

ADMINISTRACIÓN ELECTRÓNICA DEL LABORATORIO DE REDES UTB

**GIOVANNI MARTÍNEZ LEONES
ARTHUR ALBERTO PUELLO ROJAS**

**MINOR EN DESARROLLO DE APLICACIONES EN .NET
PROGRAMA DE INGENIERÍA DE SISTEMAS
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
CARTAGENA DE INDIAS, D.T Y C.**

2012

ADMINISTRACIÓN ELECTRÓNICA DEL LABORATORIO DE REDES UTB

**GIOVANNI MARTÍNEZ LEONES
ARTHUR ALBERTO PUELLO ROJAS**

Trabajo de monografía para obtener el título de Ingeniero de Sistemas

Director:

Ing. Isaac Zúñiga Silgado

**MINOR EN DESARROLLO DE APLICACIONES EN .NET
PROGRAMA DE INGENIERÍA DE SISTEMAS
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR
CARTAGENA DE INDIAS, D.T Y C.**

2012

Nota de aceptación

Jurado

Cartagena, Julio de 2012.

Señores

COMITÉ CURRICULAR

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

Ciudad

Respetados señores:

Con todo el interés me dirijo a Uds. Para presentar a su consideración, estudio y aprobación la monografía titulada **ADMINISTRACIÓN ELECTRÓNICA DEL LABORATORIO DE REDES UTB**, como requisito para obtener el título de Ingeniero de Sistemas.

Atentamente,



GIOVANNI MARTÍNEZ LEONES

Cartagena, Julio de 2012.

Señores

COMITÉ CURRICULAR

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

Ciudad

Respetados señores:

Con todo el interés me dirijo a Uds. Para presentar a su consideración, estudio y aprobación la monografía titulada **ADMINISTRACIÓN ELECTRÓNICA DEL LABORATORIO DE REDES UTB**, como requisito para obtener el título de Ingeniero de Sistemas.

Atentamente,

Arthur Puello Rosas

ARTHUR ALBERTO PUELLO ROJAS

AUTORIZACIÓN

Yo, GIOVANNI MARTÍNEZ LEONES, identificado con la cedula de ciudadanía número 1.143.327.261 de Cartagena, autorizo a la Universidad Tecnológica de Bolívar, para hacer uso de mi trabajo de monografía y publicarlo en el catalogo on-line de la biblioteca.

Giovanni Martinez

GIOVANNI MARTÍNEZ LEONES

AUTORIZACIÓN

Yo, ARTHUR ALBERTO PUELLO ROJAS, identificado con la cedula de ciudadanía número 1.128.054.768 de Cartagena, autorizo a la Universidad Tecnológica de Bolívar, para hacer uso de mi trabajo de monografía y publicarlo en el catalogo on-line de la biblioteca.

Arthur Puello Rojas

ARTHUR ALBERTO PUELLO ROJAS

DESCRIPCIÓN DEL PROBLEMA

En busca de la constante mejora en los servicios que brinda la Universidad Tecnológica de Bolívar a sus estudiantes se desarrollara una aplicación para el control de los equipos en los diferentes laboratorios de redes y software con los que cuentan las instalaciones.

Hasta el momento estos procesos se realizan manualmente perdiendo mucho tiempo en la administración de los mismos y en la identificación de los recursos; es difícil tener certeza de la totalidad de equipos en un lugar específico, el estado de los mismos y dificultando la trazabilidad sobre cada equipo.

De esta forma se depende solo de la memoria de un par de personas y la documentación física que se recopile de los equipos, para no caer en procesos ineficientes se planea la implementación de una herramienta que permita la administración electrónica del laboratorio de redes de la UTB para así con una base tecnológica poder mejorar en cada uno de los procesos y servicios para los estudiantes.

Pensando a futuro se debe prever la ampliación de los servicios que se brindaran a la comunidad estudiantil y entre estos se encuentran los brindados en el laboratorio de redes, cada vez se manejaran más equipos de última tecnología que van ligados a un alto precio de mercado por lo cual es importante mantener estricto control sobre cada uno de estos, si se le suman más laboratorios se llegara al punto en que el proceso manual no dará abasto y ya será necesaria una herramienta capaz de controlar el inventario de equipos junto con el proceso de préstamos y reserva de los mismos para ponerlo a disposición de los estudiantes de la UTB.

Para cumplir con esta meta se planea la creación de una aplicación para el manejo de los equipos de forma más eficiente y de fácil acceso, en la cual los estudiantes puedan hacer reservas de los recursos durante un periodo de tiempo y así programar sus prácticas y laboratorios con las herramientas necesarias.

Mediante un sistema se mejoran los servicios y procesos internos de la UTB permitiendo la facilidad de los procesos con total confiabilidad permitiendo dejar que el sistema se encargue del proceso dejando tiempo para optimizar los recursos disponibles y facilitar la toma de decisiones.

Tabla de contenido

INTRODUCCIÓN	11
OBTENCION DE REQUERIMIENTOS	12
JUSTIFICACIÓN	13
OBJETIVOS	15
OBJETIVOS GENERALES	15
OBJETIVOS ESPECIFICOS.....	15
1. ESPECIFICACION DE REQUERIMIENTOS	16
1.1 REQUERIMIENTOS FUNCIONALES	16
1.1.1 Subsistema de Equipos.....	16
1.1.2 Subsistema de Recursos.....	17
1.1.3 Subsistema de Préstamos y Reservas	17
1.1.4 Subsistema de Laboratorios	18
1.1.5 Subsistema de Usuarios y Roles	19
1.2 REQUERIMIENTOS NO FUNCIONALES.....	19
1.3 ACTORES DEL SISTEMA.....	20
1.4 GLOSARIO DE TERMINOS	21
1.5 CASOS DE USO	21
1.5.1 Subsistema de equipos.....	21
1.5.2 Subsistema de Recursos.....	27
1.5.3 Subsistema de Préstamos y Reservas	32
1.5.4 Subsistema de Laboratorios	40
1.5.5 Subsistema de Usuarios y Roles	45
1.6 DIAGRAMA DE CASOS DE USO	48
2 ANALISIS DEL SISTEMA	49
2.1 DIAGRAMA DE CLASES	49
2.2 DIAGRAMAS DE SECUENCIA	51
3 DISEÑO	70
3.1 CONTROL DE ACCESO.....	70
3.2 CONTROL DE FLUJO.....	72

3.3 ALMACENAMIENTO PERSISTENTE.....	72
3.4 RELACIÓN HARDWARE/SOFTWARE	73
3.4.1 DIAGRAMA DE COMPONENTES	73
3.4.2 DIAGRAMA DE DESPLIEGUE	75
3.4.3 REQUISITOS MINIMOS DE HARDWARE	75
4 VALIDACIÓN DEL SISTEMA	77
CONCLUSIONES	80
RECOMENDACIONES	81
BIBLIOGRAFIA	83

INTRODUCCIÓN

La Universidad Tecnológica de Bolívar es una entidad interesada con respecto al tema de la sistematización de sus procesos, debido a que esto proporciona una buena infraestructura para estar a la vanguardia en cuanto a tecnología se refiere.

A raíz de esto, la Universidad cuenta con su propia página de Internet, la cual se encuentra dispuesta las 24 horas del día para que cualquier persona que esté interesada en estudiar alguna carrera en la Tecnológica de Bolívar, tenga acceso a la información que necesita, además de que se puede enterar de los últimos acontecimientos que están ocurriendo en la comunidad académica.

Por otro lado, la Universidad Tecnológica de Bolívar cuenta con su propia plataforma de aprendizaje virtual llamada SAVIO, la cual apoya tecnológicamente a las asignaturas de todas las carreras, con el objetivo de facilitar la interacción de los alumnos con sus docentes y ayudar en lo que a la realización de actividades se refiere.

Por último, se encuentra la plataforma para la gestión de los alumnos, docentes y asignaturas, la cual es SIRIUS, y se encarga de administrar todo lo que son los horarios, las matrículas, la información personal de cada estudiante, entre otros muchos datos.

Ahora, como apoyo a los estudiantes y en pro de ofrecer mejores servicios para mejorar sus procesos y tener un acercamiento a sus alumnos se piensa implementar una herramienta para la administración electrónica del laboratorio de redes con la ayuda de los estudiantes participantes del Minor de desarrollo en .NET.

OBTENCION DE REQUERIMIENTOS

Mediante entrevistas con el director de la oficina de adquisiciones y compras se logró obtener la forma en que se administran los equipos asignados a la dirección de sistemas, de la misma forma se llegó a conocer el proceso de préstamos de libros de la biblioteca de la UTB para aplicarlo de forma similar en la herramienta a desarrollar.

JUSTIFICACIÓN

El uso de sistemas informáticos permite optimizar y agilizar los procesos llevados a cabo para la realización de actividades, así como el manejo de la información que estos utilizan. El software dejó de ser un lujo de grandes empresas y se transformó en una necesidad para toda entidad que tenga vocación educacional o empresarial.

Con la herramienta a desarrollar se optimizan tiempos y se mantiene mayor control, cualquier tarea de gestión de inventario que se realice demorará menos que si se hace un proceso manual, por esta razón se implementará la aplicación basados en los conocimientos adquiridos en el Minor en desarrollo de aplicaciones en .NET para tratar el problema en cuestión.

Los estudiantes de pregrado de los programas de ingeniería de sistemas y tecnología en sistemas de la UTB serán los principales beneficiados de la implementación de la herramienta desarrollada ya que esta permitirá la gestión de los recursos del laboratorio de redes, facilitando de esta forma la realización de las diferentes prácticas y experiencias analizadas en clase para afianzar sus conocimientos.

Afianzar estos conocimientos permitirá a la UTB formar profesionales más competitivos en el campo de las redes y telemática, permitiendo ampliar el campo de acción a nivel empresarial de los egresados de la institución. Estas habilidades adquiridas a nivel práctico agregarán a los estudiantes un entendimiento más profundo de lo aprendido en clase y permitirá que se apropien de conocimientos más especializados sobre temas específicos.

Es importante tener en cuenta que la teoría es parte importante del conocimiento adquirido en la academia pero solo con la práctica se llegará a tener dominio sobre los campos teóricos ya que con ella afianzas el aprendizaje, como dice el refrán “La práctica hace al maestro”,

A nivel de manejo de inventarios la herramienta permite la trazabilidad y gestión de los elementos del laboratorio de redes, teniendo en cuenta un crecimiento a futuro. Con un estricto control sobre los equipos permitirá al administrador del laboratorio tomar decisiones rápidas y acertadas sobre los servicios que se brindan en el laboratorio de redes, la herramienta esta en capacidades de controlar múltiples laboratorios mientras se siga el mismo proceso, de esta forma se pueden habilitar más y mejores espacios para los estudiantes de la UTB.

Automatizando los diferentes procesos relacionados al laboratorio de redes como la reserva de recursos, el préstamo de los mismos, el control del inventario y el historial de movimiento sobre ellos se logrará construir una herramienta poderosa para los administradores en la que con pocos clics conocer el estado, cantidad y características de sus equipos y recursos disponibles.

Finalmente es importante mencionar que la herramienta es diseñada con características que le permitirán ser extensible en un futuro para poder ofrecer nuevos y mejores servicios permitiendo así el poder brindar más y mejor servicios al cuerpo estudiantil de la UTB que finalmente serán los grandes beneficiados de la implementación de la aplicación.

OBJETIVOS

OBJETIVOS GENERALES

- Desarrollar una herramienta de software que permita mejorar u optimizar el proceso de control de recursos en el laboratorio de la UTB.

OBJETIVOS ESPECIFICOS

- Identificar las características de la herramienta mediante el análisis del proceso actual.
- Diseñar la herramienta con base en el resultado del análisis.
- Diseñar una arquitectura base de la herramienta que promueva la flexibilidad, escalabilidad y control de acceso.
- Validar la herramienta mediante pruebas funcionales.

1. ESPECIFICACION DE REQUERIMIENTOS

Los requerimientos del sistema son las principales funciones y características que debe satisfacer el sistema, para ello creamos un formato de tablas donde colocamos cada requerimiento a nivel de comprensión del usuario, el cual es inexperto en el tema de ingeniería de software y requiere que las especificación de los requerimientos del sistema se realicen igualmente en un lenguaje que pueda entender. Los siguientes requerimientos del sistema, están organizados de acuerdo al sistema al que pertenecen de la siguiente forma: 102, 1 = primer número, 02 = segundo número.

Primer número: Subsistema al que pertenece.

Segundo número: Posición del requerimiento en la tabla.

1.1 REQUERIMIENTOS FUNCIONALES

Son aquellos que especifican una funcionalidad del sistema Deben ser verificables y medibles, para poder garantizar la evaluación de estos cuando se implemente el sistema.

1.1.1 Subsistema de Equipos

Nº	Descripción	Prioridad	Actor
101	Que se puedan agregar equipos definiendo cada una de sus características.	Esencial	Administrador
102	Que se puedan consultar todos los equipos disponibles, con sus características (nombre, descripción).	Esencial	Todos
103	Que se puedan modificar las características y/o propiedades de los equipos.	Esencial	Administrador

104	Que se pueda deshabilitar equipos cuando sea necesario y así lo determine el Administrador del sistema.	Deseable	Administrador
105	Que el sistema permita exportar la relación de equipos a un archivo.	Deseable	Profesor

1.1.2 Subsistema de Recursos

Nº	Descripción	Prioridad	Actor
201	El sistema debe permitir agregar recursos para ponerlos a disposición de los interesados	Esencial	Administrador
202	Que se puedan modificar las existencias de los diferentes recursos disponibles	Esencial	Administrador
203	El sistema debe permitir consultar los diferentes recursos disponibles para los interesados	Esencial	Todos
204	Que se pueda exportar la lista de recursos con los cuales cuenta el sistema a un archivo	Deseable	Profesor

1.1.3 Subsistema de Préstamos y Reservas

Nº	Descripción	Prioridad	Actor
301	El sistema debe permitir la reserva de los recursos requeridos por los estudiantes.	Esencial	Estudiantes / Administrador
302	Que se pueda hacer cancelación de reservas	Esencial	Estudiantes /

	hechas por los estudiantes.		Administrador
303	Que el administrador pueda anular las reservas de un recurso específico en caso de daños o fallas en el mismo	Deseable	Administrador
304	Que el administrador pueda cambiar el estado en que se encuentran los recursos (Reservado, prestado, en mora, entre otros)	Esencial	Administrador
305	Que se pueda exportar el estado de las reservas a un archivo	Deseable	Profesor
306	El sistema debe controlar que los estudiantes solo reserven recursos disponibles y mantener control sobre estos	Esencial	Sistema
307	Que el estudiante pueda ampliar fácilmente la reserva de un equipo si este está disponible	Deseable	Estudiantes / Administrador
308	Que permita consultar el histórico de los préstamos al administrador	Deseable	Administrador

1.1.4 Subsistema de Laboratorios

Nº	Descripción	Prioridad	Actor
401	El sistema debe permitir agregar laboratorios	Esencial	Administrador
402	Que se puedan modificar los parámetros de los laboratorios existentes	Esencial	Administrador
403	El sistema debe permitir consultar los diferentes recursos disponibles en los diferentes laboratorios	Deseable	Todos

404	Que el sistema permita exportar a archivo los recursos por cada uno de los laboratorios	Deseable	Profesor
-----	---	----------	----------

1.1.5 Subsistema de Usuarios y Roles

Nº	Descripción	Prioridad	Actor
501	El sistema debe permitir la creación de usuarios solamente por parte del administrador	Esencial	Administrador
502	El sistema debe tener el control del acceso de los usuarios con su rol específico	Esencial	Sistema
503	Que el sistema pueda recuperar la contraseña de un usuario con su correo	Deseable	Todos

1.2 REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales son aquellos que hacen parte de un proyecto de software, pero estos no le agregan funcionalidad al software final, sino que buscan asegurar el debido funcionamiento del mismo. Los requerimientos no funcionales que debe tener el software que se va a desarrollar, son los siguientes:

Nº	Descripción	Prioridad
1	La carga de la aplicación, no debe demorar más de 4 segundos.	Esencial
2	El servidor debe soportar como mínimo la conexión de 100 usuarios simultáneamente.	Deseable

3	Debe haber un acceso limitado al sistema para todos los usuarios, de acuerdo a su rol, excepto para el administrador.	Esencial
4	La interfaz gráfica debe permitir que un usuario en específico llegué a la parte que desea navegando máximo por 5 páginas previamente (haciendo máximo 5 clic).	Esencial
5	Que no ocurran fallos en el sistema, y si ocurren, que al menos en 3 meses no se presente un fallo.	Deseable
6	Que se le pueda agregar funcionalidad al sistema de una manera sencilla, a medida que surjan nuevas necesidades o requerimientos.	Esencial
7	El sistema debe estar implementado en tecnologías .NET	Esencial

1.3 ACTORES DEL SISTEMA

Son todas las personas que interactúan y tienen contacto con el sistema, también se pueden denominar como roles de usuarios, y son los siguientes:

N°	Descripción	Nombre
01	Es aquel que administra todo el sistema. Tiene acceso y dominio de todas las partes del sistema	Administrador
02	Son los profesores encargados de los laboratorios y que pueden consultar el estado de los recursos y/o equipos	Profesor
03	Son todos los alumnos que se encuentren activos financieramente. Pueden realizar uso de los recursos mediante los préstamos y reservas	Estudiantes

1.4 GLOSARIO DE TERMINOS

Recursos: Son los diferentes elementos disponibles que pueden ser reservados y utilizados por los estudiantes.

Laboratorios: Son las diferentes aulas o salones que cuentan con equipos disponibles para los estudiantes.

Estudiante: Son todas aquellas personas que están inscritas en pregrado en el período actual en la Universidad Tecnológica de Bolívar en los programas de Ingeniería de Sistemas o Tecnología en Sistemas.

Profesor: Son los responsables de los laboratorios por parte del cuerpo de maestros de la Universidad Tecnológica de Bolívar.

Administrador: Son aquellos que tienen acceso total al sistema, como lo dice su nombre son los que administran este junto con sus servicios.

Formularios: Objetos frontera que capturan o muestran información a algún usuario específico.

1.5 CASOS DE USO

Los siguientes son los casos de usos del sistema, y representan un esquema de cómo se realizarían los procesos que llevarían al cumplimiento de los requerimientos del sistema. También están divididos en subsistemas como el apartado de Requerimientos del sistema.

1.5.1 Subsistema de equipos.

Identificación	101
Nombre	Agregar equipos
Descripción	En este proceso se agregarán equipos para que el sistema pueda asignarlos como recursos.
Actor principal	Administrador
Actor secundario	Estudiante, Profesor

Precondiciones	<ul style="list-style-type: none"> • Que el administrador haya iniciado sesión • Que el administrador ingrese en la opción de agregar equipos
Pos condiciones	<ul style="list-style-type: none"> • Se agregará un nuevo equipo a la aplicación
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la interfaz para la creación de equipos. 2. El administrador diligencia la información del nuevo equipo. 3. El administrador hace clic sobre el botón de guardar. 4. El sistema verifica que la información digitada por el administrador este correcta 5. El sistema agrega el nuevo equipo
Flujos alternativos	<ol style="list-style-type: none"> 4. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los mensajes de error correspondientes. • Vuelve al paso 1.
Prioridad	Esencial
Objetos participantes	Control: EquiposController; Entidad: Equipos; Frontera: Equipos/Create

Identificación	102
Nombre	Consultar equipos

Descripción	En este proceso se realizan consultas de las características de los equipos
Actor principal	Administrador
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan equipos registrados • Que el usuario ingrese en la página de administración de equipos
Pos condiciones	<ul style="list-style-type: none"> • Que se muestre la información de los equipos
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los equipos. 2. El sistema lista todos los equipos mostrando su información básica. 3. El usuario selecciona el equipo del que desea ver más detalles 4. El sistema ingresa a la sección de detalles del equipo seleccionado 5. El sistema permite devolverse para ver detalles de otros equipos si se desea <p>Los pasos 1, 2, 3 y 4, se repiten hasta que el usuario se salga del proceso.</p>
Flujos alternativos	Ninguno
Prioridad	Esencial
Objetos participantes	Control: EquiposController; Entidad: Equipos; Frontera: Equipos/Index, Equipos/Details

Identificación	103
Nombre	Edición de equipos
Descripción	En este proceso se realiza la edición de las características de los equipos disponibles.
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que existan equipos registradas • Que el usuario ingrese en la página de administración de equipos
Pos condiciones	<ul style="list-style-type: none"> • Que los datos de los equipos queden actualizados
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los equipos. 2. El sistema muestra la lista generada en el paso anterior. 3. El usuario selecciona el equipo al que desea editarle la información. 4. El sistema carga la información del equipo que el usuario seleccionó en el paso anterior. 5. El usuario modifica la información que desea actualizar. 6. El usuario confirma la modificación. 7. El sistema valida la información. 8. El sistema guarda la actualización.

Flujos alternativos	7. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los errores. • Regresar al paso 4
Prioridad	Esencial
Objetos participantes	Control: EquiposController; Entidad: Equipos; Frontera: Equipos/Index, Equipos/Edit

Identificación	104
Nombre	Deshabilitar equipos
Descripción	En este proceso se deshabilitan equipos según el criterio del administrador.
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que existan equipos registrados • Que el usuario ingrese en la página de administración de equipos
Pos condiciones	<ul style="list-style-type: none"> • Que los datos de los equipos queden deshabilitados
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los equipos. 2. El sistema muestra la lista generada en el paso anterior.

	<ol style="list-style-type: none"> 3. El usuario selecciona el equipo que desea deshabilitar 4. El sistema carga la información del equipo que el usuario seleccionó en el paso anterior. 5. El usuario confirma deshabilitar el equipo. 6. El sistema guarda la actualización.
Flujos alternativos	<ol style="list-style-type: none"> 6. Si el equipo tiene reservas: <ul style="list-style-type: none"> • Cancela los préstamos sobre el equipo. • Regresar al panel de administración
Prioridad	Deseable
Objetos participantes	Control: EquiposController; Entidad: Equipos; Frontera: Equipos/Index, Equipos/Delete

Identificación	105
Nombre	Exportar equipos
Descripción	En este proceso se realiza la exportación de los equipos a un archivo
Actor principal	Profesor
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan equipos registrados • Que el usuario ingrese en la página de exportar equipos

Pos condiciones	<ul style="list-style-type: none"> • Que se exporte la información de los equipos
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página para exportar 2. El sistema muestra la opción de exportar 3. El usuario hace clic sobre la opción 4. El sistema recopila la información 5. El sistema crea el archivo 6. El sistema pregunta donde desea guardar 7. El usuario confirma la ubicación
Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: EquiposController; Entidad: Equipos; Frontera: Equipos/Export

1.5.2 Subsistema de Recursos

Identificación	201
Nombre	Agregar recursos
Descripción	En este proceso se agregarán recursos para que los estudiantes puedan hacer uso de ellos.
Actor principal	Administrador

Actor secundario	Estudiante, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que el administrador haya iniciado sesión • Que el administrador ingrese en la opción de agregar recursos
Pos condiciones	<ul style="list-style-type: none"> • Se agregará un nuevo recurso a la aplicación
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la interfaz para la creación recursos. 2. El administrador diligencia la información del nuevo recurso. 3. El administrador hace clic sobre el botón de guardar. 4. El sistema verifica que la información digitada por el administrador este correcta 5. El sistema agrega el nuevo recurso
Flujos alternativos	<ol style="list-style-type: none"> 4. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los mensajes de error correspondientes. • Vuelve al paso 1.
Prioridad	Esencial
Objetos participantes	Control: RecursosController; Entidad: Recursos; Frontera: Recursos/Create

Identificación	202
-----------------------	-----

Nombre	Edición de recursos
Descripción	En este proceso se realizan la edición de las características de los recursos disponibles.
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que existan recursos registrados • Que el usuario ingrese en la página de administración de recursos
Pos condiciones	<ul style="list-style-type: none"> • Que los datos de los recursos queden actualizados
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los recursos. 2. El sistema muestra la lista generada en el paso anterior. 3. El usuario selecciona el recurso al que desea editarle la información. 4. El sistema carga la información del recurso que el usuario seleccionó en el paso anterior. 5. El usuario modifica la información que desea actualizar. 6. El usuario confirma la modificación. 7. El sistema valida la información. 8. El sistema guarda la actualización.
Flujos alternativos	<ol style="list-style-type: none"> 7. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los errores. • Regresar al paso 4

Prioridad	Esencial
Objetos participantes	Control: RecursosController; Entidad: Recursos; Frontera: Recursos/Index, Recursos/Edit

Identificación	203
Nombre	Consultarrecursos
Descripción	En este proceso se realizan consultas de las características de los recursos
Actor principal	Administrador
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan recursos registrados • Que el usuario ingrese en la página de administración de recursos
Pos condiciones	<ul style="list-style-type: none"> • Que se muestre la información de los recursos
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los recursos. 2. El sistema lista todos los recursos mostrando su información básica. 3. El usuario selecciona el recurso del que desea ver más detalles 4. El sistema ingresa a la sección de detalles del recurso seleccionado 5. El sistema permite devolverse para ver detalles de otros

	<p>recursos si se desea</p> <p>Los pasos 1, 2, 3 y 4, se repiten hasta que el usuario se salga del proceso.</p>
Flujos alternativos	Ninguno
Prioridad	Esencial
Objetos participantes	Control: RecursosController; Entidad: Recursos; Frontera: Recursos/Index, Recursos/Details

Identificación	204
Nombre	Exportar recursos
Descripción	En este proceso se realiza la exportación de los recursos a un archivo
Actor principal	Profesor
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan recursos registrados • Que el usuario ingrese en la página de exportar recursos
Pos condiciones	<ul style="list-style-type: none"> • Que se exporte la información de los recursos
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página para exportar 2. El sistema muestra la opción de exportar 3. El usuario hace clic sobre la opción

	<p>4. El sistema recopila la información</p> <p>5. El sistema crea el archivo</p> <p>6. El sistema pregunta donde desea guardar</p> <p>7. El usuario confirma la ubicación</p>
Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: RecursosController; Entidad: Recursos; Frontera: Recursos/Export

1.5.3 Subsistema de Préstamos y Reservas

Identificación	301
Nombre	Reservar recursos
Descripción	En este proceso se reservan recursos por parte de estudiantes o el administrador
Actor principal	Estudiante
Actor secundario	Administrador
Precondiciones	<ul style="list-style-type: none"> • Que el usuario haya iniciado sesión • Que el usuario ingrese en la opción de reservar recursos
Pos condiciones	<ul style="list-style-type: none"> • Se reservara un nuevo recurso en la aplicación

Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la vista de reservas 2. El sistema solicita los datos requeridos 3. El usuario diligencia los datos 4. El usuario hace clic en reservar 5. El sistema verifica la información 6. El sistema reserva el recurso
Flujos alternativos	<ol style="list-style-type: none"> 5. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los mensajes de error correspondientes. • Vuelve al paso 1.
Prioridad	Esencial
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Prestamos/Reservar

Identificación	302
Nombre	Cancelar reservas
Descripción	En este proceso se cancelan reservas realizadas por un usuario
Actor principal	Estudiante
Actor secundario	Administrador

Precondiciones	<ul style="list-style-type: none"> • Que existan reservas registradas • Que el usuario ingrese en la página de administración de reservas
Pos condiciones	<ul style="list-style-type: none"> • Que se cancele una reserva de un recurso
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de reservas. 2. El sistema lista todas las reservas mostrando su información básica. 3. El usuario da clic en cancelar reserva sobre la deseada 4. El sistema cancela la reserva
Flujos alternativos	Ninguno
Prioridad	Esencial
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Presetamos/Index

Identificación	303
Nombre	Anulación de reservas
Descripción	En este proceso se anula un grupo de reservas a cambiar el estado de un recurso.
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor

Precondiciones	<ul style="list-style-type: none"> • Que existan recursos registradas • Que el usuario ingrese en la página de administración de recursos
Pos condiciones	<ul style="list-style-type: none"> • Que las reservas sobre un recurso queden anuladas
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los recursos. 2. El sistema muestra la lista generada en el paso anterior. 3. El usuario selecciona el recurso al que necesita cambiar el estado 4. El sistema carga la información del recurso que el usuario seleccionó en el paso anterior. 5. El usuario modifica el estado del recurso. 6. El usuario confirma la modificación. 7. El sistema valida la información. 8. El sistema guarda el estado del equipo. 9. El sistema anula las reservas sobre ese recurso
Flujos alternativos	<ol style="list-style-type: none"> 7. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los errores. • Regresar al paso 4 9. Si no puede eliminar una reserva específica: <ul style="list-style-type: none"> • Muestra los errores. • Regresar al paso 4
Prioridad	Deseable
Objetos	Control: RecursosController; Entidad: Prestamos,

participantes	Recursos; Frontera: Recursos/Edit
----------------------	-----------------------------------

Identificación	304
Nombre	Confirmar reserva
Descripción	En este el administrador confirma una reserva
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que existan reservas registradas • Que el usuario ingrese en la página de administración de reservas
Pos condiciones	<ul style="list-style-type: none"> • Que las reservas se hagan efectivas
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de reservas. 2. El sistema lista todas las reservas mostrando su información básica. 3. El usuario da clic en editar reserva sobre la deseada 4. El sistema carga los detalles de la reserva 5. El usuario cambia el estado de la reserva 6. El usuario hace clic en guardar 7. El sistema confirma la reserva
Flujos alternativos	Ninguno

Prioridad	Esencial
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Prestamos/Index, Prestamos/Edit

Identificación	305
Nombre	Exportar reservas
Descripción	En este proceso se realiza la exportación de las reservas a un archivo
Actor principal	Profesor
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan reservas registradas • Que el usuario ingrese en la página de exportar reservas
Pos condiciones	<ul style="list-style-type: none"> • Que se exporte la información de las reservas
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página para exportar 2. El sistema muestra la opción de exportar 3. El usuario hace clic sobre la opción 4. El sistema recopila la información 5. El sistema crea el archivo 6. El sistema pregunta donde desea guardar 7. El usuario confirma la ubicación

Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Prestamos/Export

Identificación	307
Nombre	Ampliar reserva
Descripción	En este proceso los usuarios amplían las reservas sobre un recurso
Actor principal	Estudiante
Actor secundario	Administrador
Precondiciones	<ul style="list-style-type: none"> • Que existan reservas registrados • Que el usuario ingrese en la página de reservas
Pos condiciones	<ul style="list-style-type: none"> • Se amplía la reserva de un recurso
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página de reservas. 2. El usuario hace clic sobre ampliar reserva 3. El sistema pide verificación del usuario 4. El sistema amplía la reserva un intervalo de tiempo específico

Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Prestamos/Index

Identificación	308
Nombre	Consultar prestamos
Descripción	En este proceso el administrador puede consultar el histórico de los prestamos
Actor principal	Administrador
Actor secundario	Ninguno
Precondiciones	<ul style="list-style-type: none"> • Que existan prestamos registrados • Que el usuario ingrese en la página de prestamos
Pos condiciones	<ul style="list-style-type: none"> • Se obtiene información del histórico de prestamos
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los préstamos completo. 2. El sistema lista todos los prestamos mostrando su información básica. 3. El usuario selecciona el préstamo del que desea ver más detalles 4. El sistema ingresa a la sección de detalles del

	<p>préstamos seleccionados</p> <p>5. El sistema permite devolverse para ver detalles de otros préstamos si se desea</p> <p>6. Los pasos 1, 2, 3 y 4, se repiten hasta que el usuario se salga del proceso.</p>
Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: PrestamosController; Entidad: Prestamos; Frontera: Prestamos/Index

1.5.4 Subsistema de Laboratorios

Identificación	401
Nombre	Agregar laboratorios
Descripción	En este proceso se agregarán laboratorios para poderles asignar recursos
Actor principal	Administrador
Actor secundario	Estudiante, Profesor
Precondiciones	<ul style="list-style-type: none"> • Que el administrador haya iniciado sesión • Que el administrador ingrese en la opción de agregar laboratorios
Pos condiciones	<ul style="list-style-type: none"> • Se agregará un nuevo laboratorio a la aplicación

Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la interfaz para la creación laboratorios. 2. El administrador diligencia la información del nuevo laboratorio. 3. El administrador hace clic sobre el botón de guardar. 4. El sistema verifica que la información digitada por el administrador este correcta 5. El sistema agrega el nuevo laboratorio
Flujos alternativos	<ol style="list-style-type: none"> 4. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los mensajes de error correspondientes. • Vuelve al paso 1.
Prioridad	Esencial
Objetos participantes	Control: LaboratoriosController; Entidad: Laboratorios; Frontera: Laboratorios/Create

Identificación	402
Nombre	Edición de laboratorios
Descripción	En este proceso se realizan la edición de las características de los laboratorios disponibles.
Actor principal	Administrador
Actor secundario	Estudiantes, Profesor

Precondiciones	<ul style="list-style-type: none"> • Que existan laboratorios registrados • Que el usuario ingrese en la página de administración de laboratorios
Pos condiciones	<ul style="list-style-type: none"> • Que los datos de los laboratorios queden actualizados
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los laboratorios. 2. El sistema muestra la lista generada en el paso anterior. 3. El usuario selecciona el laboratorio al que desea editarle la información. 4. El sistema carga la información del laboratorio que el usuario seleccionó en el paso anterior. 5. El usuario modifica la información que desea actualizar. 6. El usuario confirma la modificación. 7. El sistema valida la información. 8. El sistema guarda la actualización.
Flujos alternativos	<ol style="list-style-type: none"> 7. Si la información no es correcta: <ul style="list-style-type: none"> • Muestra los errores. • Regresar al paso 4
Prioridad	Esencial
Objetos participantes	Control: LaboratoriosController; Entidad: Laboratorios; Frontera: Laboratorios/Index, Laboratorios/Edit

Identificación	403
-----------------------	-----

Nombre	Consultarlaboratorios
Descripción	En este proceso se realizan consultas de las características de los laboratorios
Actor principal	Administrador
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan laboratorios registrados • Que el usuario ingrese en la página de administración de laboratorios
Pos condiciones	<ul style="list-style-type: none"> • Que se muestre la información de los laboratorios
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga el listado de los laboratorios. 2. El sistema lista todos los laboratorios mostrando su información básica. 3. El usuario selecciona el laboratorio del que desea ver más detalles 4. El sistema ingresa a la sección de detalles del laboratorio seleccionado 5. El sistema permite devolverse para ver detalles de otros laboratorios si se desea <p>Los pasos 1, 2, 3 y 4, se repiten hasta que el usuario se salga del proceso.</p>
Flujos alternativos	Ninguno
Prioridad	Esencial
Objetos	Control: LaboratoriosController; Entidad: Laboratorios;

participantes	Frontera: Laboratorios/Index, Laboratorios/Details
----------------------	--

Identificación	404
Nombre	Exportar laboratorios
Descripción	En este proceso se realiza la exportación de los laboratorios a un archivo
Actor principal	Profesor
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan laboratorios registrados • Que el usuario ingrese en la página de exportar laboratorios
Pos condiciones	<ul style="list-style-type: none"> • Que se exporte la información de los laboratorios
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página para exportar 2. El sistema muestra la opción de exportar 3. El usuario hace clic sobre la opción 4. El sistema recopila la información 5. El sistema crea el archivo 6. El sistema pregunta donde desea guardar 7. El usuario confirma la ubicación

Flujos alternativos	Ninguno
Prioridad	Deseable
Objetos participantes	Control: LaboratoriosController; Entidad: Laboratorios; Frontera: Laboratorios/Export

1.5.5 Subsistema de Usuarios y Roles

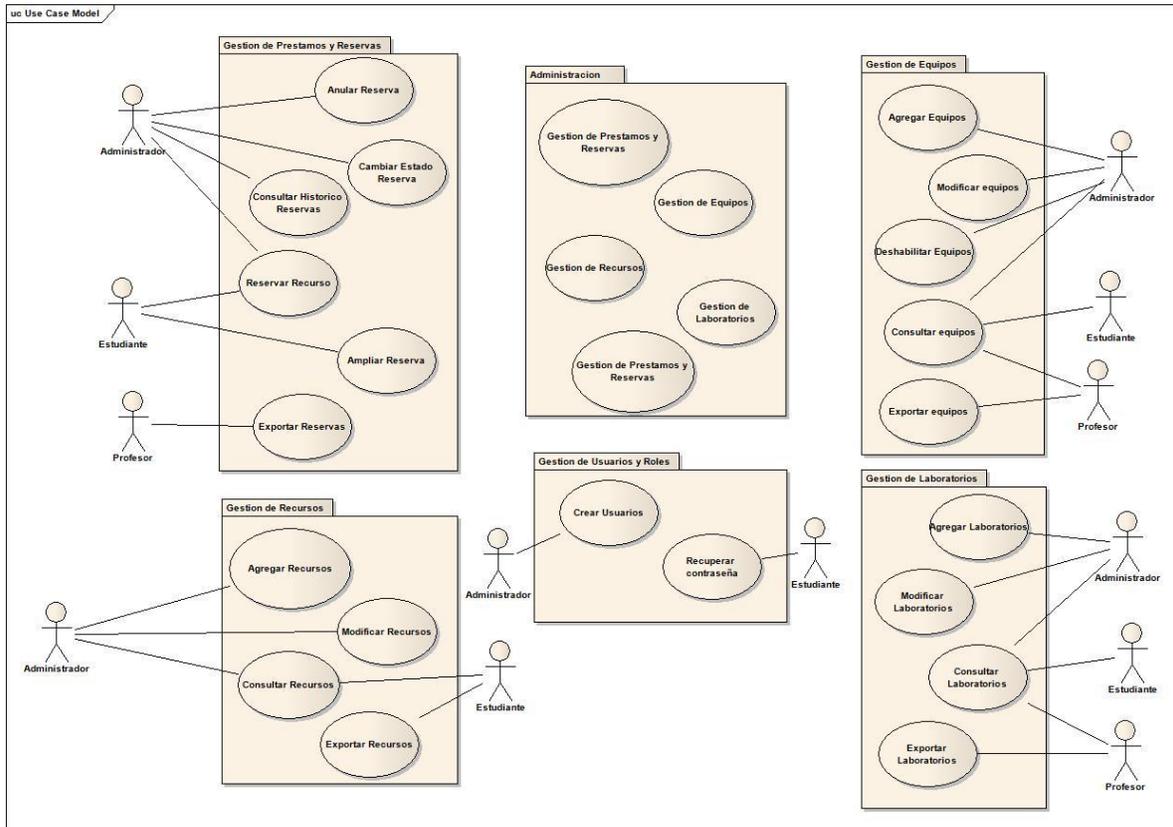
Identificación	501
Nombre	Agregar usuarios
Descripción	En este proceso se agregarán nuevos usuarios a la aplicación, solamente lo puede hacer el administrador del sistema
Actor principal	Administrador
Actor secundario	Estudiante
Precondiciones	<ul style="list-style-type: none"> • Que el administrador haya iniciado sesión • Que el administrador ingrese en la opción de agregar usuarios
Pos condiciones	<ul style="list-style-type: none"> • Se agregará un nuevo usuario a la aplicación
Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la interfaz para la creación usuarios. 2. El administrador diligencia la información del nuevo usuario. 3. El administrador hace clic sobre el botón de guardar.

	<p>4. El sistema verifica que la información digitada por el administrador este correcta</p> <p>5. El sistema agrega el nuevo usuario</p>
Flujos alternativos	<p>5. Si la información no es correcta:</p> <ul style="list-style-type: none"> • Muestra los mensajes de error correspondientes. • Vuelve al paso 1.
Prioridad	Esencial
Objetos participantes	Control: UsuariosController; Entidad: Usuarios; Frontera: Usuarios/Create

Identificación	503
Nombre	Recuperar contraseña
Descripción	En este proceso los usuarios recuperaran su contraseña en caso de olvido
Actor principal	Estudiante
Actor secundario	(ninguno)
Precondiciones	<ul style="list-style-type: none"> • Que existan estudiantes registrados • Que el usuario ingrese en la página de recuperar contraseña
Pos condiciones	<ul style="list-style-type: none"> • Se envía un mensaje al estudiante con su contraseña

Flujo de eventos	<ol style="list-style-type: none"> 1. El sistema carga la página de recuperar contraseña. 2. El sistema solicita el código del estudiante 3. El usuario ingresa su código 4. El usuario hace clic en “Recuperar” 5. El verifica que el código sea valido 6. El sistema envía un correo con la contraseña a la cuenta de correo asignada al estudiante en la base de datos
Flujos alternativos	<ol style="list-style-type: none"> 1. Si el código no es válido: <ul style="list-style-type: none"> • Muestre mensaje de error • Se devuelve al paso 1
Prioridad	Deseable
Objetos participantes	Control: AccountController; Entidad: Account; Frontera: Account/Recovery

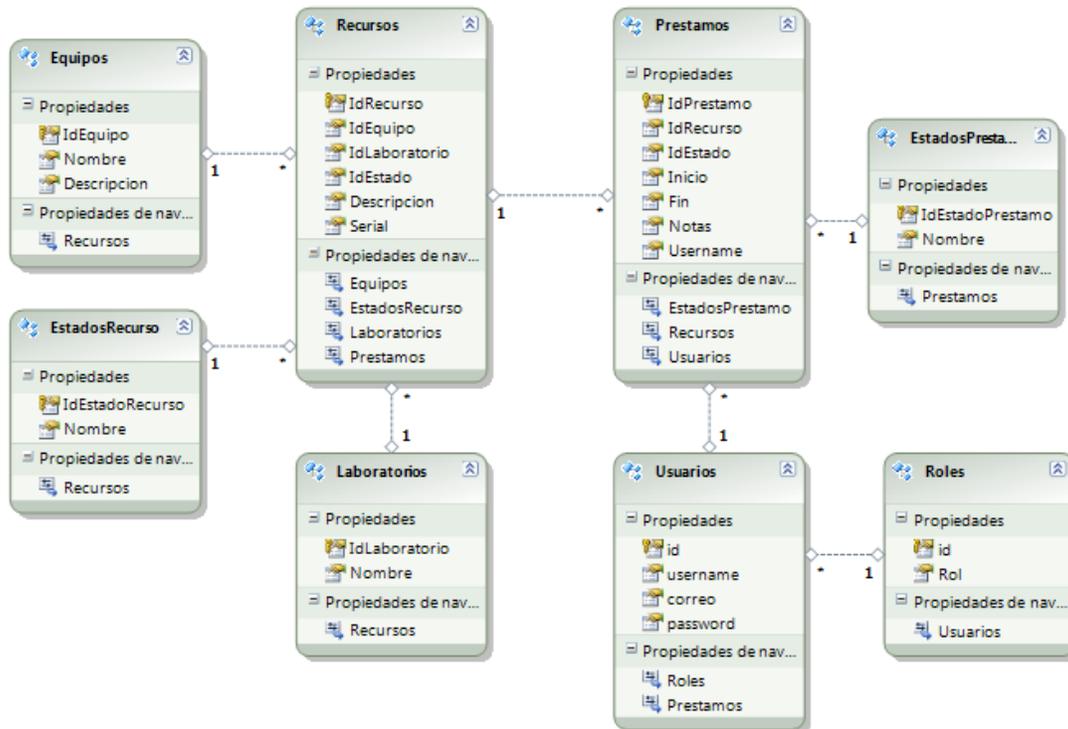
1.6 DIAGRAMA DE CASOS DE USO



2 ANALISIS DEL SISTEMA

En esta parte se determinaran los objetivos y límites del sistema a implementar haciendo énfasis en su estructura y funcionamiento. Se establecerán normas que permitan alcanzar los objetivos propuestos. También se creará el diseño conceptual de la información que se manejará, los componentes que se encargaran de su funcionamiento y las relaciones que existen entre uno y otro. Se mostrará una vista estática del sistema por medio de un diagrama de clases y una vista de interacción usando diagramas de secuencia, detallando en su totalidad la comunicación entre los distintos objetos en tiempo de ejecución dando una idea de cómo será el funcionamiento del sistema.

2.1 DIAGRAMA DE CLASES



Equipos: Son la generalización de cada uno de los artículos, son la categoría mayor de los recursos, por ejemplo “Switch HP V1910 24G”, solamente puede haber un equipo de este tipo en la aplicación.

Recursos: Son las diferentes existencias que tiene cada equipo, se caracterizan por estar en un lugar específico, con un serial único y puede haber más de uno del mismo equipo, por ejemplo “El Switch HP V1910 24G que se encuentra en el laboratorio A1 405 con serial: 2345678” y el “El Switch HP V1910 24G que se encuentra en el laboratorio A1 405 con serial: 3456789”, estos muestran 2 recursos diferentes del mismo equipo.

EstadosRecurso: Son los diferentes estados que pueden tener los recursos dentro de la aplicación (Disponible, Deshabilitado, En reparación).

Laboratorios: Son los diferentes salones o aulas en los cuales hay existencias de equipos, pueden ser gestionados en la consola del administrador.

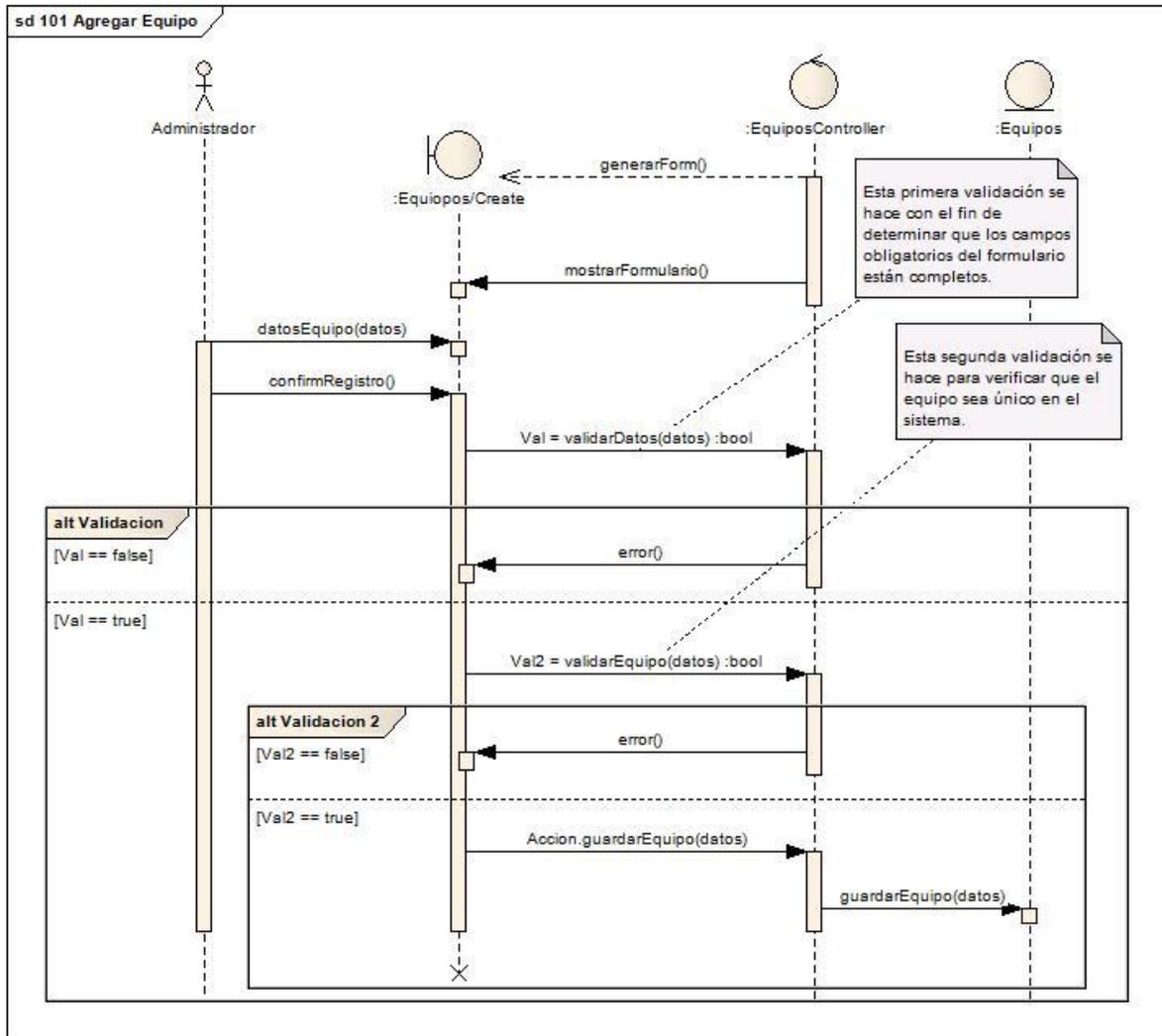
Prestamos: Son los movimientos que se realizan sobre los equipos, en estos se tiene seguimiento de los movimientos de los recursos y el uso de los estudiantes sobre los mismos, están conformados por un recurso, un estado del recurso, un usuario, una fecha y hora de inicio y una fecha y hora final, adicionalmente se pueden agregar notas si así se desea.

EstadosPrestamos: Son los diferentes estados que pueden tener los prestamos dentro de la aplicación (Reservado, Anulado, Cancelado, Prestado, Devuelto).

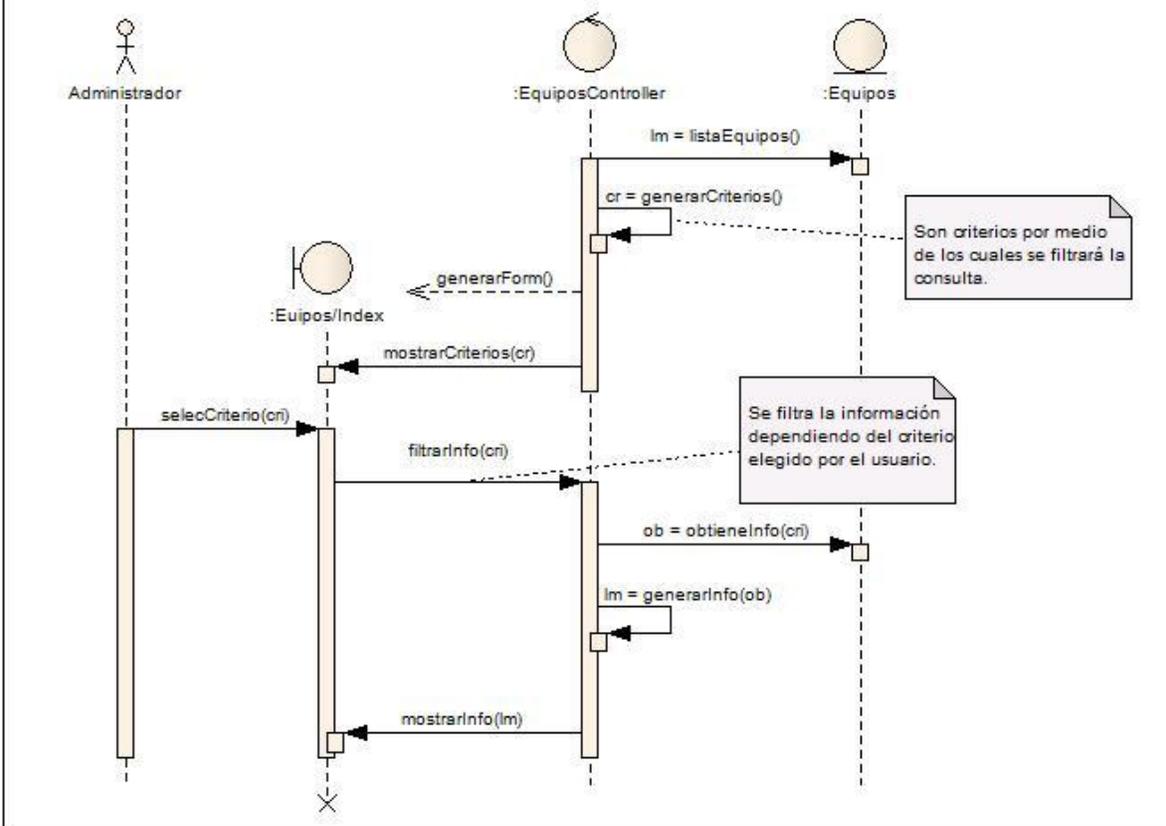
Usuarios: Son los diferentes usuarios que tendrá la aplicación, en realidad no hace parte de las clases del sistema ya que se están manejando en la herramienta de administración de sitios web de ASP.NET y por lo tanto esta desligada de la arquitectura, de esta clase solo se usa el Username.

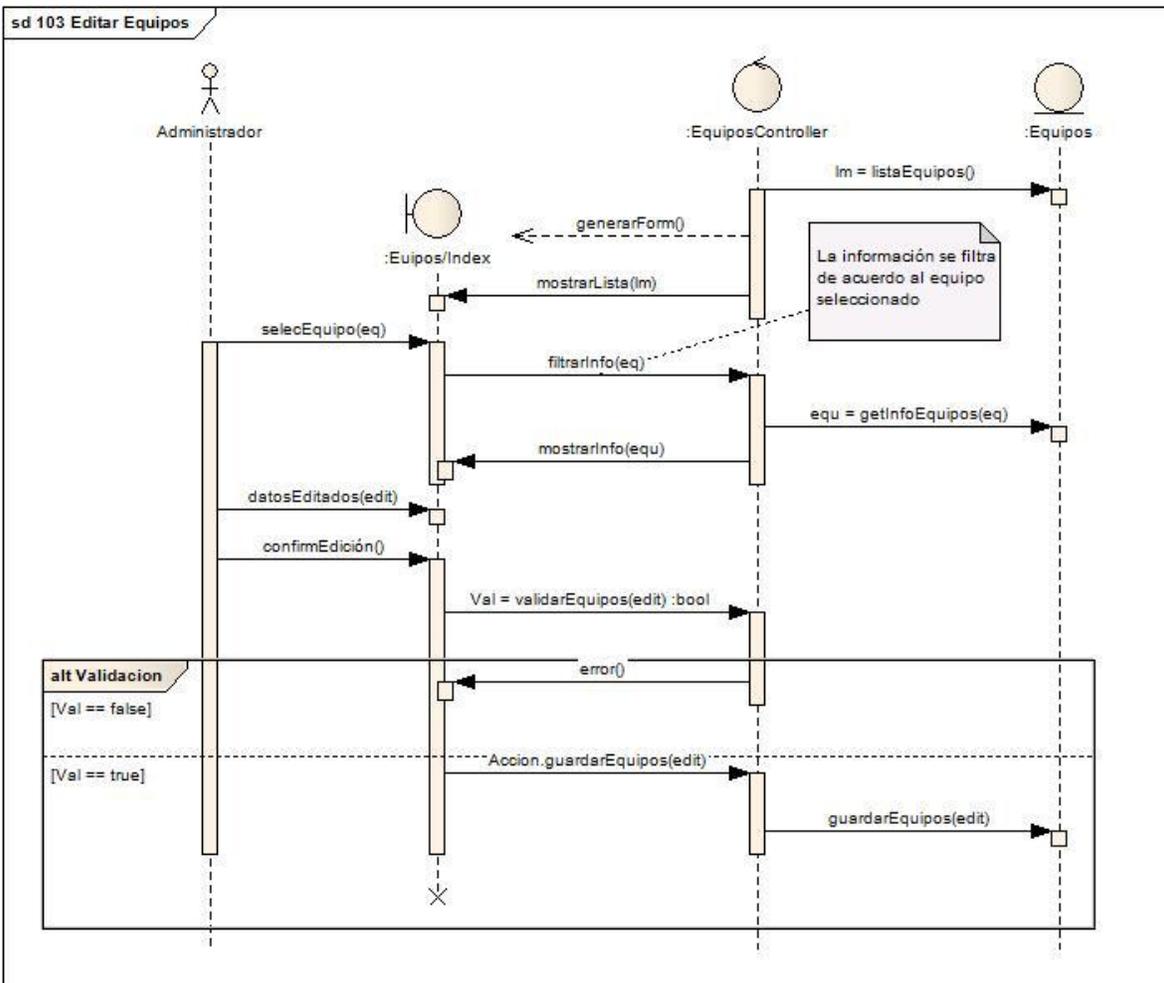
Roles: Los roles describen que tipo de usuario ingresa a la aplicación, al igual que los usuarios no hace parte de las clases del sistema ya que se está manejando en la herramienta de administración de sitios web de ASP.NET y por lo tanto esta desligada de la arquitectura.

2.2 DIAGRAMAS DE SECUENCIA

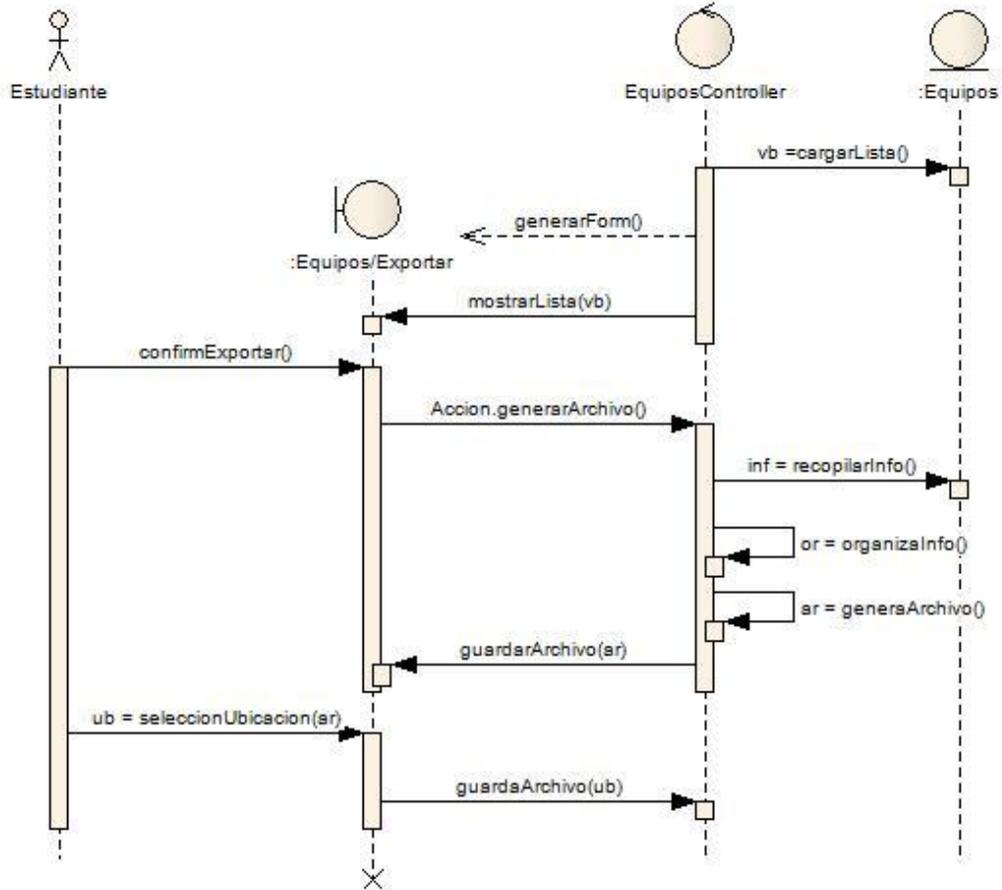


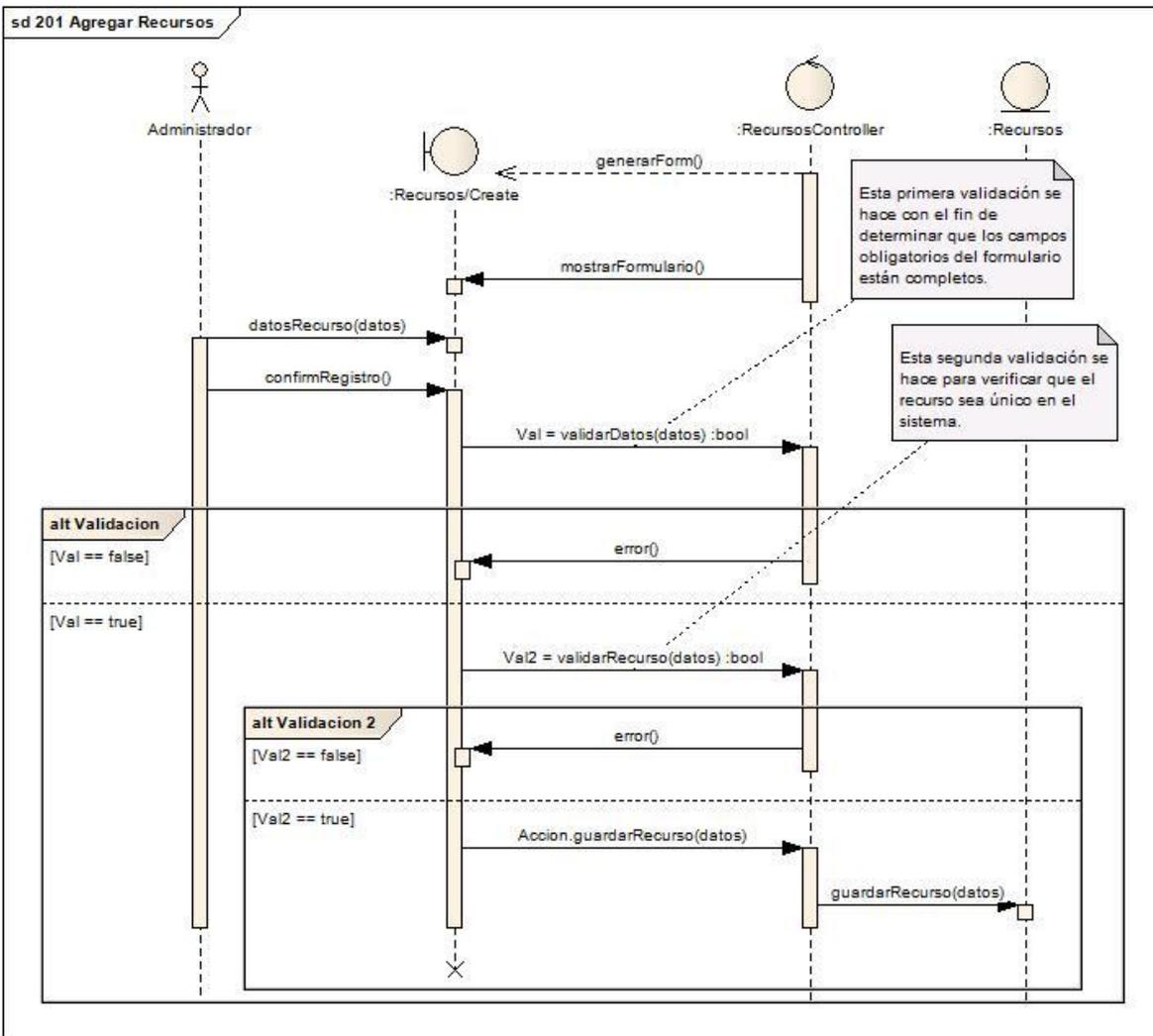
sd 102 Consultar Equipos

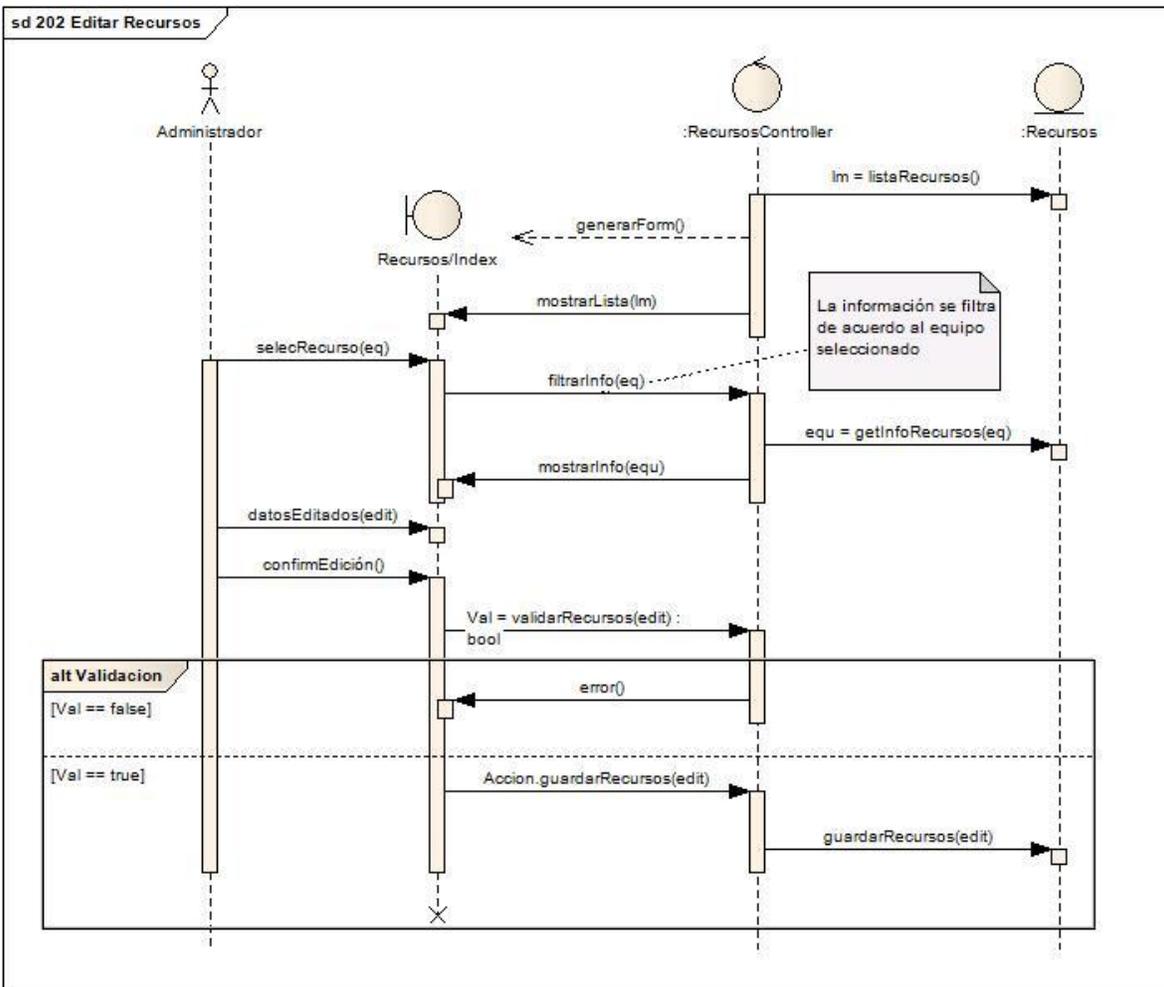


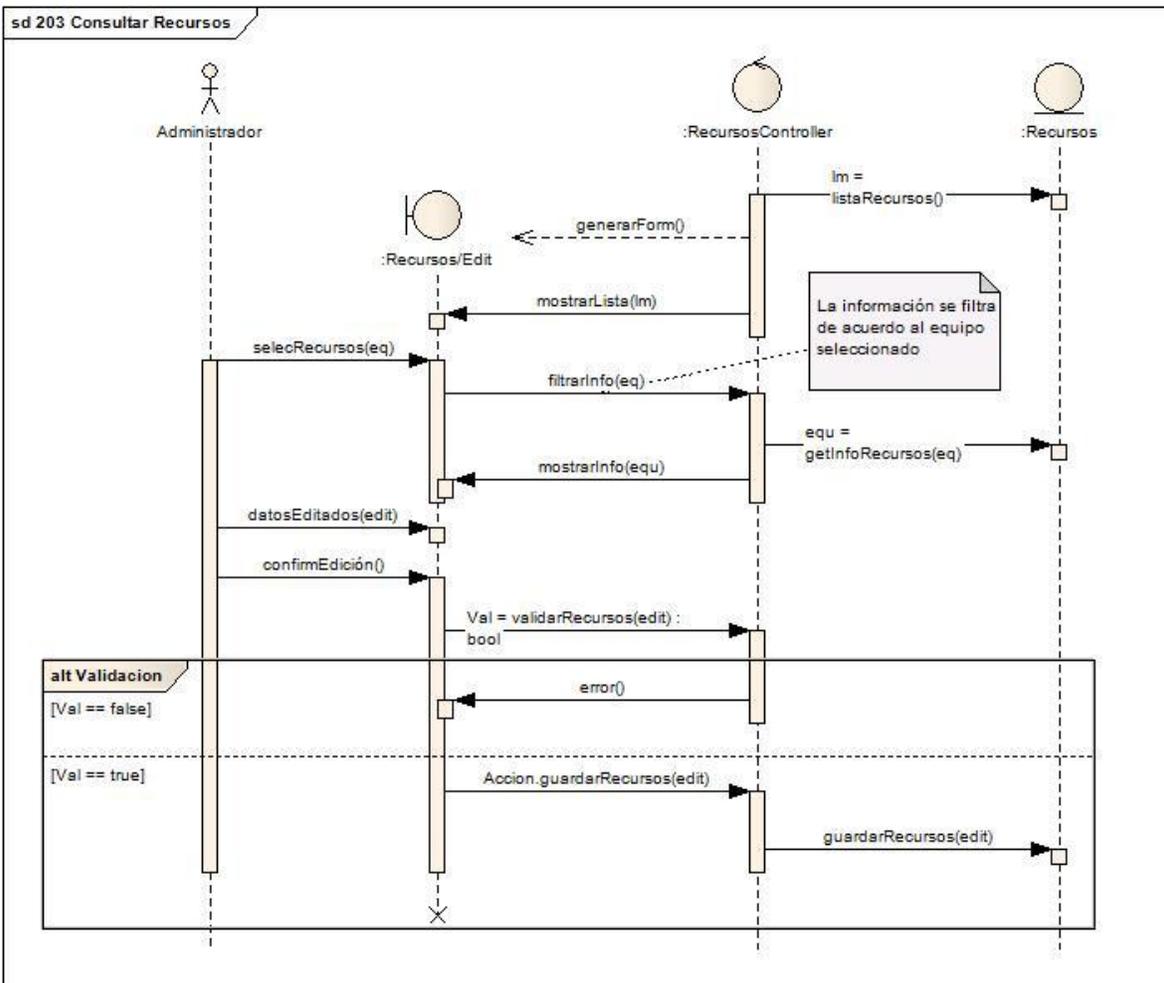


sd 104 Exportar Equipos

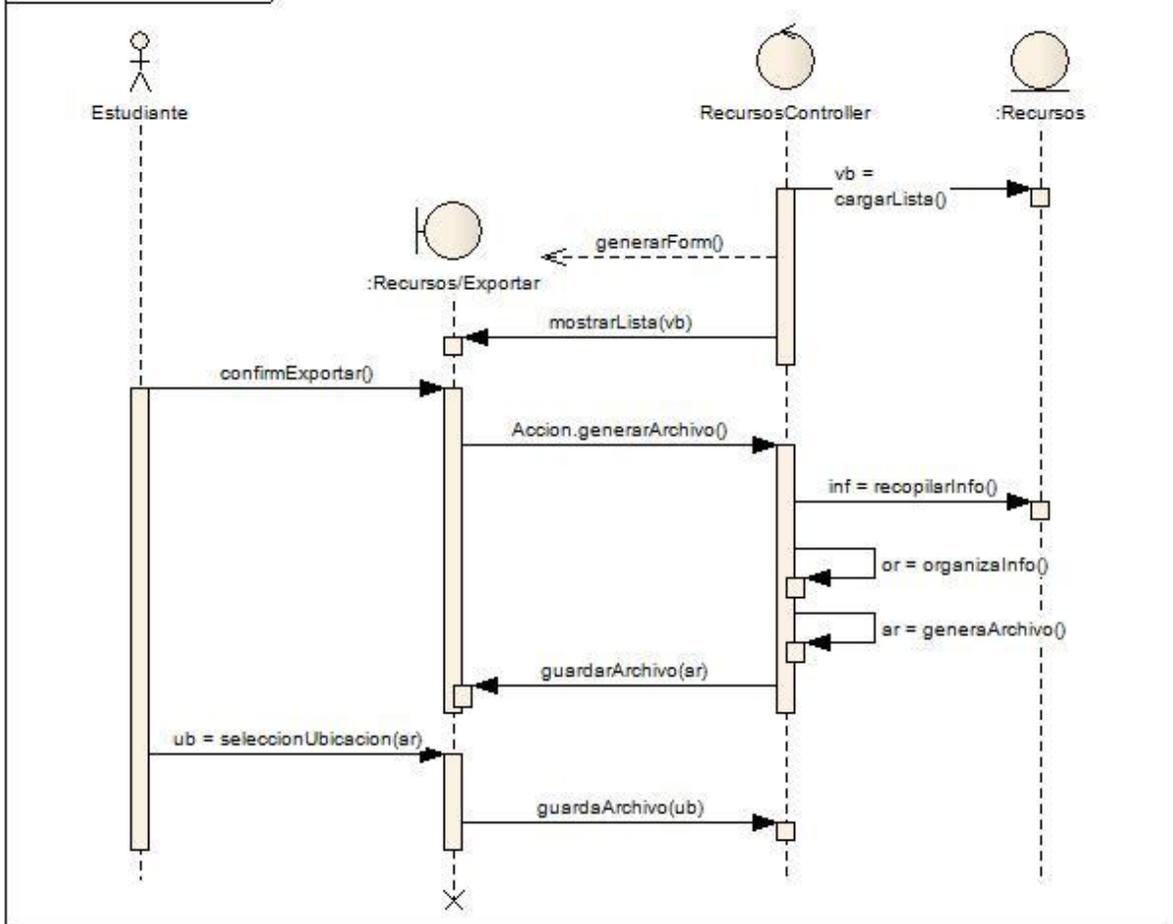


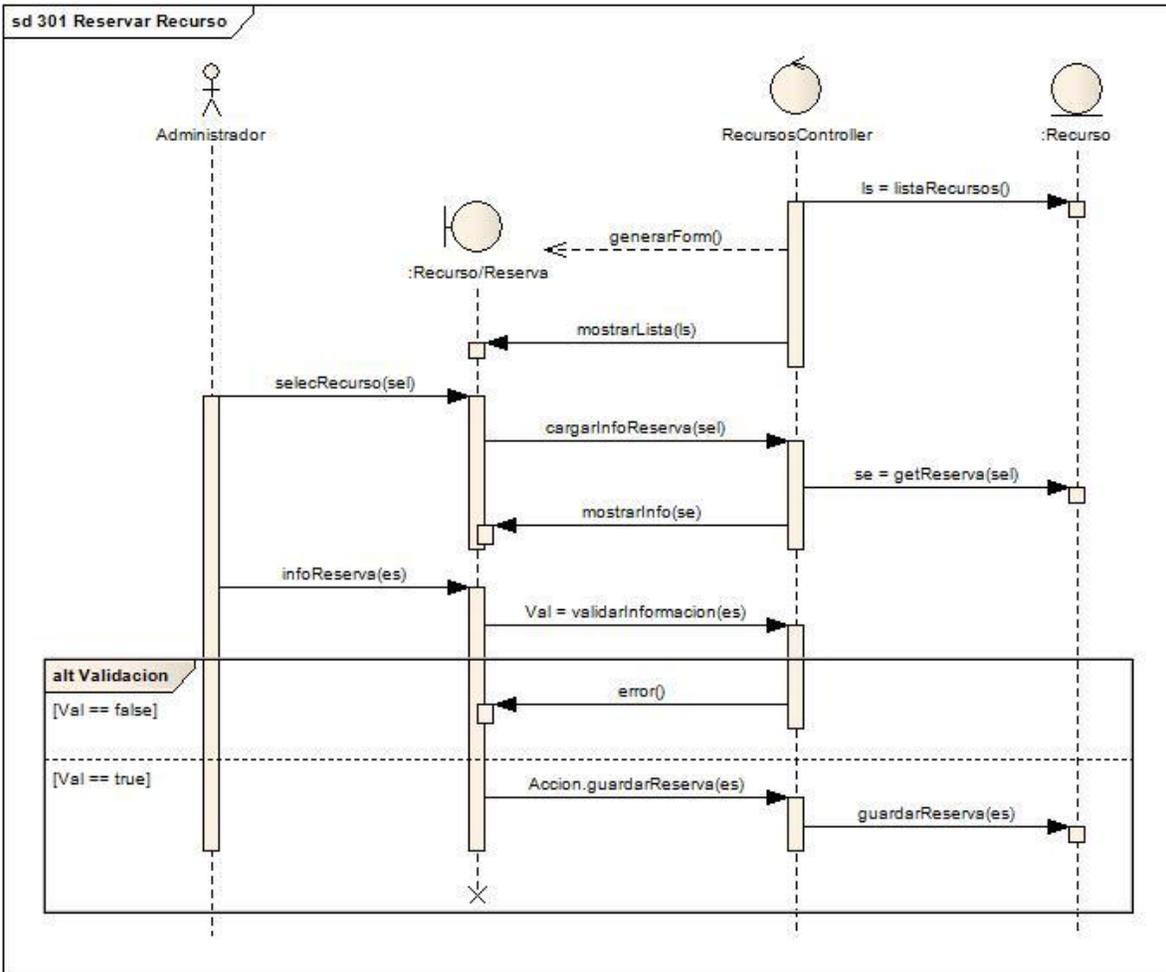


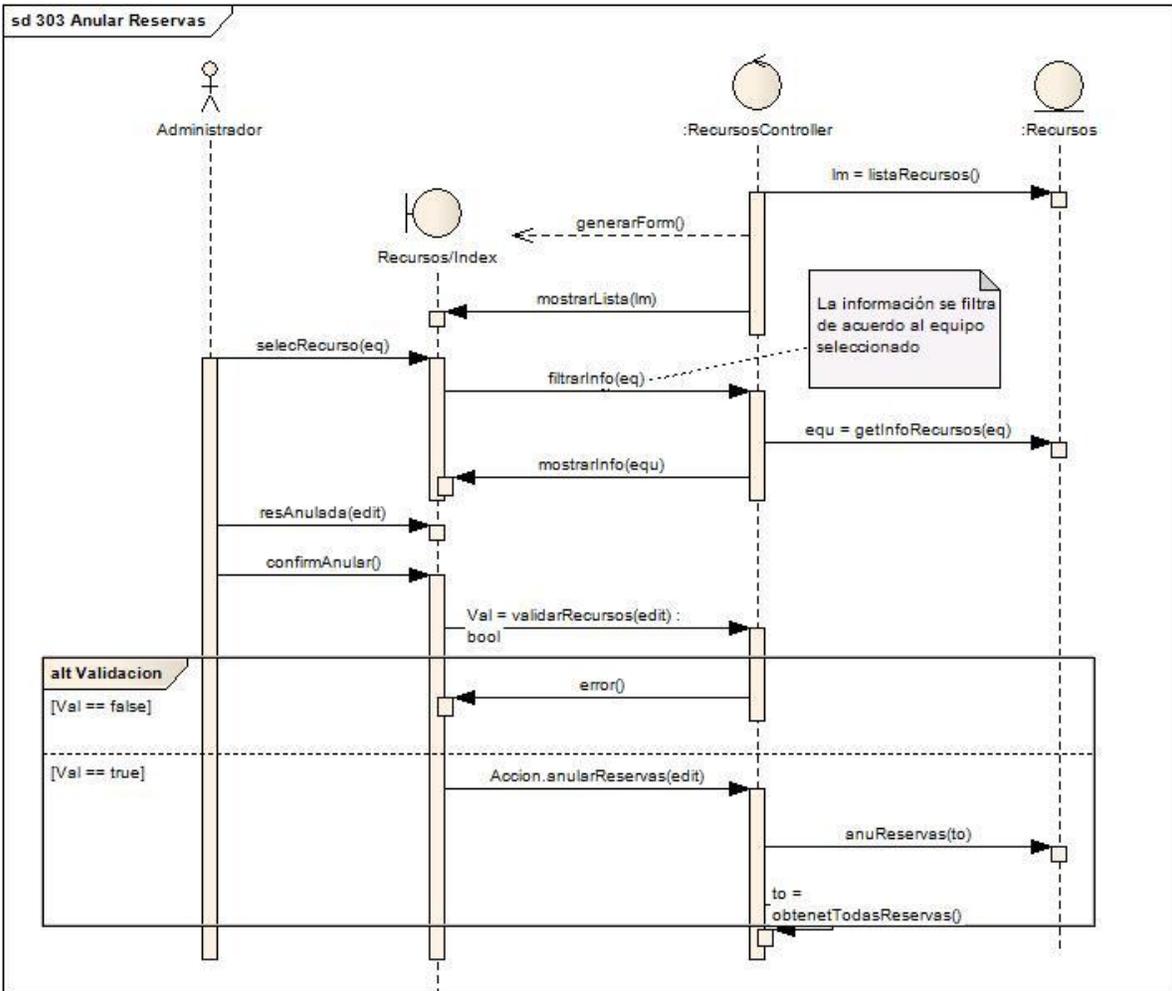


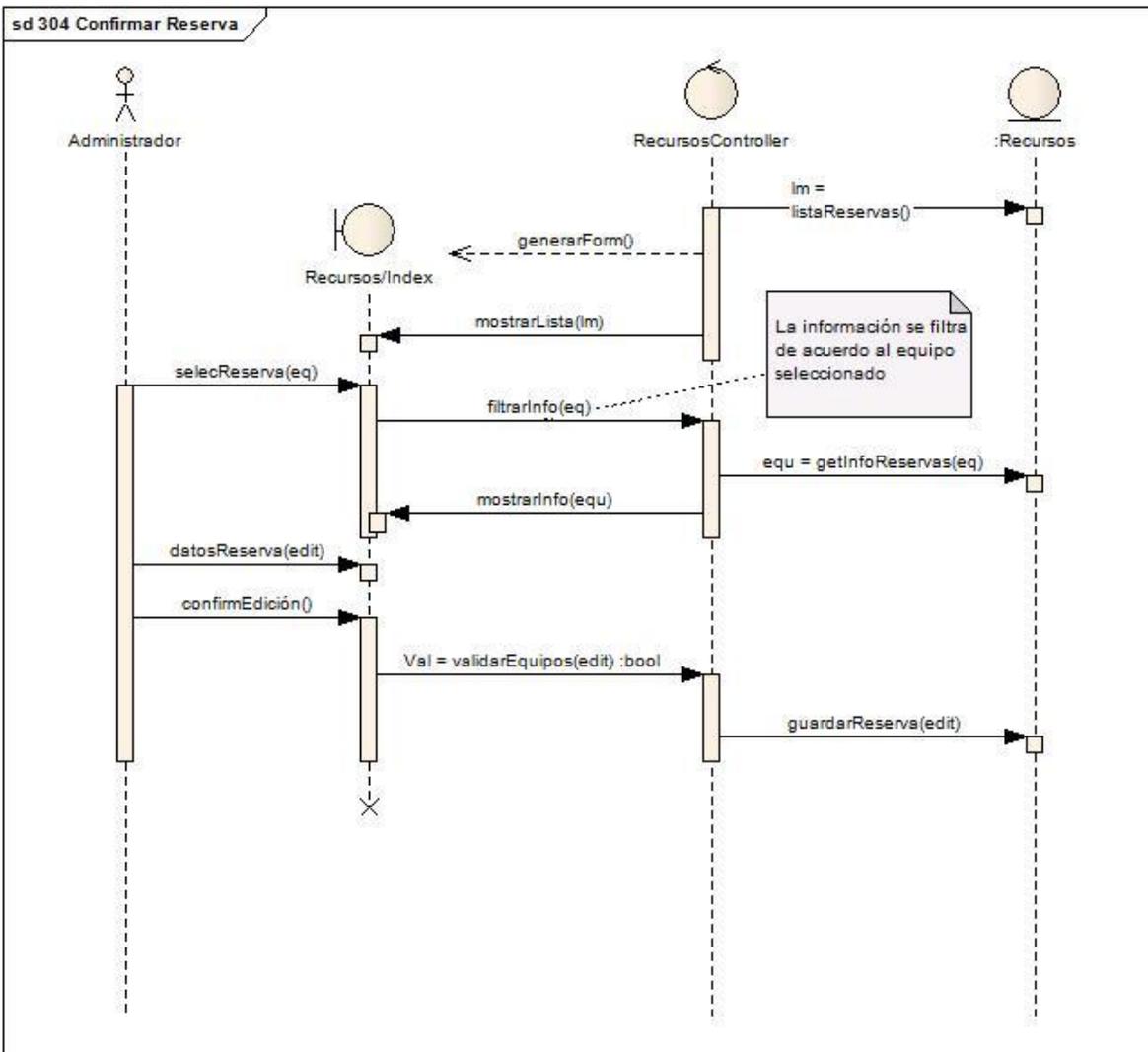


sd 204 Exportar Recursos

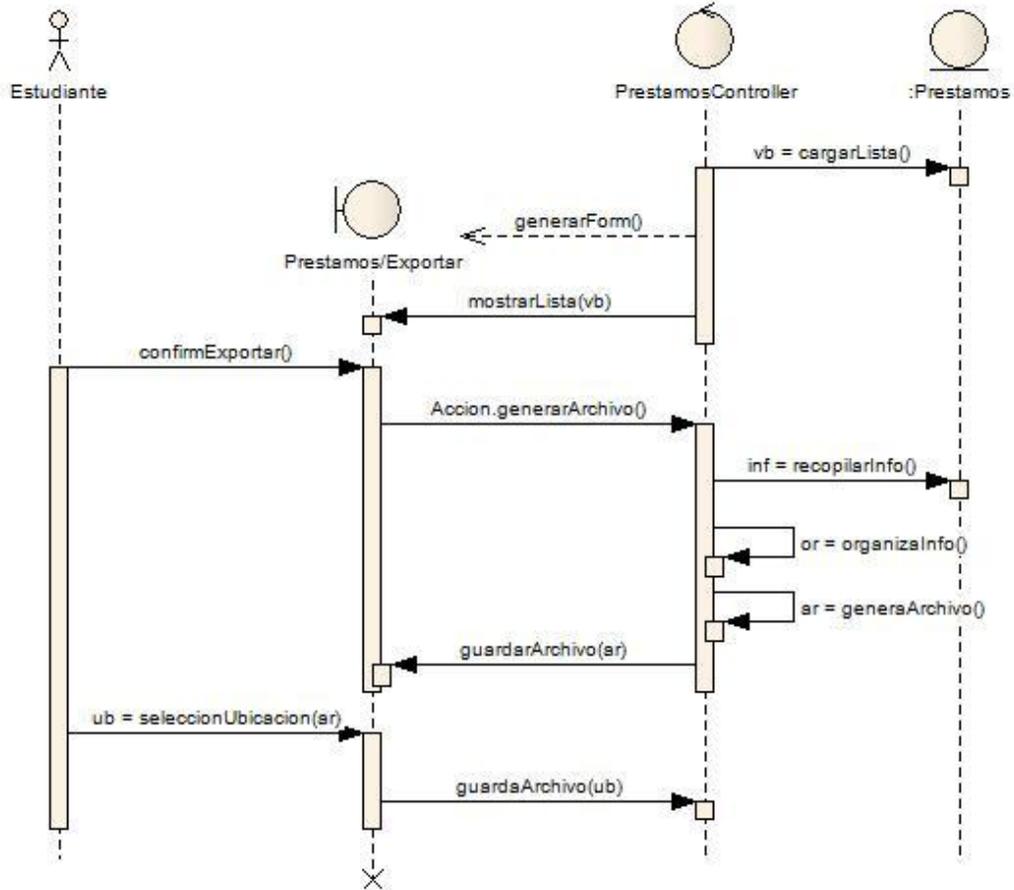




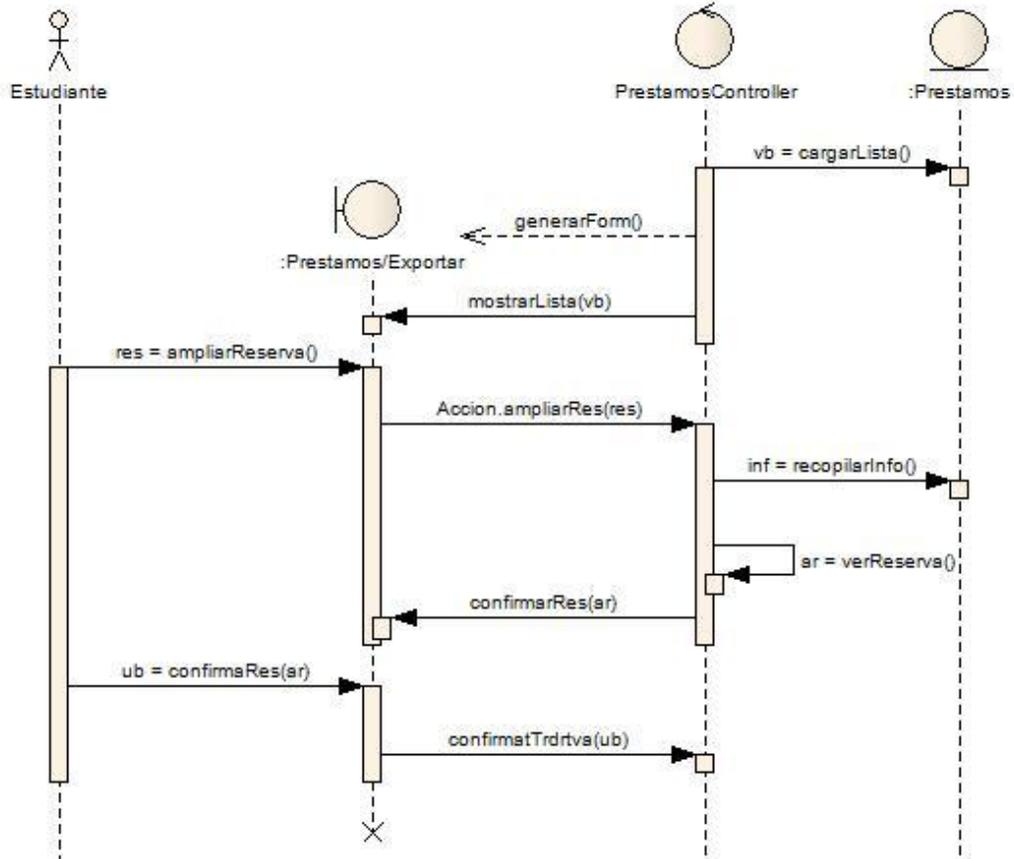


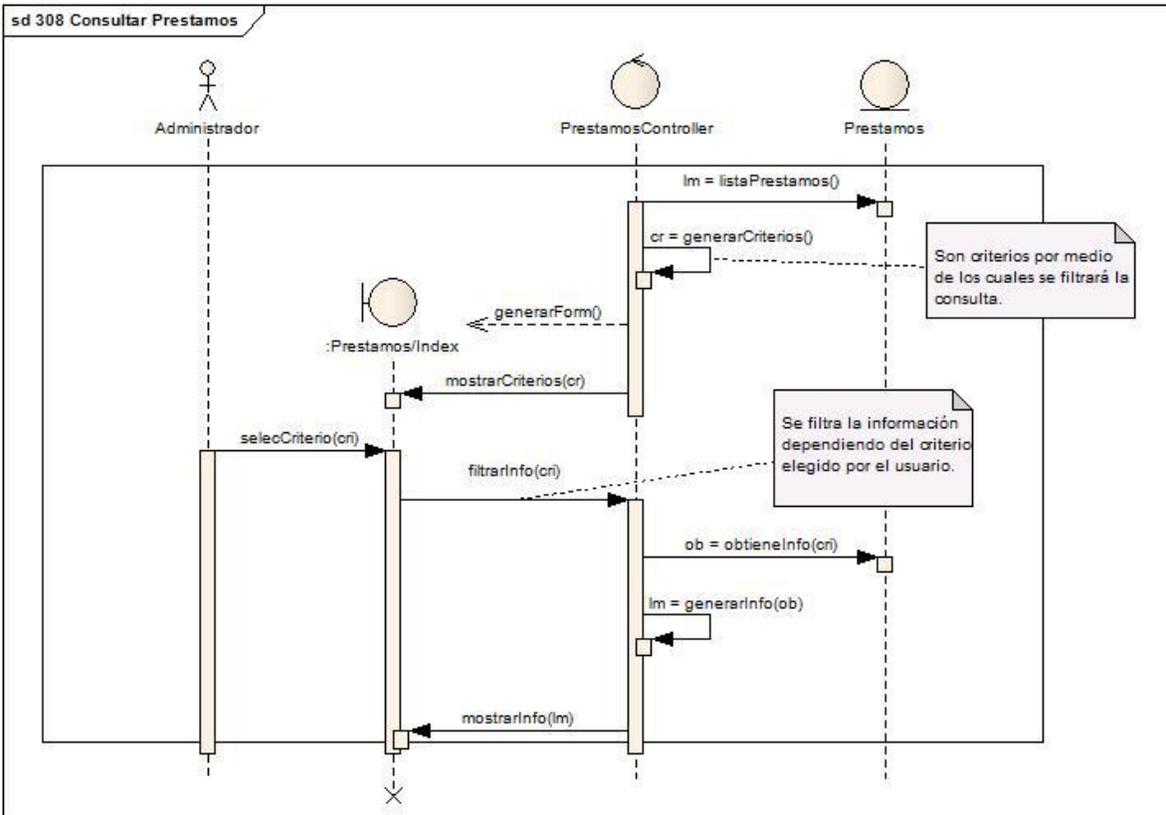


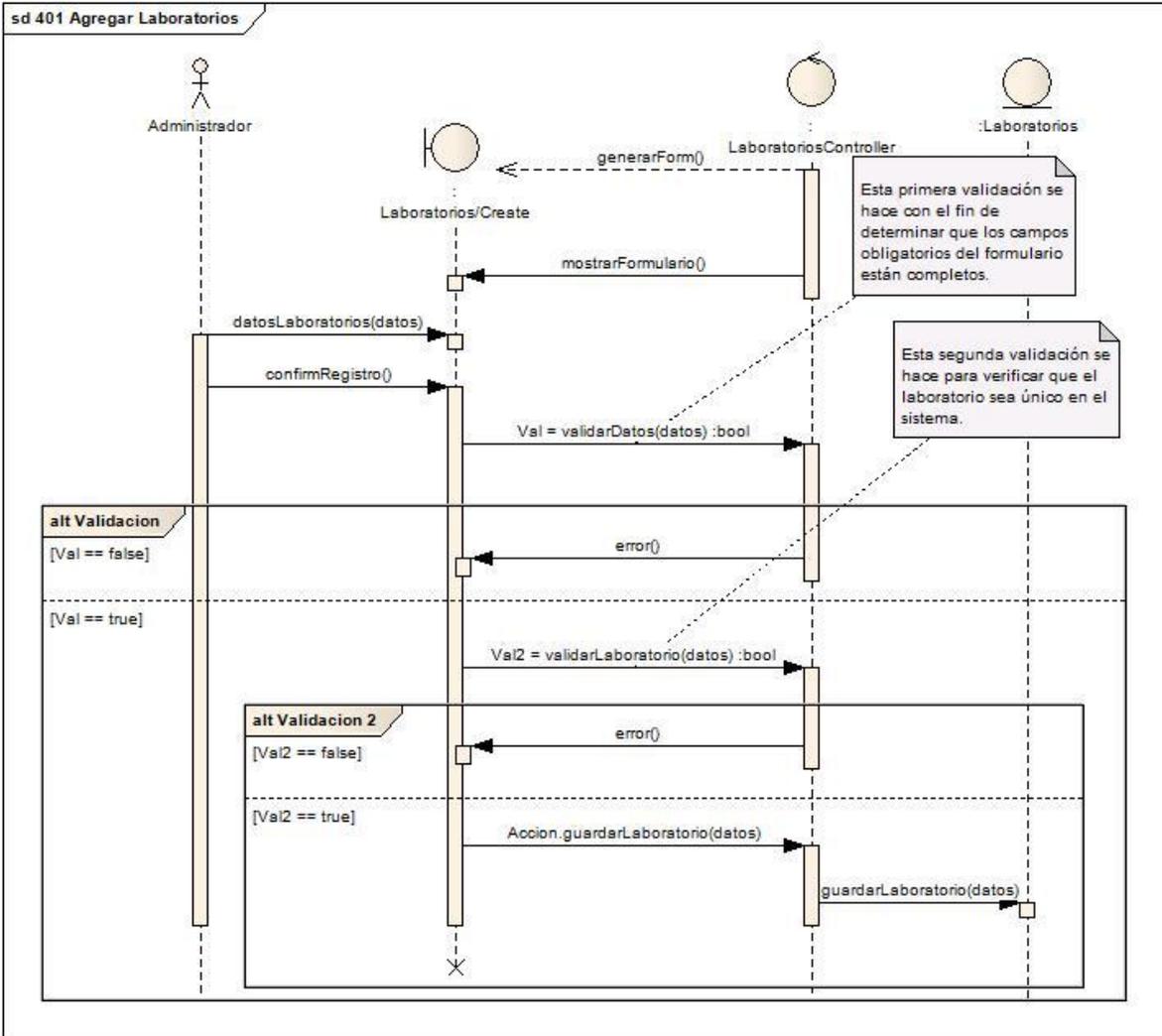
sd 305 Exportar Reservas

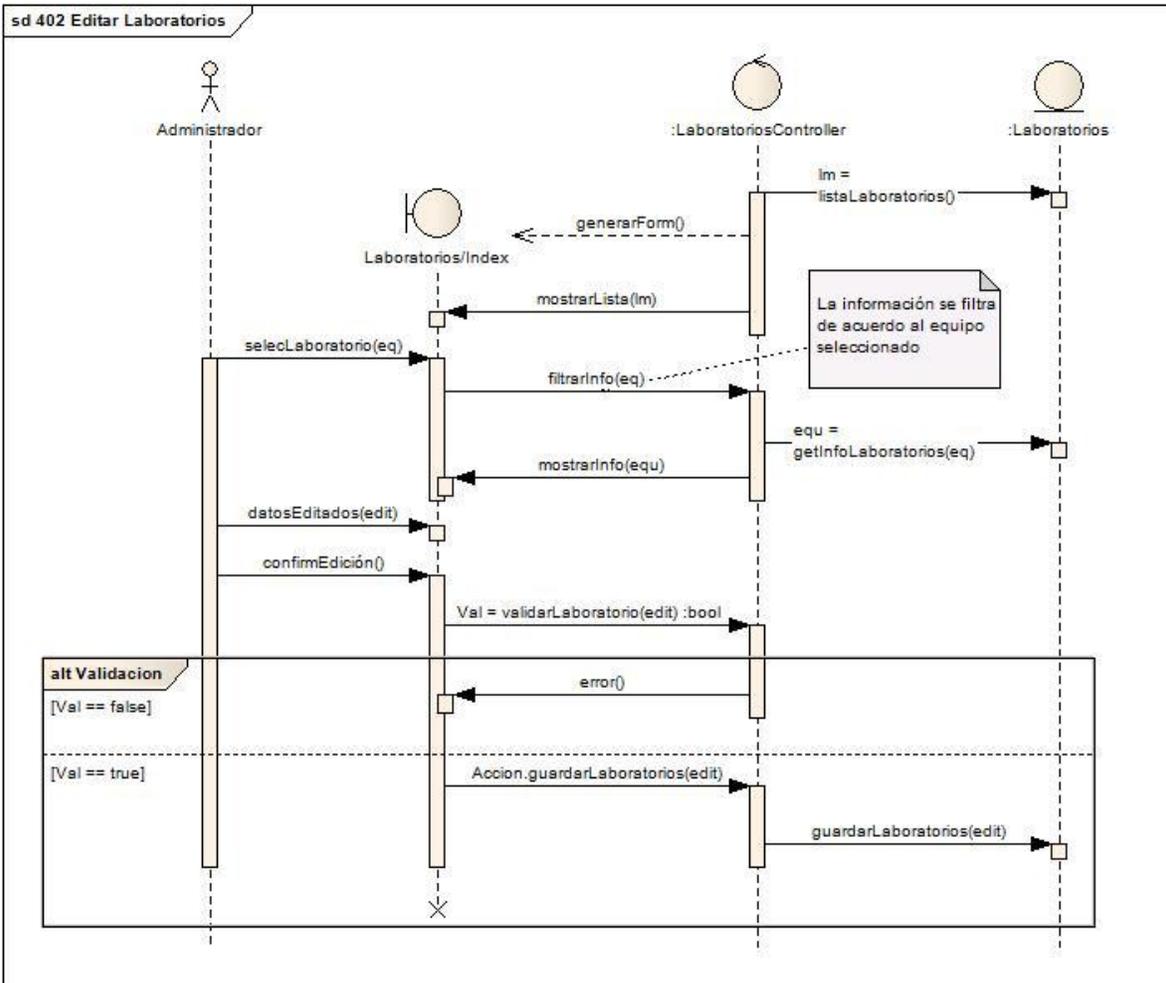


sd 307 Ampliar Reserva

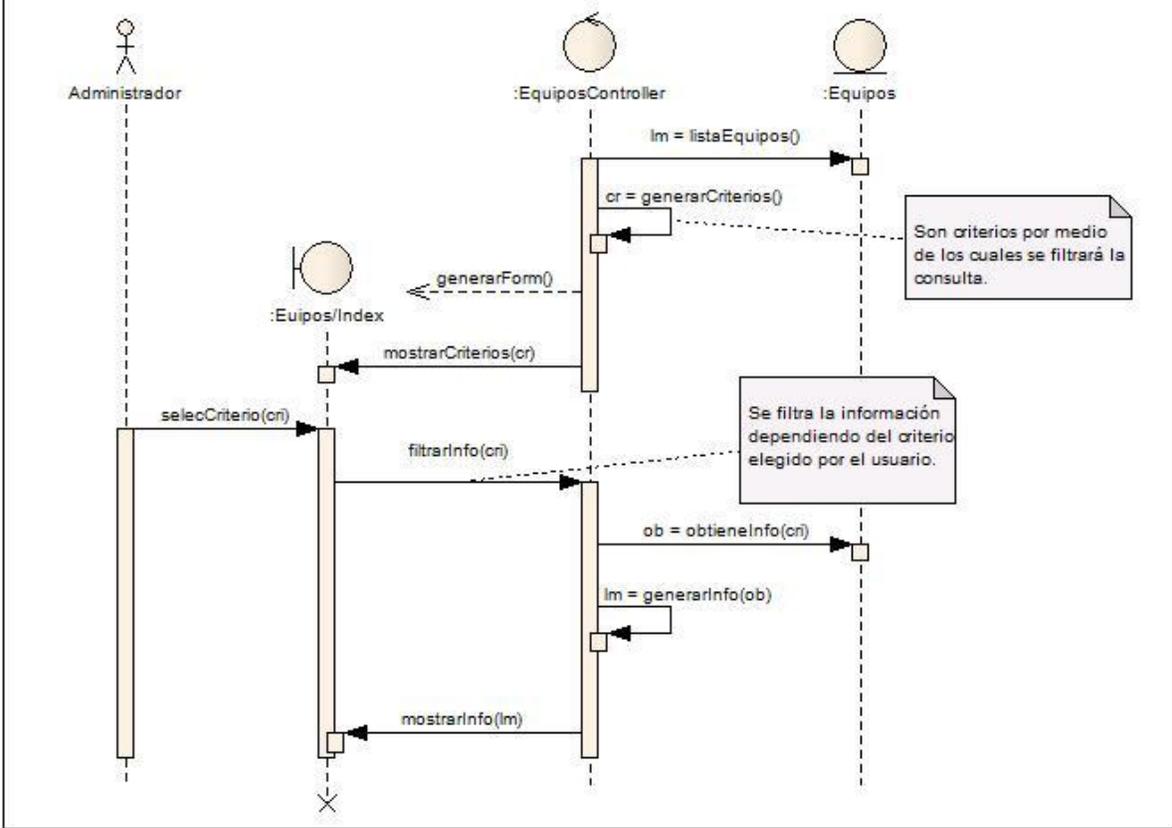


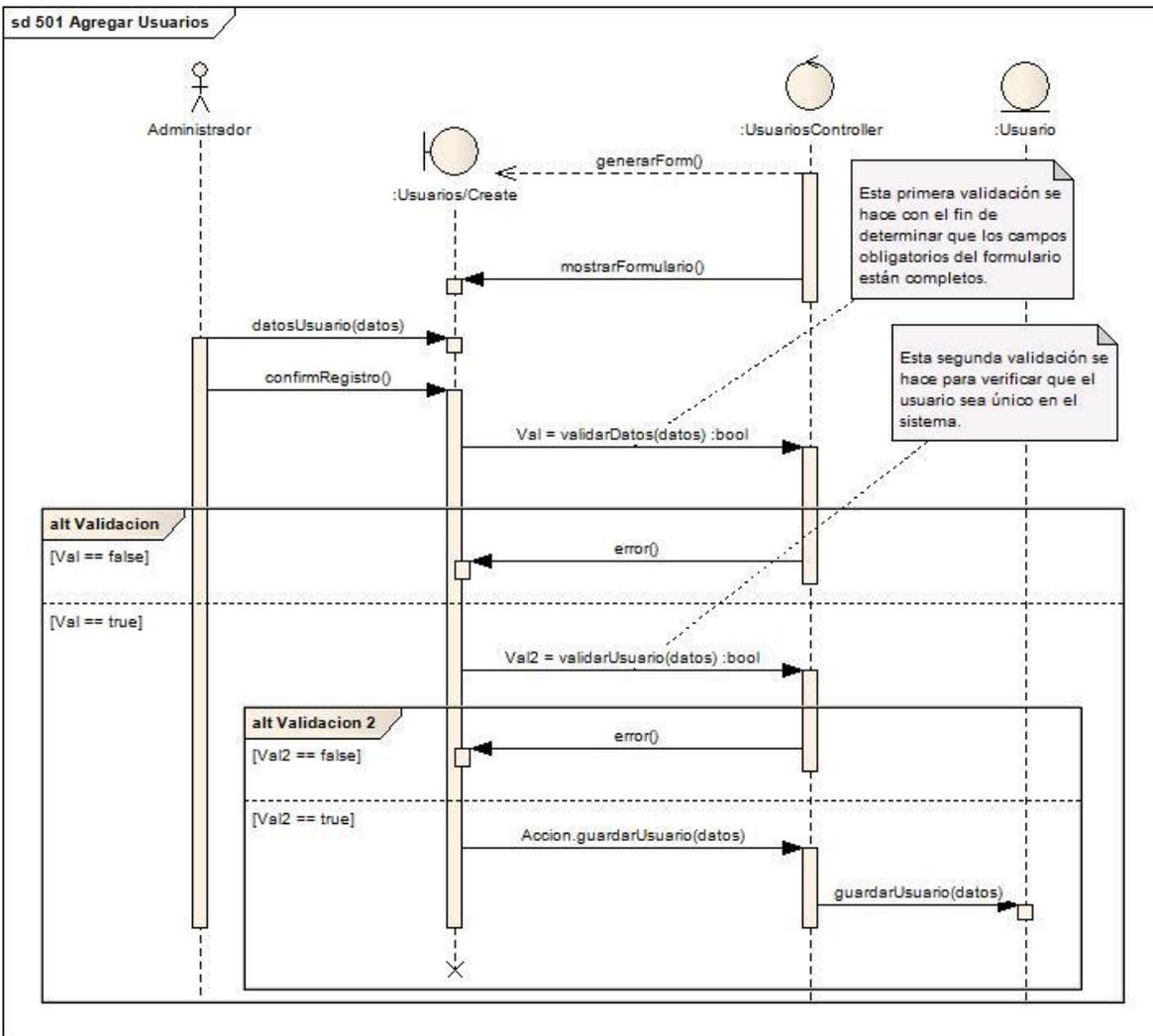




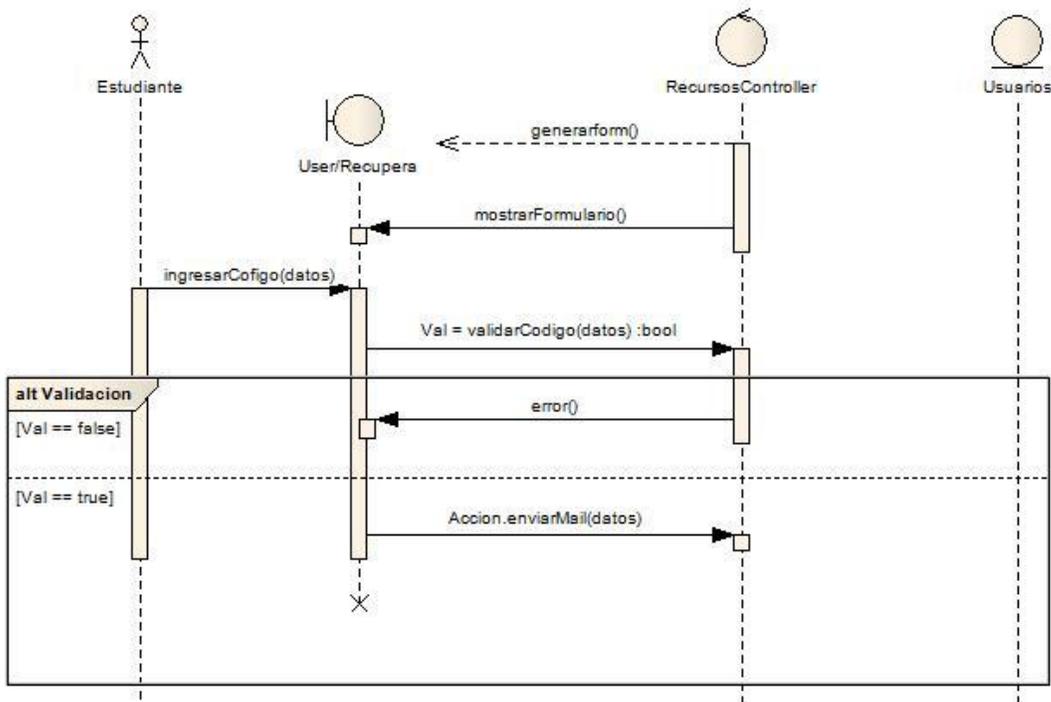


sd 403 ConsultarLaboratorios





sd 503 Recuperar Contraseña



3 DISEÑO

El diseño de un software se puede definir como el proceso de plantear y planear un sistema que sea la solución a un problema dado, el cual va a ser resultado por medio de un software. En esta etapa de un proceso de software es donde se consideran ciertos aspectos que van a ser tenidos en cuenta al momento de la implementación, entre esos unos de los más importantes son las metas de diseño.

Las metas de diseño guardan cierta relación con los requerimientos no funcionales. Como por ejemplo, una meta de diseño puede ser la robustez del sistema, o la seguridad del mismo, que a su vez vienen siendo requerimientos no funcionales del sistema. Por lo tanto en esta etapa se definen detalles importantes del sistema como los patrones de diseño a usar, los cuales se siguen para tratar de cumplir esas metas de diseño, aunque esto se detalla de acuerdo al nivel de prioridad definido en los requerimientos no funcionales.

3.1 CONTROL DE ACCESO

Para establecer la seguridad del sistema, se manejará un control de acceso estático que consiste en un sistema de usuario y contraseña basado en roles de usuario, los cuales, dependiendo del rol que se desempeñe dentro de la aplicación, tendrán visibilidad de ciertas funcionalidades definidas dentro de la herramienta. El privilegio está dado por la siguiente matriz de acceso donde se detalla cada usuario y las partes que le son visibles.

Servicio	Administrador	Profesor	Estudiante
Agregar equipos	X		
Consultar equipos	X	X	X
Editar equipos	X		
Deshabilitar equipos	X		

Exportar equipos	X	X	
Agregar recursos	X		
Consultar recursos	X	X	X
Editar recursos	X		
Exportar recursos		X	
Reservar recursos	X		X
Cancelar reservar	X		
Anular reservas	X		
Confirmar reserva	X		
Exportar reserva		X	
Ampliar reserva	X		X
Consultar prestamos	X	X	
Agregar laboratorio	X		
Editar laboratorio	X		
Consultar laboratorio	X	X	X
Exportar laboratorios		X	

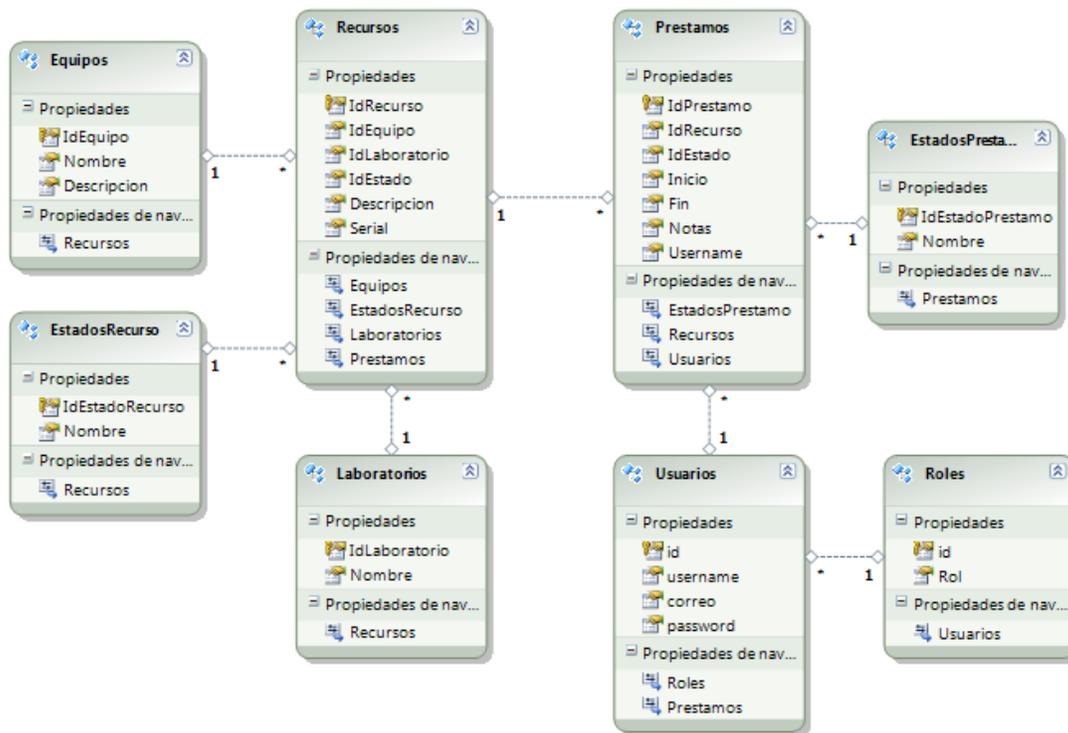
Agregar Usuarios	X		
Recuperar contraseña		X	X

3.2 CONTROL DE FLUJO

Para este sistema se va a usar un control de flujo basado en eventos, puesto que toda la interacción que se va a dar entre el usuario y el sistema va a ser basada en los eventos que realice el usuario en el lado del cliente, y las debidas respuestas generadas por parte del servidor que se envían de vuelta al cliente.

3.3 ALMACENAMIENTO PERSISTENTE

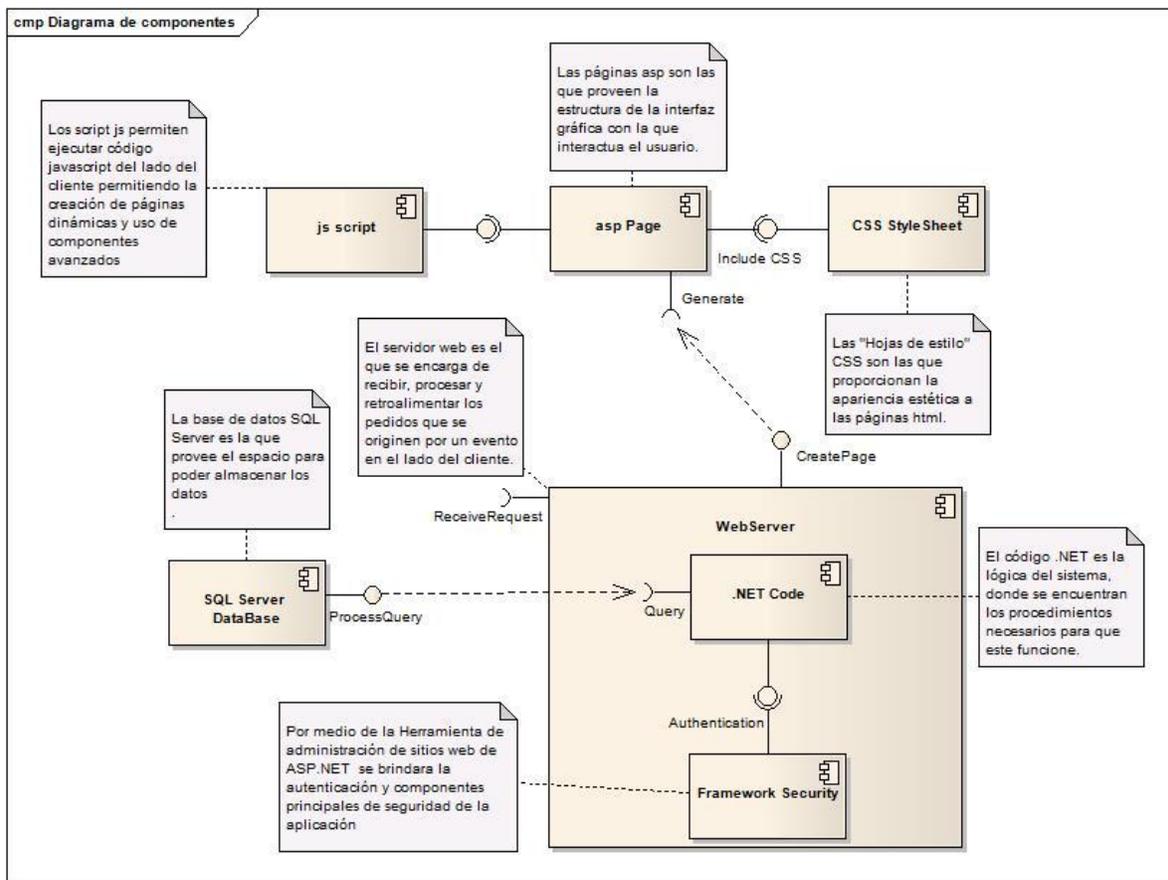
Para asegurar la persistencia de los datos en este sistema se va usar una base de datos relacional, la cual va a almacenar los datos que el usuario quiera que permanezcan para uso futuro en caso de que se cierre el sistema, esta base de datos será mapeada por medio de EntityFramework permitiendo un manejo más dinámico a nivel de aplicación y una integración más sencilla con ASP.NET MVC 3. El diagrama entidad-relación que describe la base de datos a usar, es el siguiente:



3.4 RELACIÓN HARDWARE/SOFTWARE

La descripción de la relación hardware/software se hace por medio de dos diagramas UML que son el diagrama de componentes y el diagrama de despliegue.

3.4.1 DIAGRAMA DE COMPONENTES



Asp Page: Las páginas asp son las que proveen la estructura de la interfaz gráfica con la que interactúa el usuario.

Js Script: Permiten la ejecución de código javascript del lado del cliente permitiendo la creación de páginas dinámicas y el uso de componentes avanzados como los de jQuery.

CSS Style Sheet: Las hojas de estilo son las que proporcionan la apariencia estética a las interfaces de usuario.

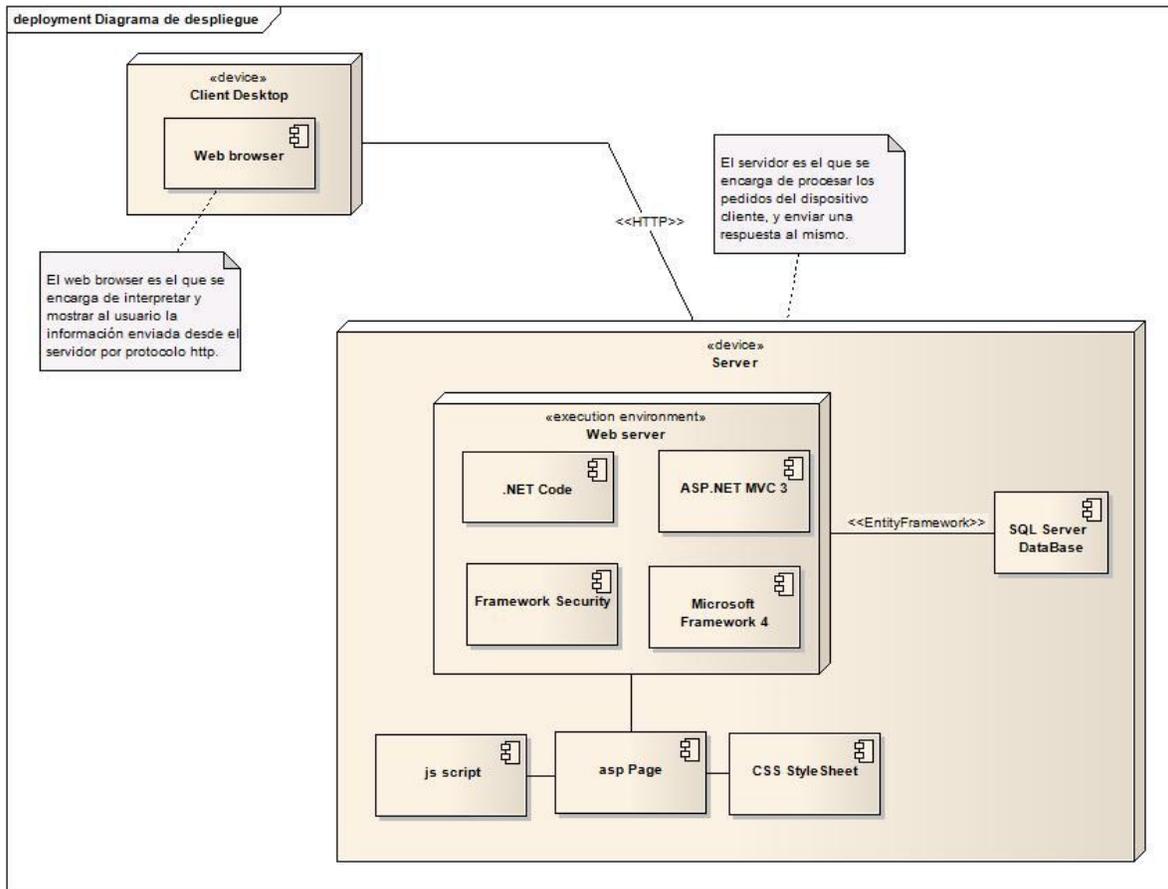
SQL Server DataBase: La base de datos SQL Server provee el espacio de almacenamiento de datos de la aplicación.

.NET Code: El código .NET brinda la lógica del sistema, es donde se encuentran los procedimientos necesarios para que la aplicación funcione.

Framework Security: Por medio de la herramienta de administración de sitios web de ASP.NET se brinda la autenticación y los componentes principales de seguridad

de la aplicación, brinda flexibilidad a la aplicación al poder migrar a futuro a otro sistema de autenticación como es SIRIUS.

3.4.2 DIAGRAMA DE DESPLIEGUE



ASP.NET MVC 3: El framework de desarrollo ASP.NET MVC 3 brinda un conjunto de herramientas para la creación de aplicaciones bajo la arquitectura MVC.

Microsoft Framework 4: El paquete Microsoft Framework 4 instala los componentes de .NET Framework necesarios para la ejecución en la arquitectura de máquina y el sistema operativo de destino. Contiene algunas librerías utilizadas en la aplicación, sin estas la aplicación no podrá ejecutarse satisfactoriamente.

3.4.3 REQUISITOS MINIMOS DE HARDWARE

Velocidad de conexión de red	Mínimo: <ul style="list-style-type: none"> • 1 Gb.
------------------------------	---

	<p>Recomendado:</p> <ul style="list-style-type: none"> • 10 Gb.
Memoria	<p>Mínimo:</p> <ul style="list-style-type: none"> • 4 GB. <p>Recomendado:</p> <ul style="list-style-type: none"> • Al menos 6 GB, debería aumentar a medida que crece la base de datos para evitar problemas de rendimiento.
Velocidad de procesamiento	<p>Mínimo:</p> <ul style="list-style-type: none"> • Procesador x86: 1,4 GHz. • Procesador x64: 1,8 GHz. <p>Recomendado:</p> <ul style="list-style-type: none"> • 2 GHz o superior.
Tipos de procesador	<ul style="list-style-type: none"> • Procesador x86: Compatible con Pentium III o superior. • Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T Intel Pentium IV compatible con EM64T.
Disco duro	<p>Mínimo:</p> <ul style="list-style-type: none"> • 10 GB. <p>Recomendado:</p> <ul style="list-style-type: none"> • 60 GB, debería aumentar a medida que crece la base de datos para evitar problemas de rendimiento.

4 VALIDACIÓN DEL SISTEMA

Para verificar el correcto funcionamiento del sistema se realizaron pruebas unitarias a cada uno de los subsistemas del mismo partiendo de las funciones principales de escritura y lectura, el Framework utilizado permite la creación de estas funciones CRUD (Create Read Update Delete) con los estándares requeridos.

Para estas pruebas se hicieron pruebas funcionales en las cuales nos pusimos en los zapatos de un usuario y se agregaron equipos básicos; por medio de un estudiante se hace la simulación de préstamos y de cancelación de los mismos. El usuario administrador gestiona cada uno de los equipos y recursos a nivel de aplicación permitiendo tener una interacción completa con el sistema.

En estas pruebas se confirmó la operatividad de los requerimientos funcionales y la integridad de la información en cada uno de los subsistemas, después de la interacción sin fallas en el sistema.

En la siguiente matriz se describen los componentes implementados y se hacen observaciones de su implementación.

Sistema	Req Implementados	Comentarios
Equipos	Agregar equipos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Consultar equipos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Editar equipos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Deshabilitar equipos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente

Recursos	Agregar recursos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Consultar recursos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Editar recursos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Reservar recursos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
Reservas y Prestamos	Cancelar reservar	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Anular reservas	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Confirmar reserva	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Control de Reservas	Se implementó mediante un stored procedure en la tabla Prestamos
	Ampliar reserva	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Consultar prestamos	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
Laboratorios	Agregar laboratorio	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente

	Editar laboratorio	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
	Consultar laboratorio	Se implementó con ASP.NET MVC 3, se corrieron pruebas unitarias superadas satisfactoriamente
Usuarios	Agregar Usuarios	Se encuentra implementado mediante la herramienta de administración de sitios web de ASP
	Control de Usuarios	Se encuentra implementado mediante la herramienta de administración de sitios web de ASP
	Recuperar contraseña	Se encuentra implementado mediante la herramienta de administración de sitios web de ASP

CONCLUSIONES

- Se desarrolló una aplicación para la administración electrónica del laboratorio de redes de la Universidad Tecnológica que satisface las necesidades puntuales y facilita el manejo y control de los recursos disponibles.
- Aunque la herramienta fue diseñada para la administración electrónica del laboratorio de redes se pueden agregar más laboratorios y gestionar los inventarios en cada uno de estos.
- La reserva de los equipos puede realizarse fácilmente con un nombre de usuario, contraseña y pocos clics permitiendo el acceso a los recursos más rápida y eficientemente.
- Se puede llevar un control de las reservas y prestamos realizados de los recursos disponibles a los estudiantes
- Se elaboró la documentación de la herramienta de software detallando a cabalidad cada uno de sus servicios y componentes, lo cual ayuda al posterior mantenimiento de la aplicación y mejor comprensión por parte de los interesados.
- El modelamiento del sistema se realizó bajo el patrón arquitectónico MVCy se implementó con ASP.NET MVC 3 que finalmente es una herramienta muy útil para la escritura del código y permitir la escalabilidad del mismo.
- EntityFramework permite la integración dinámica de los datos permitiendo una interacción más amigable con los mismos y ASP.NET MVC 3, permite el manejo de los datos como objetos y por ende menos desgaste con las consultas TSQL.
- Se agregaron funcionalidades sugeridas por el solicitante (deseables) para mejorar al brindar los servicios a los usuarios del sistema.
- Al usar el sistema de autenticación de la herramienta de administración de sitios web de ASP no se liga la arquitectura a una tabla usuario permitiendo la flexibilidad para el ingreso de usuarios si en un futuro se desea integrar a SIRIUS o cualquier otro sistema que se designe.

RECOMENDACIONES

Para futuras modificaciones y mantenimiento en el sistema:

- Revisar completamente el documento identificando servicios y estructura interna (diagramas de secuencias, diagrama de casos de uso, clases y tablas de la base de datos).
- Las modificaciones que se quieran realizar tienen que ser con el objetivo de mejorar los servicios ofrecidos apoyándose en los requerimientos existentes y agregando más funcionalidad al sistema a medida que surjan nuevos requerimientos.
- Tener en cuenta los objetivos que se quieren alcanzar con la implementación del software para no saturar el sistema de servicios innecesarios que solo causen un desmejoramiento en el desempeño del mismo.
- Las metas de diseño son una base importante a tener en cuenta para la satisfacción del cliente. Como no son medidas por los servicios ofrecidos hay que elaborar una buena arquitectura para cumplir con los requerimientos no funcionales.
- El sistema es fácilmente escalable y por lo tanto se pueden agregar nuevas funcionalidades mientras se cumpla con el patrón. Un ejemplo claro es que usando ASP MVC 3 podemos llegar incluso al campo de dispositivos móviles permitiendo dividir las vistas según el dispositivo con la etiqueta #if MOBILE, así seguiremos asociando un controlador a una sola vista específica sin afectar la filosofía del patrón.
- Como módulos de ampliación se puede pensar en una primera instancia en un módulo de catalogación (Agrupar los equipos por características similares) y un agrupamiento (Agrupamientos manuales según las experiencias que se vayan a realizar en el laboratorio) manual para facilidad de los profesores y alumnos para realizar prácticas en el laboratorio.
- El control de acceso se logra mediante un rol, si a futuro se desea cambiar la forma de autenticación solamente se necesita el nuevo rol y asignarle los

permisos respectivos a nivel de aplicación. De la autenticación solo se necesita un código SIRIUS y un rol.

BIBLIOGRAFIA

- [1] Pressman, Roger. Ingeniería de Software, Un enfoque práctico, Quinta edición, Mc Graw Hill.
- [2] Somerville, Ian. Software Engineering, Sexta edición, Addison Wesley.
- [3] Diagramas de componentes de UML: Referencia (Articulo de Internet)
<http://msdn.microsoft.com/es-es/library/dd409390.aspx>
- [4] ASP.NET MVC 3 – Desarrollar Híbridos Nativos y Mobile web Apps (Articulo de Internet) <http://msdn.microsoft.com/es-es/magazine/hh852592.aspx>
- [5] Diagrama de despliegue UML 2 (Articulo de Internet)
http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html