

UNIVERSIDAD AMAZÓNICA DE PANDO
ÁREA CIENCIAS Y TECNOLOGÍA
CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“IMPLEMENTACIÓN DE UN SISTEMA INFORMATICO PARA EL
CONTROL DE AULAS DEL CAMPUS UNIVERSITARIO DE LA
UNIVERSIDAD AMAZONICA DE PANDO”**

PROYECTO DE GRADO PRESENTADO PARA OBTENER EL TITULO
ACADEMICO DE LICENCIADO EN INGENIERIA DE SISTEMAS

Postulante : Univ. Gaddy Huaygua Mamani

Tutor : Ing. Christian Miahuchi Nataly

Asesor : Ing. Juan Carlos Huanca Guanca

Cobija – Pando – Bolivia

2015

DEDICATORIA

Dedico este trabajo a mis padres Mario Huaygua y Aquilas Mamani quienes me apoyaron económica y moralmente en todos los momentos buenos y malos brindándome su cariño y comprensión para culminar con uno de mis objetivos dentro de mi vida.

Y a mis hermanas que fueron de gran ayuda en la finalización de este proyecto.

Gracias por la confianza y por sus esfuerzos diarios!!.

AGRADECIMIENTO

Agradezco a Dios por haberme dado fuerzas en los momentos más difíciles y la sabiduría para tomar las decisiones correctas.

Agradecer a mis padres y mis hermanas que siempre me ayudaron dándome su confianza y apoyo.

A mi docente asesor Lic. Juan Carlos Huanca Guanca y tutor Ing. Christian Miahuchi Nataly, al Lic. Javier Patty Magne y al Lic. Samuel Fuente por haberme guiado durante este tiempo en el proceso de desarrollo del presente proyecto.

RESUMEN

A la presente sinopsis proporciona un enfoque general sobre la elaboración del presente proyecto de grado, con el propósito de proveer una mejor comprensión al lector.

A través del análisis realizado a la unidad, se identifica el siguiente problema “Control Deficiente de las Aulas del campus Universitario dependientes de la Unidad de Administración de Ambientes por la pérdida de información no sistematizada”.

De acuerdo al problema se realiza el siguiente objetivo “Implementar un sistema informático para el control de aulas del campus universitario de la UAP, utilizando la metodología Proceso Unificado Racional (RUP).”.

Para ello se realiza el análisis de flujo de información en la unidad lo cual permite definir los requerimientos funcionales del sistema, y obtener la información necesaria para el desarrollo de la de la base de datos y los módulos del sistema.

Concluido el sistema informático en base a la metodología RUP, se cuenta con un sistema amigable que permite realizar el registro de solicitud de aulas para la asignación del aulas requeridos, además genera diferentes tipos de búsqueda y listados, de la misma forma permite la asignación de usuarios y roles, también genera reportes de ambientes en uso y ambientes libres, con características que presenta la información de una manera gráfica y amigable almacenando bloques y ambientes del campus universitario.

De esta manera el “sistema de control de aulas” facilita el control que realiza constantemente la unidad siendo de vital importancia tanto para la institución (universidad amazónica de pando) como para la población universitaria

INDICE GENERAL

CAPITULO I MARCO REFERENCIAL

1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	3
1.3 OBJETIVOS	4
1.3.1 <i>Objetivo general</i>	4
1.3.2 <i>Objetivos Específicos</i>	4
1.4 METODOLOGIA.....	4
1.5 ALCANCES	6

CAPITULO II MARCO METODOLOGICO

2.1 METODOLOGIA RACIONAL UNIFIED PROCESS (PROCESO UNIFICADO RACIONAL)	11
2.1.1 <i>ESTRUCTURA DEL RUP</i>	11
2.1.2 <i>CICLO DE VIDA DE LA METODOLOGÍA RUP</i>	13
2.2 HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO	14
2.2.1 <i>LENGUAJE UNIFICADO DE MODELADO (UML)</i>	14
2.2.2 <i>DIAGRAMAS DE UML</i>	19
2.3 LENGUAJE DE PROGRAMACION	20
2.3.1 <i>JAVA</i>	20
2.3.2 <i>HTML</i>	21
2.3.3 <i>JAVA SCRIPT</i>	22
2.3.4 <i>HOJAS DE ESTILO CASCADING STYLE SHEETS (CSS)</i>	23
2.3.5 <i>IREPORT</i>	23
2.4 SERVIDOR DE APLICACIONES WEB	24
2.4.1 <i>APACHE TOMCAT</i>	24
2.5 BASE DE DATOS	25
2.5.1 <i>POSTGRESQL</i>	25

CAPITULO III MARCO APLICATIVO

3.1 DESARROLLO DEL SISTEMA.....	26
3.1.1 <i>FASE I: FASE DE INICIO</i>	26
3.1.2 <i>FASE II: FASE DE ELABORACION</i>	31

3.1.3 FASE III: FASE DE CONSTRUCCION65
3.1.4 FASE IV: FASE DE TRANSICION.....75

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES80
4.2. RECOMENDACIONES81
BIBLIOGRAFIA82
ANEXOS A.....
ANEXO B81
ANEXO C82
ANEXO D.....

ÍNDICE DE DIAGRAMAS

<i>Diagrama 3.1: Modelo de proceso de negocios asignación de aulas</i>	26
<i>Diagrama 3.2: diagrama de Actividades: Asignación de Aulas</i>	28
<i>Diagrama 3.3: Modelo de negocios</i>	32
<i>Diagrama 3.4: identificación de usuarios del sistema: Autenticar usuario</i>	34
<i>Diagrama 3.5: Modelo de casos de uso del sistema</i>	35
<i>Diagrama 3.6: Caso de uso del módulo administrar usuario</i>	36
<i>Diagrama 3.7: Caso de uso Gestor datos personales</i>	36
<i>Diagrama 3.8: Caso de uso Gestor datos personales</i>	37
<i>Diagrama 3.9: Caso de uso Gestor Cargo</i>	39
<i>Diagrama 3.10: Caso de uso Gestor rol</i>	40
<i>Diagrama 3.11: Caso de uso Gestor Usuario</i>	42
<i>Diagrama 3.12: Caso de uso Gestor Usuario</i>	44
<i>Diagrama 3.13: Diagrama de Caso de Uso: Administrar bloque</i>	45
<i>Diagrama 3.14: Caso de uso Gestor Bloque</i>	45
<i>Diagrama 3.15: Caso de uso Gestor tipo aula</i>	47
<i>Diagrama 3.16: Caso de uso Gestor aula</i>	47
<i>Diagrama 3-0.17: Caso de uso Gestor Horario</i>	48
<i>Diagrama 3.18: Diagrama de Caso de Uso: Administrar Solicitud</i>	50
<i>Diagrama 3.19: Caso de uso Gestor Solicitud</i>	50
<i>Diagrama 3.20: Diagrama de Clases: Autenticación de Usuario</i>	52
<i>Diagrama 3.21: Diagrama de Clase: Usuario</i>	52
<i>Diagrama 3.22: Diagrama de Clases: Registro de Bloque</i>	53
<i>Diagrama 3.23: Diagrama de Clases: Solicitud</i>	54
<i>Diagrama 3.24: Diagrama de Secuencia: Autenticar Usuario</i>	55
<i>Diagrama 3.25: Diagrama de Secuencia: Registrar Usuario</i>	56
<i>Diagrama 3.26: Diagrama de Secuencia: Asignar rol a usuario</i>	57
<i>Diagrama 3.27: Diagrama de Secuencia: Asignar opción a usuario</i>	58
<i>Diagrama 3.28: Diagrama de Secuencia: Registrar Bloque</i>	59
<i>Diagrama 3.29: Diagrama de Secuencia: Listar Bloque</i>	60
<i>Diagrama 3.30: Diagrama de Secuencia: Registrar Solicitud</i>	61
<i>Diagrama 3.31: Diagrama de Secuencia: Asignación de ambiente</i>	62
<i>Diagrama 3.32: Modelo Conceptual de la Base de Datos del sistema de información SAAC-UAP</i>	66
<i>Diagrama 3.33: de Despliegue: Sistema informático para el control de ambientes</i>	75
<i>Diagrama A.1: Árbol de problemas</i>	54
<i>Diagrama A.2: Árbol de Objetivos</i>	54
<i>Diagrama A.3: Cuadro de Involucrado</i>	54
<i>Diagrama A.4: Matriz de marco lógico</i>	¡Error! Marcador no definido.

INDICE DE TABLAS

<i>Tabla 1.1: Especificación de Aulas</i>	2
<i>Tabla 1.2: Metodología RUP</i>	5
<i>Tabla 3.3: Requerimientos del sistema SAAC-UAP.</i>	29
<i>Tabla 3.4: Requisitos no funcionales.</i>	30
<i>Tabla 3.5: Identificación de Actores</i>	31
<i>Tabla 3.6: Catalogo de caso de uso de negocio</i>	33
<i>Tabla 3.7: Descripción del caso de uso autenticar</i>	37
<i>Tabla 3.8: Descripción del caso de uso Gestor datos personales</i>	38
<i>Tabla 3.9: Descripción del caso de uso Gestor cargo</i>	40
<i>Tabla 3.10: Descripción del caso de uso Gestor Rol</i>	41
<i>Tabla 3.11: Descripción del caso de uso Gestor bloque</i>	46
<i>Tabla 3.12: Descripción del caso de uso Gestor tipo aula</i>	47
<i>Tabla 3.13: Descripción del caso de uso Gestor Aula</i>	48
<i>Tabla 3.14: Descripción del caso de uso Gestor Horario</i>	49
<i>Tabla 3.15: Descripción del caso de uso Gestor Solicitud</i>	51
<i>Tabla 3.16: tabla persona del modelo entidad</i>	67
<i>Tabla 3.17: tabla usuario del modelo entidad</i>	68
<i>Tabla 3.18: tabla rol del modelo entidad</i>	68
<i>Tabla 3.19: tabla Cargo del modelo entidad</i>	68
<i>Tabla 3.20: tabla Opción del modelo entidad</i>	69
<i>Tabla 3.21: tabla Solicitud del modelo entidad</i>	69
<i>Tabla 3.22: tabla bloque del modelo entidad</i>	70
<i>Tabla 3.23: tabla tipo ambiente del modelo entidad</i>	70
<i>Tabla 3.24: tabla ambiente del modelo entidad</i>	70
<i>Tabla 3.25: tabla dia habil del modelo entidad</i>	71
<i>Tabla 3.26: tabla hora habil del modelo entidad</i>	71

INDICE DE FIGURAS

<i>Figura 1. 1: Organigrama de la unidad de administración de ambientes</i>	3
<i>Figura 3.1: Interfaz de validación de usuario</i>	72
<i>Figura 3.2: Interfaz de usuario</i>	72
<i>Figura 3.3: Interfaz de bloque</i>	73
<i>Figura 3.5: Registro de solicitud</i>	73
<i>Figura 3.6: solicitudes pendientes</i>	74
<i>Figura 3.7: Interfaz de asignación de ambientes</i>	74

CAPITULO I
MARCO REFERENCIAL

1.1 ANTECEDENTES

La Universidad Amazónica de Pando fue creada en la gestión 1993, iniciando sus actividades académicas con dos carreras: Licenciatura en biología y Licenciatura en enfermería.

En agosto de 1996 se incorporó la carrera de informática a nivel técnico Superior, en el 2001 al 2007 se aprobó la apertura del área de ciencias económicas y financieras, área ciencias sociales con dos carreras y la unidad académica las piedras, en el municipio de Gonzalo Moreno. Actualmente, la Universidad Amazónica de Pando, cuenta con diferentes áreas: ciencias económicas y financieras, ciencias y tecnología, ciencias de la salud, ciencias sociales y humanísticas, ciencias jurídicas y políticas, ciencias biológicas y naturales.

Debido al crecimiento de la infraestructura del campus universitario de la Universidad Amazónica de Pando, se vio con la necesidad de llevar una administración a las aulas del campus dando origen a la *unidad de administración de ambientes del campus universitario*.

La unidad de administración de ambientes del campus universitario en el año 2010 desempeñaba sus funciones en un ambiente dentro de la biblioteca universitaria posteriormente en la gestión 2012 se vio con la necesidad de cambiar su ambiente por un ambiente más amplio para su mejor desempeño llevando un control frecuente del estado de las aulas y es así que actualmente cuenta con un ambiente propio.

El campus universitario de la Universidad Amazónica de Pando (UAP) tiene en la actualidad aproximadamente 6000 estudiantes distribuidos en las tres jornadas: matutina, vespertina y nocturna que deben ser repartidos en 7 Bloques con un total de 55 aulas o ambientes.

Las aulas son ambientes apropiados para el aprendizaje de los estudiantes, con infraestructura adecuada, medios didácticos multimedios de última generación.

A continuación se detalla los Bloques de aulas que actualmente están en funcionamiento en la UAP.

Bloques de Aulas		
Bloque	Aulas	Capacidad
A	1 - 4	100 a 150 estudiantes
B	1 - 6	100 estudiantes
C	1 - 8	40 estudiantes
D	1- 12	70 estudiantes
E	1 - 8	70 a 100 estudiantes
F	1 - 10	70 estudiantes
G	1 - 7	70 a 200 estudiantes
Total	7	55

Tabla 1.1: Especificación de Aulas

Fuente: *Unidad de Administración de Ambientes*

Esta unidad de administración de ambientes tiene bajo su control 41 aulas regulares de los cuales 14 aulas se encuentran ocupadas por direcciones de Area. Asumiendo como objetivo principal administrar la asignación de aulas de acuerdo a horarios y requerimientos solicitados con una anticipación de 48 horas máximo y mínimo de 24 horas para su respectivo procesamiento.

El proceso que se realiza para solicitar un aula es que cada solicitud de ambientes tenga los requerimientos necesarios y la autorización de una dirección, posteriormente esto se entrega a la unidad de administración de ambientes.

La asignación de aulas la lleva acabo el administrador de la unidad de acuerdo a horarios y aulas disponibles, esta información pasa hacer almacenada en un cuaderno de registros. Si no existen aulas disponibles para alguna hora en particular, se informa al solicitante, para que éste decida un nuevo horario.

Una solicitud de aula puede ser realizada por: vicerrectorado, direcciones académicas y direcciones de áreas como también la comunidad universitaria (docentes, estudiantes y administrativos) y personal externo a través de una dirección.

La unidad de administración de ambientes del campus universitario está organizada bajo la siguiente estructura:

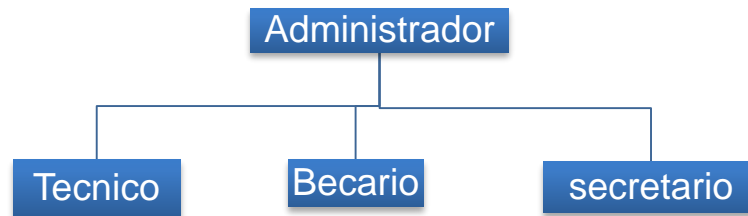


Figura 1. 1: Organigrama de la unidad de administración de ambientes
Fuente: Elaboración propia

1.2 PLANTEAMIENTO DEL PROBLEMA

Habiendo realizado un análisis se evidencio algunos problemas en la Unidad de administración de ambientes del campus universitario, como la falta de información oportuna y confiable, debido a que los registros se hacen de manera manual en cuadernos, pizarra acrílica y algunas veces usando Microsoft Excel una herramienta donde no necesariamente es adecuada para el uso y manejo de este tipo de Información. Además esta información almacenada en disco duro queda al riesgo de ser eliminada y deteriorada con el pasar del tiempo. Es por esta razón que con el presente proyecto se pretende eliminar toda esta deficiencia en cuanto al manejo y administración de las aulas del campus universitario. De lo descrito anteriormente podemos plantear el siguiente problema de Investigación

“Administración Deficiente de las Aulas del campus Universitario dependientes de la Unidad de Administración de Ambientes por la pérdida de información no sistematizada”

Generado como efectos choques de horario por la desorganización de la información, Retraso y demora en la toma de decisiones debido a la falta de información en tiempo real finalmente causando molestias a administrativos, docentes y estudiantes por la respuesta tardía a sus solicitudes.

1.3 OBJETIVOS

1.3.1 Objetivo general

Control eficiente de las aulas del campus Universitario mediante el desarrollo de un sistema informático bajo la metodología RUP. (Proceso Unificado Racional)”.

Implementar un sistema informático para el control de aulas del campus universitario de la UAP, utilizando la metodología Proceso Unificado Racional (RUP).

1.3.2 Objetivos Específicos

- Diagnosticar especificaciones de los requerimientos para el análisis del sistema.
- Diseñar el sistema especificando la base de datos que contemplen los módulos a desarrollar.
- Desarrollar módulos del sistema a partir de las especificaciones técnicas.
- Realizar pruebas al sistema mediante casos de prueba para verificar su correcto funcionamiento.

1.4 METODOLOGIA

Se utiliza como guía para el desarrollo del sistema informático metodología del Proceso Unificado Racional (RUP por sus siglas en ingles), basado en iteraciones, que se aplican sobre las siguientes fases. Inicio, elaboración, construcción, y transición. En cada una de las etapas, la metodología incorpora técnicas que utilizan herramientas de UML (lenguaje unificado de modelado) para modelar cada uno de los flujos de trabajo realizados durante la elaboración del sistema.

El proceso unificado racional, es una de las metodologías de construcción de software, dado sus características de conducción por casos de uso y orientados a objetos. RUP es la metodología estándar de la industria para la construcción completa del ciclo de ingeniería de software, tanto para sistemas tradicionales como para sistemas web.

A continuación se muestra en la tabla 1-2 a detalle las fases que se ejecuta y las herramientas con las que se desarrollan cada una de ellas de acuerdo a la metodología RUP.

ACTIVIDAD	OBJETIVO	TECNICOS	HERRAMIENTAS	RESULTADO
REQUERIMIENTOS	Capturar los requerimientos para identificar las funcionalidades del sistema con técnicas de búsqueda de información.	Cuestionario, entrevistas y visitas(técnicas de búsqueda de información)	<ul style="list-style-type: none"> - Grabadoras - filmadoras - documentos impresos etc. 	información
ANALISIS	Analizar la información obtenida para identificar las funcionalidades del sistema y obtener los objetos que lo complementan.	<ul style="list-style-type: none"> - Modelos de análisis. Técnicas de caso de uso. 	Diagrama de Casos de uso.	Modelo de análisis.
DISEÑO	Diseña la arquitectura a detalle y la macroarquitectura para implementar de forma eficiente las funcionalidades del sistema.	- Modelo de Diseño.	<ul style="list-style-type: none"> - Diagrama de clases. Diagrama de Secuencia. - Diagrama de Componentes. - Diagrama de Paquetes. 	Modelo de Diseño.
IMPLEMENTACION	Traducir el modelo de diseño en un lenguaje de programación para obtener el sistema informático.	P.O.O	<ul style="list-style-type: none"> - JAVA - NETBEANS 	Modelo implementación. Sistema informático con las funcionalidades
PRUEBA	Probar el sistema informático para verificar el cumplimiento de los requerimientos iniciales	<ul style="list-style-type: none"> • Prueba Unitaria. • Prueba Integración. Prueba del Sistema. 	<ul style="list-style-type: none"> • JUnit 	Modelo de pruebas. Plan de pruebas. Diseño de los Casos de Uso. <ul style="list-style-type: none"> ▪ Casos de Prueba. ▪ Resultados de las pruebas.

Tabla 1.2: Metodología RUP

Fuente: Elaboración propia

1.5 ALCANCES

El sistema para control de aulas, es capaz de proveer información que maneja el registro de asignación de aulas a la hora de realizar una asignación mediante los siguientes módulos:

❖ **Módulo de administración.**

- Control de acceso al sistema
- Altas, bajas y modificaciones de Usuario
- Altas, bajas y modificaciones de Datos Persona
- Asignación de usuarios y roles

❖ **Modulo Bloque**

- Altas, bajas y modificaciones de Bloques
- Altas, bajas y modificaciones de Ambientes
- Altas, bajas y modificaciones de Tipo Ambiente
- Altas, bajas y modificaciones de horarios
- Habilitación de ambientes
- Reporte de Ambientes libres
- Reporte de ambientes en Uso
- Reporte de ambientes en mantenimiento

❖ **Módulo de solicitud**

- Altas, bajas y modificaciones de solicitudes de ambientes
 - Asignación de ambientes
 - Asignación de horario
 - Reportes de solicitud
-
- El sistema estará en red de área local (en la unidad de administración de ambientes ubicada en el bloque F del campus universitario de la universidad amazónica de pando) y funcionará vía intranet.

CAPITULO II
MARCO METODOLOGICO

2.1 METODOLOGIA RACIONAL UNIFIED PROCESS (Proceso Unificado Racional)

El proceso unificado de racional es un proceso de ingeniería del software que proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en una organización de desarrollo su propósito es asegurar la producción de software de alta calidad que se ajuste a las necesidades de los usuarios finales (PRESMAN, 2005)

2.1.1 ESTRUCTURA DEL RUP

RUP es una metodología en la que se describe quien, como, que, en que tiempo y que actividades se van a desarrollar en el proyecto, el quien representa los distintos roles que puede desempeñar un individuo en la organización, el cómo se refiere a la unidad de trabajo que se asigne a un trabajador y el que en la pieza de información utilizada por un proceso. Para el desarrollo satisfactorio de este proceso, es necesario cumplir con ciertas etapas, las mismas que en conjunto determinaran el ciclo de vida y el éxito de la aplicación. RUP hace uso de cuatro etapas en su metodología descrita de la siguiente manera: una fase de elaboración, una fase de construcción y una última de transición. Dentro de cada una de ellas se realizan varias iteraciones dependiendo de la dimensión del proyecto.

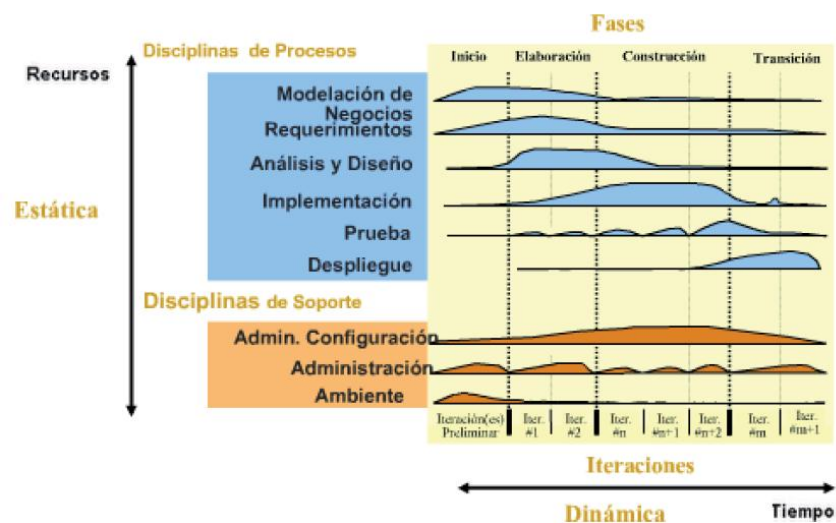


Figura 2.1: Estructura del RUP
Fuente: (Torossi, 2010)

La figura 2.1 describe la estructura del RUP en dos dimensiones, el eje horizontal representa el tiempo y proyecta una visión dinámica del proceso ya que muestra aspectos como las fases del ciclo de vida de un software RUP, metas e iteraciones; mientras que el eje vertical representa los diferentes recursos proyectados desde una vista estática, en estos se detallan las diferentes actividades el personal encargado de las mismas, los roles, artefactos, disciplinas y los flujos de trabajo.

➤ **Visión estática de la metodología RUP**

La visión estática de un proceso RUP permite describir los siguientes elementos:

- ✓ **Roles.**- entre los más comunes se encuentran los roles de gestión, de administración, de analista de sistemas y diseñador de pruebas.
- ✓ **Actividades.**- representa el trabajo que debe desempeñar cada rol.
- ✓ **Artefactos.**- son elementos tangibles de entrada y salida de las actividades, es decir, cosas que el proyecto produce o usa para componer un producto final. Los artefactos en las fases de inicio están representados por una especificación de requerimientos, en la fase de elaboración por los diagramas de caso de uso y en la fase de construcción por los diagramas de clase, de secuencia, de estados y modelos E-R.
- ✓ **Flujos de trabajo.**- constituyen la secuencia de actividades que producen resultados visibles.
- ✓ **Disciplinas.**- sirven para organizar las actividades del proceso, comprende 6 disciplinas de proceso y 3 de soporte.

En las de proceso tenemos: modelación de negocios, requerimientos, análisis y diseño, implementación, prueba y desarrollo.

En las de soporte tenemos: gestión de configuración y cambio, gestión de proyecto y entorno.

➤ **Visión dinámica de la metodología RUP**

La visión dinámica de un proceso RUP permite apreciar su ciclo de vida mediante una descripción de procesos que son iterativos e incrementales, siempre respaldados por hitos

que determinan su avance y permiten planear su continuidad en la base a posibles cambios en su estructura.

Las cuatro fases que describen un proceso RUP son las siguientes:

- ✓ **Inicio.**- en esta fase se obtiene una misión inicial del producto así como su alcance, además se identifican los principales casos de uso.
- ✓ **Elaboración.**- en esta fase se hace una planificación de las actividades y del equipo de trabajo del proyecto, también se identifican las necesidades y el diseño de la arquitectura.
- ✓ **Construcción.**- comprende el desarrollo mismo del producto hasta la entrega al usuario final.
- ✓ **Transición.**- esta fase comprende la instalación del producto a los usuarios y la formación de los mismos, en ocasiones suelen surgir nuevos requisitos para el desarrollo.

2.1.2 CICLO DE VIDA DE LA METODOLOGÍA RUP

De acuerdo a lo anteriormente mencionado las fases pertenecen a la parte dinámica de la metodología, el ciclo de vida del software del RUP se descompone en cuatro fases secuenciales. (PRESMAN, 2005)

✓ **Fase de inicio**

La fase de inicio trata de responder a estas preguntas más comunes que se debe realizar antes de iniciar un proyecto, sin embargo no se pretende una estimación precisa o la captura de todos los requisitos más bien se trata de explorar el problema lo justo para decidir si va a continuar el proyecto, uno de los objetivos principales de esta fase es establecer el ámbito del proyecto y encontrar los escenarios básicos que definen la funcionalidad mediante el modelado de caso de uso.

✓ **Fase de colaboración**

El propósito de la fase de colaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan de proyecto y eliminar los mayores riesgos, en esta fase se construye un prototipo de la arquitectura, que debe evolucionar en

iteraciones sucesivas hasta convertirse en el sistema final. Cuando termina esta fase se llega al punto de no retorno del proyecto es a partir de ese momento pasamos de las relativamente ligeras y de poco riesgo dos primeras fases, a afrontar la fase de construcción, es por esto que la fase de colaboración es de gran importancia.

✓ **Fase de construcción**

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones, durante esta fase, todos los componentes, características y requisitos que hayan sido elaborados hasta ahora, serán implementados, integrados, obteniéndose una versión del producto que se pueda poner en manos de los usuarios.

✓ **Fase de transición**

El objetivo de esta fase es poner el producto en manos de los usuarios finales, para lo que típicamente se requerirá desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto y en general, tareas relacionadas con el ajuste, configuración, instalación y usabilidad del producto.

2.2 HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO

2.2.1 LENGUAJE UNIFICADO DE MODELADO (UML)

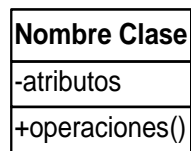
UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software de una forma consistente demostrando su utilidad en sistemas de cierto tamaño (Ivar J., Grady B., James R., 2005)

➤ **Bloques de construcción de UML**

Son bloques básicos de construcción orientada a objetos de UML. Se utilizan para escribir modelos bien formados. Existen cuatro tipos de bloques a los que llamaremos elementos: elementos estructurales, elementos de comportamiento, elementos de agrupación, elementos de anotación (Vega, 2009)

➤ **Elementos estructurales.** Son los nombres de los modelos UML. En su mayoría son las partes estáticas de un modelo, y representan cosas que son conceptuales o materiales. En total hay 6 tipos de elementos estructurales. (Ivar J., Grady B., James R., 2005)

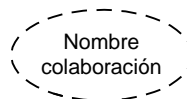
✓ **Clases:** describe un conjunto de objetos con características y comportamiento idéntico (Atributos, operaciones, relaciones y semántica)



✓ **Interfaz:** es una colección de operaciones que especifican un servicio de una clase o componente, describe el comportamiento visible externamente de ese elemento.



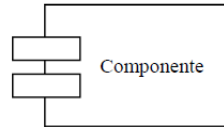
✓ **Colaboración:** una colaboración define una iteración y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos.



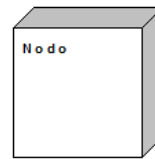
✓ **Caso de uso:** denota un requerimiento solucionado por el sistema es una descripción de un conjunto de secuencias de acciones que un sistema realiza y que produce un resultado.



- ✓ **Componente:** un componente es una parte física y reemplazable de un sistema que conforma un conjunto de interfaces y proporciona la implementación de dicho conjunto.

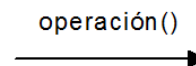


- ✓ **Nodo:** es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que por lo general dispone de algo de memoria y, con frecuencia, capacidad de procesamiento. Un conjunto de comportamientos puede residir en un nodo y puede también migrar de un nodo a otro.

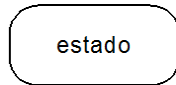


- **Elementos de comportamiento:** Son las partes dinámicas de los elementos UML. Representan comportamiento en el tiempo y el espacio. En total existen dos tipos principales de elementos de comportamiento.

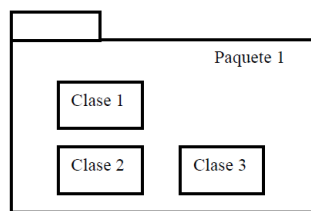
- ✓ **Interacción:** es un comportamiento que comprende un conjunto de mensajes intercambiados entre un conjunto de objetos, dentro de un contexto particular, para alcanzar un propósito específico.



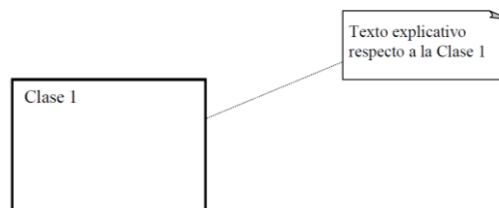
- ✓ **Máquina de estados:** una máquina de estados es un comportamiento que especifica las secuencias de estado por las que pasa un objeto o una interacción durante su vida en respuesta a eventos, junto con sus reacciones a estos eventos.



- **Elementos de agrupación:** son las partes organizativas de los modelos UML
 - ✓ **Paquete:** un paquete es un mecanismo de propósito general para organizar elementos en grupos. Los elementos estructurales, los elementos de comportamiento, e incluso otros elementos de agrupación pueden incluirse en un paquete.



- **Elementos de anotación:** son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clarificar y hacer observaciones sobre cualquier elemento de un modelo. Hay un tipo principal de elemento de anotación llamada nota.
 - ✓ **Nota:** Es frecuente que alguna parte del diagrama no presente una clara explicación del porqué está allí o la manera en que trabaja. Cuando éste sea el caso, la nota UML será útil. La nota tiene una esquina doblada y se adjunta al elemento del diagrama conectándolo mediante una línea punteada.



- **Relaciones:** las relaciones en UML son bloques básicos de construcción para relaciones, se utilizan para escribir modelos bien formados. Existen cuatro tipos de relaciones en UML.

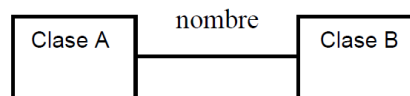
- ✓ **Dependencia:** es una relación semántica entre dos elementos, en la cual un cambio a un elemento (llamado elemento independiente) puede afectar a la semántica del otro elemento (elemento dependiente).
- ✓ **Composición:** Composición es un tipo especial de agregación que denota una fuerte posesión de la Clase “Todo”, a la Clase “Parte”. Se grafica con un rombo diamante relleno contra la clase que representa el todo.



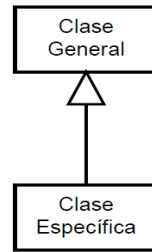
- ✓ **Agregación:** La agregación es una relación en la que la clase “Todo” juega un rol más importante que la Clase "Parte", pero las dos clases no son dependientes una de otra. Se grafica con un rombo diamante vacío contra la Clase “Todo”.



- ✓ **Asociación:** Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos.

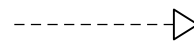


- ✓ **Generalización:** una generalización es una relación de especialización y/o generalización en la que los objetos del elemento especializado, o hijo, puede sustituir a los objetos del elemento general (padre). Así, el hijo comparte el comportamiento de la estructura del padre.



- ✓ **Realización:** Es una relación semántica entre clasificadores, en donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá.

2.2.2 DIAGRAMAS DE UML



Los diagramas representan gráficamente un conjunto de elementos con sus relaciones, el lenguaje cuenta con nueve diagramas para visualizar el sistema desde varias perspectivas las cuales son las siguientes:

- **Diagrama de casos de uso:** Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario.
- **Diagrama de clases:** muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y como se relacionan entre sí.
- **Diagrama de objetos:** Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases.
- **Diagramas de comportamiento:** dentro de estos diagramas se encuentran:
 - **Diagrama de estados:** modela el comportamiento del sistema de acuerdo con eventos.
 - **Diagrama de actividades:** simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades. También se pueden utilizar caminos verticales para mostrar los responsables de cada actividad.

- **Diagramas de interacción:** estos diagramas a su vez se dividen en dos tipos de diagramas que enfatizan.
 - **Diagrama de secuencia:** enfatiza la interacción entre los objetos y los mensajes que intercambian entre si junto con el orden temporal de los mismos.
 - **Diagrama de colaboración:** igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- **Diagrama de componentes:** muestran la organización y las dependencias entre un conjunto de componentes.
- **Diagrama de despliegue:** muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo.

Para la elaboración del modelado del sistema se utiliza los diagramas de casos de uso, diagrama de clases, diagrama de estados y diagrama de secuencia.

2.3 LENGUAJE DE PROGRAMACION

2.3.1 JAVA

Java es un lenguaje simple. Orientado al objeto, distribuido, interpretado, solido, seguro, de arquitectura neutral, portable, de multihilos y dinámico según (oracle, s.f.)

- **Orientado a objeto:** java da buen soporte a las técnicas de desarrollo OOP y en resumen a la utilización de componentes de software.
- **Distribuido:** java se ha diseñado para trabajar en ambientes de redes y contiene una gran biblioteca de clases para la utilización del protocolo TC/IP, incluyendo HTTP y FTP.
- **Interpretado:** el compilador java traduce cada fichero fuente de clase a código de bytes, que puede ser interpretado por todas las máquinas que den soporte a un visualizador que funcione con java.
- **Solido:** El código java no se quiebra fácilmente ante errores de programación.

- **Seguro:** como java suele trabajar en ambientes de redes el tema de seguridad debe interesar de sobremanera, las mismas características antes descritas que evitan la corrupción de código evitan la manipulación.
- **Alto desempeño:** al ser código interpretado la ejecución no es tan rápida como el código compilado para una plataforma particular. El compilador Java suele ofrecer la posibilidad de compilar Bytecode en código máquina de determinadas plataformas y según Sun este código resulta de una eficacia similar a compilaciones de C y C++.
- **Multihilos:** java puede aplicarse a la realización de aplicaciones en las que ocurra más de una cosa a la vez.
- **Dinámico:** al contrario de C++ que exige se compile de nuevo la aplicación al cambiar una clase madre Java utiliza un sistema de interfaces que permite aligerar esta dependencia.

2.3.2 HTML

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de

actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

2.3.3 JAVA SCRIPT

JavaScript es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

2.3.4 HOJAS DE ESTILO CASCADING STYLE SHEETS (CSS)

Hoja de estilo en cascada o CSS (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo «style».

2.3.5 IREPORT

iReport es una poderosa herramienta OpenSource y basado completamente en Java que facilita la construcción y el diseño de informes de manera visual: es capaz de utilizar datos procedentes de distintas fuentes, permite generar graficas (histogramas, grafico de burbujas, gráficos lineales, etc.) gracias a que integra JFreeChart, además es posible imprimir o exportar los reportes en varios formatos (HTML, PDF, Excel y Word).

CARACTERISTICAS DE IREPORT

- Escrito en JAVA
- OPENSOURCE
- Soporta JDBC.
- Soporta JavaBeans como orígenes de datos
- Incluye asistente para generar reportes y subreportes
- Facilidad de instalación

2.4 SERVIDOR DE APLICACIONES WEB

2.4.1 APACHE TOMCAT

Apache Tomcat es un servidor de código abierto, potente que ofrece un servicio estable y sencillo de mantener y configurar. El servidor apache tomcat se desarrolla dentro del proyecto HTTP server.

Para entender lo que es apache tomcat, primeramente se define lo que es un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir páginas web con todos sus elementos.

Características del servidor apache

Se destaca las siguientes características de apache tomcat.

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Apache tomcat es un servidor altamente configurable de diseño modular.
- Apache tomcat es una tecnología gratuita de código fuente abierta.
- Posee diversos módulos que permiten incorporarle nuevas funcionalidades.

Estructura

La jerarquía de directorios de instalación de tomcat incluye:

- bin – arranque, cierre, y otros scripst y ejecutables.
- Common – clases comunes que pueden utilizar Catalina y las aplicaciones web.
- Conf – ficheros XML y los correspondientes de DTD para la configuración de tomcat.
- Logg – logs de catalina y de las aplicaciones.
- Server – clases utilizadas solamente por catalina.
- Shared – clases compartidas por todas las aplicaciones web.
- Webapps – directorio que contiene las aplicaciones web.
- Work – almacenamiento temporal de ficheros y directorios.

2.5 BASE DE DATOS

2.5.1 POSTGRESQL

PostgreSQL es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto de y el sistema continuará funcionando. (LIB5)

Características de PostgreSQL

- Object-Relational. PostgreSQL busca los datos con un modelo objeto-relacional y es capaz de manejar rutinas y reglas complejas concurrencia multiversión, soporte a multiusuario, transacciones, optimización de consultas, herencia y arreglos.
- Alta extensibilidad. PostgreSQL soporta operadores definidos por el usuario.
- Soporte comprensivo de SQL.
- Integridad referencial.
- Consultas complejas.
- Triggers
- Vistas.
- Integridad transaccional.
- Control de concurrencia multiversión. (LIB)

CAPITULO III
MARCO APLICATIVO

3.1 DESARROLLO DEL SISTEMA

Durante este capítulo se presentan los resultados obtenidos del producto de desarrollo de software para el control de aulas del campus universitario, desarrollando los modelos y diagramas de las diferentes disciplinas de la metodología RUP con la ayuda del Lenguaje de Modelado Unificado (UML) para su mejor entendimiento.

3.1.1 FASE I: FASE DE INICIO

De acuerdo a la metodología y siguiendo los pasos de las disciplinas se realizaron entrevistas y cuestionarios a la unidad de administración de ambientes del campus universitario con la finalidad de obtener el flujo de información para el control de aulas y para adquirir los requerimientos del sistema.

3.1.1.1 ANALISIS DE FLUJO DE INFORMACION

Para conseguir los objetivos y obtener un mejor panorama del proceso que sigue una solicitud de aula e identificar los actores, se organizó las actividades que realiza la unidad por medio de un conjunto de procesos de negocios.

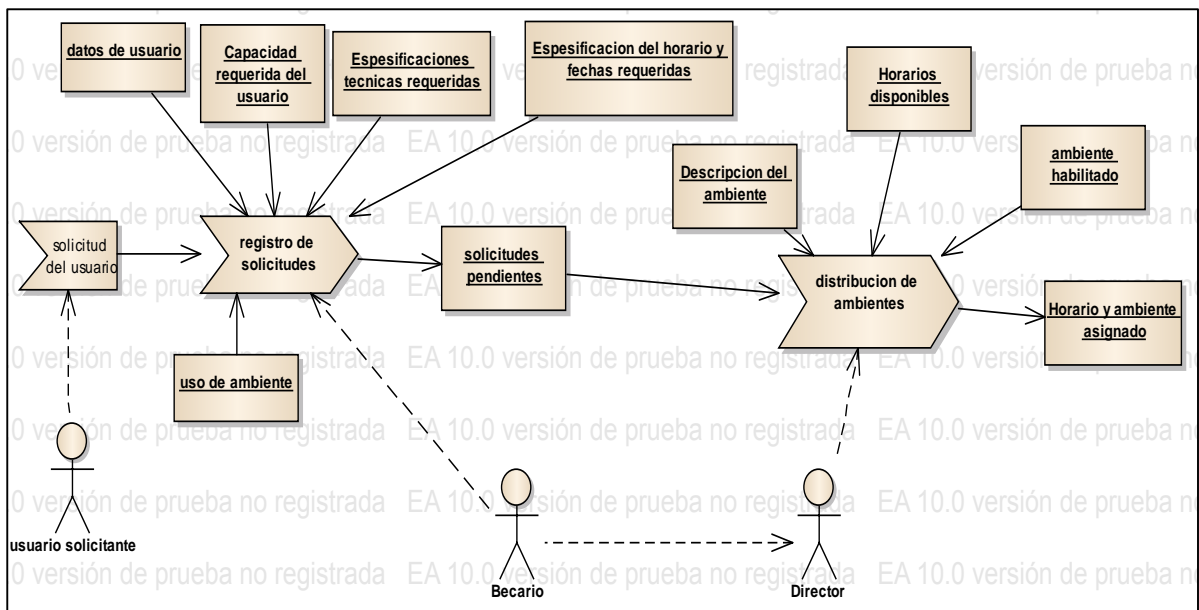


Diagrama 3.1: Modelo de proceso de negocios asignación de aulas

Fuente: Elaboración propia

En el modelo del proceso de negocio que se menciona anteriormente demuestra las actividades de trabajo que realiza la unidad para la asignación de aulas, comenzando desde que llega la solicitud a la unidad y terminando hasta realizar la asignación del aula correspondiente, manteniendo un registro de solicitud y de distribución de aulas.

A continuación en la Diagrama 3.2 se describe mediante el diagrama de actividad el flujo o proceso que se realiza para la asignación de aulas iniciando desde que la unidad solicitante interna o externa solicita un ambiente y finalizando hasta que la solicitud haya sido atendida.

Proceso de la Asignación de Aula

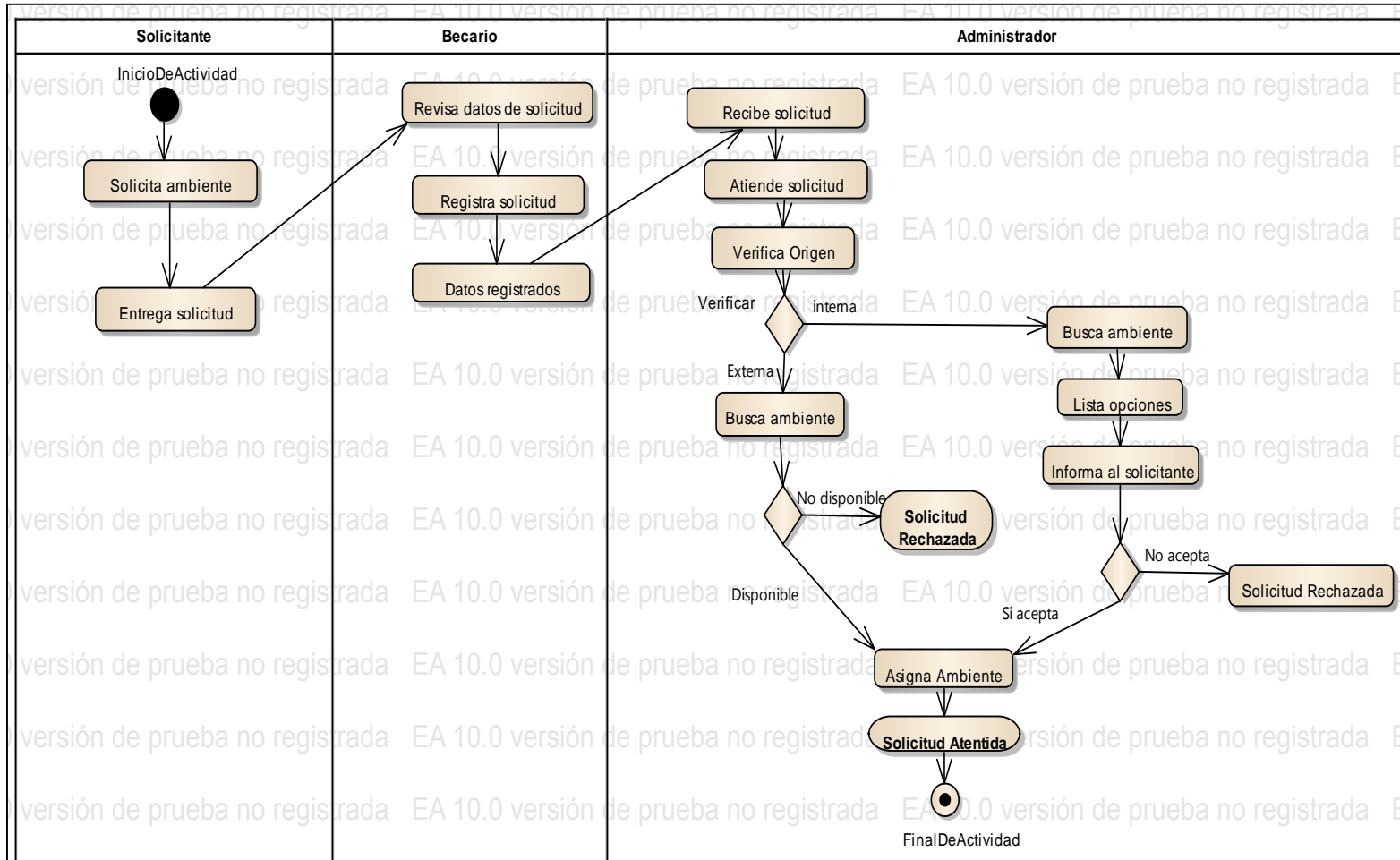


Diagrama 3.2: diagrama de Actividades: Asignación de Aulas

Fuente: Elaboración propia

3.1.1.2 ANALISIS DE REQUERIMIENTOS

Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema de software para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto.

3.1.1.2.1 REQUERIMIENTOS DEL SISTEMA SAAC-UAP

Los requerimientos del sistema son los que se establecen en la siguiente tabla:

Nº	Requerimientos
R1	Administrar usuarios del sistema
R2	Administrar bloque
R3	Administrar ambientes
R4	Administrar solicitud

Tabla 3.3: *Requerimientos del sistema SAAC-UAP.*

Fuente: Elaboración propia

Requerimiento de usuario

El sistema debe verificar que el acceso del mismo ha sido correcto y que la conexión de la base de datos se encuentra disponible.

El sistema recibirá los datos de clientes y solicitudes almacenándolos en la base de datos para futuras consultas.

- **Log del sistema**

Se debe registrar todos los sucesos que ocurre en el sistema a fin de reconstruir todas las actividades realizadas en el sistema pero siempre garantizando la privacidad del cliente.

- **Para usuario:**

Es deseable que el login de usuario se realice en la página de inicio, desde un formulario que solicite el nombre de usuario, contraseña y un botón "Entrar" que inicie la verificación de datos enviados a la Base de Datos

- **Para administradores:**

El sistema deberá brindarle al administrador la posibilidad de iniciar sesión.

3.1.1.2.2 REQUERIMIENTOS NO FUNCIONALES.

Los requisitos no funcionales describen los atributos que deben tener el software una vez construido (portabilidad, eficiencia, confiabilidad, robustez, rendimiento, etc.), actúan restringiendo las alternativas de la solución.

Nº	Requisito	Descripción
1	Interfaz para el usuario	Diseño sencillo, una interfaz simple de usar e interactiva para que al usuario le sea fácil el trabajo con el sistema.
2	Software	<ul style="list-style-type: none"> • La aplicación se hospeda en un servidor web Apache Tomcat. • El servidor de base de datos debe ser PostgreSQL. • Todas las maquinas clientes deben tener un navegador instalado, preferentemente Mozilla FireFox. Netcape y Google Chrome (versiones recientes). • Además instalado el Flash Player versión 6.0 o superior para visualizar el banner superior.
3	Manejo	La aplicación se podrá ejecutar en la mayoría de los sistemas operativos tales como Microsoft Windows XP/7 y Linux

Tabla 3.4: Requisitos no funcionales.

Fuente: *Elaboración propia*

3.1.1.3 IDENTIFICACIÓN DE ACTORES

Siguiendo metodología y durante la captura de requisitos se identificaron a los siguientes actores:




Actores	Descripción
 Adminitrador	Administrador: Es responsable de la unidad de administración del campus.
 Becarios	Becarios: Realiza el control de aulas en uso y aulas desocupadas para posteriormente cerrarlas.
 Secretario	Secretaria: Realiza la tarea de recepcionar solicitudes de las distintas unidades e instituciones que requieren ambientes.

Tabla 3.5: Identificación de Actores

Fuente: Elaboración propia

Habiendo realizado el análisis de flujo de la información y la identificación de los actores el siguiente paso es la identificación de diagrama de caso de uso los cuales describen la funcionalidad del sistema a partir de los requerimientos de los usuarios del sistema.

3.1.2 FASE II: FASE DE ELABORACION

El principal objetivo de la fase de elaboraciones formular la línea base de la arquitectura. Esto implica desarrollar alrededor del 80 por ciento de los casos de uso y abordar los riesgos que interfieran en la consecución de este objetivo. En esta fase acrecentaremos el entorno de desarrollo, no solo para llevar a cabo las actividades de esta fase sino para prepararnos para la fase de construcción. Hacia el final de esta fase habremos acumulado la información necesaria para planificar la fase de construcción. También en ese momento tendremos la

información suficiente para realizar un análisis de negocio fiable, trabajo que comenzamos durante la fase de inicio.

A continuación se procede a elaborar los siguientes diagramas de acuerdo a la identificación de las necesidades planteadas en la lista de requisito: Diagrama de casos de uso, diagrama de clases y diagrama de secuencias, (Descripción de los diagramas de casos de uso).

3.1.2.1 DIAGRAMA DE CASOS DE USO

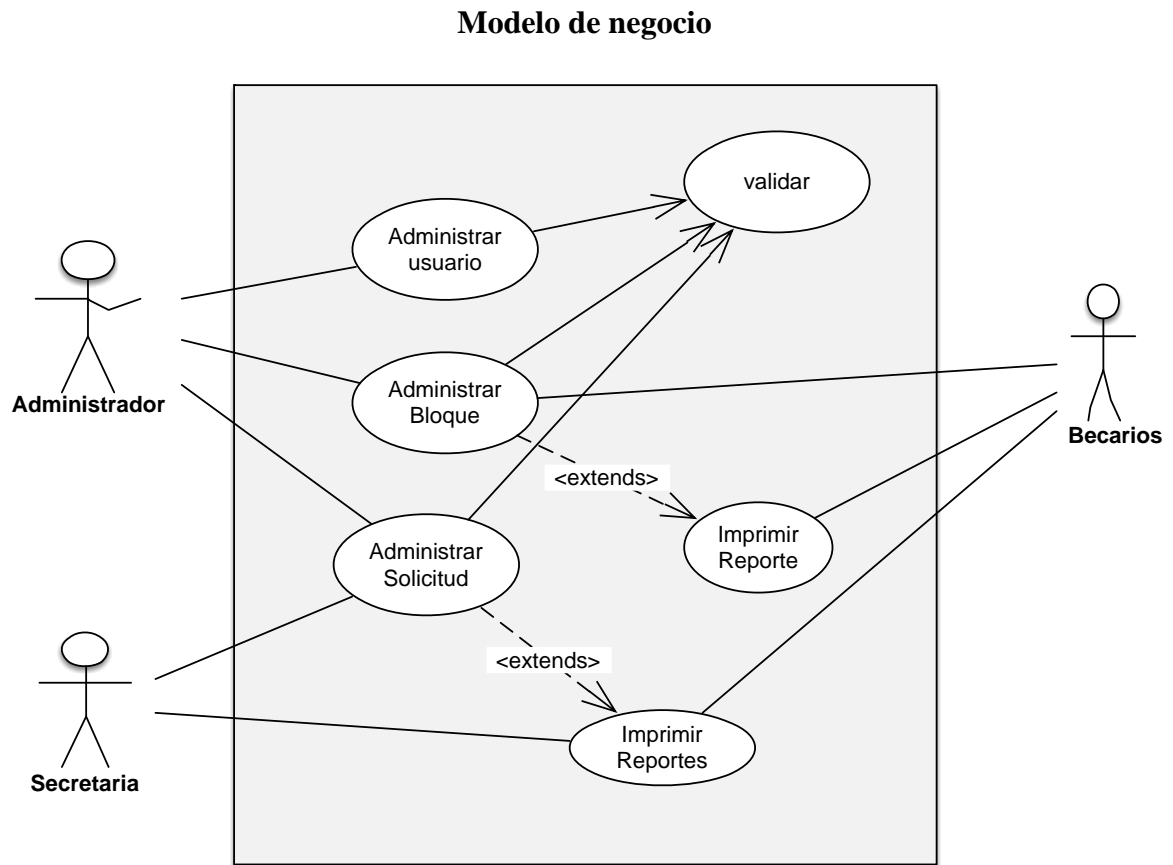


Diagrama 3.3: Modelo de negocios
Actor: Administrador, secretario, becarios
Fuente: Elaboración propia

✓ **Catálogo de caso de uso de negocio**

Caso de uso	Actor	Descripción
Administrar usuario	Administrador	Permite crear, eliminar, listar y modificar los usuarios que utilizaran el sistema, los datos personales, la opción y el rol para cada usuario.
Administrar Bloque	Administrador/Becarios	Permite crear, eliminar, modificar los parámetros que se utilizarán en la creación de bloques, como aula, tipo de aula y horario también genera reportes de aula en uso y en reparación.
Administrador Solicitud	Administrador /Secretaria	Permite crear, eliminar, modificar los parámetros que se utilizarán en el registro de Solicitud hacienda la asignación de aulas en horas y capacidad solicitada, también genera reportes de solicitudes pendientes, atendidas y rechazadas

Tabla 3.6: Catalogo de caso de uso de negocio

Fuente: *Elaboración propia*

Identificación de Usuarios

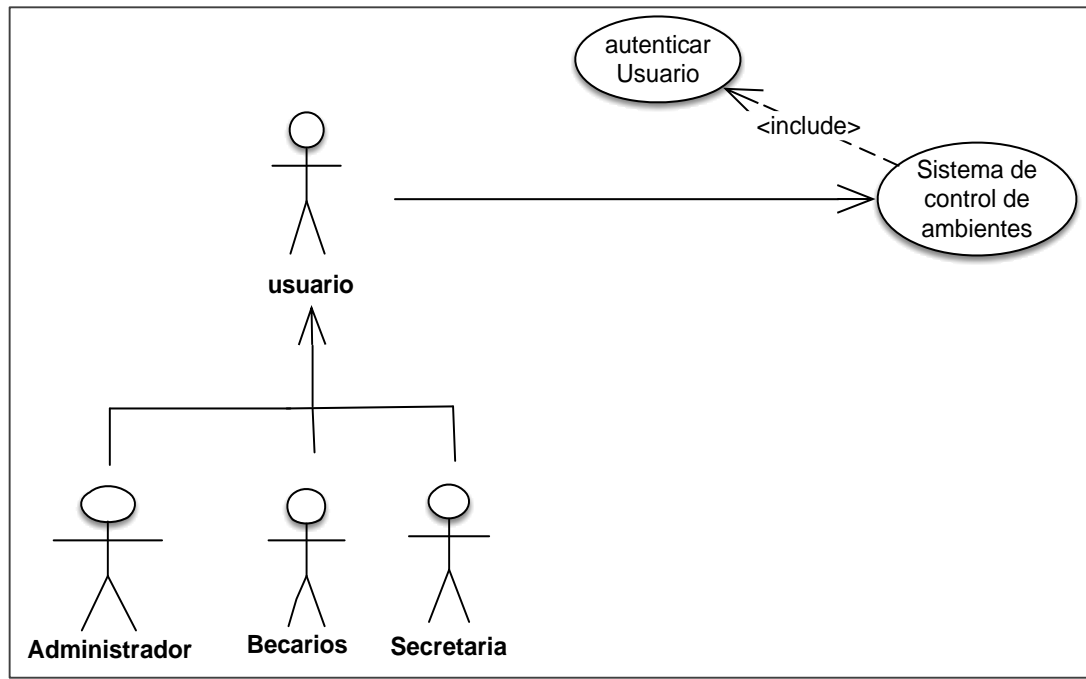


Diagrama 3.4: identificación de usuarios del sistema: Autenticar usuario

Fuente: Elaboración propia

3.1.2.2 MODELO DE CASOS DE USO DEL SISTEMA

Diagrama completo del sistema

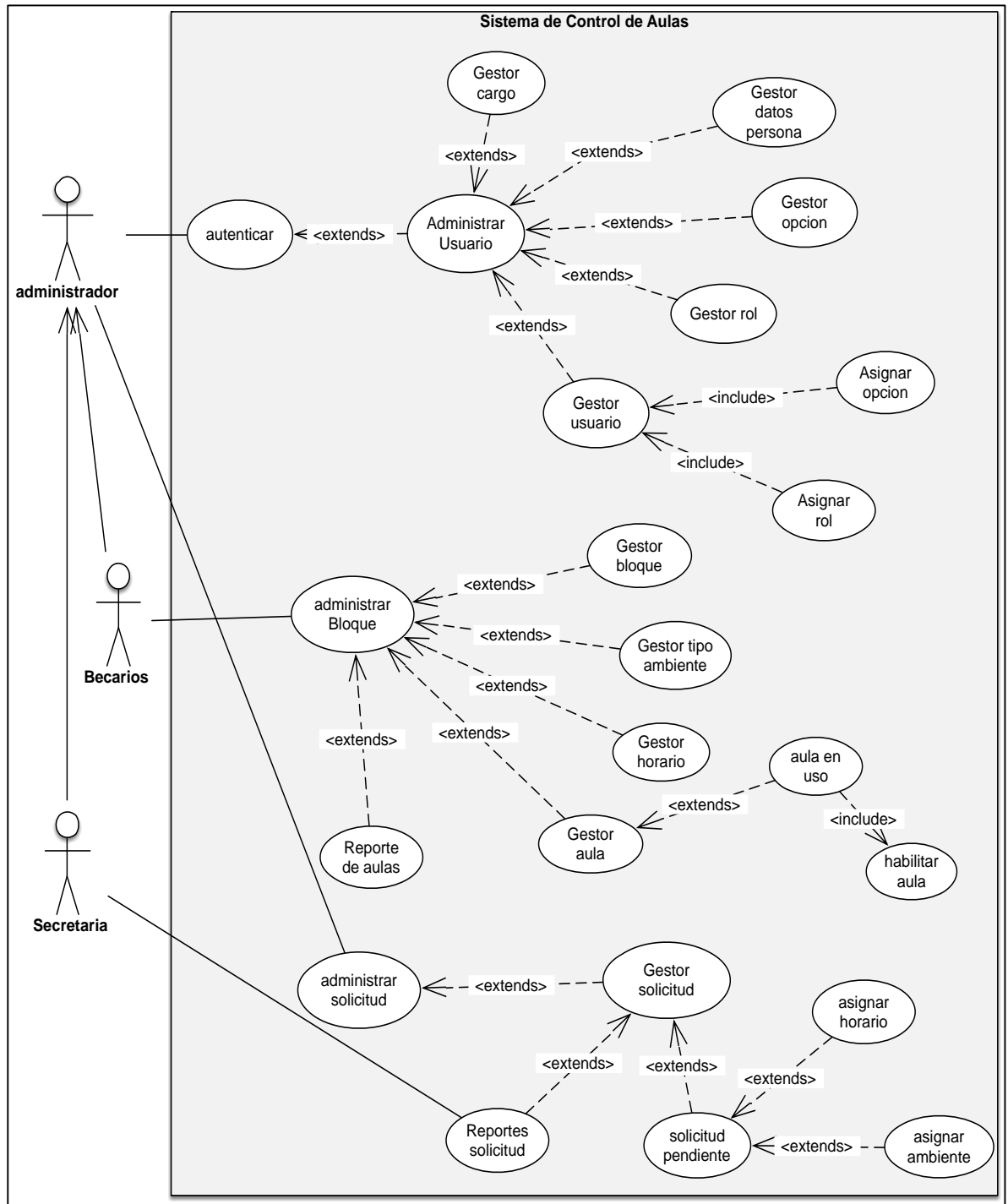


Diagrama 3.5: Modelo de casos de uso del sistema

Actor: Administrador, secretario, becarios

Fuente: Elaboración propia

3.1.2.2.1 Modulo Administrar usuario

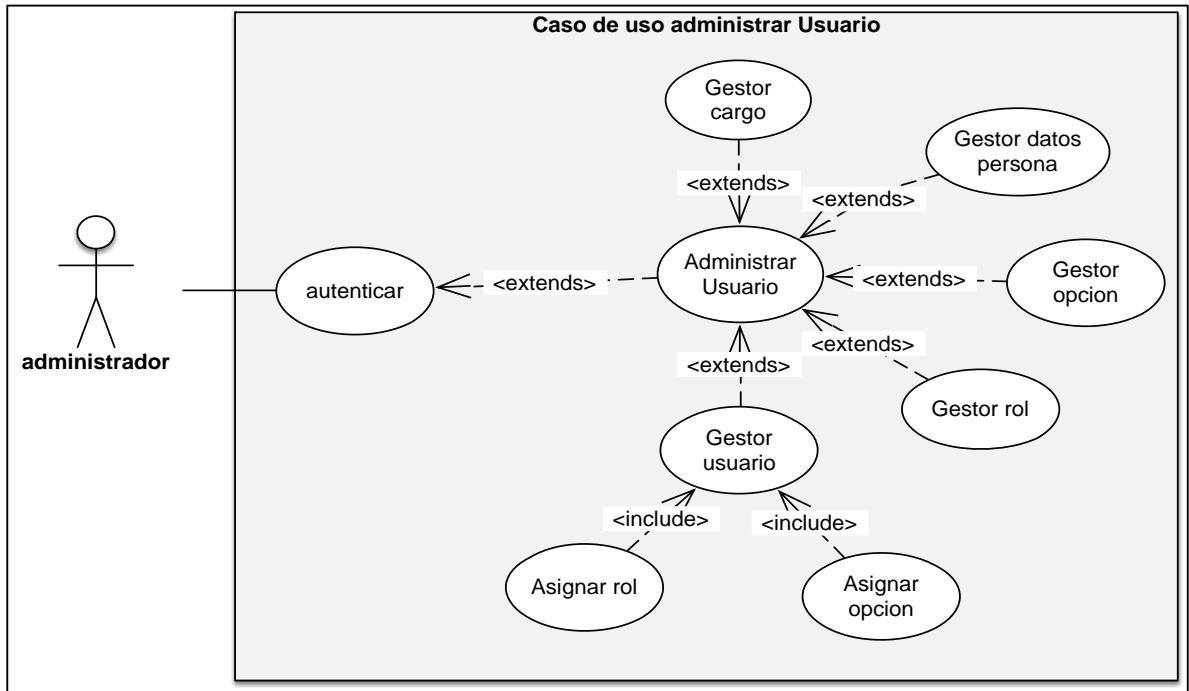


Diagrama 3.6: Caso de uso del módulo administrar usuario

Actor: Administrador

Fuente: Elaboración propia

El diagrama de caso de uso del módulo administrar usuario comprende con 5 casos de uso dentro de este módulo que son los que se describen a continuación.

❖ CU.01: Autenticar

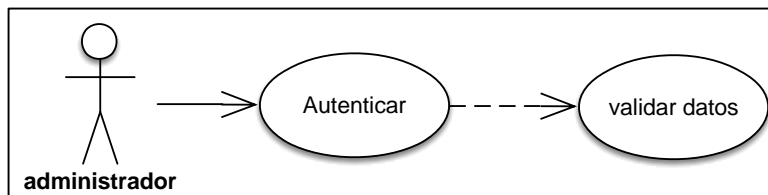


Diagrama 3.7: Caso de uso Gestor datos personales

Fuente: Elaboración propia

Autenticar	
Nombre:	Autenticar usuario
Descripción general	El usuario provee su nombre y contraseña para iniciar sesión en el sistema
Actores	Administrador
Precondiciones	No haber iniciado sesión.

	Estar registrado en el sistema. Estar habilitado. Ingresar nombre de usuario correcto. Ingresar contraseña correcta																											
Flujo normal:																												
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Inicia el sistema</td> <td></td> </tr> <tr> <td></td> <td>Ingresar nombre de usuario y contraseña</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td></td> <td></td> <td>Recupera datos de usuario en la base de datos</td> </tr> <tr> <td></td> <td></td> <td>Autentica al usuario</td> </tr> <tr> <td></td> <td></td> <td>Verifica estado de usuario</td> </tr> <tr> <td></td> <td></td> <td>Registra nueva sesión</td> </tr> <tr> <td></td> <td></td> <td>Redirección a la página principal de la aplicación.</td> </tr> </tbody> </table>		Administrador	Sistema	1	Inicia el sistema			Ingresar nombre de usuario y contraseña				Valida datos ingresados			Recupera datos de usuario en la base de datos			Autentica al usuario			Verifica estado de usuario			Registra nueva sesión			Redirección a la página principal de la aplicación.
	Administrador	Sistema																										
1	Inicia el sistema																											
	Ingresar nombre de usuario y contraseña																											
		Valida datos ingresados																										
		Recupera datos de usuario en la base de datos																										
		Autentica al usuario																										
		Verifica estado de usuario																										
		Registra nueva sesión																										
		Redirección a la página principal de la aplicación.																										
Excepciones																												
	<ol style="list-style-type: none"> 1. Si uno o todos los campos están vacíos, regresar a la página de logeo y mostrara mensaje de error. 2. Se recuperan nombre de usuario, contraseña, estado de usuario, rol y privilegios. 3. Autenticar usuario 4. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 																											

Tabla 3.7: Descripción del caso de uso autenticar

Fuente: Elaboración propia

❖ CU.02: Gestor datos personales



Diagrama 3.8: Caso de uso Gestor datos personales

Fuente: Elaboración propia

Gestor datos Persona													
Nombre:	Registrar datos persona												
Descripción general	Se desea Registrar una persona												
Actores	administrador												
Precondiciones	Iniciar sesión												
Flujo normal:													
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción registrar persona</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra el formulario de registro persona</td> </tr> <tr> <td></td> <td>Ingresar datos de formulario</td> <td></td> </tr> </tbody> </table>		Administrador	Sistema	1	Elige la opción registrar persona				Muestra el formulario de registro persona		Ingresar datos de formulario	
	Administrador	Sistema											
1	Elige la opción registrar persona												
		Muestra el formulario de registro persona											
	Ingresar datos de formulario												

		Valida datos	
		Almacena datos	
Excepciones			
5.	Validar datos		
6.	Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de datos persona		
7.	Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios.		
8.	Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo.		

Nombre:	Modificar datos persona		
Descripción general	Se desea Modificar datos de una persona		
Actores	Administrador		
Precondiciones	Iniciar sesión Los datos deben estar registrado en la base de datos		
Flujo normal:			
	Administrador	Sistema	
1	Selecciona la opción "listar datos persona"		
	Busca o selecciona los datos		
	Elige la opción "Editar"		
		Muestra detalles del registro datos persona	
	Actualiza formulario de modificación seleccionando "Validar"		
		Valida datos ingresados	
		Actualiza los datos en la base de datos	
Excepciones			
1.	Validar datos		
2.	Si no cumple con las especificaciones requeridas, despliega mensaje de error.		

Nombre:	Lista datos persona		
Descripción general	Se lista datos de personas registradas		
Actores	Administrador		
Precondiciones	Iniciar sesión Tener registrado el personas		
Flujo normal:			
	Administrador	Sistema	
1	Elige la opción de listar datos persona		
		Consulta base de datos	
		Lista resultados	
Excepciones			
1.	Lista resultados		
2.	Si no existe datos, se muestra el registro de listado vacío.		

Tabla 3.8: Descripción del caso de uso Gestor datos personales

Fuente: Elaboración propia

❖ CU.02: Gestor Cargo

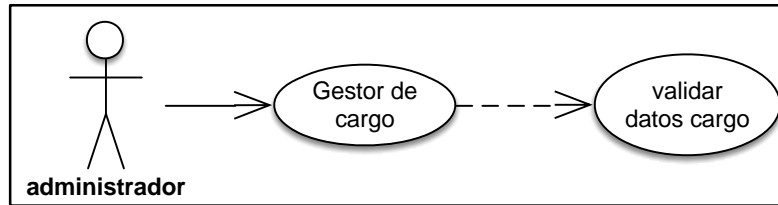


Diagrama 3.9: Caso de uso Gestor Cargo

Fuente: Elaboración propia

Gestor Cargo																			
Nombre:	Registrar cargo																		
Descripción general	Se desea Registrar un nuevo cargo																		
Actores	administrador																		
Precondiciones	Iniciar sesión																		
Flujo normal:																			
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción crear cargo</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra el formulario de creación de cargo</td> </tr> <tr> <td></td> <td>Ingresa datos de formulario</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos</td> </tr> <tr> <td></td> <td></td> <td>Almacena datos</td> </tr> </tbody> </table>		Administrador	Sistema	1	Elige la opción crear cargo				Muestra el formulario de creación de cargo		Ingresa datos de formulario				Valida datos			Almacena datos
	Administrador	Sistema																	
1	Elige la opción crear cargo																		
		Muestra el formulario de creación de cargo																	
	Ingresa datos de formulario																		
		Valida datos																	
		Almacena datos																	
Excepciones																			
	9. Validar datos 10. Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol 11. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 12. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo.																		

Modificar cargo																			
Nombre:	Modificar cargo																		
Descripción general	Se desea Modificar un nuevo cargo																		
Actores	Administrador																		
Precondiciones	Iniciar sesión El cargo debe estar registrado en la base de datos																		
Flujo normal:																			
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Selecciona la opción "listar cargo"</td> <td></td> </tr> <tr> <td></td> <td>Busca o selecciona el cargo</td> <td></td> </tr> <tr> <td></td> <td>Elige la opción "Editar"</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra detalles del registro cargo</td> </tr> <tr> <td></td> <td>Actualiza formulario de modificación seleccionando "Validar"</td> <td></td> </tr> </tbody> </table>		Administrador	Sistema	1	Selecciona la opción "listar cargo"			Busca o selecciona el cargo			Elige la opción "Editar"				Muestra detalles del registro cargo		Actualiza formulario de modificación seleccionando "Validar"	
	Administrador	Sistema																	
1	Selecciona la opción "listar cargo"																		
	Busca o selecciona el cargo																		
	Elige la opción "Editar"																		
		Muestra detalles del registro cargo																	
	Actualiza formulario de modificación seleccionando "Validar"																		

		Valida datos ingresados	
		Actualiza los datos en la base de datos	
Excepciones			
	3.	Validar datos	
	4.	Si no cumple con las especificaciones requeridas, despliega mensaje de error.	

Nombre:		Lista cargos	
Descripción general		Se lista cargos de persona	
Actores		Administrador	
Precondiciones		Iniciar sesión Tener registrado el cargo	
Flujo normal:			
		Administrador	Sistema
	1	Elige la opción de listar cargo	
			Consulta base de datos
			Lista resultados
Excepciones			
	3.	Lista resultados	
	4.	Si no existe datos, se muestra el registro de listado vacío.	

Tabla 3.9: Descripción del caso de uso Gestor cargo

Fuente: Elaboración propia

❖ CU.03: Gestor rol

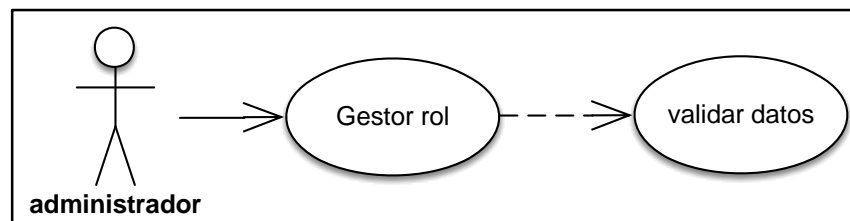


Diagrama 3.10: Caso de uso Gestor rol

Fuente: Elaboración propia

Gestor Rol			
Nombre:		Registrar rol	
Descripción general		Se desea Registrar un nuevo rol de sistema	
Actores		Administrador	
Precondiciones		Iniciar sesión	
Flujo normal:			
		Administrador	Sistema
	1	Elige la opción crear rol	
			Muestra el formulario de creación de rol
		Ingresa datos de formulario	
			Valida datos

		Almacena datos	
Excepciones			
5.	Validar datos		
6.	Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol		

Nombre:		Modificar rol	
Descripción general		Se desea Modificar un rol	
Actores		administrador	
Precondiciones		Iniciar sesión Rol debe existir en la base de datos	
Flujo normal:			
	Administrador	Sistema	
1	Busca o selecciona rol		
	Elige la opción editar rol		
		Muestra formulario de modificar rol	
	Actualiza datos de formulario		
		Valida datos	
		Almacena datos	
		Muestra mensaje que los datos fueron almacenados correctamente	
Excepciones			
1.	Validar datos		
2.	Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol		

Nombre:		Lista roles	
Descripción general		Se lista los roles registrados	
Actores		administrador	
Precondiciones		Iniciar sesión	
Flujo normal:			
	Administrador	Sistema	
1	Elige la opción de listar roles		
		Consulta base de datos	
		Lista resultados	
Excepciones			
5.	Lista resultados		
6.	Si no existe datos, se muestra el registro de listado vacío.		

Tabla 3.10: Descripción del caso de uso Gestor Rol

Fuente: Elaboración propia

❖ **CU.04: Gestor usuario**

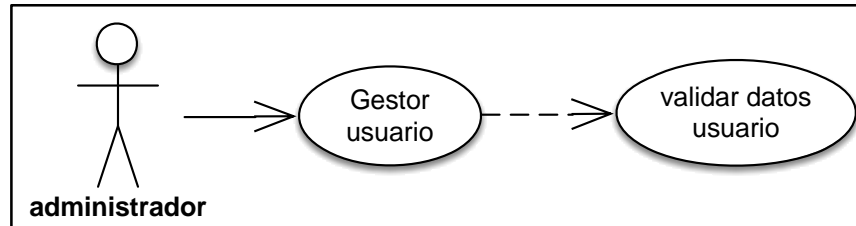


Diagrama 3.11: Caso de uso Gestor Usuario

Fuente: Elaboración propia

Gestor Usuario																			
Nombre:	Registrar usuario																		
Descripción general	Se desea registrar un nuevo usuario para que utilice el sistema																		
Actores	Administrador																		
Precondiciones	Iniciar sesión El usuario no debe estar registrado																		
Flujo normal:																			
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción registrar usuario</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra el formulario de registro usuario</td> </tr> <tr> <td></td> <td>Ingresa datos del usuario</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td></td> <td></td> <td>Almacena datos en la base de datos</td> </tr> </tbody> </table>		Administrador	Sistema	1	Elige la opción registrar usuario				Muestra el formulario de registro usuario		Ingresa datos del usuario				Valida datos ingresados			Almacena datos en la base de datos
	Administrador	Sistema																	
1	Elige la opción registrar usuario																		
		Muestra el formulario de registro usuario																	
	Ingresa datos del usuario																		
		Valida datos ingresados																	
		Almacena datos en la base de datos																	
Excepciones																			
	<ol style="list-style-type: none"> 1. Valida datos ingresados 2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 																		

Nombre:	Modificar usuario																			
Descripción general	Se desea Modificar los datos de un usuario																			
Actores	administrador																			
Precondiciones	Iniciar sesión El usuario debe estar registrado en la base de datos																			
Flujo normal:																				
	<table border="1"> <thead> <tr> <th></th> <th>Administrador</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Busca o selecciona el usuario</td> <td></td> </tr> <tr> <td></td> <td>Elige la opción modificar usuario</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra formulario de modificar usuario</td> </tr> <tr> <td></td> <td>modifica datos del usuario</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos ingresados</td> </tr> </tbody> </table>		Administrador	Sistema	1	Busca o selecciona el usuario			Elige la opción modificar usuario				Muestra formulario de modificar usuario		modifica datos del usuario				Valida datos ingresados	
	Administrador	Sistema																		
1	Busca o selecciona el usuario																			
	Elige la opción modificar usuario																			
		Muestra formulario de modificar usuario																		
	modifica datos del usuario																			
		Valida datos ingresados																		

		Almacena datos	
Excepciones			
<ol style="list-style-type: none"> 1. Validar datos 2. Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol 			

Nombre:		Eliminar usuario	
Descripción general		El administrador elimina un usuario del sistema	
Actores		administrador	
Precondiciones		Iniciar sesión El usuario debe estar registrado en la base de datos	
Flujo normal:			
	Administrador	Sistema	
1	Busca o selecciona el usuario		
	Elige la opción eliminar usuario		
		Muestra mensaje de confirmación	
	Confirma eliminación		
		Valida datos ingresados	
		Modifica estado de usuario en la base de datos	
Excepciones			
<ol style="list-style-type: none"> 1. Validar datos 2. Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol 			

Nombre:		Lista usuario	
Descripción general		Se lista los usuarios registrados	
Actores		administrador	
Precondiciones		Iniciar sesión	
Flujo normal:			
	Administrador	Sistema	
1	Elige la opción de listar usuarios		
		Consulta base de datos	
		Lista resultados	
Excepciones			
<ol style="list-style-type: none"> 1. Lista resultados 2. Si no existe datos, se muestra el registro de listado vacío. 			

Tabla 3.9: Descripción del caso de uso Gestor usuario

Fuente: Elaboración propia

❖ **CU.05: Asignación rol**

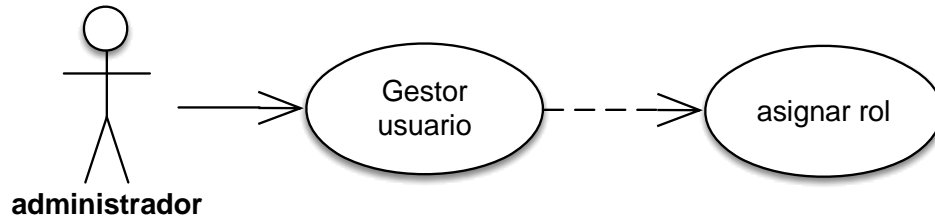


Diagrama 3.12: Caso de uso Gestor Usuario

Fuente: Elaboración propia

Asignación rol																			
Nombre:	Asignación rol																		
Descripción general	Se asigna rol de usuario																		
Actores	administrador																		
Precondiciones	Iniciar sesión Debe existir el rol de usuario																		
Flujo normal:																			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 60%;">Administrador</th> <th style="width: 35%;">Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Selecciona el rol</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra el formulario de asignar rol</td> </tr> <tr> <td></td> <td>Ingresar datos de usuario y rol</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td></td> <td></td> <td>Almacena datos en la base de datos</td> </tr> </tbody> </table>		Administrador	Sistema	1	Selecciona el rol				Muestra el formulario de asignar rol		Ingresar datos de usuario y rol				Valida datos ingresados			Almacena datos en la base de datos
	Administrador	Sistema																	
1	Selecciona el rol																		
		Muestra el formulario de asignar rol																	
	Ingresar datos de usuario y rol																		
		Valida datos ingresados																	
		Almacena datos en la base de datos																	
Excepciones																			
	<ol style="list-style-type: none"> 1. Valida datos ingresados 2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 																		

Tabla 3.10: Descripción del caso de uso asignar rol

Fuente: Elaboración propia

3.1.2.2.2 Modulo administrar Bloque

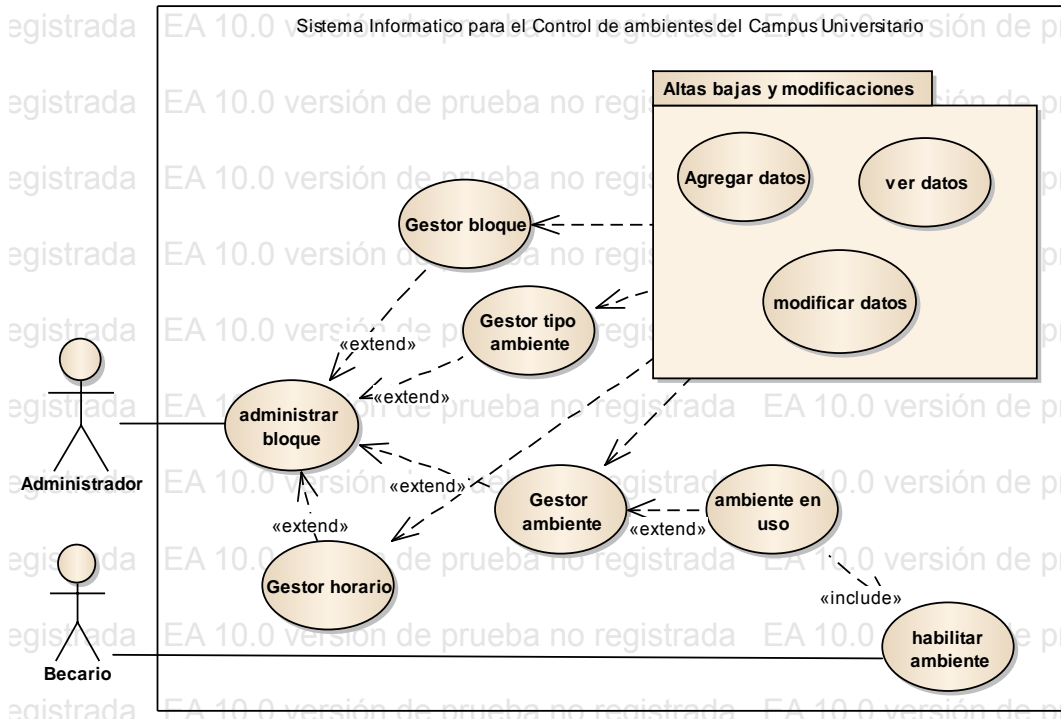


Diagrama 3.13: Diagrama de Caso de Uso: Administrar bloque
Actor: Administrador, becario
Fuente: Elaboración propia

❖ CU.06: Gestor bloque

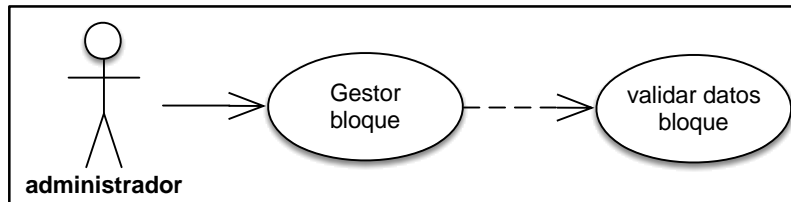


Diagrama 3.14: Caso de uso Gestor Bloque
Fuente: Elaboración propia

Gestor bloque	
Nombre:	Registrar bloque
Descripción general	Se registrar un nuevo bloque
Actores	Administrador becarios
Precondiciones	Iniciar sesión El bloque no debe estar registrado
Flujo normal:	
	Usuario
1	Elige la opción "Registrar Bloque"
	Sistema

		Muestra el formulario de registro bloque	
	Ingresar datos del bloque		
		Valida datos ingresados	
		Almacena datos en la base de datos	
Excepciones			
1. Valida datos ingresados			
2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios.			
3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo.			

Nombre:		Modificar bloque	
Descripción general		Se desea Modificar los datos de un bloque	
Actores		Administrador becarios	
Precondiciones		Iniciar sesión El ubloque debe estar registrado	
Flujo normal:			
	Usuario	Sistema	
1	Busca o selecciona el Bloque		
	Elige la opción modificar Bloque		
		Muestra formulario de modificar Bloque	
	Modifica datos del formulario		
		Valida datos ingresados	
		Actualiza datos en la base de datos	
Excepciones			
1. Validar datos			
2. Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol			

Nombre:		Lista bloques	
Descripción general		lista de bloques registrados	
Actores		Administrador becarios	
Precondiciones		Iniciar sesión	
Flujo normal:			
	Administrador	Sistema	
1	Elige la opción de listar bloques		
		Consulta base de datos	
		Lista resultados	
Excepciones			
1. Lista resultados			
2. Si no existe datos, se muestra el registro de listado vacío.			

Tabla 3.11: Descripción del caso de uso Gestor bloque

Fuente: Elaboración propia

❖ **CU.07: Gestor tipo aula**

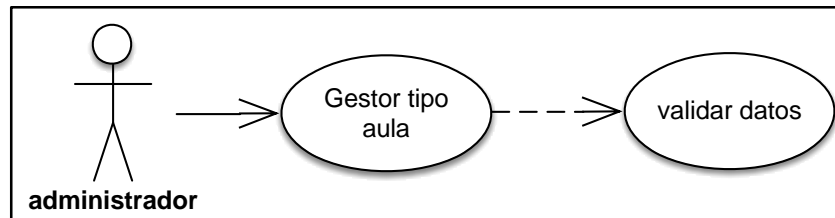


Diagrama 3.15: Caso de uso Gestor tipo aula

Fuente: Elaboración propia

Gestor tipo_Aula																						
Nombre:	Registrar tipo_Aula																					
Descripción general	Se registrar un nuevo tipo de Aula																					
Actores	Administrador becarios																					
Precondiciones	Iniciar sesión El tipo de aula no debe estar registrado																					
Flujo normal:																						
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 45%;">Usuario</th> <th style="width: 50%;">Sistema</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Elige la opción "Registrar Tipo aula"</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Muestra el formulario de registro tipo_aula</td> </tr> <tr> <td></td> <td>Ingresa datos de tipo aula</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td></td> <td></td> <td>Almacena datos en la base de datos</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Usuario	Sistema	1	Elige la opción "Registrar Tipo aula"				Muestra el formulario de registro tipo_aula		Ingresa datos de tipo aula				Valida datos ingresados			Almacena datos en la base de datos			
	Usuario	Sistema																				
1	Elige la opción "Registrar Tipo aula"																					
		Muestra el formulario de registro tipo_aula																				
	Ingresa datos de tipo aula																					
		Valida datos ingresados																				
		Almacena datos en la base de datos																				
Excepciones																						
	<ol style="list-style-type: none"> 1. Valida datos ingresados 2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 																					

Tabla 3.12: Descripción del caso de uso Gestor tipo aula

Fuente: Elaboración propia

❖ **CU.08: Gestor Aula**

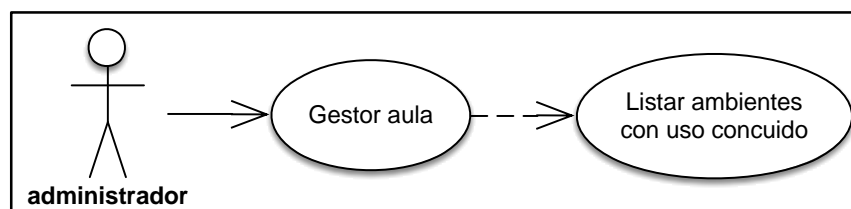


Diagrama 3.16: Caso de uso Gestor aula

Fuente: Elaboración propia

Gestor Aula													
Nombre:	Registrar Aula												
Descripción general	Se registra un Aula												
Actores	Administrador becarios												
Precondiciones	- Iniciar sesión - El de aula no debe estar registrado												
Flujo normal:													
	<table border="1"> <thead> <tr> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción "Registrar Aula"</td> </tr> <tr> <td></td> <td>Muestra el formulario de registro Aula</td> </tr> <tr> <td></td> <td>Ingresa datos del Aula</td> </tr> <tr> <td></td> <td>Valida datos ingresados</td> </tr> <tr> <td></td> <td>Almacena datos en la base de datos</td> </tr> </tbody> </table>	Usuario	Sistema	1	Elige la opción "Registrar Aula"		Muestra el formulario de registro Aula		Ingresa datos del Aula		Valida datos ingresados		Almacena datos en la base de datos
Usuario	Sistema												
1	Elige la opción "Registrar Aula"												
	Muestra el formulario de registro Aula												
	Ingresa datos del Aula												
	Valida datos ingresados												
	Almacena datos en la base de datos												
Excepciones													
	<ol style="list-style-type: none"> 1. Valida datos ingresados 2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 												

Tabla 3.13: Descripción del caso de uso Gestor Aula

Fuente: Elaboración propia

❖ CU.09: Gestor Horario

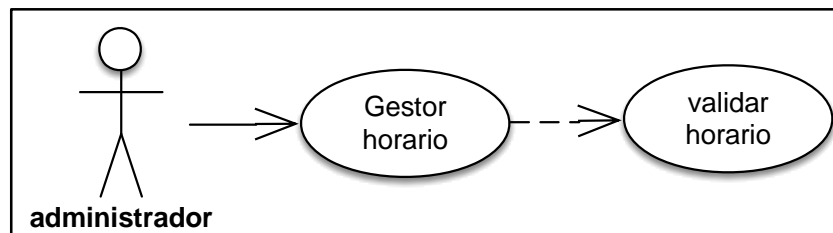


Diagrama 3.17: Caso de uso Gestor Horario

Fuente: Elaboración propia

Gestor horario									
Nombre:	Registrar horario								
Descripción general	Se registrar una nueva hora								
Actores	Administrador becarios								
Precondiciones	Iniciar sesión La hora no debe estar registrado								
Flujo normal:									
	<table border="1"> <thead> <tr> <th>Usuario</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Elige la opción "Registrar Horario"</td> </tr> <tr> <td></td> <td>Muestra el formulario de registro horario</td> </tr> <tr> <td></td> <td>Llena formulario de registro de horario</td> </tr> </tbody> </table>	Usuario	Sistema	1	Elige la opción "Registrar Horario"		Muestra el formulario de registro horario		Llena formulario de registro de horario
Usuario	Sistema								
1	Elige la opción "Registrar Horario"								
	Muestra el formulario de registro horario								
	Llena formulario de registro de horario								

		Valida datos ingresados	
		Almacena datos en la base de datos	
Excepciones			
1.		Valida datos ingresados	
2.		Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios.	
3.		Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo.	

Nombre:		Modificar horario	
Descripción general		Se desea Modificar los datos de un horario	
Actores		Administrador becarios	
Precondiciones		Iniciar sesión La hora debe estar registrado	
Flujo normal:			
	Usuario	Sistema	
1	Busca o selecciona Hora_		
	Elige la opción modificar hora		
		Muestra formulario de modificar hora	
	Modifica datos del formulario		
		Valida datos ingresados	
		Actualiza datos rn la base de datos	
Excepciones			
1.		Validar datos	
2.		Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol	

Tabla 3.14: Descripción del caso de uso Gestor Horario

Fuente: Elaboración propia

3.1.2.2.3 Modulo Administrar Solicitud

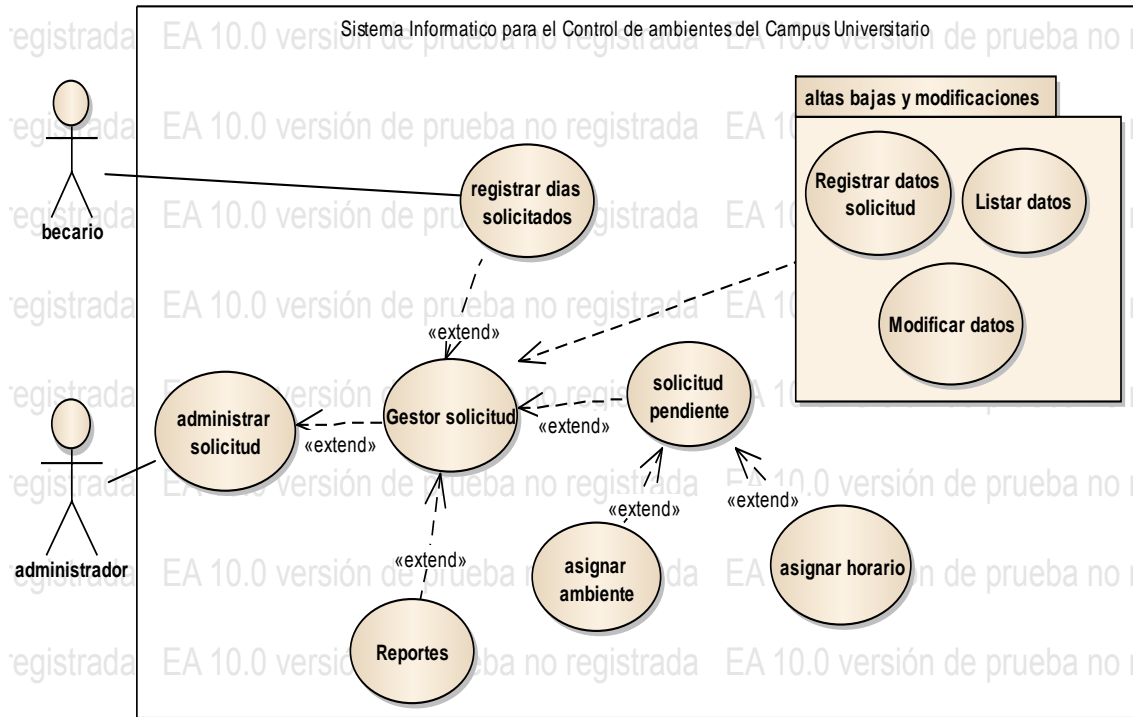


Diagrama 3.18: Diagrama de Caso de Uso: Administrar Solicitud
Fuente: Elaboración propia

❖ CU.10: Gestor Solicitud

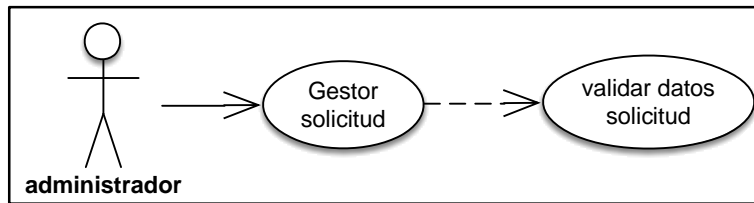


Diagrama 3.19: Caso de uso Gestor Solicitud
Fuente: Elaboración propia

Gestor Solicitud	
Nombre:	Registrar solicitud
Descripción general	Se registra una solicitud
Actores	Administrador becarios
Precondiciones	Iniciar sesión La solicitud no debe estar registrado
Flujo normal:	
	Usuario
1	Elige la opción "Registrar"
	Sistema

	Solicitud"	
		Muestra el formulario de registro Solicitud
	Ingresa datos de la solicitud	
		Valida datos ingresados
		Almacena datos en la base de datos
Excepciones		
<ol style="list-style-type: none"> 1. Valida datos ingresados 2. Si uno o más campos obligatorios son NULOS, el sistema mostrara un mensaje de error en campos obligatorios. 3. Si uno o más campos no cumplen con las especificaciones establecidas, se mostrara un mensaje de error en formato de campo. 		

Nombre:	Modificar Solicitud	
Descripción general	Se desea Modificar los datos de una solicitud	
Actores	Administrador becarios	
Precondiciones	Iniciar sesión La solicitud debe estar registrado	
Flujo normal:		
	Usuario	Sistema
1	Busca o selecciona la Solicitud	
	Elige la opción modificar Solicitud	
		Muestra formulario de modificar Bloque
	Modifica datos del formulario	
		Valida datos ingresados
		Actualiza datos rn la base de datos
Excepciones		
<ol style="list-style-type: none"> 1. Validar datos 2. Si no cumple con las especificaciones requeridas, despliega mensaje que debe llenar los campos de rol 		

Tabla 3.15: Descripción del caso de uso Gestor Solicitud

Fuente: Elaboración propia

3.1.2.3 DIAGRAMA DE CLASES

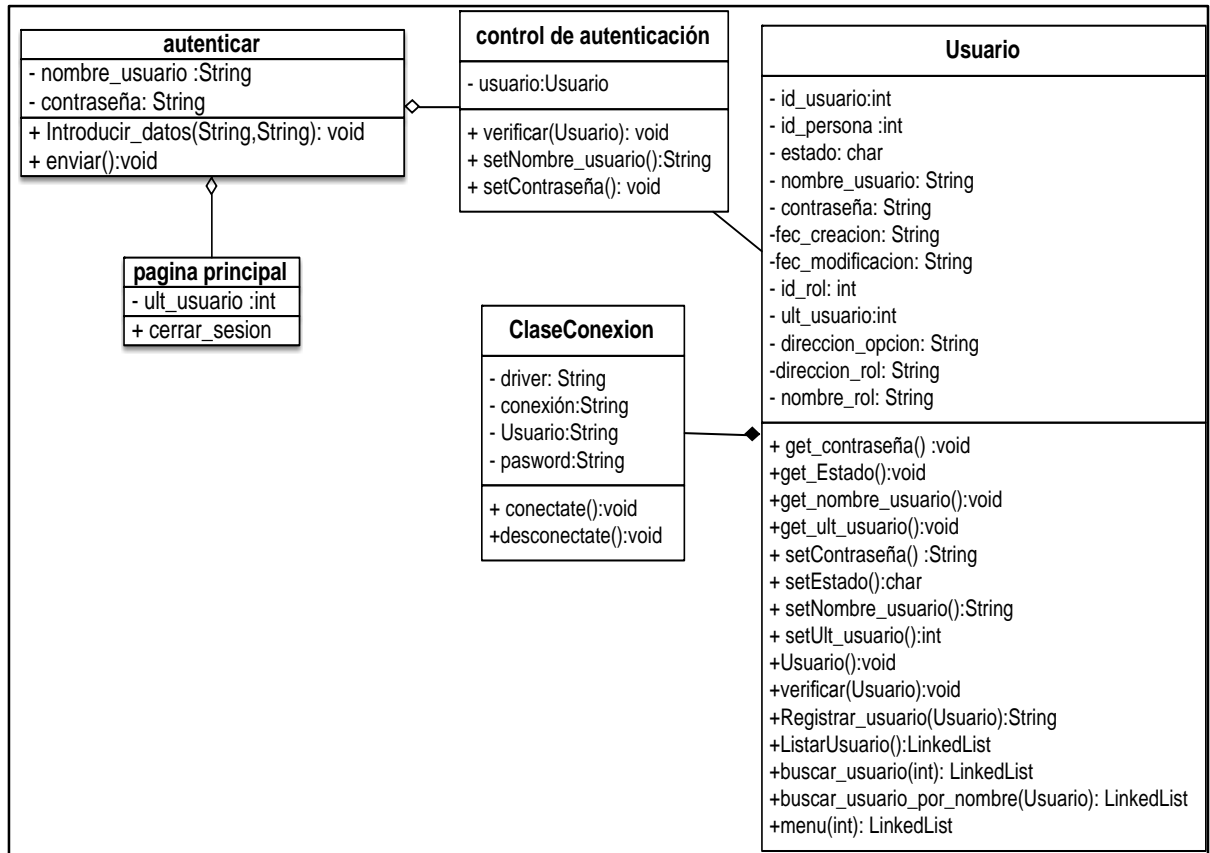


Diagrama 3.20: Diagrama de Clases: Autenticación de Usuario

Fuente: Elaboración propia

El diagrama 3-18 de la clase usuario comienza con la página principal de autenticación de usuario capturando los datos en el formulario realiza el control de autenticación de usuario que se encarga de la parte lógica del sistema llamando a los métodos de la clase Usuario manteniendo la continua conexión a la Base de Datos.

3.1.2.3.1 Diagrama de clases autenticar

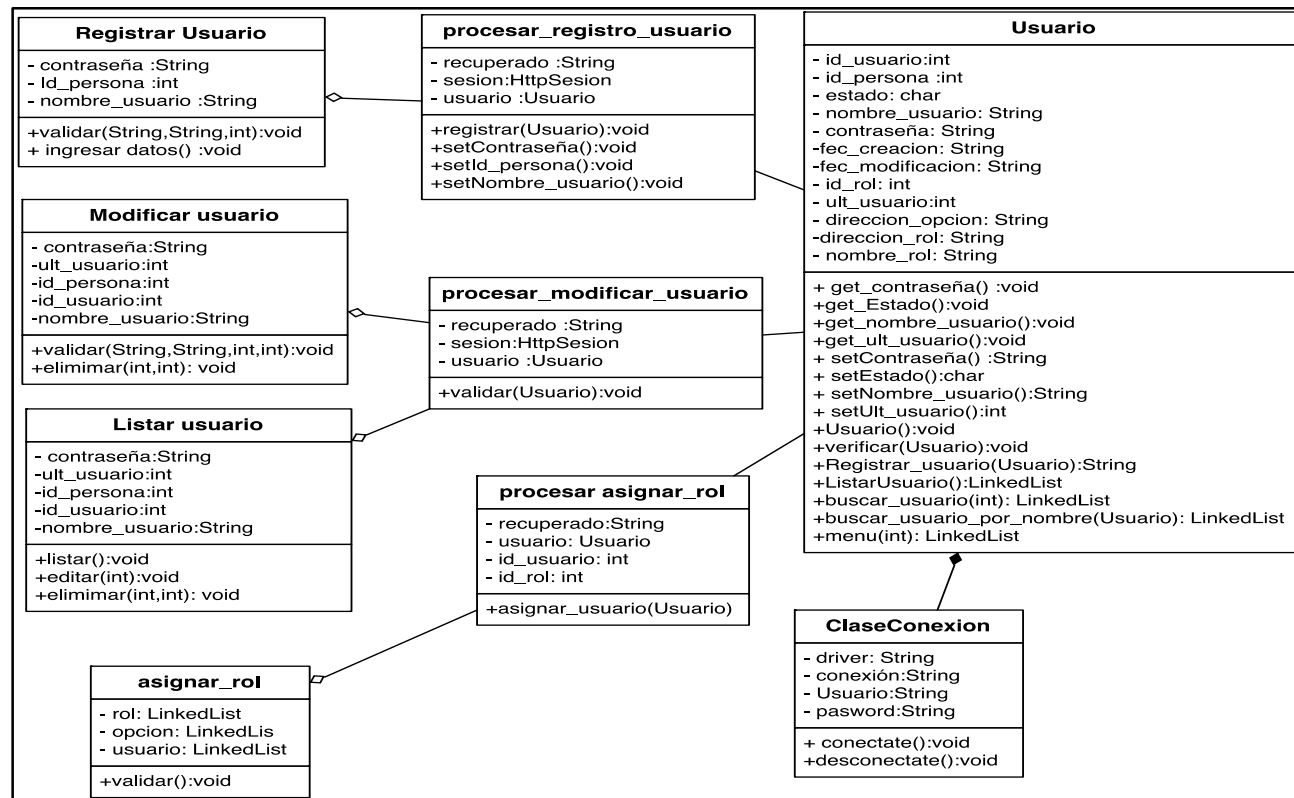


Diagrama 3.21: Diagrama de Clase: Usuario

Fuente: Elaboración propia

El diagrama 3-19 de la clase usuario nos muestra las cuatro funcionalidades que se realiza dentro del módulo administrar usuario que son: Registrar, modificar, listar y asignar rol cada una llama a la clase procesar que le corresponda realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Usuario manteniendo la continua conexión a la Base de Datos.

3.1.2.3.2 Diagrama de clases Gestor bloque

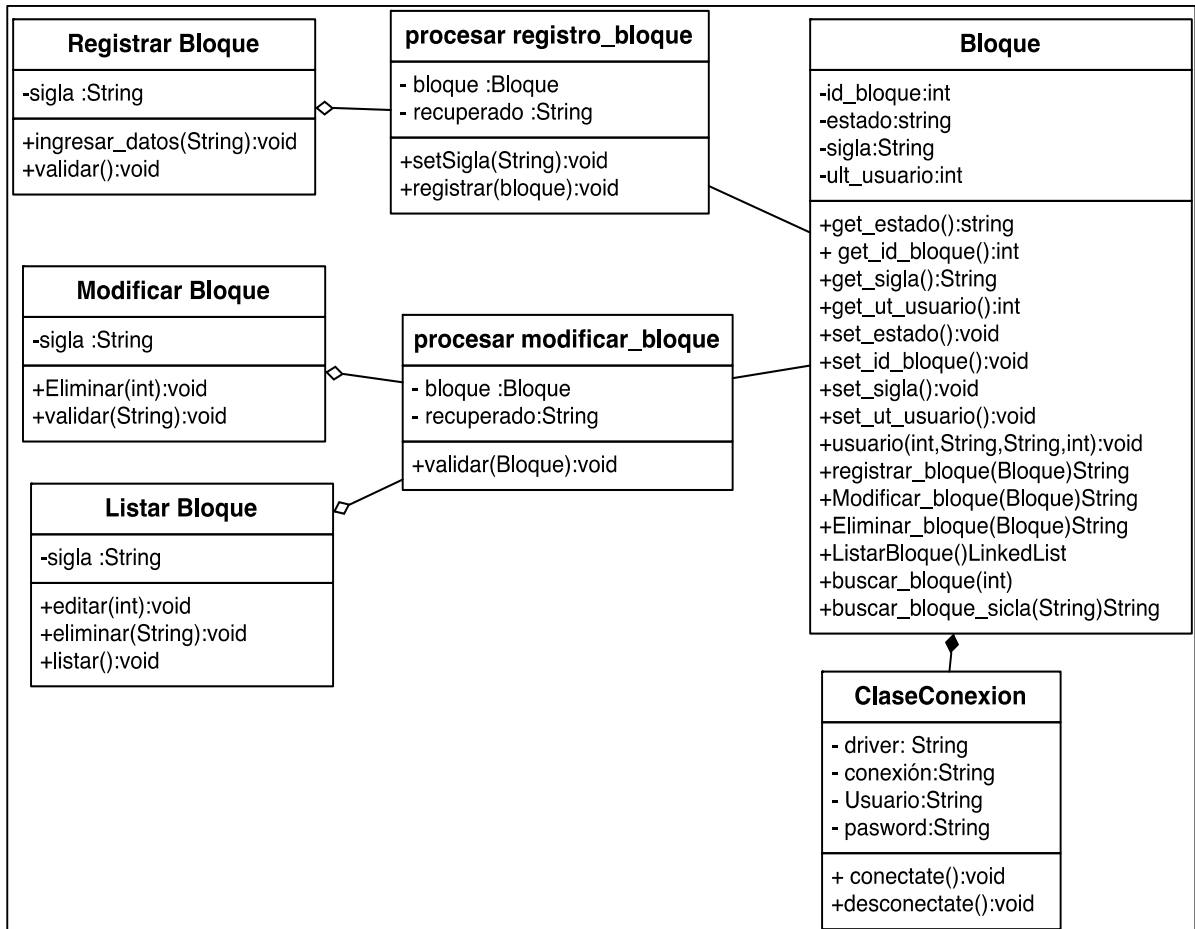


Diagrama 3.22: Diagrama de Clases: Registro de Bloque

Fuente: Elaboración propia

El diagrama 3-20 de la clase Gestor bloque nos muestra las tres funcionalidades que se realiza dentro del módulo administrar Bloque: que son registrar, modificar y listar cada una llama a la clase procesar que le corresponda realizando internamente la parte lógica del sistema pasa a llamar los métodos de la clase Bloque manteniendo la continua conexión a la Base de Datos.

3.1.2.3.3 Diagramas de clases Gestor solicitud

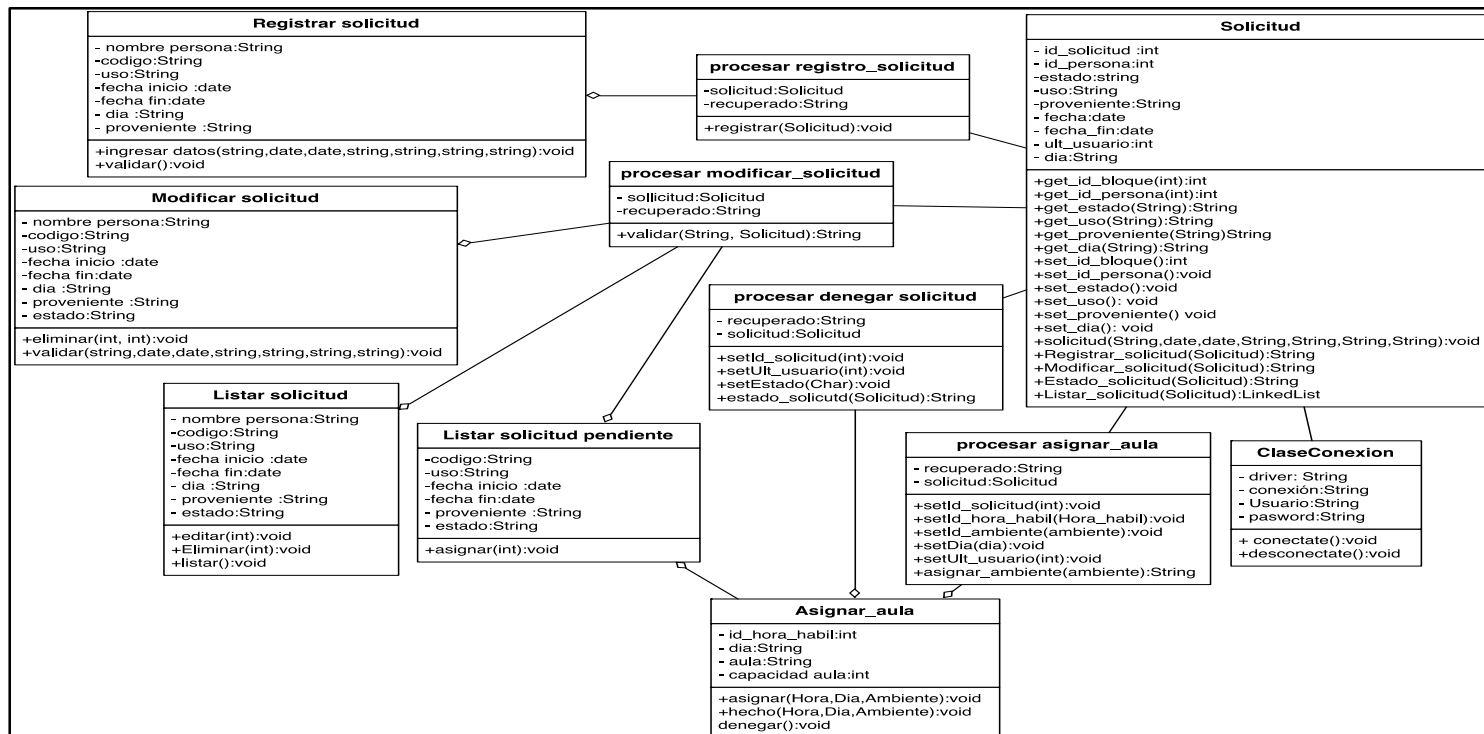


Diagrama 3.23: Diagrama de Clases: Solicitud

Fuente: Elaboración propia

El diagrama 3-19 de la clase solicitud nos muestra las cinco funcionalidades que se realiza dentro del módulo administrar Usuario que son: Registrar, modificar, listar y listar solicitud pendiente después de haber visualizado las solicitudes pendientes se realiza la asignación del aula, cada una llama a la clase procesar que le corresponda realizando internamente la parte lógica del sistema llamando al método de la clase solicitud manteniendo la continua conexión a la Base de Datos.

3.1.2.4 DIAGRAMA DE SECUENCIA

- **Autenticar usuario:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clase: Autenticar usuario

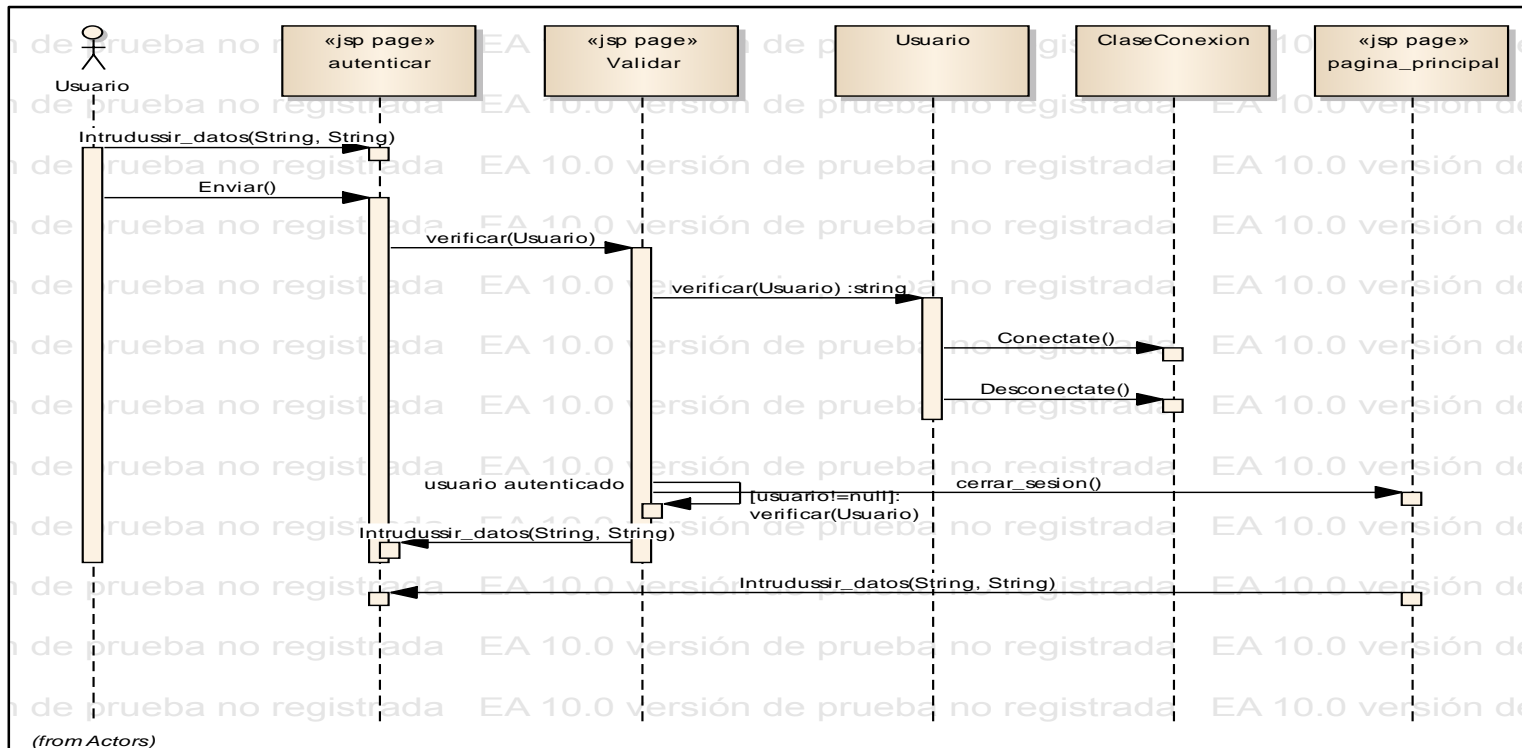


Diagrama 3.24: Diagrama de Secuencia: Autenticar Usuario
Fuente: Elaboración propia

- **Registrar usuario:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: Gestor usuario

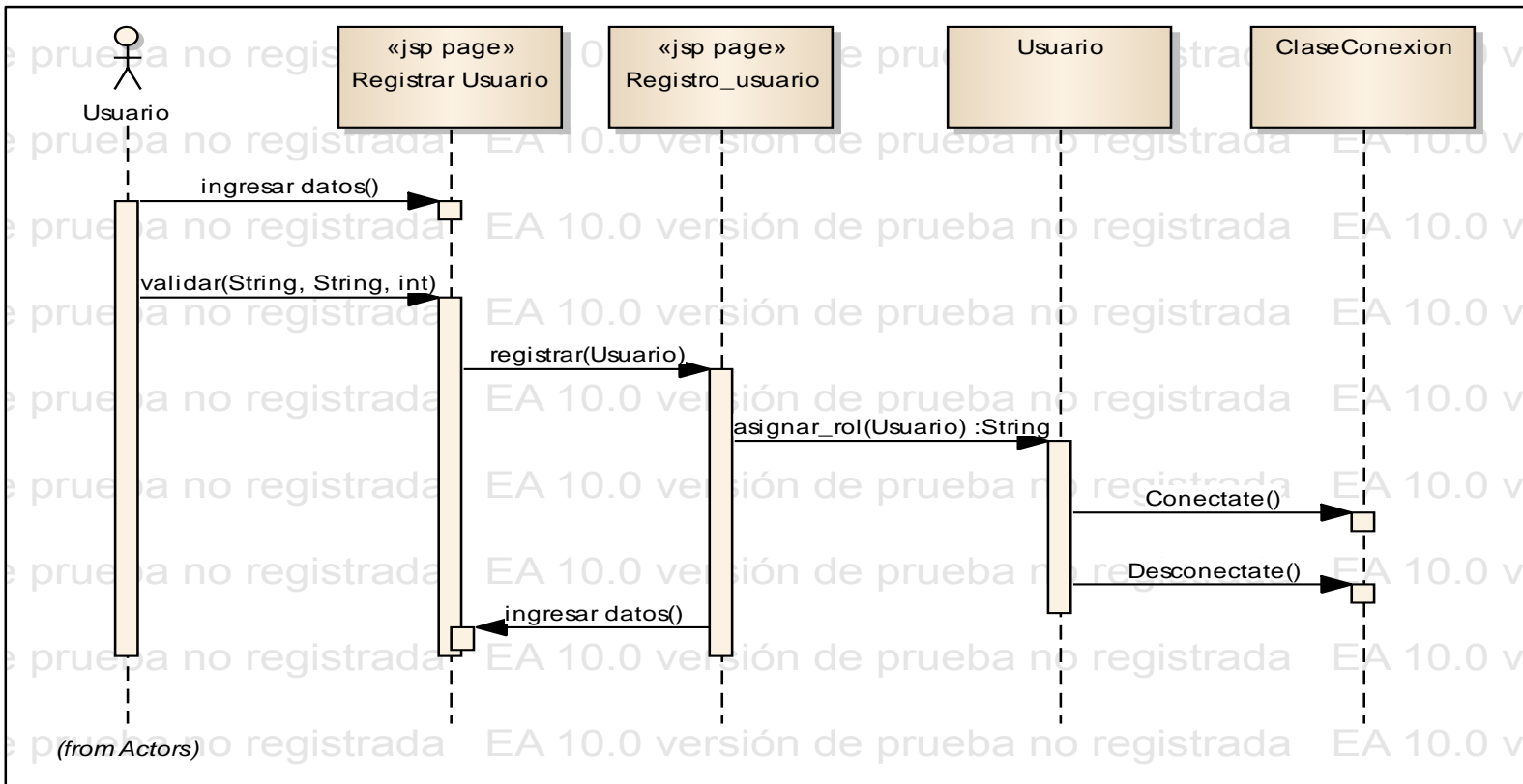


Diagrama 3.25: Diagrama de Secuencia: Registrar Usuario
Fuente: Elaboración propia

- **Asignar rol a usuario del sistema:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: Gestor usuario

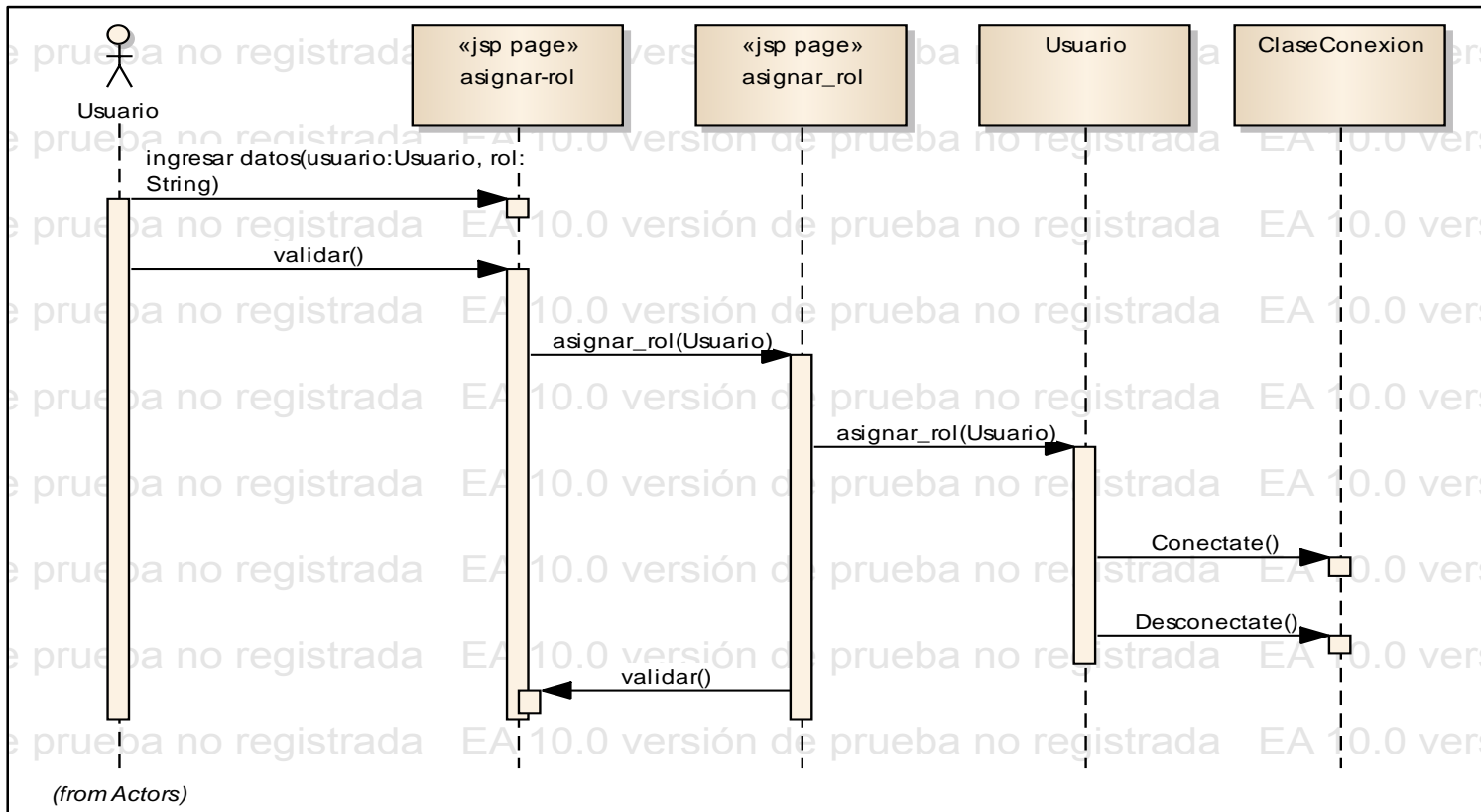


Diagrama 3.26: Diagrama de Secuencia: Asignar rol a usuario
Fuente: Elaboración propia

- **Asignar opción a usuario del sistema:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: Gestor usuario.

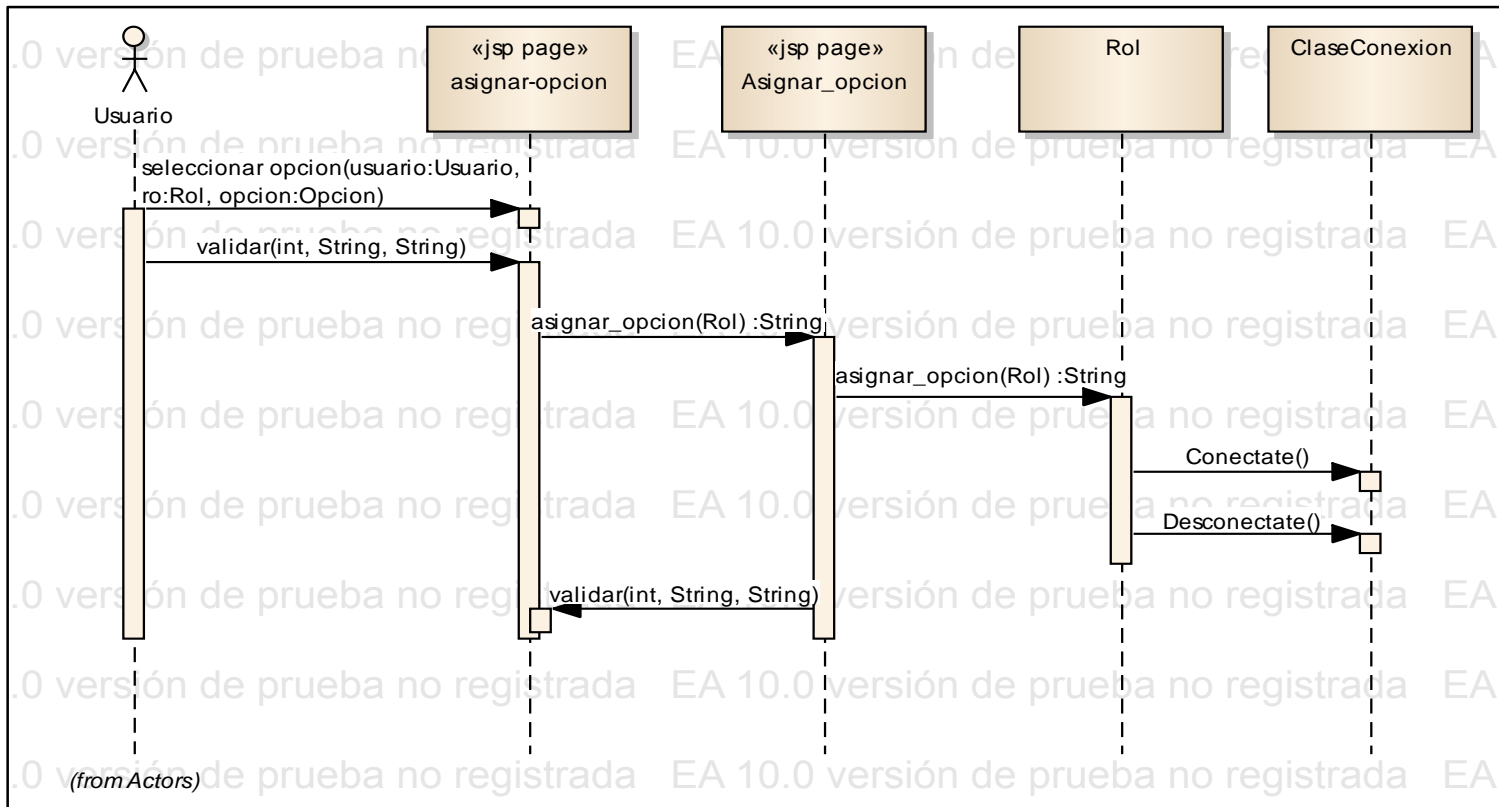


Diagrama 3.27: Diagrama de Secuencia: Asignar opción a usuario
Fuente: Elaboración propia

- **Registrar Bloque:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: Gestor Bloque.

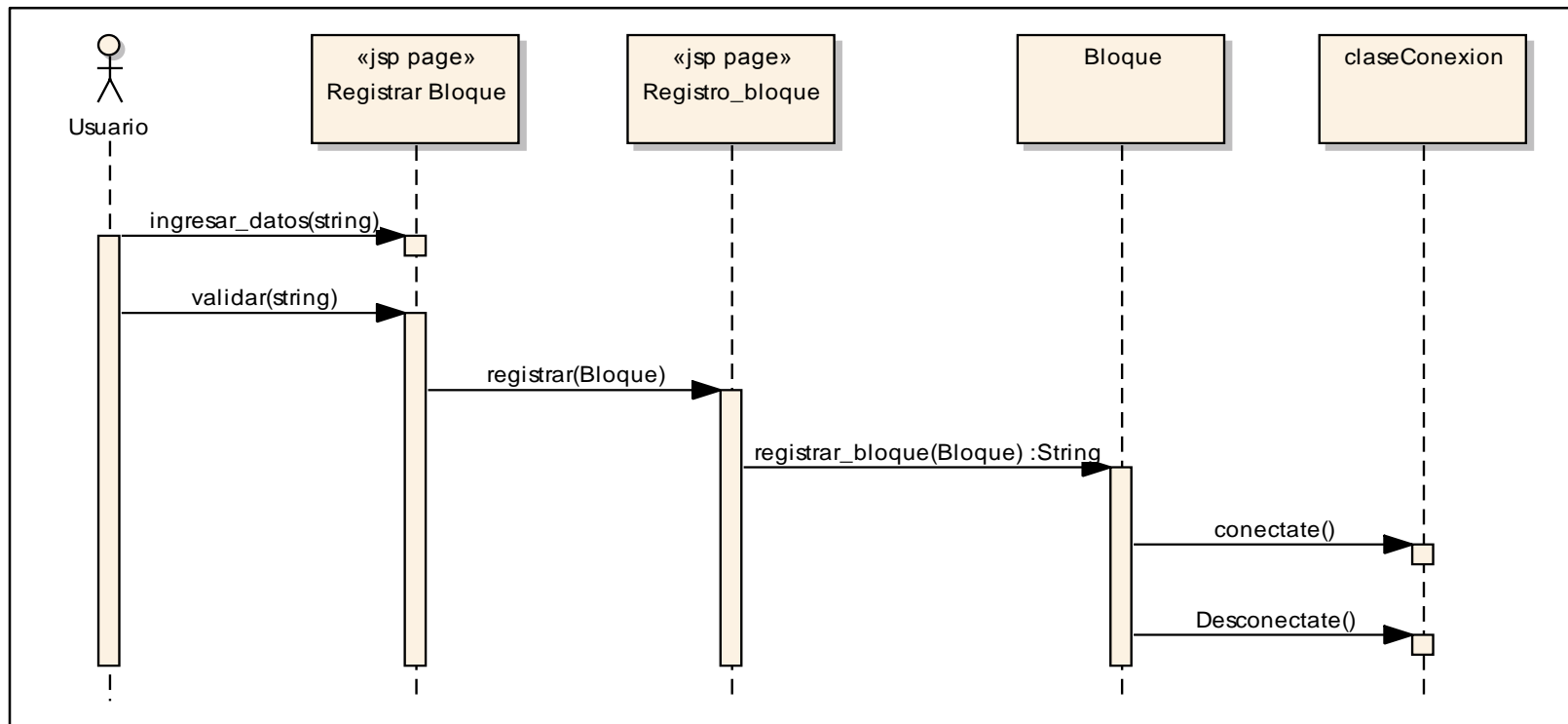


Diagrama 3.28: Diagrama de Secuencia: Registrar Bloque

Fuente: Elaboración propia

- **Listar bloque:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: Gestor Bloque.

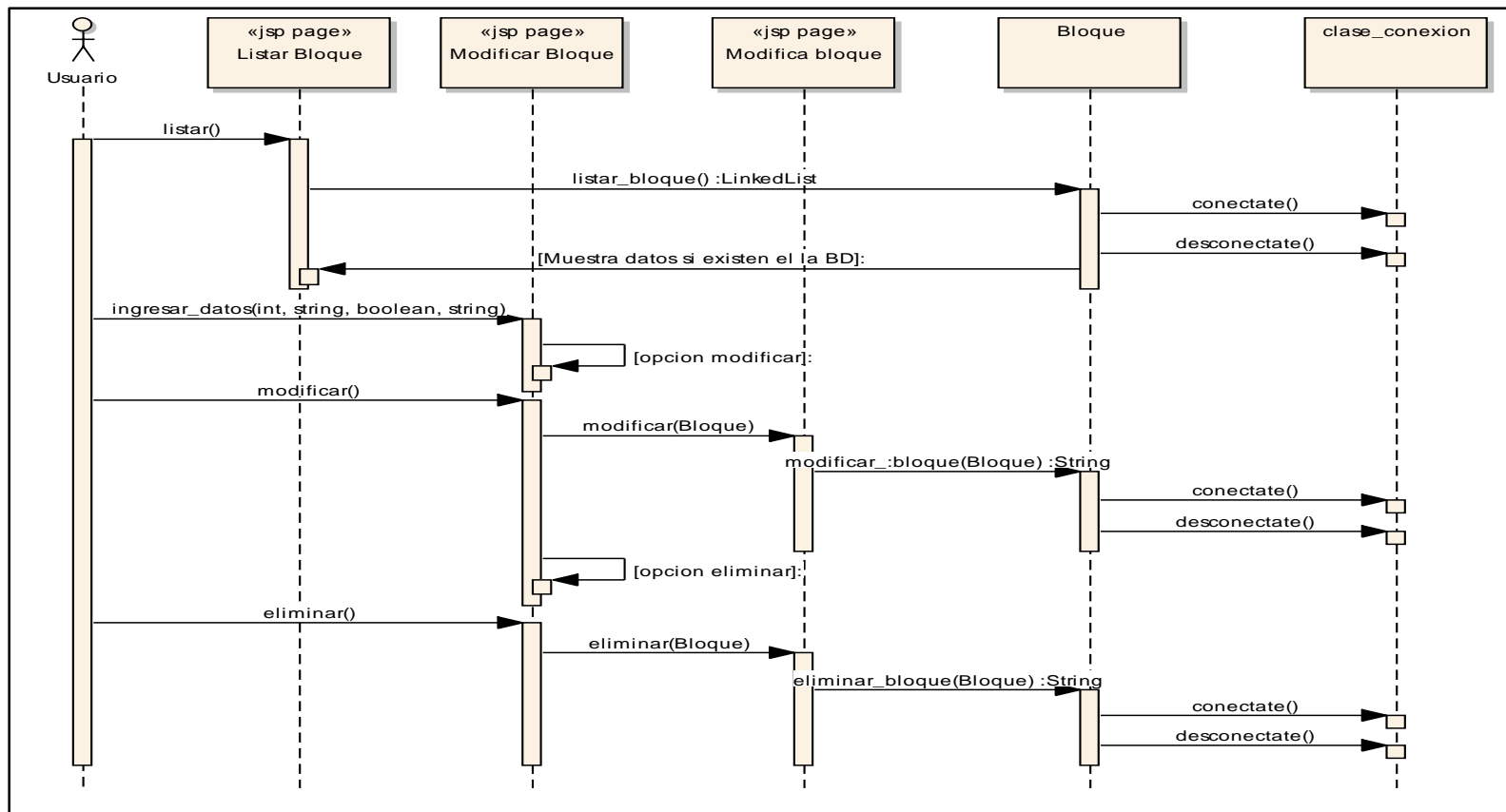


Diagrama 3.29: Diagrama de Secuencia: Listar Bloque

Fuente: Elaboración propia

- **Registrar Solicitud:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: administrar Solicitud.

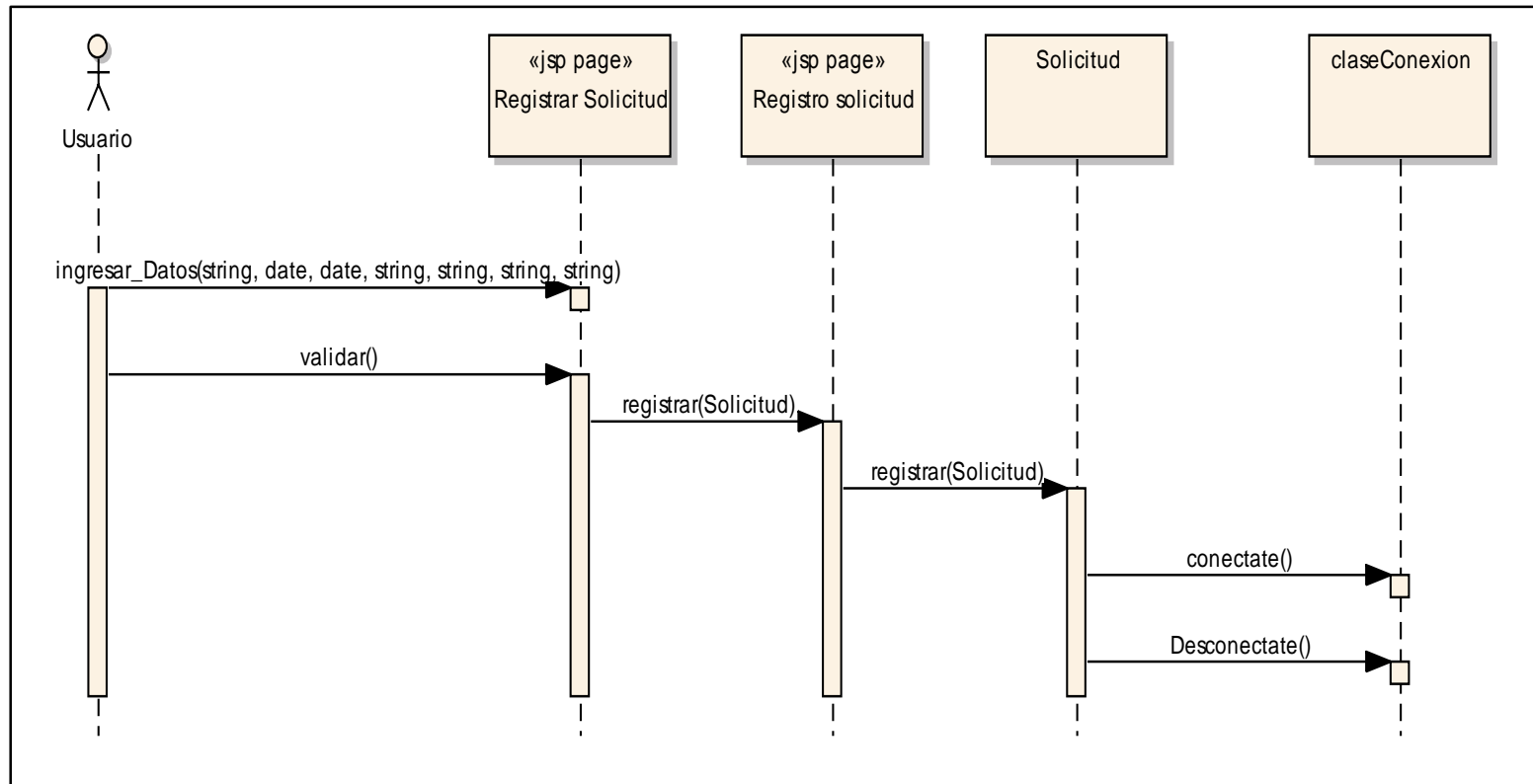


Diagrama 3.30: Diagrama de Secuencia: Registrar Solicitud

Fuente: Elaboración propia

- **Ver Solicitudes pendientes:** El siguiente es un diagrama de secuencia que representa los flujos de proceso de acuerdo al diagrama de clases: administrar solicitud.

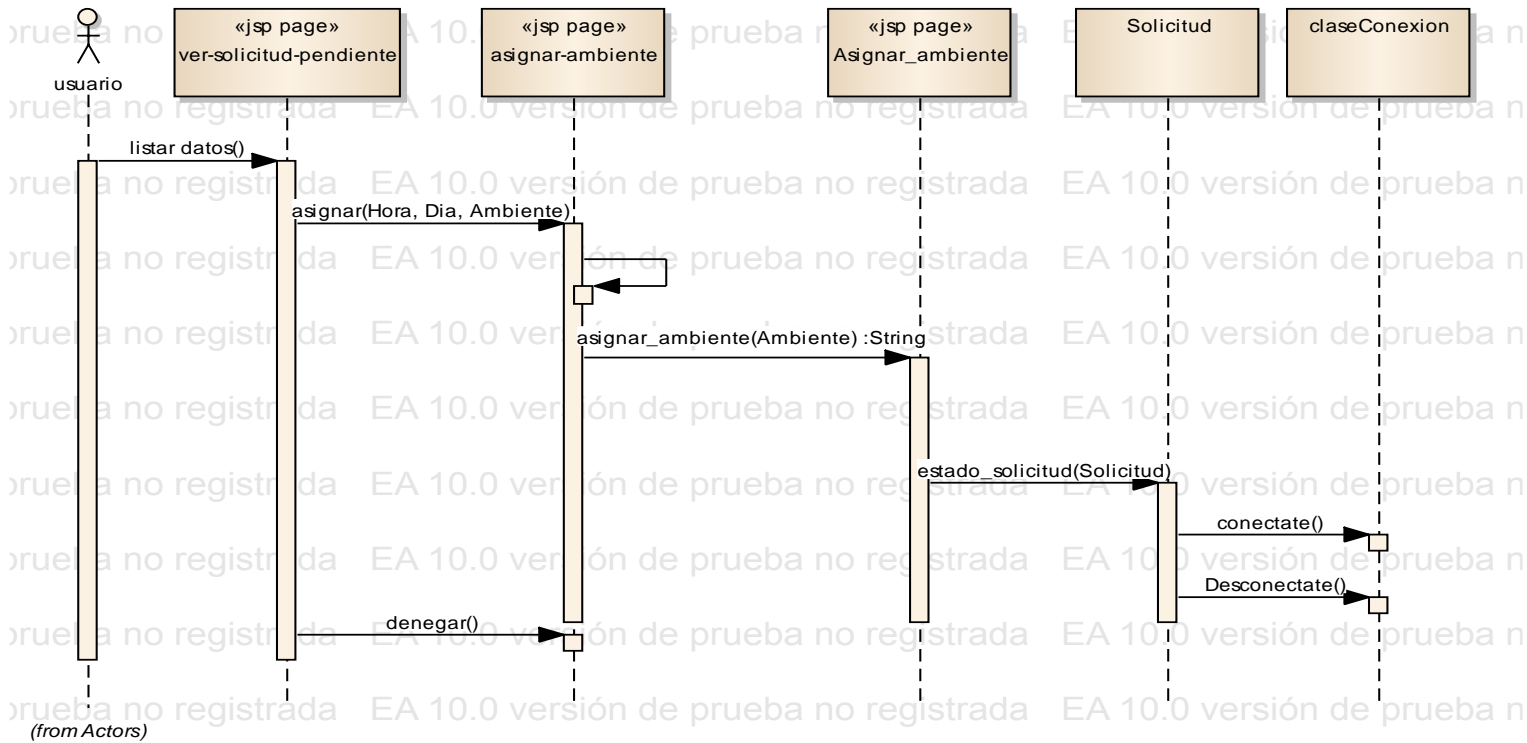


Diagrama 3.31: Diagrama de Secuencia: Asignación de ambiente

Fuente: Elaboración propia

3.1.2.5 DISEÑO DE LA ARQUITECTURA

Para la implementación del sistema SAAC-UAP se utiliza la arquitectura en capas.

3.1.2.5.1 Arquitectura en capas

Define un conjunto de niveles o capas organizados jerárquicamente; cada capa le presta servicios a capa superior y es cliente de la capa inferior.

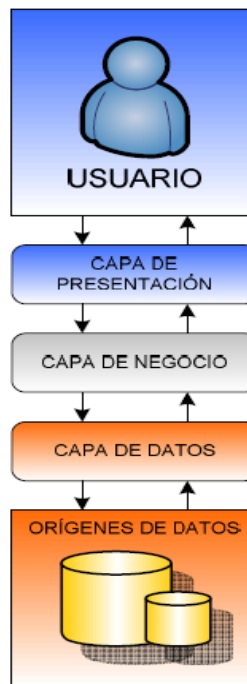


Figura 3. 2: Arquitectura en capas

Fuente: [cobo,2010]

- **Capa de presentación:** es la capa que presenta la interfaz del sistema al usuario e interactúa con el mismo, permitiéndole ingresar datos y recibir datos.
- **Capa de negocios:** es la capa que realiza los procesos una vez ingresados los datos necesarios, utilizando reglas empresariales o del negocio ya implementadas. Las reglas del negocio implementan la lógica empresarial de la aplicación.

- **Capa de Datos:** es la capa que realiza el acceso al origen de datos que requiere la capa de negocio, es capa independiente de componentes lógicos de acceso a datos, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.

3.1.2.6 DISEÑO DE LA JERARQUIA DE MENU DEL SISTEMA

Se debe recalcar que RUP no define un diseño jerárquico de menú del sistema en ninguna de las fases, por lo cual se vio en la necesidad de mostrar el diseño para un mejor entendimiento.

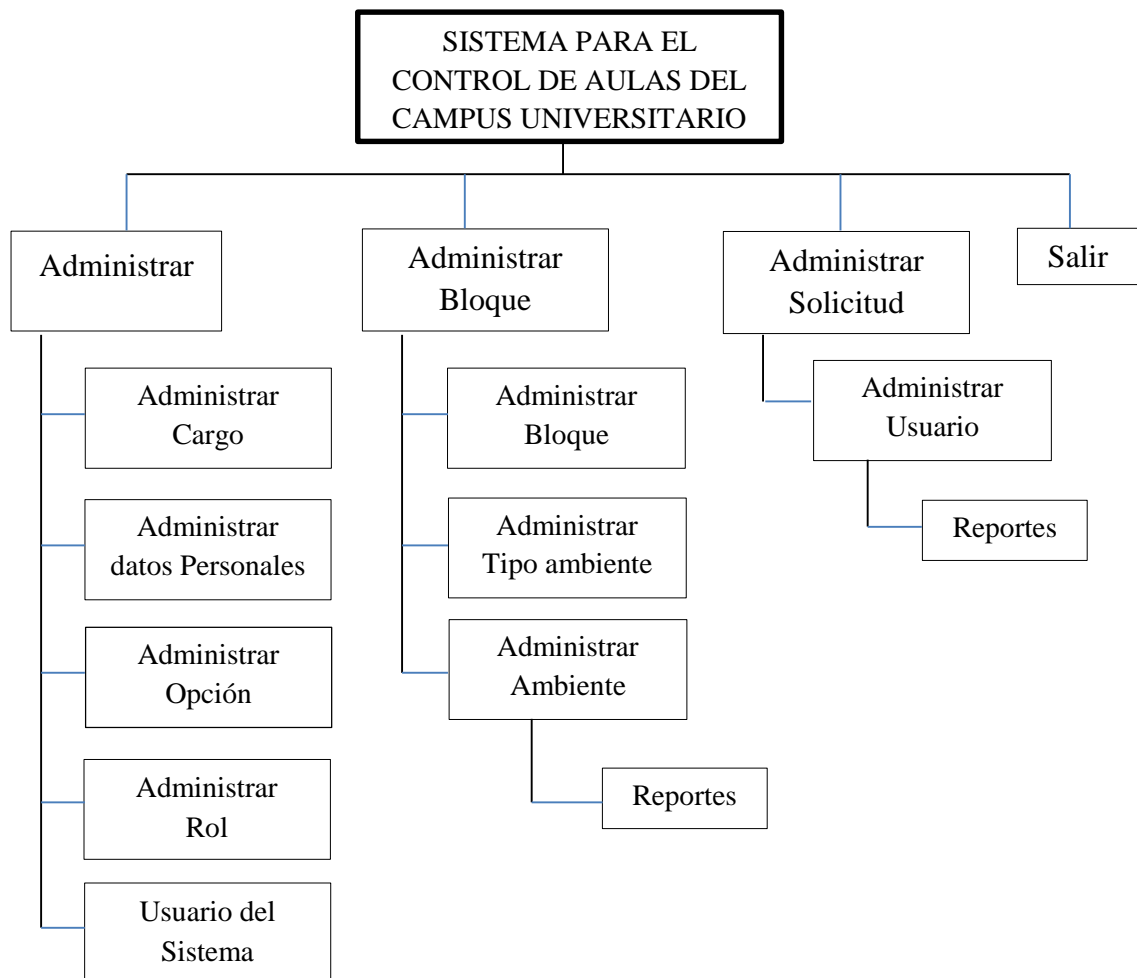


Figura 3.1: Diseño de la jerarquía del menú de sistema SAAC-UAP
Fuente: Elaboración propia

3.1.3 FASE III: FASE DE CONSTRUCCION

3.1.3.1 ELABORACION DEL MODELO LÓGICO DE LA BASE DE DATOS

El diseño lógico en el proceso de construir un esquema de la información que utiliza el sistema, basándose en un modelo de base de datos específico, independiente del SGBD que se vaya a utilizar.

En esta etapa se transforma el esquema conceptual en un esquema lógico que utilizara las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar.

El diseño lógico es una fuente de información para el diseño físico. Además desempeña un papel importante durante la etapa de mantenimiento del sistema, ya que permite que los futuros cambios que se realicen sobre los programas de aplicación o sobre los datos, se representen correctamente en la base de datos.

MODELO CONCEPTUAL

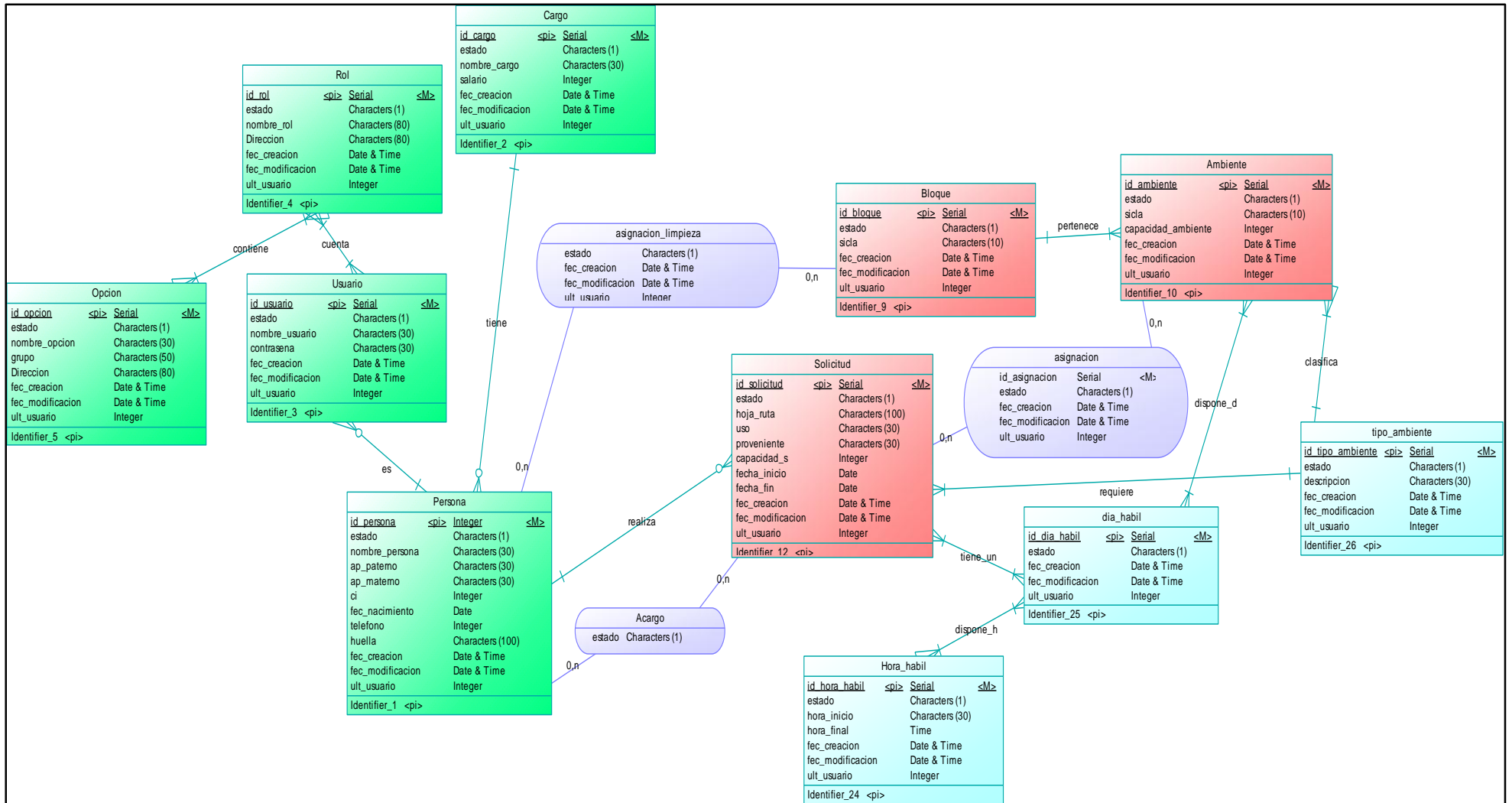


Diagrama 3.32: Modelo Conceptual de la Base de Datos del sistema de información SAAC-UAP

Fuente: Elaboración propia

MODELO LOGICO

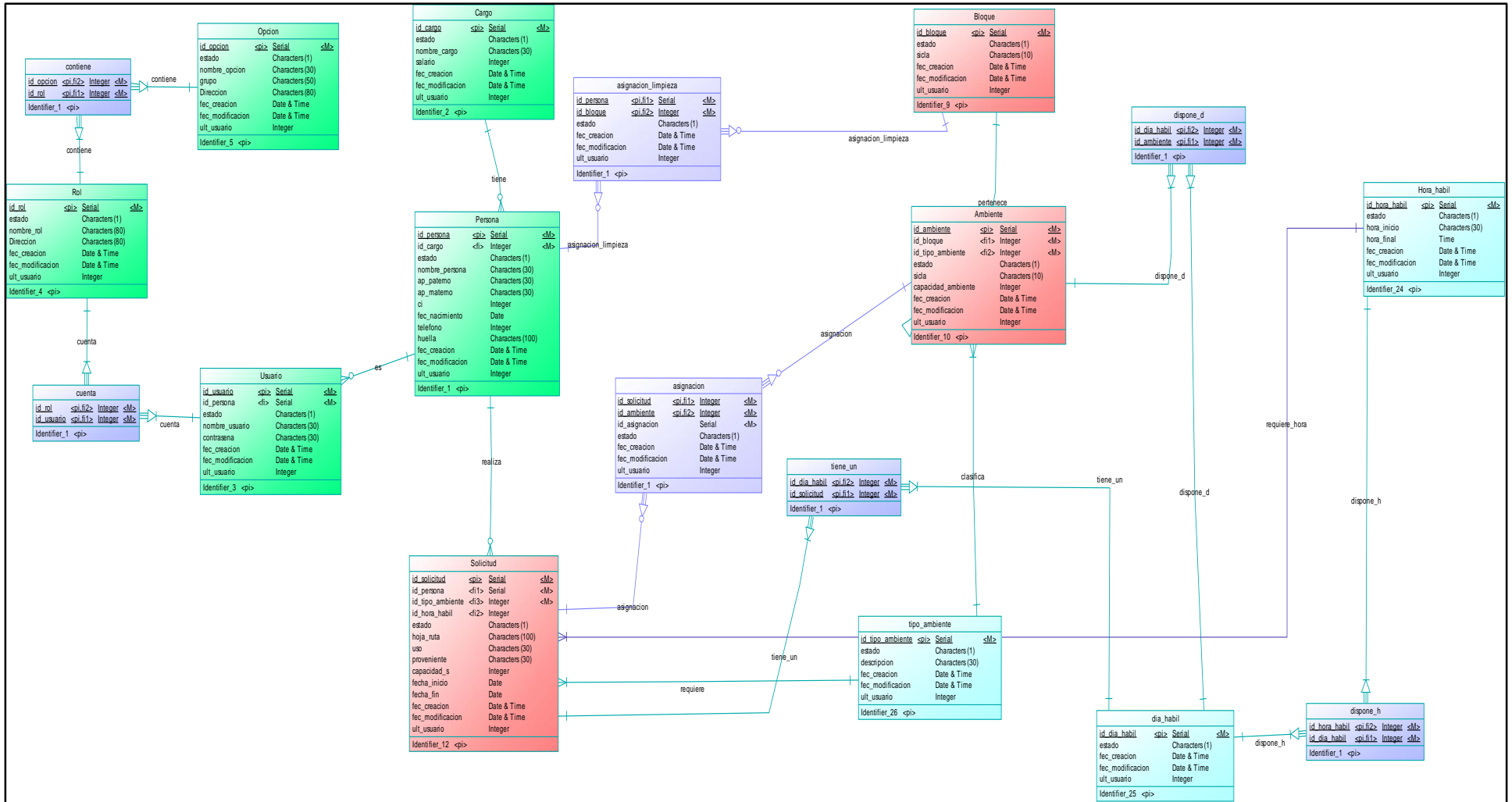


Diagrama 3.32: Modelo Logico de la Base de Datos del sistema de información SAAC-UAP
Fuente: Elaboración propia

Habiendo culminado con los requerimientos, el análisis, diseño del sistema y concluido el diseño de la base de datos, queda realizar la conclusión de los módulos utilizando las herramientas mencionadas anteriormente. A continuación se presentan las interfaces del sistema ya desarrollado según a los requerimientos del usuario.

3.1.3.2 DICCIONARIO DE DATOS

La base de datos diseñada está compuesta de 12 tablas y 3 tablas intermedias acorde a las necesidades para la asignación aulas de la unidad de administración de ambientes del campus universitario.

A continuación se describirá los atributos de cada tabla de la base de datos y su correspondiente observación de ser necesaria:

Tabla persona

Persona					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_persona	serial	Pk	No		Identificador de la tabla
Id_cargo	Integer				
estado	String		No		
Nombre_persona	String		No		
Ap_paterno	String		No		
CI	integer		No		
Fec_nacimiento	Date		No		
telefono	integer		No		
Fec_creacion	Date		No		Fecha de creación de cada
Fec_modificacion	Date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del último usuario que accedió al sistema.

Tabla 3.16: tabla persona del modelo entidad

Fuente: Elaboración propia

Tabla Usuario

Usuario					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_usuario	serial	Pk	No		Identificador único de la tabla
Id_persona	serial			fk	
estado	String		No		
Nombre_usuario	String		No		
contraseña	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del ultimo usuario que accedió al sistema.

Tabla 3.17: tabla usuario del modelo entidad

Fuente: Elaboración propia

Tabla Rol

Rol					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_rol	serial	Pk	No		Identificador único de la tabla
estado	String		No		
Nombre_rol	String		No		
Dirección	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del ultimo usuario que accedió al sistema.

Tabla 3.18: tabla rol del modelo entidad

Fuente: Elaboración propia

Tabla Cargo

Cargo					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_cargo	serial	Pk	No		Identificador único de la tabla
estado	String		No		
Nombre_cargo	String		No		
salario	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		no		Código del último usuario que accedió al sistema.

Tabla 3.19: tabla Cargo del modelo entidad

Fuente: Elaboración propia

Tabla Opcion

Opcion					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_opcion	serial	Pk	No		Identificador único de la tabla
estado	String		No		
Nombre_opcion	String		No		
grupo	String		No		
Dirección	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del ultimo usuario que accedió al sistema.

Tabla 3.20: tabla Opción del modelo entidad

Fuente: Elaboración propia

Tabla Solicitud

Solicitud					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_solicitud	serial	Pk	No		Identificador único de la tabla
Id_persona	serial			Fk	
Id_tipo_ambiente	serial			Fk	
Id_hora_habil	serial			Fk	
estado	String		No		
Hoja_ruta	String		No		
uso	String		No		
proveniente	String		No		
Capacidad_s	integer		No		
Fec_inicio	date		No		
Fec_fin	date		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del último usuario que accedió al sistema.

Tabla 3.21: tabla Solicitud del modelo entidad

Fuente: Elaboración propia

Tabla Bloque

Bloque					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_bloque	serial	Pk	No		Identificador único de la tabla
estado	String		No		
Sigla	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del ultimo usuario que accedió al sistema.

Tabla 3.22: tabla bloque del modelo entidad

Fuente: Elaboración propia

Tabla Tipo ambiente

Tipo Ambiente					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_tipo_ambiente	serial	Pk	No		Identificador único de la tabla
estado	String		No		
descripcion	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		No		Código del ultimo usuario que accedió al sistema.

Tabla 3.23: tabla tipo ambiente del modelo entidad

Fuente: Elaboración propia

Tabla Ambiente

Ambiente					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_ambiente	serial	Pk	No		Identificador único de la tabla
Id_bloque	integer		No		
Id_tipo_ambiente	integer		No		
Estado	String		No		
sicla	String		No		
Capacidad_ambiente	integer		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		no		Código del ultimo usuario que accedió al sistema.

Tabla 3.24: tabla ambiente del modelo entidad

Fuente: Elaboración propia

Tabla Dia habil

Dia_habil					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_dia_habil	serial	Pk	No		Identificador único de la tabla
Estado	String		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		no		Código del ultimo usuario que accedió al sistema.

*Tabla 3.25: tabla dia habil del modelo entidad**Fuente: Elaboración propia***Tabla Hora_hábil**

Hora_habil					
Atributo	Tipo	Key	Null	FK requerida a tabla	Observación
Id_hora_habil	serial	Pk	No		Identificador único de la tabla
Estado	String		No		
hora_inicio	time		No		
Hora_final	time		No		
Fec_creacion	date		No		Fecha de creación de cada
Fec_modificacion	date		No		Fecha en la cual se realizó la última modificación
Ult_usuario	integer		no		Código del último usuario que accedió al sistema.

*Tabla 3.26: tabla hora habil del modelo entidad**Fuente: Elaboración propia*

Una vez culminado con los requerimientos, el análisis, diseño del sistema y concluido el diseño de la base de datos, queda realizar la construcción de los módulos utilizando las herramientas mencionadas en el anterior capítulo. A continuación se presentan las interfaces del sistema ya desarrollado según a los requerimientos del usuario.

3.1.3.3 INTERFAZ DEL SISTEMA

La interfaz de usuario es el mecanismo a través del cual se establece un diálogo entre el sistema y el usuario, la comunicación puede ser físico mediante le teclado o símbolo por medio de iconos

Las siguientes imágenes muestra la interfaz del sistema informático para el control de ambientes del campus universitario ya concluido de acuerdo a los alcances definidos en el primer capítulo.

- **Control de acceso al sistema**

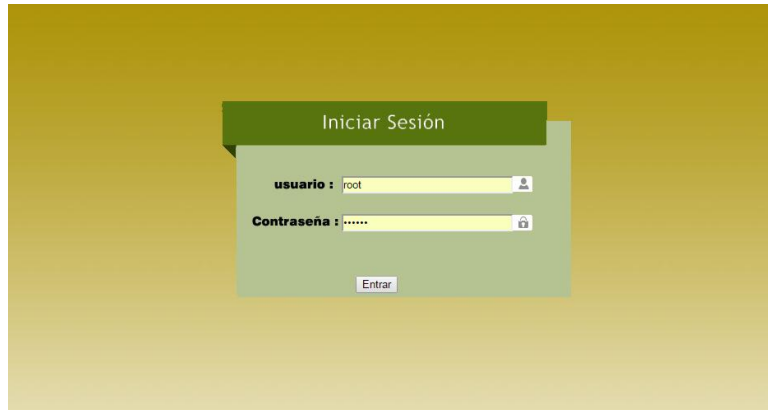


Figura 3.2: Interfaz de validación de usuario

Fuente: Elaboración propia

Esta interfaz permite el acceso al sistema por autenticación de usuarios donde cada usuario puede acceder al sistema de acuerdo al rol que le corresponda debe ingresar su nombre de usuario y su contraseña asignado por el usuario administrador del sistema de esta se establece un control eficiente para acceder al sistema.

- **Interfaz de usuario**

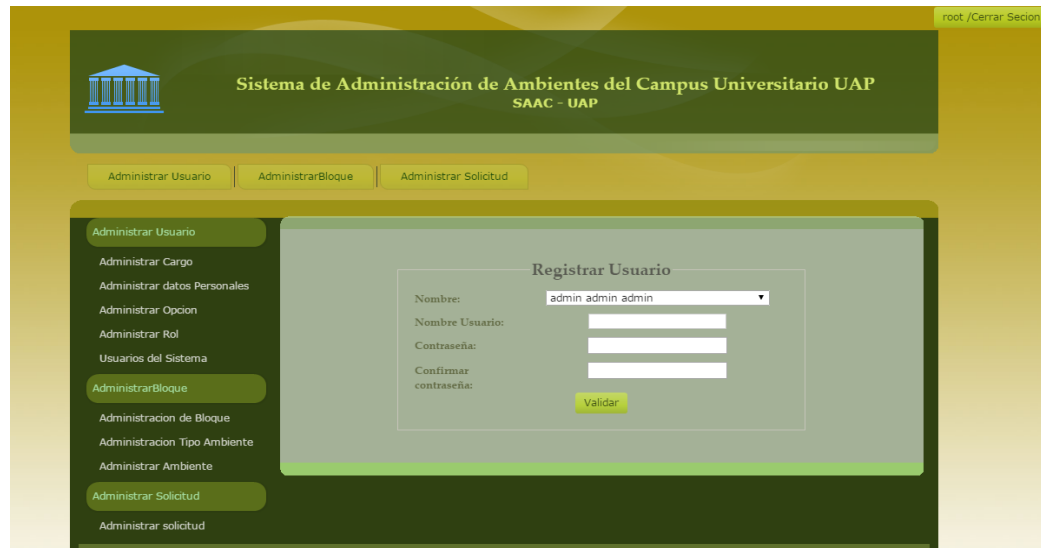


Figura 3.3: Interfaz de usuario

Fuente: Elaboración propia

En la figura 3.2 se presenta la pantalla principal del Sistema, en la cual se podrá acceder a los diferentes módulos para realizar las funciones asignadas tanto en el menú horizontal como en el vertical.

- **Registro de usuario**



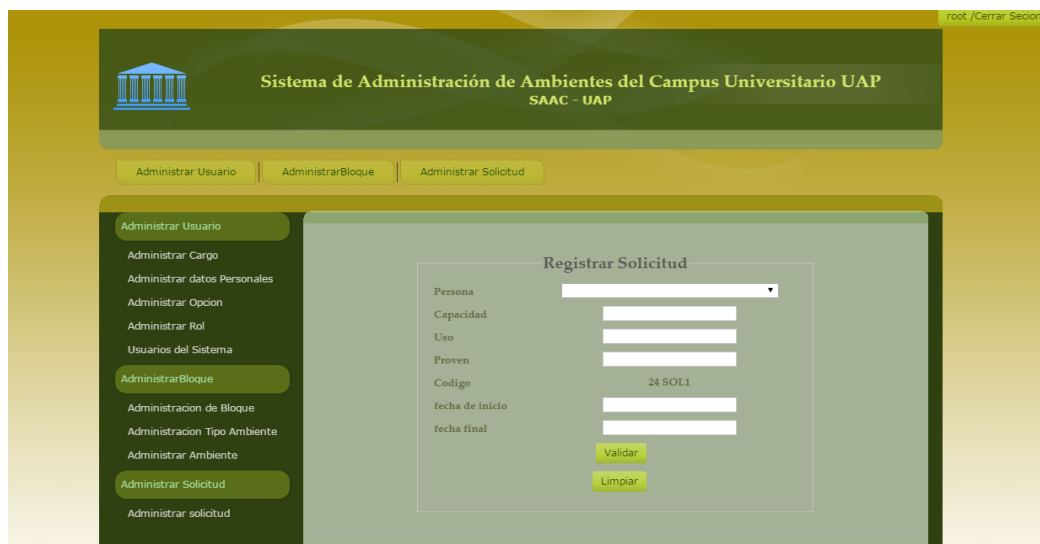
The screenshot shows the main interface of the 'Sistema de Administración de Ambientes del Campus Universitario UAP SAAC - UAP'. At the top, there is a logo of a classical building and the system name. Below the logo, there are three main menu items: 'Administrar Usuario', 'AdministrarBloque', and 'Administrar Solicitud'. A vertical sidebar on the left contains a list of sub-menus under each of these categories. The 'Administrar Usuario' sub-menu is currently selected, and the 'Registrar Usuario' form is displayed in the main content area. The form includes fields for 'Nombre' (with a dropdown menu showing 'admin admin admin'), 'Nombre Usuario', 'Contraseña', and 'Confirmar contraseña', along with 'Validar' and 'Limpiar' buttons.

Figura 3.4: Interfaz de bloque

Fuente: Elaboración propia

La figura 3.4 se visualiza el registro de usuario anteriormente debe registrar sus datos personales y el cargo que ocupa dentro de la unidad de administración, luego pasa a registrarse como usuario.

➤ **Registro de solicitud**



The screenshot shows the same main interface as Figure 3.4, but with the 'Administrar Solicitud' sub-menu selected. The 'Registrar Solicitud' form is displayed in the main content area. It includes fields for 'Persona' (with a dropdown menu), 'Capacidad', 'Uso', 'Proven', 'Codigo' (with the value '24 SOL1'), 'fecha de inicio', and 'fecha final'. There are 'Validar' and 'Limpiar' buttons at the bottom of the form.

Figura 3.5: Registro de solicitud

Fuente: Elaboración propia

➤ **Solicitudes pendientes**

Codigo	Estado	Uso	Proveniente	Fecha de Inicio	Fecha Final	Asignar
17 SOL1	P	rerer	erer	2015-07-09	2015-07-09	
18 SOL1	P	AQUI	dg	2015-07-10	2015-07-10	
19 SOL1	P	capacitacion	acyt	2015-07-11	2015-07-12	
20 SOL1	P	capacitacion	acyt	2015-07-12	2015-07-14	
21 SOL1	P	capacitacion	acyt	2015-07-12	2015-07-13	
22 SOL1	P	capacitacion	acbn	2015-07-12	2015-07-14	

Figura 3.6: solicitudes pendientes

Fuente: Elaboración propia

En la figura 3.7 muestra el listado de las solicitudes que están pendientes para su posterior asignación de ambientes dependiendo si el ambientes se encuentra disponible la solicitud es asignada con éxito si no es denegada.

- **Asignación de ambientes**

En la figura 3.8 esta interfaz permite realizar la asignación de ambientes de acuerdo a los ambientes y horarios disponibles.

Proveniente	Uso	Fechas	Codigo
MARCELO FEDERICO	std	2015-05-19 - 2015-06-19	12 SOL1

Horario: Dia: Ambiente:

Asignar Hecho Denegar

Figura 3.7: Interfaz de asignación de ambientes

Fuente: Elaboración propia

3.1.4 FASE IV: FASE DE TRANSICION

En esta fase inicia la etapa de implementación del sistema dentro de la unidad de administración de ambientes del campus universitario. Para esta fase se realiza la utilización de los diagramas de despliegue y el diagrama de componentes.

3.1.4.1 DIAGRAMA DE DESPLIEGUE

Con este diagrama podemos observar que el funcionamiento del sistema se despliega en una red LAN donde se compone por un servidor y 2 clientes que representa a los actores identificados, en caso de contar con una red de computadoras el sistema no presentara ningún problema en su funcionamiento ni en el acceso de usuarios ya que la arquitectura del sistema está desarrollado para que funcione en una red o simplemente en una sola computadora.

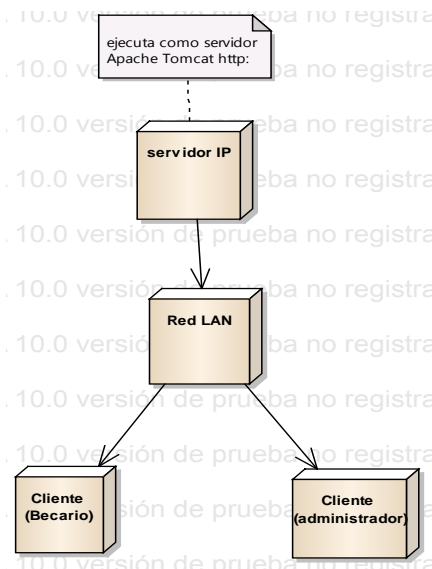


Diagrama 3.33: de Despliegue: Sistema informático para el control de ambientes
Fuente: Elaboración propia

3.1.4.2 PRUEBAS DE INTEGRACION

Prueba de integración Autenticar			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
nom_usuario, contr_usuario			
root , 123456	nom_usuario: presente contr_usuario: presente	Mensaje de bienvenida	nom_usuario == BD_nombre_usuariocontr_usuario == BD_contr_usuario
root , 123456	nom_usuario: presente contr_usuario: presente	Mensaje de usuario no encontrado	nom_usuario == BD_nombre_usuariocontr_usuario != BD_contr_usuario
root , 123457	nom_usuario: presente contr_usuario: presente	Mensaje de usuario no encontrado	nom_usuario != BD_nombre_usuariocontr_usuario == BD_contr_usuario
, 123457	nom_usuario: no presente contr_usuario: presente	Mensaje de error en nom_usuario	
root,	nom_usuario: presente contr_usuario: no presente	Mensaje de error en contr_usuario	
,	nom_usuario: no presente contr_usuario: no presente	Mensaje de error en nom_usuario y contr_usuario	

Tabla 3.14: Prueba de integración: Autenticar

Fuente: Elaboración propia

Prueba de integración Registrar usuario			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
id_persona, nom_usuario, contr_usuario, ult_usuario			
1, root, 123456, 1	Id_persona: presente nom_usuario: presente contr_usuario: presente ult_usuario: presente	Mensaje de datos almacenados correctamente	
, root, 123456, 1	Id_persona: no presente nom_usuario: presente contr_usuario: presente ult_usuario: presente	Mensaje de error en id_persona	
, , 123456, 1	Id_persona: no presente nom_usuario: no presente contr_usuario: presente ult_usuario: presente	Mensaje de error en id_persona y nom_usuario	
, , , 1	Id_persona: no presente nom_usuario: no presente contr_usuario: no presente ult_usuario: presente	Mensaje de error en id_persona, nom_usuario y contr_usuario	

Tabla 3.15: Prueba de integración: Registrar Usuario
Fuente: *Elaboración propia*

Prueba de integración Modificar usuario			
Entrada	Condición de entrada	Salida esperada	Condiciones de salida
Id_usuario, id_persona, nom_usuario, contr_usuario, ult_usuario			
1, 1, root, 123456, 1	Id_usuario: presente Id_persona: presente nom_usuario: presente contr_usuario: presente ult_usuario: presente	Mensaje los datos fueron modificados correctamente	
1, 1, root, , 1	Id_usuario: presente Id_persona: presente nom_usuario: presente contr_usuario: no presente ult_usuario: presente	Mensaje de error en contr_usuario	
1, 1, , , 1	Id_usuario: presente Id_persona: presente nom_usuario: no presente contr_usuario: no presente ult_usuario: presente	Mensaje de error en nom_usuario y contr_usuario	
1, , , , 1	Id_usuario: no presente Id_persona: no presente nom_usuario: no presente contr_usuario: no presente ult_usuario: presente	Mensaje de error en Id_persona, nom_usuario y contr_usuario	
, 1, root, 123456, 1	Id_usuario: no presente Id_persona: presente nom_usuario: presente contr_usuario: presente ult_usuario: presente	Mensaje de error en id_usuario	

Tabla 3.16: Prueba de integración: Modificar Usuario

Fuente: Elaboración propia

Prueba de integración Registrar Bloque			
Entrada	Condicion de entrada	Salida esperada	Condiciones de salida
sicla			
F	sicla: presente	Mensaje "Sus datos fueron ingresados correctamente"	Dato registrado en BD
null	sicla: no presente	Mensaje "Ingrese datos"	Dato registrado en BD

Tabla 3.17: Prueba de integración: Registrar bloque

Fuente: Elaboración propia

3.1.4.3 PRUEBAS UNITARIAS

Prueba unitaria Autenticar	
Entrada	Salida
Nom_usuario , cont_usuario	Id_usuario
Nom_usuario , cont_usuario	Null

Tabla 3.18: Prueba unitaria: Autenticar

Fuente: Elaboración propia

Prueba unitaria Registrar Bloque	
Entrada	Salida
sicla, ult_usuario	1
Sicla , null	0
Null , ult_usuario	0
Null , null	0

Tabla 3.19: Prueba de unitaria: Registrar bloque

Fuente: Elaboración propia

Prueba unitaria Registrar Ambiente	
Entrada	Salida
Id_bloque, id_tipo_ambiente, sicla, capacidad, ult_usuario	1
null, id_tipo_ambiente, sicla, capacidad, ult_usuario	0
Id_bloque, null, sicla, capacidad, ult_usuario	0
Id_bloque, id_tipo_ambiente, null, capacidad, ult_usuario	0
Id_bloque, id_tipo_ambiente, sicla, null, ult_usuario	0
Id_bloque, id_tipo_ambiente, sicla, capacidad, null	0

Tabla 3.20: Prueba de integración: Registrar bloque

Fuente: Elaboración propia

CAPITULO IV
CONCLUSIONES Y
RECOMENDACIONES

4.1. CONCLUSIONES

- Se ha diseñado un sistema que se pueda integrar con nuevos desarrollos siendo adaptable a las necesidades de la unidad de administración de ambientes.
- Con el uso de herramientas de software libre, una de las ventajas es el bajo o nulo coste de los productos libres que permiten proporcionar a las empresas medianas y pequeñas permisos, ampliar sus infraestructuras sin que se vean reducidos sus intentos de crecimiento por no poder hacer el pago de las licencias.
- Se ha mejorado el proceso de asignación de aulas y laboratorios mediante el sistema ya que se puede ver que aulas están en uso, también la persona encargada de realizar la asignación ya no demorara mucho tiempo.
- Los reportes permiten ayudar al administrador de ambientes a apreciar e interpretar la información de una manera más rápida y eficiente.

4.2. RECOMENDACIONES

- Se recomienda para la obtención de requerimientos hacer entrevistas planificadas porque de ese modo se podrá obtener los requerimientos reales del sistema.
- Se recomienda durante el proceso de desarrollo una comunicación continua con el cliente para el tratamiento de asuntos concernientes al sistema ya que este será el usuario final que manipule el sistema.
- Se recomienda en la captura de requerimientos por parte de los usuarios utilizar prototipos por la riqueza interpretativa que conlleva.

BIBLIOGRAFIA

- UAP.2011, *www.uap.edu.bo*. (s.f.). Recuperado el 28 de 10 de 2013, de *www.uap.edu.bo*:
<http://www.uap.edu.bo/index.php/launiversidad/resena-historica>
- Kendall, K. (1991). *Análisis y Diseño de Sistemas*. . México: Prentice HillCaballero, J. M. (2010). *Asignacion de Horarios de Clases Universitarias Mediante Algoritmos Evolutivos*. Colombia: Universidad de la Guajira, Riochacha.
- cadario, I. (s.f.). *uap.bo.edu*. Recuperado el 18 de 06 de 2014, de *www.uap.edu.bo*
- Fontela, C. (2011). *Modelado de Software para Profesionales*. Mexico: Alfaomega Grupo Editor.
- Galio, R. (2012). *Desarrollo e implementación de un Sistema de Gestión de Asignación de Aulas y Laboratorios*. Guaquil - Ecuador: Escuela Superior Politecnica del Litoral.
- Ivar Jacobson, G. B. (1999). *El Proceso Unificado de Desarrollo de Software* . Pearson Addisson-Wesley.
- Larma, C. (1999). *UML y Patrones 1ra Edicion*. Mexico.
- Martinez, R. (2 de 10 de 2010). *www.postgresql.org*. Recuperado el 2 de 9 de 2013, de *www.postgresql.org.es.htm*
- McFarland, D. S. (2012). *JavaScript y jQuery*. Madrid: ANAYA MULTIMEDIA.
- Mercado, W. (Abril de 2010). *SISTEMA DE GESTIÓN DEL DESEMPEÑO PARA LABORATORIOS INFORMÁTICOS UTILIZANDO UN MODELO ESTRATÉGICO DECISIONAL BASADO EN BPM Y BI*. Guayana.
- Mora, F. (15 de 03 de 2002). *UML lenguaje unificado de modelado*. Universidad alicante. Obtenido de <http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>.
- O'Brien. (2001). *Sistema de Informacion Gerencial*. McGraw Hill Interamericana S.A.
- oracle. (s.f.). *www.oracle.com*. Recuperado el 26 de 06 de 2015, de *www.oracle.com/technetwork/java/index.html*
- Torossi, A. G. (2010). *El Proceso Unificado de Desarrollo de Software*. españa.

ANEXOS A

ARBOL DE PROBLEMAS

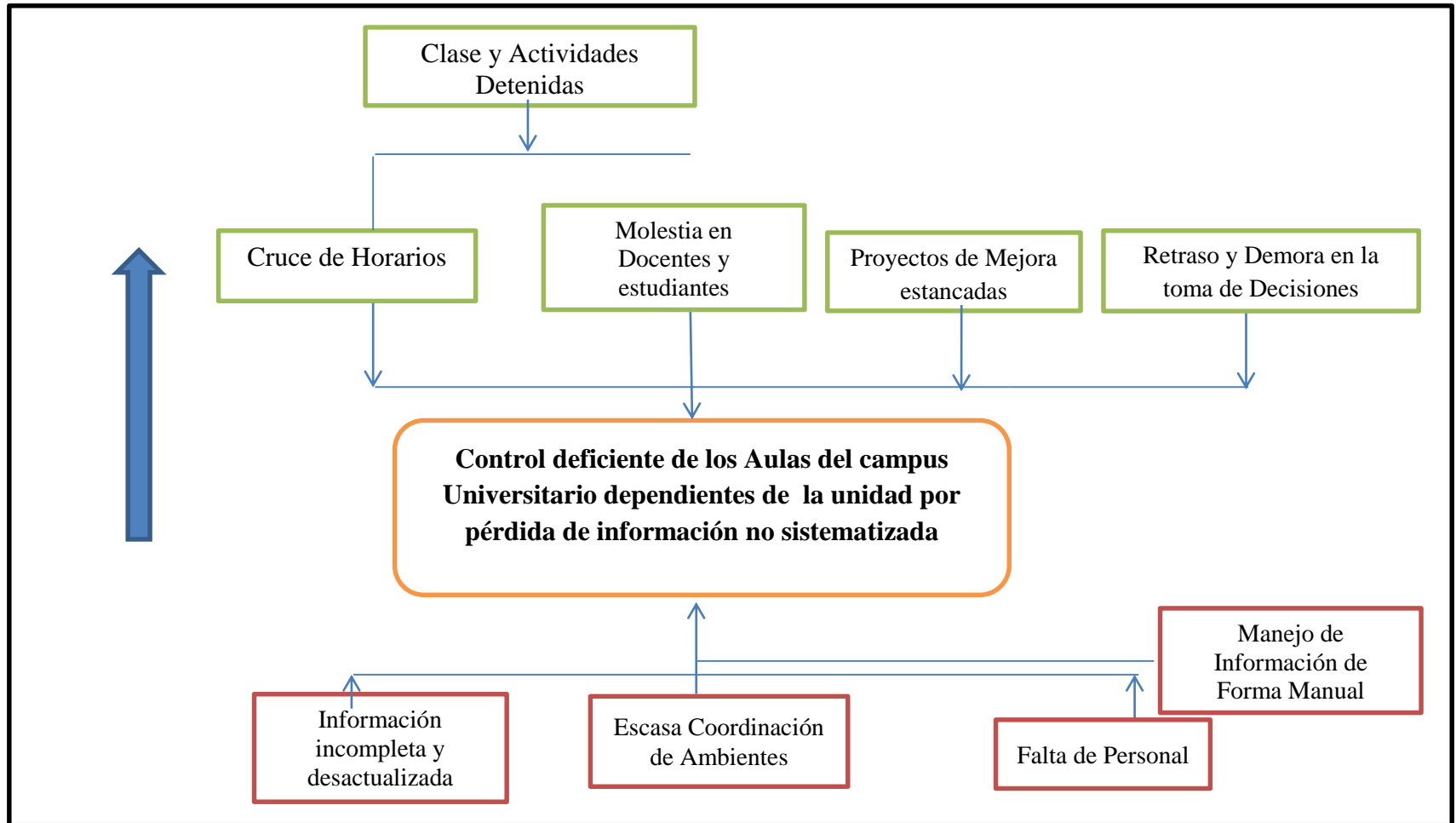


Diagrama A.1: Árbol de problemas
Fuente: Elaboración propia

ARBOL DE OBJETIVOS

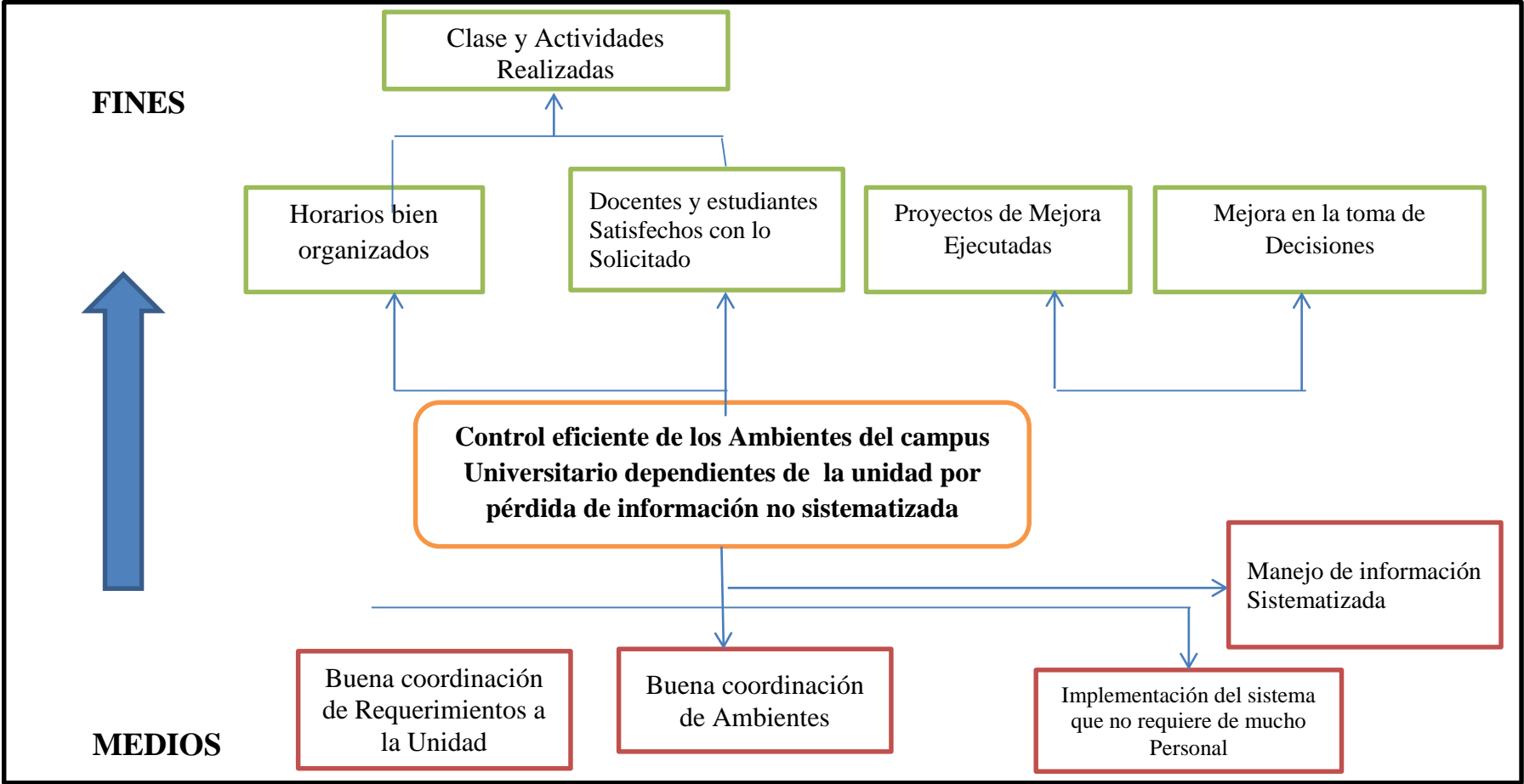


Diagrama A.2: Árbol de Objetivos
Fuente: Elaboración propia

CUADRO DE INVOLUCRADOS

GRUPOS	INTERÉS	PROBLEMAS PERSIVIDOS	RECURSOS Y MANDATOS
Vicerrectorado de la universidad amazónica de pando	- Mejorar continuamente y sistemáticamente los servicios académicos y de información con el uso de las modernas Tecnologías de la Información y Comunicación TIC.	- Capacidad de inversión limitada.	- Tiene una unidad que cuenta con recursos Económicos.
Unidad de administración de ambientes del campus universitario	- Contar con un sistema de informático para el control de ambientes de la UAP, para brindar información oportuna y confiable	- Información incompleta debido al manejo de forma manual de la información para la toma de decisiones emergentes.	- Se cuenta con todos los recursos a disposición, normas y estatutos que rigen el buen funcionamiento de la unidad
Direcciones de área	- Obtener ambientes de acuerdo a lo requerido por parte de los docentes	- Retardo en la asignación de ambientes de acuerdo a lo solicitado.	- Cuenta con recursos anualmente mediante el POA
Población universitaria	- Contar con un sistema para el control de ambientes del campus de la UAP.	- Retardo en la asignación de ambientes de acuerdo a lo solicitado.	- cuenta con recursos económicos del IDH mediante la federación universitaria local.

Diagrama A.3: Cuadro de Involucrado

Fuente: *Elaboración propia*

ANEXOS B

ANEXOS C

CUESTIONARIO DE REQUERIMIENTOS

Nombre:.....
.....

Cargo:.....
...../11/2012

Fecha:

¿Esta unidad cuenta con algún reglamento?

Fase de conocimiento del personal administrativo

¿Cómo está compuesta el organigrama de la Administración del Campus Universitario?

¿Con cuanto personal cuenta la unidad para sus distintas actividades?

Fase de las actividades de los administrativos

¿Quién es el responsable de la asignación de horario de limpieza?

¿Quién es el responsable de la asignación de horario y Ambientes en el Campus?

¿Quién es el responsable de la asignación de horarios de los Becarios?

¿Quién es el responsable de dar de baja a los activos?

¿Quiénes realizan el proceso de recolección de datos y que tipos de datos recolectan?

Fase para el proceso de recolección de datos

¿Quiénes pueden hacer las solicitudes de Aulas?

¿Tiene información de la capacidad y características de todos las Aulas?

Si es cierto que medios utiliza para almacenar dicha información

- a) Herramientas informáticas como ser Word, Excel.
- b) Herramientas de escritorio como ser papeles, folders
- c) Otros.....

¿Como organiza la información de Ambientes Requeridos?

- a) Utiliza alguna herramienta informática como ser Word, Excel.
- b) Utiliza material de escritorio como ser papel, lápiz.
- c) Otros.....

¿Con cuanto de anticipación se debe presentar una solicitud?

Fase de conocimiento de los activos fijos

¿Qué tipos de activos fijos tiene a su cargo?

¿Hace el control de los activos?

Si

- a) ¿Con cuanta frecuencia hacen el control de los activos fijos y quienes lo realizan?

No

¿Necesita usted un Registro de los bienes asignados y que datos necesita almacenar?

Fase de conocimiento de los Ambientes

¿Que tipos de Ambientes tiene bajo su cargo?

- a) Ambientes informáticos como ser laboratorios
- b) Ambientes de uso común como ser Aulas
- c) Salón
- d) Coliseo
- e) Otros.....

¿Con cuántos Aulas dispone para la asignación de horarios?

Otros

¿Que medios consulta para realizar el control de las actividades que se deben realizar?

¿Tiene equipos (PC) disponibles para el control del campus?

Si es cierto ¿estas cuentan con una red interna?

ANEXOS D