2.	Comandos de instalación MySQL . . . . .	36
4.1.	Tabla de requerimientos para el caso identificar y autenticar administrador, donde se describen los puntos que se deben llevar a cabo para la correcta identificación . . . . .	44
4.2.	Plantilla de requerimientos para caso de uso registrar estudiante . . . . .	49
4.3.	Plantilla de requerimientos para el caso de uso eliminar estudiante . . . . .	51
4.4.	Tabla de requerimientos para el caso editar estudiante . . . . .	54
4.5.	Tabla que muestra el equipo existente en el laboratorio B-402 . . . . .	58
4.6.	Tabla de requerimientos para el caso registro de equipo . . . . .	60
4.7.	Tabla de requerimientos para el caso préstamo . . . . .	62
4.8.	Tabla de requerimientos del caso: Realizar Devolución . . . . .	69
5.1.	Tabla de prueba de caja negra de módulo registrar fase 1 . . . . .	78
5.2.	Tabla de prueba de caja negra de módulo eliminar fase 1 . . . . .	79
5.3.	Tabla de prueba de caja negra de módulos préstamo-devolución fase 1 . . . . .	80
5.4.	Tabla de prueba de caja negra del módulo editar registro fase 1 . . . . .	80
5.5.	Tabla de prueba de caja negra del módulo consultar registro fase 2 . . . . .	81
5.6.	Tabla de prueba de funcionamiento de módulo cambios en préstamo y devolución fase 2 . . . . .	81
5.7.	Tabla de operación del sistema fase 1 . . . . .	82
5.8.	Tabla de operación del sistema caso devolución fase 1 . . . . .	82



# Capítulo 1

## Planteamiento del problema

*Hay dos formas de diseñar software: la primera es hacerlo tan simple que obviamente no hay deficiencias y la segunda es hacerlo tan complicado que no hay deficiencias obvias. La primera forma es mucho más difícil.*

Martin Fowler

### 1.1. Introducción

El laboratorio B-402 de la UACM-SLT, es un espacio de estudio que sirve para practicar lo aprendido en las clases teóricas de las carreras de ingeniería en el área de electrónica. Este lugar alberga equipo electrónico como: fuentes de energía, generadores de funciones, osciloscopios, multímetros, borradores de memorias de luz UV, tarjetas de conmutación de telefonía, tarjetas de adquisición de datos serial, USB, PCI y antenas de radiocomunicación. Dicho equipo es utilizado por los estudiantes de la carrera de ingeniería a través de préstamos.

Actualmente el proceso préstamo-devolución comienza con la solicitud de un estudiante, éste llena un registro en papel llamado: *papeleta de préstamo*, en la cual se registra la información del estudiante y del material utilizado; cuando el estudiante termina de utilizar el material, el administrador revisa el estado del material y genera la devolución. Este proceso manual se ha llevado a cabo desde la creación del laboratorio de electrónica hasta la fecha, y ha resuelto la problemática inicial, que es tener registro del equipo en uso, sin embargo, el número de estudiantes va incrementando semestre con semestre, lo cual genera la necesidad de mejorar la administración del equipo y de usuarios en el laboratorio para tener un mejor control de préstamos y devoluciones.

## 1.2. Definición del problema

La mejor forma de solucionar un problema es conocerlo, a continuación, se describe la forma como se llevaba a cabo la administración en el laboratorio de electrónica, identificado en este trabajo como: *proceso manual de préstamo-devolución*, dicho proceso fue tomado como referencia central para el diseño de SAMLAB UACM-SLT v1.0, además para identificar los objetivos del proyecto y los alcances de este trabajo.

### 1.2.1. Estado de administración antes de SAMLAB UACM-SLT v1.0

El préstamo y devolución de equipo electrónico en la UACM-SLT, puede representarse por medio de un diagrama de flujo, el cual se muestra en la figura 1.1. Dicho proceso está considerado como el caso base, ya que considera que un estudiante puede solicitar todo el equipo que necesite y entregarlo cuando lo desee. Una consecuencia importante del constante uso del equipo, es la avería parcial o total, y esto se debe principalmente a los siguientes puntos:

- **Creciente aumento de usuarios en el laboratorio:** El número de usuarios en el laboratorio de la UACM-SLT va crecido semestre con semestre, de atender a un grupo de alrededor de 5 a 10 estudiantes en el año 2005, pasó a atender grupos de 15 a 25 estudiantes en el año 2015.
- **Equipo disponible para cada estudiante:** El equipo disponible para cada estudiante no estaba definido, ya que un usuario podía solicitar todo el equipo que quisiera, incluso equipo de más en algunos casos; esta situación era una desventaja para otro estudiante ya que quizá no habría equipo disponible.
- **Tiempo de uso del equipo por cada sesión de préstamo:** El tiempo de uso del equipo tampoco estaba definido, un estudiante podía utilizar el equipo todo el día si lo deseaba, incluso tenerlo en calidad de préstamo y no utilizarlo durante el día.

Considerando los puntos anteriores, se tiene como consecuencia principal, la falla o avería de equipo electrónico, parcial o total, al no contar con registro preciso de la hora, día, ni momento exacto en el que un equipo dejó de funcionar o comenzó a fallar. Esto da como resultado equipo fuera de uso, o en espera de mantenimiento o en su defecto funcionado a la mitad de su capacidad. Esta es una de las principales razones por las que es necesario desarrollar una nueva forma de administración del equipo electrónico.

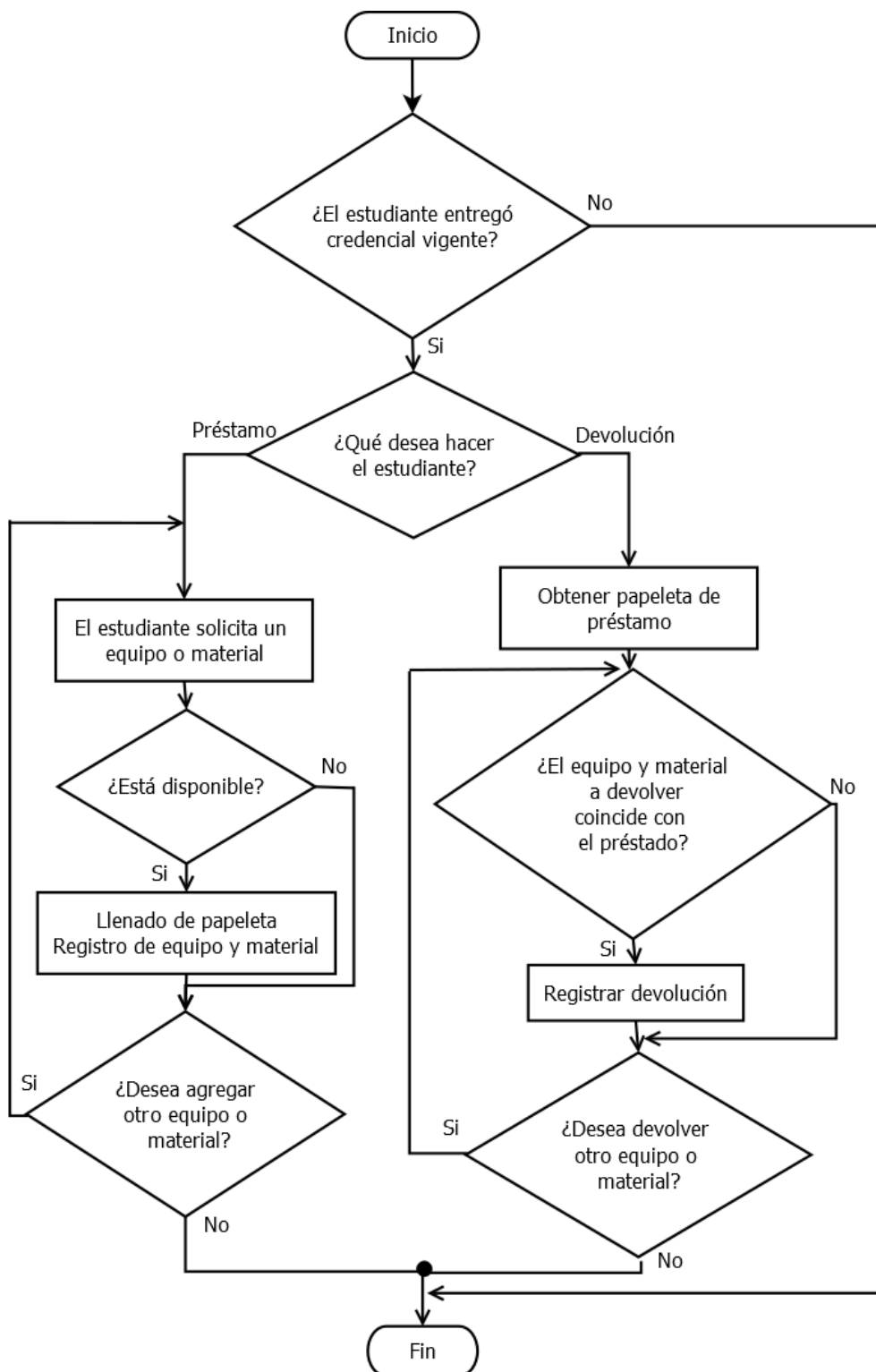


Figura 1.1: Diagrama de flujo que representa el proceso manual préstamo-devolución





Figura 1.3: Pequeña parte del conjunto de papeletas recolectadas a través del tiempo en el laboratorio de electrónica de la UACM-SLT

### 1.3. Objetivos

**Objetivo general:** Diseñar, desarrollar e implementar el sistema computacional llamado SAMLAB UACM-SLT v1.0, utilizando una topología de red, un servidor y una base de datos para gestionar el proceso préstamo-devolución del equipo electrónico, de esta forma el administrador del sistema realizará actividades de administración, utilizando una interfáz gráfica web.

**Objetivos específicos:** Implementar una arquitectura cliente-servidor, que cuente con seguridad de acceso solo para usuarios permitidos a un servidor, el cual en conjunto con una base de datos almacene y procese la información referente a los estudiantes y al equipo, para llevar a cabo el proceso préstamo-devolución. Como producto final se deberá diseñar una interfáz gráfica de usuario, que a través de la web tendrá intercambio de información entre una computadora cliente (LAB-B402) y una computadora servidor (LAB-B404). Para organizar las tareas se dividirá el problema en módulos de desarrollo, los cuales podrán ser desarrollados independientemente e incluirlos cuando se desee, los módulos a desarrollar son:

1. **Adquisición de datos:** Implementar una tecnología de adquisición de datos de barras bidimensional, que sirva como el medio de recolección de información referente a la matrícula de los estudiantes y a un código creado para el equipo.

2. **Interfáz gráfica de usuario:** Diseñar, desarrollar e implementar una interfáz gráfica de usuario GUI utilizando PHP, java script y HTML, la cual será la forma en la que el usuario podrá interactuar con el sistema.
3. **Procesamiento y almacenamiento de información:** Diseñar una base de datos e implementar un DBMS utilizando MySQL para almacenar toda la información recolectada.

## 1.4. Justificación

La realización de un producto de software que apoye la administración en el laboratorio de electrónica de la UACM-SLT, se debe a que al cursar las asignaturas correspondientes a la carrera de ingeniería, identificamos varias problemáticas en la forma como se realizaba la administración del proceso préstamo-devolución de equipo; por ejemplo, el creciente aumento de usuarios en el laboratorio y la alta demanda de uso de los equipos da como resultado la avería parcial o total del equipo electrónico, sin que alguien pueda hacer algo ni mucho menos tener registro de vida útil restante del equipo. Nosotros decidimos crear un producto operacional de software, para comenzar a solucionar en parte dichas problemáticas, y así mejorar la administración manual mediante una administración apoyada por un sistema de software y que sea una herramienta que apoye a la optimización de las actividades administrativas del laboratorio B-402.

## 1.5. Alcances del proyecto

La solución que se plantea para el desarrollo de SAMLAB UACM-SLT v1.0, cubre principalmente los requerimientos recolectados del proceso préstamo-devolución en el laboratorio B-402, de igual manera, el desarrollo de software se apegó a recursos de hardware disponibles en la universidad. Considerando que el desarrollo de software se dividió en tres importantes módulos, a continuación se describen los alcances por cada módulo.

- Para la adquisición de datos se utilizó un escaner de códigos de barras bidimensional, para tomar lectura de los códigos ya existentes en las credenciales de cada estudiante de la UACM y para la lectura de códigos creados para el equipo.
- La seguridad que se manejará en la red será de acceso, es decir, ningún usuario administrador podrá ingresar al sistema si no ingresa el nombre de usuario y la contraseña asignados. Un firewall actuará siempre redireccionando paquetes de un origen a un destino especificados previamente.

- El sistema estará disponible para ser operado considerando el proceso préstamo-devolución de forma manual, llevado a forma totalmente digital, además se incluyen validaciones de caracteres para la matrícula y para los códigos del equipo, para así garantizar que no existirá información erronéa o duplicada.
- La base de datos utilizada se manejará de manera local, es decir, no tendrá ninguna relación con otra base de datos en el servidor ni en una red externa.
- El sistema SAMLAB UACM-SLT v1.0, puede ser visitado desde cualquier red interna en la universidad, esto será posible con ayuda del firewall, es aquí donde el super administrador tendrá que otorgar o denegar permisos a una dirección IP seleccionada.
- El sistema recopilará información referente a estudiantes, equipo y material, préstamos activos e historial de préstamos, en tablas que contendrá la base de datos SAMLAB, las consultas que realiza será de manera general, y organizada para mejor visualización del administrador.

A continuación se muestra la fundamentación teórica que se necesita saber para seguir el desarrollo del sistema final.



## Capítulo 2

# Fundamentación teórica

*Hazlo tan simple como sea posible, pero  
no más.*

Albert Einstein

### 2.1. Antecedentes

Los sistemas de software son tan utilizados hoy en día que desde su aparición han evolucionado de manera exponencial, hasta convertirse en una herramienta indispensable para estudiar y trabajar, de modo que en este momento es imposible pensar en una computadora o un equipo electrónico sin un sistema de software para su operación. En este capítulo, se abordarán temas relacionados al software para subrayar su importancia.

#### 2.1.1. Clasificación general del software

El término "*software*" no entró en uso sino hasta 1958 y probablemente fue acuñado por el profesor de la Universidad de Princeton John W. Tukey, en el artículo mensual "*The American Mathematical*" en enero del mismo año [1].

La palabra software proviene del idioma inglés (literalmente partes blandas o suaves), que en el idioma español no tiene una traducción formal, por lo que se utiliza sin traducir y fue admitida por la real academia española (RAE) como[2]: "*El software es un conjunto de programas, instrucciones y reglas informáticas, que permiten ejecutar múltiples tareas en una computadora, se desarrolla mediante distintos lenguajes de programación, permitiendo la interconexión de sistemas*"

La manera en la que se clasifica el software se ha ido determinando por el uso y aplicación que con el tiempo se le ha dado, así lo explica el artículo

*The history of history of software* [2] y lo clasifica en tres categorías como se indica a continuación.

- **Software de sistema:** Se conoce también como software base, ya que es el encargado de interconectar los diferentes módulos físicos y lógicos llamados programas que componen una computadora, para ponerlos al servicio de un usuario. Como su nombre lo indica, el software de sistema es la base de desarrollo para otros programas y aplicaciones. Algunos ejemplos son:
  - Sistema operativo (Mac OS, Windows, Linux, Unix, entre otros)
  - Controladores de dispositivos

- **Software de programación:** El software de programación es el medio de desarrollo para llegar a una aplicación específica, es decir, son los compiladores e intérpretes que ayudan al programador en su tarea de desarrollar la base de lo que se convertirá, en la mayoría de los casos en software de aplicación. Los lenguajes de programación como PHP, Java, C++ entre otros, son claros ejemplos de software de programación, los cuales permiten adaptar las necesidades del usuario para que un programa informático resuelva alguna situación de la vida diaria.

Para desarrollar un sistema de software informático es necesario plantear los requerimientos que el sistema debe resolver una vez desarrollado, es aquí donde radica la principal importancia del desarrollo de software de programación y aplicación.

- **Software de aplicación:** El software de aplicación es creado para satisfacer una problemática o necesidad específica de una persona o un grupo de personas. Como ejemplo claro se tienen las aplicaciones para los dispositivos móviles, conocidas popularmente por *Apps* (del inglés Application), dichas aplicaciones funcionan en un sistema completo cumpliendo una o varias tareas específicamente. Algunos ejemplos son:
  - Editores de fotografías
  - Editores de música
  - Sistemas ofimáticos

Las aplicaciones de software existen para apoyar en algún sentido las tareas diarias de las personas, ya que una aplicación se concentra en realizar una tarea y hacerla bien, esto es quizás lo que hace que existan un sin fin de aplicaciones disponibles. En este trabajo se encuentra el desarrollo de una aplicación web que apoyará la administración del equipo en el laboratorio.

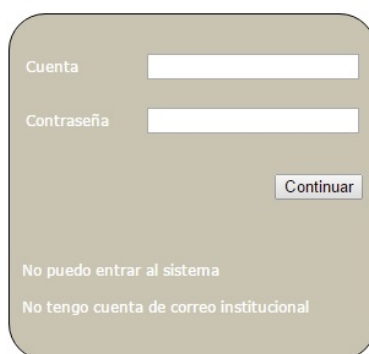
### 2.1.2. Software para la administración

Hoy en día el software de aplicación orientado a la administración, está siendo utilizado por muchas personas, empresas, instituciones y gobierno para facilitar tareas como procesos de facturación, inventarios, ingresos, egresos, nómina, bibliotecas digitales entre otros. La UACM no se queda fuera de utilizar software de aplicación para la administración de algún activo, algunos de los más usados y que servirán como ejemplo de lo que se quiere lograr son: Ex libris (sistema bibliotecario) y el sistema de administración de estudiantes.

En la UACM como en cualquier otra institución se utilizan distintos sistemas vía software para administrar cualquier rubro, en este caso se detallarán dos de ellos para tener una referencia de cómo funciona un sistema de administración vía software, solo se mencionarán dos pero es seguro que la universidad utilice muchos más.

#### 2.1.2.1. Sistema de estudiantes UACM

El sistema de estudiantes de la UACM es un ejemplo de un sistema de administración vía software. Dicho sistema está desarrollado para administrar la información de todos los estudiantes en la institución, de la misma forma se pueden realizar consultas del historial académico, inscripciones a las asignaturas de cada semestre e impresión de tira de materias entre otras acciones. Dicho sistema maneja una cuenta única por estudiante, para ingresar al sistema se requiere de una cuenta y una contraseña. Como se puede ver en la figura 2.1, este tipo de seguridad básica es muy popular en los sistemas informáticos y será utilizada para el sistema SAMLAB UACM-SLT v1.0.



El formulario de acceso al sistema de estudiantes de la UACM se muestra en un recuadro con un fondo gris claro y bordes redondeados. En la parte superior, hay dos campos de entrada de texto etiquetados como "Cuenta" y "Contraseña". Debajo de estos campos, hay un botón rectangular con el texto "Continuar". En la parte inferior del formulario, hay dos líneas de texto que sirven como enlaces o mensajes de ayuda: "No puedo entrar al sistema" y "No tengo cuenta de correo institucional".

Figura 2.1: Representación gráfica de acceso al sistema de estudiantes de la UACM

### 2.1.2.2. Sistema bibliotecario UACM

La biblioteca de la UACM utiliza un sistema de administración vía web para la gestión del acervo bibliográfico de todos los planteles, al momento de realizar una consulta el sistema busca en todas las bibliotecas de cada plantel y regresa la información a la pantalla cliente. La definición formal del sistema Exlibris tomado de su sitio web oficial es el siguiente: "Sistema con capacidad de desarrollar avanzadas soluciones de software que permite que una institución maneje y procese en su totalidad todo el espectro de material bibliográfico para obtener la información que necesita en cualquier momento y lugar"

En la figura 2.2 se muestra el sistema bibliotecario el cual está diseñado para realizar búsqueda de material bibliográfico a través de varios campos de búsqueda: búsqueda básica, búsqueda multicampo, búsqueda multibase y búsqueda avanzada. El sistema también es capaz de limitar la búsqueda a un idioma, un rango de tiempo, un formato y una biblioteca específica. Un sistema de administración bibliotecario vía web como Exlibris tiene muchos beneficios, algunos de ellos son: consulta de material bibliográfico disponible en cada plantel, centralización de información para consulta rápida y fácil, mejor administración y un control de todo el material bibliográfico en todas las sedes de la UACM.

**Búsqueda básica**

<b>Escriba una palabra o frase</b>	<input type="text"/>
<b>Campo de búsqueda</b>	Todos los campos ▾
<b>Palabras adyacentes?</b>	<input type="radio"/> No <input type="radio"/> Si
<b>Base de búsqueda</b>	CATALOGO UACM ▾
<input type="button" value="Enviar"/> <input type="button" value="Limpiar"/>	

---

**Limitar búsqueda a:**

<b>Idioma:</b>	Todos ▾	<b>Del año:</b>	<input type="text"/>	<b>Al año:</b>	<input type="text"/>
<b>Formato:</b>	Todos ▾	<b>Biblioteca:</b>	Todas ▾		

Figura 2.2: Sistema bibliotecario de la UACM con el software Ex Libris

## 2.2. Enfoque de desarrollo

Para comenzar el desarrollo de cualquier software es indispensable tener un puente de comunicación entre los desarrolladores y el usuario final (o cliente), dicho puente consiste en analizar y recaudar todos los requisitos que el software deberá cumplir para satisfacción del cliente, el objetivo principal es el común acuerdo entre el cliente y los desarrolladores para el comportamiento del sistema. En otras palabras los requerimientos identifican el *qué*, mientras que el diseño establece el *cómo* del sistema [3]. Existe

un estándar producido por la organización más importante a nivel internacional IEEE (Institute of Electrical and Electronic Engineer) llamado IEEE Std 830-1998, el cual habla de los requerimientos de software en su apartado SRS (Software Requirements Specifications); dicho estándar describe las características que deben tener los requerimientos, los tipos de requerimientos, así como lo que se debe tomar en cuenta para elaborarlos y documentarlos. Los requerimientos indican a los diseñadores qué funcionalidades y qué características va a tener el sistema resultante. Las características que deben cumplir los requerimientos según el estándar IEEE 830 son: deben ser correctos, deben ser consistentes, deben estar completos, deben ser realistas y deben ser verificables [3]. La documentación de los requerimientos incluye varios medios, el que se utilizó fue una entrevista verbal general con el usuario o el cliente del sistema (en este caso el administrador del laboratorio B-404), para recolectar los requerimientos para el desarrollo del sistema SAMLAB UACM-SLT v1.0, de esta forma, se recurrió a realizar una entrevista con el administrador del laboratorio utilizando una comunicación verbal lo más organizada posible y así generar un documento que defina de forma completa, precisa y verificable los requisitos del diseño y el comportamiento del sistema. La documentación de la estructura de los requerimientos de este trabajo se realizará tomando en cuenta aspectos importantes y muy específicos del estándar IEEE Std 830-1998. Debido a que es necesario contar con permisos de socio activo de la institución para tener acceso al estándar original, se tomó como referencia una traducción del mismo cuyo acceso es libre [4].

### 2.2.1. Análisis general de requerimientos

A continuación, se describen los requerimientos para el sistema, los cuales están enfocados en lo que el usuario final (administrador del laboratorio B-404) espera que haga el software SAMLAB UACM-SLT v 1.0. Los requerimientos recaudados servirán para especificar el diseño de éste trabajo los cuales serán descritos mas adelante.

#### 2.2.1.1. Introducción:

- **Propósito:** El sistema desarrollado deberá registrar las actividades préstamo-devolución, tomando en cuenta la forma actual de gestión del proceso.
- **Alcance:** El sistema de software SAMLAB UACM-SLT v1.0, debe ser capaz de apoyar al administrador para realizar el proceso-préstamo devolución, generando registros de préstamos activos e historial de préstamos realizados. En este caso el sistema sustituye las papeletas por una base de datos, no al actor de las acciones realizadas.

#### 2.2.1.2. Descripción global:

- **Perspectiva del producto:**

- \* El sistema de software debe contener un lugar donde se almacene la información relacionada al equipo y a los estudiantes, así como los préstamos activos y un historial general de préstamos.

- \* Se debe contar con acceso por usuario y contraseña para cada administrador.

- \* La recolección de datos debe operar de manera certera utilizando alguna tecnología viable.

- \* Las papeletas de préstamo desaparecerán.

- **Funciones del producto:**

- \* Se considerarán 3 principales funciones: Registros (estudiantes y equipo) Operaciones (préstamo, devolución y cambios) y Consultas (préstamos activos e historial)

- **Características de los usuarios:**

- \* Los usuarios que operarán el sistema SAMLAB UACM-SLT v1.0 serán de 2 tipos: administradores de laboratorio y profesores.

- **Restricciones:**

- \* El estudiante deberá mostrar siempre la credencial re-sellada de estudiante activo para poder realizar un préstamo, y corresponderá un préstamo por estudiante a la vez.

- \* El sistema deberá mostrar la situación del estudiante con una pestaña de activo/inactivo la cual será gestionada por el administrador del laboratorio, y consistirá en que el estudiante debe cumplir dos cosas importantes: credencial re-sellada del año en curso y debe estar inscrito a por lo menos una materia de las carreras de ingeniería.

- \* El sistema deberá permitir realizar un préstamo de hasta 10 equipos

Los requerimientos del administrador mostrados anteriormente serán los objetivos a cumplir en este trabajo, es momento de comenzar a estructurar la solución de forma sistemática para poder atacar cada parte e ir desarrollando el sistema, lo primero que se realizó fue dividir el sistema en bloques para organizar la solución y se realizó como sigue.

### 2.2.2. Módulos que conforman al sistema SAMLAB

El sistema SAMLAB UACM-SLT v1.0 está conformado por tres módulos importantes: adquisición de datos, interfaz con el usuario, y procesamiento y almacenamiento de información; cada módulo está formado por estructuras independientes y al mismo tiempo conviven todas las partes formando el sistema en su totalidad. Para cambios futuros en el sistema SAMLAB, bastará con elegir un módulo, realizar los cambios o mejoras pertinentes y volver a incluirlo. Cada módulo incluye submódulos que son identificados como extensiones en este documento. La figura 2.3 muestra los módulos y la relación entre sus partes.

Los módulos que se proponen en la figura 2.3, fueron creados tomando como referencia el proceso manual *préstamo-devolución*. El módulo de adquisición de datos se realiza con un lector de códigos de propósito general, la interfaz del usuario se realizó a través de una hoja de estilo CSS y HTML, y el procesamiento y almacenamiento de información se realiza con PHP Y Javascript, la información es almacenada en una base de datos utilizando comandos SQL.

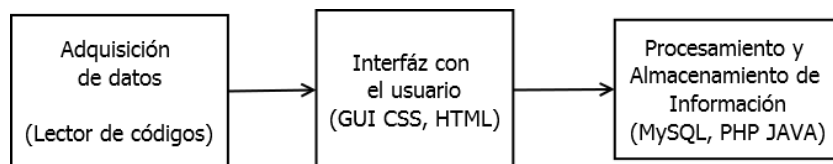


Figura 2.3: Módulos que conforman el sistema SAMLAB UACM-SLT v1.0

La organización del sistema en módulos es muy importante para los siguientes desarrolladores y para el administrador del sistema mismo, ya que habrá acciones que no tendrá que repetir e información que podrá consultar de manera ordenada, las acciones en concreto son las siguientes:

- El administrador podrá consultar la lista de equipo y lista de estudiantes, al registrar un estudiante nuevo en el sistema, la información se agregará a la lista correspondiente.
- El sistema garantiza que el estudiante que solicite préstamo este vigente como estudiante activo de la UACM SLT, ya que el sistema mostrará

la situación de cada usuario en la pantalla, esta información es cargada al sistema previamente por el administrador del laboratorio B-402.

- El administrador puede consultar la tabla de estudiantes, la tabla material, tabla de préstamos activos e historial de préstamos realizados. Esta es una gran ventaja ya que tendrá información clara y ordenada en cualquier momento.

### 2.2.3. Módulo de adquisición de datos

La recopilación de datos necesaria para el funcionamiento del sistema se realiza a través de un lector de códigos de barras, el hardware utilizado es de la marca EC Line de propósito general, dicho lector soporta lecturas tipo code-128, code-11, code-39 y ASCII completo. Los códigos que tendrá que leer el aparato son: la matrícula del estudiante (elemento único por alumno) y el código creado para el equipo electrónico (elemento único por dispositivo).

#### 2.2.3.1. Matrícula del estudiante

La credencial del estudiante de la UACM contiene los datos correspondientes al alumno como son: fotografía, carrera, plantel y matrícula; este último elemento es el más importante para este trabajo, ya que dicha información estará relacionada en cada préstamo y devolución en todo momento en el sistema. En el reverso de la credencial se encuentra el código de barras lineal de aplicación general del tipo code 39 o (Code 3 de 9), los caracteres que guarda este código están mostrados en la figura 2.4. Los primeros 2 dígitos corresponden al año de registro, los siguientes 3 dígitos corresponden al plantel y los siguientes 4 dígitos son el número único asignado a cada estudiante.

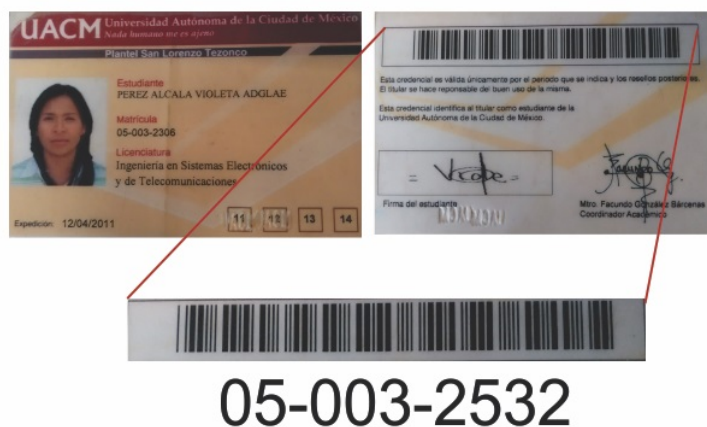


Figura 2.4: Código que representa la matrícula en la credencial de estudiante de la UACM

### 2.2.3.2. Creación de código para el equipo

El segundo dato importante a manejar, es el código referente al equipo disponible en el laboratorio, dicha representación de barras lineal se creó para identificar de manera única a cada dispositivo electrónico en el laboratorio B-402. Este código utiliza codificación 128, a diferencia del código de la matrícula (code 39), se optó por este tipo de codificación ya que genera un gráfico más corto y es legible para el lector de códigos disponible. Dichos códigos se imprimieron en etiquetas adhesivas y fueron colocados en las carcasas del equipo electrónico. Como prueba, se utilizaron tres kits de experimentación que están conformados por: 1 osciloscopio, 1 multímetro, 1 fuente de poder y 1 generador de funciones. La codificación consta de 14 símbolos que se asignaron como sigue: el año de registro en el sistema, el plantel donde se registro, la marca, el modelo y dos cifras numéricas consecutivas, la codificación se forma como se muestra en la figura 2.5. Los códigos de barras para cada dispositivo fueron creados con un generador en línea, gratuito y de propósito general de la empresa TEC-IT disponible en: <http://barcode.tec-it.com/es/barcode-generator.aspx>. Se llenó un formulario sencillo mostrado en la figura 2.5 inciso (a) para la creación del código tipo code 128. En la misma figura se muestra un ejemplo de identificación de un código de prueba. Por ejemplo, el código de prueba que corresponde al código de la figura 2.5 sería: (b) un multímetro registrado el año 2015, (c) UACM-SLT, (d) marca meterman, (e) con modelo 38XR9 y (f) número de dispositivo 10. Cabe mencionar que la codificación y el etiquetado fue llevado a cabo solo en tres kits de experimentación de todo el equipo disponible en el laboratorio, para realizar las pruebas necesarias con el software, se tendrían que crear todos los códigos siguiendo las especificaciones propuestas en este trabajo.

- **15:** Año de registro del equipo (2 cifras).
- **003:** Plantel asignado en la UACM, así como los dígitos 003 (tres cifras) de la credencial de estudiante que hacen referencia a San Lorenzo Tezonco.
- **ME:** Las primeras dos letras corresponden a la marca del equipo, ya que existen diferentes marcas para un mismo equipo, las letras propuestas son: BK para Bkprecision, ES para Escort, AG para Agilent, ME para Meterman, TX para Tektronix, NI para National Instruments, FE para Feedback (2 letras).
- **Modelo:** Modelo o número de serie del equipo, en algunos casos solo se tomaron los últimos dígitos del mismo (5 dígitos).
- **Activo:** Número de activo en el laboratorio (2 cifras)

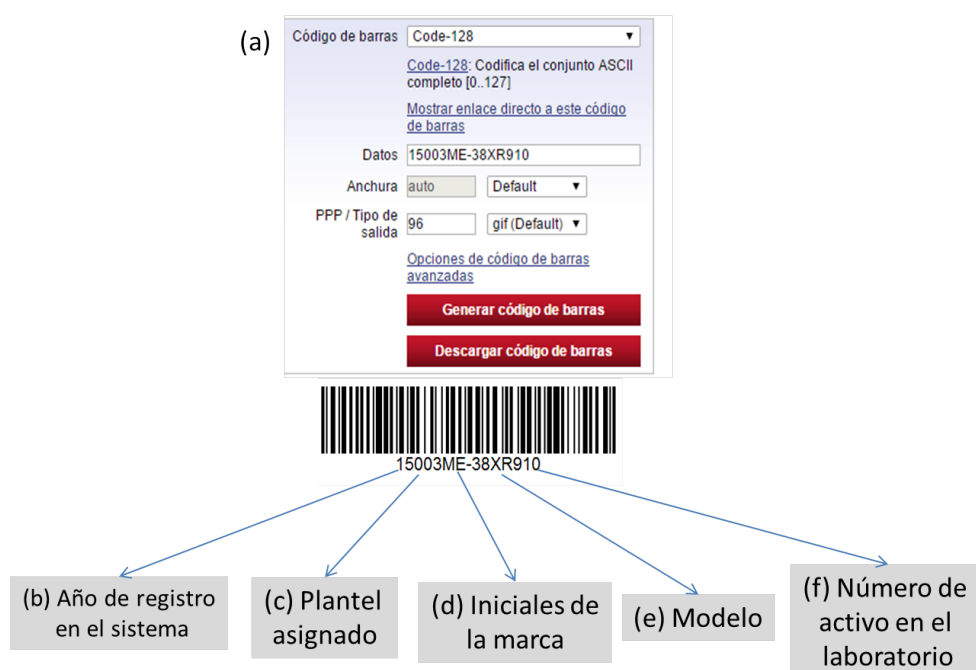


Figura 2.5: Creación de códigos de equipo

#### 2.2.4. Módulo de interfaz con el usuario

La interfaz con el usuario es el canal por el cual el administrador interactúa con el servidor para el cumplimiento del proceso préstamo y devolución. En el entorno informático este tipo de interfaz es denominada GUI (Graphic User Interface), el diseño de dicha interfaz es muy importante para este trabajo ya que será la puerta de entrada para que el usuario maneje el sistema SAMLAB UACM-SLT v1.0 a su favor así como lo propone Alliey: *El diseño de una GUI se dirige hacia la interacción entre usuario y máquina, el dominio del diseño es el dominio de la interface* [3].

En la figura 2.6, se observa la primera interfaz gráfica(GUI) que el administrador verá cuando trate de ingresar al sistema, cada administrador del laboratorio contará con una cuenta (usuario y contraseña) para poder ingresar al sistema y realizar operaciones. Existe un superusuario quien será el encargado de crear cuentas para cada nivel de usuario del sistema, además de poder crear cuentas provisionales; también tendrá acceso total a la base de datos SAMLAB.

La construcción de cada interfaz gráfica que verá el administrador, fue desarrollada utilizando el lenguaje para páginas web HTML (Hyper Text Markup Language) en su versión HTML 4.01, la versión más actual es HTML 5, no se utilizó debido a que HTML 4.01 satisfacía todas nuestras necesi-

dades, solo por mencionar algunas mejoras que nos brinda HTML 5 es que se puede incluir video y audio directamente en la web [4]. HTML está compuesto por una serie de etiquetas que el navegador interpreta y muestra de forma gráfica en la pantalla del cliente, de esta forma la información que se muestra siempre esta organizada, dándole estructura y contenido.

La parte de presentación de colores, decoración y organización visual de la GUI, está a cargo de CSS(Cascading Style Sheets), el cual es un mecanismo simple que se encarga de definir reglas aplicadas al documento HTML, para definir cómo se mostrará la página al cliente, una de sus principales bondades de utilizar una hoja de estilo CSS, es que la estructura del contenido completo está centralizado y los cambios que se realicen se muestran de manera general o de forma específica dependiendo el caso [4].



Figura 2.6: Interfaz gráfica del sistema SAMLAB UACM-SLT v1.0

La figura 2.6 corresponde a la interfaz gráfica que corresponde al acceso al sistema SAMLAB UACM-SLT v 1.0, el entorno visual que usa imágenes, figuras y texto es lo que se llama una GUI [5].

### 2.2.5. Almacenamiento y procesamiento de información

Para que el procesamiento de información que se lleva a cabo en el sistema SAMLAB UACM-SLT v1.0 sea objetivo, se construyó una base de datos desarrollada con MySQL, para explicar el funcionamiento considere el siguiente caso:

El estudiante solicita al administrador el equipo para préstamo, en ese momento el sistema debe ser capaz en primera instancia de reconocer al administrador a través de un usuario y contraseña, enseguida tendrá que reconocer la matrícula del estudiante en la tabla de estudiantes activos creada en la base de datos. Ya que realizó dichas validaciones se realizará el registro del equipo a usar, en una nueva tabla llamada préstamo, la cual se irá construyendo con el equipo solicitado por el estudiante, el préstamo debe estar activo en el sistema hasta que el estudiante regrese todo el equipo prestado. Por último, el administrador realizará el proceso de devolución del equipo y cerrará el proceso en el sistema.

Mientras ocurre el proceso anterior, el sistema debe ser capaz de relacionar datos entre tablas y crear una nueva tabla estática llamada préstamo, con estructura de registros que van cambiando conforme se van llenando los campos requeridos. Como se observa, se necesita de un lenguaje de programación y un manejador de bases de datos que cumplan con las características siguientes:

- Se requiere una respuesta rápida, ya que cada que se consulta o guarda un dato, el tiempo de respuesta debe de ser casi inmediato, por los que las tareas a realizar son continuas
- Requiere administración basada en usuarios y privilegio dentro del ambiente de MySQL, ya que es elemental contar con un filtro de acceso para el manejo de información.
- Opción de conectividad TCP/IP. Por seguridad física del sistema es necesario considerar la conexión remota, consideremos el caso en el que se requiere hacer modificaciones al software o mejoras al sistema.
- Datos siempre disponibles y confiables

De acuerdo a los requerimientos planteados dentro del desarrollo, se utilizó SQL (Structured Query Language) para el manejo de información y para facilitar las operaciones de manera gráfica, se utilizó un DBMS(Data Base Management System), versión 5.1.73 con licencia GPL(General Public License). Existe una nueva alternativa a MySQL, llamada MariaDB con mayores funciones que MySQL, sin embargo para el desarrollo de este trabajo se ocupo MYSQL ya que aun no estaba disponible la versión de MariaDB, la cual se puede utilizar migrando en cualquier momento el desarrollo de MySQL a MariaDB, ya que cada versión de MariaDB esta basada en una versión homologa de MySQL [6].

## 2.3. RUP como metodología de diseño para el software SAMLAB

El Proceso Unificado de Rational (RUP) ver figura 2.7, es un compendio de pasos para desarrollar software que ayuda a diseñar un sistema informático de cualquier tamaño, utilizando una metodología continua de pruebas y realimentación, garantizando el cumplimiento de estándares de calidad [7]. RUP captura muchas de las mejores prácticas para diseñar software moderno, de modo que es adecuado utilizar sus bondades para una amplia gama de proyectos y organizaciones. El proceso puede ser representado en dos dimensiones:

- **El eje horizontal:** representa tiempos, y muestra los aspectos dinámicos de los procesos, se expresa en términos de ciclos, fases, iteraciones e hitos.
- **El eje vertical:** representa los aspectos estáticos del proceso, se describe en términos de actividades, trabajadores y flujos de trabajo.

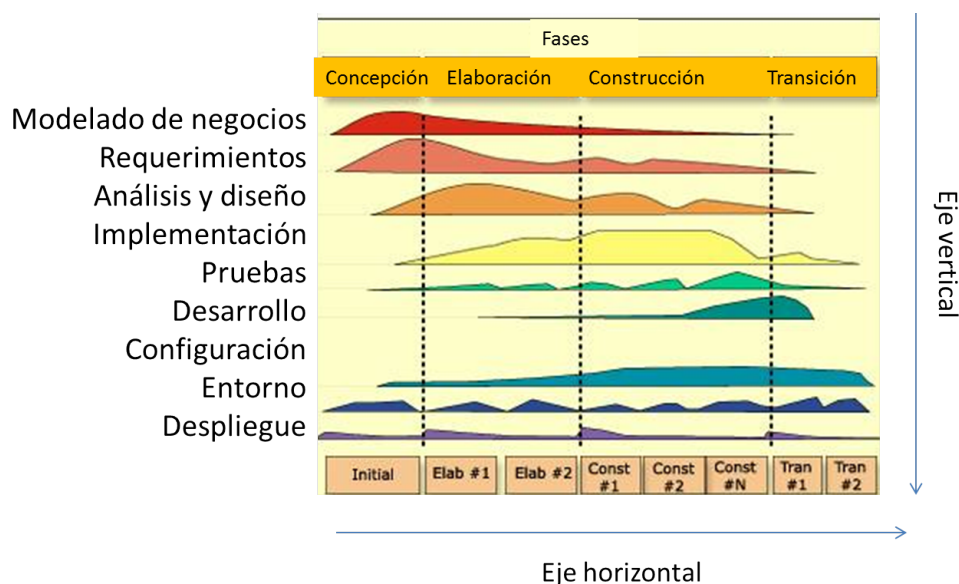


Figura 2.7: Diagrama de negocios RUP

RUP es una técnica utilizada para el desarrollo de software, debido a los aspectos siguientes:

- El desarrollo es dirigido por casos de uso
- Es centrado en la arquitectura

- Es iterativo e incremental

Para que el desarrollo de un proyecto se lleve a cabo utilizando RUP se deben seguir las siguientes fases:

- Concepción o inicio: Durante esta fase se definen los requisitos de negocio, los alcances del proyecto, los actores, casos de uso, se estiman gastos de recursos, tiempo de ejecución y estimación de riesgos.

*Hito: Plantear el objetivo general del ciclo de vida del proyecto*

- Elaboración: El propósito de esta fase es establecer los cimientos de la arquitectura desde el punto de vista del equipo de desarrollo. En este punto se define la solución técnica que el problema debe contener y los casos de uso críticos identificados en la fase de inicio. El objetivo principal es crear un plan fiable para la fase de construcción.

*Hito: Generar un marco de desarrollo general de todo el proyecto con profundidad mínima*

- Construcción: El propósito de esta fase es alcanzar el desarrollo operacional del proyecto en su nivel más aceptable, es decir, conseguir la primera versión; deben estar descritos todos los requisitos y componentes que se utilizaron para la construcción del proyecto. En esta fase se hace uso de metodologías iterativas de modelos completos (descripción de casos de uso, análisis, diseño y diagramas de despliegue, entre otros).

*Hito: Tener un producto listo para ser probado por el cliente o usuario final*

- Transición: Esta fase es la meta a corto plazo que se estaba esperando, es cuando el sistema, en su versión beta, pasa del área de desarrollo al área de producción. Es aquí donde se recibirá la mayor realimentación por parte del usuario para posibles mejoras del producto.

*Hito: Realizar pruebas al sistema y considerar mejoras a corto plazo*

Cada fase RUP se puede descomponer en cuantas iteraciones sean necesarias por el equipo de desarrolladores, el objetivo de cada iteración es desarrollar algún producto de software funcional, que pueda ser mostrado y resulte un componente importante para el desarrollo del proyecto. RUP utiliza el principio "*divide y vencerás*".

RUP junto con UML consideran buenas prácticas de desarrollo de software, su lema es "*mejoras continuas en cada iteración*", en este desarrollo se han adoptado en la medida de lo posible dichas prácticas las cuales son:

- Gestión de requerimientos y control de calidad continua

- Permitir el desarrollo iterativo e incremental
- Utilizar casos de uso
- Permitir el reuso de partes comunes ya diseñadas
- Generar un producto más robusto cada vez que existe una iteración, los errores se van corrigiendo en cada iteración.
- Modelar con gráficos visuales utilizando UML
- Manejar un enfoque estructurado para decidir quién va a hacer qué, cuándo lo va a hacer y cómo lo va a hacer.

Dado que el desarrollo del sistema SAMLAB UACM-SLT v1.0 se encuentra en la primera versión, se comenzó utilizando recursos mínimos en cada módulo para asegurar el funcionamiento base. Existen 2 fases del proceso RUP que se llevaron a cabo en este trabajo, llamadas iteración de fase 1 y 2 respectivamente. La primera fase se encargó de obtener tablas con respecto a los estudiantes y el equipo, mostrado en la tabla 2.1 y la segunda corresponde a la realización de un proceso completo préstamo-devolución mostrado en la tabla 2.2. Las dos fases están complementadas con una sub-iteración cada una, en las tablas siguientes se observan los requerimientos del caso, los alcances de la iteración y el hito que se pretende alcanzar.

Iteración de fase 1	
<b>Iteración de fase1</b>	<b>Requisitos:</b> Servidor, base de datos, códigos de matrícula y equipo <b>Alcances de la iteración:</b> Que el administrador pueda realizar registro, eliminación y consulta de estudiantes y equipo. <b>Definición de casos de uso:</b> Caso estudiantes (Registrar-Eliminar-Consultar) Caso equipo (Registrar-Eliminar-Consultar) <b>Actores:</b> Estudiante y administrador del laboratorio B-402
<b>Iteración de fase 1.1</b>	<b>Alcances de la iteración:</b> Que el administrador pueda realizar edición y consulta particular de un registro específico de estudiantes y equipo. <b>Definición de casos de uso:</b> Editar estudiante, editar equipo, consultar estudiante, consultar equipo
<b>Hito:</b>	<b>Tener una base de datos con 2 tablas, estudiantes y material, para que el administrador pueda realizar las siguientes acciones de administración:</b> <b>Registrar-Eliminar-Consultar-Editar</b>

Tabla 2.1: Proceso RUP 1 fase 1

Iteración de fase 2	
Iteración de fase 2	<p>Requisitos: Servidor, base de datos, tabla estudiantes, tabla material, tabla préstamos activos, tabla historial de préstamos.</p> <p>Alcances de la iteración: Que el administrador pueda realizar un préstamo y una devolución de equipo, y crear un registro en la base de datos</p> <p>Definición de casos de uso: Realizar préstamo, Realizar devolución</p> <p>Actores: Estudiante y administrador del laboratorio B-402</p>
Iteración de fase 2.1	<p>Alcances de la iteración: Que el administrador pueda realizar cambios en un préstamo y cambios en una devolución</p> <p>Definición de casos de uso: Consultar préstamos activos y Consultar historial de préstamos</p>
Hito:	<p>Tener una base de datos con 4 tablas, estudiantes, material, préstamo y préstamos activos</p> <p>para que el administrador pueda realizar las siguientes acciones de administración: Realizar un préstamo y una devolución, además de consultar préstamos activos e historial de préstamos realizados.</p>

Tabla 2.2: Proceso RUP 2 fase 2

## 2.4. UML como lenguaje de diseño de software

No es conveniente comenzar un desarrollo de software sin antes diseñarlo, visualizarlo, definir requerimientos, definir casos de uso y documentarlo sistemáticamente, de esta forma se tiene una guía que ayuda a no perderse a la hora de programar y de cumplir con las especificaciones del sistema al pie de la letra. Existen diferentes tipos de modelado de diseño de software una de ellas es UML. Este lenguaje está basado en gráficos que permite especificar, construir, visualizar y documentar los objetos de un sistema para ser programado [8]. Este tipo de lenguaje utiliza representaciones gráficas para describir los casos, algunos diagramas utilizados son: diagrama de caso de uso, diagrama de clases, diagrama de estados, diagrama de actividades entre otros. Dichas representaciones gráficas permiten expresar el comportamiento de un módulo antes de programarlo y definir actividades, estados e iteraciones antes de escribir una sola línea de código. No es necesario incluir todos los diagramas disponibles en UML para algún desarrollo, en este caso se hará uso de 3 tipos de diagramas: Diagramas de casos de uso, diagramas de actividades y diagramas de iteración específicamente de diagramas de secuencia, como se observa en la figura 2.8.

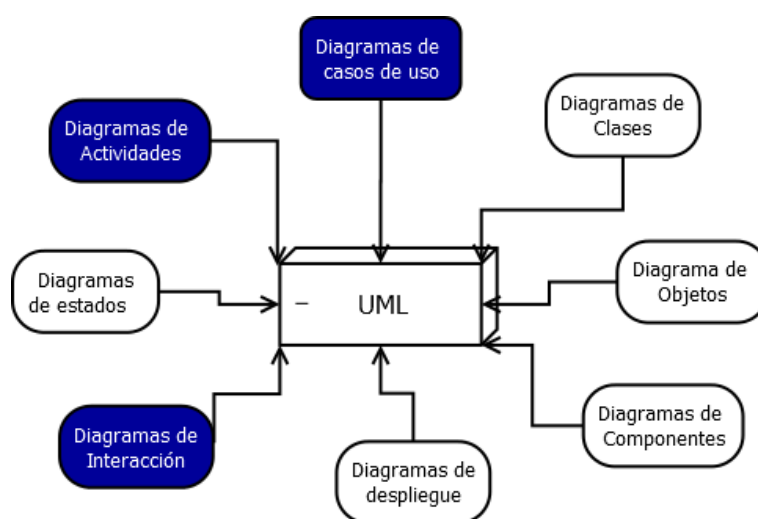


Figura 2.8: Esquema de diagramas disponibles en UML

- **Diagramas de casos de uso.**- Es un escenario que refleja el comportamiento de un sistema de software o de hardware y se utilizará para describir los siguientes casos: registrar, eliminar y editar registro de estudiante, además registrar, eliminar y editar registro de equipo, préstamo, devolución.
- **Diagramas de actividades.**- El diagrama de actividades permite decidir el orden en el que se lleva a cabo un caso de uso, su representación gráfica es muy parecida a un diagrama de flujo, y es muy utilizado en programación orientada a objetos.
- **Diagramas de iteración.**- Existen 2 tipos de diagramas de iteración: de secuencia y de colaboración, en este caso se utilizaron diagramas de secuencia, los cuales van a permitir describir el comportamiento de un caso de uso y la iteración con sus objetos.

Este tipo de lenguaje de modelado ayuda a minimizar algunos de los siguientes problemas que se presentan con mucha frecuencia al momento de desarrollar software:

- Los sistemas no responden a las expectativas de los usuarios
- Los programas fallan con cierta frecuencia
- La modificación del software es tarea difícil y costosa
- El aprovechamiento óptimo de recursos (personal, tiempo, herramienta, entre otros) no suele cumplirse.
- No se realiza la gestión de riesgos formalmente

## 2.5. Los casos de uso para el desarrollo del sistema

Los casos de uso son la forma gráfica que tienen los desarrolladores para comunicarse de una manera sencilla con el cliente (usuario final del sistema de software) [8], son de gran utilidad para describir casos específicos para garantizar que el sistema haga exactamente lo que se solicitó. El descubrimiento de los casos de uso es quizás la parte más importante del desarrollo del sistema SAMLAB UACM-SLT v1.0 ya que incluye la descripción detallada de requerimientos que se deben identificar, de manera que el caso de uso cumpla con lo que realmente va a programar el desarrollador. En la figura 2.9 se muestran de manera general los casos de uso que forman parte de todo el sistema.

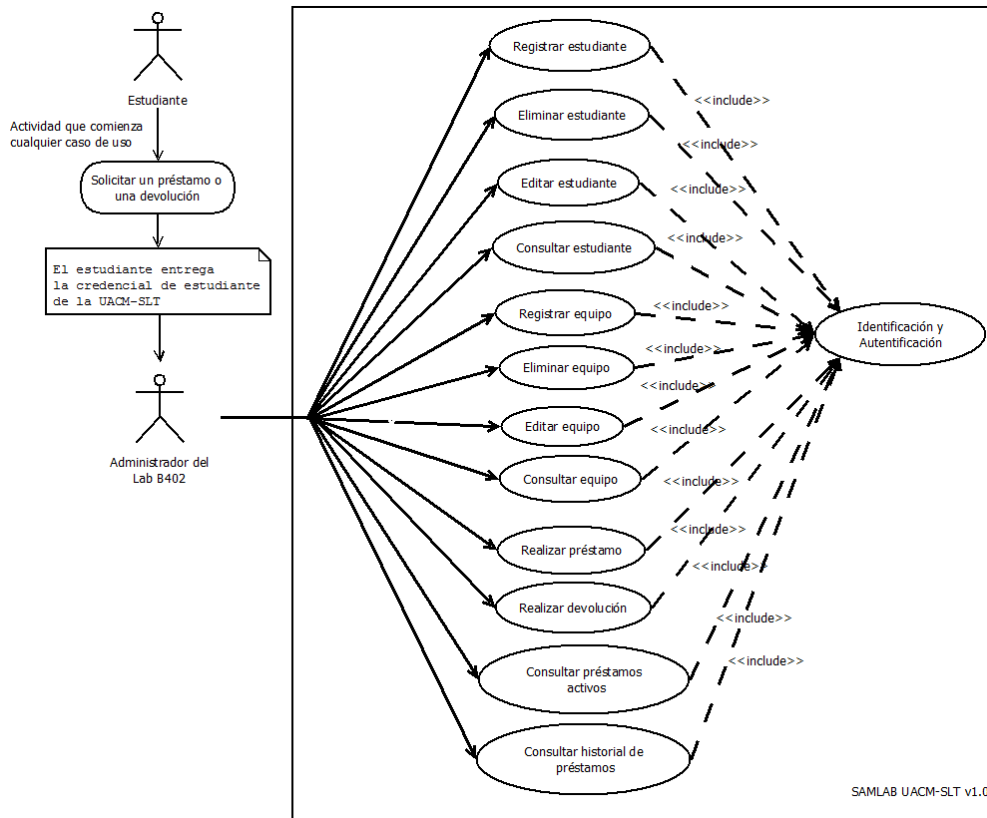


Figura 2.9: Diagrama de casos de uso en su nivel operacional básico

Los casos de uso mostrados en la figura 2.9 son la base del desarrollo, ya que contienen los cinco apartados con las que operará el sistema: Estudiantes-Equipo y Equipo-Préstamo-Devolución-Consultas. Cabe mencionar que existe un caso de uso muy importante y que es la llave de entrada al sistema SAMLAB UACM-SLT v1.0 este es: identificación y autentificación

de los usuarios-administradores en el sistema, pero el actor que comienza el proceso es un estudiante cuando solicita un préstamo o una devolución.

## 2.6. La red de computadoras

Para realizar la conexión de manera remota con el sistema SAMLAB UACM-SLT v1.0, se utilizó como herramienta la implementación de una red de computadoras conectadas a Internet, con la cual un equipo cliente pueda disponer del software de administración a través de un servicio web hacia un equipo servidor, para ello se explicará cómo llega la red a ser capaz de compartir información y trabajo en común.

### 2.6.1. Provisión de seguridad de red

Hoy en día cualquier tipo de red necesita contar con cierto tipo de seguridad para garantizar la integridad de sus datos, además de controlar el acceso de personas autorizadas a dicha red. El objetivo de seguridad en una red podría ser el siguiente:

Mantener la integridad, disponibilidad, privacidad, control y autenticidad de la información manejada por computadora, a través de procedimientos basados en una política de seguridad tales que permiten el control de lo adecuado [9].

Proveer de seguridad a una red de datos no es una tarea fácil. De acuerdo a la tabla 2.3, existen 2 grandes grupos de seguridad de red: seguridad de infraestructura (física) y seguridad de contenido (lógica)[9].

Seguridad de Infraestructura	Seguridad de Contenido
Desastres	Controles de acceso
Incendio	Identificación
Equipamiento	Roles
Inundaciones	Transacciones
Picos y Ruido electromagnético	Limitación a los servicios
Cableado	Control de acceso interno

Tabla 2.3: Tipos de seguridad en una red de datos, tomada de [10]

El sistema SAMLAB UACM-SLT v1.0 cuenta con algunos niveles de seguridad disponibles en cuanto a infraestructura y contenido. Por el lado

de la infraestructura se tienen aspectos de seguridad en el servidor y en el firewall con un resguardo personal, el equipo se encuentra montado en un rack de 19", utilizando la seguridad que ofrece la UACM-SLT en cuanto a instalación eléctrica evitando picos y ruidos de corriente. Por otro lado la seguridad de contenido está a cargo del firewall, el cual está montado en una computadora con recursos mínimos, el sistema de software utilizado para el firewall es IPCOP, quien es el encargado de detener los paquetes con posibles ataques. IPCOP es un firewall con posibilidades generales y gasto de recursos mínimos, a continuación se explica cómo se realizó la implementación de IPCOP como firewall.

### 2.6.2. IPCOP como Firewall

Para proveer de seguridad en la red del laboratorio B-404, se instaló el sistema IPCOP v2.1.7 el cual es una distribución de Linux para implementar software de tipo firewall o cortafuegos sobre una computadora con recursos básicos. El objetivo principal de IPCOP es revisar cada uno de los paquetes que deseen ingresar a la red SAMLAB, el acceso de dichos paquetes será controlado a través de reglas que se deben establecer previamente en el firewall. IPCOP tiene la bondad de adaptarse a las necesidades del usuario mediante diferentes opciones como pueden ser: port forwarding (re-direccionamiento de puertos), DMZ (demilitarized zone), pinholes (excepciones de tráfico entre redes), acceso seguro por SSL, traffic shaping (asignación de tráfico), URL filter (filtrado de URLs), entre otras posibilidades. El firewall está montado en un hardware dedicado con características básicas (procesador Intel Pentium 4, RAM 1 GB, 40 GB en disco duro), utilizando 3 tarjetas de red ethernet 10/100 Mbps. En total se administra el tráfico de paquetes en 3 redes. Se definió el acceso entre las 3 tarjetas de red como sigue: la tarjeta número uno corresponde a la red roja = entrada y salida a Internet, la tarjeta número dos corresponde a la red verde = red LAN para administración de servidor y la última tarjeta Red naranja = DMZ y granja de servidores. El acceso a la red SAMLAB está definido por el firewall, a través de creación de usuarios con prioridades de acceso al contenido del servidor, los usuarios definidos en este desarrollo son:

1. Superusuario: Acceso total al firewall y al servidor SAMLAB (puede realizar modificaciones en la base de datos) de manera local, remota o presencial.
2. Usuarios administradores: Acceso remoto mediante la interfaz gráfica al servidor SAMLAB, desde un cliente asignado en los laboratorios B-404 Y B-403, puede realizar altas, bajas, cambios, consultas, préstamos y devoluciones.

## 2.7. S.O. Centos 6 como servidor web

Un servidor, como la misma palabra lo indica, es una computadora que está al servicio de uno o varios dispositivos en una red interna o externa. Debido a que el administrador del laboratorio desde su ubicación del laboratorio B-404, deberá tener acceso al servidor que contenga el sistema SAMLAB en el laboratorio B-402 y desde cualquier punto de la red UACM-SLT, se instaló un servidor de aplicaciones web con servicio básicamente de HTTP, el cual es el protocolo de acceso cliente-servidor. Para instalar dicho servicio, se utilizó una distribución de sistema operativo linux, sin embargo, existen diferentes opciones para ello en primera instancia se encuentra la decisión entre Windows, Linux o MAC, esta es quizás la decisión más sencilla ya que descartamos servidor MAC por ser costoso en infraestructura, la opción Windows podría funcionar ya que existen distribuciones de Windows Web Server (por una módica cantidad), la última opción pero no la menos importante es utilizar un servidor basado en Linux en su distribución **Centos 6.0** (*Community Enterprise Operating System*) la cuál fue la versión más actual al realizar este trabajo. Centos es una distribución de Red Hat Enterprise Linux diseñado principalmente para ser un sistema operativo robusto de código abierto, gratuito y fácil de usar, a continuación se describen sus principales características:

- Estabilidad: Su estabilidad se debe a que ejecuta sólo las versiones básicas y más estables de software, para garantizar que no existan fallas de manera pronta, además no requiere reiniciar por configuraciones o actualizaciones del sistema.
- Seguridad: Pronta reacción ante alguna vulnerabilidad o fallos, a diferencia de windows que reacciona más lento. Además un servidor con base Linux/Unix es menos vulnerable por virus o malware.
- Libre y gratuito: Al ser un software derivado de Red Hat Enterprise (líder en servidores) podemos tener con Centos un software de clase empresarial gratuito.
- Flexibilidad: Centos y cualquier instalación de Linux puede adaptarse a las necesidades del usuario tanto como él desee, no importa si desea montar un servidor que atienda a 10 usuarios o que atienda a 10,000 usuarios o más, los desarrollos de dicho sistema operativo están preparados para soportar redes de cualquier tamaño, lo que no sucede en Windows server o cualquier otra distribución que en su versión más básica soportan de 5 a 75 usuarios máximo y de requerir mayor capacidad se tendrá que adquirir su siguiente versión.
- Económico en uso de recursos de cómputo: Para instalar Centos se

requiere un equipo de cómputo básico con requisitos mínimos, con procesador pentium 4, 512 MB de memoria RAM, 40 GB en disco duro.

Existen varias distribuciones de software libre que podrían haberse utilizado como sistema operativo para el servidor que se ocupa en este desarrollo, por ejemplo: Ubuntu y Fedora, pero sólo Centos tiene la oportunidad de contar con un sistema de grado empresarial de manera gratuita. De esta forma Centos 6.0 se instaló en un equipo con procesador AMD Phenom 3.0 GHZ, 2 GB en RAM, 500 GB en disco duro el cual estará funcionando como servidor web del sistema (ver figura 2.10).



Figura 2.10: Servidor SAMLAB UACM-SLT v 1.0

### 2.7.1. LAMP

LAMP es el conjunto de servicios y utilidades más instalados en servidores ya que integra una solución completa para comenzar un desarrollo web que podrá ir creciendo como el usuario lo desee. Los servicios disponibles en el servidor SAMLAB son: Linux (SO) + Apache (Servidor web) + MySQL/MariaDB (Gestor de bases de datos) + Perl/PHP/Python (Lenguaje de programación) Este conjunto de servicios serán indispensables para instalar el sistema en desarrollo, la instalación será por medio de consola y comandos importantes para que al final se tenga un servidor capaz de manejar intercambio de información con el usuario cliente.

## Capítulo 3

# Implementación de la red

*Ahora pueden construir lo que quieran,  
la web es una plataforma para que  
ustedes contruyan lo que imaginen*

Tim Berners-Lee

### 3.1. La UACM como proveedor de red

El plantel San Lorenzo Tezonco es el más grande de la UACM, su ER (Equipment Room) conecta 4 edificios vía fibra óptica donde se ubican los TR (Telecommunications Room) en cada edificio de la UACM-SLT, la infraestructura de red del edificio B es la encargada de proveer servicios de Internet al laboratorio B-404 y demás laboratorios, ya que a través de canales de cobre se puede tener acceso a Internet desde la red local del laboratorio B-404. En la figura 3.1 se muestra el diagrama general de llegada de la red de datos desde el ER de la UACM SLT al laboratorio B-404. Cabe mencionar que para comprender mejor la topología de red utilizada se muestra una visión general de red.

- Visión general de red

La figura 3.1 muestra una visión general de abastecimiento de la red de datos desde el ER de la UACM SLT, pasando por 2 cuartos de telecomunicaciones (TR) situados en el primer y tercer piso respectivamente. Los laboratorios B-402 y B-404 están conectados a un switch en el TR del tercer piso del edificio B y será por esta red que se comunicará el cliente (administrador del laboratorio) ubicado en el laboratorio B-402 y el servidor en el laboratorio de redes de computadoras ubicado en el B-404, F0 corresponde al tendido de fibra óptica.

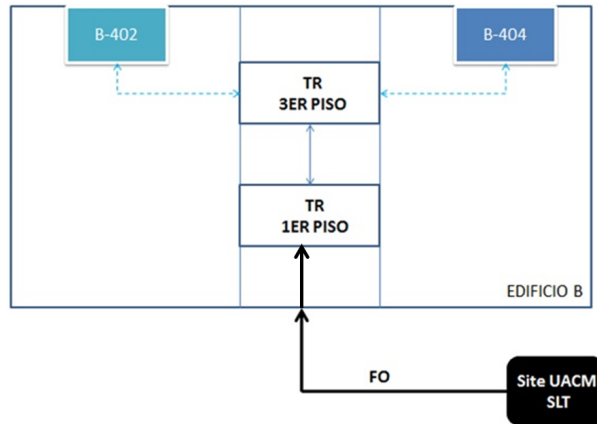


Figura 3.1: Esquema básico de abastecimiento de red a los laboratorios B-402 y B-404

- Topología Física

La topología física de red se muestra en la figura 3.2, desde el exterior hasta el primer elemento importante que es el firewall de la red, está representado por la leyenda IPCOP SERVER LAB B-404. El único acceso permitido a la red es a través de la red roja, cabe mencionar que no se tiene ningún control sobre esta red, ya que ésta es administrada por la coordinación de informática y de telecomunicaciones de la UACM.

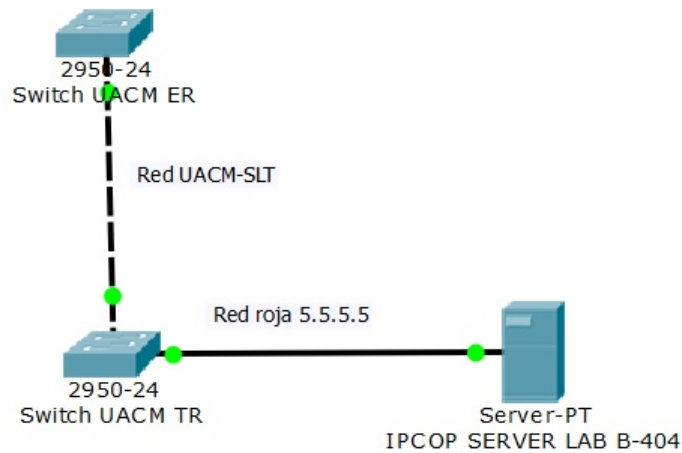


Figura 3.2: Topología física de última milla

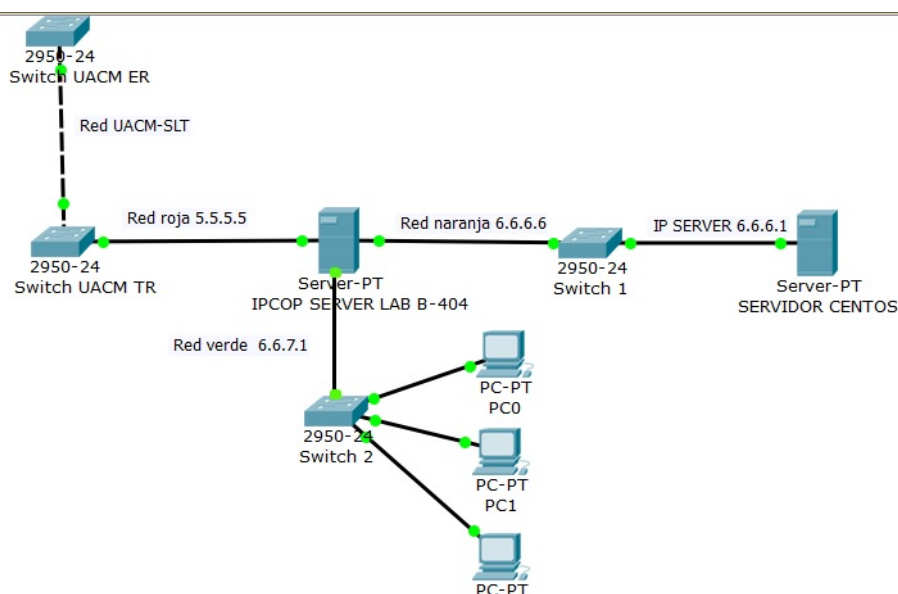


Figura 3.3: Topología física de la red SAMLAB

La red de datos que soporta el sistema SAMLAB se muestra en la figura 3.3, el tipo de red utilizado fue *estrella*, ya que cada estación de trabajo está conectada a un dispositivo central, por ejemplo en la red verde 6.6.7.1 el equipo central es un switch, el cual atiende a tres host cliente, de la misma forma el firewall IPCOP centraliza las comunicaciones de las tres redes implementadas en este trabajo (Red roja, Red verde y Red naranja). Las razones por las que se utilizó una topología de red tipo estrella, se debe a que permite agregar nuevos dispositivos fácil y rápido, además se pueden encontrar y prevenir fallos en la red de manera fácil, la reconfiguración es sencilla y la centralización de la red ocurre en todo momento.

Los accesos a la red SAMLAB se definieron desde la configuración del firewall IPCOP, a través de denegar todo el tráfico de entrada al servidor y haciendo excepciones con direcciones IP que el usuario administrador desea permitir, en la figura 3.4 se muestra la configuración que se realizó en el firewall para que los usuarios administradores tengan acceso desde una computadora maestro en el laboratorio B-404 y desde una computadora de reserva en el laboratorio B-403, también se puede tener acceso al sistema desde una computadora de administración en la red verde de la red SAMLAB.

#	Red Ifaz	Origen		Red Ifaz	Destino
1	VERDE	192.168.50.254	➤➤	IPCop	IPCop : http
2	VERDE	Green Network	➤➤	IPCop	IPCop : http
<b>Tráfico interno:</b>					
#	Red Ifaz	Origen		Red Ifaz	Destino
1	VERDE	Green Network	➤➤	NARANJA	Any : http
2	NARANJA	Orange Network	➤➤	VERDE	Green Network : http
<b>Reenvío de puertos:</b>					
#	Red Ifaz	Origen		Red Ifaz	Destino interno
1	Cualquiera	172.17.120.173 : 80	➤➤	NARANJA	192.168.51.10 : http
2	Cualquiera	Any : 80	➤➤	NARANJA	192.168.51.10 : http
3	Cualquiera	192.168.50.14 : 80	➤➤	NARANJA	192.168.51.10 : http
4	Cualquiera	172.17.142.49 : 80	➤➤	NARANJA	192.168.51.10 : http
5	Cualquiera	172.17.120.129 : 80	➤➤	NARANJA	192.168.51.10 : http
6	Cualquiera	192.168.50.19 : 80	➤➤	NARANJA	192.168.51.10 : http
7	Cualquiera	172.17.120.181 : 80	➤➤	NARANJA	192.168.51.10 : http
8	Cualquiera	172.17.120.187 : 80	➤➤	NARANJA	192.168.51.10 : http
9	Cualquiera	172.17.120.185 : 80	➤➤	NARANJA	192.168.51.10 : http
10	Cualquiera	172.17.120.186 : 80	➤➤	NARANJA	192.168.51.10 : http
11	Cualquiera	172.17.120.180 : 80	➤➤	NARANJA	192.168.51.10 : http
12	Cualquiera	192.168.50.6 : 80	➤➤	NARANJA	192.168.51.10 : http

Figura 3.4: Accesos permitidos a la red SAMLAB

El firewall proporciona a la red donde se encuentra el sistema SAMLAB seguridad de acceso, ya que nadie puede acceder al contenido del servidor si su dirección IP no se encuentra en la tabla de ruteo del firewall.

### 3.2. Instalación de servidor web HTTP

El servicio web que se utilizó es Apache versión 2.2.15, el cual tiene como objetivo principal el envío de páginas web estáticas y dinámicas a través de la WWW (World Wide Web) a un usuario cliente, en este caso el administrador del laboratorio B-402. La forma de compartir información se basa en colocar archivos y carpetas en la raíz de documentos apache /var/www/html, para que dicha información esté disponible para el cliente en cuanto lo solicite. Además de aceptar o denegar las peticiones del exterior, apache también lo hace de manera eficiente ya que puede aceptar un gran número de peticiones antes de colapsar, es decir, maneja ejecución multitarea para las páginas no encontradas, apache también tiene una serie de archivos precargados que restringen el acceso o redireccionamiento a páginas predeterminadas. Algo de suma importancia es la gestión de autenticación de usuarios y registro de logs lo cual será de gran utilidad para el desarrollo de SAMLAB. La instalación es relativamente sencilla, se realizó a través de la consola de

comandos de Centos y se utilizaron los comandos de la tabla 3.1.

Instalación de Apache	
Comando	Objetivo
<code>#yum install httpd</code>	Instalación de demonios HTTP
<code>#service httpd start</code>	Inicia el servicio HTTP
<code>#chkconfig httpd on</code>	Inicia Apache tras un reboot

Tabla 3.1: Comandos para instalación de apache

Para este paso el servicio web ha sido creado, para verificar su correcta instalación se debe de escribir en un navegador web la dirección IP local asignada al servidor y tendrá que abrir la página de bienvenida *Apache 2 Test Page* la cual es mostrada en la figura 3.5.

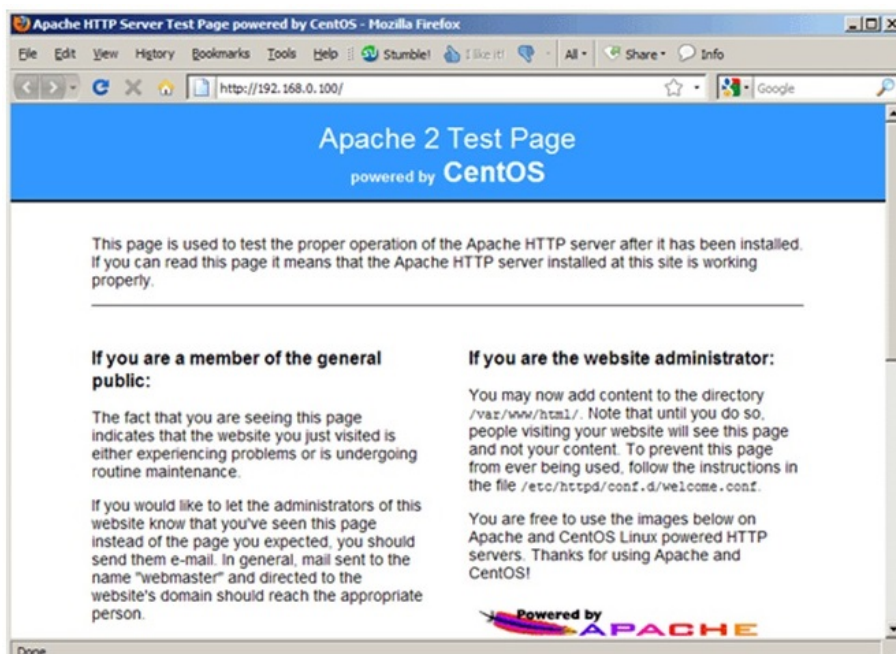


Figura 3.5: Servicio web apache

### 3.3. Instalación MySQL

Para el almacenamiento de la información se hará uso de MySQL v5.1.73 el cual es un manejador de contenidos de bases de datos, la instalación se realizó con los comandos mostrados en la tabla 3.2, además se agregó el acceso con autenticación de súper usuario (root) y la contraseña asignada.

Instalación de MySQL	
Comando	Objetivo
<code>#yum install mysql mysql-server -y</code>	Instalación de demonios MySQL
<code>#mysql admin-u root password "la que se definió"</code>	Definición de password para base de datos
<code>#service mysqld start</code>	Inicia MySQL tras un reboot
<code>#chkconfig mysqld on</code>	Inician los servicios MySQL

Tabla 3.2: Comandos de instalación MySQL

La base de datos es llamada SAMLAB, todas las tablas interactúan entre sí durante las operaciones realizadas por el administrador. Para acceder a Mysql y crear dicha base de datos primero habrá que autenticarse en el sistema Centos como superusuario(root) con la contraseña establecida anteriormente, los comandos a seguir se muestran a continuación.

```
[servidor@servidor SAMLAB]$ mysql u root p
Enter password: *****
mysql>
```

Ingresando a mysql se creó la base de datos que contiene todas las tablas que se utilizan en la operación, a continuación los comandos utilizados.

```
mysql> CREATE DATABASES SAMLAB;
Query OK, 1 row affected (0.03 sec)
mysql> SHOW DATABASES;
mysql>
```

Ahora se ha creado ya la base de datos donde se guardará toda la información del sistema SAMLAB, es momento de definir las tablas a utilizar, para continuar el proceso de creación y edición de tablas se utilizó un sistema de gestión de bases de datos llamado phpMyAdmin 4.4.7 (ver figura 3.6), aquí se editan y visualizan todas las tablas realizadas en la base de datos SAMLAB de manera gráfica.

### 3.4. Definición y creación de tablas con phpMyAdmin

La base de datos que se desarrollo tiene por nombre SAMLAB la cual forma parte del sistema SAMLAB UACM-SLT v1.0, está conformada por cinco tablas, la relación entre ellas depende del propósito que se tenga en cada momento. En la figura 3.6 se observan las relaciones entre las tablas, la tabla *admin* no tiene intercambio de información con ninguna tabla, pero es importante que el administrador se autentifique, la relación es indirecta ya que es la llave de acceso para que el usuario interactue con las otras tablas.

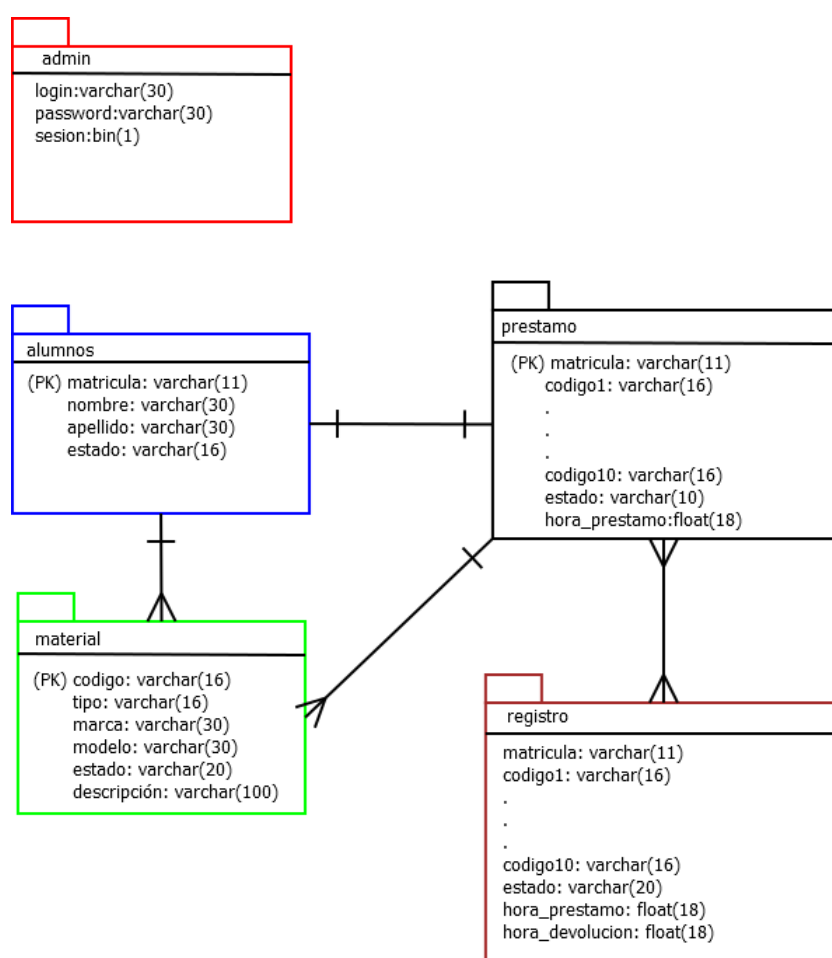


Figura 3.6: Diseño de las tablas en la base de datos SAMLAB

En la tabla *alumnos* se definió como llave primaria (primary key PK) el campo matrícula ya que este dato es único por estudiante y es el punto de relación entre tablas, por lo tanto la tabla *alumnos* se relaciona con la tabla

material de tipo uno a muchos (un alumno puede tener más de un equipo en préstamo), de la misma forma la relación entre la tabla *alumnos* y la tabla *prestamo* es uno a uno (un alumno solo podría tener un préstamo a la vez). La tabla *material* se relaciona con la tabla *prestamo* de forma uno a muchos (un préstamo puede contener más de un equipo) la llave primaria PK en esta tabla es el código del material. Por último la tabla *prestamo* se relaciona con la tabla *registro* de manera muchos a muchos (muchos préstamos pueden corresponder a muchos registros), en realidad en este sentido en la tabla *registro* no existe llave primaria PK porque esta tabla no tiene que intercambiar datos con las demás tablas.

La primera tabla que se creó fue la que guarda la autenticación e identificación de administrador, la cual es mostrada en la figura 3.7 llamada *admin*, dicha tabla está conformada por tres columnas; la primera llamada *login* la cual guarda hasta 30 caracteres, ya sea número, letra y signo tipo `varchar(30)` y servirá para guardar el nombre del usuario administrador, la segunda columna llamada *password* tipo `varchar(30)`, además de contar con 30 caracteres disponibles, se definió de tipo `password` para que se oculte la contraseña con puntos negros cada vez que el administrador la escriba.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acc
<input type="checkbox"/>	1 login	varchar(30)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	2 password	varchar(30)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/>	3 sesion	int(4)			No	Ninguna		

Marcar todos    Para los elementos que están marcados: Examinar Cambiar

Vista de impresión    Planteamiento de la estructura de tabla    Mover columnas

Agregar  columna(s)     Al final de la tabla     Al comienzo de la tabla     Después de

+ Índices

Figura 3.7: Definición de la tabla *admin*

Existe una columna importante llamada *sesion* de tipo `int(4)`, los campos de esta columna permanecen en un valor predeterminado 0, el valor del campo cambia a 1 cuando el administrador inicia sesión, ninguna persona ajena al sistema podrá entrar, ni podrá saltar la página de acceso si es que llegara a conocer alguna ruta de directorios; para lograrlo se utilizarán dos funciones básicas, la primera (y posiblemente la más importante) es la función «*function conexiones*» la cual es llamada en todas las páginas para revisar que el administrador que desea ingresar haya iniciado una sesión, de otra forma el sistema le enviará de manera automática al index (página de inicio del sis-

tema), donde nuevamente debe de ingresar usuario y contraseña, la segunda función va muy ligada a la primera y es «*function verificar usuario*», dicha función identifica los valores ingresados por el administrador.

La segunda tabla que contiene la base de datos SAMLAB es *alumnos*, los valores se muestran en la figura 3.8, dicha tabla contiene cuatro columnas, en la primera columna *matricula*, se guarda la información escaneada con el lector de códigos de barras tomada directamente de la credencial del estudiante, para respetar el formato de la matrícula se ha utilizado código java script, el cual verifica cada carácter ingresado para que cumpla con el formato de símbolos definido por XX-00X-XXX (por ejemplo: 05-003-2306). La segunda columna *nombre* guarda el nombre del estudiante con cualquier carácter y con una longitud de hasta 30 símbolos, es preciso decir que los nombres en este campo si pueden repetirse incluso idénticamente. La tercera columna es *apellido* tipo varchar(30), la última columna corresponde al *estado* tipo varchar(40), ésta es una desición que toma el administrador a la hora de dar de alta a un usuario en el sistema, estado es igual a: estudiante activo o estudiante inactivo, esto ayudará al administrador a garantizar que el estudiante que solicita un préstamo se encuentre como estudiante vigente en la universidad.<sup>1</sup>

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 <b>matricula</b>	varchar(30)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	2 <b>nombre</b>	varchar(30)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	3 <b>apellido</b>	varchar(30)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	4 <b>estado</b>	varchar(40)	latin1_swedish_ci		Sí	NULL

Figura 3.8: Definición de tabla alumnos, la cual cuenta con 4 campos de tipo varchar

La siguiente tabla corresponde al equipo y material (ver figura 3.9) las columnas que la forman son: *codigo* tipo varchar(20) contiene un script para verificar que el formato adquirido corresponda a un código de equipo asignado, la siguiente columna llamada *tipo* de valor varchar(20), guarda en cada campo el nombre de equipo seleccionado, se utilizó una selección multiple a

<sup>1</sup>El administrador dará de alta en el sistema SAMLAB al estudiante que cumpla 2 reglas: 1.- Que el estudiante cuente con credencial vigente y re-sellada en el año del registro, 2.- Que sea estudiante de alguna ingeniería impartida en la UACM-SLT

través de pestañas dinámicas que el usuario podrá seleccionar; la siguiente columna corresponde a *marca* tipo `varchar(20)` para seleccionar la marca de un menú desplegable, *modelo* tipo `varchar(20)` el administrador escribirá el modelo del equipo en el campo *estado* en este apartado el administrador elige el estado del equipo en el momento que lo registre en el sistema y cuando se desee prestar dicho equipo de antemano el administrador y el usuario sabrán si se encontrará óptimo o averiado, y un último campo llamado *descripción* es un espacio en blanco para que el administrador escriba algún comentario que considere importante mencionar.



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predetermin
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	Ninguna
<input type="checkbox"/> 2	<b>codigo</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 3	<b>tipo</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 4	<b>marca</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 5	<b>modelo</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 6	<b>estado</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 7	<b>descripcion</b>	varchar(50)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/> 8	<b>dateofbirth</b>	date			Sí	NULL

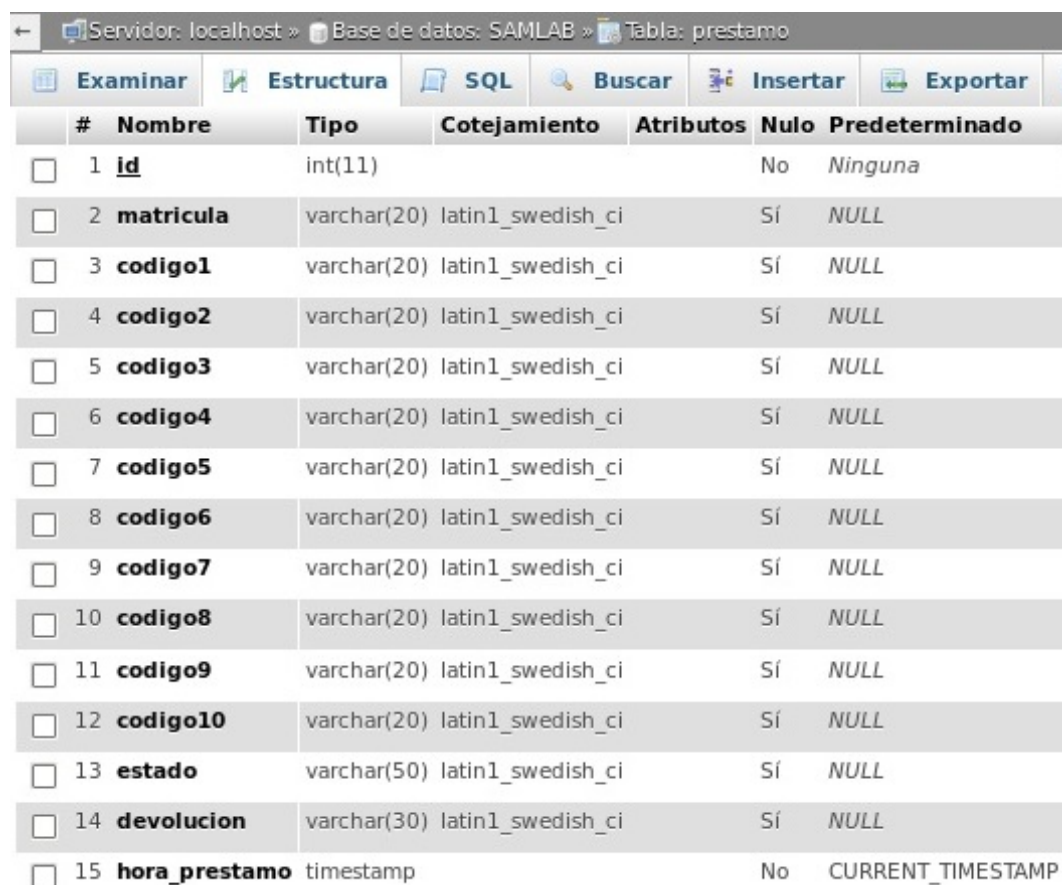
↑  Marcar todos Para los elementos que están marcados:  Examinar

Figura 3.9: Definición de la tabla material y los campos que la componen

Hasta este momento sólo se tiene registro por separado de los estudiantes y el equipo, las siguientes tablas deberán registrar la acción de préstamo y devolución en cada tabla de la base de datos, a continuación la descripción de dichas tablas.

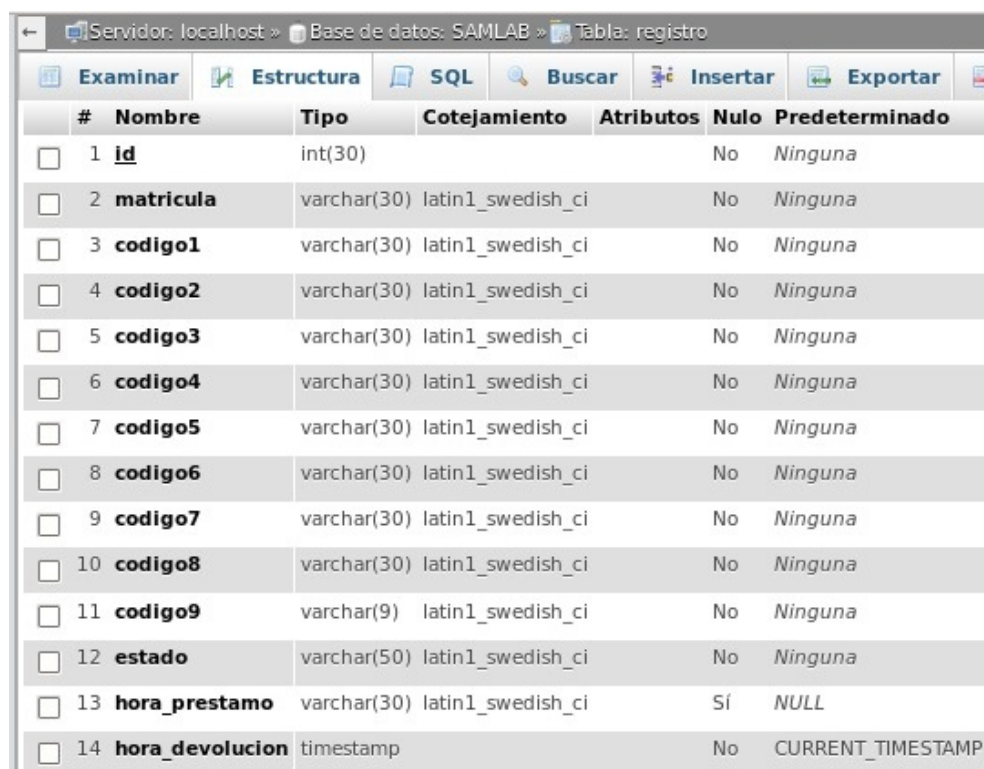
Se tiene ahora una nueva tabla llamada préstamo (ver figura 3.10), esta tabla es muy importante ya que es la parte central de este trabajo, dicha tabla interactúa con la tabla alumnos, material y registro; la primera columna es *matricula* tipo `varchar(20)`, las siguientes 10 columnas conforman los campos para almacenar los códigos del equipo seleccionado tipo `varchar(20)`, con nombre *codigo1, codigo2...codigo10*, la siguiente columna es *estado* tipo `varchar(20)` este campo es un identificador para el sistema de que ha sido guardado un préstamo, el campo cambia a (activo) si se ha guardado un préstamo y cambia a estado pasivo cuando haya sido devuelto el equipo de

manera correcta, la última columna, *hora prestamo* tipo timestamp, registra la fecha y hora del préstamo en formato YYYY-MM-DD HH:MM.SS tomada directamente del servidor, utilizando una función reservada de mysql. Al realizar un préstamo se tiene: la hora de inicio, hora de devolución así como toda la información relacionada al estudiante y al equipo, dicha acción se debe registrar en algún lugar para poder utilizar la información posteriormente, esta información se va a almacenar en la siguiente tabla *registro* (ver figura 3.11) la estructura de esta última tabla será idéntica a la tabla *prestamo*, una columna más llamada *hora devolucion* tipo timestamp toma la hora directamente del servidor y llena el campo correspondiente dejando un registro de la acción realizada.



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 <b>id</b>	int(11)			No	Ninguna
<input type="checkbox"/>	2 <b>matricula</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	3 <b>codigo1</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	4 <b>codigo2</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	5 <b>codigo3</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	6 <b>codigo4</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	7 <b>codigo5</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	8 <b>codigo6</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	9 <b>codigo7</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	10 <b>codigo8</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	11 <b>codigo9</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	12 <b>codigo10</b>	varchar(20)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	13 <b>estado</b>	varchar(50)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	14 <b>devolucion</b>	varchar(30)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	15 <b>hora_prestamo</b>	timestamp			No	CURRENT_TIMESTAMP

Figura 3.10: Tabla préstamo



The screenshot shows a database management interface with the following table structure:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 <b>id</b>	int(30)			No	Ninguna
<input type="checkbox"/>	2 <b>matricula</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	3 <b>codigo1</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	4 <b>codigo2</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	5 <b>codigo3</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	6 <b>codigo4</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	7 <b>codigo5</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	8 <b>codigo6</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	9 <b>codigo7</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	10 <b>codigo8</b>	varchar(30)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	11 <b>codigo9</b>	varchar(9)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	12 <b>estado</b>	varchar(50)	latin1_swedish_ci		No	Ninguna
<input type="checkbox"/>	13 <b>hora_prestamo</b>	varchar(30)	latin1_swedish_ci		Sí	NULL
<input type="checkbox"/>	14 <b>hora_devolucion</b>	timestamp			No	CURRENT_TIMESTAMP

Figura 3.11: Tabla registro

Durante todo el desarrollo de sistema de software es de suma importancia que los desarrolladores sigan una serie de especificaciones, que les permita tener la disciplina que haga que todas las etapas del diseño sean, no solo coherentes sino formales para conseguir el objetivo planteado en cada iteración. Los cimientos del sistema SAMLAB han sido creados hasta este momento, es hora de comenzar la aplicación conjuntamente. La creación de una topología física y lógica de red como la que se mencionó en este capítulo es la base que soporta el desarrollo del sistema SAMLAB; en el siguiente capítulo se plantean los procesos de desarrollo y diseño que se llevaron a cabo para crear la aplicación.

## Capítulo 4

# Diseño e implementación de Software SAMLAB

*La brecha entre la teoría y la práctica no es tan grande en teoría como lo es en la práctica*

Anónimo

Un sistema de software es creado para servir a un grupo de usuarios con un objetivo en común, el usuario puede ser una persona, un grupo de personas, otro sistema de software y hasta una máquina. Para asegurar que cada parte del proyecto cumpla el objetivo para el cual fue hecho, es recomendable adoptar un proceso de modelado de software para que la asignación de tareas y responsabilidades sea óptima para el desarrollo del sistema, a continuación se describe cómo el proceso unificado de Rational RUP junto con el lenguaje unificado de modelado UML aportaron organización de roles para la creación del sistema SAMLAB UACM-SLT v1.0. Además se describe el desarrollo de cada caso de uso que se utilizó para el desarrollo del sistema.

### 4.1. Caso identificar y autenticar administrador

**Alcance:** El sistema pedirá identificación y autenticación del administrador a través de un usuario y contraseña asignados, para esto existe una cuenta única de acceso por usuario (superusuario y usuario administrador). Los requerimientos del *caso de uso identificar y autenticar administrador* se encuentran en la tabla 4.1.

Identificar y autenticar administrador			
<b>Objetivos asociados</b>	Gestionar acceso de administradores al sistema SAMLAB		
<b>Requisitos asociados</b>	Contar con claves de usuario y contraseña		
<b>Descripción</b>	El sistema será capaz de administrar el acceso de administradores al sistema SAMLAB		
<b>Precondición</b>	El estudiante debe solicitar un préstamo o una devolución		
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>	
	Normal	1	El usuario administrador del laboratorio, accede a la dirección IP asignada para ingresar a la red SAMLAB
	2	El usuario administrador del laboratorio ingresa el usuario y la contraseña establecidos y da click en ingresar.	
	3	El sistema muestra al administrador del laboratorio la página de bienvenida y le pide continuar o salir del sistema.	
<b>Postcondición</b>	El usuario administrador ha iniciado una sesión dentro del sistema SAMLAB		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y la contraseña no coinciden con el registro realizado, el sistema mostrará la misma ventana para que intente de nuevo.	

Tabla 4.1: Tabla de requerimientos para el caso identificar y autenticar administrador, donde se describen los puntos que se deben llevar a cabo para la correcta identificación

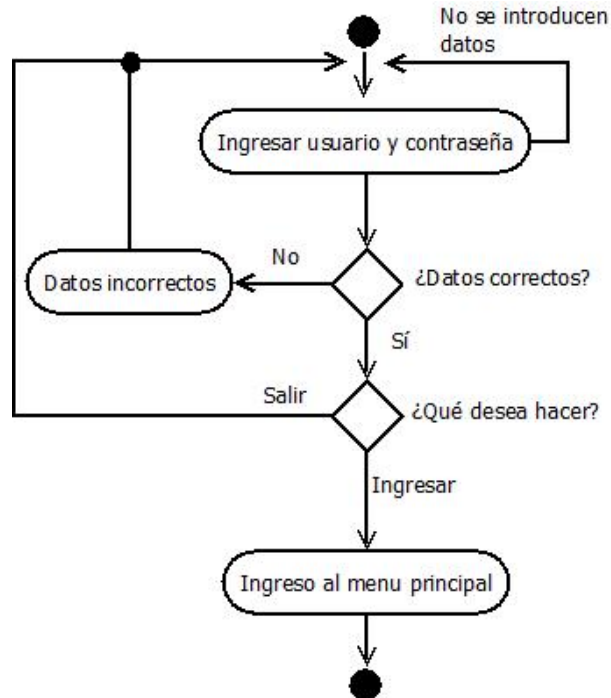


Figura 4.1: Diagrama de actividad de acceso del administrador

**Diagrama de actividades de caso identificar y autentificar administrador:** El sistema SAMLAB UACM-SLT v1.0 puede atender a la demanda estudiantil de las carreras de ingeniería, es decir. Para acceder a él, basta con conocer la dirección IPv4 asignada, el firewall direcciona la petición y muestra la página de inicio del sistema (index.php) para que el administrador se autentifique, al hacerlo correctamente el sistema reconoce al administrador mostrando su nombre como bienvenida, si el administrador continua navegando llegará al menu principal del sistema, el diagrama de actividad mostrado en la figura 4.1 detalla los pasos y condiciones que se deben cumplir y realizar para una autentificación de usuario-administrador correcta, es decir el caso ideal.

**Diagrama de secuencia del caso identificar y autentificar administrador:**

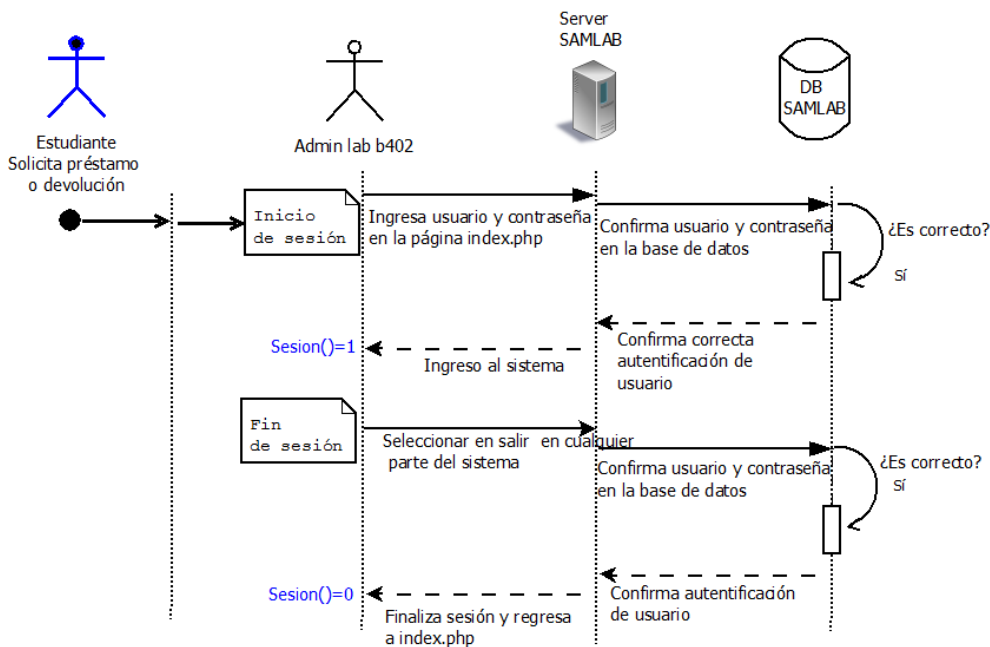


Figura 4.2: Diagrama de secuencia del caso identificar y autentificar administrador

El sistema registra el acceso de un administrador en el sistema a través de un inicio de sesión, cuando las claves de usuario y contraseña sean autenticadas correctamente en el servidor, dicha acción quedará registrada en la tabla admin en la columna *sesión*, el valor del campo estará inicialmente en cero y pasará a uno cuando un usuario inicie una sesión y volverá a cero cuando la cierre. La figura 4.1 muestra el proceso de inicio de sesión y el intercambio de mensajes entre la base de datos, el servidor y el administra-

dor. Al final se observa que el usuario puede salir desde cualquier parte del sistema.

El resultado del *caso identificar y autenticar administrador* es el acceso al sistema y lo primero que ve el administrador es una sencilla página de bienvenida, la cual contendrá dos opciones: entrar al sistema o salir, al continuar el administrador verá el menú principal del sistema (ver figura 4.3), el cual es muy importante ya que es donde el administrador tendrá una vista general de todo el sistema SAMLAB UACM-SLT v1.0 y así decidir lo que va hacer, a continuación se describe el uso de cada área.

- **Registros:** Es el área donde se organiza toda la información respecto a estudiantes y equipo (registros, eliminación, consultas y cambios).
- **Operación:** En esta área el administrador realiza operaciones como préstamos, devoluciones y cambios.
- **Consultas:** Las consultas que puede realizar el administrador se pueden revisar en esta sección, en el primer apartado se revisan los préstamos activos en cualquier momento, en el segundo apartado se consulta el historial de préstamos realizados a través del tiempo. Este apartado podría ser desarrollado en el futuro para realizar estadísticas de cualquier tipo, por ejemplo: qué equipo es el más usado por los estudiantes, o horarios de mayor demanda del uso de equipo, entre otros.



Figura 4.3: Menú principal de acceso al sistema SAMLAB

Para cada sección del menú principal existe un menú particular o submenú para entrar a las opciones disponibles, en el caso de estudiantes se tiene el submenú mostrado en la figura 4.4 y por cada opción mostrada se desarrolló un caso de uso para comenzar con la programación del mismo.



Figura 4.4: Submenú de la sección estudiantes

A continuación se describe el caso estudiantes con todas las acciones de manejo de información, se muestran las tablas de requerimientos, los diagramas de casos de uso, diagramas de secuencia, diagramas de actividad y plantillas de casos de uso.

## 4.2. Caso estudiantes

La gestión de estudiantes en el sistema SAMLAB UACM-SLT v1.0 se realiza con cuatro casos muy importantes: *registrar estudiante*, *eliminar estudiante*, *consultar estudiantes* y *editar registro de un estudiante*. Para comenzar el desarrollo de cada sección, se deben definir: el caso de uso para realizar la gestión anterior, los diagramas de secuencia y diagramas de actividad, a continuación la descripción de cada parte.

**Diagrama de casos de uso registro de estudiantes:** En el *caso de uso registrar estudiante* interviene el caso de uso *identificación y autenticación de administrador* y la extensión del caso *matrículas duplicadas*, todos en conjunto forman el registro de un estudiante con la certeza de que no se tendrán registros de matrículas duplicadas, el diagrama de casos de uso se muestra en la figura 4.5.

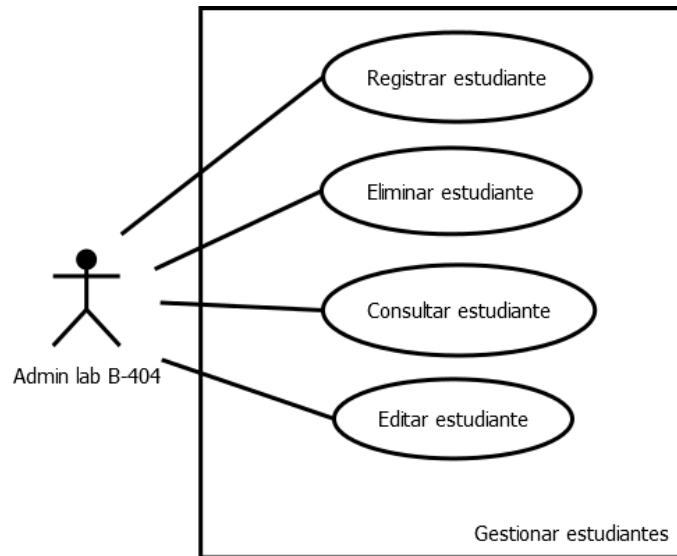


Figura 4.5: Diagrama de casos de uso registro de estudiantes

**Descripción de requerimientos:** La tabla 4.2 describe los requerimientos del *caso registrar estudiante*, dichos requerimientos sirvieron como referencia para programar el caso, se muestra un caso base (secuencia normal) en la siguiente plantilla de requerimientos.

#### 4.2.0.1. Diagrama de secuencia registrar estudiante:

En la figura 4.6 se observa el diagrama de secuencia incluyendo la extensión 3 (E3), en este diagrama se observa la iteración que tiene la base de datos con el servidor SAMLAB y los mensajes que enviará a la pantalla del administrador en cualquiera de los dos casos, para el primer caso se considera el caso base, donde el estudiante se da de alta por primera vez en la base de datos del sistema, para el segundo caso se muestra la extensión del caso (E3) donde ya existe la matrícula que se desea ingresar, y como respuesta el sistema manda un mensaje en la pantalla del administrador.

Registrar estudiante			
Objetivos asociados	Gestionar estudiantes		
Requisitos asociados	Presentar credencial de estudiante activo		
Descripción	El sistema será capaz de registrar un nuevo estudiante en la base de datos SAMLAB		
Precondición	El estudiante no está dado de alta en el sistema y cuenta con credencial de estudiante activo.		
Secuencia	<b>Paso</b>	<b>Acción</b>	
	Normal	1	El usuario administrador del laboratorio, solicita acceso al sistema e introduce el usuario y contraseña asignado.
	2	El usuario administrador del laboratorio solicita al sistema comenzar el proceso de registro de nuevo estudiante	
	3	El sistema solicita los siguientes datos del nuevo estudiante: nombre, apellidos, estado y matrícula	
	4	El usuario administrador comprueba que los datos ingresados sean correctos	
	5	El usuario administrador entrega la credencial al estudiante	
Postcondición	El estudiante esta dado de alta en el sistema y puede realizar préstamos cuando lo desee		
Excepciones	<b>Paso</b>	<b>Acción</b>	
	1	Si el usuario y la contraseña no coinciden con el registro realizado, el sistema mostrará la misma ventana para que intente de nuevo.	
	3	Si el sistema detecta que el nuevo estudiante ya registrado en la base de datos no realizará el registro y le pedirá que intente de nuevo.	
	4	Si el usuario administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina	

Tabla 4.2: Plantilla de requerimientos para caso de uso registrar estudiante

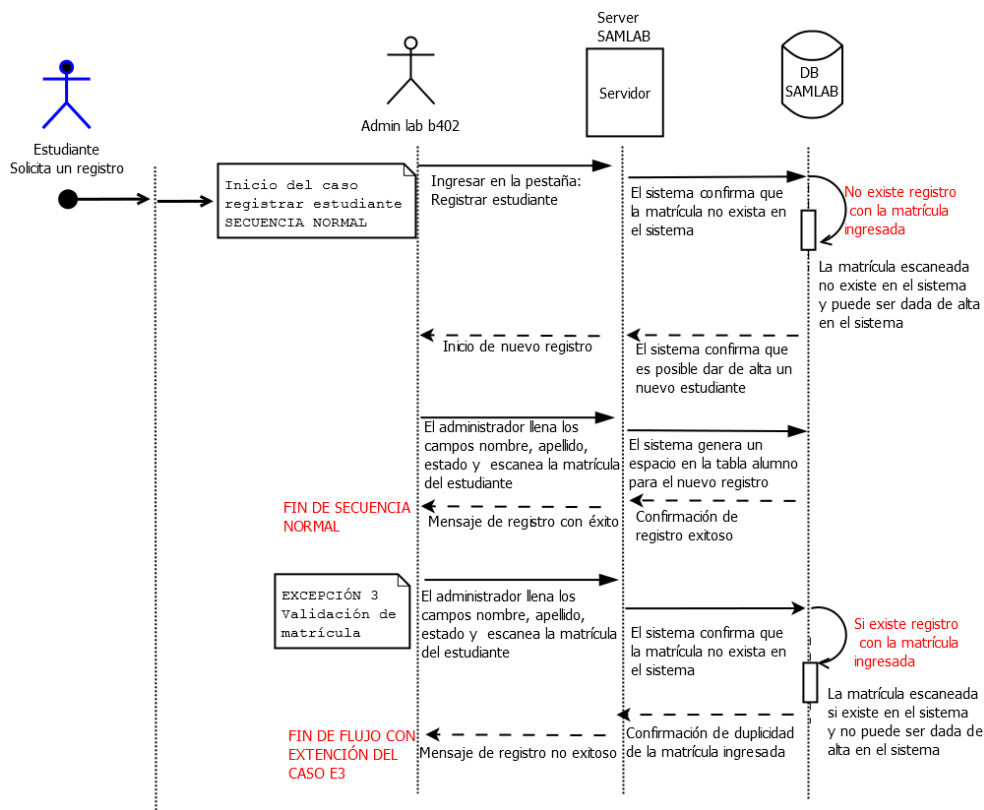


Figura 4.6: Diagrama de secuencia de caso registrar estudiante

#### 4.2.0.2. Diagrama de actividad del caso registrar estudiante:

El diagrama de actividad mostrado en la figura 4.7 muestra la serie de pasos y condiciones que verá el administrador en el proceso de alta de un estudiante, la extensión 3 está representada como la decisión que realiza el sistema debido a la consulta de la matrícula ingresada.

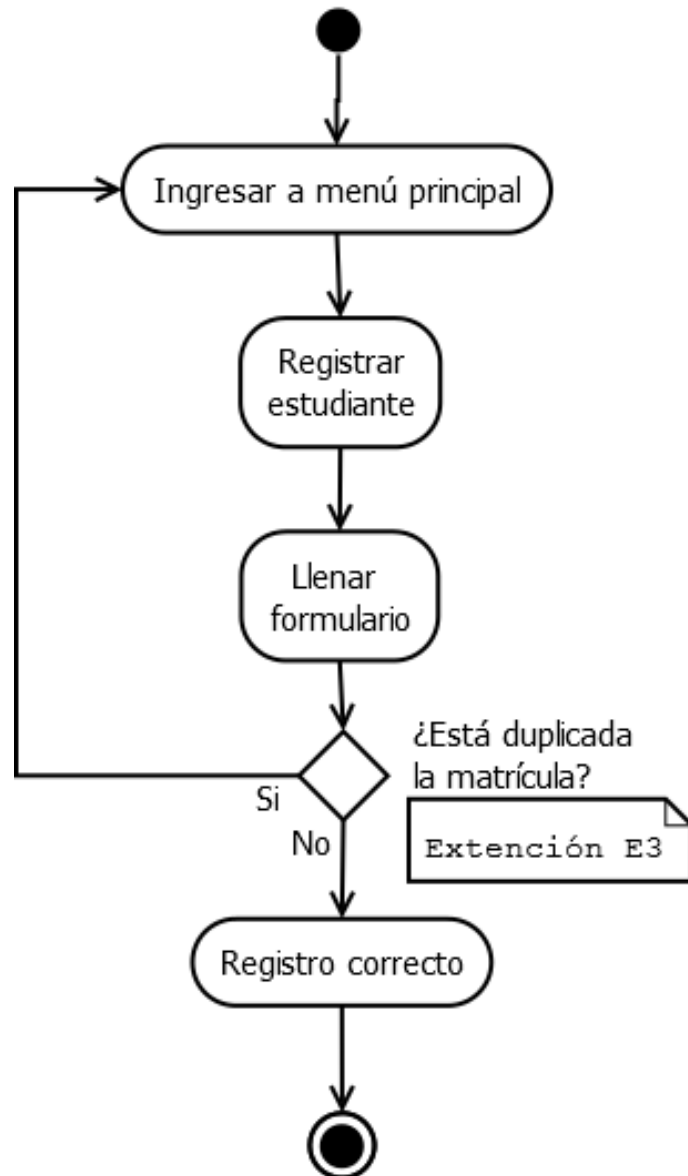


Figura 4.7: Diagrama de actividad del caso registrar estudiante

Así como se da de alta un estudiante en la base de datos en el siste-

ma SAMLAB UACM-SLT v1.0, es posible dar de baja algún estudiante del sistema a continuación la descripción del caso.

#### 4.2.0.3. Caso eliminar estudiante

El caso eliminar estudiante existe por varias razones, una de ellas corresponde a una petición especial del administrador para poder decidir la eliminación de un registro a través de la verificación visual de la etiqueta *estado* en el registro de un estudiante, ya que esta etiqueta identifica como activo o inactivo a cada estudiante; la siguiente razón corresponde a la posibilidad que tiene el administrador para eliminar cada registro individual o si lo desea toda la información de la tabla estudiantes.

Eliminar estudiante		
Objetivos asociados	Gestionar estudiantes	
Requisitos asociados	Que el estudiante esté registrado en el sistema y presentar la credencial de estudiante activo	
Descripción	El sistema será capaz de eliminar un estudiante de la base de datos SAMLAB	
Precondición	El estudiante está dado de alta en el sistema y cuenta con credencial de estudiante activo.	
Secuencia	<b>Paso</b>	<b>Acción</b>
	1	El usuario administrador del laboratorio selecciona eliminar estudiante del menú principal
	2	El sistema solicita la matrícula del estudiante, y el administrador la escanea con el lector de códigos
	3	El sistema muestra los datos relacionados con la matrícula ingresada y revisa la información obtenida, a continuación se selecciona en eliminar
	4	El sistema muestra un mensaje de baja exitosa y el usuario administrador entrega la credencial al estudiante
Postcondición	El estudiante esta dado de baja en el sistema y no puede realizar operaciones de préstamo ni devolución	
Excepciones	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema detecta que no existe la matrícula escaneada mandará un mensaje de no existencia de matrícula
	4	Si el usuario administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Tabla 4.3: Plantilla de requerimientos para el caso de uso eliminar estudiante

#### 4.2.0.4. Descripción de requerimientos:

Los requerimientos para dar de baja un estudiante de la base de datos SAMLAB, están descritos en la tabla 4.3, se describe el caso normal y una extensión importante del caso el cual valida la matrícula ingresada y se encarga de verificar que la matrícula que se desea dar de baja exista en el sistema previamente.

#### 4.2.0.5. Diagrama de secuencia del caso eliminar estudiante:

El diagrama de secuencia mostrado en la figura 4.8 muestra el intercambio de mensajes entre el servidor, la base de datos y el administrador. El inicio del flujo normal está del lado superior izquierdo donde se considera que el registro que se desea dar de baja existe en la base de datos. En la misma figura 4.8 se muestra al final la extensión del caso (E4) donde se considera que la matrícula que se desea dar de baja no existe en la base de datos y el mensaje que recibirá el administrador será *intenta de nuevo*. El diagrama de secuencia eliminar estudiante considera que el administrador se encuentra en el menú correspondiente a usuarios.

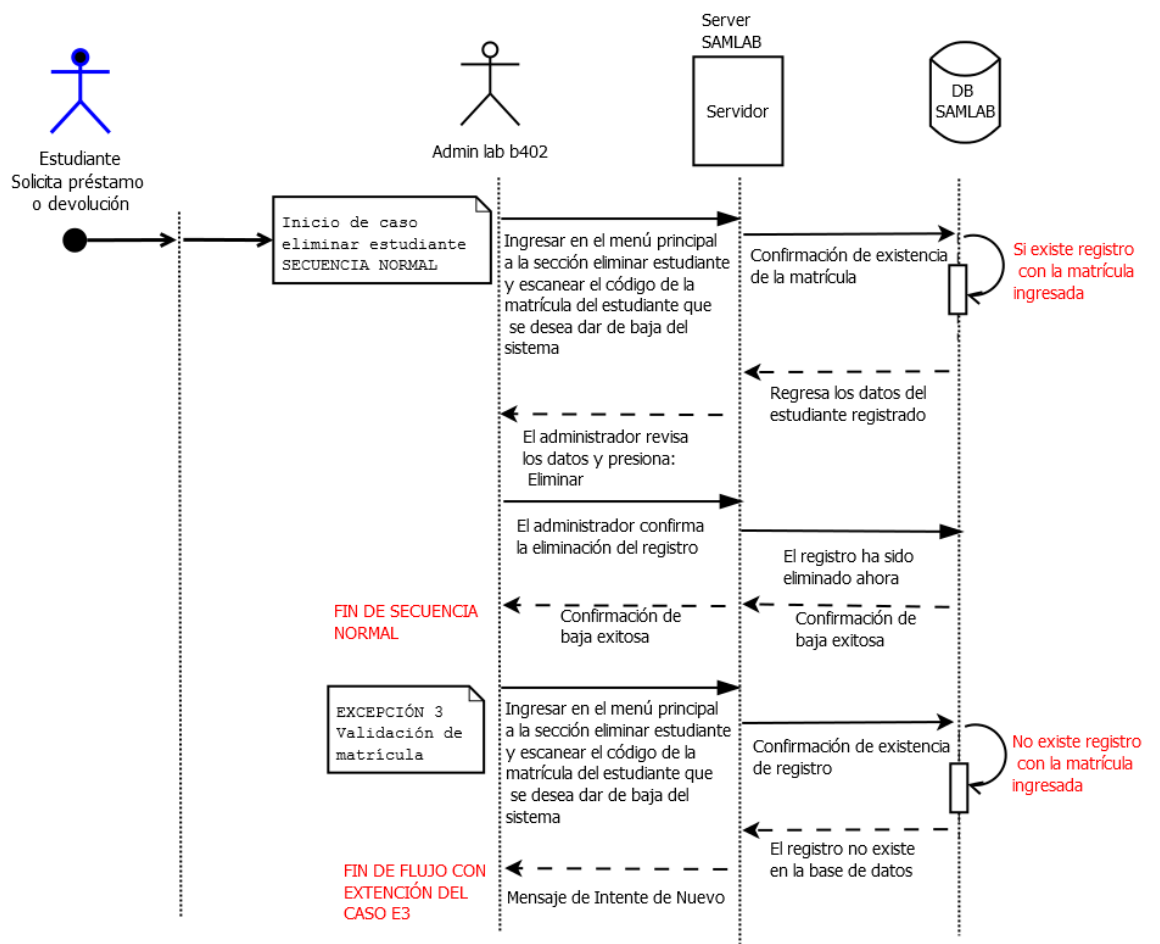


Figura 4.8: Diagrama de secuencia caso de uso eliminar estudiante

#### 4.2.0.6. Diagrama de actividad del caso de uso eliminar estudiante:

El diagrama de actividad referente a caso eliminar estudiante se muestra en la figura 4.9, donde se puede observar la continuidad de la baja de un estudiante, de igual forma se puede cancelar el proceso de eliminación en cualquier momento.

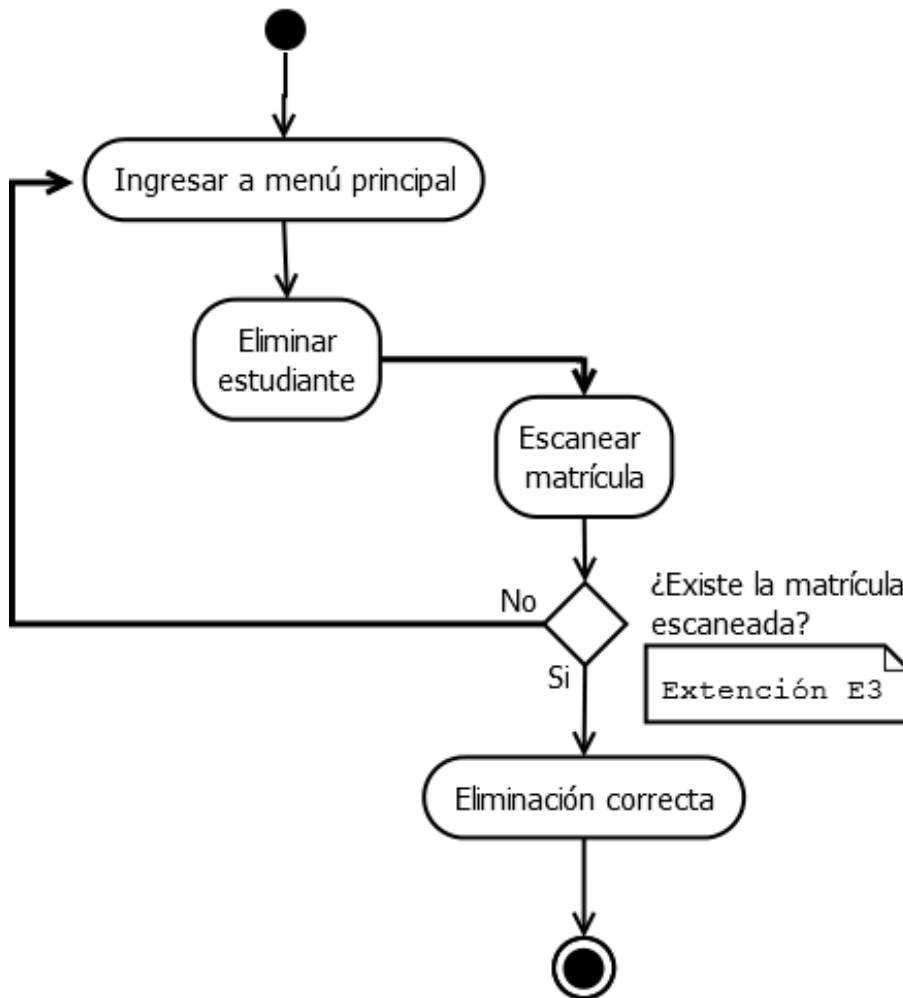


Figura 4.9: Diagrama de actividad del caso eliminar estudiante

#### 4.2.0.7. Caso editar estudiante:

Este caso es una extensión del caso *consultar estudiantes*, ya que además de consultar la información de un estudiante en particular se podrá cambiar cualquier campo registrado, excepto la matrícula. El propósito de este caso es

la posibilidad de cambiar la pestaña estado entre pasivo y activo, esta leyenda sirve para que el administrador clasifique a los estudiantes como alumnos inscritos a las materias en relación al semestre correspondiente. Esta leyenda es mostrada al administrador en el momento de que un estudiante desee realizar un préstamo. Los requerimientos del caso se muestran en la tabla 4.4.

#### 4.2.0.8. Diagrama de requerimientos:

Editar estudiante		
<b>Objetivos asociados</b>	Editar registro de estudiante	
<b>Requisitos asociados</b>	Que el estudiante esté registrado en el sistema y presentar la credencial de estudiante activo	
<b>Descripción</b>	El sistema será capaz de editar campos en el registro de un estudiante de la base de datos SAMLAB	
<b>Precondición</b>	El estudiante está dado de alta en el sistema y cuenta con credencial de estudiante activo.	
<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario administrador del laboratorio solicita al sistema comenzar el proceso editar registro en particular del menú principal
	2	El sistema solicita la matrícula del estudiante y el administrador la escanea con el lector de códigos
	3	El sistema muestra los datos relacionados con la matrícula ingresada, se muestran los campos a editar, el administrador realiza los cambios necesarios y selecciona en guardar
	4	El sistema muestra un mensaje de edición exitosa y el usuario administrador entrega la credencial al estudiante
<b>Postcondición</b>	El registro ha sido editado puede realizar operaciones de préstamo y devolución	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema detecta que no existe la matrícula escaneada mandará un mensaje de no existencia de matrícula
	4	Si el usuario administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Tabla 4.4: Tabla de requerimientos para el caso editar estudiante

#### 4.2.0.9. Diagrama de secuencia editar estudiante:

En la figura 4.10 se presenta el diagrama de secuencia que muestra la conexión con la base de datos después de haber verificado la matrícula, ahora se puede cambiar cualquier dato relacionado a la matrícula, como resultado se guardan los cambios y se imprime en pantalla que los cambios se realizaron con éxito, el diagrama de secuencia se muestra en la figura 4.10.

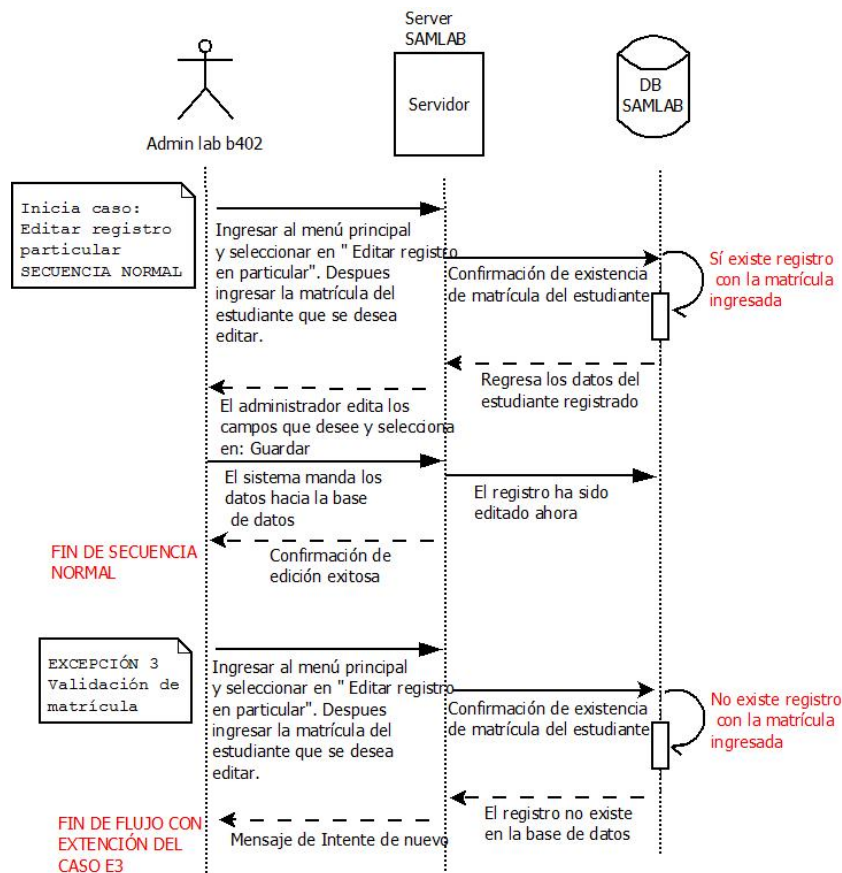


Figura 4.10: Diagrama de secuencia del caso editar estudiante

#### 4.2.0.10. Caso consultar estudiantes

El sistema tiene la posibilidad de hacer una consulta en la base de datos SAMLAB y traer el contenido de la tabla alumnos, el caso consultar estudiantes es el encargado de mostrar al administrador la relación de estudiantes que podrían realizar un préstamo, a continuación la descripción del caso.

#### 4.2.0.11. Descripción de requerimientos:

Los requerimientos del *caso consultar estudiantes* consiste en realizar una consulta general a la base de datos SAMLAB en la tabla *alumnos*, el alcance del caso es que el administrador pueda revisar la relación de los estudiantes de forma gráfica; sin embargo considerando la información que maneja podría ser uno de los casos que más se puede aprovechar en un futuro para

obtener estadísticas de todo tipo. El desarrollo en particular para este caso, se centró en mostrar la tabla alumnos en la pantalla del administrador.

#### 4.2.0.12. Diagrama de secuencia consultar estudiantes:

La iteración entre el servidor y la base de datos en este caso es muy sencilla, la consulta se realiza en tres sencillos pasos, se comienza seleccionando la pestaña consulta de estudiantes, el sistema se comunica con la base de datos y se imprime en la pantalla una lista con información de los estudiantes, en el diagrama de la figura 4.11 se observa el caso consulta.

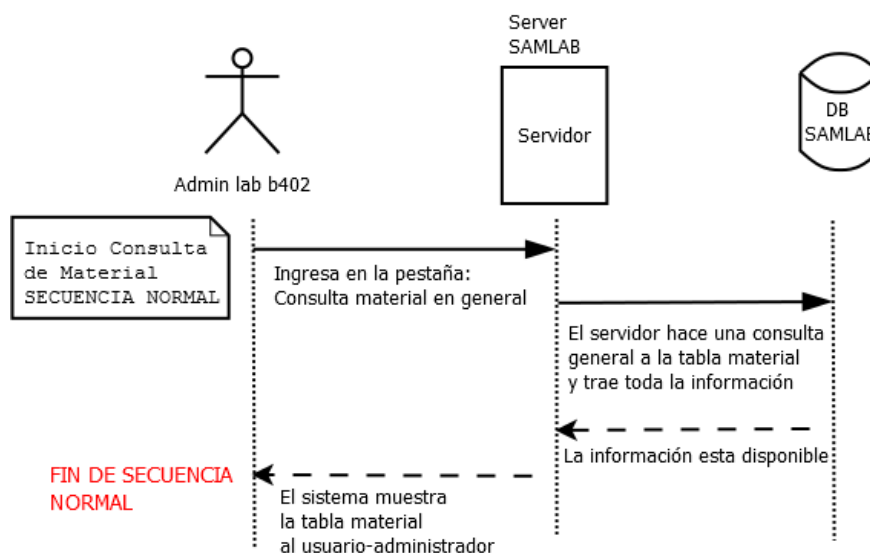


Figura 4.11: Diagrama de secuencia del caso consultar estudiantes

#### 4.2.0.13. Diagrama de actividad consultar estudiantes:

El diagrama de actividades del caso *consultar estudiantes* se muestra en la figura 4.12. En este momento se tiene listo el módulo completo de estudiantes en su forma operacional básica, ahora se tiene registro de todo estudiante que desee realizar un préstamo. Con este nuevo caso *consultar estudiantes* es posible realizar una consulta general donde muestre en la pantalla toda la información de los estudiantes que se encuentran dados de alta en el sistema, con este último caso ya tenemos concluida una iteración completa de RUP, por lo que se puede comenzar con una nueva iteración, buscando mejoras en el sistema SAMLAB v 1.0.



Figura 4.12: Diagrama de actividad del caso de uso consultar estudiantes

En este momento se tiene completo el caso estudiantes, a continuación se describirá el caso *registro de equipo*, el cual debe ser un conjunto de casos que se enfoquen en administrar todo el equipo disponible en el laboratorio de electrónica básica. Una característica importante del proceso de diseño utilizado RUP es la reutilización de casos ya desarrollados, en este caso para el diseño del *caso registro de equipo* se utilizó el diseño del *caso registro de estudiantes*.

### 4.3. Caso equipo

El equipo disponible para hacer préstamos en el laboratorio está conformado por diferentes tipos de equipos y material, sin embargo las pruebas que se realizaron en el sistema fueron con equipo del laboratorio, dejando al material solo para anexar al sistema en un futuro, por lo que en este documento solo nos enfocaremos en hablar del equipo del laboratorio. Al momento de realizar este trabajo el total de equipos fue de 107, para organizar el equipo se ha clasificado en tres tipos de equipos: equipo de electrónica básica, equipo de electrónica experimental, y equipo de electrónica didáctico, estas clasificaciones pueden ir evolucionando conforme lo haga el equipo en el laboratorio B-404, a continuación la descripción de los tipos de equipo y una recopilación de la tabla de equipos disponible.

- Equipo de electrónica básica: Es utilizado principalmente por estudiantes de los primeros semestres del ciclo superior principalmente,

por ejemplo: multimetros, fuentes, puente de impedancias y generador de señales entre otros.

- Equipo de electrónica experimental: Es utilizado por estudiantes con asignaturas de experimentación y aplicación por ejemplo: osciloscopios, analizadores de espectro, programador universal de memorias entre otros.
- Equipo de electrónica didáctico: El equipo didáctico es utilizado en todos los semestres y sirve para simulación de circuitos electrónicos en este caso son módulos de: modulación y decodificación, amplificadores, osciloscopios, circuitos, filtros entre otros.

La relación del equipo disponible en el laboratorio a la fecha de la realización de este trabajo se encuentra en la tabla 4.5, en total se cuenta con 107 equipos en existencia, de los cuales 8 se encuentran fuera de operación, en la tabla 4.5 se describe el equipo en términos del modelo, marca, número de piezas y el código que se generó para cada equipo electrónico, el cual es único por equipo. Los dos caracteres al final *XX* identifican el número de equipo, este número puede ser cualquiera y forma parte de la etiqueta.

Tipo	Modelo	Marca	Total	ISEI	ISET	FÍSICA	Dañado	Comentarios	Código
Analizadores de Espectro	2630	BKPRECISION	3	0	0	0	1		15003BK-2630-XX
Borradores de Memoria UV		CHIPHERASER	4	0	0	0	0		
Fuente	3030TD	ESCORT	6	0	0	0	1		15003ES-3030-XX
Fuente	1672	BKPRECISION	11	2	1	0	0		15003BK-1672-XX
Generador de Señales	LXI-33220	AGILENT	7	0	1	0	0	20Mhz	15003AG-3220-XX
Generador de Señales	4011A	BKPRECISION	6	0	0	0	2	5Mhz	15003BK-4011-XX
Generador de Señales	3235A	ESCORT	7	1	0	0	2		15003ES-3235-XX
Multimetro	38XR	METERMAN	16	1	0	1	1		15003ME-38XR-XX
Multimetro	UI242A	AGILENT	5	0	0	0	0	Desaparecido 1	15003AG-UI24-XX
Osciloscopio	TDS-1001	TEXTRONIX	4	0	0	0	0	40Mhz	15003TX-1001-XX
Osciloscopio	TDS-2022B	TEXTRONIX	6	0	0	0	1	200Mhz	15003TX-2022-XX
Osciloscopio	DS05032A	AGILENT	10	0	0	0	0	No existen puntas de ningún osciloscopio	15003AG-5032A-XX
Tarjeta de Adquisición de datos PCI	6221	NATIONAL INSTRUMENTS	2	1	0	0	0		15003NI-6221-XX
Programador Universal de Memorias	ADPV10	XELTEX	2	0	0	0	0	Falla cables de puertos paralelos	15003XE-ADPV-XX
Proyector Digital	MP512	BENQ	2	0	0	0	0		15003BQ-MP512-XX
Puente de Impedancias	LCR55	METERMAN	2	0	0	0	0		15003ME-LCR55-XX
Punta Lógica	DP-21	TEXAS INSTRUMENTS	5	0	0	0	0		15003TI-DP-21-XX
Tarjeta de Adquisición de datos USB	6009	NATIONAL INSTRUMENTS	2	1	0	0	0		15003NI-6009-XX
Tarjeta de Adquisición de datos serial	SCC-68	NATIONAL INSTRUMENTS	1	0	0	0	0		15003NI-SCC-68-XX
Módulo de Modulación y Decodificación	53-230	FEEDBACK	2	0	0	0	0		15003FE-53-230-XX
Módulo de Circuitos y Filtros	53-220	FEEDBACK	2	0	0	0	0		15003FE-53-220-XX
Módulo de Amplificadores y Osciloscopios	53-210	FEEDBACK	2	0	0	0	0		15003FE-53-210-XX
		Total	107	6	2	1	8		

Tabla 4.5: Tabla que muestra el equipo existente en el laboratorio B-402

Para el caso *registro de equipo* el sistema utiliza un sub-menú equipo, el caso es muy similar a los estudiantes, la figura 4.13 es una impresión de

la pantalla del sub-menú de equipo, donde cada apartado corresponde a un caso de uso.



Figura 4.13: Impresión de pantalla del submenú equipo

Para organizar el diseño y desarrollo del caso se tiene el diagrama de *casos de uso registro de equipo* el cual representa la incidencia del administrador del laboratorio para acceder a los casos de registro de equipo, la figura 4.14 muestra el diagrama.

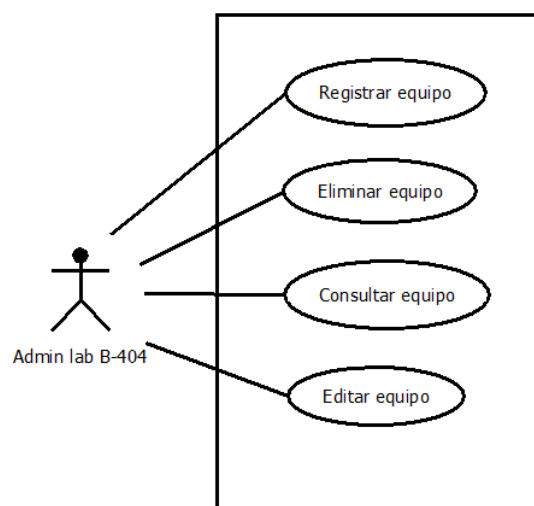


Figura 4.14: Diagrama de casos de uso registro de equipo

#### 4.3.0.14. Caso registrar equipo

El caso *registrar equipo* es similar al *caso registrar estudiante*, solo que esta vez el registro se realizará mediante un código único por equipo, el cual fue construido mediante un código de barras y las características de un dispositivo en particular, el diagrama de actividad muestra la serie de pasos y condiciones para concluir el registro de un equipo, el caso registrar equipo se muestra en la tabla 4.6 con sus debidas excepciones (número 3 y 4).

En esta sección solo se muestran algunos diagramas considerados como los más importantes, si desea ver la documentación completa revise la sección de apendices de este trabajo.

Registrar equipo		
<b>Objetivos asociados</b>	Registrar equipo	
<b>Requisitos asociados</b>	Contar con equipo etiquetado e identificado	
<b>Descripción</b>	El sistema será capaz de registrar un nuevo equipo en la base de datos SAMLAB	
<b>Precondición</b>	El equipo no está dado de alta en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El usuario administrador del laboratorio solicita al sistema comenzar el proceso de registro de nuevo equipo
	2	El sistema solicita los siguientes datos del nuevo equipo: modelo, marca, estado, nombre y código
	3	El usuario administrador comprueba que los datos ingresados sean correctos y el sistema muestra una pantalla de éxito
<b>Postcondición</b>	El nuevo equipo está disponible en la base de datos y puede ser presentado para préstamo	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3	Si el sistema detecta que el nuevo equipo ya está registrado en la base de datos no realizará el registro y le pedirá que intente de nuevo.
	4	Si el usuario administrador solicita cancelar la operación, el sistema cancela la operación, a continuación este caso de uso termina

Tabla 4.6: Tabla de requerimientos para el caso registro de equipo

#### 4.3.0.15. Caso eliminar equipo

El diagrama de actividad mostrado en la figura 4.15 corresponde al *caso eliminar equipo*, el cual comienza al escanear el código que se desea dar de baja del sistema, cabe mencionar que la validación de la matrícula es un módulo recuperado del *caso eliminar estudiante*. A consideración del administrador del laboratorio, un equipo electrónico se da de baja cuando: el equipo se ha averiado en la mayoría de sus capacidades y no es posible trabajar con el, y cuando el equipo registrado ya esté en desuso por parte de los estudiantes y el laboratorista.

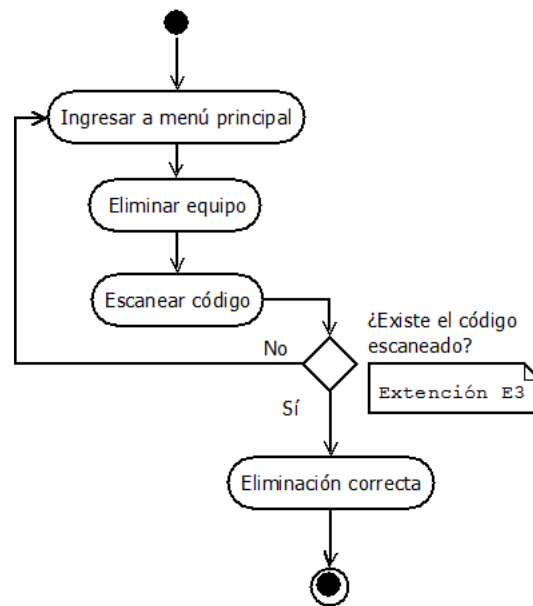


Figura 4.15: Diagrama de actividad eliminar equipo

#### 4.3.0.16. Caso editar equipo

La edición de un registro de equipo existente se realiza en ésta sección, el sistema es capaz de consultar un equipo a través del código escaneado, además se puede editar cualquier campo del registro excepto el código asignado, tal y como sucede con la matrícula de un estudiante. El diagrama de actividad para editar el registro de equipo se muestra en la figura 4.16.

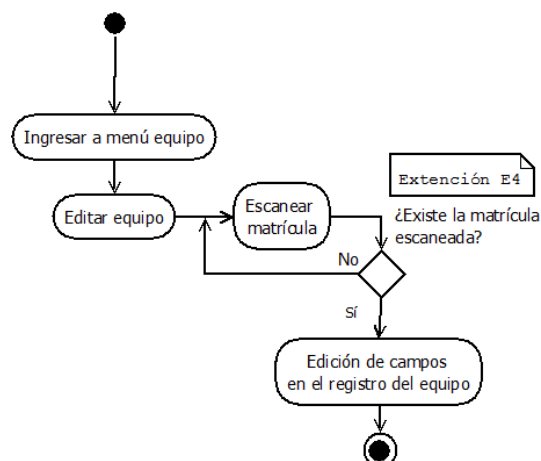


Figura 4.16: Diagrama de actividad editar registro de equipo

#### 4.4. Caso Realizar Préstamo

Se ha llegado a la parte de préstamo de equipo dentro del desarrollo del sistema SAMLAB UACM-SLT v1.0. Para poder realizar un préstamo se debe relacionar información de un estudiante (disponible en tabla alumnos) y la información de equipo (disponible en tabla material). Los requerimientos para este caso corresponden al resultado de la relación entre varios casos externos, la tabla 4.7 muestra el caso realizar préstamo, y en él todos los casos que usa para su funcionamiento. A continuación la explicación de cada caso.

Realizar Préstamo		
<b>Objetivos asociados</b>	Realizar un préstamo de equipo	
<b>Requisitos asociados</b>	Que el equipo y el estudiante en cuestión estén dados de alta en la base de datos SAMLAB	
<b>Descripción</b>	El sistema será capaz de realizar el préstamo de equipo y guardar un registro en la base de datos SAMLAB	
<b>Precondición</b>	El estudiante debe presentar la credencial de estudiante y debe estar dado de alta en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El usuario administrador del laboratorio solicita al sistema comenzar el préstamo en el menú principal y el administrador escanea la matrícula del estudiante
	2	El sistema identifica la situación del usuario y la muestra en la pantalla del administrador: las opciones son: activo/pasivo, con préstamo o sin préstamo, existe/no existe en la base de datos.
	3	El administrador escanea el código del equipo a utilizar y da click en guardar
	4	El registro del préstamo ha sido creado y terminará cuando el estudiante realice la devolución del equipo
<b>Postcondición</b>	El nuevo equipo está disponible en la base de datos y puede ser presentado para préstamo	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Si el sistema detecta que el estudiante que desea un préstamo se muestra la leyenda en pantalla, si detenta que no existe registro en la base de datos se invita a dar de alta al estudiante.
	3	El sistema detecta si el equipo escaneado no existe en la base de datos y no realiza el préstamo

Tabla 4.7: Tabla de requerimientos para el caso préstamo

Para realizar un préstamo se requiere de validaciones con respecto al estudiante y al equipo, es decir, el administrador siempre sabrá lo que sucede con respecto a un estudiante y con respecto al equipo, a continuación la explicación de dichas validaciones.

- **Validación de un estudiante:** Se verifica la situación actual del estudiante que desee un préstamo y existen tres posibilidades: el estudiante podría no estar dado de alta en el sistema, en este caso se le invita a realizar el registro con ayuda de *caso registrar estudiante*; el estudiante podría tener préstamo activo, es decir, ya tiene equipo en préstamo

y aún así podrá solicitar otro equipo y se agregará en su registro activo y por último, el estudiante se encuentra activo en el sistema y podrá realizar préstamo, en este último caso la etiqueta estudiante activo se muestra al administrador en el momento que el sistema valida la matrícula, de lo contrario aparecerá como estudiante inactivo y no podrá realizar un préstamo, la validación de este caso se muestra en una impresión de pantalla en la figura 4.17, tomando la matrícula de un estudiante registrado en el sistema.



Figura 4.17: Estudiante activo para realizar un préstamo

- **Validación de un equipo:** La situación de cada dispositivo es validado en esta sección, de la misma forma existen tres posibilidades: la primera el equipo podría no estar registrado en el sistema y se invita al administrador a darlo de alta en ese momento utilizando el caso registro de equipo; la segunda el equipo solicitado se encuentra en préstamo (es difícil que suceda este caso ya que el equipo tiene un código único e irrepetible), si llegara a suceder este caso garantiza no realizar el préstamo de un mismo equipo a dos estudiantes en el mismo periodo de tiempo; y por último, el caso donde el equipo puede salir a préstamo, en este punto se genera un ciclo repetitivo para que el administrador vaya agregando el equipo que el estudiante desee, el préstamo está limitado a nueve equipos por estudiante. En la figura 4.18 se muestra un diagrama de despliegue para agregar equipo de manera iterativa, como primer paso se escanea el código del equipo, después se debe decidir si se agrega equipo a préstamo o se cancela el proceso, y como último paso se debe guardar el préstamo o cancelar.

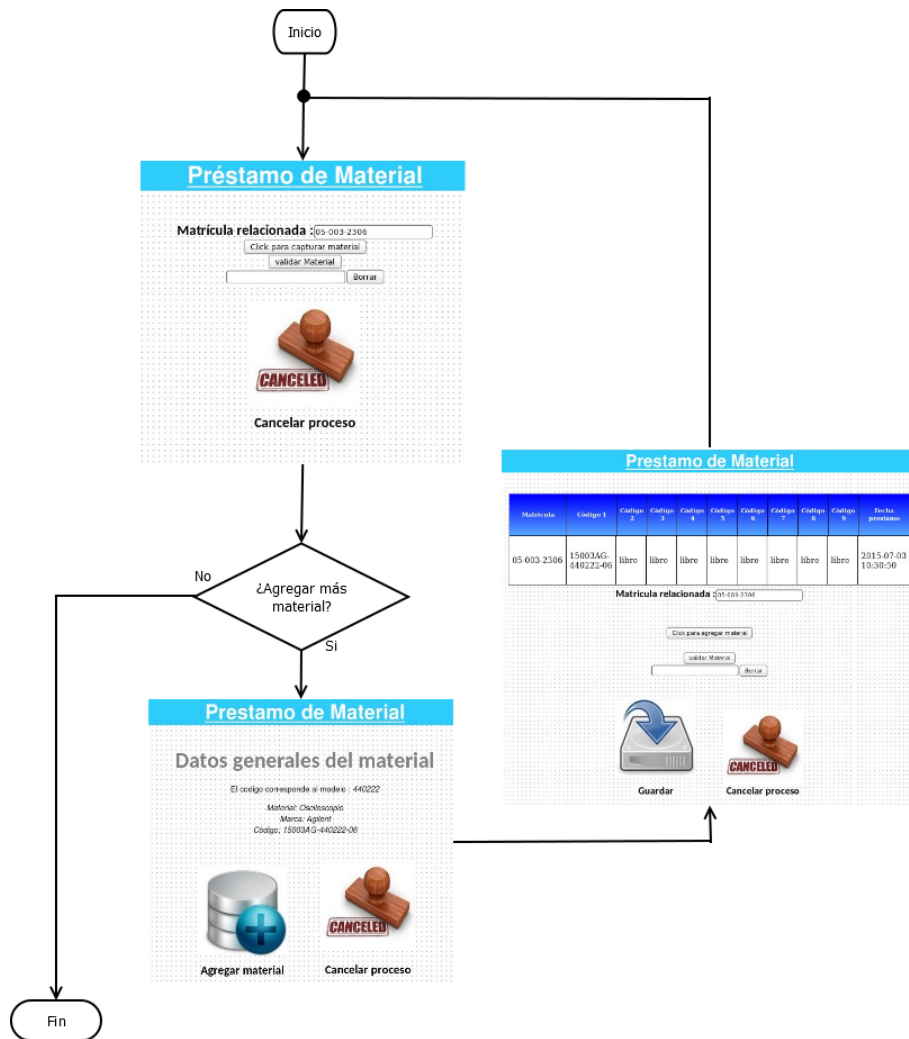


Figura 4.18: Ciclo iterativo para agregar equipo a un préstamo

Al momento que el administrador solicita un préstamo, se valida la matrícula del estudiante y se comienza a validar el equipo; el sistema ha creado un espacio en la tabla préstamo que debe ser llenado con todo lo que requiera el estudiante, si por alguna razón el préstamo no se realiza se debe eliminar dicha sesión. Para este caso se tiene la opción *cancelar proceso* en todo momento, el cual destruye el proceso realizado en cualquier ciclo del mismo, incluso desde la validación del estudiante.

El diagrama de secuencia para realizar un préstamo en el caso normal se muestra en la figura 4.19, de igual forma se consideran las validaciones que se utilizaron.

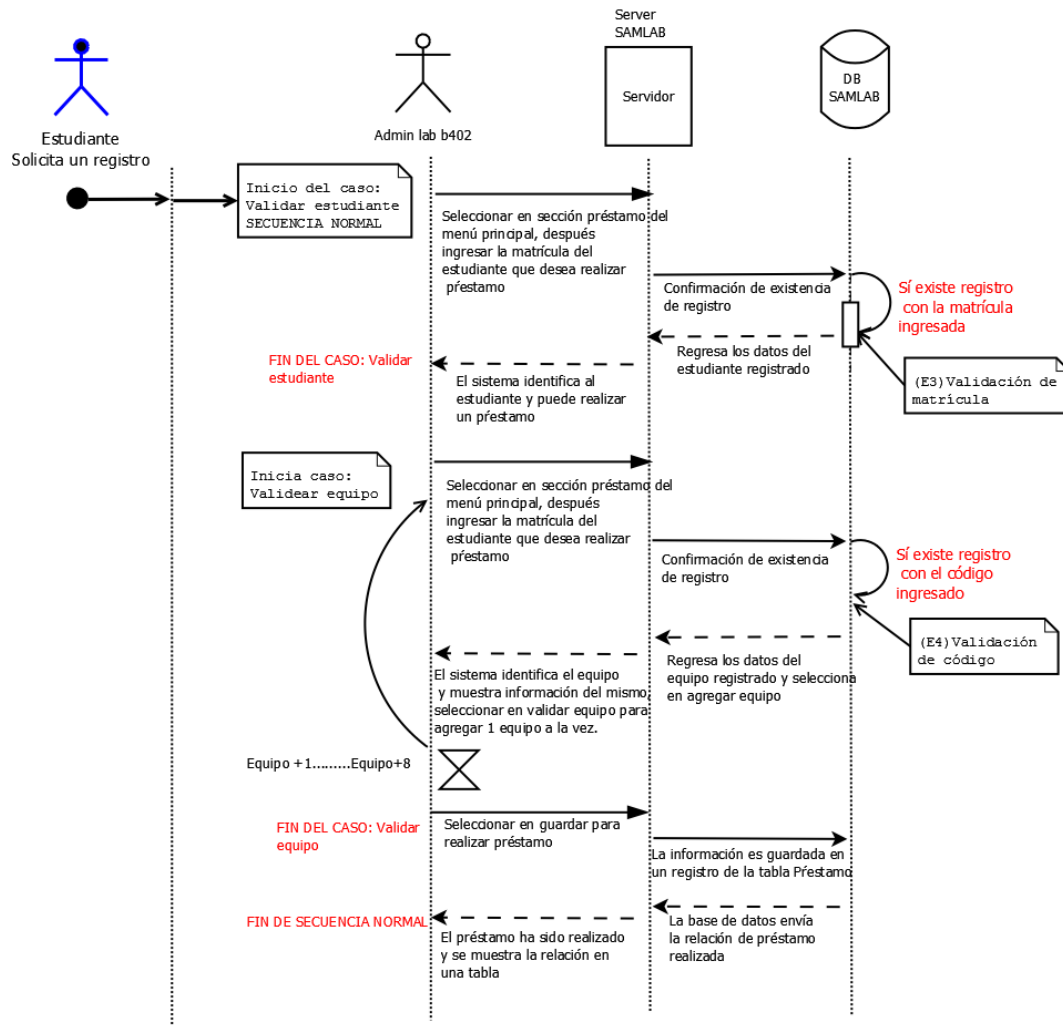


Figura 4.19: Diagrama de secuencia para realizar un préstamo, considerando un caso ideal

Para el *caso realizar préstamo* la validación de estudiante activo y equipo disponible son muy importantes a la hora de realizar un préstamo, para garantizar que un préstamo cumpla con los requerimientos establecidos, se consideraron los siguientes puntos:

- El estudiante debe de estar registrado en el sistema y estar como estudiante activo en el sistema SAMLAB UACM-SLT v1.0
- El estudiante que se encuentre con préstamo activo, podrá agregar equipo o cambiar un equipo por otro y continuar con su préstamo,

dichos módulos de cambios se revisarán más adelante.

- El equipo debe de estar registrado en el sistema SAMLAB UACM-SLT v1.0
- El equipo seleccionado para préstamo no debe encontrarse prestado a ningún otro estudiante
- El equipo se encuentra en préstamo activo, este caso solo sucedería si existen etiquetas con códigos repetidos.

En la figura 4.20 se muestra el diagrama de actividad del *caso préstamo*, incluyendo las validaciones de estudiante ya mencionadas que podrían ocurrir. La figura 4.21 es la continuación del *caso realizar préstamo* e incluye las validaciones del equipo previamente mencionadas.

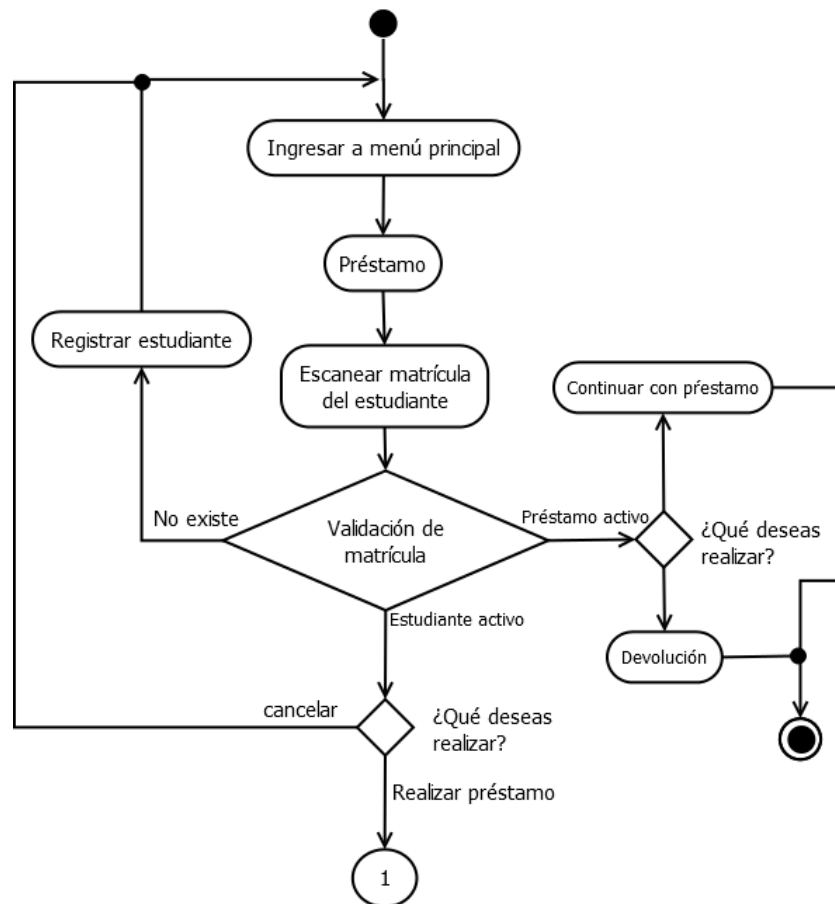


Figura 4.20: Diagrama de actividad del caso préstamo con validación de estudiante, condiciones normales

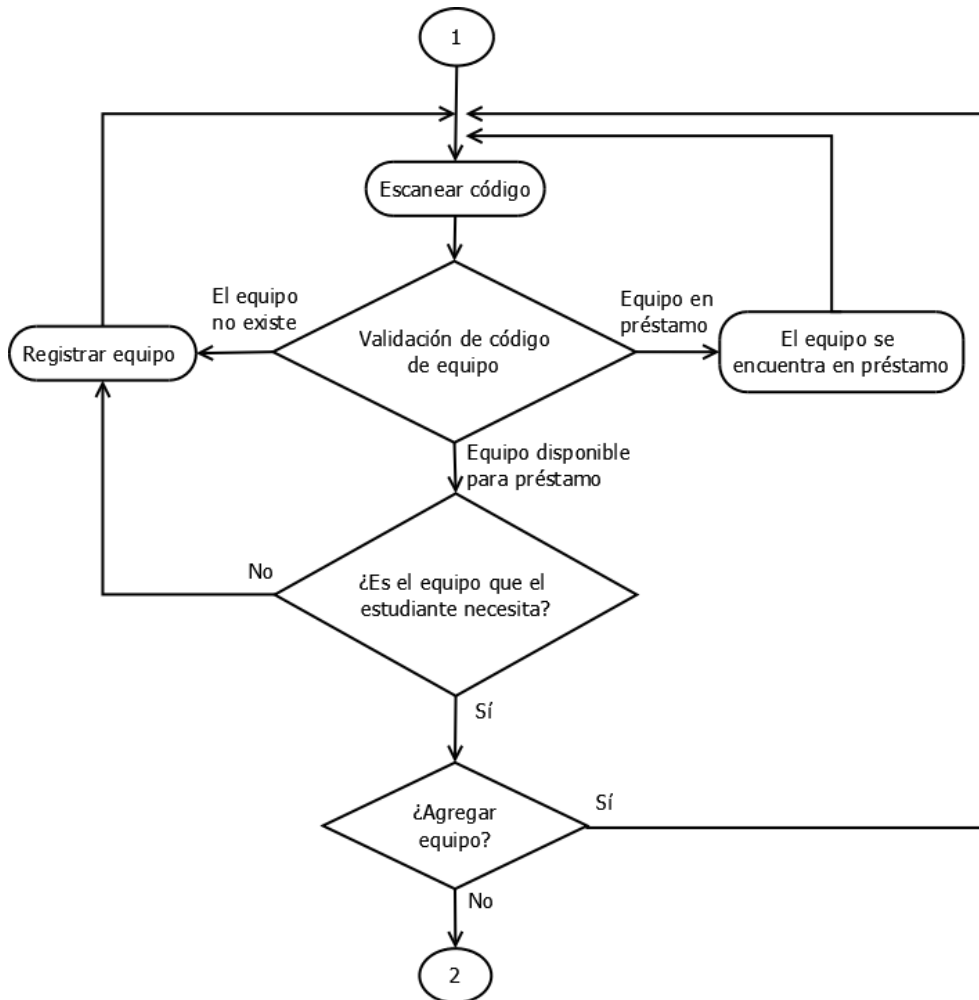


Figura 4.21: Continuación del diagrama de actividad del caso préstamo, bajo las condiciones de validación de equipo, considerando la existencia de equipo disponible en el sistema

La relación entre la información del estudiante y el equipo ha sido realizada en este momento, ya que se ha creado un registro de actividad en la tabla préstamo y solo resta guardar el proceso de préstamo o cancelarlo. La figura 4.22 muestra el diagrama de actividad que ilustra las validaciones correspondientes al préstamo y crea un campo con la opción de guardar el equipo seleccionado o eliminar el proceso. La opción cancelar préstamo es muy importante en toda la actividad, ya que esta opción destruye el registro creado en la tabla préstamo, en cualquier punto durante la operación, lo que no se debe de hacer es dejar un préstamo abierto, se selecciona guardar préstamo o cancelar préstamo. El diagrama de actividad de la figura 4.30

completa el flujo del caso préstamo.

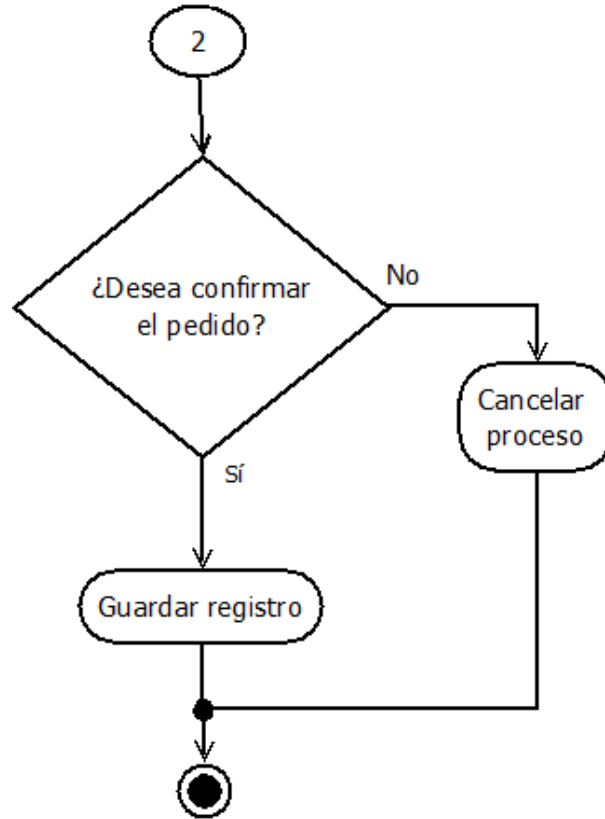


Figura 4.22: Diagrama de actividad cancelar Préstamo, muestra la opción de cancelar durante el préstamo

En este momento el registro de préstamo ha sido generado y guardado en la tabla préstamo además la fecha y hora de la operación. El registro quedará bajo la condición abierto o pendiente de entregar hasta que se realice la correcta devolución del equipo que se registro en el préstamo. A continuación la explicación del caso devolución.

## 4.5. Caso Realizar Devolución

La devolución del equipo consiste en registrar la terminación del ciclo préstamo, garantizando que el estudiante devolvió en tiempo y forma todo el equipo que le ha sido prestado, es decir, este módulo confirma la operación y genera registro de la actividad. La única forma de tener acceso a la información mencionada es mediante la petición del administrador. Los requisitos del caso están mostrados en la tabla 4.8. Es preciso comentar, que

el registro de la devolución no va a una tabla de la base de datos SAMLAB con el mismo nombre, el proceso de registro del proceso completo inicia con un préstamo y el registro se almacena temporalmente en la tabla préstamo, cuando el administrador realiza una devolución, el registro iniciado en la tabla *prestamo* es llevado a un registro nuevo en la tabla *historial*, de esta forma se ahorran recursos de manejo de información en la base de datos. La figura 4.23 muestra el diagrama de secuencia del *caso devolución* y junto

Realizar Devolución		
<b>Objetivos asociados</b>	Realizar una devolución de equipo	
<b>Requisitos asociados</b>	Que el equipo y el estudiante en cuestión tengan un préstamo activo vigente	
<b>Descripción</b>	El sistema será capaz de realizar la devolución de equipo y guardar un registro en la base de datos SAMLAB	
<b>Precondición</b>	El estudiante debe presentar la credencial y el material para realizar la devolución	
<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario administrador del laboratorio solicita al sistema comenzar la devolución en el menú principal y el administrador escanea la matrícula del estudiante
	2	El sistema identifica la situación del usuario y la muestra en la pantalla del administrador: las opciones son: devolución parcial y devolución total
	3	El administrador elige devolución total y escanea el código del material a devolver, después selecciona en guardar
	4	El registro de la devolución ha sido creado y quedará guardado en la tabla historial
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	2	Existen 2 situaciones en donde no se puede realizar la devolución: 1) Si el estudiante no se encuentra dado de alta en el sistema y 2) Si el estudiante no tiene registro de préstamo activo
	3	Existen 2 situaciones en donde no se puede realizar la devolución: 1) Si el equipo no se encuentra dado de alta en el sistema y 2) Si el equipo no tiene registro de préstamo activo

Tabla 4.8: Tabla de requerimientos del caso: Realizar Devolución

los casos externos involucrados por ejemplo, el *caso préstamo* que valida la matrícula del estudiante relacionada con el equipo prestado, dicha relación se encuentra registrada en la *tabla préstamo*.

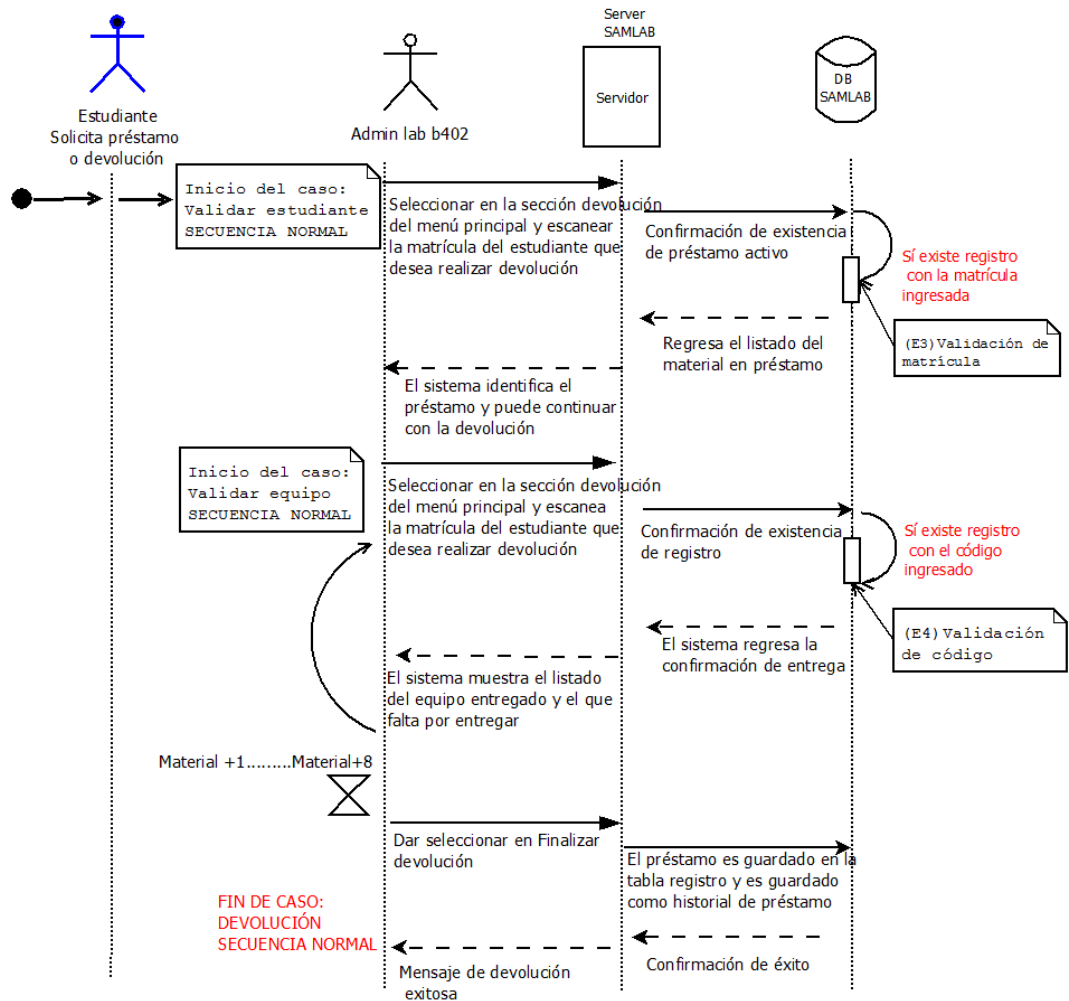


Figura 4.23: Diagrama de secuencia de la devolución de un préstamo secuencia normal

En el diagrama de actividad de la figura 4.24, se observa que lo primero que se realiza para entregar un equipo que se encuentra en préstamo es verificar la matrícula del estudiante, ya que se identificó la matrícula se genera la relación del equipo asignado a la misma, en este punto se puede entregar equipo, cambiar equipo o agregar equipo, para este caso se desea entregar, por lo que se genera un ciclo repetitivo en la entrega, la cual debe de garantizar la entrega correcta del equipo, mediante la lectura del código correspondiente, con la ayuda del lector de códigos.

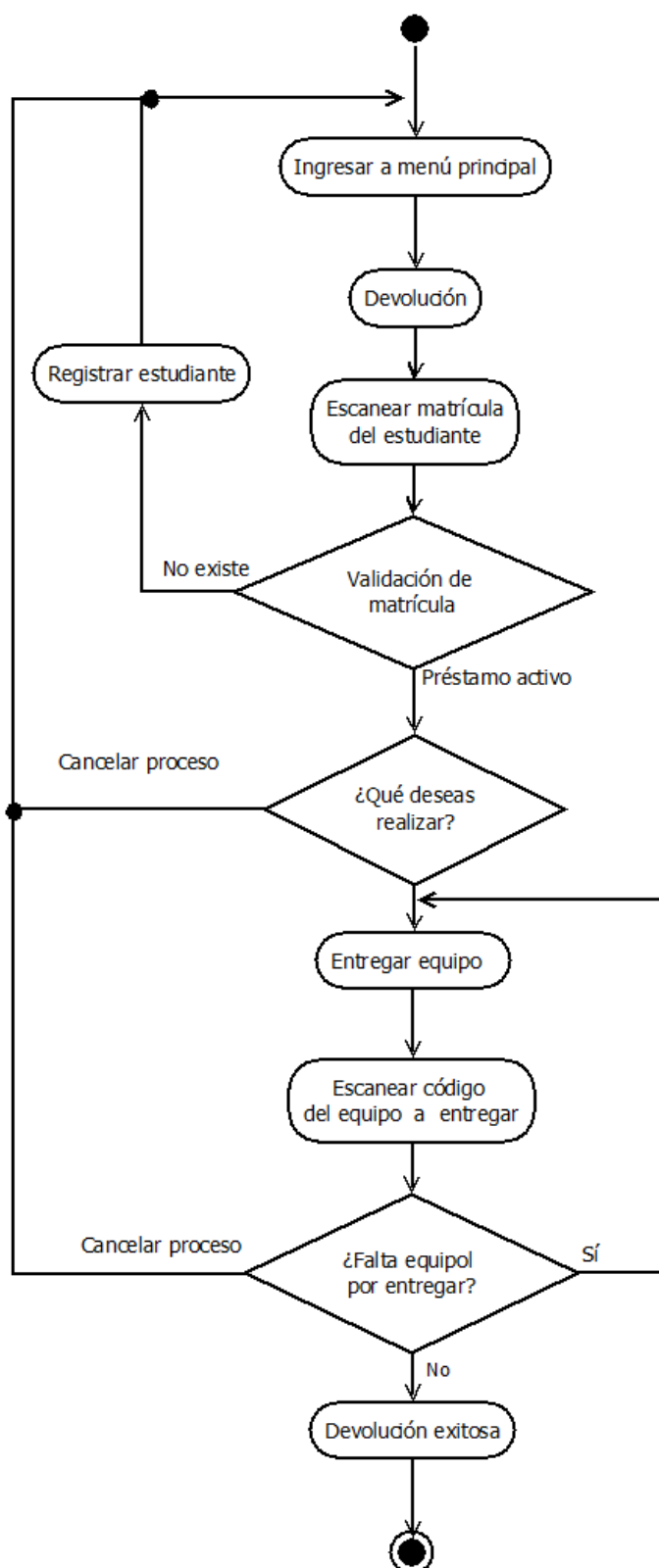


Figura 4.24: Diagrama de actividad del caso realizar devolución secuencia normal

## 4.6. Caso Realizar Cambios

Los casos préstamo y devolución no siempre ocurren de manera normal como se mostró anteriormente, para complementar el caso se ha incluido la sección *cambios en préstamo y cambios en devolución*, los cuales complementan a cada módulo con situaciones como: agregar algún equipo a un préstamo activo o devolver solo un equipo de un préstamo, a continuación la explicación de los casos.

### 4.6.0.17. Caso Realizar Cambios en un Préstamo

El administrador podrá realizar cambios en el préstamo activo de algún estudiante mediante esta sección, sucede que los préstamos en el laboratorio no siempre son estáticos sino totalmente dinámicos. Imagine el siguiente escenario: un estudiante solicita un préstamo de un kit básico que consiste en un multímetro y tres equipos más, resulta que al poner en funcionamiento el equipo, el multímetro no da valores correctos y el estudiante desea cambiarlo por otro multímetro en buen estado además olvidó solicitar un generador de señales. En la figura 4.25 se muestra el diagrama de secuencia del caso cambios en préstamo, el cual puede ilustrar la problemática antes mencionada del multímetro dañado y poder agregar un nuevo equipo al préstamo activo. El caso cambios inicia cuando el administrador elige del menú principal la opción cambios y escanea la matrícula del estudiante para validar que exista un préstamo activo, en este momento debe decidir entre dos opciones: cambiar equipo o agregar más equipos, a continuación la explicación de cada sección.

**Cambiar equipo:** El objetivo de este caso es realizar el cambio de un equipo ya en préstamo por otro nuevo, el caso comienza con la validación en la tabla préstamo ingresando el código del equipo anterior y cambiándolo por el nuevo código, el sistema guarda el cambio exactamente en el mismo lugar del anterior. Cabe mencionar que el cambio solo se realiza por la cantidad de equipos que tenga en préstamo el estudiante.

**Agregar equipo:** Esta sección se utiliza para agrandar el pedido de equipo en un préstamo ya iniciado, es decir, el estudiante podrá registrar hasta nueve códigos de equipos y agregarlos cuando desee. Este caso contiene validaciones de estudiante y equipo directamente en la tabla préstamo, para garantizar que el equipo está disponible para préstamo. La figura 4.36 muestra el diagrama de iteración del caso cambios en préstamo, el caso inicia cuando el administrador elige del menú principal la opción cambios y escanea la matrícula del estudiante para validar que exista un préstamo iniciado. Como ya se mencionó el registro del préstamo permanece en la tabla préstamo hasta que se realiza la devolución del mismo, ya que se completo la entrega del equipo

se selecciona salir, es en ese momento donde toda la información relacionada con un préstamo se guarda en la *tabla historial*.

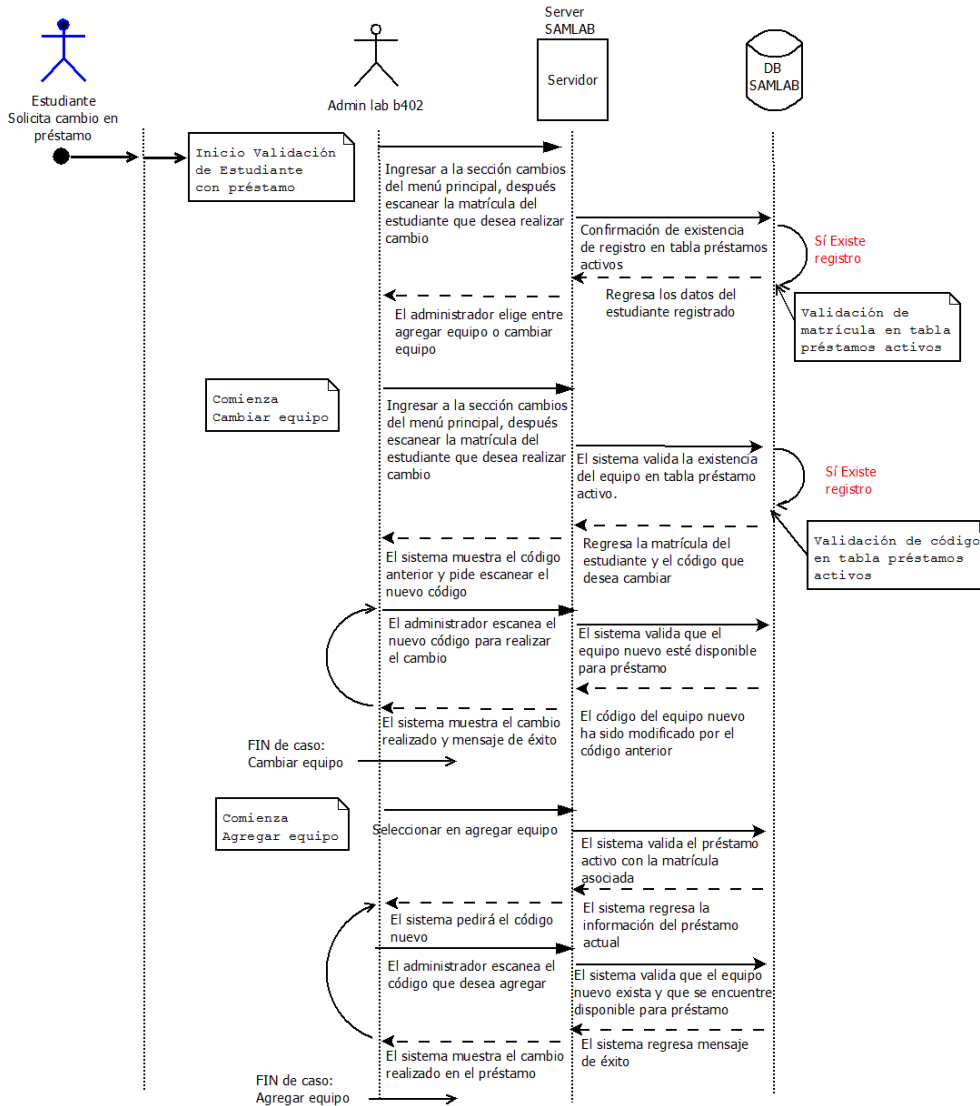


Figura 4.25: Diagrama de secuencia del caso realizar cambios en un préstamo

#### 4.6.0.18. Caso cambios en devolución

El *caso cambios en devolución* no se encuentra como tal en el apartado cambios del menú principal, sin embargo se ha mencionado en esta sección para que el lector considere este tipo de cambio en la sección devolución. La devolución parcial se realiza en el apartado devolución, el administrador puede realizar la devolución de todo el material o solo algunos de ellos, el

caso se muestra en la figura 4.26, la iteración de la devolución del equipo termina hasta que el administrador lo decida o hasta que se se agote el equipo a entregar. Cabe mencionar que la devolución parcial sí quedará registrada en el historial de préstamos de modo que el sistema solo guardará la situación final en la actividad préstamo como haya quedado después de todos los cambios que haya realizado el administrador.

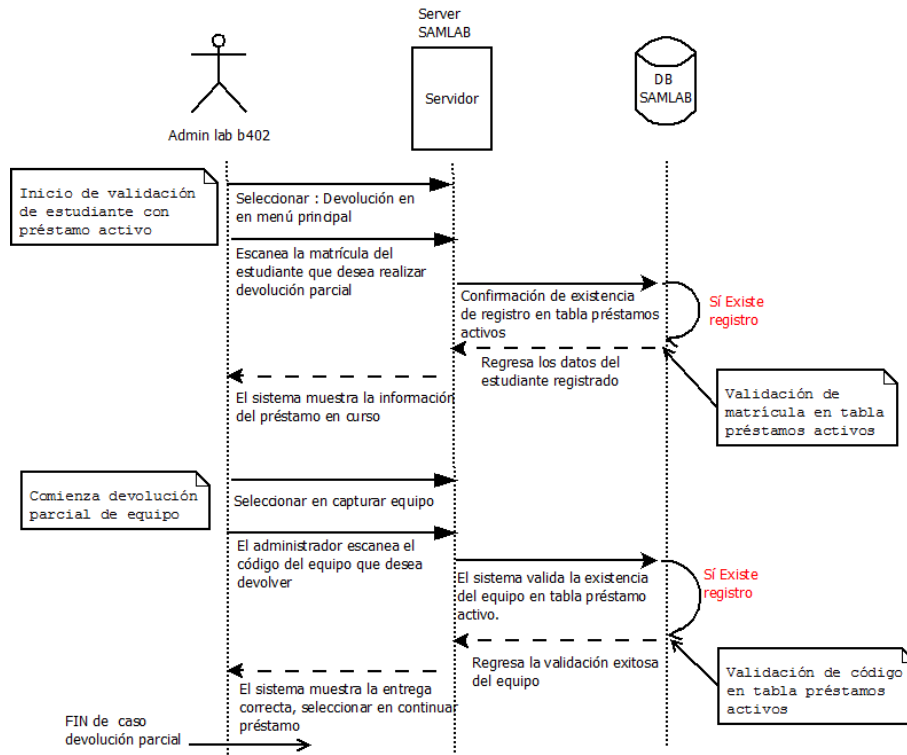


Figura 4.26: Diagrama de iteración del caso Devolución Parcial

Todas las operaciones que puede realizar el administrador han sido descritas hasta este momento, lo siguiente será explicar la sección consultas que será la forma que tendrá el administrador de verificar la información de todas las operaciones realizadas.

#### 4.6.0.19. Caso consultas

La sección consultas es de gran ayuda para el administrador ya que es el lugar donde se verifica cualquier operación realizada actualmente y a través del tiempo. El caso se ha desarrollado en conjunto, tomando en cuenta la consulta de préstamos activos y la consulta del historial. En la figura 4.27, se muestra el diagrama de secuencia juntando los dos escenarios. Las figuras

siguientes 4.28 y 4.29, son las pantallas que verá el administrador al momento de realizar dichas consultas.

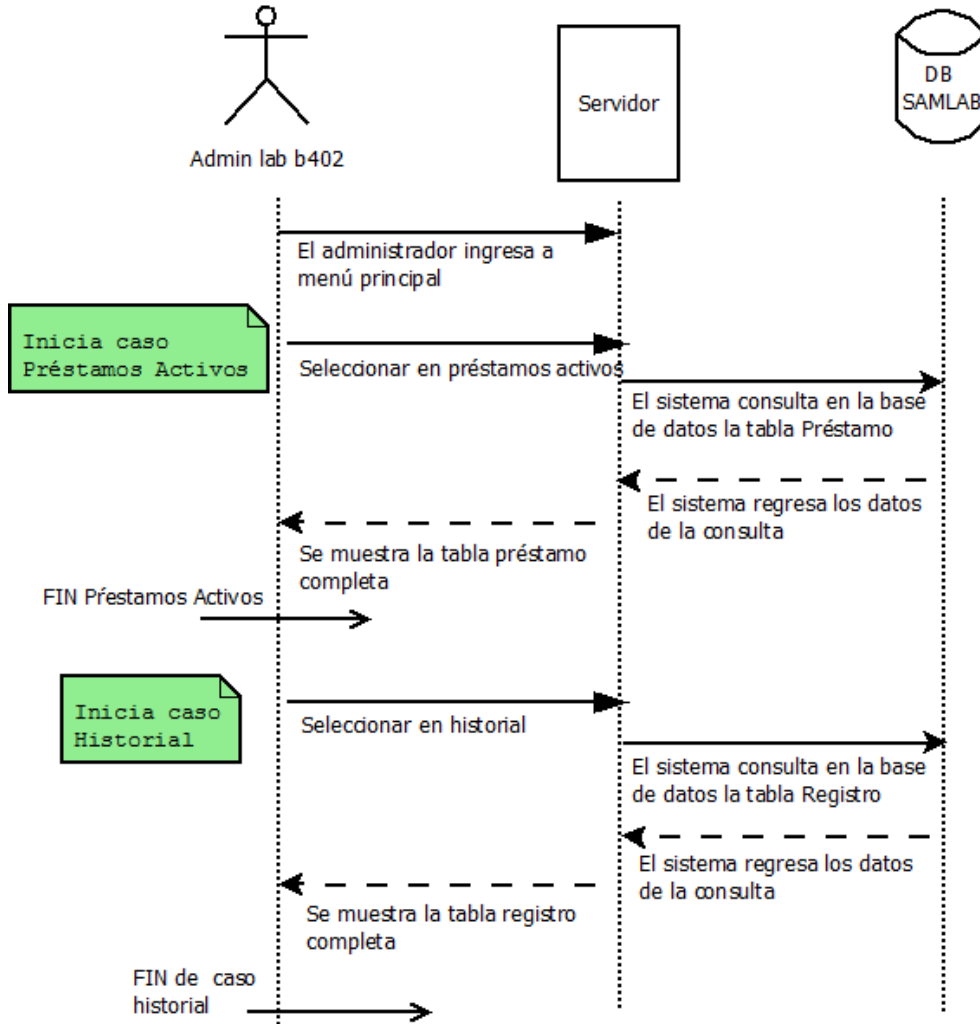


Figura 4.27: Diagrama de secuencia del caso consultas

En la figura 4.28, se muestra la *tabla préstamos* activos, esta tabla guarda la matrícula y nombre del estudiante, el código del equipo en préstamo y la hora de inicio del préstamo, al momento de concluir con el préstamo se realiza una copia de la devolución en una nueva *tabla llamada historial* mostrada en la figura 4.29, en este momento se registra y guarda la fecha y hora de devolución del préstamo en dicha tabla, la cual tiene la capacidad de guardar el número de registros que sea necesario.

RESULTADOS DE LA CONSULTA											
Prestamos de material activos											
Matricula	Nombre	código 1	código 2	código 3	código 4	código 5	código 6	código 7	código 8	código 9	Fecha Prestamo
05-003-2536	toño	15003ME-38XR9-01	libre	libre	libre	libre	libre	libre	libre	libre	2015-04-24 09:19:20
05-003-2532	Efrain	15003ME-38XR9-03	15003ME-38XR9-02	libre	libre	libre	libre	libre	libre	libre	2015-04-24 09:15:14

[Menu Principal](#)
[Usuarios](#)
[Prestamo](#)
[Cambios](#)
[Devolución](#)

Figura 4.28: Consulta de la tabla préstamos activos

Resultados de la consulta												
Matricula	Nombre	código 1	código 2	código 3	código 4	código 5	código 6	código 7	código 8	código 9	Salida	Devolución
05-003-2532	Efrain	15003ME-38XR906	15-003-TK-C057555	15-003-AG-8260161	15003ME-38XR905	15003ME-38XR909	libre	libre	libre	libre	2015-03-18 12:29:00	2015-03-18 12:54:12
05-003-2532	Efrain	15003ME-38XR906	libre	libre	libre	libre	libre	libre	libre	libre	2015-03-18 12:57:49	2015-03-20 09:52:50
05-003-2532	Efrain	15003me-38xr906	15003ag-490136-01	15003ag-490136-01	libre	libre	libre	libre	libre	libre	2015-03-20 10:53:24	2015-03-20 11:18:24
05-003-2532	Efrain	15003ME-38XR901	15003AG-490136-01	libre	libre	libre	libre	libre	libre	libre	2015-03-20 11:26:51	2015-03-20 11:26:51
05-003-2532	Efrain	15003ME-38XR906	15-003-TK-C057555	15003ME-38XR905	15003ME-38XR909	libre	libre	libre	libre	libre	2015-03-25 19:17:33	2015-03-27 11:07:07
05-003-2532	Efrain	15003ME-38XR906	15-003-TK-C057555	libre	libre	libre	libre	libre	libre	libre	2015-03-27 11:08:16	2015-04-04 19:28:01
05-003-2532	Efrain	15003AG-510239-07	libre	libre	libre	libre	libre	libre	libre	libre	2015-04-08 15:12:29	2015-04-09 09:27:25
05-003-2532	Efrain	15003AG-440222-06	libre	libre	libre	libre	libre	libre	libre	libre	2015-04-09 09:26:05	2015-04-09 09:27:25
05-003-2532	Efrain	15003AG-440222-06	libre	libre	libre	libre	libre	libre	libre	libre	2015-04-09 09:27:40	2015-04-10 11:52:03
05-003-2532	Efrain	15003ME-38XR9-01	libre	libre	libre	libre	libre	libre	libre	libre	2015-04-10 12:23:04	2015-04-14 12:04:29

Figura 4.29: Consulta de tabla historial

Ahora el sistema ha quedado terminado en su totalidad, después de haber pasado dos ciclos de mejoras RUP se obtiene la primera versión operacional, lista para ser utilizada por el laboratorio B-402.

Como ya se describió en este capítulo para desarrollar SAMLAB se siguieron las recomendaciones de RUP con el lenguaje de modelado UML, para el siguiente capítulo se abordará una última pero muy importante sección de análisis de resultados, en la cual el sistema estará puesto en marcha en el laboratorio B-402, además se aplicarán pruebas de funcionamiento y operación para analizar el producto terminado.

## Capítulo 5

# Análisis de resultados

*La mejor forma de predecir el futuro es implementarlo*

David Heinemeier Hansson - creador de  
Ruby on Rails

Para asegurar la calidad de un desarrollo de software se recurre a la realización de actividades predispuestas a descubrir errores, para encontrarlos se crean casos prueba o flujos definidos que simulan el funcionamiento en producción real, para que de esta forma se pueda encontrar la mayor cantidad de detalles, problemas, y errores en un sistema de software. Aunque no hay una clasificación formal acerca de los diversos tipos de pruebas de software, existen dos vertientes fundamentales:

- **Pruebas de caja negra:** También llamadas (*pruebas de caja opaca y pruebas de entrada/salida*) consisten en analizar el funcionamiento del sistema en módulos o de manera completa, los datos a ingresar se derivan de pruebas definidas por los desarrolladores y el objetivo de este tipo de pruebas es comprobar si los datos de salida corresponden a la entrada sin mirar el funcionamiento interno ni la parte lógica del sistema.
- **Pruebas de caja blanca:** Este tipo de pruebas son un tanto más profundas, se realizan desde la parte lógica del sistema incluyendo detalles de procedimiento, se prueban los caminos lógicos que toma una consulta, un registro o una actividad importante. Esta prueba necesita tener conocimiento específico del software para analizar los resultados.

Algunas otras pruebas que se realizan para comprobar el funcionamiento de una aplicación de software son: de unidad, de módulos, de estrés, de carga y de rendimiento entre otras.

Para probar el software desarrollado se utilizó el paradigma de las pruebas de caja negra, específicamente se implementaron pruebas de funcionamiento en 2 fases o partes y una prueba básica de operación del sistema. A continuación se describen los resultados de dichas pruebas. En esta sección se describen los resultados de las pruebas que se realizaron en el sistema, las pruebas consisten en comprobar el funcionamiento y el desempeño del sistema llevado a producción en sitio, es decir, realizando sesiones de préstamo y devolución en tiempo real. Los resultados que se obtuvieron de las pruebas son los siguientes.

### 5.1. Pruebas de funcionamiento

La aplicación de software SAMLAB UACM-SLT v1.0 debe tener características tales que cumpla con el objetivo del proyecto, es decir, que el software esté perfectamente orientado a los usuarios, en este caso los administradores del laboratorio para que realmente pueda ser aplicado para resolver el problema en cuestión, en este sentido se realizaron dos pruebas de funcionamiento y una de operación real en el laboratorio, se contó con la colaboración del encargado de los laboratorios de electrónica y de física el Ing. Ricardo Zamora Paredes, para seguir y contestar una guía de funcionamiento y una prueba de operación del sistema utilizando el paradigma de la caja negra, para revisar posibles fallas o detalles que el sistema pudiera presentar. La primera prueba que se realizó fue la prueba de funcionamiento en dos fases, a continuación la explicación y los resultados de cada fase.

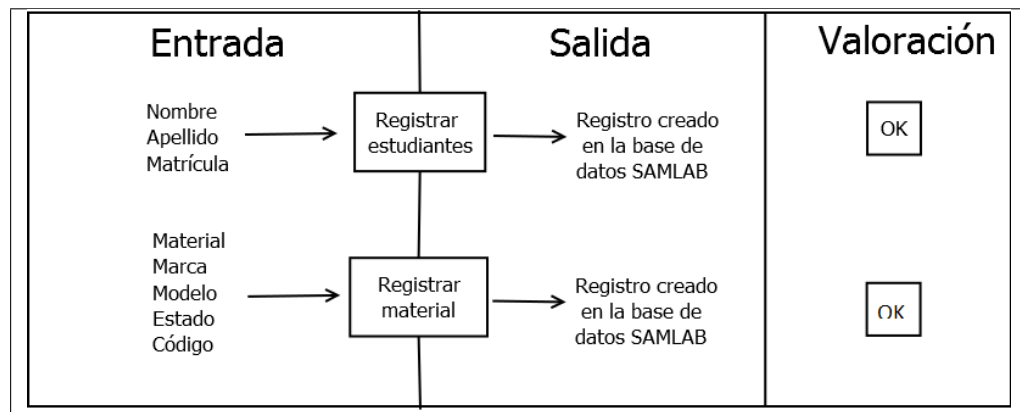


Tabla 5.1: Tabla de prueba de caja negra de módulo registrar fase 1

### 5.1.1. Prueba de funcionamiento del sistema fase 1:

Consiste en una tabla editable que el laboratorista contestó para comprobar el resultado de ingresar datos específicos en cada módulo operacional del sistema: *registro de estudiantes (registrar, eliminar, editar)*, *equipo (registrar, eliminar, editar)*, *préstamo y devolución*. Cabe mencionar que esta primera prueba corresponde a la primera iteración RUP de la cual se debe obtener la primera versión del sistema operacional. La Tabla 5.1 muestra un diagrama básico de caja negra, del lado izquierdo los datos que el administrador debe ingresar en el módulo en cuestión, el lado derecho corresponde a los datos o acciones que el sistema debe arrojar.

La valoración de cada caso esta representado por un OK si el resultado es lo que se esperaba y un KO si se ha encontrado algún error, en la prueba de caja negra que se realizó para el módulo registrar(estudiantes y equipo), se obtuvo una valoración OK.

La situación anterior es la misma para el módulo *eliminar estudiantes y eliminar equipo*, los resultados de la prueba se muestran en la Tabla 5.2.

Entrada	Salida	Valoración
Matrícula →	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Eliminar estudiantes</div> → Registro eliminado en la base de datos SAMLAB	<div style="border: 1px solid black; padding: 5px; display: inline-block;">OK</div>
Código →	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Eliminar equipo</div> → Registro eliminado en la base de datos SAMLAB	<div style="border: 1px solid black; padding: 5px; display: inline-block;">OK</div>

Tabla 5.2: Tabla de prueba de caja negra de módulo eliminar fase 1

La prueba de funcionamiento fase 1 continua con el préstamo y devolución de uno o varios equipos, en este caso se realizó la prueba con un multímetro y los resultados se muestran en la tabla 5.3. La prueba tiene una tercera parte correspondiente a las consultas de historial y consulta de préstamos activos, las cuales también resultaron funcionales en su modo más básico.

Entrada	Salida	Valoración
Matrícula Código de equipo →	Préstamo → Registro de préstamo activo	OK
Matrícula Código de equipo →	Devolución → Registro de entrega de equipo	OK

Tabla 5.3: Tabla de prueba de caja negra de módulos préstamo-devolución fase 1

### 5.1.2. Prueba de funcionamiento del sistema fase 2:

El banco de pruebas para verificar el funcionamiento de la fase 2, consiste en verificar algunas mejoras en el sistema, corresponde a la segunda iteración RUP, la cual se enfoca en que el administrador pueda editar campos en algún registro, tanto de estudiantes como de equipo, la tabla 5.4 muestra la operación realizada, cabe mencionar que en este caso los datos de entrada son la matrícula del estudiante y el código de equipo.

Entrada	Salida	Valoración
Matrícula →	Consulta específica de estudiante → Consulta de estudiante específico	OK
Código de equipo →	Consulta específica de equipo → Consulta de equipo específico	OK

Tabla 5.4: Tabla de prueba de caja negra del módulo editar registro fase 1

Una mejora más que se realizó en esta sección fue la consulta específica, el administrador podrá consultar el registro de un estudiante o un equipo en particular, la valoración fue satisfactoria la tabla 5.5 muestra los resultados obtenidos.

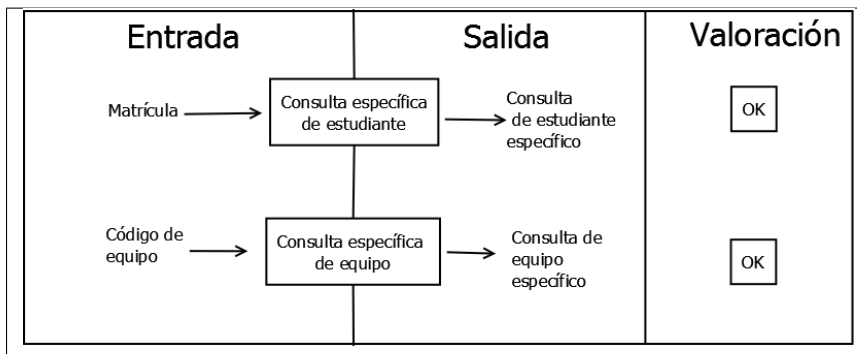


Tabla 5.5: Tabla de prueba de caja negra del módulo consultar registro fase 2

Las siguientes mejoras fueron incluidas en esta iteración y corresponden a cambios en el préstamo y cambios en la devolución, estos cambios fueron solicitados por el laboratorista desde la primera iteración RUP, ya que la acción préstamo es demasiado dinámica en situaciones reales, por ejemplo: un estudiante debe poder agregar equipo a su registro de préstamo y continuar activo, o el mismo usuario puede intercambiar un equipo por otro y todo en el mismo registro activo; lo mismo sucede con la devolución del equipo por ejemplo: un estudiante puede devolver solo un equipo de cuatro que tiene en préstamo y continuar con el préstamo activo. Las pruebas realizadas son de la misma forma utilizando el paradigma de caja negra, los resultados han sido satisfactorios y se muestran en la figura 5.6.

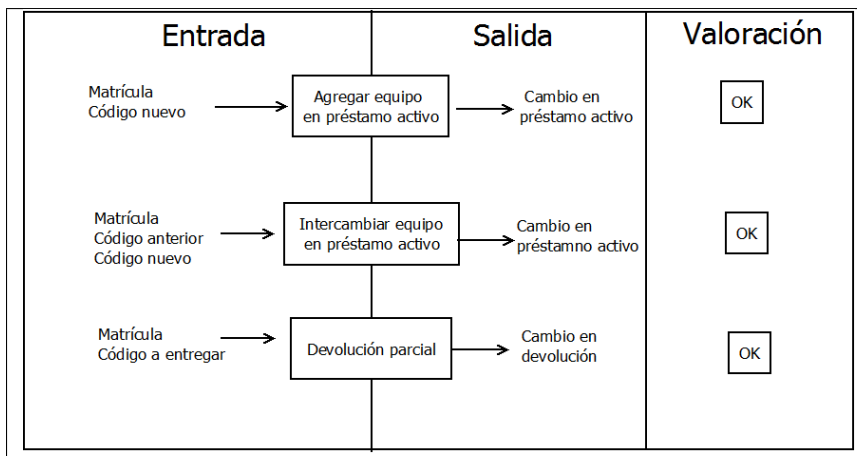


Tabla 5.6: Tabla de prueba de funcionamiento de módulo cambios en préstamo y devolución fase 2

La prueba de la caja negra fue utilizada en este desarrollo ya que este tipo de pruebas se realiza en la superficie del software, es decir, a través de una GUI. La prueba de funcionamiento completa ha sido terminada en este momento, para continuar con el análisis se realizó una prueba de operación la cual se describe a continuación.

## 5.2. Prueba de operación

Para completar las pruebas se realizó la última de ellas, la prueba de operación en situaciones reales, esta prueba consistió en comprobar el funcionamiento del sistema directamente en el laboratorio B-402, donde el administrador realizó las primeras acciones de préstamo y devolución con estudiantes activos en el laboratorio, cabe mencionar que en las tablas 5.7 y 5.8 solo se muestra una prueba de un ciclo completo (préstamo-devolución), se realizaron tres análisis de ciclos completos de préstamo-devolución.

Préstamo #1									
1.- Fecha y Hora del préstamo	2.- Estudiante	3.- ¿Cómo agregaste el equipo?		4.- ¿Se realizó algún cambio de equipo?		5.- ¿Se agregó equipo en el préstamo activo?		6.- ¿Se logró realizar el préstamo?	
		# de juego	Equipo individual	SI	NO	SI	NO	SI	NO
2015-12-16 10:22:33	Nombre: Erika Alva Suarez Matricula:07-001-0019		1.- Multímetro 2.- 3.- 4.- 5.-	¿Podrías indicar el cambio con el número de equipo?		X	¿Podrías indicar el cambio?	X	X

Tabla 5.7: Tabla de operación del sistema fase 1

Devolución #1							
1.- Fecha y Hora de la devolución	2.- Estudiante	3.- ¿Se devolvió el equipo completo?		4.- ¿Devolvieron equipo parcialmente?		5.- ¿Se logró realizar la devolución?	
		SI	NO	SI	NO	SI	NO
2015-12-16 14:22:36	Nombre: Erika Alva Suarez Matricula:07-001-0019	X		¿Podrías indicar la devolución con el número de equipo?	X	X	

Tabla 5.8: Tabla de operación del sistema caso devolución fase 1

Para finalizar con el análisis de resultados se realizó una pequeña encuesta al laboratorista, los comentarios que se obtuvieron acerca de su experiencia en relación al manejo del sistema, son los siguientes.

*Encuesta realizada el 18 de diciembre 2015 a las 10:00 am*

1.- ¿Considera que todos los módulos que conforma el sistema (registros, operación y consultas) son necesarios? de no serlo ¿podrías proponer alguno?

R: Por el momento los módulos son necesarios para el correcto funcionamiento del sistema.

2.- ¿El sistema te ayudó a agilizar el préstamo de equipo? ¿Por qué?

R: En estos momentos el sistema no ayuda a la agilización del préstamo del equipo, más que nada que los registros de equipo y alumnos se captura manualmente, pero creo que teniendo la base de datos tanto de alumnos como de equipo actualizada y guardada se agilizarían los tramites.

3.- ¿El sistema te ayudó a agilizar la devolución del equipo? ¿Por qué? R: Estamos en la misma situación que si trabajáramos con el modo de llenado de papeletas, aunque con el paso del tiempo el periodo se reduciría por la adaptación del usuario al sistema.

4.- ¿Qué módulo o módulos agregarías para completar su funcionamiento en este momento?

R: A mi forma de ver solo podríamos cuantificar cuántos módulos se necesitarían más adelante, con el simple hecho de trabajar con el sistema ya que realizando esta labor cotidianamente veremos las necesidades que fueran surgiendo o las deficiencias que podría tener el mismo sistema, podría sugerir algunos en este momento como son impresión, estadísticas, concatenación con la base de datos de los alumnos de la UACM etc.

5.- ¿Consideras importante utilizar sistemas como SAMLAB para la vida diaria en los laboratorios de electrónica de la UACM? ¿Por qué?

R: Claro que sí, con el pasar del tiempo el sistema se puede ir mejorando o adaptando a las necesidades de los laboratorios y así reducir los tiempos de entrega recepción del equipo.

6.- ¿Podrías darnos alguna opinión acerca del sistema SAMLAB?

R: En mi opinión el sistema puede resolver en gran parte algunas deficiencias que tenemos con lo que respecta a la entrega recepción del equipo, cabe mencionar que al paso del tiempo se podrían implementar algunos módulos que serían importantes, como son: las estadísticas de uso de equipo, el tiempo que utiliza un alumno en promedio el equipo, solo por mencionar algunos.

El administrador también hizo algunos comentarios importantes que deberán

ser considerados para los siguientes desarrollos a continuación sus palabras:

En esta última prueba que realice al sistema muestra algunos errores:

- 1.- Al capturar el equipo solicitado por los alumnos tiene uno que hacerlo uno por uno, es decir te pide llenar el campo, escaneas y te lleva a otra página para continuar agregando equipo al prestamo solicitado, lo ideal sería llenar todos los campos y al último darle guardar.
- 2.- Al entregar el equipo hace lo mismo que al realizar un préstamo, entonces tiene uno que escanear un equipo a la vez, cuando lo mejor sería escanear todo el equipo y darle salir.
- 3.- Considero que el servidor tendría que estar en el laboratorio de electrónica y no en el B-404 ya que por cualquier falla ya sea de energía o de red quedaría inhabilitado el sistema.

Las pruebas y la encuesta realizada demuestran que el sistema funciona basicamente para los módulos que fue hecho, sin duda como expone el Ing. Zamora en la encuesta pasada, los cambios en rapidez de ejecución para realizar un préstamo o una devolución no se ven reflejadas en las primeras acciones, sino que, se debe incluir al sistema en las actividades diarias del laboratorio para que el personal se vaya acostumbrando y pueda utilizar el sistema como una herramienta de apoyo. En cuanto a los desarrollos importantes del sistema SAMLAB UACM-SLT v.1.0 se puede trabajar en agregar mayor seguridad de contenido al sistema y seguridad de acceso.

Los analisis de resultados mostrados en este capítulo servirán para continuar con el crecimiento y la maduración del sistema SAMLAB ya que en cada iteración se pueden realizar pruebas similares o ir adecuando la prueba conforme cambia el sistema. En el siguiente apartado se describen las conclusiones a las que se llegó y el posible trabajo futuro.

# Conclusiones

El desarrollo de un sistema informático siempre deja temas nuevos a desarrollar, por ejemplo; desarrollo del modulo de estadísticas en general, implementar seguridad en la base de datos, entre otros, los cuales podrían ser tema de trabajos o proyectos incluso de tesis; esto fue precisamente parte de lo interesante de participar en el desarrollo del sistema SAMLAB UACM-SLT v1.0. A continuación se describen las conclusiones a las que se llegó al realizar este trabajo.

## **La versión más básica de SAMLAB existe y es totalmente operable**

El sistema se llevó a operación real pasando por pruebas de funcionamiento modular y pruebas de operación, en dichos resultados se nota que el sistema puede realizar un ciclo completo de préstamo y devolución, incluso con todas las variantes de cambios que un usuario administrador básico necesita.

## **Seguridad en la red y automatización de procesos como módulos prioritarios a desarrollar**

La seguridad en la red está a cargo de un firewall donde a través de multiplexaje de puertos controla el acceso de paquetes al sistema, el firewall se encarga de denegar o no el acceso, en cuanto a la seguridad de contenido se tiene una validación de autenticación básica, la cual asegura que ningún usuario podrá conocer el contenido de ninguna página del sistema si no se ha autenticado en el sistema, aunque conozca la ruta de directorios. Estos sencillos escudos para la red no son suficientes ya que los ataques a las redes son cada vez más comunes, por ello se puede pensar en mejorar la seguridad física del servidor y el firewall, aplicar seguridad a la base de datos, encriptar el contenido entre otros filtros. En cuanto a la automatización de los procesos se tiene que trabajar en la concatenación de la base de datos actual con la de estudiantes de la UACM, ya que de esta forma se podría tener información actualizada y certera de cada estudiante que solicite un préstamo.

## **Aplicación de conocimientos aprendidos durante la carrera**

La carrera de Ingeniería en Sistemas Electrónicos y de Telecomunicaciones

comprende varios aspectos de las comunicaciones y la electrónica, en este trabajo se comprobaron algunos de ellos como: creación e implementación de redes de datos, transmisión de datos, creación de codificación alfanumérica, programación orientada a objetos, diseño de sistemas informáticos, metodologías de desarrollo de sistemas, aspectos de seguridad de red, diseño, administración de bases de datos, gestión de proyectos entre otros. La combinación de un poco de cada disciplina nos llevó a la construcción de un sistema informático modular, con el cual podemos demostrar nuestro conocimiento de sistemas y comunicaciones.

### **Dejar un sistema informático para apoyar el crecimiento administrativo en el laboratorio**

El sistema SAMLAB fue desarrollado para cubrir una necesidad de gestión administrativa del laboratorio B-404 y B-402, al finalizar este trabajo se puede considerar que se obtuvieron los objetivos específicos y generales de acuerdo a los requerimientos establecidos al comienzo del desarrollo del sistema, como resultado de todo el trabajo realizado para el desarrollo contamos con la primera versión de SAMLAB UACM-SLT v.1.0 totalmente operativa. El sistema puede ser utilizado en la vida diaria de la institución, si así se propusiera, también se podrían crear planes de servicio social para asignar a una persona cada periodo de tiempo para realizar mejoras al sistema o simplemente administrar la base de datos, con esto concluimos que este trabajo es una pequeña acción pero que puede generar grandes cambios a través del tiempo en nuestra institución.

### **Trabajo futuro**

El sistema SAMLAB V1.0 es de nacimiento escalable, debido a que está conformado por los módulos descritos anteriormente, este trabajo será la base para los compañeros que deseen seguir desarrollando el sistema lo hagan hasta donde su imaginación los lleve, en esta sección se mencionan algunas posibles mejoras que consideramos en conjunto con los administradores del laboratorio a la fecha. A continuación se describen algunos puntos concretos para realizar mejoras o cambios en las diferentes secciones.

- El registro de cada estudiante podría cambiar si a cada usuario del laboratorio se le asignara una cuenta de acceso al sistema, similar a la cuenta única que tenemos como estudiantes en el sistema de estudiantes, además podría incluir un registro de estudiantes más completo, incluyendo quizás una fotografía que el estudiante pueda subir desde cualquier dispositivo.
- Un cambio importante sería la gestión administrativa para que la base de datos de SAMLAB se relacione con la base de datos de estudian-

tes de la UACM-SLT en todo momento (esto no es tarea fácil), para que al momento del registro independiente, el estudiante ingrese su matrícula y automáticamente el sistema SAMLAB tenga acceso a la situación académica actual del estudiante: tira de materias inscritas, semestre en curso y quizás hasta historial académico. Consideramos que de esta forma el registro sería más rápido y más seguro por parte del laboratorio, además la UACM- SLT podrá conocer información real y actualizada de lo que pasa en los laboratorios de electrónica de nuestra universidad.

- Se podrían generar tickets de préstamo y de devolución para tener una evidencia física del préstamo, devolución y cambios.
- Las consultas tanto de estudiantes como de equipo deben ser mucho más específicas para la búsqueda, quizás utilizando palabras claves, detalles de equipo más utilizado en un periodo de tiempo, estudiante con mayor número de préstamos realizados o buscar algún evento como préstamo o devolución conociendo solo la fecha y la hora de la operación, de igual forma ocupar toda esta información para obtener estadísticas que mejoren la administración del equipo y los tiempos de vida de los equipos entre otros.



# Bibliografía

1. Cleveland, W. S. Bell Laboratories y Murray Hill New Jersey. (1988). The Collected Works of John W. Turkey. New York: Chapman Hall.
2. The history of history of software by Campbell Kelly , M. (12 de Septiembre de 2007). The core memory. Consultado: 25 de Mayo de 2014. Disponible en: <http://www.thecorememory.com/THTHS.pdf>
3. Fuentes, Dra. María del Carmen. (2011). Notas del curso: Análisis de Requerimientos. Ciudad de México: Universidad Autónoma Metropolitana.
4. Unican. (1998). Consultado: Septiembre 2006. Disponible en: [www.ctr.unican.es/asignaturas/is1/IEEE830\\_esp.pdf](http://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf).
5. Alliey, A. M. (2009). Diseño de la Interfaz Gráfica en función de los dispositivos Móviles. Buenos Aires, Argentina.
6. Bartholomew, D. (2014). Admin Network and Security. Consultado: Mayo de 2015. Disponible en: [www.admin-magazine.com/Articles/MariaDB-vs.-MySQL](http://www.admin-magazine.com/Articles/MariaDB-vs.-MySQL)
7. Fernandez, C. A. (octubre de 2006). Universidad Tecnológica de la Mixteca. Consultado: 15 de mayo 2013. Disponible en: [http://www.utm.mx/caf/doc/El Proceso Unificado Rational.pdf](http://www.utm.mx/caf/doc/El%20Proceso%20Unificado%20Rational.pdf)

8. Fowler, M. (1999). UML GOTA A GOTA. México: Pearson Educación.
9. Sanchez, R. B. (Septiembre de 2010). Seguridad en Redes. Consultado: 11 de Enero de 2015. Disponible en: [www.uaeh.edu.mx/docencia/Tesis/icbi](http://www.uaeh.edu.mx/docencia/Tesis/icbi)
10. Silberschatz, A., Korth , H., y S. Sudarshan . (2002). Fundamentos de Bases de datos. Madrid, España: McGraw-Hill Inc
11. Camps Paré, R., Casillas Santillán, L. A., Costal Costa , D. (Mayo 2005). Software libre y las Bases de datos. Barcelona: FUOC.
12. Llonch, J. (ENERO de 2010). MySQL. Disponible en: <http://dev.mysql.com/doc/>
13. Marques, A. (2008). Sistema Integral de seguridad y acceso a la red para un departamento de la UPC. Brazil
14. Rodriguez, J. A. (2005). Java script desde cero. Consultado: 14 junio 2015. Disponible en: [www.ciudadfutura.com/javascriptdesdecero](http://www.ciudadfutura.com/javascriptdesdecero)
15. Santillan, L. A. (2009). Base de datos en MYSQL. México DF.
16. Silberschatz, A Korth , H y S. Sudarshan . (202). Fundamentos de Bases de datos. Madrid, España: McGraw-Hill Inc.
17. Sytems, C. (2010). CCNA Exploration 4.0. Capitulo 4, La red de datos.

18. Taylor, F. W. (1987). Principios de la Administración Científica. Buenos Aires Argentina Al ateneo.
  
19. (2010). Orientaciones Técnicas para la Implementación de Red de Datos y Conectividad a Internet Chile, Enlaces.