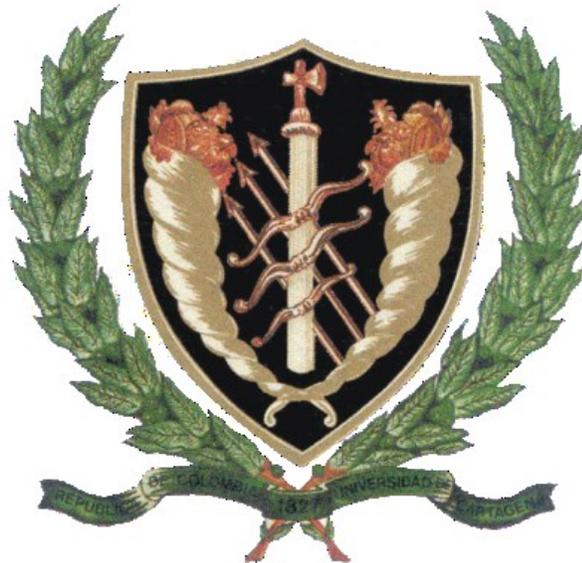


PLATAFORMA VIRTUAL PARA EL REGISTRO Y CONTROL DE  
ASISTENCIA ACADÉMICA EN LA UNIVERSIDAD DE CARTAGENA

INVESTIGADORES

Ever Cabarcas Mallarino

Jonathan Camilo Cantillo Guerrero



UNIVERSIDAD DE CARTAGENA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

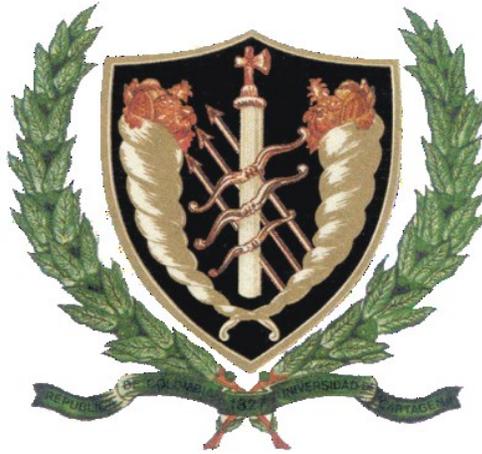
CARTAGENA DE INDIAS, 2019

PLATAFORMA VIRTUAL PARA EL REGISTRO Y CONTROL DE  
ASISTENCIA ACADÉMICA EN LA UNIVERSIDAD DE CARTAGENA

TESIS DE GRADO

Grupo de investigación: E-Soluciones

Línea de investigación: E-Servicios



INVESTIGADORES

Ever Cabarcas Mallarino

Jonathan Camilo Cantillo Guerrero

Director: Martín Monroy Ríos, MSc, PhD.

UNIVERSIDAD DE CARTAGENA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

CARTAGENA DE INDIAS, 2019



**Tesis de Grado:** PLATAFORMA VIRTUAL PARA EL REGISTRO Y CONTROL DE ASISTENCIA ACADÉMICA EN LA UNIVERSIDAD DE CARTAGENA

**Autores:** EVER CABARCAS MALLARINO  
JONATHAN CAMILO CANTILLO GUERRERO

**Director:** Msc., PhD. MARTÍN MONROY RÍOS

**Nota de Aceptación**

---

---

---

---

---

**Presidente del Jurado**

---

---

**Jurado**

**Jurado**

Cartagena de Indias, \_\_\_\_ de \_\_\_\_\_ de 2019

## TABLA DE CONTENIDO

AGRADECIMIENTOS Y DEDICATORIA .....	11
RESUMEN.....	12
ABSTRACT .....	13
1 INTRODUCCIÓN .....	14
1.1 ANTECEDENTES .....	14
1.2 PLANTEAMIENTO DEL PROBLEMA .....	16
1.2.1 Descripción del problema .....	16
1.2.2 Formulación del problema .....	19
1.3 JUSTIFICACIÓN.....	21
1.4 OBJETIVOS.....	25
1.4.1 Objetivo General .....	25
1.4.2 Objetivos Específicos .....	25
1.5 ALCANCE .....	26
2 MARCO DE REFERENCIA.....	29
2.1 ESTADO DEL ARTE.....	29
2.2 MARCO TEÓRICO .....	36
2.2.1 Registro de asistencias .....	36
2.2.2 Recurso digital.....	37
2.2.3 Plataforma virtual .....	38
2.2.4 RUP.....	39
2.2.5 UML.....	42
2.2.6 HTML5, JavaScript y CSS3.....	43
2.2.7 NoSQL Database .....	44

2.2.8	MongoDB.....	46
2.2.9	Express.....	49
2.2.10	Mongoose.....	49
2.2.11	Ionic.....	49
2.2.12	V8.....	50
2.2.13	GIT.....	51
2.2.14	REST API's.....	51
2.2.15	Arquitectura basada en componentes.....	53
3	METODOLOGÍA.....	55
4	RESULTADOS Y DISCUSIÓN.....	60
4.1	MODELO DE NEGOCIO.....	61
4.1.1	Modelo de dominio.....	61
4.1.2	Casos de uso del mundo real.....	62
4.1.3	Procesos del negocio.....	64
4.2	ESPECIFICACIÓN DE REQUISITOS.....	66
4.3	MODELO DE DISEÑO.....	67
4.3.1	Vista de escenarios.....	67
4.3.2	Vista lógica.....	69
4.3.3	Vista de procesos.....	85
4.3.4	Vista de desarrollo.....	109
4.3.5	Vista física.....	111
4.4	PRUEBAS.....	113
4.4.1	Prueba de caja negra.....	113
4.4.2	Validación del sistema.....	114

4.5	ANÁLISIS DE RESULTADOS.....	122
5	CONCLUSIONES .....	125
6	RECOMENDACIONES .....	128
7	ANEXOS.....	129
	ANEXO A: Diseño de entrevista para la identificación del problema de registro y control de asistencia y temas vistos en la Universidad de Cartagena .....	129
	ANEXO B: ACTA N° 1. Correspondiente a la entrevista realizada como método de recolección de información para conocer el problema en cuestión .....	129
	ANEXO C: Diseño de entrevista para el desarrollo e implementación de una plataforma móvil y web para un sistema de registro y control de asistencia y temas vistos en la Universidad de Cartagena .....	129
	ANEXO D: ACTA N° 2. Correspondiente a la entrevista realizada como método de recolección de información para el desarrollo de un sistema software como plataforma móvil y web enfocada en el registro y control de asistencia y temas vistos en la Universidad de Cartagena.....	130
	ANEXO E: Diseño de entrevista para conocer los sistemas de control de asistencia de la Fundación Universitaria Tecnológico Comfenalco.....	130
	ANEXO F: ACTA N° 3. Correspondiente a la entrevista realizada como método de recolección de información sobre un software para controlar la asistencia en la Fundación Universitaria Tecnológico Comfenalco .....	130
	ANEXO G: Especificación de requisitos del software (SRS).....	130
	ANEXO H: Diagrama de actividades en formato digital.....	131
	ANEXO I: Diagrama de clases funcionalidad gestionar departamentos. ....	131
	ANEXO J: Diagrama de clases funcionalidad gestionar proyectos docentes. ....	131
	ANEXO K: Diagrama de clases funcionalidad realizar registros académicos.....	131
	ANEXO L: Diagrama de clases funcionalidad visualizar cursos. ....	131

ANEXO M: Diagrama de clases funcionalidad visualizar registros académicos.....	131
ANEXO N: Diagrama de clases funcionalidad registrar comentarios sobre registros académicos.....	132
ANEXO O: Diagrama de clases funcionalidad visualizar estadísticas de cursos y registros. ....	132
ANEXO P: Diagrama de componentes. ....	132
ANEXO Q: Diagrama de secuencia del caso de uso: Consultar base de datos. ....	132
ANEXO R: Diagrama de secuencia del caso de uso: Cargar departamentos.....	132
ANEXO S: Diagrama de secuencia del caso de uso: Crear departamento.....	132
ANEXO T: Diagrama de secuencia del caso de uso: Editar departamento. ....	133
ANEXO U: Diagrama de secuencia del caso de uso: Eliminar departamento. ....	133
ANEXO V: Diagrama de secuencia del caso de uso: Visualizar departamento.....	133
ANEXO W: Diagrama de secuencia del caso de uso: Cargar proyectos docente. ....	133
ANEXO X: Diagrama de secuencia del caso de uso: Crear proyecto docente. ....	133
ANEXO Y: Diagrama de secuencia del caso de uso: Editar proyecto docente.....	133
ANEXO Z: Diagrama de secuencia del caso de uso: Eliminar proyecto docente. ....	133
ANEXO AA: Diagrama de secuencia del caso de uso: Visualizar proyecto docente.	134
ANEXO AB: Diagrama de secuencia del caso de uso: Iniciar sesión. ....	134
ANEXO AC: Diagrama de secuencia del caso de uso: Procesar mensaje de error. ...	134
ANEXO AD: Diagrama de secuencia del caso de uso: Realizar registros académicos. ....	134
ANEXO AE: Diagrama de secuencia del caso de uso: Registrar comentarios sobre registros académicos. ....	134
ANEXO AF: Diagrama de secuencia del caso de uso: Visualizar cursos.....	134

ANEXO AG: Diagrama de secuencia del caso de uso: Visualizar estadísticas de cursos y registros. ....	135
ANEXO AH: Diagrama de secuencia del caso de uso: Visualizar registros académicos. ....	135
ANEXO AI: Diagrama paquetes. ....	135
ANEXO AJ: Diagrama despliegue. ....	135
ANEXO AK: Diagrama general de casos de uso. ....	135
ANEXO AL: Diseño de pruebas de carga.....	135
ANEXO AM: Diseño de pruebas de caja negra. ....	136
ANEXO AN: Resultados de pruebas funcionales: Actor administrador. ....	136
ANEXO AO: Resultados de pruebas funcionales: Actor docente - auditor.....	136
ANEXO AP: Resultados de pruebas funcionales: Actor estudiante.....	136
8 REFERENCIAS BIBLIOGRÁFICAS.....	137

## TABLA DE ILUSTRACIONES

Ilustración 1. Cartilla de registro de asistencia.....	20
Ilustración 2. Formulario para la consignación de información del registro de asistencia. .	21
Ilustración 3. MongoDB y el trabajo de los datos .....	47
Ilustración 4. MongoDB ubicación de datos .....	48
Ilustración 5. MongoDB distribuido .....	48
Ilustración 6. Modelo de dominio .....	62
Ilustración 7. Casos de uso a nivel de mundo real.....	63
Ilustración 8. Procesos del negocio.....	65
Ilustración 9. Diagrama de casos de uso a nivel de requisitos. ....	66
Ilustración 10 Diagrama general de casos de uso .....	68
Ilustración 11 – Gestionar departamentos.....	70
Ilustración 12 – Gestionar proyectos docentes.....	72
Ilustración 13 – Realizar registros académicos .....	74
Ilustración 14 – Visualizar cursos.....	76
Ilustración 15 – Visualizar registros académicos .....	78
Ilustración 16 – Registrar comentarios sobre registros académicos .....	80
Ilustración 17 – Visualizar estadísticas de cursos y registros .....	82
Ilustración 18 – Diagrama de componentes .....	84
Ilustración 19 – Diagrama de secuencia del caso de uso: Cargar departamentos .....	86
Ilustración 20 – Diagrama de secuencia del caso de uso: Crear departamento.....	87
Ilustración 21 – Diagrama de secuencia del caso de uso: Editar departamento .....	89
Ilustración 22 – Diagrama de secuencia del caso de uso: Eliminar departamento .....	90
Ilustración 23 – Diagrama de secuencia del caso de uso: Visualizar departamento .....	91
Ilustración 24 – Diagrama de secuencia del caso de uso: Cargar proyectos docente .....	92
Ilustración 25 – Diagrama de secuencia del caso de uso: Crear proyecto docente .....	93
Ilustración 26 – Diagrama de secuencia del caso de uso: Editar proyecto docente .....	95
Ilustración 27 – Diagrama de secuencia del caso de uso: Eliminar proyecto docente .....	96
Ilustración 28 – Diagrama de secuencia del caso de uso: Visualizar proyecto docente .....	97

Ilustración 29 – Diagrama de secuencia del caso de uso: Realizar registro.....	99
Ilustración 30 – Diagrama de secuencia para el caso de uso: Registrar comentarios sobre registros académicos .....	101
Ilustración 31 – Diagrama de secuencia para el caso de uso: Visualizar cursos.....	102
Ilustración 32 – Diagrama de secuencia del caso de uso: Visualizar estadísticas de cursos y registros.....	104
Ilustración 33 – Diagrama de secuencia del caso de uso: Visualizar registros académicos .....	106
Ilustración 34 – Diagrama de secuencia para el caso de uso genérico: Consultar base de datos .....	107
Ilustración 35 – Diagrama de secuencia del caso de uso genérico: Iniciar sesión .....	108
Ilustración 36 – Diagrama de secuencia del caso de uso genérico: Procesar mensaje de error .....	109
Ilustración 37 – Diagrama de paquetes .....	110
Ilustración 38 – Diagrama de despliegue .....	112

## **AGRADECIMIENTOS Y DEDICATORIA**

---

A Dios por iluminar mi camino y llenarme de la sabiduría necesaria para este proceso, a mi madre por ser una mujer tan luchadora y convertirse en un pilar que me hizo ver hacia el futuro, a mi padrino por todo el apoyo incondicional, a mi abuelo que junto a mi madre creyeron en mí con una fe absoluta, a mi maestro y guía Martín Monroy Ríos por su dedicación y paciencia y a todos aquellos ángeles que con sus pequeñas acciones me hicieron ser quien hoy soy.

Ever Cabarcas Mallarino

A Dios por su infinita misericordia y amor, al guiar mis pasos y darme fortaleza para seguir adelante; a mi madre Nubia Guerrero y mi padre Julio Cantillo, quienes siempre han sido mi gran modelo a seguir y mi mayor admiración; a mi abuela Oilen Archbold (Q.E.P.D), porque este logro siempre fue suyo y en vida me dio lo mejor; a mi primo hermano Erick Guerrero, a quien hoy le digo que es ejemplo de vida y que cuenta conmigo siempre; a toda mi familia, quienes siempre me han brindado su cariño, apoyo, respeto y confianza; a mi mentor y guía Martín Monroy Ríos, por su invaluable labor como docente y ser humano; a la Universidad de Cartagena, por la oportunidad de estudiar y formarme en esta gran institución; y a la vida por la maravillosa oportunidad que me brinda para salir adelante y servir a los demás como siempre lo he querido.

Jonathan Camilo Cantillo Guerrero

## RESUMEN

---

Este proyecto está basado en la importancia de mejorar el proceso de registro y control de asistencia académica de la Universidad de Cartagena, teniendo en cuenta el desarrollo continuo de estrategias que permitan adecuarse al uso de las herramientas tecnológicas para procesos de gestión del control académico. Cuando un proceso no se encuentra sistematizado, normalmente produce demora en tiempos de respuesta y prestación de servicios, situación que se agudiza cuando el volumen de datos que se maneja es abundante, generando incluso agotamiento en aquellos encargados de llevar a cabo la gestión, lo que finalmente limita la obtención de información oportuna y confiable. Así, esta investigación planteó como objetivo desarrollar una plataforma virtual para el registro y control de asistencia académica en la Universidad de Cartagena, haciendo uso de la metodología de desarrollo de software RUP.

Para lograr el propósito de esta investigación se realizó una recolección de información a partir de fuentes primarias como la entrevista y la observación directa, y de fuentes secundarias como artículos científicos, tesis de grado y documentación bibliográfica; se generó un modelo de negocios, se realizó la especificación de los requerimientos, se diseñó e implementó el sistema y finalmente se ejecutaron las etapas de pruebas y correcciones necesarias para garantizar el correcto funcionamiento del producto.

Como resultado se obtuvo un sistema multiplataforma para gestionar el proceso de registro de asistencia académica en la Universidad de Cartagena, generando una primera versión estable del producto que facilita, principalmente, el diligenciamiento y almacenamiento de los registros para controlar la asistencia a clases. Se concluye que con la implementación de esta herramienta se obtienen mejoras en dicho proceso en torno a la manipulación, gestión y toma de decisiones oportuna, reducción de tiempo y esfuerzo empleado, disminución del uso de archivos físico y de costos asociados al uso de elementos como papel y/o útiles de papelería, agregando valor a la institución.

## ABSTRACT

---

This project is based on the importance of improving the process of registration and control of academic assistance at the University of Cartagena, taking into account the continuous development of strategies that allow adapting to the use of technological tools for processes for the management of academic control. When a process is not systematized, it usually produces delays in response times and service provision, a situation that becomes more acute when the volume of data handled is abundant, generating exhaustion in those responsible for carrying out the management, limiting the obtaining of timely and reliable information. Thus, this research aimed to develop a virtual platform for registration and control of academic assistance at the University of Cartagena, making use of the RUP software development methodology.

To achieve the purpose of this research, information was collected from primary sources such as interview and direct observation, and from secondary sources such as scientific articles, undergraduate theses and bibliographic documentation; a business model was generated, the requirements were specified, the system was designed and implemented, and finally, the tests and corrections necessary to guarantee the correct operation of the product were executed.

As a result, a multiplatform system was obtained to manage the registration process of academic assistance at the University of Cartagena, generating a first stable version of the product that facilitates, mainly, the filling and storage of records to control attendance at classes. It is concluded that with the implementation of this tool improvements are obtained in this process around the manipulation, management and timely decision making, reduction of time and effort used, decrease in the use of physical files and costs associated with the use of elements such as paper and/or stationery supplies, adding value to the institution.

# INTRODUCCIÓN

---

## 1.1 ANTECEDENTES

A través del tiempo, los avances tecnológicos han permitido en gran medida la automatización de procesos con el fin de garantizar la optimización, mejora y control de su realización. La tecnología de la información reúne, almacena, procesa y garantiza un mejor trato de los datos, disminuyendo los tiempos de ejecución de las tareas que, comúnmente, actúan con demora y serían cuantiosas para el ser humano llevarlas a cabo, ocasionando un manejo inapropiado de la información (Sandoval Illescas, Sigüenza Cárdenas, & Ingavélez G, 2011).

Uno de los procesos que se ha venido efectuando de manera significativa es el de registro y control de asistencia. Éste es considerado de suma importancia en las entidades, no sólo porque determina quién ingresa o abandona un espacio físico concreto, sino que también, tal como Pantoja Blyde, Lozano Leal, & Portillo Montiel (2013) detallan, garantiza la gestión de aspectos de seguridad importantes para toda organización. Igualmente las formas de controlar el registro han evolucionado a lo largo del tiempo: en un principio se hacía uso de una hoja de control que registraba tanto la hora de entrada como la de salida de los empleados; posteriormente, se implementó el uso de una cartilla marcada; hoy día se están utilizando sistemas de acceso y de asistencia por medio de un ordenador que permite la gestión y manejo de los datos de los trabajadores (Varela, 2015).

Múltiples soluciones se han presentado en orden de garantizar el control de asistencia, como por ejemplo el Sistema de Posicionamiento en Interiores desarrollado por la Universidad de Aalborg, que permite la detección de personas en un edificio haciendo uso de tecnologías WI-FI y Bluetooth; el Sistema biométrico de control de asistencia laboral realizado por la Universidad Rafael Urdueta, que permite el registro de entrada y salida de personal por medio de huellas dactilares; o el prototipo implementado para el control de los

empleados dentro de la empresa PRODICAUCHOS LTDA, desarrollado por la Corporación Universitaria Minuto de Dios.

Otra de las áreas de incidencia de este proceso a nivel organizacional, es dentro del sector educativo nacional, donde se presentan muchas fuentes de información de manera no estructurada y dispersa, y donde el registro y control de asistencia, al ser un proceso ejecutado en la mayoría de casos de forma manual, dificulta llevar a cabo un análisis adecuado de la información recolectada, tal como enuncia el Ministerio de Educación Nacional de Colombia (2001).

Por lo anterior, a nivel de la educación superior, los objetivos encaminados en el logro de una educación de calidad son cada vez notorios cuando se hace énfasis en procesos que permitan aumentar las cualidades académicas de sus estudiantes. Y al tener la Universidad de Cartagena como objetivo principal el de educar, está direccionada a implementar procesos que aumenten la calidad y añadan valor a la institución. En este orden de ideas se analizaron los procesos de registro y control de asistencia académica en esta institución de educación superior, identificando falencias tales como: ausencia de un manual de procedimiento del registro y control de asistencia académica, carencia de funciones de verificación y control sobre dichos registros, errores debido al gran número de información manejada, entre otros aspectos detallados posteriormente. Es así que el objeto de la presente investigación fue el desarrollo de una plataforma virtual para el registro y control de asistencia académica en la Universidad de Cartagena.

La investigación realizada permitió conocer a mayor detalle el actual proceso de registro y control de asistencia académica dentro de la Universidad de Cartagena, identificando procesos claves que pudieran ser sistematizados. Además, al incluir dentro de la investigación el proceso de auditoría, se permitió recolectar información de importancia para la institución como porcentaje de asistencia estudiantil y docente, porcentaje de avance de cada uno de los cursos con respecto al proyecto docente asociado, flujo de ejecución del proceso, entre otros aspectos que hasta antes de la investigación no era posible gestionar de forma fácil y detallada.

De igual manera, al sistema proponer una forma estandarizada de llevar a cabo el proceso de registro y control de la asistencia académica, se puede generar un aporte crucial de información minuciosa y detallada e insumos de entrada para otros procesos como el de acreditación institucional, nacional o internacional, donde se requiere información confiable de la forma en la que los procesos académicos se dan, en aras de medir el rendimiento de la universidad. Por lo anterior, la implementación de este sistema cambió la forma de ejecución de los procesos que involucra, buscando entonces disminuir el tiempo y esfuerzo dedicado para su realización y poder ser invertido en labores de análisis productivo para el mejoramiento de la calidad institucional.

La presente investigación se abordó dentro de la línea de investigación de E-Servicios del grupo E-Soluciones, debido a que la solución se desplegó en entornos virtuales como internet, a través de una plataforma web y una aplicación para móviles Android.

## **1.2 *PLANTEAMIENTO DEL PROBLEMA***

### **1.2.1 Descripción del problema**

Hoy día las instituciones educativas se enfocan en trabajar de la mano con el desarrollo continuo de estrategias que permitan adecuarse al uso de las herramientas tecnológicas para procesos de participación estudiantil e interacción docente, regulando las actividades relacionadas con el proceso de aprendizaje y considerando el plan curricular, es decir, la gestión del control académico. Lo anterior porque los procesos educativos están enmarcados en un conjunto de transformaciones sociales respaldadas por la innovación en el manejo de la información, la cual es el activo máspreciado de las instituciones (Zambrano, Numa, & Gómez, 2014).

También es relevante tener en cuenta otra consideración hecha por los autores Zambrano, Numa, & Gómez (2014), quienes manifiestan que cuando un proceso no se encuentra

sistematizado, normalmente produce demora en tiempos de respuesta y prestación de servicios, situación que se agudiza cuando el volumen de datos que se maneja es abundante, generando incluso agotamiento en aquellos encargados de llevar a cabo la gestión, lo que finalmente limita la obtención de información oportuna y confiable.

Por ello, el Ministerio de Educación Nacional de Colombia (2001) desde la fecha ha manifestado su preocupación por el sector educativo nacional, en el cual se presentan múltiples fuentes de información dispersa, no estructurada y cuyo procesamiento se hace de manera manual con gran dificultad para su análisis. Cabe resaltar entonces que la información presentada de esta forma genera problemas como el acceso a archivos históricos, la elevada cantidad de parámetros y criterios de análisis y el origen a una alta probabilidad de inconsistencias que afecten considerablemente la prestación de servicios e información.

Al tener en cuenta las circunstancias expuestas con anterioridad, se identificó una situación de carácter relevante en la Universidad de Cartagena, relacionada con el mecanismo utilizado para implementar el proceso de registro y control de asistencia académica, y el cual se realizaba de forma manual por parte del docente al inicio de cada clase impartida en las respectivas jornadas. A través de una cartilla (ver Ilustración 1) se consignaba información pertinente como la fecha, el(los) tema(s) visto(s) en clase, los alumnos faltantes del día y la firma del profesor (ver Ilustración 2), además de un listado de los estudiantes anexo en la parte posterior. Las cartillas en mención, una vez finaliza la jornada, eran archivadas físicamente en el departamento académico de cada programa académico.

Este proceso originaba una serie de inconvenientes, evidenciados en el acta de la entrevista realizada a la ingeniera Narlinda Espinosa Cantillo -Jefa del Departamento Académico del Programa Ingeniería de Sistemas y de la Facultad de Ingeniería (ver anexo A)-, y en las opiniones aportadas por el ingeniero David Franco – Ex Director de Programa de Ingeniería de Sistemas-. Las conclusiones a las cuales se llegaron en el momento de aplicación de la entrevista fueron las siguientes:

- No se posee un manual de procedimientos que reglamente cuáles son los lineamientos con los que debe contar el proceso, desde la etapa de ejecución a nivel académico hasta la de verificación y control a nivel administrativo. Solamente se hace mención, en el reglamento estudiantil, que el docente debe diligenciar los registros de asistencia, pero no existe un soporte oficial avalado por la Universidad en donde se defina un curso de acción.
- Se carece de funciones de verificación y control sobre los registros de asistencia académica que faciliten la identificación de variables ante sucesos como inasistencia estudiantil, cumplimiento de asistencia y labores docentes y seguimiento sobre los temas impartidos según el proyecto docente de cada asignatura, analizando la concordancia entre el contenido programático de cada una y lo diligenciado por el docente.
- El diligenciamiento del proyecto docente se hace de manera individual por cada profesor, lo que origina una variación en los contenidos impartidos para cada asignatura y no se guarda unanimidad. Además, la estructura utilizada para la consignación de los mismos es subjetiva a cada docente, disponiéndose de editores de texto u hojas de cálculo para su desarrollo. Estas herramientas son poco apropiadas al momento de mantener un record de los proyectos docente o de realizar consultas sistematizadas sobre los mismos.
- Falta de personal para ejecutar constantemente las funciones de verificación y control debido al exhaustivo trabajo que conlleva auditar la información en cuestión.
- Errores en la manipulación de la información por el cuantioso volumen de datos, que dificulta la búsqueda y manejo de registros históricos.
- Demora en el suministro de información oportuna para la toma de decisiones académicas y administrativas, por el bajo nivel de sistematización del proceso de gestión.

En consecuencia, al carecer de un sistema para el control de asistencia académica se originaban dificultades relacionadas con la forma en cómo los docentes y administrativos gestionaban el proceso de registro y control de asistencia y temáticas vistas en cada clase,

lo cual se debía, en gran medida, a la carencia de una herramienta tecnológica que sirva de apoyo para las actividades previamente expuestas. Es por eso que contar con una herramienta informática permitiría procesar, organizar, controlar, manejar y suministrar la información relativa al proceso de registro y control de asistencia académica de manera rápida y eficiente (López Escalera & Gaona Campos, 2016).

Se propuso entonces la construcción de una plataforma virtual, de característica multiplataforma (con componentes móvil y web), para llevar el proceso de registro y control de asistencia académica en la Universidad de Cartagena, como solución al problema planteado. Con esta herramienta, en comparación al mecanismo utilizado, se digitaliza la información y almacena de forma sistematizada los datos pertinentes (asistencia docente y estudiantil, y temas vistos en clase) para el proceso, con el objetivo de tener una mejor manipulación, gestión y toma de decisiones oportuna. Es importante tener en cuenta que al haber implementado un aplicativo de información, además de agregarle valor a la Universidad, como se indica en el acta de la entrevista realizada con la ingeniera Narlinda Espinosa (ver anexo C), es una herramienta esencial que facilita la toma de decisiones importantes.

### **1.2.2 Formulación del problema**

Ante la problemática expuesta previamente, surgió el siguiente interrogante: ¿Cómo mejorar el proceso de registro y control de asistencia académica de la Universidad de Cartagena?

No. \_\_\_\_\_ AÑO \_\_\_\_\_ AULA \_\_\_\_\_

FACULTAD DE \_\_\_\_\_

**REGISTRO DE ASISTENCIA  
EN EL AÑO 20 \_\_\_\_\_**

Asignatura: \_\_\_\_\_

\_\_\_\_\_

**CONFERENCIAS**

	Lunes	Martes	Miercoles	Jueves	Viernes
M					
P					

Profesor: Doctor \_\_\_\_\_

Jefe de Departamento \_\_\_\_\_

Cartagena de Indias

*Ilustración 1. Cartilla de registro de asistencia.*



deberes”. Dentro de las notas fundamentales de la constitución política, la educación es evidenciada como un derecho de la persona, un servicio de carácter público con una función social que corresponde al estado regular, ejercer inspección y vigilancia con el fin de velar por la calidad, la formación moral, intelectual y física de los educandos.

A nivel de la educación superior, los objetivos encaminados en el logro de una educación de calidad son cada vez notorios cuando se hace referencia a procesos que busquen aumentar las cualidades académicas de sus estudiantes, teniendo en cuenta que en este nivel se están formando profesionales competentes, investigadores y que aporten soluciones innovadoras en cada uno de sus campos.

Es así como la Universidad de Cartagena (2016), como institución de educación superior, en el cumplimiento de sus funciones sustantivas de docencia, investigación y extensión e internalización, forma profesionales competentes e íntegros en distintas áreas del conocimiento con actitud de emprendimiento y liderazgo, que propongan soluciones en aras de producir cambios a nivel local, regional y nacional. Para cumplir con esta misión, la universidad afronta sus objetivos desde el ambiente más familiar y básico para el estudiantado, el aula de clases, donde todo el conocimiento es impartido y donde se da forma al profesional pensante, investigador e innovador; por lo que se le hace crucial vislumbrar si el estudiantado es partícipe de las actividades académicas que se llevan a cabo dentro del aula. De igual manera, para la universidad es indispensable tener conocimiento de la participación activa de los docentes como promotores del saber dentro de las aulas de clases, con el fin de identificar si cumplen de manera oportuna con el ejercicio de sus funciones.

El control de asistencia del personal docente resulta una tarea indispensable que va más allá del pago de la nómina, involucra la profesionalización de esta noble tarea ya que, al cumplir cabalmente con sus obligaciones, la misma repercute de manera intangible en el nivel académico que se trasmite a los alumnos, derivado del cumplimiento en tiempo y forma de sus horarios de trabajo (Olivares Morales & Valeriano Assem, 2010).

Dentro del reglamento estudiantil de la Universidad de Cartagena (2009), en el título V, capítulo IV, de la asistencia a clases, se definió en su artículo 91 lo siguiente:

“Al matricularse académicamente, el estudiante adquiere el compromiso de asistir a clases. El registro de asistencia lo controlará, obligatoriamente, cada profesor, de conformidad con las listas suministradas por el Centro de Admisiones, Registro y Control Académico de la Universidad, a través de la plataforma virtual.”

De acuerdo a ello, las actividades realizadas en el aula de clases, tratando de alcanzar los objetivos mencionados con anterioridad, han sido soportadas por un proceso de registro docente para llevar el control de la asistencia, en donde se consignan datos pertinentes para el seguimiento que la universidad hace sobre los estudiantes, gracias al ejercicio de las funciones que el docente en su cargo ofrece. El proceso consta de dos partes; una donde un registro es diligenciado por el docente con los temas tratados durante su clase, fecha, firma e información de la asistencia estudiantil, y otro donde se gestiona el control realizado por jefes de departamento y/o auditores claramente definidos.

Al sistematizar un proceso crucial como este para la institución, uno de los principales aspectos a considerar es la reducción de tiempo y esfuerzo que conlleva buscar la información registrada tras las actividades académicas mencionadas anteriormente, y a su vez contribuir a la optimización de la manera como se lleva actualmente el proceso. Conocer si los estudiantes asisten a sus clases, identificar si los temas correspondientes a cada área del conocimiento se están dando por vistos y son acordes al proyecto docente y garantizar la calidad de la educación, son los objetivos a suplir como parte de una solución significativa.

Con la puesta en marcha de la solución, a nivel tecnológico se disminuyó el manejo desmesurado de grandes volúmenes de información física. Además, al hacer uso de recursos digitales, se sustituyó el almacenamiento de los registros de forma física a virtual, facilitando así aspectos como filtros de búsqueda por categorías como periodo académico, asignatura, docente, fecha; lo que otorga flexibilidad sobre su manejo y gestión, al igual

que facilidad de búsquedas históricas y procesos de auditoría desempeñados por la institución.

A nivel gerencial, las ventajas tecnológicas corresponden a que los procesos de auditoría en la universidad requieren de análisis estadísticos concernientes a la asistencia docente, estudiantil y a la relación de los temas impartidos en clases con lo estipulado en el proyecto docente de las signaturas de cada programa, por lo que el uso del módulo estadístico a nivel de auditoría facilitó las decisiones a nivel administrativo y es de gran utilidad para el proceso.

Por otra parte, con la implementación de esta solución se evidenció la reducción de costos asociados al uso de elementos como papel o útiles de papelería, que ya no se hacen necesarios en gran medida, otorgando de manera implícita un carácter ecológico relevante.

El presente proyecto fue viable económica y tecnológicamente debido a que fue una propuesta cuya solución se implementó en primera medida en la Universidad de Cartagena, al ser ésta la principal promotora del desarrollo. Por esta razón, para el desarrollo de la solución software se tuvieron a disposición los recursos económicos y a nivel de infraestructura y tecnología que la Universidad proporciona, por ser un proyecto desarrollado dentro del grupo de investigación E-Soluciones, en su línea de investigación E-Servicios. En cuanto al recurso humano necesitado para el desarrollo de la misma, se contó con la participación del personal necesario y cuyos papeles se vieron reflejados como los investigadores del proyecto y el director.

Finalmente, es así como el desarrollo de un proyecto como este agrega valor a la institución, permitiéndole estar a la altura de los tiempos en aspectos como la administración de procesos académicos, además que la posiciona como una entidad a la vanguardia en ofrecer una herramienta tecnológica como medio para el aseguramiento y fortalecimiento de la calidad académica. También facilita la ejecución de forma ágil de los procesos académicos que relacionados con el manejo de registros de asistencia y los controles de auditoría. Como se ha manifestado, es una solución trascendental que resuelve el problema expuesto y agrega valor y calidad a la institución.

## **1.4 OBJETIVOS**

### **1.4.1 Objetivo General**

Desarrollar una plataforma virtual para el registro y control de asistencia académica en la Universidad de Cartagena, haciendo uso de la metodología de desarrollo de software RUP.

### **1.4.2 Objetivos Específicos**

- Definir el modelo de negocio junto con la especificación de requisitos funcionales y no funcionales, aplicando en cada uno de los casos las técnicas de recolección de información que proporcionen datos de valor para el análisis.
- Construir un modelo de diseño apropiado a partir de la interpretación de los artefactos software evidenciados durante el modelo de negocio, teniendo en cuenta los factores de calidad pertinentes durante el proceso de desarrollo de software.
- Implementar la plataforma que atiende al problema, haciendo correcto uso de los patrones de diseño y arquitectura definidos en el modelo de diseño y con un oportuno proceso de verificación que, aplicando teorías asociadas a la experiencia de usuario, pueda brindar ambientes de trabajo agradables y confiables.
- Probar la plataforma a través de mecanismos de testeo y aceptación que validen la correlación entre lo especificado por el usuario y el producto entregado.

## 1.5 *ALCANCE*

El presente proyecto se encontró enmarcado por el desarrollo de una plataforma virtual para llevar a cabo el registro y control de asistencia académica en la Universidad de Cartagena, la cual se desarrolló y ejecutó dentro de un tiempo de 8 meses. El producto software entregado puede ser utilizado por docentes, estudiantes y administrativos. Además, cuenta con dos componentes fundamentales los cuales son una Aplicación Móvil y una Plataforma Web que permiten, a los diferentes usuarios, ejecutar diversas funcionalidades de acuerdo a los privilegios que sus roles tengan. Estas funcionalidades se soportan en mecanismos de control definidos sobre los registros realizados alrededor de la asistencia, los contenidos del proyecto docente y la auditoría. Las principales funcionalidades se definen a continuación:

- El docente, desde cualquier lugar, puede diligenciar el formato de registro de asistencia teniendo en cuenta que se configuró un tiempo límite para ello (equivalente a 48 horas). Además, se permite la notificación de novedades en caso dado el desarrollo de la clase no se dé de manera adecuada a través de un espacio de observaciones, posibilitando al docente alertar el motivo de tal evento.
- Los administrativos auditores, tales como jefes de departamento académico y jefes de departamento interno, tienen la potestad de hacer el control de la información registrada en la plataforma: registros de asistencia y temáticas vistas, además de realizar consultas pertinentes que permitan hacer seguimiento sobre los contenidos propuestos en los proyectos docente y los temas registrados en la plataforma, para identificar posibles irregularidades.
- Existe la figura de administrador de la plataforma, el cual se encarga de la alimentación, gestión y manejo de los datos de la misma, tales como: docentes y estudiantes del programa, contenidos del proyecto docente, sincronización de datos con los sistemas de información de la Universidad, ente otros.
- Cualquier estudiante tiene la opción de consultar los registros académicos diligenciados por el docente del curso en el que está matriculado, además de poder realizar observaciones con respecto a la información en cuestión, las cuales son

confidenciales. Únicamente aquellos usuarios con funciones de auditor general pueden revisar las observaciones de los estudiantes.

- El sistema le permite ver a los estudiantes los registros académicos una vez estos sean diligenciados, con el fin de que tengan acceso al proceso de consulta descrito anteriormente. Los estudiantes también pueden observar su porcentaje de inasistencia, y los auditores tienen acceso a esta información para identificar los casos en que se pueden aplicar los lineamientos estipulados en el reglamento estudiantil de la Universidad para la inasistencia.
- La plataforma cuenta con mecanismos de control para asegurar la asistencia de los estudiantes, del cuerpo docente y el desarrollo de las temáticas. Este control es factible a partir de un módulo habilitado para los auditores donde se pueden visualizar estadísticas que proporcionan información como la cantidad de registros diligenciados, el porcentaje de avance en cuanto a temas del proyecto docente y el porcentaje de asistencia estudiantil por clase.

Para la construcción de la plataforma se hizo uso del lenguaje de programación JavaScript (a través de la tecnología Node.js con el uso del framework Express) para implementar las funcionalidades lógicas del software, en acompañamiento de MongoDB, un sistema de base de datos NoSQL orientado a documentos. Para el desarrollo de la interfaz gráfica de usuario se dispuso del lenguaje de etiquetado HTML5 y del lenguaje de hojas de estilo CSS3, en acompañamiento del lenguaje de programación interpretado JavaScript, haciendo uso del framework Ionic (basado en Angular) en su versión 4.

La utilidad del producto software entregado es la de sistematizar el proceso de registro y control de asistencia académica de la Universidad de Cartagena, sirviendo como una herramienta tecnológica de apoyo que permite reducir el tiempo y esfuerzo de los docentes en la actividad del registro de asistencia. El software también permite el registro del proyecto docente por asignatura, capturando la información a través de los campos de un formulario con el fin de integrarlo al sistema como aspecto fundamental para agilizar la consignación de los temas tratados en clase y eliminando así las incoherencias que se pueden presentar cuando se consignan de forma manual.

Otra ventaja en la utilización del producto software es la supervisión y control de dichos registros, buscando atender a una problemática que vincula una gran cantidad de docentes de la Universidad y donde su manejo es bastante dispendioso. Además, un valor agregado a esta solución radica en su impacto ecológico, debido a que el manejo de los volúmenes de información física, son mitigados al deshacerse de útiles de papelería y remplazados por recursos digitales.

La conectividad con las bases de datos de la Universidad se realizó para la obtención de los datos necesarios, tales como la matrícula académica, el plan curricular por asignatura, horarios de clase e información pertinente relacionada con los estudiantes, docentes y administrativos de cada programa de la Universidad. El software tiene la capacidad de trabajar en modo de compatibilidad con el sistema de información académico-administrativo de la Universidad, garantizando el cumplimiento de los objetivos planteados en este proyecto. Los aspectos tecnológicos relacionados con la forma en cómo son consumidos los datos, se definieron detalladamente en el diseño del producto software.

Es de suma importancia aclarar que el presente proyecto no contempló la posibilidad de modificar radicalmente la manera en cómo se llevaba a cabo el proceso de registro y control de asistencia, sólo buscó garantizar un mejoramiento incremental de los recursos utilizados (tecnológicos en este caso) para tal fin. De la misma forma, con la realización del proyecto no se pretendió abarcar la gestión de procesos externos como vinculación docente, pago de nómina, inclusión o retiro de asignaturas, entre otros aspectos relacionados, ni está en la capacidad de tomar decisiones de tipo administrativas o académicas.

## MARCO DE REFERENCIA

---

### 2.1 *ESTADO DEL ARTE*

“A través del tiempo, la tecnología ha evolucionado de una manera asombrosa, permitiendo automatizar procesos para un mejor control. Los procesos rutinarios que son manejados en forma manual se resuelven de una manera lenta, es por ello que al llevarlos a un sistema automatizado garantiza un mejor trato de la información. En gran parte la tecnología de la información, reúne, almacena y procesa, es por ello que se desea disminuir tiempo en la ejecución de las tareas realizadas mediante procesos tediosos para el ser humano” (Sandoval Illescas, Sigüenza Cárdenas, & Ingavélez G, 2011).

El control de asistencia (también llamado de presencia) es importante para una organización, bien sea empresa, universidad, instituto, entidad, etc., generalmente por aspectos de seguridad, con el objetivo de identificar quien entra y sale. Es así que hoy día los sistemas de control de asistencia son imprescindibles en organizaciones, ya que no sólo funcionan como sistema de acceso seguro sino como un método para la gestión del tiempo, de las actividades y de las labores ejecutadas en el entorno de trabajo, tal como dicen Pantoja Blyde, Lozano Leal y Portillo Montiel (2013): “Los sistemas de control de asistencia son imprescindibles en toda institución porque permiten monitorear el cumplimiento de la jornada laboral de los empleados”.

Históricamente, los controles de asistencia han ido evolucionando junto con los avances tecnológicos que han surgido. En un principio las empresas utilizaban una hoja de control de asistencia en la que cada empleado proporcionaba información como la hora de entrada y de salida, acompañada de su firma como constancia. Posteriormente, los registros se trasladaron a una ficha de cartón que el trabajador marcaba, introduciéndola en una máquina a la entrada y a la salida. Ambos sistemas de control de asistencia laboral

necesitaban hacer los cálculos de forma manual después, lo que ralentizaba el proceso (Varela, 2015).

Así mismo Varela (2015) dice lo siguiente:

“Gracias a la tecnología, los sistemas de acceso y de control de asistencia se han ido modernizado y digitalizando. Ahora los controles de acceso permiten llevar los registros de asistencia desde un ordenador, donde se guardan y organizan los datos de cada trabajador. De esta forma, se pueden hacer seguimientos en tiempo real, además de conocer casi al instante las horas trabajadas, los días libres que quedan, las horas extra que se acumulan, pero no sólo a la empresa, sino también al trabajador, mejorando así en la transparencia empresarial y, por tanto, el clima laboral”.

En los últimos años las organizaciones han invertido un notorio esfuerzo en identificar la forma más adecuada de implementar procesos para el control de asistencia del personal de trabajo a sus instalaciones, indistintamente del papel que desempeñan dentro de la misma. Así, se han planteado soluciones significativas a la situación en cuestión. Con la biometría, por ejemplo, se generan sistemas de identificación y autenticación de cada individuo para gestionar accesos. Esto unido, a un reloj biométrico, permite ampliar su funcionalidad hacia el control de presencia y la gestión del tiempo en una empresa. Además, la irrupción de Internet y los nuevos dispositivos electrónicos han permitido la creación de sistemas de control de presencia multiplataforma (Varela, 2015).

Tal es el caso de un Sistema de Posicionamiento en Interiores (IPS, por sus siglas en inglés) desarrollado en la Universidad de Aalborg, Dinamarca en el año 2012. Un IPS básicamente consiste en una red de varios dispositivos que permiten localizar inalámbricamente la posición de objetos o personas dentro de un edificio. La solución propuesta por la institución se basa en un IPS haciendo uso de dispositivos móviles y con infraestructura Bluetooth instalada típicamente como parte completa del hardware / software, manejando un bajo consumo de energía en comparación con los dispositivos Wi-Fi, lo que promueve una mayor duración de la batería en los teléfonos. El sistema presentado es un IPS Bluetooth basado totalmente en productos de grado básico. La precisión de

posicionamiento se evalúa utilizando la denominada técnica de huellas digitales de localización (Frost, Svenning Jensen, S e Luckow, Thomsen, & Hansen, 2012).

Por otro lado, en Maracaibo, Venezuela, para Febrero de 2014 se implement  un Sistema biom trico de control de asistencia laboral mediante el uso de huella dactilar, como trabajo de grado de la Universidad Rafael Urdaneta en la Escuela de Ingenier a, cuyo objetivo consiste en llevar un registro sistem tico acerca de las entradas y salidas, faltas, permisos, entre otros, en un entorno laboral (P rez, El Safadi, & Araujo, 2014).

Igualmente se han propuesto desarrollos empresariales que est n orientados a la recolecci n y sistematizaci n de datos para la entrada y salida de los empleados haciendo uso de plataformas, como es el caso de la Corporaci n Universitaria Minuto de Dios en Soacha, Cundinamarca, Colombia, para el a o de 2010. “Este proyecto, el cual se titula prototipo del control de los empleados de PRODICAUCHOS LTDA, naci  a partir de la necesidad como su mismo nombre lo indica de registrar el movimiento de los empleados que residen como trabajadores al interior de la micro empresa PRODICAUCHOS LITDA...por lo que se estructur  el dise o de una p gina web que permite registrar la cantidad de empleados de la empresa, los productos, y en general una base de datos de toda la comunidad que permite el control y organizaci n de la empresa” (Reyes Puerto & Bermudez Amaya, 2010).

As  mismo, existen compa as que ya ofrecen productos software para gestionar este tipo de procesos, como es el caso de:

- SquareNet Software Solutions, en Ecuador, que ofrecen su software SquareNet gesti n Humana, “un potente software que se compone principalmente de N mina, **Control de Asistencias** y Recursos Humanos m s varios m dulos auxiliares como Control de Visitas, Control de Comedor y Control de Accesos, los cuales le brindar n completo control y manejo sobre la **asistencia**, contratos, acciones, accesos, comidas, visitas y roles de pago de sus empleados de una manera f cil, r pida y eficaz” (SquareNet Software Solutions, s.f.).
- Fusionar Soluciones Tecnol gicas, en Uruguay, que ofrecen una gama de productos software (FUSIONAR, Soluciones Tecnol gicas, s.f.) que permiten:

- Realizar de manera segura, práctica y eficiente el control en tiempo real de todos los ingresos y egresos de personas, máquinas y vehículos que ocurren diariamente (FSAcceso: Control de Acceso y Documental).
- Controlar todo lo relacionado al horario del personal de una empresa o personal tercerizado. Permite definir qué personas se van a controlar, cómo se identifican, a qué empresa pertenecen, etc. (FSAsistencia: Control de Asistencia).
- Softtronics Solutions S.A., en Cali, Colombia, “una empresa pionera en el desarrollo de soluciones que involucran tecnologías de identificación (biometría y RFID) aplicadas a diferentes procesos como lo son la planeación, control y liquidación de tiempos laborados por empleados, o el control de acceso de personas en áreas restringidas” (Softtronics Solutions S.A., s.f.). Dentro de los productos que ofrecen se encuentra BioPoint, “un sistema para capturar las marcaciones de inicio y fin de jornada en un punto de venta, una pequeña oficina o en general, las instalaciones de una empresa donde labore una cantidad de empleados limitada (menor a 50), utilizando un lector de huellas digitales en el instante en que el empleado inicia o termina la jornada laboral. El sistema guarda la fecha, hora e identidad de la persona que realiza la marcación, de una manera ágil, amigable y virtualmente inmune a fraudes” (Softtronics Solutions S.A., s.f.)

Cabe resaltar también que instituciones de carácter educativo a nivel deportivo han dado paso a la implementación de sistemas de este tipo, tal es el caso de la escuela de capoeira Nativos, ubicada en Soacha, Colombia, quienes hacen uso un software llamado STEEL, que nace de un proyecto de grado en la Corporación Universitaria Minuto de Dios el 21 de junio de 2012. El proyecto, que lleva por título ANÁLISIS Y DISEÑO DE SOFTWARE PARA EL CONTROL DE ASISTENCIA DE ESTUDIANTES, “STEEL”, consiste en una “aplicación web que permite realizar registro de datos de estudiantes, registro de asistencia y pagos ya sea por mensualidad, permitiendo el manejo de la información de una manera óptima, mejorando los procesos administrativos y ayudando a la toma de decisiones” (Alonso Bernate & Pinzón Caballero, 2012).

Las instituciones educativas como las universidades no están exentas, y en ellas se han desarrollado proyectos que permiten la sistematización del control de asistencias. Particularmente para las universidades, la asistencia tanto del cuerpo docente como del estudiantado es importante porque ambos actores son cruciales dentro de su objeto social, cuyo fin es brindar educación de calidad. Por ejemplo, la Universidad de las Fuerzas Armadas ESPE, en Ecuador, desarrolló para 2014 una aplicación web de control de asistencias para programas de postgrado, mediante la utilización de la plataforma Java Enterprise Edition Jee6 Web aplicando la metodología Scrum. Gracias a la aplicación se automatizan los “procesos de asignación de horarios de clases y el control de asistencia del profesor y alumno que se llevan a cabo en la Unidad de Gestión de Postgrados de la universidad” (Suntaxi Suntaxi, Moromenacho Caiza, Arroyo, & Parra, 2014).

También, en la Universidad Nacional Autónoma de México se llevó a cabo un proyecto de grado en marzo de 2010 para la automatización del proceso de control de asistencia del personal académico en tiempo real a través de reconocimiento biométrico. El proyecto mejoró de manera notable la rapidez con la que se realizaban anteriormente las actividades docentes, además de que la precisión de la información se garantiza por tratarse de un sistema en la que la identificación del personal es totalmente automática. Se adquirió un lector de huellas digitales económico y con una tasa muy alta de fiabilidad que permite el registro de los profesores en sus horarios de trabajo, con ello se tendrá un control exacto de las asistencias y por supuesto permite tomar decisiones de reemplazo o cobertura de algún profesor que por alguna razón falte a dar su clase (Olivares Morales & Valeriano Assem, 2010).

Así mismo, en España el “Banco Santander a través de su División Global Santander Universidades, y Vodafone España puso en marcha, para el año 2012, un proyecto piloto que permitía el control de asistencia y de acceso a las instalaciones de la Universidad Católica San Antonio de Murcia (UCAM) mediante la tecnología sin contacto NFC instalada en la tarjeta SIM de los dispositivos de los estudiantes” (ABC, 2012).

Otro referente internacional que se tiene es en Maracaibo, Zulia, Venezuela, en donde se desarrolló un proyecto para la automatización del control de asistencia del personal docente

del departamento de computación de la facultad experimental de ciencias de la Universidad del Zulia. “El objetivo de este proyecto fue automatizar el control de asistencia del personal docente del departamento de computación de la FEC (Facultad de Ciencias Experimentales)...Los resultados obtenidos facilitan el registro de las horas laborales de los docentes, ya que la automatización que produjo la misma permite llevar de manera confiable el registro de las horas activas de los profesores en la FEC” (Pantoja Blyde, Lozano Leal, & Portillo Montiel, 2013).

A nivel nacional, en la Universidad de Pamplona, Pamplona, Norte de Santander, Colombia, se desarrolló un prototipo de control de acceso a aulas y registro automático de asistencia:

El proyecto consiste en el diseño e implementación de un prototipo como prueba piloto, para el control de acceso a aulas y laboratorios, y registro automático de asistencia a clases, para reducir el tiempo que estas tareas conllevan... Como resultado cuando un docente desea ingresar a un salón de clases, sala de informática, laboratorio u otro recurso físico del instituto para desarrollar su clase, debe presentar un identificador ante un lector ubicado junto a una puerta de acceso. Luego se verifica si el docente tiene asignado el recurso físico en el horario y fecha del momento, donde luego aparecerá Acceso Denegado o Acceso Autorizado, según sea el caso. Los estudiantes, por su parte, al ingresar también deben identificarse y el sistema de control de asistencia coloca automáticamente las fallas o asistencias de cada uno, teniendo en cuenta el horario establecido. Con estos se posibilita la generación de reportes reales los cuales serían empleados por el área encargada para el seguimiento a los estudiantes y así reducir el nivel de ausentismo en la institución (Márquez C, Abdo L, & Angulo, 2014).

Al abordar un referente dentro del marco local, en la Ciudad de Cartagena de Indias, Colombia, en la Fundación Universitaria Tecnológico Comfenalco, tal como se indica en el acta de la entrevista realizada con el estudiante de ingeniería y desarrollador Carlos Mario Ramos (ver anexo E), se manejan varios tipos de sistemas para controlar los procesos de asistencia. En primera medida, los docentes hacen uso de la plataforma virtual AMSYS

para descargar las planillas en formato Excel con la relación de todos los estudiantes inscritos en una asignatura en particular, para luego utilizarlos como soporte y tomar asistencia de sus estudiantes, así como el registro de los temas dados en una jornada de clases para una materia en particular. Por otra parte, se maneja otro proceso sistematizado para controlar la asistencia del cuerpo docente a los laboratorios de sistema, a través de una plataforma desarrollada por la misma institución que se conecta con el sistema de persistencia en la nube de Google (Google Drive). De la misma manera, a nivel de toda la universidad se utilizan unos torniquetes electrónicos que registran el ingreso y la salida de todo el personal que hace parte de la institución gracias al reconocimiento digital de los carnets asociados a cada uno.

“Las áreas de tecnologías de información son claves para el fortalecimiento de nuevos cambios que apunten a esquemas de organización del trabajo de forma no solo eficientes sino también menos costosas en la utilización de tiempo y recursos, para ello se requiere el desarrollo de aplicaciones informáticas dirigidas a absorber tanto el conocimiento como los procesos, mediante la creación de soluciones integradas, orientadas a facilitar tareas además de consolidar gestión” (Pantoja Blyde, Lozano Leal, & Portillo Montiel, 2013).

De esta manera se observa que las herramientas sistematizadas y/o automatizadas para controlar la asistencia de personal como trabajadores, administrativos, docentes o estudiantes, bien sea a nivel empresarial o educativo, han venido tomando más fuerza en un mundo donde la tecnología comienza a tener un auge tan trascendental que se involucra de forma pertinente y relevante en los procesos de valor para las organizaciones o entidades. Es por ello que las instituciones que aún optan por llevar a cabo dichos procesos de manera rudimentaria deben proceder y actuar en la procura de estar a la vanguardia de los avances tecnológicos que hoy por hoy optimizan, mejoran y facilitan en gran medida las actividades del día a día.

## **2.2 MARCO TEÓRICO**

Con la llegada de las tecnologías al campo educativo se ha producido un cambio importante en los procesos que son realizados en las instituciones de toda índole. Desde instituciones de básica primaria hasta instituciones de educación superior, hacen uso de las ventajas que los avances tecnológicos pueden ofrecer para desarrollar y fortalecer sus servicios como institución en aras de mejorar y ser competitivos. “El uso de las tecnologías de la información y las comunicaciones, además de ser una ventaja para las organizaciones, en la actualidad representa una necesidad constante para superar fronteras” (Marrugo Marrugo, Nuñez Barcos, & Martelo Gómez, 2012).

Para el desarrollo de esta sección, se contextualizan los conceptos que soportaron el diseño y la implementación de la propuesta de investigación, así:

### **2.2.1 Registro de asistencias**

Aunque el término “registro” cuenta con diferentes acepciones que varían según el contexto en que es utilizado, para el caso propuesto en el presente proyecto es concebido como un conjunto de información asentada para que terceros posteriormente lo utilicen con un determinado fin.

Según la International Organization for Standardization (2015) un registro se define como un “documento que presenta resultados obtenidos o proporciona evidencia de actividades realizadas”.

De igual forma, la Universidad Nacional de Colombia (2016), en su plan de mejoramiento para la elaboración y control de documentos del sistema integrado de gestión, especifica que un registro es un “documento que presenta resultados obtenidos o proporciona evidencia objetiva de actividades desempeñadas”.

Por otra parte, la Universidad de Jaén (2012) considera que “los registros son un tipo especial de documento que proporcionan evidencia de la conformidad con los requisitos especificados”, es decir, que guardan concordancia con ciertos objetivos particulares para reflejar seguimientos encaminados a cumplirlos.

Así, tomando nuevamente como referente a esta institución, para garantizar una correcta manipulación de los mismos se tienen en cuenta los siguientes criterios generales y definiciones al establecer un procedimiento:

- **Identificación:** Los registros deben ser fácilmente identificables.
- **Almacenamiento:** Es necesario determinar cuál es la forma de almacenamiento y dónde se realiza el archivo de los registros para poder después encontrarlos fácilmente.
- **Protección:** Lugar adecuado para la conservación de registros. Es necesario determinar los niveles de protección de los registros para así evitar cambios en la información que contienen, por ejemplo, protección con contraseña o existencia de archivos con acceso restringido. Un aspecto fundamental es la realización de copias de seguridad de los registros digitales de la organización.
- **Recuperación:** Es necesario determinar la metodología para acceder y encontrar registros de actividades anteriores.

Todas las aserciones realizadas con anterioridad tienen cabida dentro del marco conceptual que se está trabajando, contextualizando así que un registro de asistencia es entendido como un documento donde se consiga información relacionada con la verificación de la asistencia de un grupo de personas, en este caso estudiantes, y otros factores como las temáticas trabajadas particularmente para cada asignatura por los docentes dentro del aula de clases, como parte de un proceso académico propio de una institución educativa.

### 2.2.2 Recurso digital

Tomando como referente a Lorduy Salas, Peña Esquivel, & Puello Marrugo (2014) se puede considerar un recurso digital como cualquier tipo de información que se encuentra

almacenada en formato digital y que facilitan el almacenamiento, la organización y la recuperación de enormes cantidades de datos.

### **2.2.3 Plataforma virtual**

Según Ezkauriatza Milagros (2011) “La tecnología ha facilitado la creación de herramientas y ambientes que permiten enriquecer las experiencias de aprendizaje. Los estudiantes del siglo XXI esperan y necesitan una educación que proporcione un aprendizaje significativo, con actividades colaborativas que les permitan interactuar e intercambiar ideas dentro y fuera del mundo virtual de la Web”. Uno de los entornos virtuales manejados para tal fin son las plataformas virtuales, definidos como programas software orientados al internet.

“La utilización de las plataformas virtuales hoy en día es inherente al quehacer docente” (Valenzuela Zambrano & Pérez Villalobos, 2013). Las plataformas virtuales representan gran importancia para la educación, ofreciendo grandes beneficios y conllevando a una mejor administración de la información, a partir de las repercusiones positivas que las tecnologías de la comunicación han generado. En definitiva, se puede determinar que una plataforma virtual es un sistema que puede ejecutar diversas aplicaciones en un mismo entorno, dando a los usuarios la posibilidad de acceder a ellas a través de internet.

Así mismo, Ysabel Zavahra Tahanian (2012) clasifica en tres categorías los diferentes tipos de plataformas que se puede encontrar, así:

- Comerciales: Debe efectuarse un pago previo para hacer uso completo de sus funcionalidades sin derecho a adquirir el código fuente o realizar modificaciones sobre la misma.
- Software Libre: Se puede adquirir sin costo alguno y se tiene acceso al código fuente para hacer modificaciones o mejores, pero con la observación que dicha modificación debe ser compartida con la comunidad y estar disponible para cualquier usuario.

- Software propio o a la medida: Son desarrolladas e implementadas a nivel educativo dentro de una institución, no buscan beneficio económico, no están dirigidas a su comercialización ni distribución masiva, solo responden a factores educativos y/o pedagógicos.

#### 2.2.4 RUP

Dentro de la ingeniería de software se encuentran tres rubros importantes a ser atendidos con antelación por todo ingeniero de software para llevar a cabo el proceso en mención, estos son: la descripción de un problema, el planteamiento de una solución y la definición del proceso para gestionar el desarrollo del software, garantizando así calidad.

El proceso en cuestión se enmarca dentro de algo conocido como Metodologías de Desarrollo de Software, que orientan la forma en la que la ingeniería de software dará solución al problema, describe artefactos de análisis, determina los requisitos, propone los modelos de diseño en la solución, y otros aspectos importantes.

RUP (Rational Unified Process), es una de esas metodologías la cual “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 sus usuarios finales con unos costos y calendario predecibles” (Martínez & Martínez, 2014). Por ello, para el cumplimiento de sus objetivos, estos autores plantean (entre otras) que las principales características que ofrece RUP son:

- **Guiado por casos de uso:** donde se describe que en RUP, un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.
- **Centrado en la arquitectura:** Esta representa la forma en que están organizadas los componentes de nuestra solución software y describe la interacción entre ellos. Es

como una radiografía del sistema que se está desarrollando, lo suficientemente completa como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo, pero lo suficientemente simple como para que si se quita algo una parte importante del sistema quede sin especificar. Se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de lo demás.

- **Iterativo e incremental:** Referencia a los diferentes ciclos de vida del software determinando que, para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Otro de los aspectos importantes de esta metodología es que está conformada por 4 fases fundamentales que permiten el desarrollo de la misma, dando forma y puesta en marcha del proyecto que se pretenda ejecutar. Las fases y sus características según Martínez y Martínez (2014) son:

- Inicio: Antes de iniciar un proyecto es conveniente plantearse algunas cuestiones: ¿Cuál es el objetivo? ¿Es factible? ¿Se construye o se compra? ¿Cuánto va a costar? La fase de inicio trata de responder a estas preguntas y a otras más. 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 se va a continuar o a dejar. Los artefactos producidos en esta fase corresponden a:
  - Visión del negocio, que describe los objetivos y restricciones a alto nivel.
  - Modelo de casos de uso.
  - Especificación adicional: requisitos no funcionales.
  - Glosario: Terminología clave del dominio.
  - Lista de riesgos y planes de contingencia.

- El caso de negocio (business case). Para más detalles ver el flujo de modelado del negocio.
- Prototipos exploratorios para probar conceptos o la arquitectura candidata.
- Plan de iteración para la primera iteración de la fase de elaboración.
- Plan de fases.
- **Elaboración:** Analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del 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. Este prototipo debe contener los casos de uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves, bien con este prototipo, bien con otros de usar y tirar. Algunas de las actividades que corresponden a esta fase están definidas así:
  - Definir, validar y cimentar la arquitectura.
  - Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones.
  - Requisitos adicionales.
  - Descripción de la arquitectura software.
  - Un prototipo ejecutable de la arquitectura.
  - Lista de riesgos y caso de negocio revisados.
  - Posiblemente un manual de usuario preliminar.
- **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 todas los componentes, características y requisitos, que no hayan sido hechos hasta ahora, han de ser implementados, integrados y testeados, obteniéndose una versión del producto que se pueda poner en manos de los usuarios. Los artefactos que deben ser entregados en esta correspondiente fase son:
  - Modelos completos (casos de uso, análisis, diseño, despliegue e implementación)
  - Arquitectura íntegra (mantenida y mínimamente actualizada)
  - Riesgos presentados mitigados
  - Plan del proyecto para la fase de transición

- Manual inicial de usuario (con suficiente detalle)
- Prototipo operacional – beta
- Caso del negocio actualizado
- Transición: La finalidad de la fase de transición 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. Los artefactos que deben ser entregados en esta correspondiente fase son:
  - Prototipo operacional.
  - Documentos legales.
  - Línea de base del producto completa y corregida que incluye todos los modelos del sistema.
  - Descripción de la arquitectura completa y corregida.
  - Manual de usuario
  - Manual del sistema

### 2.2.5 UML

UML (Unified Model Language), es un lenguaje unificado de modelado, estandarizado y aceptado por toda la comunidad de la POO (Programación Orientada a Objetos). Pressman (2010) en su libro define UML como “un lenguaje estándar para escribir diseños de software. El UML puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo”.

Otra de las definiciones es facilitada por la Object Management Group (2005), empresa creadora, quienes determinan que UML “ayuda a especificar, visualizar y documentar esquemas de sistemas de software, incluyendo su estructura y diseño, de manera que cumpla con todos estos requisitos.”, además se hace aclaración de las distintas ventajas que ofrece este lenguaje de modelado, tal como que “UML 2.0 define trece tipos de diagramas,

divididos en tres categorías: Seis tipos de diagramas representan la estructura de aplicación estática, tres representan tipos generales de comportamiento; y cuatro representan diferentes aspectos de las interacciones”. A nivel más detallado, las interacciones presentes según la Object Management Group están dadas por:

- **Diagramas de estructura**, que incluyen el diagrama de clases, diagrama de objetos, diagrama de componentes, diagrama de estructura compuesta, diagrama de paquetes, y diagrama de despliegue.
- **Diagramas de comportamiento**, que incluyen el Diagrama de casos de uso (utilizado por algunas metodologías durante la reunión de los requisitos); Diagrama de Actividades, y diagrama de la máquina del Estado.
- **Diagramas de interacción**, que son todos los derivados del Diagrama de Comportamiento más general, incluyen el diagrama de secuencia, diagrama de comunicación, Cronograma y Esquema Descripción de la Interacción.

### 2.2.6 HTML5, JavaScript y CSS3

Propiamente dicho, “HTML es el lenguaje de marcado estándar utilizado para crear páginas web y sus elementos forman los bloques de construcción de todos los sitios web” según la World Wide Web Consortium: W3C (s.f.), una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

Por otra parte Mozilla Developer Network: MDN (2017), entorno que facilita información acerca de tecnologías web abiertas, como HTML, CSS y API para sitios web y aplicaciones HTML5, pone de manifiesto que este último representa dos conceptos: “Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos; y que contiene un conjunto más amplio de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance. A este conjunto se le llama HTML5 y amigos, a menudo reducido a HTML5”.

En cuanto a JavaScript (a veces abreviado como JS), éste es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script (archivo de órdenes) para páginas web dinámicas y para mejorar interfaces de usuarios, pero también usado en muchos entornos sin navegador (del lado del servidor, Server-side JavaScript o SSJS), (Mozilla Developer Network: MDN, 2016). Está soportado sobre el estándar ECMAScript.

Finalmente las hojas de estilo en cascada (o CSS, del inglés Cascading Stylesheets) es un lenguaje que define y crea la presentación de un documento escrito en un lenguaje de etiquetado (Mozilla Developer Network: MDN, 2017). Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML. CSS3 es la última versión de CSS y, como lenguaje de diseño gráfico, proporciona a sitios web la creación de páginas visualmente atractivas, interfaces de usuario para aplicaciones web, y GUIs para muchas aplicaciones móviles.

### **2.2.7 NoSQL Database**

Las bases de datos no SQL se encargan de la gestión de bases de datos difiriendo de los modelos relacionales tradicionales. Uno de los aspectos principales de este tipo de modelos es que no utilizan como lenguaje base SQL, esto debido a que los datos no son almacenados en estructuras tipo tabla, si no estructuras definidas como documentos, tiendas de gráficos, entre otros (MongoDB, s.f.).

Algunas de las principales razones que justifican el nacimiento de este tipo de bases de datos distintas de las tradicionales son:

- NoSQL abarca una gran variedad de tecnologías de bases de datos, esto debido a las demandas presentadas en la construcción de aplicaciones modernas.
- Los desarrolladores construyen aplicaciones que permiten la creación de grandes volúmenes de nuevos tipos de datos que además cambian rápidamente, donde estos

tipos de datos pueden ser estructurados, semiestructurados, no estructurados y polimórficos.

- Las nuevas metodologías de desarrollo ágiles que enmascaran grandes cambios a velocidades vertiginosas.
- Las aplicaciones actuales, son entregadas en calidad de servicios que siempre deben estar activos y accesibles desde muchos dispositivos diferentes.
- Los servidores monolíticos están en desuso debido a que las organizaciones recurren a arquitecturas escalables con servidores de productos básicos y computación en la nube.

### *2.2.7.1 Tipos de Bases de Datos NoSQL*

Existen diferentes tipos de bases de datos NoSQL tales como:

- Las bases de datos de documentos emparejan cada clave con una estructura de datos compleja conocida como documento. Los documentos pueden contener muchos pares clave-valor diferentes, pares de pares de claves o incluso documentos anidados.
- Las tiendas de gráficos se utilizan para almacenar información sobre redes de datos, como las conexiones sociales. Las tiendas de gráficos incluyen Neo4J y Giraph.
- Las tiendas de valores clave son las bases de datos NoSQL más simples. Cada elemento de la base de datos se almacena como un nombre de atributo (o 'clave'), junto con su valor. Ejemplos de tiendas clave-valor son Riak y Berkeley DB. Algunas tiendas de valores clave, como Redis, permiten que cada valor tenga un tipo, como 'entero', que agrega funcionalidad.
- Los almacenes de gran columna, como Cassandra y HBase, están optimizados para consultas sobre grandes conjuntos de datos y almacenan columnas de datos juntas, en lugar de filas.

Para la implementación de la solución planteada se hará uso de una base de datos por documentos que contienen pares clave-valor y que maneja una estructura JSON.

Las razones por las que se eligió una base de datos NoSQL para el desarrollo del presente proyecto radican en que, en comparación con las bases de datos relacionales, las bases de datos NoSQL son más escalables, ofrecen un rendimiento superior, permiten un mejor manejo para concurrencia de usuarios y su modelo de datos resuelve varios problemas que el modelo relacional no está diseñado para abordar (GYŐRÖDI, GYŐRÖDI, PECHERLE, & OLAH, 2015), tales como:

- Grandes volúmenes de datos estructurados, semiestructurados y no estructurados que cambian rápidamente
- Sprints ágiles, iteración de esquema rápida y frecuentes cambios de código.

### **2.2.8 MongoDB**

De acuerdo al sitio oficial de MongoDB (s.f.), Mongo es una base de datos de documentos con alta escalabilidad y flexibilidad, esto debido a que los documentos almacenados son similares a JSON, lo que significa que los campos pueden tener variaciones de un documento a otro y que la estructura de los datos también puede sufrir modificaciones con el tiempo.

El modelo de documento de estas bases de datos se correlaciona con los objetos en el código de la aplicación facilitando así el trabajo de los datos.

MongoDB es una base de datos distribuida en su núcleo, por lo que la alta disponibilidad, escala horizontal y distribución geográfica están integradas y son fáciles de usar –poner referencia. Una de las razones principales por las que se eligió mongo como solución para la capa de datos es que esta base de datos de documentos es fácil de aprender y utilizar, además que proporciona herramientas para cumplir con requisitos complejos y a cualquier escala, esto debido a la gran cantidad de controladores que posee y que también ha creado la comunidad de la misma.

### 2.2.8.1 Arquitectura de MongoDB

MongoDB fue creado con el fin de satisfacer las exigencias de las aplicaciones modernas con una base tecnológica (MongoDB, s.f.), que le permite a través de:

- El modelo de datos de documento, presentar una forma adecuada de trabajar con los datos.
- Un diseño de sistemas distribuidos, colocar datos de manera inteligente y en cualquier lugar.
- Una experiencia unificada, la libertad de ejecución desde cualquier lugar, que permita las pruebas del desarrollo realizado.

Los beneficios otorgados por dicha base datos y la razón para ser elegida por parte del grupo de investigadores, se soportan en los siguientes aspectos:

#### La mejor forma de trabajar con datos



*Ilustración 3. MongoDB y el trabajo de los datos*

*Tomado de: <https://www.mongodb.com/mongodb-architecture>*

- Fácil: Se puede trabajar con datos de forma natural e intuitiva.
- Rápido: Se obtiene un gran rendimiento.
- Flexible: Se pueden adaptar cambios rápidamente.
- Versátil: Se admite una amplia variedad de datos y consultas.

## Ponga datos de manera inteligente donde lo necesite



*Ilustración 4. MongoDB ubicación de datos*

*Tomado de: <https://www.mongodb.com/mongodb-architecture>*

- Disponibilidad: Se pueden construir aplicaciones resilientes a través de una sofisticada replicación y recuperación auto-curativa.
- Escalabilidad: Se permite crecimiento horizontal a través de fragmentación nativa.
- Aislamiento de carga de trabajo: Se ejecutan cargas de trabajo analíticas y operativas en el mismo clúster.
- Localidad: Se permite la colocación de datos en dispositivos específicos y en geografías específicas, según la clase de servicio, además de garantizar acceso de baja latencia.

## Libertad para correr en cualquier lugar



*Ilustración 5. MongoDB distribuido*

*Tomado de: <https://www.mongodb.com/mongodb-architecture>*

- Portabilidad: Una base de datos que funciona igual en todas partes.

- Cobertura global: Disponible como servicio en más de 50 regiones de los principales proveedores de nubes públicas.

### **2.2.9 Express**

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla. Debido a esto, una de las principales razones por las que el grupo de investigación se poyo en este Framework es que proporciona una delgada capa de características para las aplicaciones web que se mezclan con características de Node.js. (Express, s.f.).

### **2.2.10 Mongoose**

Mongoose proporciona una solución directa basada en esquemas que, como herramienta, permite el modelado de objetos MongoDB. Está diseñada para trabajar en un entorno asíncrono (Mongoose, s.f.). Además, se encarga de comunicar la base de datos de documentos con la lógica implementada que, para el caso de este proyecto de investigación, será realizada en Node.js.

### **2.2.11 Ionic**

Tal como se refiere en la documentación oficial de Ionic Framework (Ionic, s.f.), Ionic es un marco de desarrollo de aplicaciones móviles HTML5 destinado a la creación de aplicaciones móviles híbridas. Las aplicaciones híbridas son esencialmente sitios web pequeños que se ejecutan en un Shell de navegador en una aplicación que tiene acceso a la

capa de plataforma nativa. Las aplicaciones híbridas tienen muchos beneficios sobre las aplicaciones nativas puras, específicamente en términos de soporte de plataforma, velocidad de desarrollo y acceso a código de terceros.

Ionic sirve como el marco de interfaz de usuario front-end que maneja todas las interacciones de apariencia y de interfaz de usuario. Este marco de trabajo viene con elementos de interfaz de usuario móvil de estilo nativo, brindando maneras de crear aplicaciones móviles que eclipsen los marcos de desarrollo de HTML5 existentes. Dado que es un framework HTML5, necesita un contenedor nativo como Cordova o PhoneGap para ejecutarse como una aplicación nativa.

### **2.2.12 V8**

V8 es el motor de código abierto JavaScript de alto rendimiento de Google, escrito en C++. Se usa en Google Chrome, el navegador de código abierto de Google, y en Node.js, entre otros. Implementa ECMAScript como se especifica en ECMA-262, y se ejecuta en Windows 7 o posterior, macOS 10.12+ y sistemas Linux que usan procesadores IA-32, ARM o MIPS. V8 puede ejecutarse de forma independiente o puede integrarse en cualquier aplicación de C++ (Google Developers, s.f.).

Tomando como referente el repositorio oficial de GitHub (Thompson, 2015), V8 compila y ejecuta código fuente JavaScript, manejando la asignación de memoria para objetos. JavaScript se utiliza con mayor frecuencia para las secuencias de comandos del lado del cliente en un navegador, por ejemplo, para manipular objetos del Document Object Model (DOM). Sin embargo, el DOM no suele ser proporcionado por el motor de JavaScript, sino por un navegador. Lo mismo ocurre con V8: Google Chrome proporciona el DOM. Mientras tanto, V8 proporciona todos los tipos de datos, operadores, objetos y funciones especificados en el estándar ECMA. Así V8 permite a cualquier aplicación C++ exponer sus propios objetos y funciones a código JavaScript.

### **2.2.13 GIT**

Git es un sistema de control de versiones distribuidas de código abierto y gratuito diseñado para manejar cualquier tipo de proyecto con velocidad, eficiencia y rendimiento. Supera a las herramientas de SCM como Subversion, CVS, Perforce y ClearCase con funciones como ramificación local, áreas de preparación conveniente y flujos de trabajo múltiples.

“Git es un sistema de control de versiones de código abierto conocido por su velocidad, estabilidad y modelo de colaboración distribuida. Originalmente creado en 2006 para administrar todo el kernel de Linux, ahora Git cuenta con un completo conjunto de características, un equipo de desarrollo activo y varias comunidades de alojamiento gratuito. Git fue diseñado desde cero, prestando poca atención a los estándares existentes de sistemas centralizados de control de versiones.” (Hossein Ahmadpanah, 2015).

### **2.2.14 REST API's**

Un servicio web es una aplicación de software auto descriptiva que puede invocarse en la Web utilizando un conjunto de estándares (Rosado Gomez & Jaimes Fernández, 2018). Los programas cliente usan interfaces de programación de aplicaciones (API) para comunicarse con los servicios web. En términos generales, una API expone un conjunto de datos y funciones para facilitar las interacciones entre programas informáticos y permitirles intercambiar información. Una API web es la cara de un servicio web que escucha y responde directamente a las solicitudes de los clientes. Una API web que se ajuste al estilo arquitectónico REST es una API REST. Tener una API REST hace que un servicio web sea "RESTFUL", y consiste en un conjunto de recursos interconectados (Masse, 2012).

Según la plataforma API Market del banco BBVA (2016), REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos

estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad, pero también mucha complejidad.

#### *2.2.14.1 Características de las API REST*

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI. Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI facilita el acceso a la información para su modificación o borrado, o, por ejemplo, para compartir su ubicación exacta con terceros.
- Interfaz uniforme: para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- Sistema de capas: arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.

### **2.2.15 Arquitectura basada en componentes**

Una arquitectura basada en componentes describe una aproximación de ingeniería de software al diseño y desarrollo de un sistema. Esta arquitectura se enfoca en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor que los principios de orientación por objetos y no se enfoca en asuntos específicos de los objetos como los protocolos de comunicación y la forma como se comparte el estado (Meier, y otros, 2009).

El estilo de arquitectura basado en componentes tiene las siguientes características:

- Es un estilo de diseño para aplicaciones compuestas de componentes individuales.
- Pone énfasis en la descomposición del sistema en componentes lógicos o funcionales que tienen interfaces bien definidas.
- Define una aproximación de diseño que usa componentes discretos, los que se comunican a través de interfaces que contienen métodos, eventos y propiedades.

#### ***2.2.15.1 Principios***

Un componente es un objeto de software específicamente diseñado para cumplir con cierto propósito. Los principios fundamentales cuando se diseña un componente es que estos deben ser:

- Reusable. Los componentes son usualmente diseñados para ser utilizados en escenarios diferentes por diferentes aplicaciones, sin embargo, algunos componentes pueden ser diseñados para tareas específicas.
- Sin contexto específico. Los componentes son diseñados para operar en diferentes ambientes y contextos. Información específica como el estado de los datos deben ser pasadas al componente en vez de incluirlos o permitir al componente acceder a ellos.

- Extensible. Un componente puede ser extendido desde un componente existente para crear un nuevo comportamiento.
- Encapsulado. Los componentes exponen interfaces que permiten al programa usar su funcionalidad. Sin revelar detalles internos, detalles del proceso o estado.
- Independiente. Los Componentes están diseñados para tener una dependencia mínima de otros componentes. Por lo tanto, los componentes pueden ser instalados en el ambiente adecuado sin afectar otros componentes o sistemas.

Algunos de los beneficios obtenidos al trabajar con una arquitectura son los siguientes:

- Facilidad de Instalación, ya que cuando existen nuevas versiones es fácil el reemplazar la versión existente, debido a que no se impactan otros componentes o el sistema en general.
- Costos reducidos, los componentes de terceros distribuye el costo de desarrollo y del mantenimiento.
- Facilidad de desarrollo, la implementación de interfaces que proveen funcionalidad definida permiten la facilidad del desarrollo sin el impacto de otras partes del sistema.
- Reusable. El uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.
- Mitigación de complejidad técnica. Los componentes mitigan la complejidad por medio del uso de contenedores de componentes y sus servicios. Ejemplos de servicios de componentes incluyen activación de componentes, gestión de la vida de los componentes, gestión de colas de mensajes para métodos del componente y transacciones.

## METODOLOGÍA

---

En este proyecto de investigación, la recolección de información se realizó a partir de fuentes primarias como la entrevista y la observación directa, y de fuentes secundarias como artículos científicos, tesis de grado, documentación bibliográfica, entre otros, lo cual define al proyecto como una investigación de tipo bibliográfica.

El análisis llevado a cabo sobre las fuentes primarias de información se basó en las siguientes premisas: Obtener información de individuos o grupos, facilitar opiniones e influir sobre ciertos aspectos en derredor al desarrollo y ejecución del proyecto. Las características de la información recogida surgieron de un mecanismo focalizado en un tema y un contenido concreto: El registro de asistencia académica en la Universidad de Cartagena; aplicando en cada caso los objetivos correspondientes que se quieren alcanzar en los resultados, los cuales fueron analizados y consolidados en actas que sirvan como soporte.

Por otra parte, el análisis realizado sobre las fuentes secundarias, en relación a sistemas de registro de asistencia, fue meticuloso y se aplicaron los filtros de búsqueda título, número de citas, resumen y año de publicación, categorizando y seleccionando aquellos publicados en los últimos 5 años hasta la fecha de entrega final del proyecto (2013 - 2018). Lo anterior con el propósito de identificar, seleccionar y procesar los documentos pertinentes y contribuyentes al desarrollo de los propósitos del presente proyecto de investigación, almacenándolos en formato PDF en orden cronológico y por nivel de importancia. Los principales recursos de búsqueda de la información fueron las bases de datos de la Universidad de Cartagena y el buscador de documentos académicos Google Scholar.

El proyecto también cuenta con un carácter de investigación aplicada ya que diseñó e implementó una plataforma virtual para ser ejecutada en el entorno académico de la Universidad de Cartagena, atendiendo el proceso de registro y control de asistencia académica de la misma.

Se utilizó el Lenguaje de Modelado Unificado (UML), haciendo uso del software Enterprise Architect 8.0 para diagramar los diferentes artefactos software. Con base en la metodología de desarrollo de software RUP, se desarrollaron las siguientes actividades para dar cumplimiento a cada objetivo específico y así satisfacer las necesidades y el logro del objetivo general del proyecto. Igualmente se tuvo en cuenta el aporte que hacen los autores Alejandro Martínez & Raúl Martínez al definir los flujos de trabajo de ingeniería para RUP en su artículo “Guía a Rational Unified Process”, así:

**I. Definir el modelo de negocio junto con la especificación de requisitos funcionales y no funcionales, aplicando en cada uno de los casos las técnicas de recolección de información que proporcionen datos de valor para el análisis.**

Inicialmente se aplicaron técnicas de recolección de información tales como entrevistas, con el fin de conocer el problema y posteriormente identificar una visión holística y clara del proceso de registro y control de asistencia académica llevado a cabo en la Universidad de Cartagena, atendiendo aspectos como qué inconvenientes presenta, cómo se desarrolla, quienes intervienen y qué lineamientos lo regulan; y la revisión de la bibliografía y la literatura, que permitió estar al tanto de soluciones tecnológicas que con anterioridad han sido propuestas y ejecutadas para solventar problemáticas enmarcadas dentro de los procesos de registro y asistencia.

Posteriormente se desarrollaron e hicieron entregas de los artefactos software: Modelo de Dominio, Diagrama de Actividades y Diagrama de Casos de Uso a nivel del mundo real. Esto con el fin de presentar de una forma concisa las dependencias entre el negocio y el software construido, llegando a un mejor entendimiento del entorno en donde se desenvuelve el problema en cuestión, asegurando que el producto software entregado sea de utilidad, consiguiendo que se adapte adecuadamente en la universidad, y teniendo un marco común para los stakeholders del proyecto.

Se construyó el documento para la especificación de los requisitos del software, SRS (Software Requirements Specifications) definido por el estándar IEEE 830, con el fin de

establecer detalladamente los aspectos a tener en cuenta dentro de la etapa de construcción. Este documento representa un contrato aceptado por las partes involucradas dentro del proyecto, y define las diferentes características a exhibir (requisitos no funcionales) y las funcionalidades (requisitos funcionales) con las que cuenta la solución software.

Para llevar a cabo la construcción del documento se definieron, junto con los interesados, los límites del software, teniendo en cuenta aspectos que atendieran la perspectiva del producto y las funciones del mismo, las características de los usuarios, las restricciones del software y demás aspectos definidos por el estándar.

Además, se elaboró un diagrama de casos de uso a nivel de requisitos que muestre de forma clara cuales son los actores que intervienen para hacer uso del software, y la forma en que se relacionan con los diferentes requisitos establecidos en el desarrollo del documento.

“En definitiva, en este flujo de trabajo hay que analizar el problema, comprender las necesidades de los interesados y expresarlas en forma de requisitos; construir diagramas de casos de uso para los requisitos funcionales, los no funcionales describirlos textualmente en especificaciones suplementarias. Además, hay que gestionar los cambios en los requisitos a lo largo de todo el proceso.” (Martínez & Martínez, 2014)

## **II. Construir un modelo de diseño apropiado a partir de la interpretación de los artefactos software evidenciados durante el modelo de negocio y teniendo en cuenta los factores de calidad pertinentes durante el proceso de desarrollo de software.**

Para el cumplimiento y la construcción de un modelo de diseño apropiado, se realizó un análisis exhaustivo de las funcionalidades que desempeña el software, teniendo en cuenta en todo momento los requisitos (funcionales y no funcionales) descritos en los artefactos generados a partir del modelo de negocio. Para ello, en primera medida se definió una arquitectura de software que otorgue un orden lógico a la solución y que sea trascendental para los diversos artefactos generados en el presente flujo de trabajo. Así, a partir del

“Modelo de Vistas 4+1” propuesto por Philippe Kruchten (Kruchten, 1995), se elaboraron los diferentes artefactos de software pertinentes al diseño, así:

- **vista lógica**, representada por diagrama de clases y componentes,
- **vista de desarrollo**, representada por diagrama de paquetes,
- **vista de procesos**, representada por diagrama de secuencia,
- **vista física**, representada por diagrama de despliegue, y
- **vista de escenarios**, representada por diagrama de casos de uso a nivel de diseño, descripción de casos de uso y diseño de interfaz gráfica de usuario.

Con el análisis realizado se abordaron los requisitos funcionales a partir de la pregunta ¿Qué hace el sistema?, mientras que con el diseño se garantizó el enfoque en cómo cumplir con los objetivos, atendiendo los requisitos no funcionales.

### **III. Implementar la plataforma que atiende al problema, haciendo correcto uso de los patrones de diseño y arquitectura definidos en el modelo de diseño y con un oportuno proceso de verificación que, aplicando teorías asociadas a la experiencia de usuario, pueda brindar ambientes de trabajo agradables y confiables.**

Se llevó a cabo el proceso de codificación de la plataforma virtual, implementando tanto un componente web como móvil, en aras de obtener una versión funcional del software. Para cumplir con lo anterior, se hizo uso de los patrones generales de software para la asignación de responsabilidades (GRASP), además de los propuestos por la banda de los 4 (Gang of four), haciendo énfasis en patrones como alta cohesión y bajo acoplamiento para suplir los requisitos funcionales pactados y los no funcionales imprescindibles tales como mantenibilidad, flexibilidad, seguridad, correctitud, usabilidad, etc., además de los descritos en el SRS. Como agregado al cumplimiento del presente objetivo, se desarrollaron los manuales de usuario y del sistema del software.

#### **IV. Probar la plataforma a través de mecanismos de testeo y aceptación que validen la correlación entre lo especificado por el usuario y el producto entregado.**

Se diseñaron y aplicaron diferentes etapas de prueba al software desarrollado, para verificar la corrección y exactitud de la plataforma codificada. A partir de los resultados obtenidos por cada grupo de pruebas realizadas, se hizo entrega de un análisis donde se documentaron los aspectos más relevantes y pertinentes que sean de interés sobre el desempeño del sistema.

En el presente informe, en la sección 4.4 (Pruebas), se encuentra un análisis más detallado de las funcionalidades testeadas por cada actor del sistema y el equipo de investigación, identificando el comportamiento y correcto funcionamiento de la plataforma, teniendo en cuenta las pruebas de caja negra, como mecanismo de testeo, y el diseño de pruebas (ver anexo AM) construido a partir del documento de Especificación de Requisitos del Software (SRS), como mecanismo de aceptación.

## RESULTADOS Y DISCUSIÓN

---

Debido al proceso de investigación realizado se pudo llevar a cabo la construcción de una plataforma para el registro y control de asistencia académica en la universidad de Cartagena, que permite a los docentes la gestión de las clases realizadas y de los proyectos docentes construidos; a los auditores, la verificación de los temas tratados en clases y realizar análisis de los resultados obtenidos en estas, y a los estudiantes ser partícipes del proceso como veedores del mismo. Este proceso además permite detallar, en la presente sección, los diferentes resultados obtenidos durante su construcción.

La plataforma, la cual se ha denominado como SIRECAA, es descrita a continuación a través de las siguientes cinco secciones: *modelo de negocio*, que describe a nivel de mundo real la problemática presentada y evidencia en contexto lo sucedido en la Universidad de Cartagena con respecto a cómo se lleva a cabo el proceso de registro y control de asistencia académica; *la especificación de requisitos*, que permitió la definición tanto de los requisitos funcionales como no funcionales que sustentan la construcción de la plataforma y que define las decisiones a nivel de diseño que posteriormente se tuvieron en cuenta; *el modelo de diseño*, el cual de forma detallada muestra la interacción de los diferentes componentes y piezas de software que conforman la plataforma, la arquitectura de desarrollo que la define y las decisiones a nivel del proceso de desarrollo que fueron tomadas; finalmente la *sección de pruebas*, donde se denotan los resultados obtenidos al someter a la plataforma construida a los diferentes casos de prueba diseñados, los cuales permitieron analizar el comportamiento y funcionamiento de SIRECAA.

## **4.1 *MODELO DE NEGOCIO***

Esta sección presenta el modelo de negocio del proceso de registro y control de asistencia en la Universidad de Cartagena, el cual está representado por medio de los siguientes modelos:

### **4.1.1 Modelo de dominio**

Este modelo detalla las diferentes entidades que interactúan dentro del mundo real y como es la relación entre cada una de ellas. Para esto, se identifican las relaciones, multiplicidades y atributos pertenecientes a esas entidades, del cual se puede determinar que: los docentes de la universidad de Cartagena realizan registros de las clases dictadas a diario, además realizan los proyectos docentes pertenecientes a la asignatura dictada y los cuales contienen los temas a tratar durante el periodo académico; los auditores se dirigen a un departamento académico de la institución para seleccionar y auditar los registros académicos realizados por los docentes; y los estudiantes asisten a clases y su asistencia es consignada dentro de los registros diligenciados (Ver Ilustración 6).

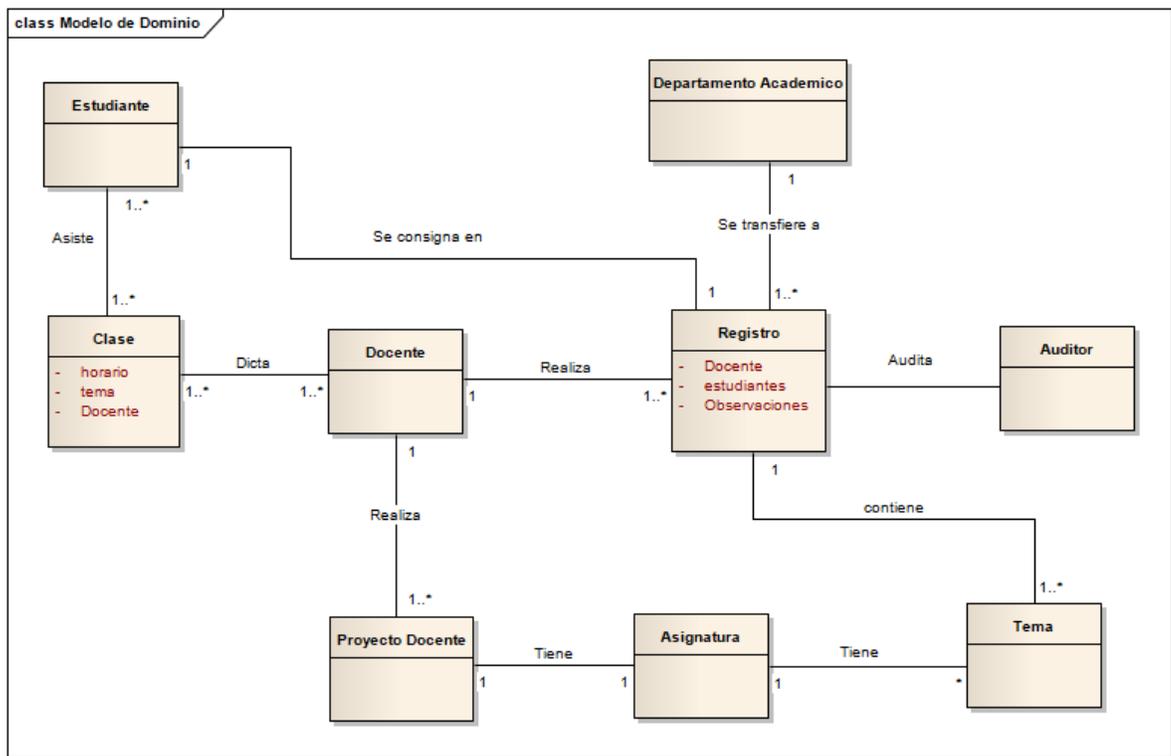
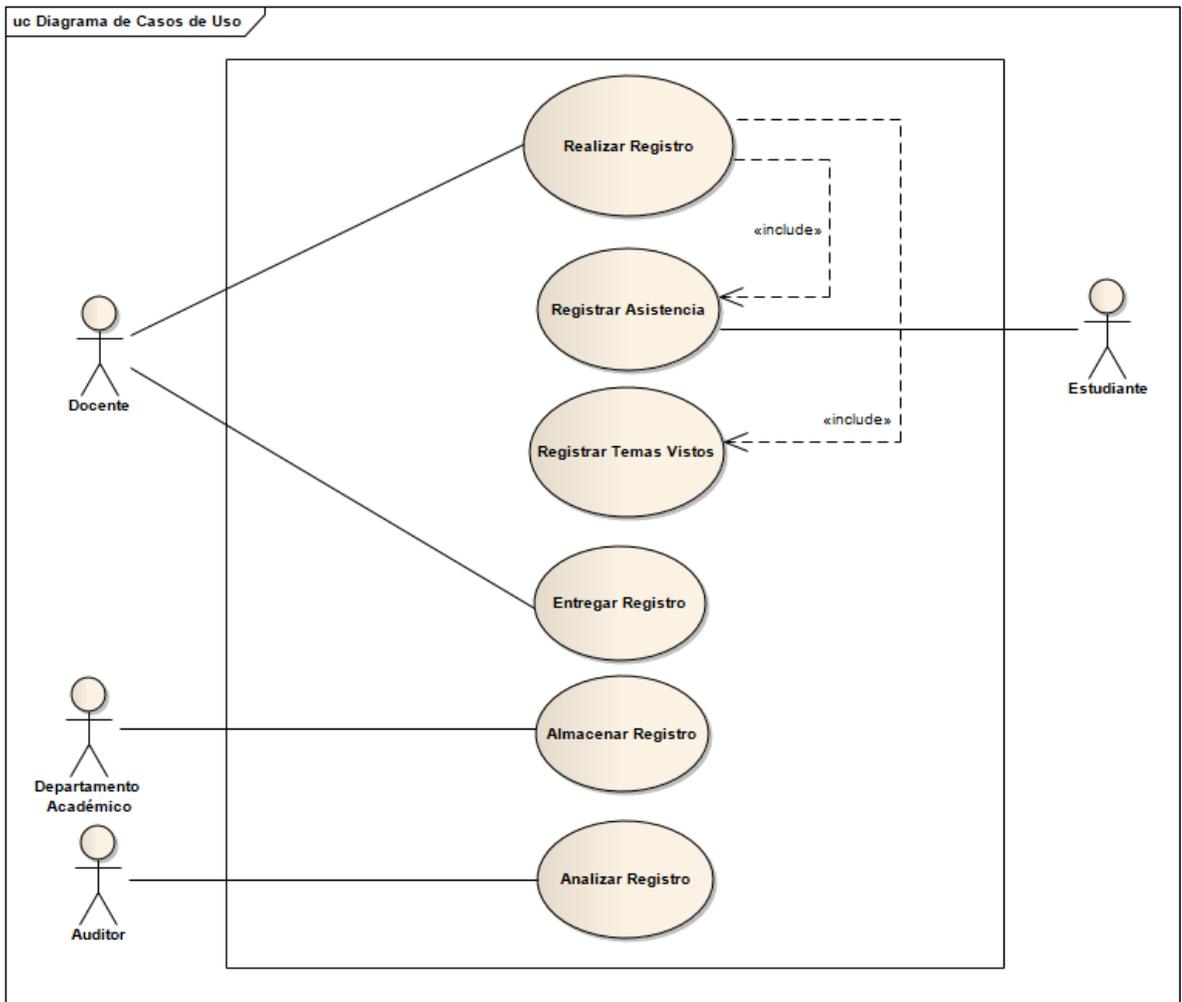


Ilustración 6. Modelo de dominio

#### 4.1.2 Casos de uso del mundo real

El diagrama de casos de uso a nivel de mundo real muestra la interacción que presentan cada uno de los actores del mundo real con el proceso de registro y control de asistencia académica de la universidad de Cartagena, donde el docente realiza registros de asistencia que le permiten registrar la asistencia y registrar los temas vistos en clase; el auditor analiza los diferentes registros que le competen a su área de interés; el estudiante como actor de apoyo es participe del proceso de registro de asistencia; y el departamento académico almacena diariamente los registros consignados (Ver Ilustración 7).



*Ilustración 7. Casos de uso a nivel de mundo real*

### **4.1.3 Procesos del negocio**

Este diagrama detalla las acciones realizadas por cada uno de los actores involucrados, mostrando la relación existente entre ellos y el orden cronológico en que se ejecutan en el contexto del mundo real. El docente y los estudiantes inician el flujo de las actividades asistiendo a la clase, luego se ejecutan las actividades de desarrollo de temas y registro de asistencia, las cuales presentan un tiempo limitado correspondiente al tiempo de duración de la clase. Posteriormente se lleva a cabo la realización completa del registro de asistencia, el cual es archivado por el departamento académico y consultado y analizado por el auditor posteriormente según sea requerido (Ver Ilustración 8. Se anexa en formato digital).

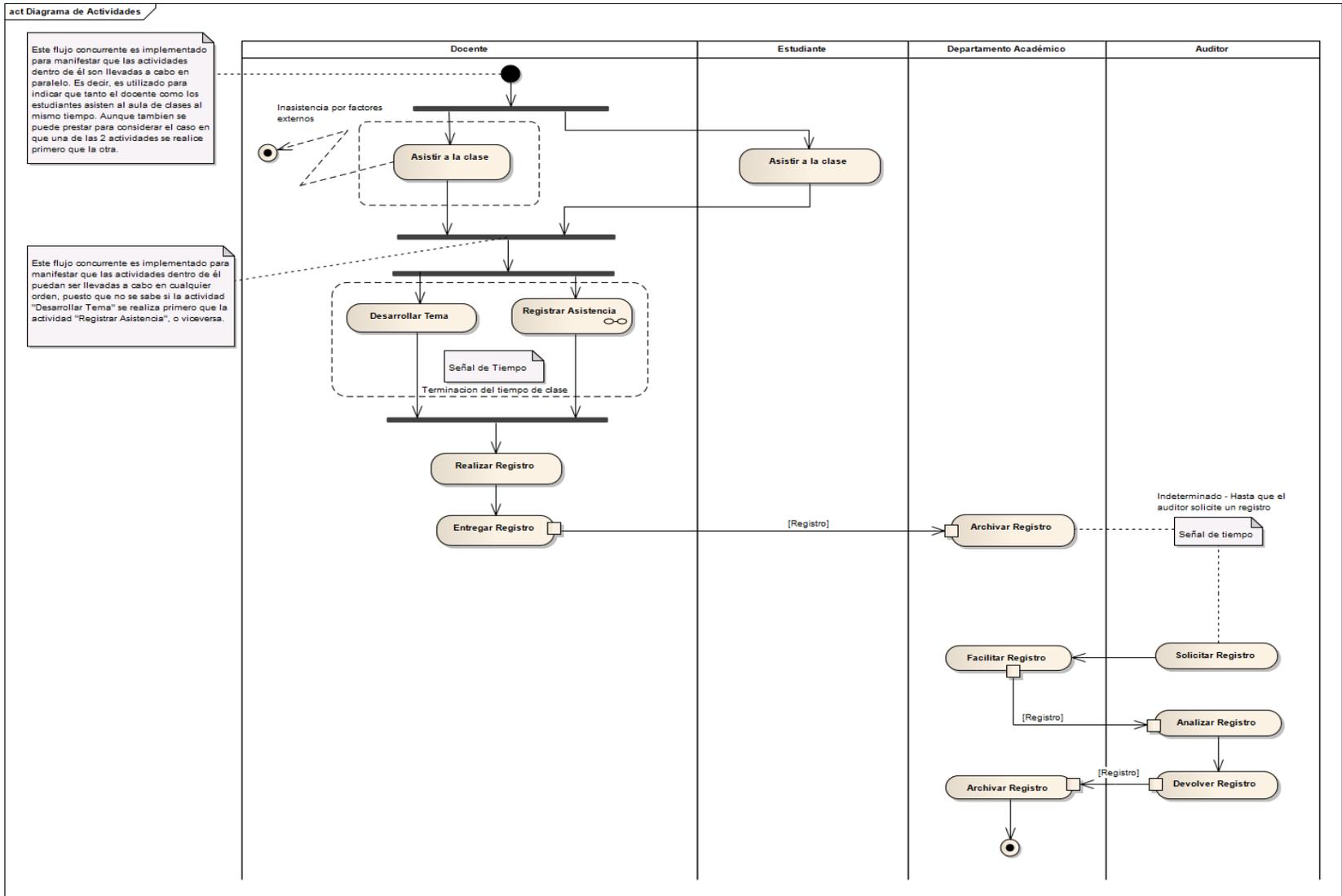


Ilustración 8. Procesos del negocio

## 4.2 ESPECIFICACIÓN DE REQUISITOS

La plataforma SIRECAA permite a docentes el registro de la asistencia académicas de acuerdo a un proyecto docente definido por un auditor. El administrador del sistema, es el encargado de la gestión del departamento académico a su cargo y maneja procesos sensibles de la plataforma, como gestión de departamentos internos, gestión de cursos y procesos de sincronización con las bases de datos de la Universidad. Adicionalmente la plataforma permite a estudiantes la visualización de los registros realizados por los docentes y ha auditores tomar decisiones con respecto a la información almacenada dentro del sistema en forma de datos y gráficos que faciliten su gestión.

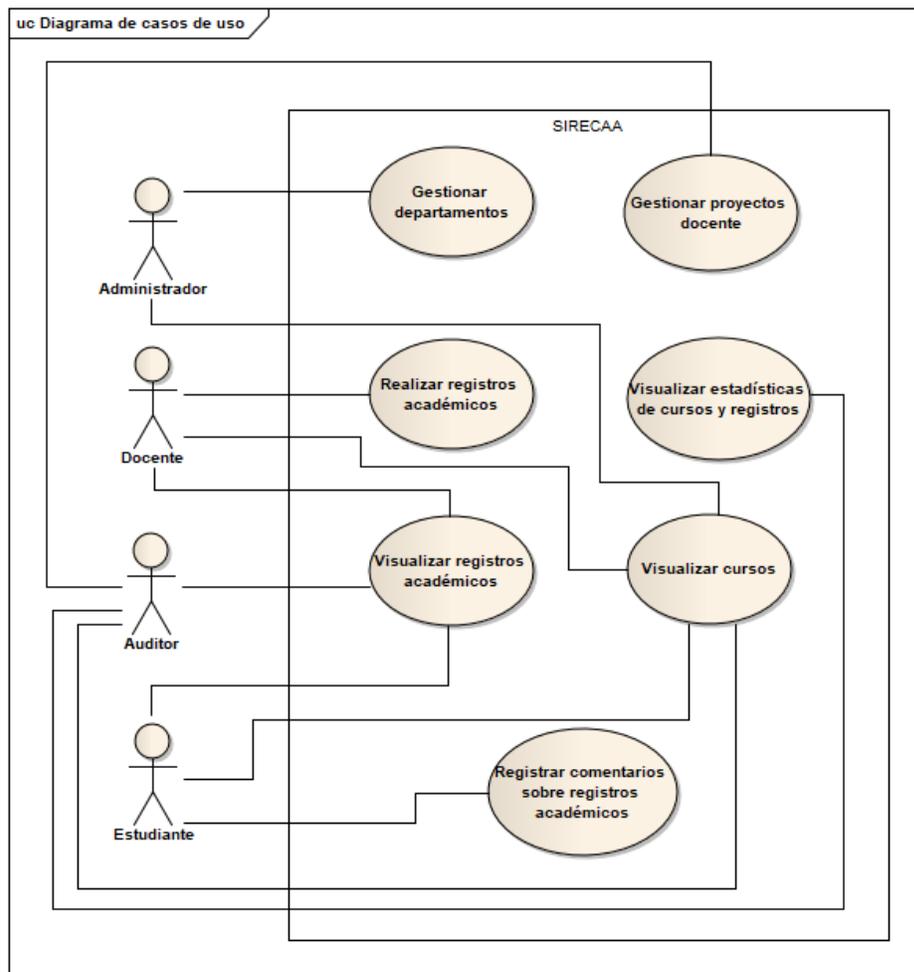


Ilustración 9. Diagrama de casos de uso a nivel de requisitos.

Cada una de las características anteriormente mencionadas son explicadas a detalle en el en el ANEXO G: Especificación de requisitos del software (SRS), del capítulo de anexos.

### **4.3 *MODELO DE DISEÑO***

La arquitectura del sistema se presenta usando el modelo de vista 4+1 propuestas por Philippe Kruchten que permiten describir la partes que componen la plataforma SIRECAA, las relaciones entre ellas y con el entorno a través de; la vista lógica, vista de procesos, vista despliegue y vista de escenarios, así:

#### **4.3.1 *Vista de escenarios***

Esta vista es la encargada de mostrar la interacción de todos los actores con el sistema, describiendo las acciones realizadas por estos y el comportamiento que tiene el sistema de acuerdo a ello. Está representada por el diagrama de casos de uso y su especificación. La descripción de los casos de uso se puede consultar como parte del manual del sistema, el cual puede ser accedido a través del siguiente enlace, en la sección *Modelo de diseño – Vista de escenarios*: <http://sirecaa.grupo soluciones.com/manuales/sistema/>.

4.3.1.1 Diagrama general de casos de uso

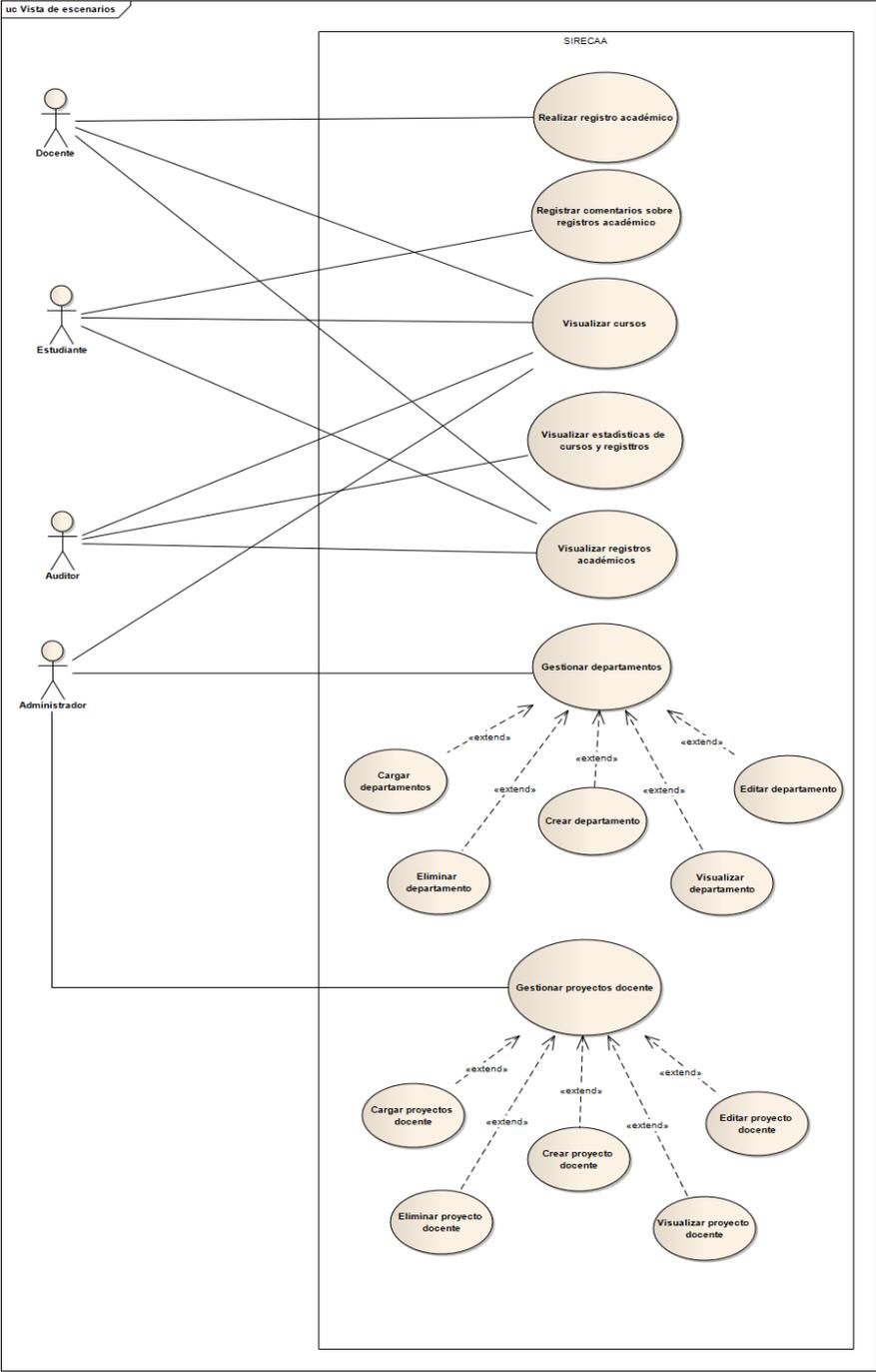


Ilustración 10 Diagrama general de casos de uso

### **4.3.2 Vista lógica**

La vista lógica representa la estructura y la funcionalidad del sistema, está representada por el diagrama de clases y el diagrama de componentes:

#### ***4.3.2.1 Diagrama de clases***

El diagrama de clases se segmentó por cada una de las funcionalidades definidas en el diagrama de especificación de requisitos.

##### 4.3.2.1.1 Funcionalidad gestionar departamentos

Esta funcionalidad realizada por el actor administrador permite a través del paquete Pages hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend para solicitar la información necesaria referente a los departamentos internos del programa académico que se encuentran en la API entregada por la universidad, para su recepción se crean modelos específicos y para la nueva información de los departamentos internos que serán creados. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 11. Se anexa en formato digital).

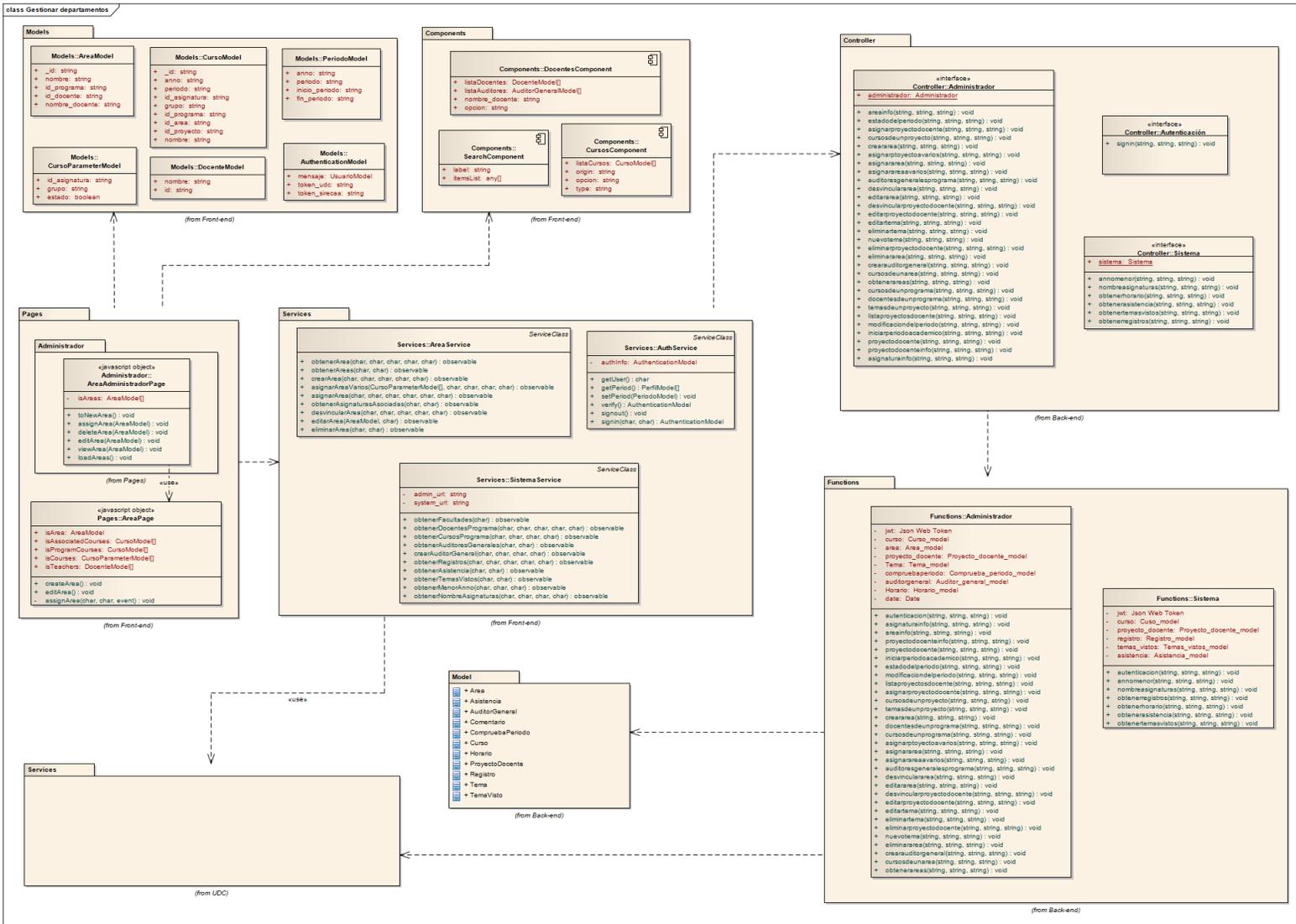


Ilustración 11 – Gestionar departamentos

#### 4.3.2.1.2 Funcionalidad gestionar proyectos docentes

Esta funcionalidad realizada por el actor auditor permite a través del paquete Pages hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend para solicitar la información necesaria referente a los proyectos docentes realizados por los docentes en el desarrollo de sus clases, toda esta información almacenada en las bases de datos de nuestra aplicación y que es independiente de la interfaz provista por la universidad. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 12. Se anexa en formato digital).



#### 4.3.2.1.3 Funcionalidad realizar registros académicos

Esta funcionalidad, realizada por el actor docente, permite la consignación de los registros académicos propios a la clase dictada en correspondencia con el plan establecido desde el proyecto docente que este tenga asociado. A través del paquete Pages se puede hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 13. Se anexa en formato digital).

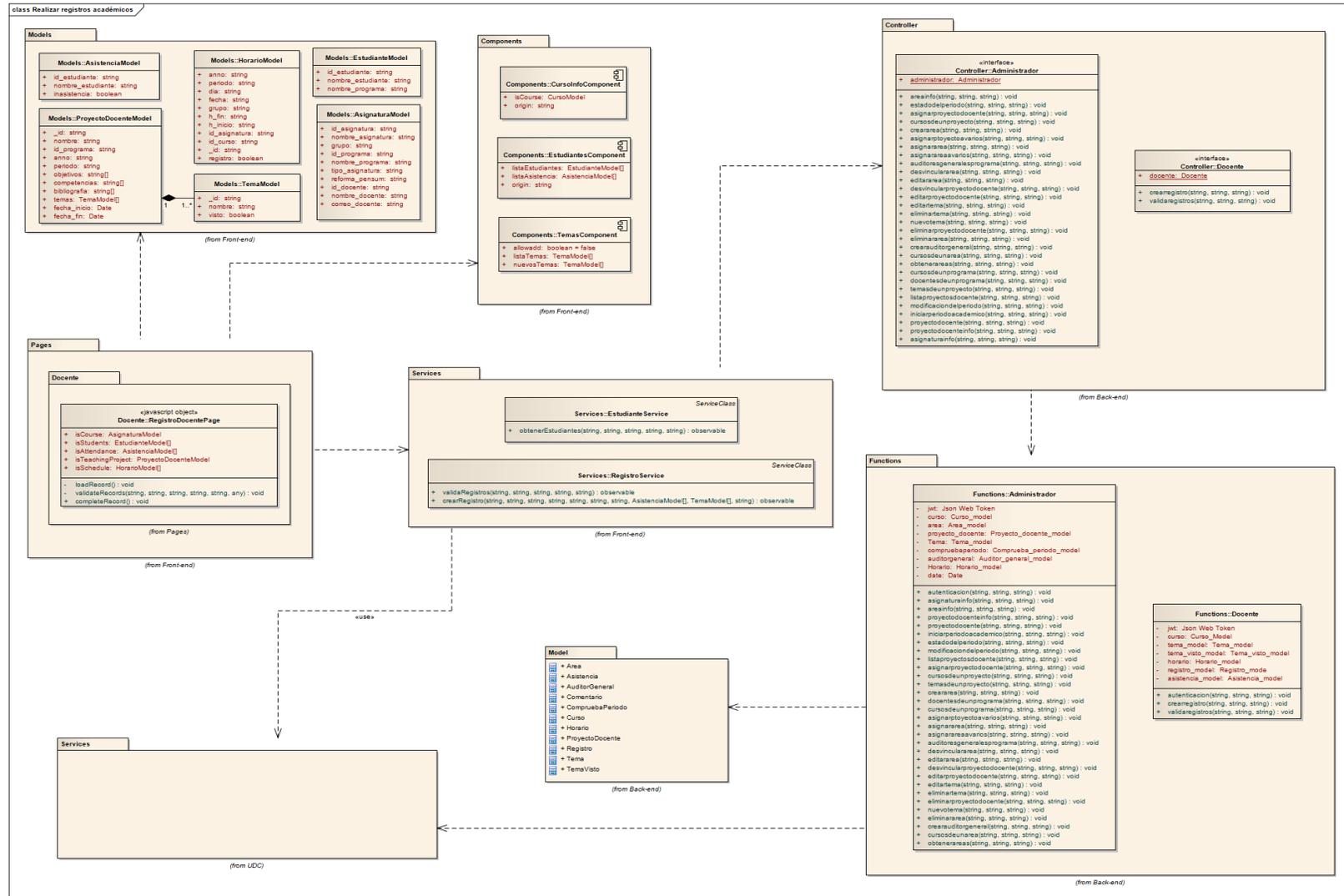


Ilustración 13 – Realizar registros académicos

#### 4.3.2.1.4 Funcionalidad visualizar cursos

Esta funcionalidad, realizada por los actores del sistema, permiten la visualización de los cursos que este tenga asociados a nivel de departamento académico. A través del paquete Pages se puede hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 14. Se anexa en formato digital).



#### 4.3.2.1.5 Funcionalidad visualizar registros académicos

Esta funcionalidad, realizada por el actor docente, permiten la visualización de los registros académicos que este haya consignado a lo largo de desarrollo del plan curricular y del proyecto docente asociado al curso, esto le permitirán tanto ver registros actuales como el histórico de los registros realizados por el mismo. A través del paquete Pages se puede hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend que para este caso se conectan con la API provista por la universidad. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 15. Se anexa en formato digital).

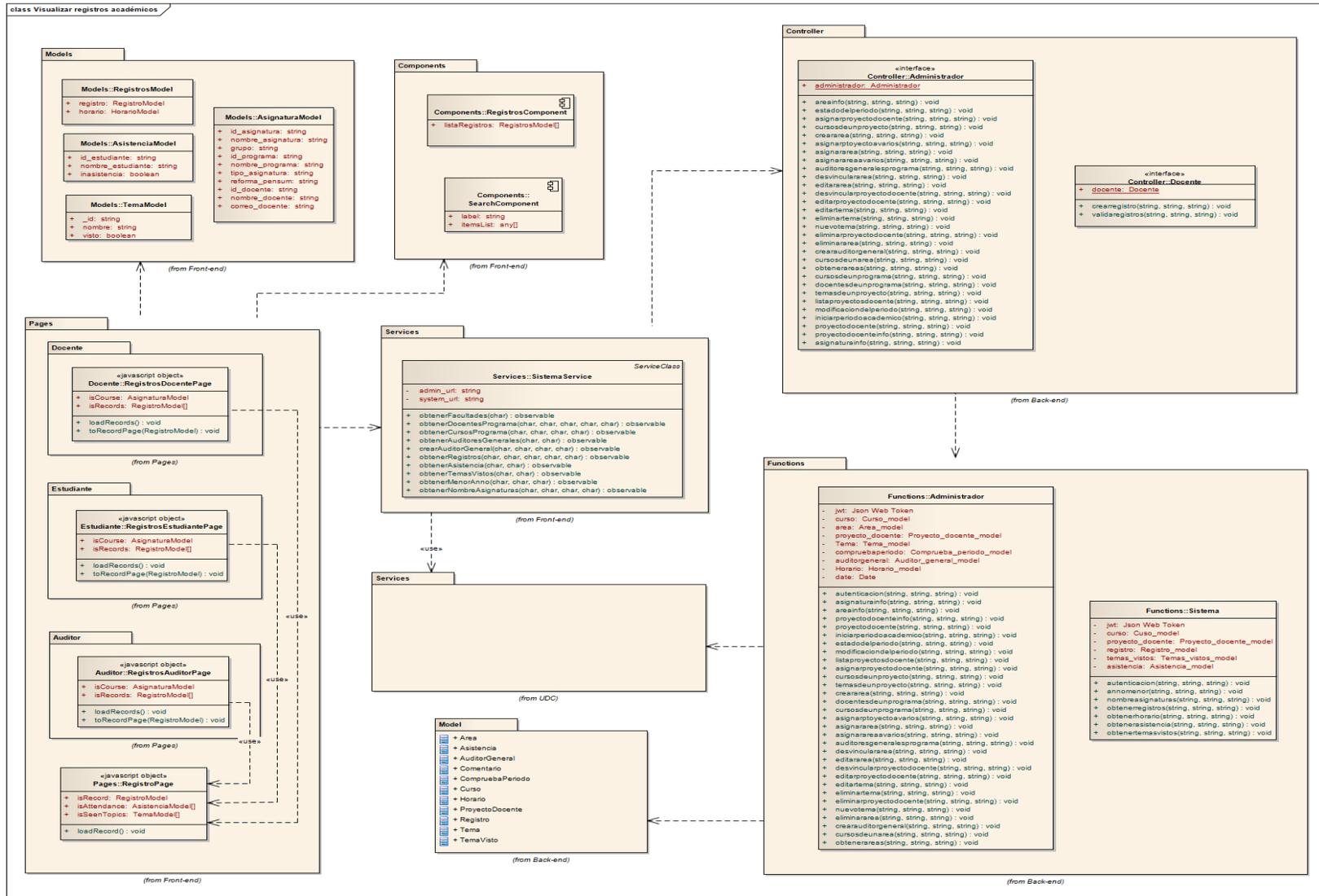


Ilustración 15 – Visualizar registros académicos

#### 4.3.2.1.6 Funcionalidad registrar comentarios sobre registros académicos

Esta funcionalidad realizada por el actor estudiante le permite el registro, edición y eliminación de comentarios sobre los registros académicos de los cursos a el cual este hace parte y que serán visualizados por los auditores. A través del paquete Pages se puede hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend que para este caso se conectan con la API provista por la universidad. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 16. Se anexa en formato digital).

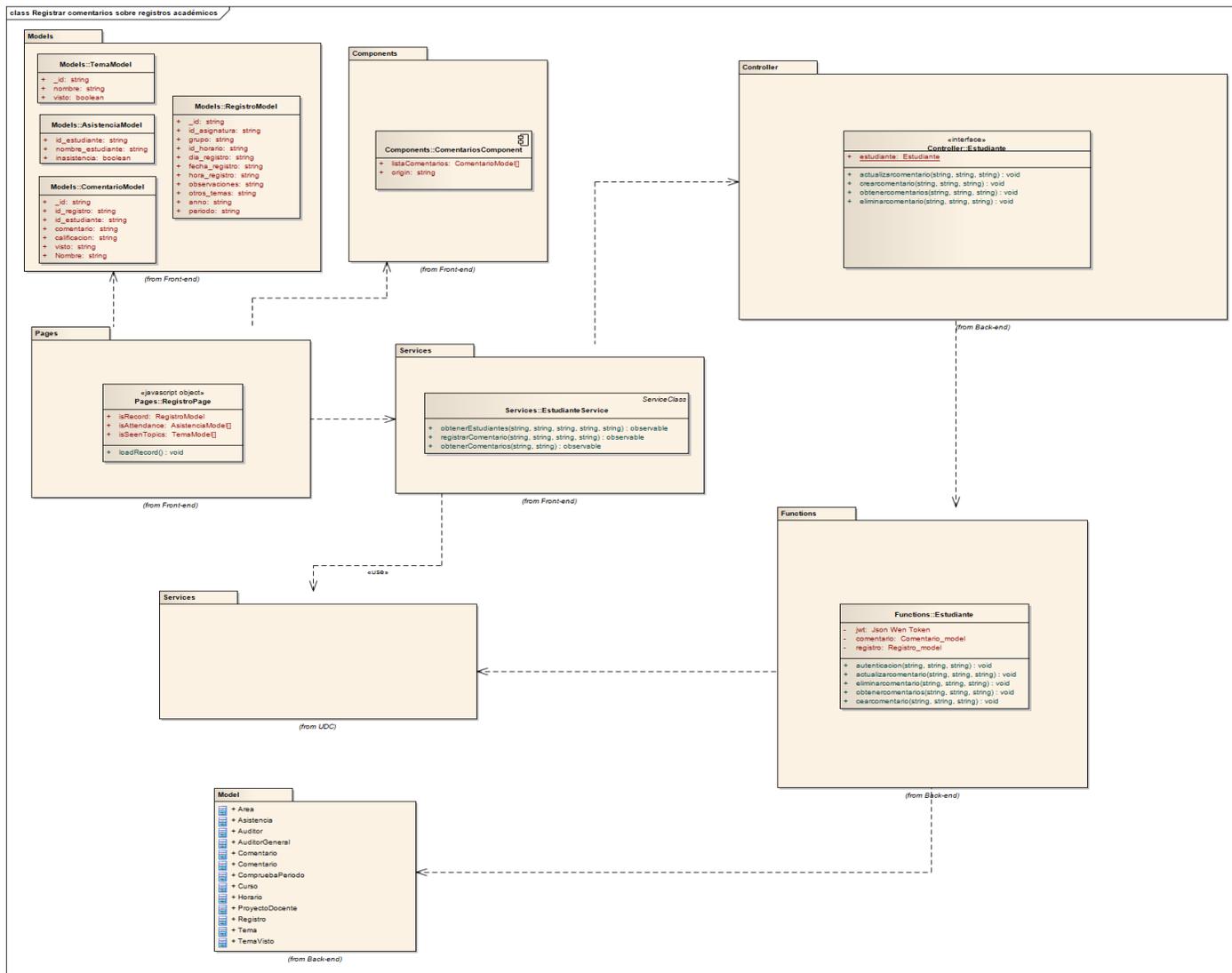


Ilustración 16 – Registrar comentarios sobre registros académicos

#### 4.3.2.1.7 Funcionalidad visualizar estadísticas de cursos y registros

Esta funcionalidad realizada por el actor auditor le permite visualizar diferentes estadísticas tales como; porcentaje de asistencia, porcentaje de cumplimiento del proyecto docente, porcentaje de asistencia docente, entre otras estadísticas que facilitan la toma de decisiones que este realiza en el ejercicio de sus funciones. A través del paquete Pages se puede hacer uso tanto de los modelos declarados; que determinan la estructura de los datos a recibir por parte de los servicios, como de los componentes que permiten la agrupación de los diferentes modelos y clases HTML que hacen parte de la vista. Los servicios a su vez se comunicarán con los controladores desarrollados desde el backend que para este caso se conectan con la API provista por la universidad. En el diagrama se especifican todas las clases tanto lógicas como HTML que intervienen. (Ver Ilustración 17. Se anexa en formato digital).

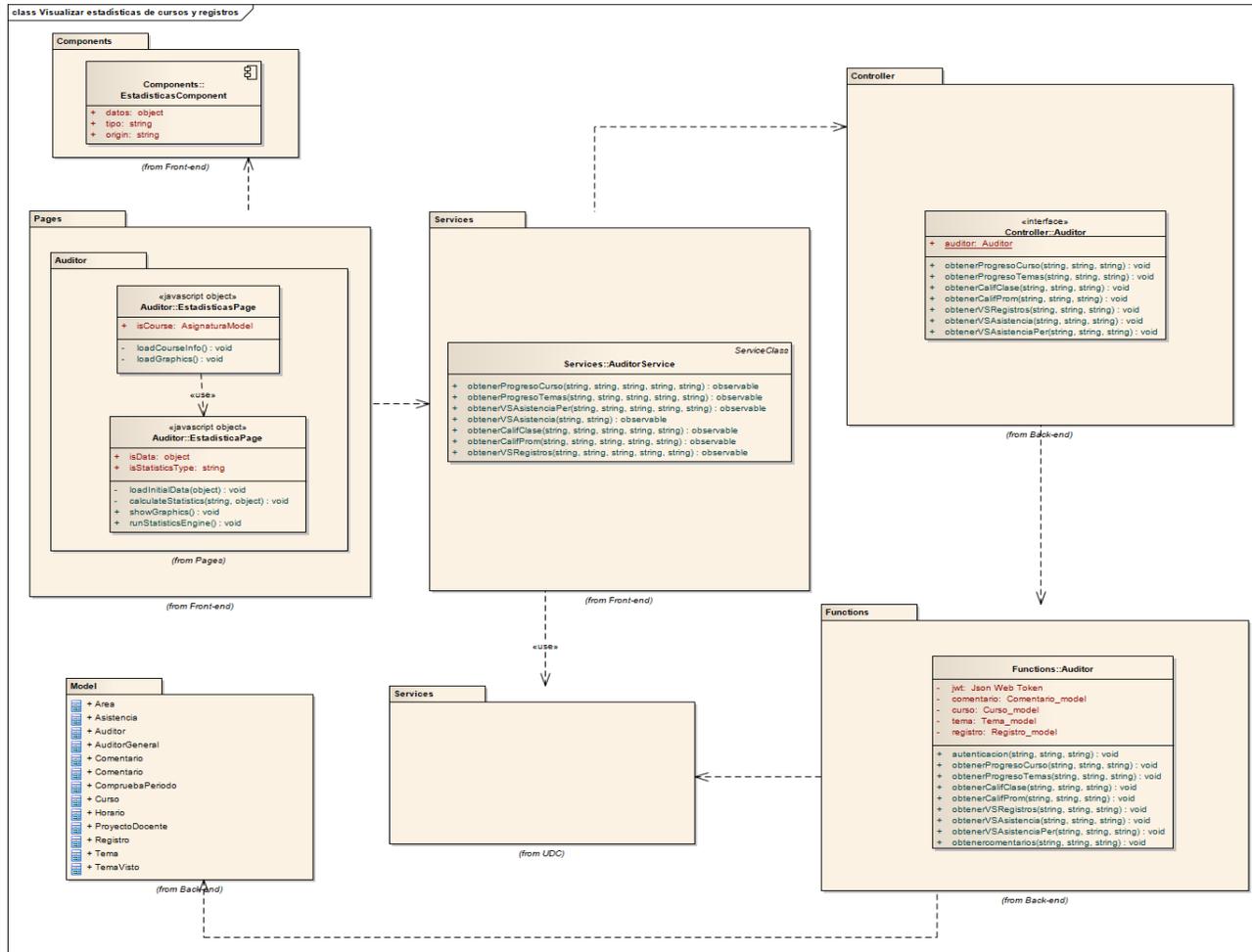


Ilustración 17 – Visualizar estadísticas de cursos y registros

#### ***4.3.2.2 Diagrama de componentes***

La composición de la plataforma SIRECAA está dada en dos grandes capas; una capa de presentación y una lógica, que determina de manera detallada los dos grandes componentes que la conforman. Para el componente la capa de presentación que representan el Front-End de la plataforma se despliega los formularios y vistas necesarios para cada uno de los actores estos a su vez se relacionan con distintos modelos que permiten tan recibir como enviar información a servicios que se comunican con el componente Back-End de la plataforma. El componente lógico que representa el Back-End de la solución ha sido expuesto como una API que hace uso la primera capa para soportar todas las operaciones y funcionalidades definidas, donde se determinan rutas de acceso particulares para cada actor como se visualiza en el diagrama y donde además se soporta la comunicación con la interfaz provista por la Universidad de Cartagena para el manejo de datos sensibles tales como credenciales, cursos, estudiantes, docentes y administrativos. Todas las dependencias y relaciones entre cada uno de los componentes tanto internos como externos son detallados en el diagrama expuesto a continuación (Ver Ilustración 18. Se anexa en formato digital):

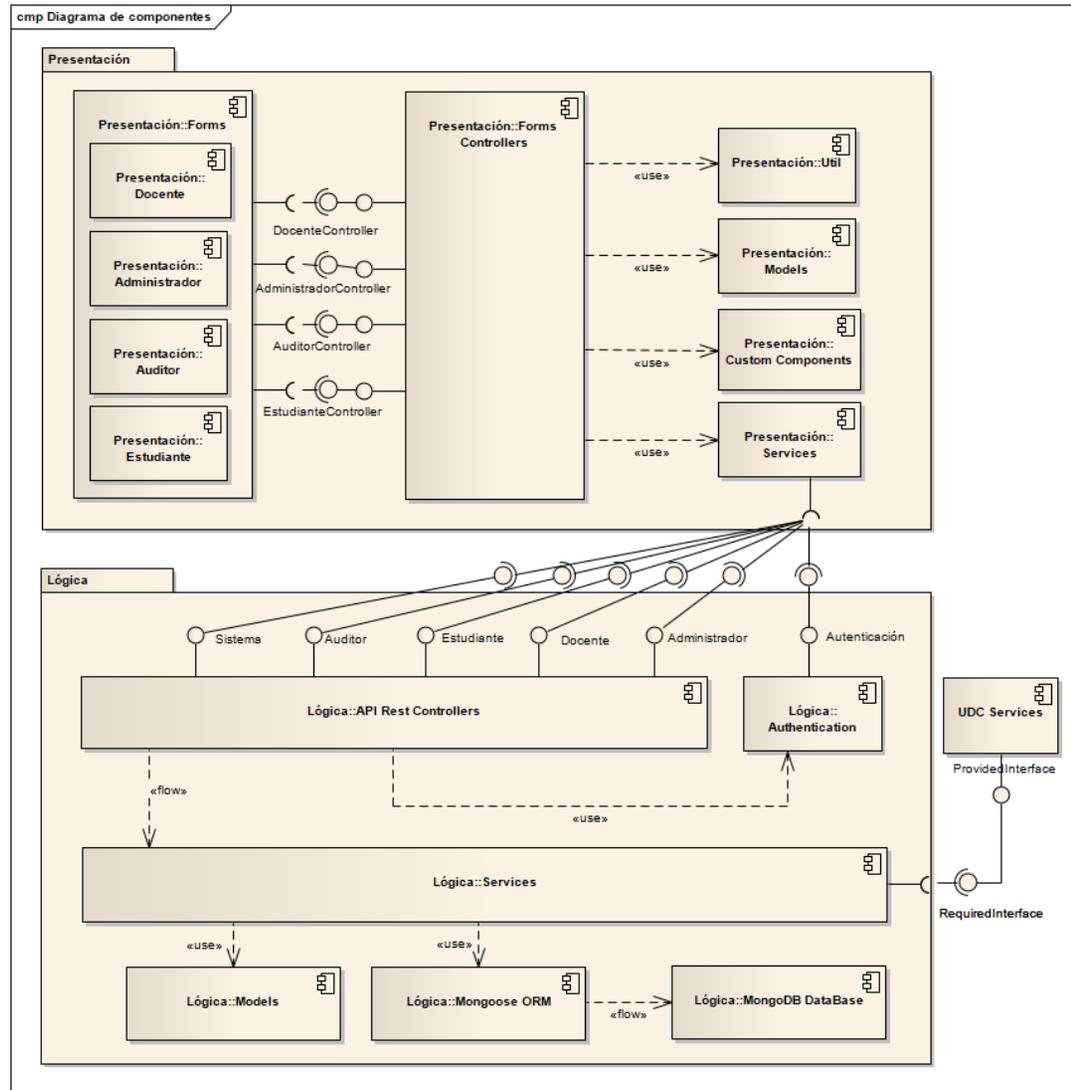


Ilustración 18 – Diagrama de componentes

### 4.3.3 Vista de procesos

Esta vista es la encargada de mostrar el flujo o comportamiento del sistema de acuerdo a las acciones de los usuarios, denotando las clases, métodos y objetos que intervienen en el flujo de una acción que retorna información para el usuario de SIRECAA. Esta vista se encuentra representada por los diagramas de secuencia los cuales para mayor legibilidad han sido desglosados por cada uno de los casos de usos del sistema.

#### 4.3.3.1 Diagrama de secuencia

El diagrama de secuencia por facilidad se segmentó por cada una de los casos de uso definidos dentro de SIRECAA.

##### 4.3.3.1.1 Caso de uso: Gestionar departamentos

La gestión de departamentos dentro de SIRECAA permite la creación, edición, eliminación y visualización de los diferentes departamentos internos asociados a un programa académico, así:

##### 4.3.3.1.1.1 Cargar departamentos

Para la carga de los departamentos internos pertenecientes a un departamento el actor Administrador debe autenticarse ante SIRECAA, después de ello se hace llamado a la clase *AreaService* que envía a la clase *AdministradorController* que se comunica con la clase *AdministradorFunction* el programa al que pertenece para obtener como respuesta la lista de los departamentos internos a los que puede acceder dicho Administrador. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir

u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 19. Se anexa en formato digital)

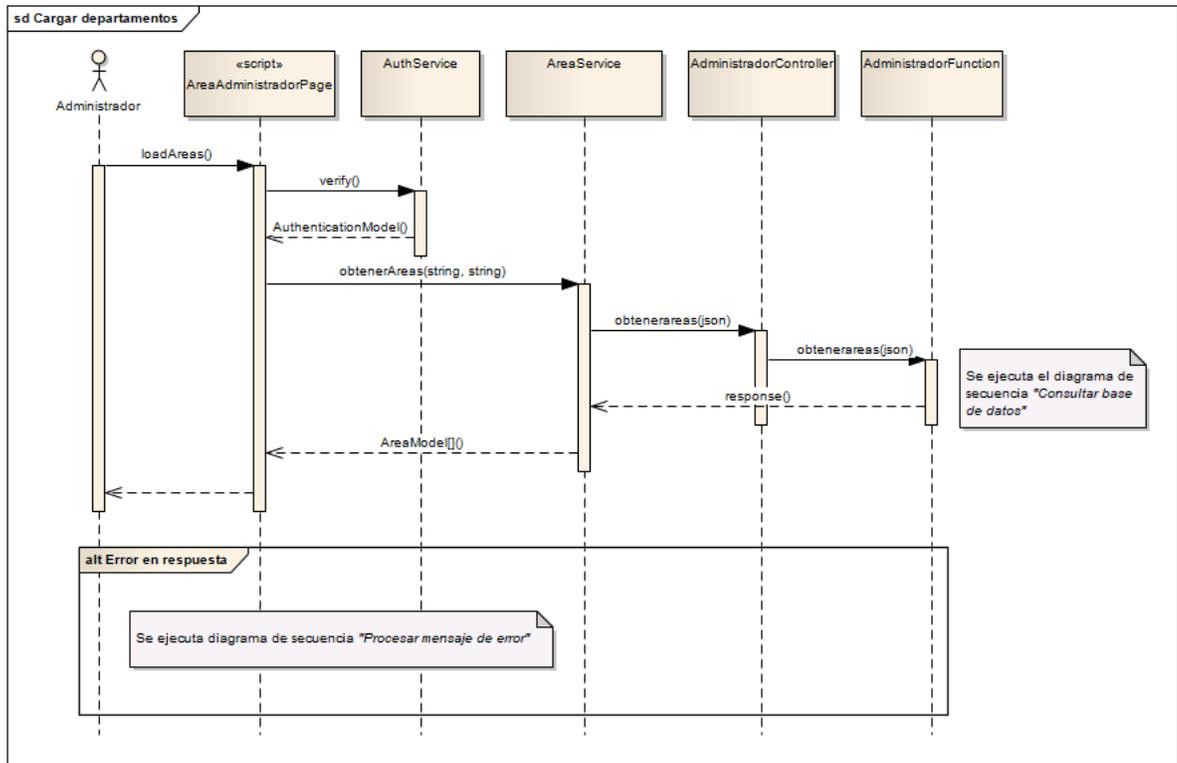


Ilustración 19 – Diagrama de secuencia del caso de uso: Cargar departamentos

#### 4.3.3.1.1.2 Crear departamento

Para la creación de los departamentos internos de un programa lo que primero que debe suceder dentro del sistema además de la autenticación ante SIRECAA es la invocación al caso de uso Cargar departamentos que le permitirá visualizar los departamentos previamente creados, cuando el actor Administrador decida la creación de una nueva área la clase `AreaAdministradorPage` ejecuta la acción `toNewArea` que a su vez ejecuta el método `loadAreas` dentro de la clase `AreaPage`, que permite obtener primero los docentes pertenecientes a un programa académico a través de las clases `SistemaService`, que realiza petición a `AdministradorController` con el método `docentesdeunprograma` y

*AdministradorFunction* que ejecuta la lógica de la clase para devolver una respuesta. Esto permite precargar datos que utilizará la clase *AreaPage* para asociar el docente que dirigirá el departamento interno y los cursos que estarán a su cargo, todos estos datos son manipulados por las clases *AdministradorController* y *AdministradorFunction* que permitirán el almacenamiento del nuevo departamento interno. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 20. Se anexa en formato digital)

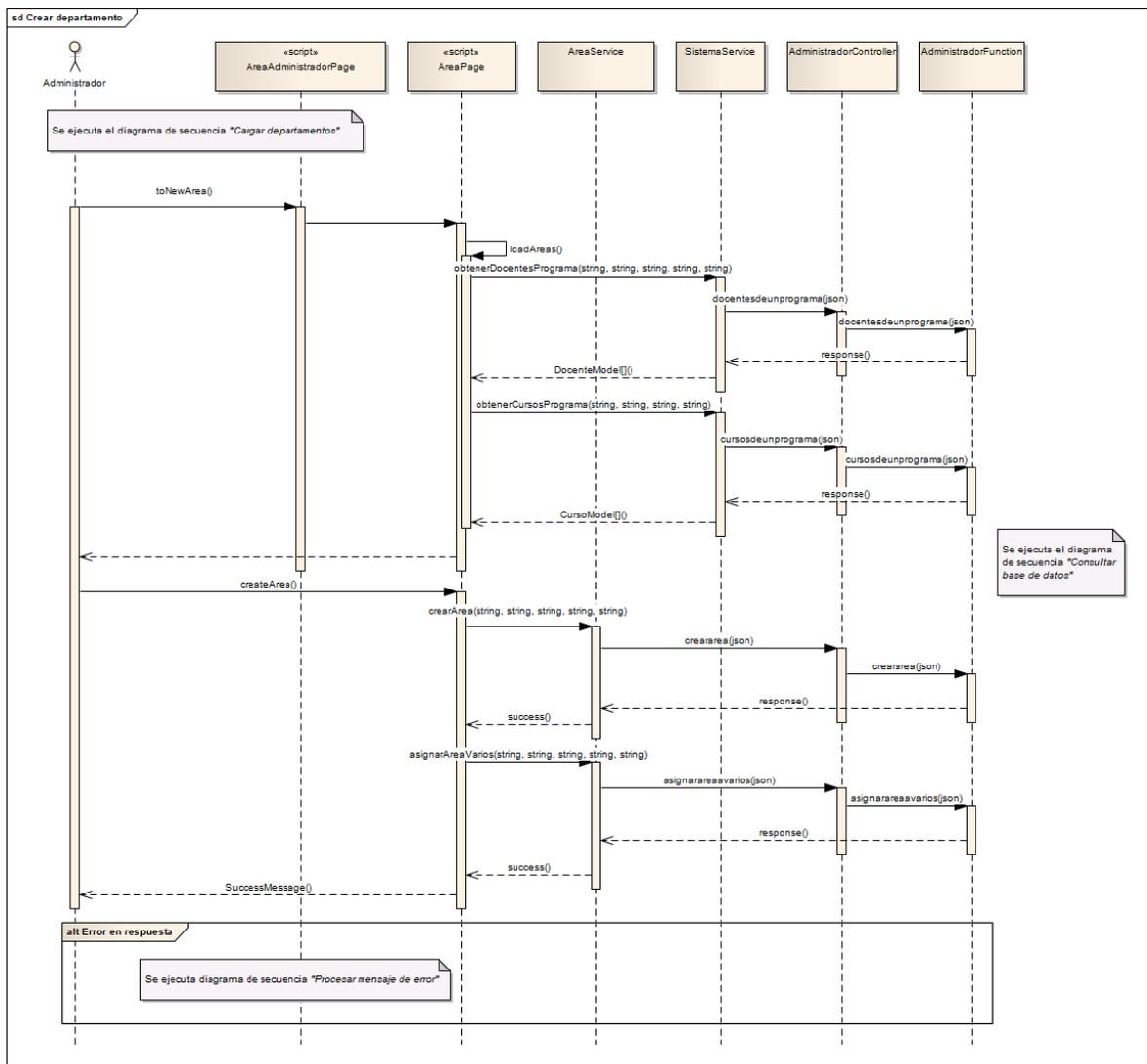


Ilustración 20 – Diagrama de secuencia del caso de uso: Crear departamento

#### 4.3.3.1.1.3 Editar departamentos

Para editar los departamentos internos de un programa lo que primero que debe suceder dentro del sistema además de la autenticación ante SIRECAA es la invocación al caso de uso Cargar departamentos que le permitirá visualizar los departamentos previamente creados, cuando el actor Administrador decide editar un departamento este debe obtener tanto las asignaturas y cursos como el docente que hacen parte del departamento interno que se quiere editar haciendo uso de los métodos *obtenerAsignaturasAsociadas*, *obtenerDocentesPrograma* y *obtenerCursosPrograma* a través de las clases *AreaService*, *SistemaService*, *AdministradorController* y *AdministradorFunction*, cuando toda esta información es proporcionada, la clase *AreaPage* ejecuta el método *editarArea* que permite la modificación del departamento interno en cuestión. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 21. Se anexa en formato digital)

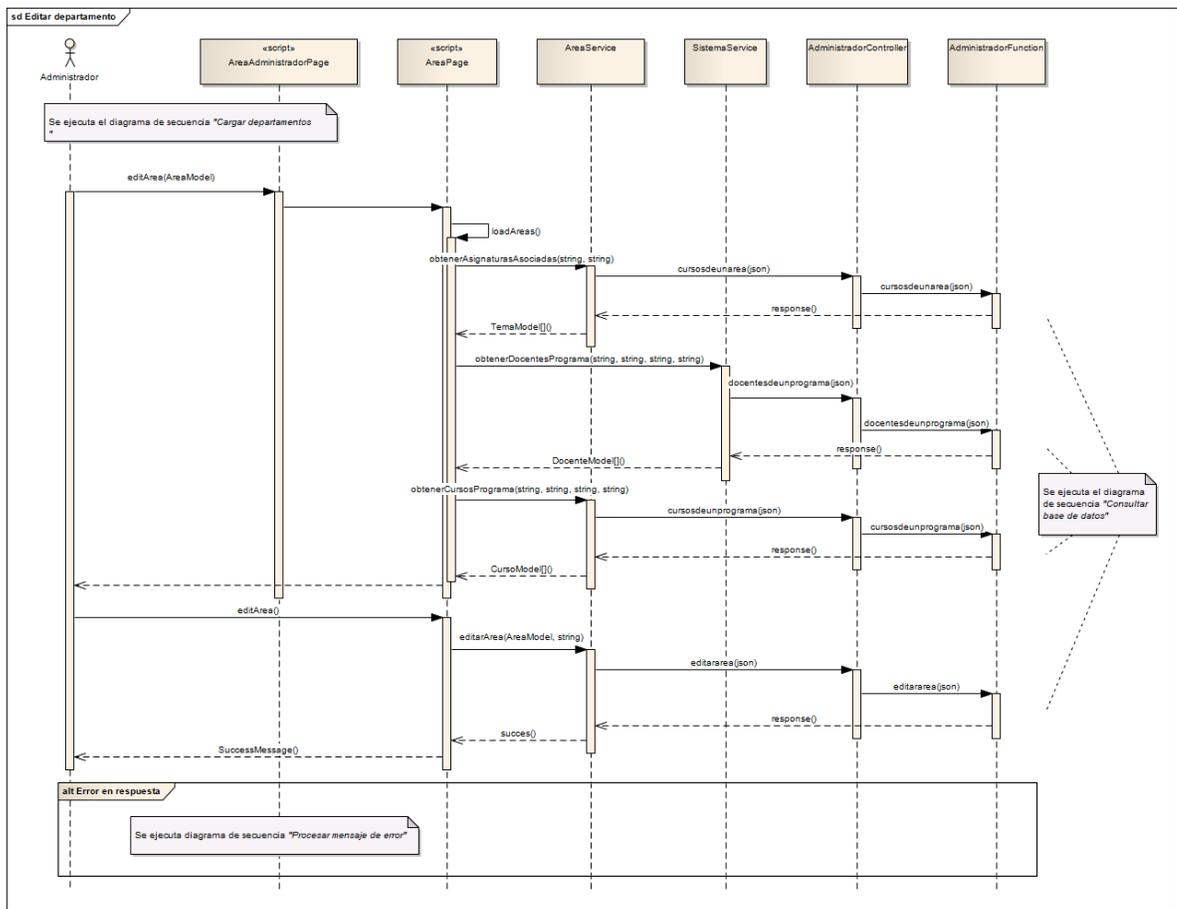


Ilustración 21 – Diagrama de secuencia del caso de uso: Editar departamento

#### 4.3.3.1.1.4 Eliminar departamento

Para la eliminación de un departamento el actor Administrador debe seleccionar un departamento, después que estos sean cargados por medio de la invocación del caso de uso Cargar departamentos. La clase *AreaAdministradorPage* ejecuta el método *eliminarArea* de la clase *AreaService* que se enlaza con los métodos *eliminarArea* de las clases *AdministradorController* y *AdministradorFunction*. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la

información solicitada por cada una de las clases que interactúan. (Ver Ilustración 22. Se anexa en formato digital)

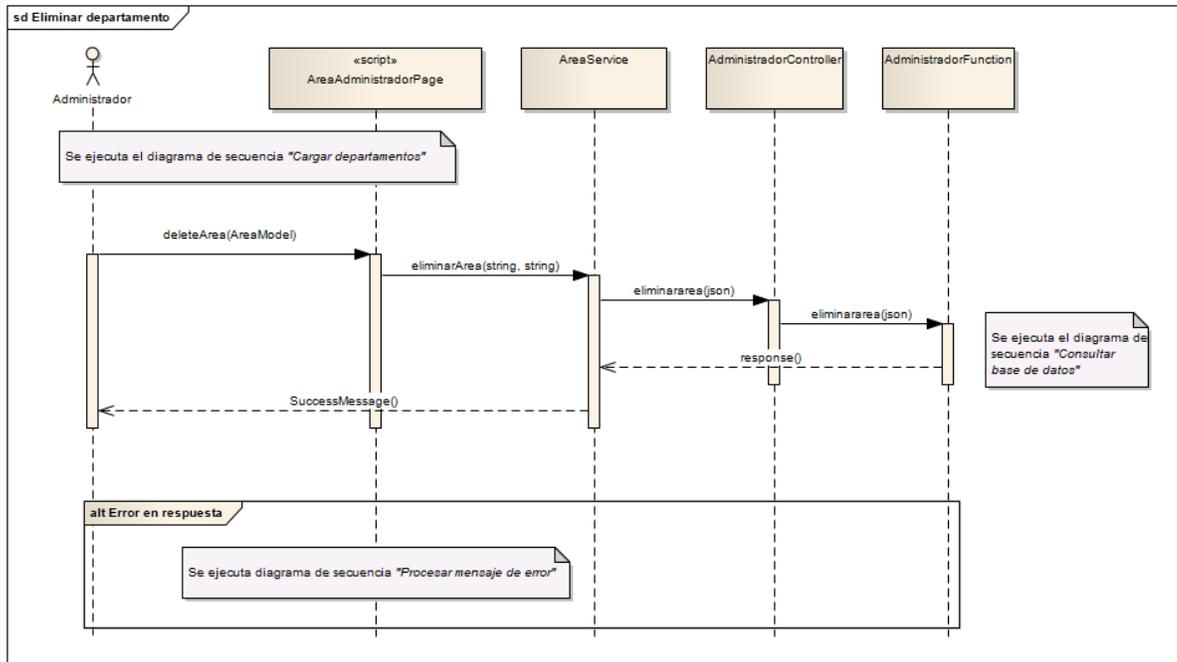


Ilustración 22 – Diagrama de secuencia del caso de uso: Eliminar departamento

#### 4.3.3.1.1.5 Visualizar departamento

La visualización de departamento le permite al actor Administrador observar cada uno de los departamentos internos que hacen parte del programa académico al que hace parte, para ello la clase *AreaAdministradorPage* ejecuta el método *viewArea* que permite por medio de las clases *AreaPage*, *AreaService*, *SistemaService*, *AdministradorController* y *AdministradorFunction* obtener los departamentos internos con las asignaturas y cursos asociados. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 23. Se anexa en formato digital)

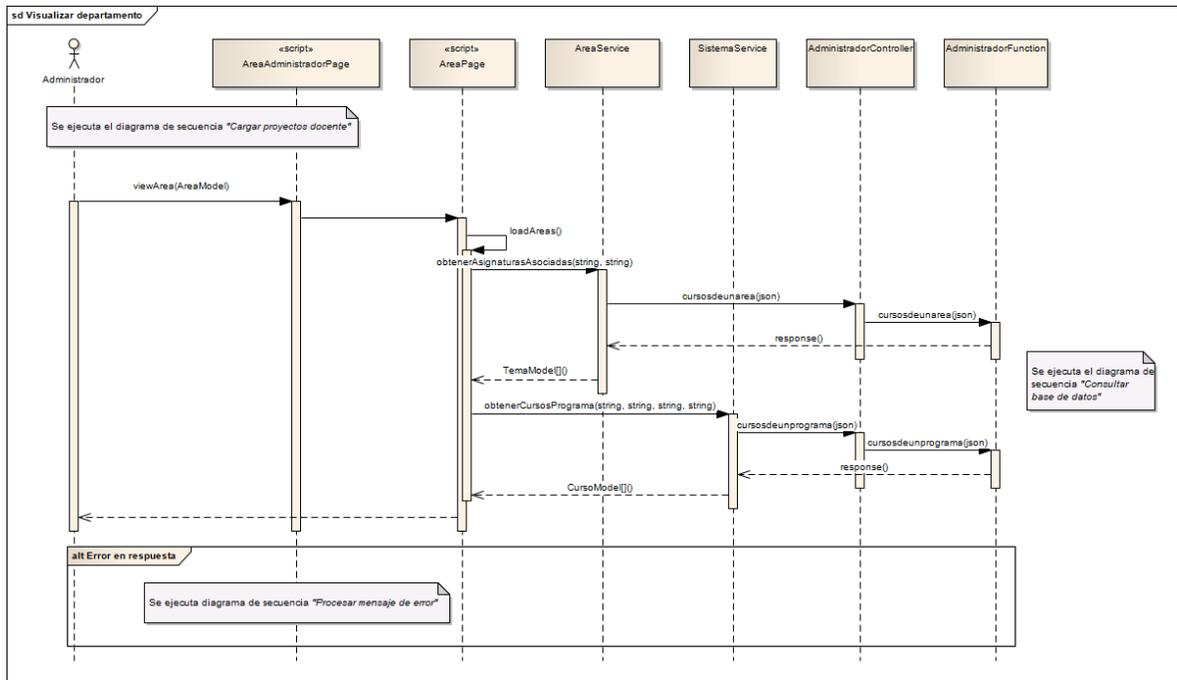


Ilustración 23 – Diagrama de secuencia del caso de uso: Visualizar departamento

#### 4.3.3.1.2 Caso de uso: Gestionar proyectos docente

La gestión de proyectos docente dentro de SIRECAA permite la creación, edición, eliminación y visualización de los diferentes proyectos docentes de los cursos de un departamento académico, así:

##### 4.3.3.1.2.1 Cargar Proyectos docente

La carga de proyectos docente permite obtener todos los proyectos docente que han sido registrados dentro de SIRECAA por los diferentes auditores de departamentos internos. Para que esta carga se pueda realizar el Administrador o el actor en cuestión debe estar autenticado ante el sistema; verificación realizada a través de la clase *AuthService*, después

de ello por medio del método *obtenerProyectosDocente* de la clase *ProyectoDocenteService* se permite la comunicación con las clases *AdministradorController* y *AdministradorFunction* que permiten por medio de los métodos definidos la recuperación de los distintos proyectos docentes almacenados. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 24. Se anexa en formato digital)

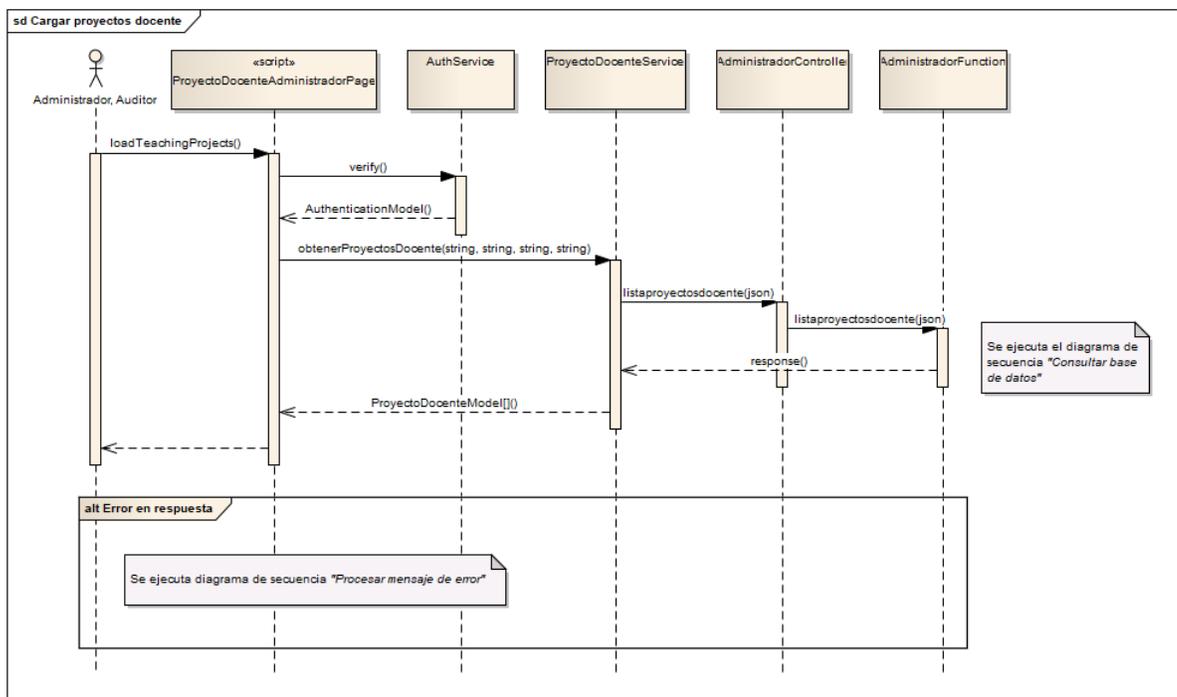


Ilustración 24 – Diagrama de secuencia del caso de uso: Cargar proyectos docente

#### 4.3.3.1.2.2 Crear proyecto docente

Para la creación de un proyecto docente el actor auditor podrá asociar distintos cursos al proyecto docente que se esté creando por ello lo primero que se hace dentro de SIRECAA

es cargar todos los cursos del departamento interno al que pertenece, después que este determina los cursos a lo que quiere asociar el proyecto se hace un llamado al método *crearProyectoDocente* de la clase *ProyectoDocentePage* que invoca al método *proyectodocente* de las clases *AdministradorController* y *AdministradorFunction*, la asociación de los cursos a un proyecto docente se realiza por medio del método *asignarProyectoDocenteVarios* de la clase *ProyectoDocenteService*. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 25. Se anexa en formato digital)

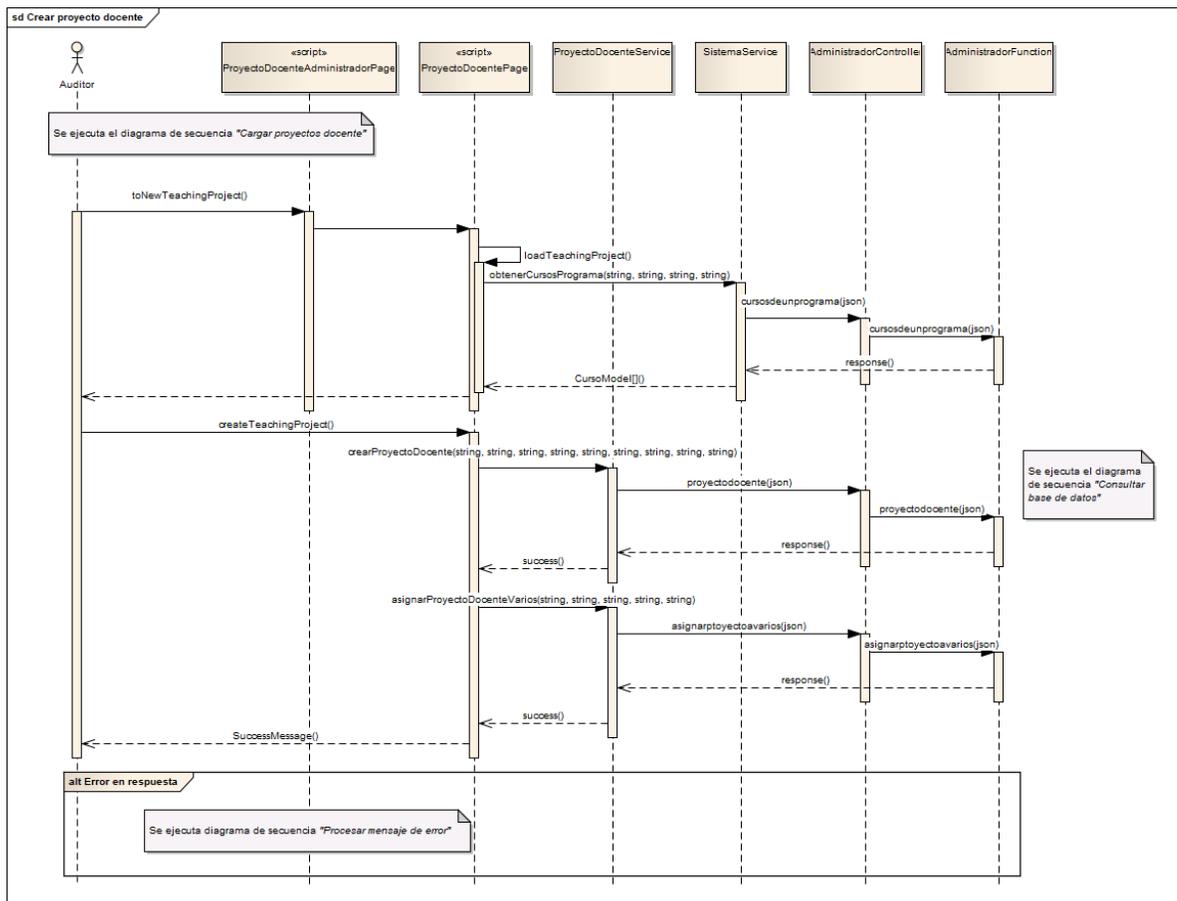


Ilustración 25 – Diagrama de secuencia del caso de uso: Crear proyecto docente

#### 4.3.3.1.2.3 Editar proyecto docente

Para la edición de un proyecto docente se deben cargar los proyectos docentes asociados al departamento interno al que pertenece el auditor. Primero el sistema muestra todos los temas del proyecto docente, sus asignaturas y cursos a través de los métodos *obtenerTemasProyectoDocente*, *obtenerAsignaturasAsociadas* y *obtenerCursosPrograma* de las clases *ProyectoDocenteService*, *ProyectoDocenteService* y *SistemaService* que se comunican con las clases backend *AdministraodorController* y *AdministradorFunction*. Después que toda la información pertinente es almacenada se procede a llamar al método *editarProyectoDocente* de la clase *ProyectoDocenteService*. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 26. Se anexa en formato digital)

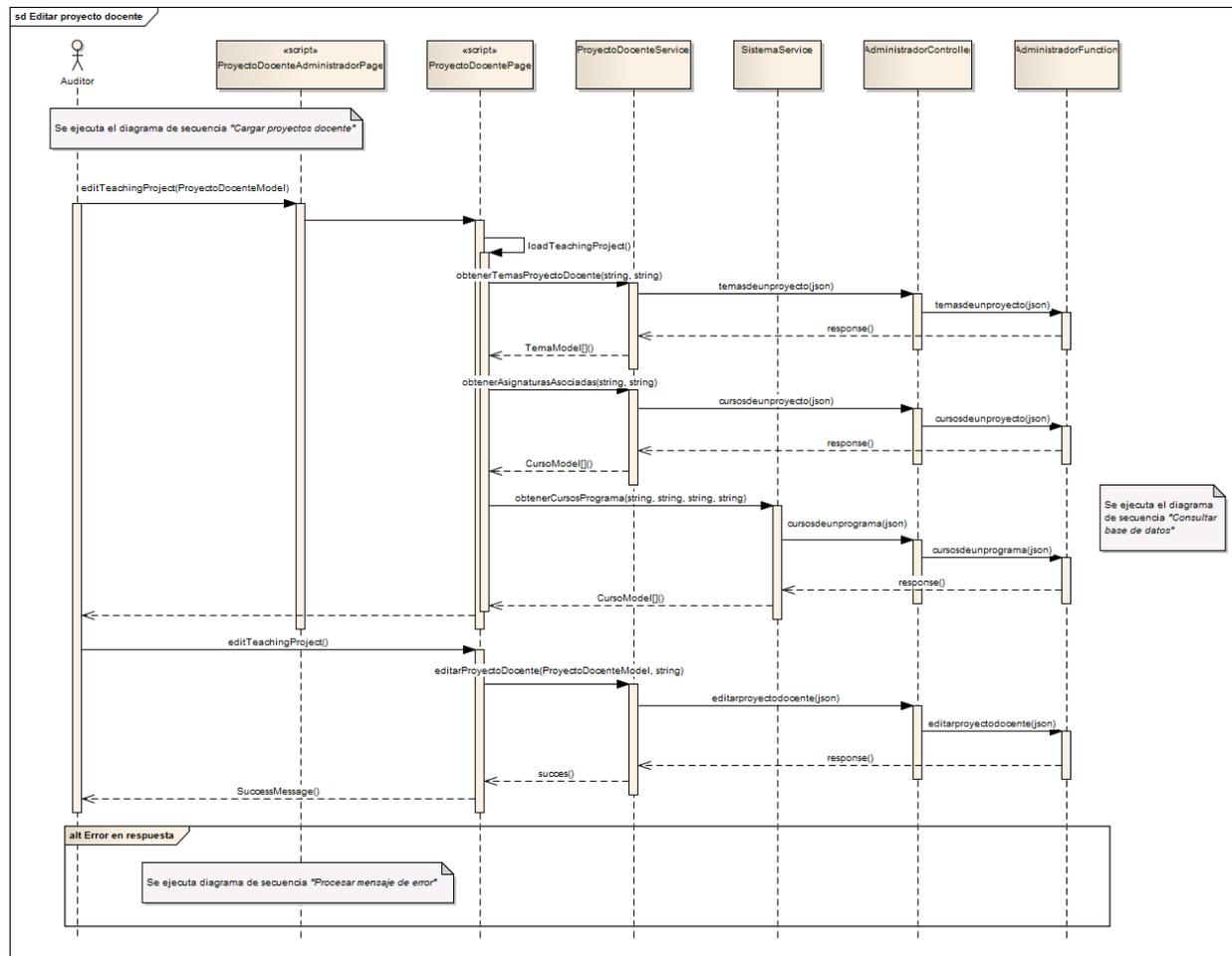


Ilustración 26 – Diagrama de secuencia del caso de uso: Editar proyecto docente

#### 4.3.3.1.2.4 Eliminar proyecto docente

Para la eliminación de un proyecto docente el sistema cargará todos los proyectos docentes asociados al departamento interno, después de ello seleccionará el que desea eliminar llamando al método *eliminarProyectoDocente* de la clase *ProyectoDocenteService* que se comunica con los métodos *eliminarproyectodocente* de las clases *AdministradorController* y *AdministradorFunction* que realizan las validaciones pertinentes para la eliminación. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 27. Se anexa en formato digital)

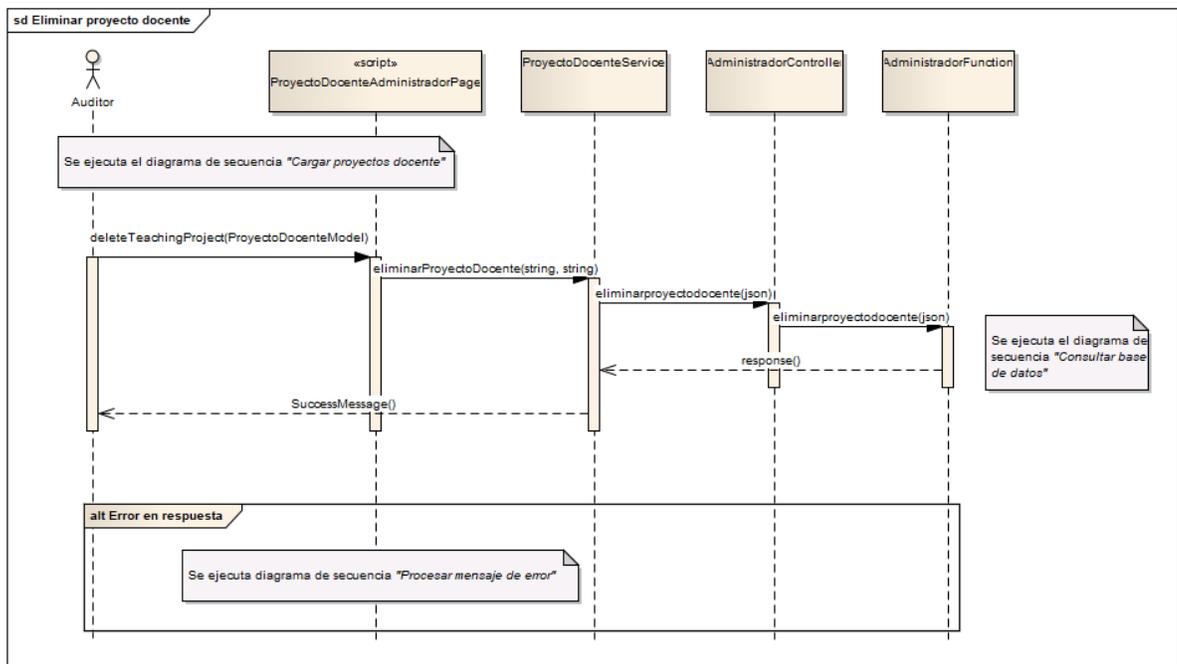


Ilustración 27 – Diagrama de secuencia del caso de uso: Eliminar proyecto docente

#### 4.3.3.1.2.5 Visualizar proyecto docente

Para la visualización de un proyecto docente el sistema cargará todos los proyectos docentes que se han creado para el departamento académico, cuando el actor Administrador o el actor en cuestión seleccione uno el sistema carga para él seleccionado los temas del proyecto por medio del método *obtenerTemasProyectoDocente* de la clase *ProyectoDocenteService* y carga los cursos que tiene asociados con el método *obtenerAsignaturasAsociadas* de la clase *ProyectoDocenteService* ambos datos son cargados con los métodos contenidos dentro de las clases *AdministradorController* y *AdministradorFunction*. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 28. Se anexa en formato digital)

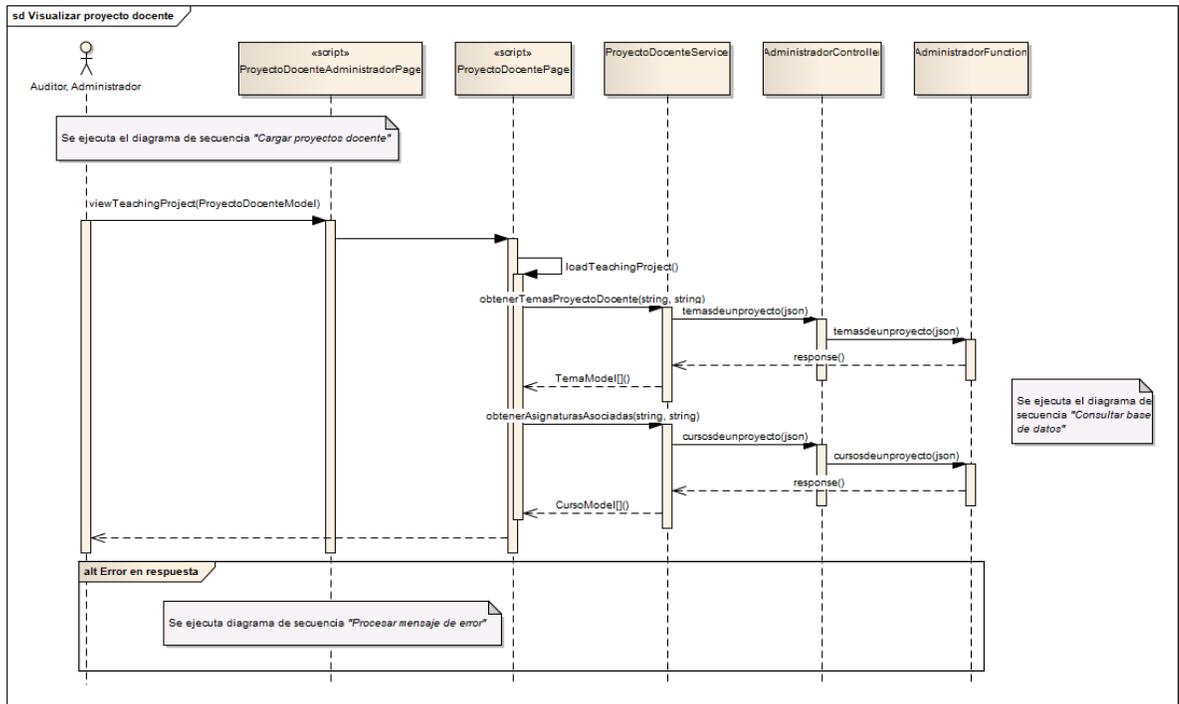


Ilustración 28 – Diagrama de secuencia del caso de uso: Visualizar proyecto docente

#### 4.3.3.1.1 Caso de uso: Realizar registros académicos

Para realizar los registros académicos el sistema debe poder visualizar los cursos que tiene asociado el actor Docente, después que este seleccione el curso de su interés en el formulario de registro debe seleccionar los estudiantes que asistieron a la clase, dicha información es visualizada a través del método *obtenerEstudiantes* de la clase *EstudianteService* que busca con el método *estudiantesMatriculados* del servicio provisto por la Universidad de Cartagena que estudiantes están matriculados. Antes de que el registro pueda ser almacenado dentro del sistema se debe validar por medio del método *validaRegistros* de la clase *RegistroService* que el registro se encuentre dentro del tiempo estipulado para poder ser diligenciado, el tiempo por defecto para esta acción de es máximo 48 horas después de que la clase ha iniciado. Después que toda la información del registro es diligenciada con el método *CompleteRecord* de la clase *RegistroDocentePage* se invoca al método *crearRegistro* de la clase *RegistroService* que se comunica con los métodos *validaregistros* de las clases *DocenteController* y *DocenteFunction* para su posterior almacenamiento. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 29. Se anexa en formato digital)

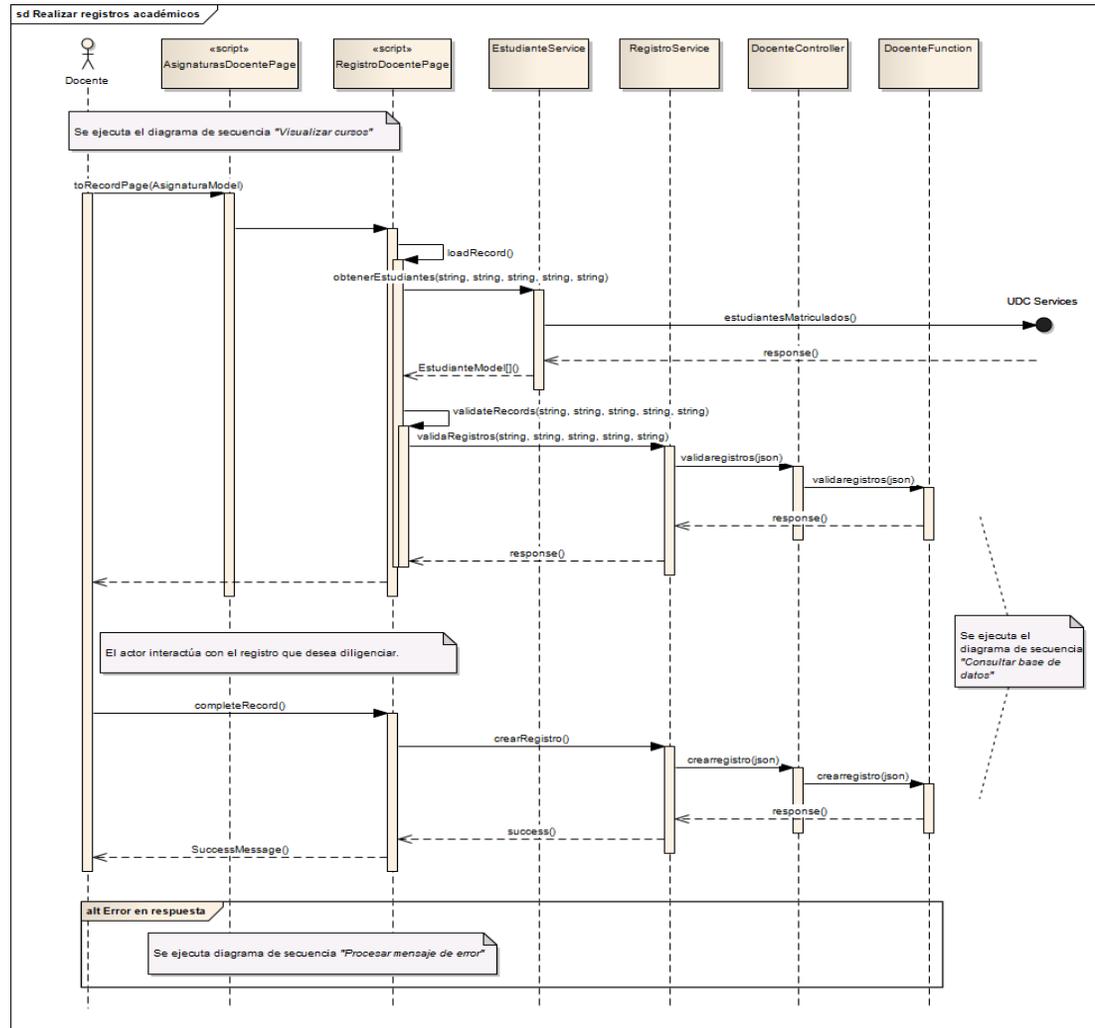


Ilustración 29 – Diagrama de secuencia del caso de uso: Realizar registro

#### 4.3.3.1.2 Caso de uso: Registrar comentarios sobre registros académicos

El actor estudiante cada vez que este lo desee puede registrar comentario u observaciones a los registros académicos de los cursos a los que pertenezca, para ello cuando este acceda a SIRECAA lo primero que debe visualizar con los cursos que tiene asociados, después que seleccione el de su preferencia se le listaran todos los registros académicos que el docente ha diligenciado, cuando seleccione el de su interés el sistema le permitirá crear un comentario por medio de la invocación del método *crearComentario* de la clase *EstudianteService*, que invoca a su vez a los métodos *crearcomentario* de las clases *EstudianteController* y *EstudianteFunction* que permiten el almacenamiento del mensaje diligenciado por el actor Estudiante. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 30. Se anexa en formato digital)

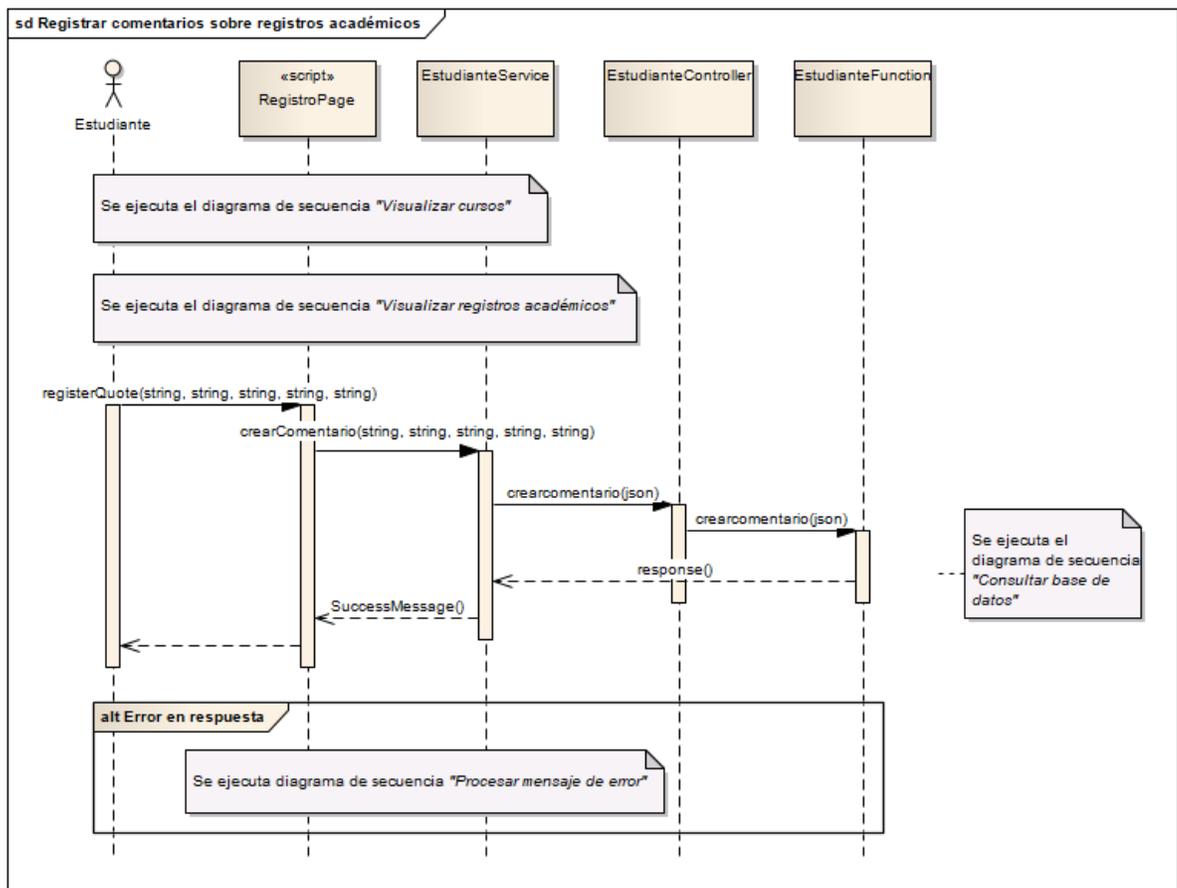


Ilustración 30 – Diagrama de secuencia para el caso de uso: Registrar comentarios sobre registros académicos

#### 4.3.3.1.3 Caso de uso: Visualizar cursos

Para la visualización de los cursos pertenecientes a un programa académico, se hace uso de la API provista por la Universidad de Cartagena, que dado ciertos parámetros específicos permite obtener la información deseada. Para ello debe existir una autenticación ante el sistema por medio de la clase *AuthService*, además se debe proporcionar tanto el periodo de los cursos a consultar y el usuario que realiza la solicitud con el fin de corroborar si tiene los permisos necesarios para visualizar, con esa información se ejecuta el método *obtenerAsignaturasActor* de la API que se almacenan dentro de la clase *AsignaturasporActorPage* que posteriormente permiten extraer la información almacenada

de los cursos para su visualización. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 31. Se anexa en formato digital)

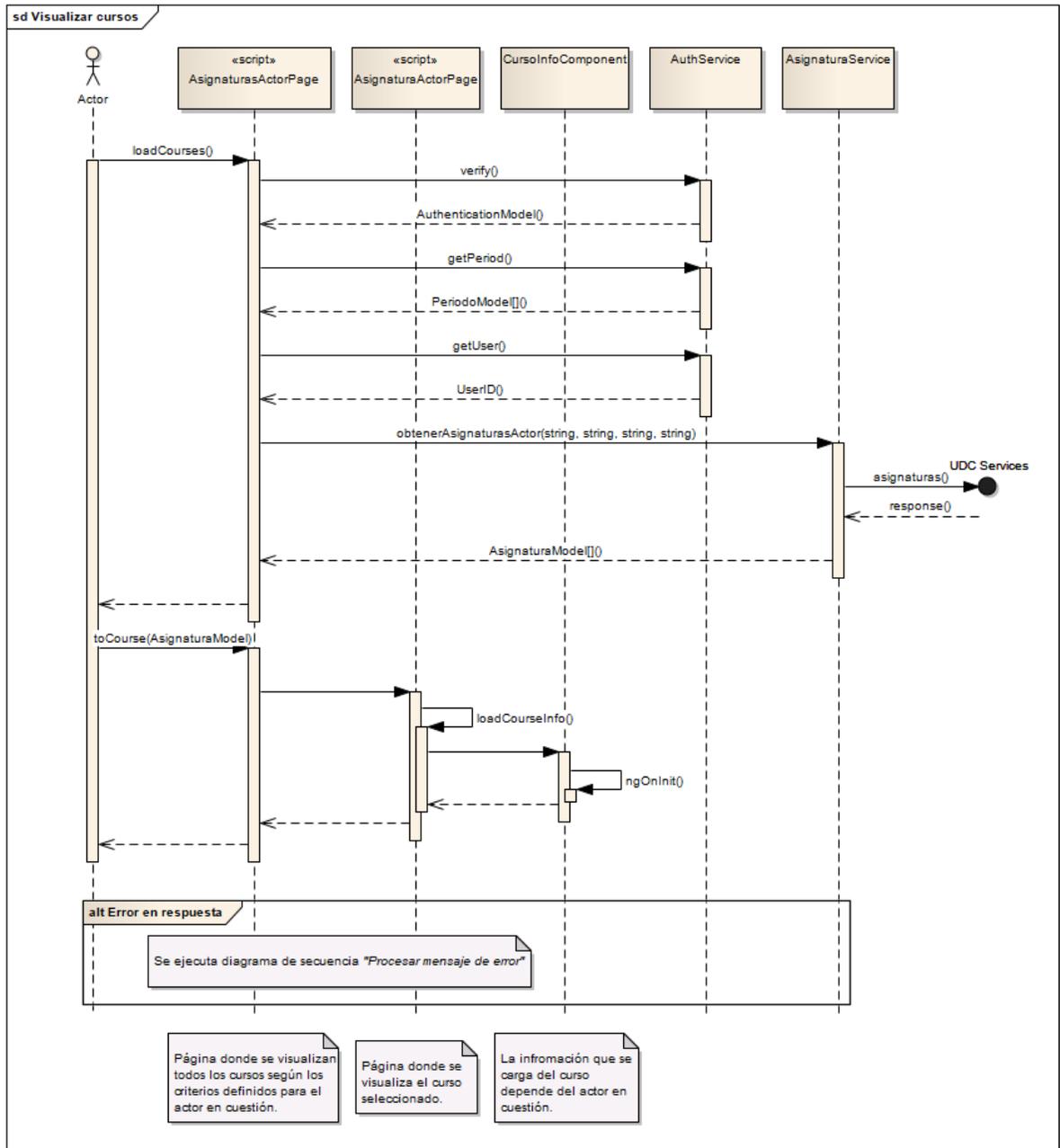


Ilustración 31 – Diagrama de secuencia para el caso de uso: Visualizar cursos

#### 4.3.3.1.4 Caso de uso: Visualizar estadísticas de cursos y registros

El actor Auditor podrá visualizar un conjunto de estadísticas que le ayudaran en la toma de decisiones de su labora para ello cuando el Auditor decida visualizarlas el sistema por medio de la clase *EstadisticasPage* que ejecuta el método *obtenerData* de la clase *AuditorService* que dependiendo de la estadística seleccionada ejecuta la lógica correspondiente en las clases *AuditorController* y *AuditorFunction*, que permite devolver información que necesitan de insumo los métodos *calculateStatistics*, *runStatisticsEngine* y *showGraphics*, que permiten la visualización de los gráficos estadísticos. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 32. Se anexa en formato digital)

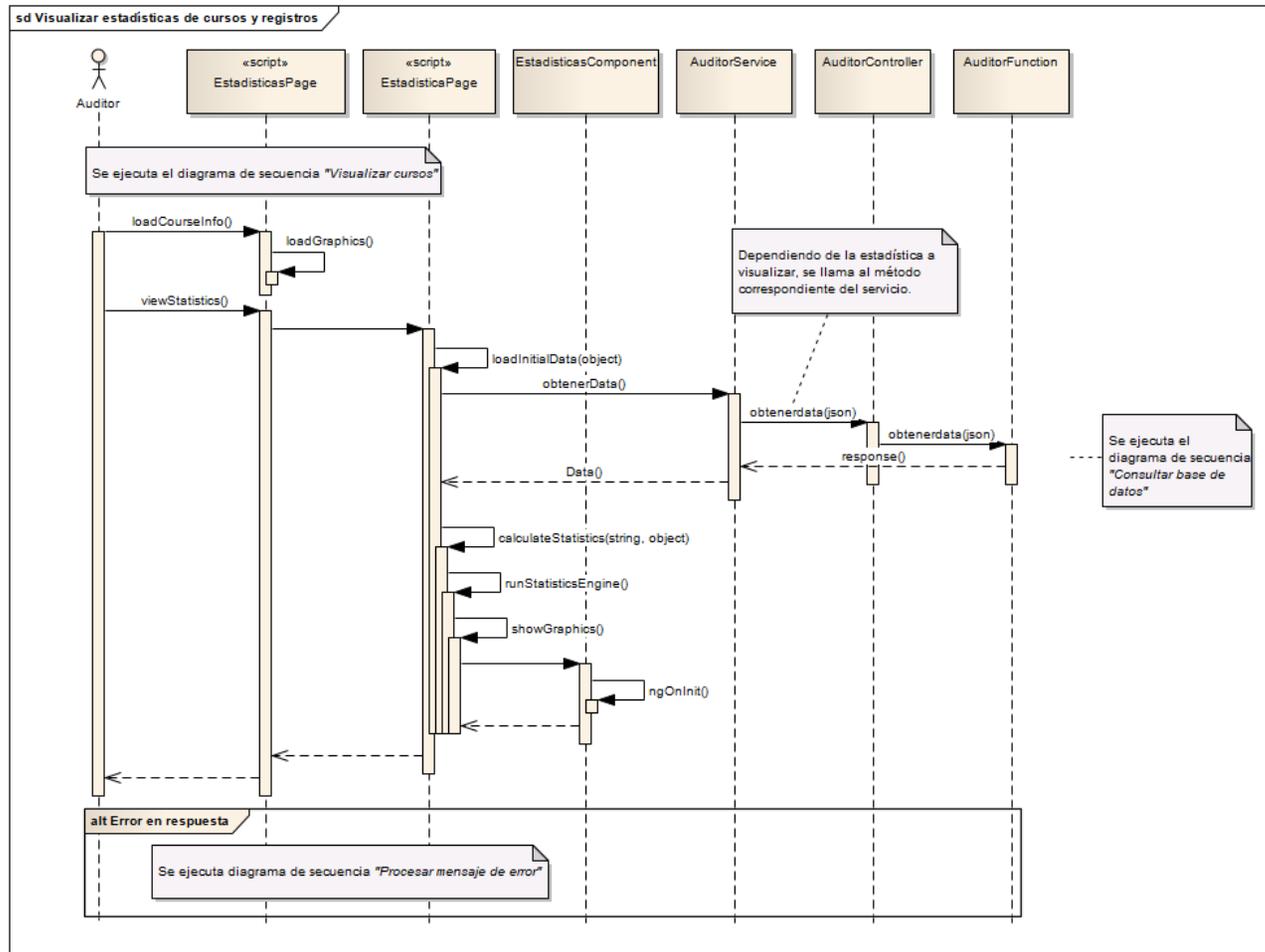


Ilustración 32 – Diagrama de secuencia del caso de uso: Visualizar estadísticas de cursos y registros

#### 4.3.3.1.5 Caso de uso: Visualizar registros académicos

Para la visualización de los registros académicos el usuario de la aplicación debe seleccionar un curso y aparecerán los registros asociados. Para ello la entidad *RegistrosActorPage* invoca al método *obtenerRegistros* que se comunican con los métodos *obtenerregistros* de las clases *SistemaController* y *DocenteFunction*, toda esta información es cargada en la clase *RegistrosActorPage* donde se podrán visualizar tanto las asistencias registradas a través del método *obtenerasistencia* de las clases *SistemaController* y *DocenteFunction* como los temas vistos; obtenidos a través del método *obtenerTemasVistos* de clase *SistemaService* que se comunica con los métodos *obtenertemasvistos* de las clases *SistemaController* y *DocenteFunction*, además por medio de estas clases también se pueden visualizar los comentarios diligenciados por los estudiantes. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 33. Se anexa en formato digital)

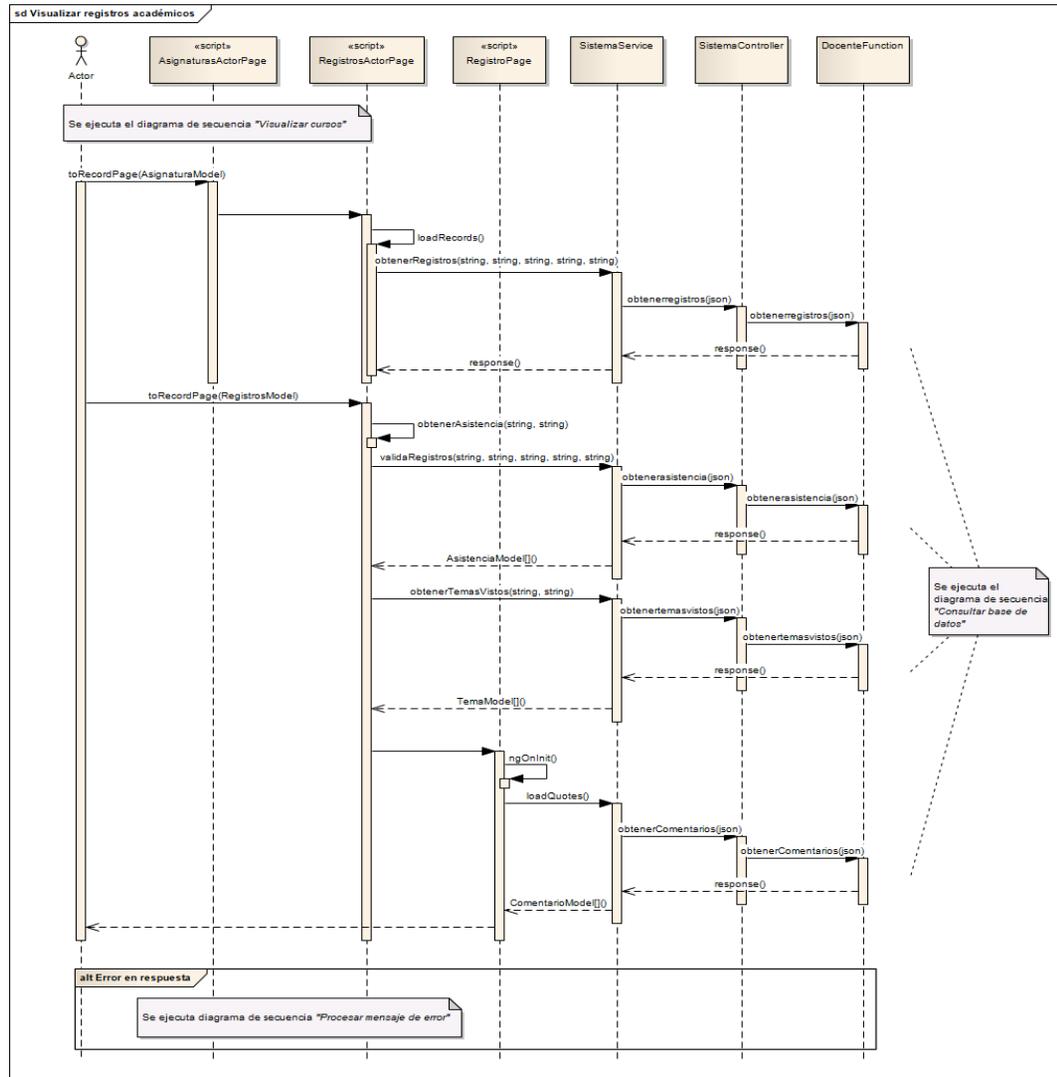


Ilustración 33 – Diagrama de secuencia del caso de uso: Visualizar registros académicos

#### 4.3.3.1.6 Caso de uso genérico: Consultar base de datos

Todos los usuarios de la aplicación al tener interacción con ella realizan peticiones o llamados a la base de datos, que están representados por el establecimiento de una conexión que permite la extracción por medio de una consulta de información utilizada como insumo para SIRECAA. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 34. Se anexa en formato digital)

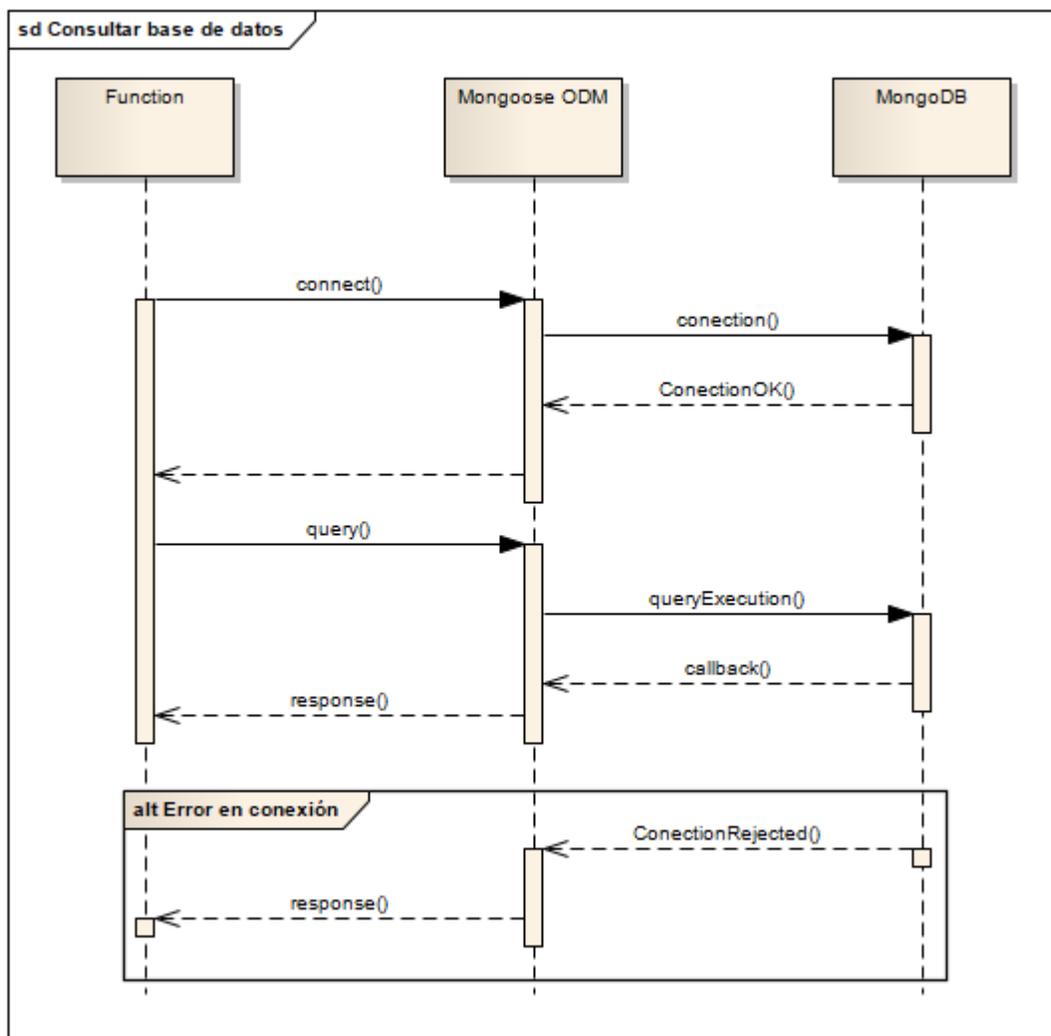


Ilustración 34 – Diagrama de secuencia para el caso de uso genérico: Consultar base de datos

#### 4.3.3.1.7 Caso de uso genérico: Iniciar sesión

Todos los actores del sistema para poder interactuar con las funcionalidades provistas por la aplicación deben pasar por un proceso de autenticación y autorización ante el sistema, el cual recibe datos de acceso que son diligenciados y que son validados por la clase *AuthService* que se comunica con el método *signin* de la clase *Signin Backend Service* que se conecta con la API provista por la Universidad de Cartagena para devolver la autorización en caso exitoso, que permite el uso de las funcionalidades de SIRECAA. Además, existe una sección encargada del manejo de errores que se pueden presentar tanto al enviar, recibir u procesar la información solicitada por cada una de las clases que interactúan. (Ver Ilustración 35. Se anexa en formato digital)

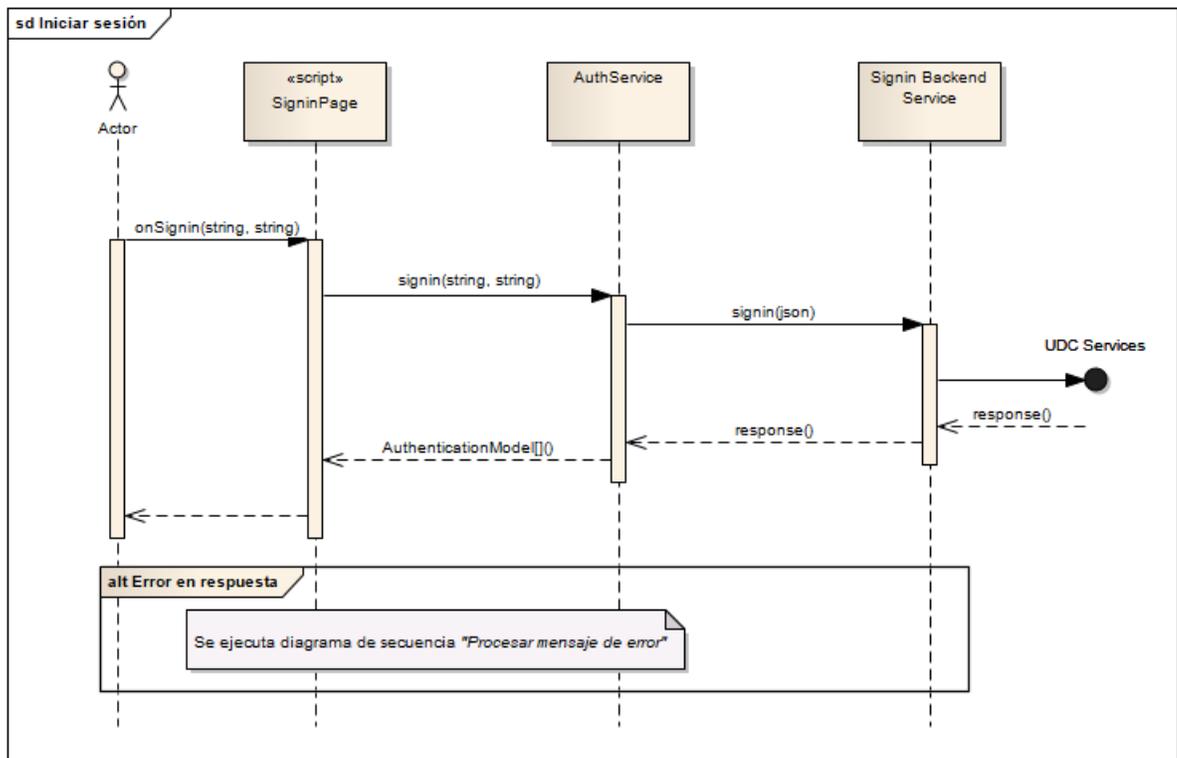


Ilustración 35 – Diagrama de secuencia del caso de uso genérico: Iniciar sesión

#### 4.3.3.1.8 Caso de uso: Procesar mensaje de error

El procesamiento de mensajes de error corresponde a la forma en la que se da la transmisión de errores entre los servicios Back-end con los elementos Pages del Front-end que lo consumen. Primero los servicios Back-end devuelven mensajes de error que son capturados por los Services del Front-end que envían a los Pages Front-end dicha información para ser desplegada a los usuarios de SIRECAA. (Ver Ilustración 36. Se anexa en formato digital)

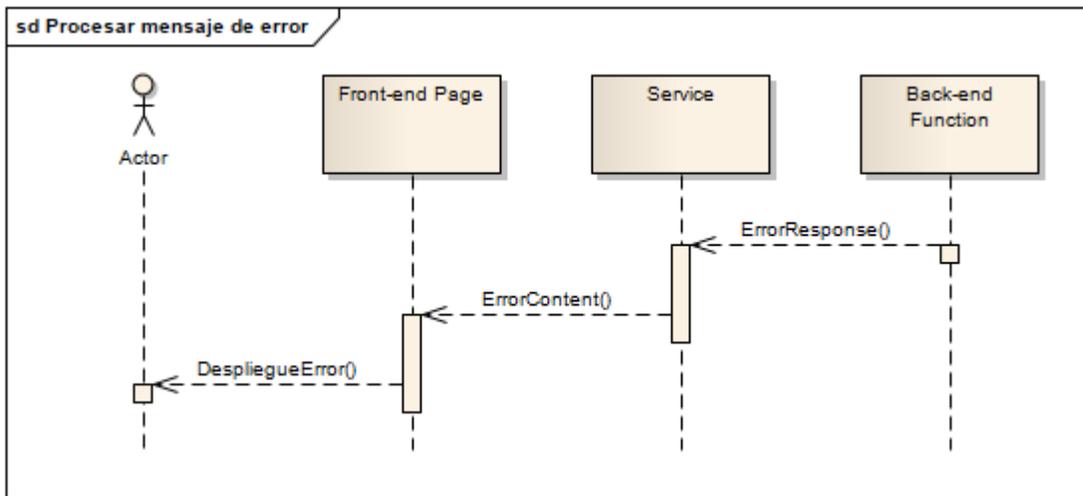


Ilustración 36 – Diagrama de secuencia del caso de uso genérico: Procesar mensaje de error

#### 4.3.4 Vista de desarrollo

Esta vista es la encargada de mostrar a nivel de paquetes las relaciones entre las diferentes clases y su interacción, se encuentra representada por el Diagrama de paquetes.

#### 4.3.4.1 Diagrama de paquetes

Este diagrama ilustra los diferentes componentes representados como paquetes que construyen la solución SIRECAA, pasando por todos los elementos Front-end, Back-end, hasta la utilización de la API provista por la Universidad. (Ver Ilustración 37. Se anexa en formato digital)

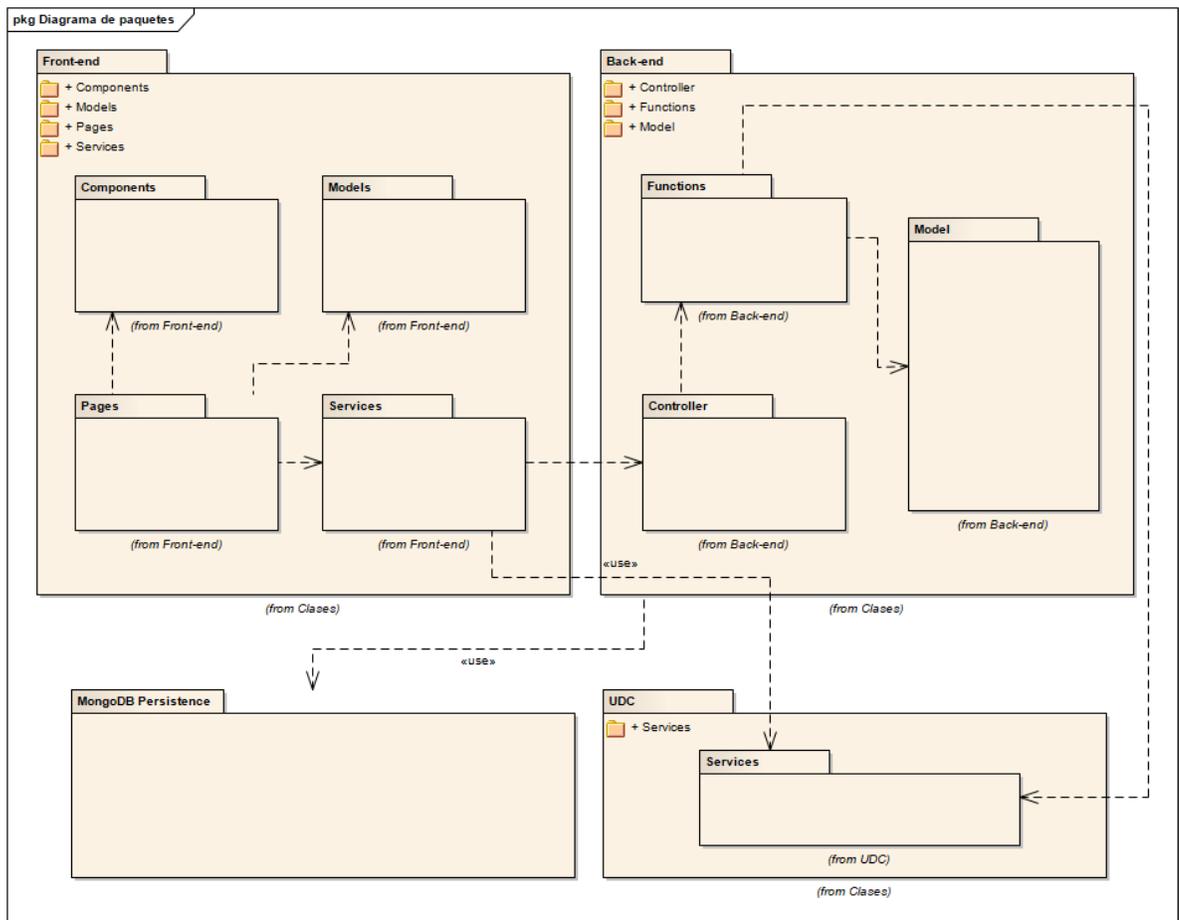


Ilustración 37 – Diagrama de paquetes

### **4.3.5 Vista física**

Esta vista es la encargada de mostrar como es el despliegue de la aplicación desde nodos físicos que almacenan nuestra solución. Esta vista se encuentra representada por el diagrama de despliegue.

#### ***4.3.5.1 Diagrama de despliegue***

Este diagrama ilustra los diferentes nodos físicos donde es almacenada SIRECAA, debido a que implementó una solución móvil y web se tienen dos nodos físicos principales, los cuales son un PC donde se puede acceder a la solución web y un SmartPhone donde se puede acceder a la solución móvil. (Ver Ilustración 38. Se anexa en formato digital)

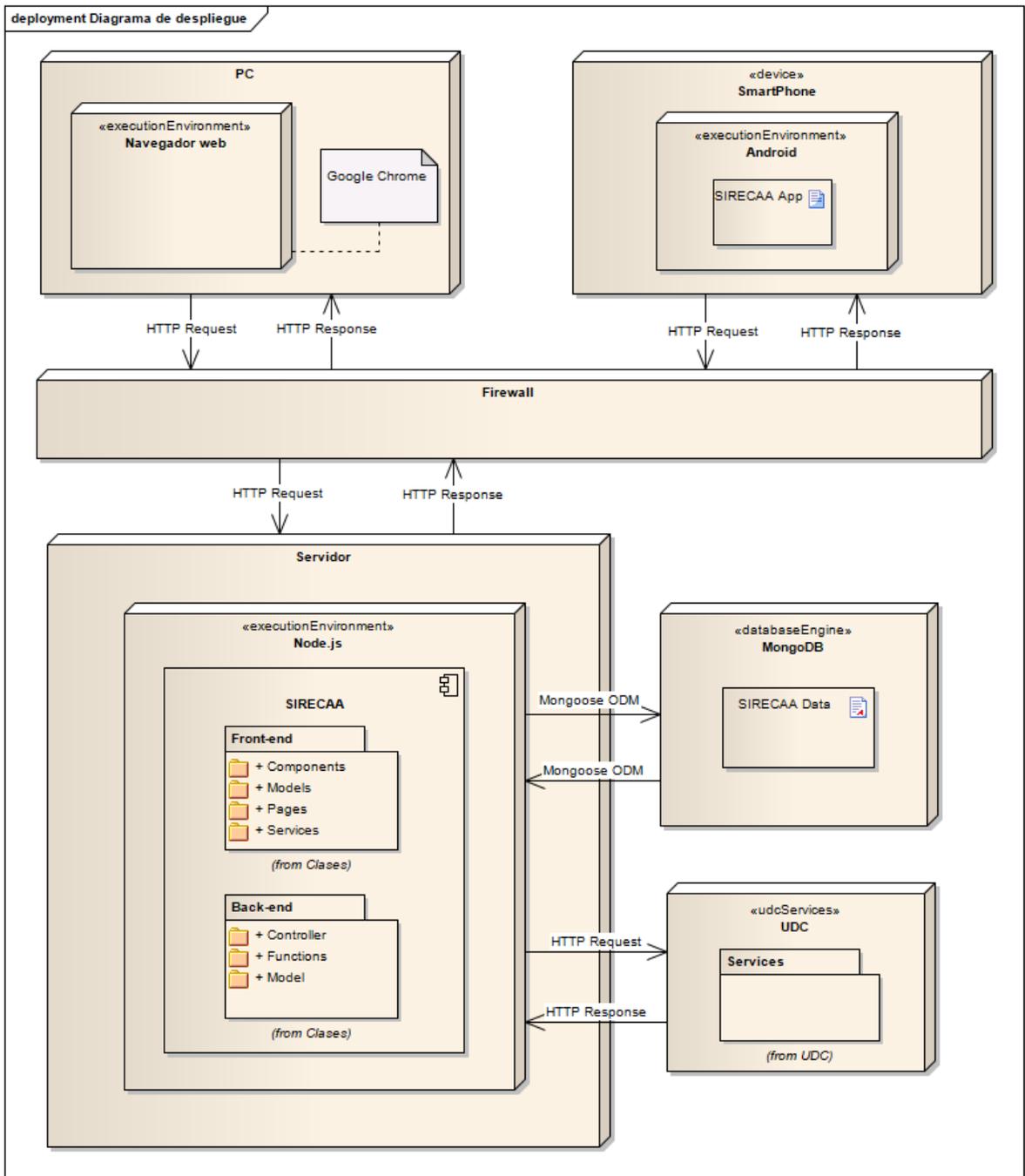


Ilustración 38 – Diagrama de despliegue

## **4.4 PRUEBAS**

Con el objetivo de verificar el correcto funcionamiento de la plataforma SIRECAA, se realizaron pruebas de caja negra. Para la realización de las pruebas se tuvo en cuenta la ejecución de un flujo de trabajo dividido en dos partes fundamentales: el diseño de las pruebas y la ejecución de las mismas.

Tal como lo expresa Pressman, Roger S. (Ingeniería del software, Un enfoque práctico, 2010), a través de las pruebas de caja negra se buscó encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación.

En esta sección se presentan los diseños y los resultados de las pruebas ya mencionadas.

### **4.4.1 Prueba de caja negra**

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa (Pressman, 2010).

Para realizar las pruebas de caja negra de SIRECAA se hizo uso de un diseño basado en una estructura que responde a las preguntas propuestas por Roger Pressman, las cuales son:

- ¿Cómo se prueba la validez funcional?
- ¿Cómo se prueban el comportamiento y el rendimiento del sistema?
- ¿Qué clases de entrada harán buenos casos de prueba?
- ¿El sistema es particularmente sensible a ciertos valores de entrada?
- ¿Cómo se aíslan las fronteras de una clase de datos?

- ¿Qué tasas y volumen de datos puede tolerar el sistema?
- ¿Qué efecto tendrán sobre la operación del sistema algunas combinaciones específicas de datos?

En el capítulo de anexos, en el anexo AM, se encuentra adjunto el archivo en formato de Excel con el diseño implementado para la realización de estas pruebas.

#### **4.4.2 Validación del sistema**

Con el objetivo de validar el funcionamiento del sistema, ejecutando las pruebas de caja negra, se hizo entrega de los diseños de cada prueba a cada uno de los actores implicados: Docente, Estudiante, Jefe de departamento y Auditor. Lo anterior se hizo para que cada uno de ellos hiciera uso de la plataforma teniendo en cuenta el diseño de los casos funcionales de prueba, los casos de uso definidos en el SRS y las limitantes dispuestas en el alcance del proyecto. Para los actores del sistema se llevaron a cabo varias etapas de prueba, y para cada una se diligenciaron todos los hallazgos encontrados y las recomendaciones realizadas, considerando el actor y el orden de prioridad de ejecución de las pruebas (esto debido a que unos actores son dependientes de otros para que el sistema pueda operar correctamente).

##### **4.4.2.1 Administrador**

Este actor se encarga de las configuraciones generales de la plataforma tal como inicio del periodo académico, creación de auditores generales, gestión de departamentos internos, gestión de proyectos docentes, visualización de cursos y asignación de departamentos y proyectos. Este rol es desempeñado por el Jefe de departamento del programa académico. Esta prueba fue realizada por la Jefe de departamento del programa de Ingeniería de Sistemas, la ingeniera Narlinda Espinoza Cantillo.

Durante la primera etapa de pruebas no se presentaron problemas al momento de probar los flujos especificados dentro de los casos de prueba para el actor, por lo que, en términos generales, el sistema se comportó acorde y sin contratiempos dadas las funciones principales. Sin embargo, se realizaron sugerencias con respecto a ciertas funcionalidades que el sistema no presentaba (ver resultados en el anexo AN) pero que son de suma importancia en el proceso de registro y control de asistencia académica. Las siguientes sugerencias funcionales fueron realizadas:

1. Permitir la eliminación de departamentos internos más fácilmente.
2. Permitir la eliminación de auditores generales.

Como respuesta a las sugerencias en cuestión, el equipo de desarrollo del proyecto estudió el impacto de las nuevas funcionalidades requeridas y el tiempo que tomaría su desarrollo, llegando a la conclusión de un impacto bajo a nivel de desarrollo y de su alto grado de significancia para el proceso. Las soluciones abordadas por el equipo fueron:

1. Implementar una solución en la capa lógica (Back-end) para evitar validaciones innecesarias al momento de eliminar un departamento interno, manteniendo la integridad referencial de la base de datos.
2. Implementar una solución en la capa de presentación (Front-end) y en la capa lógica (Back-end) para permitir la eliminación de auditores generales dentro del sistema.

Una vez implementadas las soluciones, durante la segunda etapa de pruebas todo funcionó correctamente, cumpliendo así el cuarto objetivo del presente informe en relación a lo especificado por el usuario Administrador en el documento SRS y el producto entregado.

#### ***4.4.2.2 Docente***

Este actor se encarga de la realización de los registros de asistencia de las clases que dicta. Este rol es desempeñado por los docentes pertenecientes a uno o varios programas

académicos. Esta prueba fue realizada por el docente de planta del programa de Ingeniería de Sistemas, el ingeniero Martín Monroy Ríos.

Durante la primera etapa de pruebas no se presentaron problemas al momento de probar los flujos especificados dentro de los casos de prueba para el actor, por lo que, en términos generales, el sistema se comportó acorde y sin contratiempos dadas las funciones principales. El docente no tuvo problemas con ninguno de los casos funcionales establecidos en el diseño de pruebas, debido a que pudo comprobar correctamente la visualización de los cursos dictados, la realización del registro de asistencia académica, teniendo en cuenta las validaciones pertinentes realizadas por el sistema, permitiendo la selección de los temas dictados de acuerdo al proyecto docente asociado al curso, la selección de los estudiantes que asistieron o no y el diligenciamiento de observaciones generales con respecto al desarrollo de la clase.

Sin embargo, se realizaron sugerencias con respecto a ciertos aspectos de forma que el sistema no presentaba (ver resultados en el anexo AO) pero que son de suma importancia para que el sistema cuente con buena experiencia de usuario (capacidad de uso). Las siguientes sugerencias fueron realizadas a nivel de interfaz gráfica de usuario:

1. Cambiar el tiempo de mensaje mostrado después de que el usuario intenta autenticarse sin éxito.
2. Añadir botón de cerrar para las ventanas de proyecto docente y departamento.
3. Ordenar alfabéticamente por apellido la lista de estudiantes visualizada al momento de diligenciar el registro de asistencia.
4. Cambiar el color del texto utilizado para indicar que un estudiante no asistió.
5. Cambiar el tiempo de visualización y los colores de los mensajes emergentes mostrados.

Como respuesta a las sugerencias en cuestión, el equipo de desarrollo del proyecto estudió el impacto de las mejoras sugeridas y el tiempo que tomaría su desarrollo, llegando a la conclusión de un impacto bajo a nivel de desarrollo y de su alto grado de significancia para enriquecer la capacidad de uso. Las soluciones abordadas por el equipo fueron:

1. Implementar una solución en la capa de presentación (Front-end) para incrementar el tiempo de visualización del mensaje emergente dado el evento de credenciales no válidas o fallo en el inicio de sesión. Se hizo uso de una configuración del tiempo de visualización del componente genérico de mensajes, con 5 segundos de duración.
2. Implementar una solución en la capa de presentación (Front-end) para añadir el botón de cerrado de ventana en la solución web de la plataforma. Se hizo uso de una configuración para incluir el botón en los sistemas de tipo core (navegadores web) del componente genérico de presentación de proyectos docente y departamentos.
3. Implementar una solución en la capa de presentación (Front-end) para ordenar la lista de estudiantes y de asistencia de manera alfabética a partir de los apellidos de cada estudiante. Se hizo uso de un método público de la clase utilitaria, que consiste en un algoritmo de ordenamiento ascendente o descendente de listas, dentro del componente genérico de lista de estudiantes.
4. Implementar una solución en la capa de presentación (Front-end) para cambiar los estilos del texto mostrado en la lista de asistencia. Se hizo uso de una clase en la hoja de estilos en cascada del componente genérico de lista de estudiantes.
5. Implementar una solución en la capa de presentación (Front-end) para incrementar el tiempo de visualización de los mensajes emergentes del sistema y cambiar los estilos del componente. Se hizo uso de una configuración del tiempo de visualización del componente genérico de mensajes, con 5 segundos de duración, además de hacer uso de varias clases personalizadas en la hoja de estilos en cascada del componente general de la aplicación.

Una vez implementadas las soluciones, durante la segunda etapa de pruebas todo funcionó correctamente, teniendo en cuenta que se agregaron algunas mejoras de interfaz y algunas funcionalidades para agregar valor al actor, las cuales se listan a continuación:

1. Funcionalidad de visualización de estadísticas de los cursos dirigidos por el docente.
2. Cambio de título de “Cursos dictados” a “Cursos dirigidos”.
3. Opción de cerrado de ventanas y modales alineada a la derecha.
4. Omisión de mensajes de información redundantes.

De esta manera se cumplió con el cuarto objetivo del presente informe en relación a lo especificado en el documento SRS por el usuario Docente y el producto entregado.

#### **4.4.2.3 Estudiante**

Este actor se encarga de visualizar los registros de asistencia académica de los cursos a los que pertenece, estando en la capacidad de realizar comentarios y de calificar las clases que ha recibido. Esta prueba fue realizada por los estudiantes del curso Proyecto de Grado, el cual es dirigido por el profesor Martín Monroy Ríos.

Durante la primera etapa de pruebas los flujos principales estipulados en el diseño de pruebas para los estudiantes no presentaron ningún problema, esto debido a que los estudiantes estuvieron en la capacidad de visualizar al instante todos los registros diligenciados por el profesor Martín, además que pudieron realizar comentarios y observaciones referentes a las clases impartidas, por último, pudieron dejar consignada la calificación de la clase recibida por el Docente.

Sin embargo, los estudiantes realizaron observaciones a nivel de interfaz de usuario (ver resultados en el anexo AP), las cuales se detallan a continuación:

1. En todas las ventanas modales añadir opción de cerrar.
2. En el recuadro de calificación de una clase colocar un texto explicativo antes, Ejemplo: “¿Cómo califica usted el desarrollo de esta clase?” (Versión web).
3. En el recuadro de calificación de una clase colocar un texto explicativo antes, Ejemplo: “Califique la clase” (Versión móvil).
4. Colocar mensajes flotantes informativos en las ventanas importantes, los cuales deben aparecer cuando se deslice el mouse por iconos que no son descriptivos por sí solos.

5. Cuando no se encuentre información para mostrar en los paneles secundarios estos deberán ser limpiados. Esto debe ocurrir principalmente cuando el estudiante quiera visualizar un curso que no tenga un proyecto docente asociado.

Como respuesta a las sugerencias en cuestión, el equipo de desarrollo del proyecto estudió el impacto de las mejoras sugeridas y el tiempo que tomaría su desarrollo, llegando a la conclusión de un impacto bajo a nivel de desarrollo y de su alto grado de significancia para enriquecer la capacidad de uso. Las soluciones abordadas por el equipo fueron:

1. Implementar una solución en la capa de presentación (Front-end) para añadir el botón de cerrado de ventana en la solución web de la plataforma. Se hizo uso de una configuración para incluir el botón en los sistemas de tipo core (navegadores web) del componente genérico de presentación de registros.
2. Implementar una solución en la capa de presentación (Front-end) para incluir las etiquetas informativas al momento de registrar un comentario. Se realizaron las validaciones necesarias para identificar en que plataforma se estaba ejecutando la acción y así poder variar el mensaje dependiendo del entorno, si era web o móvil.
3. Implementar una solución en la capa de presentación (Front-end) para mostrar mensajes flotantes sobre la iconografía de las ventanas modales. Se implementó una directiva del framework para hacer uso de la propiedad animada “tooltip”, configurando el mensaje a mostrar y las propiedades de estilo y visualización.
4. Implementar una solución en la capa de presentación (Front-end) para limpiar el panel de trabajo y mostrar textos informativos indicando el estado del curso seleccionado. Se aplicó una configuración a nivel del componente de panel de detalles.

De esta manera se cumplió con el cuarto objetivo del presente informe en relación a lo especificado en el documento SRS por el usuario Estudiante y el producto entregado.

#### **4.4.2.4 Auditor**

Este actor se encarga del control de todos los registros realizados por los docentes del programa académico al cual pertenece, para lo cual el sistema ofrece las funcionalidades de visualización y consulta de estadísticas que permiten, por cada curso, mostrar el porcentaje de asistencia estudiantil a las clases, la cantidad de temas vistos y no vistos con respecto al proyecto docente y la cantidad de registros diligenciados por el docente. De igual manera puede visualizar los comentarios realizados por los estudiantes en cada una de las clases, a partir de los registros diligenciados por los docentes. Esta prueba fue realizada por el jefe del departamento interno de Ingeniería de Software, el ingeniero Martín Monroy Ríos del programa de Ingeniería de Sistemas.

Durante la primera etapa de pruebas no se presentaron problemas al momento de probar los flujos especificados dentro de los casos de prueba para el actor, por lo que, en términos generales, el sistema se comportó acorde y sin contratiempos dadas las funciones principales. El auditor no tuvo problemas con ninguno de los casos funcionales establecidos en el diseño de pruebas, debido a que pudo comprobar correctamente la visualización de cada uno de los cursos que tiene habilitados para auditar (teniendo en cuenta el área que dirige), la visualización de la lista de registros de asistencia diligenciados por los docentes, y el detalle de cada registro: información general, lista de asistencia, comentarios y estadísticas. Además, pudo visualizar las estadísticas relacionadas con el porcentaje de asistencia estudiantil de cada curso, el avance de temas con respecto al proyecto docente del curso y la cantidad de registros de asistencia diligenciados y no diligenciados.

Sin embargo, el auditor realizó algunas sugerencias a nivel de interfaz de usuario (ver resultados en el anexo AO) que permiten aumentar el entendimiento y capacidad de uso de la plataforma. Las sugerencias son listadas a continuación:

1. En todas las ventanas modales añadir opción de cerrar.
2. El espacio de trabajo de un auditor debe ser limpiado en caso de que el curso a auditar seleccionado no tenga proyecto docente.

3. Mejora en los diseños de las gráficas estadísticas: Añadir etiquetas informativas, buen manejo de colores, omisión de legendas innecesarias.
4. Incluir mensajes flotantes informativos en los íconos de segmentos de vista de todas las ventanas modales.
5. En el componente de selección de estudiantes, perteneciente a la gráfica asistencia estudiantil, agregar un marco para que este componente tenga más visibilidad.
6. En el componente de selección de estudiantes, perteneciente a la gráfica asistencia estudiantil, ordenar la lista de estudiantes alfabéticamente.

Como respuesta a las sugerencias en cuestión, el equipo de desarrollo del proyecto estudió el impacto de las mejoras sugeridas y el tiempo que tomaría su desarrollo, llegando a la conclusión de un impacto bajo a nivel de desarrollo y de su alto grado de significancia para enriquecer la capacidad de uso. Las soluciones abordadas por el equipo fueron:

1. Implementar una solución en la capa de presentación (Front-end) para añadir el botón de cerrado de ventana en la solución web de la plataforma. Se hizo uso de una configuración para incluir el botón en los sistemas de tipo core (navegadores web) del componente genérico de presentación de registros y estadísticas.
2. Implementar una solución en la capa de presentación (Front-end) para limpiar el panel de trabajo y mostrar textos informativos indicando el estado del curso seleccionado. Se aplicó una configuración a nivel del componente de panel de detalles.
3. Implementar una solución en la capa de presentación (Front-end) para cambiar los estilos de todas las gráficas. Se hizo uso de un objeto JSON para manipular las propiedades de configuración de la librería de terceros de gráficas y mejorar así la forma de presentación.
4. Implementar una solución en la capa de presentación (Front-end) para mostrar mensajes flotantes sobre la iconografía de las ventanas modales. Se implementó una directiva del framework para hacer uso de la propiedad animada “tooltip”, configurando el mensaje a mostrar y las propiedades de estilo y visualización.

5. Implementar una solución en la capa de presentación (Front-end) para cambiar los estilos del campo de opciones seleccionable. Se hizo uso de una clase en la hoja de estilos en cascada en el componente de estadísticas.
6. Implementar una solución en la capa de presentación (Front-end) para ordenar la lista de estudiantes y de asistencia de manera alfabética a partir de los apellidos de cada estudiante. Se hizo uso de un método público de la clase utilitaria, que consiste en un algoritmo de ordenamiento ascendente o descendente de listas, dentro del componente genérico de lista de estudiantes.

De esta manera se cumplió con el cuarto objetivo del presente informe en relación a lo especificado en el documento SRS por el usuario Auditor y el producto entregado.

#### **4.5 ANÁLISIS DE RESULTADOS**

El sistema de registro y control de asistencia académica, en adelante SIRECAA, ha sido pensado para reducir el tiempo y esfuerzo de los docentes, estudiantes, auditores y jefes de departamento en el proceso de registro de asistencia académica llevado a cabo en la Universidad de Cartagena. Teniendo en cuenta la fase de validación del sistema, a través de las pruebas ejecutadas se pudo constatar que la solución presentada ha cumplido con todas las expectativas y requisitos funcionales solicitados como parte del alcance inicial de la investigación, comprobando así que es una solución de calidad y que aporta valor a la Universidad de Cartagena.

Al comienzo de la investigación, se realizó un análisis a nivel internacional, nacional y local de algunas soluciones que, como parte del marco de referencia de la investigación, contaban con un conjunto de características útiles relacionadas con el planteamiento del problema. Es por eso que, a continuación, se compara el producto software obtenido en contraste con las soluciones software encontradas al realizar el estado del arte, identificando ventajas y desventajas entre ellas.

Una de las soluciones encontradas fue un Sistema de posicionamiento en Interiores (IPS), desarrollado por la Universidad de Aalborg en el año 2012. Este sistema, por medio de conexiones en red, permite la localización inalámbrica de personas o posición de objetos dentro de un edificio. Difiere con SIRECAA por cuanto este último no puede localizar inalámbricamente a los docentes que realizan los registros o los estudiantes que asisten a clase. Sin embargo, si permite conocer tanto la fecha como la hora en el momento de diligenciamiento de los registros, lo cual facilita identificar si éstos se realizaron dentro del rango de tiempo estipulado por el administrador, además que cuenta con gráficas y datos estadísticos que facilitan la toma de decisiones de los auditores, así como que su base de datos permite visualizar todos los registros de un curso y los comentarios realizados sobre una clase en particular.

Otra solución muy similar a SIRECAA fue la desarrollada en Maracaibo, Venezuela, para el año 2014, la cual consiste en un Sistema biométrico de control de asistencia laboral mediante el uso de huella dactilar, cuyo objetivo principal es llevar el registro sistemático de entradas y salidas. En cuanto a esta solución es válido aclarar que SIRECAA puede ser utilizado para el registro sistemático de asistencia estudiantil dentro de un curso, pero no posee ni el software ni hardware que permite el control biométrico por huella dactilar que ofrece información confiable y precisa de donde se encuentra el personal de interés. Esto puede ser considerado como una recomendación para trabajos futuros.

Se encontraron también soluciones empresariales relacionadas con el registro y control de asistencia, tal es el caso del Sistema para el Control de Empleados de PRODICAUCHOS LTDA, desarrollada por la Corporación Universitaria Minutos de Dios en Soacha, Cundinamarca, Colombia, en el año 2010, cuyo objetivo consiste en que por medio de una página web se registra la cantidad de empleados de la empresa, los productos y en general una base de datos de toda la comunidad. Esta solución está muy relacionada con SIRECAA en cuanto a la cantidad de información que ambos sistemas manejan, ya que SIRECAA se encuentra conectada, indirectamente, con la plataforma académica SMA de la Universidad de Cartagena, por lo que todos los datos de administrativos, estudiantes, docentes y auditores son provistos por las bases de datos de la institución. No obstante, a diferencia de

la solución implementada para la empresa PRODICAUCHOS, esta no solo es información almacenada, sino que es utilizada para generar la información de los cursos, temas, lista de asistencia y demás insumos necesarios para el diligenciamiento de los registros de asistencia académica, proyectos docente y departamentos internos.

A nivel local, en la ciudad de Cartagena de Indias, en la Fundación Universitaria Tecnológico Comfenalco, se crearon varios tipos de sistemas para controlar los procesos de asistencia estudiantil. Uno de estos sistemas es una página web que permite descargar planillas con la relación curso – estudiantes matriculados, que eran utilizados como soporte para la toma de asistencia de sus estudiantes y el registro de los temas dictados en clase. Otro de los sistemas se encontraba enfocado en el control de asistencia docente a los laboratorios de sistemas de la fundación y por último se tenía un sistema a gran escala que registraba la entrada y salida de los estudiantes a la fundación universitaria. Este es uno de los sistemas más parecidos a SIRECAA debido a que ambos pueden acceder a todas las listas de estudiantes matriculados en cada curso, permiten registrar la asistencia en cada clase y consignar los temas dictados, y aunque en la solución desarrollada por el equipo de investigación se tiene un control docente, éste no es tan riguroso como el que ofrece la fundación universitaria, pero puede ser utilizado para tal fin. Otra de las desventajas es que SIRECAA no posee un sistema a gran escala para consignar el ingreso de los estudiantes a las instalaciones de la Universidad de Cartagena.

A diferencia de todos los grandes sistemas que convergen en la solución propuesta por la Fundación Universitaria Tecnológico Comfenalco, cabe destacar que SIRECAA posee todas esas funcionalidades en una única plataforma que permite, por cada uno de los cursos, observar la lista de estudiantes matriculados sin necesidad de realizar ninguna descarga, ya que el sistema permite el diligenciamiento de los estudiantes que asisten y los que no, además que se pueden determinar cada uno de los temas tratados de acuerdo a un proyecto docente que ha sido aprobado por el consejo académico de un programa y que busca mantener unanimidad en todos los cursos de una misma asignatura. Además, posee un sistema estadístico para mostrar de forma clara, precisa y condensada lo que ha sucedido a nivel académico en un curso.

## CONCLUSIONES

---

En el desarrollo de esta investigación se obtuvo como resultado un sistema multiplataforma para gestionar el proceso de registro de asistencia académica en la Universidad de Cartagena, generando una primera versión estable del producto que facilita, principalmente, el diligenciamiento y almacenamiento de los registros para controlar la asistencia a clases. Además del producto software, se generaron artefactos importantes como el manual de usuario y el manual del sistema, con el objetivo de brindar asistencia técnica y servir de guía a los actores que hacen uso del mismo.

Los resultados obtenidos de las pruebas realizadas permiten afirmar que con la implementación de esta herramienta se obtienen las siguientes mejoras en dicho proceso:

- Facilita la manipulación, gestión y toma de decisiones oportuna en relación al porcentaje de asistencia estudiantil a las clases, avance de temas de cada uno de los cursos con respecto al proyecto docente asociado, y el diligenciamiento oportuno de los registros de asistencia por parte del docente.
- Reduce de tiempo y esfuerzo empleado por los actores para cumplir con las etapas del proceso.
- Disminuye el uso de archivo físico y el manejo del espacio, ya que los volúmenes de información física se ven reducidos al ser sustituidos por virtuales, permitiendo un acceso más rápido y preciso a los datos.
- Facilita la reducción de costos asociados al uso de elementos como papel y/o útiles de papelería, que dejan de ser necesarios para el proceso, otorgando de manera implícita un carácter ecológico relevante.

En el mismo sentido cabe resaltar que la implementación de este sistema permite:

- Aportar capacidad de uso y ejecución de las funcionalidades principales definidas en el alcance y en el SRS de esta investigación.

- Integrarse de manera fácil y flexible con los sistemas de la Universidad, específicamente con la plataforma académica SMA, garantizando sincronía con el sistema de datos de la institución y con el entorno universitario sin que represente un problema de gran envergadura.

Por otra parte, se pudo corroborar que, al comparar el producto obtenido en esta investigación con aquellos expuestos en la literatura previa, existen similitudes y diferencias: aunque aquellas soluciones se encontraban muy enfocadas en el registro de asistencia, cabe resaltar que el producto obtenido es una solución única y exclusivamente para la Universidad de Cartagena, ofreciendo precisión, calidad, consistencia, inmutabilidad de la información, facilidad de entendimiento, compatibilidad y confiabilidad de los datos, al ser un producto a la medida que trabaja muy bien de la mano con la principal plataforma académica de la Universidad, el SMA.

Al comparar con las investigaciones previas, se concluye entonces que, a diferencia de los demás sistemas, éste brinda una solución particular a la Universidad de Cartagena, permitiendo:

- Registrar de la asistencia estudiantil.
- Gestionar proyectos docente para cada asignatura, generando un punto único de información y temáticas para todos los cursos pertenecientes a esa asignatura.
- Mantener unanimidad en el desarrollo de las clases, y evitando con ello que diferentes cursos de la misma asignatura dicten o manejen temáticas diferentes.
- Generar comentarios u observaciones relacionadas con la clase.
- Tomar decisiones basadas en gráficas y estadísticas con relación a los cursos y registros de asistencia.

Todos estos aspectos hacen del producto un sistema con valor agregado y único, con características que, para el entorno donde se desenvuelve el problema, que es la Universidad de Cartagena, no fueron encontrados en alguna de las soluciones halladas dentro del marco de referencia de esta investigación.

Es así como los resultados de la investigación son importantes porque representan, para la Universidad, una herramienta a partir de la cual se obtiene los beneficios mencionados al principio de este capítulo, dando solución al problema planteado. Así mismo, generan un impacto para la Universidad porque sirven de guía para estandarizar un proceso tan crucial que a día de hoy no cuenta con lineamientos definidos y bien estructurados. Para los investigadores del proyecto, la investigación es importante porque representó una oportunidad para afianzar y fortalecer su conocimiento y formación como ingenieros de sistemas, permitiéndoles también cumplir con los requisitos establecidos por el programa para obtener el título de pregrado de Ingeniero de Sistemas.

No obstante, cabe resaltar que se presentaron algunas vicisitudes durante desarrollo de la investigación en derredor a escenarios y resultados inesperados que, de una u otra forma, retrasaron la construcción de la plataforma. La primera situación fue la dificultad, por cuestiones de carácter administrativo, para contar con los permisos para construir y exponer los servicios que le permitieran a la plataforma conectarse con la base de datos de la Universidad. Así mismo, fue difícil contar con la participación activa de un ingeniero de sistemas, funcionario de la división de sistemas de la institución, que estuviera presto a modificar, añadir o eliminar servicios web en caso que fuera necesario.

Sin embargo, aunque surgieron ciertos contratiempos, se pudo proseguir con el desarrollo de la investigación gracias a las herramientas utilizadas, el enfoque lógico abordado, la metodología de desarrollo implementada, la buena planeación y la aproximación definida a nivel tecnológico (lenguajes de programación, marcos de trabajo, gestor de base de datos, etc.).

Finalmente, tal y como se evidenció en el análisis de resultados, que el desarrollo de un proyecto como este agrega valor a la Universidad de Cartagena, permitiéndole estar a la altura de los tiempos en aspectos como la administración de procesos académicos, además que la posiciona como una entidad a la vanguardia en ofrecer una herramienta tecnológica como medio para el aseguramiento y fortalecimiento de la calidad académica.

## RECOMENDACIONES

---

Con el fin de escalar el sistema, como trabajo futuro se recomienda manejar un historial accesible de información con respecto a previos periodos académicos, así como poder mostrar comparativas a nivel estadístico entre la información perteneciente a distintos periodos. Actualmente, este proyecto se limitó a manejar simultáneamente información correspondiente a un solo periodo académico, el que se encuentra vigente en el momento de la consulta (a excepción de contados escenarios como el caso de visualización de proyectos docente). No obstante, dada la arquitectura del sistema, se puede implementar la mejora anteriormente planteada a través de la construcción de servicios backend y componentes frontend que permitan obtener y visualizar la información distinguiendo el periodo académico deseado.

Igualmente, para el caso de un despliegue móvil con más cobertura de usuarios, se recomienda implementar la compilación del sistema para entornos móviles que cuentan con sistema operativo iOS, teniendo en cuenta que la cantidad de dispositivos que cuentan con esta arquitectura son cada vez más demandados en el mercado. Gracias a las herramientas tecnológicas y el lenguaje utilizado para la construcción de la interfaz gráfica del proyecto, es posible realizar este despliegue de dos maneras: haciendo uso del entorno de desarrollo de aplicaciones móviles, como Cordova, o ejecutando un proceso de migración de la plataforma (componente visual) a una aplicación web progresiva, PWA. En cualquiera de los dos escenarios anteriores, se podría lograr el objetivo propuesto.

Por último, a nivel administrativo, se recomienda estudiar la posibilidad de contar con un personal encargado de la modificación de los servicios actualmente expuestos por la universidad que gestionan la información que sirve como insumo fundamental para el funcionamiento de la plataforma, debido a que pueden implementarse mejoras en la forma en cómo se obtienen, procesan y devuelven los datos, aportando valor para que el sistema contemple otros escenarios a nivel del proceso de asistencia académica y sea más completo en términos de alcance.

## ANEXOS

---

**ANEXO A: Diseño de entrevista para la identificación del problema de registro y control de asistencia y temas vistos en la Universidad de Cartagena**



Diseño de  
entrevista N°1.pdf

**ANEXO B: ACTA N° 1. Correspondiente a la entrevista realizada como método de recolección de información para conocer el problema en cuestión**



Acta N°1.pdf

**ANEXO C: Diseño de entrevista para el desarrollo e implementación de una plataforma móvil y web para un sistema de registro y control de asistencia y temas vistos en la Universidad de Cartagena**



Diseño de  
entrevista N°2.pdf

**ANEXO D: ACTA N° 2. Correspondiente a la entrevista realizada como método de recolección de información para el desarrollo de un sistema software como plataforma móvil y web enfocada en el registro y control de asistencia y temas vistos en la Universidad de Cartagena**



Acta N°2.pdf

**ANEXO E: Diseño de entrevista para conocer los sistemas de control de asistencia de la Fundación Universitaria Tecnológico Comfenalco**



Diseño de  
entrevista N°3.pdf

**ANEXO F: ACTA N° 3. Correspondiente a la entrevista realizada como método de recolección de información sobre un software para controlar la asistencia en la Fundación Universitaria Tecnológico Comfenalco**



Acta N°3.pdf

**ANEXO G: Especificación de requisitos del software (SRS).**



Especificación de  
Requisitos de Softw

**ANEXO H: Diagrama de actividades en formato digital.**



Diagrama de  
Actividades.png

**ANEXO I: Diagrama de clases funcionalidad gestionar departamentos.**



Gestionar  
departamentos.png

**ANEXO J: Diagrama de clases funcionalidad gestionar proyectos docentes.**



Gestionar  
proyectos docente.t

**ANEXO K: Diagrama de clases funcionalidad realizar registros académicos.**



Realizar registros  
académicos.png

**ANEXO L: Diagrama de clases funcionalidad visualizar cursos.**



Visualizar  
cursos.png

**ANEXO M: Diagrama de clases funcionalidad visualizar registros académicos.**



Visualizar registros  
académicos.png

**ANEXO N: Diagrama de clases funcionalidad registrar comentarios sobre registros académicos.**



Registrar  
comentarios sobre r

**ANEXO O: Diagrama de clases funcionalidad visualizar estadísticas de cursos y registros.**



Visualizar  
estadísticas de cursc

**ANEXO P: Diagrama de componentes.**



Diagrama de  
componentes.png

**ANEXO Q: Diagrama de secuencia del caso de uso: Consultar base de datos.**



Consultar base de  
datos.png

**ANEXO R: Diagrama de secuencia del caso de uso: Cargar departamentos.**



Cargar  
departamentos.png

**ANEXO S: Diagrama de secuencia del caso de uso: Crear departamento.**



Crear  
departamento.png

**ANEXO T: Diagrama de secuencia del caso de uso: Editar departamento.**



Editar  
departamento.png

**ANEXO U: Diagrama de secuencia del caso de uso: Eliminar departamento.**



Eliminar  
departamento.png

**ANEXO V: Diagrama de secuencia del caso de uso: Visualizar departamento.**



Visualizar  
departamento.png

**ANEXO W: Diagrama de secuencia del caso de uso: Cargar proyectos docente.**



Cargar proyectos  
docente.bmp

**ANEXO X: Diagrama de secuencia del caso de uso: Crear proyecto docente.**



Crear proyecto  
docente.bmp

**ANEXO Y: Diagrama de secuencia del caso de uso: Editar proyecto docente.**



Editar proyecto  
docente.bmp

**ANEXO Z: Diagrama de secuencia del caso de uso: Eliminar proyecto docente.**



Eliminar proyecto  
docente.bmp

**ANEXO AA: Diagrama de secuencia del caso de uso: Visualizar proyecto docente.**



Visualizar proyecto  
docente.bmp

**ANEXO AB: Diagrama de secuencia del caso de uso: Iniciar sesión.**



Iniciar sesión.png

**ANEXO AC: Diagrama de secuencia del caso de uso: Procesar mensaje de error.**



Procesar mensaje  
de error.png

**ANEXO AD: Diagrama de secuencia del caso de uso: Realizar registros académicos.**



Realizar registros  
académicos.png

**ANEXO AE: Diagrama de secuencia del caso de uso: Registrar comentarios sobre registros académicos.**



Registrar  
comentarios sobre r

**ANEXO AF: Diagrama de secuencia del caso de uso: Visualizar cursos.**



Visualizar  
cursos.png

**ANEXO AG: Diagrama de secuencia del caso de uso: Visualizar estadísticas de cursos y registros.**



Visualizar estadísticas de curso

**ANEXO AH: Diagrama de secuencia del caso de uso: Visualizar registros académicos.**



Visualizar registros académicos.png

**ANEXO AI: Diagrama paquetes.**



Diagrama de paquetes.png

**ANEXO AJ: Diagrama despliegue.**



Diagrama de despliegue.png

**ANEXO AK: Diagrama general de casos de uso.**



Vista de escenarios.png

**ANEXO AL: Diseño de pruebas de carga.**



Diseño pruebas de carga.xlsx

**ANEXO AM: Diseño de pruebas de caja negra.**



Diseño pruebas de  
caja negra.xlsx

**ANEXO AN: Resultados de pruebas funcionales: Actor administrador.**



Pruebas  
funcionales - Admin

**ANEXO AO: Resultados de pruebas funcionales: Actor docente - auditor.**



Pruebas  
funcionales - Docen

**ANEXO AP: Resultados de pruebas funcionales: Actor estudiante.**



Pruebas  
funcionales - Estudi

## REFERENCIAS BIBLIOGRÁFICAS

---

- ABC. (14 de Mayo de 2012). Banco Santander y Vodafone España anuncian un proyecto piloto para el acceso y control de asistencia en la Universidad. *ABC.es*. Recuperado el 13 de Marzo de 2017, de <http://www.abc.es/20120514/tecnologia/abci-banco-santander-vodafone-espana-201205141722.html>
- Alonso Bernate, J. C., & Pinzón Caballero, L. E. (2012). *ANÁLISIS Y DISEÑO DE SOFTWARE PARA EL CONTROL DE ASISTENCIA DE ESTUDIANTES, "STEEL"*. Soacha, Cundinamarca, Colombia: Corporación Universitaria Minuto de Dios. Recuperado el 13 de Marzo de 2017, de [http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2592/TTI\\_AlonsoBernateJuanCarlos\\_2012.pdf?sequence=1](http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2592/TTI_AlonsoBernateJuanCarlos_2012.pdf?sequence=1)
- BBVA. (23 de Marzo de 2016). *API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos*. Recuperado el 9 de Julio de 2018, de BBVA API Market Web site: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- Cerda, H. (1991). Medios, Instrumentos, Técnicas y Métodos en la Recolección de Datos e Información. En H. Cerda, *Los elementos de la investigación*. Caracas, Venezuela: Universidad Nacional Abierta. Recuperado el 18 de Abril de 2017, de <http://postgrado.una.edu.ve/metodologia2/paginas/cerda7.pdf>
- Espinosa Cantillo, N. (6 de Octubre de 2015). Acta No 1, Estudio de viabilidad de Implementación de un Sistema de Registro y Control de Asistencia. *Ingeniería de Sistemas*. (E. Cabarcas, J. Cantillo, & J. Suárez, Entrevistadores) Cartagena de Indias, Bolívar, Colombia. Recuperado el 7 de Septiembre de 2015
- Espinosa Cantillo, N. (6 de Marzo de 2017). Acta No 2, Conocimiento del problema en estudio. (E. Cabarcas, & J. Cantillo, Entrevistadores) Cartagena, Bolívar, Colombia. Recuperado el 10 de Marzo de 2017

- Express. (s.f.). *Framework for Node.js*. Recuperado el 7 de Julio de 2018, de Express Web site: <http://expressjs.com/>
- Ezkauriatza, M. G. (2011). *TRABAJO COLABORATIVO EN LA WEB: ENTORNO VIRTUAL DE AUTOGESTIÓN PARA DOCENTES*. Palma de Mallorca: UNIVERSITAT DE LES ILLES BALEARS. Recuperado el 21 de 05 de 2017, de <http://www.tdx.cat/bitstream/handle/10803/59037/tmge1de1.pdf>
- Frost, C., Svenning Jensen, C., Søe Luckow, K., Thomsen, B., & Hansen, ., R. (2012). Bluetooth Indoor Positioning System Using Fingerprinting. *Mobile Lightweight Wireless Systems*, 81, 136-150. Recuperado el 13 de Marzo de 2017, de [https://link.springer.com/chapter/10.1007/978-3-642-29479-2\\_11](https://link.springer.com/chapter/10.1007/978-3-642-29479-2_11)
- FUSIONAR, Soluciones Tecnológicas. (s.f.). *Productos Software: fusionar.com.uy*. Recuperado el 13 de Marzo de 2017, de fusionar.com.uy: <http://www.fusionar.com.uy/productos-software.php>
- Google Developers. (s.f.). *V8 Introduction: Google Developers*. Recuperado el 1 de Julio de 2018, de Google Developers Web site: <https://developers.google.com/v8/>
- GYÖRÖDI, C., GYÖRÖDI, R., PECHERLE, G., & OLAH, A. (2015). A Comparative Study: MongoDB vs. MySQL. *IEEE Explore*, 7. Recuperado el 8 de Julio de 2018, de [https://www.researchgate.net/profile/Cornelia\\_Gyroedi2/publication/278302676\\_A\\_Comparative\\_Study\\_MongoDB\\_vs\\_MySQL/links/557fcb1a08aeea18b7797116.pdf](https://www.researchgate.net/profile/Cornelia_Gyroedi2/publication/278302676_A_Comparative_Study_MongoDB_vs_MySQL/links/557fcb1a08aeea18b7797116.pdf)
- Hossein Ahmadpanah, S. (2015). *What is Git ?!* USA: CreateSpace Independent Publishing Platform , USA ©2015. Obtenido de <https://dl.acm.org/citation.cfm?id=2834562>
- IBM. (s.f.). *IBM Knowledge Center*. Recuperado el 11 de Marzo de 2017, de Aplicaciones Java SE y Java EE: [http://www.ibm.com/support/knowledgecenter/es/SSQP76\\_7.5.0/com.ibm.dserver.res/Content/Business\\_Rules/\\_pubskel/Infocenter\\_Primary/ps\\_DS\\_Rule\\_Execution\\_Server1444.html](http://www.ibm.com/support/knowledgecenter/es/SSQP76_7.5.0/com.ibm.dserver.res/Content/Business_Rules/_pubskel/Infocenter_Primary/ps_DS_Rule_Execution_Server1444.html)

- International Organization for Standardization. (2015). *ISO 9000:2015(es) - Sistemas de gestión de la calidad: Fundamentos y vocabulario - Ed. 4*. Recuperado el 10 de Marzo de 2017, de ISO: Online Browsing Platform: <https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:es>
- Ionic. (s.f.). *What is Ionic, and where does it fit?* Recuperado el 1 de Julio de 2018, de Ionic Framework Web site: <https://ionicframework.com/docs/v1/guide/preface.html>
- Kruchten, P. B. (Noviembre de 1995). The 4+1 View Model of Architecture. *IEEE software*, 42-50. Recuperado el 26 de Abril de 2017, de <http://jz81.github.io/course/smt/The%204+1%20View%20Model%20of%20Architecture.pdf>
- López Escalera, R., & Gaona Campos, H. U. (Junio de 2016). Desarrollo e Implementación de un Sistema Electrónico de Control de Asistencia para los Profesores de la Universidad Michoacana de San Nicolás de Hidalgo. *Revista de la Facultad de Contaduría y Ciencias Administrativas*, I(1), 364-378. Recuperado el 21 de Abril de 2017, de <http://rfcca.umich.mx/index.php/rfcca/article/view/25/29>
- Lorduy Salas, I. J., Peña Esquivel, Á. E., & Puello Marrugo, P. (2014). *Desarrollo de una plataforma para la gestión de objetos virtuales de aprendizaje para la Facultad de Odontología en la Universidad de Cartagena*. Cartagena, Bolívar, Colombia: Universidad de Cartagena. Recuperado el 11 de Marzo de 2017, de <http://190.242.62.234:8080/jspui/bitstream/11227/421/1/TESIS%20DE%20GRADO.pdf>
- Márquez C, L. E., Abdo L, Y. N., & Angulo, F. J. (23 de Noviembre de 2014). PROTOTIPO DE CONTROL DE ACCESO A AULAS Y REGISTRO AUTOMÁTICO DE ASISTENCIA. *Revista Colombiana de Tecnologías Avanzadas*. Recuperado el 13 de Marzo de 2017, de [http://www.unipamplona.edu.co/unipamplona/portallIG/home\\_40/recursos/05\\_v25\\_30/revista\\_26/01052016/06.pdf](http://www.unipamplona.edu.co/unipamplona/portallIG/home_40/recursos/05_v25_30/revista_26/01052016/06.pdf)

- Marrugo Marrugo, Y. J., Nuñez Barcos, R., & Martelo Gómez, R. J. (2012). *Sistema software de apoyo al proceso de creación y registro de políticas de seguridad informática en organizaciones*. Cartagena, Bolívar, Colombia: Universidad de Cartagena. Recuperado el 11 de Marzo de 2017, de <http://190.242.62.234:8080/jspui/bitstream/11227/392/1/SOFTWARE%20DE%20APOYO%20AL%20PROCESO%20DE%20CREACION%20Y%20REGISTRO%20DE%20PSI%20EN%20ORGANIZACIONES%20%281%29.pdf>
- Martínez, A., & Martínez, R. (2014). *Guía a Rational Unified Process*. Albacete: Universidad de Castilla La Mancha. Recuperado el 10 de Marzo de 2017, de <https://anaylenlopez.files.wordpress.com/2011/03/trabajo-guia20rup.pdf>
- Masse, M. (2012). *Designing Consistent RESTful Web Service Interfaces*. 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA: O'Reilly Media Inc.
- Meier, J., Homer, A., Hill, D., Taylor, J., Bansode, P., Wall, L., . . . Bogawat, A. (2009). *Microsoft Application Architecture Guide*. Microsoft Corporation. Recuperado el 9 de Julio de 2018
- Ministerio de Educación Nacional de Colombia. (09 de Octubre de 2001). *Sistemas de información para mejorar la gestión*. Recuperado el 22 de 08 de 2015, de Ministerio de Educación Nacional: <http://www.mineduccion.gov.co/1621/article-87646.html>
- Ministerio de Educación Nacional de Colombia. (5 de Febrero de 2016). *Sistema Educativo Colombiano: Ministerio de Educación*. Recuperado el 10 de Marzo de 2017, de Mineduccion: <http://www.mineduccion.gov.co/1759/w3-article-231235.html>
- MongoDB. (s.f.). *MongoDB Architecture*. Recuperado el 7 de Julio de 2018, de mongoDB: <https://www.mongodb.com/mongodb-architecture>
- MongoDB. (s.f.). *What is MongoDB?: mongoDB*. Recuperado el 8 de Julio de 2018, de mongoDB Web site: <https://www.mongodb.com/what-is-mongodb>

- MongoDB. (s.f.). *What is NoSQL?: mongoDB*. Recuperado el 7 de Julio de 2018, de mongoDB Web site: <https://www.mongodb.com/nosql-explained>
- Mongoose. (s.f.). *What's Mongoose?: Mongoose*. Recuperado el 1 de Julio de 2018, de Mongoose Web site: <http://mongoosejs.com/>
- Mozilla Developer Network: MDN. (12 de Mayo de 2016). *JavaScript: MDN*. Recuperado el 11 de Marzo de 2017, de developer.mozilla.org: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mozilla Developer Network: MDN. (2 de Febrero de 2017). *CSS developer guide: MDN*. Recuperado el 27 de Abril de 2017, de developer.mozilla.org: <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- Mozilla Developer Network: MDN. (3 de Enero de 2017). *HTML5: MDN*. Recuperado el 11 de Marzo de 2017, de developer.mozilla.org: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- Object Management Group. (Julio de 2005). *UML.org*. Recuperado el 10 de Marzo de 2017, de ¿Qué es UML?: Introducción para Unified Modeling Language de OMG: <http://www.uml.org/what-is-uml.htm>
- Olivares Morales, A., & Valeriano Assem, J. (2010). *Automatización del proceso de control de asistencia del personal académico en tiempo real a través de reconocimiento biométrico*. Ciudad de México: Universidad Nacional Autónoma de México. Recuperado el 13 de Marzo de 2017, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/1188/Tesis.pdf?sequence=1>
- Oracle. (s.f.). *JavaEE: Oracle.com*. Recuperado el 11 de Marzo de 2017, de oracle.com: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- Pantoja Blyde, J., Lozano Leal, A., & Portillo Montiel, M. (1 de Mayo de 2013). AUTOMATIZACIÓN DEL CONTROL DE ASISTENCIA DEL PERSONAL

DOCENTE DEL DEPARTAMENTO DE COMPUTACIÓN DE LA FACULTAD EXPERIMENTAL DE CIENCIAS DE LA UNIVERSIDAD DEL ZULIA. *Revistas Electrónicas URBE*. Recuperado el 13 de Marzo de 2017, de <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/2306/html>

Pérez, A., El Safadi, A., & Araujo, R. (2014). *Sistema biométrico de control de asistencia laboral mediante el uso de huella dactilar*. Maracaibo, Zulia, Venezuela: Universidad Rafael Urdaneta. Recuperado el 13 de Marzo de 2017, de <http://200.35.84.131/portal/bases/marc/texto/2501-14-07814.pdf>

Pressman, R. S. (2010). *Ingeniería del Software Un Enfoque Práctico*. México D.F: McGraw Hill. Recuperado el 10 de Marzo de 2017, de <http://cotana.informatica.edu.bo/downloads/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF>

Pressman, R. S. (2010). *Ingeniería del software, Un enfoque práctico* (Séptima ed.). México, D. F.: MCGRAW-HILL INTERAMERICANA EDITORES. Recuperado el 28 de Noviembre de 2018

Ramos, C. M. (11 de Marzo de 2017). Acta No 3, Conocer los sistemas de control de asistencia de la Corporación Universitaria Tecnológico Comfenalco. (J. Cantillo, Entrevistador) Cartagena de Indias, Bolívar, Colombia.

Reyes Puerto, N., & Bermudez Amaya, M. O. (2010). *PLATAFORMA DE RECOLECCIÓN Y SISTEMATIZACIÓN DE DATOS PARA LA ENTRADA Y SALIDA DE LOS EMPLEADOS*. Soacha, Cundinamarca, Colombia: Corporación Universitaria Minuto de Dios. Recuperado el 13 de Marzo de 2017, de [http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2710/TTI\\_ReyesPuertoNathalyConcepcion\\_2010.pdf?sequence=1&isAllowed=y](http://repository.uniminuto.edu:8080/xmlui/bitstream/handle/10656/2710/TTI_ReyesPuertoNathalyConcepcion_2010.pdf?sequence=1&isAllowed=y)

Rosado Gomez, A. A., & Jaimes Fernández, J. C. (2018). Rosado Gomez, Alveiro Alonso; Jaimes Fernández, Juan Camilo. *Revista Colombiana de Tecnologías de Avanzada*, 1(31), 77-88. doi:<https://doi.org/10.24054/16927257.v31.n31.2018.2769>

- Sandoval Illescas, J. X., Sigüenza Cárdenas, R. E., & Ingavélez G, P. (2011). *Análisis, Diseño e Implementación del Sistema de Control de Asistencia de Personal Docente y Administrativo de la Escuela Fiscal Mixta Rafael Aguilar Pesantez*. Cuenca, Azuay, Ecuador: Universidad Politécnica Salesiana Sede Cuenca. Recuperado el 14 de Marzo de 2017, de <http://dspace.ups.edu.ec/bitstream/123456789/1636/14/UPS-CT002157.pdf>
- Softtronics Solutions S.A. (s.f.). *¿Quiénes somos?: stt-solutions.com*. Recuperado el 13 de Marzo de 2017, de stt-solutions.com: <http://www.stt-solutions.com/quienes.php>
- Softtronics Solutions S.A. (s.f.). *BioPoint: stt-solutions.com*. Recuperado el 13 de Marzo de 2017, de stt-solutions.com: <http://www.stt-solutions.com/producto-detalle.php?idProduct=30>
- SquareNet Software Solutions. (s.f.). *SquareNet Gestión Humana: SquareNet.com.ec*. Recuperado el 13 de Marzo de 2017, de SquareNet.com.ec: <http://www.squarenet.com.ec/>
- Suntaxi Suntaxi, S., Moromenacho Caiza, D., Arroyo, R., & Parra, P. (Mayo de 2014). Desarrollo de una aplicación web de control de asistencias para programas de postgrado, mediante la utilización de la plataforma Java Enterprise Edition JEE6 Web aplicando la metodología SCRUM. *Repositorio ESPE*. Recuperado el 13 de Marzo de 2017, de <http://repositorio.espe.edu.ec:8080/bitstream/21000/8337/1/AC-SI-ESPE-047930.pdf>
- Thompson, S. (25 de Noviembre de 2015). *V8 Introduction: GitHub*. Obtenido de GitHub Web site: <https://github.com/v8/v8/wiki/Introduction>
- Universidad de Cartagena. (2009). *Reglamento Estudiantil* (Acuerdo N° 14 del 11 de Diciembre ed.). Cartagena, Bolívar, Colombia: Universidad de Cartagena. Recuperado el 11 de Marzo de 2017
- Universidad de Cartagena. (13 de Septiembre de 2016). *Unicartagena*. Recuperado el 10 de Marzo de 2017, de Misión: Universidad de Cartagena:

<http://www.unicartagena.edu.co/inicio/transparencia-y-acceso-a-la-informacion/informacion-institucional/mision>

Universidad de Jaén. (19 de Marzo de 2012). *Sistema Integrado de Gestión de la Calidad de los Servicios y Unidades Administrativas*. Recuperado el 10 de Marzo de 2017, de Control de los Registros: Universidad del Jaén: [http://www.ujaen.es/serv/spe/sigcsua/SIGCSUA\\_PD02.pdf](http://www.ujaen.es/serv/spe/sigcsua/SIGCSUA_PD02.pdf)

Universidad Nacional de Colombia. (7 de Enero de 2016). *UNAL*. Recuperado el 10 de Marzo de 2017, de Mejoramiento de la gestión: UNAL: [http://unal.edu.co/fileadmin/user\\_upload/U-PR-11.005.020\\_Procedimiento\\_Control\\_de\\_documentos\\_V5.pdf](http://unal.edu.co/fileadmin/user_upload/U-PR-11.005.020_Procedimiento_Control_de_documentos_V5.pdf)

Valenzuela Zambrano, B., & Pérez Villalobos, M. V. (15 de Febrero de 2013). Aprendizaje autorregulado a través de la plataforma virtual Moodle. *ProQuest*. Recuperado el 11 de Marzo de 2017, de <http://search.proquest.com/openview/21c67853c577d8376809d5ae20018a0f/1?pq-origsite=gscholar&cbl=366360>

Varela, R. (2 de Abril de 2015). *Tipos de controles de presencia: Joblers*. Recuperado el 13 de Marzo de 2017, de blog.joblers.net: <http://blog.joblers.net/tipos-de-controles-de-presencia.html>

Walia, H., & Jain, N. (5 de Mayo de 2016). Fingerprint Based Attendance Systems-A Review. *International Research Journal of Engineering and Technology (IRJET)*, III. Recuperado el 28 de Marzo de 2017, de <https://www.irjet.net/archives/V3/i5/IRJET-V3I5237.pdf>

World Wide Web Consortium: W3C. (s.f.). *HTML: W3.org*. Recuperado el 11 de Marzo de 2017, de w3.org: <https://www.w3.org/html/>

Ysabel Zavahra Tahanian. (Marzo de 2012). *Plataformas Educativas*. Recuperado el 21 de 05 de 2017, de Tipos de Plataformas Virtuales: <https://sites.google.com/site/plataformaseducativasvirtuales/home/tipos>

Zambrano, J. J., Numa, M., & Gómez, A. (2014). PROPUESTA DE SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS SERVICIOS E INFORMACIÓN DEL REGISTRO ACADÉMICO EN EL LICEO BOLIVARIANO «NÉSTOR LUIS PÉREZ» EN TUCUPITA, ESTADO DELTA. *Universidad De Ciego De Ávila Máximo Gómez Báez, III(2)*. Recuperado el 20 de Abril de 2017, de <http://revistas.unica.cu/index.php/uciencia/article/view/346/217>