

UNIVERSIDAD PRIVADA ANTENOR ORREGO
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE
INGENIERÍA DE COMPUTACIÓN Y SISTEMAS



**“PROTOTIPO DE SISTEMA INTEGRADO DE GESTIÓN
ACADÉMICA PARA LA INSTITUCIÓN EDUCATIVA PRIVADA
‘SAGRADO CORAZÓN DE JESÚS’ DE CASA GRANDE UTILIZANDO
LA METODOLOGÍA RUP Y EL FRAMEWORK CAKE PHP”**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE COMPUTACIÓN Y SISTEMAS
LÍNEA DE INVESTIGACIÓN: SISTEMAS DE INFORMACIÓN**

AUTOR: Br. Guillermo Jesús Valdez Peláez
ASESOR: Ing. Jorge Luis Piminchumo Flores

TRUJILLO - PERÚ
2016

ÍNDICE PRINCIPAL

ACREDITACIONES	3
PRESENTACIÓN	4
DEDICATORIA	5
AGRADECIMIENTOS	6
RESUMEN	7
ABSTRACT	8
ÍNDICE.....	9
ÍNDICE DE TABLAS	11
ÍNDICE DE FIGURAS	12
ANEXOS	185

ACREDITACIONES

TÍTULO: “PROTOTIPO DE UN SISTEMA INTEGRADO DE GESTIÓN ACADÉMICA PARA LA IEP ‘SAGRADO CORAZÓN DE JESÚS’ UTILIZANDO LA METODOLOGIA RUP Y EL FRAMEWORK CAKE PHP”

AUTOR: Guillermo Jesús Valdez Peláez

APROBADO POR:

Ing. Walter Aurelio Lazo Aguirre
PRESIDENTE
N° CIP: 36034

Ing. Heber Gerson Abanto Cabrera
SECRETARIO
N° CIP: 106421

Ing. Silvia Ana Rodríguez Aguirre
VOCAL
N° CIP: 107615

Ing. Jorge Luis Piminchumo Flores
ASESOR
N° CIP: 137153

PRESENTACIÓN

Señores Miembros del Jurado:

Dando cumplimiento y conforme a las normas establecidas en el Reglamento de Grados y Títulos y Reglamento de la Facultad de Ingeniería de la Universidad Privada Antenor Orrego, para obtener el título profesional de Ingeniero de Computación y Sistemas, se pone a vuestra consideración el Informe del Trabajo de Investigación Titulado

“PROTOTIPO DE UN SISTEMA INTEGRADO DE GESTIÓN ACADÉMICA PARA LA IEP ‘SAGRADO CORAZÓN DE JESÚS’ UTILIZANDO LA METODOLOGIA RUP Y EL FRAMEWORK CAKE PHP”

, con la convicción de alcanzar una justa evaluación y dictamen, excusándome de antemano de los posibles errores involuntarios cometidos en el desarrollo del mismo.

Trujillo, 02 de Diciembre del 2016.

Guillermo Jesús Valdez Peláez

DEDICATORIA

A Dios, por iluminarme en este largo camino que aún falta recorrer, por darme fuerza y perseverancia para salir adelante y continuar con mis objetivos trazados.

A mis padres Guillermo y Betty por su gran amor, su apoyo incondicional y sus sabios consejos para ser una buena persona.

AGRADECIMIENTOS

Agradezco infinitamente a mis padres Guillermo y Betty quienes depositaron sus esfuerzos y esperanzas en mí desde el inicio hasta la actualidad, gracias a ellos soy lo que soy hoy por hoy. Los amo infinitamente y estaré eternamente agradecido.

A mi asesor Jorge Piminchumo Flores, por su asesoramiento profesional a lo largo del desarrollo de esta tesis, quien me proporciono ideas para realizar mi tesis de la mejor manera.

A todos mis amigos y profesionales que de alguna u otra manera me apoyaron en muchos aspectos de mi vida y a lo largo de ella.

RESUMEN

Este trabajo consiste en el desarrollo de un prototipo de Sistema Integrado de la Gestión Académica para la Institución Educativa Privada “Sagrado Corazón de Jesús” de la ciudad de Casa Grande, el cual consiste es únicamente un prototipo de un sistema de información para controlar y gestionar los procesos académicos básicos de la institución, haciendo que los procesos sean más eficientes reduciendo tiempo y costo para beneficio de todos los usuarios de las distintas áreas.

Actualmente, la Institución “Sagrado Corazón de Jesús” carece de un sistema de información que le permita gestionar eficientemente los procesos académicos que realiza, tales como admisión y matriculas, programación de horarios y clases, asistencias, gestión de egresados y promociones, etc realizando mayormente el proceso manualmente y guardando parte de la información en archivos separados como en Excel y Word, los cuales no permiten obtener eficientemente la información para la toma de decisiones

La propuesta es un prototipo del sistema de información que podrá solucionar esos problemas, el cual se desarrollará usando la metodología RUP y el framework CakePHP, y se encargará de la gestión de los procesos académicos importantes de la institución. Debido a motivos de tiempo y estudios, el investigador sólo trabajará hasta esa fase, porque no será un sistema funcionando al 100%.

Luego de desarrollar el prototipo usando RUP y CakePHP, se obtuvo un prototipo que cumple al menos con la mayoría de expectativas que tienen los clientes (el colegio y sus integrantes administrativos), lo cuales se sienten conformes con ese prototipo, el cual, de convertirse en un sistema completo, sería aún más confortante para ellos.

De antemano, pido disculpas por cualquier error que pueda encontrarse en el presente trabajo, puesto a que no ha sido de manera intencional, sino errores propios que cualquier ser humano puede cometer en cualquier situación.

ABSTRACT

This work consists of the development of a prototype of Integrated System of the Academic Management for the Educational Private Institution “Sagrado Corazón de Jesús” from Casa Grande city, which consists it is only a prototype of a system of information to control and to manage the academic basic processes of the institution, doing that the processes are more efficient reducing time and cost for benefit of all the users of the different areas.

Currently, the Institution lacks a system of information that allows him to manage efficiently the academic processes that it realizes, such as admission and you register, programming of schedules and classes, assistance, management of gone away and promotions, etc realizing mainly the process manually and keeping part of the information in files separated like in Excel and Word, which do not allow to obtain efficiently the information for the decision making.

The proposal is a prototype of the information system that will be able to solve these problems, which will be developed using the methodology RUP and the framework CakePHP, and will be responsible for the management of the processes leading academics of the institution. Due to reasons of time and studies, the researcher will only work up to that stage, because it will not be a system running at 100%.

After developing the prototype using RUP and CakePHP, was obtained a prototype that meets the majority of expectations that have clients (the school and its members administrative), they feel they are in conformity with that prototype, which become a complete system, would be even more comforting to them.

In advance, I apologize for any errors that may be found in the present work, since it has not been intentional, but errors that any human being can commit in any situation.

ÍNDICE

1. INTRODUCCIÓN	16
1.1. Realidad problemática	16
1.2. Delimitación del problema	17
1.3. Características y análisis del problema	17
1.4. Formulación del Problema	17
1.5. Formulación de la Hipótesis	18
1.6. Objetivos del estudio	18
1.7. Justificación del Estudio	18
1.8. Limitaciones del estudio	18
2. MARCO TEÓRICO	18
2.1. Antecedentes	18
2.2. Bases teóricas	20
2.2.1. Gestión Académica	20
2.2.2. Sistema de Información Gerencial	20
2.2.3. RUP	22
2.2.4. Framework Cake PHP	23
2.3. Definición de términos	25
2.3.1. Gestión Académica	25
2.3.2. Metodología de Desarrollo	25
2.3.3. Framework	25
3. MATERIAL Y MÉTODOS	26
3.1. Material	26
3.1.1. Población	26
3.1.2. Muestra	26
3.1.3. Unidad de Análisis	26
3.2. Método	26
3.2.1. Nivel de Investigación	26
3.2.2. Diseño de Investigación	26
3.2.3. Variables de estudio y operacionalización	27
3.2.4. Técnicas e Instrumentos de recolección de datos	28
3.2.5. Técnicas de Procesamiento de datos	28
3.2.6. Técnicas de análisis de datos	28
4. RESULTADOS	29
4.1. Modelado del Negocio	29

4.1.1.	Evaluación de la Organización.....	29
4.1.2.	Modelo Visual de Diagnostico	30
4.1.3.	Arquitectura del Negocio.....	34
4.1.4.	Modelo Visual Solucionador	37
4.1.5.	Reglas del Negocio	41
4.1.6.	MODELO DE CASO DE USO DEL NEGOCIO	42
4.1.7.	Especificaciones de Casos de Uso del Negocio	42
4.1.8.	Modelo de Objetos del Negocio	46
4.1.9.	Modelo De Dominio Del Problema	51
4.1.10.	Glosario del Negocio.....	51
4.1.11.	Visión Del Negocio.....	51
4.2.	Modelo de Requerimientos.....	56
4.2.1.	Necesidades de los stakeholders	56
4.2.2.	Casos De Uso De Requerimientos	58
4.2.3.	Especificaciones de los casos de uso:.....	62
4.2.4.	Diagramas de Actividad.....	73
4.3.	Analisis y Diseño.....	91
4.3.1.	Diagramas de Colaboración	91
4.3.3.	Modelo Físico	115
4.3.4.	Pantallas	116
4.3.5.	Diagrama de componentes.....	137
4.3.6.	Diagrama de despliegue	137
4.4.	IMPLEMENTACION	138
4.4.1.	Estructuras de Clases con Cake PHP	138
4.4.2.	Código.....	150
4.4.3.	Base de datos(script)	160
5.	DISCUSIÓN DE RESULTADOS	173
5.1.	Variable: El grado de cumplimiento de la elaboración del prototipo con la metodología RUP y el Framework Cake PHP.....	174
5.2.	Variable: El grado de conformidad del diseño con los requerimientos y especificaciones de los usuarios	179
6.	CONCLUSIONES.....	182
7.	RECOMENDACIONES.....	183
8.	REFERENCIAS BIBLIOGRÁFICAS	183

ÍNDICE DE TABLAS

Tabla 01: Colegios de Casa Grande	16
Tabla 02: Modelo de Caso de Uso del Negocio Realizar Admisión y Matrícula	43
Tabla 03: Modelo de Caso de Uso de Negocio Programar Horario.....	44
Tabla 04: Modelo de Caso de Uso el Negocio Realizar Ejecución de Clases	44
Tabla 05: Modelo de Caso de Uso del Negocio Realizar Evaluación de Notas.....	45
Tabla 06: Promover y Egresar Alumnos	46
Tabla 07: Declaración del Problema	52
Tabla 08: Declaración Posicionamiento del Producto.....	52
Tabla 09: Resumen de los Stakeholder	53
Tabla 10: Predecesores y Prioridades.....	55
Tabla 11: Necesidades de los Stakeholders.....	57
Tabla 12: Puntuación de Atributos No Observables por Expertos.....	175
Tabla 13: Puntuación de Atributos Observables por Expertos.....	177
Tabla 14: Resultados de Encuesta	179
Tabla 15: Conteo de Resultados de Tabla 14	179

ÍNDICE DE FIGURAS

Figura 01: Elementos de un Sistema de Información Gerencial	21
Figura 02: Ciclo de Vida de RUP.....	23
Figura 03: Flujo de Procesos de Cake PHP	25
Figura 04: Proceso Actual de Admisión y Matrícula.....	30
Figura 05: Proceso Actual de Programación Académica.....	31
Figura 06: Proceso actual de Ejecución de clases	32
Figura 07: Proceso actual de Evaluación	33
Figura 08: Proceso Actual de Promocion y Egresos	34
Figura 09: Proceso Solucionador de Admision y Matricula	37
Figura 10: Proceso Solucionador de Programación Académica	38
Figura 11: Proceso Solucionador de Ejecución de Clases	39
Figura 12: Proceso Solucionador de Evaluación.....	40
Figura 13: Proceso solucionador de Promocion y Egresos	41
Figura 14: Modelo de Casos de Uso de Requerimientos	42
Figura 15: Modelo de objetos del Negocio de Admisión y Matrícula.....	46
Figura 17: Modelo de Objetos del Negocio de Ejecución de Clases	48
Figura 18: Modelo de Objetos del Negocio de Evaluación de Estudiantes	49
Figura 19: Modelo de Objetos del Negocio de Promoción y Egreso de alumnos	50
Figura 20: Modelo de Dominio del Problema.....	51
Figura 22: Caso de Uso de Requerimiento de Programación Académica	59
Figura 23: Caso de Uso de Requerimientos de Ejecución de clases.....	60
Figura 24: Caso de Uso de Requerimientos de Evaluación de estudiantes.....	61
Figura 25: Caso de Uso de Requerimiento de Egreso de Alumnos	62
Figura 26: Diagrama de Actividad Registrar Solicitud de Admisión	73
Figura 27: Diagrama de Actividad Registrar Alumno	74
Figura 28: Diagrama de Actividad Registrar Matricula.....	75
Figura 29: Diagrama de Actividad Evaluar Solicitud	76
Figura 30: Diagrama de Actividad Registrar Docente	77
Figura 31: Diagrama de Actividad Registrar aula.....	78
Figura 32: Diagrama de Actividad Registrar Curso.....	79
Figura 33: Diagrama de Actividad Registrar Plan de Estudios.....	80
Figura 34: Diagrama de Actividad Registrar Horario	81
Figura 35: Diagrama de Actividad Registrar disponibilidad horario	82
Figura 36: Diagrama de Actividad Aprobar Horario	83
Figura 37: Diagrama de Actividad Registrar Asistencia Docente	84

Figura 38: Diagrama de Actividad Consultar tema de Clase	85
Figura 39: Diagrama de Actividad Registrar Asistencia de Alumno.....	86
Figura 40: Diagrama de Actividad Registrar tema de Clase	87
Figura 41: Diagrama de Actividad Registrar Evaluación	88
Figura 42: Diagrama de Actividad Registrar Nota.....	89
Figura 43: Diagrama de Actividad Registrar Requisitos de Promoción	90
Figura 44: Diagrama de Colaboración Registrar Solicitud de Admisión.....	91
Figura 45: Diagrama de Colaboración Registrar alumno	91
Figura 46: Diagrama de Colaboración Registrar Matricula	92
Figura 47: Diagrama de Colaboración Evaluar Solicitud	92
Figura 48: Diagrama de colaboración registrar docente	93
Figura 49: Diagrama de Colaboración Registrar Aula.....	93
Figura 50: Diagrama de Colaboración Registrar Curso	94
Figura 51: Diagrama de Colaboracion Registrar Plan de Estudios	94
Figura 52: Diagrama de Colaboración Registrar Horario	95
Figura 53: Diagrama de colaboración Registrar Disponibilidad Horaria	96
Figura 54: Diagrama de Colaboración Aprobar Horario	96
Figura 55: Diagrama de Colaboración Registrar Asistencia Docente.....	96
Figura 56: Diagrama de colaboración Consultar Tema de clase.....	97
Figura 57: Diagrama de Colaboración Registrar Asistencia de Alumnos	97
Figura 58: Diagrama de colaboración Registrar Clase.....	97
Figura 59: Diagrama de Colaboración Registrar Evaluación.....	98
Figura 60: Diagrama de Colaboración Registrar Nota.....	98
Figura 61: Diagrama de Colaboracion Registrar Requisitos de Promocion de alumnos.....	98
Figura 62: Diagrama de Secuencia Registrar Solicitud de Admisión.....	99
Figura 63: Diagrama de Secuencia Registrar Alumno.....	100
Figura 64: Diagrama de Secuencia Registrar Matrícula	101
Figura 65: Diagrama de Secuencia Evaluar Solicitud.....	102
Figura 66: Diagrama de Secuencia Registrar Docente	103
Figura 67: Diagrama de Secuencia Registrar Aula	104
Figura 68: Diagrama de Secuencia Registrar Curso	105
Figura 69: Diagrama de Secuencia Registrar Plan de Estudios	106
Figura 70: Diagrama de Secuencia Registrar Horario	107
Figura 71: Diagrama de Secuencia Registrar Disponibilidad Horaria.....	108
Figura 72: Diagrama de Secuencia Aprobar Horario.....	109
Figura 73: Diagrama de Secuencia Registrar Asistencia de Docente	110

Figura 74: Diagrama de Secuencia Consultar Tema de Clase	111
Figura 75: Diagrama de Secuencia Registrar Asistencia de Docente	112
Figura 76: Diagrama de Secuencia Registrar Clase	113
Figura 77: Registrar Método de Evaluación	113
Figura 78: Diagrama de Secuencia Registrar Nota	114
Figura 79: Diagrama de Secuencia Registrar Requisitos de Promoción.....	114
Figura 80: Modelo Físico de la Base de Datos	115
Figura 81: Pantalla Registrar Postulante	116
Figura 82: Pantalla Listar Postulante	117
Figura 83: Pantalla Registrar Apoderado	118
Figura 84: Pantalla Listar Apoderado	119
Figura 85: Pantalla Registrar Solicitud de Matricula	120
Figura 86: Listar Solicitudes de Matricula	121
Figura 87: Pantalla Registrar Alumno.....	122
Figura 88: Listar Alumno.....	123
Figura 89: Pantalla Registrar Matricula	124
Figura 90: Pantalla Registrar Aula	125
Figura 91: Pantalla Registrar Curso	126
Figura 92: Registrar Docente	127
Figura 93: Pantalla Registrar Grado.....	128
Figura 94: Pantalla Registrar Sección	129
Figura 95: Pantalla Registrar Horario	130
Figura 96: Pantalla Registrar Nivel.....	131
Figura 97: Pantalla Registrar Clase.....	132
Figura 98: Pantalla Registrar Asistencia Alumno	133
Figura 99: Pantalla Registrar Plan de Estudio.....	134
Figura 100: Pantalla Registrar Requisitos de egreso.....	135
Figura 101: Registrar Evaluaciones y Notas	136
Figura 102: Diagrama de Componentes.....	137
Figura 103: Diagrama de Despliegue.....	137
Figura 104: Servidor Xampp.....	139
Figura 105: Línea de Comandos de Windows – Direccion a la ruta htdocs	139
Figura 106: Creación del Proyecto.....	140
Figura 107: Instalación del Framework CakePHP.....	140
Figura 108: Instalación de Paquetes del Framework CakePHP.....	141
Figura 109: Otorgación de Permisos Especiales al Proyecto.....	141
Figura 110: Otorgación de Permisos Especiales al Proyecto.....	142

Figura 111: Pantalla Inicial del Proyecto	142
Figura 112: Proyecto No Conectado a la Base de Datos.....	143
Figura 113: Proyecto SCDJ y sus subcarpetas	143
Figura 114: Ubicación del archivo app.php	144
Figura 115: Sección Fuente de datos	144
Figura 116: Mensaje de conexión correcta a la base de datos	145
Figura 117: Carpeta src con los archivos del proyecto	145
Figura 118: Controladores del Proyecto.....	146
Figura 119: Modelos del Proyecto	146
Figura 120: Pantallas del Proyecto.....	147
Figura 121: Archivos de operaciones con la base de datos del proyecto.....	147
Figura 122: Ejecución del comando “cake bake”	148
Figura 123: Creación del modelo Aula	149
Figura 124: Creación del controlador Aula.....	149
Figura 125: Creación de la pantalla Aula.....	150
Figura 126: Gráfica de Puntuación de Atributos No Observables por Expertos	176
Figura 127: Gráfica de Puntuación de Atributos Observables por Expertos	178
Figura 128: Pregunta 1	180
Figura 129: Pregunta 2	180
Figura 130: Pregunta 3	181
Figura 131: Pregunta 4	181
Figura 132: Pregunta 5	182

1. INTRODUCCIÓN

1.1. Realidad problemática

El distrito de Casa Grande se encuentra ubicado en la provincia de Ascope departamento de La Libertad, tiene una población de 31174 habitantes según datos de la INEI en el año 2015.

Actualmente cuenta con 14 colegios de los cuales 4 son públicos y 10 son privados, como consta en la Tabla 01:

COLEGIO	CARÁCTER
81506 Cesar Vallejo	Público
81516 Libertad	Público
Americana	Privado
Arquímedes	Privado
Casa Grande	Público
Cristo Rey	Privado
Jorge Chávez	Privado
Liceo Roma	Privado
Sagrado Corazón de Jesús	Privado
San Jorge	Privado
San Juan Bautista	Privado
Santa María Reina	Privado
Santa Teresita de Jesús	Privado
Víctor Raúl Haya de la Torre	Público

Tabla 01: Colegios de Casa Grande

Fuente: (Ministerio de Educación, 2016)

La Gestión académica de los colegios del distrito se realiza de acuerdo a los estatutos de la UGEL de Ascope. El manejo de su información en la mayoría de centros educativos se realiza en forma manual, en otros a través de hojas Excel y en muy pocos a través de sistemas informáticos.

El proceso de Gestión Académica está conformada por los subprocesos de:

- Admisión de nuevos estudiantes
- Planificación académica
 - ✓ Definición de Planes de estudios
 - ✓ Asignación docentes
 - ✓ Asignación de aulas de clase
 - ✓ Creación de horarios
- Matricula del estudiante
- Pagos de pensiones

- Registro de clases
 - ✓ Registro de sílabos
 - ✓ Registro de sesiones de clases
 - ✓ Registro de reuniones con tutores de estudiantes
- Registro de evaluaciones
- Determinación de egresados

El colegio privado “Sagrado Corazón de Jesús” no cuenta con un Sistema Integrado de Gestión Académica que permita responder de manera rápida y eficiente a los problemas de requerimientos de información por parte de los estudiantes, docentes, administrativos y autoridades.

1.2. Delimitación del problema

Para el estudio se va tomar en cuenta el proceso de Gestión Académica de la IEP “Sagrado Corazón de Jesús” de Casa Grande en el año 2016.

1.3. Características y análisis del problema

En una observación directa in situ realizada por el investigador, se ha identificado los siguientes problemas:

- *Redundancia de información*
Información almacenada en archivos Excel, por lo que no se encuentran relacionadas entre sí, y existiendo redundancia de datos
- *Pérdida de información*
Pérdida de información académica de años anteriores, pues la información se almacena cada año en archivos Excel, conllevando a que por error se pueda eliminar uno de dichos archivos.
- *Falta de seguridad de la información*
No existe seguridad de la información cualquier usuario que accede a la PC puede copiar o eliminar los archivos.
- *Carencia de información para toma de decisiones*
Para solicitar información resumen, se requiere de mucho trabajo, pues se encuentra en varios archivos y toma mucho tiempo.

1.4. Formulación del Problema

¿Cómo diseñar un prototipo del Sistema Integrado de Gestión Académica para el colegio Privado “Sagrado Corazón de Jesús” utilizando Tecnologías de Información?

1.5. Formulación de la Hipótesis

El uso de la metodología RUP y el framework Cake PHP permite el diseño de un sistema Integrado de para el proceso de Gestión Académica de la Institución Educativa Privada “Sagrado Corazón de Jesús”.

1.6. Objetivos del estudio

Objetivo General:

Diseñar un Prototipo del Sistema Integrado de Gestión Académica para la Institución Educativa Privada “Sagrado Corazón De Jesús” de Casa Grande utilizando la metodología RUP Y el framework Cake PHP.

Objetivos Específicos:

- ✓ Realizar el Modelado del Negocio y de Requerimientos aplicando la metodología RUP.
- ✓ Elaborar el Análisis y el Diseño aplicando la metodología RUP.
- ✓ Hacer la Construcción del prototipo utilizando el framework Cake PHP.
- ✓ Efectuar la Validación del prototipo.

1.7. Justificación del Estudio

Los Sistemas de Información ocupan un lugar determinante en cualquier área del quehacer humano, permitiéndoles reducir costos de operación y seguridad de información. Este prototipo servirá como un punto de partida para “informatizar” a la Institución y permitirle entre mediano y largo plazo obtener beneficios como ahorro de costos, rapidez en los procesos, control más eficiente de la información, etc.

1.8. Limitaciones del estudio

- El tiempo para desarrollar el proyecto.
- No hay apoyo financiero ni ningún tipo de patrocinio

2. MARCO TEÓRICO

2.1. Antecedentes

Gualacata (2012) en su investigación “Desarrollo del Sistema de Gestión Académica para la Escuela ‘Gonzalo Rubio Orbe’ de Otavalo” se propuso como objetivos: Establecer una estructura adecuada de información de los alumnos; Desarrollar una interfaz sencilla que permita ingresar información

al sistema; Facilitar los procesos de gestión de notas y matrícula; y ofrecer un Acceso a información de manera rápida y eficaz, utilizando SQL Server 2005 y Visual Studio 2010. Su trabajo concluyó con una evaluación de criterios a los futuros usuarios del sistema, en la que asegura que la mayoría de los parámetros fueron cumplidos satisfactoriamente con un porcentaje del SI 85%, mientras que el NO tiene un 6%, debido a que algunos usuarios no tienen conocimiento de computación que le facilite el manejo del sistema. Como aportes podemos mencionar que el autor logro crear los Módulos de: matrícula, de gestión de notas y de usuario, que son importantes para dicha institución porque les permite la administración automatizada de procesos académicos tanto para los docentes y personal administrativos, procesos fundamentales para la gestión académica.

Méndez (2012) en su investigación “Sistema de Gestión Académica para la Unidad Educativa ‘Manuel Guerrero’”, tuvo como objetivos: Desarrollar los módulos de: matrícula, registro de calificaciones, control de asistencias, envío de correo electrónicos; y crear un foro de tareas para los alumnos, usando la metodología de James Senn. Lenguaje PHP y Mysql. Su trabajo concluyó que dicho sistema web era necesario para la institución, ya que le ofrecía tener toda su información correctamente almacenada en una base de datos, y estar a disposición de toda la comunidad educativa, en especial para la Dirección. El principal aporte es que este sistema se desarrolló con la intención de convertirlo en una fuente vital para la toma de decisiones para la dirección del colegio, el cual es un proceso clave en cualquier organización para su crecimiento a futuro.

Díaz, Duran y Quijano (2010) en su trabajo “Sistema de Información para Gestión Académica”, se propusieron como objetivos: Permitir la parametrización de lineamientos para ejecutar actividades académicas; Realizar seguimiento a la actividad académica del estudiante por medio de informes de notas y actas; y Facilitar la toma de decisiones en procesos académicos y administrativos por medio de reportes. Como principal aporte menciona que la etapa de análisis es importante para el desarrollo del sistema, ya que el éxito del desarrollo de un sistema depende del cumplimiento de las necesidades y los requerimientos de los usuarios del futuro sistema.

Rengifo, P (2007), en su trabajo “Sistema de Información para la Administración de un Colegio”, propuso como objetivos: mostrar el análisis, diseño, desarrollo e implementación de un Sistema de Información para la administración de los procesos básicos de un colegio, utilizando para ello metodología orientada a objetos. Sus conclusiones fueron que el sistema usa herramientas libres, como el gestor de Base de Datos PostgreSQL, para reducir el costo de adquisición de licencias de otras herramientas como las de Windows. También el sistema esta desarrollado en lenguaje Java, y que por

tal motivo también podía funcionar en entorno Linux. Aporta que un sistema debe tener seguridad en su acceso, por lo que solo se autoriza el ingreso a usuarios registrados previamente.

Romero, R. (2012) en su trabajo “Análisis, Diseño e Implementación de un Sistema de Información aplicado a la Gestión Educativa en Centros de Educación Especial” se propuso como objetivos: Elaborar el análisis y diseño del sistema; Seleccionar y definir la arquitectura; Elaborar un modelo de base de datos relacional; Diseñar una Interfaz gráfica amigable e intuitiva; Definir el esquema de seguridad bajo el cual se hará uso del sistema de información a implementar. Concluyó que El monitoreo continuo del cronograma de proyecto y de la estructura de descomposición del trabajo posibilitó el cumplimiento de los tiempos estipulados, y la incorporación de buenas prácticas y de la metodología AUP en las etapas de construcción de software permitieron cumplir con los tiempos de entrega. Como principales aportes menciona que ASP.NET WebForms permitió una mejor implementación de funcionalidades desde una interfaz gráfica intuitiva, y la arquitectura en capas ofrece una mejor escalabilidad para futuras integraciones con nuevas herramientas y servicios aplicando la reutilización de componentes

2.2. Bases teóricas

2.2.1. Gestión Académica

Paredes y Enríquez (2010) definen la Gestión académica así:

“consiste en potenciar una educación de calidad caracterizada por prácticas creativas e innovadoras, pero estas acciones conllevan un aumento en la cuota de responsabilidad de cada actor de una reestructuración y cambio de visión y enfoque educativo” (p.66).

Rodriguez (2010) dice que:

“La gestión académica es un proceso complejo que involucra la entrada de recursos diversos, un procesamiento de la complejidad más elevada que pueda existir, y genera salidas bajo la forma de productos de alta complejidad como: nuevos conocimientos, profesionalidad, habilidades cognoscitivas, investigativas, capacidades de solución en el descubrimiento, formulación, planteamiento y resolución de problemas profesionales” (párr. 6).

2.2.2. Sistema de Información Gerencial

Tutorials Point (2014) dice que un Sistema de Información Gerencial o 'MIS' es un “sistema planificado de recogida, almacenamiento y

difusión de datos en forma de información necesaria para llevar a cabo las funciones de gestión” (p.16)

Se compone de elementos tal como consta la siguiente figura (Figura 01):



Figura 01: Elementos de un Sistema de Información Gerencial

Fuente: (Tutorials Point, 2014)

Según Tutorials Point (2014) los siguientes son los objetivos básicos de un MIS (p. 17):

- **Captura de datos:** la captura de datos contextuales, o la información operativa que contribuya en la toma de decisiones procedente de diversas fuentes internas y externas de organización
- **Procesamiento de Datos:** los datos capturados se procesan en información necesaria para la planificación, organización, coordinación, dirección y control de las funcionalidades a nivel estratégico, táctico y operativo.

El procesamiento de Datos significa:

- Hacer cálculos con los datos
 - Ordenamiento de datos
 - Clasificación de datos
 - Resumen de datos
- **Almacenamiento de información:** La información o datos procesados deben ser almacenados para su uso en el futuro.
 - **Recuperación de Información:** el sistema debe ser capaz de recuperar esta información desde el almacenamiento cuando sea requerido por varios usuarios.
 - **Propagación de la información:** la información o el producto terminado del MIS deben distribuirse a sus usuarios periódicamente utilizando la red de la organización.

Características de un MIS:

- Debe basarse en una planificación a largo plazo.
- Se debe proporcionar una visión integral de la dinámica y estructura de la organización.
- Se debe trabajar como un sistema completo e integral que abarque todos los subsistemas interconectados dentro de la organización.
- Se debe planificar de “arriba hacia abajo”, como los responsables de la toma de decisiones o la gerencia, que deben participar activamente y proporcionar una dirección clara en la etapa de desarrollo del MIS.
- Debe basarse en las necesidades de información estratégica, táctica y operativa de los gerentes de una organización.
- También debe tener cuidado de situaciones excepcionales informando esas situaciones.
- Debe ser capaz de hacer previsiones y cálculos, y generar información avanzada, proporcionando así una ventaja competitiva. Los tomadores de decisiones pueden tomar acciones en base a tales predicciones.
- Se debe crear vínculos entre todos los subsistemas dentro de la organización, de manera que los tomadores de decisiones pueden tomar la decisión correcta basada en visión integrada.
- Se debe permitir un fácil flujo de información a través de diversos subsistemas, evitando así la redundancia y duplicidad de datos. Se debe simplificar las operaciones de con la mayor practicidad como sea posible.
- Aunque el MIS es un sistema integrado, completo, debe hacerse de una manera tan flexible que se puede dividir fácilmente en sub-sistemas más pequeños como y cuando sea necesario.
- Una base de datos central es la columna vertebral de un SIG bien construido.

2.2.3. RUP

Según IBM (1998). RUP:

“es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo enfocada hacia los casos de uso, manejo de riesgos y el manejo de la arquitectura.” (p. 3). (IBM, 1998)

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software (p.4).

Ademas, IBM (1998) señala que en el ciclo de vida RUP veremos una implementación del desarrollo en espiral. Con el ciclo de vida se establecen tareas en fases e iteraciones. El RUP maneja el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base de inicio

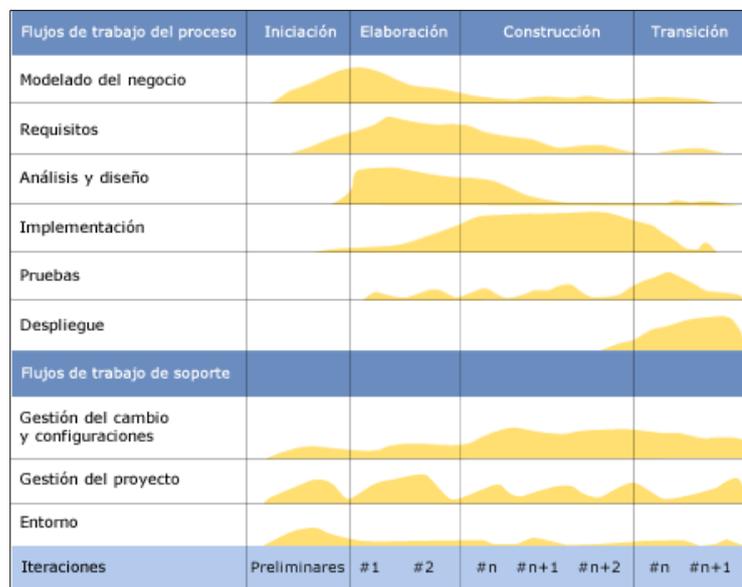


Figura 02: Ciclo de Vida de RUP

Fuente: (IBM, 1998)

2.2.4. Framework Cake PHP

Según la web de Cake Software Foundation (Foundation, 2012), dice que CakePHP es un marco de desarrollo rápido para PHP, libre, de código abierto. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web. Nuestro principal objetivo es que puedas trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

Con CakePHP el desarrollo web ya no es monótono porque ofrecemos las herramientas para que empieces a escribir el código que realmente

necesitas: la lógica específica de tu aplicación. Consigue una copia de CakePHP, empieza con lo verdaderamente importante y no reinventes la rueda cada vez que te incorpores a un nuevo proyecto.

CakePHP tiene un equipo de desarrolladores y una comunidad activos, lo que añade valor al proyecto. Con CakePHP, además de no tener que reinventar la rueda, el núcleo de tu aplicación se mejora constantemente y está bien probado.

Estas son las características de CakePHP:

- Comunidad activa y amistosa
- Licencia flexible
- Compatible con PHP4 y PHP5
- CRUD integrado para la interacción con la base de datos
- Soporte de aplicación [scaffolding]
- Generación de código
- Arquitectura Modelo Vista Controlador (MVC)
- Despachador de peticiones [dispatcher], con URLs y rutas personalizadas y limpias
- Validación integrada
- Plantillas rápidas y flexibles (sintaxis de PHP, con ayudantes [helpers])
- Ayudantes para AJAX, Javascript, formularios HTML y más
- Componentes de Email, Cookie, Seguridad, Sesión y Manejo de solicitudes
- Listas de control de acceso flexibles
- Limpieza de datos
- Caché flexible
- Localización
- Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache

CakePHP incluye las clases Controlador [Controller], Modelo [Model] y Vista [View], pero también incluye otras clases y objetos que hacen que el desarrollo en MVC sea un poco más rápido y agradable. Los Componentes [Components], Comportamientos [Behaviors], y Ayudantes [Helpers] son clases que proporcionan extensibilidad y reusabilidad; agregan rápidamente funcionalidad a las clases base MVC de las aplicaciones. En la Figura 04 podremos ver mejor dichos elementos.

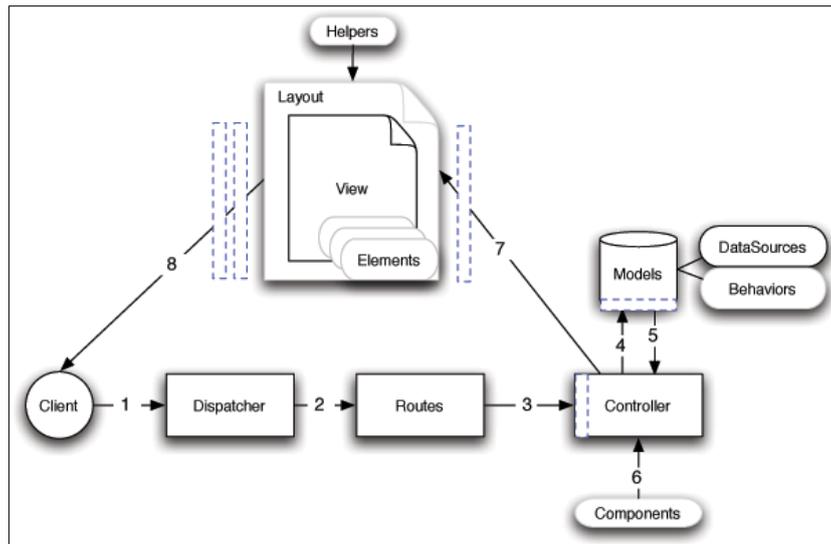


Figura 03: Flujo de Procesos de Cake PHP
Fuente: (Cake Software Foundation, 2012)

2.3. Definición de términos

2.3.1. Gestión Académica

Comprende los procesos centrales en la misión educativa, necesarios para desarrollar competencias en los estudiantes. Comprende la estructuración de planes de estudios, articulación entre grados, niveles y áreas, metodologías de enseñanza, proyectos transversales, investigación y clima del aula. (Ministerio de Educación de Colombia, 2015, párr. 4).

2.3.2. Metodología de Desarrollo

Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. (Centers For Medicare & Medicaid Services, 2015, p.1).

2.3.3. Framework

Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta (Gutiérrez, 2015, p.1).

3. MATERIAL Y MÉTODOS

3.1. Material

3.1.1. Población

Administrativos	→	4	(Director, Subdirector, Coordinador Académico y Secretaria)
Docentes	→	9	
Estudiantes	→	113	
Padres de Familia	→	78	

3.1.2. Muestra

La muestra será todos los 9 docentes y los 4 administrativos.

3.1.3. Unidad de Análisis

- Los usuarios: Los 9 docentes y los 4 administrativos.

3.2. Método

3.2.1. Nivel de Investigación

Por el propósito: Aplicado

Por la Clase de Medios: Experimental

Por el alcance: Descriptiva

3.2.2. Diseño de Investigación

La metodología aceptada para llevar a cabo el desarrollo de este proyecto comprende la siguiente secuencia:

- **Planeación:** El propósito de la etapa de planeación es producir un plan de trabajo formal para desarrollar el aplicativo de migración. El plan de trabajo es un conjunto de documentos que se utilizará para guiar y evaluar el desarrollo correcto de la investigación.
- **Implementación:** en esta fase comprende seguir los siguientes pasos basados en la metodología RUP y se procederá así:

- ❖ **Modelado de Negocio:** Recopilar Información, identificar procesos y usuarios, Modelado de Procesos, etc
- ❖ **Modelado de Requerimientos:** Listar los requerimientos para el sistema.
- ❖ **Análisis y Diseño:** Diagrama de Actividades, de Estado, de Secuencia, de Clases, de Interfaces, Diseño Físico y Lógico de la Base de Datos, Diagrama de Componentes y de Despliegue
- ❖ **Construcción:** Construcción de Prototipos, de la Base de Datos
- **Recolección de Datos:** mediante este proceso obtendremos la información necesaria para lograr los objetivos de la investigación en curso y estará apoyada con entrevistas, encuestas, y la observación.
- **Análisis de Datos:** en este proceso transformaremos la información más útil de la recolección de datos con el único propósito de sugerir conclusiones en base a los objetivos planteados.

Elaboración del Informe: este proceso final comprende resumir todo lo comprendido en las etapas anteriores en un formato establecido, con las mejoras y correcciones recomendadas.

3.2.3. Variables de estudio y operacionalización

Variable Independiente: Diseño de un Sistema Integrado de Gestión Académica de la Institución Educativa Privada “Sagrado Corazón de Jesús”.

Definición Conceptual: Es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física (Taylor, 1959; citado por Torossi, 2015).

Definición Operacional: El grado de conformidad del diseño con los requerimientos y especificaciones de los usuarios.

Variable Dependiente: Prototipo de Sistema Integrado de Gestión utilizando la Metodología RUP y el Framework Cake PHP.

Definición Conceptual: Es un modelo a escala de lo real, pero no necesariamente funcional del sistema final, ya que no lleva a cabo la totalidad de las funciones necesarias de éste. (Google Sites, 2016)

Definición Operacional: El grado de cumplimiento de la elaboración del prototipo con la metodología RUP y el Framework Cake PHP.

3.2.4. Técnicas e Instrumentos de recolección de datos

Siguiendo los pasos de una investigación aplicada, precisamos encontrar los medios de recolección de información para luego procesarlas, analizarlas y posteriormente extraer las conclusiones sobre la base de la información recolectada, aplicando las técnicas e instrumentos pertinentes como:

- Entrevistas
- Encuestas
- Observación Directa

3.2.5. Técnicas de Procesamiento de datos

La técnica de procesamiento para los datos obtenidos de la encuesta a los administrativos y docentes, sobre el grado de satisfacción en el uso del módulo de asistencias, será la codificación y tabulación

3.2.6. Técnicas de análisis de datos

La técnica de análisis de los datos obtenidos de la encuesta a los administrativos y docentes, sobre el grado de satisfacción en el uso del módulo de asistencias, será la prueba de medias.

4. RESULTADOS

4.1. Modelado del Negocio

4.1.1. Evaluación de la Organización

4.1.1.1.Contexto del Negocio

Sagrado Corazón de Jesús es una institución educativa privada ubicada en el distrito de Casa Grande, que ofrece servicios educativos en sus niveles de primaria y secundaria. Cuenta con 107 alumnos

Dentro de la institución se manejan varios procesos tales como registro matrículas de alumnos, pagos de conceptos, programación de horarios de clases, registro de notas de alumnos y registro de asistencia de alumnos.

Toda la información es archivada en archivos de Excel, corriendo el riesgo de perder la información documentaria. Por ello el colegio sugiere contar con un sistema de información Web para optimizar el proceso de gestión académica.

4.1.1.2.Estrategias en el contexto del Negocio

MISIÓN: Realizar un prototipado de sistema integrado de gestión académica que permita gestionar de una mejor manera los procesos académicos de la institución.

VISION: Llevar un mejor control de la gestión académica

4.1.1.3.Factores Externos

CLIENTES:

- Niños y adolescentes en edad escolar
- Padres de familia

COMPETIDORES:

- Santa María Reina
- Americana
- Jorge Chávez
- Cristo Rey
- San Jorge
- San Juan Bautista
- Santa Teresita del Niño Jesús

STAKEHOLDERS:

- Director
- Subdirector
- Secretaria
- Coordinador Académico

4.1.1.4. Factores Internos

- Director
- Subdirector
- Coordinador Académico
- Secretaria
- Docentes
- Alumnos

4.1.2. Modelo Visual de Diagnostico

4.1.2.1. Admisión y Matricula

El proceso actual de admisión y matricula se ve en la figura 04 a continuación:

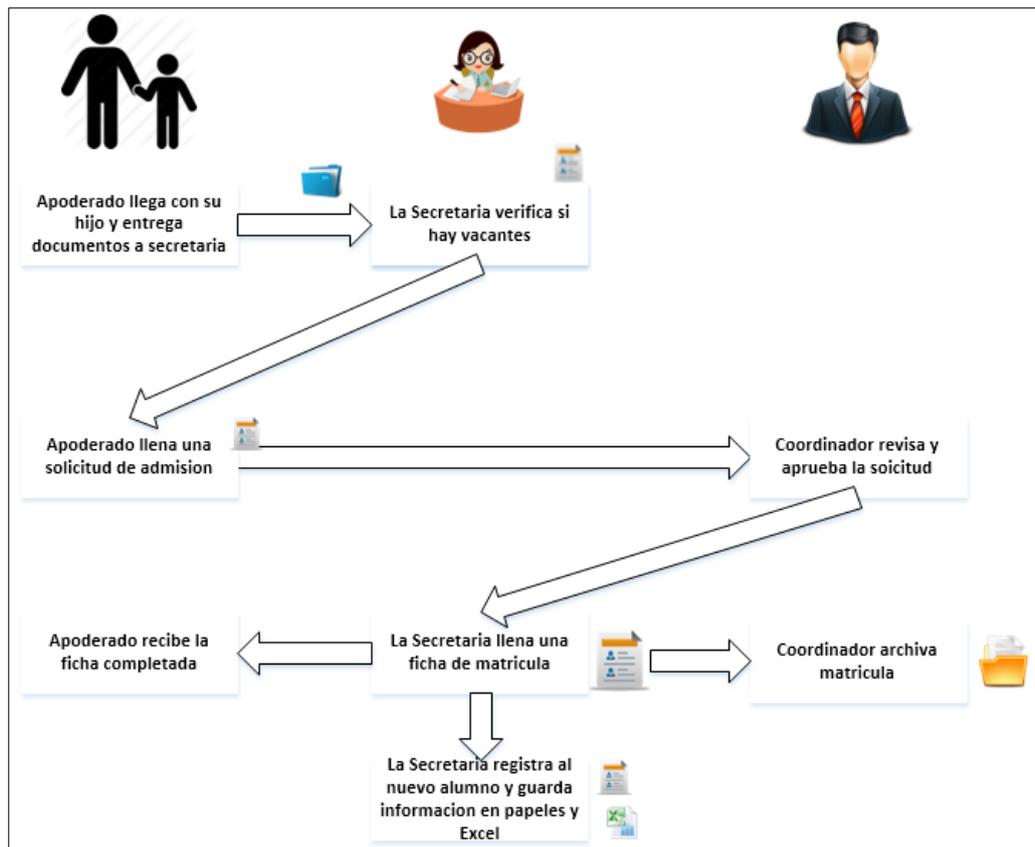


Figura 04: Proceso Actual de Admisión y Matrícula

Fuente: Elaboración Propia

4.1.2.2. Programación Académica

El proceso actual de programación académica se ve a continuación en la figura 05:

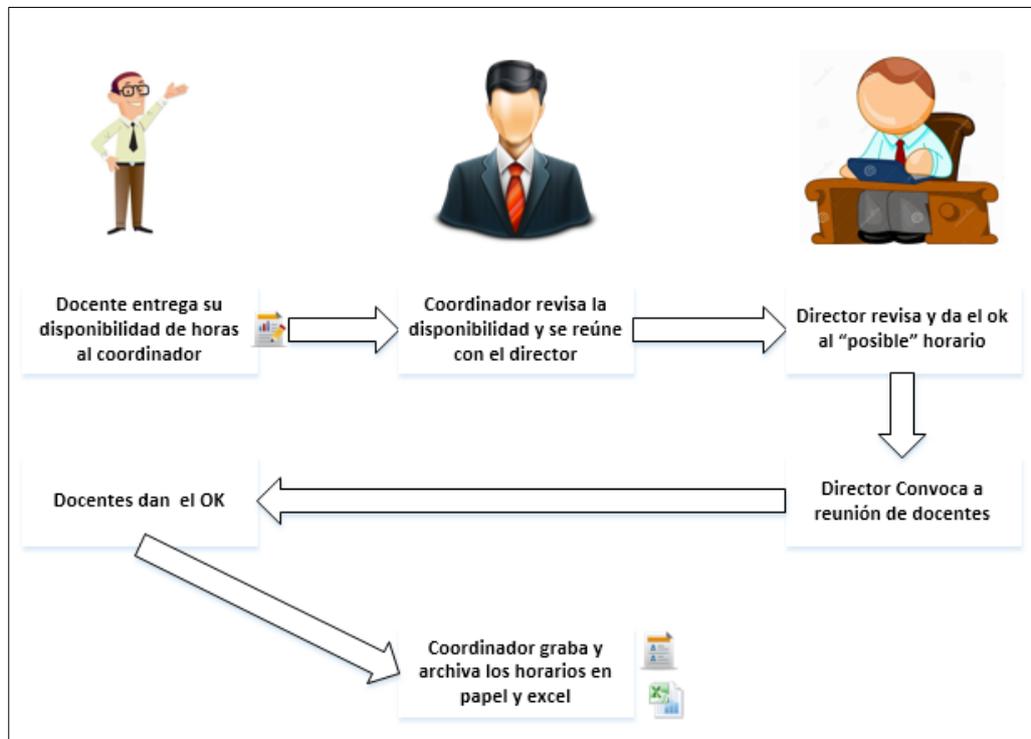


Figura 05: Proceso Actual de Programación Académica

Fuente: Elaboración Propia

4.1.2.3. Ejecución de Clases

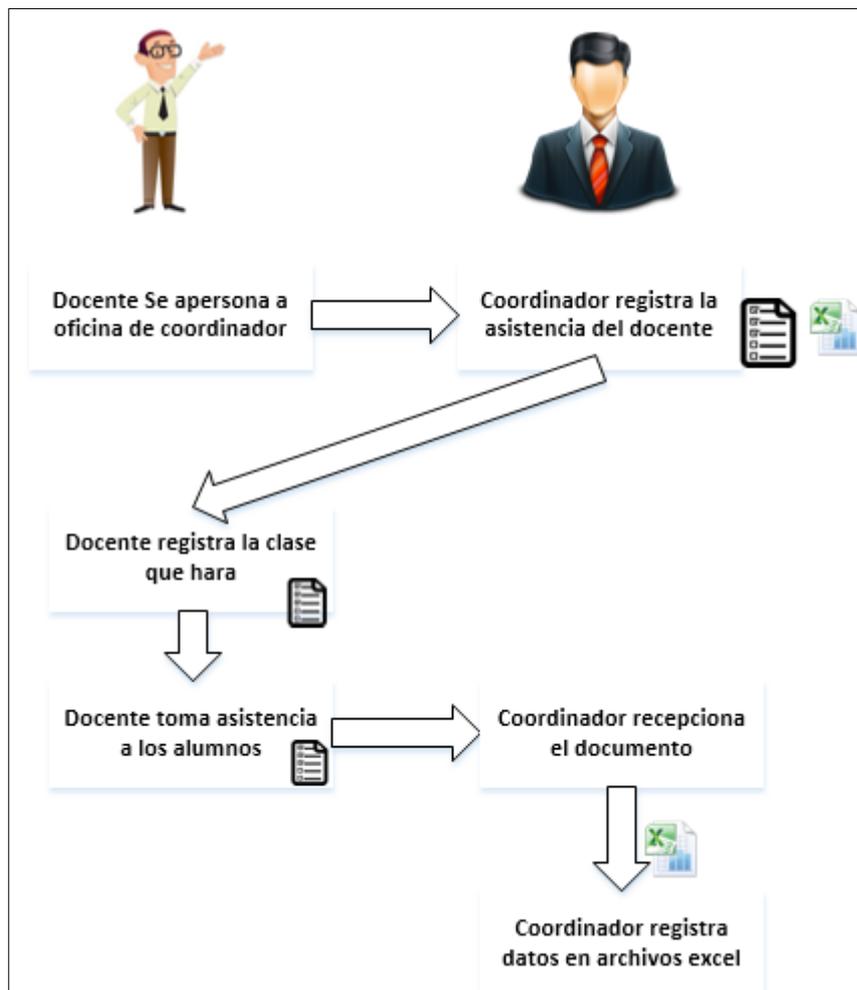


Figura 06: Proceso actual de Ejecución de clases

Fuente: Elaboración Propia

4.1.2.4.Evaluación

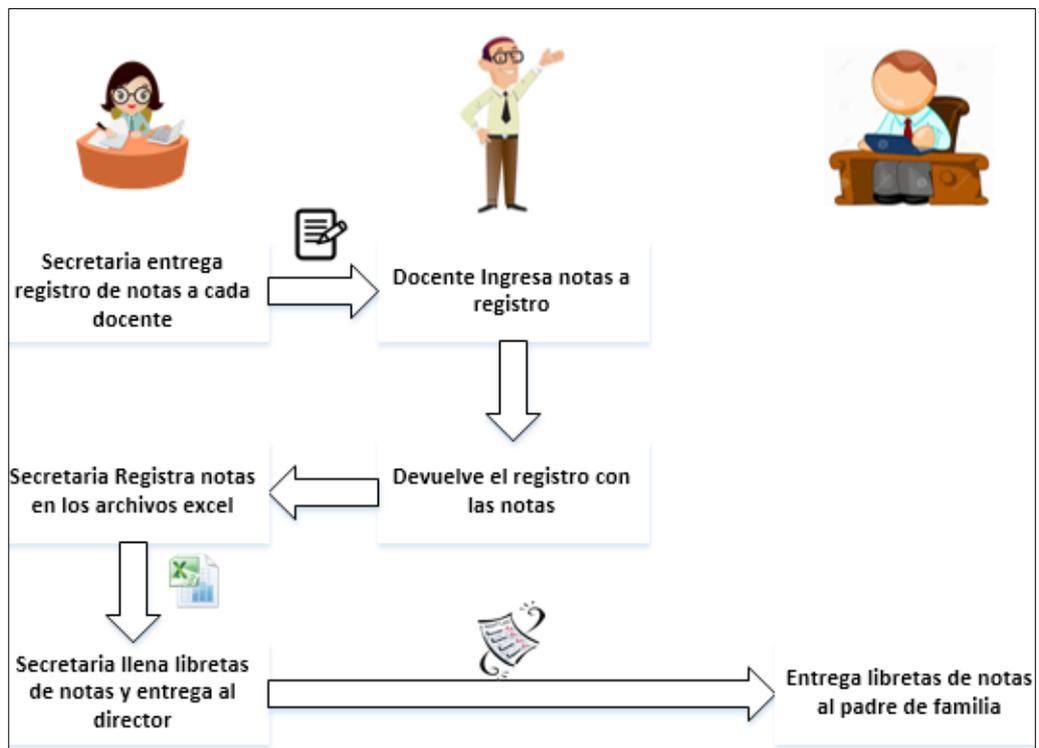


Figura 07: Proceso actual de Evaluación

Fuente: Elaboración Propia

4.1.2.5.Promover y Egresar Alumnos

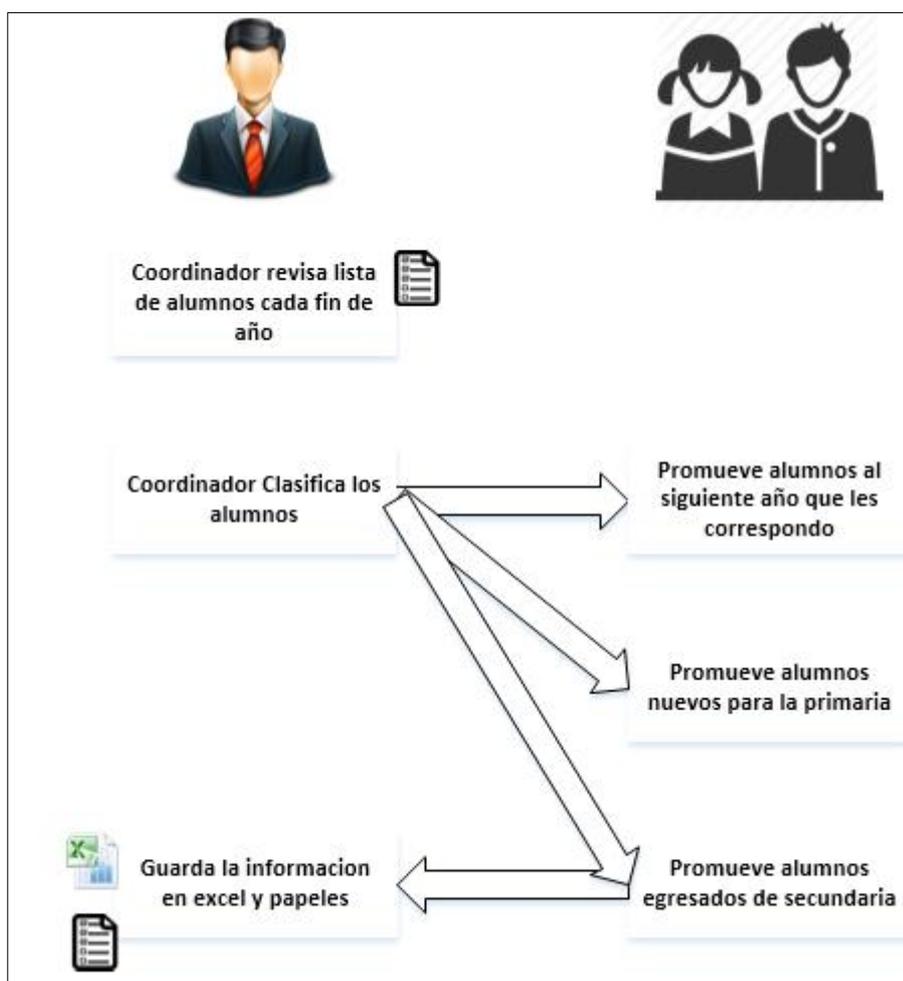


Figura 08: Proceso Actual de Promoción y Egresos

Fuente: Elaboración Propia

4.1.3. Arquitectura del Negocio

4.1.3.1.Proceso de Admisión y Matrícula

El proceso de matrícula de un alumno se inicia cuando el padre de familia o apoderado del menor se apersona en secretaria presentando la documentación respectiva de su menor hijo tales como: ficha integral del educando, partida de nacimiento, certificado de estudios, certificado de comportamiento, certificado de vacunas, fotografías y otros. La secretaria verifica tanto en sus archivos Excel, como en los papeles de su escritorio, si hay vacantes disponibles en el año solicitado para el alumno. Si no hay vacantes disponibles, el proceso termina y no es posible proceder con la matrícula. Caso contrario la secretaria realiza una búsqueda manual del alumno por nombre y apellidos tanto en los documentos apilados en su escritorio, como en los archivos Excel de su computadora. En el caso de que el alumno no este registrado (nuevo alumno), la secretaria le entrega al padre de familia una

solicitud de admisión para su hijo, donde llenará los datos del alumno y de su apoderado. Cuando el apoderado termina de llenar la solicitud, le entrega a la secretaria, la cual entrega la solicitud a su vez al coordinador académico para que lo analice. Si la solicitud es desaprobada, el proceso termina. Si la solicitud es aprobada, la secretaria registra una ficha de matrícula, para la cual rellena los campos correspondientes como son: Nro. Matrícula, fecha, grado y sección, nivel de estudios, código del alumno y condición del alumno (promovido, repitente, etc.). Si el alumno está registrado (alumno que ha estudiado el último año en la institución), entonces se procede a inscribirlo directamente para el año escolar que debe seguir. Finalmente, tanto la secretaria como el coordinador académico, en ambos casos archiva la matrícula del alumno.

4.1.3.2. Proceso de Programación Académica

El proceso de registro de horarios de clases inicia cuando los docentes entregan información al coordinador académico, sobre su disponibilidad de días y de horas para dictar clases en el colegio, además hacen llegar información sobre los cursos que pueden dictar. Luego, el coordinador se reúne con el director para armar una versión preliminar de los horarios de cada grado de estudio, tanto para el nivel primario, como para el secundario; definiendo los días, la hora, la cantidad de horas de cada curso, el aula y el docente responsable de la clase. Luego el director convoca a una reunión con la participación de todos los docentes para consultarles finalmente si están de acuerdo con el horario propuesto. De obtener conformidad por todos los docentes, el horario no sufre modificaciones y cada definido como tal.

4.1.3.3. Proceso de Ejecución De Clases

Se registra los temas a desarrollar, los trabajos dejados tanto en el aula como los dejados para casa.

El docente, al momento de empezar su clase, registra en su cuadernillo el tema que va a desarrollar de acuerdo a su silabario,

Este proceso consiste en tomar la asistencia tanto de alumnos como de docente.

Asistencia de Alumnos:

El proceso inicia cuando el docente llama por el nombre a cada alumno en cada clase. Si el alumno está presente, se le pone “Asistencia”; si no está presente, se le pone “Inasistencia”. Todo esto el docente lo apunta en su registro de asistencia, el cual es proporcionado por la dirección a cada docente. Luego, al final de la clase, el docente guarda su registro, y al final del día, entrega a la secretaria para que saque una copia de dicho papel. En caso de falta, el alumno puede presentar su justificación a la dirección mediante un documento con la información explicando la causa de su falta.

Asistencia de Docentes:

En el caso de los docentes, al momento que ellos llegan al colegio, se apersonan a la oficina del coordinador académico para firmar su asistencia sobre el casillero del curso que va a dictar en ese día, y el encargado apunta la hora y la fecha de su llegada. Lo mismo se repite cuando tenga que ir a otro salón a dictar clase de otro curso. Al final de su jornada, firma su salida de la institución. Sino se apersona a la oficina, el coordinador se apersona al aula donde tenga clase el docente para averiguar qué pasó. Esto se repite todos los días y queda archivado y guardado en los fólderes. Semanalmente, el coordinador registro la información en un archivo Excel. Los docentes también deben justificar su inasistencia, en caso la cometan, lo cual le generara irreparablemente un descuento en su sueldo, a menos que haya sido generada por enfermedad o motivos familiares importantes (por ejemplo, muerte de algún pariente). Por otra parte, el docente, en el caso del día de su cumpleaños, no está obligado a dictar la clase, y no está sujeto a descuento en su sueldo, pero esta solo se le tolerara en el mismo día de su cumpleaños, no más que ese día.

4.1.3.4. Proceso de Evaluación a los Estudiantes

Consiste en el registro de las notas, el cual se realiza mediante un registro auxiliar que entrega la secretaria a cada docente, el cual ingresa las calificaciones de cada materia y luego se lo entregan a la secretaria para que organice las notas. Se llevan dos registros: uno mensual y otro trimestral. El registro trimestral tiene tres notas las cuales son obtenidas por los estudiantes mensualmente y una nota de examen trimestral, las cuales son promediadas, y esta nota promedio trimestral, es anotada en la libreta de notas de cada alumno por la secretaria. Luego se convoca a una reunión de padres de familia, en donde el director se encarga de entregar las libretas de notas a cada padre de familia o apoderado, y estos en un plazo menor a una semana, deben devolver las libretas de notas firmadas por ellos.

4.1.3.5. Proceso de Promover y Egresar Estudiantes

El proceso de promover y egresar estudiantes comienza en las ultimas semanas de cada año escolar, en el cual el coordinador académico debe revisar la lista de alumnos: en el caso de egresados; tanto de 6 grado de primaria como del 5 grado de secundaria; en el caso de promover alumnos, a los demás grados. Verifica si todos estos alumnos ya estas aptos para figurar como nuevos egresados de Primaria y Secundaria, o como nuevos alumnos del siguiente año escolar, si previamente cumplen con los requisitos de figurar como aprobados.

4.1.4. Modelo Visual Solucionador

4.1.4.1. Admisión y Matricula

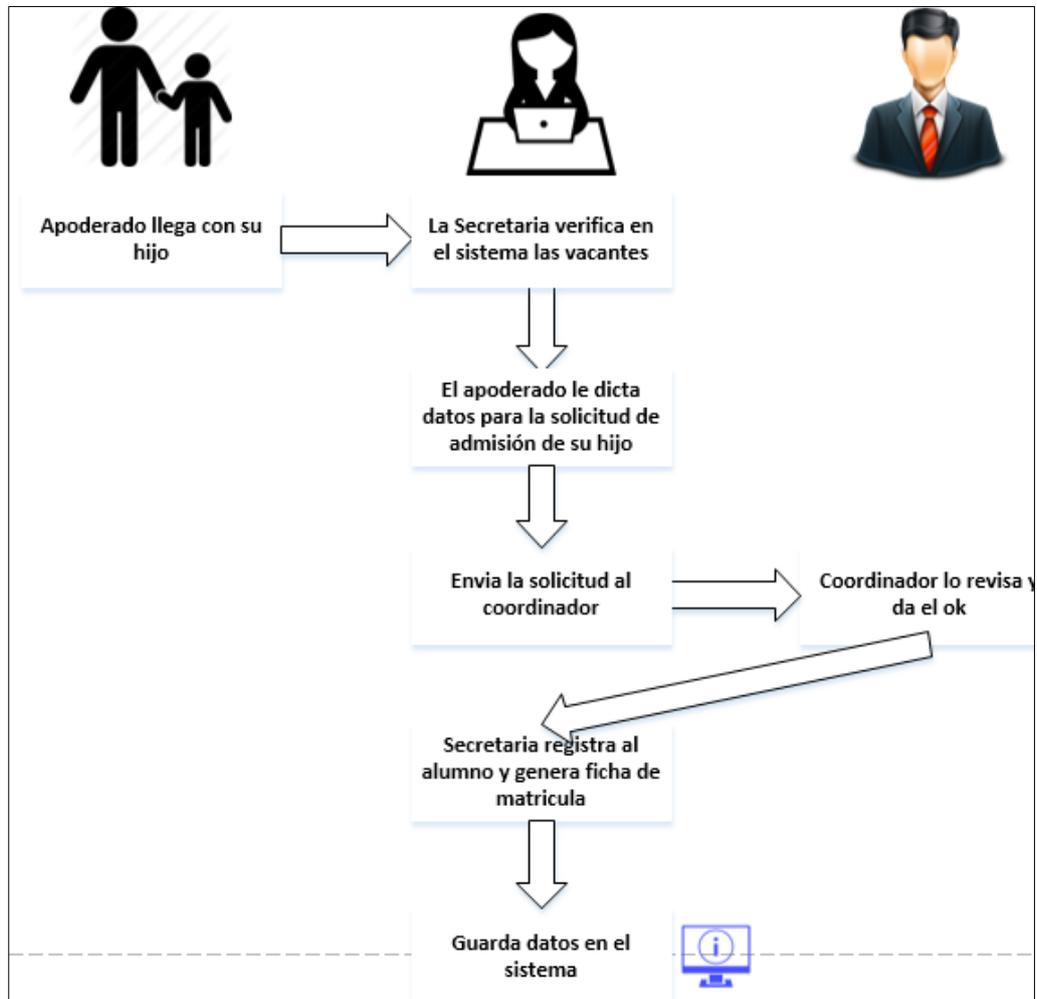


Figura 09: Proceso Solucionador de Admision y Matricula

Fuente: Elaboración Propia

4.1.4.2. Programación Académica

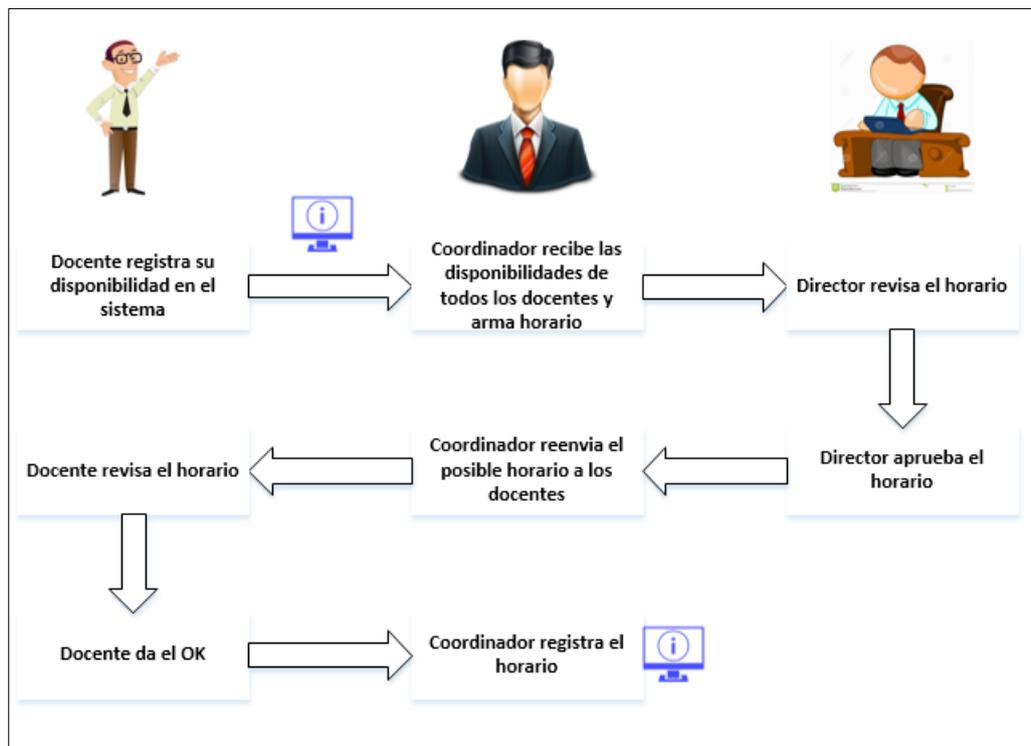


Figura 10: Proceso Solucionador de Programación Académica
Fuente: Elaboración Propia

4.1.4.3.Ejecución De Clases

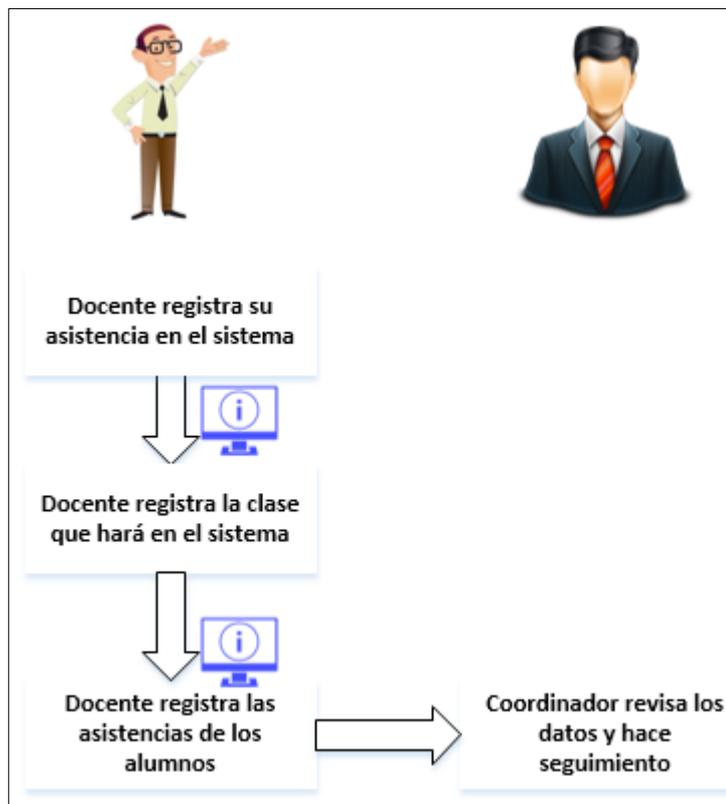


Figura 11: Proceso Solucionador de Ejecución de Clases

Fuente: Elaboración Propia

4.1.4.4.Evaluación

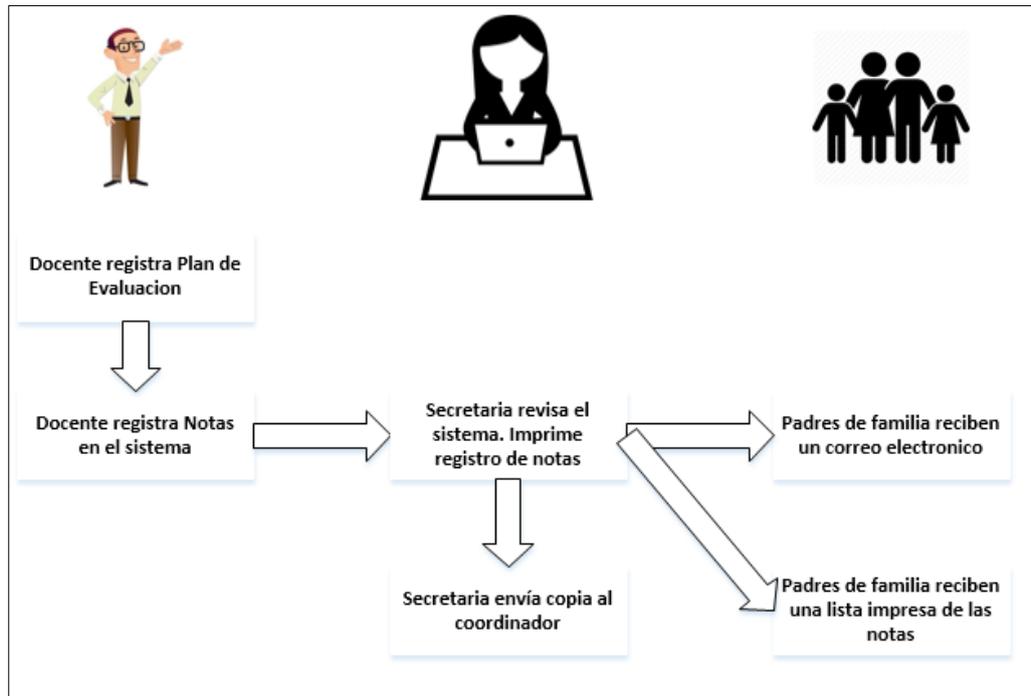


Figura 12: Proceso Solucionador de Evaluación

Fuente: Elaboración Propia

4.1.4.5.Promoción y Egresados

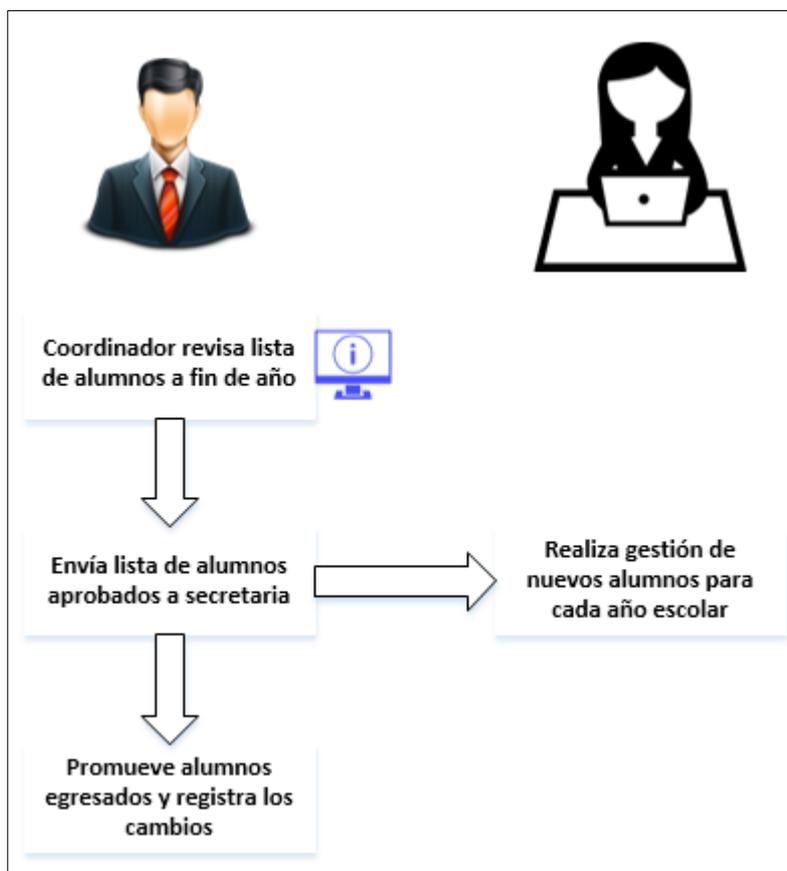


Figura 13: Proceso solucionador de Promoción y Egresos

Fuente: Elaboración Propia

4.1.5. Reglas del Negocio

- Para matricular a un nuevo alumno, primero se busca sus datos en los archivos de Excel. Si no es encontrado, se lo registra como nuevo alumno.
- Las inasistencias son injustificables tanto para alumnos como para docente, salvo los casos de enfermedad (para ambos), motivo de duelo (para ambos), o de cumpleaños (solo para docente). En el caso de los docentes, si lo hace por primera vez, se le penalizará con un descuento; si reincide, se le manda un memorándum, y si vuelve a realizarlo, se procederá a despedirlo. En el caso de los alumnos, se le pondrá como demerito en su libreta escolar, y si falta 3 veces, se llamará a sus padres de familia para obtener una explicación.
- Las notas son de 0 a 20, y se registran como numero entero en la libreta de notas. En caso de ser decimal, el 0,5 en adelante es un punto a favor del alumno. Si un alumno en su promedio final, desapueba 1 o 2 cursos, llevara estos en el verano para hacerle un reforzamiento obligado, pero

no pierde el año escolar. Si desaprueba 3 o más, automáticamente tendrá que repetir el año escolar.

- En el caso de horarios, una vez aprobados y firmados por los docentes aceptando su conformidad, no se podrá hacer cambios de este a lo largo del año escolar, salvo casos extremos similares a los explicados en las inasistencias.

4.1.6. MODELO DE CASO DE USO DEL NEGOCIO

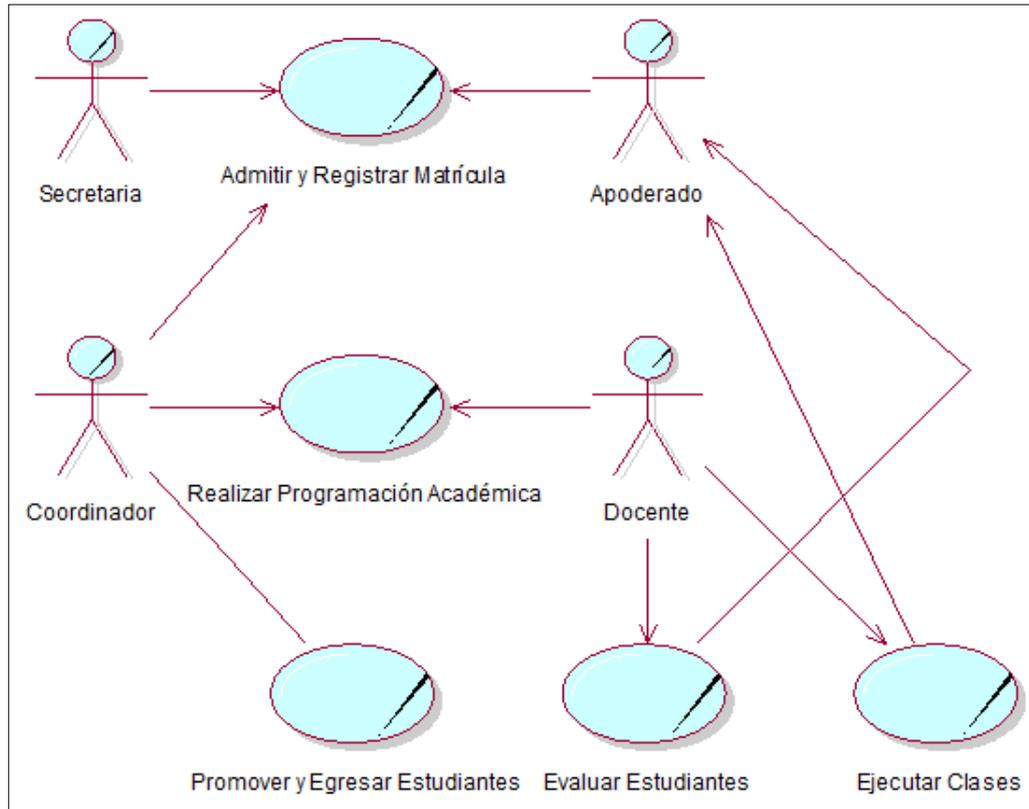


Figura 14: Modelo de Casos de Uso de Requerimientos

Fuente: Elaboración Propia

4.1.7. Especificaciones de Casos de Uso del Negocio

1.- Nombre de caso de Uso	Realizar Admisión y Matricula
2.- Objetivos	Realizar la matricula tanto de nuevos como de antiguos alumnos en la institución
3.- Flujo de Trabajo	<ul style="list-style-type: none"> - Registrar Postulante - Registrar Solicitud de Postulante - Evaluar Solicitud de postulante - Admitir Postulante - Registrar Alumno - Registrar Apoderado

	<ul style="list-style-type: none"> - Registrar Matricula - Generar Ficha de matricula
4.- Categoría	Principal
5.- Posibilidades	Permitirá guardar la información tanto del alumno como del apoderado para algún uso futuro
6.- Riesgo	No contar con la información para algún reporte
7.- Requerimientos	Fidelidad de los datos personales
8.- Propietario	Secretaria, Apoderado

Tabla 02: Modelo de Caso de Uso del Negocio Realizar Admisión y Matrícula

Fuente: Elaboración Propia

1.- Nombre de caso de Uso	Programar Horarios
2.- Objetivos	Realizar la programación académica es decir los horarios de clase
3.- Flujo de Trabajo	<ul style="list-style-type: none"> - Registrar Docente - Registrar aula - Registrar curso - Registrar Plan de Estudios por grado - Asignar Docente al curso - Asignar cursos al grado - Asignar aula al grado - Registrar horario por grado
4.- Categoría	Principal
5.- Posibilidades	Permitirá guardar la información de disponibilidad del docente Asignar nuevos cursos a un docente
6.- Riesgo	El docente se retire de la institución
7.- Requerimientos	La Disponibilidad no debe cambiar

8.- Propietario	Coordinador Académico, docente.

Tabla 03: Modelo de Caso de Uso de Negocio Programar Horario

Fuente: Elaboración Propia

1.- Nombre de caso de Uso	Realizar Ejecución de Clases
2.- Objetivos	Realizar la ejecución de clases así como las asistencias
3.- Flujo de Trabajo	<ul style="list-style-type: none"> - Registrar asistencia de docente - Registrar inasistencia de docente - Registrar tema de clase - Registrar tipo de clase (aula, laboratorio, campo) - Registrar Asistencia de alumnos - Registrar inasistencias de alumnos - Consultar todos los datos anteriores
4.- Categoría	Principal
5.- Posibilidades	<p>Tener información de las asistencias de los alumnos</p> <p>Saber que sesiones de clases se hicieron a lo largo del año</p> <p>Tener información de las asistencias de los docentes y calcular sus pagos</p>
6.- Riesgo	<p>El docente no pueda ingresar al sistema</p> <p>El docente falte a clases</p>
7.- Requerimientos	La Disponibilidad no debe cambiar
8.- Propietario	Docente, Coordinador Académico

Tabla 04: Modelo de Caso de Uso el Negocio Realizar Ejecución de Clases

Fuente: Elaboración Propia

1.- Nombre de caso de Uso	Realizar Evaluación de notas
2.- Objetivos	Realizar la evaluación y registro de notas

3.- Flujo de Trabajo	<ul style="list-style-type: none"> - Registrar requisitos para aprobar o desaprobar. - Registrar tipo de evaluación - Registrar metodología de evaluación. - Registrar notas de cada alumno. - Registrar cambios de notas. - Enviar notas finales - Listar notas por alumno - Imprimir libreta de notas del alumno - Enviar emails a los padres de familia
4.- Categoría	Principal
5.- Posibilidades	<p>Los padres de familia conocerán las notas de sus hijos más oportunamente.</p> <p>Se tendrá información exacta para promover los alumnos</p>
6.- Riesgo	<p>El docente no pueda ingresar al sistema</p> <p>El docente falte a clases</p> <p>La secretaria no imprima los registros</p>
7.- Requerimientos	La Disponibilidad no debe cambiar
8.- Propietario	Docente. Secretaria

Tabla 05: Modelo de Caso de Uso del Negocio Realizar Evaluación de Notas

Fuente: Elaboración Propia

1.- Nombre de caso de Uso	Promover y Egresar alumnos
2.- Objetivos	Realizar la promoción de alumnos a nuevos años escolares, y la salida de los que acabaron primaria y/o secundaria
3.- Flujo de Trabajo	<ul style="list-style-type: none"> - Registrar requisitos para promover o egresar - Listar alumnos a fin de año escolar. - Listar Alumnos que cumplan los requisitos. - Clasificar alumnos según sean promovidos o egresados. - Registrar alumnos promovidos y egresados.

4.- Categoría	Principal
5.- Posibilidades	Conocer los nuevos egresados. Calcular la cantidad de alumnos de los siguientes años escolares
6.- Riesgo	El coordinador no pueda ingresar al sistema El docente no ingrese las notas a tiempo
7.- Requerimientos	La Disponibilidad no debe cambiar
8.- Propietario	Coordinador

Tabla 06: Promover y Egresar Alumnos

Fuente: Elaboración Propia

4.1.8. Modelo de Objetos del Negocio

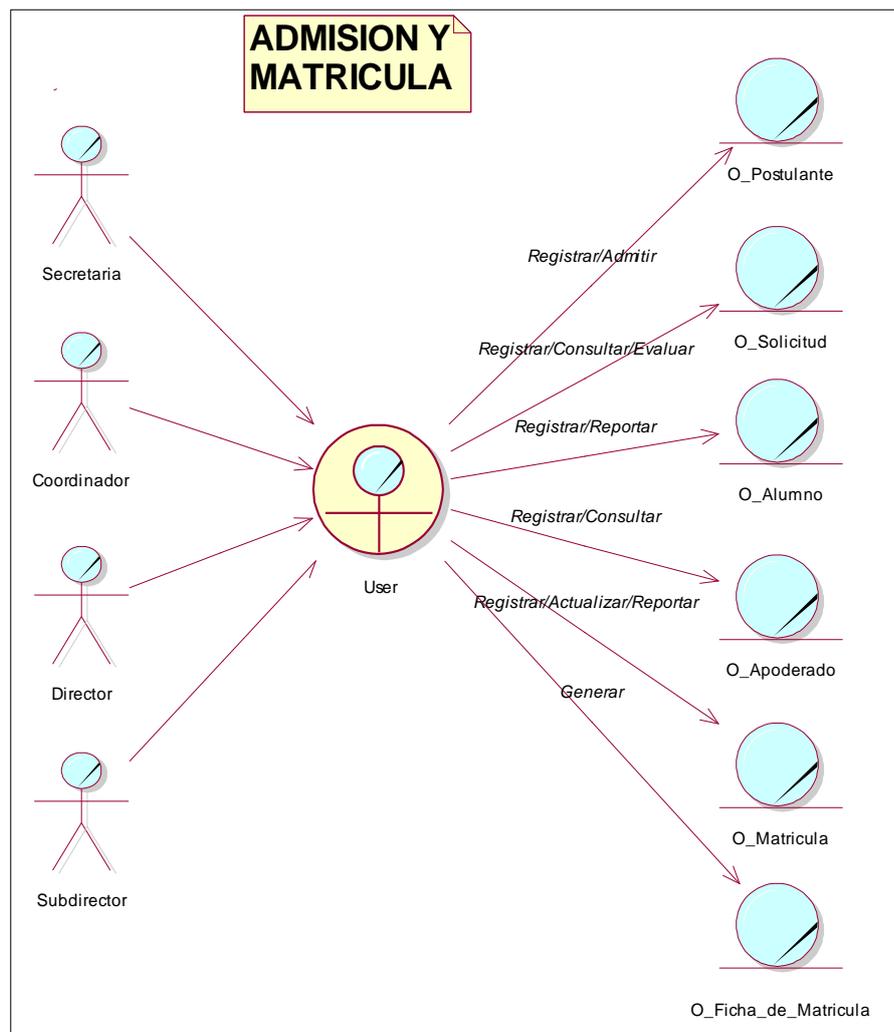


Figura 15: Modelo de objetos del Negocio de Admisión y Matrícula

Fuente: Elaboración Propia

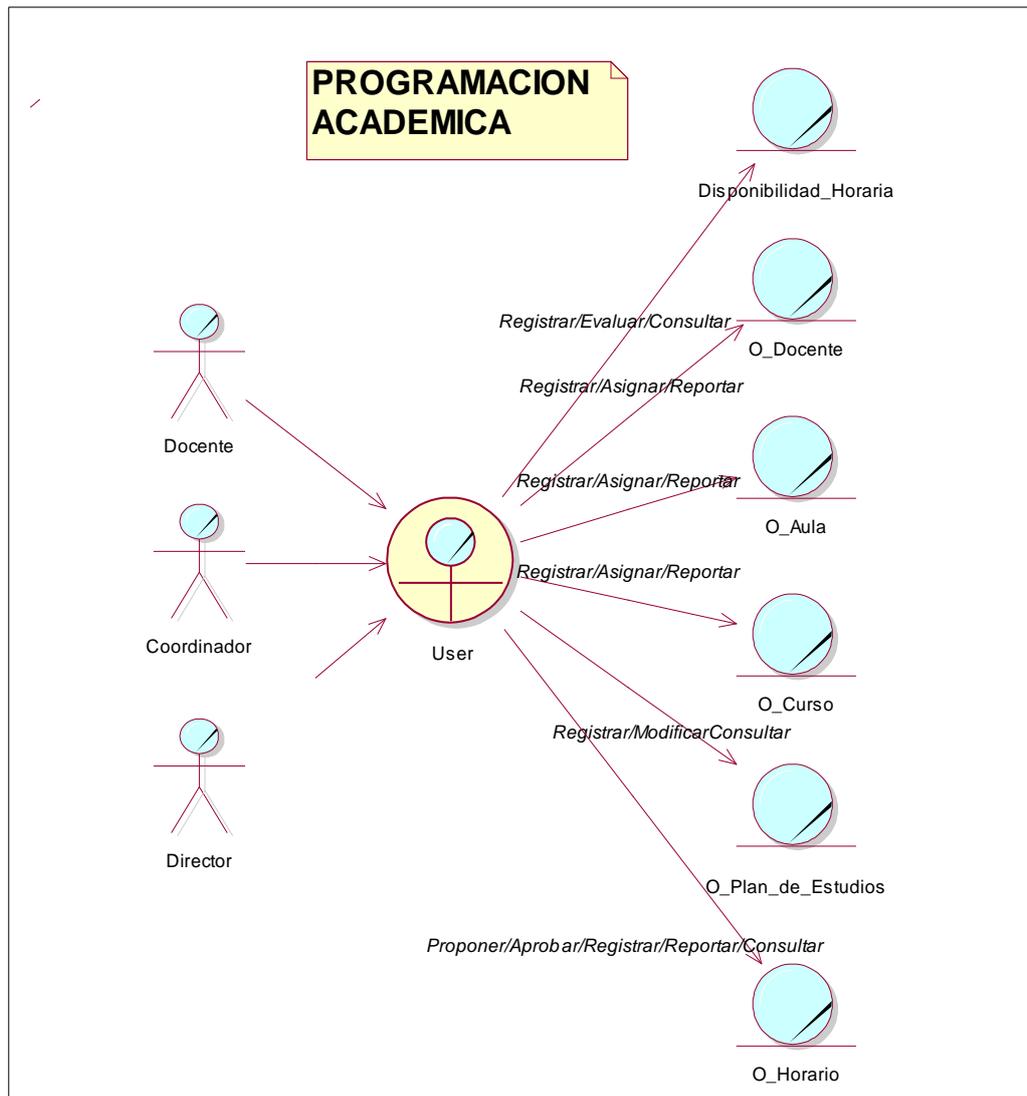


Figura 16: Modelo de Objetos del Negocio de Programación Académica
Fuente: Elaboración Propia

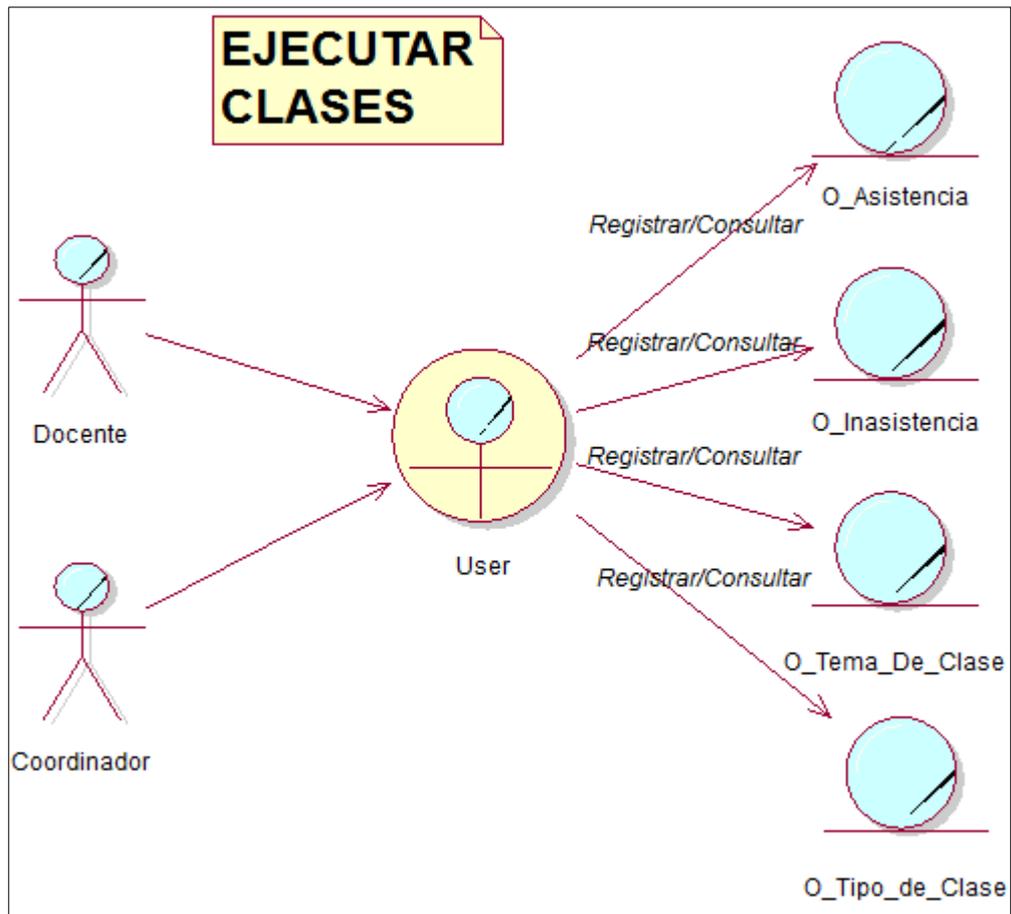


Figura 17: Modelo de Objetos del Negocio de Ejecución de Clases
Fuente: Elaboración Propia

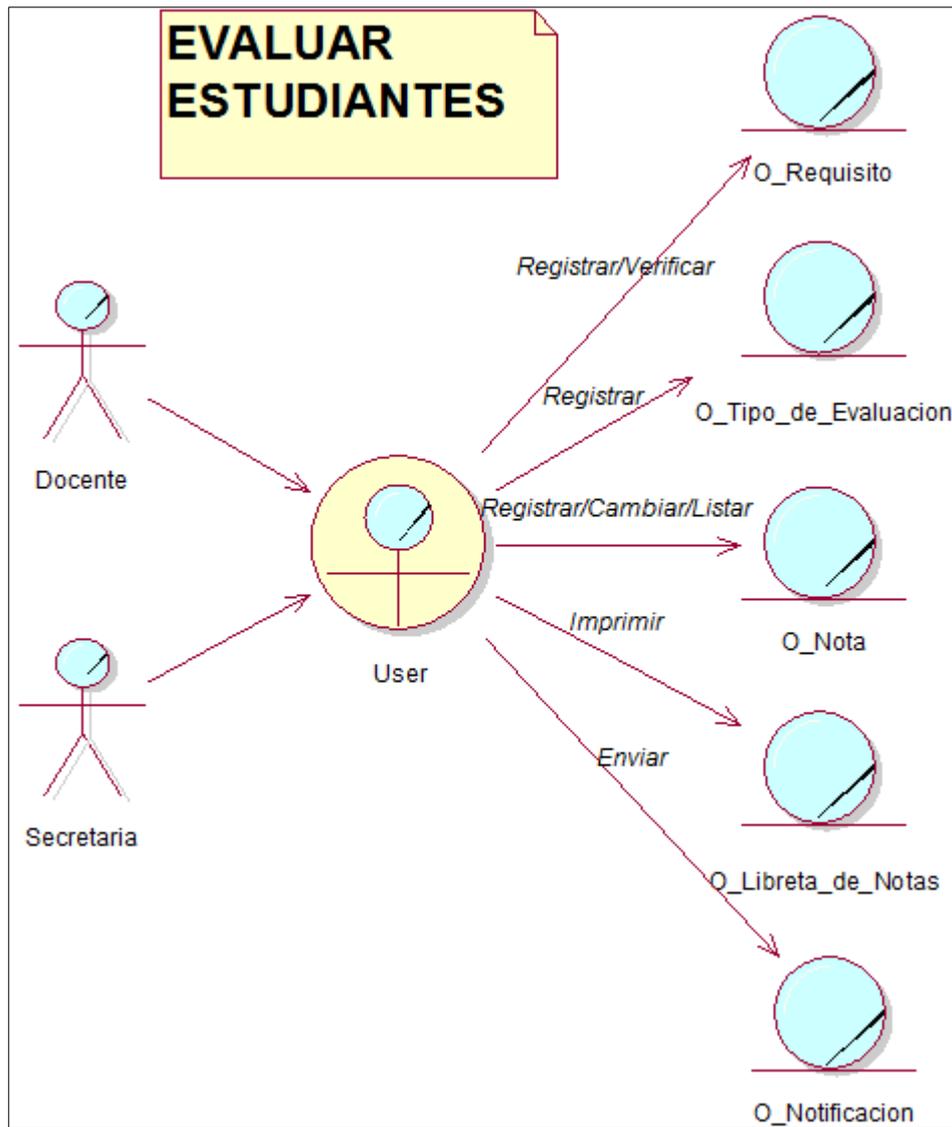


Figura 18: Modelo de Objetos del Negocio de Evaluación de Estudiantes
Fuente: Elaboración Propia

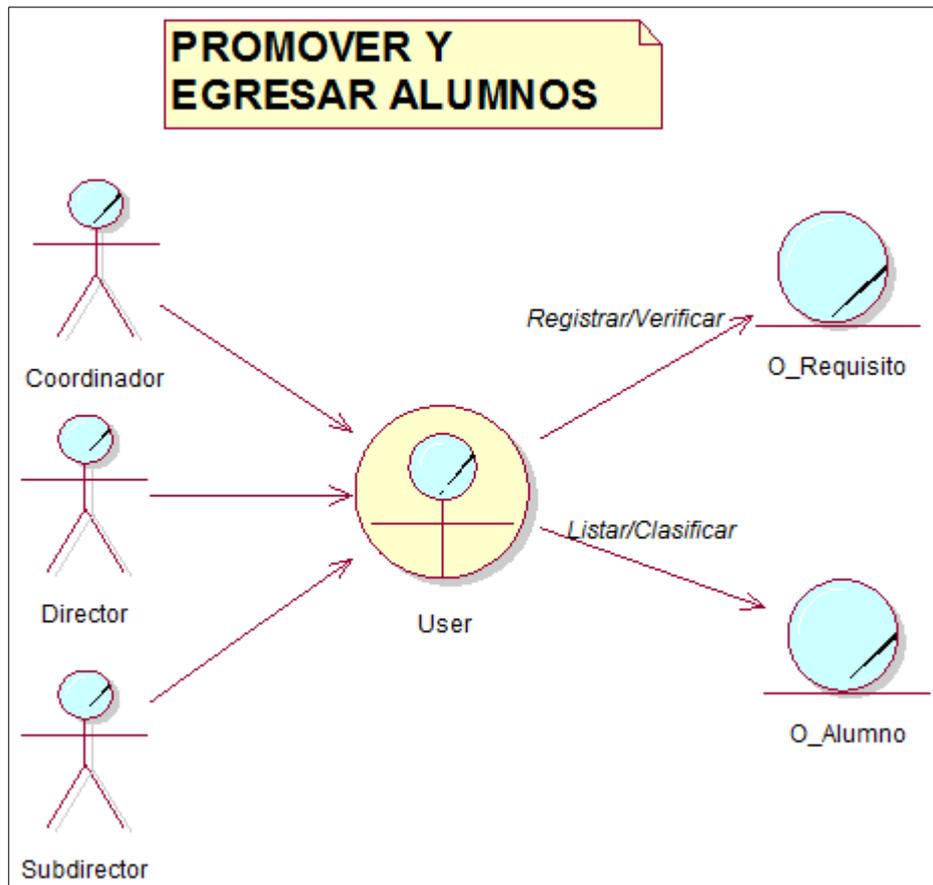


Figura 19: Modelo de Objetos del Negocio de Promoción y Egreso de alumnos

Fuente: Elaboración Propia

4.1.9. Modelo De Dominio Del Problema

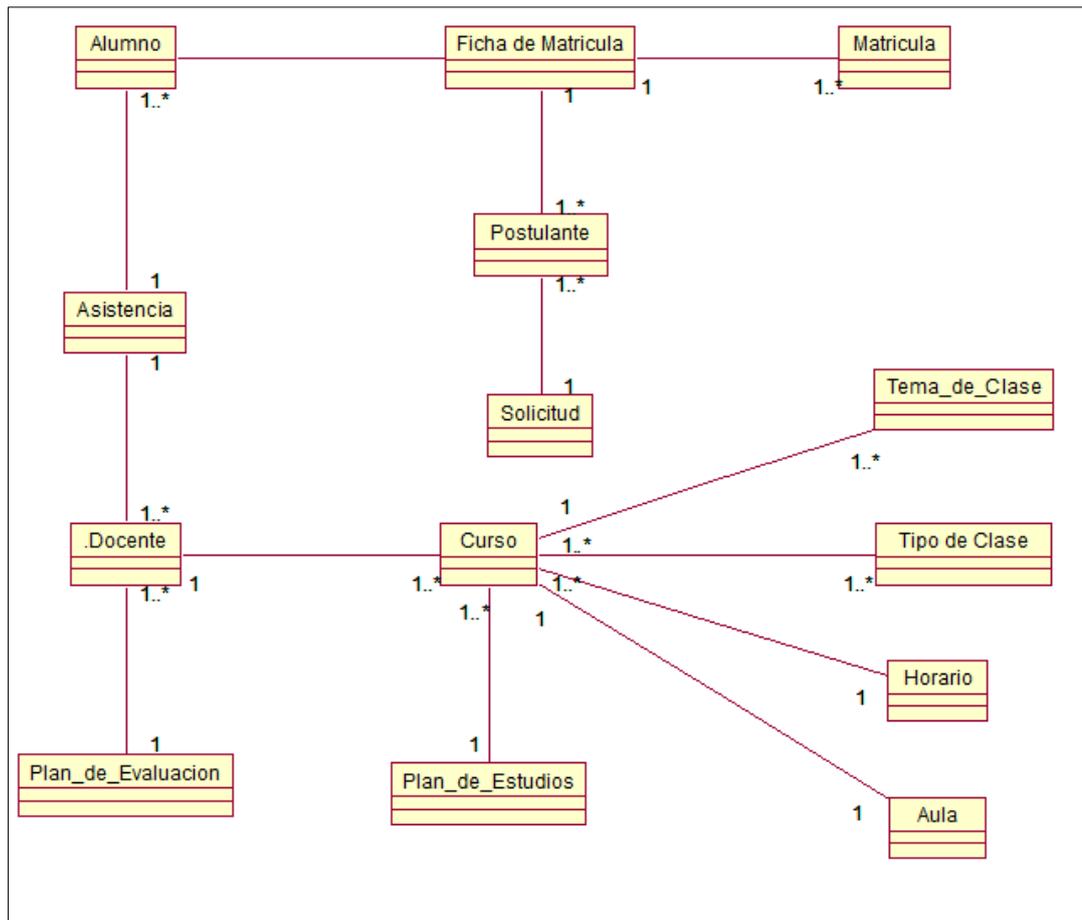


Figura 20: Modelo de Dominio del Problema

Fuente: Elaboración Propia

4.1.10. Glosario del Negocio

Stakeholders: Son elementos externos al sistema pero forma parte de la empresa.

4.1.11. Visión Del Negocio

Introducción:

En este documento describiremos la visión del negocio que se desarrollará mediante el prototipo del sistema integrado de gestión académica de la IEP “Sagrado Corazón de Jesús”

Posicionamiento:

“Sagrado Corazón de Jesús” es una institución educativa de carácter privado dedicada a brindar educación primaria y secundaria.

✓ **Posibilidad del negocio**

La empresa requiere de un sistema que le permita gestionar eficientemente sus procesos de gestión académica como las admisiones, matriculas, asistencias, programación de horarios, promoción y egresados

✓ **Declaración del problema**

Problema	La mala gestión de los procesos de gestión académica.
Afecta	Director, Subdirector, Coordinador Académico, Secretaria, Docentes, alumnos y padfres de familia.
Impacto	Insatisfacción por no poder los procesos académicos en forma eficiente.
Solución	Diseñar un sistema que permita optimizar el tiempo en gestionar los trámites documentarios.

Tabla 07: Declaración del Problema

Fuente: Elaboración Propia

✓ **Declaración Posicionamiento del Producto**

¿Para quién es el sistema?	IEP “SAGRADO CORAZON DE JESUS”
¿Quién lo necesita?	Área Académica
Nombre del Producto	SIGEA
Resumen de beneficios	<ul style="list-style-type: none"> - Proporciona información oportuna - Mejorará los tiempos de respuesta. - Presenta una base de datos uniforme para el almacenamiento de datos - Es un sistema adaptable a las modificaciones futuras de acuerdo a las necesidades.

Tabla 08: Declaración Posicionamiento del Producto

Fuente: Elaboración Propia

✓ **Descripción de los Actores**

Actores

Director: Autoridad máxima del colegio. Tiene acceso a reportes de todo tipo, para toma de decisiones

Subdirector: Asistente del director. De similares funciones que el director.

Coordinador Académico: Realiza los procesos de admisión, programación académica, asistencia de docentes y promoción y egreso de alumnos.

Secretaria: Realiza la matrícula de alumnos, registro de notas, libreta de notas

Padre de Familia: Lleva la documentación necesaria para la admisión y matrícula de sus hijos.

Alumno: el cliente del colegio.

✓ **Descripción de los Customer y de los Stakeholder**

Restricciones

- **Customer:**

IEP “SAGRADO CORAZON DE JESUS” de Casa Grande

- **Stakeholder:**

Director: Encargado de la dirección y gerencia del colegio

- **Mercado Demográfico:**

Todos los menores de edad en edad escolar

✓ **Resumen de los stakeholder**

Nombre	Descripción	Responsabilidad
Director	Tiene a su cargo la dirección del colegio	- Velar por el bienestar del colegio
Subdirector	Asistente del director	- Gestionar la información de los documentos.
Coordinador Académico	Encargado de la gestión académica	Gestionar las admisiones, programación de horarios, egresar alumnos.
Secretaria	Encargado de la atención a los clientes (padres y alumnos)	
Docentes	Encargados de la enseñanza y educación de los alumnos	
Alumnos	Son los principales clientes del negocio	
Apoderados / Padres de Familia	Familiares encargados de la tutela de los alumnos	

Tabla 09: Resumen de los Stakeholder

Fuente: Elaboración Propia

✓ **Restricciones**

Este Sistema ha sido elaborado para gestionar los procesos académicos de la institución.

- ✓ **Rangos de calidad**
 - Se registrará información de los alumnos, docentes, horarios, clases, asistencias, etc.
 - El Sistema manejará en forma eficiente el control de la información.
- ✓ **Los Predecesores y Prioridad**

Nro	Nombre	Predecesor	Prioridad
1	Admisión y Matrícula <ul style="list-style-type: none"> • Verificar Vacantes • Verificar Alumno • Registrar Solicitud • Evaluar Solicitud • Aprobar/Desaprobar Solicitud • Registrar Alumno • Registrar Matrícula 		
2	Programacion Academica <ul style="list-style-type: none"> • Verificar disponibilidad del docente • Verificar disponibilidad de salones • Registrar Docente • Registrar Curso • Registrar Salón • Registrar Grado de estudio • Asignar Docente a curso • Asignar Grado de estudio a salón de clases. • Asignar Curso a grado de estudio. • Validar Horario • Registrar Horario 	1	
3	Ejecución de clases <ul style="list-style-type: none"> • Registrar asistencia de docente • Registrar asistencia de alumno • Registrar tipo de clase • Registrar tema de clase • Registrar trabajos dejados 	2	

4	Evaluación de Alumnos <ul style="list-style-type: none"> • Registrar tipo de evaluación • Registrar notas por alumno. • Registrar cambios de notas. • Enviar notas finales 	3	
5	Promoción y Egresados <ul style="list-style-type: none"> • Reportar alumnos • Clasificar alumnos • Registrar alumnos promovidos y egresados. 	4	

Tabla 10: Predecesores y Prioridades

Fuente: Elaboración Propia

✓ **Requerimientos del Sistema**

Plataforma:

El sistema podrá ejecutarse en Plataformas como Windows y Linux.

Normas de Comunicación:

Las diferentes estaciones deberán comunicarse utilizando el Protocolo TCP/IP, ya que es la familia de protocolos más utilizados.

Seguridad:

El sistema en sí y la Base de Datos tienen un grupo de usuarios y cada uno de estos posee una contraseña para así poder restringir el acceso asegurando la integridad de la información.

✓ **Rendimiento**

El tiempo de respuesta del sistema será más rápido de lo que se haría manualmente.

✓ **Requerimientos medioambientales**

- Cualquier lugar donde se encuentre una PC con acceso a Internet (de acuerdo a las necesidades de la empresa).
- Materiales de Escritorio.

4.2. Modelo de Requerimientos.

4.2.1. Necesidades de los stakeholders

Necesidad	Prioridad	Inquietudes	Solución actual	Solución propuesta
Admisión de nuevos alumnos	Alta	Evaluación de muchas solicitudes	<p>La secretaria recibe las solicitudes de admisión de nuevos alumnos</p> <p>El Coordinador académico las revisa y evalúa</p> <p>El coordinador académico da el visto bueno</p> <p>La secretaria registra al nuevo alumno</p>	
Creación de Horarios	Media	Ver disponibilidad de docentes	<p>El coordinador académico cita personalmente a los docentes para saber su disponibilidad laboral</p> <p>Al obtener toda la información de todos los docentes, arma un horario como versión preliminar</p> <p>Luego, se reúne con el director para que este lo revise y le de su visto bueno.</p>	
Registrar asistencia	Media	Registrar asistencia de docentes	<p>Los docentes se acercan a la oficina del coordinador académico.</p> <p>Firman su asistencia en el registro de asistencias, y anotan la hora.</p>	

Registrar Notas	Media	Registrar la nota de alumnos	Los docentes, luego de realizar sus evaluaciones, registran sus notas La secretaria les envía un registro a llenar con las notas de los alumnos Luego de ser llenadas, la secretaria procede a registrarlo en un archivo Excel, y luego las apunta en la libreta de notas	
Promover alumnos egresados	Alta	Alumnos Egresados	El coordinador académico revisa la lista de alumnos a fin de año. Si estos alumnos cumplen requisitos de aprobados y están en 5 año de secundaria, se convierten en egresados.	

Tabla 11: Necesidades de los Stakeholders

Fuente: Elaboración Propia

4.2.2. Casos De Uso De Requerimientos

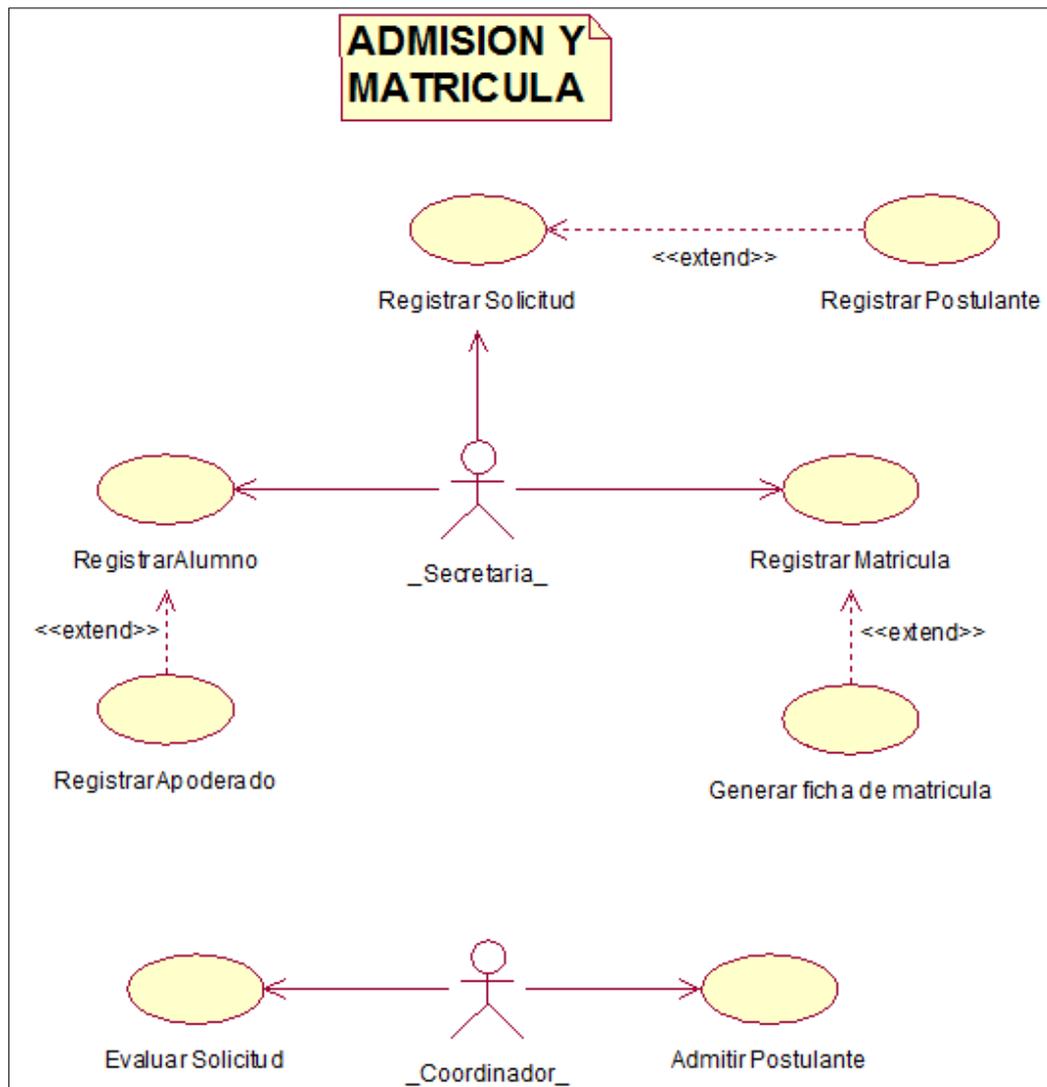


Figura 21: Caso de Uso de Requerimientos de Admisión y Matricula

Fuente: Elaboración Propia

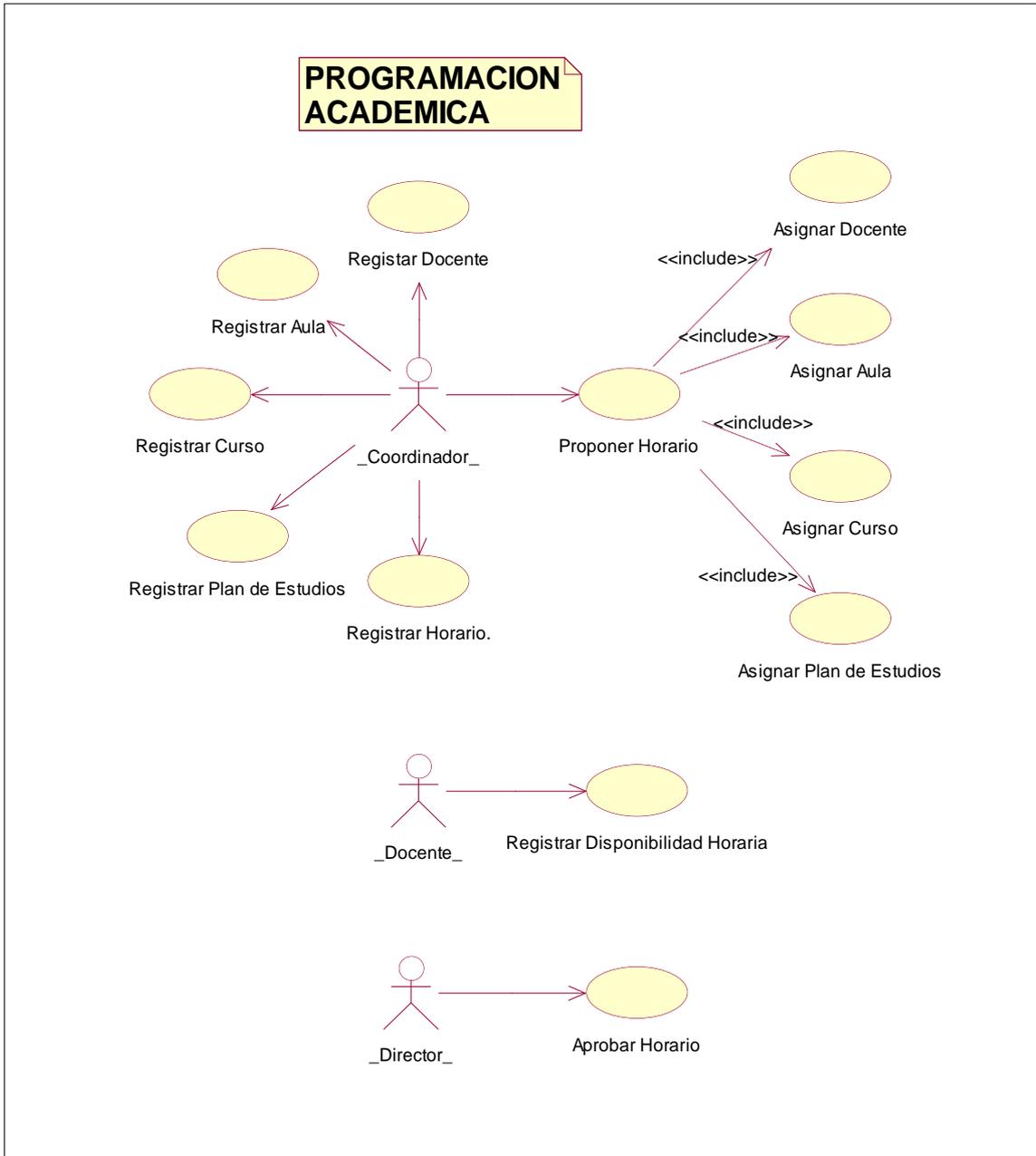


Figura 22: Caso de Uso de Requerimiento de Programación Académica
Fuente: Elaboración Propia

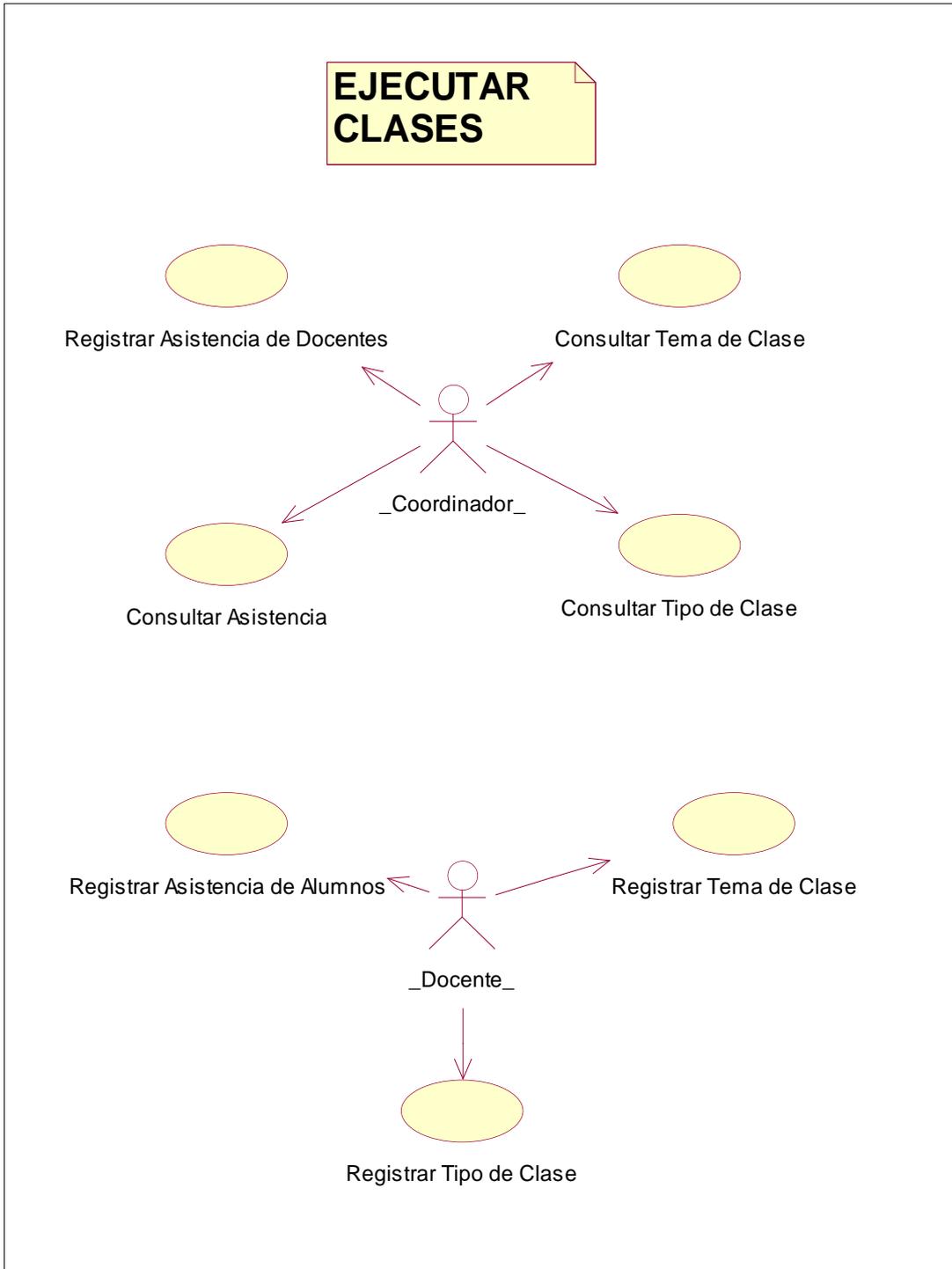


Figura 23: Caso de Uso de Requerimientos de Ejecución de clases
Fuente: Elaboración Propia

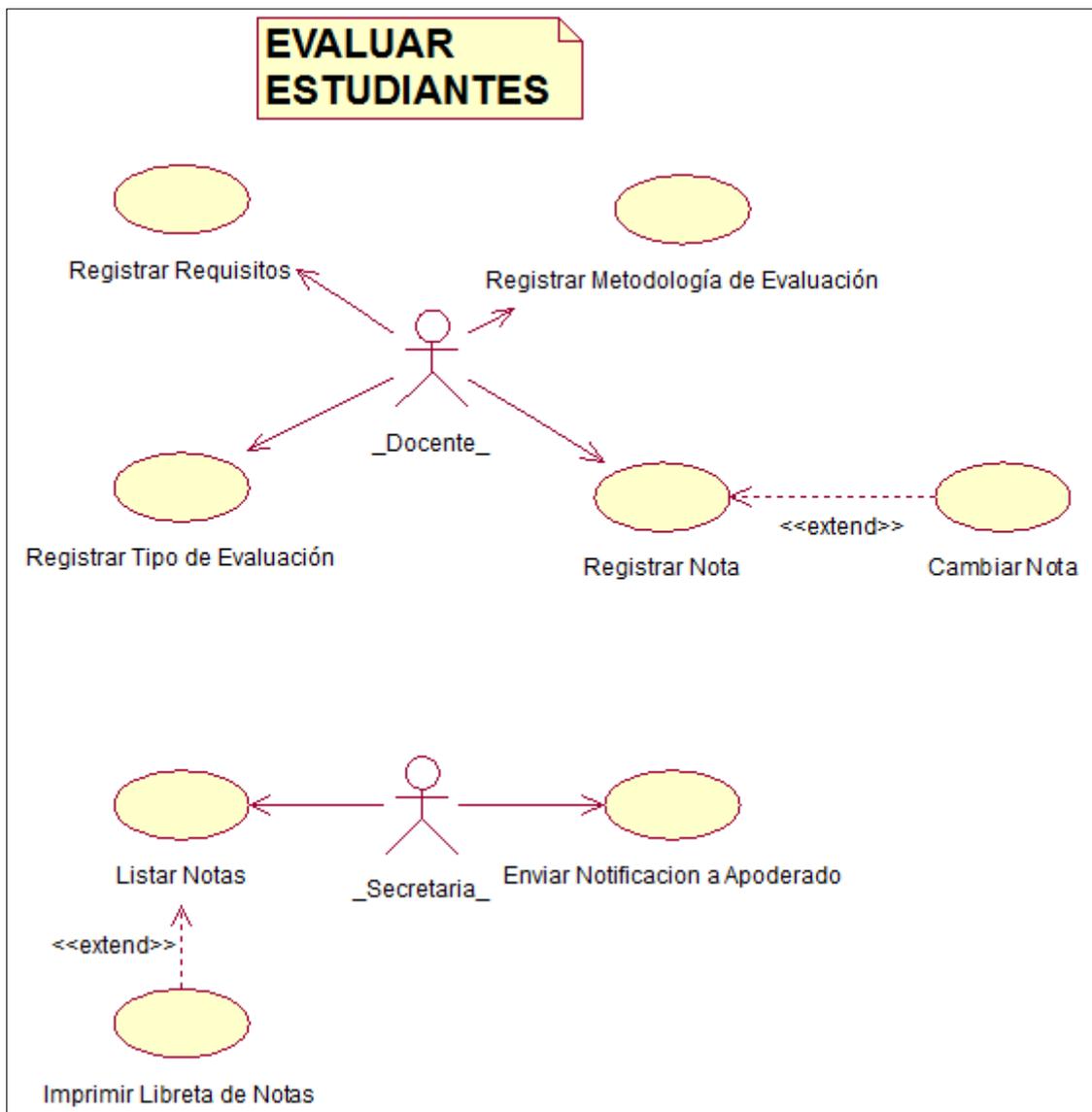


Figura 24: Caso de Uso de Requerimientos de Evaluación de estudiantes
Fuente: Elaboración Propia

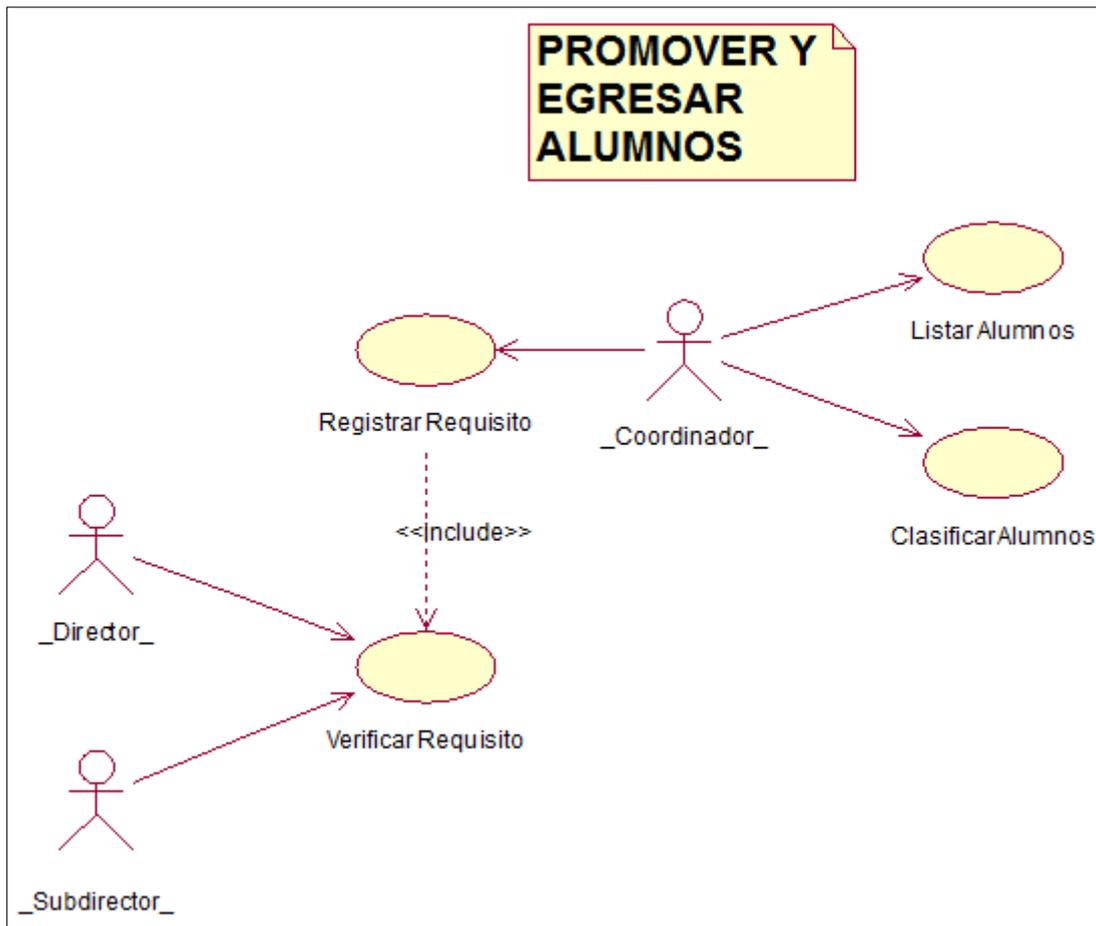


Figura 25: Caso de Uso de Requerimiento de Egreso de Alumnos
Fuente: Elaboración Propia

4.2.3. Especificaciones de los casos de uso:

4.2.3.1.Registrar Solicitud

1. Breve Descripción:

En este caso de uso se procederá a registrar las solicitudes de admisión de un nuevo estudiante

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Postulante
- Registrar Postulante
- Buscar Solicitud
- Ingresar Solicitud
- Registrar Solicitud

3. Pre Condiciones:

La solicitud a crearse no debe existir y se necesita que se haya registrado el postulante para poder registrar la solicitud.

4. Post Condiciones:

Se necesita para que el coordinador académico evalúe dicha solicitud de admisión al postulante

5. Puntos de Extensión:

Registrar Postulante

4.2.3.2.Registrar Alumno

1. Breve Descripción:

En este caso de uso se procederá a registrar los nuevos alumnos admitidos para el colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Alumno
- Ingresar Alumno
- Ingresar Apoderados
- Registrar Apoderados
- Registrar Alumno

3. Pre Condiciones:

El alumno no debe existir en el sistema

4. Post Condiciones:

Se necesita para que el coordinador académico pueda calcular la capacidad de alumnos en las aulas

5. Puntos de Extensión:

Registrar Apoderado

4.2.3.3. Registrar Matrícula

1. Breve Descripción:

En este caso de uso se procederá a registrar la matrícula de los alumnos cada inicio de año escolar

2. Flujo de Eventos:

2.1. Flujo Básico:

- Solicitar Matricula
- Buscar Alumno
- Ingresar Matricula
- Actualizar Matricula
- Registrar Matricula

3. Pre Condiciones:

El alumno debe existir en el sistema.

4. Post Condiciones:

Se necesita para que el coordinador académico arme los horarios por grado y aula.

5. Puntos de Extensión:

Generar Ficha de matricula

4.2.3.4. Evaluar Solicitud

1. Breve Descripción:

En este caso de uso se procederá a evaluar las solicitudes de admisión de un nuevo estudiante

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Postulante
- Buscar Solicitud
- Evaluar Solicitud
- Aprobar Solicitud
- Guardar solicitud

3. Pre Condiciones:

La solicitud debe existir y aparecer como “no aprobada” o “por revisar”.

4. Post Condiciones:

Se necesita para luego registrar las matrículas y nuevos alumnos

5. Puntos de Extensión:

Registrar Postulante

4.2.3.5. Registrar Docente

1. Breve Descripción:

En este caso de uso se procederá a registrar a los nuevos docentes que sean contratados por el colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Docente

- Ingresar Docente
- Registrar Docente

3. Pre Condiciones:

El docente no debe existir en el sistema

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

No hay

4.2.3.6. Registrar Aula

1. Breve Descripción:

En este caso de uso se procederá a registrar a las aulas de la institucion

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Aula
- Ingresar Aula
- Registrar aula

3. Pre Condiciones:

El aula no debe existir en el sistema

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

No hay

4.2.3.7. Registrar Curso

1. Breve Descripción:

En este caso de uso se procederá a registrar los nuevos cursos que se dicten en el colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Curso
- Ingresar curso
- Registrar Curso

3. Pre Condiciones:

El curso no debe existir en el sistema

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

No hay

4.2.3.8. Registrar Plan de Estudios

1. Breve Descripción:

En este caso de uso se procederá a registrar los planes de estudios correspondientes a cada grado en el colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar plan de estudios
- Verificar si existe
- Ingresar plan de estudios
- Registrar plan de estudios

3. Pre Condiciones:

El curso no debe existir en el sistema

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

No hay

4.2.3.9. Registrar Horarios

1. Breve Descripción:

En este caso de uso se procederá a registrar los nuevos horarios correspondientes a cada grado del colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar curso
- Buscar grado
- Buscar Aula
- Buscar docente
- Buscar plan de estudios
- Asignar curso al docente
- Asignar docente al grado

- Asignar aula al grado
- Asignar cursos al grado
- Asignar día y hora a cada curso
- Registrar Horario

3. Pre Condiciones:

Los cursos, docentes, aula y grados ya deben existir en el sistema.

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

- Asignar aula al grado

4.2.3.10. Registrar Disponibilidad Horaria

1. Breve Descripción:

En este caso de uso el docente procederá a registrar la disponibilidad horaria que tenga para asistir a laborar en la institución.

2. Flujo de Eventos:

2.1. Flujo Básico:

- Ingresar días disponibles
- Ingresar horas disponibles
- Confirmar disponibilidad horaria
- Registrar disponibilidad horaria

3. Pre Condiciones:

El curso no debe existir en el sistema

4. Post Condiciones:

Se necesita para luego programar los horarios

5. Puntos de Extensión:

No hay

4.2.3.11. Aprobar Horario

1. Breve Descripción:

En este caso de uso el director procederá a aprobar y dar el OK al nuevo horario establecido.

2. Flujo de Eventos:

2.1. Flujo Básico:

- Listar horarios por grado
- Consultar estado de horarios
- Revisar Horario
- Realizar Cambios al horario
- Aprobar horario

3. Pre Condiciones:

El horario debe existir en el sistema

4. Post Condiciones:

Se necesita para publicar el horario fina

5. Puntos de Extensión:

No hay

4.2.3.12. Registrar Asistencia de Docentes

1. Breve Descripción:

En este caso de uso el coordinador académico registrara las asistencias de los docentes que asisten a laborar

2. Flujo de Eventos:

2.1. Flujo Básico:

- Registrar Curso de Asistencia
- Registrar Hora de entrada
- Registrar Hora de salida
- Registrar Tardanza
- Registrar Asistencia
- Registrar Inasistencia

3. Pre Condiciones:

Ninguna.

4. Post Condiciones:

Se necesita para luego reportar las asistencias.

5. Puntos de Extensión:

No hay

4.2.3.13. Consultar Tema de Clase

1. Breve Descripción:

En este caso de uso el coordinador académico consultará las clases que realizan los docentes

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Grado
- Buscar Docente
- Buscar Curso
- Listar temas de clase

3. Pre Condiciones:

No

4. Post Condiciones:

No

5. Puntos de Extensión:

Generar Reporte de Clases

4.2.3.14. Registrar Asistencia de Alumnos

1. Breve Descripción:

En este caso de uso el docente procederá a registrar la asistencia de los alumnos presentes en su clase

2. Flujo de Eventos:

2.1. Flujo Básico:

- Entrar a un menú de opciones
- Ingresar asistencia de alumnos
- Registrar asistencia de alumnos

3. Pre Condiciones:

Se debe crear la sesión de clase

4. Post Condiciones:

5. Puntos de Extensión:

No hay

4.2.3.15. Registrar Clase

1. Breve Descripción:

En este caso de uso el docente procederá a registrar el tema de clase que va a desarrollar.

2. Flujo de Eventos:

2.1. Flujo Básico:

- Buscar Curso
- Registrar Curso de Asistencia
- Registrar Clase
- Registrar Inasistencia

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para conocer las clases realizadas

5. Puntos de Extensión:

No hay

4.2.3.16. Registrar Requisitos de Evaluación

1. Breve Descripción:

En este caso de uso el docente procederá a registrar los requisitos de evaluación de sus cursos

2. Flujo de Eventos:

2.1. Flujo Básico:

- Seleccionar curso
- Ingresar requisitos de evaluación
- Registrar tema de clase

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para poder calificar a los alumnos

5. Puntos de Extensión:

No hay

4.2.3.17. Registrar tipo de Evaluación

1. Breve Descripción:

En este caso de uso el docente procederá a registrar el tipo de evaluaciones que les hará a sus alumnos (trabajo a domicilio, examen de laboratorio, etc).

2. Flujo de Eventos:

2.1. Flujo Básico:

- Seleccionar curso
- Ingresar tipo de evaluación
- Registrar tipo de evaluación

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para poder calificar a los alumnos

5. Puntos de Extensión:

No hay

4.2.3.18. Registrar Nota

1. Breve Descripción:

En este caso de uso el docente procederá a registrar las notas de sus alumnos

2. Flujo de Eventos:

2.1. Flujo Básico:

- Seleccionar curso
- Ingresar nota a cada alumno
- Registrar notas

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para poder calificar a los alumnos

5. Puntos de Extensión:

No hay

4.2.3.19. Registrar Requisitos de promover egresados

1. Breve Descripción:

En este caso de uso el docente procederá a registrar los requisitos para promover y egresar alumnos del colegio

2. Flujo de Eventos:

2.1. Flujo Básico:

- Seleccionar grado de estudios
- Ingresar requisitos de egreso y promoción
- Registrar requisitos

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para poder tener la lista de nuevos alumnos por egresar

5. Puntos de Extensión:

No hay

4.2.3.20. Listar Alumnos

1. Breve Descripción:

En este caso de uso el director y subdirector listarán los nuevos alumnos que salen de la institución a modo de reportes

2. Flujo de Eventos:

2.1. Flujo Básico:

- Seleccionar grado
- Consultar por algún parámetro (aprobados, por egresar)
- Registrar tipo de evaluación

3. Pre Condiciones:

El usuario debe iniciar sesión en el sistema

4. Post Condiciones:

Se necesita para poder promover a los alumnos

5. Puntos de Extensión:

Clasificar alumnos

4.2.4. Diagramas de Actividad

4.2.4.1.Registrar Solicitud de Admisión

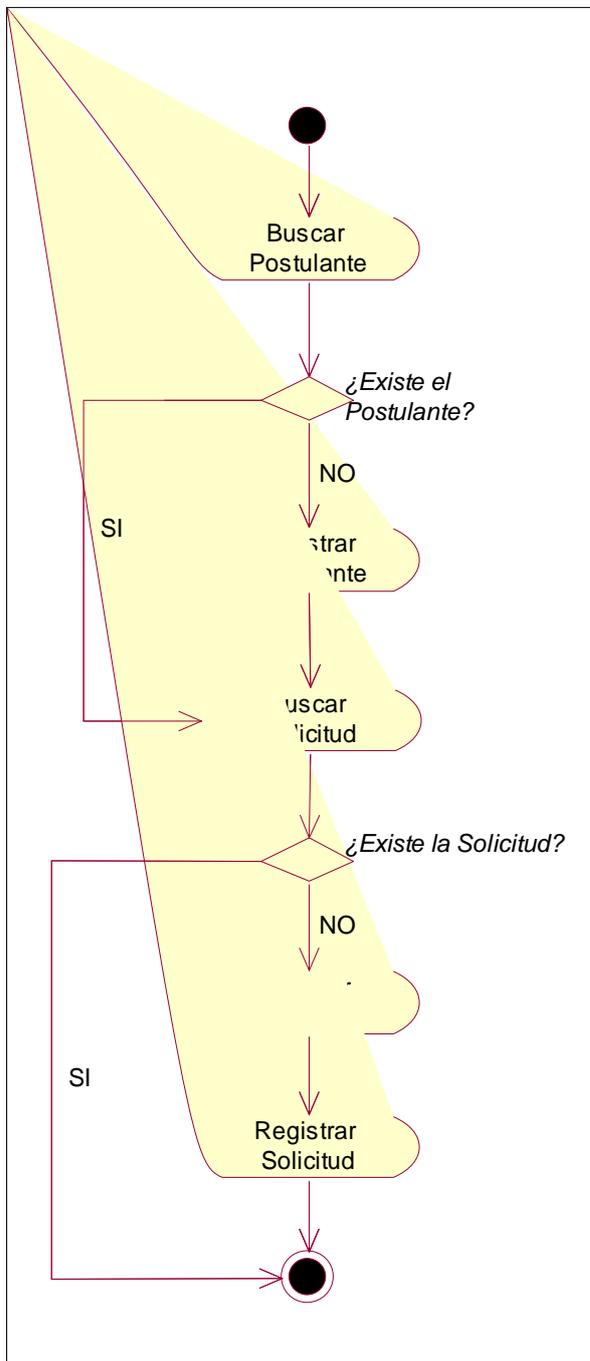


Figura 26: Diagrama de Actividad Registrar Solicitud de Admisión
Fuente: Elaboración Propia

4.2.4.2.Registrar Alumno

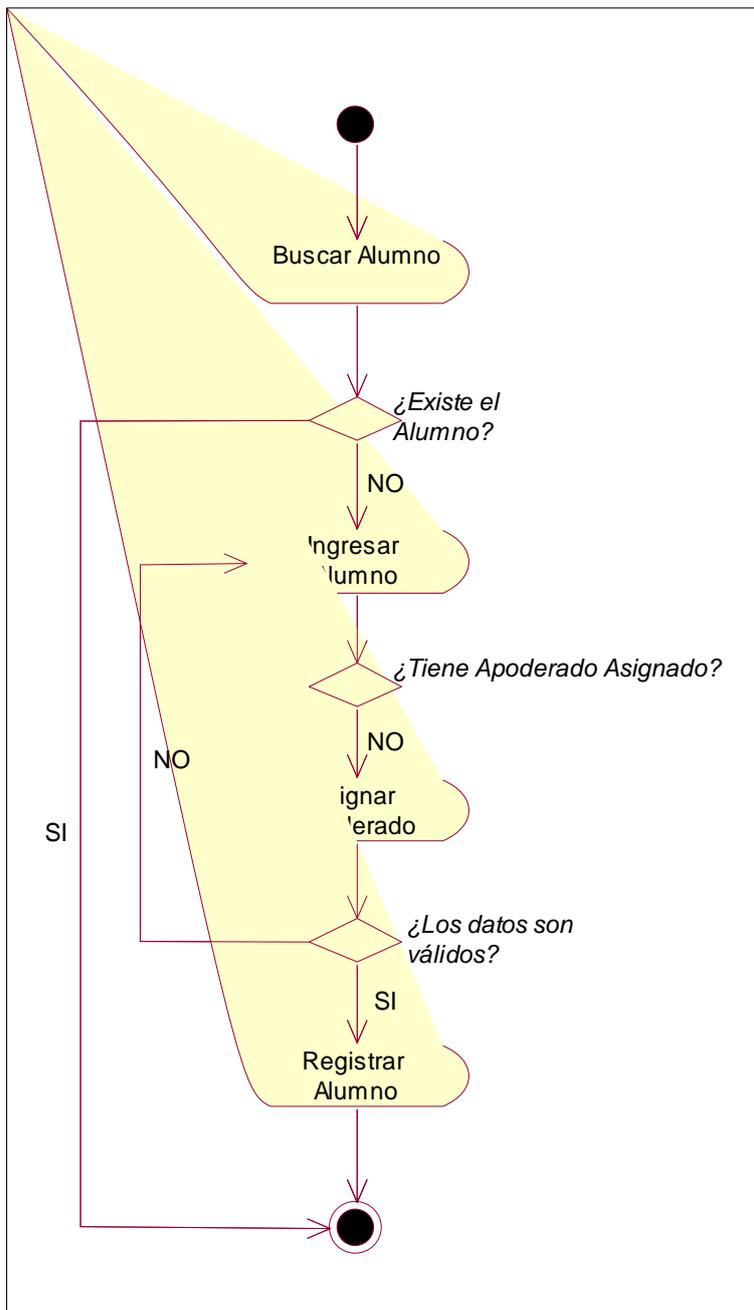


Figura 27: Diagrama de Actividad Registrar Alumno
Fuente: Elaboración Propia

4.2.4.3.Registrar Matricula

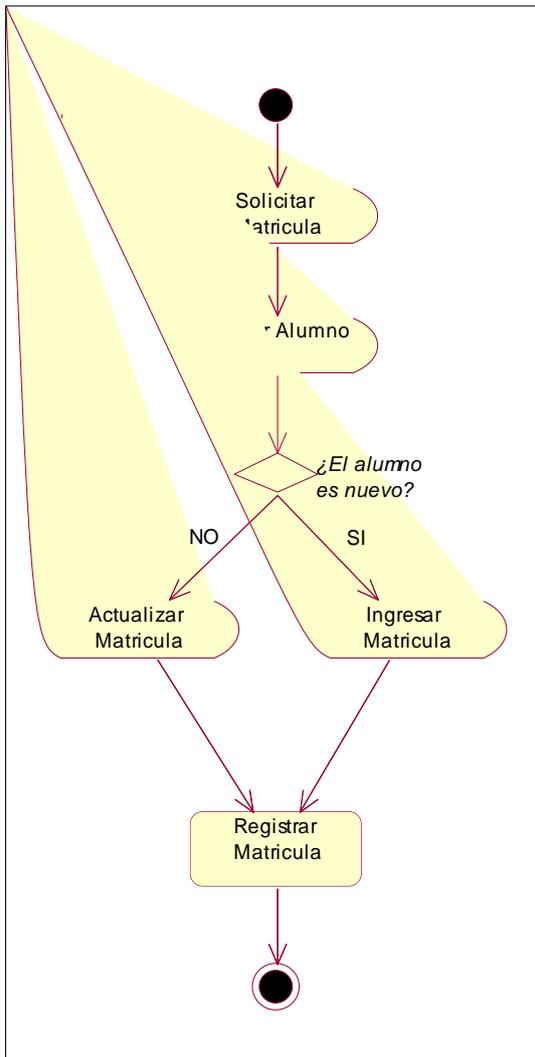


Figura 28: Diagrama de Actividad Registrar Matricula
Fuente: Elaboración Propia

4.2.4.4. Evaluar Solicitud

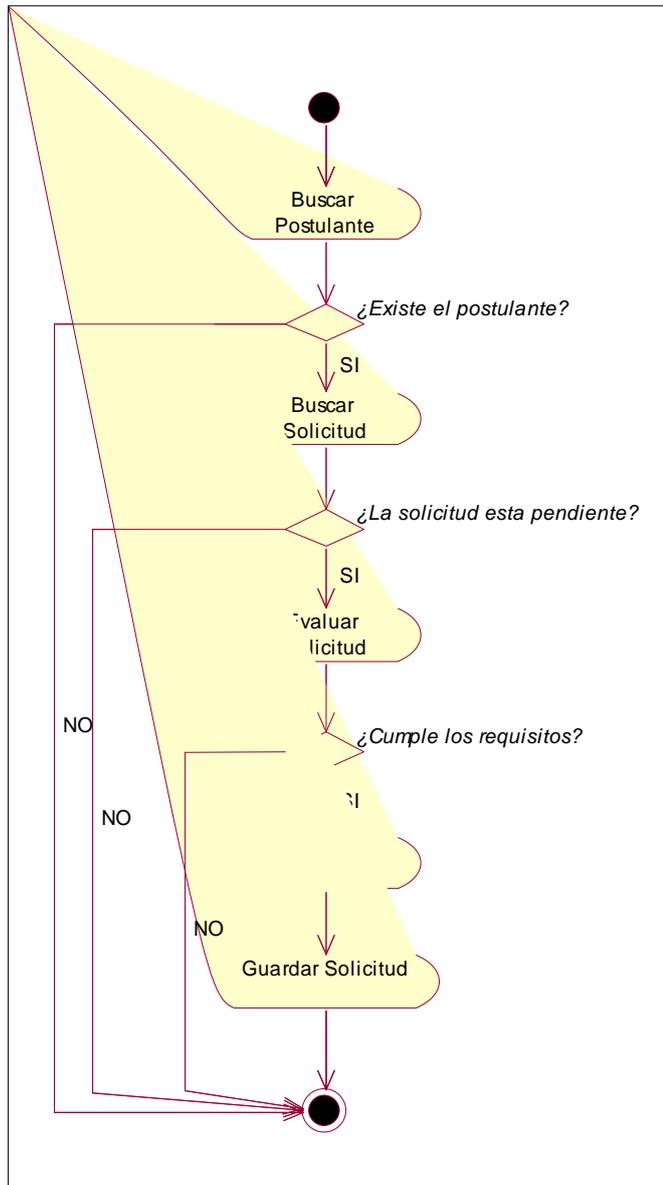


Figura 29: Diagrama de Actividad Evaluar Solicitud
Fuente: Elaboración Propia

4.2.4.5.Registrar Docente

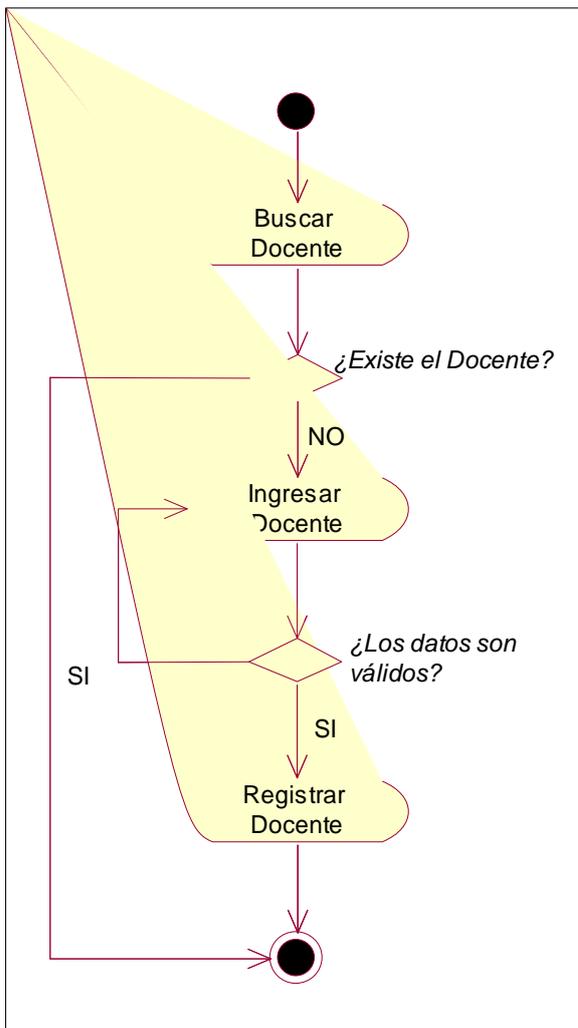


Figura 30: Diagrama de Actividad Registrar Docente
Fuente: Elaboración Propia

4.2.4.6.Registrar Aula

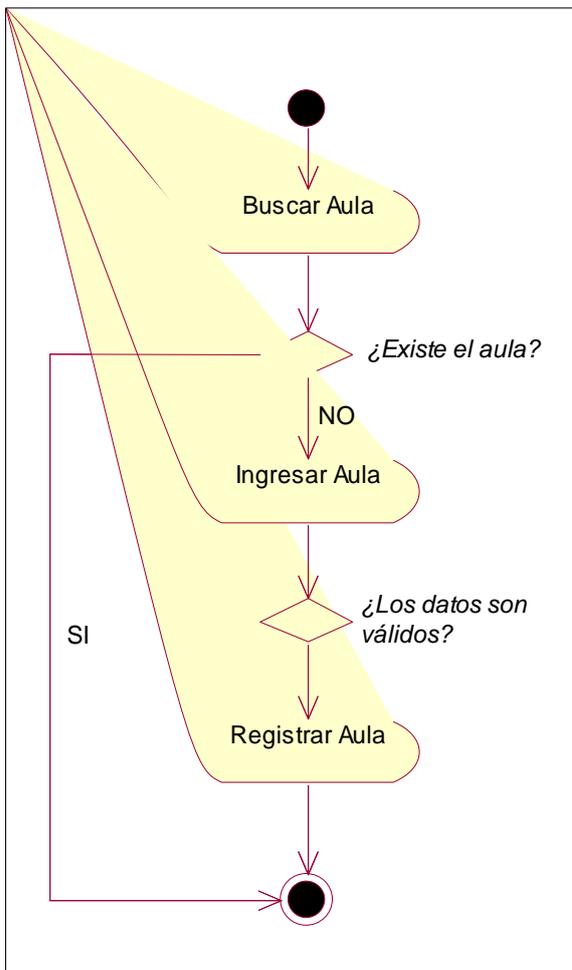


Figura 31: Diagrama de Actividad Registrar aula
Fuente: Elaboración Propia

4.2.4.7.Registrar Curso

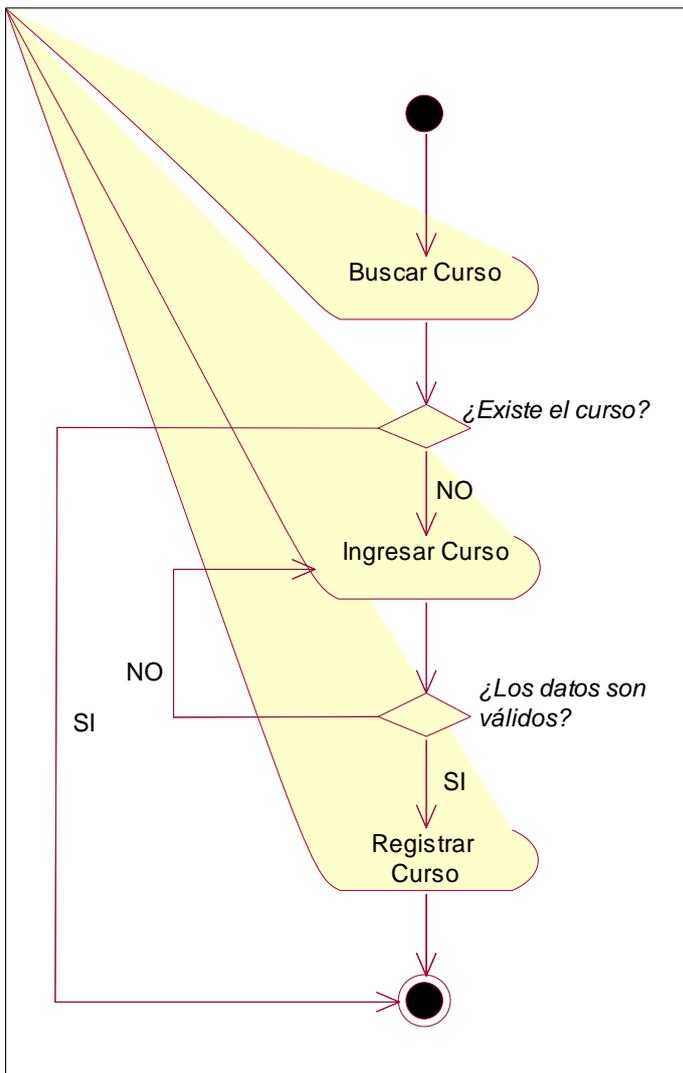


Figura 32: Diagrama de Actividad Registrar Curso
Fuente: Elaboración Propia

4.2.4.8.Registrar Plan de Estudios

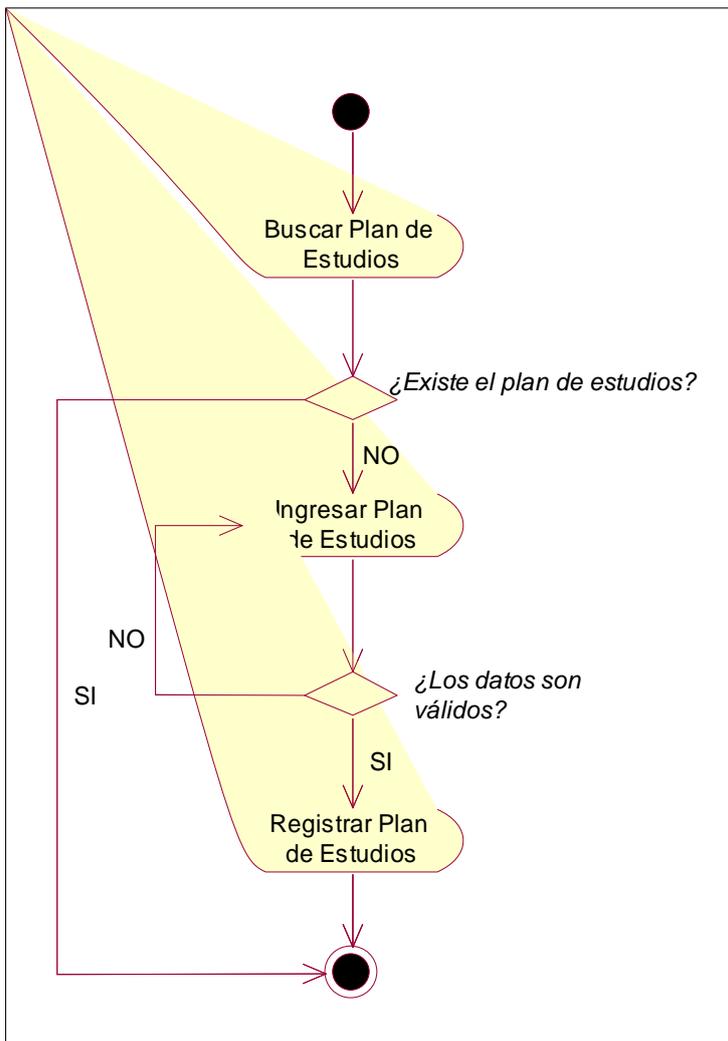


Figura 33: Diagrama de Actividad Registrar Plan de Estudios
Fuente: Elaboración Propia

4.2.4.9.Registrar Horario

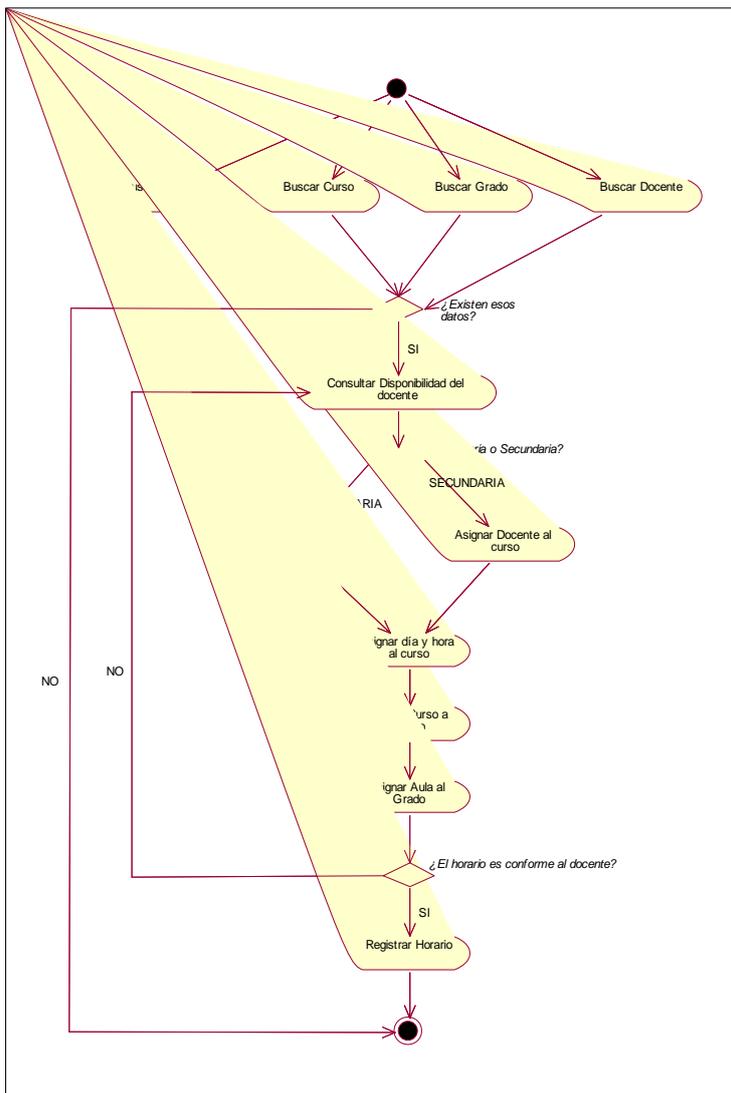


Figura 34: Diagrama de Actividad Registrar Horario
Fuente: Elaboración Propia

4.2.4.10. Registrar Disponibilidad Horaria

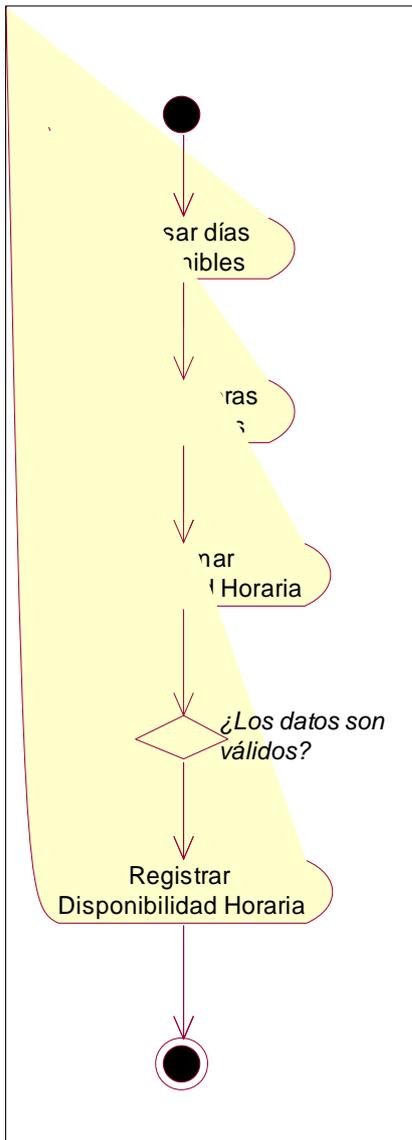


Figura 35: Diagrama de Actividad Registrar disponibilidad horario
Fuente: Elaboración Propia

4.2.4.11. Aprobar Horario

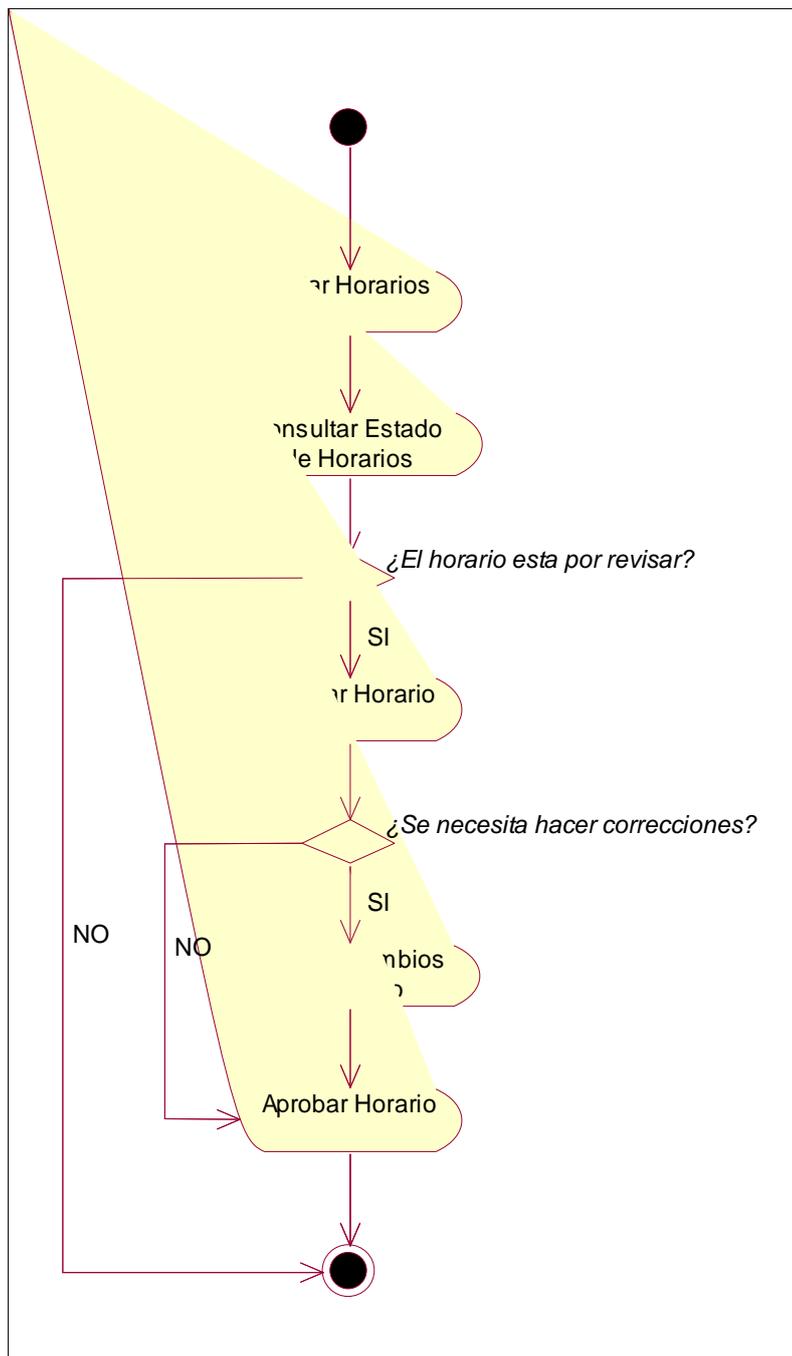


Figura 36: Diagrama de Actividad Aprobar Horario
Fuente: Elaboración Propia

4.2.4.12. Registrar Asistencia Docente

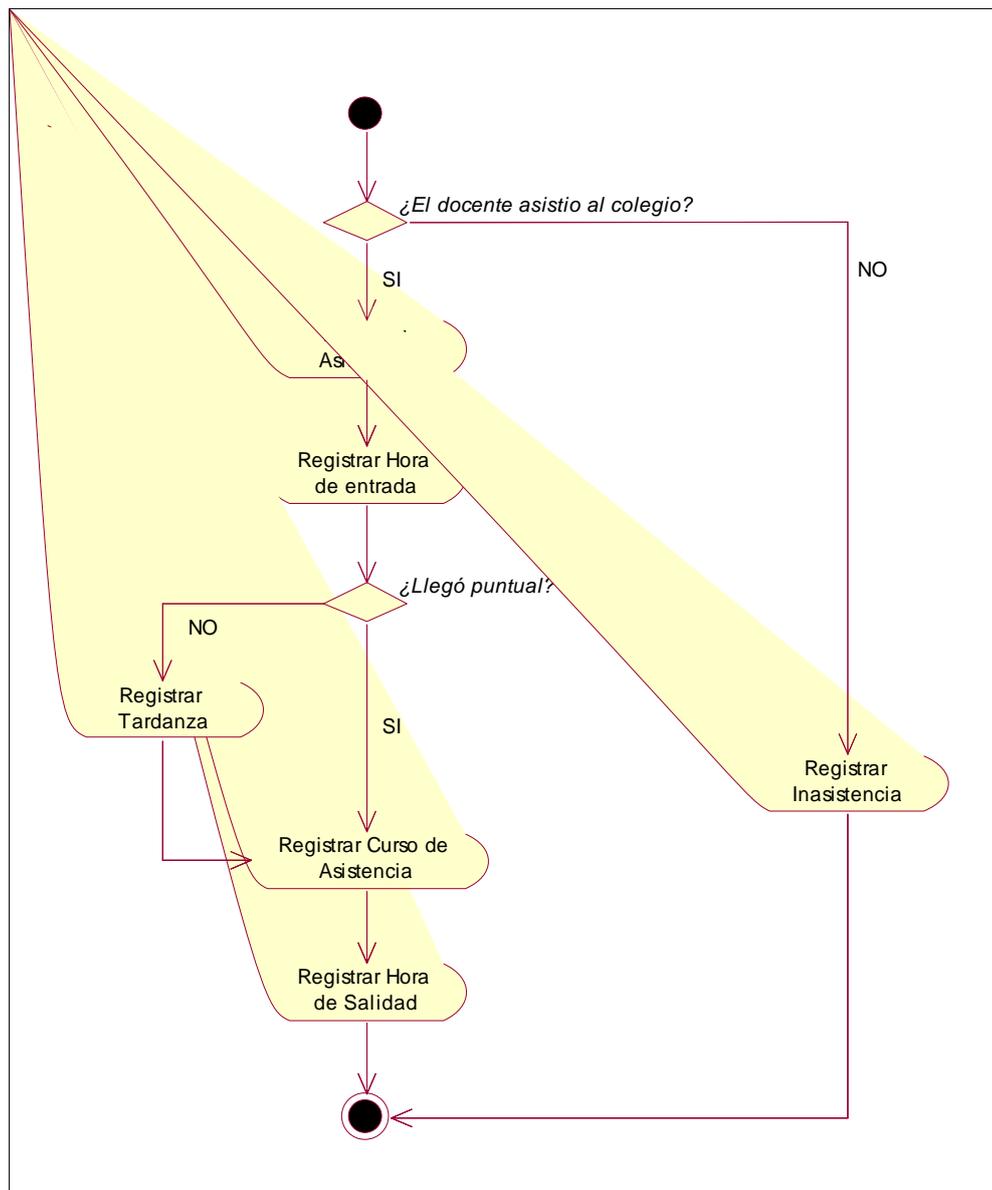


Figura 37: Diagrama de Actividad Registrar Asistencia Docente
Fuente: Elaboración Propia

4.2.4.13. Consultar Tema de Clase

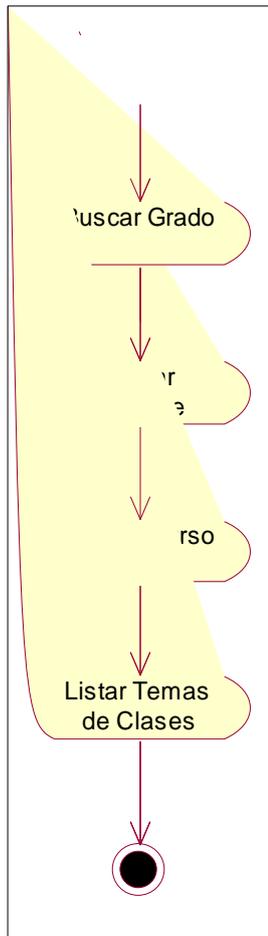


Figura 38: Diagrama de Actividad Consultar tema de Clase
Fuente: Elaboración Propia

4.2.4.14. Registrar Asistencia de Alumnos

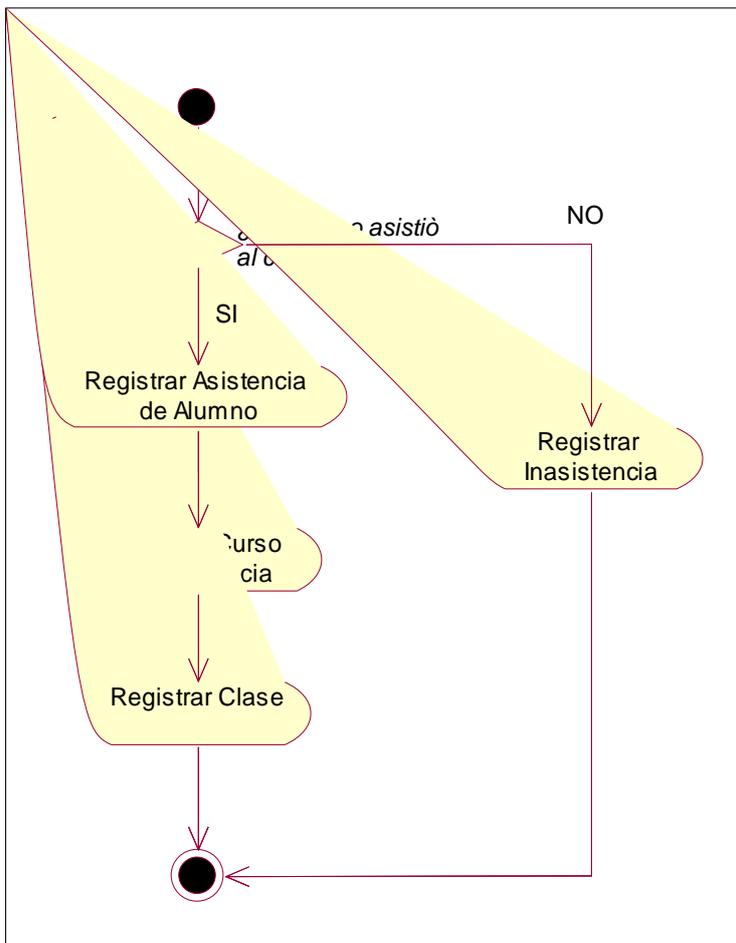


Figura 39: Diagrama de Actividad Registrar Asistencia de Alumno
Fuente: Elaboración Propia

4.2.4.15. Registrar Tema de Clase

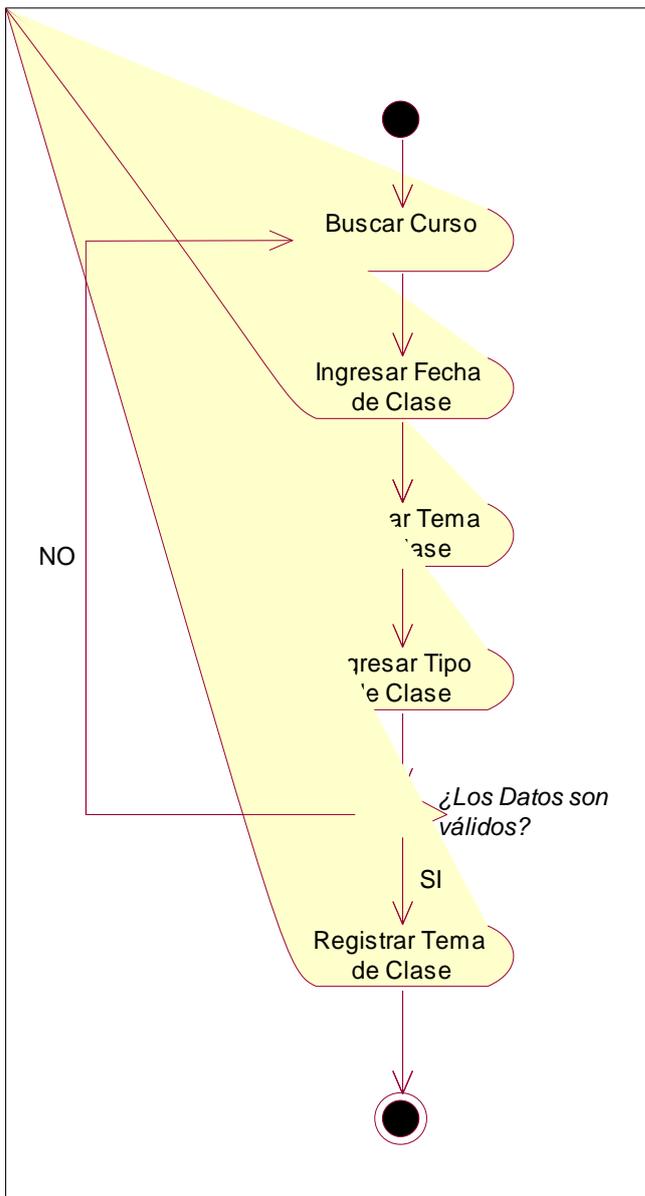


Figura 40: Diagrama de Actividad Registrar tema de Clase
Fuente: Elaboración Propia

4.2.4.16. Registrar Evaluación

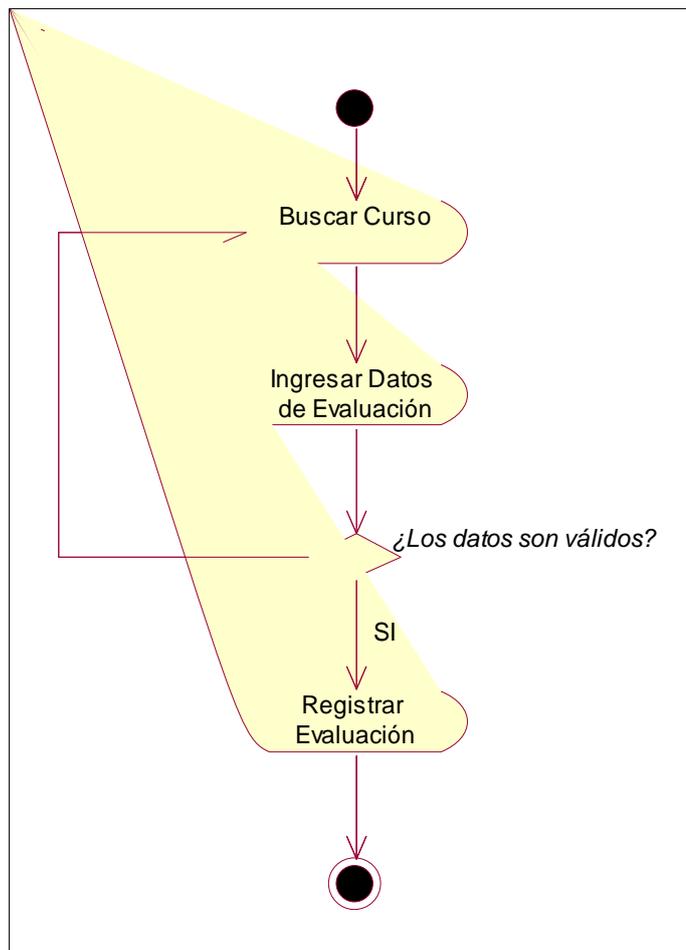


Figura 41: Diagrama de Actividad Registrar Evaluación
Fuente: Elaboración Propia

4.2.4.17. Registrar Nota

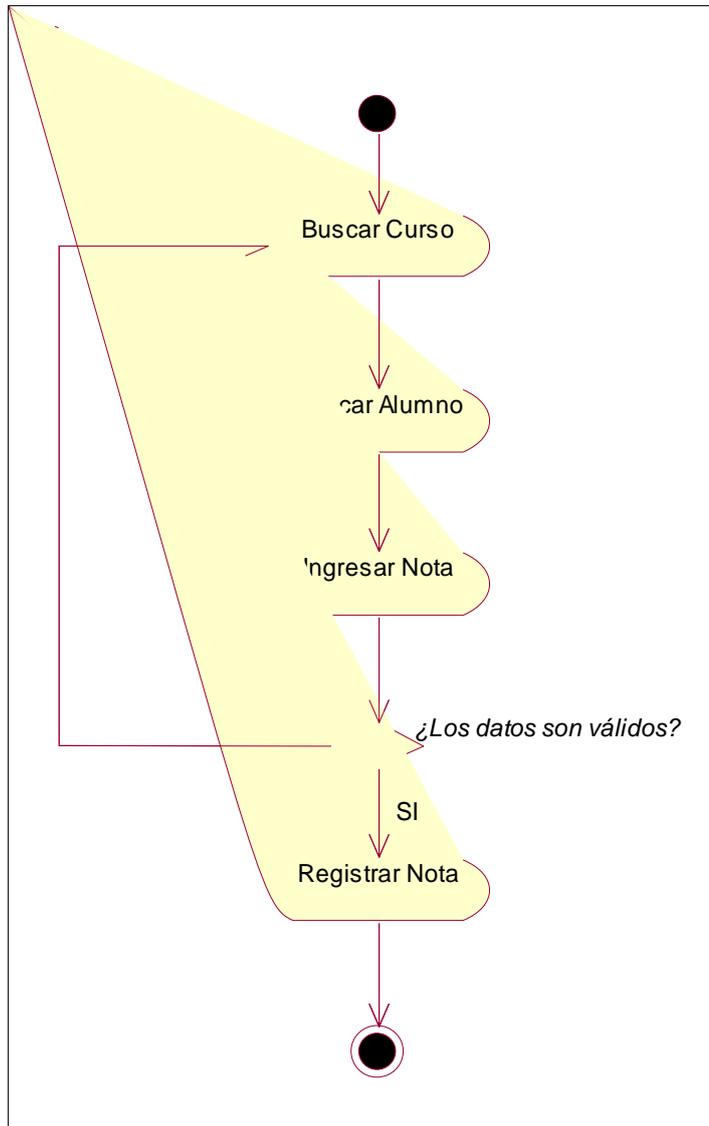


Figura 42: Diagrama de Actividad Registrar Nota
Fuente: Elaboración Propia

4.2.4.18. Registrar Requisito de promoción

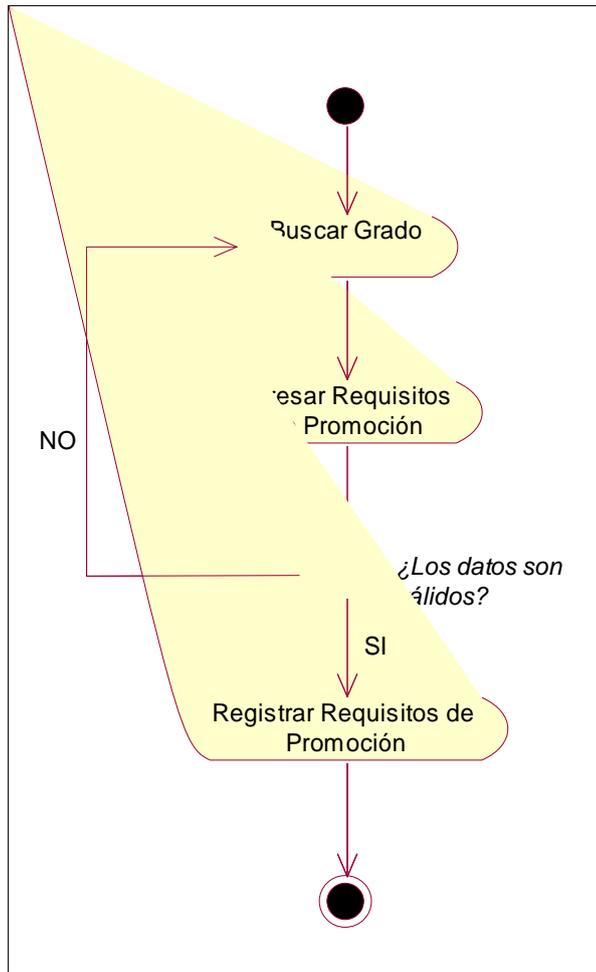


Figura 43: Diagrama de Actividad Registrar Requisitos de Promoción
Fuente: Elaboración Propia

4.3.Análisis y Diseño

4.3.1. Diagramas de Colaboración

4.3.1.1.Registrar Solicitud de Admisión

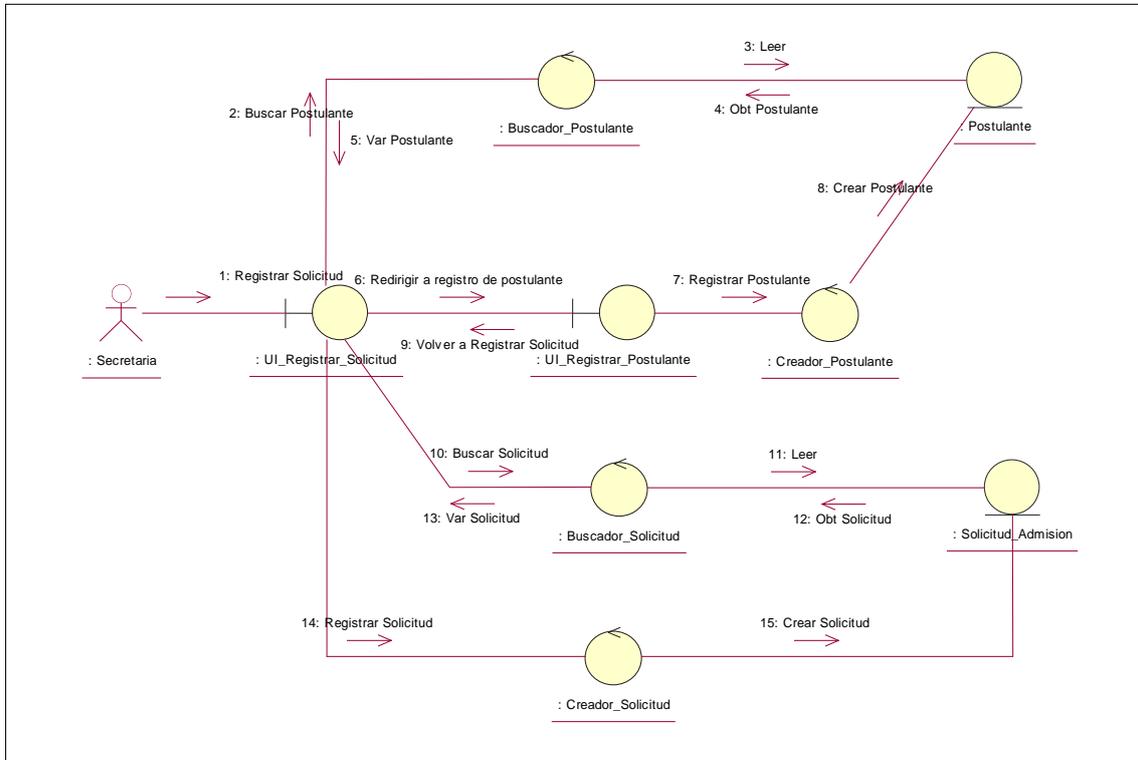


Figura 44: Diagrama de Colaboración Registrar Solicitud de Admisión
Fuente: Elaboración Propia

4.3.1.2.Registrar Alumno

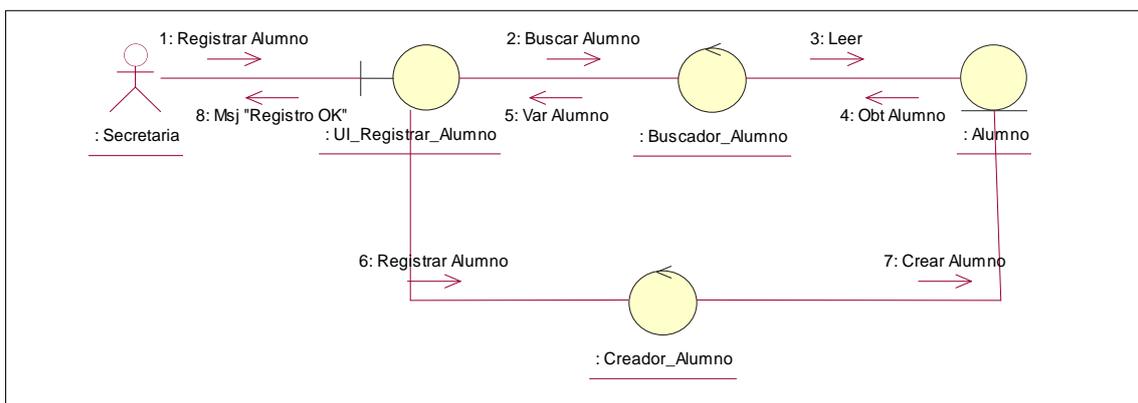


Figura 45: Diagrama de Colaboración Registrar alumno
Fuente: Elaboración Propia

4.3.1.3.Registrar Matricula

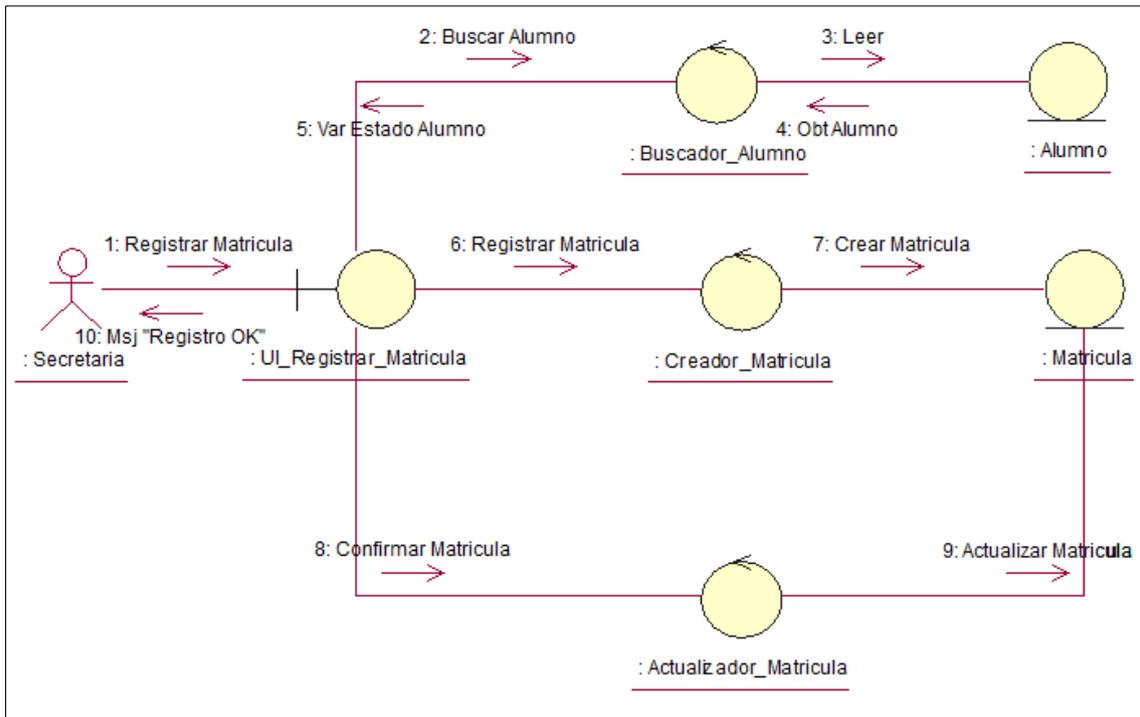


Figura 46: Diagrama de Colaboración Registrar Matricula
Fuente: Elaboración Propia

4.3.1.4.Evaluar Solicitud

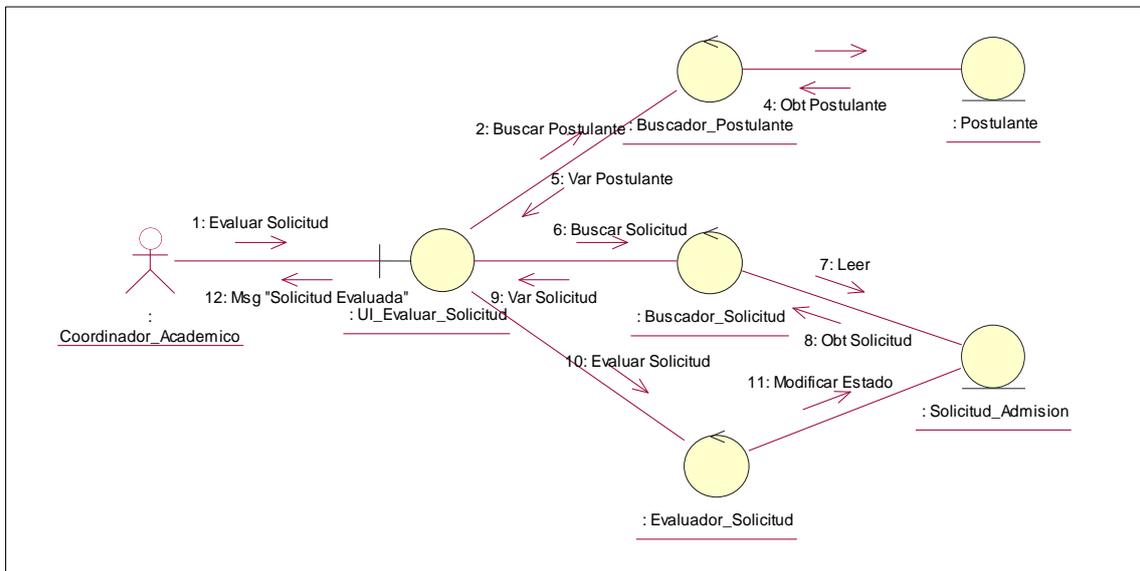


Figura 47: Diagrama de Colaboración Evaluar Solicitud
Fuente: Elaboración Propia

4.3.1.5.Registrar Docente

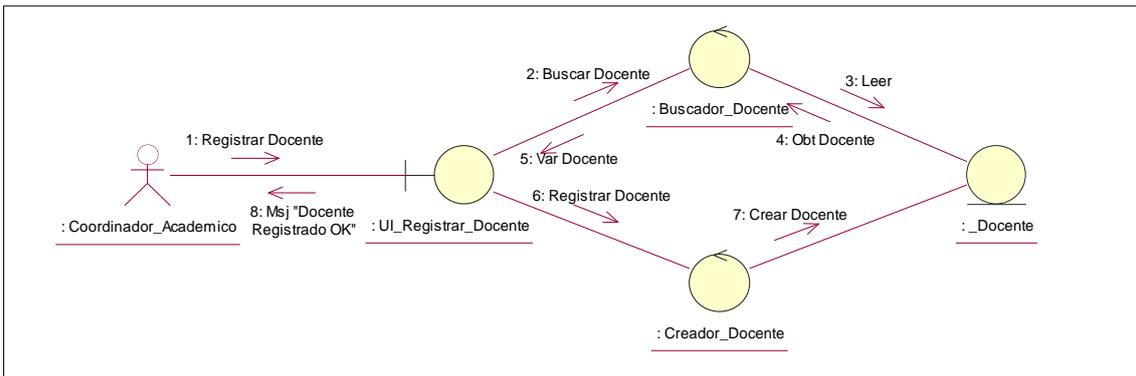


Figura 48: Diagrama de colaboración registrar docente
Fuente: Elaboración Propia

4.3.1.6.Registrar Aula

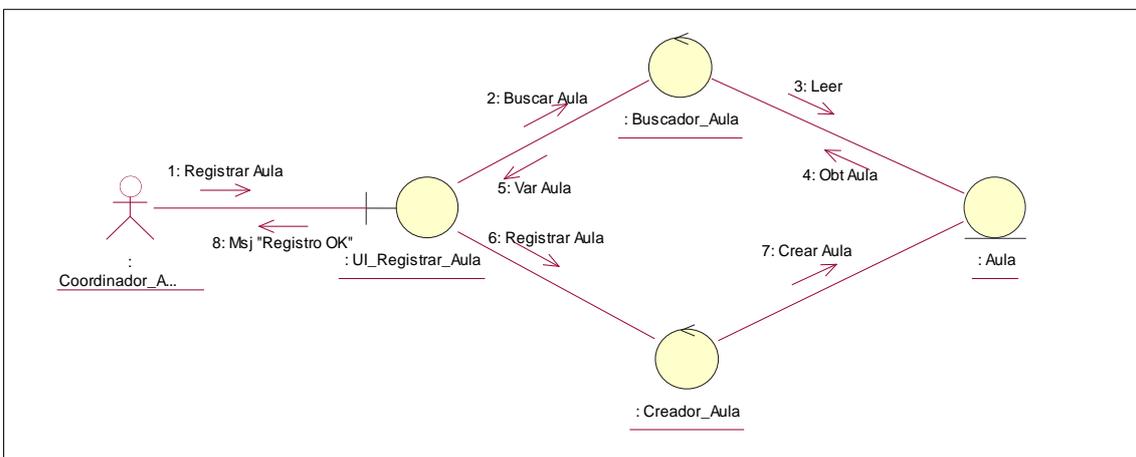


Figura 49: Diagrama de Colaboración Registrar Aula
Fuente: Elaboración Propia

4.3.1.7.Registrar Curso

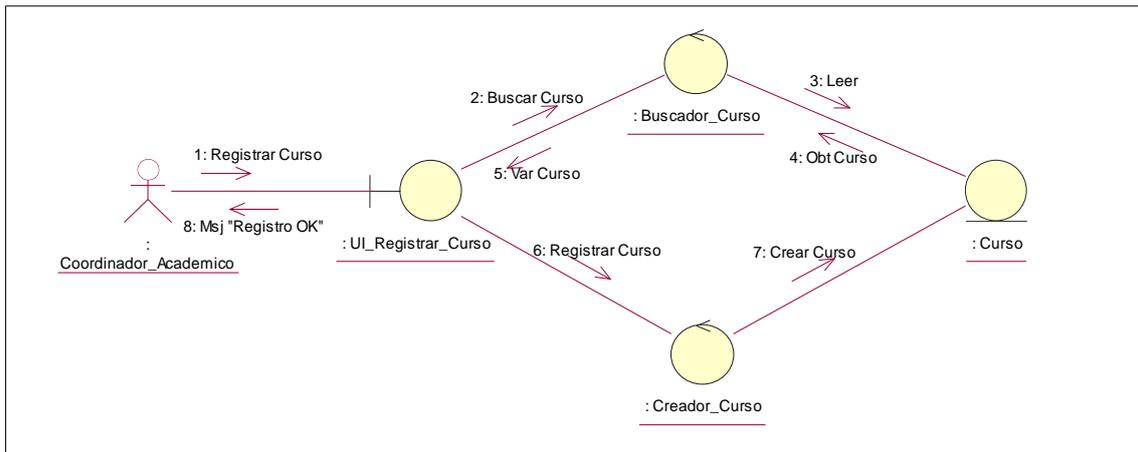


Figura 50: Diagrama de Colaboración Registrar Curso
Fuente: Elaboración Propia

4.3.1.8.Registrar Plan de Estudios

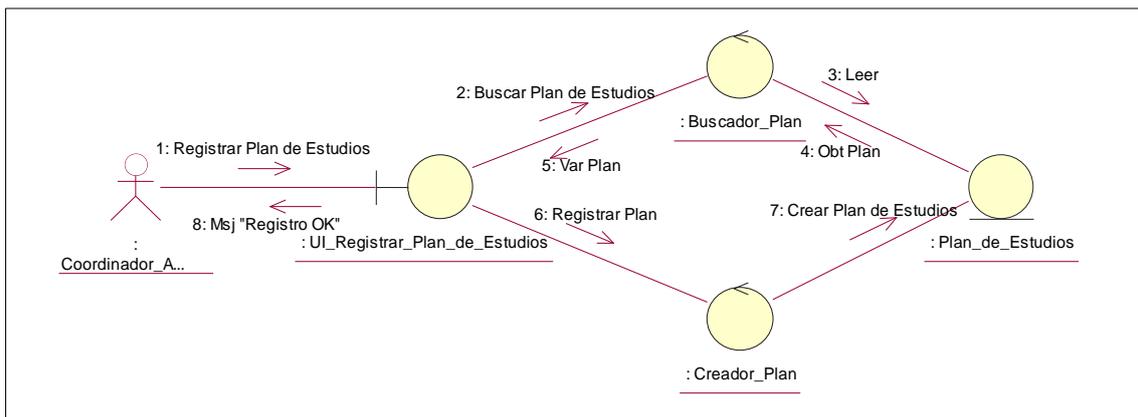


Figura 51: Diagrama de Colaboracion Registrar Plan de Estudios
Fuente: Elaboración Propia

4.3.1.9.Registrar Horario

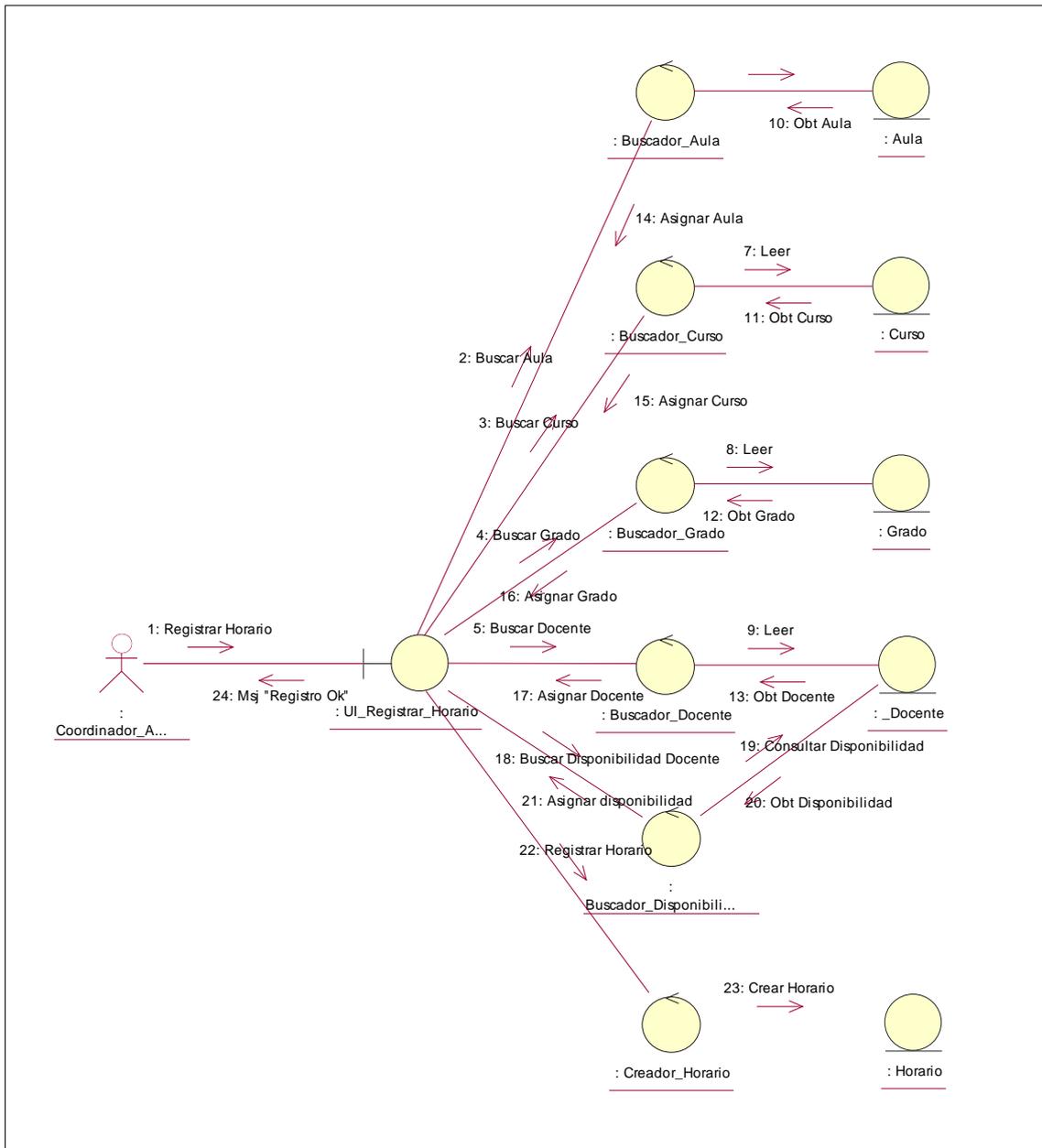


Figura 52: Diagrama de Colaboración Registrar Horario
Fuente: Elaboración Propia

4.3.1.10. Registrar Disponibilidad Horaria



Figura 53: Diagrama de colaboración Registrar Disponibilidad Horaria
Fuente: Elaboración Propia

4.3.1.11. Aprobar Horario

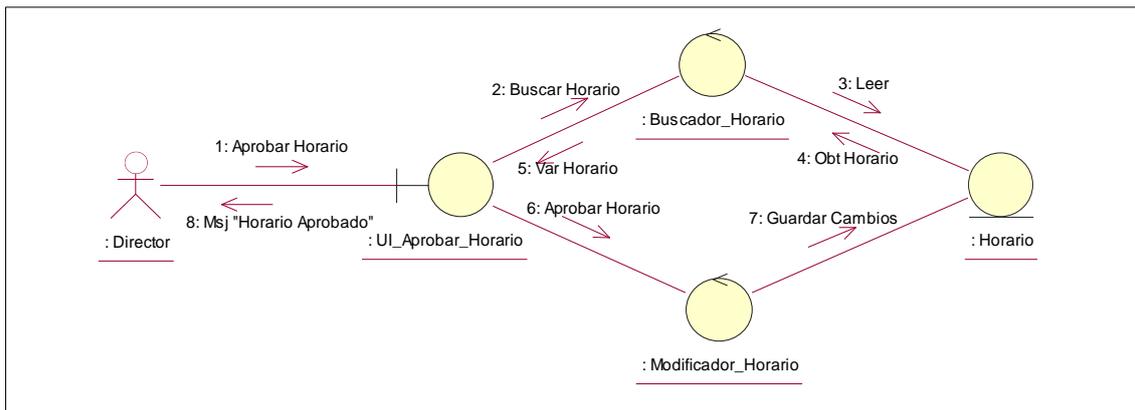


Figura 54: Diagrama de Colaboración Aprobar Horario
Fuente: Elaboración Propia

4.3.1.12. Registrar Asistencia de Docente

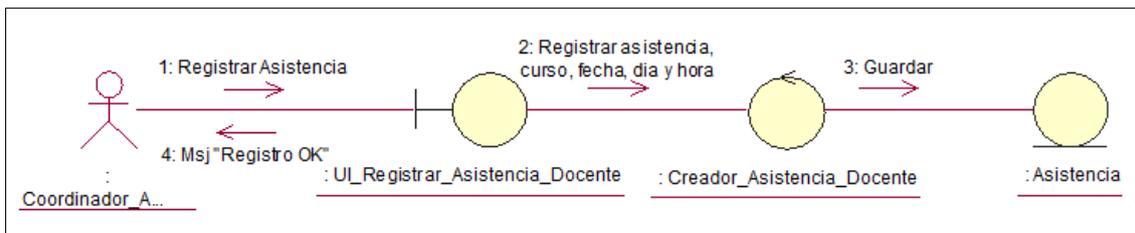


Figura 55: Diagrama de Colaboración Registrar Asistencia Docente
Fuente: Elaboración Propia

4.3.1.13. Consultar Tema de Clase

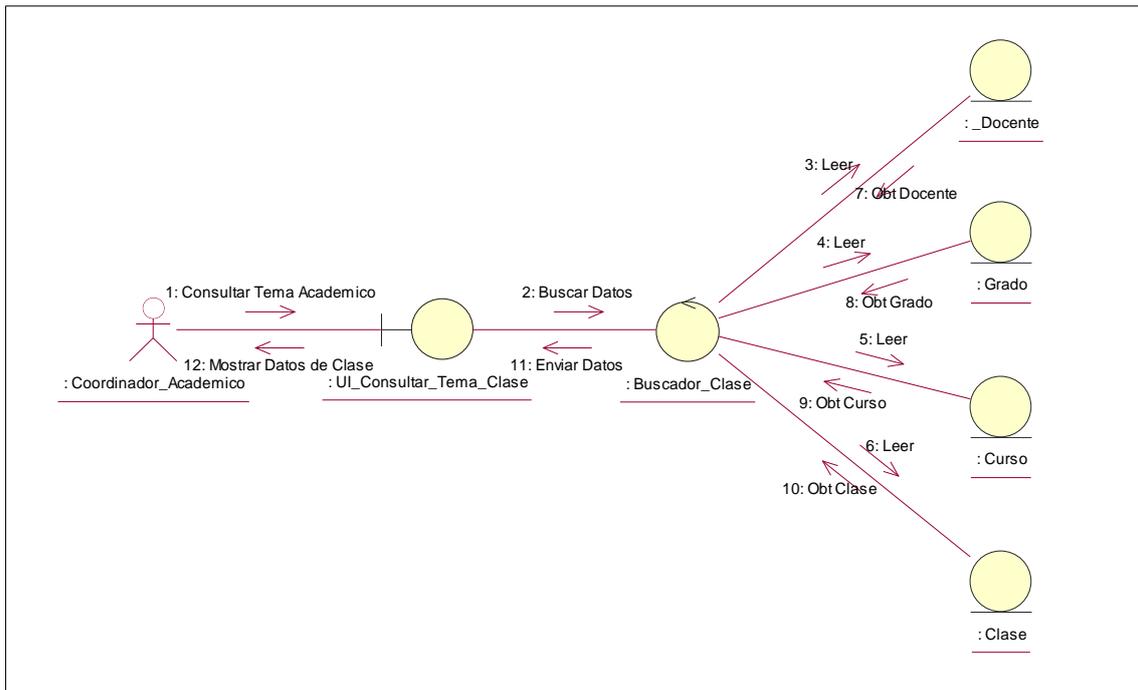


Figura 56: Diagrama de colaboración Consultar Tema de clase
Fuente: Elaboración Propia

4.3.1.14. Registrar Asistencia de Alumnos

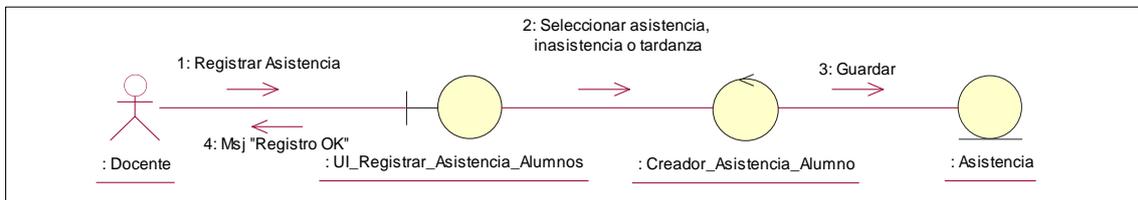


Figura 57: Diagrama de Colaboración Registrar Asistencia de Alumnos
Fuente: Elaboración Propia

4.3.1.15. Registrar Clase

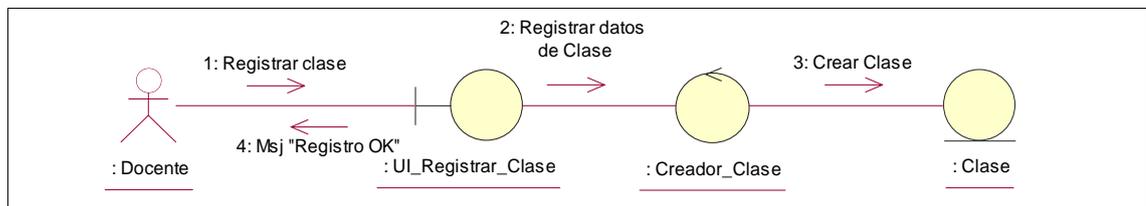


Figura 58: Diagrama de colaboración Registrar Clase
Fuente: Elaboración Propia

4.3.1.16. Registrar Evaluación

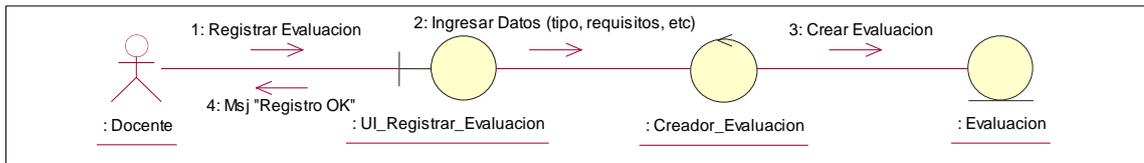


Figura 59: Diagrama de Colaboración Registrar Evaluación
Fuente: Elaboración Propia

4.3.1.17. Registrar Nota

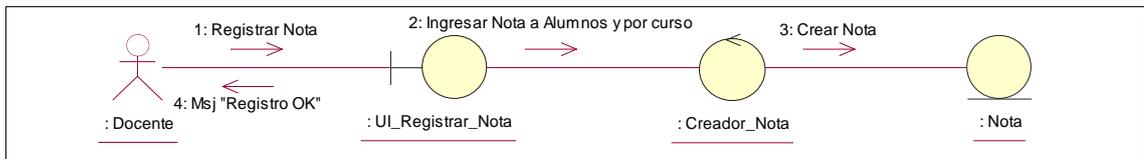


Figura 60: Diagrama de Colaboración Registrar Nota
Fuente: Elaboración Propia

4.3.1.18. Registrar Requisito de promoción

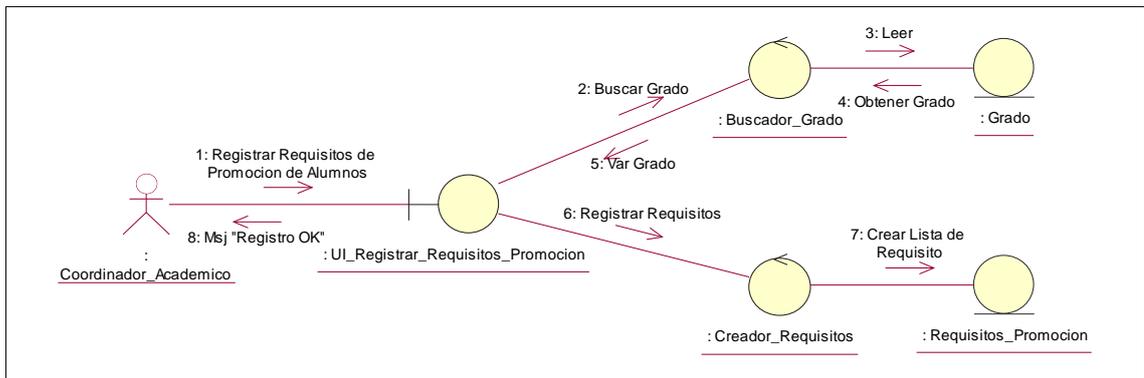


Figura 61: Diagrama de Colaboracion Registrar Requisitos de Promocion de alumnos
Fuente: Elaboración Propia

4.3.2. Diagramas de Secuencia

4.3.2.1.Registrar solicitud de Admisión

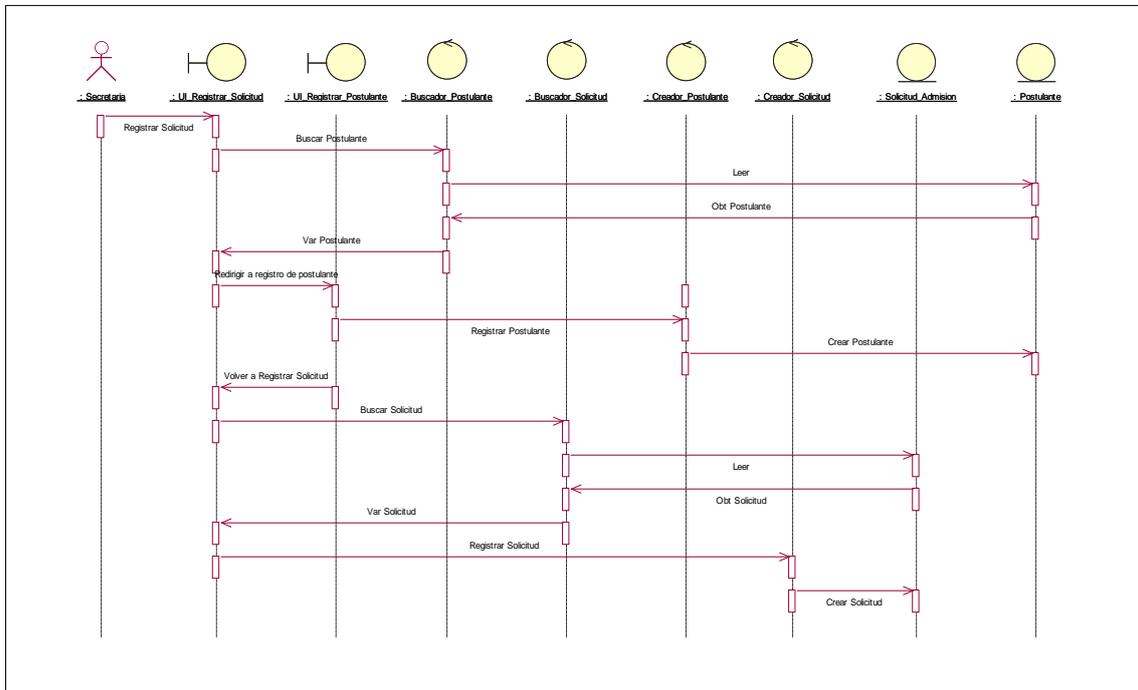


Figura 62: Diagrama de Secuencia Registrar Solicitud de Admisión
Fuente: Elaboración Propia

4.3.2.2.Registrar Alumno

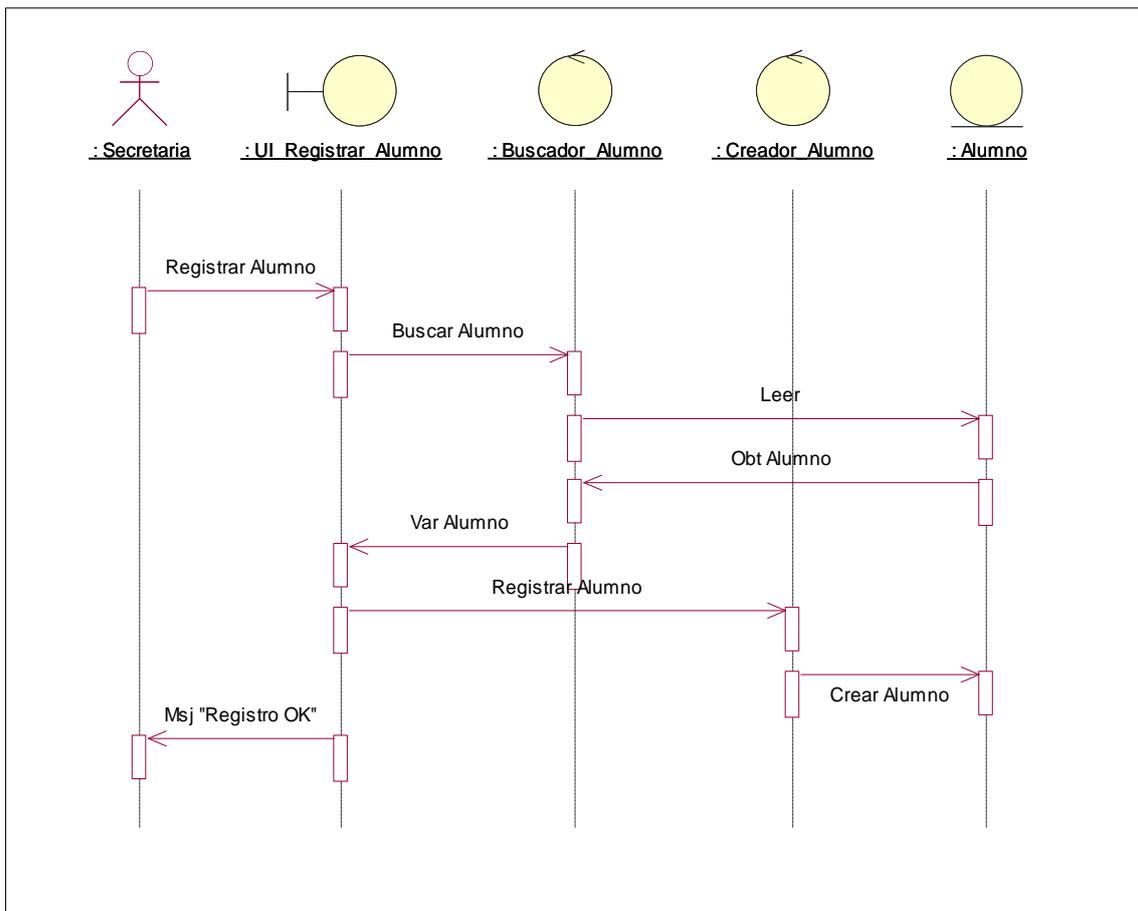


Figura 63: Diagrama de Secuencia Registrar Alumno
Fuente: Elaboración Propia

4.3.2.3.Registrar Matricula

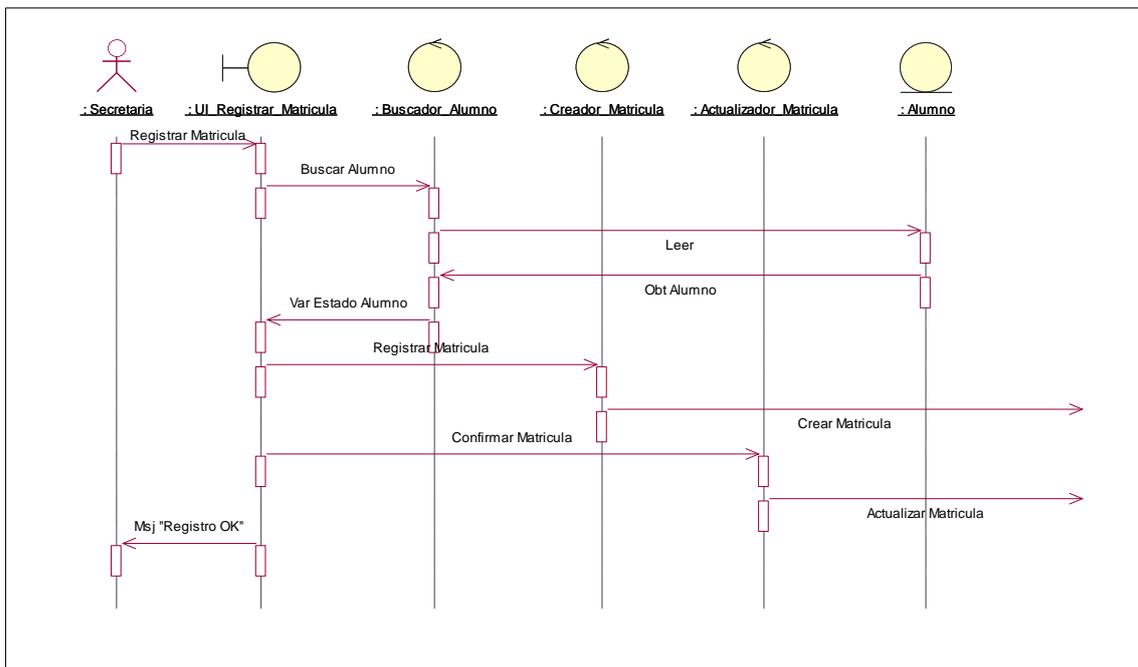


Figura 64: Diagrama de Secuencia Registrar Matrícula
Fuente: Elaboración Propia

4.3.2.4. Evaluar Solicitud

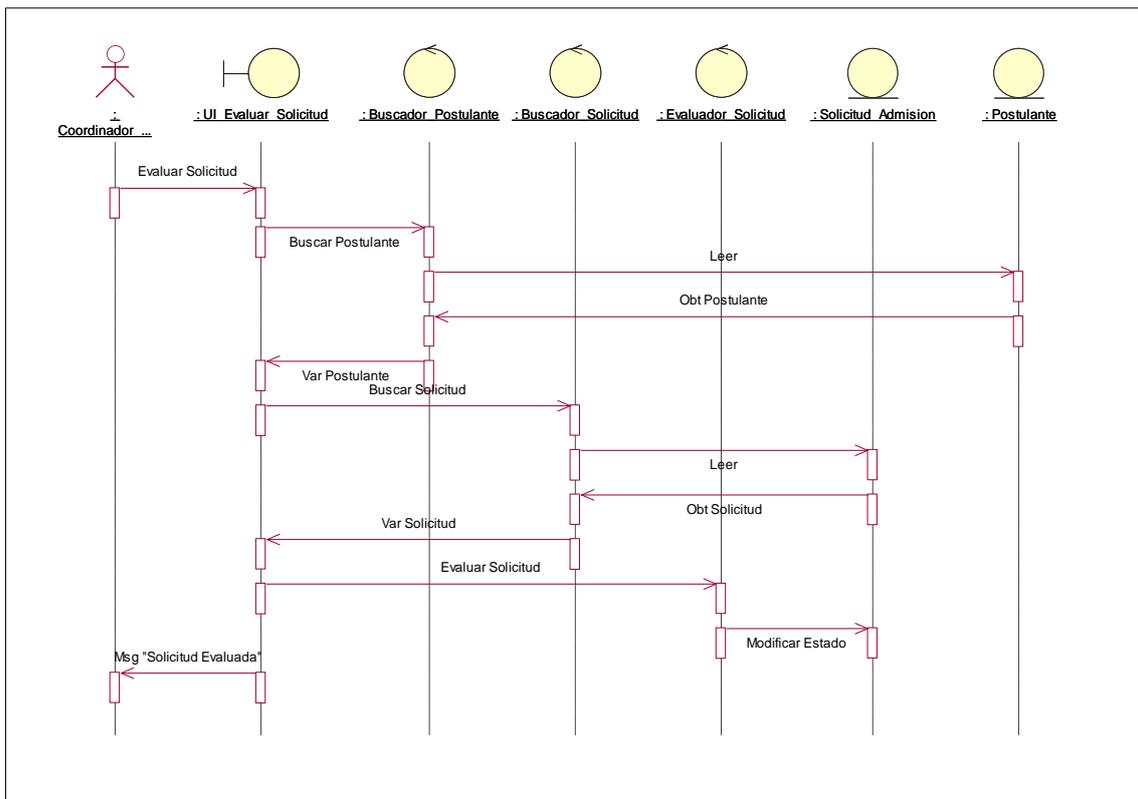


Figura 65: Diagrama de Secuencia Evaluar Solicitud
Fuente: Elaboración Propia

4.3.2.5.Registrar Docente

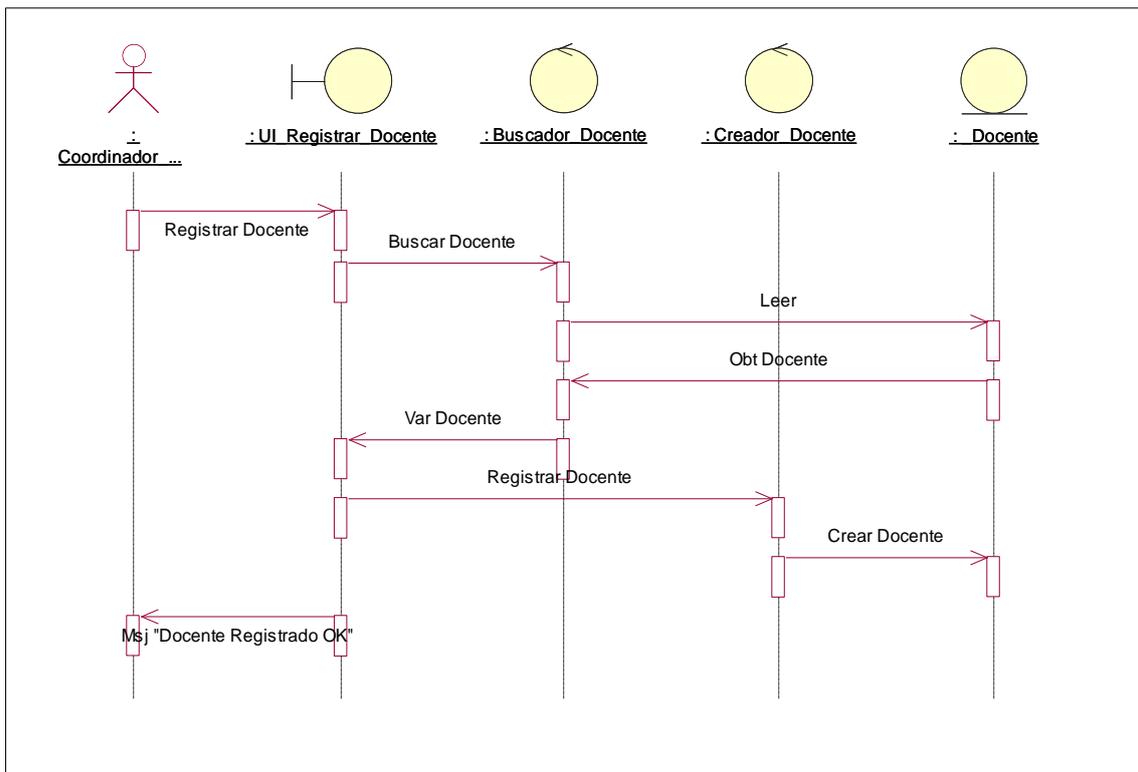


Figura 66: Diagrama de Secuencia Registrar Docente

Fuente: Elaboración Propia

4.3.2.6.Registrar Aula

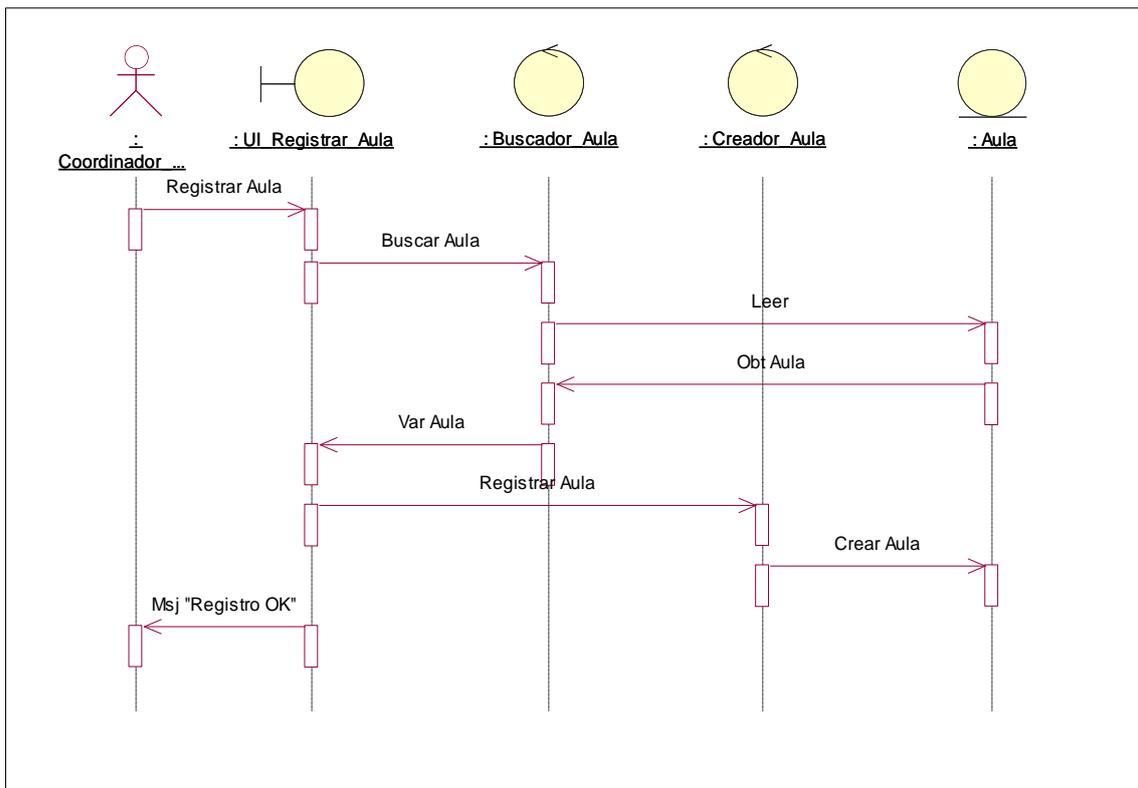


Figura 67: Diagrama de Secuencia Registrar Aula

Fuente: Elaboración Propia

4.3.2.7.Registrar Curso

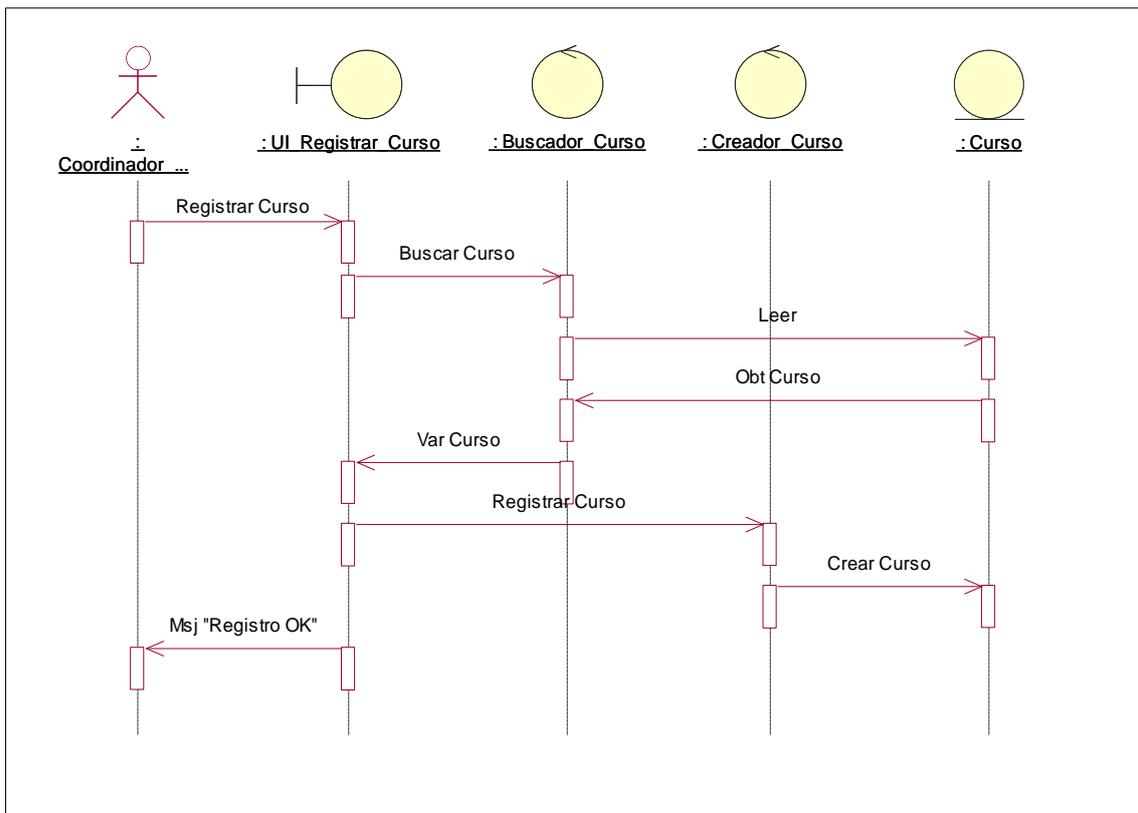


Figura 68: Diagrama de Secuencia Registrar Curso

Fuente: Elaboración Propia

4.3.2.8.Registrar Plan de Estudios

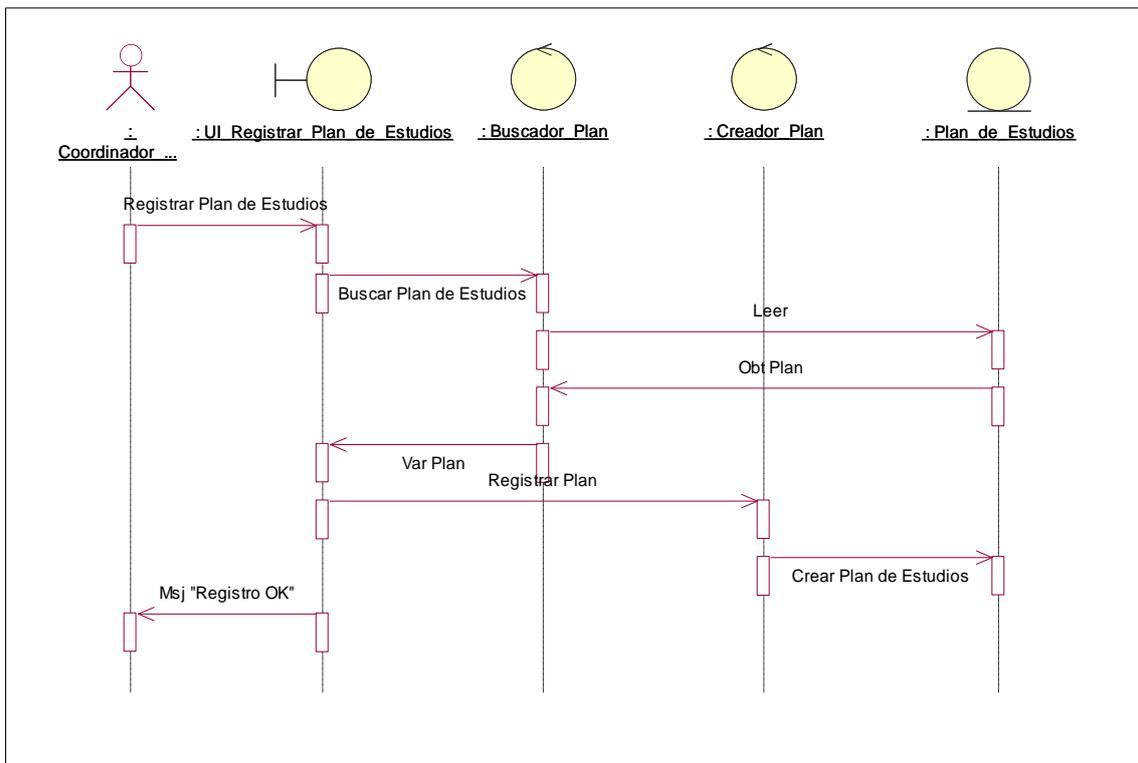


Figura 69: Diagrama de Secuencia Registrar Plan de Estudios

Fuente: Elaboración Propia

4.3.2.9.Registrar Horario

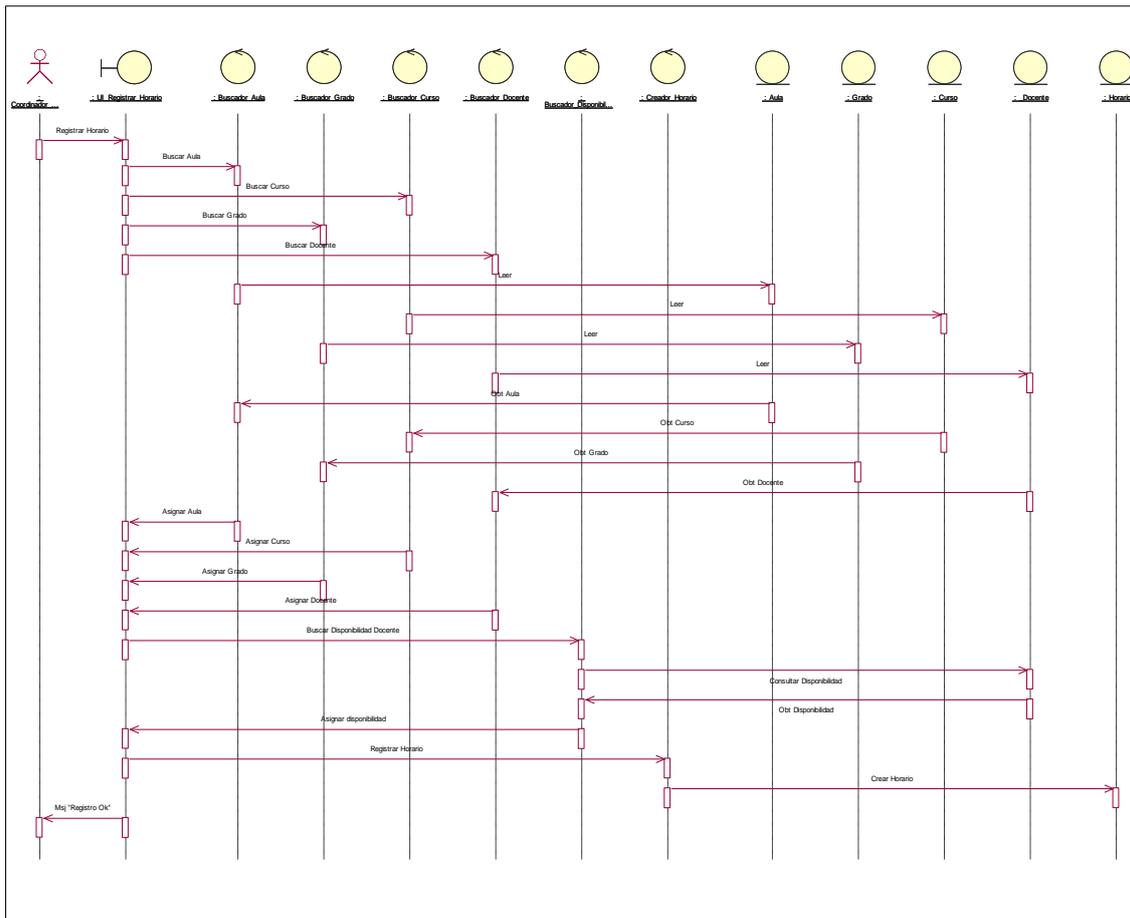


Figura 70: Diagrama de Secuencia Registrar Horario
Fuente: Elaboración Propia

4.3.2.10. Registrar Disponibilidad Horaria

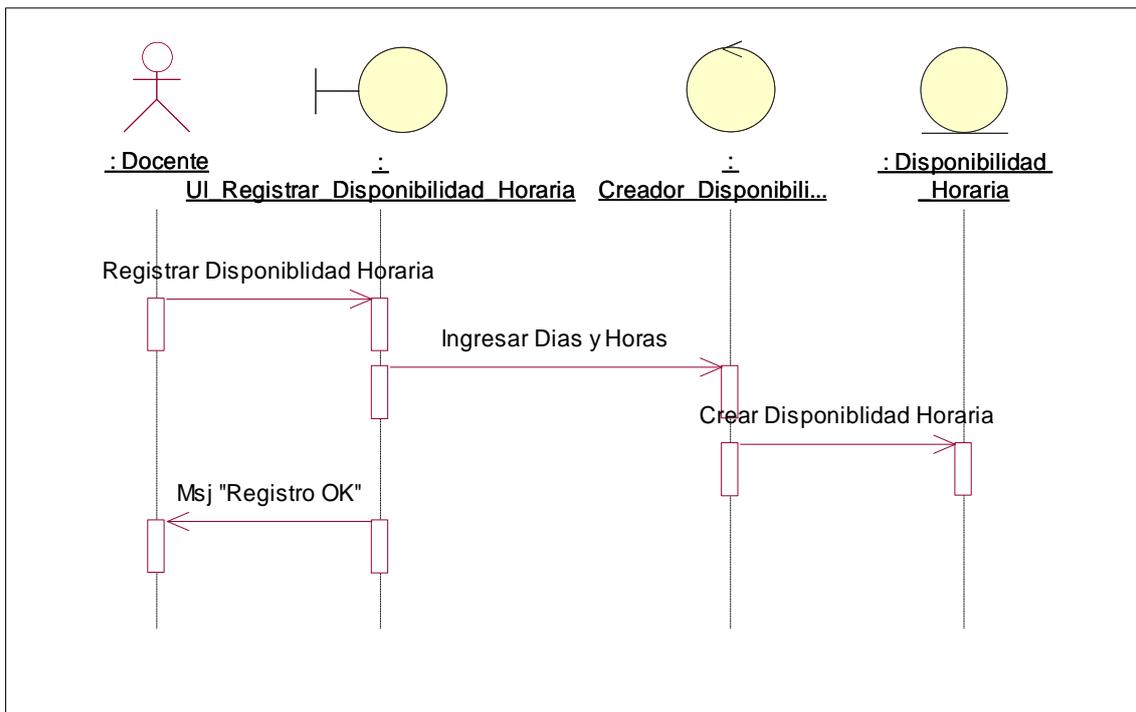


Figura 71: Diagrama de Secuencia Registrar Disponibilidad Horaria
Fuente: Elaboración Propia

4.3.2.11. Aprobar Horario

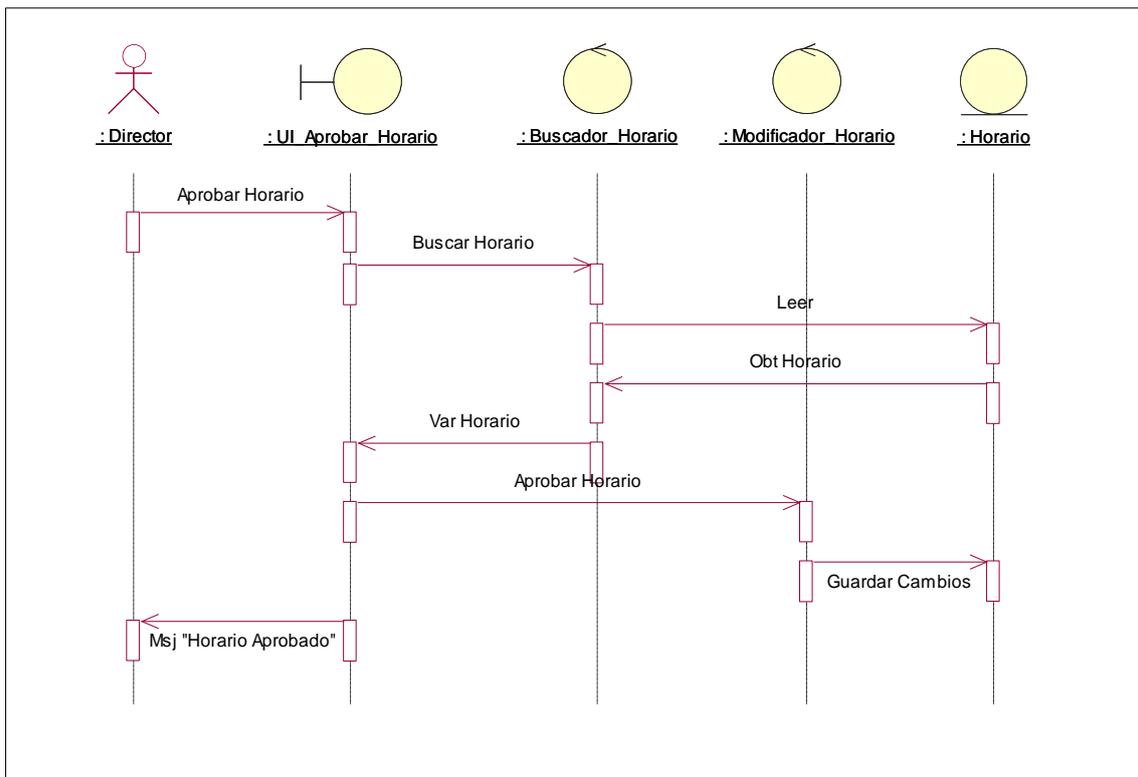


Figura 72: Diagrama de Secuencia Aprobar Horario

Fuente: Elaboración Propia

4.3.2.12. Registrar Asistencia Docente

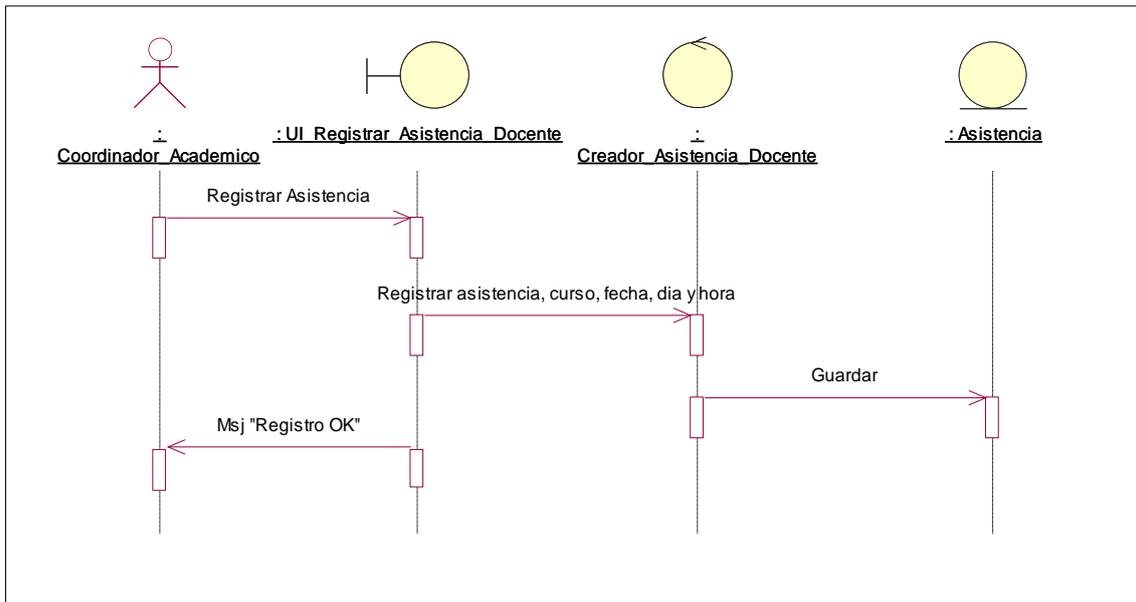


Figura 73: Diagrama de Secuencia Registrar Asistencia de Docente
Fuente: Elaboración Propia

4.3.2.13. Consultar Tema de Clase

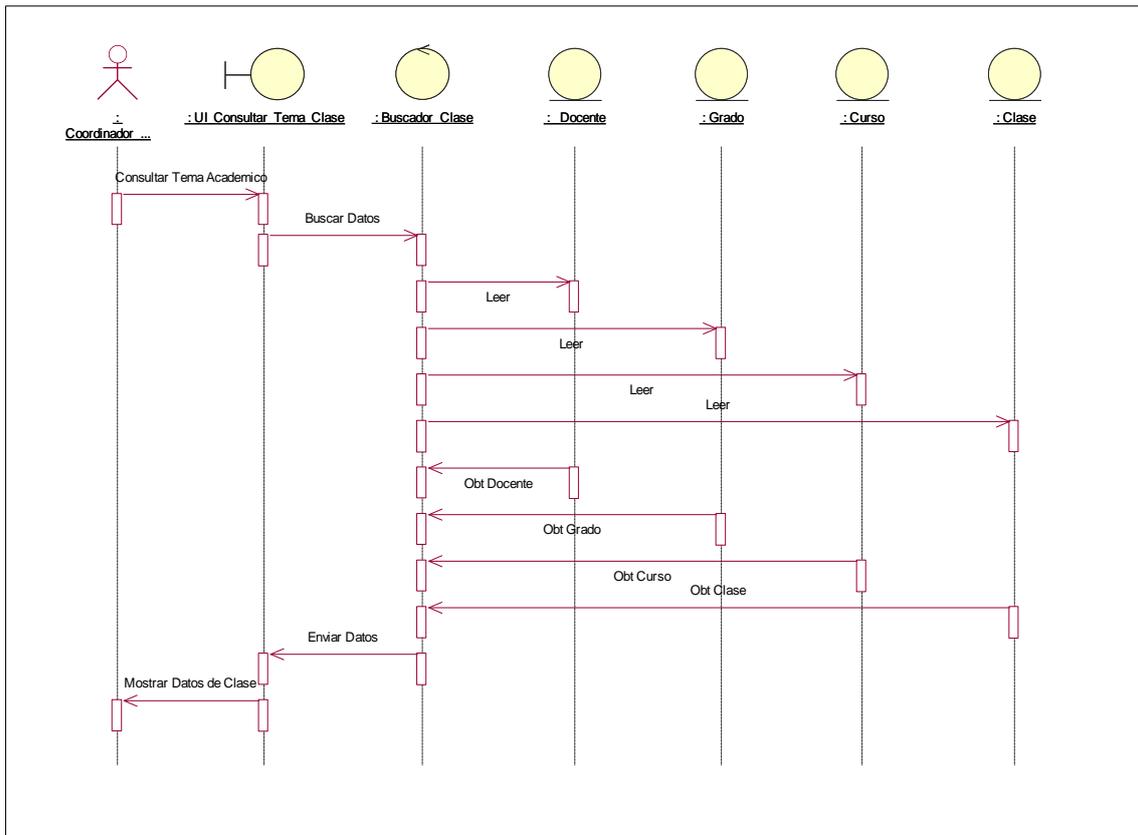


Figura 74: Diagrama de Secuencia Consultar Tema de Clase
Fuente: Elaboración Propia

4.3.2.14. Registrar Asistencia de Alumnos

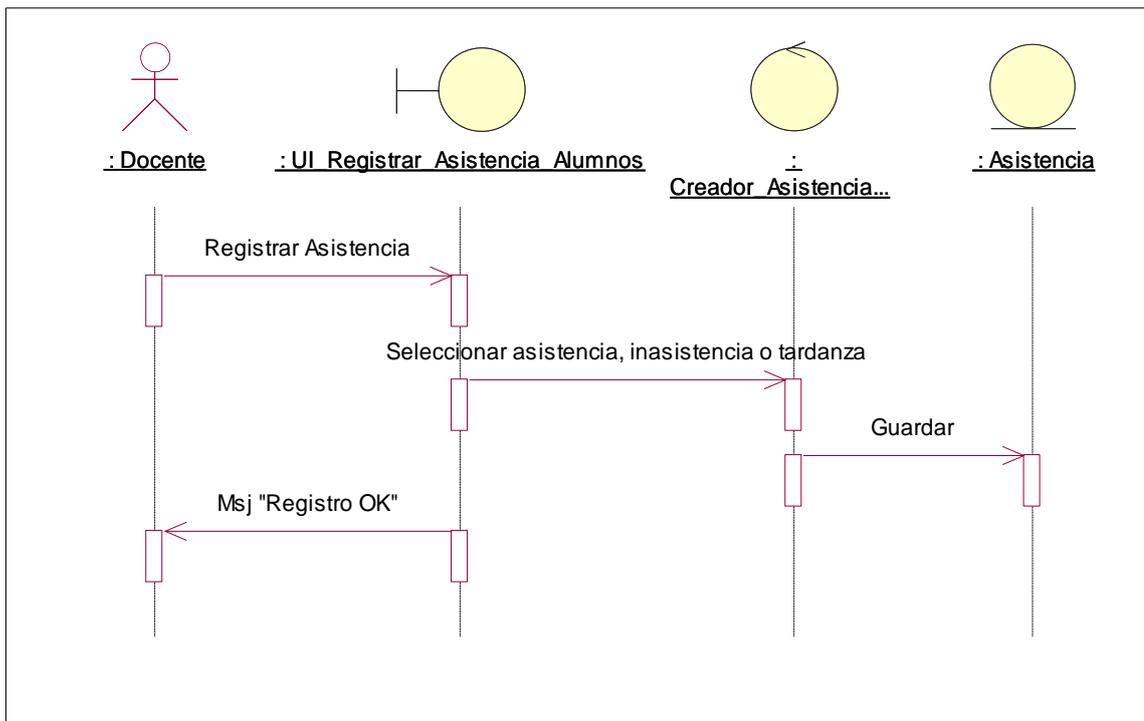


Figura 75: Diagrama de Secuencia Registrar Asistencia de Docente
Fuente: Elaboración Propia

4.3.2.15. Registrar Clase

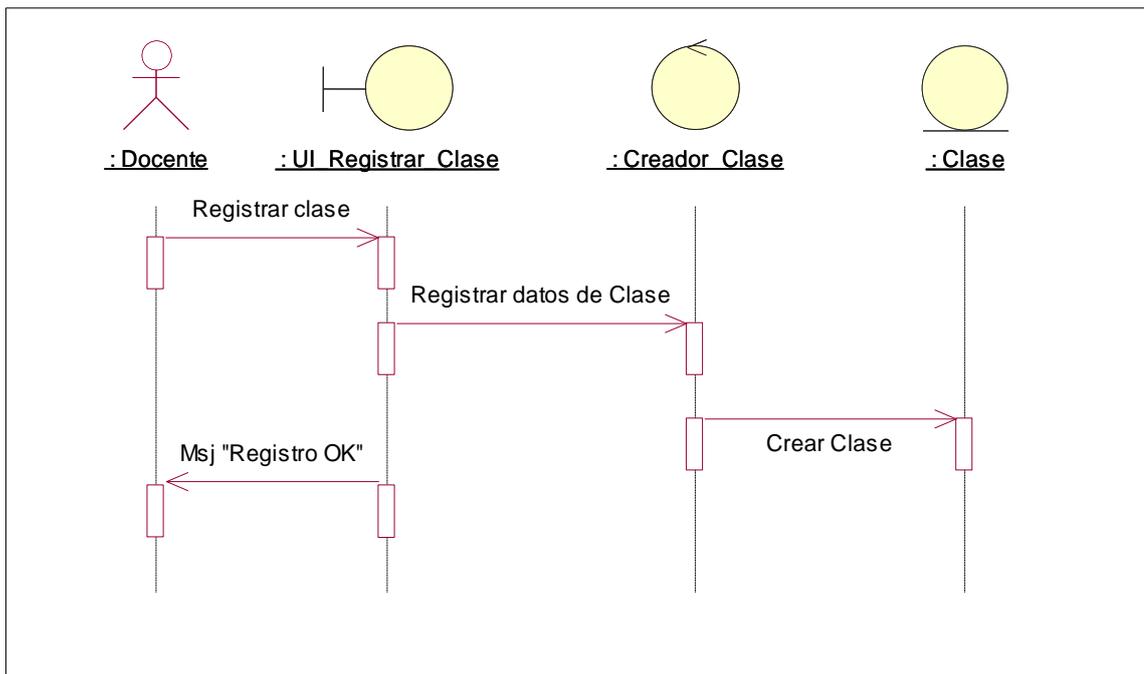


Figura 76: Diagrama de Secuencia Registrar Clase
Fuente: Elaboración Propia

4.3.2.16. Registrar Método de Evaluación

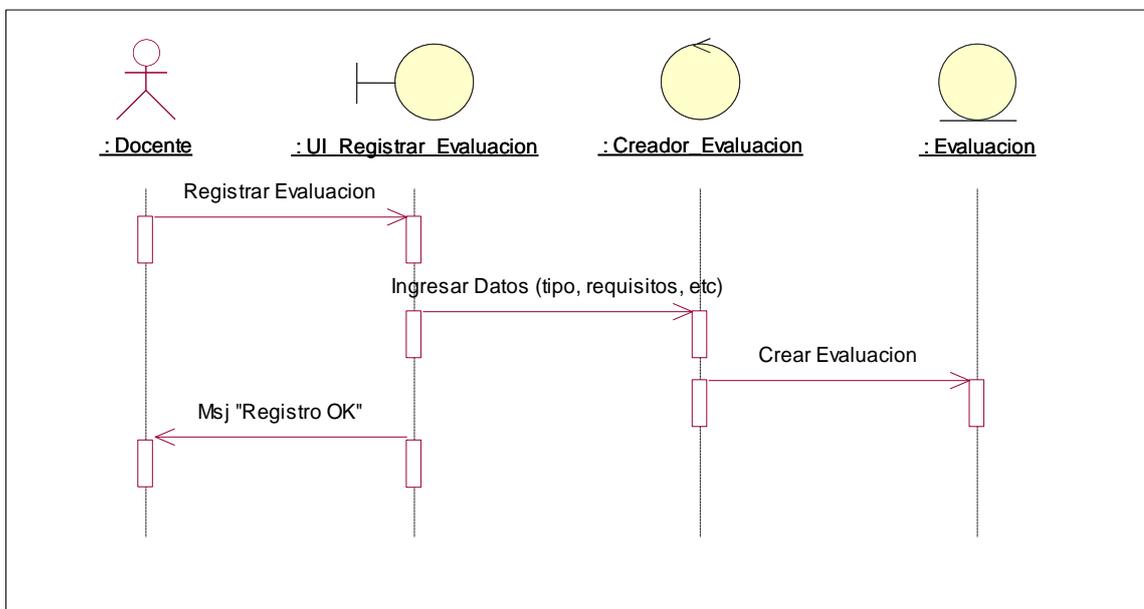


Figura 77: Registrar Método de Evaluación
Fuente: Elaboración Propia

4.3.2.17. Registrar Nota

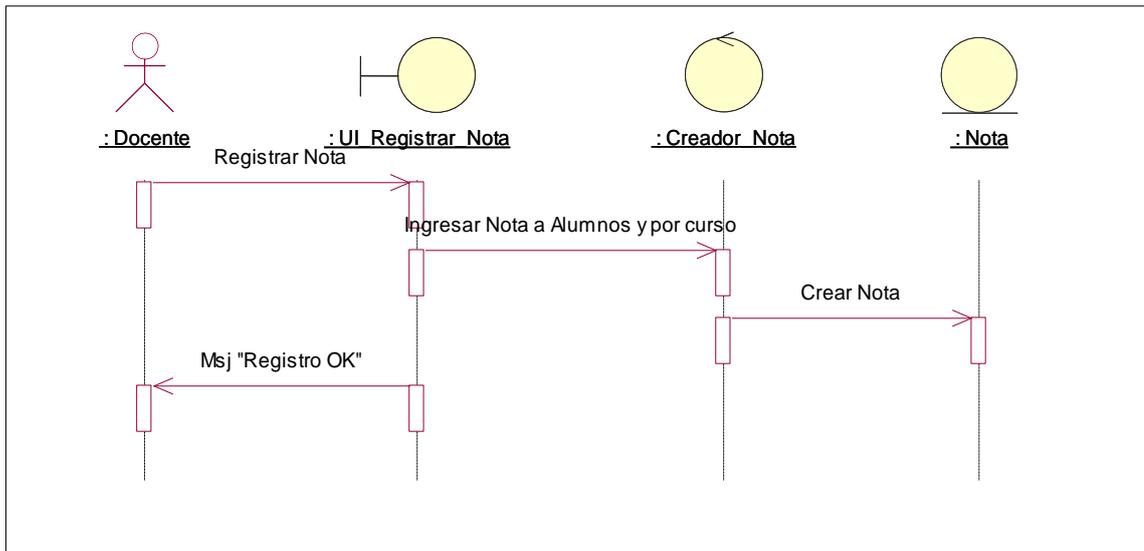


Figura 78: Diagrama de Secuencia Registrar Nota
Fuente: Elaboración Propia

4.3.2.18. Registrar Requisitos de Promoción

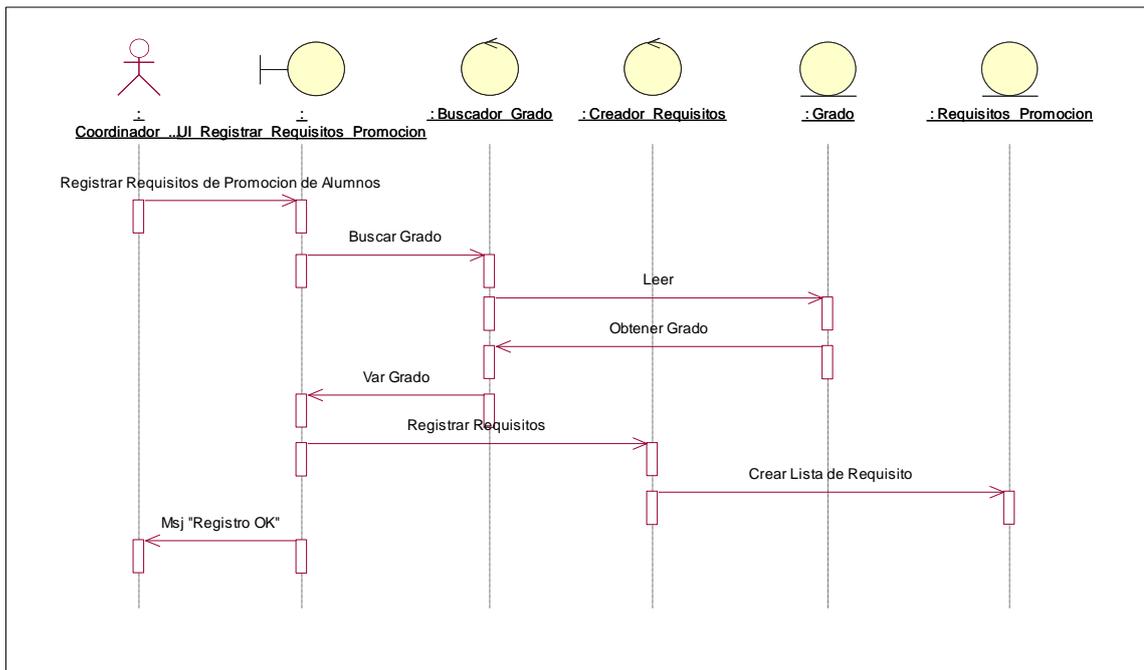


Figura 79: Diagrama de Secuencia Registrar Requisitos de Promoción
Fuente: Elaboración Propia

4.3.3. Modelo Físico

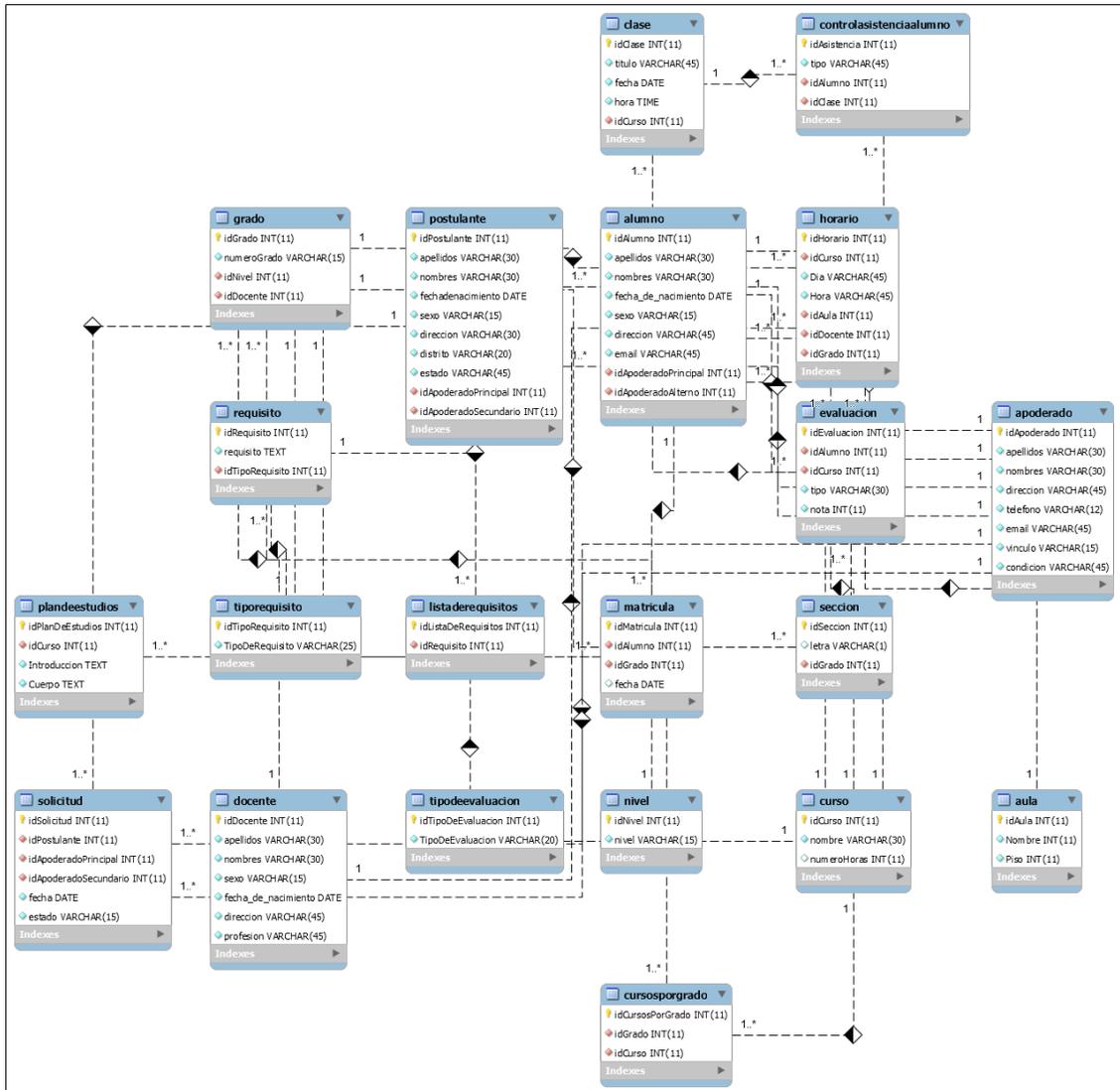


Figura 80: Modelo Físico de la Base de Datos
Fuente: Elaboración Propia

4.3.4. Pantallas

4.3.4.1.Registrar Postulante

Desarrolla tu aplicación X CakePHP: the rapid dev X

localhost/tesis/postulante/add

Documentation API

Postulante

ACTIONS

List Postulante

Add Postulante

Apellidos *

Nombres *

Fecha de nacimiento *

2016 September 19

Sexo *

Direccion *

Distrito *

Estado *

Figura 81: Pantalla Registrar Postulante
Fuente: Elaboración Propia

4.3.4.2. Listar postulante

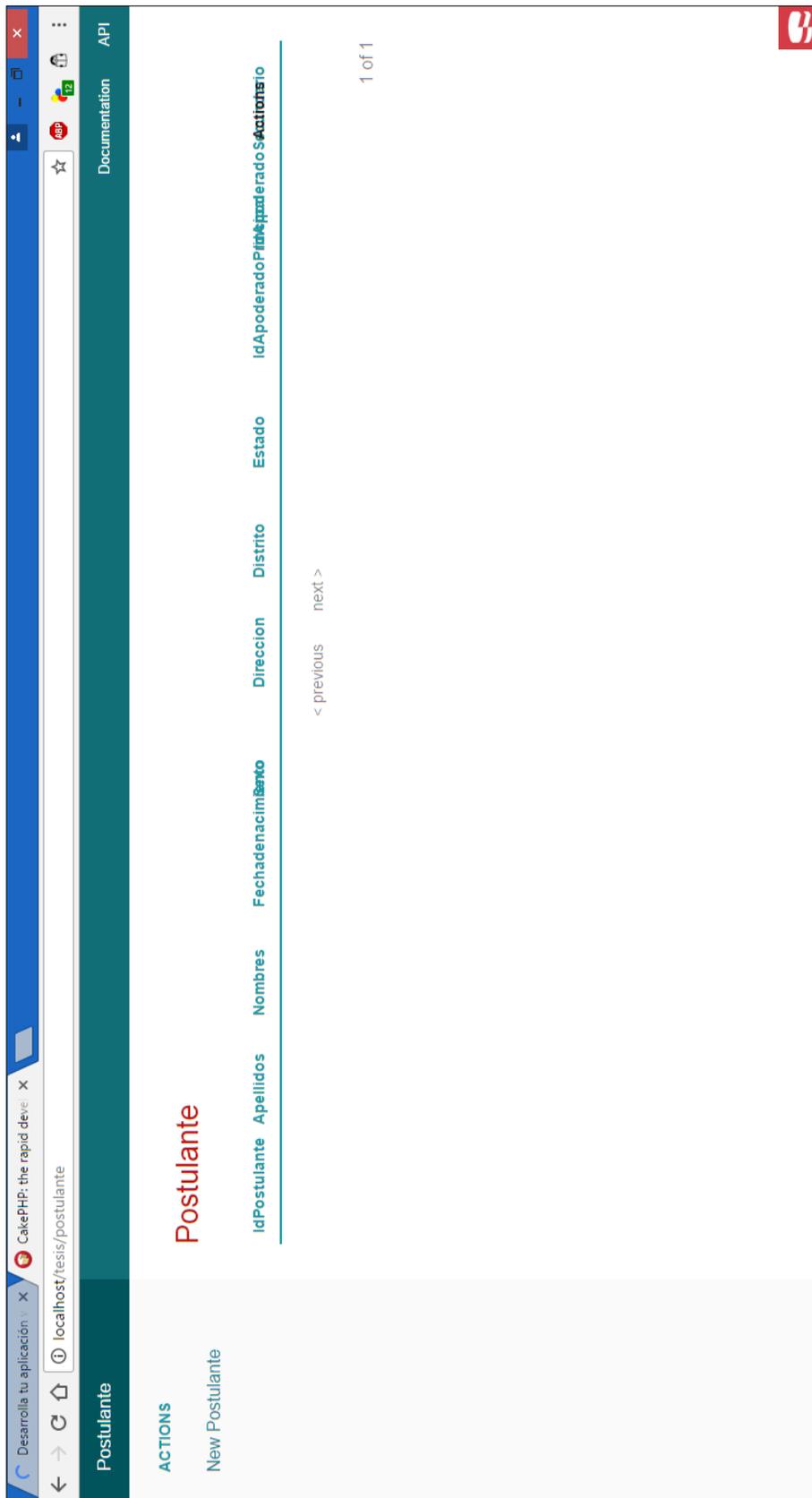


Figura 82: Pantalla Listar Postulante
Fuente: Elaboración Propia

4.3.4.3.Registrar apoderado

The screenshot shows a web browser window with the following elements:

- Browser Tabs:** 'Desarrolla tu aplicación', 'CakePHP: the rapid dev...', and 'localhost/tesis/apoderado/add'.
- Address Bar:** 'localhost/tesis/apoderado/add'.
- Page Header:** 'Apoderado' on the left, 'Documentation' and 'API' on the right, and a logo on the far right.
- Form Title:** 'Add Apoderado'.
- Form Fields:** Seven vertical input fields, each with a label and an asterisk indicating it is required:
 - Apellidos *
 - Nombres *
 - Direccion *
 - Telefono *
 - Email *
 - Vinculo *
 - Condicion *
- Actions:** A button labeled 'List Apoderado' is located below the form.
- Footer:** 'localhost/tesis/apoderado' is visible at the bottom of the page.

Figura 83: Pantalla Registrar Apoderado
Fuente: Elaboración Propia

4.3.4.4. Listar apoderados

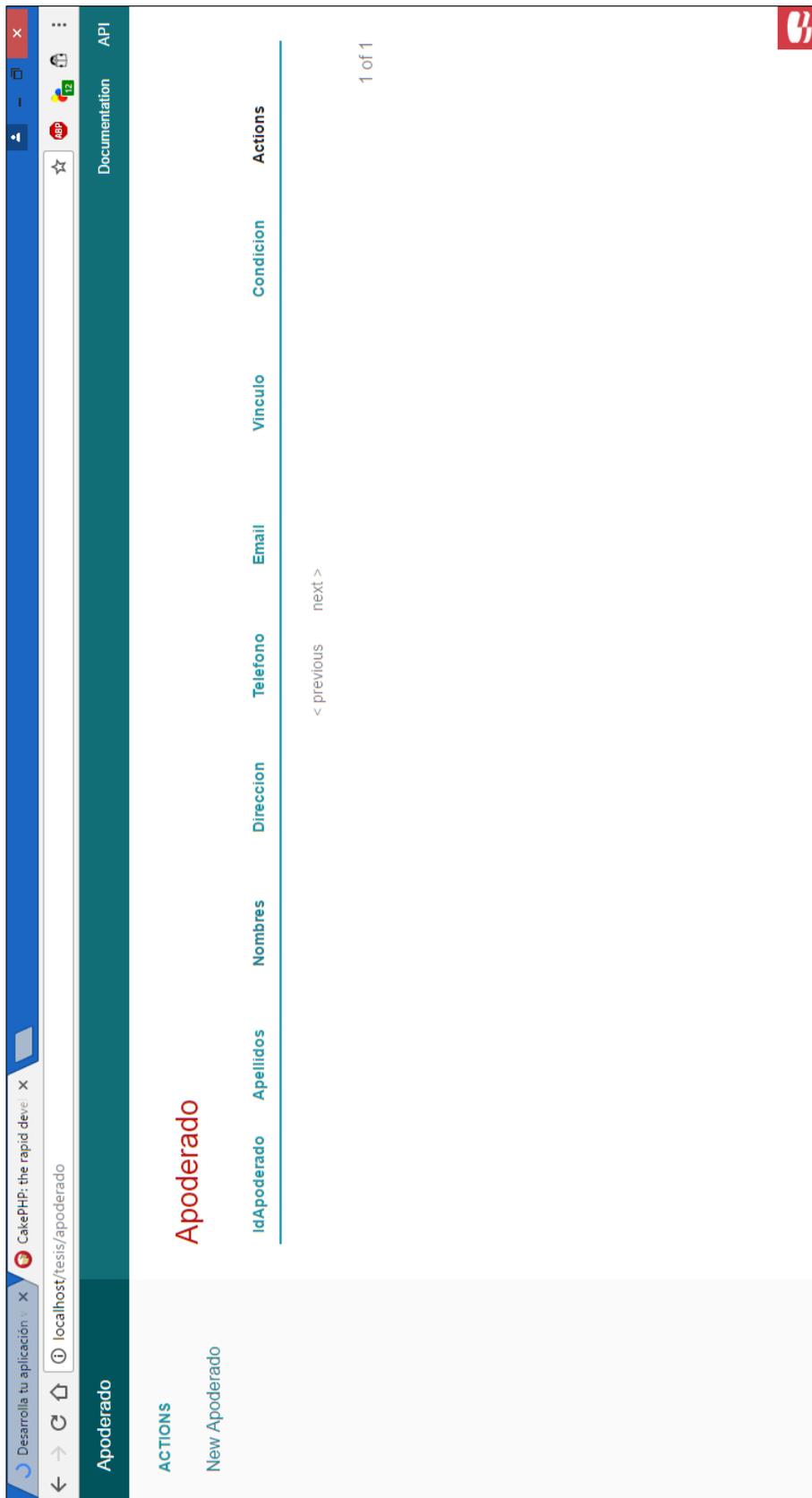


Figura 84: Pantalla Listar Apoderado
Fuente: Elaboración Propia

4.3.4.5.Registrar Solicitud de Matricula

The screenshot displays a web browser window with the following elements:

- Browser Tabs:** 'Desarrolla tu aplicación', 'CakePHP: the rapid dev...', and 'localhost/tesis/solicitud/add'.
- Address Bar:** 'localhost/tesis/solicitud/add'.
- Page Header:** 'Solicitud' and 'Documentation API'.
- Navigation:** 'ACTIONS' and 'List Solicitud'.
- Form Fields:**
 - 'Id Postulante *': A text input field.
 - 'Id Apoderado Principal *': A text input field.
 - 'Id Apoderado Secundario *': A text input field.
 - 'Fecha *': A date picker with '2016' selected for the year and 'September' for the month.
 - 'Estado *': A text input field.
- Submit Button:** A blue button labeled 'SUBMIT'.
- Logo:** A red logo with a white 'B' is located in the top right corner.

Figura 85: Pantalla Registrar Solicitud de Matricula
Fuente: Elaboración Propia

4.3.4.6. Listar solicitudes de matricula



Figura 86: Listar Solicitudes de Matricula
Fuente: Elaboración Propia

4.3.4.7.Registrar Alumno

Desarrolla tu aplicación x CakePHP: the rapid developer x

localhost/tesis/alumno/add

Alumno

ACTIONS

List Alumno

Add Alumno

Apellidos *

Nombres *

Fecha De Nacimiento *

2016 September 19

Sexo *

Direccion *

Email *

Id Apoderado Principal *

localhost/tesis/alumno

Figura 87: Pantalla Registrar Alumno
Fuente: Elaboración Propia

4.3.4.8. Listar Alumno

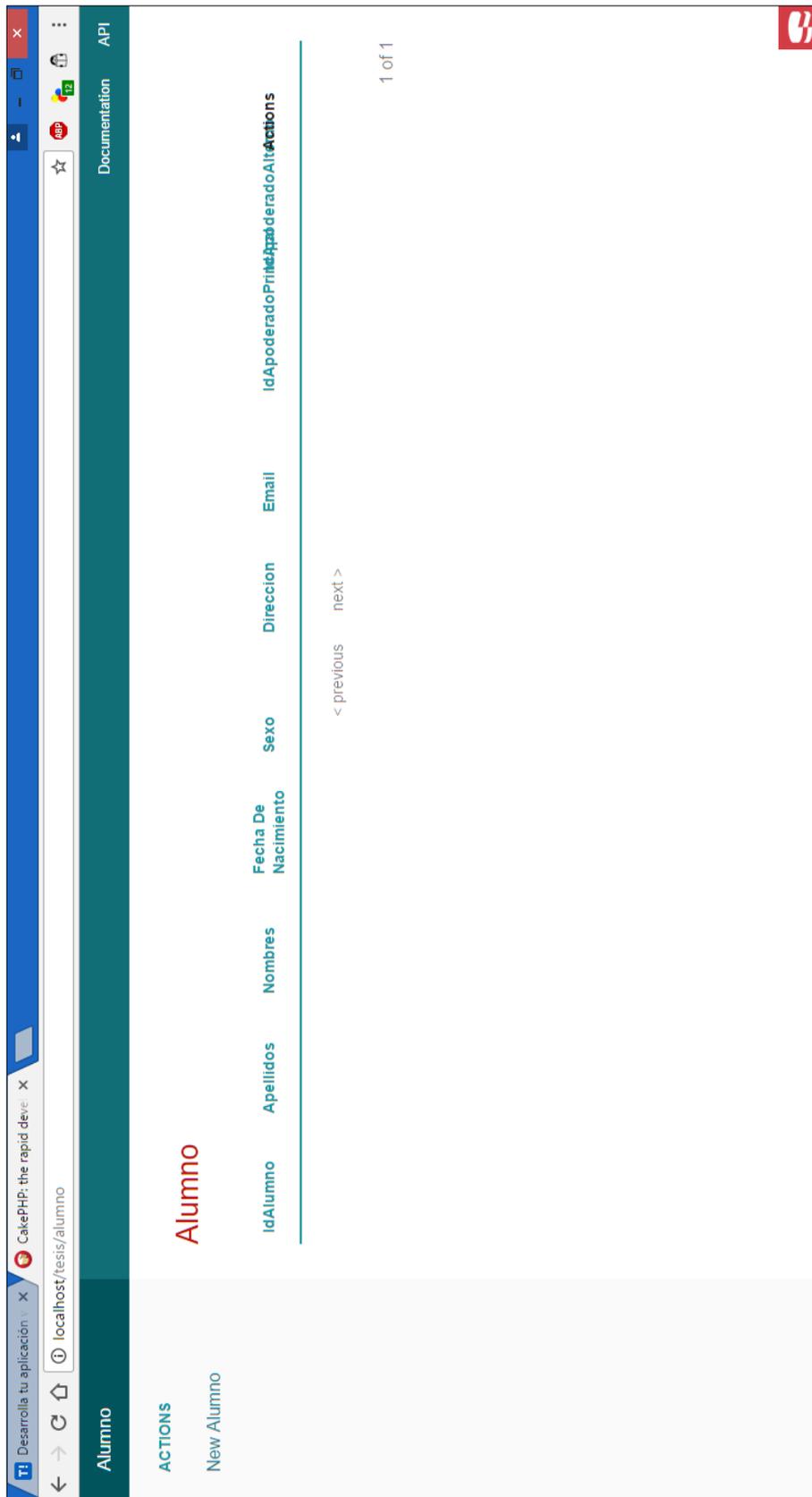


Figura 88: Listar Alumno
Fuente: Elaboración Propia

4.3.4.9.Registrar Matricula

Development environment: T: Desarrolla tu aplicación x CakePHP: the rapid dev... x

Browser address bar: localhost/tesis/matricula/add

Page Header: Matricula | Documentation | API

Form Title: Add Matricula

Form Elements:

- Link: List Matricula
- Field: Id Alumno * (text input)
- Field: Id Grado * (text input)
- Field: Fecha (date input)
- Button: SUBMIT

Figura 89: Pantalla Registrar Matricula

Fuente: Elaboración Propia

4.3.4.10. Registrar Aula

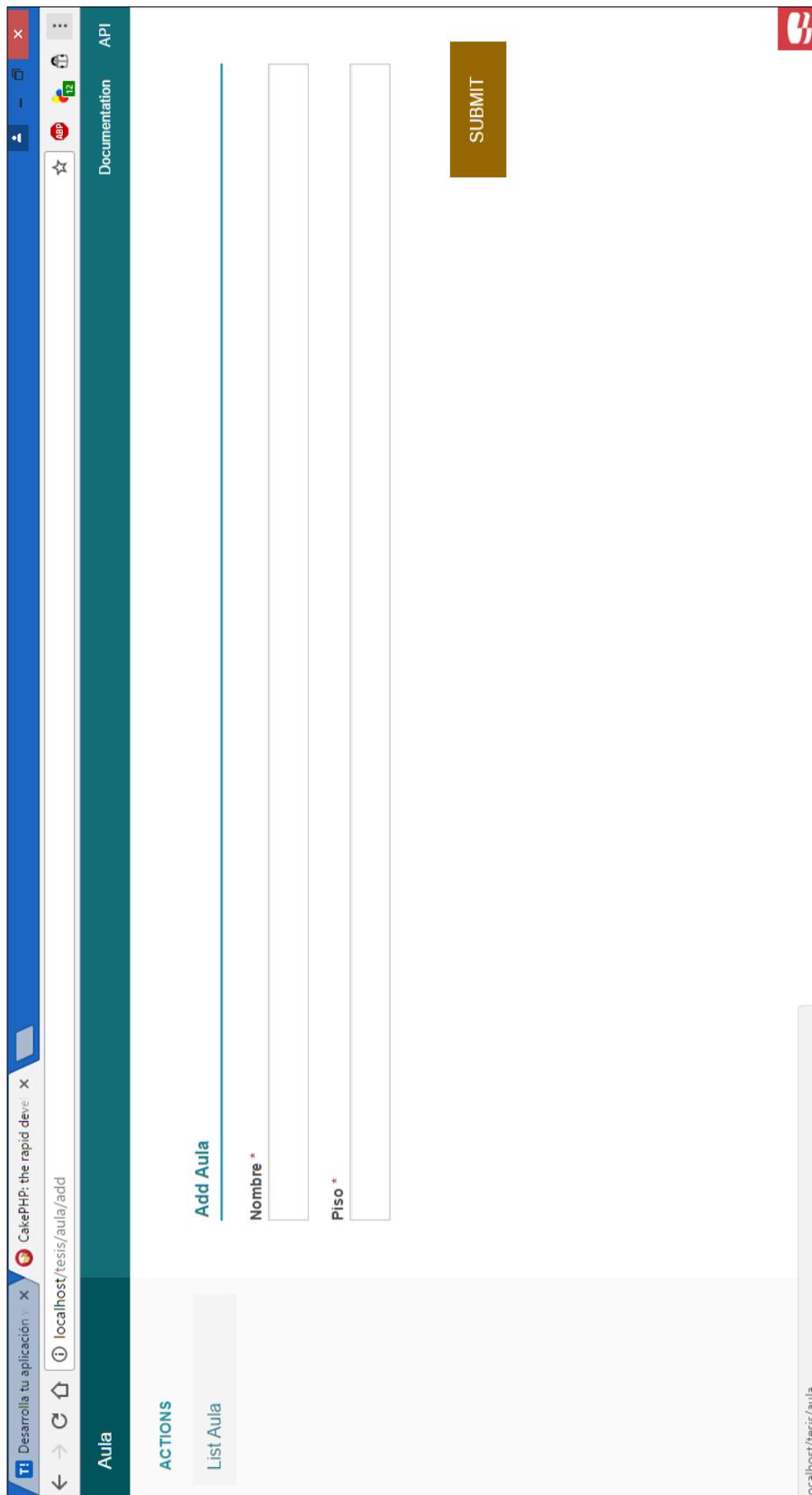


Figura 90: Pantalla Registrar Aula
Fuente: Elaboración Propia

4.3.4.11. Registrar Curso

The screenshot shows a web browser window with the following elements:

- Browser Address Bar:** Displays 'localhost/tesis/curso/add'.
- Page Header:** A dark green bar with 'Curso' on the left and 'API' on the right.
- Form Content:**
 - Section Header:** 'Add Curso' in blue text.
 - Form Fields:**
 - A text input field labeled 'Nombre *' with a red asterisk indicating it is required.
 - A text input field labeled 'Numero Horas'.
 - Submit Button:** A blue button labeled 'SUBMIT'.
- Footer:** A light gray bar with 'ACTIONS' and a link 'List Curso'.

Figura 91: Pantalla Registrar Curso
Fuente: Elaboración Propia

4.3.4.12. Registrar Docente

The screenshot shows a web browser window with the URL `localhost/tesis/docente/add`. The page has a dark green header with the text "Docente" and "ACTIONS". Below the header, there is a "List Docente" link. The main content area is titled "Add Docente" and contains a form with the following fields:

- Apellidos *
- Nombres *
- Sexo *
- Fecha De Nacimiento * (with a date picker showing 2016, September, and 10)
- Direccion *
- Profesion *

A "SUBMIT" button is located on the right side of the form.

Figura 92: Registrar Docente

Fuente: Elaboración Propia

4.3.4.13. Registrar Grado

The screenshot displays a web browser window with the following elements:

- Browser Tab:** CakePHP: the rapid developer
- Address Bar:** localhost/tesis/grado/add
- Page Header:** Grado (left), Documentation API (right)
- Actions:** List Grado
- Form Fields:**
 - Numero Grado *
 - Id Nivel *
 - Id Docente *
- Submit Button:** SUBMIT
- Footer:** localhost/tesis/grado

Figura 93: Pantalla Registrar Grado
Fuente: Elaboración Propia

4.3.4.14. Registrar Sección

The screenshot displays a web browser window with the following elements:

- Browser Address Bar:** Shows the URL `localhost/tesis/grado/add`.
- Page Header:** A dark teal bar with the text "Grado" on the left and "ACTIONS" and "List Grado" below it. On the right side of the header, the words "Documentation" and "API" are visible.
- Main Content Area:**
 - A heading "Add Grado" is positioned above the first input field.
 - Four vertical input fields are stacked vertically, labeled "Numero Grado *", "Id Nivel *", and "Id Docente *".
 - A blue "SUBMIT" button is located at the bottom right of the form area.
- Footer:** A small red logo is visible in the bottom right corner of the page.

Figura 94: Pantalla Registrar Sección
Fuente: Elaboración Propia

4.3.4.15. Registrar Horario

The screenshot shows a web browser window with the URL `localhost/tesis/horario/add`. The page has a dark teal header with the text "Horario" and "API" on the right. Below the header, there is a section titled "ACTIONS" with a button labeled "List Horario". The main content area is titled "Add Horario" and contains a form with the following fields:

- Id Curso ***: A text input field.
- Dia ***: A text input field.
- Hora ***: A text input field.
- Id Aula ***: A text input field.
- Id Docente ***: A text input field.
- Id Grado ***: A text input field.

At the bottom right of the form area, there is a yellow button labeled "SUBMIT". The browser's address bar shows the URL `localhost/tesis/horario`.

Figura 95: Pantalla Registrar Horario
Fuente: Elaboración Propia

4.3.4.16. Registrar Nivel

The screenshot displays a web browser window with the following elements:

- Browser Address Bar:** Shows the URL `localhost/tesis/nivel/add`.
- Page Header:** A dark teal bar with the text "Nivel" on the left and "API" on the right.
- Left Sidebar:** Contains the word "ACTIONS" and a link labeled "List Nivel".
- Main Content Area:**
 - Title: "Add Nivel"
 - Form Field: A single-line text input labeled "Nivel *".
 - Submit Button: A dark blue button labeled "SUBMIT" positioned to the right of the form field.
- Page Footer:** A small red logo is visible in the bottom right corner.

Figura 96: Pantalla Registrar Nivel
Fuente: Elaboración Propia

4.3.1.17.Registrar Clase

CakePHP: the rapid developer framework

localhost/tesis/clase/add

Documentation API

Clase

ACTIONS

List Class

Add Class

Titulo *

Fecha *

2016 September 19

Hora *

5 30

id Curso *

SUBMIT

Figura 97: Pantalla Registrar Clase

Fuente: Elaboración Propia

4.3.1.18.Registrar asistencia Alumno

The screenshot displays a web browser window with the following elements:

- Browser Address Bar:** Shows the URL `localhost/tesis/controlasistenciaalumno/add`.
- Page Header:** Includes the text "Controlasistenciaalumno" and "API".
- Form Title:** "Add Controlasistenciaalumno".
- Form Fields:**
 - Tipo *:** An empty text input field.
 - Id Alumno *:** An empty text input field.
 - Id Clase *:** An empty text input field.
- Submit Button:** A blue button labeled "SUBMIT" positioned to the right of the input fields.
- Actions Panel:** Located at the bottom left, it contains the text "ACTIONS" and "List Controlasistenciaalumno".

Figura 98: Pantalla Registrar Asistencia Alumno
Fuente: Elaboración Propia

4.3.1.19.Registrar Plan de Estudios

The screenshot displays a web browser window with the following elements:

- Browser Address Bar:** Shows the URL `localhost/tesis/plandeestudios/add`.
- Page Title:** `Plandeestudios`.
- Navigation Bar:** Includes a back arrow, a refresh icon, and a home icon.
- Form Title:** `Add Plandeestudio`.
- Form Fields:**
 - `Id Curso *`: A single-line text input field.
 - `Introduccion *`: A multi-line text area.
 - `Cuerpo *`: A multi-line text area.
- Actions:** A button labeled `LIST Plandeestudios` is located on the left side of the form.
- Submit Button:** A large orange button labeled `SUBMIT` is positioned at the bottom right of the form.

Figura 99: Pantalla Registrar Plan de Estudio
Fuente: Elaboración Propia

4.3.1.20.Registrar Requisitos de Egreso

The screenshot displays a web browser window with the following elements:

- Browser Tab:** CakePHP: the rapid deve
- Address Bar:** localhost/tesis/requisito/add
- Page Header:** Requisito (selected), Documentation, API
- Left Sidebar:**
 - ACTIONS
 - List Requisito
- Main Content Area:**
 - Add Requisito** (Section Title)
 - Requisito *** (Large text input field)
 - Id Tipo Requisito *** (Text input field)
 - SUBMIT** (Yellow button)
- Footer:** localhost/tesis/requisito

Figura 100: Pantalla Registrar Requisitos de egreso

Fuente: Elaboración Propia

4.3.1.21.Registrar Evaluaciones de Notas

The screenshot shows a web browser window with the address bar containing 'localhost/tesis/evaluacion/add'. The browser's developer tools are open at the top. The page content includes a dark green header with 'Evaluacion' and 'API' tabs. Below the header, there is a section titled 'Evaluacion' with a sub-section 'ACTIONS' containing a link 'List Evaluacion'. The main content area is titled 'Add Evaluacion' and contains four input fields: 'Id Alumno *', 'Id Curso *', 'Tipo *', and 'Nota *'. A 'SUBMIT' button is located at the bottom right of the form area.

Figura 101: Registrar Evaluaciones y Notas
Fuente: Elaboración Propia

4.3.2. Diagrama de componentes

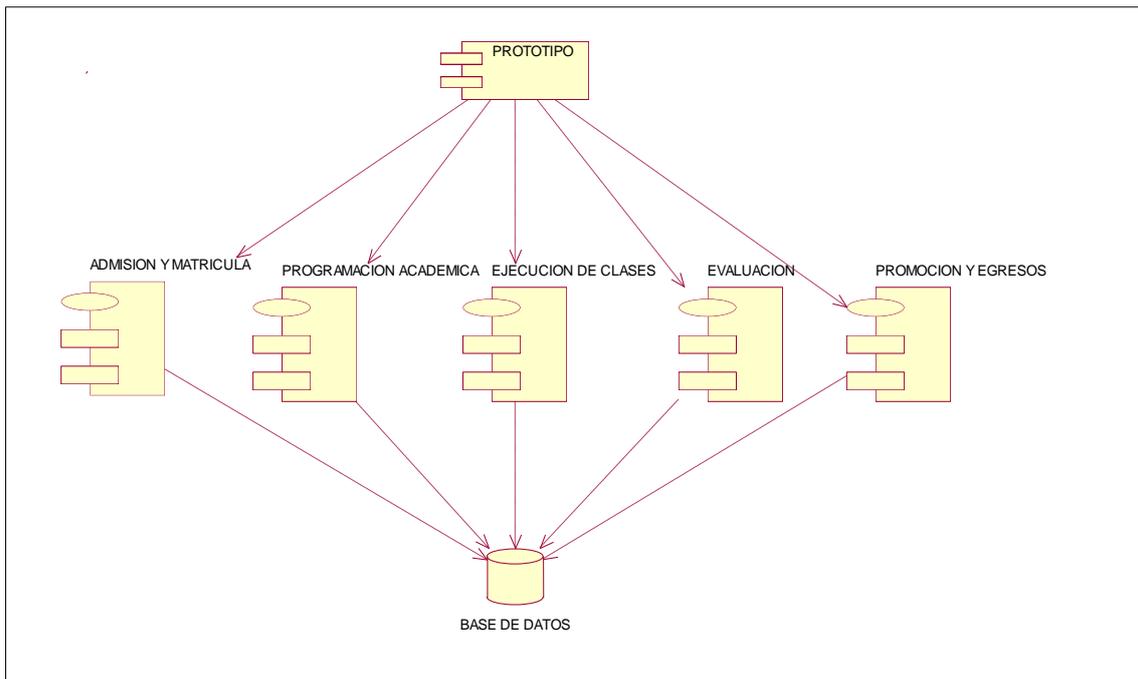


Figura 102: Diagrama de Componentes
Fuente: Elaboración Propia

4.3.3. Diagrama de despliegue

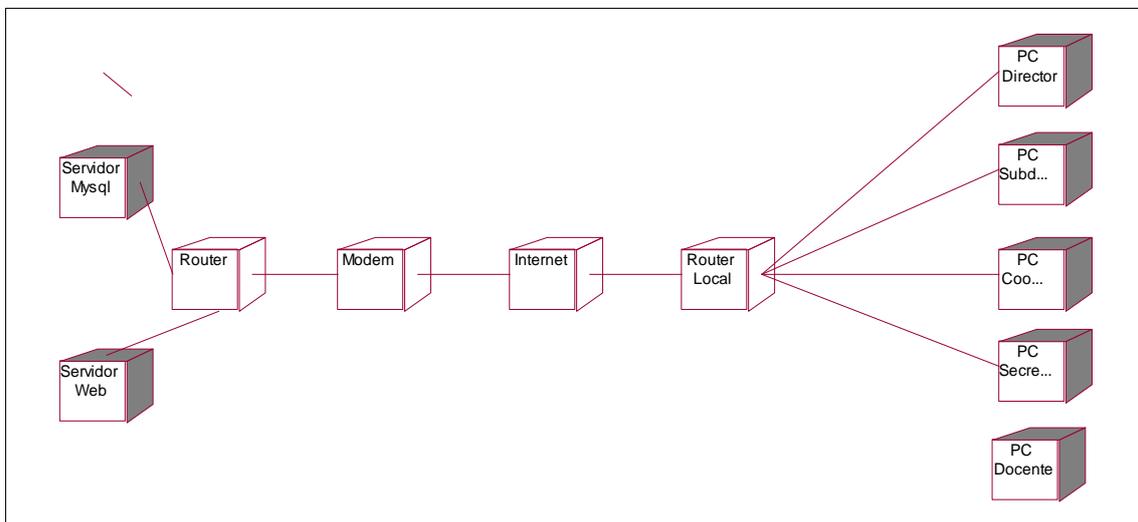


Figura 103: Diagrama de Despliegue
Fuente: Elaboración Propia

4.4. IMPLEMENTACION

4.4.1. Estructuras de Clases con Cake PHP

Para implementar el prototipo, usaremos los siguientes aplicaciones: composer y Xampp. Álvarez (2016) explica que composer “es un gestor de dependencias en proyectos, para programación en PHP. Eso quiere decir que nos permite gestionar (declarar, descargar y mantener actualizados) los paquetes de software en los que se basa nuestro proyecto PHP. Se ha convertido en una herramienta de cabecera para cualquier desarrollador en este lenguaje que aprecie su tiempo y el desarrollo ágil”.

La página PortableApps (2016) nos dice que “Xampp es un paquete integrado de servidor Apache, MySQL, PHP y Perl (AMPP en el XAMPP) que tiene todo pre-configurado y listo para funcionar simplemente descomprimiendo o instalarlo”.

Entonces, primero instalamos el Xampp, el cual es descargada de la siguiente página: <https://www.apachefriends.org/es/index.html> luego se descarga el composer, el cual es descargado de la siguiente pagina: <https://getcomposer.org/>

Luego de ser instalados ambos programas, podemos instalar el framework Cakephp, usando el siguiente comando:

```
composer create-project --prefer-dist cakephp/app [nombre_aplicacion]
```

El Xampp se instala por defecto en la raíz del disco C, y a su vez tiene varias carpetas, entre ellas la carpeta “htdocs”, en la cual se guardan las aplicaciones php, que son ejecutadas, por lo que, mediante línea de comandos, nos dirigimos a dicha carpeta

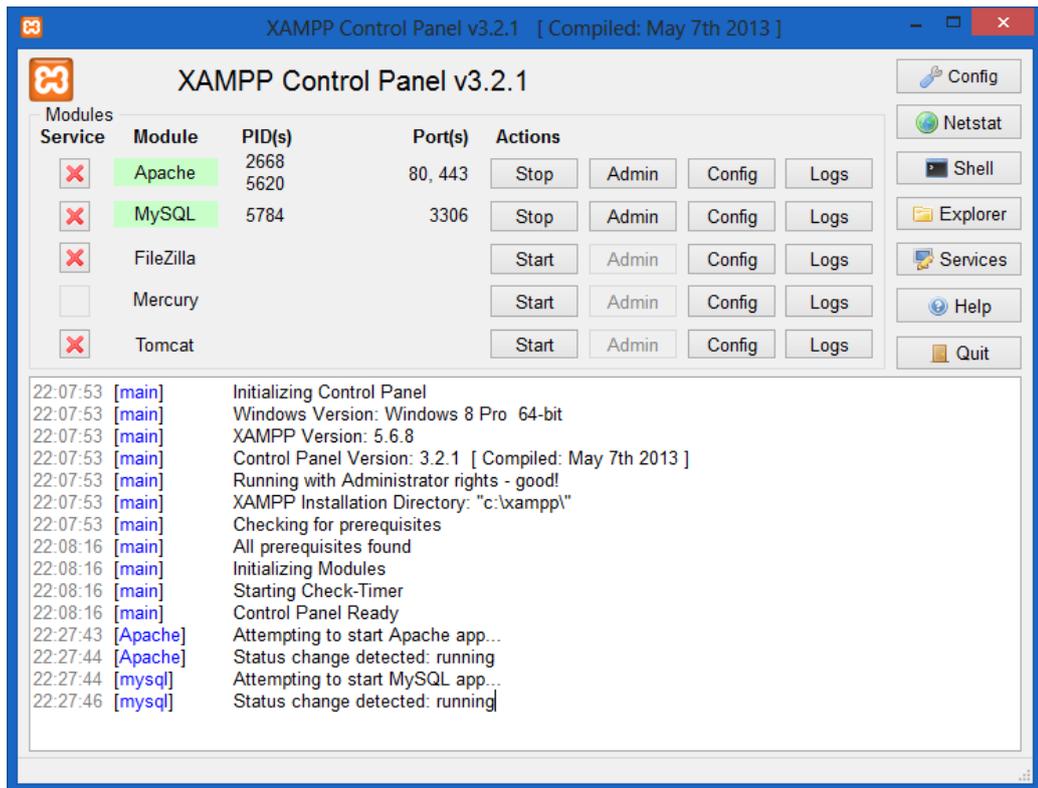


Figura 104: Servidor Xampp

Fuente: Elaboración Propia

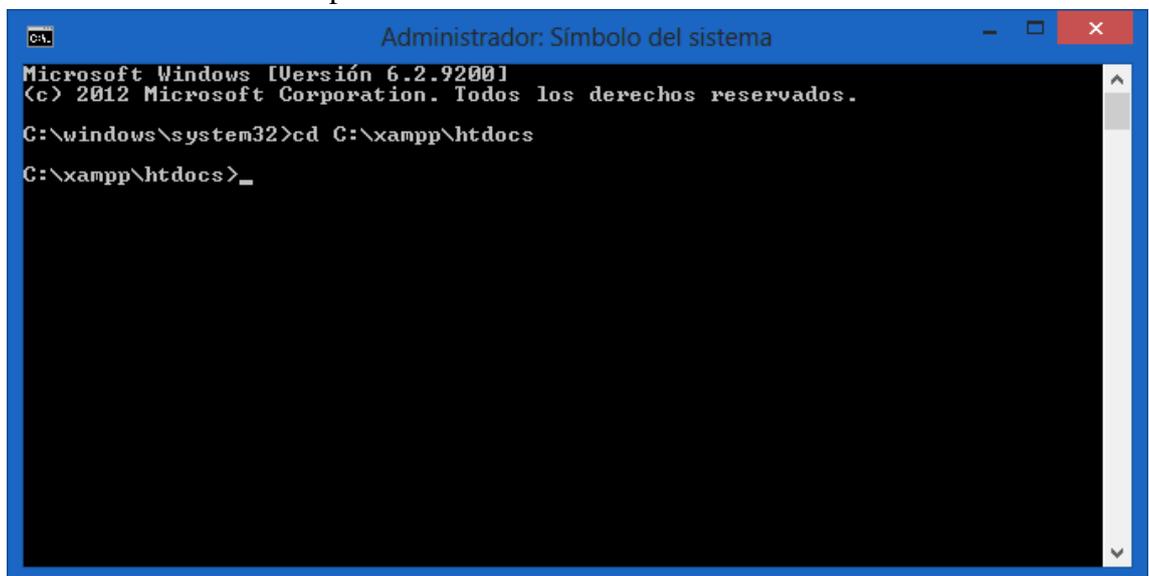
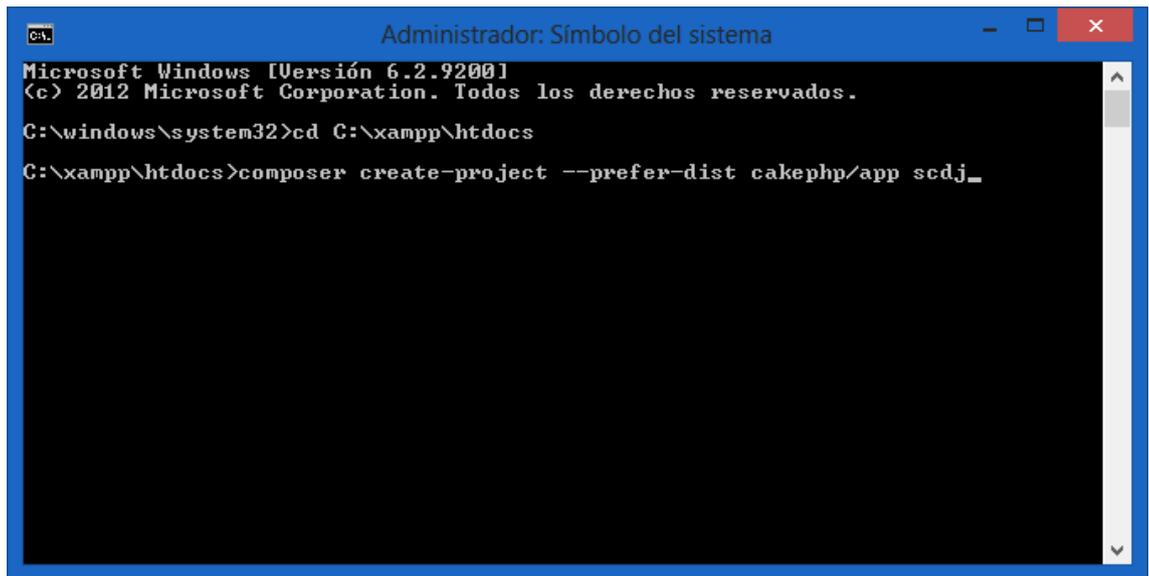


Figura 105: Línea de Comandos de Windows – Dirección a la ruta htdocs

Fuente: Elaboración Propia

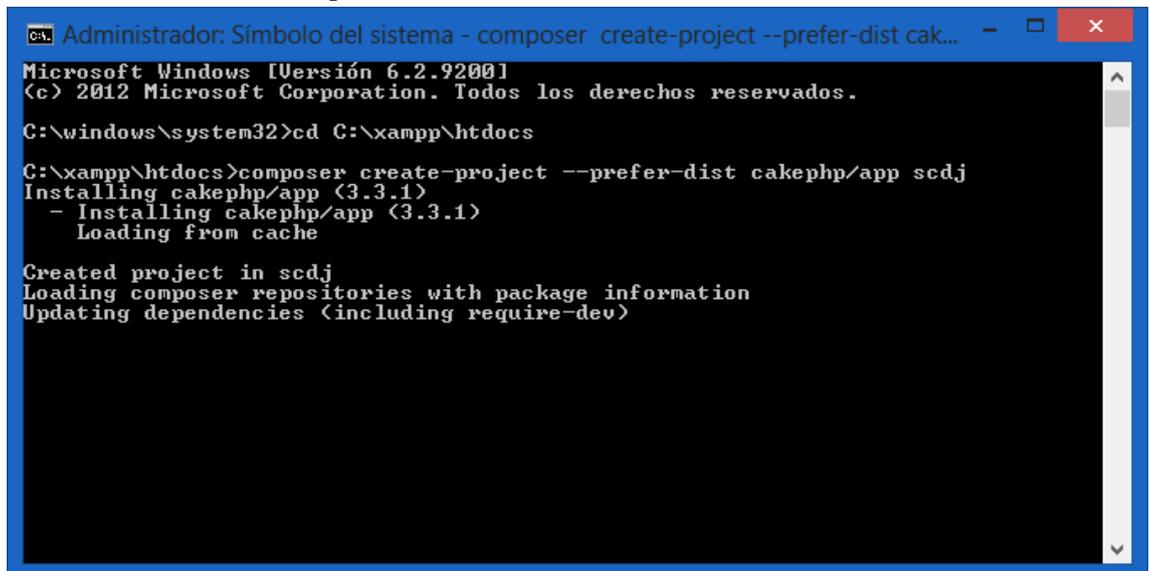
Le ponemos como nombre a la aplicación como “scdj” que son las iniciales del nombre de la institución



```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.
C:\windows\system32>cd C:\xampp\htdocs
C:\xampp\htdocs>composer create-project --prefer-dist cakephp/app scdj_
```

Figura 106: Creación del Proyecto

Fuente: Elaboración Propia



```
Administrador: Símbolo del sistema - composer create-project --prefer-dist cak...
Microsoft Windows [Versión 6.2.9200]
(c) 2012 Microsoft Corporation. Todos los derechos reservados.
C:\windows\system32>cd C:\xampp\htdocs
C:\xampp\htdocs>composer create-project --prefer-dist cakephp/app scdj
Installing cakephp/app (3.3.1)
- Installing cakephp/app (3.3.1)
  Loading from cache

Created project in scdj
Loading composer repositories with package information
Updating dependencies (including require-dev)
```

Figura 107: Instalación del Framework CakePHP

Fuente: Elaboración Propia

```
Administrador: Símbolo del sistema - composer create-project --prefer-dist cak...
Loading from cache
- Installing jakub-ondierka/php-console-color (0.1)
  Loading from cache
- Installing jakub-ondierka/php-console-highlighter (v0.3.2)
  Loading from cache
- Installing dnoegel/php-xdg-base-dir (0.1)
  Loading from cache
- Installing nikic/php-parser (v2.1.1)
  Downloading: 100%
- Installing symfony/var-dumper (v3.1.4)
  Loading from cache
- Installing psy/psysh (v0.7.2)
  Loading from cache
- Installing jdorn/sql-formatter (v1.2.17)
  Loading from cache
- Installing cakephp/debug_kit (3.3.3)
  Downloading: 50%
```

Figura 108: Instalación de Paquetes del Framework CakePHP

Fuente: Elaboración Propia

```
Administrador: Símbolo del sistema - composer create-project --prefer-dist cak...
plugin which provides file concatenation and a flexible filter system for preprocessing and minification.)
cakephp/app suggests installing phpunit/phpunit (Allows automated tests to be run without system-wide install.)
cakephp/app suggests installing cakephp/cakephp-codesniffer (Allows to check the code against the coding standards used in CakePHP.)
symfony/console suggests installing symfony/event-dispatcher (
symfony/console suggests installing symfony/process (
symfony/var-dumper suggests installing ext-symfony_debug (
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing ext-readline (Enables support for arrow-key history navigation, and showing and manipulating command history.)
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
cakephp/debug_kit suggests installing ext-sqlite (DebugKit needs to store panel data in a database. SQLite is simple and easy to use.)
Writing lock file
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
> App\Console\Installer::postInstall
Created `config/app.php` file
Set Folder Permissions ? (Default to Y) [Y,n]? y_
```

Figura 109: Otorgación de Permisos Especiales al Proyecto

Fuente: Elaboración Propia

```
Administrador: Símbolo del sistema

n without system-wide install.)
cakephp/app suggests installing cakephp/cakephp-codesniffer <Allows to check the
code against the coding standards used in CakePHP.>
symfony/console suggests installing symfony/event-dispatcher <
symfony/console suggests installing symfony/process <
symfony/var-dumper suggests installing ext-symfony_debug <
psy/psysh suggests installing ext-pcntl <Enabling the PCNTL extension makes PsyS
H a lot happier :>>
psy/psysh suggests installing ext-posix <If you have PCNTL, you'll want the POSI
X extension as well.>
psy/psysh suggests installing ext-readline <Enables support for arrow-key histor
y navigation, and showing and manipulating command history.>
psy/psysh suggests installing ext-pdo-sqlite <The doc command requires SQLite to
work.>
cakephp/debug_kit suggests installing ext-sqlite <DebugKit needs to store panel
data in a database. SQLite is simple and easy to use.>
Writing lock file
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
> App\Console\Installer::postInstall
Created 'config/app.php' file
Set Folder Permissions ? <Default to Y> [Y,n]? y
Updated Security.salt value in config/app.php

C:\xampp\htdocs>_
```

Figura 110: Otorgación de Permisos Especiales al Proyecto
Fuente: Elaboración Propia

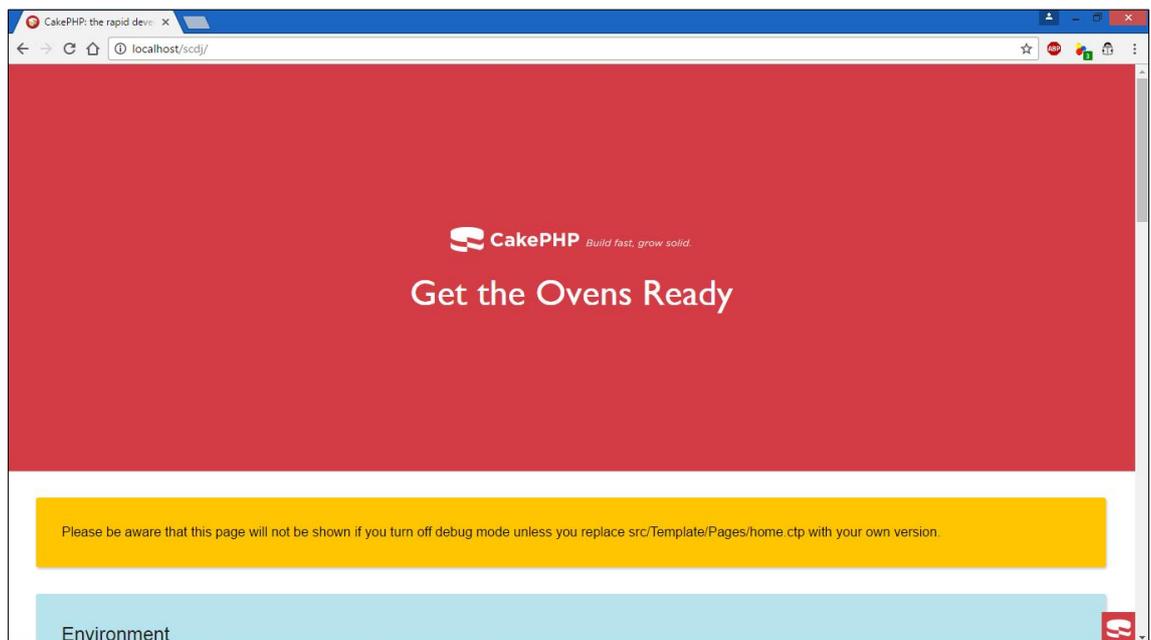


Figura 111: Pantalla Inicial del Proyecto
Fuente: Elaboración Propia

El proyecto debe estar conectado a la base de datos que creamos, puesto que en un inicio aparece desconectado, como podemos apreciar en la figura 110:

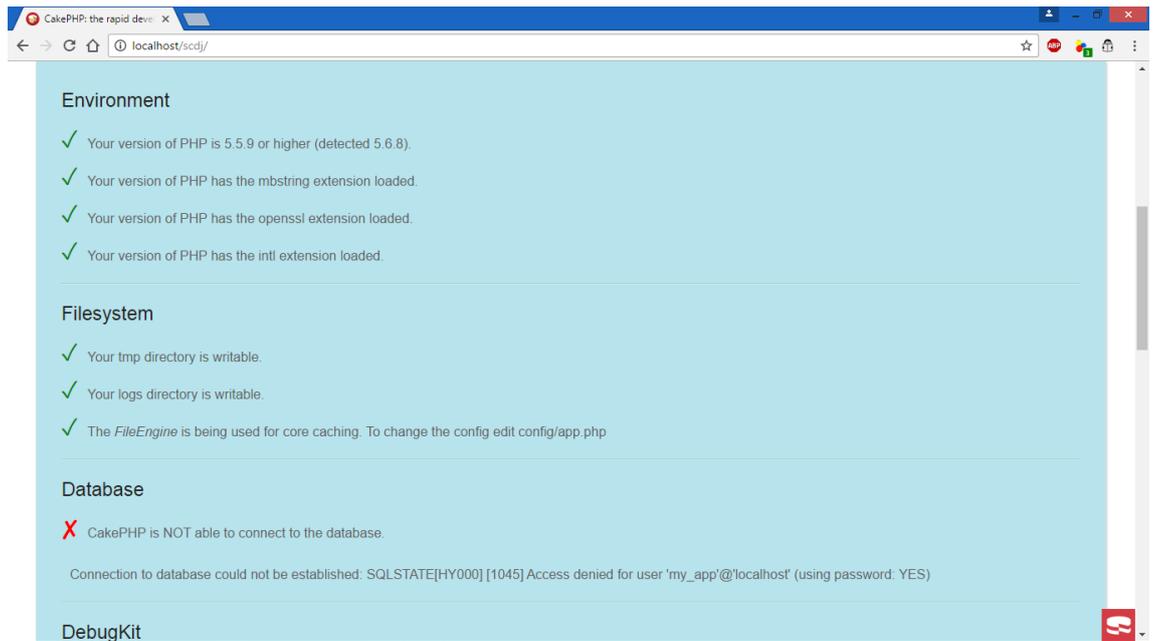


Figura 112: Proyecto No Conectado a la Base de Datos
Fuente: Elaboración Propia

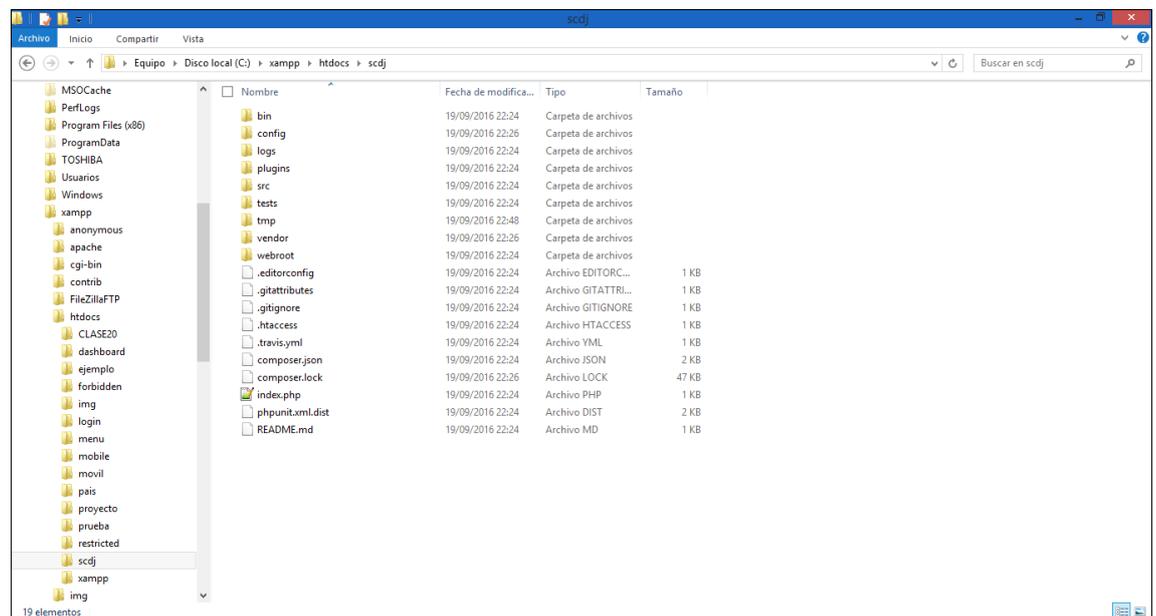


Figura 113: Proyecto SCDJ y sus subcarpetas
Fuente: Elaboración Propia

Para modificar el acceso a la base de datos, debemos dirigirnos a la carpeta del proyecto llamada config, y editar el archivo app.php, tal como vemos en la figura 112, y luego se modifica la sección datasources en dicho archivo, como vemos en la figura 113:

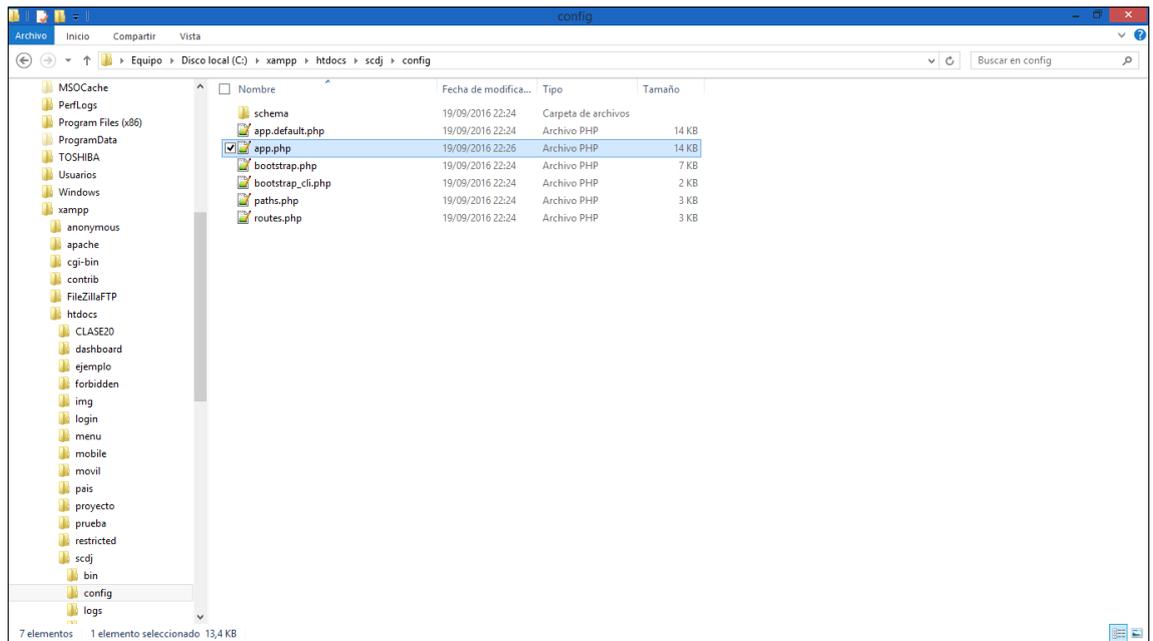


Figura 114: Ubicación del archivo app.php
Fuente: Elaboración Propia

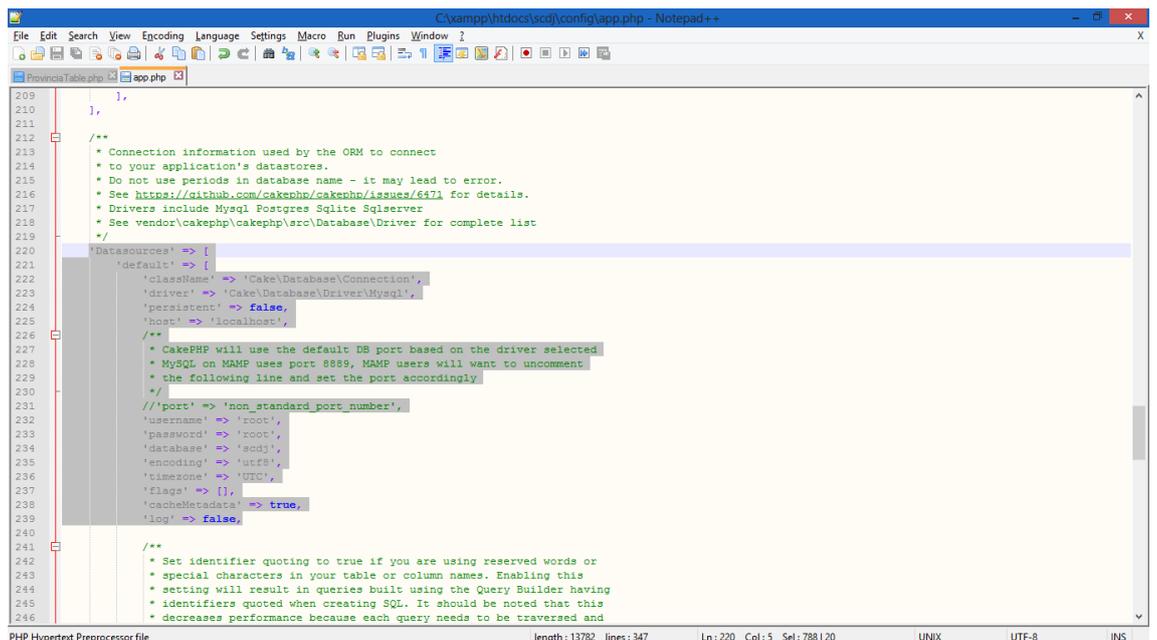


Figura 115: Sección Fuente de datos
Fuente: Elaboración Propia

Luego de editar dicho archivo con la conexión correcta, la página inicial cargara correctamente como vemos en la figura 114:

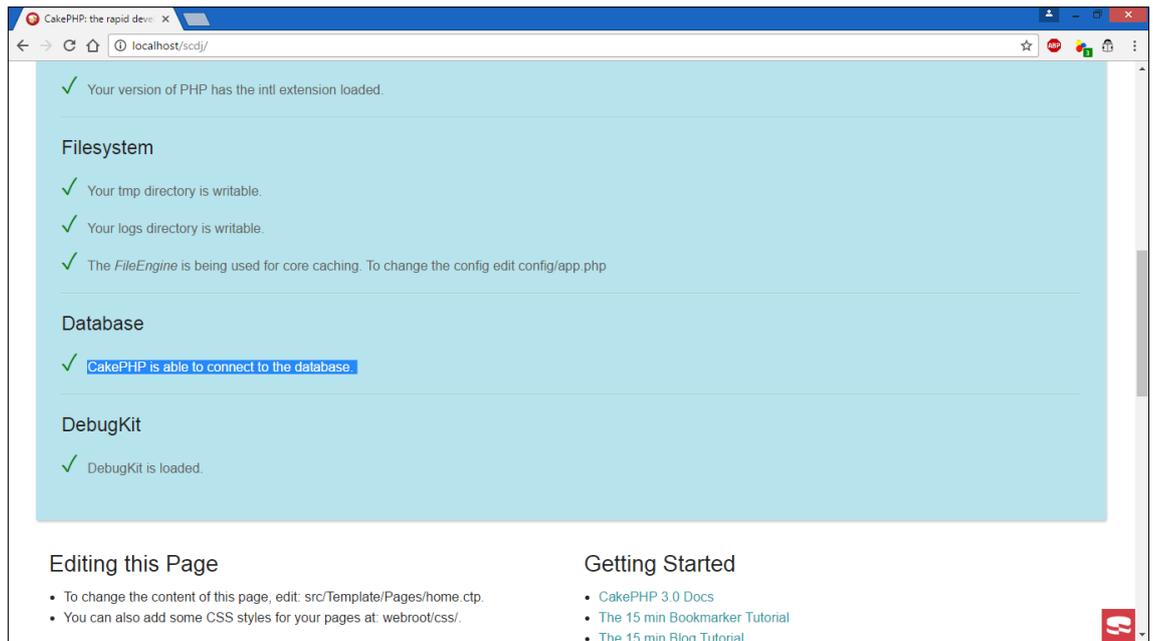


Figura 116: Mensaje de conexión correcta a la base de datos
Fuente: Elaboración Propia

En la figura 114 podemos ver las carpetas que contiene los archivos del proyecto, como los modelos, controladores y pantallas:

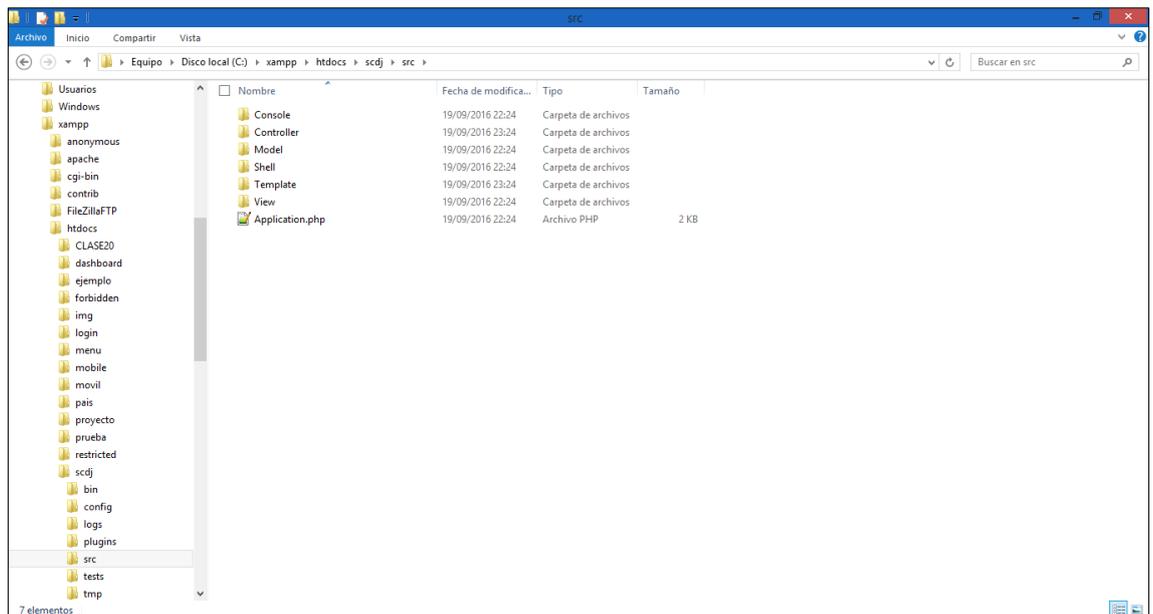


Figura 117: Carpeta src con los archivos del proyecto
Fuente: Elaboración Propia

En la figura 118 se aprecian los controladores

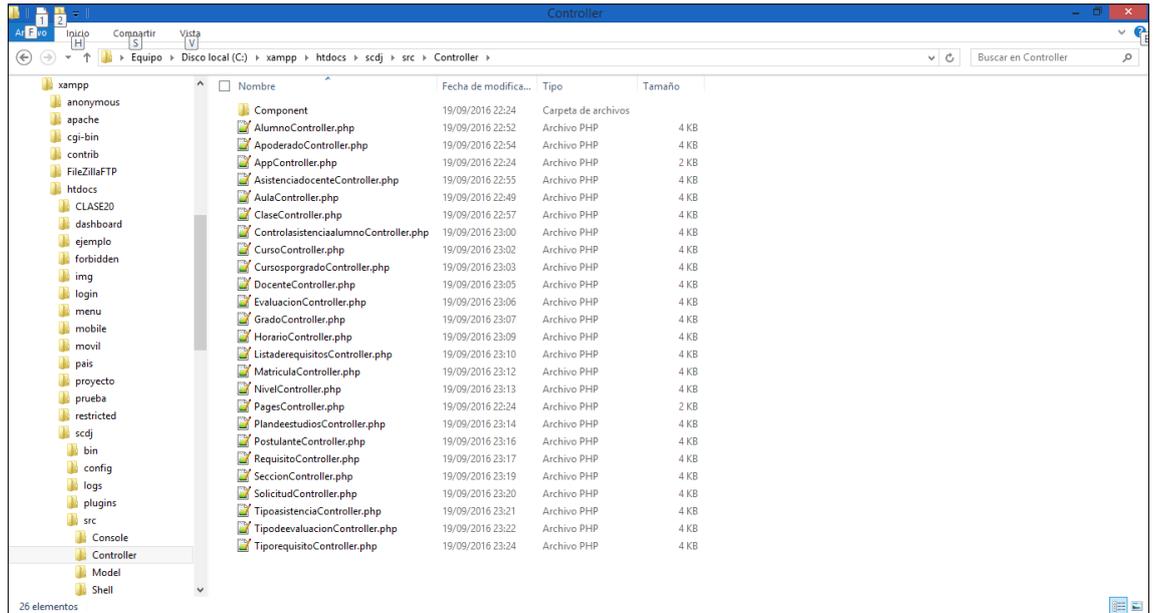


Figura 118: Controladores del Proyecto

Fuente: Elaboración Propia

En la figura 119 están los modelos del proyecto

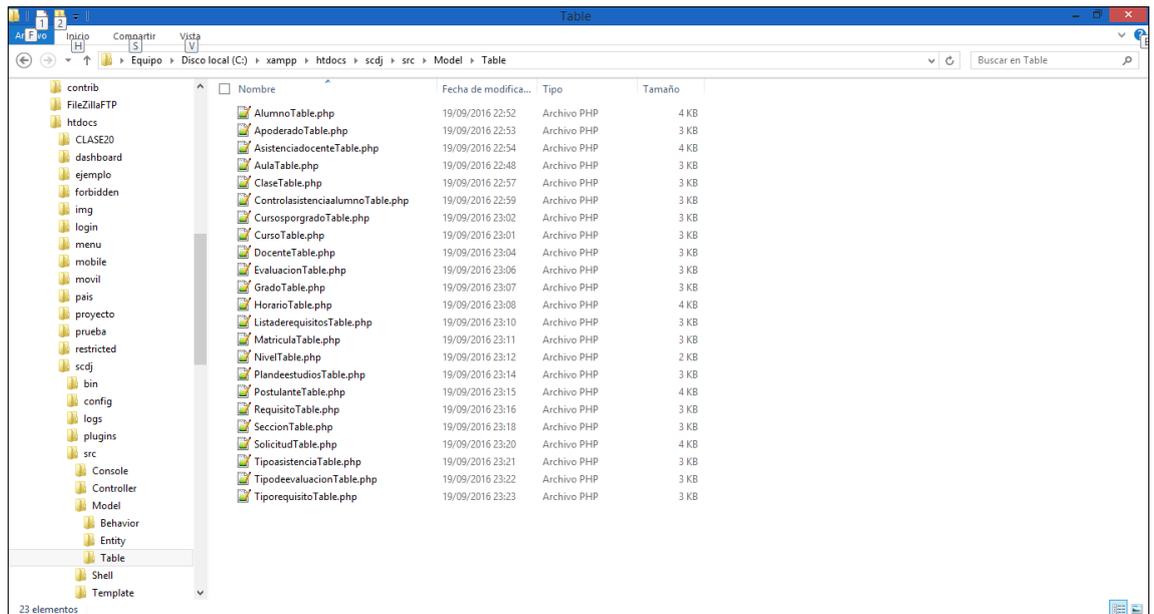


Figura 119: Modelos del Proyecto

Fuente: Elaboración Propia

En la figura 120 están las pantallas del proyecto

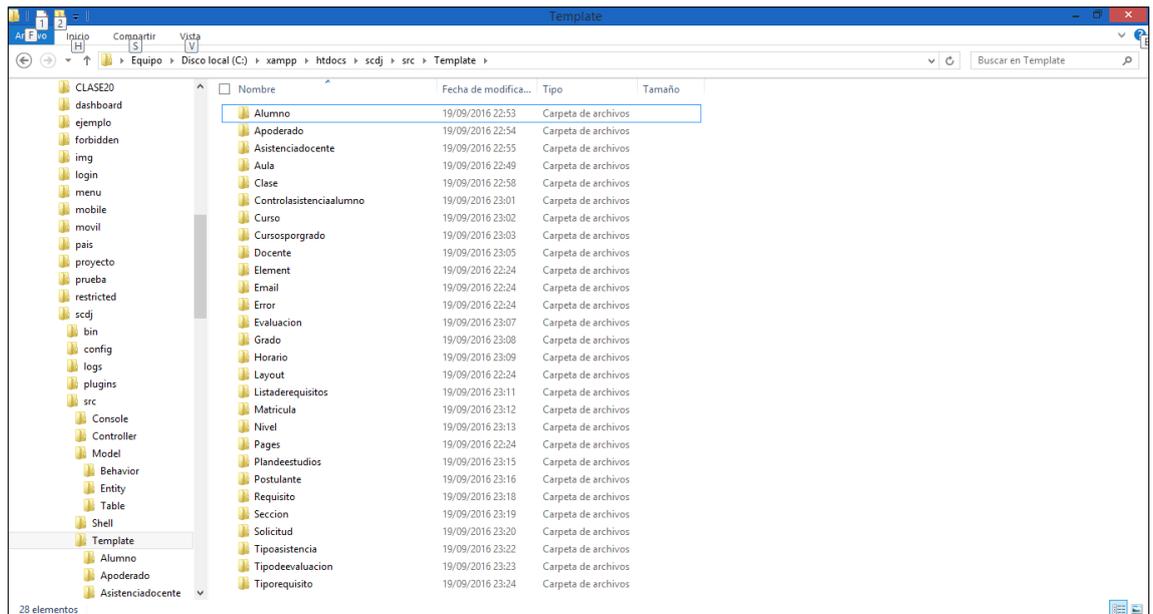


Figura 120: Pantallas del Proyecto

Fuente: Elaboración Propia

En la figura 121 vemos los archivos para hacer las operaciones con la base de datos

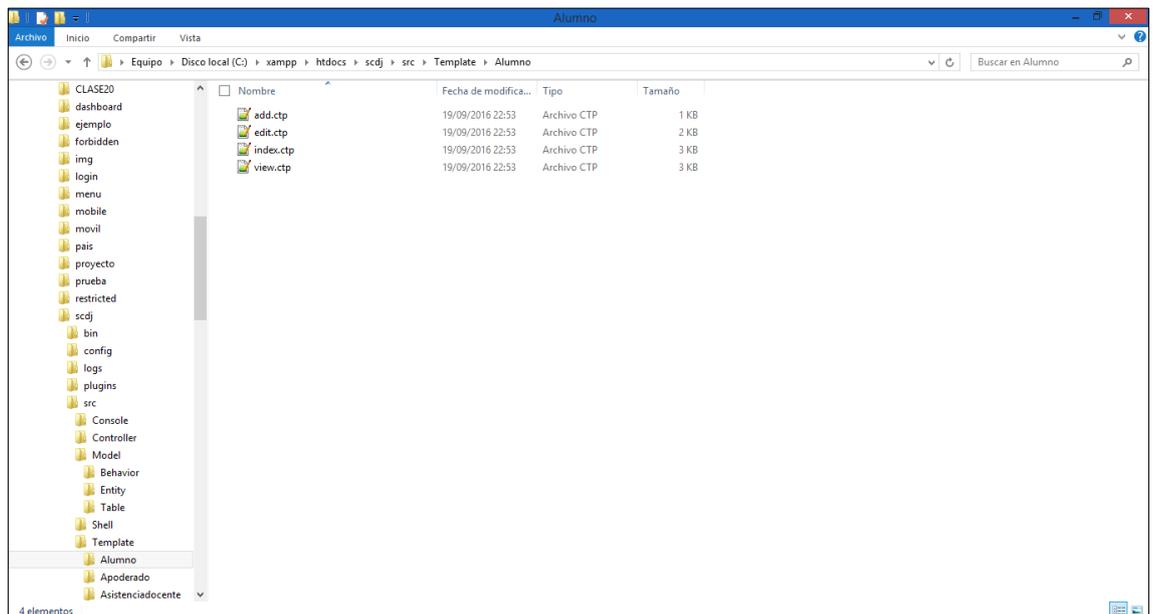


Figura 121: Archivos de operaciones con la base de datos del proyecto

Fuente: Elaboración Propia

En la figura 122 nuevamente usamos la línea de comandos de Windows para dirigirnos a la carpeta bin del proyecto, y ejecutamos el comando “cake bake” para inicializar los comandos de creacion

```
C:\xampp\htdocs>cd scdj
C:\xampp\htdocs\scdj>cd bin
C:\xampp\htdocs\scdj\bin>cake bake
Welcome to CakePHP v3.3.3 Console
-----
App : src
Path: C:\xampp\htdocs\scdj\src\
PHP : 5.6.8
-----
The following commands can be used to generate skeleton code for your application.

Available bake commands:
- all
- behavior
- cell
- component
- controller
- fixture
- form
- helper
- mailer
- migration
- migration_diff
- migration_snapshot
- model
- plugin
- seed
- shell
- shell_helper
- task
- template
- test

By using `cake bake [name]` you can invoke a specific bake task.

C:\xampp\htdocs\scdj\bin>
```

Figura 122: Ejecución del comando “cake bake”

Fuente: Elaboración Propia

La creación de los modelos, controladores y pantallas se crean de la siguiente manera:

Primero, creamos los modelos, mediante el comando `cake bake [nombre de modelo]`, como podemos ver en la figura 123 que hicimos con la entidad aula:

```
Administrador: Símbolo del sistema

C:\xampp\htdocs\scdj\bin>cake bake model aula

Welcome to CakePHP v3.3.3 Console
-----
App : src
Path: C:\xampp\htdocs\scdj\src\
PHP : 5.6.8
-----

One moment while associations are detected.

Baking table class for Aula...

Creating file C:\xampp\htdocs\scdj\src\Model\Table\AulaTable.php
Wrote 'C:\xampp\htdocs\scdj\src\Model\Table\AulaTable.php'
Deleted 'C:\xampp\htdocs\scdj\src\Model\Table\empty'

Baking entity class for Aula...

Creating file C:\xampp\htdocs\scdj\src\Model\Entity\Aula.php
Wrote 'C:\xampp\htdocs\scdj\src\Model\Entity\Aula.php'
Deleted 'C:\xampp\htdocs\scdj\src\Model\Entity\empty'

Baking test fixture for Aula...

Creating file C:\xampp\htdocs\scdj\tests\Fixture\AulaFixture.php
Wrote 'C:\xampp\htdocs\scdj\tests\Fixture\AulaFixture.php'
Deleted 'C:\xampp\htdocs\scdj\tests\Fixture\empty'
Bake is detecting possible fixtures...

Baking test case for App\Model\Table\AulaTable ...

Creating file C:\xampp\htdocs\scdj\tests\TestCase\Model\Table\AulaTableTest.php
Wrote 'C:\xampp\htdocs\scdj\tests\TestCase\Model\Table\AulaTableTest.php'

C:\xampp\htdocs\scdj\bin>
```

Figura 123: Creación del modelo Aula

Fuente: Elaboración propia

Luego se crea el controlador, como podemos ver en la figura 124

```
Administrador: Símbolo del sistema

C:\xampp\htdocs\scdj\bin>cake bake controller aula

Welcome to CakePHP v3.3.3 Console
-----
App : src
Path: C:\xampp\htdocs\scdj\src\
PHP : 5.6.8
-----

Baking controller class for Aula...

Creating file C:\xampp\htdocs\scdj\src\Controller\AulaController.php
Wrote 'C:\xampp\htdocs\scdj\src\Controller\AulaController.php'
Bake is detecting possible fixtures...

Baking test case for App\Controller\AulaController ...

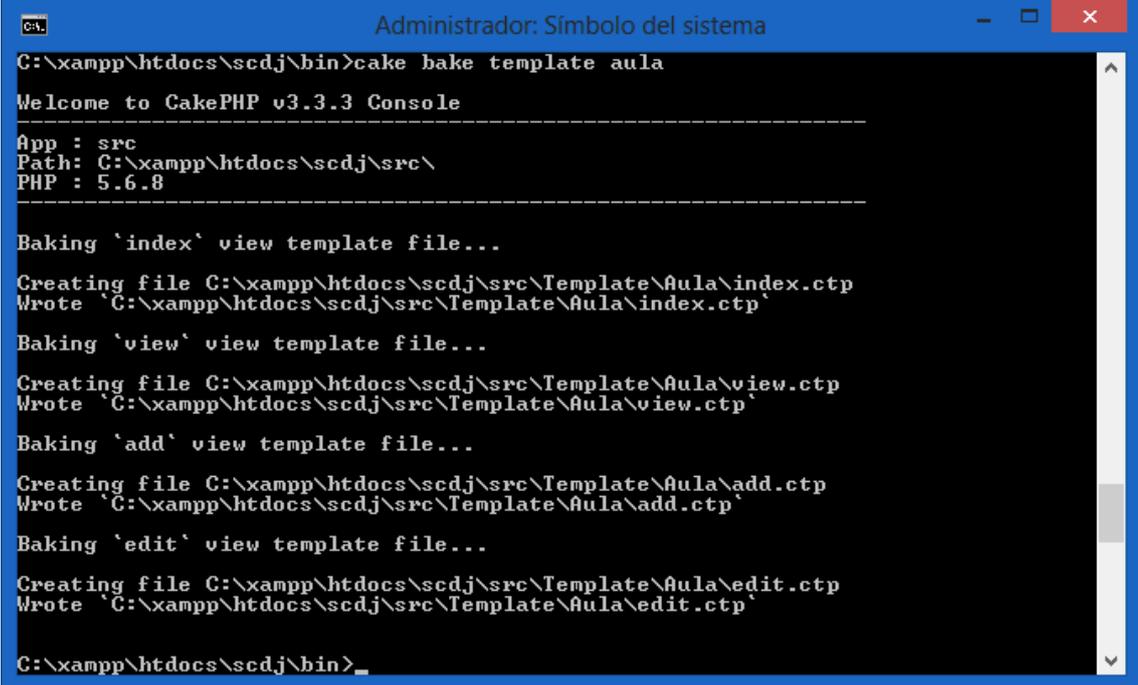
Creating file C:\xampp\htdocs\scdj\tests\TestCase\Controller\AulaControllerTest.php
Wrote 'C:\xampp\htdocs\scdj\tests\TestCase\Controller\AulaControllerTest.php'

C:\xampp\htdocs\scdj\bin>_
```

Figura 124: Creación del controlador Aula

Fuente: Elaboración propia

Finalmente, se crean la pantalla como se puede apreciar en la figura 125



```
C:\xampp\htdocs\scdj\bin>cake bake template aula
Welcome to CakePHP v3.3.3 Console
-----
App : src
Path: C:\xampp\htdocs\scdj\src\
PHP : 5.6.8
-----

Baking `index` view template file...
Creating file C:\xampp\htdocs\scdj\src\Template\Aula\index.ctp
Wrote `C:\xampp\htdocs\scdj\src\Template\Aula\index.ctp`

Baking `view` view template file...
Creating file C:\xampp\htdocs\scdj\src\Template\Aula\view.ctp
Wrote `C:\xampp\htdocs\scdj\src\Template\Aula\view.ctp`

Baking `add` view template file...
Creating file C:\xampp\htdocs\scdj\src\Template\Aula\add.ctp
Wrote `C:\xampp\htdocs\scdj\src\Template\Aula\add.ctp`

Baking `edit` view template file...
Creating file C:\xampp\htdocs\scdj\src\Template\Aula\edit.ctp
Wrote `C:\xampp\htdocs\scdj\src\Template\Aula\edit.ctp`

C:\xampp\htdocs\scdj\bin>_
```

Figura 125: Creación de la pantalla Aula

Fuente: Elaboración propia

Y de esta manera, se crea las plantillas para las demás tablas de la base de datos

4.4.2. Código

Por motivos de ahorro de espacio y practicidad, solo mostraremos el código de la entidad Alumno, puesto que todas las entidades tienen la misma estructura de programación;

AlumnoController.php

```
<?php
namespace App\Controller;

use App\Controller\AppController;

/**
 * Alumno Controller
 *
 * @property \App\Model\Table\AlumnoTable $Alumno
 */
```

```

class AlumnoController extends AppController
{

    /**
     * Index method
     *
     * @return \Cake\Network\Response|null
     */
    public function index()
    {
        $alumno = $this->paginate($this->Alumno);

        $this->set(compact('alumno'));
        $this->set('_serialize', ['alumno']);
    }

    /**
     * View method
     *
     * @param string|null $id Alumno id.
     * @return \Cake\Network\Response|null
     * @throws \Cake\Datasource\Exception\RecordNotFoundException When
record not found.
     */
    public function view($id = null)
    {
        $alumno = $this->Alumno->get($id, [
            'contain' => []
        ]);

        $this->set('alumno', $alumno);
        $this->set('_serialize', ['alumno']);
    }

    /**
     * Add method
     *
     * @return \Cake\Network\Response|void Redirects on successful add,
renders view otherwise.
     */
    public function add()
    {
        $alumno = $this->Alumno->newEntity();
        if ($this->request->is('post')) {

```

```

        $alumno = $this->Alumno->patchEntity($alumno, $this->request->data);
        if ($this->Alumno->save($alumno)) {
            $this->Flash->success(__('The alumno has been saved.));

            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__('The alumno could not be saved. Please, try
again.));
        }
    }
    $this->set(compact('alumno'));
    $this->set('_serialize', ['alumno']);
}

/**
 * Edit method
 *
 * @param string|null $id Alumno id.
 * @return \Cake\Network\Response|void Redirects on successful edit,
renders view otherwise.
 * @throws \Cake\Network\Exception\NotFoundException When record not
found.
 */
public function edit($id = null)
{
    $alumno = $this->Alumno->get($id, [
        'contain' => []
    ]);
    if ($this->request->is(['patch', 'post', 'put'])) {
        $alumno = $this->Alumno->patchEntity($alumno, $this->request->data);
        if ($this->Alumno->save($alumno)) {
            $this->Flash->success(__('The alumno has been saved.));

            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__('The alumno could not be saved. Please, try
again.));
        }
    }
    $this->set(compact('alumno'));
    $this->set('_serialize', ['alumno']);
}

/**

```

```

* Delete method
*
* @param string|null $id Alumno id.
* @return \Cake\Network\Response|null Redirects to index.
* @throws \Cake\Datasource\Exception\RecordNotFoundException When
record not found.
*/
public function delete($id = null)
{
    $this->request->allowMethod(['post', 'delete']);
    $alumno = $this->Alumno->get($id);
    if ($this->Alumno->delete($alumno)) {
        $this->Flash->success(__('The alumno has been deleted.));
    } else {
        $this->Flash->error(__('The alumno could not be deleted. Please, try
again.));
    }

    return $this->redirect(['action' => 'index']);
}
}

```

AlumnoTable.php

```

<?php
namespace App\Model\Table;

use Cake\ORM\Query;
use Cake\ORM\RulesChecker;
use Cake\ORM\Table;
use Cake\Validation\Validator;

/**
 * Alumno Model
 *
 * @method \App\Model\Entity\Alumno get($primaryKey, $options = [])
 * @method \App\Model\Entity\Alumno newEntity($data = null, array $options
= [])
 * @method \App\Model\Entity\Alumno[] newEntities(array $data, array
$options = [])
 * @method \App\Model\Entity\Alumno|bool
save(\Cake\Datasource\EntityInterface $entity, $options = [])

```

```

* @method \App\Model\Entity\Alumno
patchEntity(\Cake\Datasource\EntityInterface $entity, array $data, array
$options = [])
* @method \App\Model\Entity\Alumno[] patchEntities($entities, array $data,
array $options = [])
* @method \App\Model\Entity\Alumno findOrCreate($search, callable
$callback = null)
*/
class AlumnoTable extends Table
{

/**
* Initialize method
*
* @param array $config The configuration for the Table.
* @return void
*/
public function initialize(array $config)
{
    parent::initialize($config);

    $this->table('alumno');
    $this->displayField('idAlumno');
    $this->primaryKey('idAlumno');
}

/**
* Default validation rules.
*
* @param \Cake\Validation\Validator $validator Validator instance.
* @return \Cake\Validation\Validator
*/
public function validationDefault(Validator $validator)
{
    $validator
        ->integer('idAlumno')
        ->allowEmpty('idAlumno', 'create')
        ->add('idAlumno', 'unique', ['rule' => 'validateUnique', 'provider' =>
'table']);

    $validator
        ->requirePresence('apellidos', 'create')
        ->notEmpty('apellidos');
}

```

```
$validator
->requirePresence('nombres', 'create')
->notEmpty('nombres');
```

```
$validator
->date('fecha_de_nacimiento')
->requirePresence('fecha_de_nacimiento', 'create')
->notEmpty('fecha_de_nacimiento');
```

```
$validator
->requirePresence('sexo', 'create')
->notEmpty('sexo');
```

```
$validator
->requirePresence('direccion', 'create')
->notEmpty('direccion');
```

```
$validator
->email('email')
->requirePresence('email', 'create')
->notEmpty('email');
```

```
$validator
->integer('idApoderadoPrincipal')
->requirePresence('idApoderadoPrincipal', 'create')
->notEmpty('idApoderadoPrincipal')
->add('idApoderadoPrincipal', 'unique', ['rule' => 'validateUnique',
'provider' => 'table']);
```

```
$validator
->integer('idApoderadoAlterno')
->requirePresence('idApoderadoAlterno', 'create')
->notEmpty('idApoderadoAlterno')
->add('idApoderadoAlterno', 'unique', ['rule' => 'validateUnique',
'provider' => 'table']);
```

```
return $validator;
}
```

```
/**
```

```
* Returns a rules checker object that will be used for validating
* application integrity.
```

```
*
```

```
* @param \Cake\ORM\RulesChecker $rules The rules object to be modified.
```

```

* @return \Cake\ORM\RulesChecker
*/
public function buildRules(RulesChecker $rules)
{
    $rules->add($rules->isUnique(['email']));
    $rules->add($rules->isUnique(['idApoderadoPrincipal']));
    $rules->add($rules->isUnique(['idApoderadoAlterno']));
    $rules->add($rules->isUnique(['idAlumno']));

    return $rules;
}
}

```

Addalumno.ctp

```

<nav class="large-3 medium-4 columns" id="actions-sidebar">
    <ul class="side-nav">
        <li class="heading"><?= __('Actions') ?></li>
        <li><?= $this->Html->link(__('List Alumno'), ['action' => 'index']) ?></li>
    </ul>
</nav>
<div class="alumno form large-9 medium-8 columns content">
    <?= $this->Form->create($alumno) ?>
    <fieldset>
        <legend><?= __('Add Alumno') ?></legend>
        <?php
            echo $this->Form->input('apellidos');
            echo $this->Form->input('nombres');
            echo $this->Form->input('fecha_de_nacimiento');
            echo $this->Form->input('sexo');
            echo $this->Form->input('direccion');
            echo $this->Form->input('email');
            echo $this->Form->input('idApoderadoPrincipal');
            echo $this->Form->input('idApoderadoAlterno');
        ?>
    </fieldset>
    <?= $this->Form->button(__('Submit')) ?>
    <?= $this->Form->end() ?>
</div>

```

editAlumno.ctp

```

<nav class="large-3 medium-4 columns" id="actions-sidebar">

```

```

<ul class="side-nav">
  <li class="heading"><?= __('Actions') ?></li>
  <li><?= $this->Form->postLink(
    __('Delete'),
    ['action' => 'delete', $alumno->idAlumno],
    ['confirm' => __('Are you sure you want to delete # {0}?', $alumno-
>idAlumno)]
  )
  ?></li>
  <li><?= $this->Html->link(__('List Alumno'), ['action' => 'index']) ?></li>
</ul>
</nav>
<div class="alumno form large-9 medium-8 columns content">
  <?= $this->Form->create($alumno) ?>
  <fieldset>
    <legend><?= __('Edit Alumno') ?></legend>
    <?php
      echo $this->Form->input('apellidos');
      echo $this->Form->input('nombres');
      echo $this->Form->input('fecha_de_nacimiento');
      echo $this->Form->input('sexo');
      echo $this->Form->input('direccion');
      echo $this->Form->input('email');
      echo $this->Form->input('idApoderadoPrincipal');
      echo $this->Form->input('idApoderadoAlterno');
    ?>
  </fieldset>
  <?= $this->Form->button(__('Submit')) ?>
  <?= $this->Form->end() ?>
</div>

```

Index.ctp

```

<nav class="large-3 medium-4 columns" id="actions-sidebar">
  <ul class="side-nav">
    <li class="heading"><?= __('Actions') ?></li>
    <li><?= $this->Html->link(__('New Alumno'), ['action' => 'add']) ?></li>
  </ul>
</nav>
<div class="alumno index large-9 medium-8 columns content">
  <h3><?= __('Alumno') ?></h3>
  <table cellpadding="0" cellspacing="0">
    <thead>
      <tr>

```

```

        <th scope="col"><?= $this->Paginator->sort('idAlumno') ?></th>
        <th scope="col"><?= $this->Paginator->sort('apellidos') ?></th>
        <th scope="col"><?= $this->Paginator->sort('nombres') ?></th>
        <th scope="col"><?= $this->Paginator->sort('fecha_de_nacimiento')
?></th>
        <th scope="col"><?= $this->Paginator->sort('sexo') ?></th>
        <th scope="col"><?= $this->Paginator->sort('direccion') ?></th>
        <th scope="col"><?= $this->Paginator->sort('email') ?></th>
        <th scope="col"><?= $this->Paginator->sort('idApoderadoPrincipal')
?></th>
        <th scope="col"><?= $this->Paginator->sort('idApoderadoAlterno')
?></th>
        <th scope="col" class="actions"><?= __('Actions') ?></th>
    </tr>
</thead>
<tbody>
    <?php foreach ($alumno as $alumno): ?>
    <tr>
        <td><?= $this->Number->format($alumno->idAlumno) ?></td>
        <td><?= h($alumno->apellidos) ?></td>
        <td><?= h($alumno->nombres) ?></td>
        <td><?= h($alumno->fecha_de_nacimiento) ?></td>
        <td><?= h($alumno->sexo) ?></td>
        <td><?= h($alumno->direccion) ?></td>
        <td><?= h($alumno->email) ?></td>
        <td><?= $this->Number->format($alumno->idApoderadoPrincipal)
?></td>
        <td><?= $this->Number->format($alumno->idApoderadoAlterno)
?></td>
        <td class="actions">
            <?= $this->Html->link(__('View'), ['action' => 'view', $alumno-
>idAlumno]) ?>
            <?= $this->Html->link(__('Edit'), ['action' => 'edit', $alumno-
>idAlumno]) ?>
            <?= $this->Form->postLink(__('Delete'), ['action' => 'delete',
$alumno->idAlumno], ['confirm' => __('Are you sure you want to delete # {0}?',
$alumno->idAlumno)]) ?>
        </td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>
<div class="paginator">
    <ul class="pagination">

```

```

        <?=$this->Paginator->prev('<' . __('previous')) ?>
        <?=$this->Paginator->numbers() ?>
        <?=$this->Paginator->next(__('next') . '>') ?>
    </ul>
    <p><?=$this->Paginator->counter() ?></p>
</div>
</div>

```

viewAlumno.ctp

```

<nav class="large-3 medium-4 columns" id="actions-sidebar">
    <ul class="side-nav">
        <li class="heading"><?=$this->Paginator->prev('<' . __('previous')) ?></li>
        <li><?=$this->Html->link(__('Edit Alumno'), ['action' => 'edit', $alumno->idAlumno]) ?> </li>
        <li><?=$this->Form->postLink(__('Delete Alumno'), ['action' => 'delete', $alumno->idAlumno], ['confirm' => __('Are you sure you want to delete # {0}?', $alumno->idAlumno)]) ?> </li>
        <li><?=$this->Html->link(__('List Alumno'), ['action' => 'index']) ?> </li>
        <li><?=$this->Html->link(__('New Alumno'), ['action' => 'add']) ?> </li>
    </ul>
</nav>
<div class="alumno view large-9 medium-8 columns content">
    <h3><?=$this->Paginator->prev('<' . __('previous')) ?></h3>
    <table class="vertical-table">
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
        <tr>
            <th scope="row"><?=$this->Paginator->prev('<' . __('previous')) ?></th>
            <td><?=$this->Paginator->prev('<' . __('previous')) ?></td>
        </tr>
    </table>

```

```

</tr>
<tr>
  <th scope="row"><?= __('IdAlumno') ?></th>
  <td><?= $this->Number->format($alumno->idAlumno) ?></td>
</tr>
<tr>
  <th scope="row"><?= __('IdApoderadoPrincipal') ?></th>
  <td><?= $this->Number->format($alumno->idApoderadoPrincipal)
?></td>
</tr>
<tr>
  <th scope="row"><?= __('IdApoderadoAlterno') ?></th>
  <td><?= $this->Number->format($alumno->idApoderadoAlterno)
?></td>
</tr>
<tr>
  <th scope="row"><?= __('Fecha De Nacimiento') ?></th>
  <td><?= h($alumno->fecha_de_nacimiento) ?></td>
</tr>
</table>
</div>

```

4.4.3. Base de datos(script)

```
-- MySQL Workbench Forward Engineering
```

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

```

```

-----
-- Schema tesis
-----

-----
-- Schema tesis
-----

```

```
CREATE SCHEMA IF NOT EXISTS `tesis` DEFAULT CHARACTER SET
utf8 ;
```

```
-----  
-- Schema tesis  
-----
```

```
-----  
-- Schema tesis  
-----
```

```
CREATE SCHEMA IF NOT EXISTS `tesis` DEFAULT CHARACTER SET
latin1 ;
```

```
USE `tesis` ;
```

```
-----  
-- Table `tesis`.`curso`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tesis`.`curso` (  
  `idCurso` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(30) NOT NULL,  
  `numeroHoras` INT(11) NULL DEFAULT NULL,  
  PRIMARY KEY (`idCurso`),  
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tesis`.`clase`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tesis`.`clase` (  
  `idClase` INT(11) NOT NULL AUTO_INCREMENT,  
  `titulo` VARCHAR(45) NOT NULL,  
  `fecha` DATE NOT NULL,  
  `hora` TIME NOT NULL,  
  `idCurso` INT(11) NOT NULL,  
  PRIMARY KEY (`idClase`),  
  UNIQUE INDEX `idClase_UNIQUE` (`idClase` ASC),  
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC),  
  INDEX `curso_FK_idx` (`idCurso` ASC),  
  CONSTRAINT `clase_curso_FK`  
    FOREIGN KEY (`idCurso`)  
    REFERENCES `tesis`.`curso` (`idCurso`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-----
-- Table `tesis`.`tipoasistencia`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`tipoasistencia` (
  `idtipoasistencia` INT NOT NULL AUTO_INCREMENT,
  `tipoasistencia` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`idtipoasistencia`),
  UNIQUE INDEX `idtipoasistencia_UNIQUE` (`idtipoasistencia` ASC))
ENGINE = InnoDB;
```

```
-----
-- Table `tesis`.`asistenciadocente`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`asistenciadocente` (
  `idasistenciadocente` INT NOT NULL AUTO_INCREMENT,
  `dia` DATE NOT NULL,
  `hora entrada` TIME NOT NULL,
  `hora salida` TIME NOT NULL,
  `idclase` INT NOT NULL,
  `idtipo` INT NOT NULL,
  PRIMARY KEY (`idasistenciadocente`),
  UNIQUE INDEX `idasistenciadocente_UNIQUE` (`idasistenciadocente`
ASC),
  UNIQUE INDEX `idclase_UNIQUE` (`idclase` ASC),
  UNIQUE INDEX `idtipo_UNIQUE` (`idtipo` ASC),
  CONSTRAINT `asistenciadocente_clase`
  FOREIGN KEY (`idclase`)
  REFERENCES `tesis`.`clase` (`idClase`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `asistenciadocente_tipo`
  FOREIGN KEY (`idtipo`)
  REFERENCES `tesis`.`tipoasistencia` (`idtipoasistencia`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
USE `tesis` ;
```

-- Table `tesis`.`apoderado`

```
CREATE TABLE IF NOT EXISTS `tesis`.`apoderado` (  
  `idApoderado` INT(11) NOT NULL AUTO_INCREMENT,  
  `apellidos` VARCHAR(30) NOT NULL,  
  `nombres` VARCHAR(30) NOT NULL,  
  `direccion` VARCHAR(45) NOT NULL,  
  `telefono` VARCHAR(12) NOT NULL,  
  `email` VARCHAR(45) NOT NULL,  
  `vinculo` VARCHAR(15) NOT NULL,  
  `condicion` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idApoderado`),  
  UNIQUE INDEX `idApoderado_UNIQUE` (`idApoderado` ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1  
COMMENT = '          ';
```

-- Table `tesis`.`alumno`

```
CREATE TABLE IF NOT EXISTS `tesis`.`alumno` (  
  `idAlumno` INT(11) NOT NULL AUTO_INCREMENT,  
  `apellidos` VARCHAR(30) NOT NULL,  
  `nombres` VARCHAR(30) NOT NULL,  
  `fecha_de_nacimiento` DATE NOT NULL,  
  `sexo` VARCHAR(15) NOT NULL,  
  `direccion` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(45) NOT NULL,  
  `idApoderadoPrincipal` INT(11) NOT NULL,  
  `idApoderadoAlterno` INT(11) NOT NULL,  
  PRIMARY KEY (`idAlumno`),  
  UNIQUE INDEX `idApoderado1_UNIQUE` (`idApoderadoPrincipal` ASC),  
  UNIQUE INDEX `idApoderado2_UNIQUE` (`idApoderadoAlterno` ASC),  
  UNIQUE INDEX `idAlumno_UNIQUE` (`idAlumno` ASC),  
  CONSTRAINT `alumno_apoderadoprincipal_FK`  
    FOREIGN KEY (`idApoderadoPrincipal`)  
    REFERENCES `tesis`.`apoderado` (`idApoderado`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `alumno_apoderadosecundario_FK`  
    FOREIGN KEY (`idApoderadoAlterno`)  
    REFERENCES `tesis`.`apoderado` (`idApoderado`)
```

```

    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`aula`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`aula` (
  `idAula` INT(11) NOT NULL AUTO_INCREMENT,
  `Nombre` INT(11) NOT NULL,
  `Piso` INT(11) NOT NULL,
  PRIMARY KEY (`idAula`),
  UNIQUE INDEX `idAula_UNIQUE` (`idAula` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`controlasistenciaalumno`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`controlasistenciaalumno` (
  `idAsistencia` INT(11) NOT NULL AUTO_INCREMENT,
  `tipo` VARCHAR(45) NOT NULL,
  `idAlumno` INT(11) NOT NULL,
  `idClase` INT(11) NOT NULL,
  PRIMARY KEY (`idAsistencia`),
  UNIQUE INDEX `idAlumno_UNIQUE` (`idAlumno` ASC),
  UNIQUE INDEX `idClase_UNIQUE` (`idClase` ASC),
  UNIQUE INDEX `idAsistencia_UNIQUE` (`idAsistencia` ASC),
  INDEX `alumno_FK_idx` (`idAlumno` ASC),
  INDEX `clase_FK_idx` (`idClase` ASC),
  CONSTRAINT `controlasistencia_alumno_FK`
    FOREIGN KEY (`idAlumno`)
    REFERENCES `tesis`.`alumno` (`idAlumno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `controlasistencia_clase_FK`
    FOREIGN KEY (`idClase`)
    REFERENCES `tesis`.`clase` (`idClase`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

```

DEFAULT CHARACTER SET = latin1;

```
-----  
-- Table `tesis`.`docente`  
-----  
CREATE TABLE IF NOT EXISTS `tesis`.`docente` (  
  `idDocente` INT(11) NOT NULL AUTO_INCREMENT,  
  `apellidos` VARCHAR(30) NOT NULL,  
  `nombres` VARCHAR(30) NOT NULL,  
  `sexo` VARCHAR(15) NOT NULL,  
  `fecha_de_nacimiento` DATE NOT NULL,  
  `direccion` VARCHAR(45) NOT NULL,  
  `profesion` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idDocente`),  
  UNIQUE INDEX `idDocente_UNIQUE` (`idDocente` ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tesis`.`nivel`  
-----  
CREATE TABLE IF NOT EXISTS `tesis`.`nivel` (  
  `idNivel` INT(11) NOT NULL AUTO_INCREMENT,  
  `nivel` VARCHAR(15) NOT NULL,  
  PRIMARY KEY (`idNivel`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tesis`.`grado`  
-----  
CREATE TABLE IF NOT EXISTS `tesis`.`grado` (  
  `idGrado` INT(11) NOT NULL AUTO_INCREMENT,  
  `numeroGrado` VARCHAR(15) NOT NULL,  
  `idNivel` INT(11) NOT NULL,  
  `idDocente` INT(11) NOT NULL,  
  PRIMARY KEY (`idGrado`),  
  UNIQUE INDEX `idGrado_UNIQUE` (`idGrado` ASC),  
  UNIQUE INDEX `idNivel_UNIQUE` (`idNivel` ASC),  
  UNIQUE INDEX `idDocente_UNIQUE` (`idDocente` ASC),  
  INDEX `nivel_FK_idx` (`idNivel` ASC),
```

```

INDEX `grado_docente_FK_idx` (`idDocente` ASC),
CONSTRAINT `grado_docente_FK`
  FOREIGN KEY (`idDocente`)
  REFERENCES `tesis`.`docente` (`idDocente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `grado_nivel_FK`
  FOREIGN KEY (`idNivel`)
  REFERENCES `tesis`.`nivel` (`idNivel`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`cursosporgrado`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`cursosporgrado` (
  `idCursosPorGrado` INT(11) NOT NULL AUTO_INCREMENT,
  `idGrado` INT(11) NOT NULL,
  `idCurso` INT(11) NOT NULL,
  PRIMARY KEY (`idCursosPorGrado`),
  UNIQUE INDEX `idCursosPorGrado_UNIQUE` (`idCursosPorGrado` ASC),
  UNIQUE INDEX `idGrado_UNIQUE` (`idGrado` ASC),
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC),
  INDEX `grado_FK_idx` (`idGrado` ASC),
  INDEX `curso_FK_idx` (`idCurso` ASC),
  CONSTRAINT `cursoxgrado_curso_FK`
    FOREIGN KEY (`idCurso`)
    REFERENCES `tesis`.`curso` (`idCurso`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `cursoxgrado_grado_FK`
    FOREIGN KEY (`idGrado`)
    REFERENCES `tesis`.`grado` (`idGrado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`evaluacion`
-----

```

```

-----
CREATE TABLE IF NOT EXISTS `tesis`.`evaluacion` (
  `idEvaluacion` INT(11) NOT NULL AUTO_INCREMENT,
  `idAlumno` INT(11) NOT NULL,
  `idCurso` INT(11) NOT NULL,
  `tipo` VARCHAR(30) NOT NULL,
  `nota` INT(11) NOT NULL,
  PRIMARY KEY (`idEvaluacion`),
  UNIQUE INDEX `idEvaluacion_UNIQUE` (`idEvaluacion` ASC),
  UNIQUE INDEX `idAlumno_UNIQUE` (`idAlumno` ASC),
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC),
  INDEX `nota_alumno_FK_idx` (`idAlumno` ASC),
  INDEX `nota_curso_FK_idx` (`idCurso` ASC),
  CONSTRAINT `evaluacion_alumno_FK`
    FOREIGN KEY (`idAlumno`)
    REFERENCES `tesis`.`alumno` (`idAlumno`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `evaluacion_curso_FK`
    FOREIGN KEY (`idCurso`)
    REFERENCES `tesis`.`curso` (`idCurso`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`horario`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`horario` (
  `idHorario` INT(11) NOT NULL AUTO_INCREMENT,
  `idCurso` INT(11) NOT NULL,
  `Dia` VARCHAR(45) NOT NULL,
  `Hora` VARCHAR(45) NOT NULL,
  `idAula` INT(11) NOT NULL,
  `idDocente` INT(11) NOT NULL,
  `idGrado` INT(11) NOT NULL,
  PRIMARY KEY (`idHorario`),
  UNIQUE INDEX `idAula_UNIQUE` (`idAula` ASC),
  UNIQUE INDEX `idDocente_UNIQUE` (`idDocente` ASC),
  UNIQUE INDEX `idGrado_UNIQUE` (`idGrado` ASC),
  UNIQUE INDEX `idHorario_UNIQUE` (`idHorario` ASC),
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC),

```

```

INDEX `horario_curso_FK_idx` (`idCurso` ASC),
INDEX `horario_aula_FK_idx` (`idAula` ASC),
INDEX `horario_docente_FK_idx` (`idDocente` ASC),
INDEX `horario_grado_FK_idx` (`idGrado` ASC),
CONSTRAINT `horario_aula_FK`
  FOREIGN KEY (`idAula`)
  REFERENCES `tesis`.`aula` (`idAula`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `horario_curso_FK`
  FOREIGN KEY (`idCurso`)
  REFERENCES `tesis`.`curso` (`idCurso`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `horario_docente_FK`
  FOREIGN KEY (`idDocente`)
  REFERENCES `tesis`.`docente` (`idDocente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `horario_grado_FK`
  FOREIGN KEY (`idGrado`)
  REFERENCES `tesis`.`grado` (`idGrado`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`tiporequisito`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`tiporequisito` (
  `idTipoRequisito` INT(11) NOT NULL AUTO_INCREMENT,
  `TipoDeRequisito` VARCHAR(25) NOT NULL,
  PRIMARY KEY (`idTipoRequisito`),
  UNIQUE INDEX `idTipoRequisito_UNIQUE` (`idTipoRequisito` ASC))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`requisito`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`requisito` (

```

```

`idRequisito` INT(11) NOT NULL AUTO_INCREMENT,
`requisito` TEXT NOT NULL,
`idTipoRequisito` INT(11) NOT NULL,
PRIMARY KEY (`idRequisito`),
UNIQUE INDEX `idRequisitosEgreso_UNIQUE` (`idRequisito` ASC),
UNIQUE INDEX `idTipoRequisito_UNIQUE` (`idTipoRequisito` ASC),
CONSTRAINT `requisito_tiporequisito`
  FOREIGN KEY (`idTipoRequisito`)
  REFERENCES `tesis`.`tiporequisito` (`idTipoRequisito`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`listaderequisitos`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`listaderequisitos` (
  `idListaDeRequisitos` INT(11) NOT NULL AUTO_INCREMENT,
  `idRequisito` INT(11) NOT NULL,
  PRIMARY KEY (`idListaDeRequisitos`),
  UNIQUE INDEX `idListaDeRequisitos_UNIQUE` (`idListaDeRequisitos`
ASC),
  UNIQUE INDEX `idRequisito_UNIQUE` (`idRequisito` ASC),
  CONSTRAINT `listaderequisitos_requisito_FK`
    FOREIGN KEY (`idRequisito`)
    REFERENCES `tesis`.`requisito` (`idRequisito`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`matricula`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`matricula` (
  `idMatricula` INT(11) NOT NULL AUTO_INCREMENT,
  `idAlumno` INT(11) NOT NULL,
  `idGrado` INT(11) NOT NULL,
  `fecha` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`idMatricula`),
  UNIQUE INDEX `idAlumno_UNIQUE` (`idAlumno` ASC),

```

```

UNIQUE INDEX `idGrado_UNIQUE` (`idGrado` ASC),
UNIQUE INDEX `idMatricula_UNIQUE` (`idMatricula` ASC),
INDEX `alumno_fk_idx` (`idAlumno` ASC),
INDEX `grado_FK_idx` (`idGrado` ASC),
CONSTRAINT `matricula_alumno_fk`
  FOREIGN KEY (`idAlumno`)
  REFERENCES `tesis`.`alumno` (`idAlumno`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `matricula_grado_FK`
  FOREIGN KEY (`idGrado`)
  REFERENCES `tesis`.`grado` (`idGrado`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`plandeestudios`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`plandeestudios` (
  `idPlanDeEstudios` INT(11) NOT NULL AUTO_INCREMENT,
  `idCurso` INT(11) NOT NULL,
  `Introduccion` TEXT NOT NULL,
  `Cuerpo` TEXT NOT NULL,
  PRIMARY KEY (`idPlanDeEstudios`),
  UNIQUE INDEX `idCurso_UNIQUE` (`idCurso` ASC),
  INDEX `plan_estudios_curso_FK_idx` (`idCurso` ASC),
  CONSTRAINT `plandeestudios_curso_FK`
    FOREIGN KEY (`idCurso`)
    REFERENCES `tesis`.`curso` (`idCurso`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`postulante`
-----
CREATE TABLE IF NOT EXISTS `tesis`.`postulante` (
  `idPostulante` INT(11) NOT NULL AUTO_INCREMENT,
  `apellidos` VARCHAR(30) NOT NULL,

```

```

`nombres` VARCHAR(30) NOT NULL,
`fechadenacimiento` DATE NOT NULL,
`sexo` VARCHAR(15) NOT NULL,
`direccion` VARCHAR(30) NOT NULL,
`distrito` VARCHAR(20) NOT NULL,
`estado` VARCHAR(45) NOT NULL,
`idApoderadoPrincipal` INT(11) NOT NULL,
`idApoderadoSecundario` INT(11) NOT NULL,
PRIMARY KEY (`idPostulante`),
UNIQUE INDEX `idApoderadoPrincipal_UNIQUE` (`idApoderadoPrincipal`
ASC),
UNIQUE INDEX `idApoderadoSecundario_UNIQUE`
(`idApoderadoSecundario` ASC),
UNIQUE INDEX `idPostulante_UNIQUE` (`idPostulante` ASC),
CONSTRAINT `postulante_apoderadoprinicipal_FK`
FOREIGN KEY (`idApoderadoPrincipal`)
REFERENCES `tesis`.`apoderado` (`idApoderado`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `postulante_apoderadosecundario`
FOREIGN KEY (`idApoderadoSecundario`)
REFERENCES `tesis`.`apoderado` (`idApoderado`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

```

```

-----
-- Table `tesis`.`seccion`
-----

```

```

CREATE TABLE IF NOT EXISTS `tesis`.`seccion` (
`idSeccion` INT(11) NOT NULL AUTO_INCREMENT,
`letra` VARCHAR(1) NULL DEFAULT NULL,
`idGrado` INT(11) NOT NULL,
PRIMARY KEY (`idSeccion`),
UNIQUE INDEX `idSeccion_UNIQUE` (`idSeccion` ASC),
UNIQUE INDEX `idGrado_UNIQUE` (`idGrado` ASC),
INDEX `grado_FK_idx` (`idGrado` ASC),
CONSTRAINT `seccion_grado_FK`
FOREIGN KEY (`idGrado`)
REFERENCES `tesis`.`grado` (`idGrado`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-----
-- Table `tesis`.`solicitud`
-----
```

```
CREATE TABLE IF NOT EXISTS `tesis`.`solicitud` (
  `idSolicitud` INT(11) NOT NULL AUTO_INCREMENT,
  `idPostulante` INT(11) NOT NULL,
  `idApoderadoPrincipal` INT(11) NOT NULL,
  `idApoderadoSecundario` INT(11) NOT NULL,
  `fecha` DATE NOT NULL,
  `estado` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`idSolicitud`),
  UNIQUE INDEX `idApoderado1_UNIQUE` (`idApoderadoPrincipal` ASC),
  UNIQUE INDEX `idApoderado2_UNIQUE` (`idApoderadoSecundario` ASC),
  UNIQUE INDEX `idPostulante_UNIQUE` (`idPostulante` ASC),
  UNIQUE INDEX `idSolicitud_UNIQUE` (`idSolicitud` ASC),
  INDEX `postulante_FK_idx` (`idPostulante` ASC),
  INDEX `apoderado1_FK_idx` (`idApoderadoPrincipal` ASC),
  INDEX `apoderado2_FK_idx` (`idApoderadoSecundario` ASC),
  CONSTRAINT `apoderado1_FK`
    FOREIGN KEY (`idApoderadoPrincipal`)
    REFERENCES `tesis`.`apoderado` (`idApoderado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `apoderado2_FK`
    FOREIGN KEY (`idApoderadoSecundario`)
    REFERENCES `tesis`.`apoderado` (`idApoderado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `postulante_FK`
    FOREIGN KEY (`idPostulante`)
    REFERENCES `tesis`.`postulante` (`idPostulante`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-----
-- Table `tesis`.`tipodeevaluacion`
-----
```

```
CREATE TABLE IF NOT EXISTS `tesis`.`tipodeevaluacion` (  
  `idTipoDeEvaluacion` INT(11) NOT NULL AUTO_INCREMENT,  
  `TipoDeEvaluacion` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`idTipoDeEvaluacion`),  
  UNIQUE INDEX `idTipoDeEvaluacion_UNIQUE` (`idTipoDeEvaluacion`  
ASC))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

5. DISCUSIÓN DE RESULTADOS

La hipótesis planteada fue la siguiente:

“El uso de la metodología RUP y el framework Cake PHP permite el diseño de un sistema Integrado de para el proceso de Gestión Académica de la Institución Educativa Privada “Sagrado Corazón de Jesús”.

Y las variables son las siguientes:

Variable Independiente: Diseño de un Sistema Integrado de Gestión Académica de la Institución Educativa Privada “Sagrado Corazón de Jesús”.

Definición Conceptual: Es el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física (Taylor, 1959; citado por Torossi, 2015).

Definición Operacional: El grado de conformidad del diseño con los requerimientos y especificaciones de los usuarios.

Variable Dependiente: Prototipo de Sistema Integrado de Gestión utilizando la Metodología RUP y el Framework Cake PHP.

Definición Conceptual: Es un modelo a escala de lo real, pero no necesariamente funcional del sistema final, ya que no lleva a cabo la totalidad de las funciones necesarias de éste. (Google Sites, 2016)

Definición Operacional: El grado de cumplimiento de la elaboración del prototipo con la metodología RUP y el Framework Cake PHP.

5.3. Variable: El grado de cumplimiento de la elaboración del prototipo con la metodología RUP y el Framework Cake PHP.

Para esto usaremos un método de puntuación calificada por expertos, basándonos en la tesis de maestría del Ingeniero Vladimir Urrelo titulada “Arquitectura de Software para Web Adaptativa Manejadora Del Secuenciamiento de Objetos de Aprendizaje en el Primer Curso Universitario De Programación”, los cuales nos darán su opinión acerca del prototipo con características que nos dirán cuanto hemos cumplido con el desarrollo, y los resultados os podemos apreciar en la tabla 13 que está a continuación en la siguiente página:

Atributo de Calidad	Puntuación Mínima Experto 1	Experto 1	Puntuación Mínima Experto 2	Experto 2	Puntuación Mínima Experto 3	Experto 3	Promedio Puntuación Mínima Requerida	Promedio Puntuación Expertos	Resultado
Configurabilidad	60	90	65	97	70	95	65.0	94.0	MEJORO
Integrabilidad	65	75	70	78	75	81	70.0	78.0	MEJORO
Integridad	70	89	70	84	70	85	70.0	86.0	MEJORO
Interoperabilidad	70	87	65	77	70	87	68.3	83.7	MEJORO
Modificabilidad	75	69	65	72	70	71	70.0	70.7	MEJORO
Mantenibilidad	70	62	70	67	75	70	71.7	66.3	NO MEJORO
Portabilidad	75	89	65	90	80	92	73.3	90.3	MEJORO
Reusabilidad	75	81	65	85	75	83	71.7	83.0	MEJORO
Escalabilidad	75	87	70	90	70	89	71.7	88.7	MEJORO
Capacidad de Prueba	70	77	65	82	75	80	70.0	79.7	MEJORO
Promedio de Atributos de Calidad no Observables							70.2	82.0	MEJORO

Tabla 12: Puntuación de Atributos No Observables por Expertos

Fuente: Elaboración Propia

En la figura 126 vemos los resultados en un grafica demostrando que el prototipo tiene buena puntuación

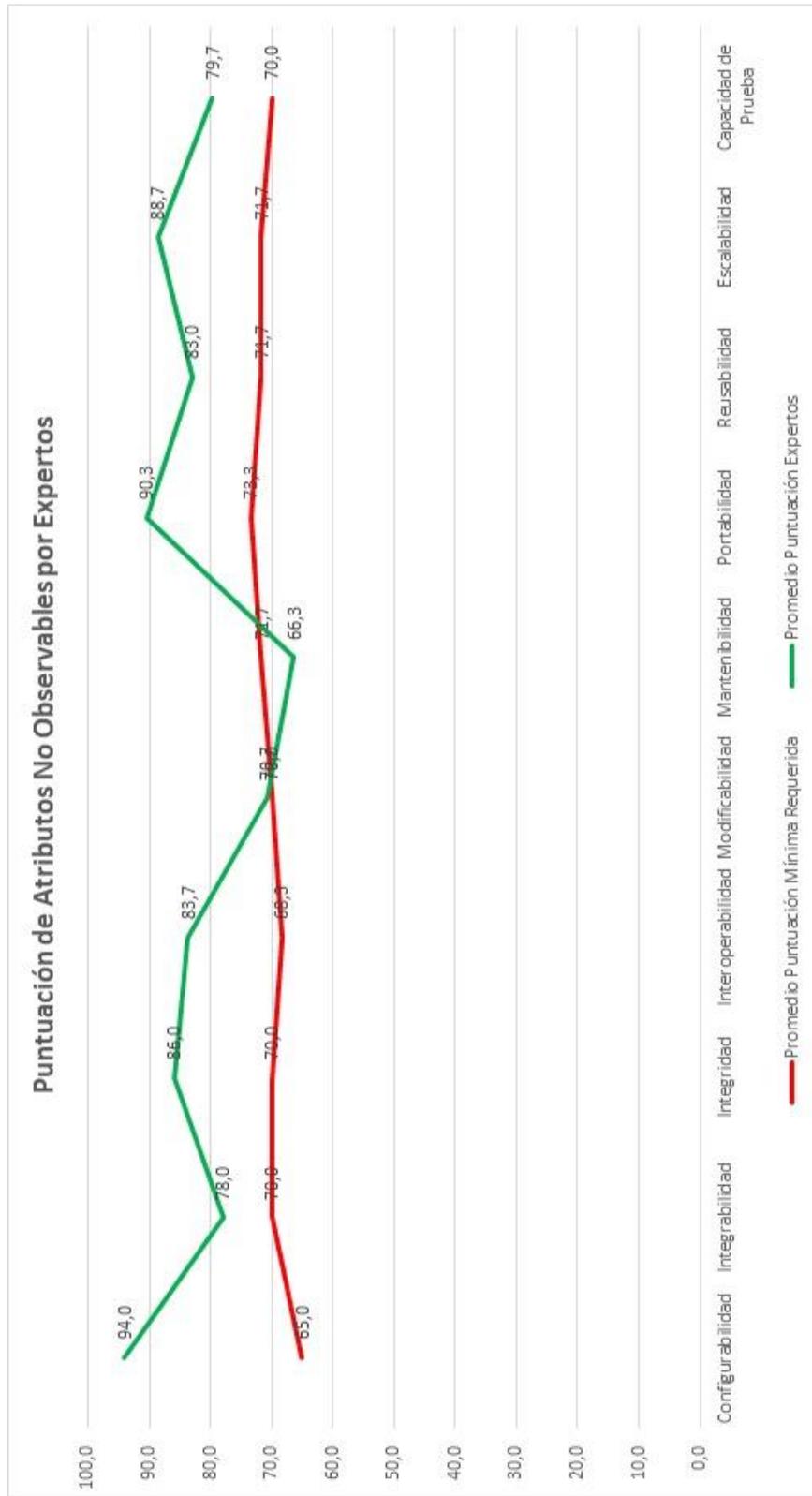


Figura 126: Gráfica de Puntuación de Atributos No Observables por Expertos
Fuente: Elaboración Propia

En la tabla 13 tenemos la Puntuación de Atributos Observables por Expertos, y luego un análisis gráfico en la figura 127:

Atributo de Calidad	Puntuación Mínima Experto 1	Experto 1	Puntuación Mínima Experto 2	Experto 2	Puntuación Mínima Experto 3	Experto 3	Promedio Puntuación Mínima Requerida	Promedio Puntuación Expertos	Resultado
Disponibilidad	75	75	70	72	75	80	73.3	75.7	MEJORO
Confidencialidad	70	60	75	58	70	53	71.7	57.0	NO MEJORO
Funcionalidad	75	82	70	84	77	89	74.0	85.0	MEJORO
Desempeño	80	86	75	87		90	77.5	87.7	MEJORO
Confiabilidad	75	80	75	88		90	75.0	86.0	MEJORO
Seguridad externa	75	68	70	57	80	64	75.0	63.0	NO MEJORO
Seguridad interna	75	75	70	80	80	77	75.0	77.3	MEJORO
Promedio de Atributos de Calidad Observables							74.5	76.0	MEJORO

Tabla 13: Puntuación de Atributos Observables por Expertos
Fuente: Elaboración Propia

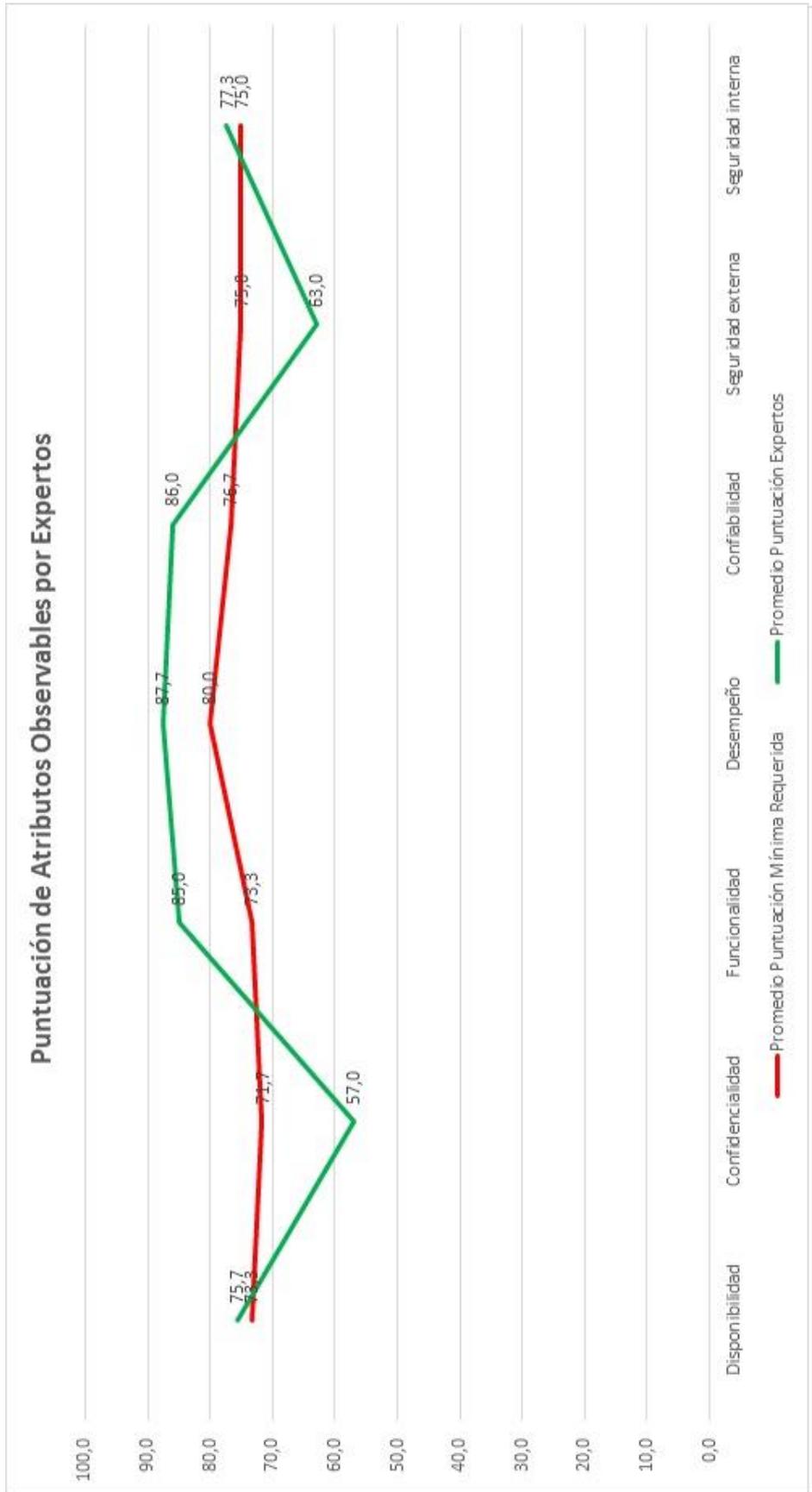


Figura 127: Gráfica de Puntuación de Atributos Observables por Expertos
Fuente: Elaboración Propia

5.4. Variable: El grado de conformidad del diseño con los requerimientos y especificaciones de los usuarios

Se aplicó la siguiente encuesta a los usuarios:

1. ¿El prototipo mejoró su trabajo?
2. ¿Considera fácil usar el prototipo?
3. ¿Qué le parece la velocidad de procesamiento de datos?
4. ¿Le parece el prototipo disponible a todas sus funciones?
5. Finalmente, ¿Considera que el prototipo funciona correctamente?

Por motivos de procesamiento de datos en Excel, se usó la siguiente escala numérica:

CALIFICACION

bueno = 3
 regular = 2
 malo = 1

Los resultados obtenidos los vemos en la tabla 14:

	PREGUNTA 1	PREGUNTA 2	PREGUNTA 3	PREGUNTA 4	PREGUNTA 5
Director	3	2	2	2	3
coordinador académico	2	2	3	2	3
Secretaria	3	3	3	2	3
docente 1	2	3	2	3	2
docente 2	3	2	2	2	3

Tabla 14: Resultados de Encuesta

Fuente: Elaboración Propia

En la tabla 15 se realizó el conteo según la escala aplicada (Bueno, Regular y Malo)

	PREGUNTA 1	PREGUNTA 2	PREGUNTA 3	PREGUNTA 4	PREGUNTA 5
BUENO	3	2	2	1	2
REGULAR	2	3	3	4	3
MALO	0	0	0	0	0

Tabla 15: Conteo de Resultados de Tabla 14

Fuente: Elaboración Propia

Los resultados son los siguientes (Figuras 128, 129, 130, 131 y 132):

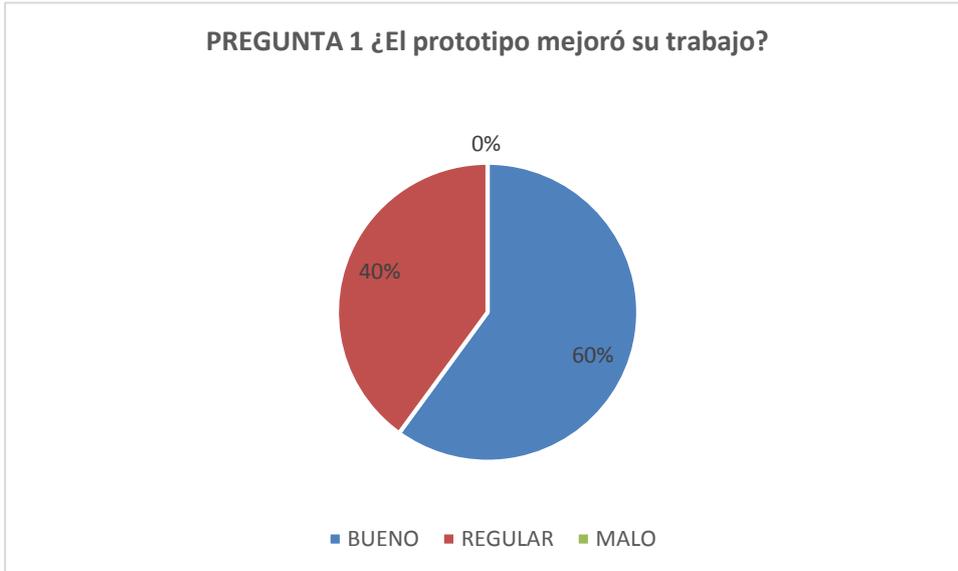


Figura 128: Pregunta 1
Fuente: Elaboración Propia

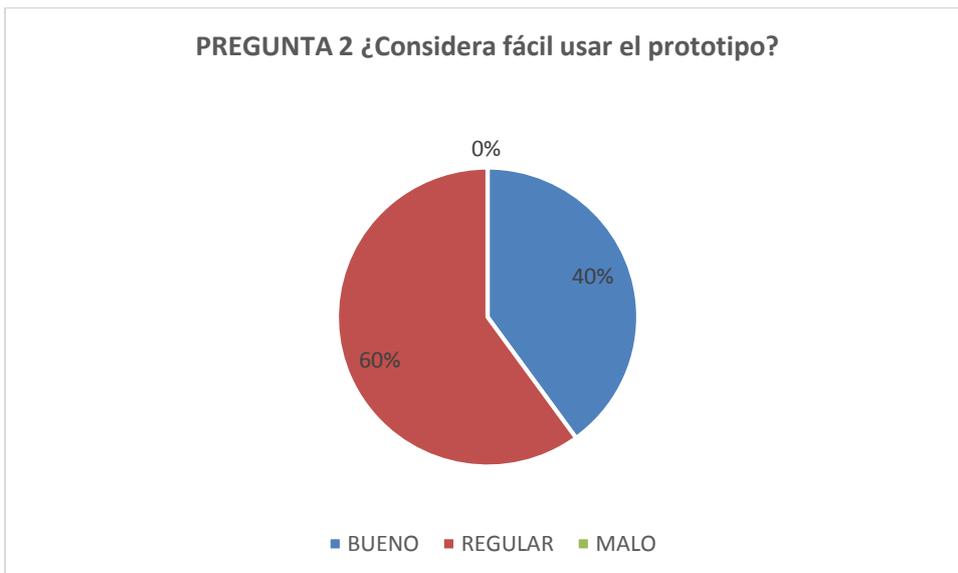


Figura 129: Pregunta 2
Fuente: Elaboración Propia

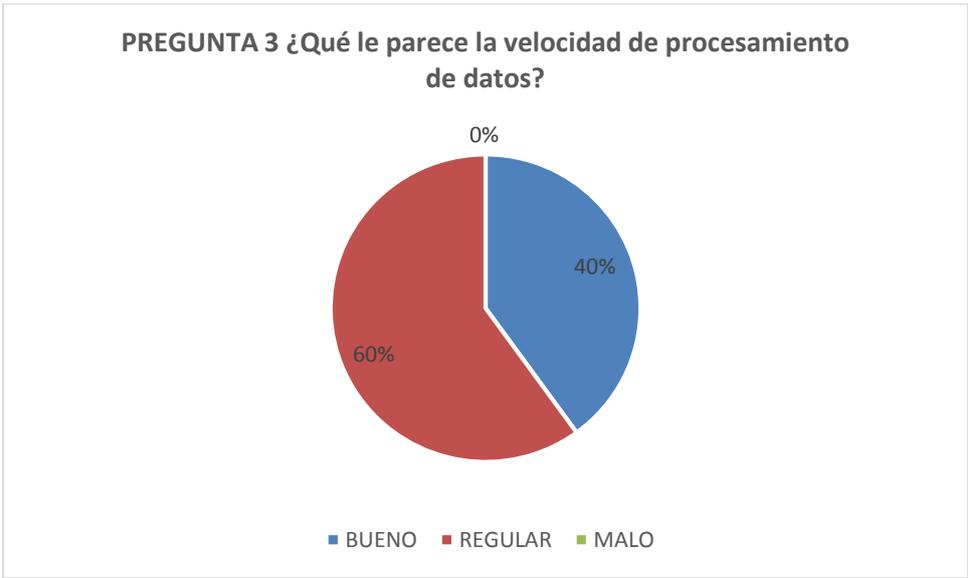


Figura 130: Pregunta 3
Fuente: Elaboración Propia

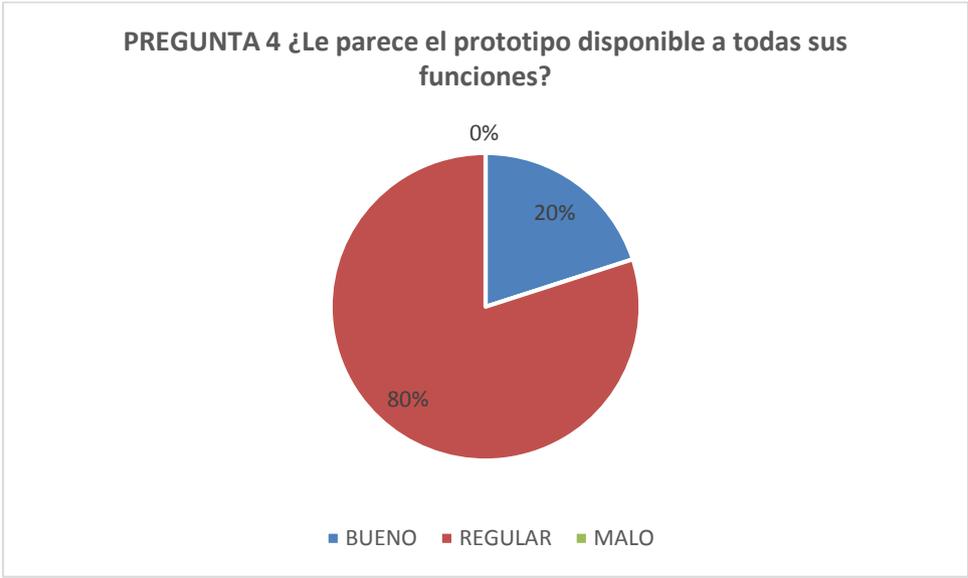


Figura 131: Pregunta 4
Fuente: Elaboración Propia

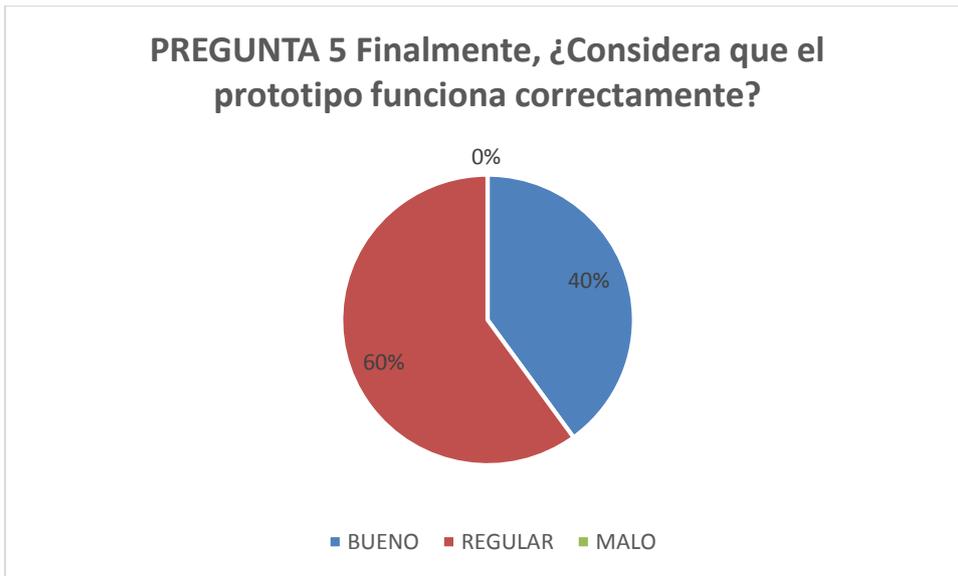


Figura 132: Pregunta 5
Fuente: Elaboración Propia

La primera variable, el promedio de la puntuación de los expertos ha mejorado en un **16,9%** con respecto al promedio del promedio de las puntuaciones mínimas dadas por los mismos.

La segunda variable ha mejorado en un **100%**, ya que se obtuvo un promedio de:
 40% aceptan el prototipo como bueno
 60% aceptan el prototipo como regular
 0% aceptan el prototipo como malo

Como se ve, podemos deducir que todos están al menos conformes, puesto que antes no había ningún prototipo ni sistema de información, aunque la mayoría lo considera regular debido a que es aún un prototipo de sistema y no un sistema completado al 100%, caso contrario, muy probablemente la gran mayoría calificarían al ya sistema como bueno.

6. CONCLUSIONES

- El framework cakephp nos permitio crear 40 interfaces, pues al crear las opciones de modificar, listar y eliminar datos, lo hace más fácilmente que los métodos convencionales de programación, porque genera los modelos y plantillas automáticamente para cumplir dichas funciones.
- La metodología RUP ha permitido crear 18 casos de uso, 18 Especificaciones de casos de uso, permitiendo cubrir todas las necesidades de los usuarios y en general de la institución para lograr desarrollar este prototipo lo mejor posible

- También la metodología RUP ha permitido crear 18 diagramas de actividad, 18 diagramas de colaboración y 18 diagramas de secuencia, además 25 entidades de base de datos.
- La validación del prototipo resultó correcta y satisfactoria tanto para los expertos como los usuarios de este “futuro” sistema, superando en un 16,9% y en un 100% respectivamente.

7. RECOMENDACIONES

- La institución debe instalar una red local in situ para garantizar el correcto funcionamiento del sistema y puedan aprovecharse al máximo todas las ventajas que ofrece
- Inicialmente, este prototipo se ha probado en el sistema operativo Windows, pero también podría funcionar en un entorno Linux, ya que utiliza herramientas de software libre como php, Xampp y Mysql.
- Este prototipo, al no ser un sistema terminado, sería más fácil de adaptarse fácilmente a otras instituciones puesto a que no está en un estado maduro
- Se puede mejorar las interfaces de manera que puedan ingresar y gestionar mejor la entrada y salida de datos e información, por ejemplo, en la interface de postulante, en vez de la columna id_apoderado, podría mostrarse directamente el nombre del apoderado.

8. REFERENCIAS BIBLIOGRÁFICAS

- Álvarez, M. (2016). *Desarrollo Web*. Obtenido de <http://www.desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>
- Cake Software Foundation. (2012). *Principios básicos de CakePHP*. Obtenido de <http://book.cakephp.org/1.3/es/The-Manual/Beginning-With-CakePHP/What-is-CakePHP-Why-Use-it.html>
- Castro, J. L. (2008). *Gestión Académica De Alumnos Con Necesidades Educativas Especiales*. Madrid.
- Centers For Medicare & Medicaid Services. (2015). *SELECTING A DEVELOPMENT APPROACH*. Obtenido de <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>
- Foundation, C. S. (2012). *Principios básicos de CakePHP*. Obtenido de <http://book.cakephp.org/1.3/es/The-Manual/Basic-Principles-of-CakePHP.html>
- Google Sites. (2016). *Metodología Gestión de Requerimientos*. Obtenido de <https://sites.google.com/site/metodologiareq/capitulo-iv>
- Gutierrez, J. (2015). *¿Qué es un framework web?* Obtenido de http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf

- IBM. (1998). *Rational Unified Process - Best Practices for Software*. Obtenido de http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- Mazaira Rodríguez, Z. (2010). *El proceso académico y su administración en la educación superior*. Obtenido de <http://www.gestiopolis.com/el-proceso-academico-y-su-administracion-en-la-educacion-superior/>
- Ministerio de Educación. (2016). <http://www.colegiosenperu.com>. Obtenido de <http://www.colegiosenperu.com>
- Ministerio de Educación de Colombia. (2015). *Colombia Aprende - La Red del Conocimiento*. Obtenido de <http://www.colombiaaprende.edu.co/html/home/1592/article-227402.html>
- PortableApps. (2016). *PortableApps*. Obtenido de <http://portableapps.com/apps/development/xampp>
- Rational Software. (1998). *Rational Unified Process - Best Practices for Software*. Obtenido de http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- Torossi, G. M. (2015). *Diseño Estructurado de Sistemas*. Obtenido de http://exa.unne.edu.ar/informatica/anasistem2/public_html/apuntes/de1.pdf
- Tutorials Point. (2014). *MIS Tutorial*. Obtenido de http://119.235.48.163/library/Ebooks/Others/mis_tutorial.pdf
- Urrelo, V. (2012). *Arquitectura de Software para Web Adaptativa Manejadora Del Secuenciamiento de Objetos de Aprendizaje en el Primer Curso Universitario De Programación*.

ANEXOS

ANEXO 01 FORMATOS DE PROCESOS ACADÉMICOS DEL COLEGIO

Anexo 01.a: Formato de Registro de Solicitud de Matricula



IEP
"Suprema Corazón de Jesús"
 ED 00694 - 04 - UGRA
 Los Esteros Nº 109 Casa Grande Telf. 433165

FOTO
 4 X 4
 ACTUAL

AÑO QUE CURSA	<input type="checkbox"/> 1 ^{er.} <input type="checkbox"/> 2 ^{do.} <input type="checkbox"/> 3 ^{er.} <input type="checkbox"/> 4 ^{to.} <input type="checkbox"/> 5 ^{to.} <input type="checkbox"/> 6 ^{to.} <input type="checkbox"/> 7 ^{mo.}
DATOS DEL ESTUDIANTE	Datos Filiatorios Apellido _____ Nombre Completo _____ Fecha de Nacimiento _____ DNI _____ Domicilio Actualizado Calle _____ Altura _____ Piso _____ Dpto _____ Mzna. _____ Casa _____ Barrio _____ Localidad _____ Código Postal _____ Teléfono Fijo _____ Teléfono Celular _____ Email _____
DEL PADRE	Datos Filiatorios Apellido _____ Nombre Completo _____ Fecha de Nacimiento _____ DNI _____ Domicilio Actualizado Calle _____ Altura _____ Piso _____ Dpto _____ Mzna. _____ Casa _____ Barrio _____ Localidad _____ Código Postal _____ Teléfono Fijo _____ Teléfono Celular _____ Email _____ Domicilio Laboral Empresa _____ Calle _____ Altura _____ Piso _____ Dpto _____ Mzna. _____ Casa _____ Barrio _____ Localidad _____ Código Postal _____ Teléfono Contacto _____ Email _____
DE LA MADRE	Datos Filiatorios Apellido De Soltera _____ Nombre Completo _____ Fecha de Nacimiento _____ DNI _____ Domicilio Actualizado Calle _____ Altura _____ Piso _____ Dpto _____ Mzna. _____ Casa _____ Barrio _____ Localidad _____ Código Postal _____ Teléfono Fijo _____ Teléfono Celular _____ Email _____ Domicilio Laboral Empresa _____ Calle _____ Altura _____ Piso _____ Dpto _____ Mzna. _____ Casa _____ Barrio _____ Localidad _____ Código Postal _____ Teléfono Contacto _____ Email _____

Continúa al Dorsó

ANEXO 02:

ENCUESTA SOBRE USO DEL PROTOTIPO

Encuesta aplicada a los administrativos y docentes del colegio para conocer su opinión acerca del prototipo como futuro sistema a usar

1. ¿El prototipo mejoró su trabajo?
 - a) Bueno
 - b) Malo
 - c) Regular

2. ¿Considera fácil usar el prototipo?
 - a) Bueno
 - b) Malo
 - c) Regular

3. ¿Qué le parece la velocidad de procesamiento de datos?
 - a) Bueno
 - b) Malo
 - c) Regular

4. ¿Le parece el prototipo disponible a todas sus funciones?
 - a) Bueno
 - b) Malo
 - c) Regular

5. Finalmente, ¿Considera que el prototipo funciona correctamente?
 - a) Bueno
 - b) Malo
 - c) Regular

ANEXO 03 Tabla para Puntuación de Atributos de Calidad del Software No Observables

		Puntuación del Atributo por Expertos (Máxima Puntuación 100)			Promedio de la Arquitectura a Propuesta	Puntuación Mínima Requerida para el Proyecto
Atributo de Calidad	Experto 1	Experto 2	Experto 3			
Configurabilidad						
Integrabilidad						
Integridad						
Interoperabilidad						
Modificabilidad						
Mantenibilidad						
Portabilidad						
Reusabilidad						
Escalabilidad						
Capacidad de Prueba						
Promedio de Atributos de Calidad no Observables						

ANEXO 04: Tabla De Puntuación De Atributos De Calidad del Software Observables

		Puntuación del Atributo por Expertos (Máxima Puntuación 100)			Promedio de la Arquitectura a Propuesta	Puntuación Mínima Requerida para el Proyecto
Atributo de Calidad	Experto 1	Experto 2	Experto 3			
Disponibilidad						
Confidencialidad						
Funcionalidad						
Desempeño						
Confiabilidad						
Seguridad externa						
Seguridad interna						
Promedio de Atributos de Calidad Observables						