

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 10



Título: Sistema para la Gestión del Traslado de los Estudiantes en la UCI

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor:
David León Rivas

Tutores:
MC. Ing. María Antonia Montesino Menéndez
Ing. Yoandy Lichilin Ríos

Ciudad de la Habana
Julio 2009
“Año del 50 aniversario del triunfo de la Revolución.”

Declaración de Autoría.

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

David León Rivas

Autor

Lic. María Antonia Montesino Menéndez

Tutora

Ing. Yoandy Lichilin Ríos

Tutor

Dedicatoria.

A mi abuela y mi madre por dedicar sus vidas en mi formación y por esperar este momento para ver realizado este sueño que es de todos.

A mi tutora por ser la que pensó en este trabajo y a la que va dirigido.

A todas esas personas que forman parte de mi vida y que sienten tanto como yo este éxito.

Agradecimientos.

A mi amigo Luis Alberto por el apoyo que me han dado incondicionalmente durante el desarrollo del trabajo, y por estar siempre ahí.

A mis tutores María Antonia y Yoandy por haberme brindado toda la ayuda posible y por confiar en mí.

A mi familia por saber guiarme por el camino correcto, por convertirme en la persona que soy, por estar y quererme en todo momento, por el apoyo que me dan, por tener la confianza plena en mí, por ser los capaces de mantenerme en pie y por darme las fuerzas para seguir adelante.

A mi profesor Hugo por ser como un padre en esta Universidad que me ha brindado todo su apoyo.

A mis amigos de la Universidad por compartir estos años juntos, Radel, Felipe, Yoel, Héctor, Livan, a los de mi grupo.

A mi amiga Solangel por ser mi hermana, por brindarme su cariño y su apoyo.

A mi profesora Susel.

A mi amigo Fernando por el apoyo y cariño que me ha dedicado.

A todas las personas que no menciono y que de una forma u otra me ayudaron a llegar donde estoy.

Resumen.

La investigación realizada presenta como objetivo fundamental la creación de un software que gestione el proceso de traslado de los estudiantes de la UCI, dirigido a solucionar diversos problemas existentes en el área de Secretaría General: el aprovechamiento óptimo del tiempo empleado, en el análisis del proceso de traslado ya que se efectúa manualmente.

Como paso previo se estudió el procedimiento para efectuar el traslado de los estudiantes en la UCI, tanto los que ocurren entre facultades como los que se producen entre la UCI y otro CES del país, con el objetivo de determinar las principales insuficiencias del mismo y las necesidades del personal de secretaría para realizarlo más eficientemente. Tomando como base lo anteriormente expresado se propone el sistema de gestión de traslados que se presenta.

En el desarrollo de la investigación se emplearon métodos de los niveles teóricos y empíricos, entre ellos se utilizaron, el analítico, el histórico-lógico, la observación y la entrevista.

Palabras Claves

CES: Centro de Educación Superior.

BD: Base de datos.

CU: Casos de uso.

SG: Secretaría General.

SD: Secretaría Docente.

Tabla de contenido

Declaración de Autoría. I

Dedicatoria. II

Agradecimientos. III

Resumen. V

Palabras Claves. VI

Índice de tablas. XI

Índices de imágenes. XIV

Introducción. 1

Capítulo 1 “Fundamentación teórica”. 6

1.1. Introducción. 6

1.2. Situación actual de los traslados en el país. 6

1.3. Procedimiento de solicitud de traslado para otro CES en la UCI. 9

1.4. Proceso de solicitud de traslado para la UCI. 10

1.5. Procedimiento para los traslados internos de la UCI. 10

1.6. Reglas del negocio. 11

1.7. Propuesta de solución. 11

1.8. Tecnologías utilizadas para el desarrollo del sistema. 12

1.8.1. Arquitectura Cliente\Servidor. 12

1.8.2. Frameworks. 13

 1.8.2.1. CodeIgniter. 13

 1.8.2.2. Ext JS. 15

1.8.3. Javascript. 15

1.8.4. PHP. 16

1.8.5. PostgreSQL. 17

1.8.6. Servidor Web Apache. 19

1.8.7. Metodologías de Desarrollo de Software. 19

 1.8.7.1. RUP. 19

 1.8.7.2. Microsoft Solution Framework (MSF). 21

1.8.7.3. Extreme Programming (XP).....	23
1.8.8. ¿Por qué RUP?	24
1.8.9. UML.....	24
1.8.10. Visual Paradigm.	25
1.8.11. Zend Studio.....	26
1.9. Conclusiones parciales.	27
Capítulo 2 “Modelación del negocio y características del sistema propuesto”.	28
2.1. Introducción.....	28
2.2. Actores, trabajadores y procesos del negocio.	28
2.2.1. Actores del negocio.	28
2.2.2. Trabajadores del negocio.....	29
2.2.3. Diagrama de Casos de Uso del Negocio.	29
2.2.4. Descripción de los Casos de uso del negocio.....	30
2.2.5. Diagrama de Actividades.....	34
2.2.5.1. Sección solicitar traslados para otra Facultad.	34
2.2.5.2. Sección solicitar traslados para otro CES.....	35
2.2.5.3. Sección informar traslado.....	36
2.2.6. Diagrama de clases del modelo de objetos del negocio.....	37
2.3. Requerimientos Funcionales.	37
2.4. Requerimientos no funcionales.	39
2.4.1. Apariencia o interfaz externa.....	39
2.4.2. Usabilidad.....	39
2.4.3. Rendimiento.....	40
2.4.4. Hardware.....	40
2.4.5. Software.....	40
2.4.6. Políticos culturales.....	40
2.4.7. Portabilidad.....	40
2.4.8. Interoperabilidad.....	41
2.4.9. Seguridad.....	41
2.4.10. Confiabilidad.....	41
2.5. Descripción del sistema propuesto.....	41

2.5.1.	Actores del sistema.....	41
2.5.2.	Casos de uso del sistema.....	42
2.5.3.	Diagrama de casos de uso del sistema.....	43
2.5.4.	Descripción de los casos de uso.....	44
2.6.	Conclusiones parciales.....	56
Capítulo 3 “Diseño del sistema”.....		57
3.1.	Introducción.....	57
3.2.	Diseño.....	57
3.2.1.	Definición del modelo de diseño.....	57
3.2.1.1.	Modelo Vista Controlador (MVC).....	57
3.2.2.	Descripción de las clases.....	59
3.2.2.1.	Paquetes del diseño.....	59
3.2.2.2.	Paquete Administración. Descripción de las Clases.....	60
3.2.2.3.	Paquete Solicitudes. Descripción de las Clases.....	62
3.2.2.4.	Paquete Estudiantes. Descripción de las Clases.....	66
3.2.2.5.	Paquete Reportes. Descripción de las Clases.....	69
3.2.2.6.	Paquete Nomencladores. Clases.....	72
3.2.3.	Diagramas de clases del diseño.....	77
3.2.3.1.	CU: Autenticar usuario.....	77
3.2.3.2.	CU: Gestionar usuario.....	78
3.2.3.3.	CU: Gestionar estudiante interno.....	78
3.2.3.4.	CU: Gestionar estudiante externo.....	79
3.2.3.5.	CU: Gestionar solicitud de traslado interna.....	79
3.2.3.6.	CU: Gestionar solicitud de traslado externa.....	80
3.2.3.7.	CU: Gestionar resolución.....	80
3.2.3.8.	CU: Registrar expediente.....	81
3.2.3.9.	CU: Generar reportes.....	81
3.2.4.	Diseño de la Base de Datos.....	83
3.2.4.1.	Diagrama Entidad-Relación de la Base de Datos.....	83
3.2.4.2.	Diagramas de interacción.....	84
3.3.	Conclusiones parciales.....	84
Capítulo 4 “Implementación y prueba del sistema”.....		85
4.1.	Introducción.....	85
4.2.	Estructura del modelo de implementación.....	85
4.2.1.	Diagrama de componentes.....	85
4.2.2.	Diagrama de despliegue.....	86

4.3.	Prueba	86
4.3.1.	CU: Autenticar usuario.	88
4.3.2.	CU: Gestionar usuario.	89
4.3.3.	CU: Gestionar estudiante interno.	90
4.3.4.	CU: Gestionar estudiante externo.	91
4.3.5.	CU: Gestionar solicitud interna.....	92
4.3.6.	CU: Gestionar solicitud externa.....	93
4.3.7.	CU: Gestionar resolución.....	94
4.3.8.	CU: Registrar expediente.	94
4.4.	Conclusiones parciales.	95
	Conclusiones generales.	96
	Recomendaciones.	97
	Referencias bibliográficas.	98
	Bibliografía consultada.....	99
	Anexos.	100
	Glosario.....	108

Índice de tablas.

Tabla 1: Descripción de los actores del negocio. 28

Tabla 2: Descripción de los trabajadores del negocio. 29

Tabla 3: CU Solicitud de traslado. 32

Tabla 4: CU Informar solicitud. 33

Tabla 5: Requisitos funcionales del sistema. 38

Tabla 6: Actores del sistema. 42

Tabla 7: Casos de uso del sistema. 42

Tabla 8: Descripción del CU: Autenticar usuario. 44

Tabla 9: Descripción del CU: Gestionar usuario. 46

Tabla 10: Descripción del CU: Gestionar estudiante interno. 47

Tabla 11: Descripción del CU: Gestionar estudiante externo. 49

Tabla 12: Descripción del CU: Gestionar solicitud interna. 51

Tabla 13: Descripción del CU: Gestionar solicitud externa. 53

Tabla 14: Descripción del CU: Gestionar resolución. 55

Tabla 15: Descripción del CU: Registrar expediente. 55

Tabla 16: Descripción del CU: Generar reportes. 56

Tabla 17: Descripción de la clase CE_Usuario. 60

Tabla 18: Descripción de la clase CC_Usuario. 61

Tabla 19: Descripción de la clase CE_Solicitud. 63

Tabla 20: Descripción de la clase CE_Solicitud externa. 63

Tabla 21: Descripción de la clase CE_Solicitud interna. 63

Tabla 22: Descripción de la clase CE_Expediente. 63

Tabla 23: Descripción de la clase CE_Resolucion. 64

Tabla 24: Descripción de la clase CC_Solicitud salida. 64

Tabla 25: Descripción de la clase CC_Solicitud entrada. 64

Tabla 26: Descripción de la clase CC_Solicitud facultad. 65

Tabla 27: Descripción de la clase CC_Resolucion. 65

Tabla 28: Descripción de la clase CC_Expediente. 66

Tabla 29: Descripción de la clase CE_Estudiante. 67

Tabla 30: Descripción de la clase CE_Estudiante externo. 67

Tabla 31: Descripción de la clase CE_Estudiante interno. 67

Tabla 32: Descripción de la clase CC_Estudiante externo. 67

Tabla 33: Descripción de la clase CC_Estudiante interno. 68

Tabla 34: Descripción de la clase CC_SolAprobada. 70

Tabla 35: Descripción de la clase CC_SolUniversidad. 70

Tabla 36: Descripción de la clase CC_SolCarrera. 70

Tabla 37: Descripción de la clase CC_SolCurso. 70

Tabla 38: Descripción de la clase CC_SolFacultad. 70

Tabla 39: Descripción de la clase CC_SolNoAprobada.....	71
Tabla 40: Descripción de la clase CE_Municipio.....	73
Tabla 41: Descripción de la clase CE_Provincia.....	73
Tabla 42: Descripción de la clase CE_Facultad.....	73
Tabla 43: Descripción de la clase CE_Grupo.....	73
Tabla 44: Descripción de la clase CE_Tipo de curso.....	73
Tabla 45: Descripción de la clase CE_Curso escolar.....	74
Tabla 46: Descripción de la clase CE_Universidad.....	74
Tabla 47: Descripción de la clase CE_Carrera.....	74
Tabla 48: Descripción de la clase CE_Rol.....	74
Tabla 49: Descripción de la clase CC_Municipio.....	74
Tabla 50: Descripción de la clase CC_Provincia.....	75
Tabla 51: Descripción de la clase CC_Facultad.....	75
Tabla 52: Descripción de la clase CC_Grupo.....	75
Tabla 53: Descripción de la clase CC_Tipo de curso.....	75
Tabla 54: Descripción de la clase CC_Curso escolar.....	76
Tabla 55: Descripción de la clase CC_Universidad.....	76
Tabla 56: Descripción de la clase CC_Carrera.....	76
Tabla 57: Descripción de la clase CC_Rol.....	77
Tabla 58: Pasos para obtener los casos de pruebas.....	87
Tabla 59: Escenarios del CU.....	88
Tabla 60: Casos de pruebas.....	88
Tabla 61: Casos de prueba con los valores de los datos.....	88
Tabla 62: Escenarios del CU.....	89
Tabla 63: Casos de pruebas.....	89
Tabla 64: Casos de prueba con los valores de los datos.....	89
Tabla 65: Escenarios del CU.....	90
Tabla 66: Casos de pruebas para el CU.....	90
Tabla 67: Casos de prueba con los valores de los datos.....	90
Tabla 68: Escenarios del CU.....	91
Tabla 69: Casos de pruebas para el CU.....	91
Tabla 70: Casos de prueba con los valores de los datos.....	91
Tabla 71: Escenarios del CU.....	92
Tabla 72: Casos de pruebas para el CU.....	92
Tabla 73: Casos de prueba con los valores de los datos.....	93
Tabla 74: Escenarios del CU.....	93
Tabla 75: Casos de pruebas para el CU.....	93
Tabla 76: Casos de prueba con los valores de los datos.....	93
Tabla 77: Escenario del CU.....	94
Tabla 78: Casos de pruebas para el CU.....	94
Tabla 79: Casos de prueba con los valores de los datos.....	94

Tabla 80: Escenario del CU.	94
Tabla 81: Casos de pruebas para el CU.....	94
Tabla 82: Casos de prueba con los valores de los datos.	95

Índices de imágenes.

Figura 1: Diagrama de casos de uso del negocio.....	29
Figura 2: Diagrama de actividades. CU solicitar traslado para otra facultad.	34
Figura 3: Diagrama de actividades. CU solicitar traslado para otro CES.....	35
Figura 4: Diagrama de actividades. CU informar traslado.	36
Figura 5: Diagrama de objetos.....	37
Figura 6: Diagrama de casos de uso del sistema.....	43
Figura 7: Funcionamiento del Modelo Vista Controlador.	58
Figura 8: Diagrama de paquetes del sistema.....	59
Figura 9: Diagrama de clases incluidas en el paquete de administración.	60
Figura 10: Diagrama de clases incluidas en el paquete de solicitudes.	62
Figura 11: Diagrama de clases incluidas en el paquete de estudiantes.	66
Figura 12: Diagrama de clases incluidas en el paquete de reportes.	69
Figura 13: Diagrama de clases incluidas en el paquete de nomencladores.....	72
Figura 14: Diagrama de clase del diseño CU: Autenticar usuario.....	77
Figura 15: Diagrama de clase del diseño CU: Gestionar usuario.	78
Figura 16: Diagrama de clase del diseño CU: Gestionar estudiante interno.	78
Figura 17: Diagrama de clase del diseño CU: Gestionar estudiante externo.	79
Figura 18: Diagrama de clase del diseño CU: Gestionar solicitud de traslado interna.	79
Figura 19: Diagrama de clase del diseño CU: Gestionar solicitud de traslado externa.....	80
Figura 20: Diagrama de clase del diseño CU: Gestionar resolución.	80
Figura 21: Diagrama de clase del diseño CU: Registrar expediente.....	81
Figura 22: Diagrama de clase del diseño CU: Generar reportes.	82
Figura 23: Diagrama Entidad-Relación.....	83
Figura 24: Diagrama de componentes.....	85
Figura 25: Diagrama de despliegue.	86
Figura 26: Diagrama de secuencia CU: Autenticar usuario.....	100
Figura 27: Diagrama de secuencia CU: Gestionar usuario.	101
Figura 28: Diagrama de secuencia CU: Gestionar estudiante interno.	102
Figura 29: Diagrama de secuencia CU: Gestionar estudiante externo.	103
Figura 30: Diagrama de secuencia CU: Gestionar solicitud interna.	104
Figura 31: Diagrama de secuencia CU: Gestionar solicitud externa.	105
Figura 32: Diagrama de secuencia CU: Gestionar resolución.	106
Figura 33: Diagrama de secuencia CU: Registrar expediente.....	107
Figura 34: Diagrama de secuencia CU: Generar reportes.	107

Introducción.

Con el surgimiento de la Universidad de las Ciencias Informáticas se definió como su principal Misión formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática, así como también producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

La UCI cuenta con un plan de estudio diferente al del resto de las universidades del país en los que se estudian carreras similares a esta. Se hace énfasis en la producción como parte del proceso de aprendizaje, por lo que vincula a los estudiantes a múltiples proyectos productivos e investigativos de interés en campos como los de la educación, la salud, el deporte, el gobierno en línea, el software libre, los portales web, y productos multimedia por solo citar algunos. Otras actividades complementan la formación curricular brindando espacios a la investigación y la innovación, así como al desarrollo de habilidades en el uso de la tecnología. Cuenta además con una tecnología informática avanzada para el desarrollo de todos los proyectos.

Al mismo tiempo que se desarrollan las actividades académicas y productivas, la Universidad enfrenta una gigantesca obra constructiva para crear las condiciones de vida y de trabajo necesarias para el desarrollo y ampliación de sus capacidades. Cada año se realiza una rigurosa selección en todo el país entre los interesados en estudiar la Ingeniería en Ciencias Informáticas

La Secretaría General es la responsable de realizar todos los procesos relacionados con los estudiantes desde que estos matriculan hasta que egresan, controlando toda la información asociada a los mismos. Por la diversidad de procesos que se llevan a cabo en esta área se hace necesario el uso de la tecnología de manera que se facilite su desarrollo con más eficiencia y calidad.

Entre los múltiples procesos que realiza el área, está el proceso de gestión de traslados de los estudiantes de la UCI hacia otros Centros de Educación Superior del país, de estos hacia la UCI y el proceso de traslado interno, es decir entre facultades de la propia Universidad.

La UCI cuenta con un Sistema de Gestión Académica, "Akademos", que pone a la disposición de la comunidad universitaria varios módulos desarrollados para mantener informados a los estudiantes sobre su desempeño académico. Posee un registro que controla el desarrollo del proceso docente, calificaciones y asistencia, al cual solo pueden acceder, los profesores y las secretarías docentes de cada facultad, quienes además tienen otros permisos en el sistema acordes a la labor que desempeñan. La versión actual de "Akademos" no permite gestionar el traslado de los estudiantes, solo brinda información sobre aquellos que ya fueron trasladados.

El proceso de gestión de traslados de los estudiantes en la UCI se dificulta en la actualidad porque el mismo se realiza manualmente, empleándose gran cantidad de tiempo en la organización, análisis y búsqueda de la información recogida; por todo esto el proceso es indiscutiblemente ineficiente y muy trabajoso y se requiere por tanto de la informatización del mismo.

Por todo lo planteado anteriormente se asume como **pregunta de la investigación:**

¿Cómo mejorar la eficiencia del proceso de traslado y el análisis de la información asociado a este?

El objeto de estudio:

El sistema automatizado para el proceso de traslado de los estudiantes de la UCI.

El campo de acción:

La gestión de la información del proceso de traslado de los estudiantes en la UCI.

Para dar solución al problema se plantea como **objetivo general:**

Desarrollar una solución de software que gestione el proceso de traslado de los estudiantes de la UCI con mayor eficiencia.

Se plantean los siguientes **objetivos específicos:**

1. Informatizar el proceso de traslado de los estudiantes en la UCI.
2. Determinar las principales reglas que el sistema debe cumplir.

3. Definir los requerimientos del sistema a diseñar.
4. Realizar el diseño del sistema a utilizar.
5. Implementar el sistema.
6. Realizar las pruebas para verificar el correcto funcionamiento del sistema.

Para desarrollar la investigación se plantean las siguientes **preguntas científicas:**

1. ¿Cómo se efectúa el proceso de traslado de los estudiantes en la UCI?
2. ¿Qué información se maneja durante el proceso de traslado?
3. ¿Cuáles son las funcionalidades que debe tener el sistema para controlar la gestión de traslado de los estudiantes en la UCI?

Para dar cumplimiento a los objetivos planteados anteriormente se proponen las siguientes **tareas de la investigación:**

1. Realizar el estudio sobre el proceso de traslado de los estudiantes en la UCI.
2. Modelar el negocio.
3. Realizar el levantamiento de requisitos para el sistema.
4. Realizar el diseño del sistema propuesto.
5. Implementar el sistema.
6. Estimar y realizar las pruebas necesarias para el correcto funcionamiento del sistema.

Para la realización de este trabajo se emplearon los siguientes **métodos científicos e instrumentos de investigación:**

1. Métodos teóricos.

- 1.1. Analítico: Con el objetivo de analizar documentos para extraer elementos importantes relacionados con el objeto de estudio.

1.2. Histórico-Lógico: El método histórico estudia la trayectoria de los fenómenos y acontecimientos en su devenir histórico, mientras que el método lógico investiga las leyes generales del funcionamiento y desarrollo de los fenómenos. Lo lógico no reproduce lo histórico en todos sus detalles, sino que reproduce en el plano teórico lo más importante del fenómeno.

2. Métodos empíricos.

2.1. Entrevistas: A especialistas que laboran en el área de Secretaría General y Secretarías Docentes con el objetivo de recoger la información de interés y comprender a fondo la situación actual del problema a resolver y levantar los requisitos que el sistema debe cumplir.

2.2. Observación: Se creó un registro visual para concentrar y resumir la información plasmándola en una libreta de trabajo.

El presente trabajo se desarrolla en cuatro capítulos, las conclusiones generales, recomendaciones y la bibliografía utilizada durante el desarrollo de la investigación. Cuenta con un glosario de términos y los anexos que son necesarios para una mejor comprensión del trabajo.

Capítulo 1: Fundamentación teórica.

En este capítulo se definen los principales conceptos relacionados con el tema y con los cuales se trabajó. Se describe detalladamente cómo funcionan actualmente los procesos del negocio. Se aborda el tema de las herramientas y metodologías empleadas para dar solución al problema y su utilización en el mundo en general.

Capítulo 2: Modelación del negocio. Características del sistema.

En este capítulo se hace una descripción detallada del negocio, se identifican los actores y trabajadores del negocio, los casos de uso del negocio y su descripción. Se identifican las características que tiene el sistema, se muestran los requisitos con los que debe contar, y se identifican los actores del sistema, los casos de uso y una descripción de los mismos.

Capítulo 3: Diseño del sistema.

En este capítulo se describe el diseño del sistema propuesto, se describe el patrón de diseño que se utiliza, se muestra el diagrama de paquetes en los que se divide la propuesta de desarrollo, presenta los diagramas de clases y las relaciones por paquetes, se muestran también los diagramas de clases del diseño, se muestra el diagrama entidad relación y los diagramas de secuencia por cada CU.

Capítulo 4: Implementación y prueba del sistema.

En este capítulo se muestra el diagrama de componente y se hace una descripción del diagrama de despliegue. Se explica el proceso que se realiza para efectuar las pruebas y probar el correcto funcionamiento de las funcionalidades del sistema.

Capítulo 1 “Fundamentación teórica”.

1.1. Introducción.

En el presente capítulo se realiza una descripción del proceso de traslado de los estudiantes en el país; se presenta el procedimiento seguido por los estudiantes al solicitar el traslado hacia otro centro universitario o internamente en la UCI, así como el procedimiento cuando se solicita el traslado de otro centro universitario para la UCI; se especifican las reglas del negocio para el proceso que se desea automatizar y se plantea una propuesta de solución para resolver los problemas que ha generado dicho proceso. Asimismo se propone el lenguaje, las herramientas necesarias a utilizar y también la tecnología necesaria para implementar dicho sistema.

1.2. Situación actual de los traslados en el país.

El traslado de carrera se otorga excepcionalmente y se concede sólo por una vez cuando existan elementos de peso en el orden social, personal o académico. Se exceptúan de esta disposición, los que obtengan plaza por la vía de examen de concurso y por el Decreto No. 91/81, y los casos de enfermedad, accidentes y otras causas que le impidan continuar estudios en la carrera que cursa y sea aconsejable su traslado a otra por pérdida de requisitos.

Cuando los estudiantes matriculados en la educación superior obtengan plazas mediante el examen de concurso con el propósito de cambiar de carrera, sin haber causado baja en la educación superior, se les considerará traslado.

Los estudiantes que por alguna razón pierdan algunos de los requisitos adicionales, que se exigen en la carrera que cursan, independientemente del año en que estén matriculados, serán

trasladados de carrera. El Rector queda facultado para disponer dichos traslados de carrera, a solicitud del Decano de la Facultad de que se trate.

Los requisitos para considerar la solicitud de traslado de carrera de estudiantes matriculados en cursos regulares diurnos son los siguientes:

Haber aprobado dos años completos en la carrera matriculada.

Tener una destacada trayectoria docente integral.

Cumplir los requisitos adicionales de las carreras que los tengan establecidos.

Cuando exista plena justificación, se podrá considerar la solicitud de traslado de carrera de los estudiantes de cursos regulares diurnos que hayan vencido todas las asignaturas de primer año con índice académico promedio de cuatro o más y una evaluación integral satisfactoria. El Rector evaluará la solicitud en el Consejo de Dirección del Centro, tomando en cuenta las opiniones del Colectivo de Año, de la Federación Estudiantil Universitaria (FEU) y de la Unión de Jóvenes Comunistas (UJC).

La solicitud de traslado se hace al Decano de la Facultad en que se encuentra formalizada la matrícula, especificándole las causas, y debe realizarse dentro de los sesenta días anteriores a la fecha de terminación del curso académico.

Los traslados de carreras enmarcadas en la misma Facultad, siempre que cumplan los requisitos establecidos, se deciden por el Decano tomando en consideración la opinión de la UJC y la FEU, o de la sección sindical en el caso del "Curso para trabajadores". Los trámites se realizan por la Secretaría Docente de la Facultad. Los traslados entre carreras de facultades diferentes del mismo Centro de Educación Superior, se analizan y deciden por ambos Decanos, tomando en consideración las referidas opiniones. En casos de discrepancias, informarán sus opiniones al Rector, el que decidirá al respecto.

El traslado de un centro de educación superior a otro, será analizado por el Rector del centro donde se encuentra formalmente matriculado el estudiante (centro de origen) y se remitirá al Rector del centro para el que se solicita el traslado (centro receptor). El Rector del centro receptor decidirá si lo acepta o

no, según sus posibilidades y tomando en consideración las opiniones de la organizaciones juveniles o sindicales, según corresponda. Toda la tramitación y remisión de los documentos se hará por la Secretaría General del centro de origen, para lo que dispondrá de 20 días hábiles. El centro receptor deberá dar respuesta al interesado en un plazo no mayor de 20 días hábiles, a partir de la fecha en que recibió la documentación correspondiente.

Los traslados se autorizarán una vez concluido el curso académico. El Rector queda facultado para autorizar traslados entre centros fuera de dicho período y en el transcurso del calendario académico conforme al procedimiento establecido, cuando existan causas de extrema justificación.

No se aprobará el traslado si:

En el expediente académico del estudiante no aparecen reflejadas las calificaciones correspondientes a todas las asignaturas por él evaluadas en la carrera que cursa.

Al concluir el segundo semestre del curso académico, el estudiante tuviese asignaturas que debe evaluar en los exámenes extraordinarios del mes de agosto. El traslado sólo se tramitará cuando haya realizado dichas evaluaciones y estén anotadas en su expediente académico las calificaciones obtenidas.

El estudiante obtuviere resultados negativos en los exámenes extraordinarios del mes de agosto salvo que, por excepción se le concediere previamente el derecho a repetir el año académico.

Es responsabilidad del Decano de la facultad para la cual se aprobó el traslado una vez analizado el expediente del estudiante, decidir el año académico y las asignaturas que le corresponde matricular según el plan de estudio vigente, así como establecer un ajuste del plan de estudio, si lo requiere el caso.

No se autorizará traslado hacia carreras de Ciencias Médicas a estudiantes de otras carreras. El Decano podrá hacer excepción en aquellos casos de estudiantes con índice igual o superior a cuatro, cuya actitud social sea avalada por el Consejo de Dirección de la Facultad.

1.3. Procedimiento de solicitud de traslado para otro CES en la UCI.

El proceso comienza en el mes de abril; en ese momento el estudiante redacta la solicitud de traslado por escrito en una carta dirigida al Decano, explicando los motivos por los cuales realiza la solicitud, especificando tanto la carrera como el centro universitario de destino. La carta es entregada en la SD de su facultad.

La Secretaria Docente verifica que el estudiante cumpla los requisitos establecidos, de no ser así le informa al estudiante las razones por las que su solicitud no puede ser atendida. De cumplirlos se le hace llegar la carta al Decano de la facultad para que este en un Consejo de Dirección, decida sobre su aprobación. En caso de que se decida no aprobar la solicitud, se le informa al estudiante las razones consideradas para ello.

Las diferentes solicitudes de traslado aprobadas a nivel de facultad se entregan en la SG de la Universidad. La información recogida en los documentos de solicitudes se refleja en un documento de Excel después de ser revisados para verificar que los estudiantes cumplen con los requisitos establecidos.

Procesada toda la información y organizada por años académicos se procede a presentar las solicitudes a la comisión creada por el Rector la cual las analiza y oído el parecer de las organizaciones juveniles y del Decano de la facultad en cuestión las propone al Rector para que decida sobre su aprobación.

Aquellas solicitudes que resultaran aprobadas se envían a la SG del centro receptor donde se aprueba finalmente el traslado de considerarse pertinente por parte de las autoridades docentes del centro. De no ser aprobado el traslado ya sea por el centro de origen o por el centro receptor se le comunica al estudiante la decisión explicándole las razones consideradas al efecto.

Aprobado el traslado en el centro de origen se procede a la confección de la resolución del Rector que lo hace firme y finalizado el curso escolar se hace efectivo siempre que el estudiante no haya perdido alguno de los requisitos establecidos. De ocurrir esto, o bien se retiene el traslado hasta tanto concluyan los exámenes extraordinarios de fin de curso o se cancela el mismo si el estudiante siendo de 1er año, no alcanzara los cuatro puntos de promedio requeridos.

1.4. Proceso de solicitud de traslado para la UCI.

Cuando un estudiante matriculado en otro Centro de Educación Superior quiere solicitar el traslado para la UCI el procedimiento a seguir es muy similar al explicado en el epígrafe 1.3. La diferencia radica esencialmente en que el estudiante en cuestión es objeto de investigaciones más profundas por parte de la UCI.

1.5. Procedimiento para los traslados internos de la UCI.

El procedimiento a seguir es muy similar al explicado con anterioridad; las diferencias fundamentales vienen dadas porque las solicitudes no son analizadas por el Rector sino por cada uno de los Decanos de las facultades de origen y de destino quienes por Resolución lo hacen firme.

De estar de acuerdo ambos Decanos, la Secretaria Docente de la facultad de origen informa a la Secretaria General para que proceda al cambio de facultad en el sistema de Gestión Académica, lo refleje en los controles pertinentes y lo informe a las Direcciones de Residencia involucradas.

1.6. Reglas del negocio.

Durante todos los procesos explicados anteriormente se concluye con las reglas que se deben seguir para efectuar el traslado y la elaboración de la aplicación a desarrollar, siempre respetando las diferentes restricciones que estas implican:

- a) Solo se concederá el traslado por una vez.
- b) Solo se procederá al traslado de carrera si el estudiante cumple con los requisitos de la misma.
- c) Tener aprobado el segundo año de la carrera.
- d) Tener un índice académico igual o mayor que cuatro puntos de estar matriculado en primer año.
- e) Los traslados se efectuarán una vez terminado el curso.
- f) Las calificaciones correspondientes a todas las asignaturas ya evaluadas deberán reflejarse en el expediente.
- g) El estudiante no podrá tener asignaturas pendientes.
- h) Solo se recibirán solicitudes de traslado aprobadas por el Decano de la facultad en cuestión.
- i) Solamente obtendrán información de la aplicación los administradores y trabajadores que laboren en ella.
- j) Solo el personal de las SD y de la SG podrán modificar los datos del sistema.
- k) Toda solicitud tendrá un número que la identifique.
- l) La solicitud deberá reflejar las causas por las que fue efectuada.
- m) Los estudiantes a los que se les aprueba el traslado tienen que entregar todas sus pertenencias antes de abandonar la Universidad al finalizar el curso (modelo de tramitación de baja).

1.7. Propuesta de solución.

Este trabajo describe el desarrollo de una aplicación web que facilita a la SG en la UCI gestionar el proceso de traslado en cada curso escolar.

El sistema que se diseñe debe permitir al personal de las secretarías llevar un mejor control de los traslados efectuados en un curso escolar; adicionalmente debe permitir:

- ✓ Mostrar individualmente cualquier información que les sea necesaria y que exista en la base de datos a partir de introducir cualquiera de estos como elemento de búsqueda.
- ✓ Conocer en qué momento del proceso se encuentra una solicitud.
- ✓ Obtener cifras y estadísticas de los estudiantes que realizan traslados.
- ✓ Conocer la relación nominal de los traslados efectuados por Universidad de destino.
- ✓ Mostrar las solicitudes aprobadas y los traslados por tipo de curso.

Para dar cumplimiento a la propuesta de solución planteada en esta investigación se desarrollará una aplicación Web que permitirá que el sistema pueda ser utilizado desde distintos lugares y sin más requerimientos que una computadora con navegador Web y conexión a la red. Se usará PHP como lenguaje de programación, PostgreSQL como base de datos y Apache como servidor de aplicación Web, todos estos por las potencialidades que ofrecen. Para planificar y controlar la propuesta planteada, y con las características y facilidades que aporta a todo el proceso, se empleará la metodología RUP, usando como herramienta case el Visual Paradigm y que utiliza el lenguaje de modelado UML.

1.8. Tecnologías utilizadas para el desarrollo del sistema.

1.8.1. Arquitectura Cliente\Servidor.

Se utilizará una arquitectura Cliente/Servidor en el sistema a desarrollar con el objetivo de que las informaciones procesadas se puedan dividir en procesos independientes que cooperen entre sí para poder intercambiar informaciones, servicios o recursos.

Este modelo permite dividir el servidor de forma que contenga la parte que debe ser compartida por varios usuarios, y en el cliente permanezca sólo lo particular de cada usuario.

Cuando el cliente necesite un servicio lo solicita al servidor correspondiente y éste le responderá proporcionándole la información que solicita. Pero no necesariamente el cliente y el servidor están

ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

1.8.2. Frameworks.

El framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto, además provee una estructura y una metodología de trabajo; son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Framework es un concepto sumamente genérico, se refiere a “ambiente de trabajo, y ejecución”. En general los frameworks son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

Fuera de las aplicaciones en la informática, un framework puede ser considerado como el conjunto de procesos y tecnologías usados para resolver un problema complejo. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada.

1.8.2.1. CodeIgniter.

CodeIgniter (CI) está registrado bajo Apache/BSD-estilo, la licencia de la fuente abierta por EllisLab Inc, permite copiar, modificar y distribuir el software y su documentación, con o sin modificación, para cualquier propósito con tal de que se cumplan las condiciones establecidas.

Desarrollado originalmente por Rick Ellis, el armazón fue escrito para la actuación en el mundo real con muchas de las bibliotecas incorporadas, que los auxiliares, y subalterno-sistemas pidieron prestado

del código-base de ExpressionEngine. Se desarrolla actualmente y responde al mantenimiento del equipo de desarrollo de ExpressionEngine.

Es una herramienta para hacer más fácil el uso de PHP, mantiene una biblioteca amplia y la estructura lógica para acceder a la misma, tiene una interfaz simple. Mediante el CI se minimiza la cantidad de líneas de código en el desarrollo de una aplicación, lo que permite que cargue más rápido y que consuma menos espacio en memoria. Es perfectamente compatible con todas las versiones de PHP y hasta su última versión y en caso de interés de la migración de una versión a otra han sido específicos en los mínimos cambios a realizarle a la configuración de las aplicaciones para compatibilizarla con la nueva versión.

Es de peso ligero. El sistema requiere sólo unas bibliotecas muy pequeñas. Esto está en contraste severo con muchos armazones que requieren significativamente más recursos. Las bibliotecas adicionales están dinámicamente cargadas en la demanda, basado en sus necesidades para un proceso dado.

CodeIgniter se basa en el modelo de desarrollo Modelo-Vista-Controlador (MVC) es un acercamiento del software que separa la lógica de la aplicación y de la presentación. En la práctica, permite a sus páginas web contener el scripting mínimo desde que la presentación está separada del scripting de PHP.

CodeIgniter permite acceder a cualquier base de datos ya sea PostgreSQL, MySQL (4.1+), MySQLi, MS SQL, Oracle, SQLite, and ODBC, permite el envío de correo electrónico, validación de los datos de un formulario, mantenimiento de sesiones, manipulación de imágenes, que trabaja con los datos de XML-RPC y mucho más. Es extensible, el sistema puede extenderse fácilmente a través del uso de plugins y bibliotecas del auxiliador, o a través de extensiones de la clase o ganchos del sistema. CI también permite poner el URL del sitio en un archivo de la configuración pudiendo acceder al resto del sitio. Cuando se desee mover el sitio, sólo se necesita cambiar el archivo de la configuración, y todos sus hipervínculo se ponen al día automáticamente. Se utilizó el CodeIgniter versión 1.7.1.

1.8.2.2. Ext JS.

El framework Ext JS implementado en javascript, se comenzó a desarrollar a principios del 2006 por Jack Slocum como un conjunto de extensiones para la librería Yahoo User Interface (YUI) enfocado a componentes o widgets.

Ext JS es una librería de código abierto para la creación de interfaces usuarios en el desarrollo de aplicaciones Web. Para ello ofrece una serie de librerías (compatibles con PHP 4 y 5) para integrar Ext JS en nuestro sistema. Es de peso ligero ya que contiene un conjunto reducido de bibliotecas con un alto rendimiento ofrecidas en un archivo con un tamaño de 25 KB portable. Funciona como un mapeado en clases de la librería JS. Entre las posibilidades que ofrece nos encontramos con la creación de formularios, combos, grids o menús. A parte ayuda a la comunicación entre el cliente y el servidor.

Ext JS es compatible con varios navegadores como Internet explorer 6+, Firefox 1.5+, Safari 3+ y Ópera 9+. También tiene varias opciones disponibles para la autorización con una variedad de necesidades que incluyen fuente abierta y licencias del anuncio. La versión utilizada de Ext JS fue la 2.2.

1.8.3. Javascript.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar con Javascript se tienen dos vertientes.

Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.

Por el otro, javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

1.8.4. PHP.

PHP (acrónimo de Hypertext Preprocessor) es un lenguaje del lado del servidor, es decir PHP funciona en un servidor remoto que procesa la página Web antes de que sea abierta por el navegador del usuario, especialmente creado para el desarrollo de páginas Web dinámicas. Puede ser incluido con facilidad dentro del código HTML, y permite una serie de funcionalidades extraordinarias. Para la implementación del sistema se utilizó PHP 5.

Este lenguaje se caracteriza por:

Costo: PHP es gratuito, puede descargarse y utilizarse en cualquier aplicación, personal o profesional cuando se desee sin costo alguno.

Rendimiento: PHP es muy eficiente. Con escaso mantenimiento y mediante el uso de un único servidor gratuito (en nuestro caso, Apache), puede soportar sin problema millones de visitas diarias.

Integración de base de datos: PHP dispone de una conexión propia a todos los sistemas de base de datos. Además de PostgreSQL; que es el SGBD que se utilizará en la implementación del sistema, puede conectarse directamente a las bases de datos de MySQL, mSQL, Oracle, dbm, filepro, Hyperwave, Informix, InterBase y Sybase, entre otras.

Bibliotecas incorporadas: Como se ha diseñado para el desarrollo de páginas Web dinámicas, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.

Portabilidad: PHP está disponible para una gran cantidad de sistemas operativos diferentes. Puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix, como Linux y FreeBSD, versiones comerciales de Unix, como Solaris e IRIX o en las diferentes versiones de Microsoft Windows. Su código funcionará sin necesidad de aplicar modificación alguna a los diferentes sistemas que ejecuten PHP.

Código fuente: Al tratarse de código abierto posibilita el acceso al código fuente de PHP. A diferencia de los productos comerciales y de código cerrado, si se desea modificar algo o agregar un elemento al programa, se puede hacer con total libertad.

1.8.5. PostgreSQL.

PostgreSQL es un Sistema Gestor de Base de Datos (SGBD) relacional orientado a objetos de software libre, liberado bajo la licencia BSD. El desarrollo de PostgreSQL no es manejado por una sola compañía, como otros proyectos Open Source, es dirigido y desarrollado por la comunidad PGDG (PostgreSQL Global Development Group).

La licencia BSD otorgada principalmente para los sistemas BSD (Berkeley Software Distribution, Distribución de Software Berkeley). Esta licencia pertenece al grupo de licencias de software libre y tiene

menos restricciones en comparación con otras como la GPL, estando muy cercana al dominio público. La licencia BSD al contrario de la GPL permite el uso del código fuente en software no libre.

PostgreSQL está caracterizado por su alta concurrencia, pues mediante su sistema denominado Acceso Concurrente Multi Versión (MVCC) permite el acceso a las tablas que están bajo un proceso que se está ejecutando en ellas sin necesidad de bloqueos. Por otra parte posee una amplia variedad de tipos nativos, entre otros son los casos de:

1. Texto de largo ilimitado.
2. Números de precisión arbitraria.
3. Figuras geométricas.
4. Direcciones IP.
5. Arrays.

Integra otras características que son generales en cada uno de los distintos SGBD como:

1. Llaves ajenas o Llaves Foráneas (foreign keys).
2. Disparadores (triggers).
3. Vistas.
4. Integridad transaccional.
5. Herencia de tablas.
6. Tipos de datos y operaciones geométricas.

Los bloques de código que se ejecutan en el servidor (funciones), pueden ser escritos en varios lenguajes, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional. Se utilizó PostgreSQL versión 8.3.4 que utiliza SQL 4.2.0.1

1.8.6. Servidor Web Apache.

El servidor Apache es el complemento perfecto para las páginas dinámicas desarrolladas con PHP y PostgreSQL. Comparte con éstos muchas de sus características, como son la gratuidad (también se trata de *software* libre), su popularidad, sencillez de manejo y versatilidad, ya que puede ser instalado sobre *Unix* o sobre *Windows*, por lo que se recomienda y se utiliza en dicho trabajo. Su sencillez de manejo lo hace ideal para instalarlo en un ordenador para realizar todo tipo de pruebas y ejercicios. Se contó con el AppServ versión 2.5.10.

1.8.7. Metodologías de Desarrollo de Software.

Todo desarrollo de software es riesgoso y difícil de controlar, en este punto es donde se hace necesario el uso de una metodología de software, la cual plantea un conjunto de pasos y procedimientos que deben seguirse para desarrollar un producto. Puede seguir uno o varios modelos de ciclo de vida los cuales indican qué es lo que hay que obtener a lo largo del desarrollo del proyecto. Es también objetivo de la metodología elevar la calidad del producto final trabajando en función de esta en cada una de las fases que define.

1.8.7.1. RUP.

El Proceso Unificado de Rational (RUP, en inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El Proceso Unificado es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, tipos de organización, niveles de aptitud y tamaños de proyecto.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Es un proceso que define claramente quién, cuándo, cómo y qué debe hacerse, y como su enfoque está basado en modelos, utiliza un lenguaje bien definido para tal fin, el UML.

RUP posee tres características esenciales está dirigido por Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden debe desarrollarse, y es iterativo e incremental: divide el proyecto en mini proyectos y los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

El desarrollo del software se divide en 4 fases fundamentales:

- ✓ **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- ✓ **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- ✓ **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- ✓ **Transición:** El objetivo es llegar a obtener el **release** del proyecto.

El ciclo de vida que se desarrolla por cada iteración es llevada bajo dos disciplinas:

Disciplina de Desarrollo o Flujo de Trabajo

- ✓ Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- ✓ Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- ✓ Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- ✓ Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- ✓ Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

1. Configuración y administración del cambio: Guardando todas las versiones del proyecto.

2. Administrando el proyecto: Administrando horarios y recursos.
3. Ambiente: Administrando el ambiente de desarrollo.
4. Distribución: Hacer todo lo necesario para la salida del proyecto

Los elementos del RUP son:

- ✓ Actividades: Son los procesos que se llegan a determinar en cada iteración.
- ✓ Trabajadores: Son las personas o entes involucrados en cada proceso.
- ✓ Artefactos: Pueden ser documentos, modelos, o elementos de modelo.

Una particularidad de esta metodología es que en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo una de las más importantes para alcanzar un grado de certificación en el desarrollo del software.

1.8.7.2. Microsoft Solution Framework (MSF).

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF proporciona prácticas comprobadas para planificar, crear y poner en marcha soluciones exitosas. En oposición a la metodología prescriptiva, MSF proporciona una estructura flexible y escalable para conocer las necesidades de una organización o equipo encargado de un proyecto de cualquier tamaño. La orientación de MSF consiste en proporcionar principios, modelos y disciplinas para manejar personas, procesos y elementos de tecnología con los que se encuentran la mayoría de los proyectos.

Características de MSF.

- ✓ **Adaptable:** es parecida a un compás, utilizado en cualquier parte como un mapa, del cual su uso es limitado a un lugar específico.
- ✓ **Escalable:** puede organizar equipos pequeños (entre 3 ó 4 personas), así como también, proyectos que requieran 50 personas o más.

- ✓ **Flexible:** es utilizado en el ambiente de desarrollo de cualquier cliente.
- ✓ **Tecnología Agnóstica:** porque puede ser utilizada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el Modelo de Aplicación.

- ✓ **Modelo de Arquitectura del Proyecto:** Diseñado para acortar la planificación del ciclo de vida. Este modelo define las pautas para construir proyectos empresariales a través del lanzamiento de versiones.
- ✓ **Modelo de Equipo:** Este modelo ha sido diseñado para mejorar el rendimiento del equipo de desarrollo. Proporciona una estructura flexible para organizar los equipos de un proyecto. Puede ser escalado dependiendo del tamaño del proyecto y del equipo de personas disponibles.
- ✓ **Modelo de Proceso:** Diseñado para mejorar el control del proyecto, minimizando el riesgo, y aumentar la calidad acortando el tiempo de entrega. Proporciona una estructura de pautas a seguir en el ciclo de vida del proyecto describiendo las fases, las actividades, la liberación de versiones y explicando su relación con el Modelo de Equipo.
- ✓ **Modelo de Gestión del Riesgo:** Diseñado para ayudar al equipo a identificar las prioridades, tomar las decisiones estratégicas correctas y controlar las emergencias que puedan surgir. Este modelo proporciona un entorno estructurado para la toma de decisiones y acciones valorando los riesgos que puedan provocar.
- ✓ **Modelo de Diseño del Proceso:** Diseñado para distinguir entre los objetivos empresariales y las necesidades del usuario. Proporciona un modelo centrado en el usuario para obtener un diseño eficiente y flexible a través de un enfoque iterativo. Las fases de diseño conceptual, lógico y físico proveen tres perspectivas diferentes para los tres tipos de roles: los usuarios, el equipo y los desarrolladores.
- ✓ **Modelo de Aplicación:** Diseñado para mejorar el desarrollo, el mantenimiento y el soporte, proporciona un modelo de tres niveles para diseñar y desarrollar aplicaciones software. Los

servicios utilizados en este modelo son escalables, y pueden ser utilizados en un solo ordenador o incluso en varios servidores.

1.8.7.3. Extreme Programming (XP).

Esta metodología se basa en la idea de que existen cuatro variables que guían el desarrollo de los sistemas: Costo, Tiempo, Calidad y Alcance. La manera de encarar los desarrollos avalados por este modelo de desarrollo es permitir a las fuerzas externas (gerencia, clientes) manejar hasta tres de estas variables, quedando el control de la restante en manos del equipo de desarrollo. El modelo hace visibles de manera más o menos continua estas cuatro variables.

Características de XP.

No es posible prever todo antes de comenzar a programar y si lo fuera resultaría demasiado costoso e innecesario, ya que muchas veces se emplean demasiado tiempo y recursos en cambiar la documentación de la planificación para que se parezca al código. Para evitar esto, XP intenta implementar una forma de trabajo que se adapte fácilmente a las circunstancias.

Básicamente consiste en trabajar estrechamente con el cliente, haciendo pequeñas iteraciones (mini-entregas) cada dos semanas, y en las que no existe más documentación que el código en sí; cada versión contiene las modificaciones necesarias según el cliente vaya retroalimentando el sistema.

Para suplir la falta de requisitos, casos de uso y demás herramientas, XP utiliza historias de usuarios; la historia de usuario es una frase corta que representa alguna función que realizará el sistema. Cada historia de usuario no puede demorar en desarrollarse más de una semana, si así lo requiriera, debe segmentarse.

Es requisito para XP definir un estándar en el tipo de codificación, esto hace que los programadores tengan definido ya el estilo de programación y no que cada uno programe a su estilo. El testing en cada

iteración es muy importante; de eso se trata en este paradigma de programación, corregir mientras se programa. De esta forma se van supliendo todas las deficiencias que en cada versión exista.

Cada dos semanas se entrega una versión al cliente, quien lo verifica, realiza el feedback y se continúa el desarrollo; este ciclo continúa hasta que el sistema cumpla con las expectativas del cliente, acto que concluirá el proyecto.

No existe documentación del proyecto, lo que más se acerca a la documentación son las historias de usuario, pero al concluir el proyecto se descartan. Inclusive se recomienda hacer dos secciones, una con todas las historias de usuario que faltan por desarrollar, y otra donde se archiven las concluidas, esto permite tener una idea del estado de avance del proyecto.

1.8.8. ¿Por qué RUP?

Una vez estudiada las metodologías anteriores, se decide escoger RUP atendiendo a sus características y el alcance de la biblioteca que utiliza para el desarrollo del sistema. Estas características vienen dadas porque aporta todos los elementos necesarios para el desarrollo de aplicaciones que requieren de una documentación, XP le da poca importancia al desarrollo de la documentación y al lenguaje de modelado y MSF le resta importancia a la selección de la tecnología a utilizar en el presente trabajo debido a que está destinado a realizarse en software libre.

1.8.9. UML.

El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado de propósito general, permite el modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para utilizarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia basada en técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye

conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos. El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos.

1.8.10. Visual Paradigm.

Visual Paradigm para UML es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

El VP-UML es de gran alcance. Proporciona a los desarrolladores del software una plataforma de desarrollo para construir aplicaciones de calidad más rápidas, mejores y más baratas. Facilita una interoperabilidad excelente con otras herramientas CASE.

Esta herramienta da soporte al modelado visual que permite UML ofreciendo un entorno de creación de diagramas. Permite el diseño centrado en casos de uso y enfocado al negocio posibilitando la generación de un software de gran calidad. Posee un lenguaje estándar y común para todo el equipo de

desarrollo que hace posible una mejor comunicación. Proporciona la ventaja de realizar ingeniería directa e inversa. El modelo y el código permanecen sincronizados en todo el ciclo de desarrollo.

El Visual Paradigm puede construir diferentes tipos de diagramas que permiten ver el sistema desde diferentes perspectivas, entre ellos se encuentran los diagramas de casos de uso, los diagramas de clase, actividad, estado, componentes, secuencia, entre otros. Otra de sus ventajas es que permite hacer paquetes de trabajo que proveen de un mecanismo de organización de los modelos agrupando elementos de modelado, siendo esto de gran ayuda para desarrollar sistemas de gran envergadura y complejidad.

1.8.11. Zend Studio.

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones web, en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Sus versiones se pueden utilizar en diferentes sistemas operativos como; Windows, Linux y MacOS. Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

Contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento ofreciendo nombres de las funciones y parámetros que deben recibir. Esta ayuda contextual no se limita a las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que van creándose, incluso en páginas que estén incluidas con la función include.

1.9. Conclusiones parciales.

El presente capítulo recogió el resultado del análisis de los procesos fundamentales relacionados con el trabajo. Se hace una descripción de la tecnología que se utiliza para el desarrollo del sistema, se concluye que la metodología utilizada para el desarrollo del sistema sea RUP; UML como lenguaje de modelado; Visual Paradigm como herramienta case; PHP y javascript como lenguajes de programación con el apoyo de los frameworks CodeIgniter y Ext JS y PostgreSQL como gestor de base de datos.

Capítulo 2 “Modelación del negocio y características del sistema propuesto”.

2.1. Introducción.

En el presente capítulo se hace una descripción detallada de algunos procesos que se desarrollan en el área de Secretaría General de la Universidad para efectuar el traslado de los estudiantes que lo solicitan. Se lleva a cabo un análisis del negocio que se plantea en cada uno de los procesos de solicitud de traslado con el objetivo de lograr un sistema que controle todos los temas a desarrollar; también serán vistos los actores y trabajadores involucrados en dichos procesos.

Se presentan también las características fundamentales que debe tener el sistema propuesto, todas las ideas que los clientes, usuarios y miembros del equipo de trabajo tengan acerca de lo que debe hacer el sistema son analizadas y previstas como requisitos.

2.2. Actores, trabajadores y procesos del negocio.

2.2.1. Actores del negocio.

Actores del negocio	Justificación
Solicitante de traslado.	Es el individuo que acude a Secretaría para solicitar el traslado. Este rol pueden ocuparlo estudiantes y trabajadores matriculados en el Curso para Trabajadores del centro.
Secretario General de otro CES.	Es el individuo que pertenece a otro CES y que informa a la Secretaría General de la UCI las solicitudes de traslados que se han efectuado para la misma.

Tabla 1: Descripción de los actores del negocio.

2.2.2. Trabajadores del negocio.

Trabajadores del negocio	Justificación
Decano	Autoridad encargada de aprobar o denegar el traslado.
Secretaria Docente	Autoridad encargada de recibir la solicitud de traslado por escrito, procesarla y presentarla al Decano para su análisis.
Secretaria General	Autoridad encargada de procesar los datos de la solicitud de traslado y presentarla a la comisión de la dirección de la UCI para su aprobación final.
Rector	Autoridad encargada de aprobar el traslado en la Universidad de origen y destino.

Tabla 2: Descripción de los trabajadores del negocio.

2.2.3. Diagrama de Casos de Uso del Negocio.

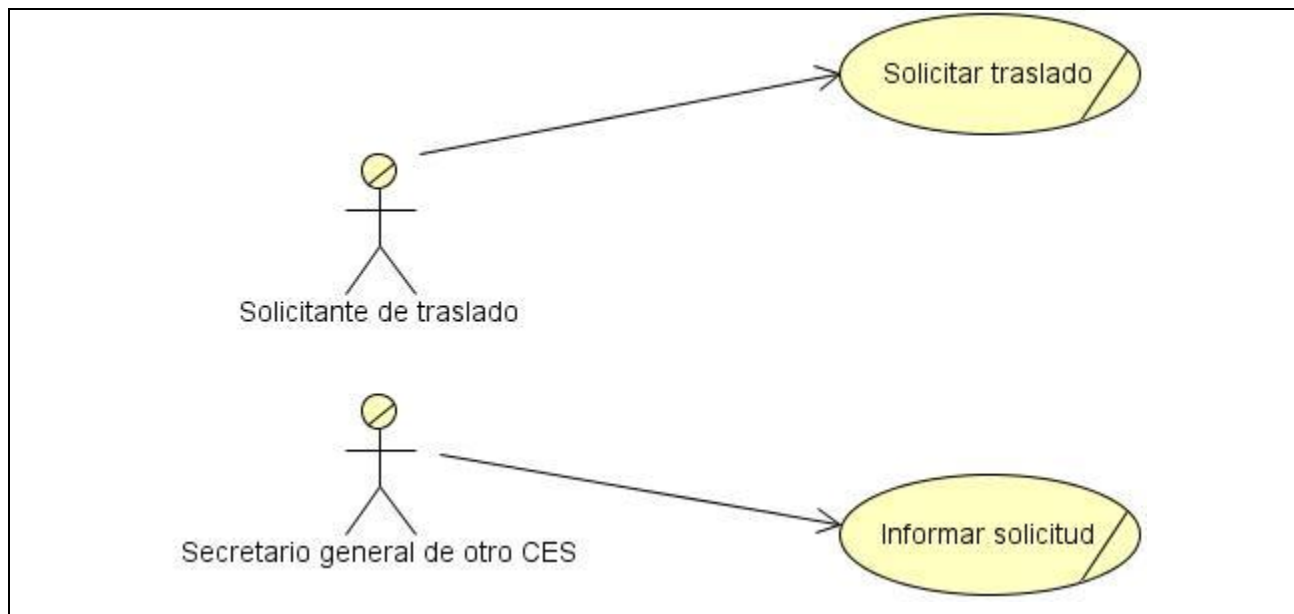


Figura 1: Diagrama de casos de uso del negocio.

2.2.4. Descripción de los Casos de uso del negocio.

Caso de Uso:	Solicitar traslado.	
Actores:	Solicitante del traslado.	
Trabajadores:	Secretaria General, Rector, Secretaria Docente y Decano.	
Resumen:	El caso de uso se inicia cuando el solicitante del traslado se dirige a Secretaría Docente de la Facultad y presenta la solicitud por escrito; este puede ser: solicitud de traslado para otra facultad de la UCI o solicitud de traslado para otro CES.	
Flujo Normal de Eventos.		
Sección: Solicitar traslado para otra Facultad.		
Acción del Actor	Respuesta del Negocio	
1. El solicitante del traslado se dirige al área de Secretaría Docente y presenta la solicitud de traslado por escrito.	2. La Secretaria Docente recibe la solicitud	
	3. La Secretaria Docente identifica el tipo de solicitud y refleja los datos en un documento creado al efecto (en Excel).	
	4. La Secretaria Docente presenta la solicitud al Decano.	
	5. El Decano aprueba la solicitud. Si no ver flujo alternativo 1.	
	6. La Secretaria Docente informa la aprobación del traslado a la homóloga de la facultad para la cual el estudiante lo solicita.	
	7. La Secretaria Docente de la facultad para la cual el solicitante pide traslado confecciona un resumen de la solicitud recibida.	
	8. La Secretaria Docente de la facultad para la cual el solicitante pide traslado informa al Decano de esa Facultad.	

Capítulo 2: Modelación del negocio y características del sistema

	9. El Decano de la facultad de destino aprueba solicitud. Si no ver flujo alterno 2.
	10. La Secretaria Docente informa a la homóloga de la facultad de la cual proviene el estudiante y a la Secretaria General.
	11. La Secretaria General realiza el cambio en el sistema de gestión académica y lo refleja en un documento creado al efecto (Excel).
	12. La Secretaria Docente confecciona la Resolución de traslado y la pone a la firma del Decano.
	13. La Secretaria Docente de la facultad de origen prepara la documentación necesaria para efectuar el traslado y entrega el expediente. Termina el caso de uso.
Flujos Alterno	
1.	El Decano no aprueba la solicitud. Se informa al solicitante
2.	El Decano de la facultad de destino no aprueba la solicitud. Se informa al solicitante.
Sección: Solicitar traslado para otro CES.	
Flujo Normal de Eventos.	
Acción del actor	Respuesta del negocio
1. El solicitante del traslado se dirige al área de Secretaría Docente y presenta la solicitud de traslado por escrito.	2. La Secretaria Docente recibe la solicitud.
	3. La Secretaria Docente identifica el tipo de solicitud y refleja los datos en un documento creado al efecto (en Excel).
	4. La Secretaria Docente presenta la solicitud al Decano.
	5. El Decano aprueba la solicitud. Si no ver flujo alterno 1
	6. La Secretaria Docente informa a la Secretaria General de la solicitud.
	7. La Secretaria General confecciona un resumen de

Capítulo 2: Modelación del negocio y características del sistema

	la solicitud, y recoge los datos en un Excel.
	8. La Secretaria General presenta la solicitud a la comisión conformada para estos efectos en la Dirección de la Universidad.
	9. La comisión aprueba la solicitud. Si no ver flujo alterno 2.
	10. La Secretaria General pone a la firma del Rector la Resolución de traslado.
	11. La Secretaria General informa a la Secretaria Docente de la facultad del estudiante la aprobación del traslado.
	12. La Secretaria Docente informa al estudiante la aprobación del traslado.
	13. La Secretaria General de la UCI informa al Secretario General del CES de destino la solicitud del estudiante. Si no ver flujo alterno 3. Termina el caso de uso.
Flujos Alternos	
1.	El Decano no aprueba la solicitud y se le informa al solicitante.
2.	La comisión de la Dirección de la Universidad no aprueba la solicitud. La Secretaria General informa a la Secretaria Docente de la facultad del estudiante. La Secretaria Docente informa al estudiante que no fue aceptada su solicitud.
3.	El Secretario General del CES de destino informa a la Secretaria General de la UCI que el Rector no aceptó la solicitud. La Secretaria General de la UCI informa a la Secretaria Docente de la Facultad del estudiante que no fue aceptado el traslado. La Secretaria Docente informa al estudiante que no fue aceptado el traslado.

Tabla 3: CU Solicitud de traslado.

Capítulo 2: Modelación del negocio y características del sistema

Caso de Uso:	Informar solicitud.	
Actores:	Secretario General de otro CES.	
Trabajadores:	Secretaria General, Rector.	
Resumen:	El caso de uso se inicia cuando el Secretario General de otro CES informa a Secretaría General la solicitud efectuada para la UCI, comenzando el proceso de análisis para su aprobación.	
Flujo Normal de Eventos.		
Acción del Actor	Respuesta del Negocio	
1. Informa a la Secretaria General la solicitud efectuada para la UCI.	2. La Secretaria General confecciona un resumen de la nueva solicitud, recogiendo los datos necesarios en un Excel.	
	3. La Secretaria General presenta la solicitud a la comisión conformada para esos efectos en la Dirección de la UCI.	
	4. La comisión aprueba dicha solicitud. Si no ver flujo alterno 1.	
	5. La Secretaria General realiza un resumen con los datos de la solicitud aceptada y prepara todo para matricular al estudiante.	
	6. La Secretaria General informa la aceptación del traslado al Secretario General del otro CES. Termina el caso de uso.	
Flujo Alterno		
1. La comisión no aprueba la solicitud. La Secretaria General de la UCI informa al homólogo del CES de procedencia la no aceptación de la solicitud de traslado recibida. Termina el caso de uso.		

Tabla 4: CU Informar solicitud.

2.2.5. Diagrama de Actividades.

2.2.5.1. Sección solicitar traslados para otra Facultad.

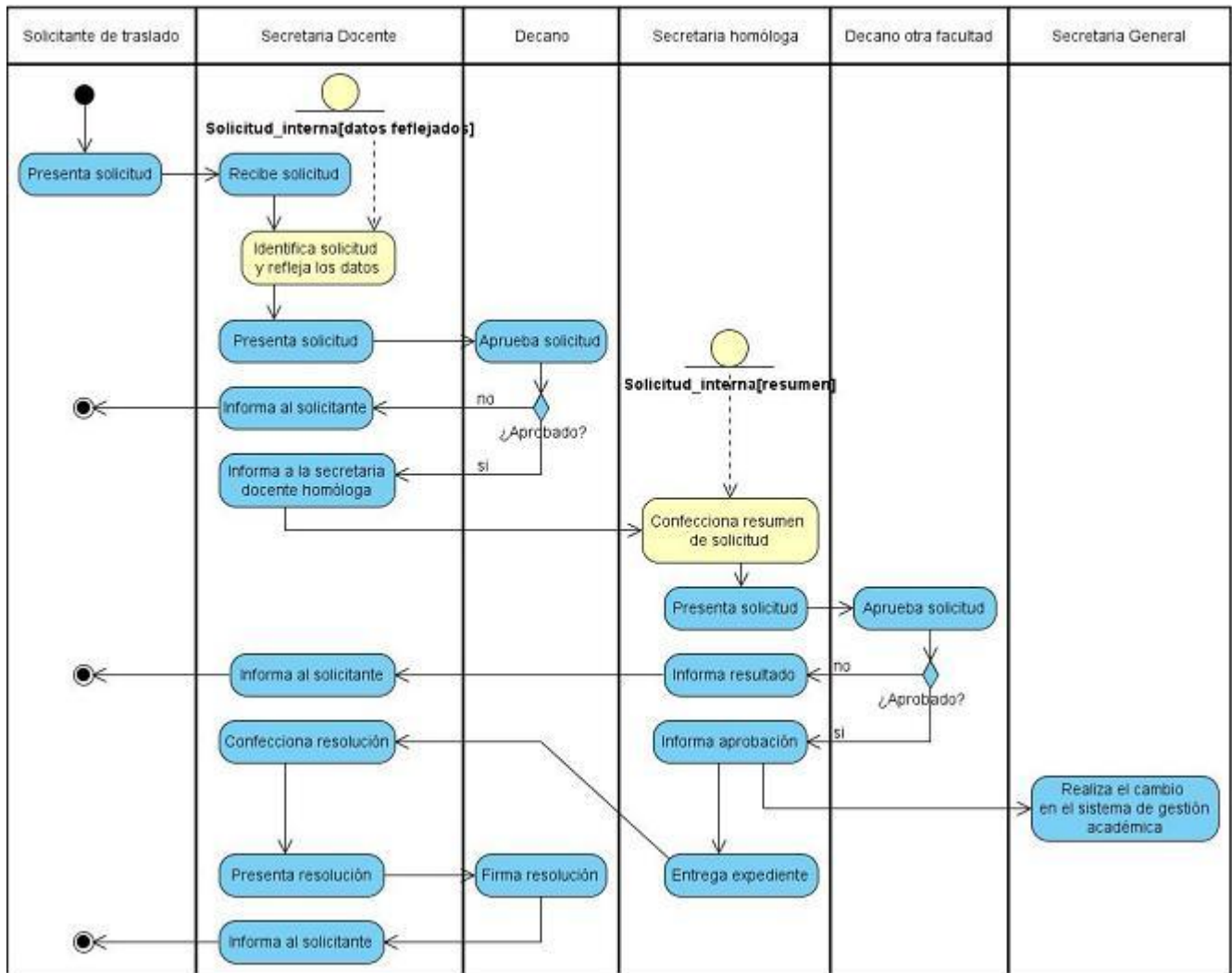


Figura 2: Diagrama de actividades. CU solicitar traslado para otra facultad.

2.2.5.2. Sección solicitar traslados para otro CES.

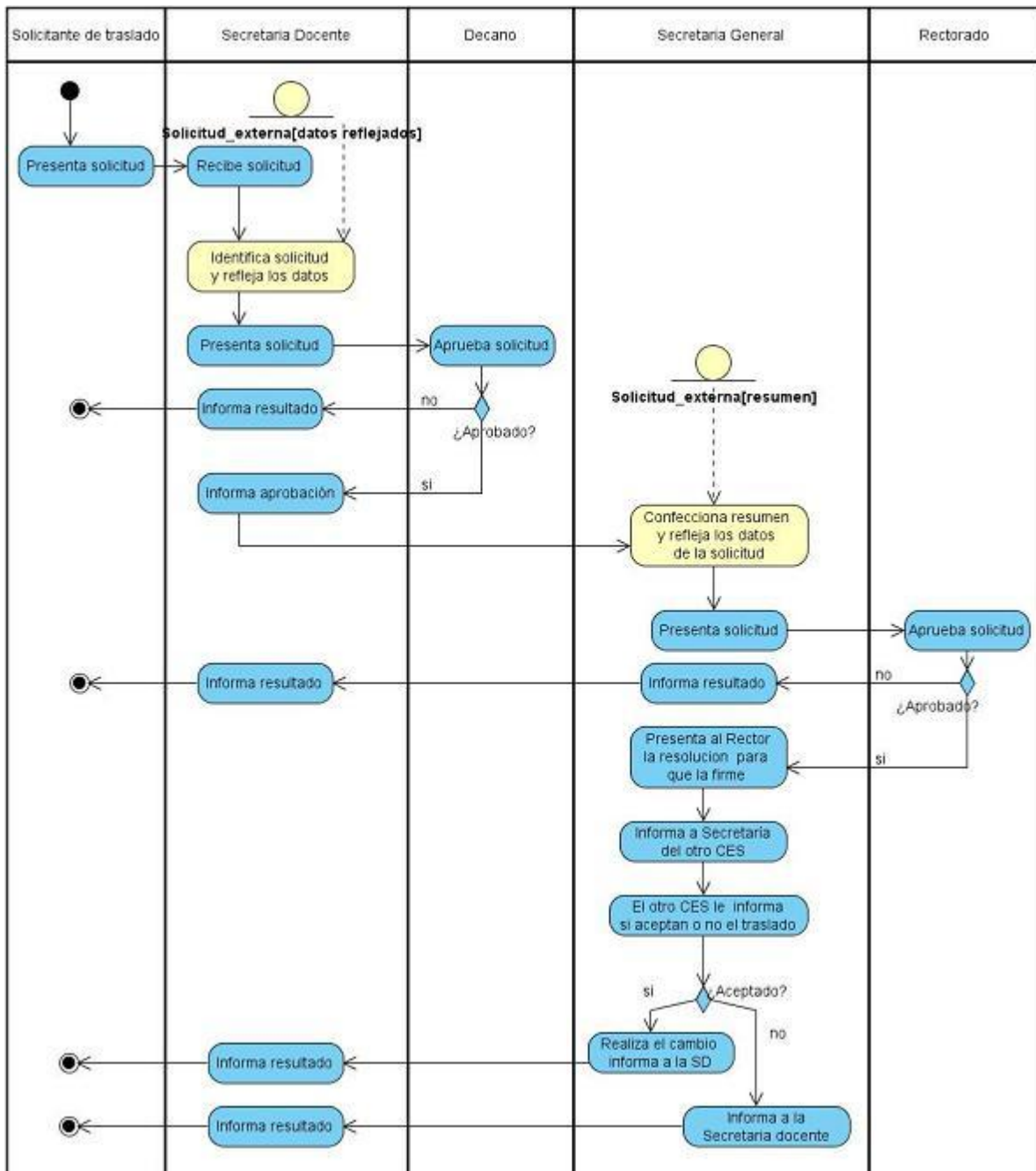


Figura 3: Diagrama de actividades. CU solicitar traslado para otro CES.

2.2.5.3. Sección informar traslado.

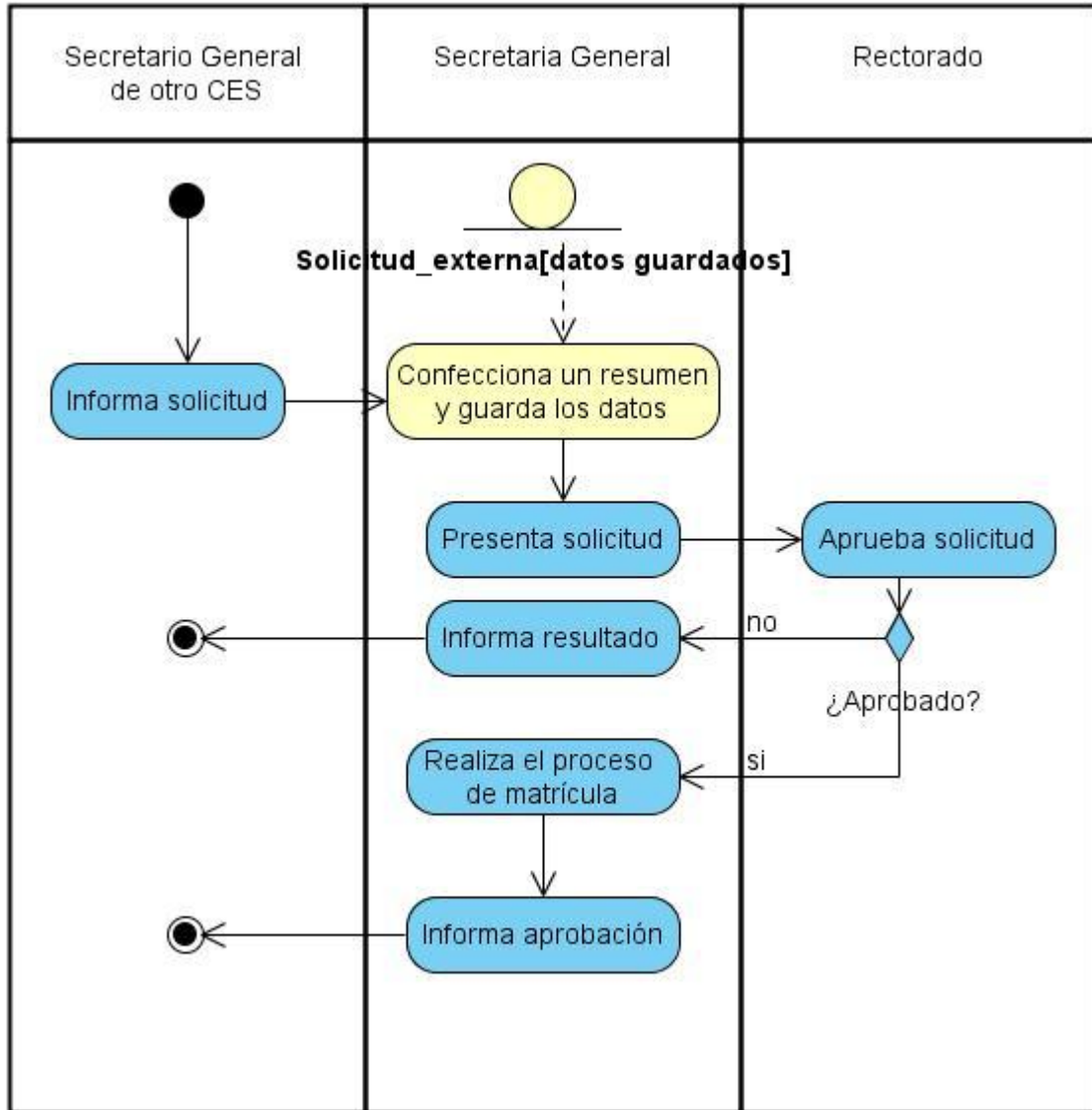


Figura 4: Diagrama de actividades. CU informar traslado.

2.2.6. Diagrama de clases del modelo de objetos del negocio.

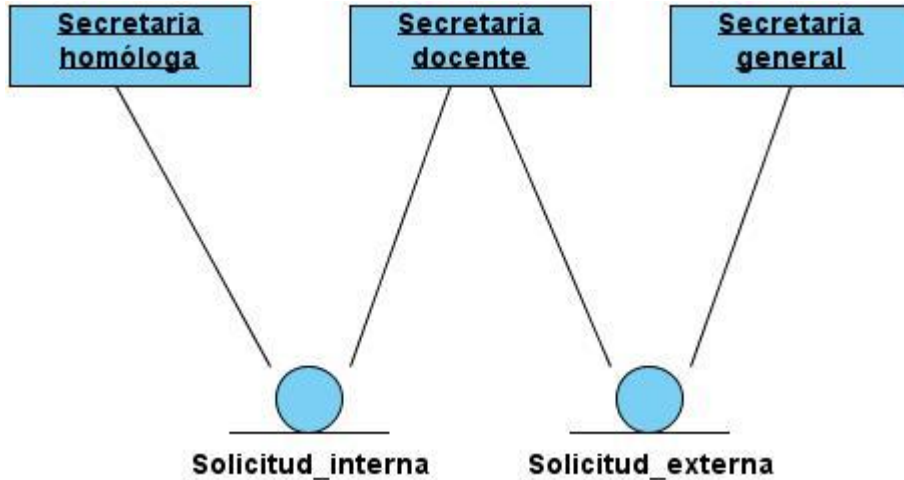


Figura 5: Diagrama de objetos.

2.3. Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Deben ser comprensibles por los clientes, usuarios y desarrolladores. Deben tener una sola interpretación y estar definidos en forma medible y verificable. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. No alteran la funcionalidad del producto. A continuación se mencionan los requisitos funcionales del sistema en desarrollo:

Referencia	Requisito funcional
R1	Permitir que los usuarios se autenticuen.
R2	Gestionar los usuarios que trabajarán con el sistema.
R2-1	Registrar usuario.
R2-2	Modificar usuario.
R2-3	Eliminar usuario.
R3	Gestionar los estudiantes que solicitan traslado.

Capítulo 2: Modelación del negocio y características del sistema

R3-1	Registrar estudiante.
R3-2	Modificar estudiante.
R3-3	Eliminar estudiante.
R4	Buscar usuarios y estudiantes.
R5	Gestionar las solicitudes de traslado.
R5-1	Registrar solicitud de traslado.
R5-2	Modificar solicitud de traslado.
R5-3	Eliminar solicitud de traslado.
R6	Aprobar las solicitudes de traslado registradas.
R7	No aprobar las solicitudes de traslado registradas.
R8	Gestionar las resoluciones de las solicitudes.
R8-1	Registrar resolución.
R8-2	Modificar resolución.
R8-3	Eliminar resolución.
R9	Registrar expediente de solicitud.
R10	Generar reportes.
R10-1	Listar estudiantes que solicitan traslados.
R10-2	Listar los traslados que se han aprobados.
R10-3	Listar las solicitudes por cualquier criterio de dato.
R10-4	Listar los expedientes de las solicitudes.
R10-5	Listar los usuarios registrados.
R11	Enviar correo de aviso cuando se registre una solicitud interna a la Secretaria Docente.
R12	Generar las solicitudes como archivos.pdf.

Tabla 5: Requisitos funcionales del sistema.

2.4. Requerimientos no funcionales.

Los requerimientos no funcionales especifican propiedades del sistema, cualidades que el producto debe tener, estas pueden ser características que hagan al producto atractivo, usable, rápido confiable, como restricciones del entorno o de la implementación, rendimiento, dependencias de las plataformas, facilidad de mantenimiento, extensibilidad, y fiabilidad. La fiabilidad hace referencia a características como la disponibilidad, exactitud, tiempo medio entre fallos, defectos por varias líneas de código, y defectos por clases. Un requisito por rendimiento impone condiciones sobre los requisitos funcionales, como la velocidad, rendimiento, tiempo de respuesta y uso de memoria. Están relacionados con los requerimientos funcionales pues una vez definido lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener y cuán rápido o grande debe ser. A continuación se especifican los requisitos no funcionales que debe tener el sistema en desarrollo:

2.4.1. Apariencia o interfaz externa.

- El sistema tendrá una interfaz fácil de usar y amigable para el buen entendimiento de la aplicación con los usuarios.
- Estará diseñado para la resolución deseada por el usuario, aunque debe soportar estándar de 800 x 600 píxeles.

2.4.2. Usabilidad.

- El sistema debe ser usado por las personas autorizadas y capacitadas para manejar los datos dentro del proceso de traslado.
- El sistema debe estar disponible las 24 h del día.

2.4.3. Rendimiento.

- Las funcionalidades estarán divididas en secciones, de modo que no se sobrecarguen los pedidos.
- Las conexiones a la BD deben ser de manera rápida y segura, las respuestas deben ser en el mínimo de tiempo posible.

2.4.4. Hardware.

- Para los usuarios los requerimientos mínimos son: Procesador Pentium IV a 1.6GHz con 256 Mb de memoria. Tarjeta de red.
- Para el servidor los requerimientos mínimos son: Procesador Pentium IV a 3GHz y 1Gb de memoria RAM. Tarjeta de red.

2.4.5. Software.

- Para los usuarios deben tener instalado cualquier navegador web.
- Para el servidor tener instalado SO Windows, gestor de BD PostgreSQL y tener instalado AppServ 2.5.10.

2.4.6. Políticos culturales.

- El producto no debe contener palabras en otros idiomas.
- Debe contener información acorde a los principios éticos puestos en vigor en las áreas de Secretaría General y Secretaría Docente.

2.4.7. Portabilidad.

- El sistema deberá ser multiplataforma, es decir debe correr en diferentes plataformas (Linux, Windows, etc.).

2.4.8. Interoperabilidad.

- La interoperabilidad del sistema es necesaria para que los usuarios tengan un acceso completo a la información disponible.

2.4.9. Seguridad.

- El sistema requerirá de la autenticación de los usuarios para ser identificado y acceder a determinadas funcionalidades.
- La autenticación será a través del servicio, Lightweight Directory Access Protocol (LDAP) de la universidad.

2.4.10. Confiabilidad.

- Alto nivel de confiabilidad.
- El sistema informará los errores que puedan ocurrir mostrando los mensajes de la efectividad de las funcionalidades.

2.5. Descripción del sistema propuesto.

2.5.1. Actores del sistema.

Los actores del Sistema son aquellos terceros fuera del sistema que interactúan con él y pueden ser individuos u otros sistemas, por lo que a su vez son trabajadores del sistema y ejercen diferentes roles.

Actores del sistema	Justificación
Secretaria General	Es el rol que controla el sistema en su conjunto, envía las solicitudes de traslado hacia otros CES y recibe las provenientes de estos.
Especialista de Secretaría General	Es el rol que realiza los procesos de solicitudes de traslados para otros CES y para la UCI, archivando toda la información.
Secretaria Docente	Es el rol que realiza el proceso de solicitud de traslado para otra Facultad en la UCI, archivando toda la información.

Tabla 6: Actores del sistema.

2.5.2. Casos de uso del sistema.

Identificador	Caso de uso
CU1	Autenticar usuarios
CU2	Gestionar usuarios
CU3	Gestionar estudiantes internos
CU4	Gestionar estudiantes externos
CU5	Gestionar solicitudes internas
CU6	Gestionar solicitudes externas
CU7	Gestionar resolución de solicitud
CU8	Registrar expediente
CU9	Generar reportes

Tabla 7: Casos de uso del sistema.

2.5.3. Diagrama de casos de uso del sistema.

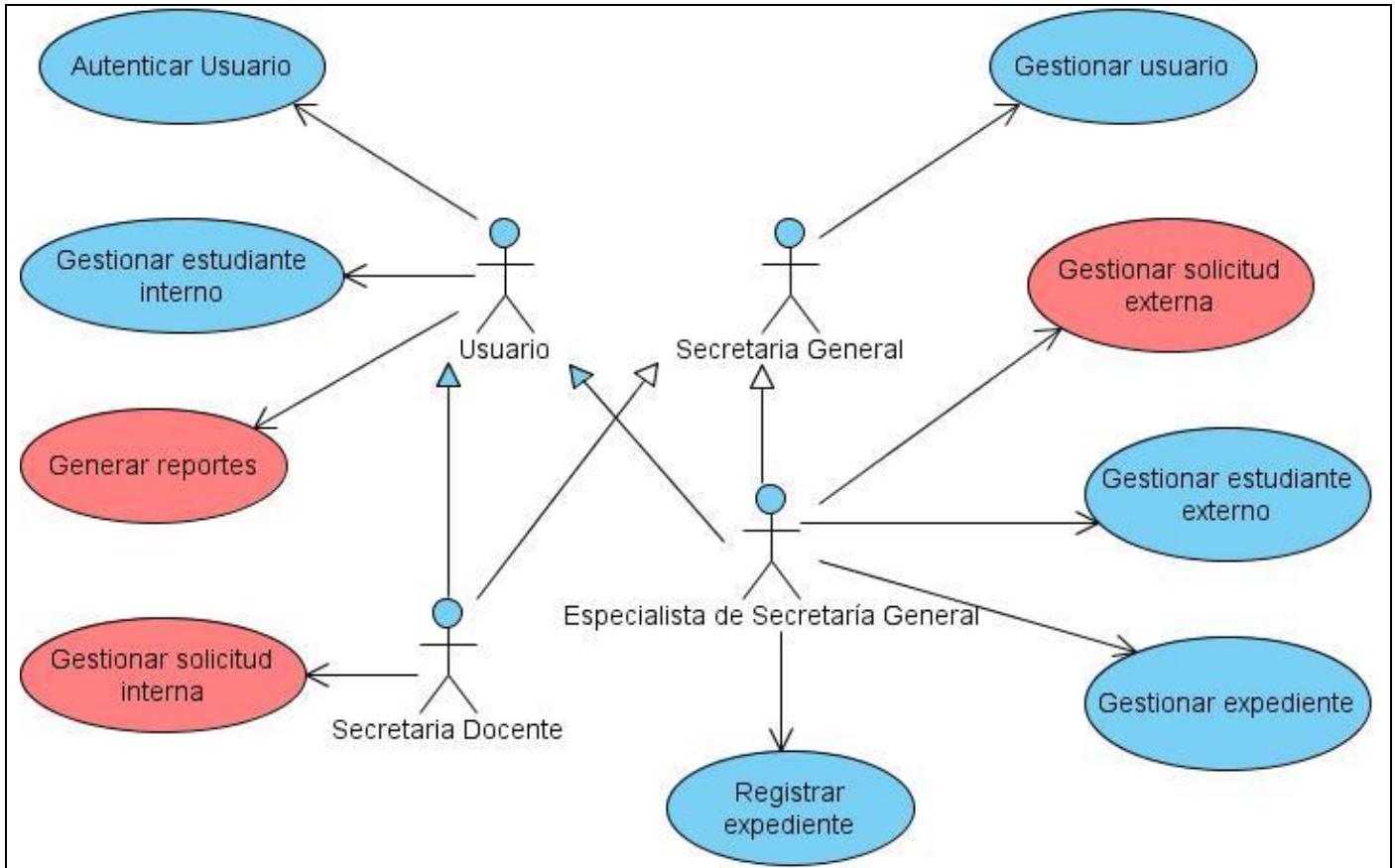


Figura 6: Diagrama de casos de uso del sistema.

2.5.4. Descripción de los casos de uso.

Caso de uso:	Autenticar usuarios.	
Propósito	Iniciar sesión en el sistema.	
Actores:	Secretaria General, Especialista de Secretaría General y Secretaria Docente.	
Resumen:	El usuario accede al sistema para trabajar con los diferentes tipos de Módulos de Trabajo en los que tiene permiso para laborar.	
Referencias:	R1	
Precondiciones:	Los usuarios deben estar registrados en el sistema.	
Pos condiciones:	Guarda en la base de dato las informaciones procesadas.	
Acción del actor	Respuesta del sistema	
	1. Muestra en pantalla el formulario que le permite al usuario introducir su nombre de usuario y contraseña.	
2. Introduce el nombre de usuario y contraseña, pulsa acceder.	3. Comprueba los datos introducidos en el directorio activo de la UCI verificando que son los correctos para dar el acceso. En caso de ser incorrectos, ver flujo alterno 1.	
	4. Verifica que esté registrado en el sistema. Si no está registrado ver flujo alterno 2.	
	5. Muestra la página con las funcionalidades a las cuales el usuario tiene permiso y finaliza el proceso.	
Flujo Alterno		
1. Muestra un mensaje de error indicándole que no tiene permiso para acceder al sistema.		
2. Muestra un mensaje indicando que no está registrado en el sistema y que se ponga en contacto con la Secretaria General.		

Tabla 8: Descripción del CU: Autenticar usuario.

Capítulo 2: Modelación del negocio y características del sistema

Caso de uso:	Gestionar usuario.	
Propósito	Permite agregar, modificar o eliminar los usuarios que van a trabajar con el sistema y le asigna el rol.	
Actores:	Secretaria General.	
Resumen:	Se realiza una búsqueda de los usuarios del dominio UCI, se agregarán en el sistema y se les asigna el rol que ocuparán para que tengan acceso a las distintas funcionalidades.	
Referencias:	R2, R2-1, R2-2, R2-3, R3, R5	
Precondiciones:	1. La Secretaria General debe estar registrada en el sistema.	
Pos condiciones:	1. Guarda en la BD las informaciones procesadas.	
Sección: Registrar usuario.		
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo administración.	2. Muestra las operaciones que se pueden realizar en ese módulo.	
3. Selecciona la opción usuarios.	4. Muestra un formulario para llenar los datos del usuario que se va a registrar.	
5. Introduce datos de interés y asigna el rol al usuario	6. Registra en la BD los datos entrados y el rol que se le asignó. Termina el proceso.	
Precondiciones:	El usuario debe estar registrado.	
Sección: Modificar usuario		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción usuarios.	2. Lista los usuarios registrados y la opción modificar.	
3. Selecciona el usuario que va a modificar.	4. Muestra el formulario con la posibilidad de modificar el rol y otros datos.	
	5. Actualiza los nuevos valores en la BD. Termina el proceso.	
Sección: Eliminar usuario		
Precondiciones:	El usuario debe estar registrado.	
Acción del actor	Respuesta del sistema	
1. Selecciona la opción usuarios.	2. Lista los usuarios que están registrados en el sistema y la opción eliminar.	
3. Selecciona el usuario que desea eliminar.		

Capítulo 2: Modelación del negocio y características del sistema

4. Selecciona la opción eliminar.	5. Procesa la información y elimina el usuario de la BD. Termina el proceso.
-----------------------------------	--

Tabla 9: Descripción del CU: Gestionar usuario.

Caso de uso:	Gestionar estudiante interno.	
Propósito	Gestionar los datos de los estudiantes que solicitan traslado para otro CES o para otra Facultad.	
Actores:	Secretaria General, Especialista de Secretaría General y Secretaria Docente.	
Resumen:	Se busca el estudiante por el carné de identidad (CI), el sistema muestra los datos en un formulario que permite modificar los datos y registrar el estudiante en la BD de sistema, también se puede eliminar cualquier estudiante que haya sido registrado.	
Referencias:	R4, R4-1, R4-2, R5	
Precondiciones:	1. Los usuarios deben estar autenticados.	
Pos condiciones:	1. Guardar en la BD las informaciones procesadas.	
Sección: Registrar estudiante interno.		
Acción del actor	Respuesta del sistema	
1. El usuario selecciona el módulo solicitudes.	2. Muestra las operaciones que se pueden efectuar en el módulo de solicitudes.	
3. El usuario selecciona buscar estudiante.	4. Muestra un formulario que le permite introducir el CI para buscar los datos del estudiante.	
5. Introduce el CI del estudiante que solicita traslado.	6. Muestra los datos del estudiante obtenidos de la BD de la Universidad, si no ver flujo alterno 1	
7. Acepta la operación.	8. Muestra un formulario con los datos encontrados y la opción Registrar.	
9. Selecciona la opción Registrar.	10. Procesa la opción seleccionada y guarda en la BD los datos entrados en el formulario. Termina el proceso.	
Flujo alterno		
1. El sistema muestra un mensaje de error informando que no se ha encontrado el estudiante.		
Sección: Modificar estudiante interno.		
Acción del actor	Respuesta del sistema	
1. Selecciona la opción solicitudes.	2. Muestra las operaciones que se pueden efectuar	

Capítulo 2: Modelación del negocio y características del sistema

	en el módulo de solicitudes.
3. Selecciona la opción salida.	4. Lista los estudiantes registrados y las opciones adicionar, modificar y eliminar.
5. Selecciona el estudiante que va a modificar los datos y la opción modificar.	6. Muestra el formulario con los datos del estudiante registrado.
7. Cambia los datos incorrectos.	8. Actualiza en la BD los nuevos valores. Termina el proceso.
Sección: Eliminar estudiante interno.	
Acción del actor	Respuesta del sistema
1. Selecciona la opción solicitudes.	2. Muestra las operaciones que se pueden efectuar en el módulo de solicitudes.
3. Selecciona la opción salida.	4. Lista los estudiantes registrados y las opciones adicionar, modificar y eliminar.
5. Selecciona el estudiante que va a eliminar y la opción eliminar.	6. Elimina de la BD al estudiante. Termina el proceso.

Tabla 10: Descripción del CU: Gestionar estudiante interno.

Capítulo 2: Modelación del negocio y características del sistema

Caso de uso:	Gestionar estudiante externo.	
Propósito	Gestionar los datos de los estudiantes de otros CES que solicitan traslado para la UCI.	
Actores:	Secretaría General y Especialista de Secretaría General	
Resumen:	Una vez aceptada la solicitud por el Rector del centro de origen, la Especialista de Secretaría General accede al sistema para registrar los datos del estudiante que desea realizar el cambio para la UCI.	
Referencias:	R4, R4-1, R4-2.	
Precondiciones:	La Secretaria General o la Especialista de Secretaría General deben estar autenticadas en el sistema.	
Pos condiciones:	Guardar en la BD las informaciones procesadas.	
Sección: Registrar estudiante externo.		
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo solicitudes.	2. Muestra las operaciones que pueden realizarse en ese módulo.	
3. Selecciona la opción entrada.	4. Muestra una ventana que permite adicionar, modificar y eliminar.	
5. Selecciona la opción adicionar.	6. Muestra un formulario que permite recoger los datos del estudiante que solicita traslado para la UCI.	
7. Llena el formulario con los datos del nuevo estudiante, acepta operación.	8. Se guardan en la BD los datos del estudiante registrado. Termina el proceso.	
Sección: Modificar estudiante externo.		
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo solicitudes.	2. Muestra las operaciones que pueden realizarse en ese módulo.	
3. Selecciona la opción entrada.	4. Lista los estudiante que se han registrado y las opciones adicionar	
1. Selecciona el estudiante.	2. Muestra el formulario con los datos del estudiante, permite cambiar algún dato.	
3. Cambia los datos que encuentre incorrectos y selecciona modificar.	4. Procesa la información y actualiza los datos del estudiante en la BD. Termina el proceso.	
Sección: Eliminar estudiante externo.		

Capítulo 2: Modelación del negocio y características del sistema

Acción del actor	Respuesta del sistema
1. Selecciona la opción solicitudes.	2. Muestra las operaciones que se pueden efectuar en el módulo de solicitudes.
3. Selecciona la opción entrada.	4. Lista los estudiantes registrados y las opciones adicionar, modificar y eliminar.
5. Selecciona el estudiante que va a eliminar y la opción eliminar.	6. Elimina de la BD al estudiante. Termina el proceso.

Tabla 11: Descripción del CU: Gestionar estudiante externo.

Caso de uso:	Gestionar solicitud interna.
Propósito	Gestionar las solicitudes de traslado entre facultades realizadas por los estudiantes, permite saber en qué estado se encuentra la solicitud.
Actores:	Secretaria Docente.
Resumen:	El caso de uso comienza cuando la Secretaria Docente se identifica en el sistema y agrega la nueva solicitud de traslado para otra facultad.
Referencias:	R6, R6-1, R6-2, R7, R8
Precondiciones:	<ol style="list-style-type: none"> 1. La secretaria debe autenticarse en el sistema. 2. Las solicitudes deben estar aprobadas por el Decano.
Pos condiciones:	<ol style="list-style-type: none"> 1. Guardar en la BD las informaciones procesadas
Sección: Registrar solicitud interna.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las operaciones que pueden realizarse en ese módulo.
3. Selecciona la opción traslados.	4. Muestra una ventana con las opciones adicionar, modificar o eliminar.
5. Selecciona la opción adicionar.	6. Muestra una ventana con la opción de estudiante y solicitud.
7. Selecciona la opción solicitud.	8. Muestra el formulario con los campos para recoger los datos de la solicitud de traslado para otra facultad.
9. Llena el formulario con los datos	10. Procesa la información y guarda los datos en la

Capítulo 2: Modelación del negocio y características del sistema

correspondientes y la opción aceptar.	BD.
	11. El sistema envía un correo de aviso a la secretaria docente de la facultad para la cual se registró la solicitud.
12. La Secretaria Docente de la facultad para la cual se dirige la solicitud entra al sistema.	
13. Revisa las solicitudes de traslado para otra facultad "traslados".	
14. Selecciona la solicitud que le fue avisada por correo.	15. Muestra los datos de la solicitud y la opción aprobar
16. Selecciona la opción aprobar solicitud, una vez que el Decano haya aceptado la solicitud para su facultad de lo contrario ver flujo alterno 1	17. Procesa la información y registra el estado en que queda la solicitud (aprobado o no aprobado).
	18. Le envía un correo a la Secretaria Docente de la facultad de destino informando el resultado de la operación. Termina el proceso.
Flujo alterno	
1. La Secretaria Docente no aprueba la solicitud.	
Sección: Modificar solicitud interna.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.
3. Selecciona la opción "traslados".	4. Lista las solicitudes que se han registrado en el sistema y la opción de modificar.
5. Selecciona la opción buscar.	6. Muestra un formulario que le permite entrar el nombre.
7. Entra el nombre del estudiante.	8. Muestra la solicitud encontrada, si no ver flujo alterno 1.
9. Cambia los datos que encuentre incorrectos y selecciona modificar.	10. Procesa la información y actualiza los datos de la solicitud en la BD. Termina el proceso.
Flujo Alterno	
1. Muestra un mensaje de error que expresa: no existe ninguna solicitud con ese nombre.	

Capítulo 2: Modelación del negocio y características del sistema

Sección: Eliminar solicitud interna.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.
3. Selecciona la opción "traslados".	4. Lista las solicitudes que se han registrado en el sistema y la opción eliminar.
5. Selecciona la opción buscar.	6. Muestra un formulario que le permite entrar el nombre.
7. Entra el nombre del estudiante.	8. Muestra la solicitud encontrada, si no ver flujo alterno 1.
9. Selecciona la opción eliminar	10. Elimina la solicitud de la BD. Termina el proceso.
Flujo Alterno	
1. Si no muestra un mensaje que la solicitud no existe.	

Tabla 12: Descripción del CU: Gestionar solicitud interna.

Caso de uso:	Gestionar solicitud externa.
Propósito	Gestionar las solicitudes de traslado externas realizadas por los estudiantes, permite saber en que estado se encuentra la solicitud. Estas solicitudes pueden ser para otra Universidad o hacia la UCI.
Actores:	Secretaria General y Especialista de Secretaría General
Resumen:	El caso de uso comienza con la autenticación de los actores, luego registran los datos de las solicitudes efectuadas.
Referencias:	R6, R6-1, R6-2, R7, R8
Precondiciones:	Los actores deben autenticarse en el sistema.
Pos condiciones:	Se guardarán en la BD las informaciones procesadas.
Sección: Registrar solicitud externa.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene este módulo.
3. Selecciona la opción "Entrada", si no ver flujo alterno 1.	4. Muestra una ventana con las solicitudes efectuadas y las opciones adicionar, modificar o eliminar.
5. Selecciona la opción adicionar.	6. Muestra una ventana que le permite seleccionar el estudiante o la solicitud.

Capítulo 2: Modelación del negocio y características del sistema

7. Selecciona solicitud.	8. Muestra un formulario que le permite recoger los datos de la solicitud que se desea registrar.
9. Entra los datos y acepta la operación.	10. Guarda en la BD los datos de la solicitud. Termina el proceso.
Flujo alternativo	
1. El actor selecciona la opción "Salida".	
Sección: Modificar solicitud externa.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.
3. Selecciona el tipo de solicitud ya sea de entrada o salida que desea modificar.	4. Muestra una ventana que lista las solicitudes registradas y las opciones adicionar, modificar, eliminar y buscar.
5. Selecciona el buscador.	6. Muestra el formulario que le permite entrar el nombre para buscar la solicitud.
7. Entra el nombre del estudiante que realizó la solicitud.	8. Muestra la solicitud encontrada, si no ver flujo alternativo 1.
9. Selecciona la solicitud encontrada y la opción modificar.	10. Muestra el formulario con los datos de la solicitud y permite realizar los cambios.
11. Cambia los datos que desee.	12. Actualiza en la BD la nueva información entrada por el usuario. Termina el proceso.
Flujo Alterno	
1. Muestra un mensaje que expresa: no existe ninguna solicitud con ese nombre.	
Sección: Eliminar solicitud externa.	
Acción del actor.	Respuesta del sistema.
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.
3. Selecciona el tipo de solicitud ya sea de entrada o salida que desea modificar.	4. Muestra una ventana que lista las solicitudes registradas y las opciones adicionar, modificar, eliminar y buscar.
5. Selecciona el buscador.	6. Muestra el formulario que le permite entrar el nombre para buscar la solicitud.
7. Entra el nombre del estudiante que realizó la solicitud.	8. Muestra la solicitud encontrada, si no ver flujo alternativo 1.

Capítulo 2: Modelación del negocio y características del sistema

9. Selecciona la solicitud encontrada y la opción eliminar.	10. Elimina la solicitud de la BD. Termina el proceso.
Flujo Alterno	
1. Muestra un mensaje que expresa: no se ha encontrado ninguna solicitud con ese nombre.	

Tabla 13: Descripción del CU: Gestionar solicitud externa.

Capítulo 2: Modelación del negocio y características del sistema

Caso de uso:	Gestionar resolución de solicitud.	
Propósito	Gestionar la situación de las resoluciones salidas por cada solicitud de traslado hacia otro centro universitario.	
Actores:	Secretaría General y Especialista de Secretaría General.	
Resumen:	Una vez registra una solicitud de traslado para otro CES, se desea controlar el estado en que se encuentra la resolución, gestionando las resoluciones en el sistema.	
Referencias:	R9, R9-1, R9-2.	
Precondiciones:	<ol style="list-style-type: none"> 1. Los autores deben estar autenticados en el sistema. 2. Las resoluciones de las solicitudes deben estar confeccionadas para que el expediente pueda salir. 	
Pos condiciones:	Se guardarán en la BD las informaciones procesadas.	
Sección: Registrar resolución.		
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo solicitudes.	2. Muestra las operaciones que se pueden realizar en ese módulo.	
3. Selecciona la opción "Resoluciones".	4. Muestra una ventana con las resoluciones creadas y las opciones adicionar, modificar y eliminar.	
5. Selecciona la opción adicionar.	6. Muestra un formulario que le permite entrar los datos de la resolución.	
7. Entra los datos de la resolución.	8. Guarda en la BD los datos entrados por el usuario.	
9. Selecciona la solicitud del estudiante que se ha confeccionado la resolución.		
10. Refleja en la solicitud la resolución confeccionada.	11. Guarda en la BD los datos entrados por el usuario. Termina el proceso.	
Sección: Modificar resolución.		
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.	
3. Selecciona la opción "Resoluciones".	4. Muestra una ventana con las resoluciones que se han efectuado y las opciones adicionar, modificar y eliminar.	
5. Selecciona la resolución que va a modificar y la opción modificar.	6. Muestra el formulario con los datos de la resolución y posibilita cambiar algún dato.	

Capítulo 2: Modelación del negocio y características del sistema

7. Cambia los datos que encuentre incorrectos y selecciona aceptar.	8. Procesa la información y actualiza los datos de la resolución en la BD. Termina el proceso.
Sección: Eliminar resolución.	
Acción del actor	Respuesta del sistema
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que tiene el módulo.
3. Selecciona la opción "Resoluciones".	4. Muestra una ventana con las resoluciones que se han confeccionado y las opciones adicionar, modificar y eliminar.
5. Selecciona la resolución que va a eliminar y la opción eliminar.	6. Elimina la resolución de la BD. Termina el proceso.

Tabla 14: Descripción del CU: Gestionar resolución.

Caso de uso:	Registrar expediente.	
Propósito	Registrar en el sistema los datos relacionados con el expediente de los estudiantes.	
Actores:	Secretaria General, Especialista de Secretaría General.	
Resumen:	Una vez registradas las solicitudes de traslado para otro CES, se registra el expediente de cada estudiante recogiendo los datos de interés para saber el estado en que se encuentran los expedientes.	
Referencias:	R9	
Precondiciones:	Los actores se autentifiquen en el sistema.	
Pos condiciones:	Las informaciones procesadas se guardarán en la BD.	
Acción del actor	Respuesta del sistema	
1. Selecciona el módulo solicitudes.	2. Muestra las opciones que se pueden efectuar en el módulo.	
3. Selecciona la opción Salida	4. Muestra una ventana con las solicitudes registradas y las opciones adicionar, modificar, eliminar y expediente.	
5. Selecciona la solicitud a la cuál se le registrará el expediente y la opción expediente.	6. Muestra un formulario que permiten recoger los datos de interés sobre el expediente.	
7. Entra los datos y acepta.	8. Se guardan los datos en la BD. Termina el CU.	

Tabla 15: Descripción del CU: Registrar expediente.

Caso de uso:	Generar reportes.	
Propósito	Mostrar en pantalla cualquier información que el usuario necesite conocer.	
Actores:	Secretaria General, Especialista de Secretaría General y Secretaria Docente	
Resumen:	Una vez que los usuarios se registren en el sistema de acuerdo con el rol que le fue asignado podrá tener acceso a las diferentes opciones de reportes que el sistema proveerá.	
Referencias:	R10, R10-1, R10-2, R10-3	
Precondiciones:	Los actores se autentifiquen en el sistema.	
Pos condiciones:	Las informaciones procesadas se guardarán en la BD.	
Acción del actor	Respuesta del sistema	
9. Selecciona el módulo reportes.	10. Muestra las opciones que le permitirán conocer cualquier información sobre: los estudiantes registrados, usuarios registrados, el estado de las solicitudes y los expedientes.	
11. Selecciona la opción de la cual precisa obtener información.	12. Muestra una interfaz que le permita entrar el criterio por el cual desea buscar y la opción buscar.	
13. Entra el criterio por el cual desea buscar.	14. Muestra la lista correspondiente a la opción deseada (de estudiantes, usuarios, solicitudes o expedientes). Termina el proceso.	

Tabla 16: Descripción del CU: Generar reportes.

2.6. Conclusiones parciales.

En el presente capítulo se explican cómo se realiza el proceso de traslado de los estudiantes en el área de Secretaría General, y se especifican cuales son los actores y los trabajadores que laboran en el negocio. Además se explican las características del sistema propuesto, se especifican las funcionalidades del mismo y se plantean los casos de usos necesarios para el sistema.

Capítulo 3 “Diseño del sistema”.

3.1. Introducción.

Con el presente capítulo se persigue una mayor comprensión de los requisitos del sistema ya mencionados en el capítulo anterior, a través de la construcción del Modelo de Diseño. Esto abarca la descripción de los diagramas de clases de la aplicación y los diagramas de interacción correspondientes a los casos de uso.

3.2. Diseño.

3.2.1. Definición del modelo de diseño.

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Para dar soporte al diseño de este sistema, existe una arquitectura definida, de la cual se expondrán algunos elementos importantes a tener en cuenta para la realización del diseño.

3.2.1.1. Modelo Vista Controlador (MVC).

El patrón Modelo Vista Controlador (MVC) es un patrón de diseño de software en el cual todo el proceso está dividido en 3 capas, típicamente estas capas son el Modelo, la Vista y el Controlador (Reynoso, Kiccillof, & AIREs, 2004).

El Modelo representa las estructuras de los datos. Incorpora la capa del dominio y persistencia, la cual es la encargada de guardar los datos en un medio persistente (ya sea una base de datos, un archivo de texto, XML, registro, etc.). En el modelo es donde se realiza el levantamiento de todos los objetos que el sistema debe utilizar, es el proveedor de los recursos.

La Vista es la información que se presenta al usuario, se encarga de presentar la interfaz en prototipos web, aunque pueden existir otros tipos de vistas. En la vista solo se deben llevar a cabo operaciones simples, sentencias condicionales, ciclos, y formateo. Una Vista en CodeIgniter, puede ser también un fragmento de la página como un título o pie de página. También puede ser una página de RSS, o cualquier otro tipo de página.

El Controlador es el que capta los cambios en la vista y se los envía al modelo, el cual le regresa los datos a la vista, generando una página web, es un ciclo donde cada acción del usuario causa el inicio de un nuevo ciclo.

La forma más sencilla de implementar este patrón es utilizando capas; como regla los accesos a la base de datos se hacen en el modelo; la vista y el controlador no deben saber si se usa o no una base de datos. El controlador es el que decide qué vista se debe imprimir y qué información es la que se envía.

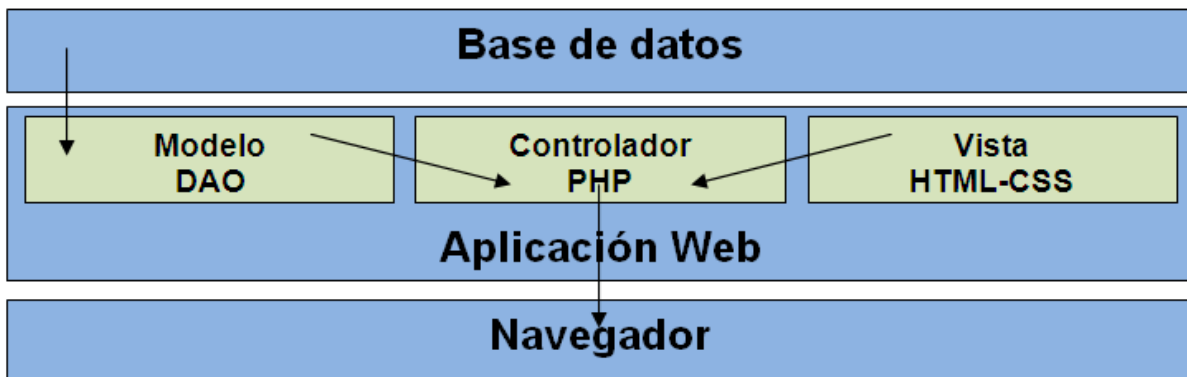


Figura 7: Funcionamiento del Modelo Vista Controlador.

3.2.2. Descripción de las clases.

3.2.2.1. Paquetes del diseño.

Para una mejor comprensión a continuación se explica como queda distribuido el sistema en paquetes para tener un mejor acceso a los datos y un mayor control de la información que se maneja, también se muestran las relaciones entre los mismos. Cada paquete contiene las clases del negocio correspondientes.

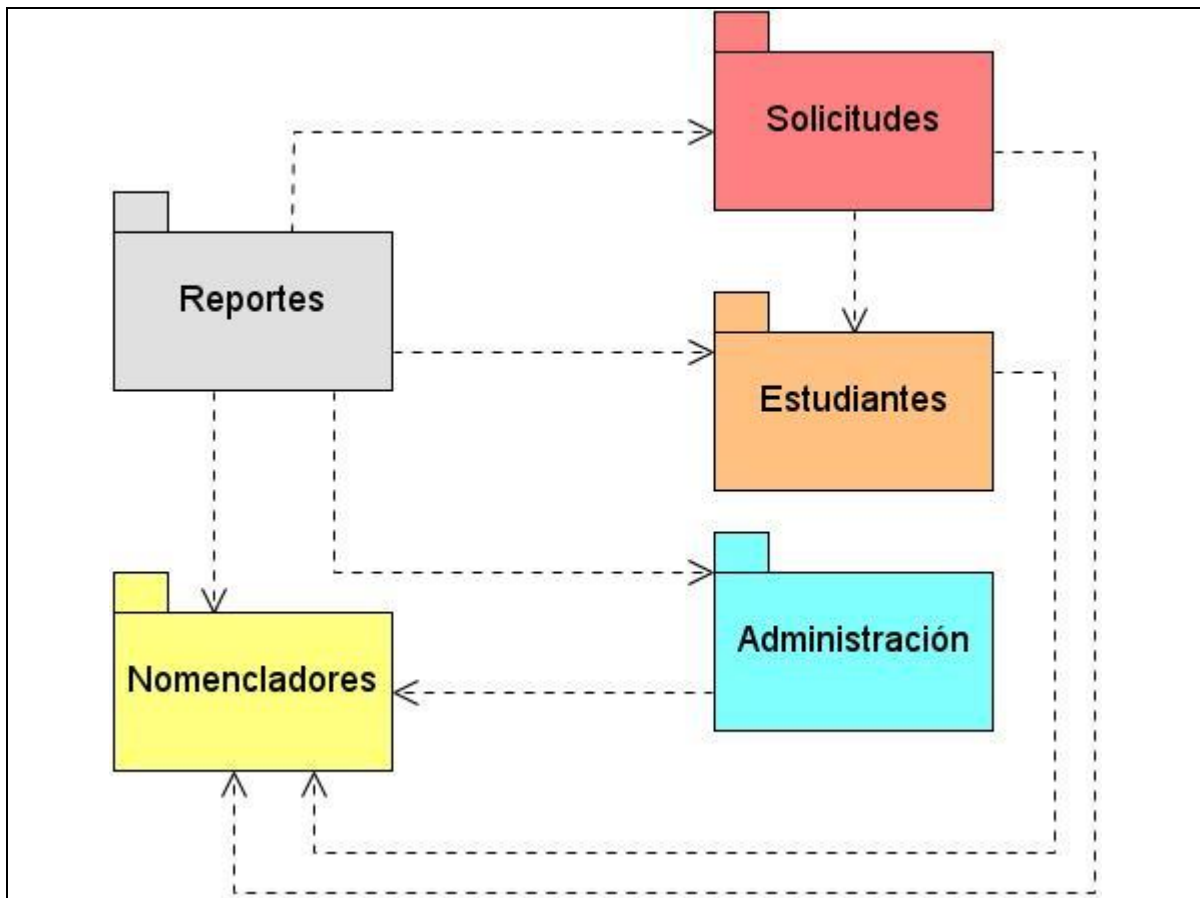


Figura 8: Diagrama de paquetes del sistema.

3.2.2.2. Paquete Administración. Descripción de las Clases.

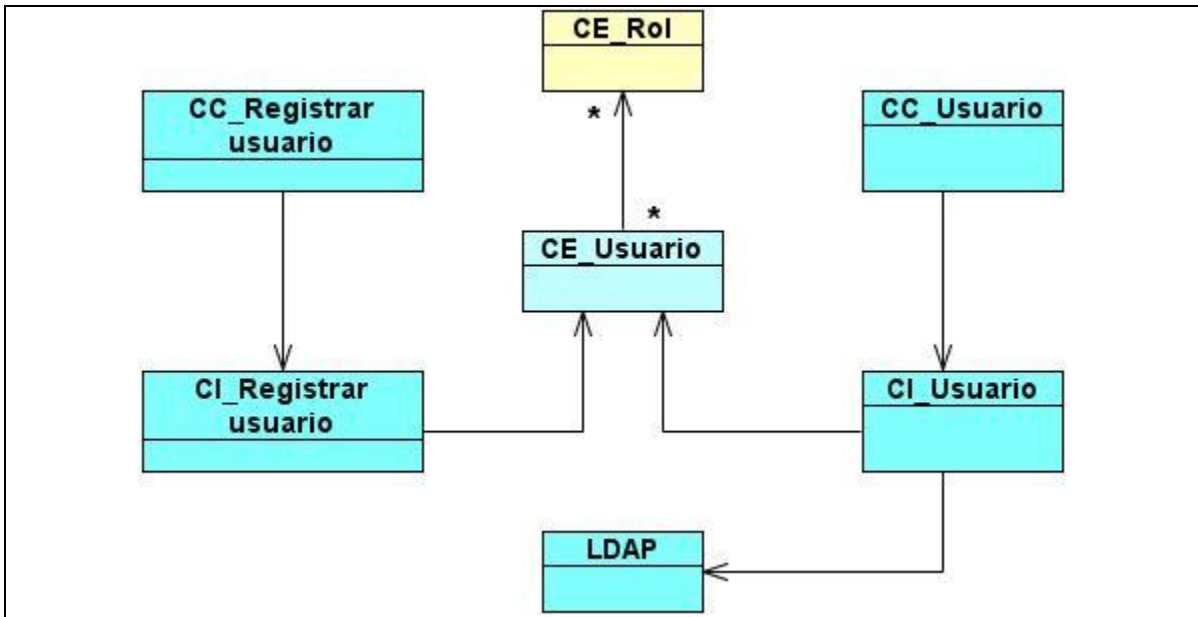


Figura 9: Diagrama de clases incluidas en el paquete de administración.

Nombre:	CE_Usuario
Tipo de clase:	Entidad
Atributos:	Tipo de dato:
idusuario	numeric
usuario	varchar
telefono	numeric
area	varchar
idrol	numeric

Tabla 17: Descripción de la clase CE_Usuario.

Nombre:	CC_Usuario
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de los usuarios que trabajarán con el sistema.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de cualquier usuario registrado en el sistema.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier usuario.
Responsabilidad:	load_usuario ()
Descripción:	Este método permite cargar los datos de los usuarios.
Responsabilidad:	get ()
Descripción:	Este método permite obtener los datos de los usuarios.
Responsabilidad:	log_in ()
Descripción:	Este método permite verificar que los datos entrados para acceder al sistema son los correctos, verifica en la BD de la Universidad que sea usuario UCI y luego verifica que está registrado en el sistema, dando permiso a acceder al sistema.

Tabla 18: Descripción de la clase CC_Usuario.

3.2.2.3. Paquete Solicitudes. Descripción de las Clases.

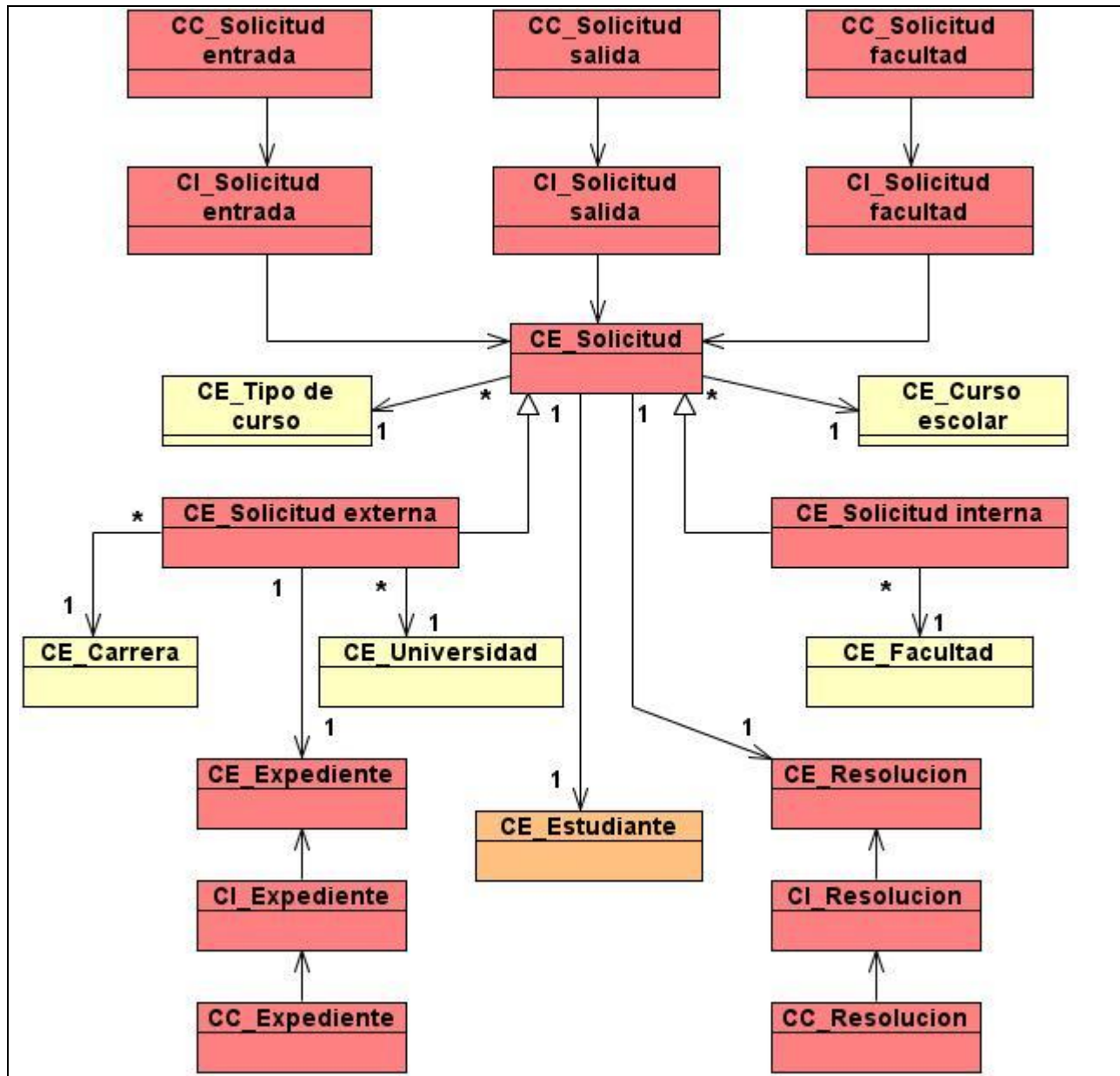


Figura 10: Diagrama de clases incluidas en el paquete de solicitudes.

Nombre:	CE_Solicitud	
Tipo de clase:	Entidad	
Atributos:	Tipo de dato:	
idsolicitud	numeric	
licencia	numeric	
motivos	varchar	
ci	numeric	
idcurso	numeric	
idtipocurso	numeric	
idresolucion	numeric	

Tabla 19: Descripción de la clase CE_Solicitud.

Nombre:	CE_Solicitud externa	
Tipo de clase:	Entidad	
Atributos:	Tipo de dato:	
idsolicitud	numeric	
iduniversidad	numeric	
idcarrera	numeric	
entrada	numeric	

Tabla 20: Descripción de la clase CE_Solicitud externa.

Nombre:	CE_Solicitud interna	
Tipo de clase:	Entidad	
Atributos:	Tipo de dato:	
idsolicitud	numeric	
idfacultad	numeric	

Tabla 21: Descripción de la clase CE_Solicitud interna.

Nombre:	CE_Expediente	
Tipo de clase:	Entidad	
Atributos:	Tipo de dato:	
idsolicitud	numeric	
fecha	date	
salida	numeric	
recibidopor	varchar	

Tabla 22: Descripción de la clase CE_Expediente.

Nombre:	CE_Resolucion	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idsolicitud		numeric
estado		varchar
fecha		date

Tabla 23: Descripción de la clase CE_Resolucion.

Nombre:	CC_Solicitud salida
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de las solicitudes de traslado para la UCI.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de cualquier solicitud para la UCI ya registrada en caso de que los datos requieran cambios.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier solicitud.
Responsabilidad:	load_solicitud ()
Descripción:	Este método permite cargar los datos de la solicitud.
Responsabilidad:	get ()
Descripción:	Este método permite obtener los datos de la solicitud.

Tabla 24: Descripción de la clase CC_Solicitud salida.

Nombre:	CC_Solicitud entrada
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de las solicitudes de traslado para la UCI.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de cualquier solicitud para la UCI ya registrada en caso de que los datos requieran cambios.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier solicitud.
Responsabilidad:	load_solicitud ()
Descripción:	Este método permite cargar los datos de la solicitud.
Responsabilidad:	get ()
Descripción:	Este método permite obtener los datos de la solicitud.

Tabla 25: Descripción de la clase CC_Solicitud entrada.

Nombre:	CC_Solicitud facultad
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de las solicitudes de traslado para otra facultad.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de cualquier solicitud para otra facultad ya registrada en caso de que los datos requieran cambios.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier solicitud.
Responsabilidad:	load_solicitud ()
Descripción:	Este método permite cargar los datos de la solicitud.
	get ()
	Este método permite obtener los datos de la solicitud.

Tabla 26: Descripción de la clase CC_Solicitud facultad.

Nombre:	CC_Resolucion
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de las resoluciones que generan las solicitudes.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de cualquier resolución ya registrada en caso de que los datos requieran cambios.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier solicitud.
Responsabilidad:	load_solicitud ()
Descripción:	Este método permite cargar los datos de la solicitud.
Responsabilidad:	load_resolucion()
Descripción:	Este método permite cargar los datos de las resoluciones.
Responsabilidad:	get ()
Descripción:	Este método permite obtener las resoluciones registradas.

Tabla 27: Descripción de la clase CC_Resolucion.

Nombre:	CC_Expediente
Tipo de clase:	Controladora
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de los expedientes de cada estudiante.
Responsabilidad:	load_expediente()
Descripción:	Este método permite cargar los datos de los expedientes.
Responsabilidad:	get ()
Descripción:	Este método permite obtener los expedientes registrados.

Tabla 28: Descripción de la clase CC_Expediente.

3.2.2.4. Paquete Estudiantes. Descripción de las Clases.

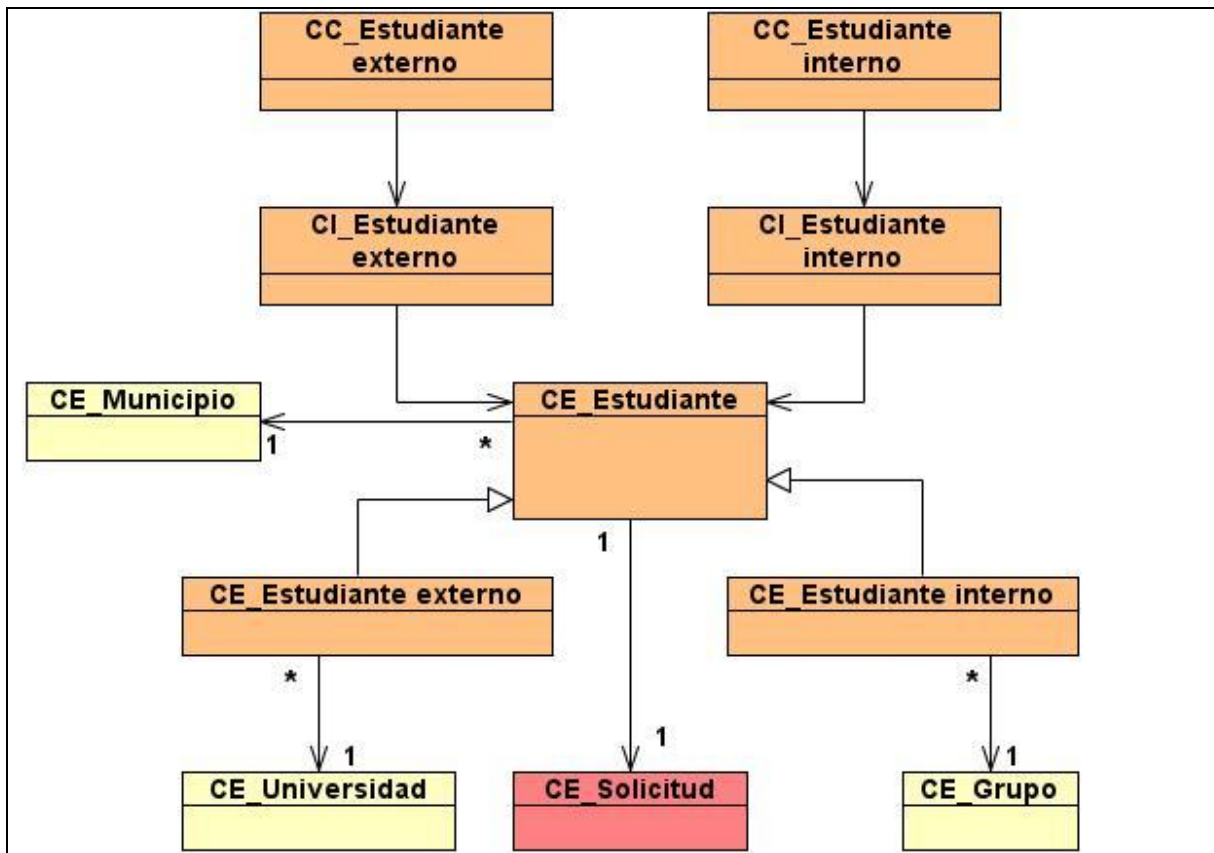


Figura 11: Diagrama de clases incluidas en el paquete de estudiantes.

Nombre:	CE_Estudiante	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
ci		numeric
nombre		varchar
primerapellido		varchar
segundoapellido		varchar
idmunicipio		numeric

Tabla 29: Descripción de la clase CE_Estudiante

Nombre:	CE_Estudiante externo	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
ci		numeric
iduniversidad		numeric
idcarrera		numeric

Tabla 30: Descripción de la clase CE_Estudiante externo.

Nombre:	CE_Estudiante interno	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
ci		numeric
idgrupo		numeric

Tabla 31: Descripción de la clase CE_Estudiante interno.

Nombre:	CC_Estudiante externo	
Tipo de clase:	Controladora	
Responsabilidad:	add ()	
Descripción:	Este método permite agregar los datos de los estudiantes externos.	
Responsabilidad:	upd ()	
Descripción:	Este método permite modificar los datos de los estudiantes externos.	
Responsabilidad:	remove ()	
Descripción:	Este método permite eliminar cualquier estudiante.	
Responsabilidad:	load_estudiante()	
Descripción:	Este método permite cargar los datos del estudiante.	
Responsabilidad:	get ()	
Descripción:	Este método permite obtener los estudiantes externos registrados.	

Tabla 32: Descripción de la clase CC_Estudiante externo.

Nombre:	CC_Estudiante interno
Tipo de clase:	Controladora
Responsabilidad:	load_upd_estudiante ()
Descripción:	Este método permite cargar los datos de los estudiantes de la UCI que se van a registrar en el sistema, esta búsqueda se hace por el CI.
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de los estudiantes internos.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de los estudiantes internos.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier estudiante.
Responsabilidad:	get ()
Descripción:	Este método permite obtener los estudiantes internos registrados.

Tabla 33: Descripción de la clase CC_Estudiante interno.

3.2.2.5. Paquete Reportes. Descripción de las Clases.

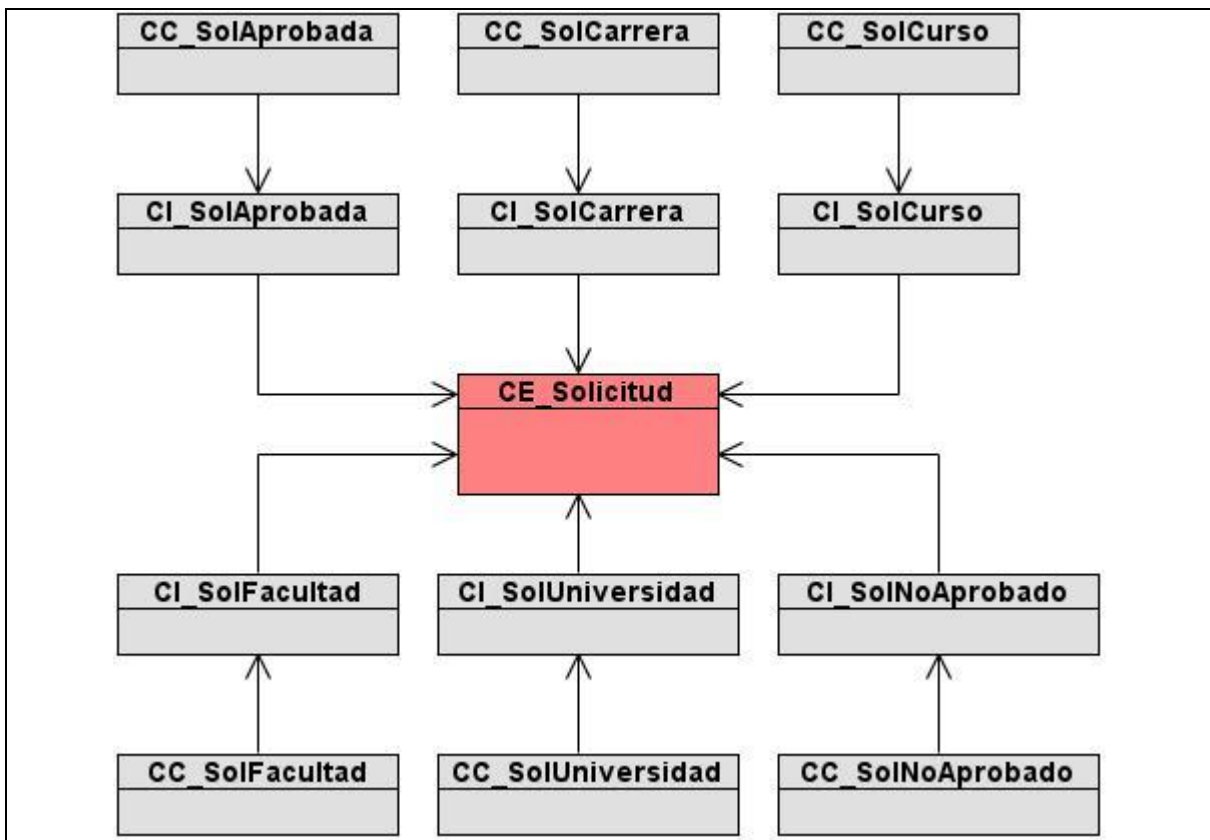


Figura 12: Diagrama de clases incluidas en el paquete de reportes.

Nombre:	CC_SolAprobada
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las solicitudes que se han aprobado.

Tabla 34: Descripción de la clase CC_SolAprobada.

Nombre:	CC_SolUniversidad
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las solicitudes que se han realizado para alguna universidad.

Tabla 35: Descripción de la clase CC_SolUniversidad.

Nombre:	CC_SolCarrera
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las solicitudes que se han efectuado pidiendo cambio para alguna carrera.

Tabla 36: Descripción de la clase CC_SolCarrera.

Nombre:	CC_SolCurso
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las solicitudes que se han efectuado durante un curso escolar.

Tabla 37: Descripción de la clase CC_SolCurso.

Nombre:	CC_SolFacultad
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las solicitudes que se han efectuado por cada facultad.

Tabla 38: Descripción de la clase CC_SolFacultad.

Nombre:	CC_SolNoAprobada
Tipo de clase:	Controladora
Responsabilidad:	get ().
Descripción:	Este método permite obtener las solicitudes que no se han aprobado por algún motivo.

Tabla 39: Descripción de la clase CC_SolNoAprobada.

3.2.2.6. Paquete Nomencladores. Clases.

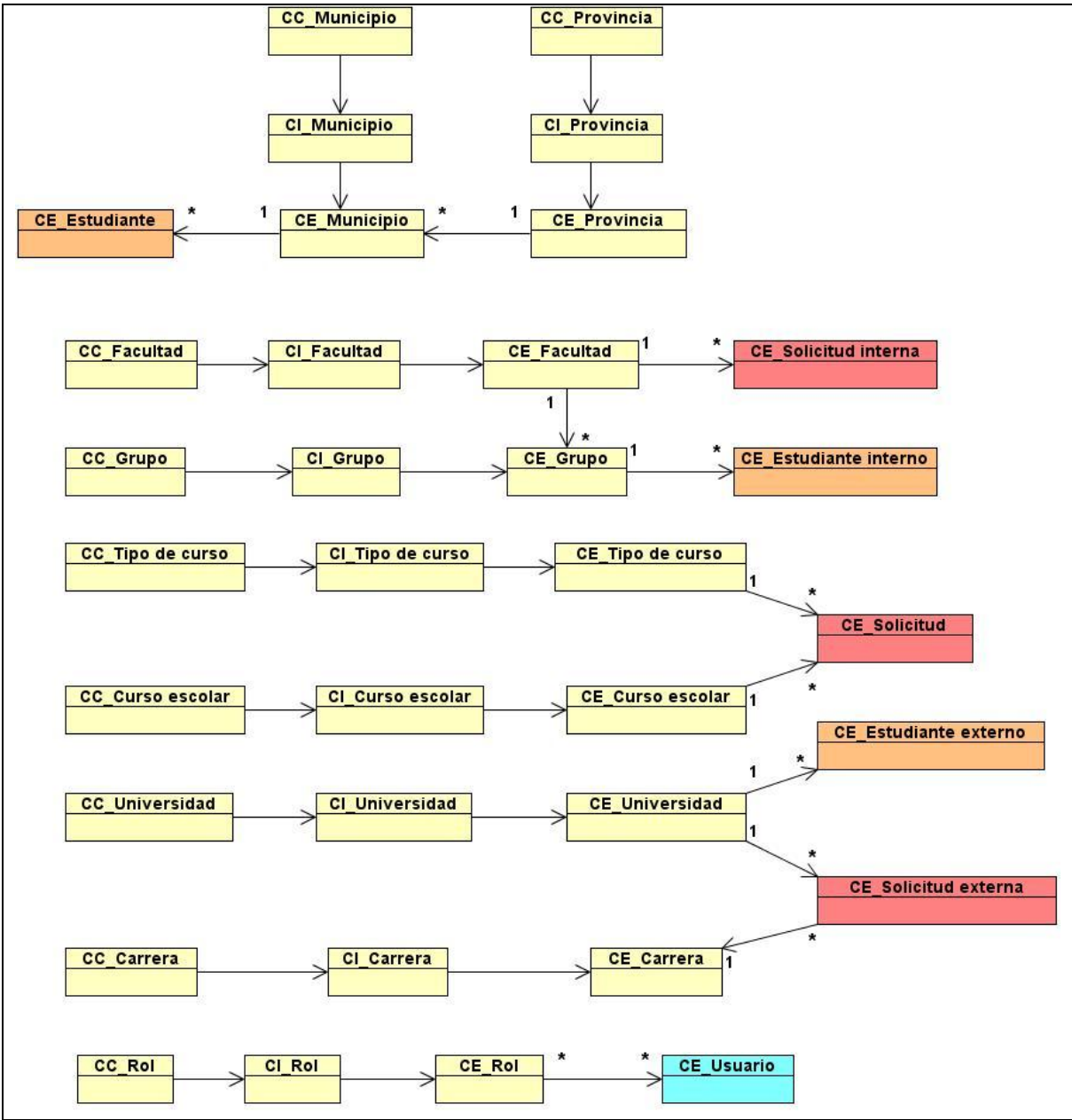


Figura 13: Diagrama de clases incluidas en el paquete de nomencladores.

Nombre:	CE_Municipio	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idmunicipio		numeric
municipio		varchar
idprovincia		numeric

Tabla 40: Descripción de la clase CE_Municipio.

Nombre:	CE_Provincia	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idprovincia		numeric
provincial		varchar

Tabla 41: Descripción de la clase CE_Provincia.

Nombre:	CE_Facultad	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idfacultad		numeric
facultad		numeric

Tabla 42: Descripción de la clase CE_Facultad.

Nombre:	CE_Grupo	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idgrupo		numeric
grupo		numeric

Tabla 43: Descripción de la clase CE_Grupo.

Nombre:	CE_Tipo de curso	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
ldtipocurso		numeric
tipocurso		varchar

Tabla 44: Descripción de la clase CE_Tipo de curso.

Nombre:	CE_Curso escolar	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idcurso		numeric
cursoescolar		varchar

Tabla 45: Descripción de la clase CE_Curso escolar.

Nombre:	CE_Universidad	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
iduniversidad		numeric
universidad		varchar

Tabla 46: Descripción de la clase CE_Universidad.

Nombre:	CE_Carrera	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idcarrera		numeric
carrera		varchar

Tabla 47: Descripción de la clase CE_Carrera.

Nombre:	CE_Rol	
Tipo de clase:	Entidad	
Atributos:		Tipo de dato:
idrol		numeric
rol		varchar

Tabla 48: Descripción de la clase CE_Rol.

Nombre:	CC_Municipio	
Tipo de clase:	Controladora	
Responsabilidad:	get ()	
Descripción:	Este método permite obtener los municipios por cada provincia, dando la opción de seleccionar alguno.	

Tabla 49: Descripción de la clase CC_Municipio.

Nombre:	CC_Provincia
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener los tipos de provincia de nuestro país, dando la opción de seleccionar alguna.

Tabla 50: Descripción de la clase CC_Provincia.

Nombre:	CC_Facultad
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener las facultades existentes en la Universidad, dando la opción de seleccionar alguna.

Tabla 51: Descripción de la clase CC_Facultad.

Nombre:	CC_Grupo
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener los grupos existentes, dando la opción de seleccionar alguno.

Tabla 52: Descripción de la clase CC_Grupo.

Nombre:	CC_Tipo de curso
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener los tipos de cursos existentes, dando la opción de seleccionar alguno.

Tabla 53: Descripción de la clase CC_Tipo de curso.

Nombre:	CC_Curso escolar
Tipo de clase:	Controladora
Responsabilidad:	get ()
Descripción:	Este método permite obtener los cursos en que se efectúan las solicitudes, dando la opción de seleccionar alguno.
Responsabilidad:	load_curso ()
Descripción:	Este método permite cargar los datos de los cursos que se van a registrar en el sistema.
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de los cursos escolares que se van a trabajar con el sistema.

Responsabilidad:	upd ()
Descripción:	Este método permite modificar el dato de los cursos escolares.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier curso escolar.

Tabla 54: Descripción de la clase CC_Curso escolar.

Nombre:	CC_Universidad
Tipo de clase:	Controladora
Responsabilidad:	load_universidad ()
Descripción:	Este método permite cargar los datos de las universidades que se van a registrar en el sistema.
Responsabilidad:	add ()
Descripción:	Este método permite agregar los datos de las universidades que se van a trabajar con el sistema.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de las universidades.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier universidad.
Responsabilidad:	get ()
Descripción:	Este método permite obtener las universidades en que se efectúan las solicitudes, dando la opción de seleccionar alguno.

Tabla 55: Descripción de la clase CC_Universidad.

Nombre:	CC_Carrera
Tipo de clase:	Controladora
Responsabilidad:	load_carrera ()
Descripción:	Este método permite cargar los datos de las carreras que se van a registrar en el sistema.
Nombre:	add ()
Descripción:	Este método permite agregar los datos de las carreras que se van a entrar en el sistema.
Responsabilidad:	upd ()
Descripción:	Este método permite modificar los datos de las carreras.
Responsabilidad:	remove ()
Descripción:	Este método permite eliminar cualquier carrera.
Responsabilidad:	get ()
Descripción:	Este método permite obtener las carreras registradas en la BD, dando la opción de seleccionar alguno.

Tabla 56: Descripción de la clase CC_Carrera.

Nombre:	CC_Rol
Tipo de clase:	Controladora
Nombre:	get ()
Descripción:	Este método permite obtener los roles para asignarle a los usuarios, dando la opción de seleccionar alguno.

Tabla 57: Descripción de la clase CC_Rol.

3.2.3. Diagramas de clases del diseño.

Un artefacto generado dentro del flujo de trabajo diseño es el diagrama de clases, que describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Normalmente contiene clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, información sobre los tipos de los atributos, navegabilidad y dependencias. Un diagrama de este tipo contiene las definiciones de las entidades del software. (Larman, 1999).

3.2.3.1. CU: Autenticar usuario.

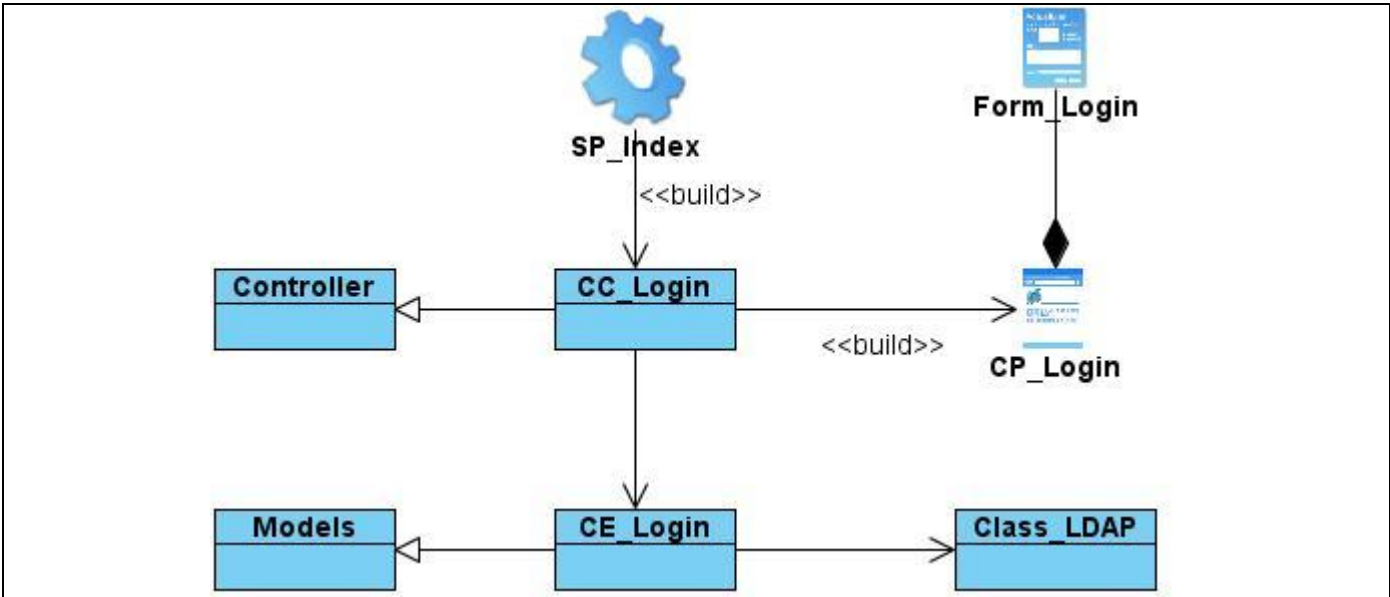


Figura 14: Diagrama de clase del diseño CU: Autenticar usuario.

3.2.3.2. CU: Gestionar usuario.

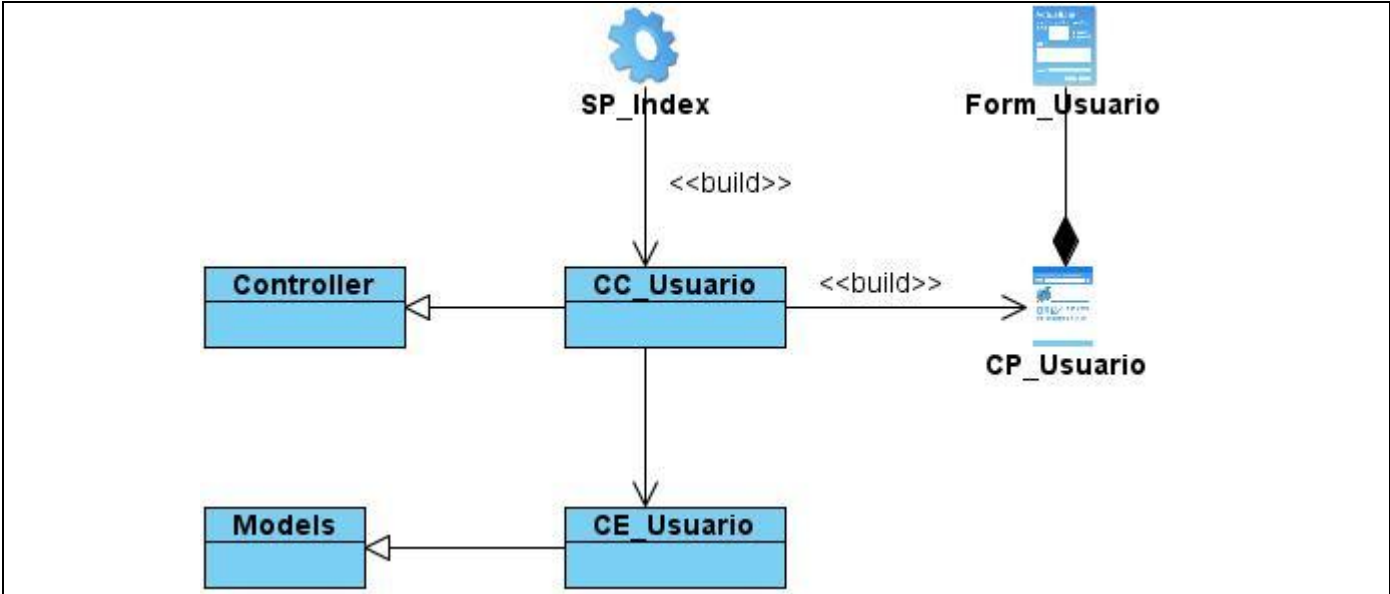


Figura 15: Diagrama de clase del diseño CU: Gestionar usuario.

3.2.3.3. CU: Gestionar estudiante interno.

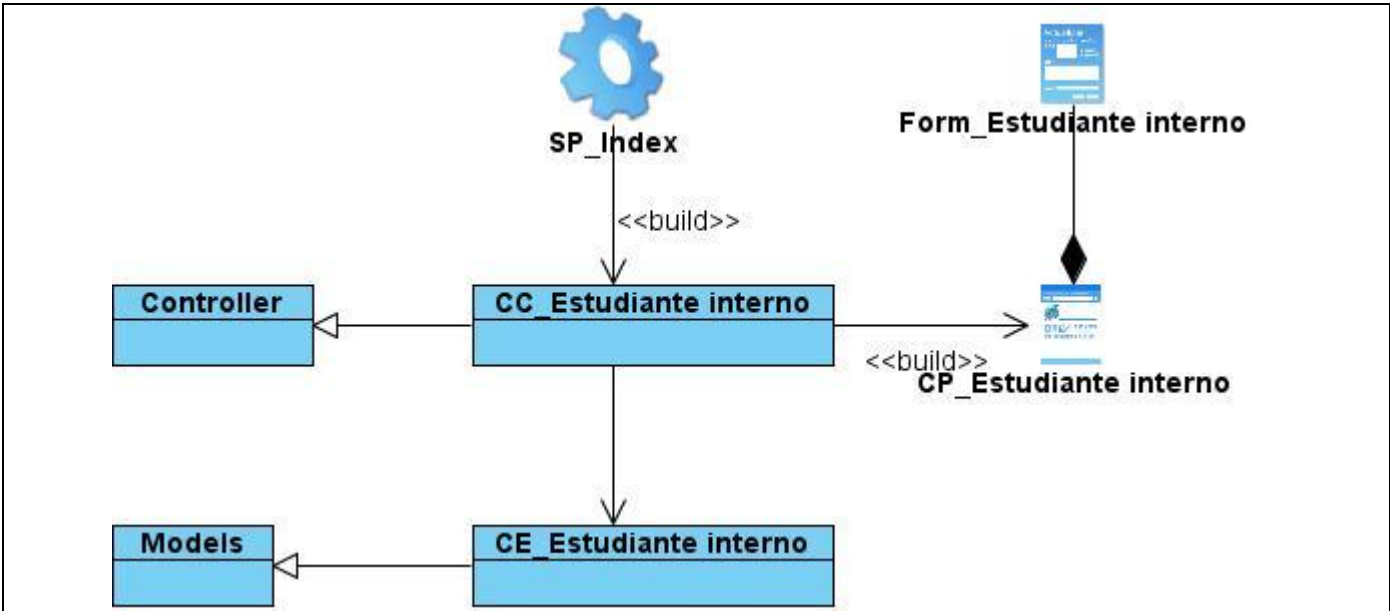


Figura 16: Diagrama de clase del diseño CU: Gestionar estudiante interno.

3.2.3.4. CU: Gestionar estudiante externo.

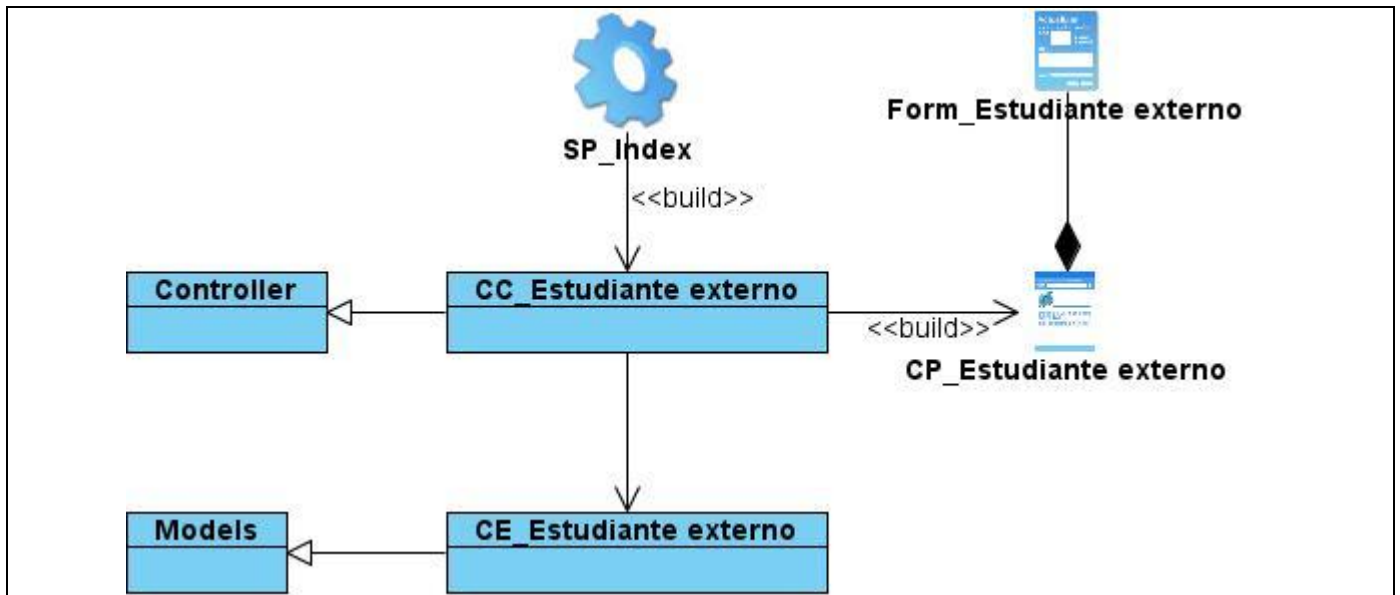


Figura 17: Diagrama de clase del diseño CU: Gestionar estudiante externo.

3.2.3.5. CU: Gestionar solicitud de traslado interna.

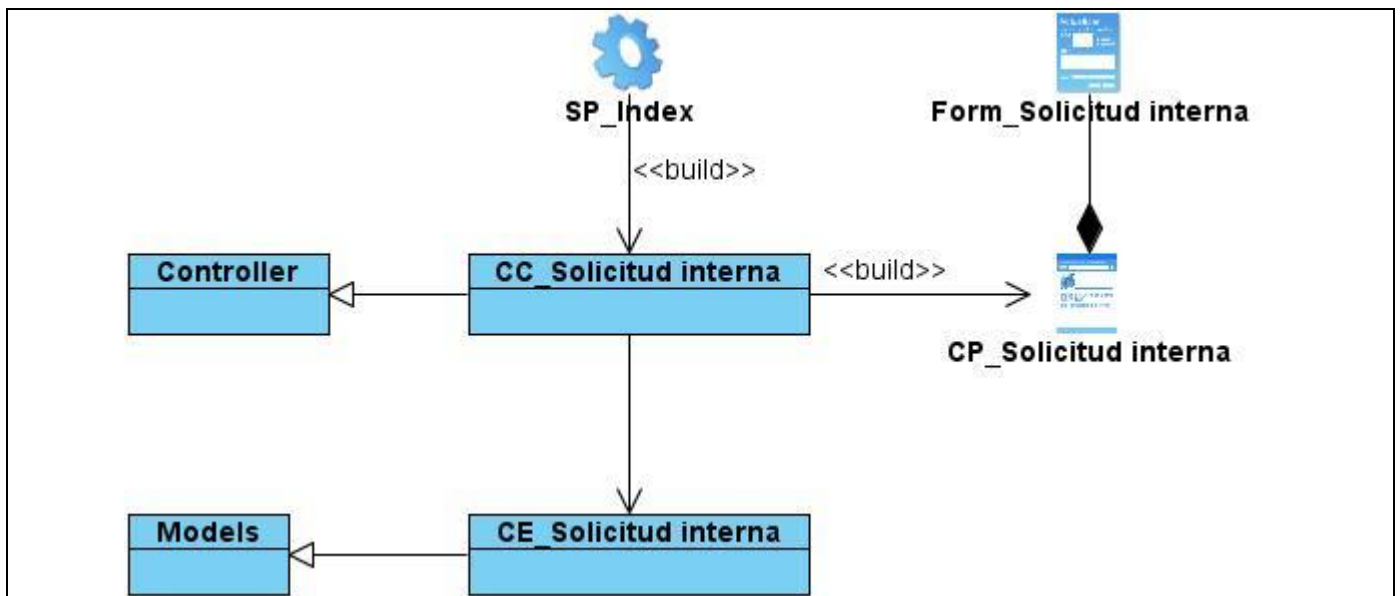


Figura 18: Diagrama de clase del diseño CU: Gestionar solicitud de traslado interna.

3.2.3.6. CU: Gestionar solicitud de traslado externa.

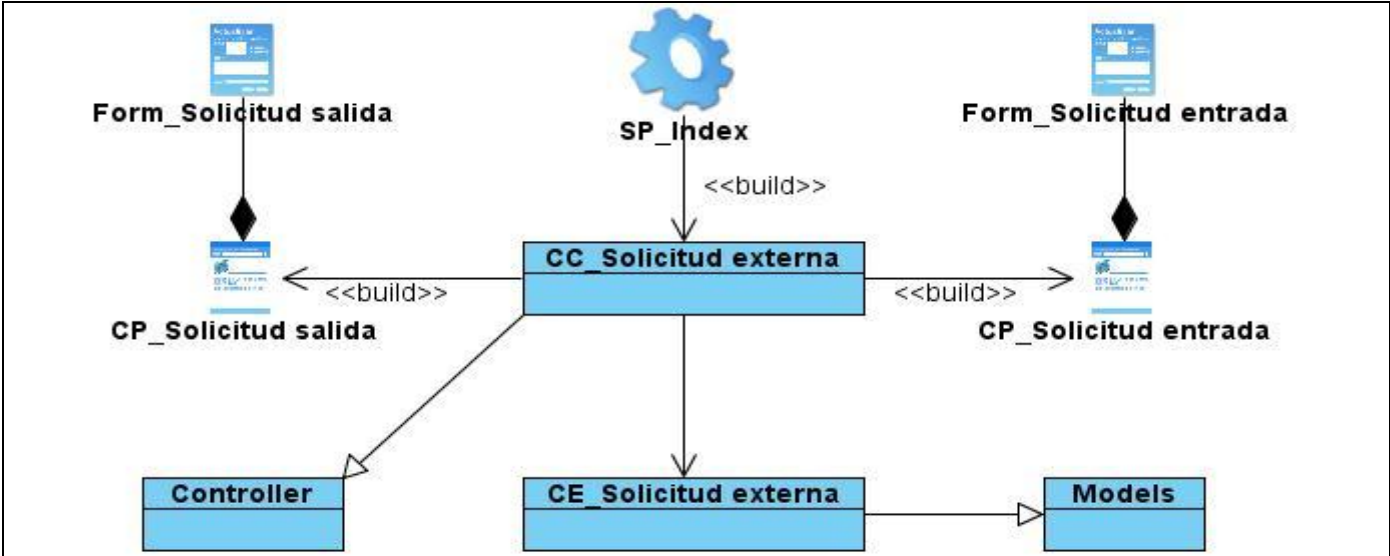


Figura 19: Diagrama de clase del diseño CU: Gestionar solicitud de traslado externa.

3.2.3.7. CU: Gestionar resolución.

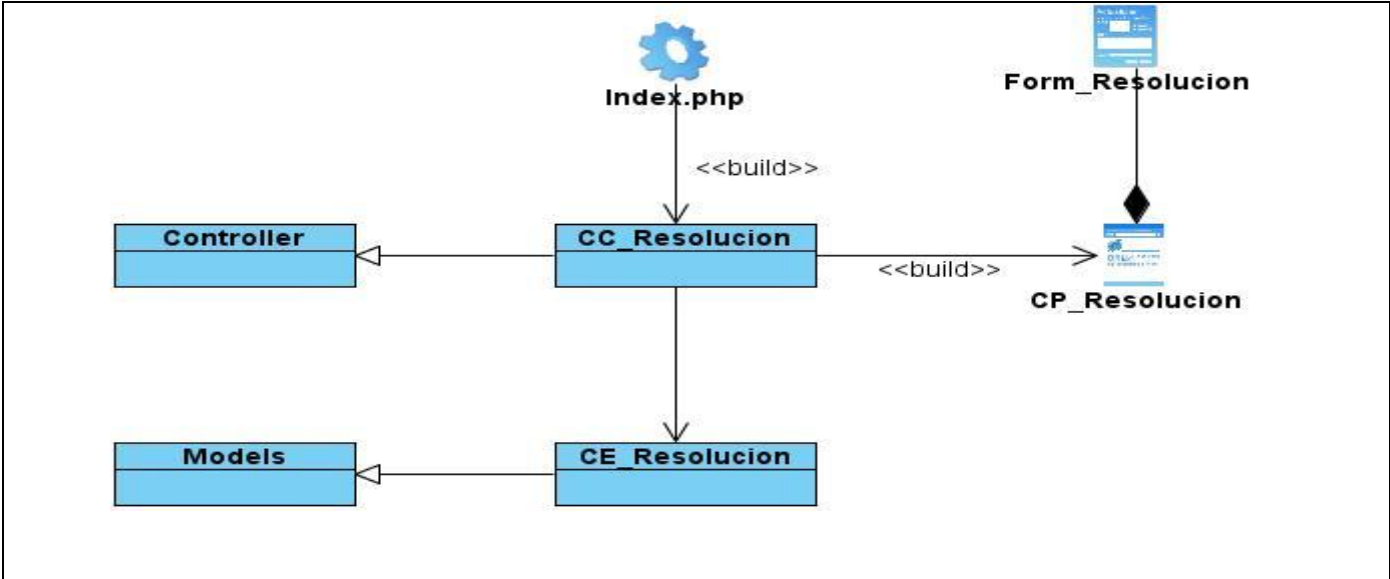


Figura 20: Diagrama de clase del diseño CU: Gestionar resolución.

3.2.3.8. CU: Registrar expediente.

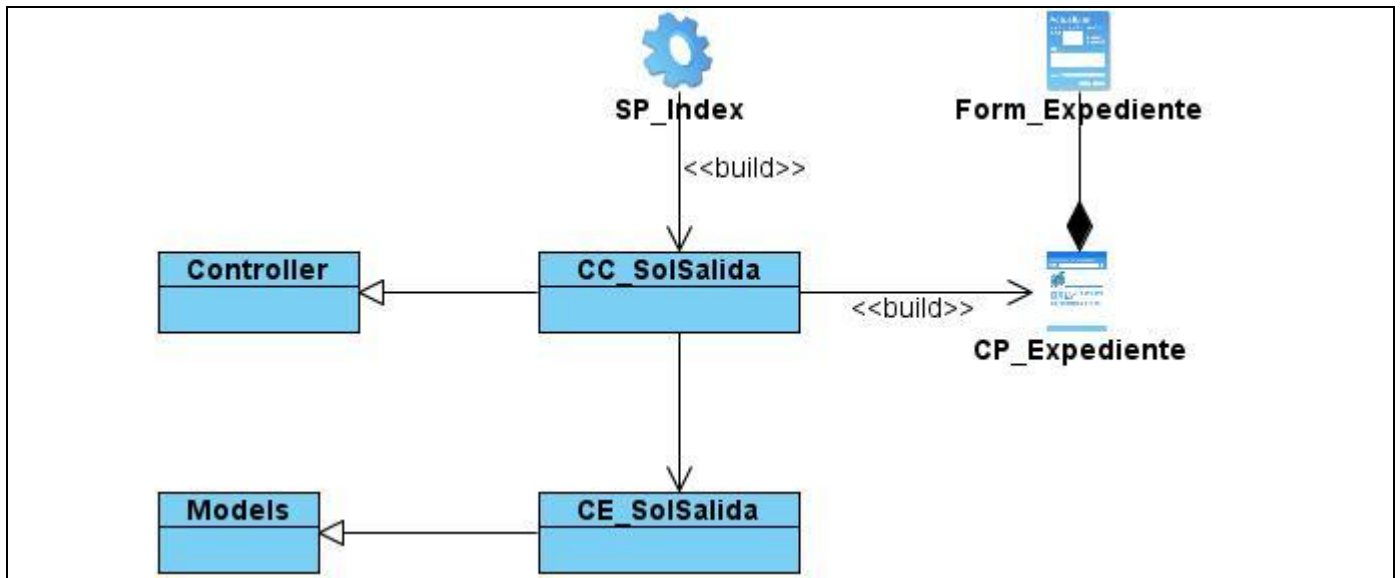


Figura 21: Diagrama de clase del diseño CU: Registrar expediente.

3.2.3.9. CU: Generar reportes.

El siguiente diagrama muestra las clases del diseño generado por el reporte Solicitudes Aprobadas, de igual forma los reportes solicitudes no aprobadas, solicitudes por carrera, solicitudes por curso, solicitudes por facultad y solicitudes por universidad generan un diagrama parecido al mostrado solo cambia el nombre de las clases generadas.

Solicitudes no aprobadas (SP_Index, CC_SolNoAprobadas, CP_SolNoAprobadas, Form_SolNoAprobadas, SolNoAprobadas, CE_Solicitudes).

Solicitudes por carrera (SP_Index, CC_SolCarrera, CP_SolCarrera, Form_SolCarrera, SolCarrera, CE_Solicitud).

Solicitudes por curso (SP_Index, CC_SolCurso, CP_SolCurso, Form_SolCurso, SolCurso, CE_Solicitud).

Solicitudes por facultad (SP_Index, CC_SolFacultad, CP_SolFacultad, Form_SolFacultad, SolFacultad, CE_Solicitud).

Solicitudes por universidad (SP_Index, CC_SolUniversidad, CP_SolUniversidad, Form_SolUniversidad, SolUniversidad, CE_Solicitud).

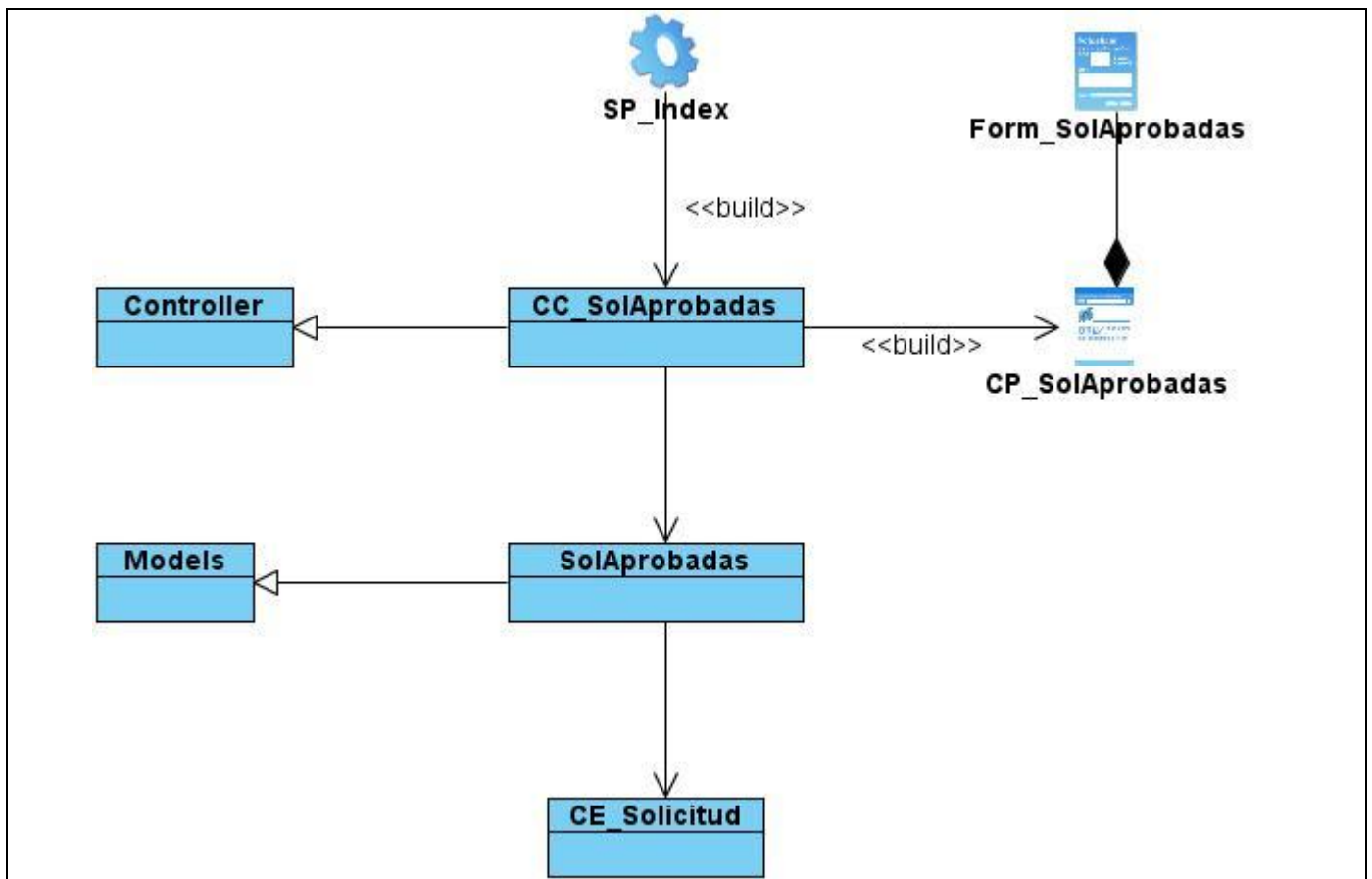


Figura 22: Diagrama de clase del diseño CU: Generar reportes.

3.2.4. Diseño de la Base de Datos.

3.2.4.1. Diagrama Entidad-Relación de la Base de Datos.

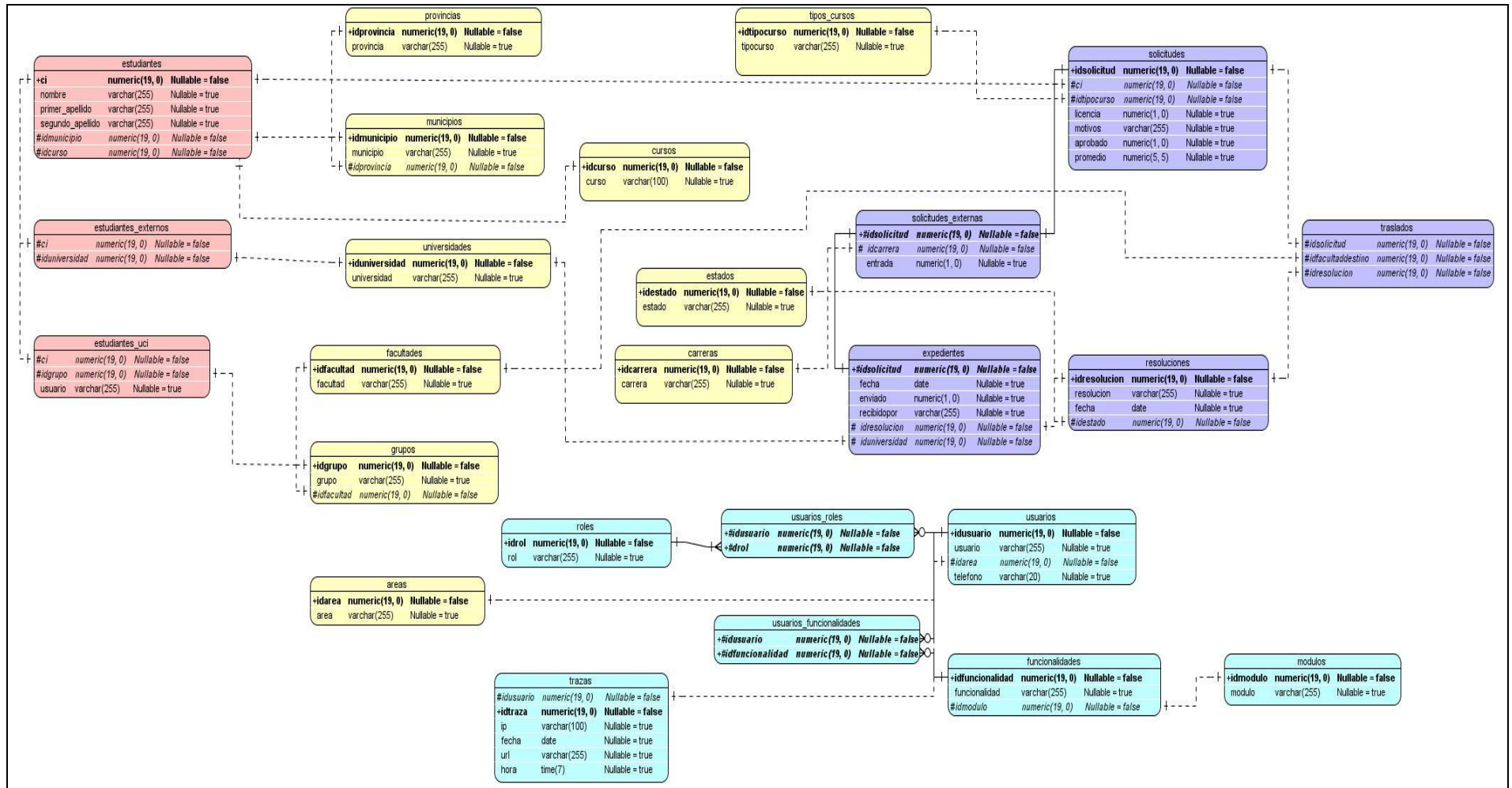


Figura 23: Diagrama Entidad-Relación.

3.2.4.2. Diagramas de interacción.

Los diagramas de interacción son otro artefacto generado dentro del flujo de trabajo de diseño, estos explican gráficamente las interacciones existentes entre las instancias de las clases y sus modelos. El UML define dos tipos de estos diagramas; ambos sirven para expresar interacciones semejantes o idénticas de mensaje:

- Diagramas de colaboración.
- Diagramas de secuencia.

Los Diagramas de colaboración describen las interacciones entre los objetos en un formato de grafo o red y los diagramas de secuencia describen las interacciones en una especie de formato de cerca o muro, que a su vez muestra en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema.

En el [anexo #1](#) se muestran los diagramas de secuencia que se realizaron.

3.3. Conclusiones parciales.

En el presente capítulo se muestra el diseño detallado del sistema, se explica el patrón utilizado para el diseño, se describen las clases utilizadas y la relaciones entre ellas, se muestran los diagramas de clases del diseño, se muestra los diagramas de secuencia y se hace una descripción de la BD para el sistema.

Capítulo 4 “Implementación y prueba del sistema”.

4.1. Introducción.

En este capítulo se especifican los componentes del sistema, se describe la estructura del mismo y se ejemplifica mediante el diagrama de despliegue. Se presentan los casos de prueba necesarios para probar el éxito del sistema; se describe cada caso de prueba, mediante el método de caja negra que sigue tres pasos principales para efectuar las pruebas.

4.2. Estructura del modelo de implementación.

4.2.1. Diagrama de componentes.

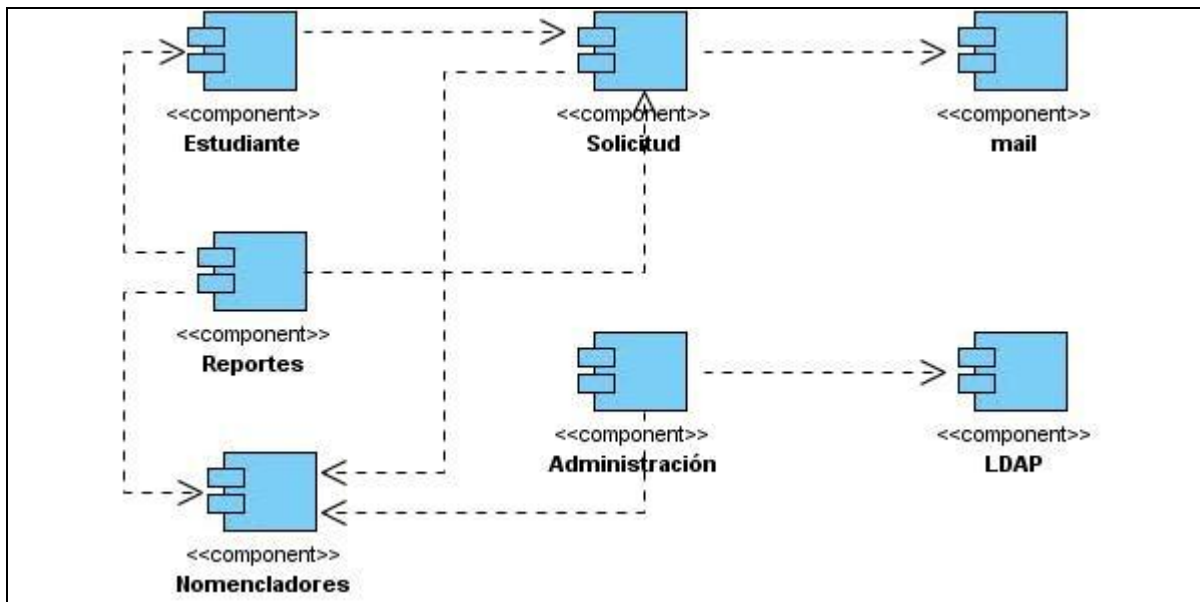


Figura 24: Diagrama de componentes.

4.2.2. Diagrama de despliegue.

El siguiente diagrama de despliegue muestra la forma en que estará distribuido el sistema y los elementos necesarios para que funcione correctamente. Tiene que existir una PC cliente con un navegador instalado ya sea Internet Explorer o Mozilla Firefox, necesario para levantar la aplicación mediante el protocolo http, que se conectará al servidor donde el sistema estará alojado y cuyas condiciones deben ser las siguientes: tener 2GB de memoria RAM como mínimo, y tener instalado el AppServ 2.5.10 y PostgreSQL 8.3.4.

El servidor del sistema se conecta mediante el protocolo LDAP al servidor de la Universidad para obtener los datos de los trabajadores y estudiantes de la UCI con los que se trabajará en el sistema. También la PC cliente debe contener una impresora conectada por el puerto USB para imprimir cualquier información necesaria.

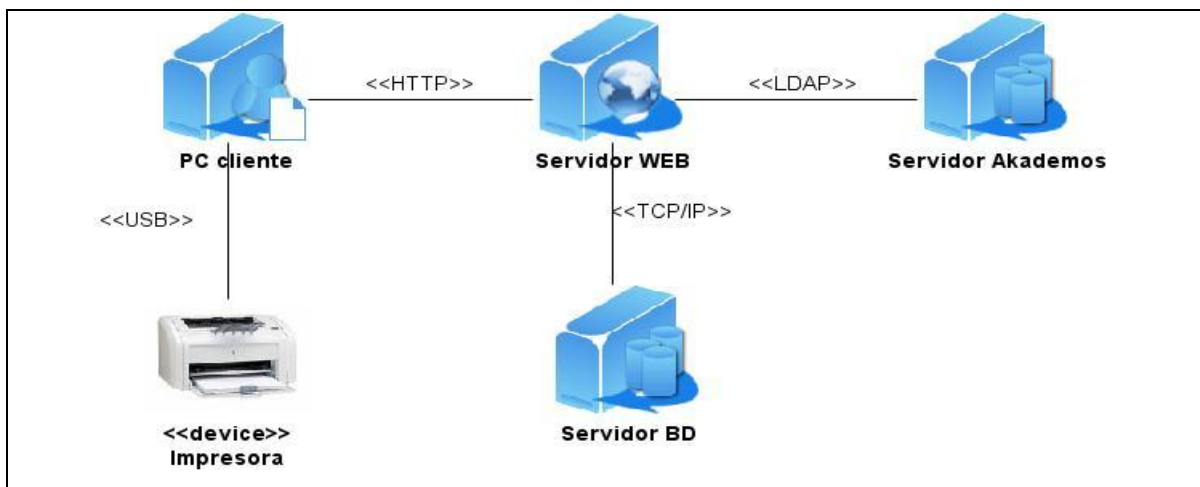


Figura 25: Diagrama de despliegue.

4.3. Prueba.

Para realizar las pruebas del sistema se realizaron pruebas de cajas negras basado en un método de tres pasos para obtener un conjunto de casos de prueba del sistema a partir de los CU; los pasos son los siguientes:

Paso	Descripción	Resultado
1	Generar escenarios de uso	Obtener los posibles caminos de ejecución de cada caso de uso. Cada camino es un escenario de uso.
2	Identificar casos de prueba	Conjunto de casos de prueba a partir de los escenarios anteriores.
3	Identificar los valores a probar	Valores de prueba asociados a cada caso de prueba anterior.

Tabla 58: Pasos para obtener los casos de pruebas.

Para generar los casos de prueba primero se generan todos los posibles escenarios, o caminos de ejecución, de cada caso de uso. Después, se identifican los casos de prueba a partir de esos escenarios y por último se identifican los valores a probar de cada caso de prueba.

Lo más importante de los CU para generar los casos de prueba son los caminos de ejecución. Este camino se divide en dos: en el camino principal o secuencia normal y en el camino alterno. El camino principal son los pasos que sigue el sistema siempre y cuando no surja algún imprevisto, mientras que el camino alterno son las variaciones que van surgiendo durante el camino principal a causa de los errores. Cada uno de estos caminos constituye el escenario de casos de uso.

Un caso de prueba es el conjunto de entradas de datos, las condiciones para la ejecución y los resultados esperados. Esto se hace con el propósito de identificar y comunicar las condiciones que se llevarán a cabo en la prueba. Los casos de prueba son necesarios para verificar el éxito de la aplicación y la aceptabilidad del producto.

Para generar los casos de prueba se seguirán los siguientes pasos:

1. Para cada CU, generar un sistema completo de los escenarios.
2. Para cada escenario, identificar los casos de pruebas y las condiciones que hagan que se ejecute.
3. Para cada caso de prueba, identificar los valores de los datos con los cuales se harán las pruebas.

Las celdas de la tabla contienen V o I. V indica válido, I indica inválido.

CP: Casos de pruebas

4.3.1. CU: Autenticar usuario.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Autenticación exitosa.	Flujo normal	
2	Datos incorrectos.	Flujo normal	Flujo alternativo 1
3	Usuario no registrado.	Flujo normal	Flujo alternativo 2

Tabla 59: Escenarios del CU.

ID	Escenario	Usuario	Contraseña	Resultado esperado
CP1	1	V	V	Accede al sistema, mostrando las funcionalidades debidas, dependiendo del tipo de usuario identificado.
CP2	2	I	V	Mensaje de error: Acceso denegado, vuelve a la página que permite identificarse.
CP3	2	V	I	Mensaje de error: Acceso denegado, vuelve a la página que permite identificarse.
CP4	3	V	V	Mensaje de error: usuario no registrado, para acceder al sistema es necesario estar registrado, póngase en contacto con el administrador del sistema, vuelve a la página que permite identificarse.

Tabla 60: Casos de pruebas.

ID	Escenario	Usuario	Contraseña	Resultado esperado
CP1	1	Usuario UCI,	Password UCI.	Accede al sistema, mostrando las funcionalidades debidas, dependiendo del tipo de usuario identificado.
CP2	2	Usuario UCI mal escrito	Password UCI.	Mensaje de error: Acceso denegado, vuelve a la página que permite identificarse.
CP3	2	Usuario UCI	Password UCI incorrecto.	Mensaje de error: Acceso denegado, vuelve a la página que permite identificarse.
CP4	3	Usuario UCI (no registrado en el sistema)	Password UCI	Mensaje de error: usuario no registrado, para acceder al sistema es necesario estar registrado, póngase en contacto con el administrador del sistema, vuelve a la página que permite identificarse.

Tabla 61: Casos de prueba con los valores de los datos.

4.3.2. CU: Gestionar usuario.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar usuario.	Flujo normal	
2	Modificar usuario.	Flujo normal	
3	Eliminar usuario.	Flujo normal	

Tabla 62: Escenarios del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Si el proceso fue correcto se inserta los datos en la BD del sistema y queda registrado el usuario que trabajará con el sistema con el rol asignado y podrá acceder a las funcionalidades que le son permitidas.
CP2	1	I	Mensaje de error, no se pueden registrar los datos, todos los campos deben ser llenados.
CP3	2	V	Se actualiza los datos en la BD del sistema.
CP4	3	V	Se elimina el usuario de la BD del sistema.

Tabla 63: Casos de pruebas.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se registran los datos correctos. Usuario: Usuario UCI. Rol: Se le asigna el rol al usuario. Área: Se registra el área dónde radia el usuario. Teléfono.	Si el proceso fue correcto se inserta los datos en la BD del sistema y queda registrado el usuario que trabajará con el sistema, con el rol asignado y podrá acceder a las funcionalidades que le son permitidas.
CP2	1	Cuando se van a registrar los datos, que se deje algún campo vacío.	Mensaje de error, no se pueden registrar los datos, todos los campos deben ser llenados.
CP3	2	Cambiar el dato de algún usuario registrado.	Se actualiza los datos en la BD del sistema.
CP4	3	Eliminar los datos de algún usuario registrado.	Se elimina el usuario de la BD del sistema.

Tabla 64: Casos de prueba con los valores de los datos.

4.3.3. CU: Gestionar estudiante interno.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Buscar estudiante	Flujo normal	Flujo Alterno 1
2	Registrar estudiante	Flujo normal	
3	Modificar estudiante.	Flujo normal	
4	Eliminar estudiante.	Flujo normal	

Tabla 65: Escenarios del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Muestra un formulario con los datos sacados de la BD de la universidad.
CP2	1	F	Mensaje de error, dato incorrecto, vuelve a la página del escenario 1.
CP3	2	V	Se guarda en la BD del sistema los datos del estudiante.
CP4	3	V	Se actualiza en la BD del sistema los datos que fueron cambiados.
CP5	4	V	Se elimina de la BD el estudiante seleccionado.

Tabla 66: Casos de pruebas para el CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se entra el CI del estudiante.	Muestra un formulario con los datos sacados de la BD de la universidad.
CP2	1	Que el CI entrado esté incorrecto o que le falten dígitos.	Mensaje de error, dato incorrecto, vuelve a la página del escenario 1
CP4	2	Los datos del formulario fueron llenados con la búsqueda efectuada automáticamente.	Se guarda en la BD del sistema los datos del estudiante.
CP4	3	Se cambia algún dato que se encuentre incorrecto o que deba ser cambiado.	Se actualiza en la BD del sistema los datos que fueron cambiados.
CP5	4	Se elimina algún estudiante registrado.	Se elimina de la BD el estudiante seleccionado.

Tabla 67: Casos de prueba con los valores de los datos.

4.3.4. CU: Gestionar estudiante externo.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar estudiante.	Flujo normal	
2	Modificar estudiante.	Flujo normal	
3	Eliminar estudiante.	Flujo normal	

Tabla 68: Escenarios del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se realiza el proceso y se guarda en la BD del sistema los datos entrados.
CP2	1	I	Mensaje de error, campos vacíos.
CP3	2	V	Se actualizan los datos en la BD del sistema.
CP4	3	V	Se elimina de la BD el estudiante seleccionado.

Tabla 69: Casos de pruebas para el CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se entran los datos del estudiante correctamente.	Se realiza el proceso y se guarda en la BD del sistema los datos entrados.
CP2	1	Se deja algún campo vacío.	Mensaje de error, campos vacíos.
CP3	2	Se cambian los datos de algún estudiante.	Se actualizan los datos en la BD del sistema.
CP4	3	Se elimina el estudiante registrado.	Se elimina de la BD el estudiante seleccionado.

Tabla 70: Casos de prueba con los valores de los datos.

4.3.5. CU: Gestionar solicitud interna.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar solicitud.	Flujo normal	
2	Modificar solicitud.	Flujo normal	
3	Eliminar solicitud.	Flujo normal	
4	Aprobar solicitud.	Flujo normal	Flujo alterno 1

Tabla 71: Escenarios del CU.

ID	Escenario	Datos.	Correo enviado.	Resultado esperado
CP1	1	V	V	Se guardan los datos en la BD y se genera un mensaje informando la operación efectuada.
CP2	1	I	I	Mensaje de error, vuelve a la página para entrar los datos de nuevo.
CP3	2	V	--	Se aprueba la solicitud y se guarda aprobada en la BD.
CP4	3	V	--	Se anula la solicitud y se guarda como anulada en la BD.
CP5	4	V	V	Se guardan los datos en la BD y se genera un mensaje informando la operación efectuada.
CP6	4	I	V	Se guardan los datos en la BD y se genera un mensaje informando la operación efectuada.

Tabla 72: Casos de pruebas para el CU.

ID	Escenario	Datos.	Correo enviado.	Resultado esperado
CP1	1	Se introducen los datos correctos.	El sistema envía un correo a la SD de la facultad para la cual se destina la solicitud.	Se guardan los datos en la BD y se genera un mensaje informando la operación.
CP2	1	Se dejan campos vacíos.	No envía correo.	Mensaje de error, vuelve a la página para entrar los datos de nuevo.
CP3	2	Se cambia los datos de la solicitud.	--	Se actualiza en la BD los nuevos datos de la solicitud.
CP4	3	Se selecciona la solicitud a eliminar.	--	Se elimina en la BD del sistema la solicitud seleccionada.
CP5	4	Se aprueba la solicitud.	El sistema envía un correo a la SD de la	Se guardan los datos en la BD y se genera un mensaje

			facultad que pidió la solicitud informando la aprobación.	informando la operación.
CP6	4	No se aprueba la solicitud.	El sistema envía un correo a la SD de la facultad que pidió la solicitud informando la no aprobación.	Se guardan los datos en la BD y se genera un mensaje informando la operación.

Tabla 73: Casos de prueba con los valores de los datos.

4.3.6. CU: Gestionar solicitud externa.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar solicitud.	Flujo normal	
2	Modificar solicitud.	Flujo normal	
3	Eliminar solicitud.	Flujo normal	

Tabla 74: Escenarios del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se guardan los datos en la BD.
CP2	1	I	Mensaje de error, campos vacíos o datos incorrectos.
CP3	2	V	Se actualizan los datos en la BD.
CP4	3	V	Se elimina la solicitud seleccionada.

Tabla 75: Casos de pruebas para el CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se introducen los datos correctamente.	Se guardan los datos en la BD.
CP2	1	Se dejan campos vacíos.	Mensaje de error, campos vacíos o datos incorrectos.
CP3	2	Se cambian datos necesarios.	Se actualizan los datos en la BD.
CP4	4	Se selecciona la solicitud a eliminar.	Se elimina la solicitud seleccionada.

Tabla 76: Casos de prueba con los valores de los datos.

4.3.7. CU: Gestionar resolución.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar resolución.	Flujo normal.	
2	Modificar resolución.	Flujo normal.	
3	Eliminar resolución.	Flujo normal.	

Tabla 77: Escenario del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se guardan los datos en la BD.
CP2	1	I	Mensaje de error, campos vacíos o datos incorrectos.
CP3	2	V	Se actualizan los datos en la BD.
CP4	3	V	Se elimina la resolución seleccionada.

Tabla 78: Casos de pruebas para el CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se introducen los datos correctamente.	Se guardan los datos en la BD.
CP2	1	Se dejan campos vacíos.	Mensaje de error, campos vacíos o datos incorrectos.
CP3	2	Se cambian datos necesarios.	Se actualizan los datos en la BD.
CP4	4	Se selecciona la resolución a eliminar.	Se elimina la resolución seleccionada.

Tabla 79: Casos de prueba con los valores de los datos.

4.3.8. CU: Registrar expediente.

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Registrar expediente.	Flujo normal	

Tabla 80: Escenario del CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se guardan los datos en la BD.

Tabla 81: Casos de pruebas para el CU.

ID	Escenario	Datos	Resultado esperado
CP1	1	Se introducen los datos del expediente.	Se guardan los datos en la BD.

Tabla 82: Casos de prueba con los valores de los datos.

4.4. Conclusiones parciales.

En el presente capítulo se muestra como queda el sistema expresado en componentes de implementación, se muestra el diagrama de despliegue y se describen las pruebas realizadas para verificar que el sistema cumple con las funcionalidades previstas.

Conclusiones generales.

Una vez finalizado el trabajo se plantean las siguientes conclusiones:

1. Se informatizó el proceso de gestión de traslado para los estudiantes de la UCI con las ventajas que esto conlleva para el trabajo del personal de Secretaría General y Secretaría Docente.
2. El mismo cumple con todos los requerimientos solicitados por el área de Secretaría General y Secretaría Docente.
3. El sistema está siendo utilizado en el presente curso.
4. Se realizó el diseño del sistema completo obteniendo las clases y las relaciones entre las mismas.

Con la puesta en práctica del sistema de gestión de traslado para Secretaría General se logra una mayor organización en el manejo de las informaciones y datos, se logra un aumento en la calidad de trabajo y se tiene un mejor control de los traslados efectuados en un curso escolar.

Recomendaciones.

Continuar en la investigación del sistema con la finalidad de aportar nuevas funcionalidades.

Incluir en el sistema la gestión de las bajas académicas de la UCI que son reinsertadas en otro CES del país.

Que el sistema permita tomar los datos de akademos no solo de los matriculados sino de los que estén en otros estados como licencia de Matrícula y bajas.

Referencias bibliográficas.

- Universidad de Málaga. Métodos generales de la investigación científica. [En línea][Citado el: 14 de abril de 2009]
<http://www.eumed.net/libros/2007c/306/METODOS%20GENERALES%20DE%20LA%20INVESTIGACION%20CIENTIFICA.htm>
- Visual Paradigm. [En línea] 2004. [Citado el: 20 de diciembre de 2007.] <http://www.visual-paradigm.com/product/vpuml/>.
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000. 84-7829-036-2.
- Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código.[En línea]2003[Citado el: 02 de abril de 2009]
<http://www.desarrolloweb.com/articulos/1178.php>,
<http://www.maestrosdelweb.com/editorial/zendstudio/>
- Introducción a los lenguajes del web.[En línea][Citado el: 02 de abril de 2009]
<http://www.desarrolloweb.com/manuales/27/>
- Patrón “Modelo-Vista-Controlador”. [En línea]. [Citado: 4 de mayo de 2009] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
- ProgramacionWeb.net [En línea]. [2003-2009]. [Citado: 4 de mayo de 2009]
<http://www.programacionweb.net/articulos/articulo/?num=505>
- Desarrollo, Javascript, Tutoriales.[En línea][2008][Citado: 4 de mayo de 2009]
<http://www.crysfel.com/index.php/2008/03/24/introduccion-a-ext-js/>
- <http://www.extjs.com/>

Bibliografía consultada.

- Biblioteca. [En línea] UCI. <http://biblioteca.uci.cu>.
- <http://bibliodoc.uci.cu/pdf/reg02138.pdf>. Desarrollo Web con PHP y MySQL. José Antonio Gallego Vázquez. 2003
- <http://bibliodoc.uci.cu/pdf/reg03050.pdf>. El Lenguaje Unificado de Modelado. Manual de Referencia. James Rumbaugh, Ivar Jacobson, Grady Booch
- <http://bibliodoc.uci.cu/pdf/reg00060.pdf>. El Proceso Unificado de Desarrollo de Software. James Rumbaugh, Ivar Jacobson, Grady Booch. 2000
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. Madrid: Addison Wesley, 2000. 84-7829-036-2.
- Larman, Craig. *UML Y PATRONES Introducción al análisis y diseño orientado a objetos*. México: PRENTICE HALL, 1999. 970-17-0261-1.
- Universidad de las Ciencias Informáticas. Entorno Virtual de Aprendizaje. *Ingeniería del Software I y II*. [En línea] UCI. teleformacion.uci.cu.
- Equipo de desarrollo de PostgreSQL. Tutorial de PostgreSQL. Editado por Thomas Lockhart, 1996-9.
- Equipo de desarrollo de PostgreSQL. Manual del usuario de PostgreSQL. Editado por Thomas Lockhart, 1996-9.
- Frederick, Shea, Ramsay, Colin, Blades, Steve. Learning Ext JS. 2008. 978-1-847195-14-2.
- Upton, David. CodeIgniter for Rapid PHP Application Development. 2007. 978-1-847191-74-8

Anexos.

Anexo # 1: Diagramas de secuencias.

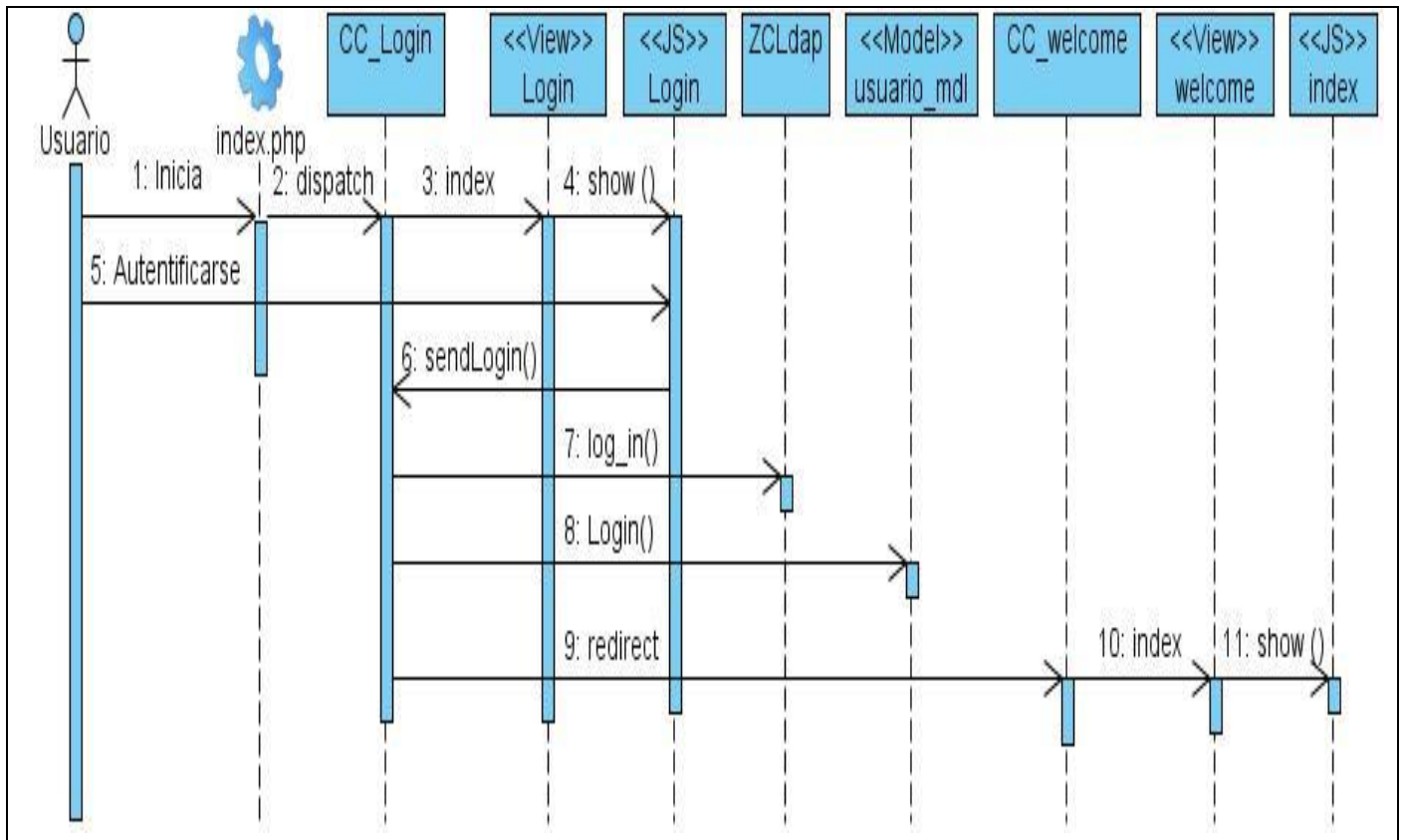


Figura 26: Diagrama de secuencia CU: Autenticar usuario.

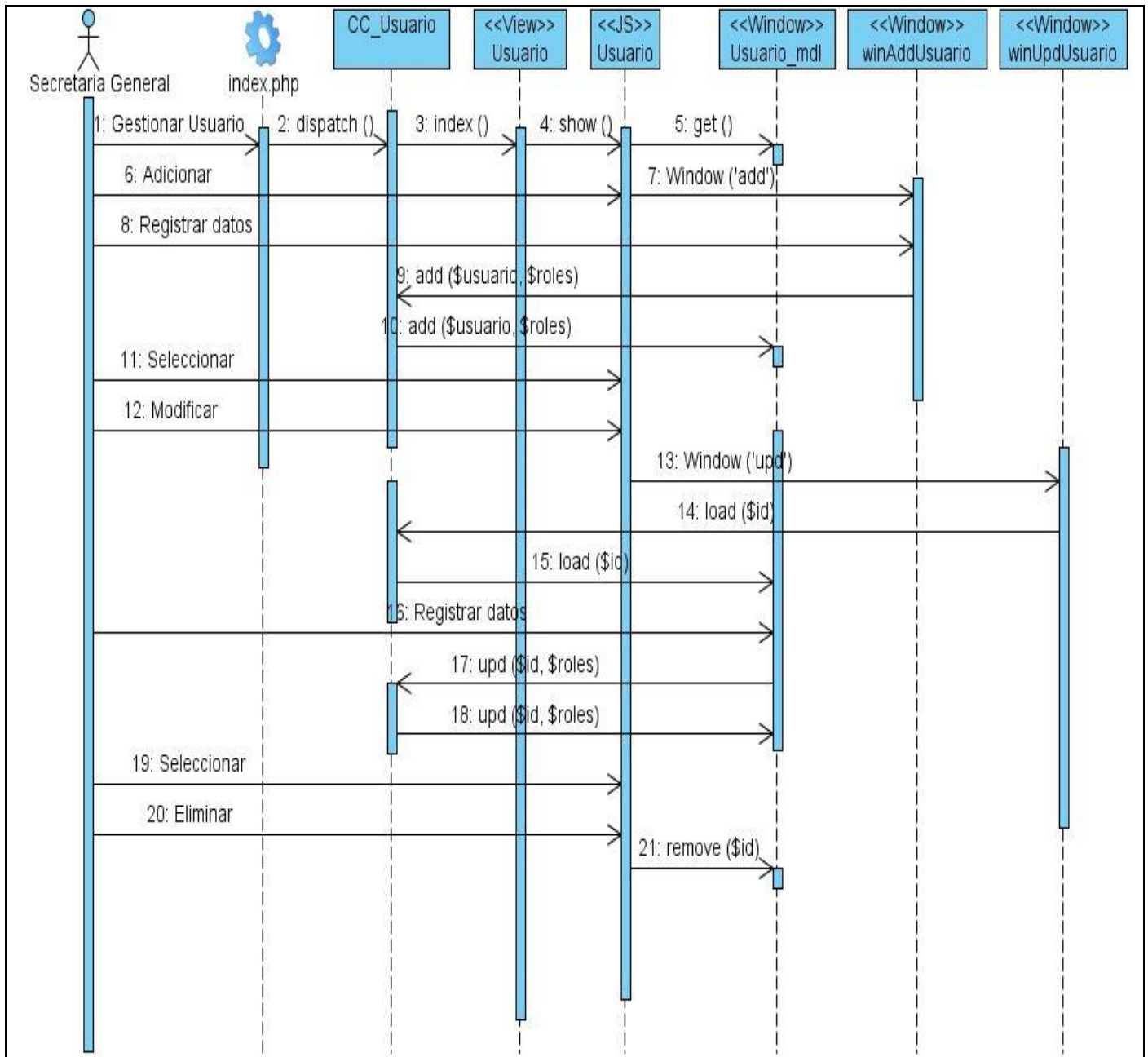


Figura 27: Diagrama de secuencia CU: Gestionar usuario.

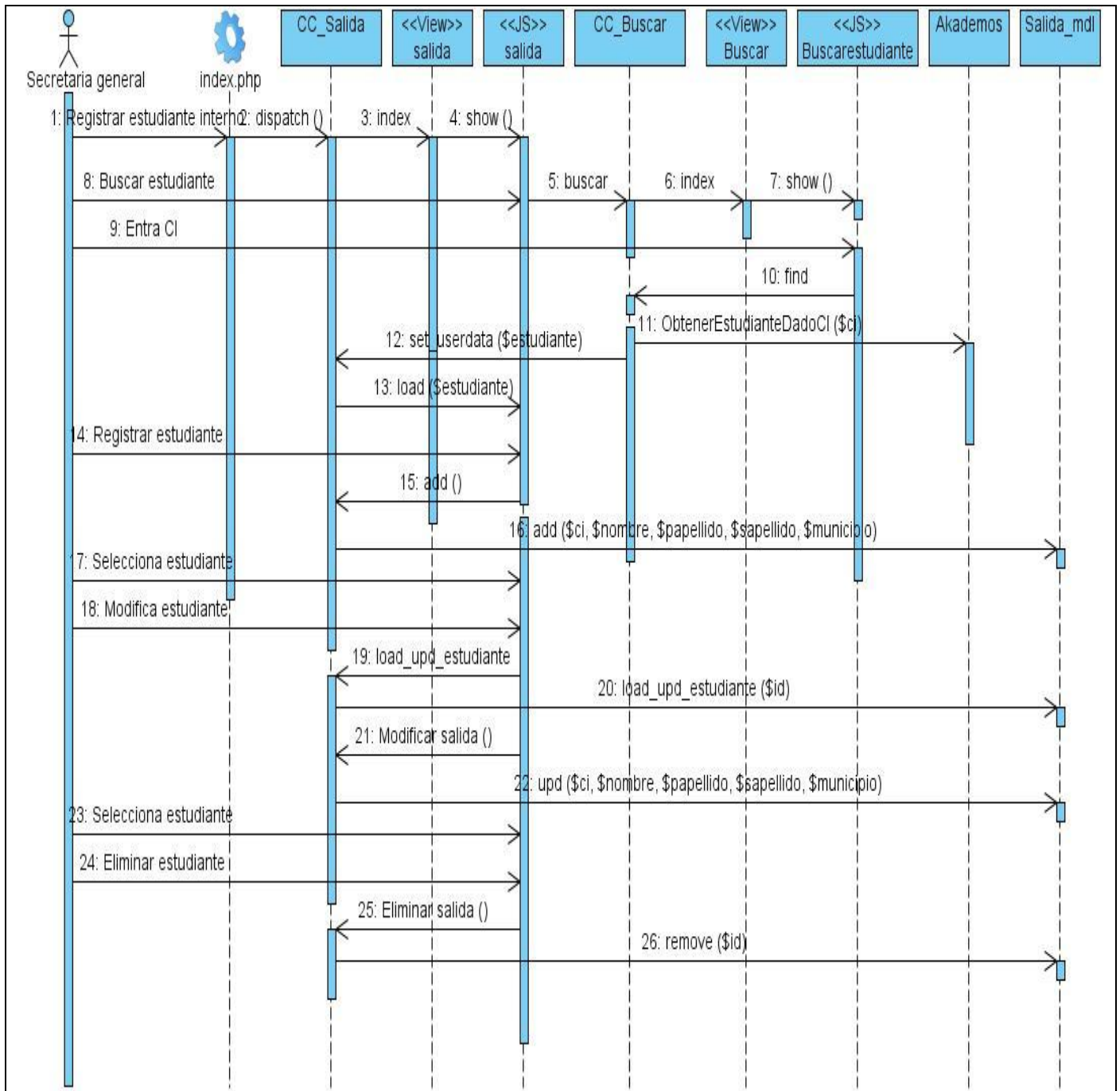


Figura 28: Diagrama de secuencia CU: Gestionar estudiante interno.

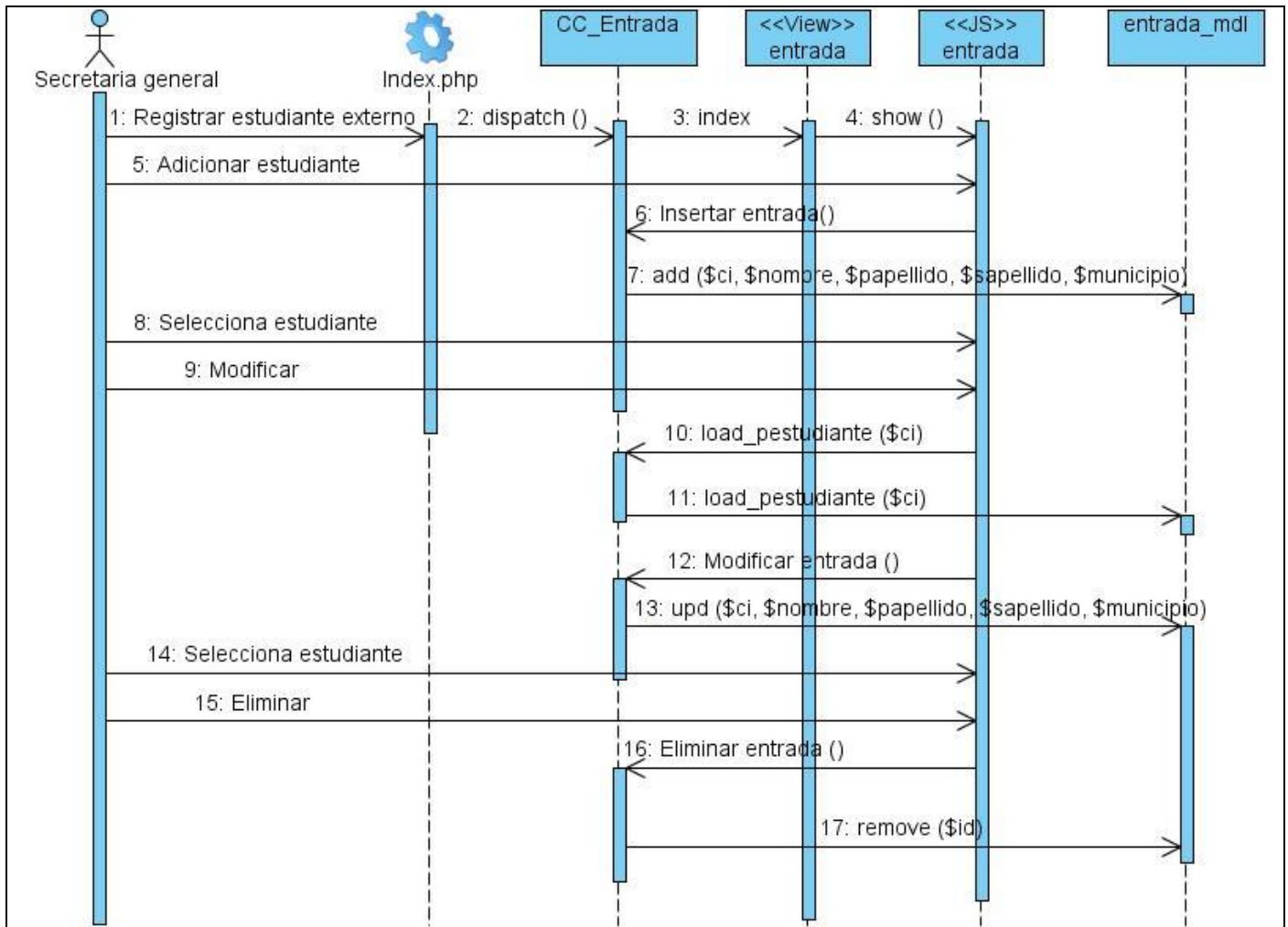


Figura 29: Diagrama de secuencia CU: Gestionar estudiante externo.

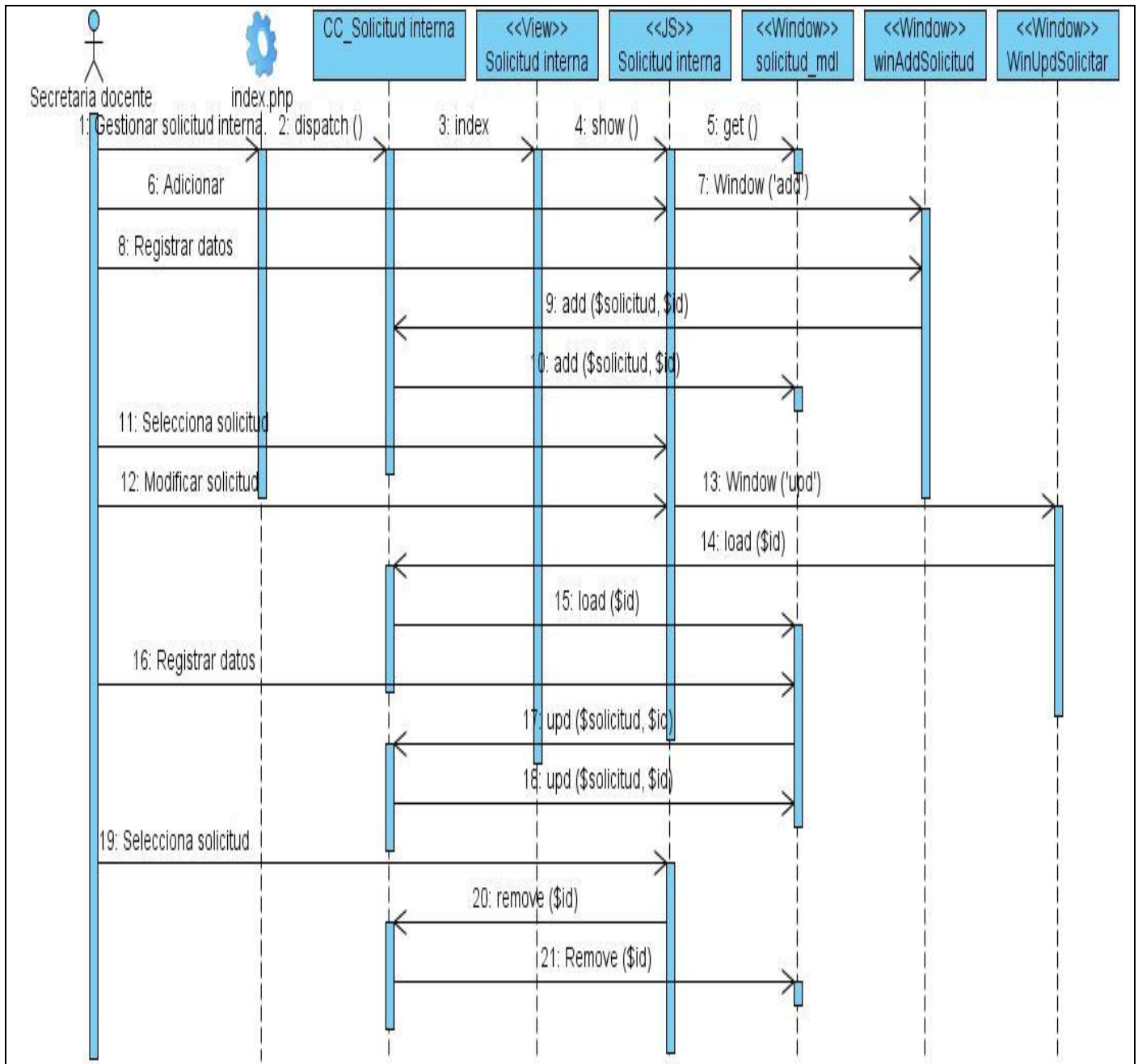


Figura 30: Diagrama de secuencia CU: Gestionar solicitud interna.

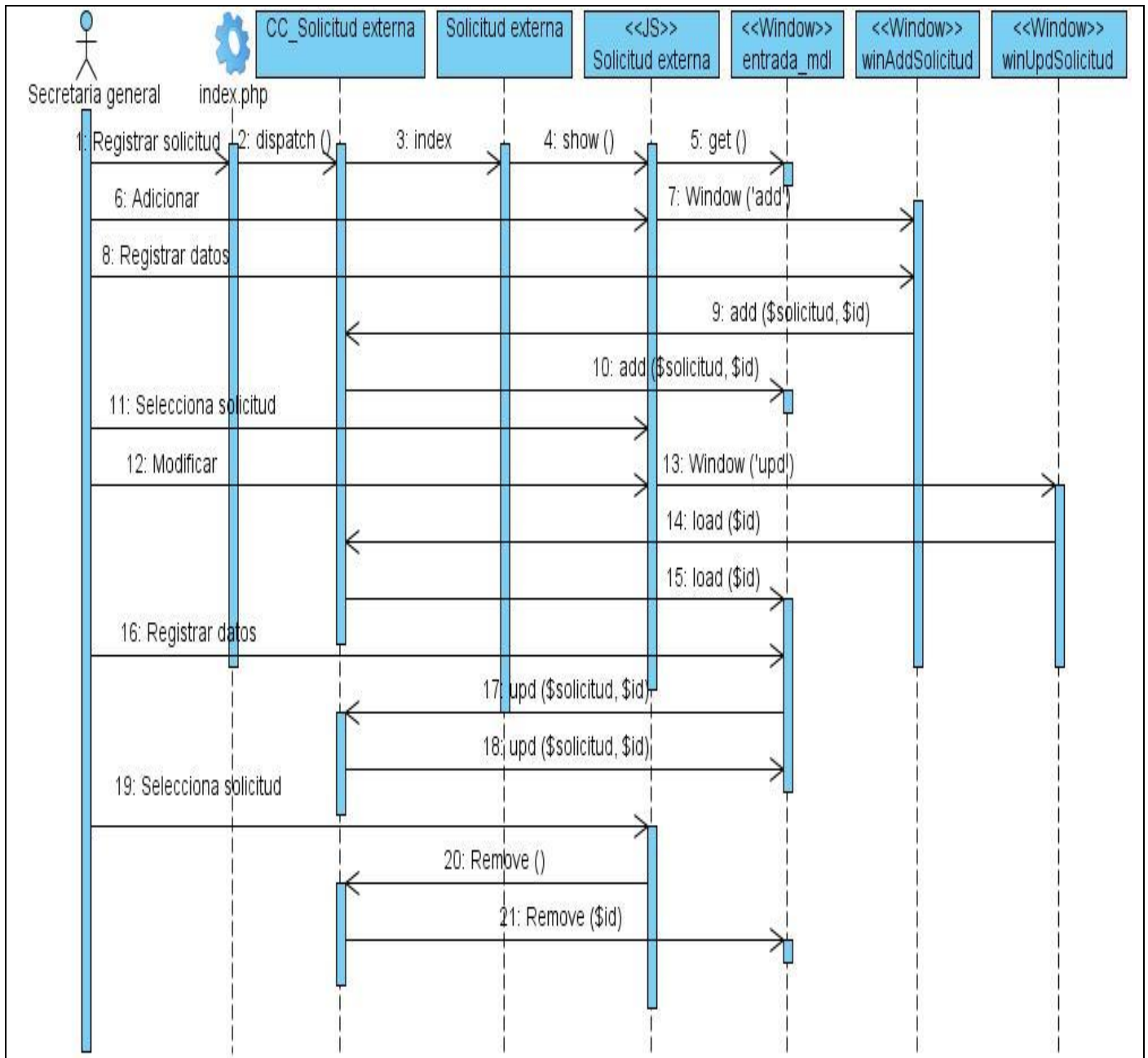


Figura 31: Diagrama de secuencia CU: Gestionar solicitud externa.

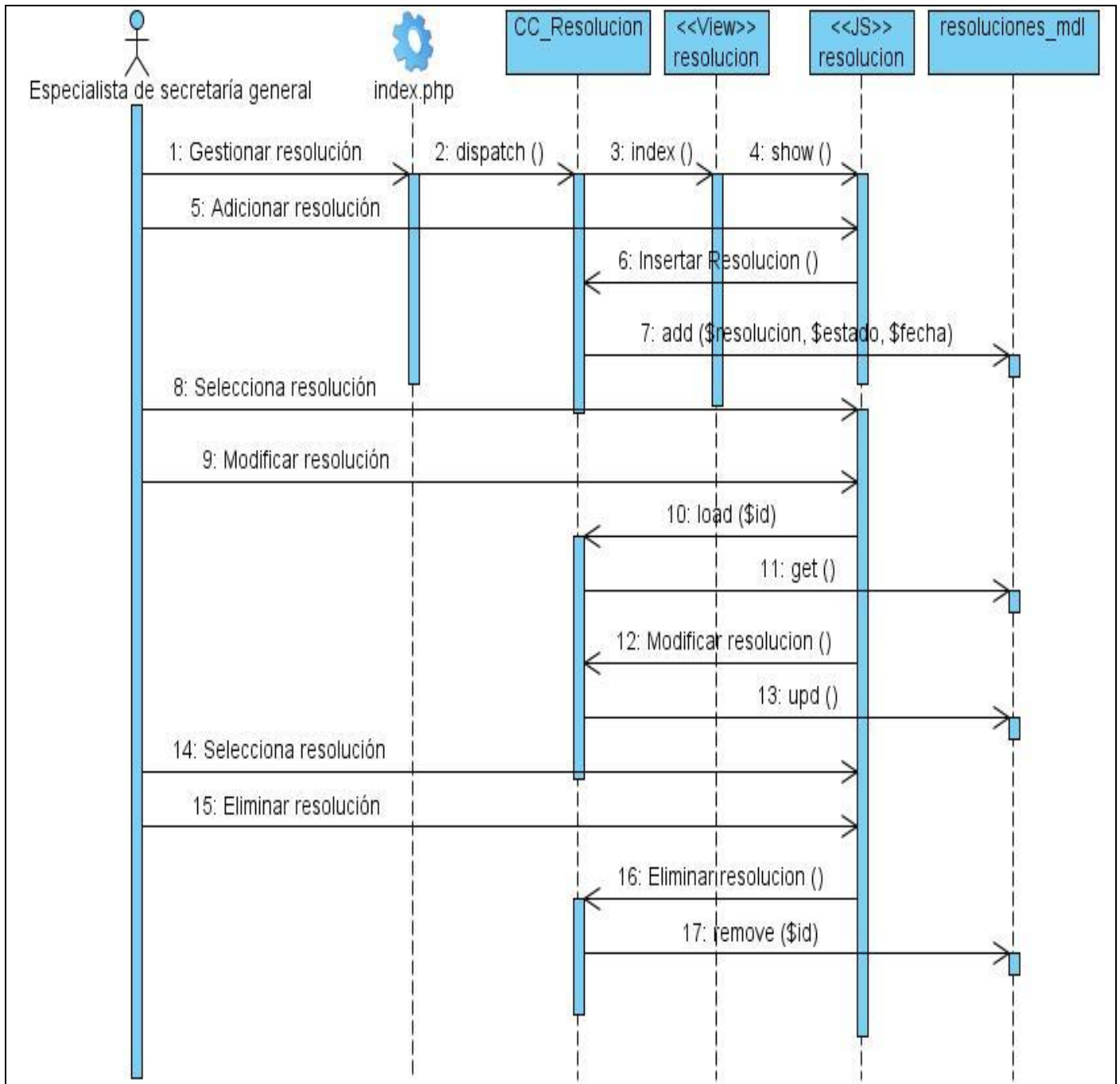


Figura 32: Diagrama de secuencia CU: Gestionar resolución.

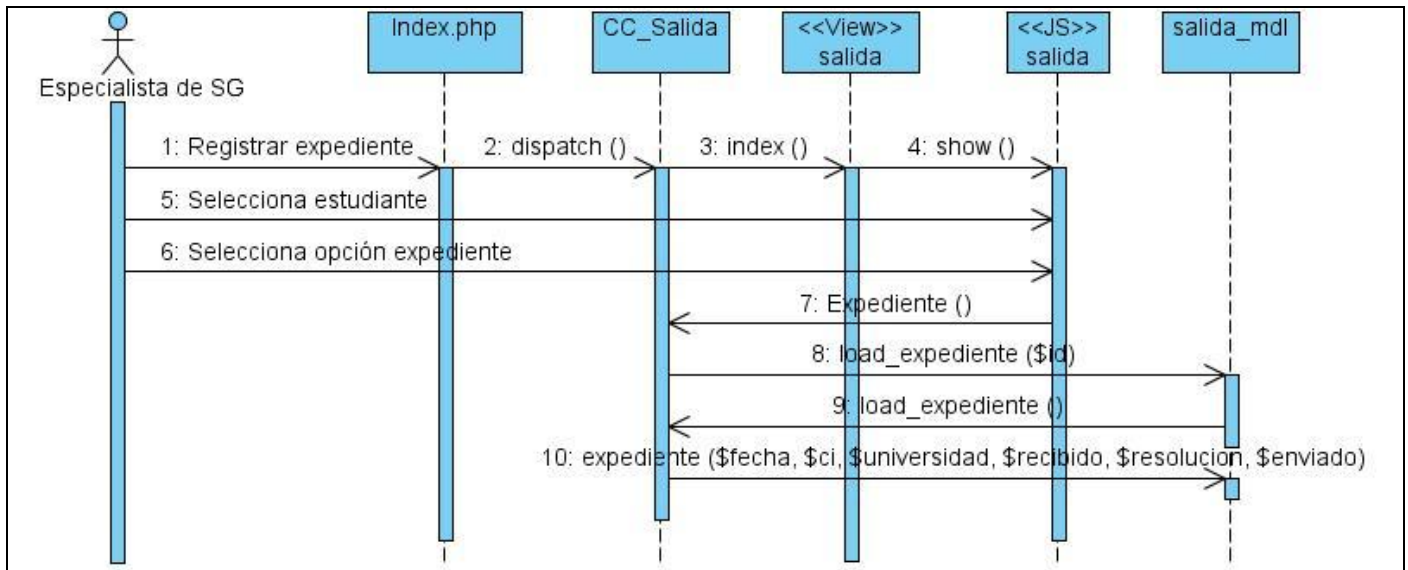


Figura 33: Diagrama de secuencia CU: Registrar expediente.

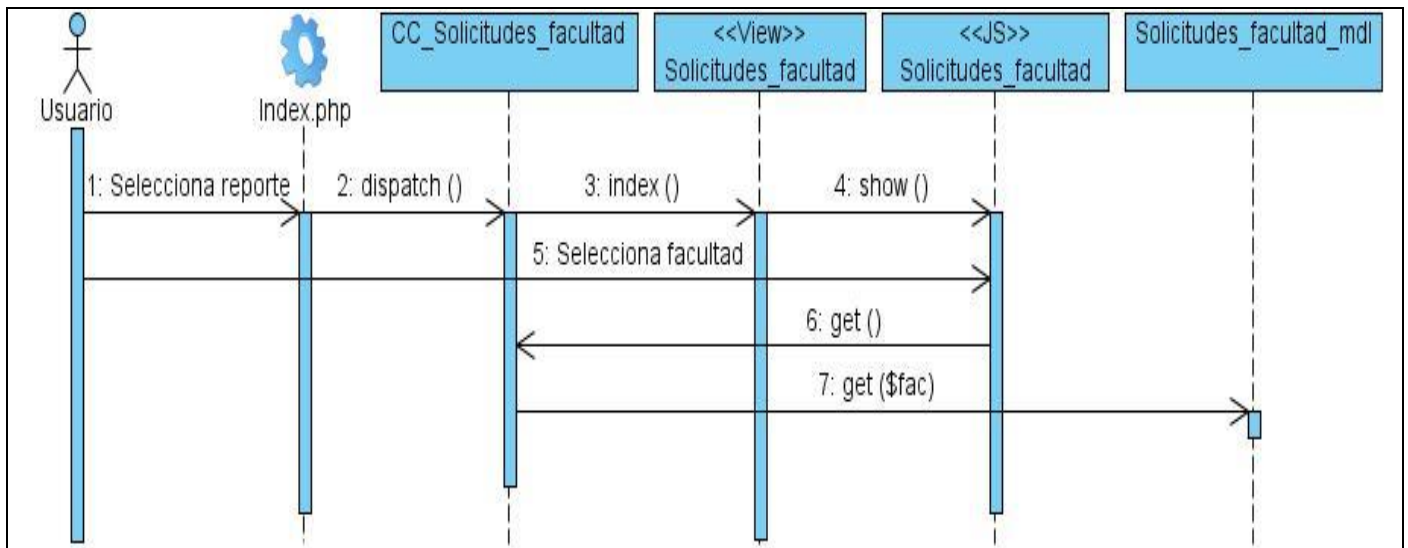


Figura 34: Diagrama de secuencia CU: Generar reportes.

Glosario.

Aquí se incluye una explicación de determinados términos y siglas utilizados en el texto para facilitar su comprensión. Generalmente se incluyen los términos que tienen menor difusión en nuestro campo profesional, los de otro campo profesional, o aquellos términos conocidos, pero que se usan con un significado diferente en el texto. Además, pueden ser incluidas las siglas utilizadas en el documento para facilitar su lectura.

Feedback: Es un proceso por el cual cierta proporción de la señal de salida de un sistema se pasa (alimentado de nuevo) a la entrada. Esta se usa a menudo para controlar el comportamiento dinámico del sistema. Ejemplos de información se pueden encontrar en la mayoría de los sistemas complejos, tales como la ingeniería, arquitectura, economía, la termodinámica y la biología. Un ejemplo de un complejo sistema de votos es el sistema de dirección de un automóvil. Mientras conduce, una persona que recibe las señales del medio ambiente, tales como signos y peligros, el cerebro del conductor procesa la información y envía señales al automóvil a través del volante y los pedales. El automóvil en consecuencia responde al cambiar de dirección o velocidad.

Frameworks: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Iteración: En *matemática* se refiere al proceso de iteración de una función o a las técnicas que se usan en métodos iterativos para la resolución de problemas numéricos y en *programación* es la repetición de una serie de instrucciones en un programa de computadora. Puede usarse tanto como un término genérico (como sinónimo de repetición) así como para describir una forma específica de repetición con un estado mutable.

LDAP: es un protocolo a nivel de aplicación, que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversas informaciones en el entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.

Plugin: Es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande. Un programa puede tener uno o más conectores. Son muy utilizados en los programas navegadores para ampliar sus funcionalidades.

Testing: Pruebas que se le realizan a los software que permiten encontrar las fallas antes de la puesta en producción. Esto garantiza la calidad de un software.