

1. TÍTULO

“DISEÑO E IMPLEMENTACIÓN DEL SISTEMA INFORMÁTICO PARA LA GESTIÓN ACADÉMICA DE LA ESCUELA FISCAL MIXTA JULIO MARÍA MATOVELLE”.

## 2. RESUMEN

La presente investigación se ha desarrollado bajo los lineamientos que plantea el Sistema Académico Modular por Objetos de Transformación “SAMOT” que implementa la Universidad Nacional de Loja. Acorde con la materia de Diseño de Sistemas del presente Módulo y el docente encargado se delimitó el tema de Investigación denominado: “DISEÑO E IMPLEMENTACIÓN DEL SISTEMA INFORMÁTICO PARA LA GESTIÓN ACADÉMICA DE LA ESCUELA FISCAL MIXTA JULIO MARÍA MATOVELLE”.

Con el análisis realizado previamente se dio inicio a la elaboración de los diagramas de robustez, diagramas de secuencia y diagramas de colaboración y con esto se pudo realizar el diagrama de paquetes, modelado de clases y modelado de ventanas para poder construir un Sistema Informático que permita agilizar los procesos de matriculación y acreditación de la escuela en cuestión.

El presente trabajo de investigación tiene los siguientes objetivos específicos: Diseñar y realizar la descripción de los casos de uso. Diseñar los diagramas de robustez y de secuencia. Diseñar el modelo del dominio del problema. Desarrollar el sistema informático basado en el análisis preliminar de la escuela. Definir de manera apropiada la metodología a utilizar para el Diseño del Sistema. Codificar el programa y realizar el mapeo de la base de datos. Implementar el Sistema Informático correspondiente al Diseño establecido. Capacitar a los usuarios que van a hacer uso del Sistema Informático.

### 3. INTRODUCCIÓN

El presente proyecto investigativo pretende dar solución a los problemas que se presentan al realizar los procesos de matriculación y acreditación en la Escuela Julio María Matovelle mediante la construcción de un Sistema Informático. Luego de haber realizado un previo análisis del Sistema de Información que tiene esta institución para llevar a cabo el proceso de gestión académica se pudo notar que él no identificar los problemas que aquejan a la institución puede llevar a una desorganización de información, ya que para la buena gestión de datos es necesario. En muchas instituciones locales, nacionales e internacionales se realizan con demasiada frecuencia por lo que éstas tienen un desempeño muy bueno.

Para el desarrollo del presente trabajo investigativo se ha desarrollado un estudio profundo sobre “LA FALTA DE IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO EN LA ESCUELA FISCAL MIXTA “JULIO MARÍA MATOVELLE” QUE AGILICE LA GESTION ACADEMICA EN EL PROCESO DE MATRICULACION Y CALIFICACION DE LOS ESTUDIANTES”, incide en la pérdida de tiempo y desorganización de información en la escuela Julio M. Matovelle de la ciudad de Loja.

Por tal motivo se ha construido un Sistema Informático que disminuya en gran medida los problemas que se presentan al realizar el proceso de Gestión Académica dentro de la Institución en cuestión. Este Sistema Informático presenta las siguientes características:

- Administra la Información de los estudiantes, directora, secretaria y docentes.
- Administra un periodo académico, los años de educación básica, los paralelos dentro de un año de educación básica y los ciclos de estudio.
- Permite registrar las matrículas correspondientes a cada estudiante.
- Permite registrar acreditaciones trimestrales por estudiante.
- Permite registrar días asistidos y no asistidos por cada estudiante así como las faltas justificadas e injustificadas.
- Permite realizar búsquedas de la información de los estudiantes y búsquedas de paralelos.

- Permite calcular el promedio trimestral y el promedio final de alumno.
- Permite generar reportes de alumnos vigentes como: Certificado de matrícula, libreta de calificaciones, lista de alumnos, lista de alumnos con notas ordenadas.
- Permite generar reportes de ex-alumnos como: Informe sobre la fecha en que un alumno culminó la Educación Primaria, Informe sobre las acreditaciones de séptimo.
- Permite generar un listado de padres de familia.

El desarrollo del presente trabajo investigativo es de suma importancia ya que el mismo ha dado lugar a la elaboración de un programa informático que permita dar solución a los problemas que se presentan en el proceso de Gestión Académica en la institución antes mencionada.

Los objetivos que se han planteado en la presente investigación son los siguientes:

- **Objetivo General:** Realizar el diseño e implementación del sistema informático para la gestión académica de la Escuela Fiscal Mixto Julio María Matovelle.
- **Objetivos Específicos:** Diseñar y realizar la descripción de los casos de uso. Diseñar los diagramas de robustez y de secuencia. Diseñar el modelo del dominio del problema. Desarrollar el sistema informático basado en el análisis preliminar de la escuela. Definir de manera apropiada la metodología a utilizar para el Diseño del Sistema. Codificar el programa y realizar el mapeo de la base de datos. Implementar el Sistema Informático correspondiente al Diseño establecido. Capacitar a los usuarios que van a hacer uso del Sistema Informático.

#### 4. OBJETIVOS INVESTIGACIÓN

##### **OBJETIVO GENERAL**

Realizar el diseño e implementación del sistema informático para la gestión académica de la Escuela Fiscal Mixto Julio María Matovelle.

##### **OBJETIVOS ESPECÍFICOS**

- ✓ Diseñar y realizar la descripción de los casos de uso. Diseñar los diagramas de robustez y de secuencia.
- ✓ Diseñar el modelo del dominio del problema.
- ✓ Desarrollar el sistema informático basado en el análisis preliminar de la escuela.
- ✓ Definir de manera apropiada la metodología a utilizar para el Diseño del Sistema.
- ✓ Codificar el programa y realizar el mapeo de la base de datos. Implementar el Sistema Informático correspondiente al Diseño establecido.
- ✓ Capacitar a los usuarios que van a hacer uso del Sistema Informático.

## 5. ÍNDICE

1. Título.....	2
2. Resumen.....	3
3. Introducción.....	4
4. Objetivos de la investigación.....	6
5. Índice.....	7
6. Revisión de Literatura.....	14
<b>6.1</b> Escuela Fiscal Mixta Julio María Matovelle.....	14
<b>6.1.1</b> Antecedentes.....	14
<b>6.1.2</b> Visión.....	14
<b>6.1.3</b> Misión.....	15
<b>6.1.4</b> Objetivos de la Institución.....	15
<b>6.1.5</b> Requerimientos de la Institución.....	15
<b>6.2</b> Análisis de Sistemas.....	16
<b>6.2.1</b> Definición.....	16
<b>6.2.2</b> Roles del Analista.....	16
<b>6.2.3</b> Cualidades.....	17
<b>6.2.4</b> Tipos de Usuario.....	17
<b>6.3</b> Sistemas de Información.....	18
<b>6.3.1</b> Definición.....	18
<b>6.3.2</b> Categorías de los Sistemas de Información.....	18
<b>6.3.3</b> Integración de las Tecnologías de Sistemas.....	19
<b>6.3.4</b> Análisis y Diseño de Sistemas.....	20
<b>6.3.5</b> Ciclo de Desarrollo del Sistema.....	21
<b>6.3.6</b> Análisis Orientado a Objetos.....	23
<b>6.3.7</b> Requerimientos.....	23
<b>6.3.8</b> Casos de Uso.....	23
<b>6.4</b> Modelado del Contexto de un Sistema.....	24
<b>6.4.1</b> Modelo Conceptual.....	24
<b>6.4.2</b> Atributos.....	24
<b>6.5</b> Diseño de sistemas.....	24
<b>6.5.1</b> Iconix.....	24
<b>6.5.1.1</b> El Dominio del Problema.....	25

<b>6.5.1.2</b>	Tareas de Iconix.....	25
<b>6.5.1.2.1</b>	Análisis de Requisitos.....	25
<b>6.5.1.2.2</b>	Análisis y Diseño Preliminar.....	26
<b>6.5.1.2.3</b>	Diseño.....	26
<b>6.5.1.2.4</b>	Implementación.....	26
<b>6.5.2</b>	Diagramas de Robustez.....	27
<b>6.5.3</b>	Diseño Orientado a Objetos.....	27
<b>6.5.3.1</b>	Diagrama de Interacción.....	27
<b>6.5.3.1.1</b>	Diagramas de Secuencia.....	28
<b>6.5.3.1.2</b>	Diagramas de Colaboración..	29
<b>6.5.4</b>	Diagramas de Clases.....	30
<b>6.5.4.1</b>	Diagramas de Paquetes.....	32
<b>6.5.4.2</b>	Diagramas de Componentes.....	33
<b>6.5.4.3</b>	Diagramas de despliegue.....	33
<b>6.6</b>	Base de Datos.....	34
<b>6.6.1</b>	Base de Datos Estadísticas.....	34
<b>6.6.2</b>	Base de datos Dinámicas.....	34
<b>6.6.2.1</b>	Nivel Físico.....	34
<b>6.6.2.2</b>	Nivel Conceptual.....	35
<b>6.6.2.3</b>	Nivel Visión.....	35
<b>6.6.3</b>	Modelo de Base de Datos.....	35
<b>6.7</b>	Librerías Java.....	36
<b>6.7.1</b>	Toplink.....	37
<b>6.8</b>	JPA.....	38
<b>6.8.1</b>	Asociaciones JPA.....	38
<b>6.8.1.1</b>	Especificación de Entidades JPA.....	38
<b>6.8.1.2</b>	@Column.....	38
<b>6.8.1.3</b>	@Id.....	39
<b>6.8.1.4</b>	@GeneratedValue.....	39
<b>6.8.1.5</b>	@Temporal.....	39
<b>6.8.1.6</b>	@JoinColumn.....	39
<b>6.8.1.7</b>	@OneToOne.....	39
<b>6.8.1.8</b>	@OneToMany.....	39

<b>6.8.1.9</b>	@ManyToOne.....	40
<b>6.8.1.10</b>	@ManyToMany.....	40
<b>6.9</b>	Graphics2d.....	40
7.	Métodos y Materiales .....	42
<b>7.1</b>	Métodos.....	42
<b>7.2</b>	Materiales .....	43
8.	Resultados.....	45
<b>8.1</b>	Requerimientos no funcionales.....	45
<b>8.1.1</b>	Funciones Básicas.....	45
<b>8.1.2</b>	Funciones de Matricula.....	48
<b>8.1.3</b>	Funciones de Acreditación.....	51
<b>8.2</b>	Casos de Uso.....	54
<b>8.3</b>	Descripción de Casos de Uso Expandidos.....	55
<b>8.3.1</b>	<b>Caso de Uso 1: Ingreso al Sistema.....</b>	<b>55</b>
<b>8.3.1.1</b>	Curso Normal de Eventos.....	55
<b>8.3.1.2</b>	Diagrama de Robustez.....	56
<b>8.3.1.3</b>	Diagrama de Secuencia.....	56
<b>8.3.1.4</b>	Diagrama de Colaboración.....	57
<b>8.3.2</b>	<b>Casos de Uso 2: Administrar Usuario.....</b>	<b>57</b>
<b>8.3.2.1</b>	Curso Normal de Eventos.....	58
<b>8.3.2.2</b>	Sección: Agregar Usuario.....	58
<b>8.3.2.2.1</b>	Diagrama de Robustez.....	59
<b>8.3.2.2.2</b>	Diagrama de Secuencia.....	60
<b>8.3.2.2.3</b>	Diagrama de Colaboración....	61
<b>8.3.2.3</b>	Sección: Modificar Usuario.....	61
<b>8.3.2.3.1</b>	Diagrama de Robustez.....	62
<b>8.3.2.3.2</b>	Diagrama de Secuencia.....	63
<b>8.3.2.3.3</b>	Diagrama de Colaboración....	64
<b>8.3.2.4</b>	Sección: Asignar Año .....	64
<b>8.3.2.4.1</b>	Diagrama de Robustez.....	65
<b>8.3.2.4.2</b>	Diagrama de Secuencia.....	66
<b>8.3.2.4.3</b>	Diagrama de Colaboración....	66
<b>8.3.3</b>	Casos de Uso 3: Administrar Periodo.....	67



<b>8.3.3.1</b>	Curso Normal de Eventos.....	67
<b>8.3.3.2</b>	Sección: Crear Periodo.....	68
<b>8.3.3.2.1</b>	Diagrama de Robustez.....	69
<b>8.3.3.2.2</b>	Diagrama de Secuencia.....	69
<b>8.3.3.2.3</b>	Diagrama de Colaboración....	70
<b>8.3.3.3</b>	Sección: Modificar Periodo.....	70
<b>8.3.3.3.1</b>	Diagrama de Robustez.....	71
<b>8.3.3.3.2</b>	Diagrama de Secuencia.....	71
<b>8.3.3.3.3</b>	Diagrama de Colaboración....	72
<b>8.3.4</b>	Caso de Uso 4: Administrar Año de Educación Básica	72
<b>8.3.4.1</b>	Curso Normal de Eventos.....	73
<b>8.3.4.2</b>	Sección: Agregar Año.....	73
<b>8.3.4.2.1</b>	Diagrama de Robustez.....	74
<b>8.3.4.2.2</b>	Diagrama de Secuencia.....	75
<b>8.3.4.2.3</b>	Diagrama de Colaboración....	75
<b>8.3.4.3</b>	Sección: Modificar Año.....	75
<b>8.3.4.3.1</b>	Diagrama de Robustez.....	77
<b>8.3.4.3.2</b>	Diagrama de Secuencia.....	77
<b>8.3.4.3.3</b>	Diagrama de Colaboración....	78
<b>8.3.5</b>	Caso de Uso 5: Administrar Paralelos.....	89
<b>8.3.5.1</b>	Curso Normal de Eventos.....	80
<b>8.3.5.2</b>	Sección: Ingresar Paralelo.....	80
<b>8.3.5.2.1</b>	Diagrama de Robustez.....	81
<b>8.3.5.2.2</b>	Diagrama de Secuencia.....	82
<b>8.3.5.2.3</b>	Diagrama de Colaboración....	82
<b>8.3.5.3</b>	Sección: Modificar Paralelo.....	83
<b>8.3.5.3.1</b>	Diagrama de Robustez.....	84
<b>8.3.5.3.2</b>	Diagrama de Secuencia.....	84
<b>8.3.5.3.3</b>	Diagrama de Colaboración....	85
<b>8.3.6</b>	Caso de Uso: Administrar Materia.....	86
<b>8.3.6.1</b>	Curso normal de eventos.....	87
<b>8.3.6.2</b>	Sección: Agregar Materia.....	88
<b>8.3.6.2.1</b>	Diagrama de Robustez.....	88

8.3.6.2.2	Diagrama de Secuencia.....	89
8.3.6.2.3	Diagrama de Colaboración....	89
<b>8.3.6.3</b>	<b>Sección: Modificar Materia.....</b>	<b>89</b>
8.3.6.3.1	Diagrama de Robustez.....	90
8.3.6.3.2	Diagrama de Secuencia.....	90
8.3.6.3.3	Diagrama de Colaboración....	91
<b>8.3.6.4</b>	<b>Sección: Eliminar Materia.....</b>	<b>91</b>
8.3.6.4.1	Diagrama de Robustez.....	91
8.3.6.4.2	Diagrama de Secuencia.....	92
8.3.6.4.3	Diagrama de Colaboración....	92
<b>8.3.7</b>	<b>Caso de Uso: Administrar alumno.....</b>	<b>93</b>
<b>8.3.7.1</b>	<b>Curso Normal de eventos.....</b>	<b>94</b>
<b>8.3.7.2</b>	<b>Sección: Buscar Alumno.....</b>	<b>95</b>
8.3.7.2.1	Diagrama de Robustez.....	95
8.3.7.2.2	Diagrama de Secuencia.....	96
8.3.7.2.3	Diagrama de Colaboración....	96
<b>8.3.7.3</b>	<b>Sección: Agregar Alumno.....</b>	<b>97</b>
8.3.7.3.1	Diagrama de Robustez.....	98
8.3.7.3.2	Diagrama de Secuencia.....	98
8.3.7.3.3	Diagrama de Colaboración....	99
<b>8.3.7.4</b>	<b>Sección Modificar alumno.....</b>	<b>99</b>
8.3.7.4.1	Diagrama de Robustez.....	100
8.3.7.4.2	Diagrama de Secuencia.....	101
8.3.7.4.3	Diagrama de Colaboración....	102
<b>8.3.8</b>	<b>Caso de Uso: Registrar Matricula.....</b>	<b>102</b>
<b>8.3.8.1</b>	<b>Curso Normal de Eventos.....</b>	<b>103</b>
<b>8.3.8.2</b>	<b>Sección: Agregar Matricula.....</b>	<b>104</b>
8.3.8.2.1	Diagrama de Robustez.....	104
8.3.8.2.2	Diagrama de Secuencia.....	105
8.3.8.2.3	Diagrama de Colaboración....	106
<b>8.3.8.3</b>	<b>Sección: Modificar Matricula.....</b>	<b>107</b>
8.3.8.3.1	Diagrama de Robustez.....	107
8.3.8.3.2	Diagrama de Secuencia.....	108

8.3.8.3.3	Diagrama de Colaboración....	109
<b>8.3.9</b>	<b>Caso de Uso: Registrar Asistencias.....</b>	<b>110</b>
8.3.9.1	Curso Normal de Eventos.....	111
8.3.9.2	Sección: Agregar Asistencia.....	111
8.3.9.2.1	Diagrama de Robustez.....	112
8.3.9.2.2	Diagrama de Secuencia.....	113
8.3.9.2.3	Diagrama de Colaboración....	114
8.3.9.3	Sección: Modificar Asistencia.....	114
8.3.9.3.1	Diagrama de Robustez.....	115
8.3.9.3.2	Diagrama de Secuencia.....	116
8.3.9.3.3	Diagrama de Colaboración....	117
<b>8.3.10</b>	<b>Caso de Uso: Registrar Acreditaciones.....</b>	<b>118</b>
8.3.10.1	Curso Normal de Eventos.....	119
8.3.10.2	Sección: Agregar Acreditaciones.....	120
8.3.10.2.1	Diagrama de Robustez.....	121
8.3.10.2.2	Diagrama de Secuencia.....	122
8.3.10.2.3	Diagrama de Colaboración....	123
8.3.10.3	Sección: Modificar Acreditación.....	123
8.3.10.3.1	Diagrama de Robustez.....	124
8.3.10.3.2	Diagrama de Secuencia.....	125
8.3.10.3.3	Diagrama de Colaboración....	126
<b>8.3.11</b>	<b>Caso de Uso: Generar Reportes.....</b>	<b>126</b>
8.3.11.1	Curso Normal de Eventos.....	127
8.3.11.2	Sección: Generar Certificado de Materia..	128
8.3.11.2.1	Diagrama de Robustez.....	129
8.3.11.2.2	Diagrama de Secuencia.....	130
8.3.11.2.3	Diagrama de Colaboración....	131
8.3.11.3	Sección: Generar Informe de Acreditaciones	132
8.3.11.3.1	Diagrama de Robustez.....	134
8.3.11.3.2	Diagrama de Secuencia.....	135
8.3.11.3.3	Diagrama de Colaboración....	136
8.3.11.4	Sección: Generar Lista de notas Finales..	137
8.3.11.4.1	Diagrama de Robustez.....	139

8.3.11.4.2 Diagrama de Secuencia.....	140
8.3.11.4.3 Diagrama de Colaboración....	141
<b>8.3.11.5</b> Sección: Generar Lista de Alumnos...	142
8.3.11.5.1 Diagrama de Robustez.....	143
8.3.11.5.2 Diagrama de Secuencia.....	144
8.3.11.5.3 Diagrama de Colaboración....	145
<b>8.3.11.6</b> Sección: Generar Lista de Representantes...	145
8.3.11.6.1 Diagrama de Robustez.....	147
8.3.11.6.2 Diagrama de Secuencia.....	148
8.3.11.6.3 Diagrama de Colaboración....	149
<b>8.3.12</b> Caso de Uso: Administrar Ciclo de estudio.....	149
8.3.12.1 Curso Normal de Eventos.....	150
<b>8.3.12.2</b> Sección: Agregar Ciclo de Estudio....	151
8.3.12.2.1 Diagrama de Robustez.....	152
8.3.12.2.2 Diagrama de Secuencia.....	153
8.3.12.2.3 Diagrama de Colaboración....	153
<b>8.3.12.3</b> Sección: Modificar Ciclo de Estudio...	154
8.3.12.3.1 Diagrama de Robustez.....	154
8.3.12.3.2 Diagrama de Secuencia.....	155
8.3.12.3.3 Diagrama de Colaboración....	155
<b>8.4</b> Modelo de Paquetes.....	156
<b>8.5</b> Diagrama de Ventanas.....	157
<b>8.6</b> Diagrama de Controladores.....	158
<b>8.7</b> Diagramas de Clases.....	159
<b>8.8</b> Modelo de la Base de Datos.....	160
<b>9.</b> Conclusiones y Recomendaciones.....	161
9.1 Conclusiones .....	161
9.2 Recomendaciones.....	162
<b>10.</b> Bibliografía.....	163
<b>11.</b> Anexos.....	164

## **6. REVISIÓN DE LA LITERATURA**

### **6.1 ESCUELA FISCAL MIXTA JULIO MARÍA MATOVELLE**

#### **6.1.1 ANTECEDENTES**

El código de la niñez y la adolescencia publicado el 3 de enero de 2003, en su Título I, Art 1 dispone “La protección integral que el Estado, la sociedad y la familia deben garantizar a todos los niños/as y adolescentes a fin de lograr el desarrollo integral y el disfrute de sus derechos, en su masco de libertad dignidad y equidad”.

Se ha certificado que en las instituciones educativas, sus reglamentos internos son elaborados desde esquemas de autoridad y poder sin la concertación y conceso de la comunidad educativa. Muchas veces los estudiantes y padres de familia desconocen el contenido de estos reglamentos y únicamente cuando se suscitan problemas o conflictos es cuando se hace referencia a los mismos.

Nuestra institución considera acertada la decisión del Ministerio de Educación de institucionalizar los Códigos de Convivencia con la finalidad de prevenir los problemas y conflictos que se dan entre integrantes de una comunidad educativa. Sin un buen clima escolar no se puede dar una educación de calidad. Ante esta realidad la Escuela Fiscal Mixta “Julio María Matovelle” ha creído conveniente elaborar su Código de Convivencia que optimice las relaciones humanas en nuestra institución educativa a fin de hacer de ella un espacio de vivencia democrática, de respeto de derechos de cumplimiento de obligaciones, de construcción de ciudadanía y un ambiente propicio para el aprendizaje; en tal virtud nuestro plantel está dando cumplimiento a lo dispuesto por el Ministerio de Educación; mediante Acuerdo Ministerial N° 182, de mayo del 2007.

#### **6.1.2 VISIÓN**

Contar con un establecimiento educativo, considerado como el lugar propicio para la formación humana y convivencia que promueva el desarrollo integral de la comunidad educativa para poder afrontar los nuevos retos de acuerdo a los signos de los tiempos.

### **6.1.3 MISIÓN**

Asumir con voluntad y poner en práctica las normas establecidas en el presente código de Convivencia para aprender a convivir en armonía, solucionar problemas, a tener confianza en nuestras potencialidades de estudio como de trabajo.

### **6.1.4 OBJETIVOS DE LA INSTITUCIÓN**

Contribuir al mejoramiento de las relaciones de convivencia en la institución educativa, lo que permitirá un ambiente de participación, respeto, compromiso de todos los actores del quehacer educativo: maestros, alumnos, padres de familia, autoridades y comunidad en general, para que en un marco de respeto de derechos y cumplimiento de deberes, propender el mejoramiento de la calidad de educación.

Asumir un compromiso personal y comunitario de convivencia educativa, pacífica y productiva mediante el conocimiento de deberes, derechos, responsabilidades y compromisos; capaz de ayudarnos a todos a madurar y superar posibles deficiencias.

### **6.1.5 REQUERIMIENTOS DE LA INSTITUCIÓN**

#### **DOCENTES**

- ❖ Ampliación y actualización de conocimientos
- ❖ Mejorar la de práctica de valores
- ❖ Responsabilidad en el desarrollo de trabajo

#### **ESTUDIANTES**

- ❖ Reducir el elevado porcentaje de inasistencias y atrasos de alumnos.
- ❖ Controlar la indisciplina y no cumplimiento de las tareas escolares.
- ❖ Reducir el descuido en la presentación de útiles escolares y uniformes.

#### **REPRESENTANTE**

- ❖ Ampliar la práctica de valores cívicos y sociales.
- ❖ Proporcionar materiales indispensables para el trabajo en el aula.
- ❖ Controlar de mejor manera las tareas y deberes de sus representados.

## 6.2 ANÁLISIS DE SISTEMAS

### 6.2.1 DEFINICIÓN

El analista de sistemas evalúa de manera sistemática el funcionamiento de un negocio mediante el examen de la entrada y el procesamiento de datos y su consiguiente producción de información, con el propósito de mejorar los procesos de una organización.

### 6.2.2 ROLES DEL ANALISTA

Los principales papeles que puede cumplir el analista son

#### a. *Como Consultor:*

- Para analizar ciertos tópicos de la información que genera la organización
- Es un consultor externo que trae nuevas ideas que no poseen otros miembros de la organización, el mismo que deberá conocer e implantar los métodos necesarios que implican el análisis y el diseño de sistemas
- Debe contar con la ayuda del usuario de los S.I., para entender el funcionamiento de la organización.

#### b. *Como especialista de apoyo:*

- El cual trabaja dentro de la organización de manera regular en el departamento de sistemas.
- Es un profesional con amplia experiencia en hardware y software.
- Como especialista de apoyo no dirige ningún proyecto sino más bien presta su coloración para desarrollo.

#### c. *Como agente de Cambio*

- Este papel implica una muy alta responsabilidad dentro de la organización
- Un agente de cambio puede definirse como aquella persona que sirve como catalizador para el cambio, que desarrolla un plan o proyecto y que colabora con otras para agilizarlo

### 6.2.3 CUALIDADES

- El analista de sistemas es un mediador. Es responsable de mostrar a los usuarios del sistema y a los directivos cómo pueden las nuevas tecnologías favorecer a su empresa y a sus operaciones. Para alcanzar dicho fin, el analista ha de ser conocedor de las tecnologías y las técnicas existentes. Una de las mejores formas para mantenerse al día es desarrollar la disciplina y el hábito de hojear y leer diversas publicaciones periódicas sobre sistemas de información.
- El analista debe tener experiencia y dominio de la programación informática, es decir, escriba o no programas, es evidente que el analista debe saber programar; debido a que el analista constituye el principal eslabón entre los usuarios de la empresa y los programadores informáticos.
- Un analista debe tener la capacidad de relación interpersonal, es decir, el trabajo con sistemas se basa en personas, y los analistas deben ser extrovertidos o proclives a tratar con personas.
- El analista debe ser flexible y tener capacidad de adaptación. No es posible encontrar dos proyectos de desarrollo de sistemas donde los analistas de sistemas tengan idéntica función.

### 6.2.4 TIPOS DE USUARIO

- ***Usuarios Primarios.*** Que son las personas que usan de manera directa en sus tareas, los resultados que emiten los sistemas de información. Por ejemplo los gerentes que reciben informes diarios de los diferentes departamentos.
- ***Usuarios secundarios.*** Que son las personas encargadas de introducir los datos al sistema de información, pero no utilizan de forma directa los informes que se emiten para el cumplimiento de sus tareas.



- **Los propietarios.** Los cuales patrocinan los sistemas basados en ordenadores.

En otras palabras pagan para disponer del sistema desarrollado, aprueban la tecnología utilizada en el sistema y pagan para el mantener dicho sistema en funcionamiento. Y lo que es más importante, esperan del sistema que, en el conjunto de su tiempo de vida, suministren mayores beneficios de coste, evidentemente, el propietario puede ser también un usuario primario o secundario.

### 6.3 SISTEMAS DE INFORMACIÓN

#### 6.3.1 DEFINICIÓN

La información es un elemento determinante dentro de todas las operaciones de la organización, porque gracias a la información que se genera de las operaciones dentro de los diferentes departamentos, se logrará establecer si se están desempeñando correctamente las funciones para las que fueron designados.

#### 6.3.2 CATEGORÍAS DE LOS SISTEMAS DE INFORMACIÓN

##### a. *Sistemas de Procesamiento de datos:*

Los sistemas de procesamiento de transacciones (TPS, *Transaction Processing Systems*) son sistemas de información computarizada creados para procesar grandes cantidades de datos relacionadas con transacciones rutinarias de negocios, como las nóminas y los inventarios.

##### b. **Sistemas de Automatización de la Oficina y Sistemas de Trabajo del Conocimiento.**

Existen dos clases de sistemas en el nivel del conocimiento de una organización. Los sistemas de automatización de la oficina [OAS, *Office Automation Systems*] apoyan a los trabajadores de datos, quienes por lo general no generan conocimientos nuevos, sino más bien analizan la información con el propósito de transformar los datos o manipularlos de alguna manera antes de compartirlos o, en su caso, distribuirlos formalmente con el resto de la organización y en ocasiones más allá de ésta.

##### c. *Sistemas de Información para la Administración*

Los MIS (*Management Information Systems*) son sistemas de información computarizados cuyo propósito es contribuir a la correcta interacción entre los usuarios y las computadoras. Debido a que requieren que los usuarios, el software [los programas de cómputo] y el hardware (las computadoras, impresoras, etc.), funcionen de manera coordinada, los sistemas de información gerencial dan apoyo a un espectro de tareas organizacionales mucho más amplio que los sistemas de procesamiento de transacciones, como el análisis y la toma de decisiones.

**d. Sistema de apoyo para la toma de decisiones.**

Los sistemas de apoyo a la toma de decisiones (DSS, *Decisión Support Systems*) constituyen una clase de alto nivel de sistemas de información computarizada. Los DSS coinciden con los sistemas de información gerencial en que ambos dependen de una base de datos para abastecerse de datos. Sin embargo, difieren en que el DSS pone énfasis en el apoyo a la toma de decisiones en todas sus fases, aunque la decisión definitiva es responsabilidad exclusiva del encargado de tomarla.

**e. Sistemas expertos**

La inteligencia artificial (AI, *Artificial Intelligence*) se puede considerar como el campo general para los sistemas expertos. La motivación principal de la AI ha sido desarrollar máquinas que tengan un comportamiento inteligente. Dos de las líneas de investigación de la AI son la comprensión del lenguaje natural y el análisis de la capacidad para razonar un problema hasta su conclusión lógica. Los sistemas expertos utilizan las técnicas de razonamiento de la AI para solucionar los problemas que les plantean los usuarios de negocios (y de otras áreas).

### **6.3.3 INTEGRACIÓN DE LAS TECNOLOGÍAS DE SISTEMAS**

**a. Aplicaciones de Comercio Electrónico y Sistemas Web**

El Internet es una estrategia preferida para buscar el crecimiento de sus negocios. Esta respuesta duplicó a la de aquellos que manifestaron su inclinación por realizar alianzas estratégicas como medio para crecer. Hay muchos beneficios derivados de la implementación de una aplicación en la Web:

- Una creciente difusión de la disponibilidad de un servicio, producto, industria,

persona o grupo.

- La posibilidad de que los usuarios accedan las 24 horas.
- La estandarización del diseño de la interfaz.
- La creación de un sistema que se puede extender a nivel mundial y llegar a gente en lugares remotos sin preocuparse por la zona horaria en que se encuentren.

#### **b. Sistemas de Planeación de Recursos Empresariales**

Esta integración es precisamente el propósito de los sistemas de planeación de recursos empresariales (ERP, *Enterprise Resource Planning*). El establecimiento de los sistemas ERP implica un enorme compromiso y cambio por parte de la organización.

Entre el software más conocido de ERP se encuentran SAP, PeopleSoft y paquetes de Oracle y J.D. Edwards. Algunos de estos paquetes están diseñados para migrar a las empresas a la Web. Por lo general, los analistas y algunos usuarios requieren capacitación, apoyo técnico y mantenimiento por parte del fabricante para diseñar, instalar, dar mantenimiento, actualizar y utilizar de manera apropiada un paquete de ERP en particular.

#### **c. Sistemas para dispositivos Inalámbricos y Portátiles**

Los analistas tienen la exigencia de diseñar una gran cantidad de nuevos sistemas y aplicaciones, muchos de ellos para dispositivos inalámbricos y computadoras portátiles como la popular serie de computadoras Palm y otros asistentes personales digitales (PDAs, *Personal Digital Assistants*). Además, los analistas podrían llegar a diseñar redes de comunicaciones estándar o inalámbricas que integren voz, vídeo y correo electrónico en intranets para una organización o extrañéis para la industria.

#### **d. Software de Código Abierto**

Es una alternativa al desarrollo de software tradicional cuyo código patentado se oculta a los usuarios. Representa un modelo de desarrollo y filosofía de distribución de software gratuito y publicación de su código fuente. Bajo este esquema, el código (las instrucciones para la computadora) se puede estudiar y compartir, y muchos usuarios y programadores tienen la posibilidad de modificarlo.

### **6.3.4 ANÁLISIS Y DISEÑO DE SISTEMAS**

**a. El análisis:** Es el proceso mediante el cual se clasifica e interpreta hechos para

diagnosticar el problema a través del empleo de la información, para recomendar mejoras al sistema actual.

**b. El diseño:** Es el proceso por el cual se planifica, diseña, modela, reemplaza o completa en sistema organizacional existente, para lo cual es necesario comprender en su totalidad el viejo sistema y determinar la mejor forma de reemplazarlo.

### 6.3.5 CICLO DE DESARROLLO DEL SISTEMA

#### a. Identificación del problema, alternativas y objetivos:

- **Identificación del problema.** Esto involucra que el analista observe de forma objetiva y sistemática lo que ocurre en la empresa, e incluso necesitara el apoyo de otros miembros de la empresa para hacer notar los problemas
- **Alternativas.** Son aquellas situaciones que el analista considera que puedan ser perfeccionadas con el uso del sistema de información computarizado. Al plantear estas alternativas los directivos pueden aprovecharse de las mismas para lograr un mejor desarrollo de la empresa.
- **Objetivos.** En base a estudio de los logros que la empresa intenta alcanzar, el analista estará en la posibilidad de establecer si el uso de los sistemas de información apoyaría a la empresa para alcanzar sus metas.

#### b. Determinación de los requerimientos de información

Esta etapa trata de identificar en todo lo posible la información que requieren los usuarios involucrados para desempeñar sus tareas. Puede utilizarse varios instrumentos para este fin, entre los cuales se tiene: el muestreo, la entrevista, los cuestionarios, la observación, y también el desarrollo de prototipos.

Esta es una etapa en la cual el analista se verá en la necesidad de establecer una relación directa con los usuarios. Esta etapa le sirve al analista para elaborar una imagen de la organización y los objetivos que persigue.

#### c. Análisis de las necesidades

En esta etapa se analizan las necesidades propias del sistema y se hace uso de varias técnicas y herramientas que ayudan al analista para determinar los requerimientos del sistema.

Se utilizan los diagramas de flujo para representar el forma gráfica la entrada de los datos de la empresa, los procesos y la salida de información. A partir de estos diagramas de flujo se desarrollan los diccionarios de datos que contienen todos los elementos que utiliza el sistema.

En esta etapa el analista puede hacer uso de las decisiones estructuradas, que son decisiones donde las condiciones, condiciones alternativas, acciones y reglas de acción podrán determinarse.

#### **d. Diseño del sistema recomendado**

En base a la información recolectada en el análisis, se realiza el diseño lógico del sistema de información. El analista diseña procedimientos para la captura de datos, que permitan verificar que los datos que se ingresan al sistema sean correctos. También debe diseñar formato de entrada y salida de datos por pantalla.

#### **e. Desarrollo y documentación del software**

En esta etapa el analista trabaja con los programadores para crear el software que sea necesario para desarrollar el sistema de información

En proyectos pequeños es analista de sistema se encarga de realizar las tareas del programador para desarrollar el sistema. En esta etapa el analista de sistemas se encarga de instruir al programador sobre los requerimientos de programación que requiere el sistema.

Es también necesario que el analista trabaje con las usuarios del sistema para desarrollar la documentación que estos requieren para manejarlo (manuales del usuario, manuales de procedimientos)

#### **f. Pruebas y mantenimiento**

El sistema deberá de ser probado antes de utilizarse, debido a que ciertos factores imprevisto ocasionar falla en el sistema, esto implica que el costo sea menor si se detecta a tiempo los problemas antes de la entrega

Los programadores encargados de la realización del sistema deberán realizar sus propias pruebas, y otras deberá llevarlas a cabo con el analista de sistemas y los usuarios que emplearan el sistema.

#### **g. Implantación y evaluación del sistema**

En esta etapa el analista ayuda a implantar el sistema de información, lo cual involucra el adiestramiento de los usuarios que manejarán el sistema. Uno de los criterios fundamentales que debe satisfacerse, es que en el futuro los usuarios manejen bien el sistema.

### 6.3.6 ANÁLISIS ORIENTADO A OBJETOS

La mayoría de proyectos de software son complejos, y la estrategia primaria para superar la complejidad, es la *descomposición* (divide y vencerás). La estrategia es dividir el problema en unidades más pequeñas que sean *manejables*. Un enfoque tradicional para realizar esto fue el *análisis y diseño estructurados*, donde se trata de descomponer el problema en *funciones* o *procesos*.

### 6.3.7 Requerimientos

Los *requerimientos* son una descripción de las necesidades o deseos de un producto. La meta principal en esta etapa es identificar y documentar lo que en realidad se necesita, en una forma en que pueda fácilmente ser transmitido al cliente y al equipo de desarrollo. Se recomienda aquí definir al menos los siguientes puntos:

- **Panorama general**

En el proyecto se crea un sistema de terminal donde se utilizará métodos o procedimientos requeridos en la empresa.

- **Metas**

- Automatizar los diferentes procesos necesarios en el sistema.

- **Funciones del sistema**

Las *funciones del sistema* son lo que éste deberá de *hacer*. Hay que identificar estas funciones y listarlas en grupos lógicos.

- **Atributos del sistema**

Los *atributos del sistema* son cualidades no funcionales que a menudo se confunden con las funciones.

### 6.3.8 CASOS DE USO

Es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso. Para especificar los casos de uso en el lenguaje UML

## **CASO DE USO EXPANDIDO**

Un *caso de uso expandido* muestra más detalles que un *caso de uso de alto nivel*. Los casos de uso expandidos son útiles para alcanzar un conocimiento más profundo de los procesos y los requerimientos.

## **6.4 MODELADO DEL CONTEXTO DE UN SISTEMA**

### **6.4.1 MODELO CONCEPTUAL**

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones, y es la herramienta más importante del *análisis orientado a objetos*. Los casos de uso son una importante herramienta para el análisis de requerimientos, pero realmente no están *orientados a objetos*. Un modelo conceptual representa cosas del mundo real, no componentes del software. En UML se representa mediante un grupo de *diagramas de estructura estática* donde no se define ninguna operación. En estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos (atributos). La siguiente figura muestra un modelo conceptual parcial del dominio de la tienda y las ventas.

### **6.4.2 ATRIBUTOS**

Un *atributo* es un valor lógico de un dato de un objeto. Es preferible que los atributos sean simples. Entre los *tipos* de atributos más comunes se encuentran: booleanos (o lógicos), fechas, números, texto y horas. Algunos tipos comunes son: dirección, color, teléfono, RUT, código de barras, código postal

Los atributos no deberían usarse para *relacionar* conceptos en el modelo conceptual, solamente para *describir* estos conceptos. Una de las violaciones más comunes a esta regla consiste en agregar atributos como *llaves foráneas*

## **6.5 DISEÑO DE SISTEMAS**

### 6.5.1 ICONIX

ICONIX es un modelado de objetos conducido por casos de uso, como RUP; también es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño de éste. Este proceso también hace uso aerodinámico del UML (estándar del Object Management Group) mientras guarda un enfoque afilado en el seguimiento de requisitos. Y, el proceso se queda igual a la visión original de Jacobson del “manejo de casos de uso”, esto produce un resultado concreto, específico y casos de uso fácilmente entendibles, que un equipo de un proyecto puede usar para conducir el esfuerzo hacia un desarrollo real. El enfoque es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos. Además de las características ya mencionadas, presenta las siguientes:

- Centrado en datos: se descompone en fronteras de datos
- Basado en escenarios que descomponen los casos de uso
- Enfoque iterativo e incremental
- Ofrece trazabilidad

#### 6.5.1.1 EL DOMINIO DEL PROBLEMA

El modelo del dominio es una parte esencial del proceso de ICONIX. Construye la porción estática inicial de un modelo que es esencial al manejar su plan de la aplicación, antes de los casos del uso.

El enfoque de este tema es el modelo del dominio. El término "dominio del problema" se refiere al área que abarca cosas del mundo real y conceptos relacionados al problema que el sistema está diseñándose para resolver. El modelo del dominio es la tarea de descubrir " los objetos " (las clases) estos representan cosas y conceptos.

#### 6.5.1.2 TAREAS DE ICONIX

##### 6.5.1.2.1 ANÁLISIS DE REQUISITOS

**A.** Identificar en el “mundo real” los objetos y todas las relaciones de agregación y generalización entre ellos. Utilizar un diagrama de clases de alto nivel definido como **modelo de dominio**.

**B.** Presentar, una **prototipación** rápida de las interfaces del sistema, los diagramas de navegación, etc., de forma que los clientes puedan comprender mejor el sistema propuesto.



- C. Identificar los casos de uso del sistema mostrando los actores involucrados. Utilizar para representarlo el modelo de casos de uso.
- D. Organizar los casos de uso en grupos, o sea, utilizar los **diagramas de paquetes**.
- E. Asociar los requisitos funcionales con los casos de uso y con los objetos del dominio (trazabilidad).

#### **6.5.1.2.2 ANÁLISIS Y DISEÑO PRELIMILAR**

- A. Describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción. La principal sugerencia de ICONIX, en esta actividad, es que no se debe perder mucho tiempo con la descripción textual. Debería usarse un estilo consistente que sea adecuado al contexto del proyecto.
- B. Realizar un **diagrama de robustez**. Se debe ilustrar gráficamente las interacciones entre los objetos participantes de un caso de uso. Este diagrama permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso.
- C. Actualizar el diagrama de clases ya definido en el modelo de dominio con las nuevas clases y atributos descubiertas en los diagramas de robustez.

#### **6.5.1.2.3 DISEÑO**

- A. Especificar el comportamiento a través del **diagrama de secuencia**. Para cada caso de uso identificar los mensajes entre los diferentes objetos. Es necesario utilizar los diagramas de colaboración para representar la interacción entre los objetos.
- B. Terminar el modelo estático, adicionando los detalles del diseño en el **diagrama de clases**.
- C. Verificar si el diseño satisface todos los requisitos identificados

#### **6.5.1.2.4 IMPLEMENTACIÓN**

- A. Utilizar el diagrama de componentes, si fuera necesario para apoyar el desarrollo. Es decir, mostrar la distribución física de los elementos que componen la estructura interna del sistema.

#### **B. ESCRIBIR/ GENERAR EL CÓDIGO**

La importancia de la interactividad, interactividad, accesibilidad y navegación en el software harán que el usuario se sienta seguro y cómodo al poder hacer uso de la aplicación sin inconvenientes tales como son los

problemas de comunicación. Este y otros problemas como la realización de cambios, son factores que deben ser tenidos en cuenta.

#### **Factores a tomar en cuenta**

- **La Reusabilidad:** que es la posibilidad de hacer uso de los componente en diferentes aplicaciones.
- **La Extensibilidad:** que consiste en modificar con facilidad el software.
- **La Confiabilidad:** realización de sistemas descartando las posibilidades de error.

c. Realizar pruebas. Test de unidades, de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.

### **6.5.2 DIAGRAMA DE ROBUSTEZ**

El Análisis de Robustez, ayuda a identificar los objetos que participaran en cada caso de uso, estos objetos que forman parte de los diagramas de robustez se clasifican dentro de los tres tipos siguientes:

- **Objetos de interfaz:** Usados por los actores para comunicarse con el sistema.
- **Objetos entidad:** Son objetos del modelo del dominio.
- **Objetos de control:** Es la unión entre la interfaz y los objetos entidad.

#### **Reglas**

- Los Actores sólo pueden interactuar con los objetos límite.
- Los objetos límite sólo pueden interactuar con controladores y actores.
- Los objetos entidad sólo pueden interactuar con controladores.
- Los controladores pueden interactuar con objetos limite y objetos entidad, y con otros controladores, pero no con actores.

### **6.5.3 DISEÑO ORIENTADO A OBJETOS**

#### **6.5.3.1 DIAGRAMA DE INTERACCIÓN**

El objetivo de esta técnica es describir el comportamiento dinámico del sistema de información mediante el paso de mensajes entre los objetos del mismo. Además representa un medio para verificar la coherencia del sistema mediante la validación con el modelo de clases.

#### **Descripción**

Un diagrama de interacción describe en detalle un determinado escenario de un caso

de uso. En él se muestra la interacción entre el conjunto de objetos que cooperan en

la realización de dicho escenario. Suele ser conveniente especificar en la parte izquierda del diagrama el caso de uso que se está representando para que resulte más sencilla su validación.

### **Elementos que componen el diagrama de Interacción**

- **Un objeto** es una entidad que tiene un estado, un comportamiento e identidad. La estructura y el comportamiento común de diferentes objetos se recoge en una clase. En un diagrama de interacción, los objetos serán al final instancias de una determinada clase o de un actor.
- **Un mensaje** es una comunicación entre dos objetos. El envío de un mensaje por parte de un objeto (emisor) a otro (receptor), puede provocar que se ejecute una operación, se produzca un evento o se cree o destruya un objeto.

Hay dos tipos de diagramas de interacción: diagramas de secuencia y diagramas de colaboración. Ambos tipos de diagramas tratan la misma información pero cada uno hace énfasis en un aspecto particular en cuanto a la forma de mostrarla.

#### **6.5.3.1.1 DIAGRAMA DE SECUENCIA**

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos.

##### **Descripción**

Un diagrama de secuencia tiene dos dimensiones, el eje vertical representa el tiempo y el eje horizontal los diferentes objetos. El tiempo avanza desde la parte superior del diagrama hacia la inferior. Normalmente, en relación al tiempo sólo es importante la secuencia de los mensajes, sin embargo, en aplicaciones de tiempo real se podría introducir una escala en el eje vertical.

Respecto a los objetos, es irrelevante el orden en que se representan, aunque su colocación debería poseer la mayor claridad posible.

Cada objeto tiene asociados una línea de vida y focos de control. La línea de vida

indica el intervalo de tiempo durante el que existe ese objeto. Un foco de control o activación muestra el periodo de tiempo en el cual el objeto se encuentra ejecutando alguna operación, ya sea directamente o mediante un procedimiento

concurrente.

### **Notación**

#### **a. Objeto y línea de vida**

Un objeto se representa como una línea vertical discontinua, llamada línea de vida, con un rectángulo de encabezado con el nombre del objeto en su interior. También se puede incluir a continuación el nombre de la clase, separando ambos por dos puntos.

La línea de vida de un objeto puede desplegarse en dos o más líneas para mostrar los diferentes flujos de mensajes que puede intercambiar un objeto, dependiendo de alguna condición.

#### **b. Foco de control o activación**

Se representa como un rectángulo delgado superpuesto a la línea de vida del objeto. Su largo dependerá de la duración de la acción. La parte superior del rectángulo lo indica el inicio de una acción ejecutada por el objeto y la parte inferior su finalización.

#### **c. Mensaje**

Un mensaje se representa como una flecha horizontal entre las líneas de vida de los objetos que intercambian el mensaje. La flecha va desde el objeto que envía el mensaje al que lo recibe. Además, un objeto puede mandarse un mensaje a sí mismo, en este caso la flecha comienza y termina en su línea de vida.

### **Tipos**

- **Mensaje sincrónico:** Es el utilizado con mayor frecuencia. Su uso significa que el expeditor del mensaje espera que la activación del método mencionado por el destinatario finalice antes de continuar su actividad.
- **Mensaje asincrónico:** El expeditor no espera el término de la activación invocada por el destinatario. Esto se produce al modelar sistemas en los que los objetos pueden funcionar en paralelo (es el caso de los sistemas multi-thread, donde los tratamientos se efectúan en paralelo).

#### **6.5.3.1.2 DIAGRAMAS DE COLABORACION**

El diagrama de colaboración es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información mostrando

cómo interactúan los objetos entre sí, es decir, con qué otros objetos tiene vínculos o intercambia mensajes un determinado objeto.

### **Descripción**

Un diagrama de colaboración muestra la misma información que un diagrama de secuencia pero de forma diferente. En los diagramas de colaboración no existe una secuencia temporal en el eje vertical; es decir, la colocación de los mensajes en el diagrama no indica cuál es el orden en el que se suceden. Además, la colocación de los objetos es más flexible y permite mostrar de forma más clara cuáles son las colaboraciones entre ellos.

### **Notación**

- **Objeto:** Se representa con un rectángulo dentro del que se incluye el nombre del objeto y, si se desea, el nombre de la clase, separando ambos por dos puntos.
- **Vínculo:** En el diagrama, un vínculo se representa como una línea continua que une ambos objetos y que puede tener uno o varios mensajes asociados en ambas direcciones. Como un vínculo instancia una relación de asociación entre clases, también se puede indicar la navegabilidad del mismo mediante una flecha.
- **Mensaje:** Se representa con una pequeña flecha colocada junto a la línea del vínculo al que está asociado. La dirección de la flecha va del objeto emisor del mensaje al receptor del mismo. Junto a ella, se coloca el nombre del mensaje y sus argumentos.

## **6.5.4 DIAGRAMAS DE CLASES**

El objetivo principal de este modelo es la representación de los aspectos estáticos del sistema, utilizando diversos mecanismos de abstracción (clasificación, generalización, agregación).

### **Descripción**

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal.

### **Clases**

Una clase describe un conjunto de objetos con propiedades (atributos) similares y un comportamiento común. Los objetos son instancias de las clases. No existe un procedimiento inmediato que permita localizar las clases del diagrama de clases.

## **Relaciones**

### **a. Asociación.**

Las relaciones de asociación representan un conjunto de enlaces entre objetos o instancias de clases. Es el tipo de relación más general, y denota básicamente una dependencia semántica.

### **b. Herencia**

Las jerarquías de generalización/especialización se conocen como herencia. Herencia es el mecanismo que permite a una clase de objetos incorporar atributos y métodos de otra clase, añadiéndolos a los que ya posee. Con la herencia se refleja una relación “es un” entre clases. La clase de la cual se hereda se denomina superclase, y la que hereda subclase

### **c. Agregación**

La agregación es un tipo de relación jerárquica entre un objeto que representa la totalidad de ese objeto y las partes que lo componen. Permite el agrupamiento físico de estructuras relacionadas lógicamente.

### **d. Composición**

La composición es una forma de agregación donde la relación de propiedad es más fuerte, e incluso coinciden los tiempos de vida del objeto completo y las partes que lo componen.

### **e. Dependencia**

Una relación de dependencia se utiliza entre dos clases o entre una clase y una interfaz, e indica que una clase requiere de otra para proporcionar alguno de sus servicios.

### **f. Notación**

**Clase:** Se representa como una caja, separada en tres zonas por líneas horizontales. En la zona superior se muestra el nombre de la clase y propiedades generales como el estereotipo. El nombre de la clase aparece centrado y si la clase es abstracta se representa en cursiva. El estereotipo, si se muestra, se sitúa sobre el nombre y entre el símbolo: << .... >>.

**Relaciones:** Se representa como una línea continua entre las clases asociadas. En una relación de asociación, ambos extremos de la línea pueden conectar con la misma clase, indicando que una instancia de una clase, está asociada a otras instancias de la misma clase, lo que se conoce como asociación reflexiva.

La relación puede tener un nombre y un estereotipo, que se colocan junto a la

línea. El nombre suele corresponderse con expresiones verbales presentes en las especificaciones, y define la semántica de la asociación. Los estereotipos permiten clasificar las relaciones en familias y se escribirán entre el símbolo: << ... >>.

### Propiedades de la Relación

- **Multiplicidad:** La multiplicidad puede ser un número concreto, un rango o una colección de números. La letra „n” y el símbolo „\*” representan cualquier número.
- **Orden:** Se puede especificar si las instancias guardan un orden con la palabra clave „{ordered}”. Si el modelo es suficientemente detallado, se puede incluir una restricción que indique el criterio de ordenación.
- **Navegabilidad:** La navegación desde una clase a la otra se representa poniendo una flecha sin relleno en el extremo de la línea, indicando el sentido de la navegación.

### Notas específicas

- **Agregación:** Se representa con un rombo hueco en la clase cuya instancia es una agregación de las instancias de la otra.
- **Composición:** Se representa con un rombo lleno en la clase cuya instancia contiene las instancias de la otra clase.
- **Dependencia:** Una línea discontinua con una flecha apuntando a la clase cliente. La relación puede tener un estereotipo que se coloca junto a la línea, y entre el símbolo: << ... >>.
- **Herencia:** Esta relación se representa como una línea continua con una flecha hueca en el extremo que apunta a la superclase.

## 6.5.4.1 DIAGRAMAS DE PAQUETES

- Los paquetes ofrecen un mecanismo general para la organización de los modelos/subsistemas agrupando elementos de modelado
- Cada paquete corresponde a un submodelo (subsistema) del modelo (sistema)
- Un paquete puede contener otros paquetes, sin límite de anidamiento pero cada elemento pertenece a (está definido en) sólo un paquete
- Una clase de un paquete puede aparecer en otro paquete por la importación a través de una relación de dependencia entre paquetes
- Todas las clases no son necesariamente visibles desde el exterior del paquete, es

decir, un paquete encapsula a la vez que agrupa

- El operador “::” permite designar una clase definida en un contexto distinto del actual.

#### **6.5.4.2 DIAGRAMAS DE COMPONENTES**

- Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, bibliotecas cargadas dinámicamente, etc.
- Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente
- Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones.
- Muestran las opciones de realización incluyendo código fuente, binario y ejecutable

#### **6.5.4.3 DIAGRAMA DE DESPLIEGUE**

- Los Diagramas de Despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos
- Los estereotipos permiten precisar la naturaleza del equipo:
  - ❖ Dispositivos
  - ❖ Procesadores
  - ❖ Memoria
- Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse.



## **6.6 BASE DE DATOS**

### **6.6.1 BASES DE DATOS ESTÁTICAS**

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

### **6.6.2 BASES DE DATOS DINÁMICAS**

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub.

Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Base de Datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

Los SBD pueden ser estudiados desde 3 niveles distintos:

#### **6.6.2.1 NIVEL FÍSICO.**

Es el nivel real de los datos almacenados. Es decir cómo se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar cualificadas para ello. Este nivel lleva asociada una representación de los datos, que es lo que denominamos Esquema Físico.

### **6.6.2.2 NIVEL CONCEPTUAL.**

Es el correspondiente a una visión de la base de datos desde el punto de vista del mundo real. Es decir tratamos con la entidad u objeto representado, sin importarnos como está representado o almacenado. Este nivel lleva asociado el Esquema Conceptual.

### **6.6.2.3 NIVEL VISIÓN.**

Son partes del esquema conceptual. El nivel conceptual presenta toda la base de datos, mientras que los usuarios por lo general sólo tienen acceso a pequeñas parcelas de ésta. El nivel visión es el encargado de dividir estas parcelas. Un ejemplo sería el caso del empleado que no tiene por qué tener acceso al sueldo de sus compañeros o de sus superiores. El esquema asociado a éste nivel es el Esquema de Visión. Los 3 niveles vistos, componen lo que conocemos como arquitectura de base de datos a 3 niveles.

A menudo el nivel físico no es facilitado por muchos DBMS, esto es, no permiten al usuario elegir como se almacenan sus datos y vienen con una forma estándar de almacenamiento y manipulación de los datos. La arquitectura a 3 niveles se puede representar como sigue: Subesquema de Visión, Subesquema de Visión, Subesquema de Visión.

## **6.6.3 MODELOS DE BASES DE DATOS**

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- **Encapsulación** - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia** - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo** - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado.

Esto podría denominarse independencia entre programas y operaciones.

SQL: 2003, es el estándar de SQL92 ampliado, soporta los conceptos orientados a objetos y mantiene la compatibilidad con SQL92.

## 6.7 LIBRERÍAS JAVA

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código toma la forma, normalmente, de un conjunto de librerías dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Pero la Plataforma Java está pensada para ser independiente del sistema operativo subyacente, por lo que las aplicaciones no pueden apoyarse en funciones dependientes de cada sistema en concreto. Lo que hace la Plataforma Java, es ofrecer un conjunto de librerías estándar, que contiene mucha de las funciones reutilizables disponibles en los sistemas operativos actuales.

Las librerías de Java tienen tres propósitos dentro de la Plataforma Java. Al igual que otras librerías estándar, ofrecen al programador un conjunto bien definido de

funciones para realizar tareas comunes, como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres.

Además, las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas tales como manejo de las funciones de red o acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. En el caso concreto anterior, las librerías `java.net` y `java.io` implementan el código nativo internamente, y ofrecen una interfaz estándar para que aplicaciones Java puedan ejecutar tales funciones.

Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente.

#### **Implementaciones:**

- TopLink
- Swing ec
- SyntheticaBlueMoon

### **6.7.1 TOPLINK**

El Enterprise JavaBeans (EJB) 3.0 incluye una especificación de persistencia adicional llamado Java Persistence API (JPA). Se trata de una API para crear, eliminar y consultar a través de los objetos Java de peso ligero y se puede utilizar tanto dentro de un contenedor EJB 3.0 compatible y un entorno estándar de Java EE5.

Cuando se utiliza TopLink JPA, puede configurar el comportamiento de la APP de las entidades que utilizan sus anotaciones. Una anotación es un medio simple y expresivo del código de Java con la decoración de fuente de metadatos que se compila a los correspondientes ficheros de clase Java para la interpretación en tiempo de ejecución TopLink JPA para manejar la conducta de la APP.

Oracle TopLink es un líder en la comunidad de la APP(Application Persistence ) entrega de los 1,0 implementación de referencia de la APP con TopLink Essentials, y

ahora el desarrollo de la implementación de referencia de la APP 2,0 con EclipseLink. A partir de Oracle TopLink 11g, TopLink Essentials ha sido sustituido por EclipseLink APP. EclipseLink APP implementa JPA 1.0 y es responsable de entregar el marco de la persistencia.

## **6.8 JPA**

Java Persistence API, más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma Java EE (Java Platform, Enterprise Edition o Java EE anteriormente conocido como Java 2 Platform, Enterprise Edition) e incluida en el estándar EJB3 (Los Enterprise JavaBeans también conocidos por sus siglas EJB).

Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs acrónimo de Plain Old Java Object).

### **6.8.1 ASOCIACIONES JPA**

#### **6.8.1.1 Especificación de Entidades JPA**

Cada tabla de la BD estará representada en Java por una Entidad

- Una entidad es una clase Java marcada con la anotación @Entity
- Cada instancia de esa clase Entidad mapeará una tupia de esa tabla
- Entidades deben ser JavaBeans deben contener 1 constructor vacío y get() y set() cada atributo
- Todos los atributos de la Entidad son persistentes y se corresponderán con columnas de la tabla mapeada
- Cada instancia de una Entidad es única y debe estar identificada por uno o más de sus atributos
- Anotaciones para el mapeo de atributos

Pueden asociarse a la declaración del atributo en la entidad o al método get() de ese atributo

#### **6.8.1.2 @Column:**

Especifica una columna de la tabla a mapear sobre un atributo de la entidad.

Es opcional: si no se especifica lo contrario se asume que todos los atributos de la entidad mapean una columna de la BD con su mismo nombre y tipo

- name (nombre de la columna)
- unique (atributo con valor único)
- nullable (permite nulos)
- insertable, updatable
- length, precisión, longitud

### 6.8.1.3 @Id

Especifica que se trata de un atributo identificador, JPA exige que toda entidad debe tener un Id

### 6.8.1.4 @GeneratedValue

Asociado a atributos clave primaria, indica como se debe generar Strategy estrategia para la generación de clave

- GenerationType.AUTO( valor por defecto autoincremental gestionado por la BD)
- GenerationType.IDENTITY(utiliza un contador autoincremental gestionado por la BD)
- GenerationType.SEQUENCE( utiliza una secuencia)
- GenerationType.TABLE( utiliza una tabla de identificadores)

### 6.8.1.5 @Temporal

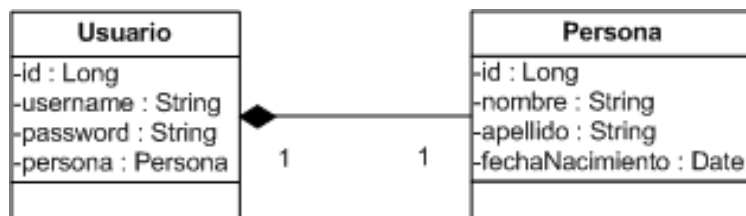
- Especifica el mapeo de campos de BD que almacenan fechas.

### 6.8.1.6 @JoinColumn

Especifica un atributo que actúa como clave foránea de otra entidad mappedBY campo de la entidad relacionada que posee la relación (únicamente se especifica en uno de los lados de la relación[el que no tiene la clave foránea])

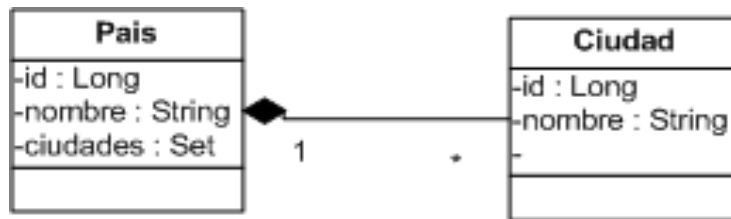
### 6.8.1.7 @OneToOne

Cada entidad se relaciona con una sola instancia de otra entidad. Donde ambas se referencian por la misma llave PK.



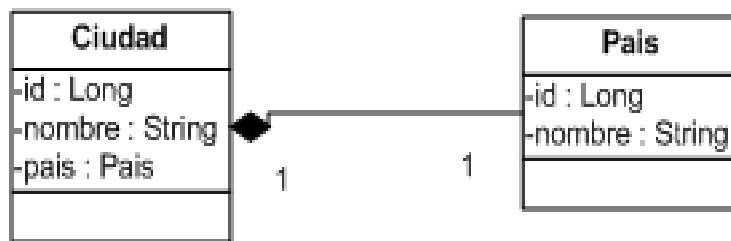
### 6.8.1.8 @OneToMany

Una entidad, puede estar relacionada con varias instancias de otras entidades



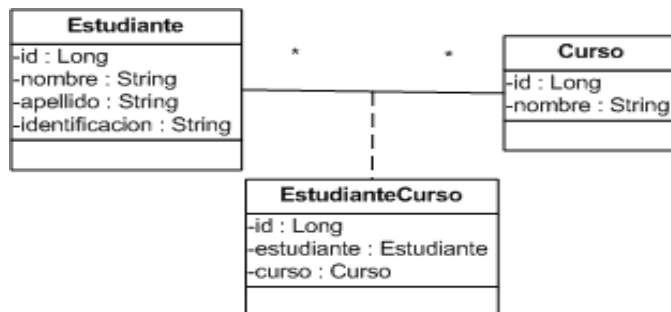
### 6.8.1.9 @ManyToOne

Múltiples instancias de una entidad pueden estar relacionadas con una sola instancia de otra entidad. Esta multiplicidad es lo contrario a la relación uno a muchos. Usado también como si fuera una relación OneToOne



### 6.8.1.10 @ManyToMany

En este caso varias instancias de una entidad pueden relacionarse con múltiples instancias de otras entidades.



## 6.9 GRAPHICS2D

Esta clase Graphics2D extiende la clase Graphics para proporcionar un control más sofisticados en la geometría, transformaciones de coordenadas, gestión de color y el diseño de texto. Esta es la clase fundamental para la prestación de 2-formas tridimensionales, textos e imágenes en Java (TM).

### Espacios de coordenadas:

Todas las coordenadas pasa a un objeto Graphics2D se especifican en un dispositivo independiente del sistema de coordenadas llamado el espacio del usuario, que es utilizado por las aplicaciones. El objeto Graphics2D contiene un objeto AffineTransform como parte de su estado de representación que define cómo convertir coordenadas de espacio de usuario al dispositivo dependen de las coordenadas en el espacio de dispositivo.

Algunas de las operaciones realizadas por los objetos de atributo de representación se producen en el espacio de dispositivo, pero todos los métodos Graphics2D tener en el espacio de usuario coordenadas.

Cada objeto Graphics2D se asocia con un objetivo que define dónde se lleva a cabo la prestación. Un objeto GraphicsConfiguration define las características de la meta de representación, como el formato de píxeles y resolución. El objetivo de representación se utiliza el mismo durante toda la vida de un objeto Graphics2D. Al crear un objeto Graphics2D el GraphicsConfiguration especifica el valor predeterminado de transformar para el objetivo de la Graphics2D. Este valor por defecto de transformar los mapas del sistema de coordenadas del espacio de usuario a la pantalla y el dispositivo de impresora coordenadas tal que los mapas de origen a la esquina superior izquierda de la región de orientación del dispositivo con el aumento de las coordenadas X se extiende a la derecha y las coordenadas Y cada vez se extiende hacia abajo.



## 7. MATERIALES Y METODOS

### 7.1 METODOS

Para la elaboración y solución del presente trabajo investigativo, recurriremos al empleo de métodos, técnicas y procedimientos que permitirán desarrollar con éxito el mismo.

Con la autorización previa de la directora de la Escuela Fiscal Mixta Julio María Matovelle se tendrá acceso a la recolección de datos, lo cual permitirá realizar el diseño requerido para la construcción del Sistema Informático para la Gestión Académica de la Escuela “Julio María Matovelle”.

La presente investigación basada principalmente en el diseño se desarrollará enfocándose a los siguientes métodos:

**El Método Deductivo y el Inductivo**, los cuales siguen el proceso analítico-sintético y satisfacen los requerimientos propios de las ciencias informáticas (recolección de datos, análisis de la información, diseño del sistema e implementación).

Con los resultados cualitativos y analíticos que obtuvimos permitirán establecer las conclusiones y desarrollar el sistema informático que beneficiará a la institución en la gestión académica.

- Para poder recolectar la información útil nos apoyaremos en las siguientes técnicas.

**La Observación:** El propósito de la observación es múltiple, nos permitirá determinar ¿qué se está haciendo?, ¿cómo se está haciendo?, ¿quién lo hace?, ¿cuándo se lleva a cabo?, ¿cuánto tiempo toma?, ¿dónde se hace? y ¿por qué se hace?.

**La Entrevista:** Determinar que la información manifestada por la asistente administrativa sea verídica, la misma permitirá reflejar la situación real de dicha organización, así también servirá para cumplir los objetivos planteados.

**Iconix:** es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos, con el objetivo de abarcar todo el ciclo de vida de un proyecto.

Características fundamentales de ICONIX son:

- **Iterativo e Incremental.-** Varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- **Trazabilidad:** cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre diferentes artefactos producidos.
- **Dinámica UML.-** La metodología ofrece un uso dinámico del UML como los diagramas del caso de uso, diagramas de secuencia y de colaboración.

## 7.2 METODOS

### Recursos Humanos

RECURSOS HUMANOS	Numero Horas	COSTO
<ul style="list-style-type: none"> <li>• Diseñadores:</li> </ul>		
Darwin Gualotuña	• 400	00.00
Carlos Iñiguez	• 400	00.00
Luis Soto	• 400	00.00
Mariuxi Salinas	• 400	00.00
<ul style="list-style-type: none"> <li>• <b>Coordinador:</b></li> </ul>		
Ing. Patricio Villamarín	• 128	00.00
<b>TOTAL:</b>		00.00

Tabla 1.1

### Recursos Materiales

RECURSOS MATERIALES	COSTO
Resma de papel	5.00
Impresiones	10.00
Copias	5.00
Anillados	5.00
<b>TOTAL:</b>	25.00

Tabla 1.2

**Recursos Técnicos**

<b>RECURSOS TECNICOS</b>	<b>COSTO</b>
Alquiler de Computadores	15.00
Cartuchos de Impresora	8.00
Programas de Diseño	00.00
<b>TOTAL:</b>	23.00

**Tabla 1.3****Recursos Tecnológicos**

<b>RECURSOS TECNOLÓGICOS</b>	<b>COSTO</b>
Internet	15.00
Proyector	8.00
<b>TOTAL:</b>	23.00

**Tabla 1.4**

## 8. RESULTADOS

### 8.1. REQUERIMIENTOS NO FUNCIONALES

#### 8.2. 1. FUNCIONES BÁSICA:

Referencia	Función	Categoría	Atributo	Detalle y descripción	Categoría
R1.1	Reduce el tiempo de ejecución de los procesos de gestión académica	Oculta	Tiempo de respuesta	2 segundos para guardar la información. Reduce el tiempo en un 80% con respecto al proceso manual	Obligatorio
			Metáfora de interfaz	Ventanas amigables con el docente por sus bajos conocimientos	Obligatorio
R 1.2	Digitalización de clave para el ingreso al sistema.	Evidente	Tolerancia a fallas	Si la clave es errónea	Obligatorio
			Tiempo de respuesta	1 segundo para la comprobación de clave	Obligatorio
			Metáfora de interfaz	Cuadro de dialogo que indique si la clave es errónea o correcta	Obligatorio
R 1.3	Reduce el uso de	Evidente	Metáfora	Ventanas con el	Obligatorio

	registros manuales en la gestión académica		de interfaz	esquema de los parámetros establecidos en registros manuales	
R 1.4	Muestra la descripción del estado actual del estudiante.	Evidente	Tiempo de respuesta	2 segundos como máximo para mostrar	Obligatorio
			Metáfora de interfaz	Ventana de fácil acceso para mostrar esa información	Obligatorio
R 1.5	Ofrece un mecanismo de almacenamiento persistente.	Oculta	Tolerancia a fallas	Recuperar datos almacenados por si acaso ocurriera alguna falla en el equipo	Obligatorio
			Tiempo de respuesta	25 segundos para almacenar. Siendo disponible en todo el año escolar	Obligatorio
R 1.6	Permite ingresar datos del docente y actualizarlos	Evidente	Tiempo de Respuesta	2 segundos máximo para almacenar satos	Obligatorio

			Metáfora de Interfaz	Ventana que permita ingresar los datos del docente y que permita modificarlos (actualizarlos) si es necesario	Obligatorio
R 1.7	Permite controlar y actualizar los datos de la Directora.	Evidente	Tiempo de respuesta	2 seg como máximo para almacenar datos	Obligatorio
			Metáfora de interfaz	Ventana donde se puedan ingresar y actualizar los datos de la Directora	Obligatorio
R 1.9	Permite controlar y actualizar los datos de la Secretaria	Evidente	Tiempo de respuesta	2 seg como máximo para almacenar la información	Obligatorio
			Metáfora de interfaz	Ventana en donde se puedan ingresar los datos de la secretaria y donde contenga un campo para ingresar la clave	Obligatorio

				ya que ella administrará el sistema	
--	--	--	--	-------------------------------------	--

Tabla 8.2.1

### 8.1.2. FUNCIONES DE MATRICULACIÓN

Referencia	Función	Categoría	Atributo	Detalles y descripción	Categoría
R 2.1	Registra la matrícula del estudiante con los datos personales respectivo	Evidente	Tiempo de respuesta	2 segundos para guardar la información	Obligatorio
			Metáfora de interfaz	Ventana para llenar los requisitos (información estudiantes) de matrícula. Cuadro de dialogo que presente un mensaje si no están llenos todos los datos obligatorios.	Obligatorio
R 2.2	Emite comprobante de matrícula del estudiante al representante.	Superflua	Tiempo de respuesta	5 segundos como máximo	Opcional
			Metáfora de interfaz	Ventana con un modelo establecido para emitir el	Opcional

				comprobante	
R 2.3	Permite modificar (actualizar) los datos del estudiante y docente.	Evidente	Tiempo de respuesta	Para actualizar (guardar esta información) 1 segundo. Siendo posible para docentes y alumnos.	Obligatorio
			Metáfora de interfaz	Ventana con datos del estudiante y docente para poder modificarlos previa su búsqueda. Cuadro de dialogo que muestre que se modificaron correctamente	Obligatorio
R 2.4	Calcula la edad del estudiante cada año para matricularlo.	Oculto	Tiempo de respuesta	Según fecha de nacimiento calcula la edad cada año en un máximo de 2 segundos	Obligatorio
R 2.5	Emite un listado de los alumnos matriculados por cada paralelo y uno de sus	Evidente	Tiempo de respuesta	10 segundos por ser dos listados. Siendo disponible durante todo el año lectivo	Obligatorio



	representantes.				
R 2.6	Permite actualizar el listado de un paralelo con respecto a los alumnos nuevos o retirados.	Evidente	Tiempo de respuesta	2 segundos para actualizar(guardar información)	Obligatorio
			Metáfora de interfaz	Ventana donde se presente el listado de un paralelo que se necesite para poder actualizarlo fácilmente	Obligatorio
R 2.7	Permite emitir un certificado de matrícula con el número respectivo y las asistencias del estudiante.	Evidente	Tiempo de respuesta	6 segundos como máximo	Obligatorio
			Metáfora de interfaz	Ventana con un modelo establecido del certificado de matrícula	Obligatorio

Tabla 8.1.2

### 8.1.3. FUNCIONES DE ACREDITACIÓN

Referencia	Función	Categoría	Atributo	Detalles y descripción	Categoría
R 3.1	Registra las notas de cada estudiante	Evidente	Tiempo de respuesta	2 segundos para registrar. Habilitado en los 3 trimestres y en días establecidos	Obligatorio
			Metáfora de interfaz	Ventanas para almacenar las	Obligatorio

				notas de cada área y para cada estudiante	
R 3.2	Calcula el promedio trimestral y final del estudiante	Oculta	Tiempo de respuesta	2 segundos. Siendo realizado en los 3 trimestres y al finalizar el año escolar	Obligatorio
			Tolerancia a fallas	No admite notas de cero ni mayores a 20	Obligatorio
R 3.3	Emite una boleta de calificaciones de cada estudiante por trimestre	Evidente	Tiempo de respuesta	6 segundos como máximo	Obligatorio
			Metáfora de interfaz	Ventana con un modelo de libreta de calificaciones	Obligatorio
R 3.4	Permite la corrección de errores al ingresar mal las notas:	Evidente	Tiempo de respuesta	1 segundo para guardar cambio. Habilitado todo el año escolar	Obligatorio
			Metáfora de interfaz	Cuadro de dialogo con un mensaje que especifique que se guardaron todos los cambios	Obligatorio
			Tolerancia a fallas	Permite(tolera) el cambio de la nota	Obligatorio

				mal ingresada	
R 3.5	Emite un informe al finalizar el año escolar de cada estudiante sobre el rendimiento académico y conducta (informe final)	Evidente	Tiempo de respuesta	6 segundos como máximo	Obligatorio
			Metáfora de interfaz	Ventana con el modelo de libreta de calificaciones de todos los trimestres, promedio final y con un informe final	Obligatorio
R 3.6	Permite contabilizar el total de días asistidos y no asistidos al finalizar el año escolar	Oculta	Tiempo de respuesta	Se tiene como plazo 24 horas después de haber ingresado las notas del tercer trimestre para su posterior entrega al padre de familia	Obligatorio
			Tolerancia a fallas	Que el sistema detecte después de la fecha establecida si algún docente no ha ingresado asistencias	Obligatorio
R 3.7	Emite un reporte de las notas de 7mo año, para los alumnos que ya terminan la	Evidente	Tiempo de respuesta	6 a 7 segundos en cualquier día laborable del año escolar	Obligatorio
			Tolerancia a fallas	Se toma en cuenta que solo se puede emitir a alumnos	Obligatorio

	educación básica.			que ya terminaron la educación primaria	
			Metáfora de interfaz	En caso de que el alumno no haya culminado la educación primaria que presente un cuadro de dialogo de que existe un error por no encontrar esa información	Obligatorio

Tabla 8.1.3

## 8.2 CASOS DE USO

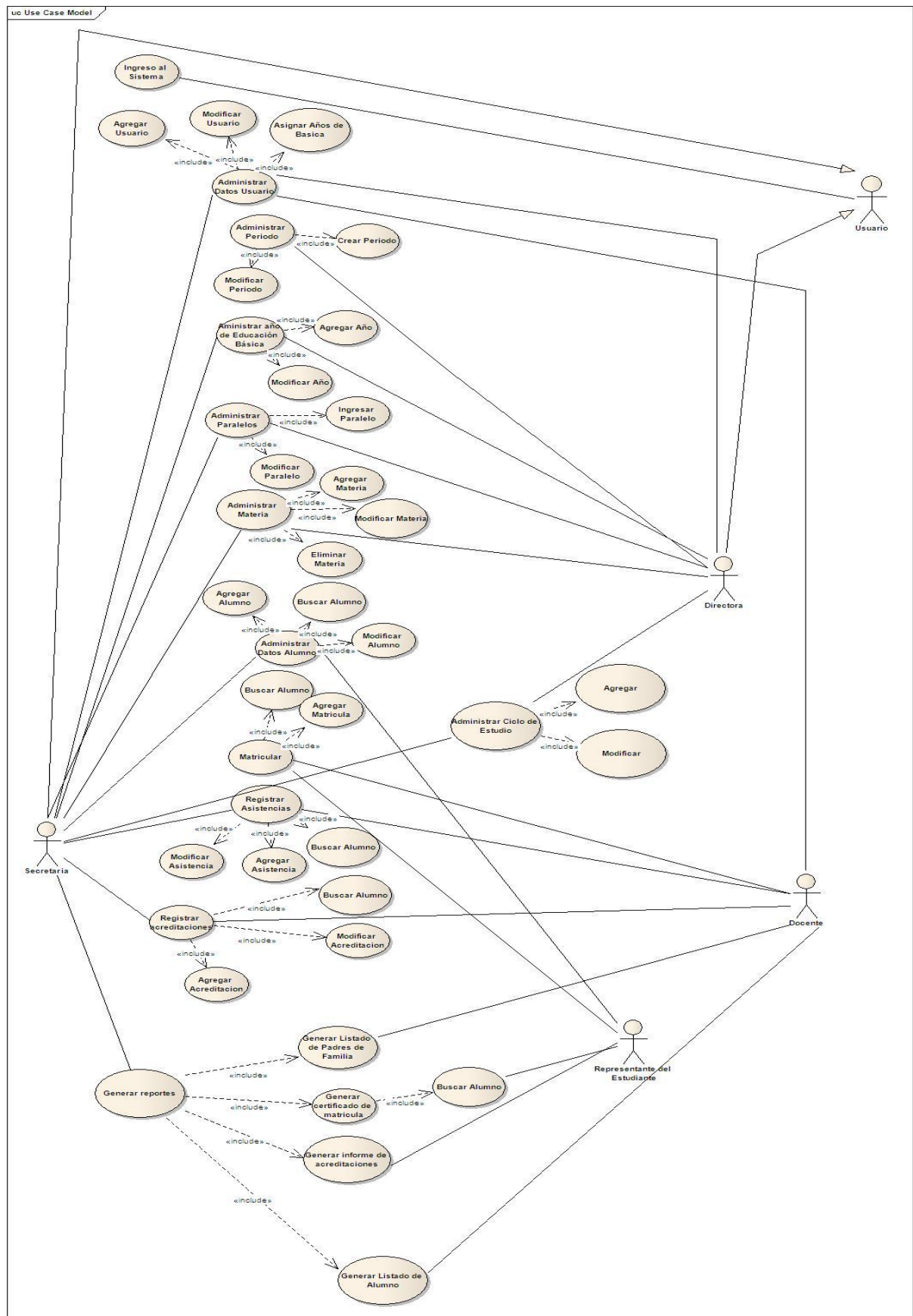


Fig. 8.2

### 8.3. DESCRIPCIÓN DE CASOS DE USO EXPANDIDOS

#### 8.3.1. Caso de Uso 1: Ingreso al Sistema



fig. 8.3.1

<b>Caso de uso 001</b>	Ingreso al Sistema
<b>Actores</b>	Docente, Secretaria, Directora (iniciadores)
<b>Propósito</b>	Ingreso al sistema por parte de los usuarios
<b>Resumen</b>	Al momento de abrir el programa se presentara una ventana para que se pueda ingresar; donde podrán acceder ya sea la secretaria el docente o la directora pero con la respectiva clave.
<b>Tipo</b>	Primario y esencial
<b>Referen. Cruzadas</b>	R1.2

##### 8.3.1.1. CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Comienza cuando la Directora, Secretaria o Docente desean ingresar al sistema.	2. El sistema presenta la ventana principal.
3. Los usuarios ingresan la clave en la caja de contraseña y hace clic en ingresar.	4. Presenta la ventana principal correspondiente al usuario.
5. Si desea cancelar el ingreso del sistema el usuario hace clic en cancelar.	6. El sistema cierra el sistema

### Cursos alternos

- Ítem 4: El sistema controla que la clave ingresada sea correcta y verifica el tipo de Usuario, de lo contrario presentará un mensaje de error. Controla tres intentos como máximo al ingreso al sistema. Caso contrario el sistema se cerrará automáticamente.

#### 8.3.1.2. Diagrama de robustez

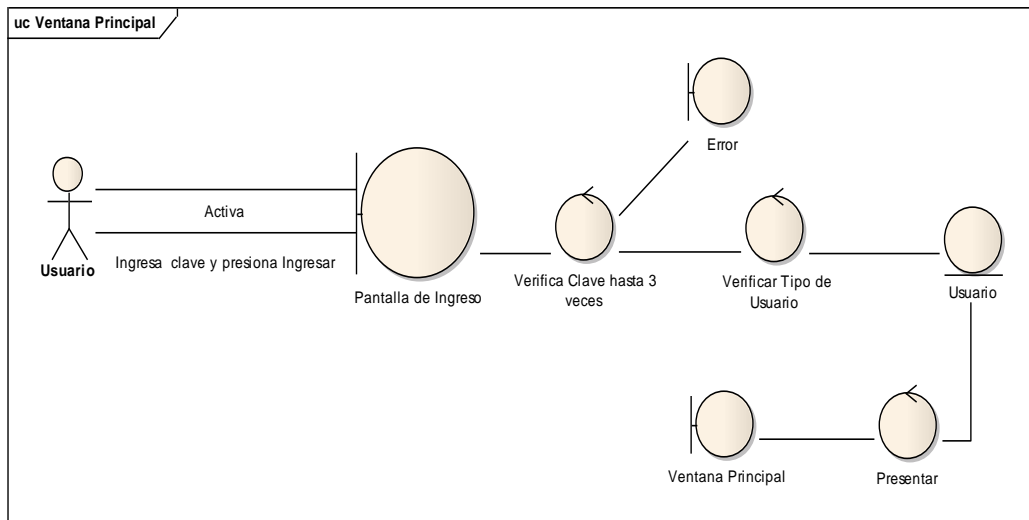


fig. 8.3.1.2

#### 8.3.1.3. Diagrama de Secuencia

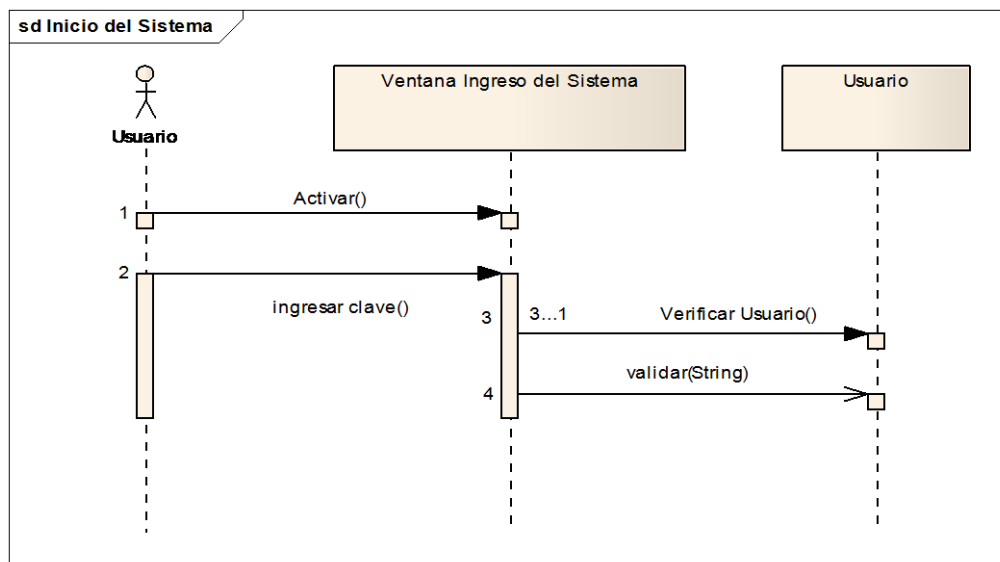


fig. 8.3.1.3

### 8.3.1.4. Diagramas de Colaboración

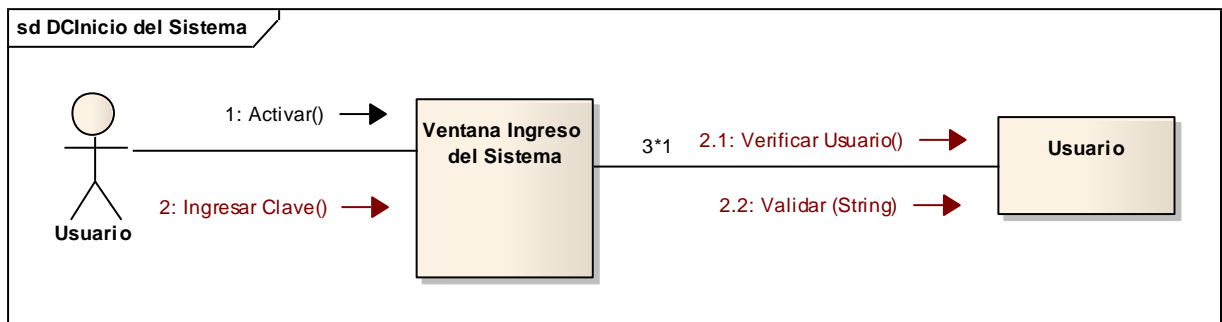


fig. 8.3.1.4

### 8.3.2. Caso de Uso 2: Administrar datos Usuario

fig. 8.3.2

<b>Caso de uso 002</b>	Administrar datos Usuario
<b>Actores</b>	Secretaria (iniciadora), Docente, Directora
<b>Propósito</b>	Ingresar y actualizar los datos de los usuarios
<b>Resumen</b>	La secretaria ingresa al sistema a la ventana administrar usuario e ingresa los datos correspondientes ya sea del docente directora o datos suyos.
<b>Tipo</b>	Primario y esencial
<b>Referencias</b>	R1.7, R1.9, R 2.0



<b>cruzadas</b>	
-----------------	--

**Tabla. 8.3.2****8.3.2.1. CURSO NORMAL DE EVENTOS**

Acción del actor	Respuesta del sistema
1. Comienza cuando hay nuevos Usuarios (Directora, Secretaria y Docente) o actualizaciones de sus datos.	
2. La secretaria va al menú administrar y al ítem administrar Usuario.	3. El sistema presenta la ventana Administrar Usuario.
4. Luego si se desea agregar Usuario ir a la sección Agregar Usuario	
5. Si se requiere alguna modificación con respecto a los datos del Usuario ir a la sección Modificar Usuario.	
6. Si el Usuario es un Docente, ir a la sección Asignar Año.	
7. Si se desea salir de la ventana la secretaria hace clic en cancelar.	8. El sistema cierra la ventana administrar Usuario.

**Tabla 8.3.2.1****8.3.2.2. Sección: Agrega Usuario**

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea agregar nuevos Usuarios	2. En la ventana Administrar Usuario presenta el botón Nuevo. 3. El sistema habilita todos los campos para ingresar un Usuario.
4. La secretaria llena los datos de los	5. Agrega los nuevos Usuarios

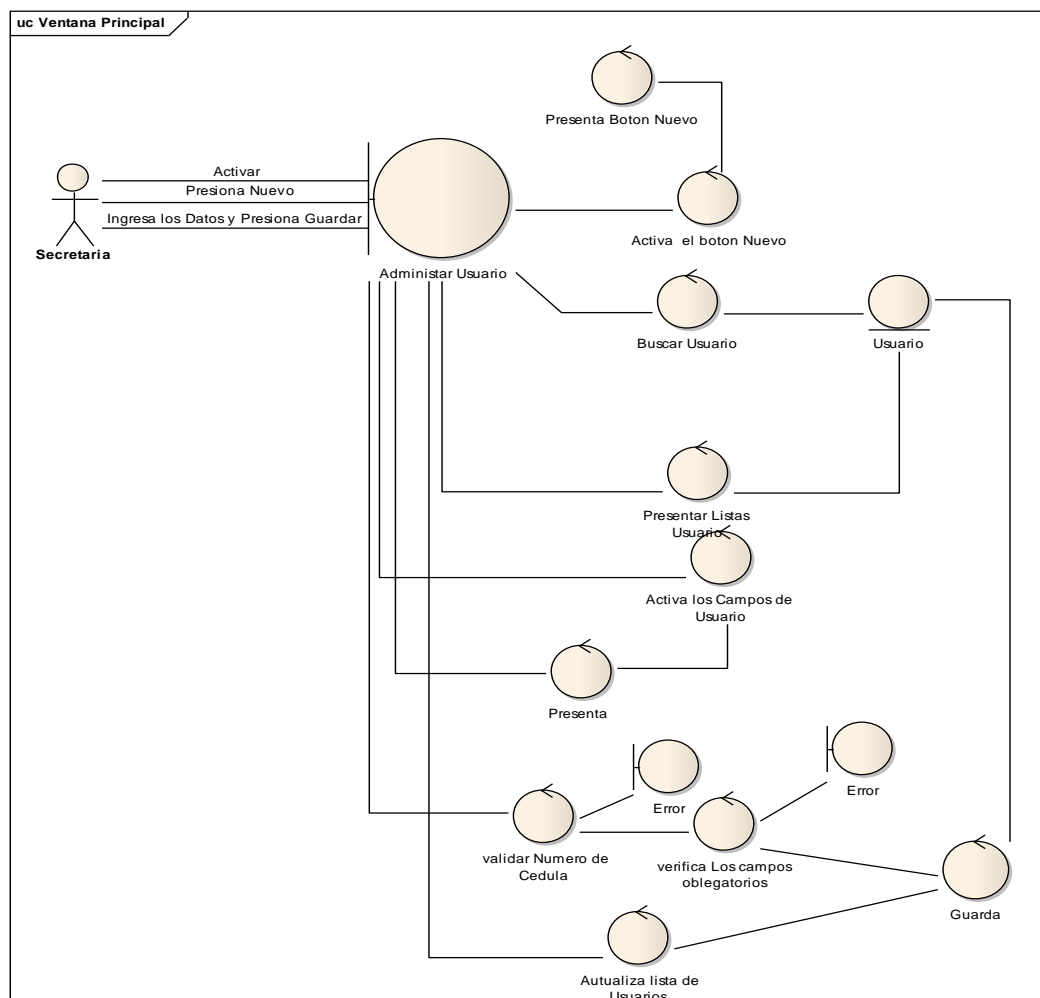
Usuarios a Agregar y presiona el botón guardar.	<p>almacenando la información en la base de datos.</p> <p>6. El sistema agrega los usuarios en una lista usuario, ubicada en el panel lista usuarios.</p>
---	---

**Tabla. 8.3.2.2**

### Cursos alternos

- Ítem 4: El sistema controla que todos los campos obligatorios no estén vacíos de estarlo al momento de almacenar presentará un mensaje de error.

### 8.3.2.1. Diagrama de Robustez



**fig. 8.3.2.1**

### 8.3.2.2.2. Diagrama de Secuencia

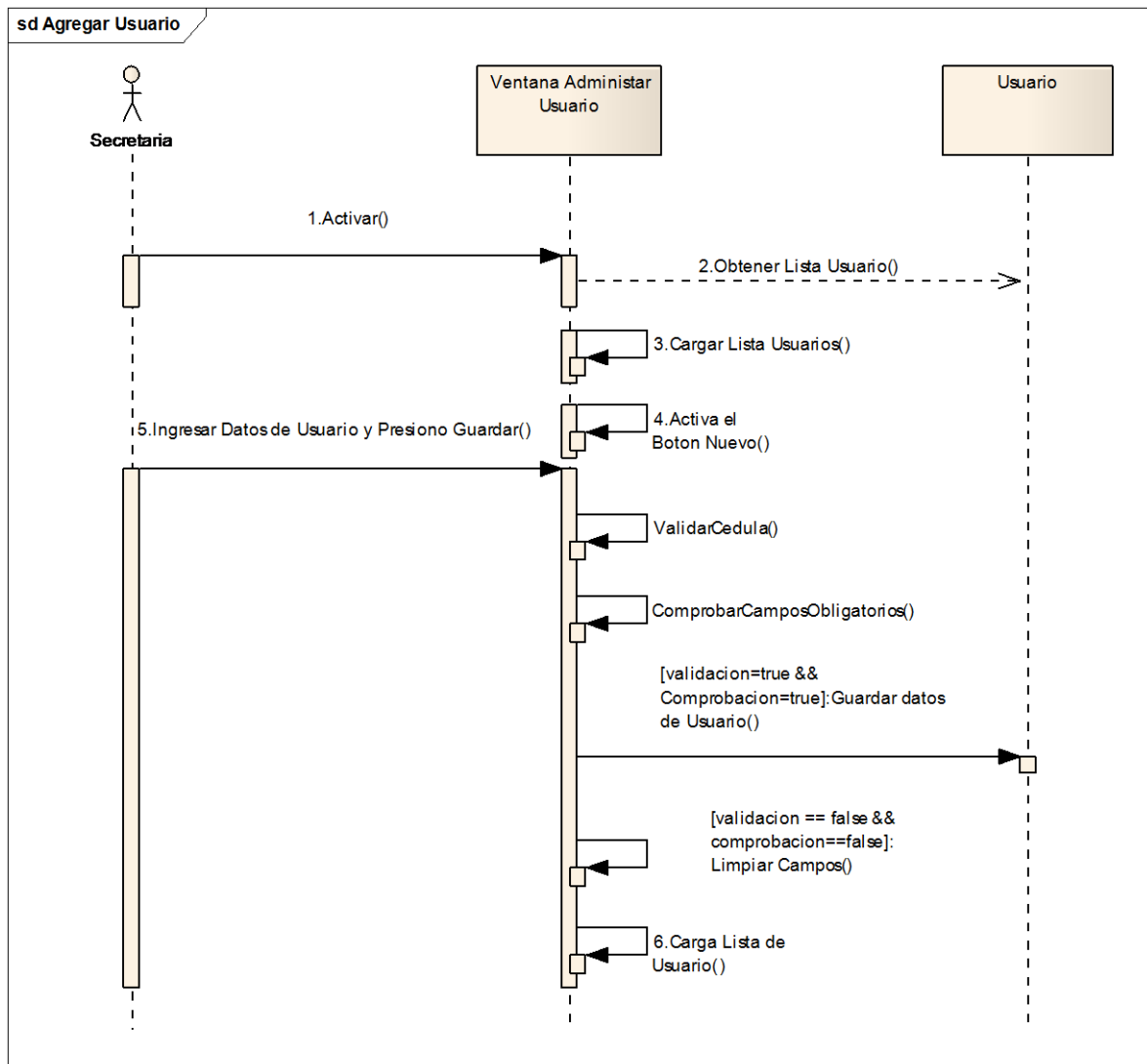


fig. 8.3.2.2.2

### 8.3.2.2.3. Diagrama de Colaboración

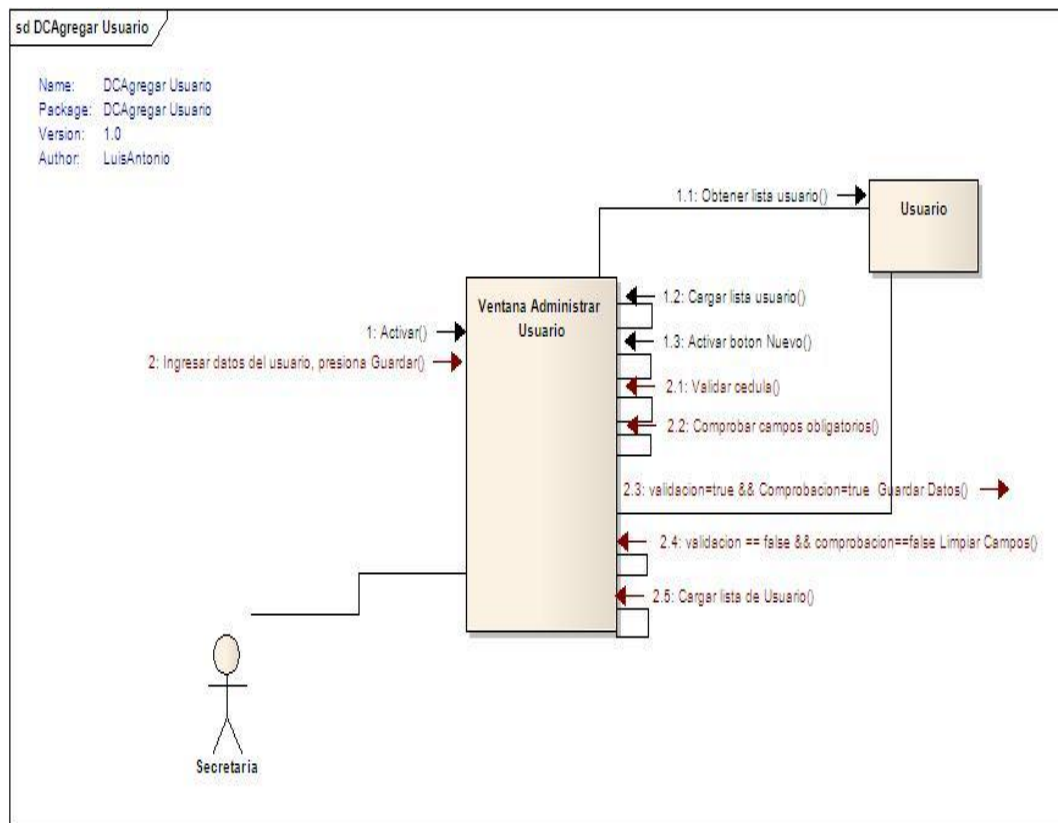


fig. 8.3.2.2.3

### 8.3.2.3. Sección: Modificar Usuario

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea Modificar datos de Usuarios	2. En la ventana Administrar Usuario el sistema presenta una tabla de Usuarios.
3. La secretaria selecciona el usuario a ser modificado.	4. En la ventana presenta un botón modificar
5. La secretaria hace un clic en modificar.	6. El Sistema carga los datos del usuario Seleccionado en los campos en el Panel Agregar.
7. La secretaria modifica los datos de Usuario y hace clic en Guardar	8. Almacenar y actualizará los datos de Usuario
9. Si se desea modificar Clave, Ir a la sección Cambiar clave.	

Tabla. 8.3.2.3

## Cursos alternos

- Ítem 8: El sistema controla que todos los campos obligatorios no estén vacíos de estarlo al momento de almacenar presentará un mensaje de error.

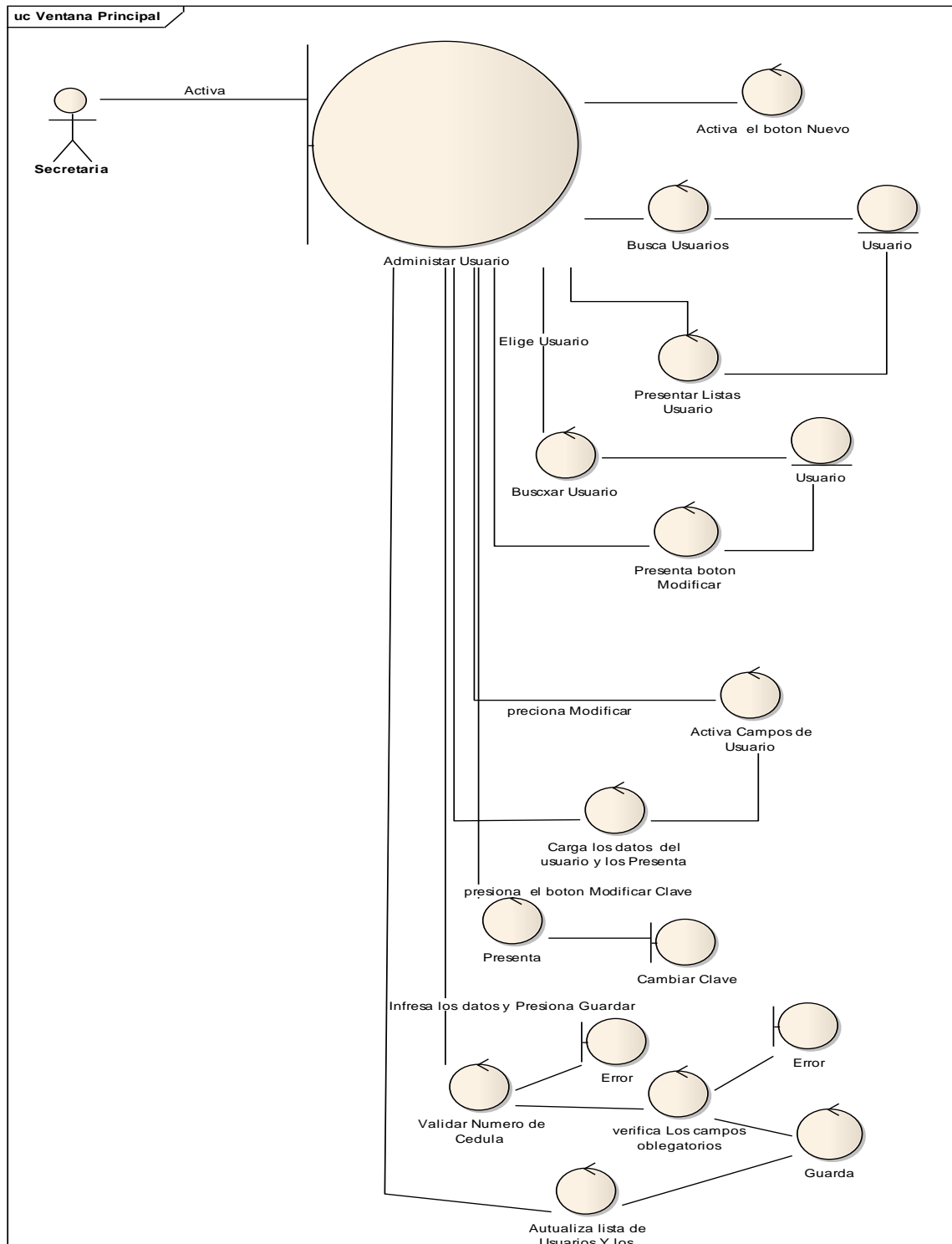
**8.3.2.3.1. Diagrama de Robustez**

fig. 8.3.2.3.1

### 8.3.2.3.2. Diagrama de Secuencia

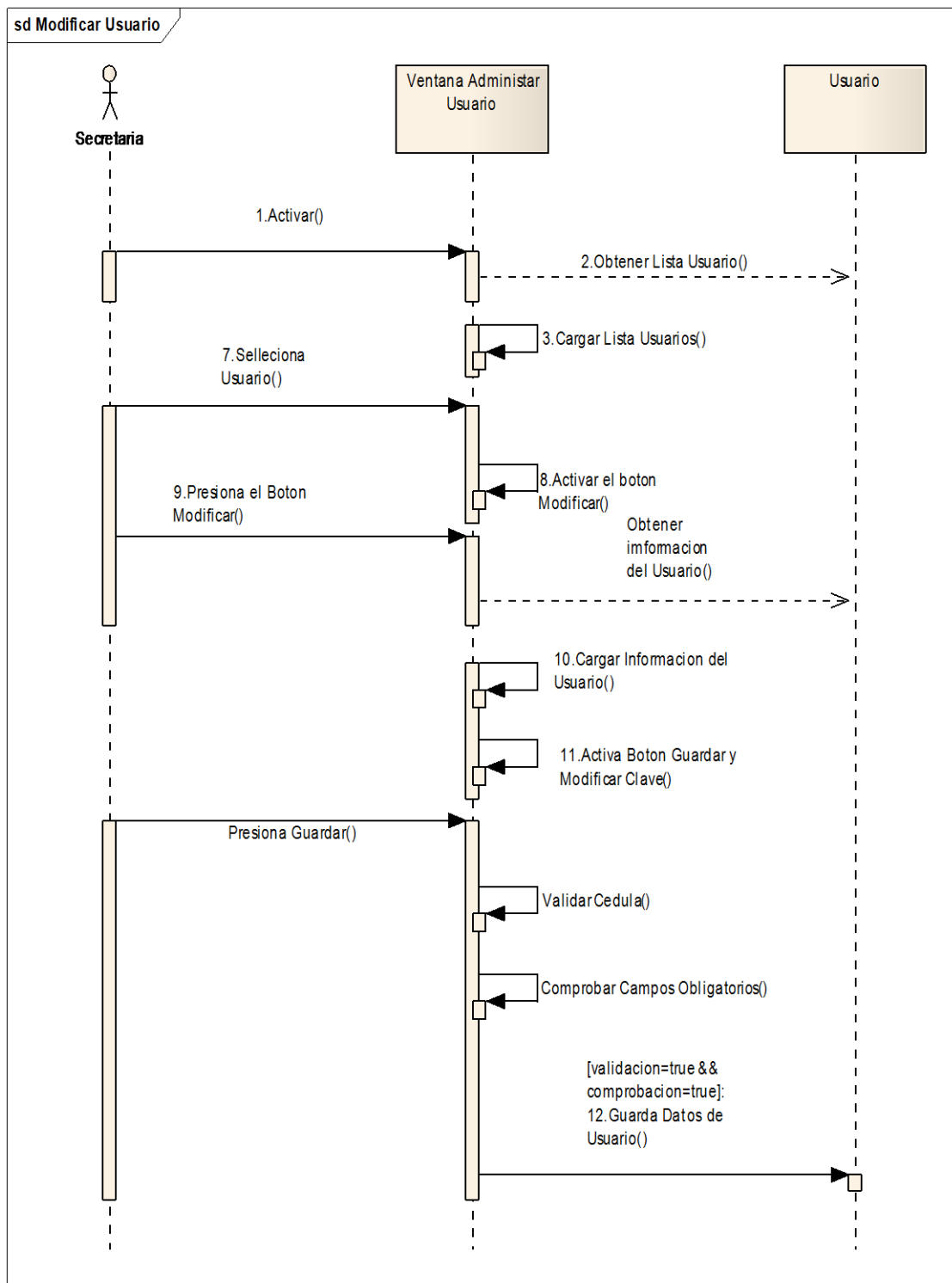


fig. 8.3.2.3.2

### 8.3.2.3. Diagrama de Colaboración

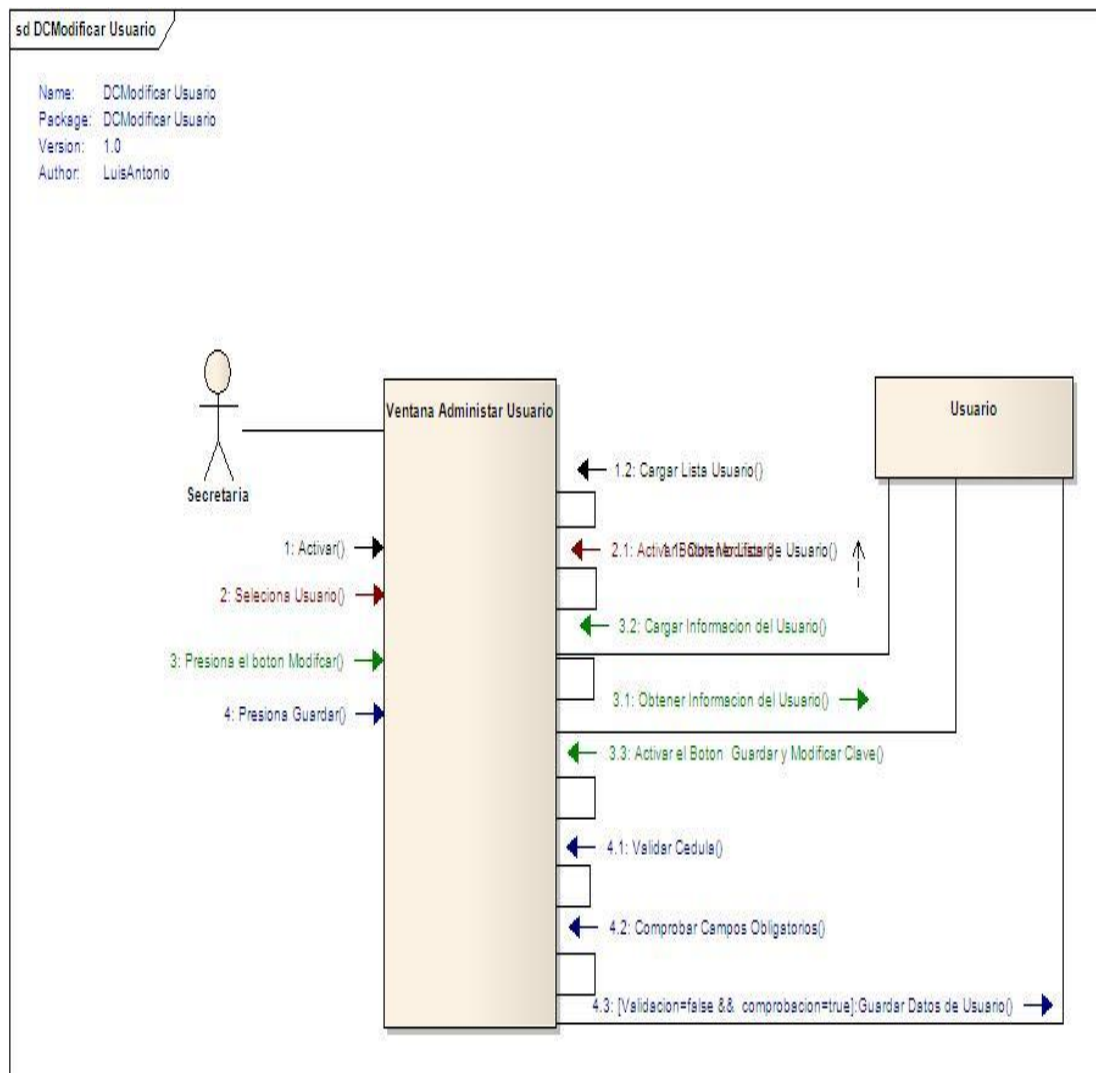


fig. 8.3.2.2.3

### 8.3.2.4. Sección: Asignar Año

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria selecciona un Usuario Docente	2. El Sistema carga los años de educación básica como sus paralelos.
3. Si el docente es de aula	4. El sistema activa el componente JComboBox de años de educación básica, así como el JComboBox del paralelo.
5. La secretaria dependiendo del tipo	6. En la ventana presenta un botón

de docente selecciona el año de educación básica, como el número de paralelos	modificar
7. Luego se hace clic en APLICAR	8. El Sistema guarda los datos.

Tabla. 8.3.2.4

8.3.2.4.1. Diagrama de Robustez

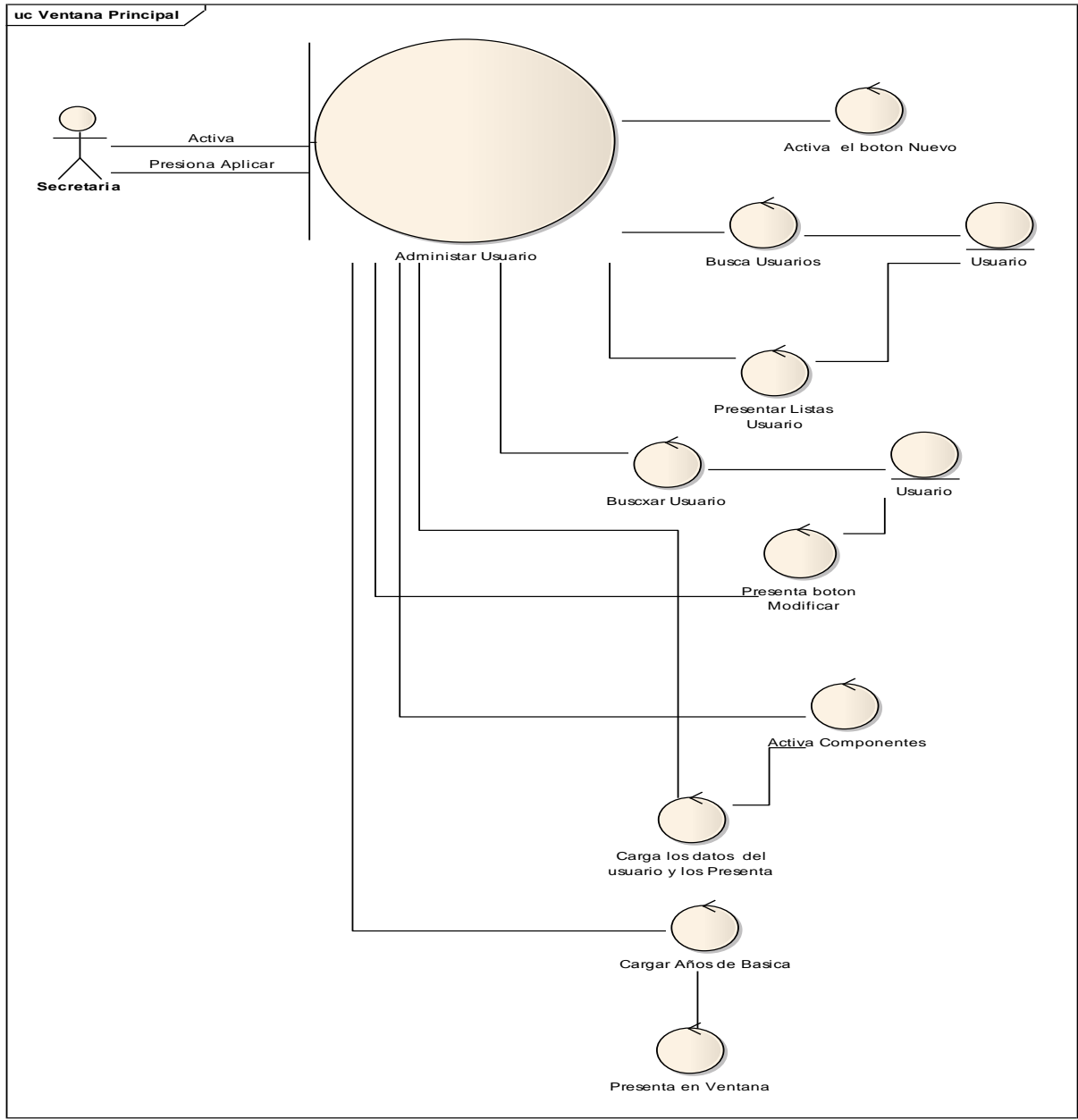


fig. 8.3.2.4.1



### 8.3.3.2.1. Diagrama de Secuencia:

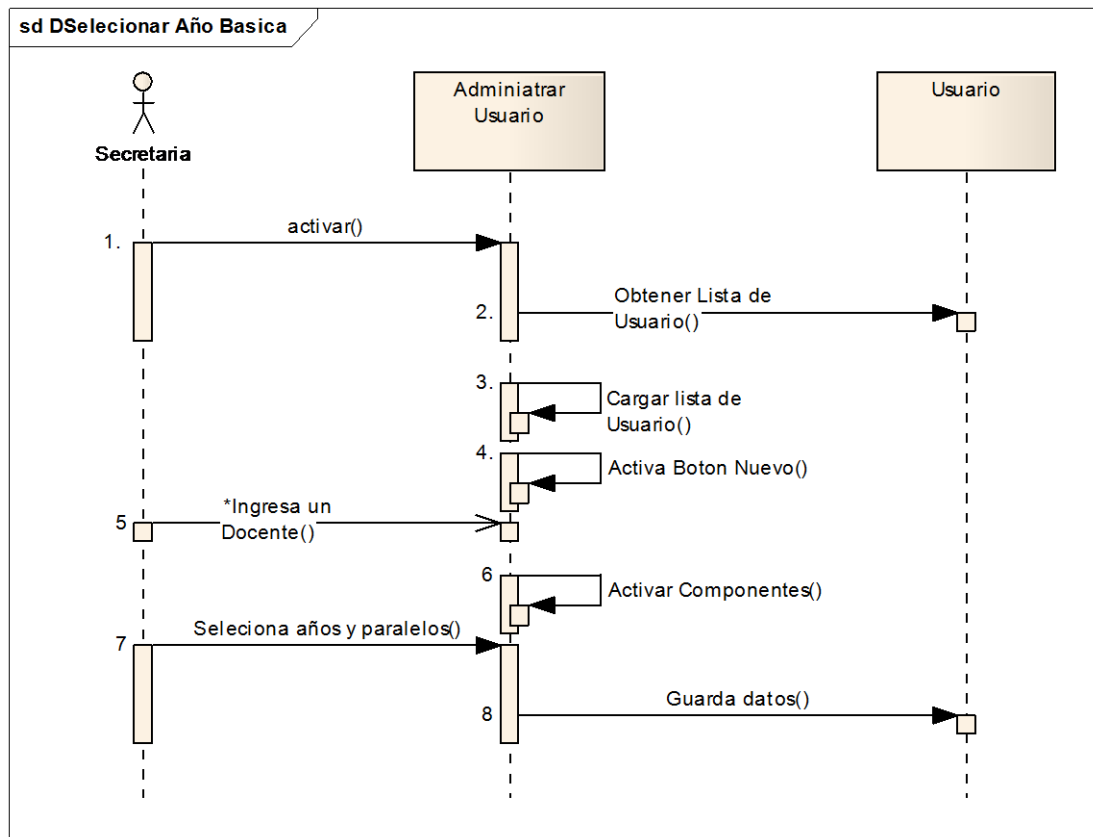


fig. 8.3.3.2.1

### 8.3.3.2.1. Diagrama de Colaboración:

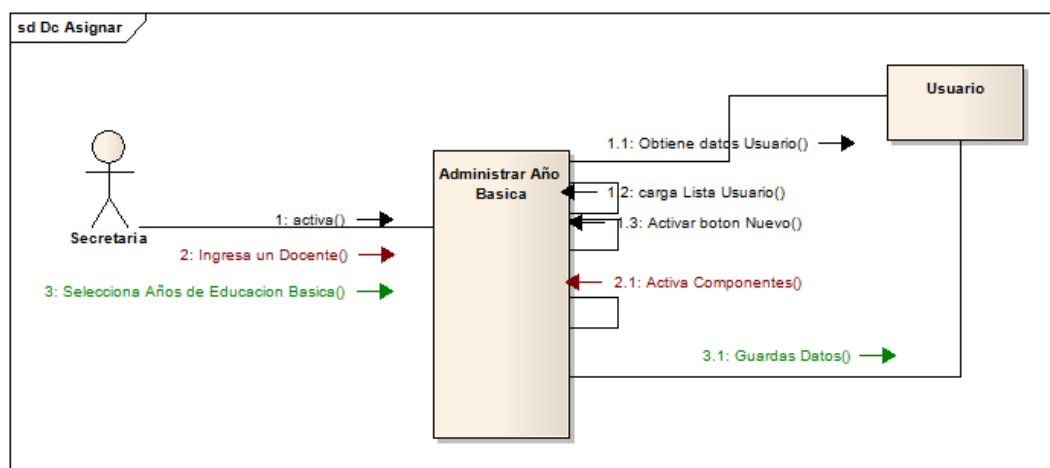


fig. 8.3.3.2.1

### 8.3.3. Caso de Uso: Administrar Periodo

fig. 8.3.3

<b>Caso de uso 003</b>	Administrar Periodo
<b>Actores</b>	Directora (iniciadora), Secretaria
<b>Propósito</b>	Crear un año lectivo, iniciarlo y posteriormente terminarlo.
<b>Resumen</b>	La Directora ingresa al sistema a la ventana administrar Periodo, ingresa la fecha de inicio y del final del periodo académico.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R 1.10

Tabla. 8.3.3

#### 8.3.3.1. CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Comienza cuando la directora inaugura un periodo lectivo.	
2. La Directora va al menú	3. El sistema presenta la ventana

administrar y selecciona el ítem Administrar Periodo.	Administrar Periodo.
4. Sección Crear Periodo	
5. Y al culminar el periodo la Directora desactiva el periodo selecciona el link desactivar en la columna de estado.	6. El sistema desactiva el periodo académico.
7. Si la directora desea modificar el año lectivo va a la sección modificar periodo.	

**Tabla. 8.3.3.1****8.3.3.2. Sección: Crear Periodo**

Acción del actor	Respuesta del Sistema
1. Empieza cuando la Directora ingresa a la ventana Administrar Periodo	2. En la ventana administrar periodo se cargan los años, el estado, el inicio y final de cada periodo académico recuperados en la base de datos y se ubican en la tabla de administrar periodos.
3. La directora procede a ingresar la fecha de inicio y de culminación del periodo, y hace un clic en aplicar.	4. El Sistema actualiza la tabla de administrar periodo académico.
5. La directora da clic en el botón guardar	6. El sistema guarda los datos creados.

**Tabla. 8.3.3.2****Cursos alternos**

- Item 3: Si no consta alguna fecha el Sistema presenta un mensaje de error ya que no puede almacenar el año que se desee crear

### 8.3.3.2.1. Diagrama de Robustez:

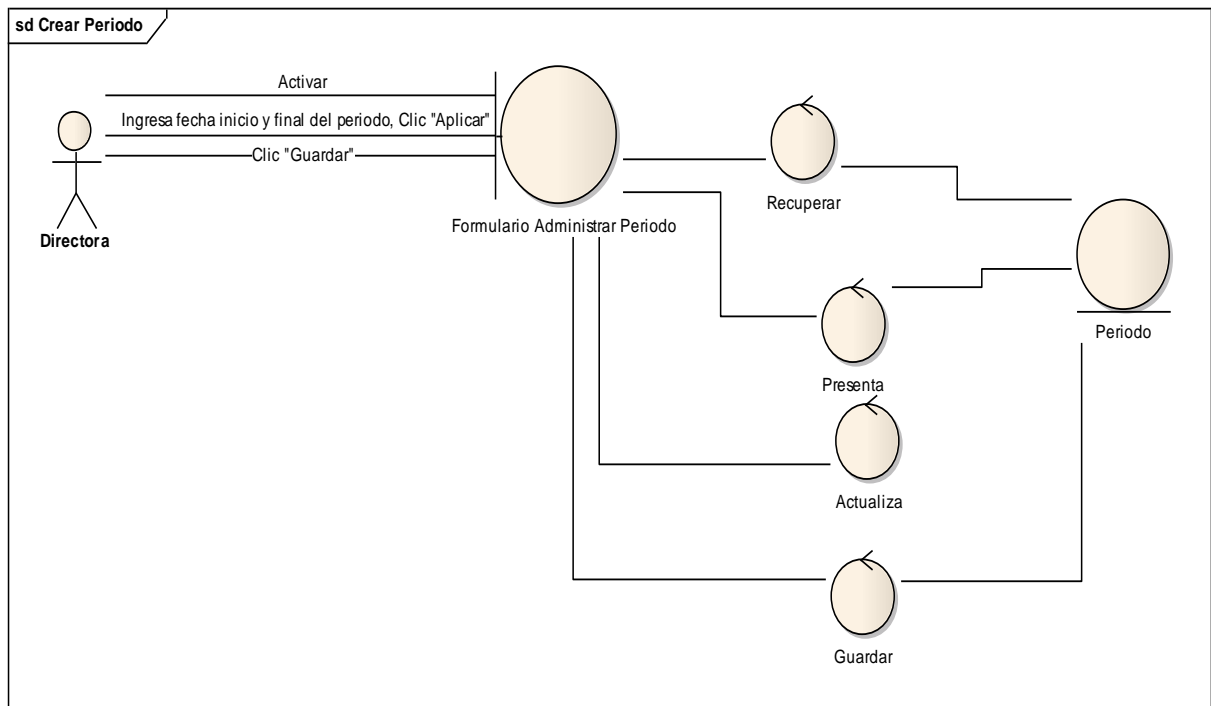


fig. 8.3.3.2.1

### 8.3.3.2.2. Diagrama de Secuencia:

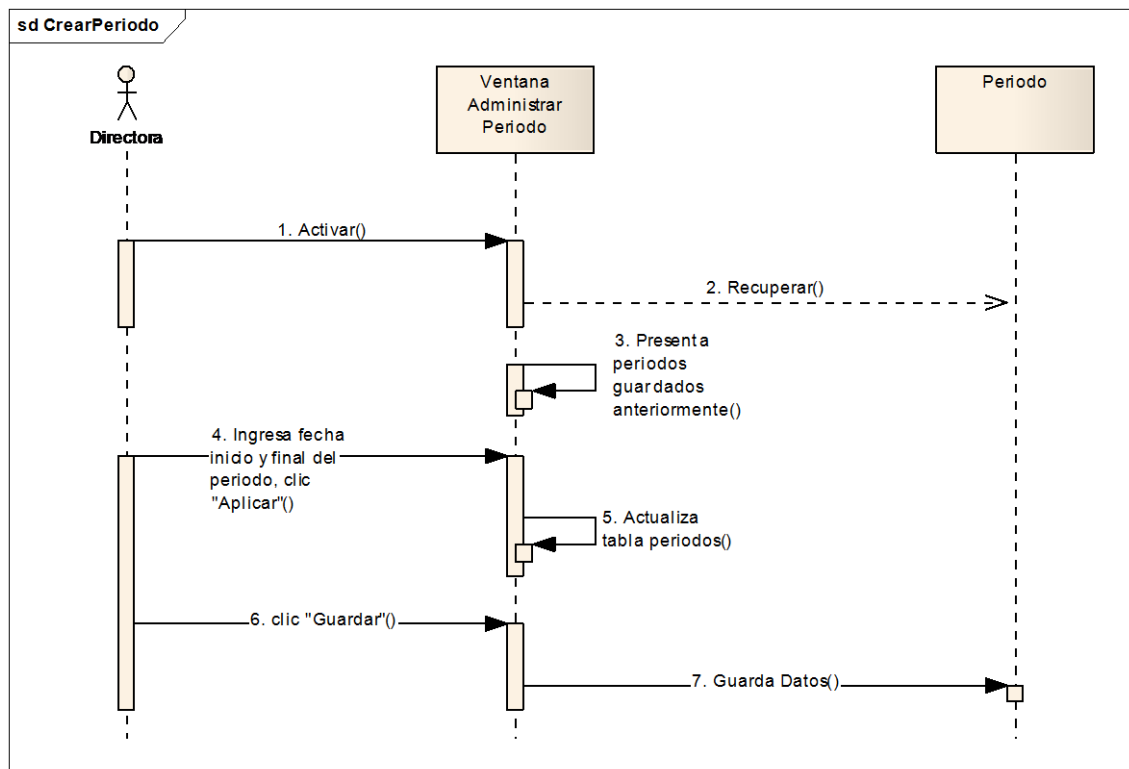


fig. 8.3.3.2.2

### 8.3.3.2.3. Diagrama de Colaboracion:

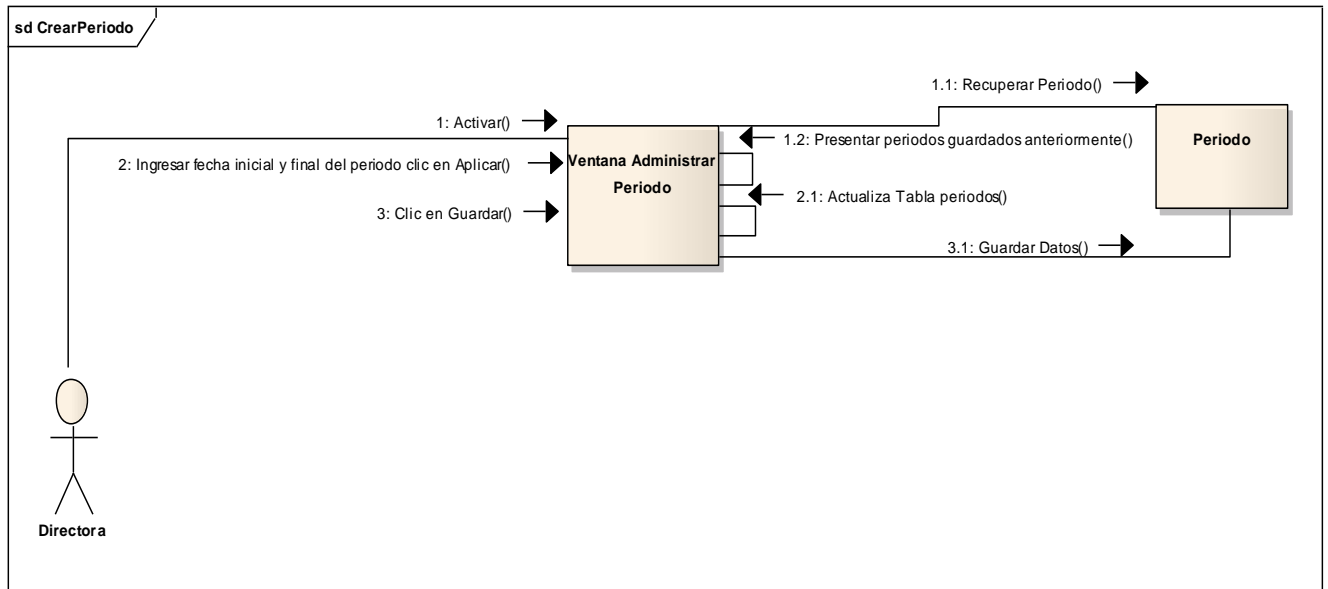


fig. 8.3.3.2.3

### 8.3.3.3. Sección: Modificar Periodo

Acción del actor	Respuesta del Sistema
1. Empieza cuando la Directora ingresa a la ventana Administrar Periodo	2. En la ventana administrar periodo se cargan los años, el estado, el inicio y final de cada periodo académico recuperados en la base de datos y se ubican en la tabla de administrar periodos.
3. La directora selecciona un año lectivo y procede a dar clic en el botón modificar.	4. El Sistema ubica los datos en las casillas correspondientes al inicio, al fin del periodo respectivamente.
5. La directora procede a modificar los datos y el estado del periodo, da clic en Aplicar	6. El sistema actualiza la tabla de administrar periodo.
7. La directora da clic en guardar	8. El sistema guarda los datos actualizados.

fig. 8.3.3.3

Curso alterno

- Ítem 5: Presenta un mensaje de Error si no se hace clic en Aplicar, ya que no se han modificado los datos

### 8.3.3.3.1 Diagrama de Robustez:

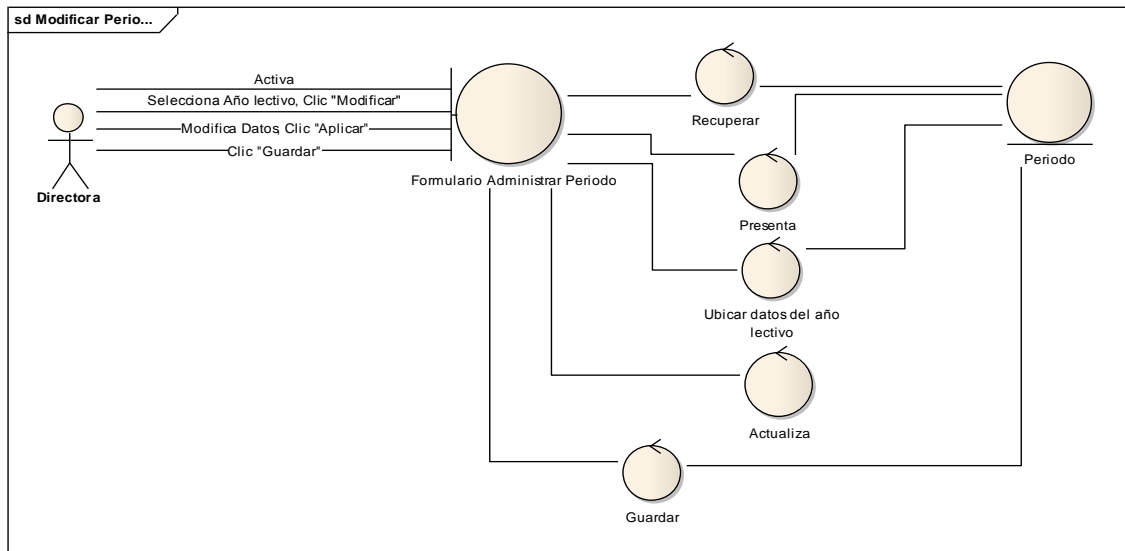


fig. 8.3.3.3.1

### 8.3.3.3.2 Diagrama de Secuencia

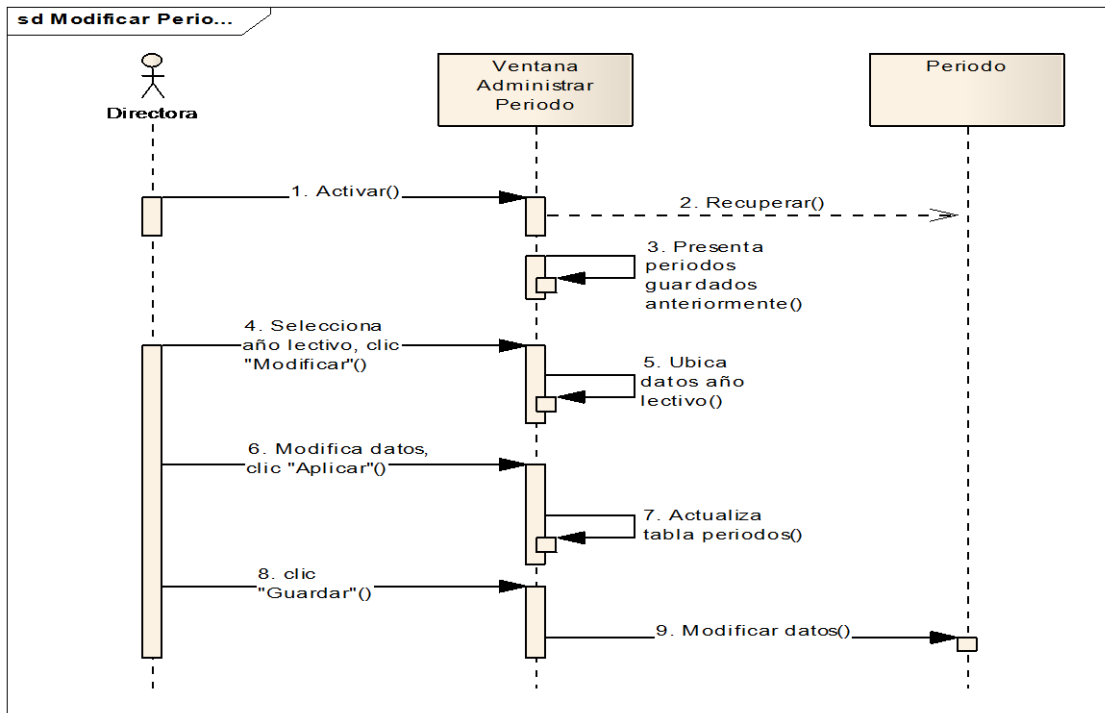


fig. 8.3.3.3.2

### 8.3.3.3 Diagrama de Colaboracion:

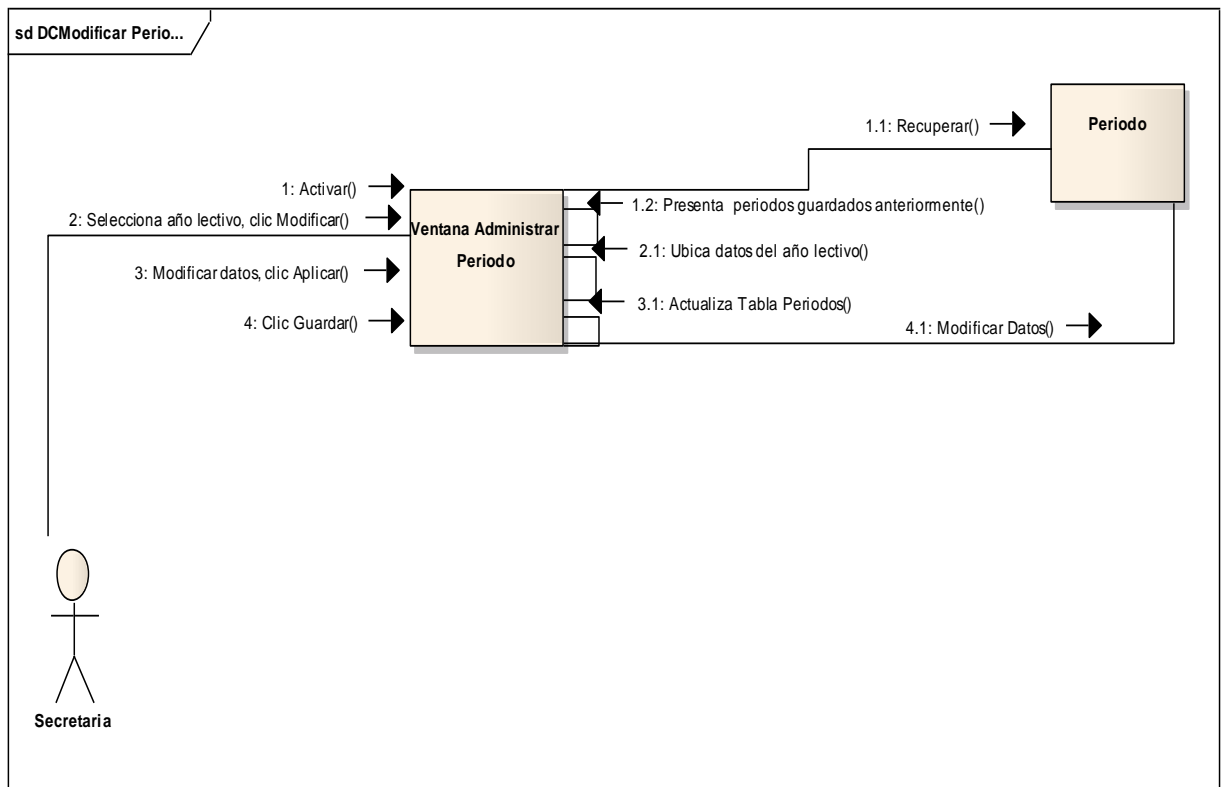


fig. 8.3.3.3.3

### 8.3.4. Caso de Uso: Administrar Año de Educación Básica

<b>Caso de uso 004</b>	Administrar Año de Educación Básica
<b>Actores</b>	Secretaria (iniciadora), Directora
<b>Propósito</b>	Crear los años correspondientes
<b>Resumen</b>	La secretaria ingresa al sistema a la ventana administrar Año de Educación Básica y con la Directora crean los años que estarán disponibles en un año lectivo o periodo.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R 1.11

Tabla. 8.3.4

### 8.3.4.1 CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Comienza cuando la directora autoriza el incremento o disminución de años de educación básica.	2. En la ventana Administrar Año de Educación Básica ubica el periodo académico actual.
3. La secretaria va al menú administrar y selección el ítem administrar Año de Educación Básica.	4. En la ventana Administrar Año de Educación Básica se cargan los años de educación básica y su respectivo estado recuperados en la base de datos y se ubican en la tabla de administrar Años de Básica.
	5. El campo de texto Año de Educación Básica y el campo de su estado encuentra desactivado.
6. Luego si la secretaria desea agregar año ir a la sección agregar año.	
7. Y si la secretaria desea modificar los años creados ir a la sección modificar año.	

**Tabla. 8.3.4.1**

### 8.3.4.2 Sección: Agregar Año

Acción del actor	Respuesta del sistema
7. Empieza cuando la secretaria desea agregar años de educación básica.	8. En la ventana Administrar Años presenta spinner para ingresar el número de años de básica.
9. La secretaria escoge el número de años de básica y hace un clic en aplicar.	10. Ubica los años de educación básica en la tabla Administrar Años de Básica y con el estado como activado.
11. La secretaria da clic en Guardar	12. Guarda los años de educación



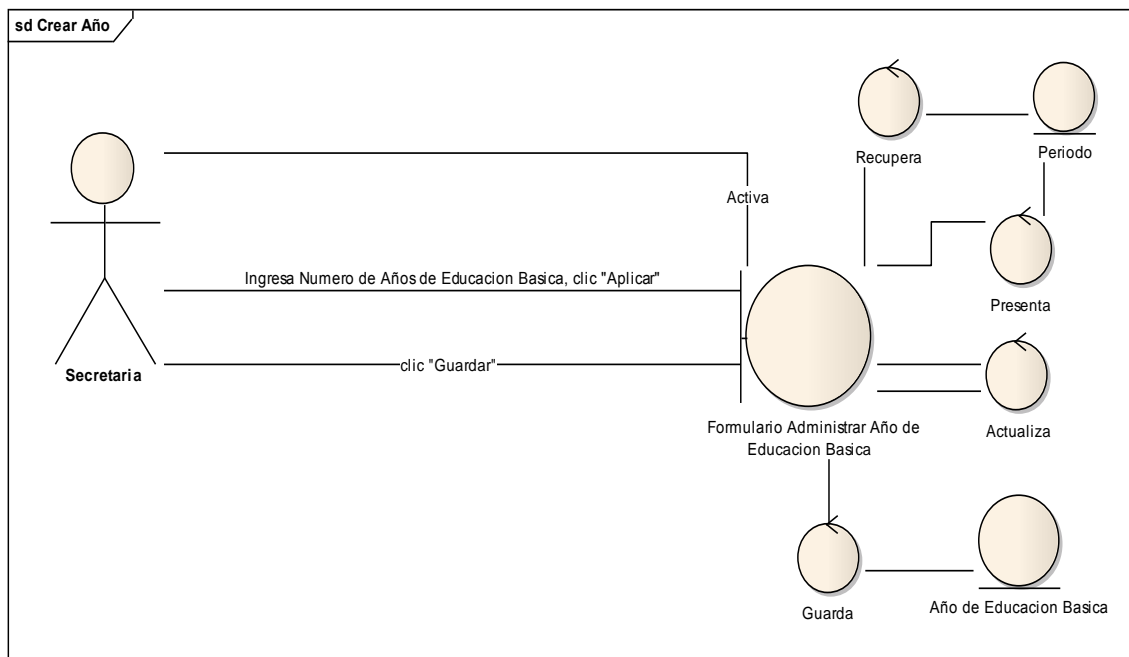
	básica y su respectivo estado.
--	--------------------------------

**Tabla. 8.3.4.2**

Cursos alternos:

- Ítem 5: Si no se ha da clic en Guardar presentará un mensaje de error indicando que no se han guardado los datos.

#### 8.3.4.3.1 Diagrama de Robustez:

**fig. 8.3.4.3.1**

### 8.3.4.3.2 Diagrama de Secuencia:

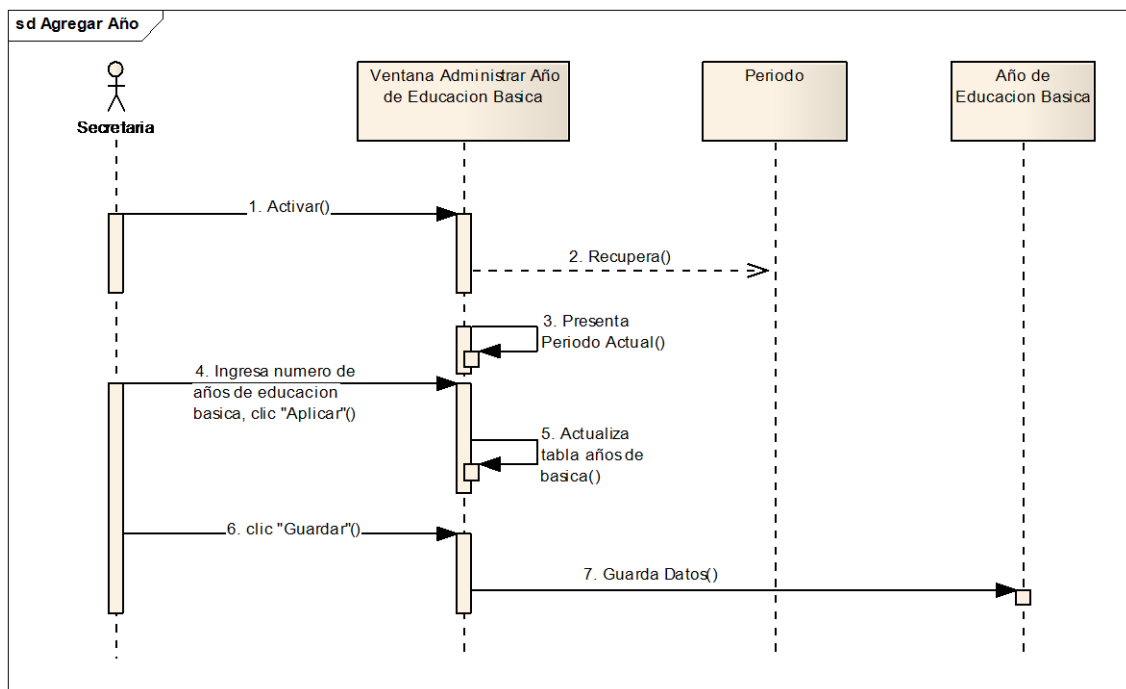


fig. 8.3.4.3.2

### 8.3.4.2. Diagrama de Colaboracion:

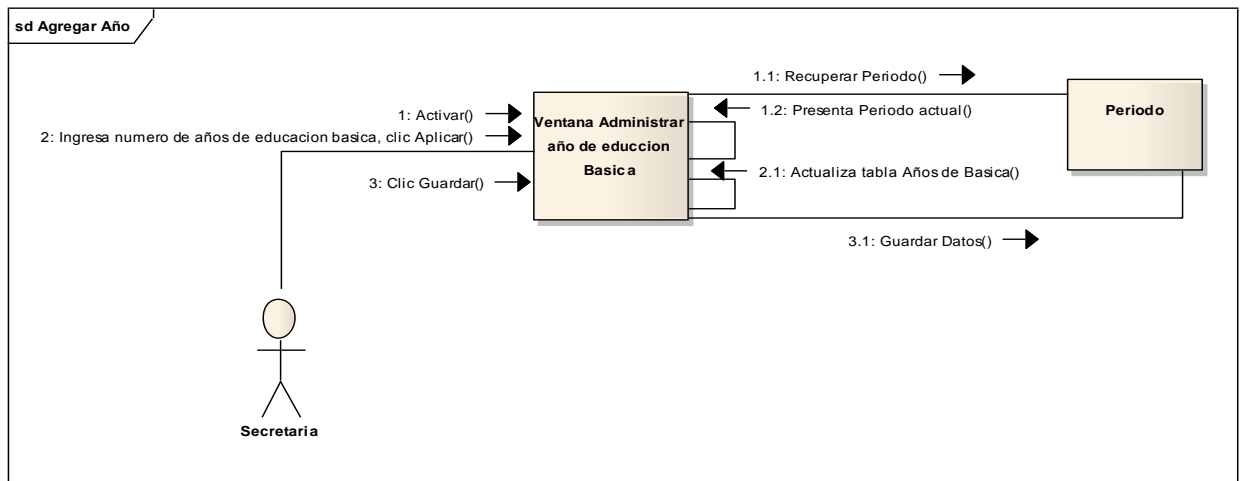


fig. 8.3.4.2

### 8.3.4.3. Sección: Modificar Año

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea modificar un año de educación básica.	2. En la ventana Administrar Año de Educación Básica se cargan los años de educación básica y los

	respectivos estados recuperados en la base de datos y se ubican en la tabla de administrar Años de Básica
	3. El campo de texto Año de Educación Básica y el campo de su estado encuentra desactivado.
4. La secretaria selecciona el año de educación básica que desea modificar, Y hace un clic en modificar.	5. El sistema fija el año de educación básica en el campo de Año de educación básica y se activa el campo de su estado.
6. La secretaria modifica el estado del año de educación básica seleccionado. Y hace clic en aplicar.	7. El sistema actualiza la información, ubicando los datos respectivos en la tabla de los años de educación básica.
8. La secretaria da clic en guardar	9. Guarda los datos modificados de los años de educación básica y su estado.

**Tabla. 8.3.4.3**

Cursos alternos:

- Ítem 8: Si no se ha da clic en Guardar presentará un mensaje de error indicando que no se han guardado los datos modificados.

### 8.3.4.3.1 Diagrama de Robustez:

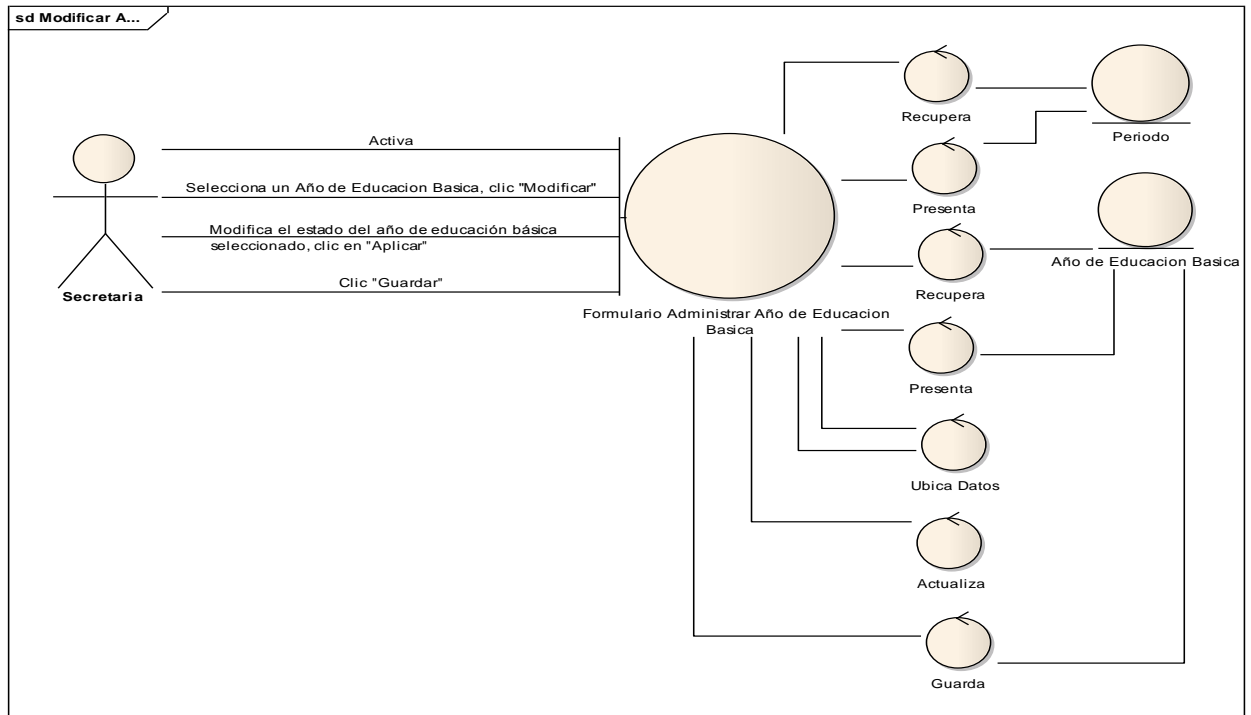


fig. 8.3.4.3.1

### 8.3.4.3.2 Diagrama de Secuencia:

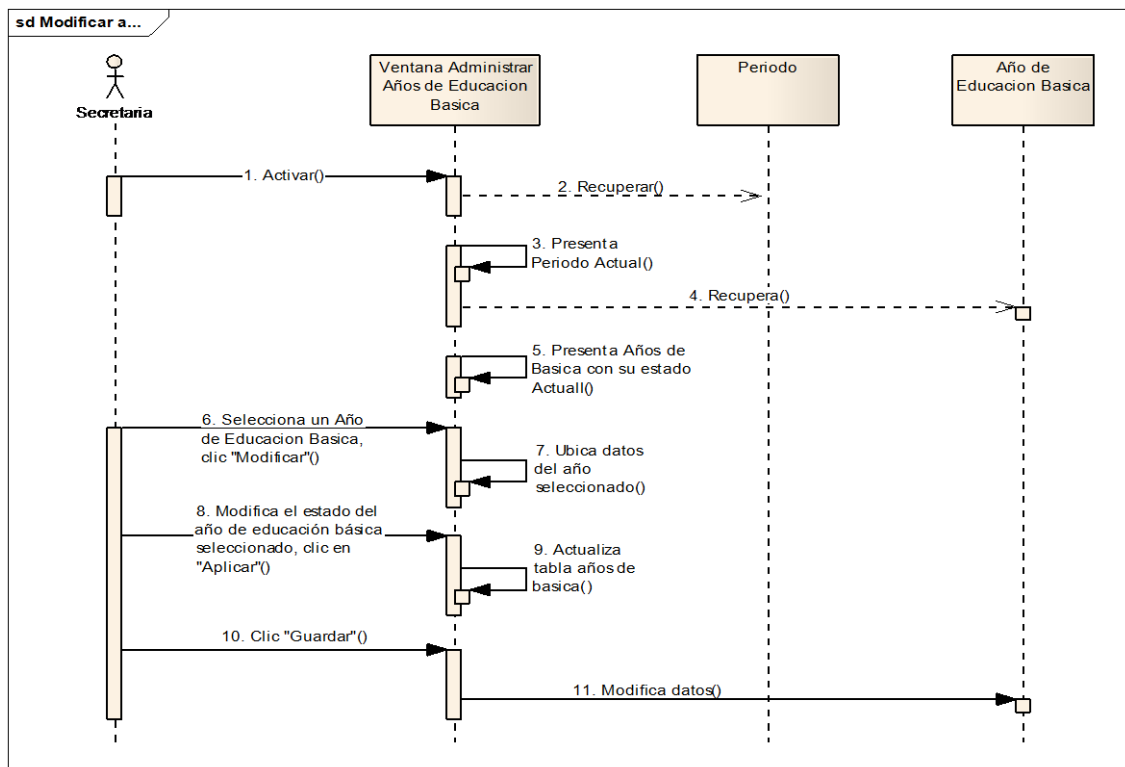


fig. 8.3.4.3.2

### 8.3.4.3.3 Diagrama de Colaboracion:

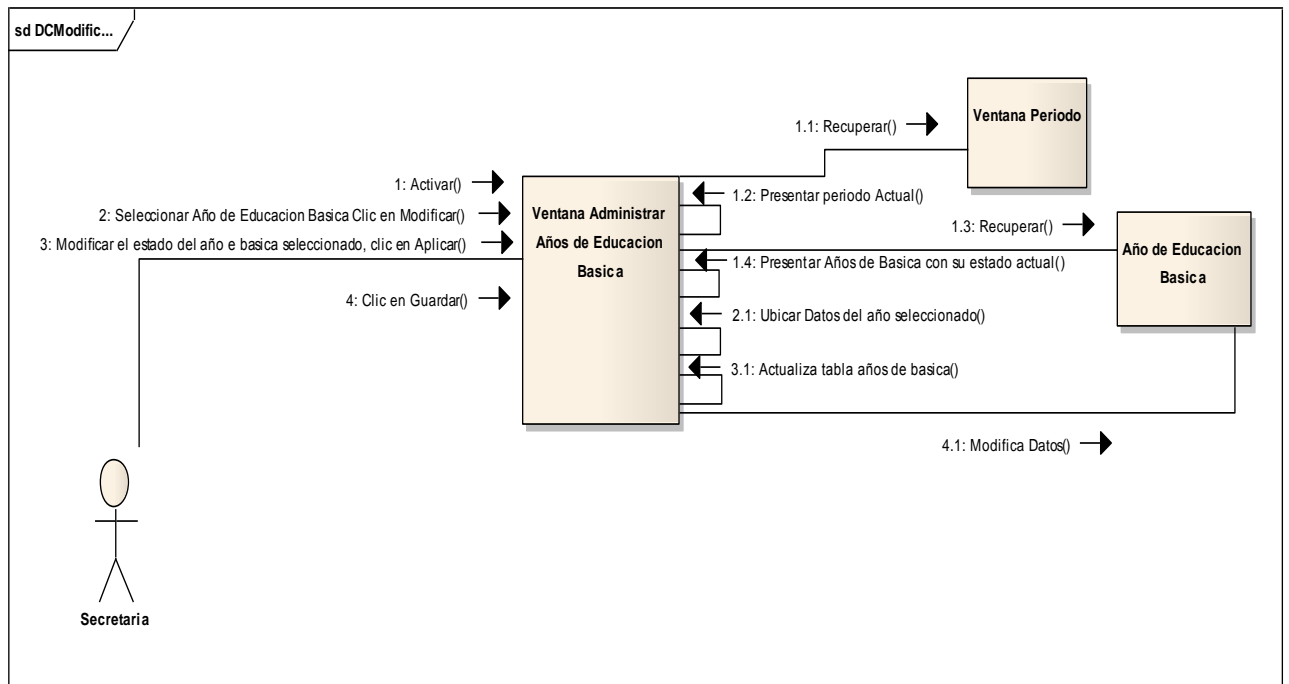


fig. 8.3.4.3.3

### 8.3.5 Caso de Uso: Administrar Paralelos

**Para Ingresar Paralelos Seleccionar Un Año de la lista**

Año de Educacion Basica 1° Año de ... **APLICAR**

Paralelos: 0

**Seleccione un año de basica para modificar estado de sus paralelos**

AÑO DE EDUCACION BASICA

Seleccione un paralelo para modificar su estado A

Estado ☐

**TABLA AÑOS DE BASICA Y PARALELOS**

Año de Educacion Basica	Paralelos
1° Año de Educación Básica	[Paralelos sin asignar]
2° Año de Educación Básica	[Paralelos sin asignar]
3° Año de Educación Básica	[Paralelos sin asignar]
4° Año de Educación Básica	[Paralelos sin asignar]
5° Año de Educación Básica	[Paralelos sin asignar]
6° Año de Educación Básica	[Paralelos sin asignar]
7° Año de Educación Básica	[Paralelos sin asignar]

**GUARDAR** **CANCELAR**

**MODIFICAR**

Synthetica - Unregistered Evaluation Copy!

fig. 8.3.5

<b>Caso de uso 004</b>	Administrar Paralelos
<b>Actores</b>	Secretaria (iniciadora), Directora
<b>Propósito</b>	Crear los Paralelos correspondientes a cada año de básica.
<b>Resumen</b>	La secretaria ingresa al sistema a la ventana administrar paralelos y con la Directora crean los paralelos que estarán disponibles para cada año de básica en un periodo académico.
<b>Tipo</b>	Primario y esencial
<b>Referencias</b>	R 1.11

<b>cruzadas</b>	
-----------------	--

**Tabla. 8.3.5****8.3.5.1 CURSO NORMAL DE EVENTOS**

Acción del actor	Respuesta del sistema
1. Comienza cuando la directora autoriza el incremento o disminución de paralelos de cada año de educación básica.	
2. La secretaria va al menú administrar y selección el ítem administrar Paralelos.	3. En la ventana Administrar Año de Educación Básica se cargan los años de educación básica y los respectivos paralelos recuperados en la base de datos y se ubican en la tabla de Años de básica y paralelos.
	4. El campo año de educación básica, la lista de paralelos y la casilla de estado de paralelos se encuentra desactivada.
5. Luego si la secretaria desea agregar paralelos respectivos a cada año. Ir a la sección ingresar paralelo.	
6. Si la secretaria desea modificar paralelos respectivos a cada año. Ir a la sección modificar paralelo.	

**Tabla. 8.3.5.1****8.3.5.2 Sección: Ingresar Paralelo**

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea ingresar paralelos según el año de	2. En la ventana Administrar Años se carga la lista de años de educación

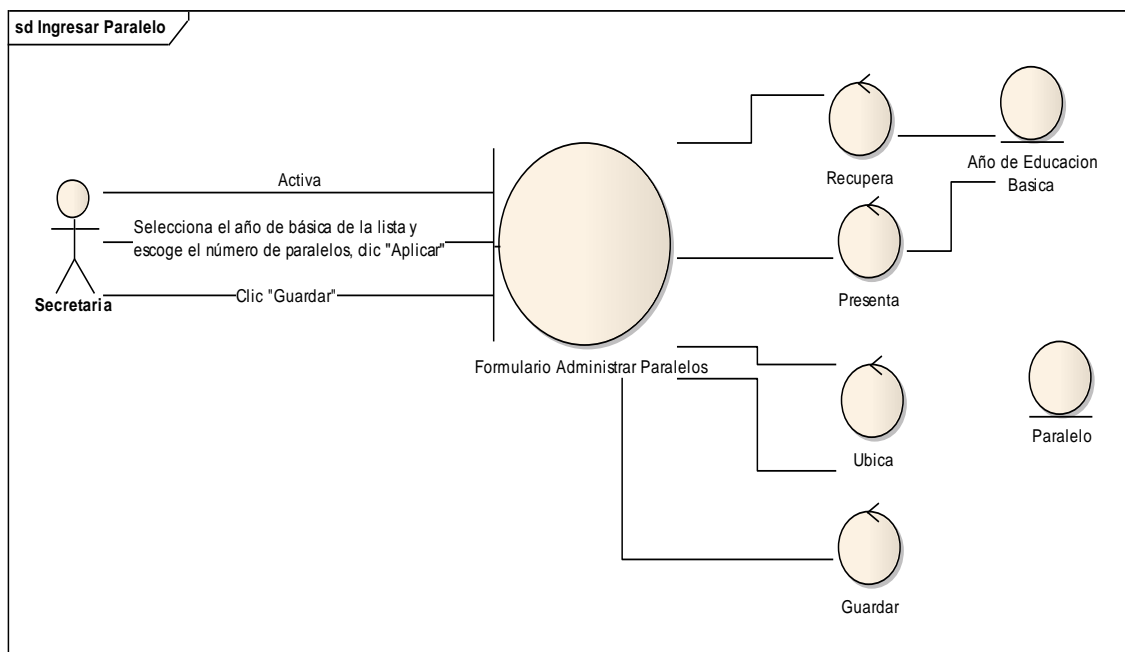
educación básica.	básica al igual el que la tabla años de básica y paralelos se cargan todos los años de básica activos.
	3. El campo año de educación básica, la lista de paralelos y la casilla de estado de paralelos se encuentra desactivada.
4. La secretaria selecciona el año de básica de la lista y escoge el número de paralelos y hace un clic en aplicar.	5. Ubica los paralelos respectivos al año de básica seleccionado en la tabla Años de Básica y paralelos.
6. La secretaria da clic en Guardar.	7. El sistema guarda los datos.

**Tabla. 8.3.5.2**

Cursos alternos:

- Ítem 6: Si no se ha ingresado los paralelos respectivos por cada año de educación básica el sistema no almacenará los datos y presentará un mensaje de error.

#### 8.3.5.2.1 Diagrama de Robustez:



**fig. 8.3.5.2.1**



### 8.3.5.2.2 Diagrama de Secuencia:

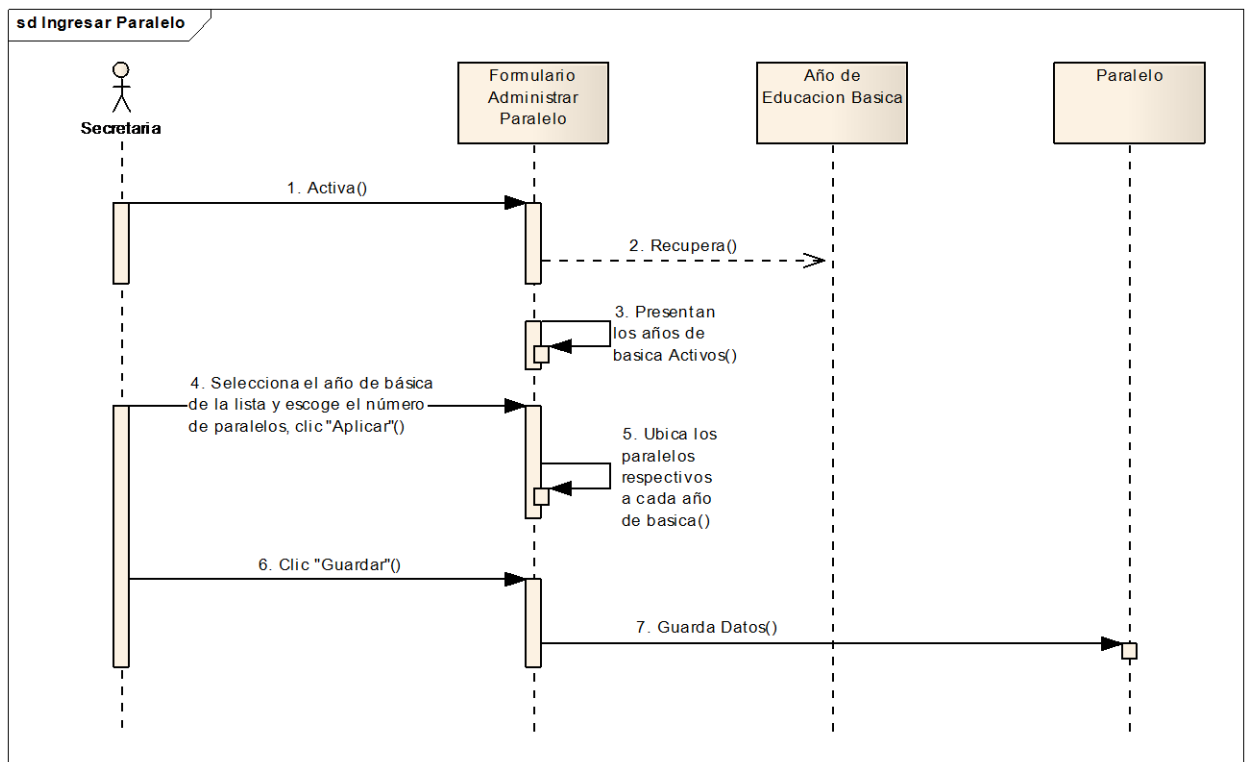


fig. 8.3.5.2.2

### 8.3.5.2.3 Diagrama de Colaboracion:

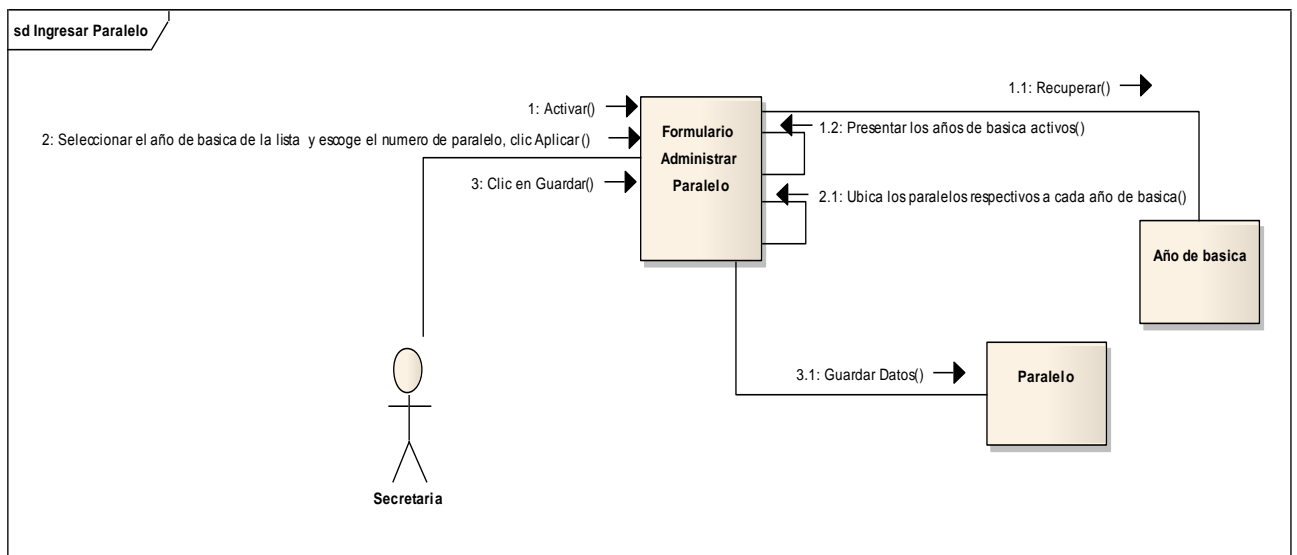


fig. 8.3.5.2.3

### 8.3.5.3 Sección: Modificar Paralelo

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea modificar el estado de los paralelos de cada año de educación básica.	2. En la ventana Administrar Años se carga la lista de años de educación básica al igual que la tabla años de básica y paralelos se cargan todos los años de básica activos.
3. La secretaria selecciona un año de básica de la tabla y da clic en modificar	4. Se ubica el año de básica seleccionado en el campo de año de educación básica y se cargan todos los paralelos en la lista de paralelos.
5. Selecciona un paralelo de la lista de paralelos y modifica su estado, da clic en actualizar	6. Se actualiza la tabla años de básica y paralelos con todos los años de educación básica activos, al igual q los paralelos activos de cada año de básica.
7. La secretaria da clic en guardar.	8. Guarda los datos modificados.

**fig. 8.3.5.3**

Cursos alternos:

- Ítem 7: Si no se ha da clic en Guardar presentará un mensaje de error indicando que no se han guardado los datos modificados.

### 8.3.5.3.1 Diagrama de Robustez:

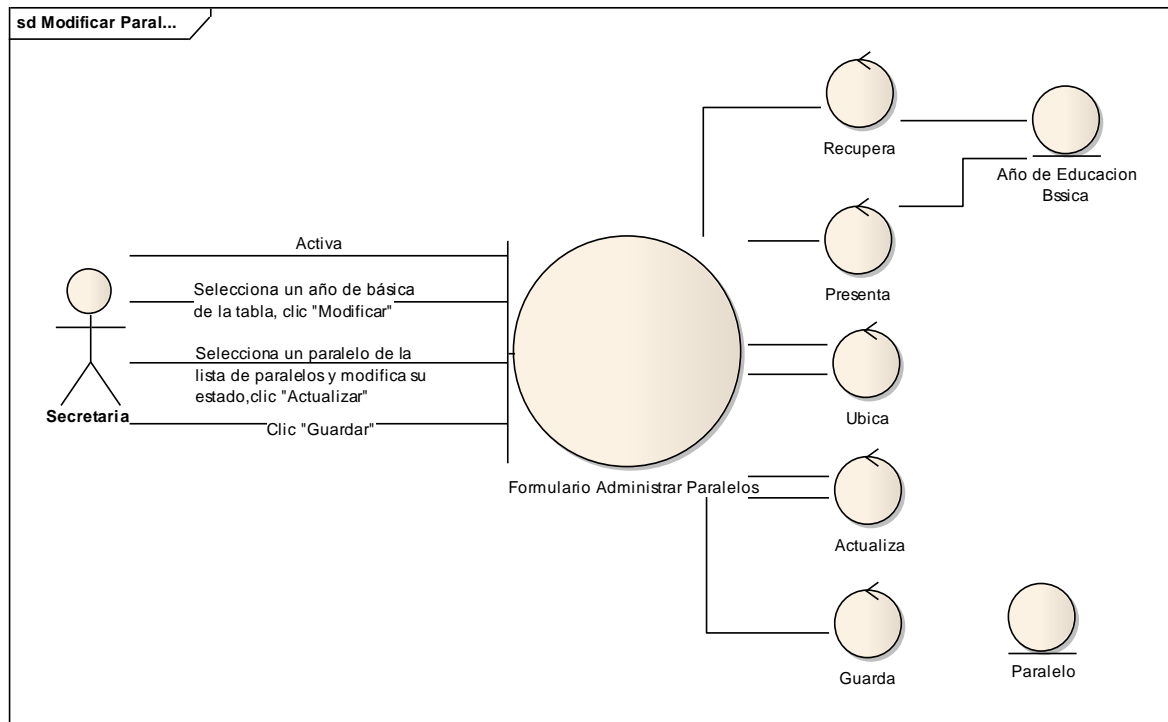


fig. 8.3.5.3.1

### 8.3.5.3.2 Diagrama de Secuencia:

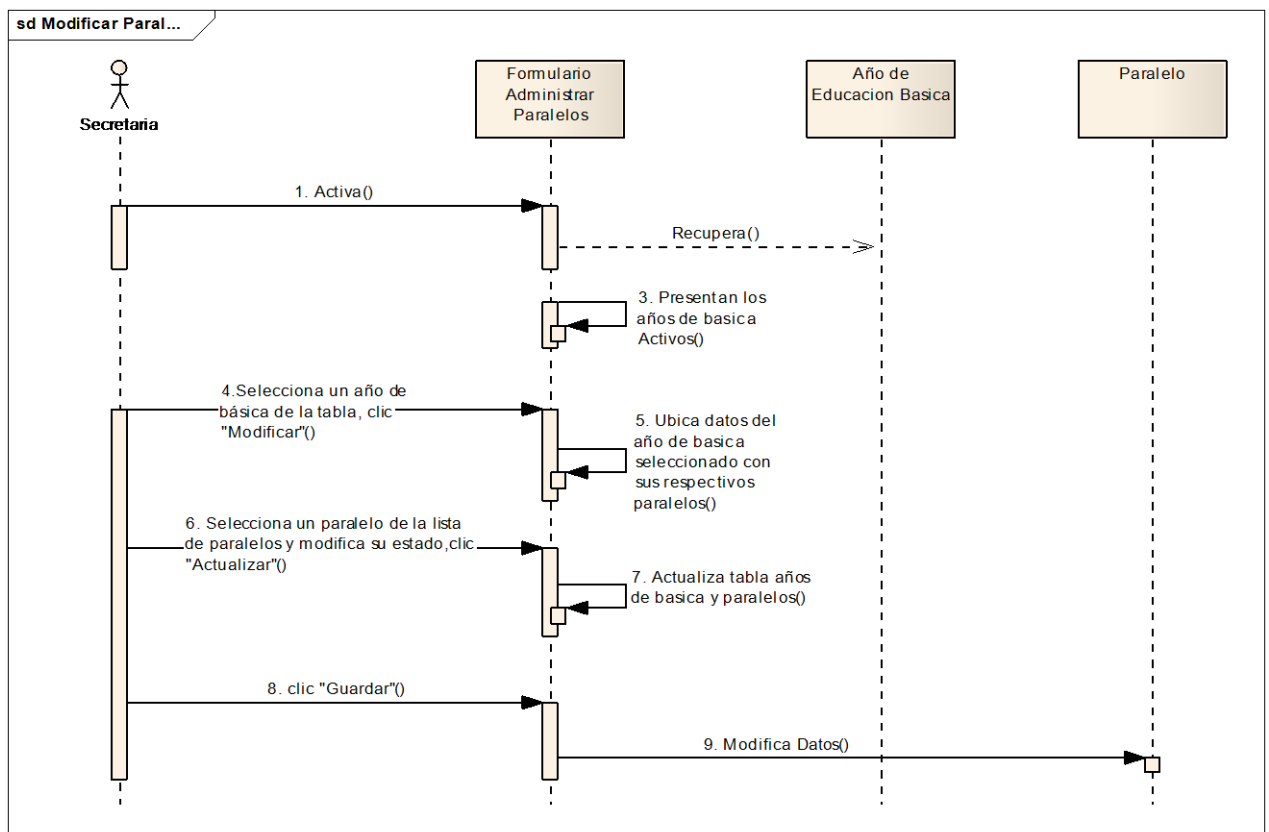


fig. 8.3.5.3.3

### 8.3.5.3.3 Diagrama de Colaboracion:

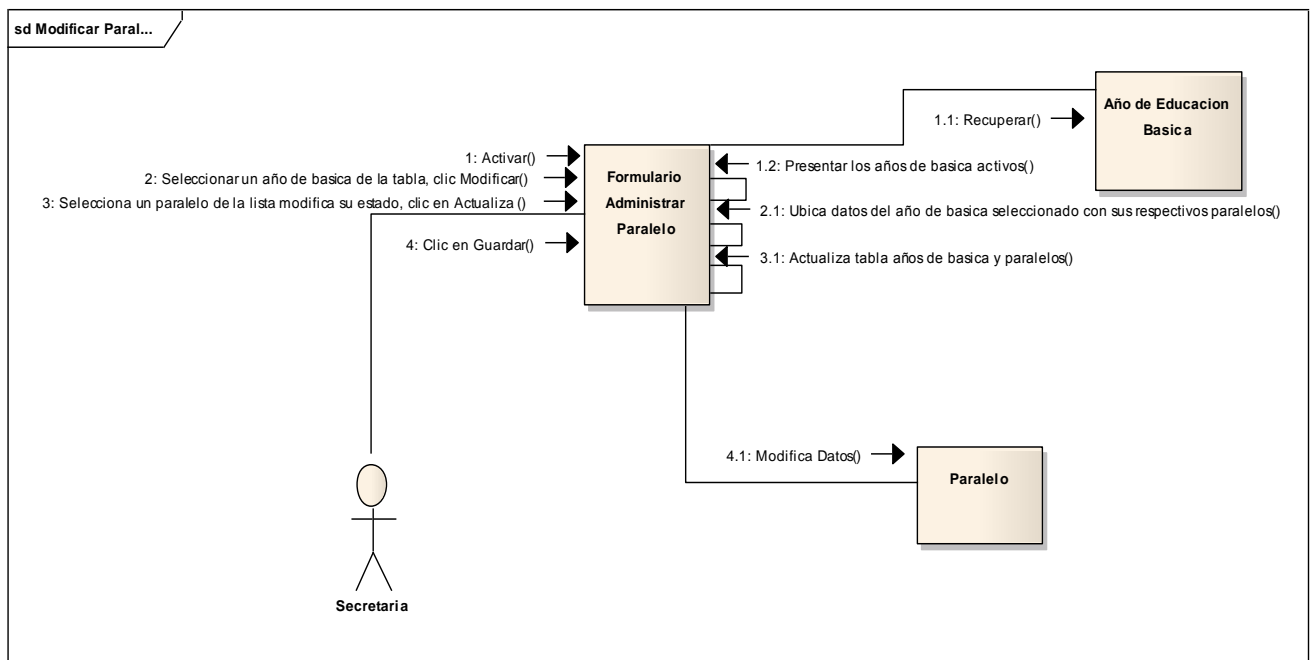


fig. 8.3.5.3.3

### 8.3.6 Caso de Uso: Administrar Materia

**Diseño previo [Materias]**

**Agregar Materia**

Haga clic en nuevo si desea agregar materia

Seleccione el Año de Basica y el año a Asignar

Año:

Escriba el nombre la materia que desea Agregar

Nombre:

☐ Seleccione si la Materia es Especial

En la columna Tipo de Matera Agregue las materias que son especiales:

Materia	Tipo de Materia
Lenguaje y Comunicacion	<input type="checkbox"/>
Matematicas	<input type="checkbox"/>
Ciencias Naturales	<input type="checkbox"/>
Estudios Sociales	<input checked="" type="checkbox"/>
Cultura Fisica	<input type="checkbox"/>
Cultura Estetica	<input type="checkbox"/>
Lenguaje Extran (Ingles)	<input type="checkbox"/>
Optativa	<input type="checkbox"/>
Formacion Cristiana	<input type="checkbox"/>
Entono Natural y Social	<input type="checkbox"/>
Musica	<input type="checkbox"/>
Computacion	<input type="checkbox"/>

Si desea Eliminar o Modificar Seleccione la Materia:

fig. 8.3.6

<b>Caso de uso 005</b>	Administrar Materia
<b>Actores</b>	Secretaria (iniciadora), Directora
<b>Propósito</b>	Crear las materias para cada año de básica
<b>Resumen</b>	La secretaria ingresa al sistema a la ventana administrar Materia y con la Directora crean las materias correspondientes a cada Año de Básica.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R 1.12

Tabla. 8.3.5

### 8.3.6.1 CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Comienza cuando la directora autoriza y especifica las materias que se lleven a cabo por año de educación básica.	
2. La secretaria va al menú administrar y al ítem administrar Materia.	3. El sistema presenta la ventana Administrar Materia
4. Luego la secretaria seleccionará las materias correspondientes a cada año de básica de la lista establecida.	5. El sistema presentará las materias seleccionadas con un visto.
6. Si la secretaria desea agregar una materia que no esté en la lista deberá escoger la opción <b>NUEVO</b> , seleccionará el año de básica a asignar dicha materia e ingresará el nombre de la nueva materia, luego seleccionará si es materia especial o no, y finalmente pulsará el botón guardar para que se almacene la información ingresada.	7. El sistema permitirá visualizar la nueva materia ingresada en la lista.
8. Si desea modificar alguna materia ingresada o las materias seleccionadas de la lista podrá hacerlo en el botón modificar.	9. El sistema permitirá hacer los cambios que desee realizar la secretaria.
10. Si desea eliminar alguna materia deberá seleccionarla de la lista y presionar el botón <b>QUITAR</b> .	11. El sistema mostrará la lista actualizada, es decir no aparecerán las materias que hayan sido eliminadas.

**Tabla. 8.3.6.1**

### 8.3.6.2 Sección: Agregar Materia

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea agregar una Materia.	2. El Sistema presenta en la Ventana Administrar Materia presenta un botón Nuevo.
3. La secretaria presiona el botón nuevo y selecciona el año de Básica.	4. El sistema habilita la caja de texto en donde se agrega el nombre la nueva Materia.
5. Luego la secretaria digita la nueva materia. Y presiona el botón <b>AGREGAR</b> .	6. Guarda los datos ingresados y agrega la materia ingresada en la lista.
7. Finalmente presiona <b>GUARDAR</b> para que los datos queden registrados en el sistema de forma permanente.	8. El sistema guarda la información ingresada.

fig. 8.3.6.2

#### Cursos alternos

- Item 5: Si no se ha ingresado el nombre de la materia el sistema no agregará la materia y presentará un mensaje de error.

#### 8.3.6.2.1 DIAGRAMA DE ROBUSTES

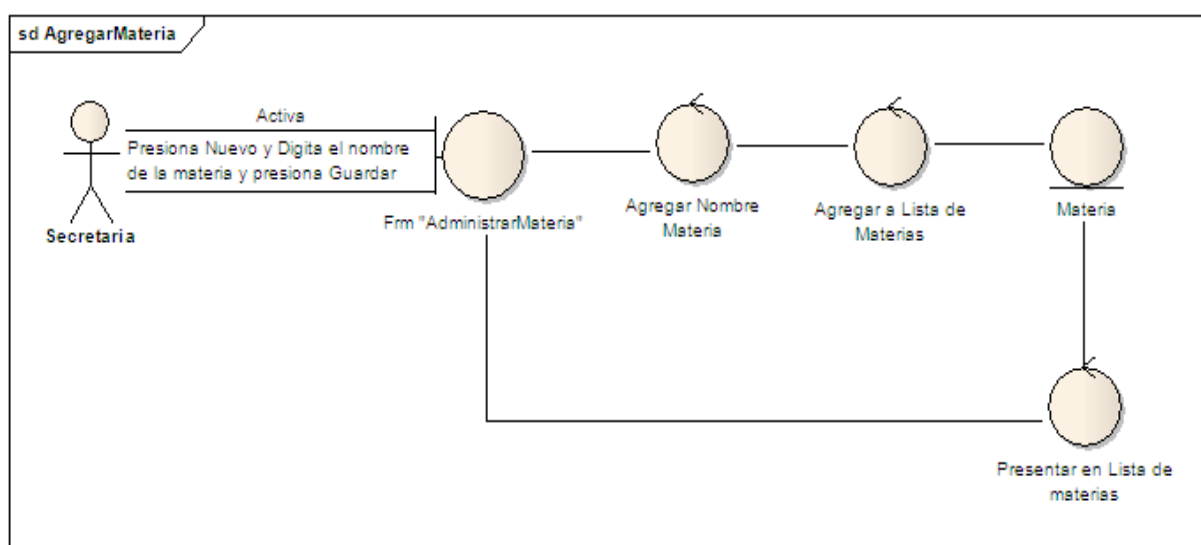


fig. 8.3.6.2.1

### 8.3.6.2.2 DIAGRAMA DE SECUENCIA

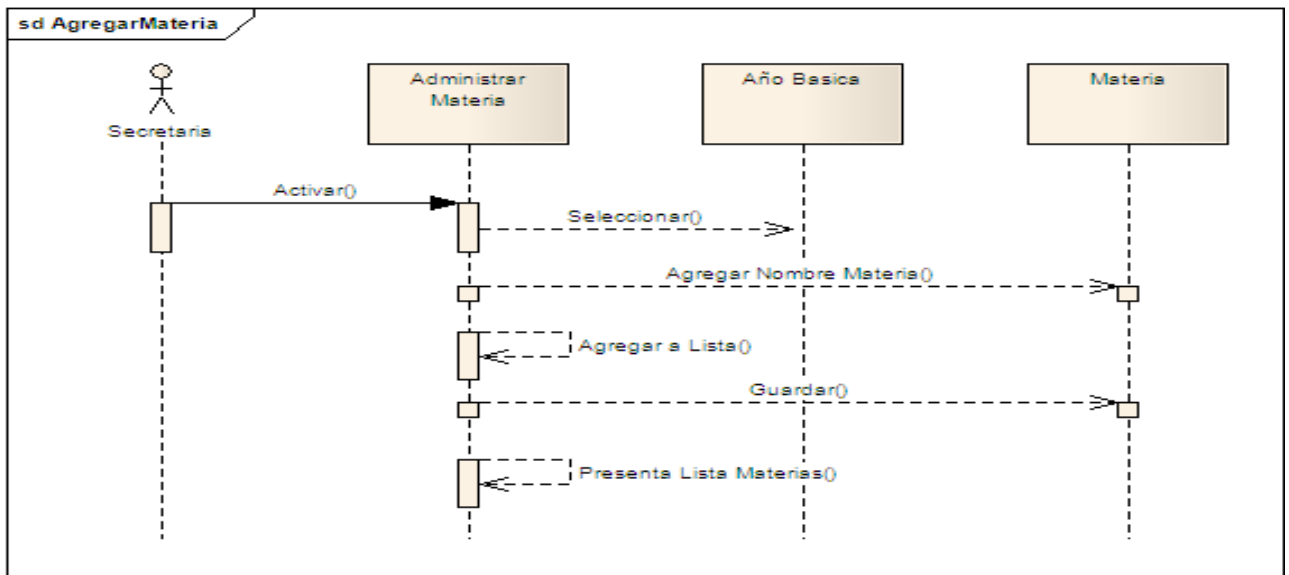


fig. 8.3.6.2.2

### 8.3.6.2.3 DIAGRAMA DE COLABORACIÓN

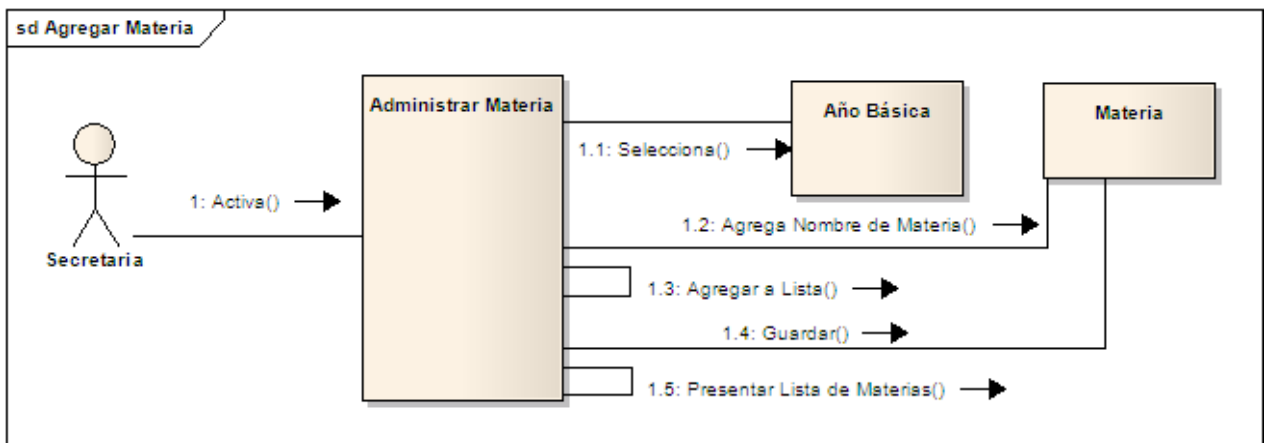


fig. 8.3.6.2.3

### 8.3.6.3 Sección: Modificar Materia

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea modificar Materia.	2. En la ventana Administrar Materia presenta una tabla con la lista de materias.



3. La secretaria selecciona en la tabla la materia a modificar y pulsa el botón modificar.	4. El sistema habilita y carga los datos de la materia seleccionada.
5. Luego la secretaria modifica los datos de la materia y da un clic en <b>AGREGAR</b> .	6. El sistema almacena temporalmente los datos de Materia modificada.
7. Finalmente presiona <b>GUARDAR</b> para que los datos queden registrados en el sistema de forma permanente.	8. El sistema guarda la información ingresada.

Tabla. 8.3.6.3

### 8.3.6.1.1 DIAGRAMA DE ROBUSTES

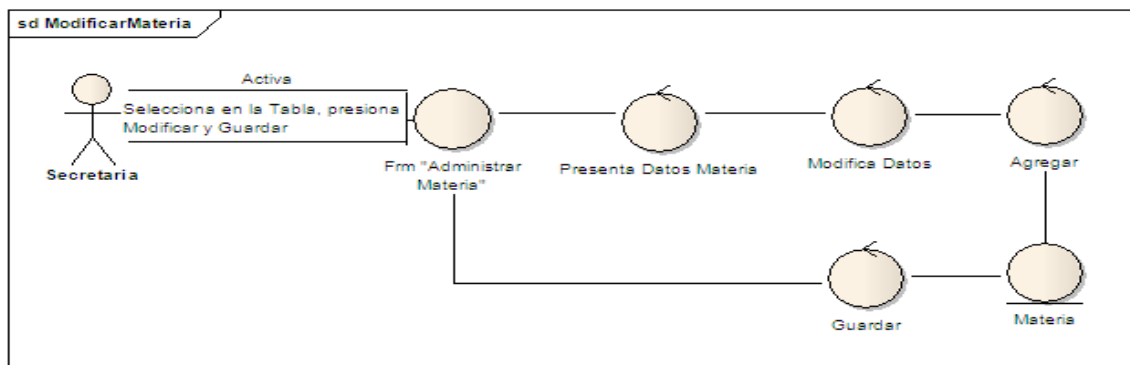


fig. 8.3.6.1.1

### 8.3.6.2.2 DIAGRAMA DE SECUENCIA

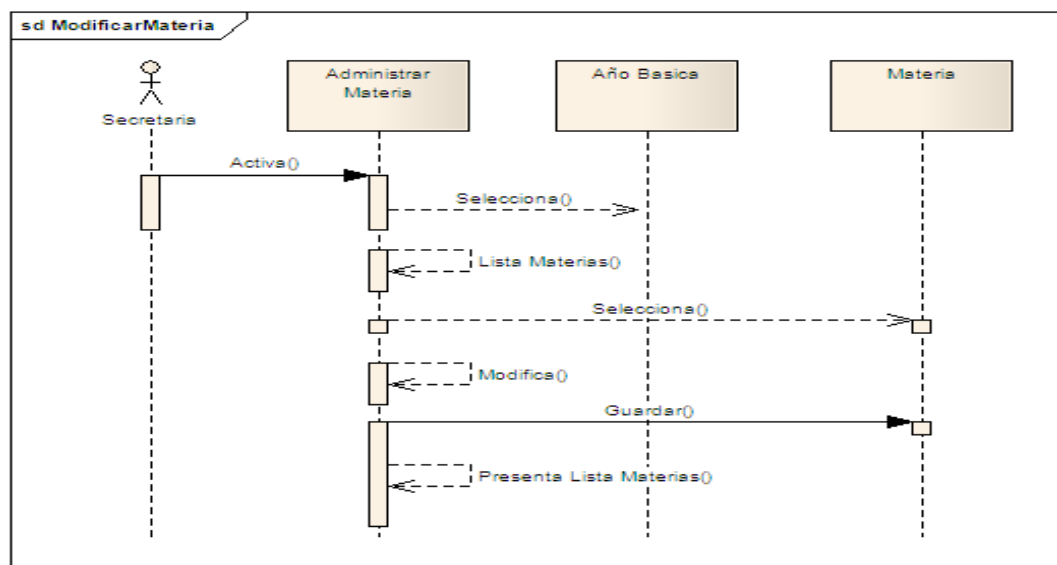


fig. 8.3.6.2.2

### 8.3.6.3 DIAGRAMA DE COLABORACIÓN

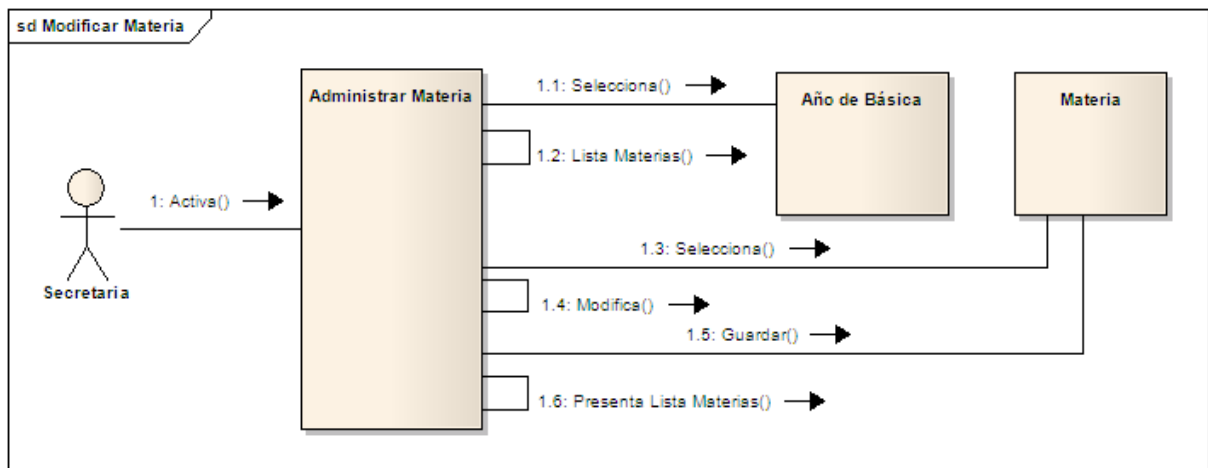


fig. 8.3.6.3.3

### 8.3.6.4 Sección: Eliminar Materia

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea Eliminar una Materia.	2. El sistema presenta una tabla con la lista de materias.
3. La secretaria en la tabla selecciona la materia a Eliminar y pulsa el botón <b>QUITAR</b> .	4. El sistema elimina la materia de la lista, presentando la nueva lista de materias actualizada.

Tabla. 8.3.6.4

### 8.3.6.4.1 DIAGRAMA DE ROBUSTEZ

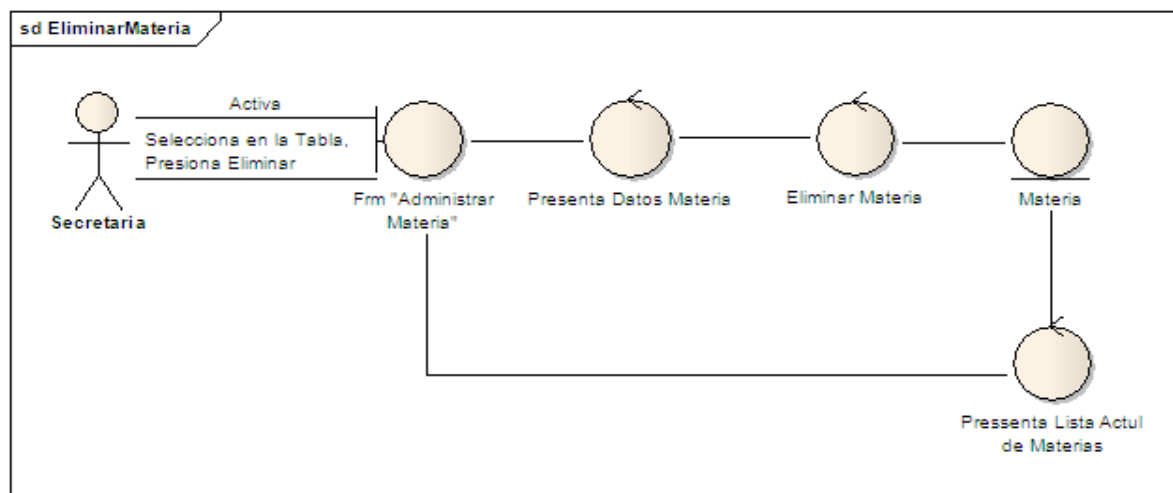


fig. 8.3.6.4.1

8.3.6.4.2 DIAGRAMA DE SECUENCIA

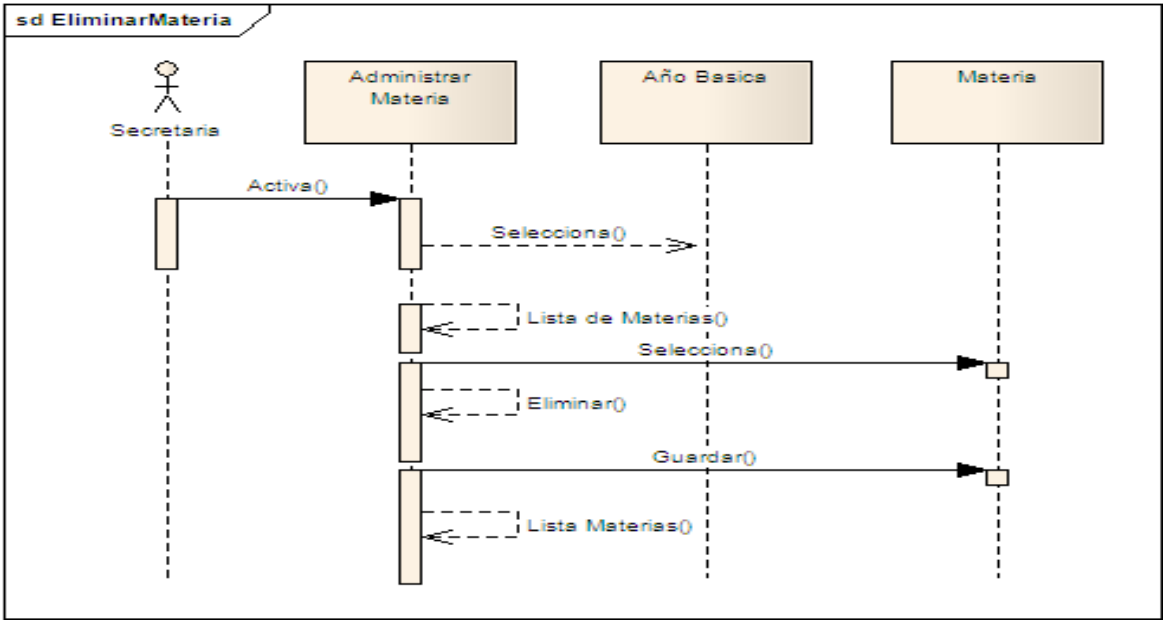


Tabla. 8.3.6.4.2

8.3.6.4.3 DIAGRAMA DE COLABORACIÓN

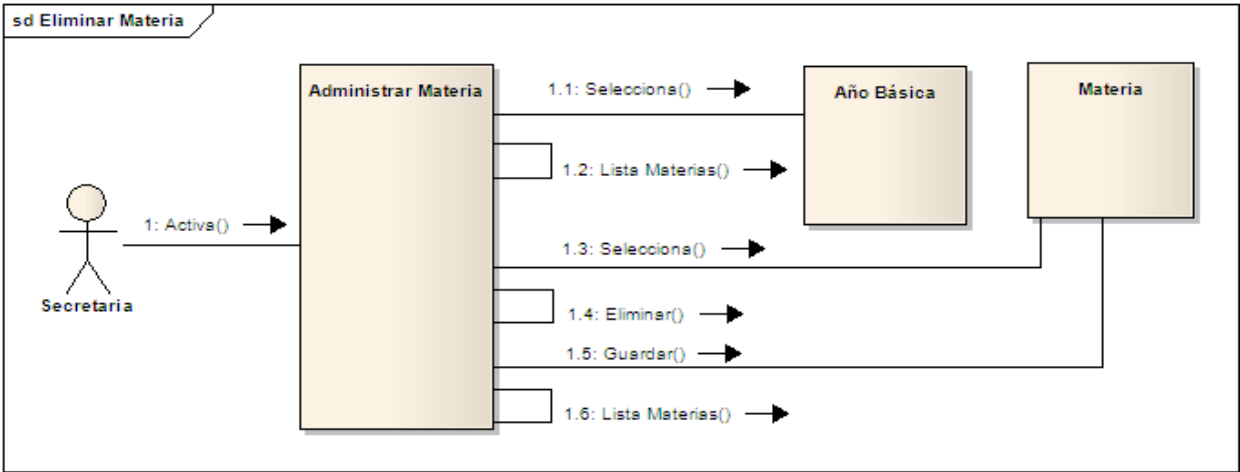


Tabla. 8.3.6.4.3

### 8.3.7 Caso de Uso: Administrar Alumno

The screenshot shows a window titled "Administrar Alumno". Inside, there is a section titled "Datos Alumno". The form includes the following fields and controls:

- Nombre: Text input field.
- Apellido: Text input field.
- Cedula: Text input field.
- Sexo: Radio buttons for "Masculino" and "Femenino".
- Fecha de Nacimiento: Dropdown menu showing "lunes 13 de diciembre de 2010".
- Edad: Text input field.
- Direccion Domicilio: Text input field.
- Estado: Text input field.
- Años Repetidos: Text input field with "0" and a spinner.
- Procedente de otra escuela: Checkable box.
- Escuela de Procedencia: Text input field.
- Direccion de la Escuela de Procedencia: Text input field.
- Buttons: "GUARDAR" (with a floppy disk icon), "MODIFICAR" (with a speech bubble icon), and "SALIR" (with a red X icon).

fig. 8.3.7. A

The screenshot shows a window titled "Diseño previo [AdministrarRepresentante]". Inside, there is a section titled "Datos Representante". The form includes the following fields and controls:

- Nombre: Text input field.
- Apellido: Text input field.
- Cedula: Text input field.
- Sexo: Radio buttons for "Masculino" and "Femenino".
- Domicilio: A container with two sub-fields: "Direccion" and "Telefono", each with a text input field.
- Ocupacion: Text input field.
- Trabajo: A container with two sub-fields: "Direccion" and "Telefono", each with a text input field.
- Buttons: "GUARDAR" (with a floppy disk icon), "MODIFICAR" (with a speech bubble icon), and "SALIR" (with a red X icon).

fig. 8.3.7. B

<b>Caso de uso 006</b>	Administrar datos Alumno
<b>Actores</b>	Secretaria, Representante (iniciador)
<b>Propósito</b>	Mantener Actualizada la información del alumno.
<b>Resumen</b>	El Representante llega a la dirección para actualizar los datos del estudiante, informa a la secretaria y ella ingresa aquellos datos en el sistema, en la ventana Administrar Alumno o también los datos se actualizan cuando el representante llega a matricular al alumno.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R1.5 R2.1 y R2.3

**Tabla. 8.3.7**

#### **8.3.7.1 CURSO NORMAL DE EVENTOS**

Acción del actor	Respuesta del sistema
1. Empieza cuando hay alumnos nuevos o se necesita realizar alguna modificación la secretaria pide datos al representante.	2. El sistema muestra la ventana Administrar Alumno.
3. Si el alumno ya consta en el sistema deberá ir a la sección buscar alumno.	4. El sistema muestra la ventana buscar alumno.
5. Luego de buscarlo si se necesita hacer alguna modificación deberá ir a la sección Modificar alumno.	6. El sistema presenta en la ventana Administrar Alumno el botón modificar.
7. Si el alumno es nuevo deberá ir a la sección Agregar alumno.	8. El Sistema presenta en la ventana el botón nuevo.

**Tabla. 8.3.7.1**

### 8.3.7.2. Sección: buscar alumno

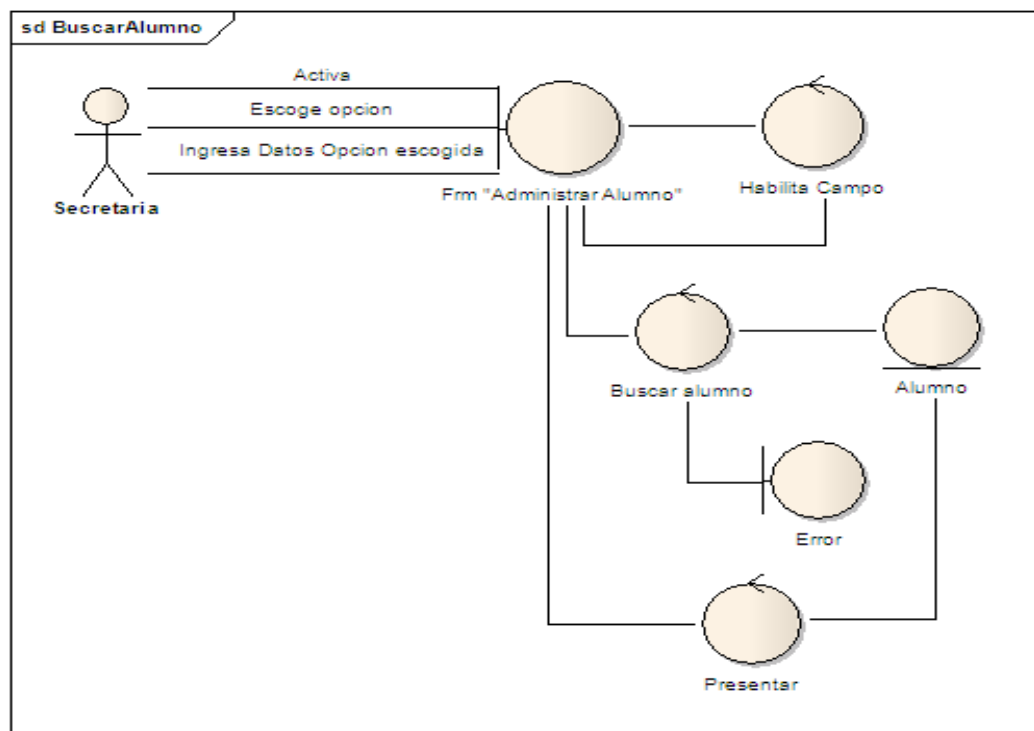
Acción del actor	Respuesta del sistema
13. Empieza cuando la secretaria desea buscar un alumno, escogerá buscar por apellido o cédula.	14. Presenta en una caja de texto diferentes opciones de apellidos o cédulas similares.
15. La secretaria escoge el apellido o cédula correspondiente.	16. Presenta la ventana administrar alumno con los datos encontrados del estudiante.

**Tabla. 8.3.7.2**

Cursos alternos

- Ítem 3: Si no hay el apellido o cédula se presentará un mensaje donde indica que el alumno no consta en el sistema.

#### 8.3.7.2.1 DIAGRAMA DE ROBUSTEZ



**fig. 8.3.7.2**

### 8.3.7.2.2 DIAGRAMA DE SECUENCIA

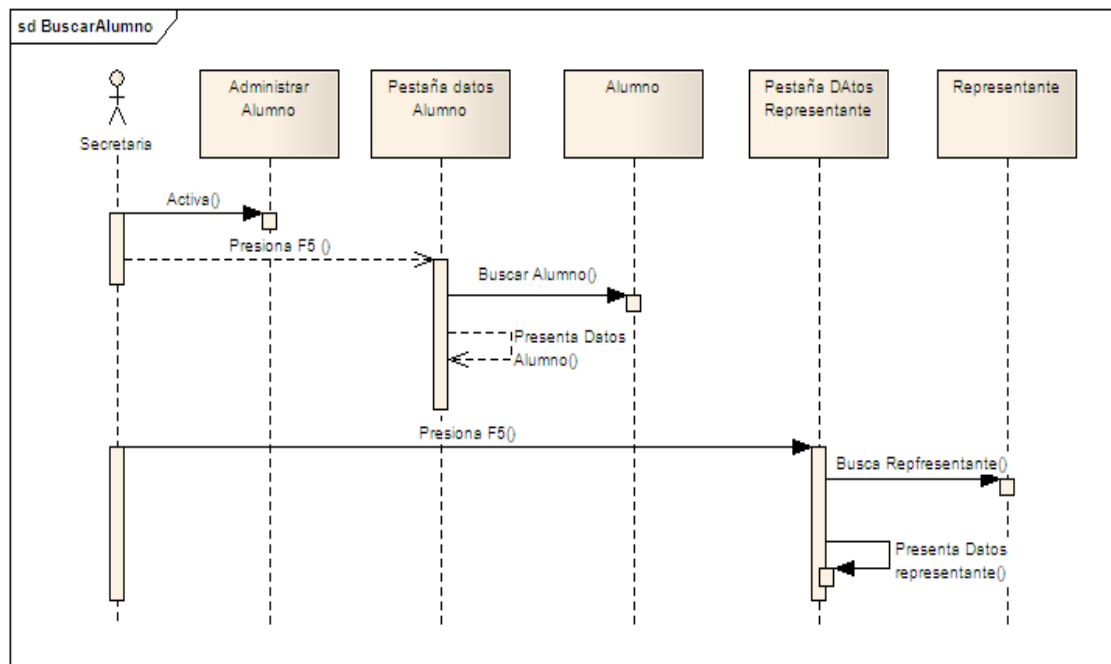


fig. 8.3.7.3

### 8.3.7.2.3 DIAGRAMA DE COLABORACIÓN

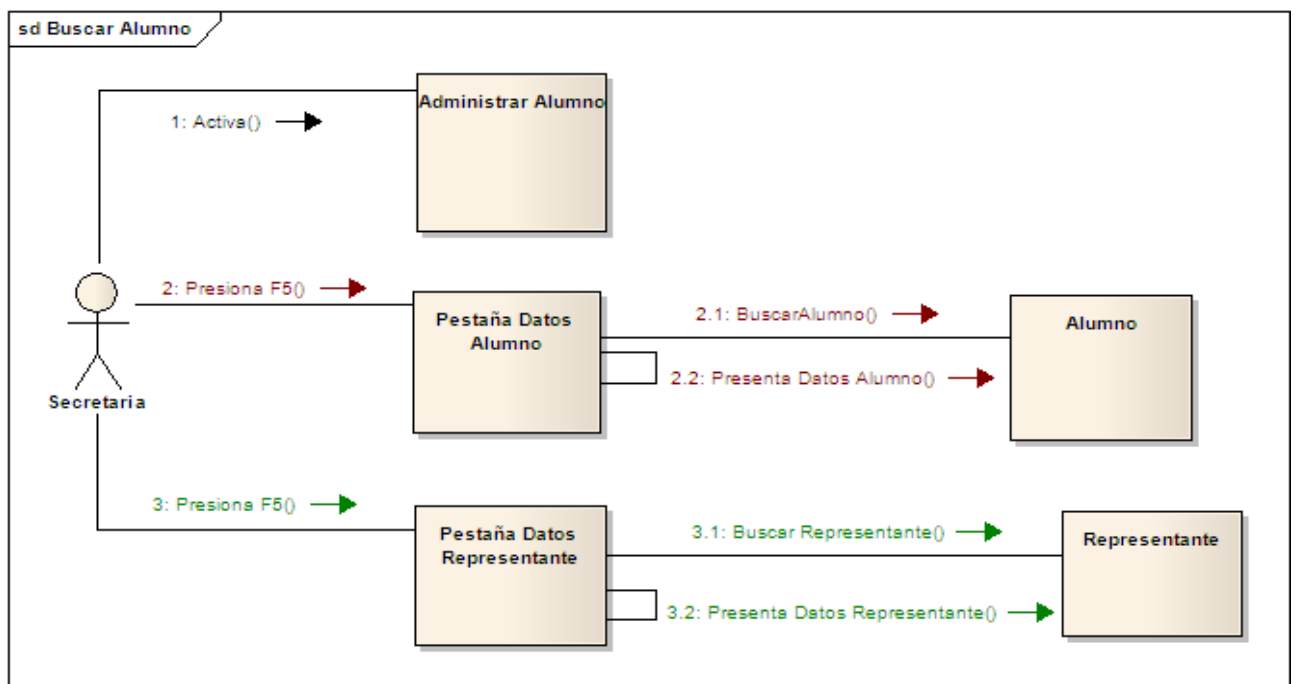


fig. 8.3.7.2.3

**8.3.7.3 Sección: agregar alumno**

Acción del actor	Respuesta del sistema
1. Empieza cuando el alumno es nuevo, la secretaria presionará el botón Nuevo.	2. El Sistema presenta la ventana Administrar Alumno con todos los campos vacíos y con la opción buscar alumno deshabilitada.
3. La Secretaria procede a ingresar los datos del alumno llenando todos los campos ya que son obligatorios.	4. El sistema valida el número de cédula y la información ingresada.
5. La secretaria presiona el botón guardar.	6. El Sistema almacena los datos ingresados y lo agrega al año correspondiente.
7. La secretaria procede a ingresar los datos de la persona que representa en la Ventana Administrar Representante al estudiante ingresado y presiona el botón guardar.	8. El Sistema almacena los datos del representante validando primero el número de cédula.

**Tabla. 8.3.7.3****Cursos alternos**

- Item 4: Si la cédula no es válida el sistema presentara un mensaje de error.
- Item 6: Si no están llenos todos los campos obligatorios el sistema presentará un mensaje mostrando que no puede almacenar los datos.
- Item 8: Si no están llenos todos los campos obligatorios acerca del representante o si la cédula es inválida, el sistema presenta un mensaje de error



### 8.3.7.3.1D IAGRAMA DE ROBUSTEZ

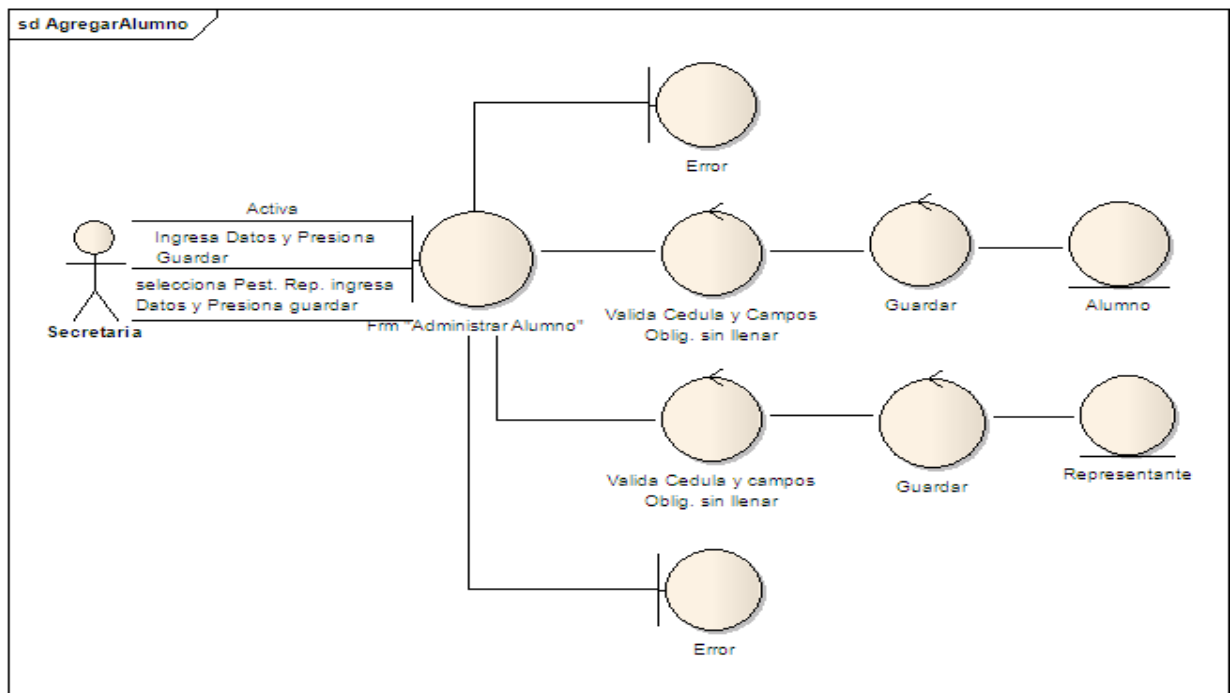


fig. 8.3.7.3.1

### 8.3.7.3.1 DIAGRAMA DE SECUENCIA

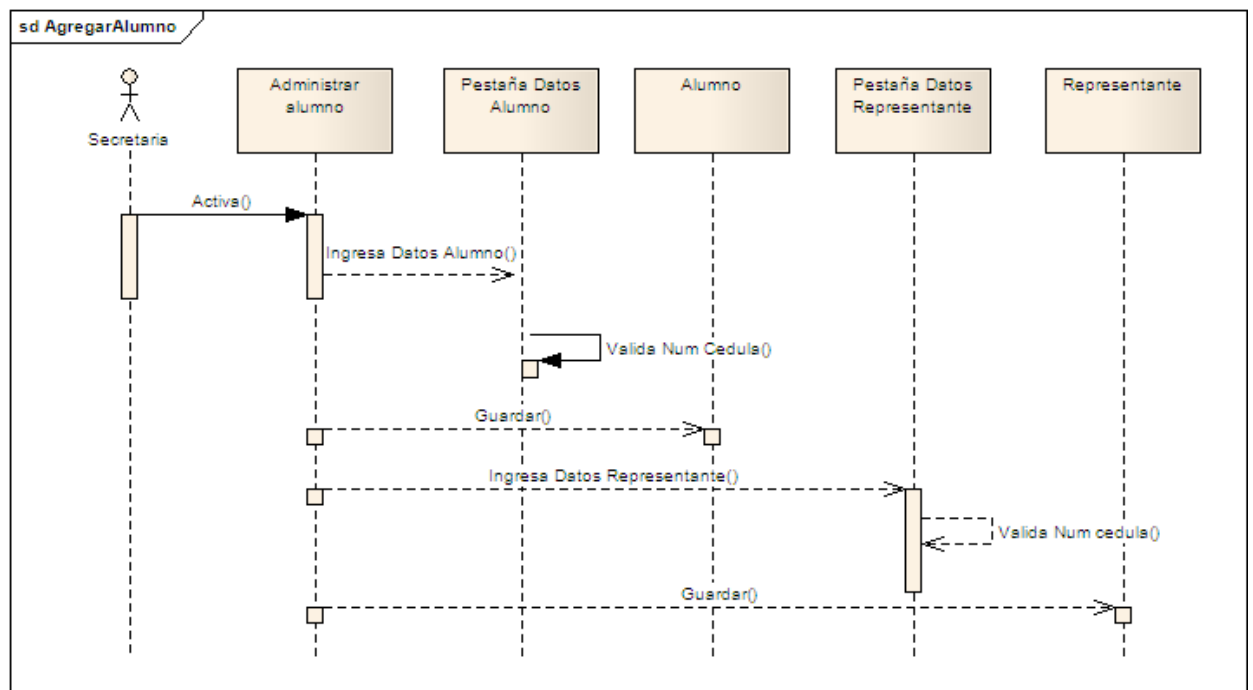


fig. 8.3.7.3.1

### 8.3.7.3.2 DIAGRAMA DE COLABORACIÓN

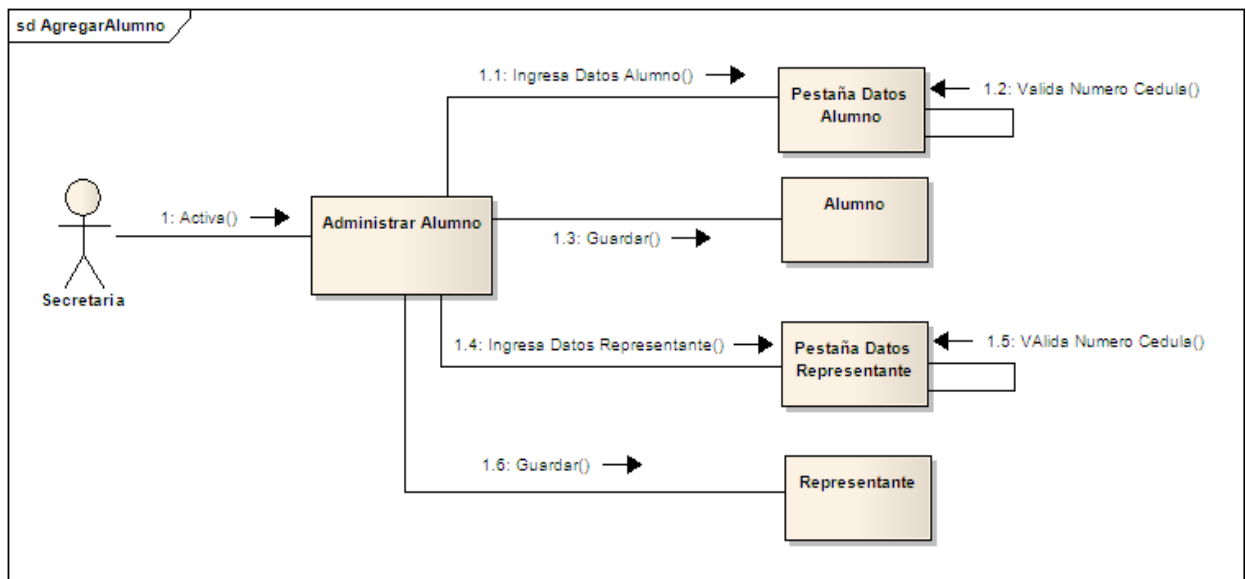


fig. 8.3.7.3.2

### 8.3.7.4 Sección: modificar alumno

Acción del actor	Respuesta del sistema
1. Empieza cuando el alumno ya consta en el sistema.	
2. La secretaria procede a buscarlo, ya sea por apellido o cédula.	3. El Sistema presenta una caja de texto con las opciones apellido y cédula.
4. La secretaria escoge la opción correspondiente y presiona Buscar.	5. Presenta la ventana administrar alumno con los datos ya encontrados.
6. La secretaria presiona el botón modificar si requiere modificar algo y luego presiona el botón guardar.	7. El Sistema permite que se modifiquen, almacenen y actualicen los datos ingresados, presentándolos en la ventana respectiva.

Tabla. 8.3.7.4

### 8.3.7.4.1 DIAGRAMA DE ROBUSTES

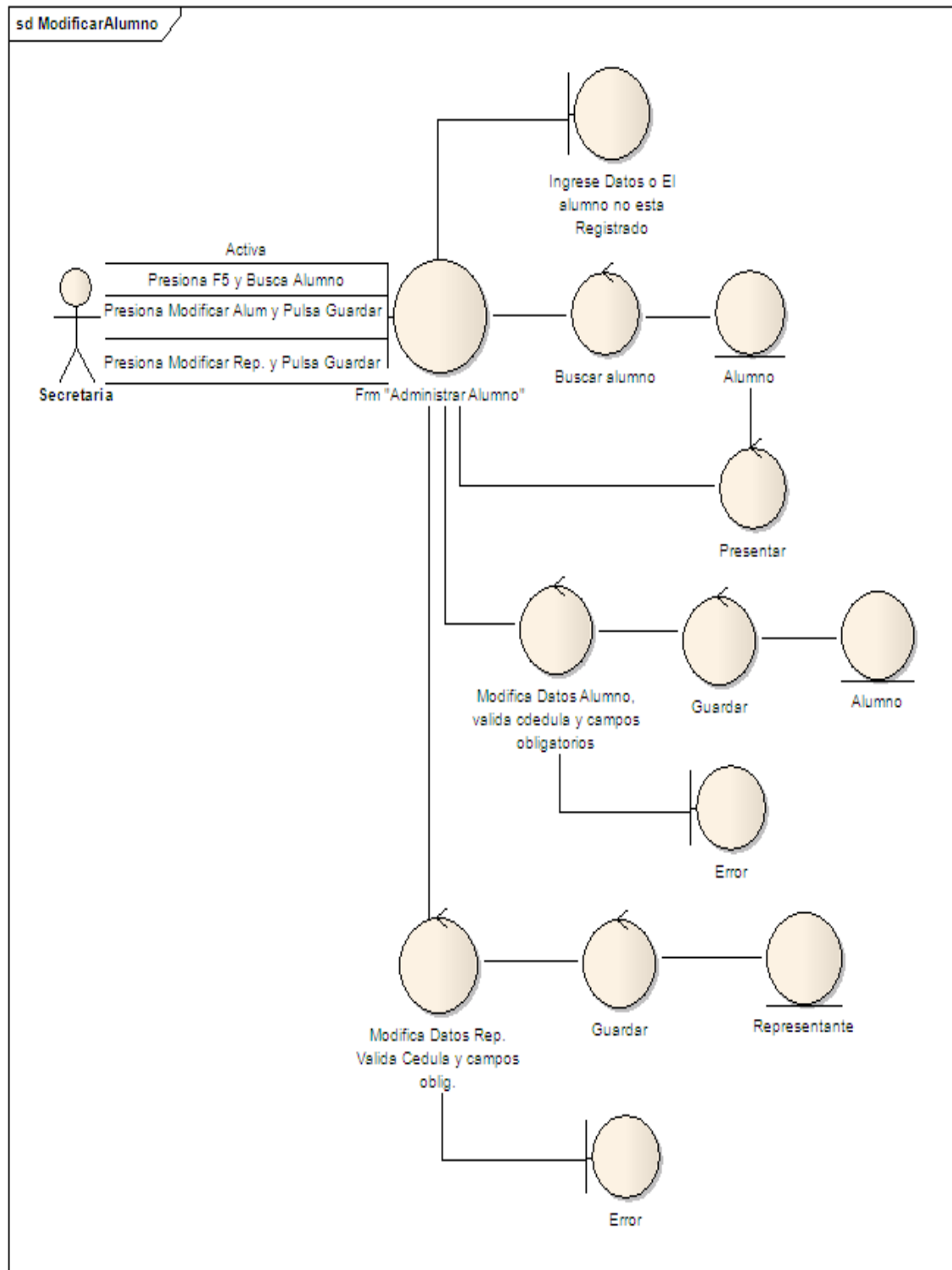


fig. 8.3.7.4.1

### 8.3.7.4.2 DIAGRAMA DE SECUENCIA

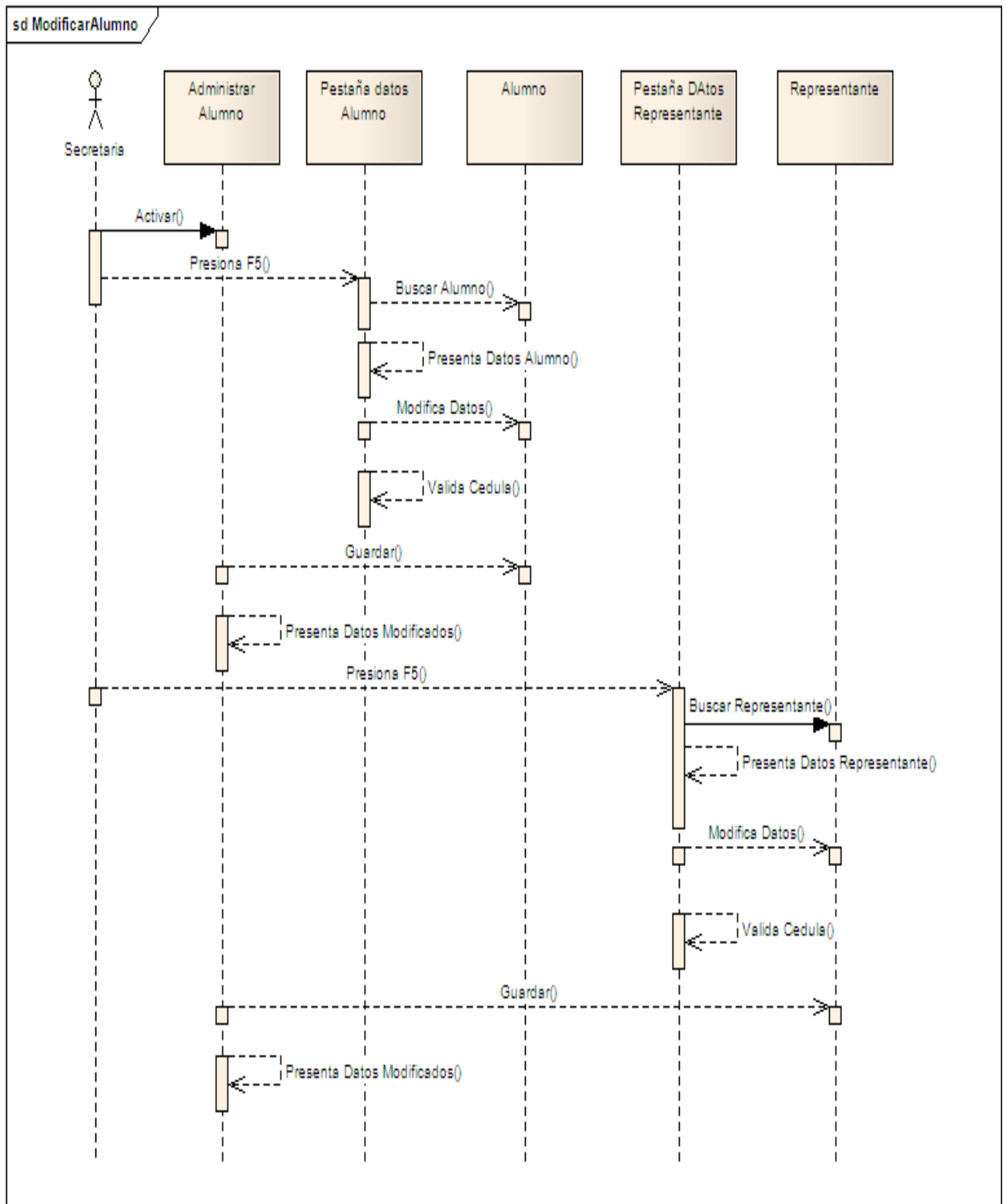


fig. 8.3.7.4.2

### 8.3.7.4.3 DIAGRAMA DE COLABORACIÓN

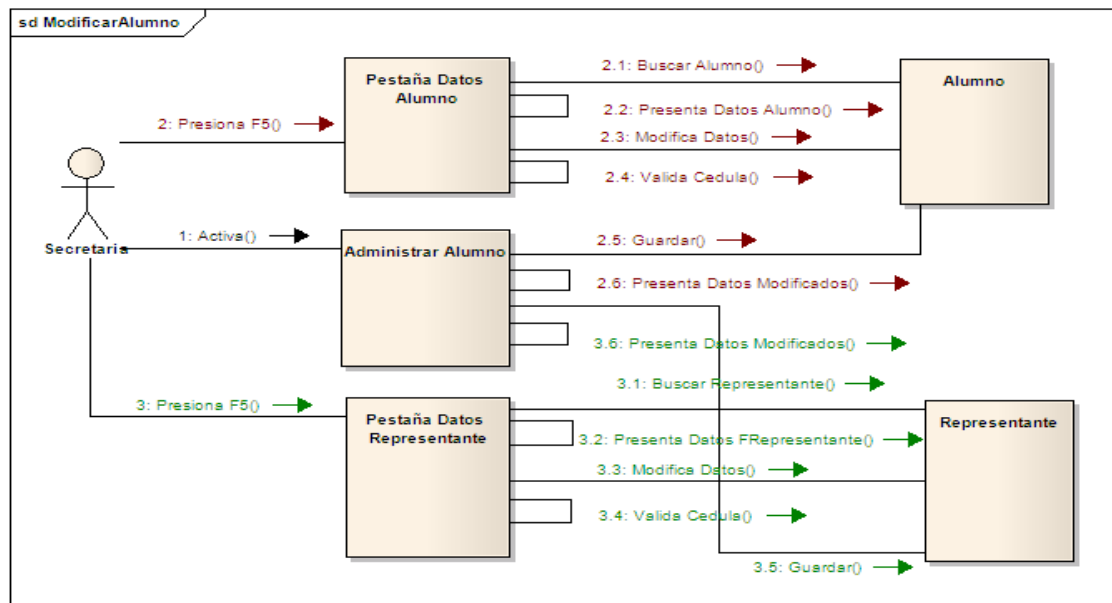


fig. 8.3.7.4.3

### 8.3.8 Caso de Uso: Registrar Matricula

fig. 8.3.8

<b>Caso de uso 007</b>	Registrar Matricular
<b>Actores</b>	Secretaria, Representante (iniciador)
<b>Propósito</b>	Ingresar los datos del niño en el sistema para que conste como

	matriculado.
<b>Resumen</b>	La secretaria ingresa al sistema en la ventana Administrar Alumno y posteriormente empieza a ingresar los datos personales importantes del alumno si es nuevo o, si el alumno ya consta en el sistema los presenta en la ventana para que se modifiquen si es necesario.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R 1.5, R2.1, R2.3, R2.4, R2.5

**Tabla. 8.3.8**

### 8.3.8.1 CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. El caso de uso inicia cuando el representante llega a la dirección a matricular al niño.	
2. El docente activa la ventana de Matriculación de alumnos. 4. Si el alumno existe la secretaria va al menú buscar y pulsa el ítem buscar alumno. En caso contrario pulsa el botón nuevo alumno. Ir a la sección Administrar Alumno.	3. Se carga la pantalla de Matriculación de alumnos 5. Presenta opciones para buscar por apellido o por cédula.
6. Luego de Administrar el alumno ya sea nuevo o un alumno que ya consta en el sistema presiona Guardar.	7. El sistema guarda la matricula del alumno.

**Tabla. 8.3.8.1**

#### Cursos alternos

- Ítem 3: Controla mediante mensajes de diálogo si el alumno existe o no.

### 8.3.8.2 Sección: Agregar Matricula

Acción del actor	Respuesta del sistema
1. Empieza cuando el docente activa la ventana matriculación.	2. Presenta la ventana matriculación.
3. Las listas despegables y las cajas de texto se llenan con los datos del alumno y las listas despegables con los años y paralelos activados.	4. El sistema presenta estas cajas de texto y estas listas retornando lo almacenado en la base de datos.
5. El docente luego de haber escogido el año y paralelo presiona el botón guardar.	6. El Sistema almacena o agrega la matrícula.

fig. 8.3.8.2

#### 8.3.8.2.1 Diagrama de Robustez

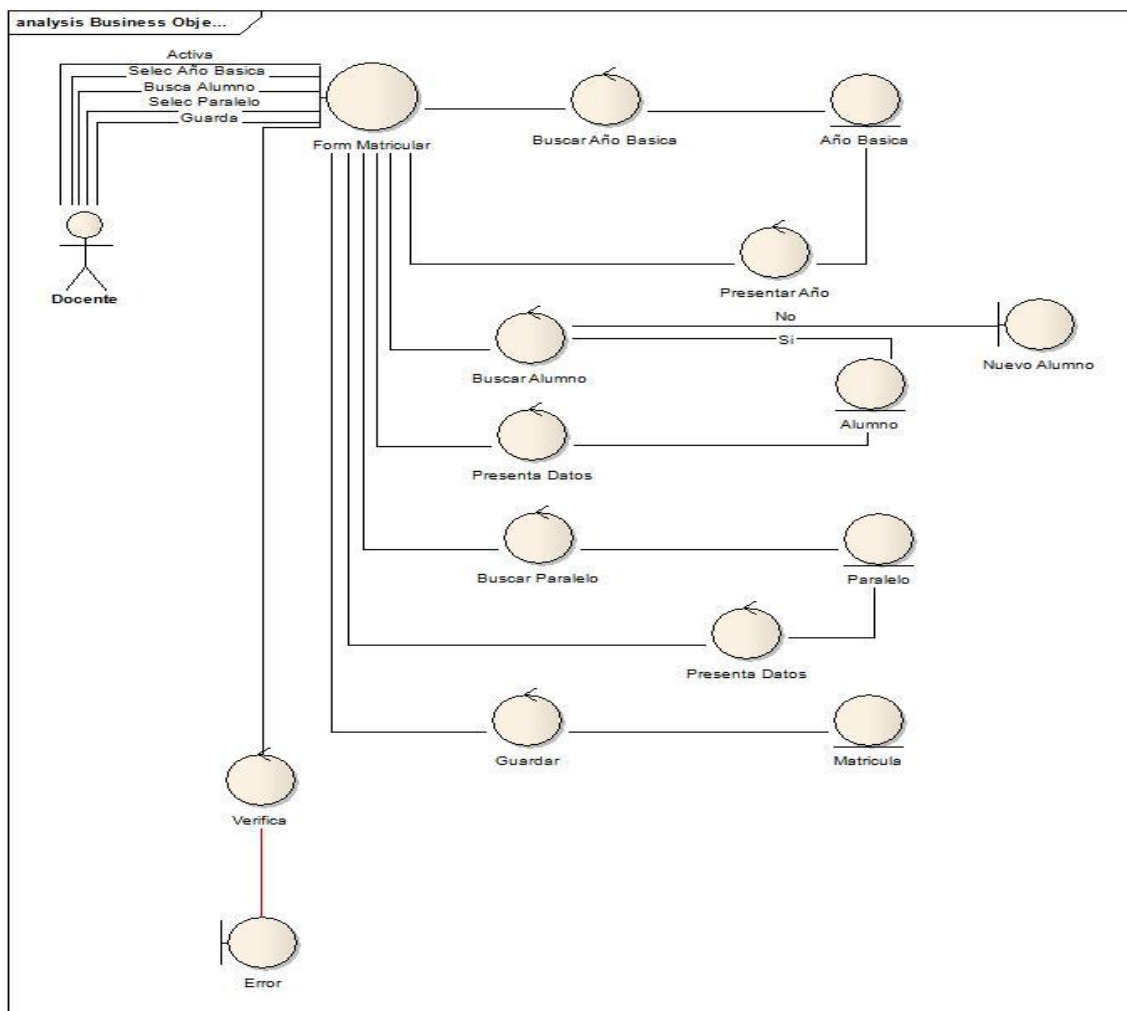


fig. 8.3.8.2.1

## 8.3.8.2.2 Diagrama de Secuencia

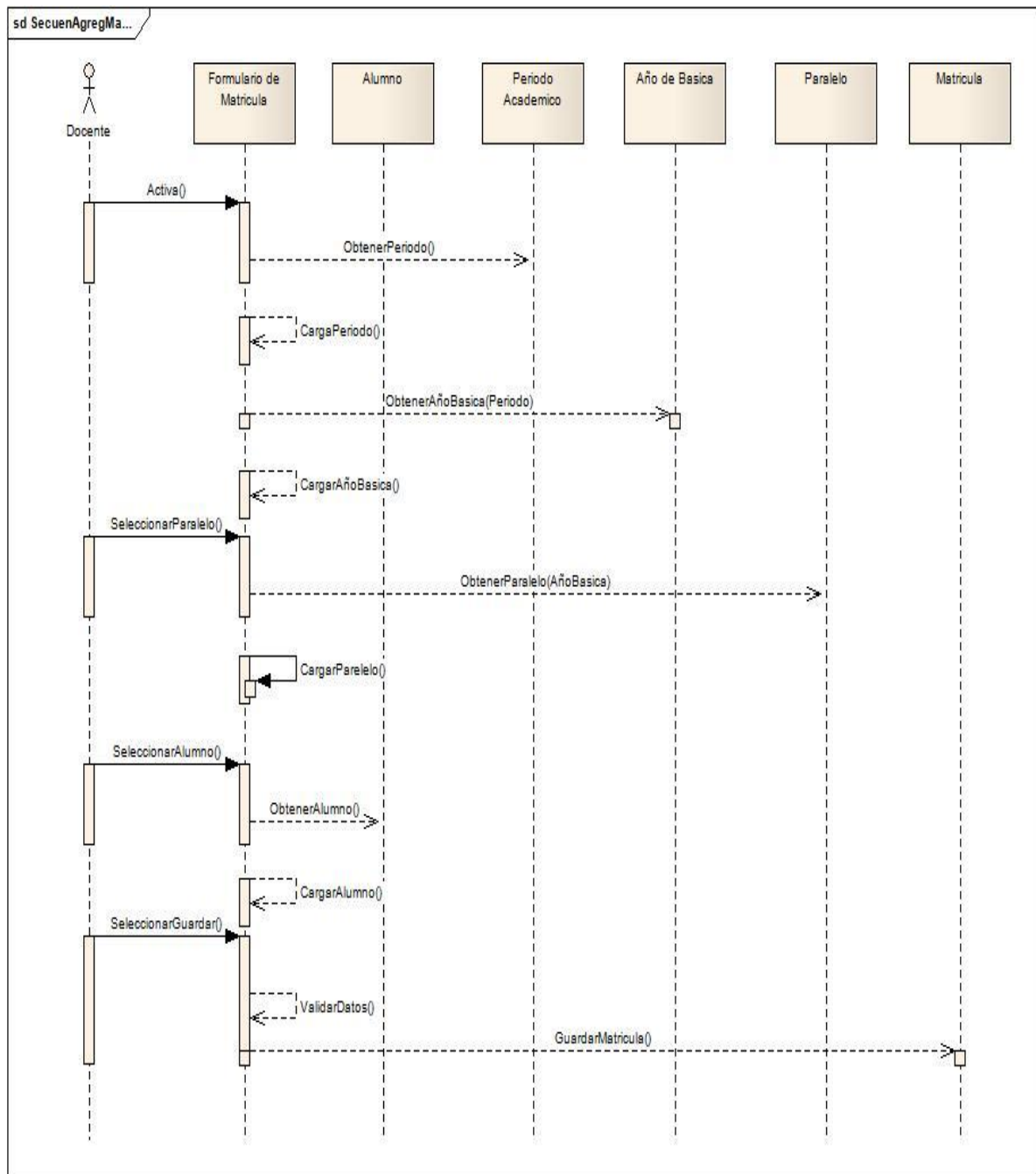


fig. 8.3.8.2.2



## 8.3.8.2.3 Diagrama de Colaboración

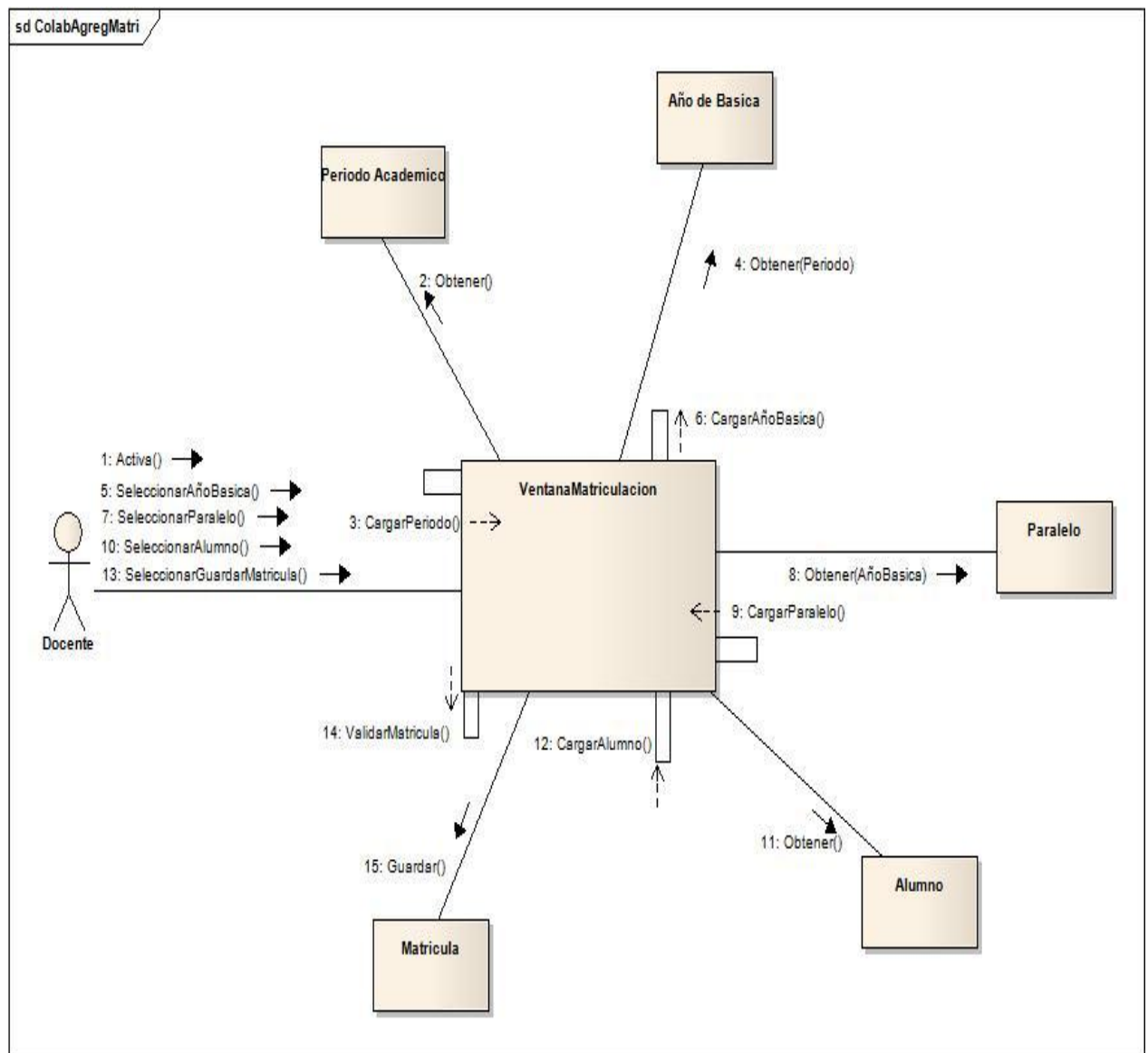


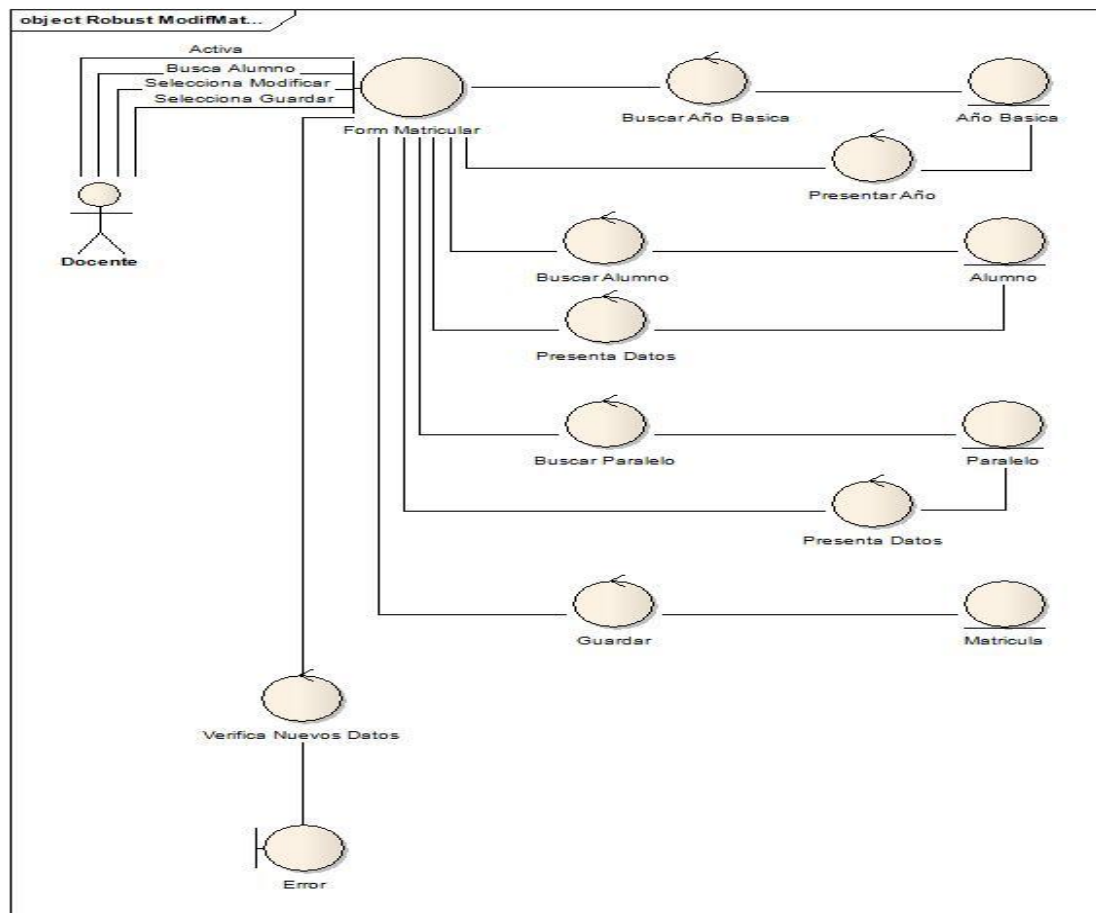
fig. 8.3.8.2.3

### 8.3.8.3 Sección: Modificar Matricula

Acción del actor	Respuesta del sistema
1. Empieza cuando el docente activa la ventana matriculación.	2. Presenta la ventana matriculación.
3. El docente ingresa el número de cedula del alumno en la caja de texto Cedula y presiona enter o en caso de no saberlo da clic en el botón buscar alumno ya se por cedula o por apellido	4. El sistema busca en la base de datos al alumno matriculado y presenta esos datos en la ventana de matriculación.
5. El docente luego de haber escogido el año y paralelo presiona el botón guardar matricula.	6. El Sistema almacena la matricula.

**Tabla. 8.3.8.3**

#### 8.3.8.3.1 Diagrama de Robustez



**fig. 8.3.8.3.1**

## 8.3.8.3.2 Diagrama de Secuencia

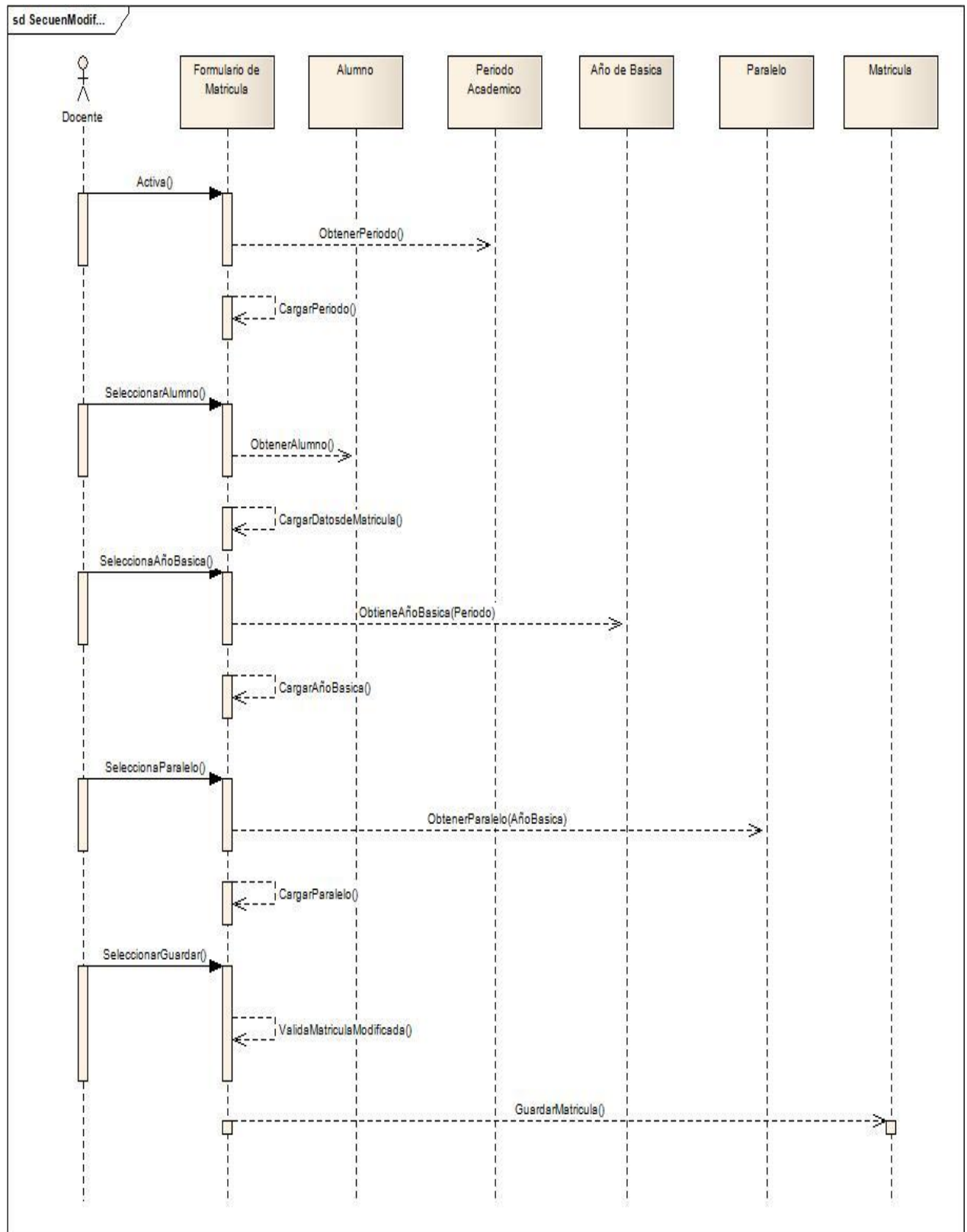


fig 8.3.8.3.2

### 8.3.8.3.3 Diagrama de Colaboración

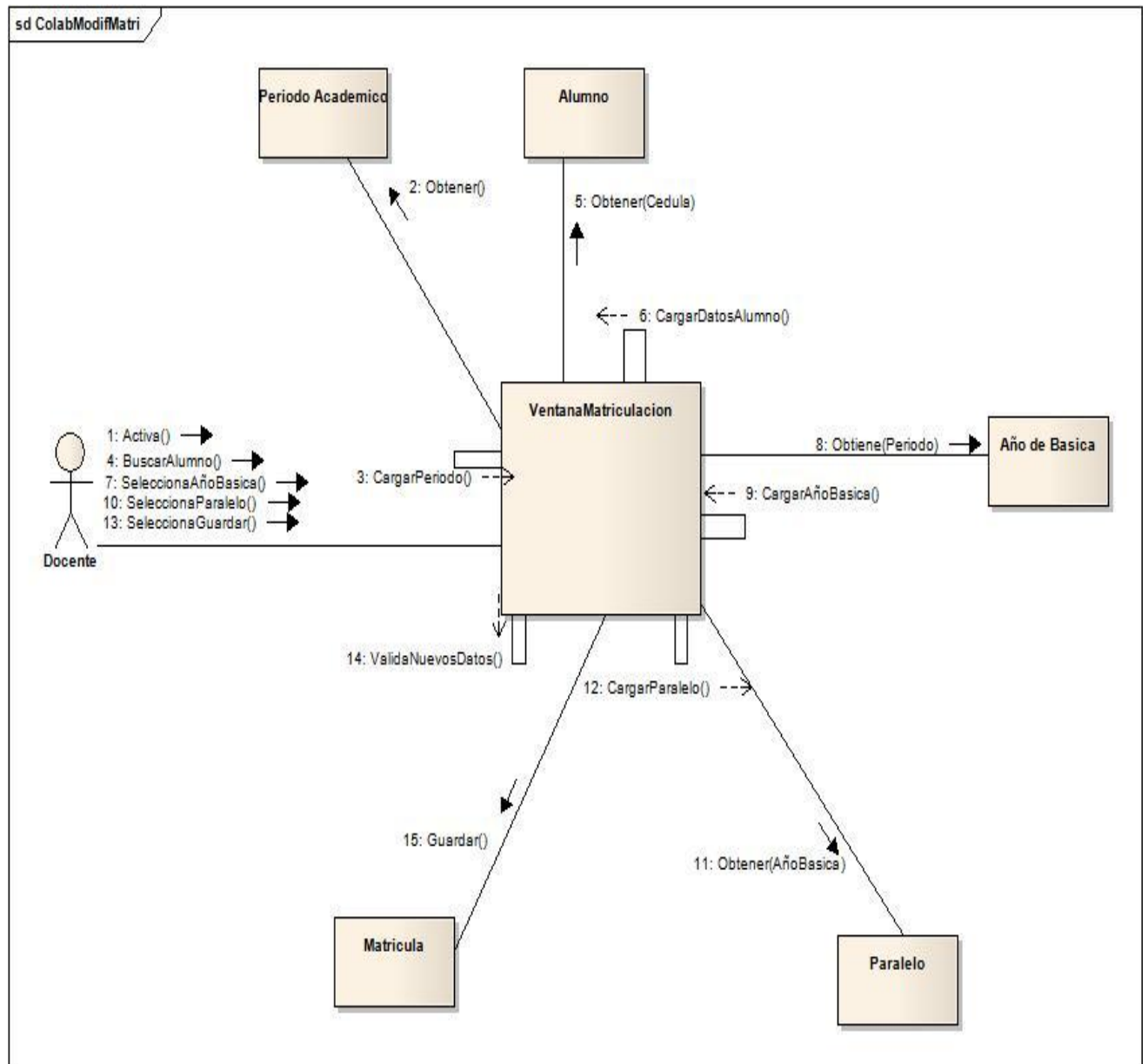


Fig. 8.3.8.3.3

### 8.3.9 Caso de Uso: Registrar Asistencias

[illegible]

**Fig. 8.3.9**

<b>Caso de uso 008</b>	Registrar Asistencias
<b>Actores</b>	Secretaria, Docente (iniciador)
<b>Propósito</b>	Controlar que el estudiante asiste lo que es parte de la participación
<b>Resumen</b>	El docente lleva a la secretaria las asistencias correspondientes a su paralelo y la secretaria ingresa al sistema a la ventana Asistencias la cual permite que se busque un paralelo y se pueda ingresar sus asistencias correspondientes al ciclo de estudio.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R3.6

**Tabla. 8.3.9**

### 8.3.9.1 CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Empieza cuando el docente lleva el total de las asistencias trimestrales de los alumnos.	
2. El docente entrega ese informe a la secretaria ir a la sección Agregar Asistencia	3. Presenta la ventana Asistencias
4. Si las asistencias ya constan y ocurrió una equivocación y se desea modificar ir a la sección Modificar Asistencias	5. Presenta en la ventana Asistencias el botón modificar.
6. Luego de que se haya ingresado o haya modificado la secretaria presiona el botón calcular.	7. El sistema guarda las nuevas asistencias en la base de datos. 8. Al finalizar el año lectivo el sistema calcula el total de días asistidos y no asistidos del alumno.

**Tabla. 8.3.9.1**

#### Cursos alternos

- Ítem 6: Si el alumno no existe se visualizara un mensaje de diálogo que presentará un error de que el alumno no consta en el sistema.

### 8.3.9.2 Sección: Agregar Asistencia

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria va a buscar un paralelo las listas despegables se llenan de acuerdo con los años y paralelos creados	2. El Sistema retorna los años y paralelos de la base de datos a las listas.

3. La secretaria ingresa las asistencias alumno por alumno en un cuadro de dialogo interno.	4. Presenta en la ventana un botón Guardar
5. La secretaria presiona el botón guardar.	6. El sistema almacena las asistencias las agrega a cada alumno en la base de datos.

Tabla. 8.3.9.2

## 8.3.9.2.1 Diagrama de Robustez

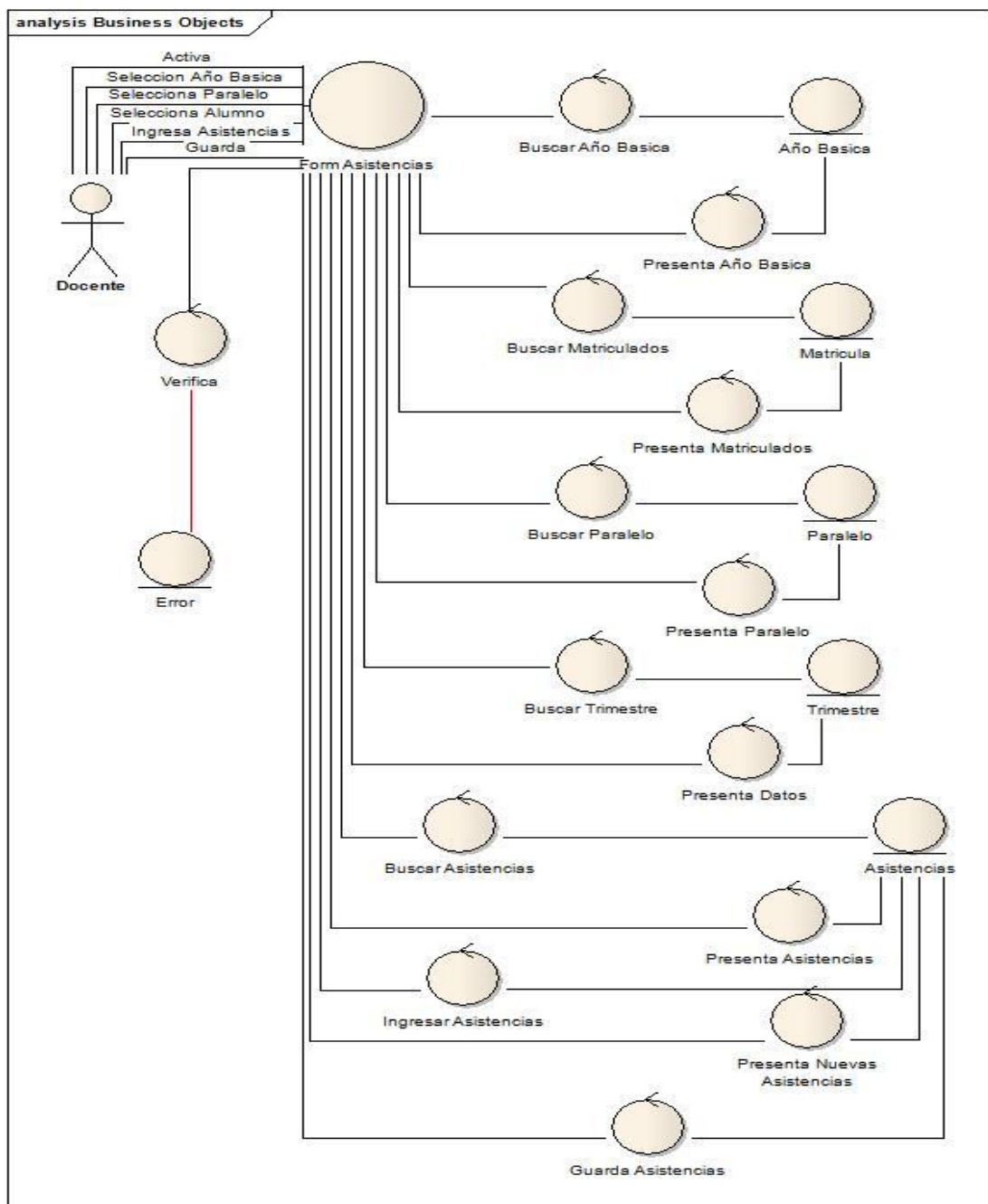


Fig. 8.3.9.2.1

### 8.3.9.2.2 Diagrama de Secuencia

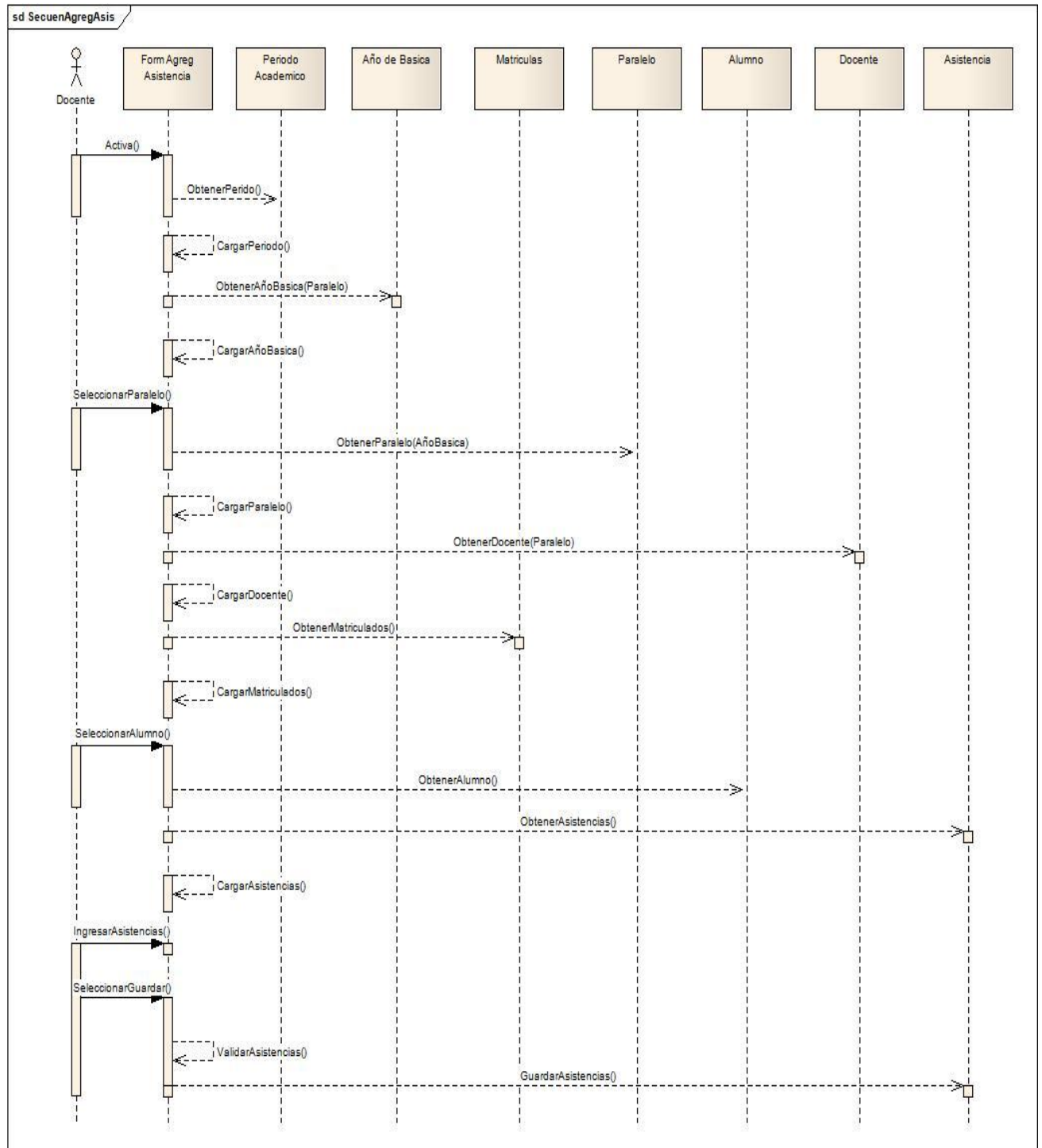


fig. 8.3.9.2.2



### 8.3.9.2.3 Diagrama de Colaboración

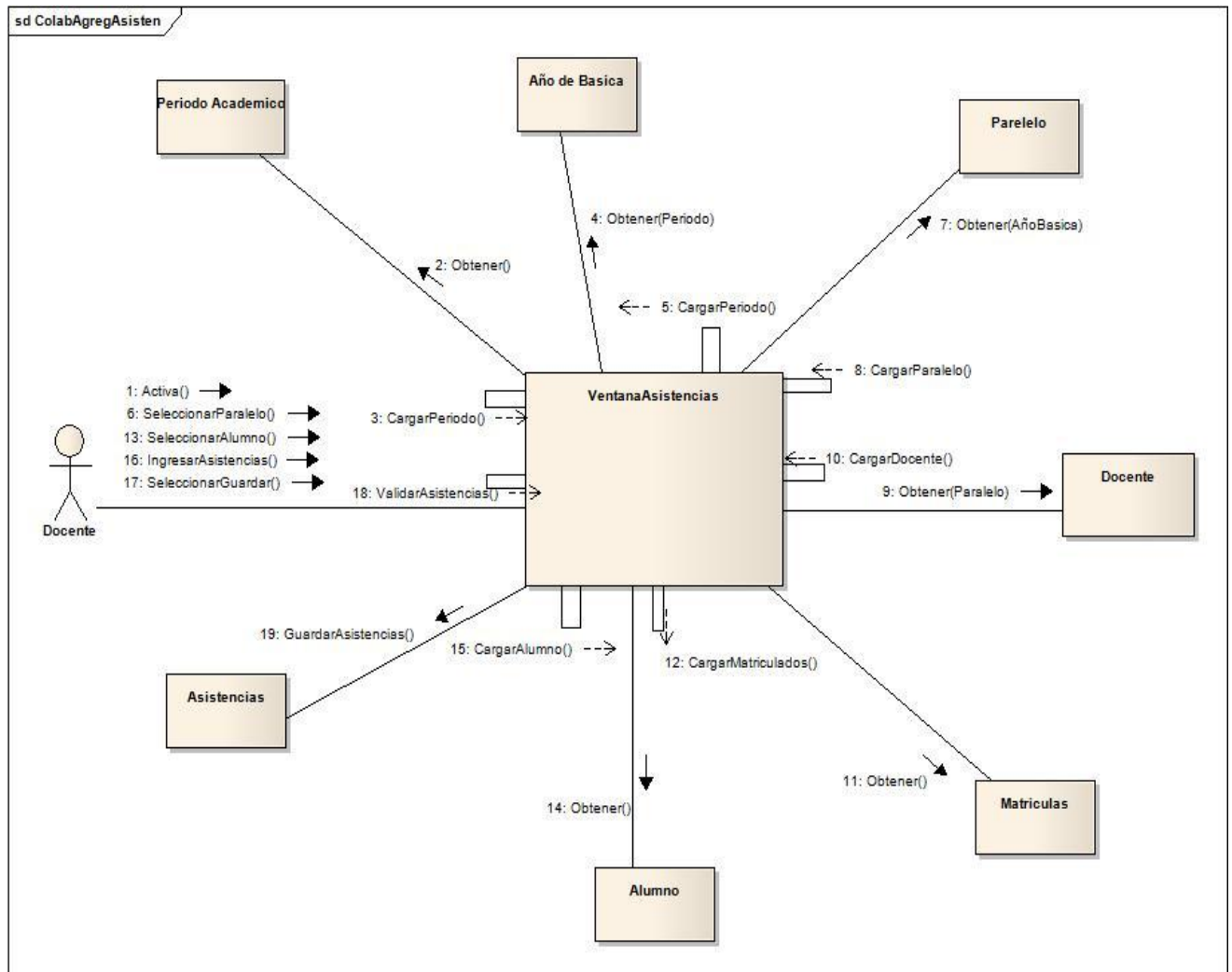


Tabla. 8.3.9.2.3

### 8.3.9.3 Sección: Modificar Asistencia

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria desea modificar las Asistencias.	2. Presenta en un botón modificar.
3. La secretaria da click en modificar. Y a continuación edita los campos	4. Presenta en la ventana el botón aceptar.

que desea.	
5. La secretaria presiona el botón guardar.	6. Actualizara las asistencias del alumnos en la base de datos.

Tabla. 8.3.9.3

## 8.3.9.3.1 Diagrama de Robustez

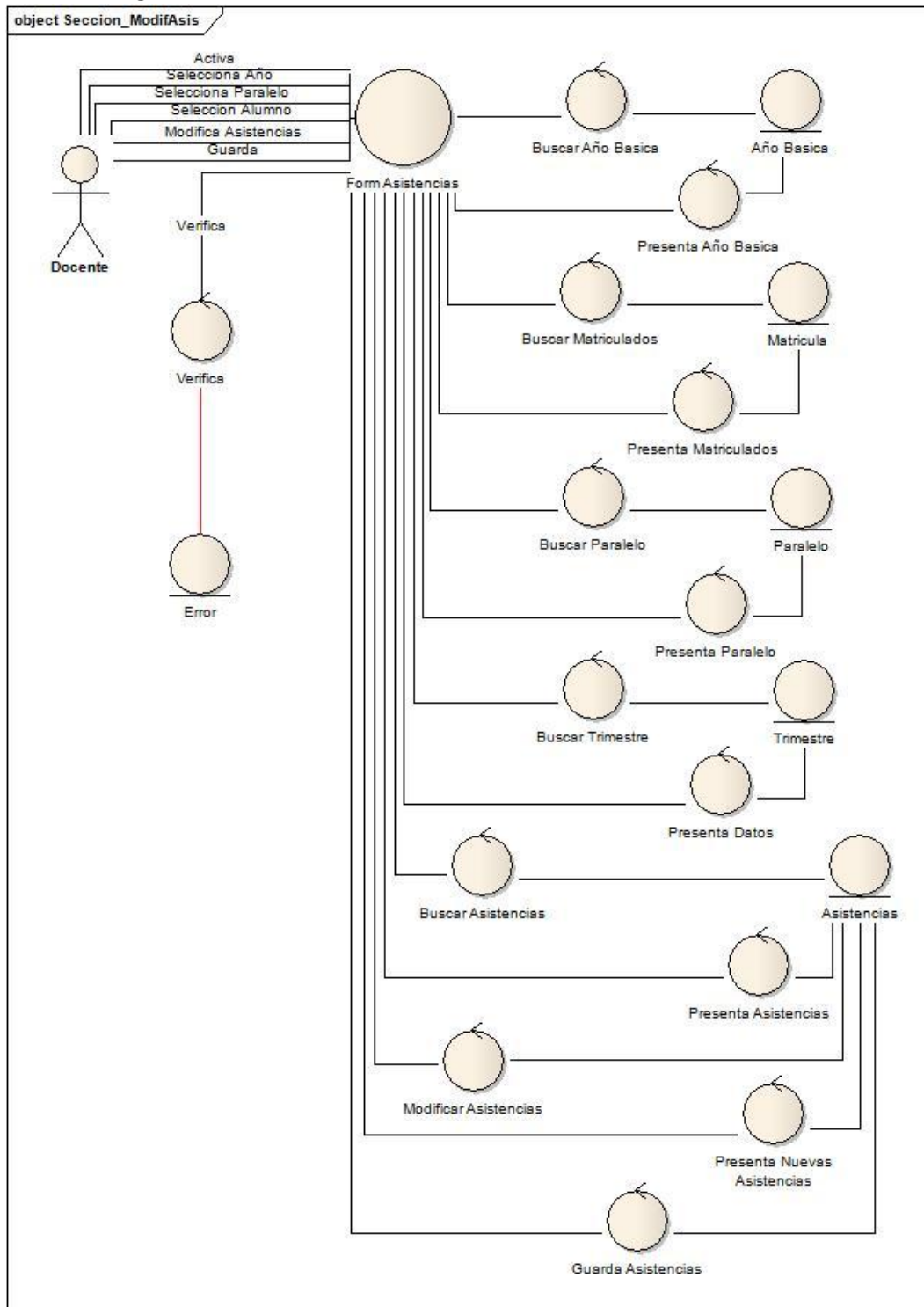


Tabla. 8.3.9.3.1

## 8.3.9.3.2 Diagrama de Secuencia

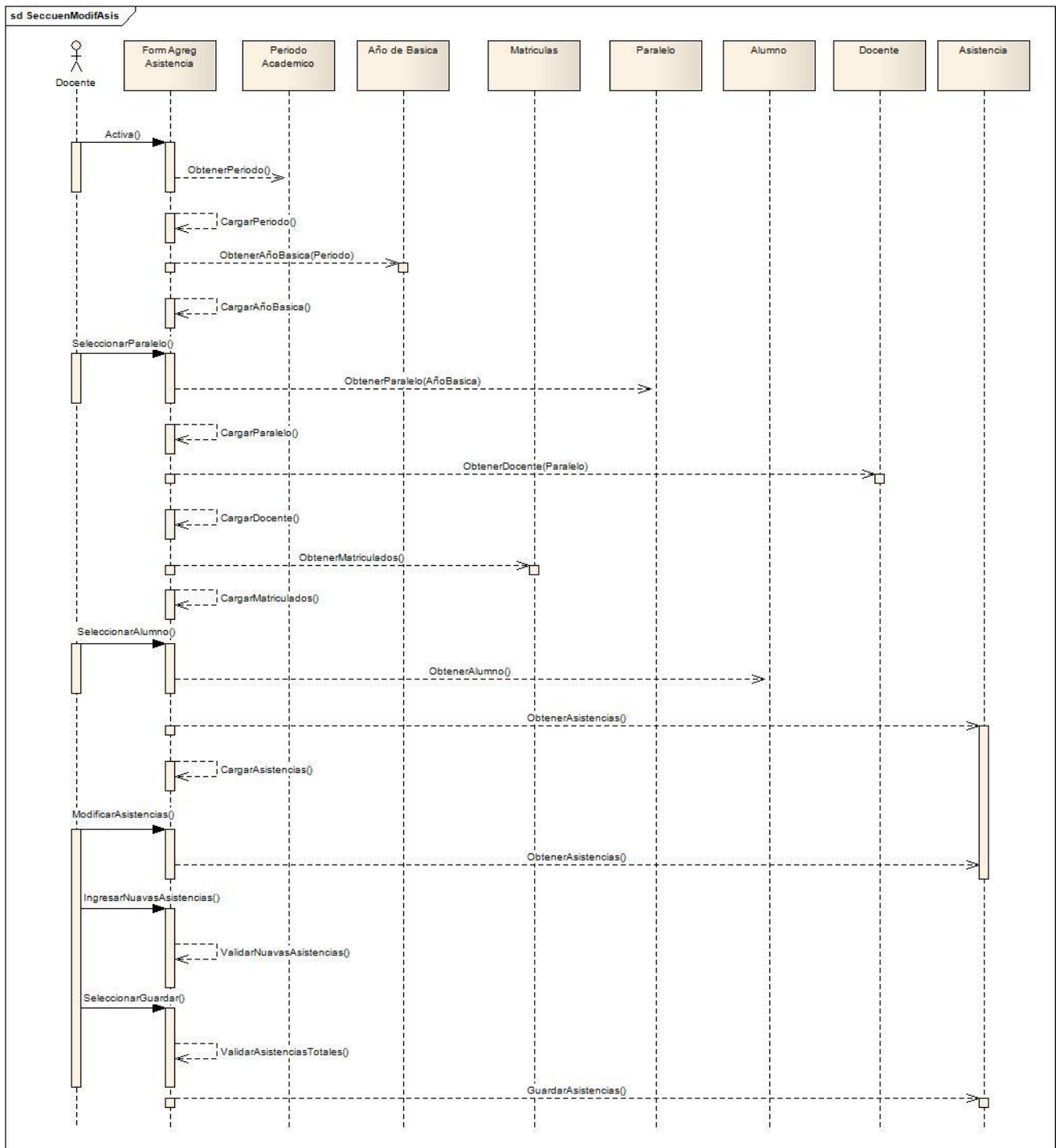


fig. 8.3.9.3.2

## 8.3.9.3.3 Diagrama de Colaboración

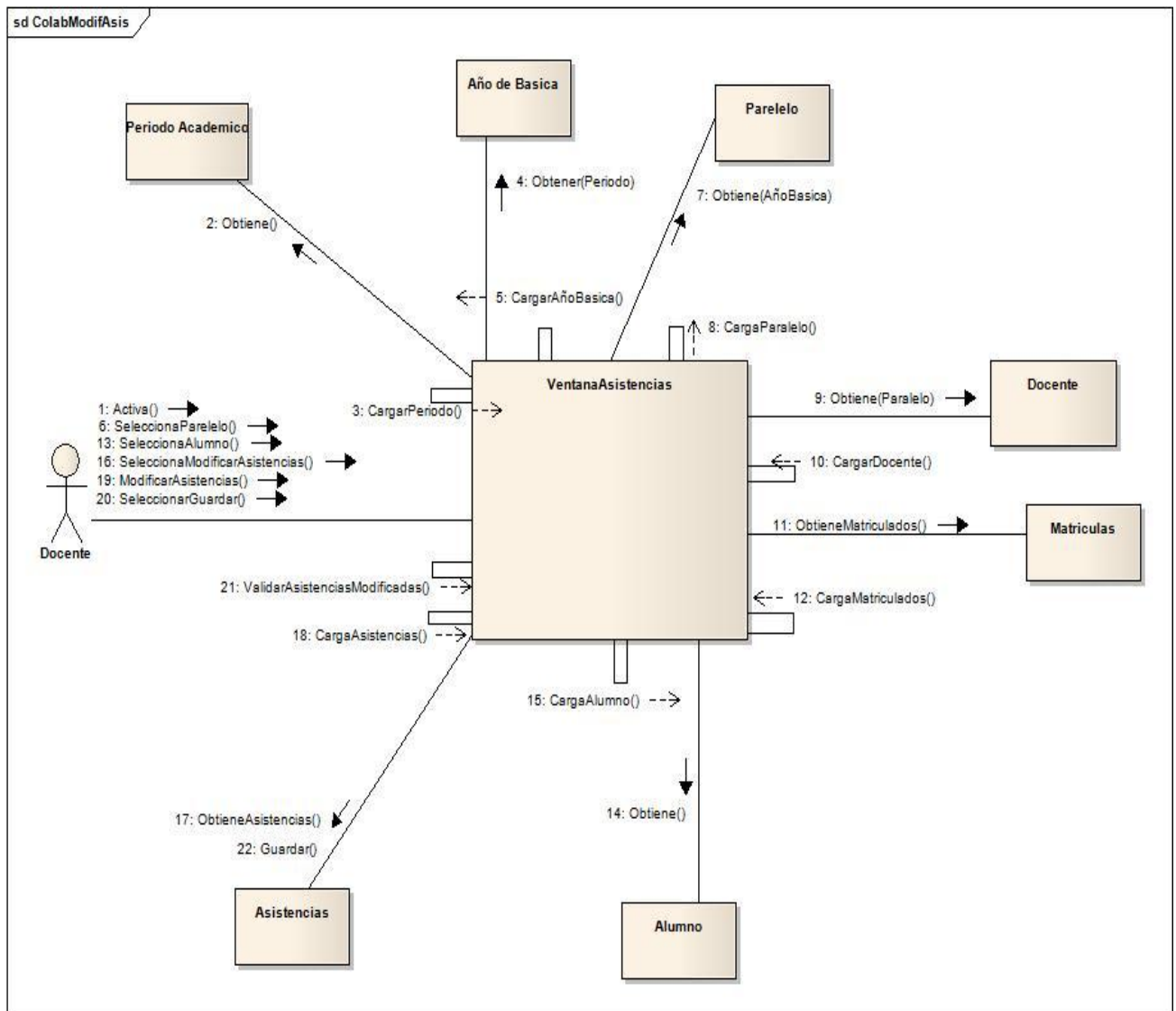


fig. 8.3.9.3.3

## 8.3.10 Caso de Uso: Registrar Acreditaciones



**Opciones**

Acreditación

**Proceso de Acreditación**

**DATOS ESTUDIANTE**

Cedula: 1104896202

Nombre: Mayra

Apellido: Villa

Año Basica:

Paralelo:

Año Lectivo:

Ciclo de Estudio: 1 Estado: ☒ Estado

**ACREDITACION**

**Materia** **Nota**

Disciplina:

**GUARDAR** **CANCELAR** **MODIFICAR**

fig. 8.3.10

<b>Caso de uso 009</b>	Registrar Acreditaciones
<b>Actores</b>	Secretaria, Docente (iniciador)
<b>Propósito</b>	Registrar las notas trimestralmente en el sistema que los estudiantes han obtenido en la ventana Acreditaciones para poder entregar los reportes de aprovechamiento y otros fines
<b>Resumen</b>	El docente llega a la dirección para entregar las notas trimestrales de sus alumnos a la secretaria y ella se encarga de ingresarlas al sistema entrando a la ventana Acreditaciones la cual permite buscar un paralelo y escoger un trimestre.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R3.1, R3.2, R3.4

Tabla. 8.3.10

### 8.3.10.1 CURSO NORMAL DE EVENTOS

Acción del actor	Respuesta del sistema
1. Empieza cuando el docente lleva las acreditaciones trimestrales de los alumnos	
2. El docente entrega ese informe a la secretaria ir a la sección Agregar acreditaciones.	
3. Si ocurrió una equivocación en las acreditaciones y se desea modificar ir a la sección Modificar acreditaciones	
4. La secretaria dentro de la ventana Acreditaciones presiona el botón calcular	5. El sistema calcula el promedio trimestral y al finalizar el año el promedio final.

Cursos alternos

### 8.3.10.2 Sección: Agregar Acreditación

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria va al menú acreditar al ítem Acreditación por paralelo y hace clic en el ítem	2. Presenta la ventana Acreditación 3. Retorna desde la base de datos el año lectivo actual a la caja de texto, los años de básica a la lista despegable dependiendo del año lectivo y los trimestres a la otra lista despegable.
4. La secretaria escoge un año de básica.	5. Busca y retorna desde la base de datos los paralelos dependiendo del año de básica escogido.
6. La secretaria escoge un paralelo y un trimestre	7. Muestra en la tabla el número y el nombre de los alumnos del paralelo correspondiente.

	8. Muestra en la caja de texto el docente a cargo del paralelo.
9. La secretaria ingresa las acreditaciones alumno por alumno y presiona el botón guardar	10. El sistema almacena las acreditaciones y las agrega a cada alumno.

### 8.3.10.2.1 Diagrama de robustez

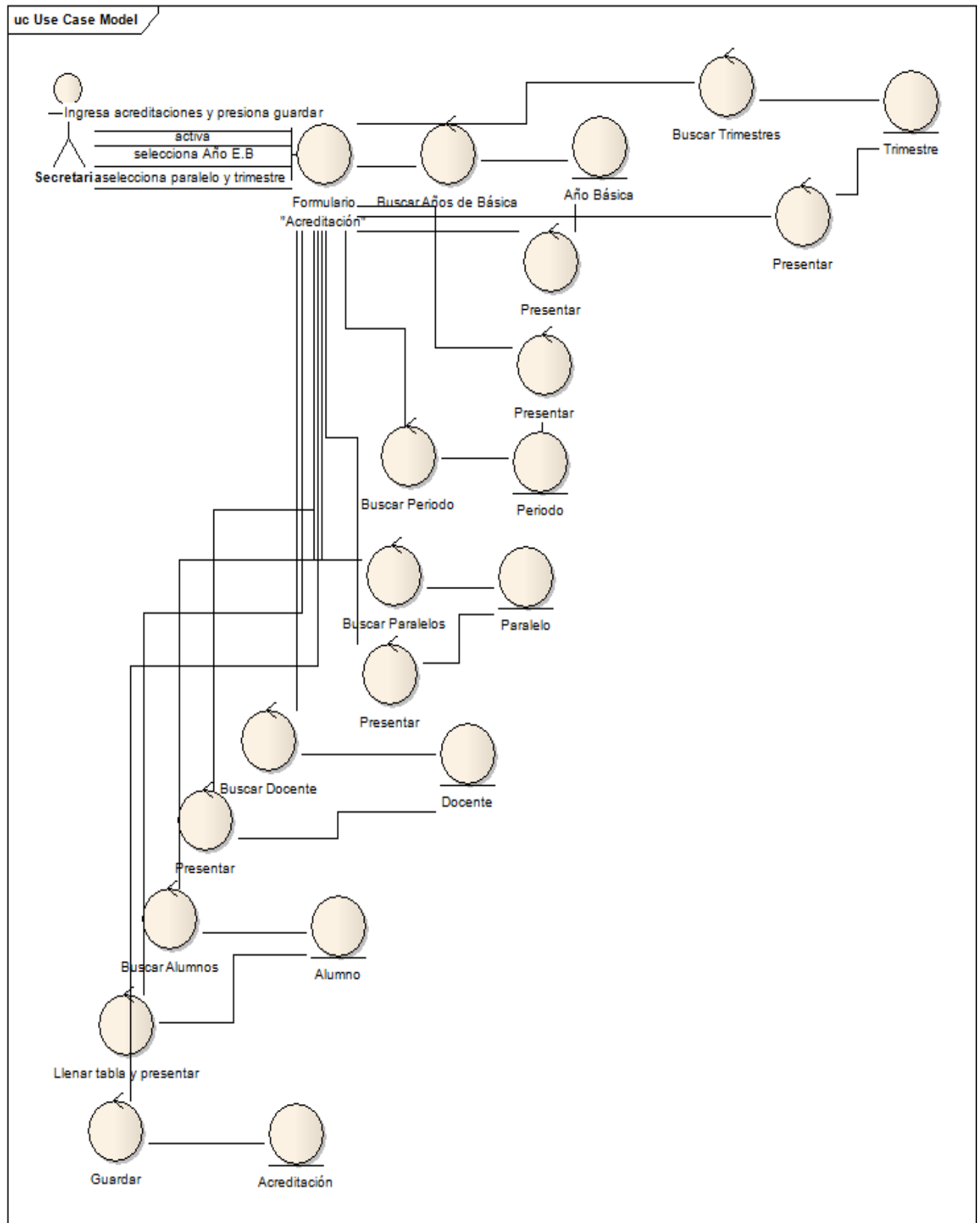


Fig. 8.3.10.2.1



### 8.3.10.2.2 Diagrama de secuencia

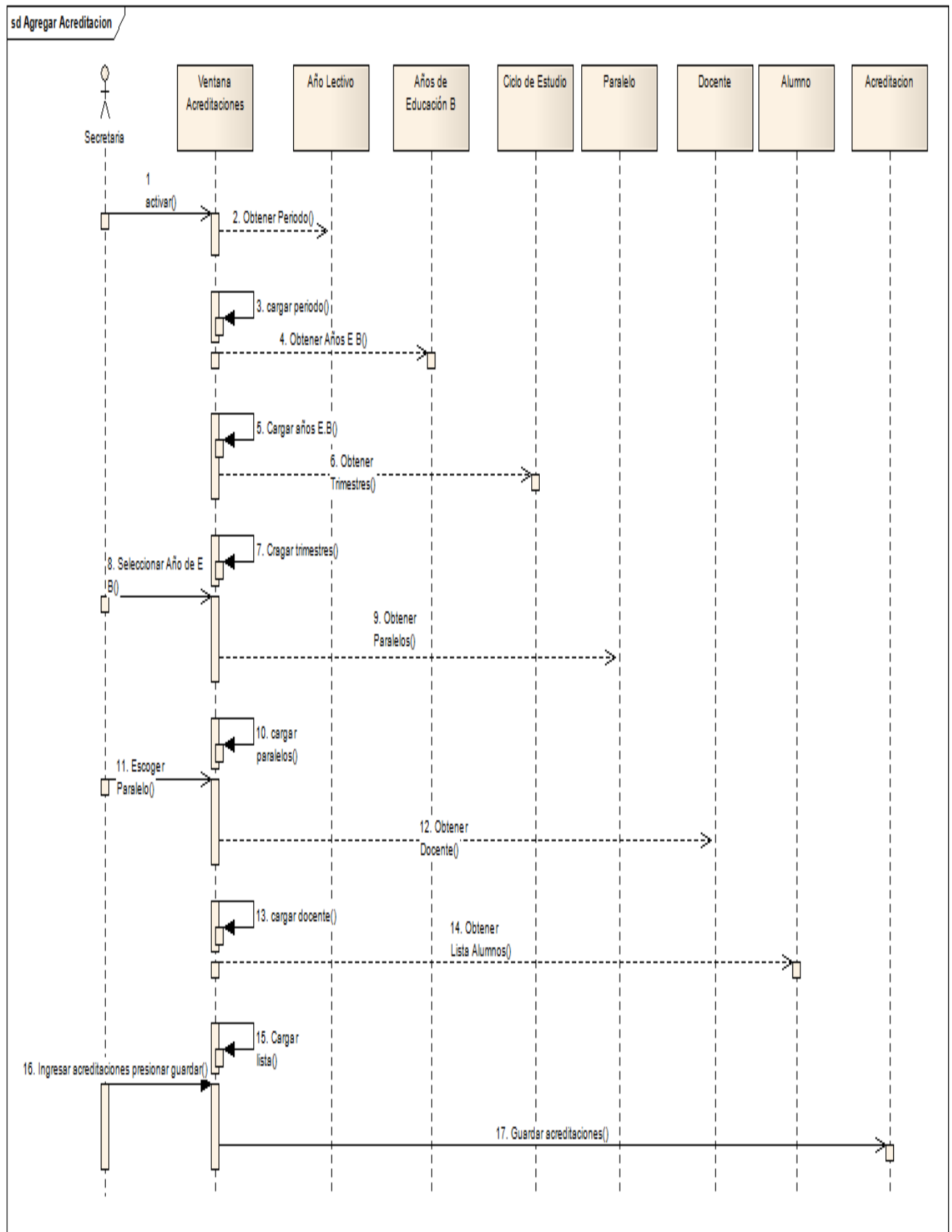


Fig. 8.3.10.2.2

## 8.3.10.2.3 Diagrama de colaboración

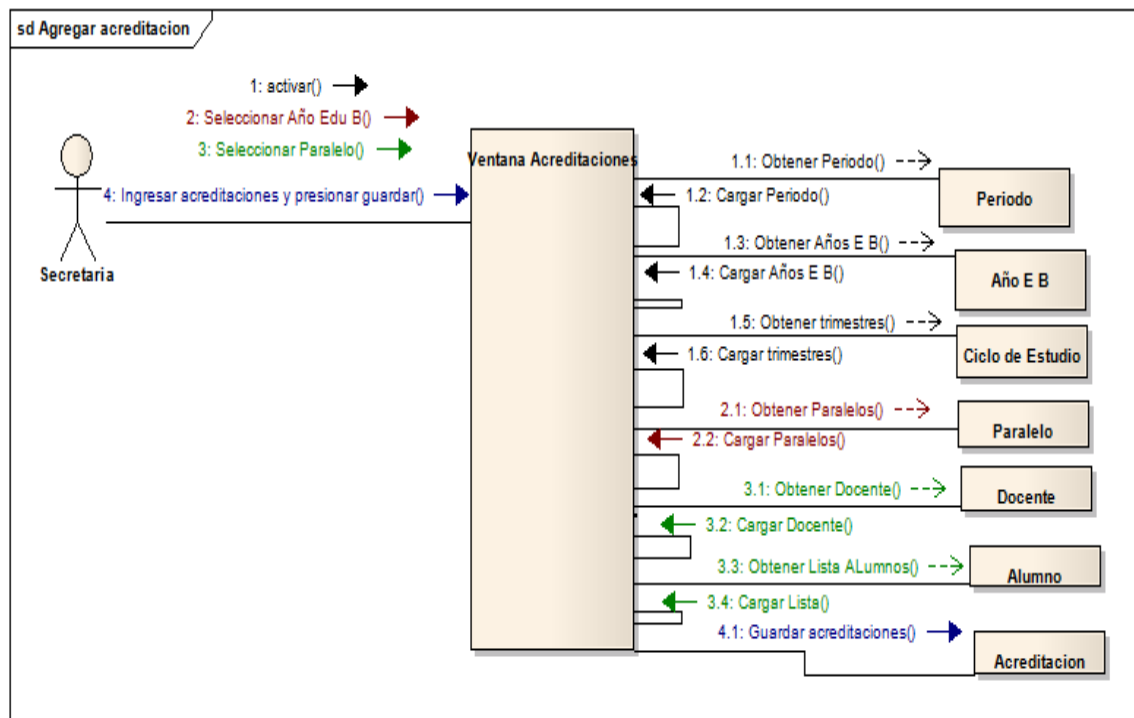


Fig. 8.3.10.2.3

## 8.3.10.3 Sección: Modificar Acreditación

Acción del actor	Respuesta del sistema
1. Empieza cuando la secretaria va al menú acreditar al ítem Acreditación por paralelo y hace clic en el ítem.	2. Presenta la ventana Acreditación. 3. Retorna desde la base de datos el año lectivo actual a la caja de texto, los años de básica a la lista despegable dependiendo del año lectivo y los trimestres a la otra lista despegable.
4. La secretaria escoge un año de básica.	5. Busca y retorna desde la base de datos los paralelos dependiendo del año de básica escogido.
6. La secretaria escoge un paralelo y un trimestre	7. Muestra en la tabla el número, el nombre de los alumnos y los datos

	de las acreditaciones ingresadas.
	8. En la una caja de texto el docente a cargo del paralelo
9. La secretaria hace clic en modificar.	10. Permite que se editen los campos en la tabla.
11. La secretaria ingresa los nuevos datos en los campos y hace clic en guardar.	12. Actualizará y guardará las nuevas calificaciones de los alumnos que han sido modificados.

Tabla 8.3.10.3

### 8.3.10.3.1 Diagrama de robustez

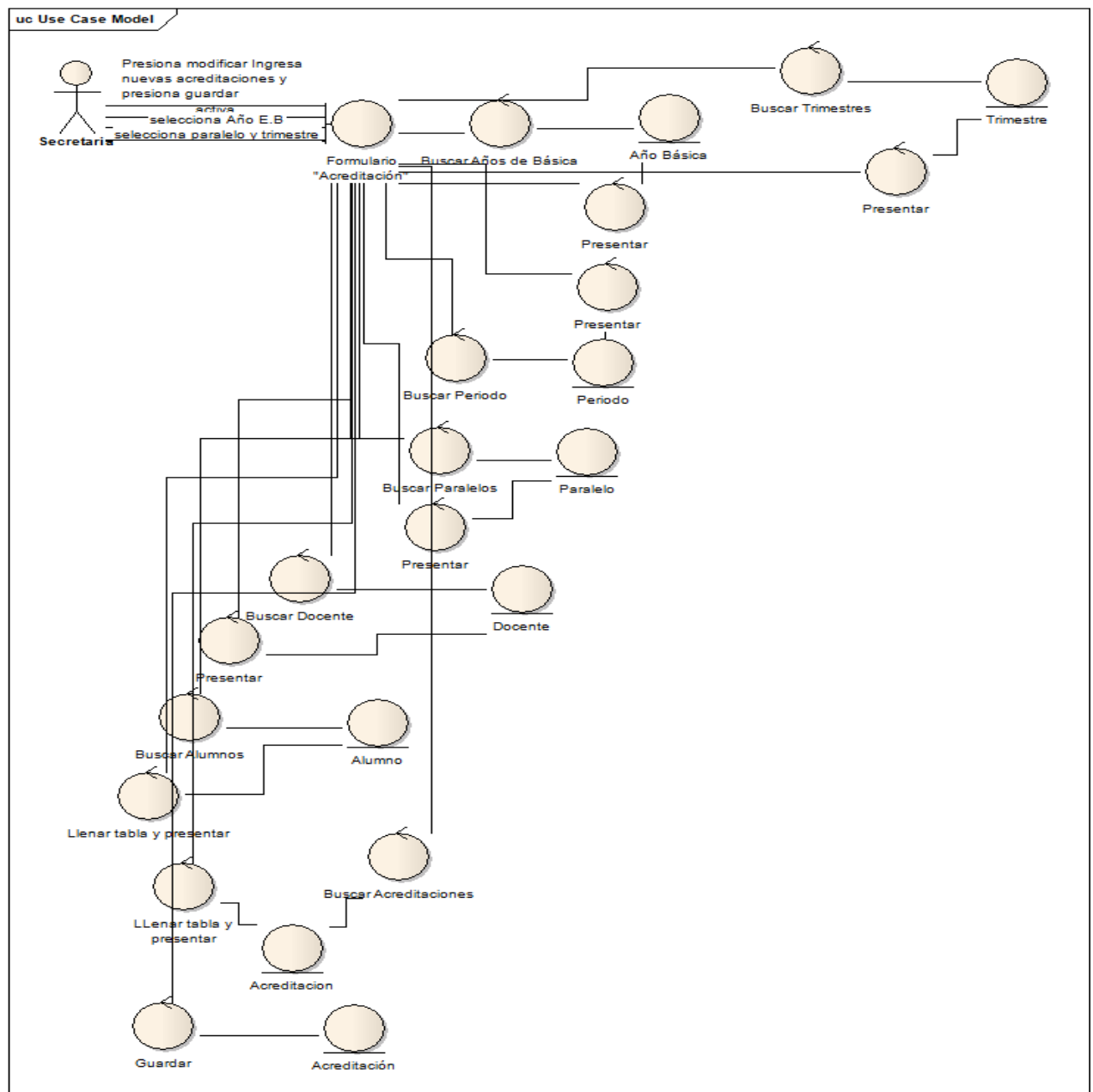


Fig. 8.3.10.3.1

## 8.3.10.3.1 Diagrama de secuencia

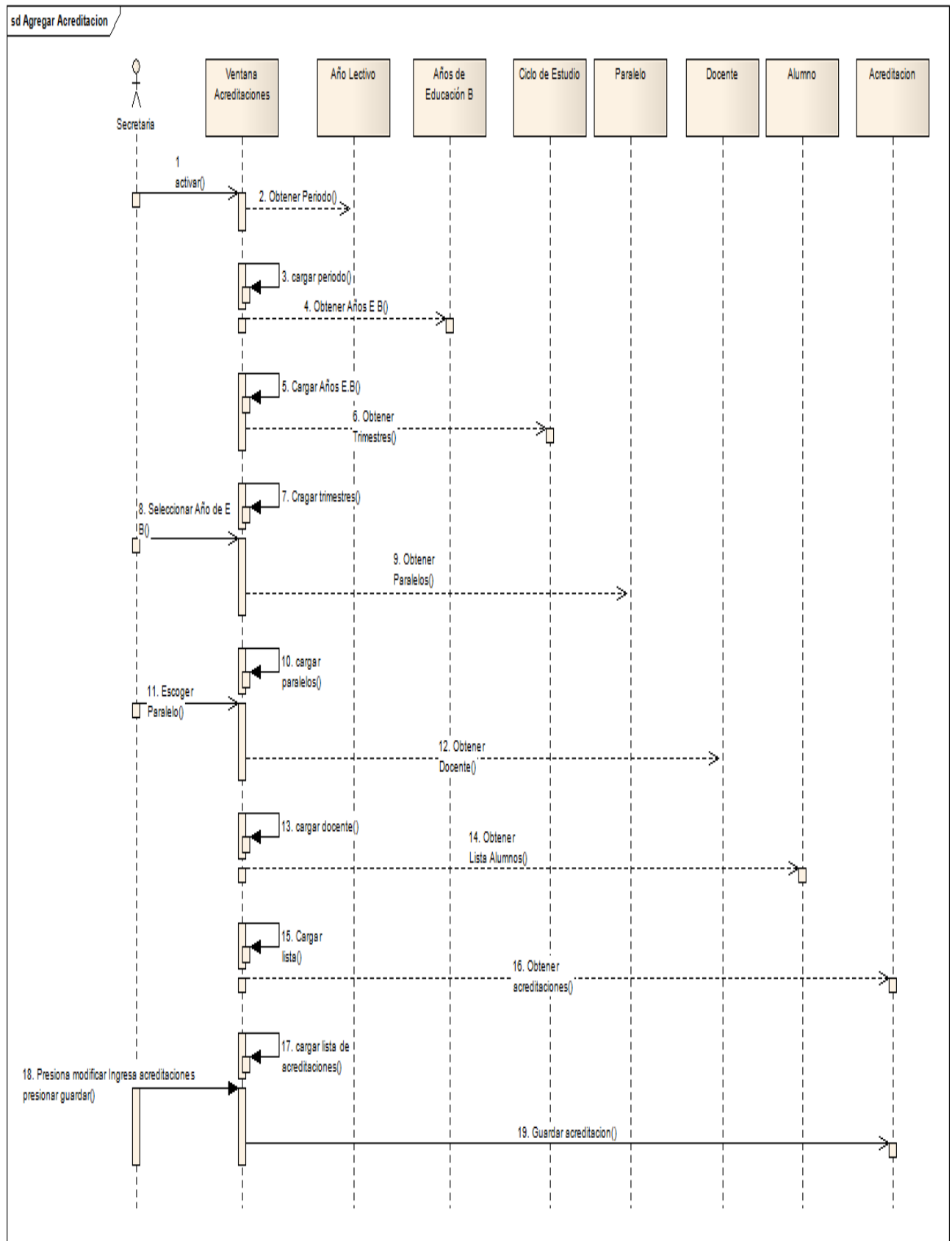


Fig. 8.3.10.3.1

## 8.3.10.3.2 Diagrama de colaboración

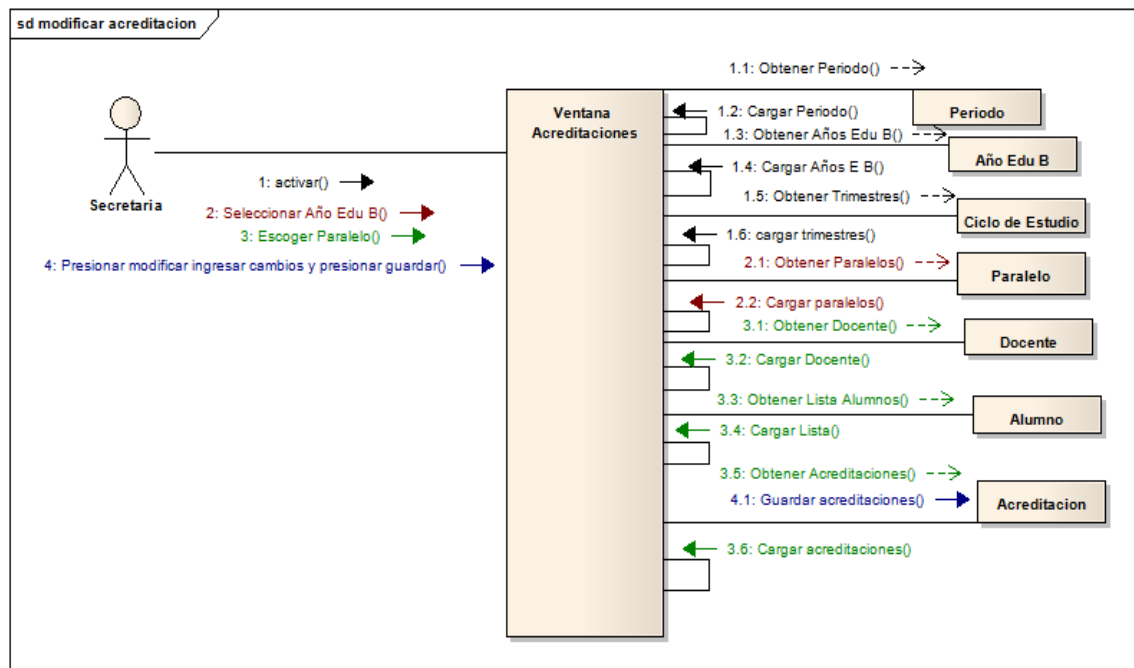


Fig. 8.3.10.3.2

## 8.3.11 Caso de Uso: Generar Reportes



Fig. 8.3.11

### 8.3.11.2 Curso Normal de Eventos

Acción del actor	Respuesta del Sistema
1. Comienza cuando el representante, la Directora o el Docente piden que se les genere un reporte.	
2. Si el Representante pide un reporte de matrícula ir a la Sección Generar certificado de matrícula.	3. El sistema presenta el menú reportes y el ítem certificado de matrícula.
4. Si el Representante pide un reporte acerca de la fecha de culminación Ir a la Sección Generar reporte de fecha de culminación de escuela	5. El Sistema presenta el menú reportes el ítem reportes ex alumnos y el ítem Fecha de Culminación.
6. Si hay que entregar notas trimestrales al padre de familia Ir a la Sección Generar reporte de calificaciones.	7. El sistema en la ventana principal presenta el menú y el ítem acreditaciones
8. Si un padre de familia pide las acreditaciones de séptimo de un ex alumno Ir a la sección Generar informe de acreditaciones de séptimo.	9. El Sistema presenta el menú reportes el ítem reportes ex alumnos y el ítem reporte acreditaciones.
10. Si la Directora pide un reporte de notas Ir a la Sección Generar Listado de Notas finales	11. El sistema presenta el menú reportes y el ítem Notas finales.
12. Si el Docente pide un listado de alumnos Ir a la Sección Generar lista de Alumnos.	13. El sistema presenta el menú reportes y el ítem Lista de alumnos.
14. Si el Docente pide un listado de padres de familia Ir a la Sección Generar lista de Padres de Familia	15. El sistema presenta el menú reportes y el ítem Lista de Representantes.

**Tabla 8.3.11.2**

### 8.3.11.2 Sección: Generar Certificado de matrícula

Fig. 8.3.11.2 A

Fig. 8.3.11.2 B

Acción del actor	Respuesta del Sistema
1. Empieza cuando la secretaria va al menú reportes al ítem certificado de matrícula.	2. El sistema muestra la ventana buscar alumno
3. Ir a la sección Buscar alumno	4. El Sistema retorna desde la base de

	datos el nombre el año en el que está matriculado los días asistidos y el año lectivo actual sobre el alumno a los espacios en blanco.
5. La secretaria ingresa la fecha actual y hace clic en el botón imprimir.	6. El sistema imprime el certificado.

**Tabla. 8.3.11.2**

Cursos alternos:

- Item 4: Si el alumno no consta el sistema presentara un mensaje de error.

#### 8.3.11.2.1 Diagrama de robustez

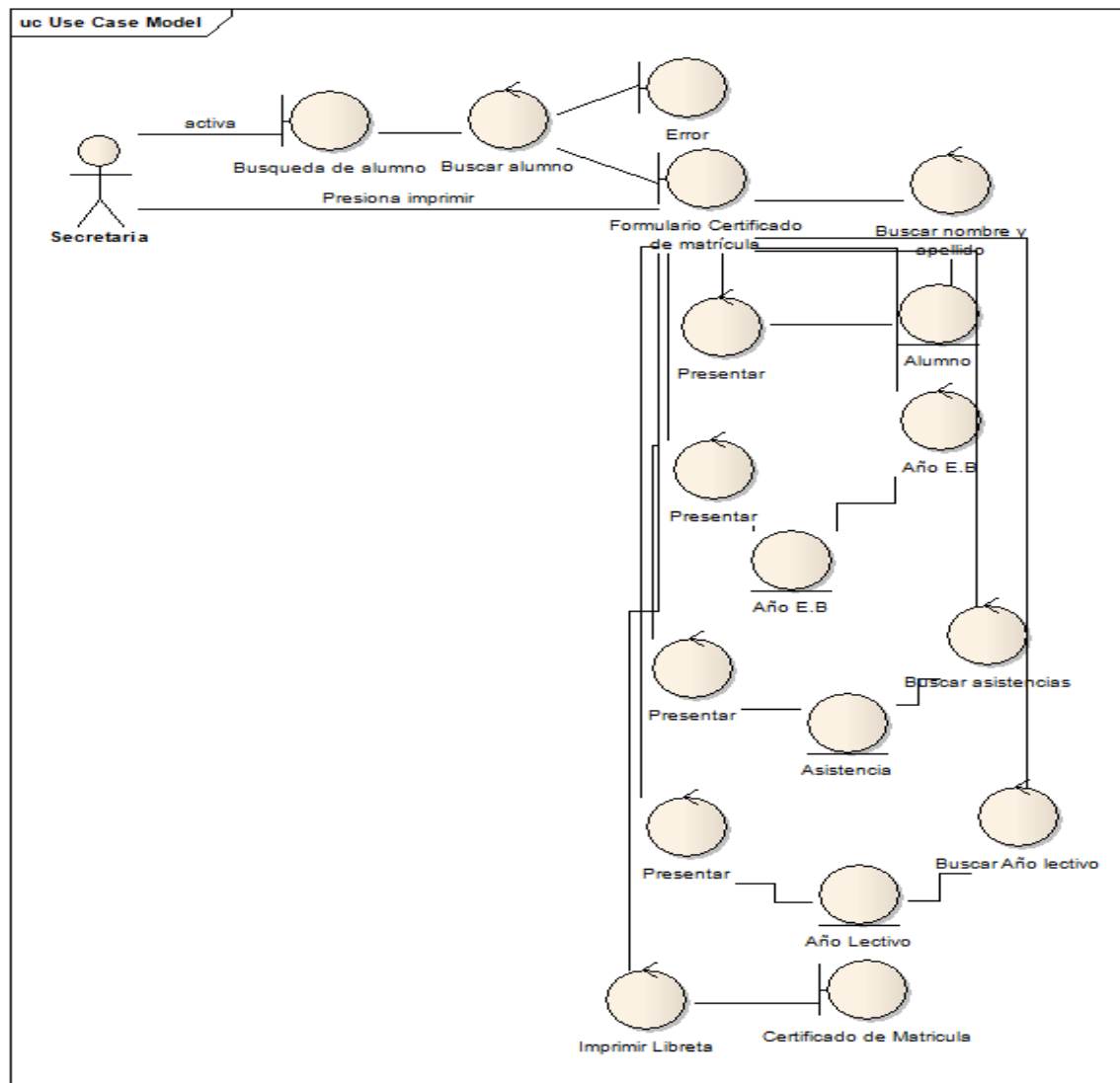
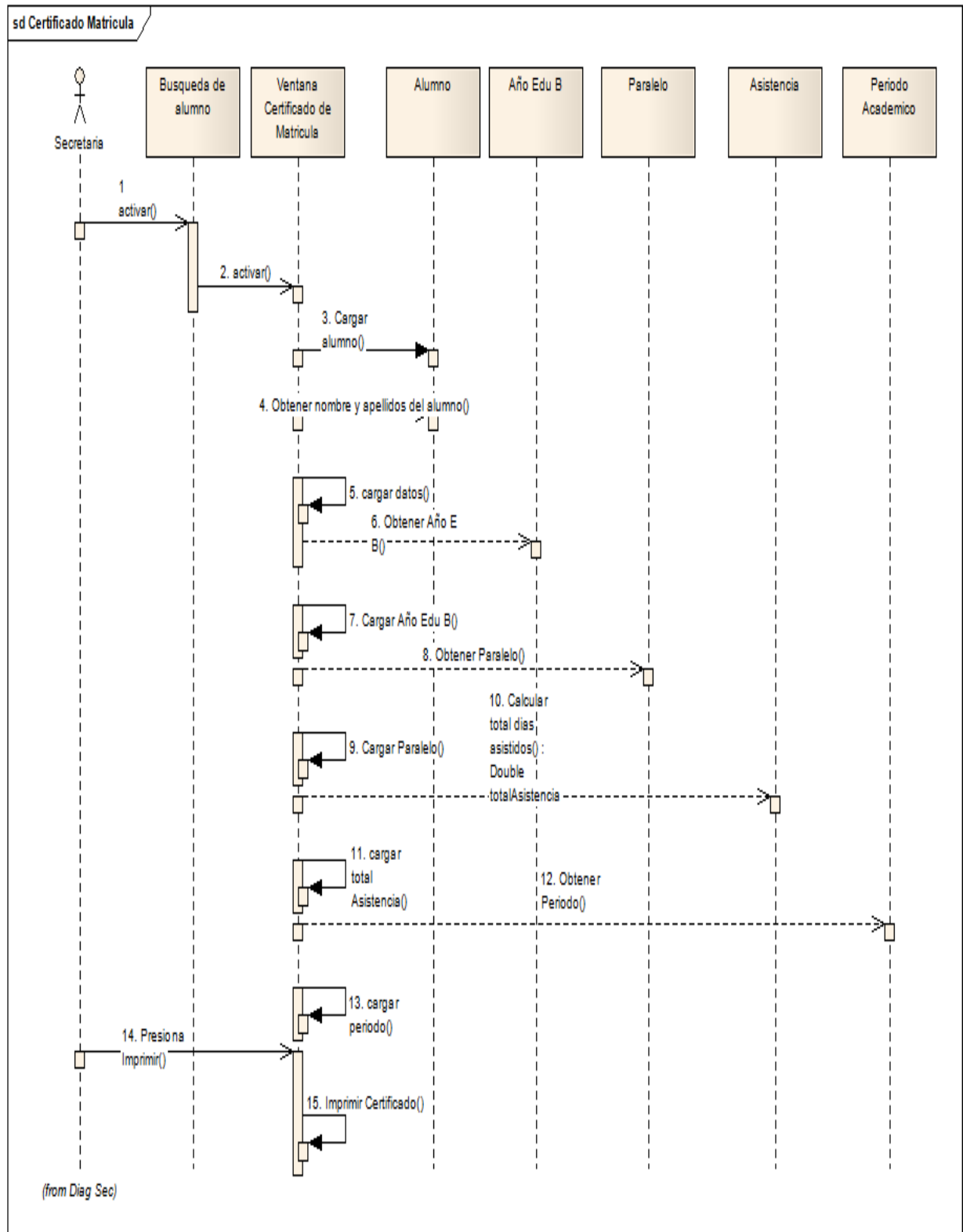




Fig. 8.3.11.2.1

## 8.3.11.2.2 Diagrama de secuencia



**Fig. 8.3.11.2.2**

### 8.3.11.2.3 Diagrama de colaboración

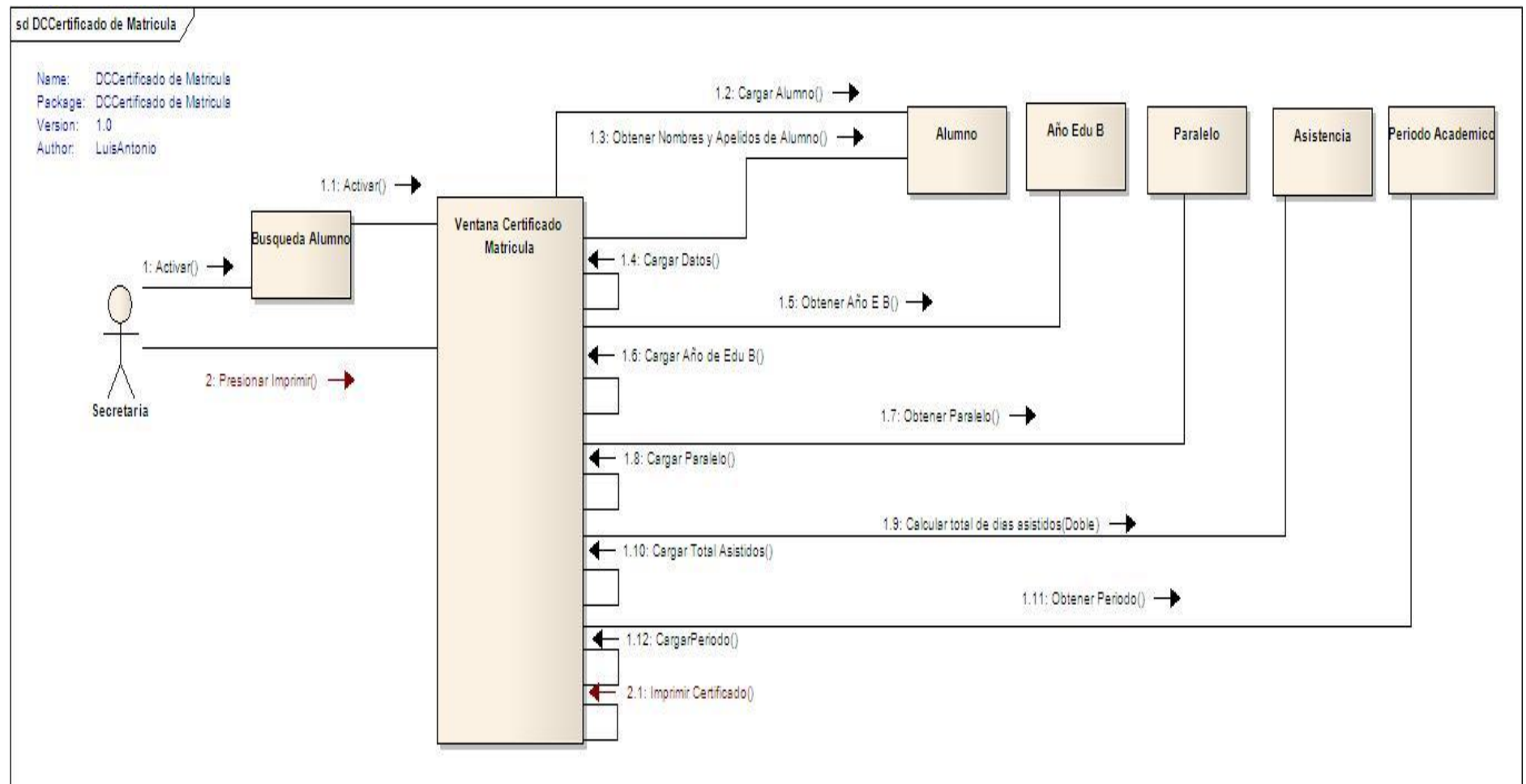



Fig. 8.3.11.2.3

### 8.3.11.3 Sección: Generar informe de acreditaciones



**Escuela Fiscal Mixta Julio Maria Matovelle**

LIBRETA

Materia	1er Tri
Lenguaje y Comunicación	
Matemáticas	
Ciencias Naturales	
Estudios Sociales	
Cultura Física	

Disciplina

SUMAN

PROMEDIO

Total de Dias Asistidos  Faltas Justificadas

Total de Dias Faltados  Faltas Injustificadas

20-19 Sobresaliente  
18-15 Muy Buena  
15-14 Buena  
13-12 Regular  
11-0 Insuficiente

DIRECTORA  DIRIGENTE

**Fig. 8.3.11.3**

Acción del actor	Respuesta del Sistema
1. Empieza cuando la secretaria va a la ventana Reportes al ítem Libreta de calificaciones hace clic en el ítem.	
2. Procede a buscar un alumno Ir a la sección buscar alumno	3. El Sistema muestra la ventana Libreta de calificaciones. 4. El Sistema retorna de la base de datos a la ventana Libreta de calificaciones el año lectivo actual el número de matrícula y el nombre del alumno el año de

	básica y el paralelo en el que está
5. La secretaria escoge el trimestre para que se muestre la libreta.	6. Presenta las calificaciones y las asistencias en la tabla y en los espacios en blanco los promedios totales y las asistencias totales.
7. La secretaria hace clic en imprimir	8. El sistema imprime la libreta.

**Tabla 8.3.11.3**

Cursos alternos:

- Item 4: Si el alumno no consta el sistema presentara un mensaje de error.

### 8.3.11.3 Diagrama de robustez

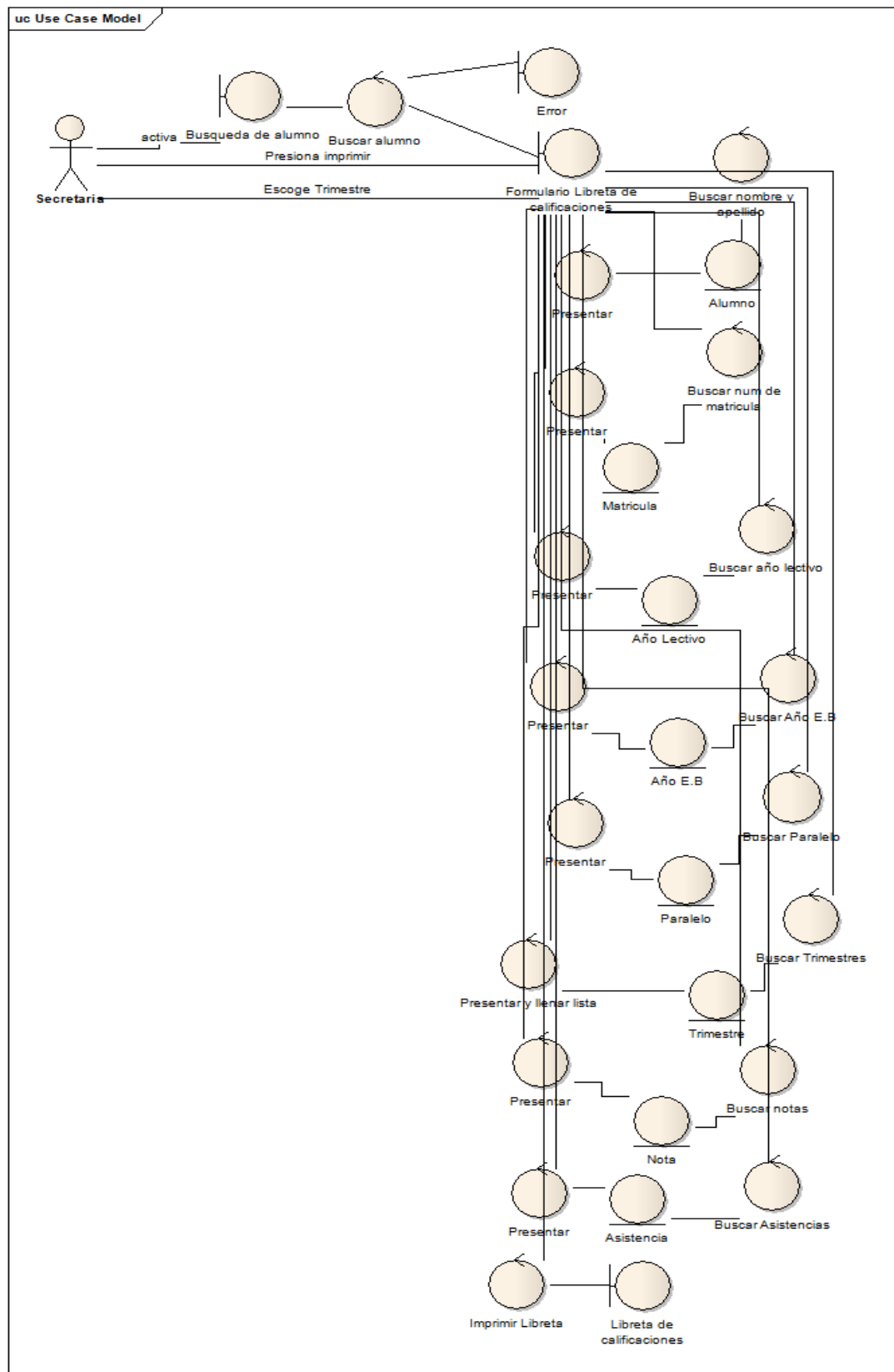


Fig. 8.3.11.3

### 8.3.11.3.3 Diagrama de secuencia

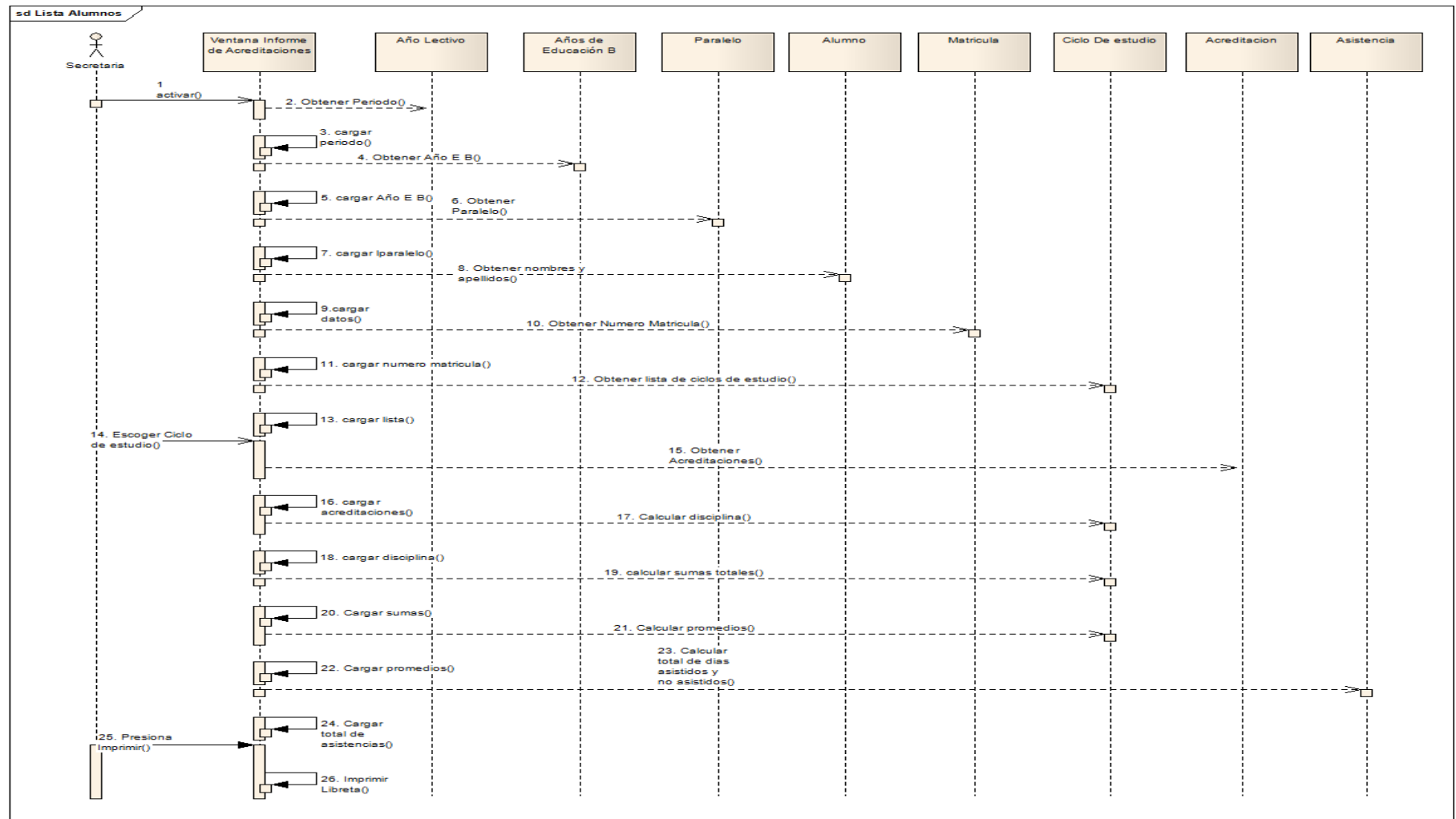


Fig. 8.3.11.3.3

### 8.3.11.3.3 Diagrama de colaboración

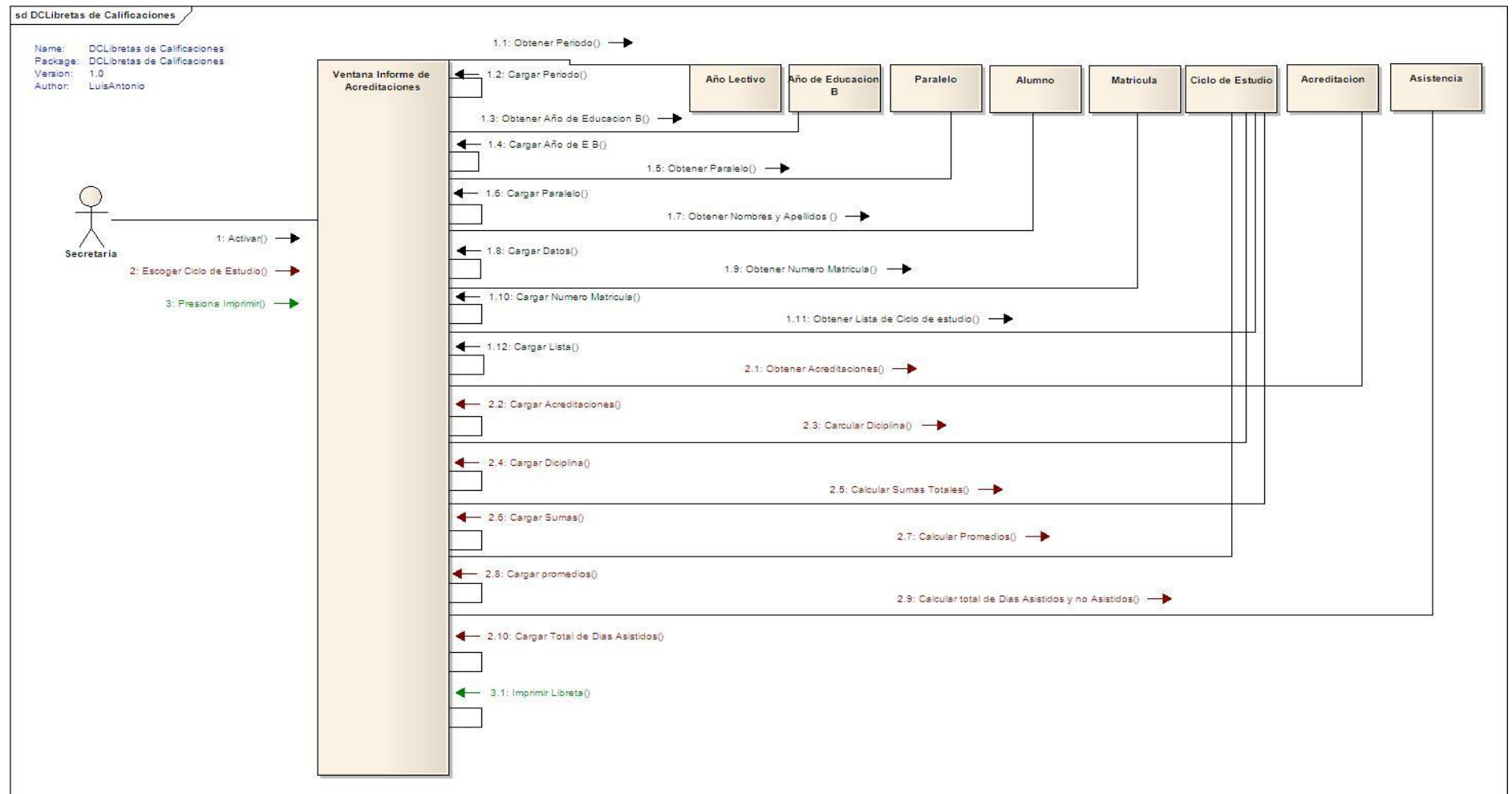


Fig. 8.3.11.3.3



### 8.3.11.3 Sección: Generar Lista de notas finales

**Lista Alumnos con Notas Finales**

**Escuela Fiscal Mixta Julio Maria Matovelle**

Año:  Paralelo:

Docente:

Año Lectivo:  Decendente:

Nº	Nombre	Nota	Disciplina
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			

**IMPRIMIR** **SALIR**

Fig. 8.3.11.3

Acción del actor	Respuesta del Sistema
1. Empieza cuando la secretaria va a la ventana reportes al ítem notas finales.	2. El Sistema retorna desde la base de datos el año lectivo actual y con los años de básica correspondientes. 3. Presenta dos opciones para ordenar las notas ascendente o descendente.
4. La secretaria escoge un año	5. El Sistema retorna de la base de

	datos los paralelos correspondiente al año de básica escogido
6. La secretaria escoge un paralelo	7. El sistema retorna desde la base de datos el docente a cargo.
8. La secretaria escoge una opción de orden.	9. El Sistema ordena las notas y las presenta con el número y el nombre del alumno.
10. La secretaria pulsa el botón imprimir.	11. El sistema imprime la lista.

**Tabla. 8.3.11.3**



### 8.3.11.3.2 Diagrama de robustez

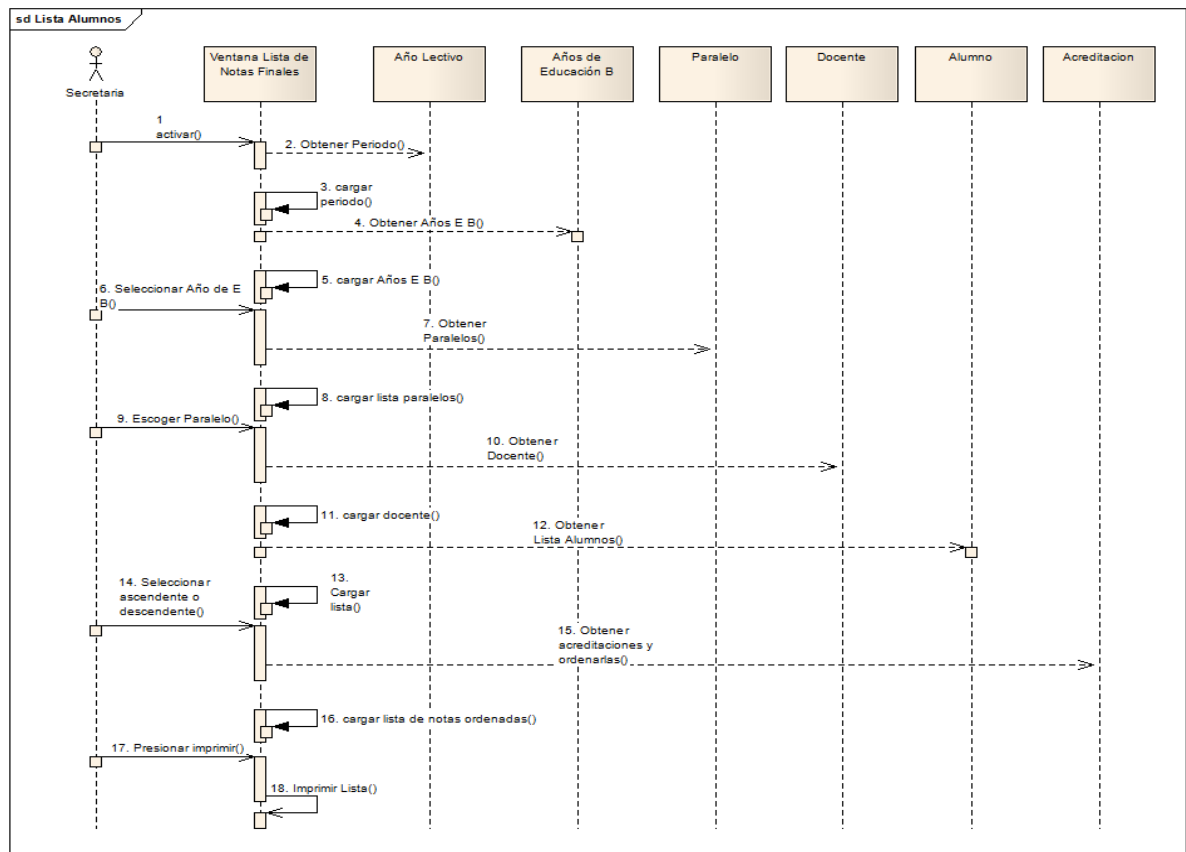


Fig. 8.3.11.3.2



**Fig. 8.3.11.4**

Acción del actor	Respuesta del Sistema
1. Empieza cuando la secretaria va a la ventana reportes al ítem lista de representantes	2. El Sistema muestra la ventana lista de alumnos. 3. Muestra el año lectivo actual y en la lista despegable los años retornados de la base de datos correspondiente al año lectivo.
4. La secretaria escoge un año de básica.	5. El Sistema retorna de la base de datos los paralelos correspondientes al año de básica.
6. La secretaria escoge un paralelo	7. El sistema retorna desde la base de datos el docente a cargo y la lista de alumnos con el número correspondiente.
8. La secretaria hace clic en el botón imprimir.	9. El sistema imprime la lista de alumnos.

**Tabla. 8.3.11.4**

### 8.3.11.4.1 Diagrama de robustez

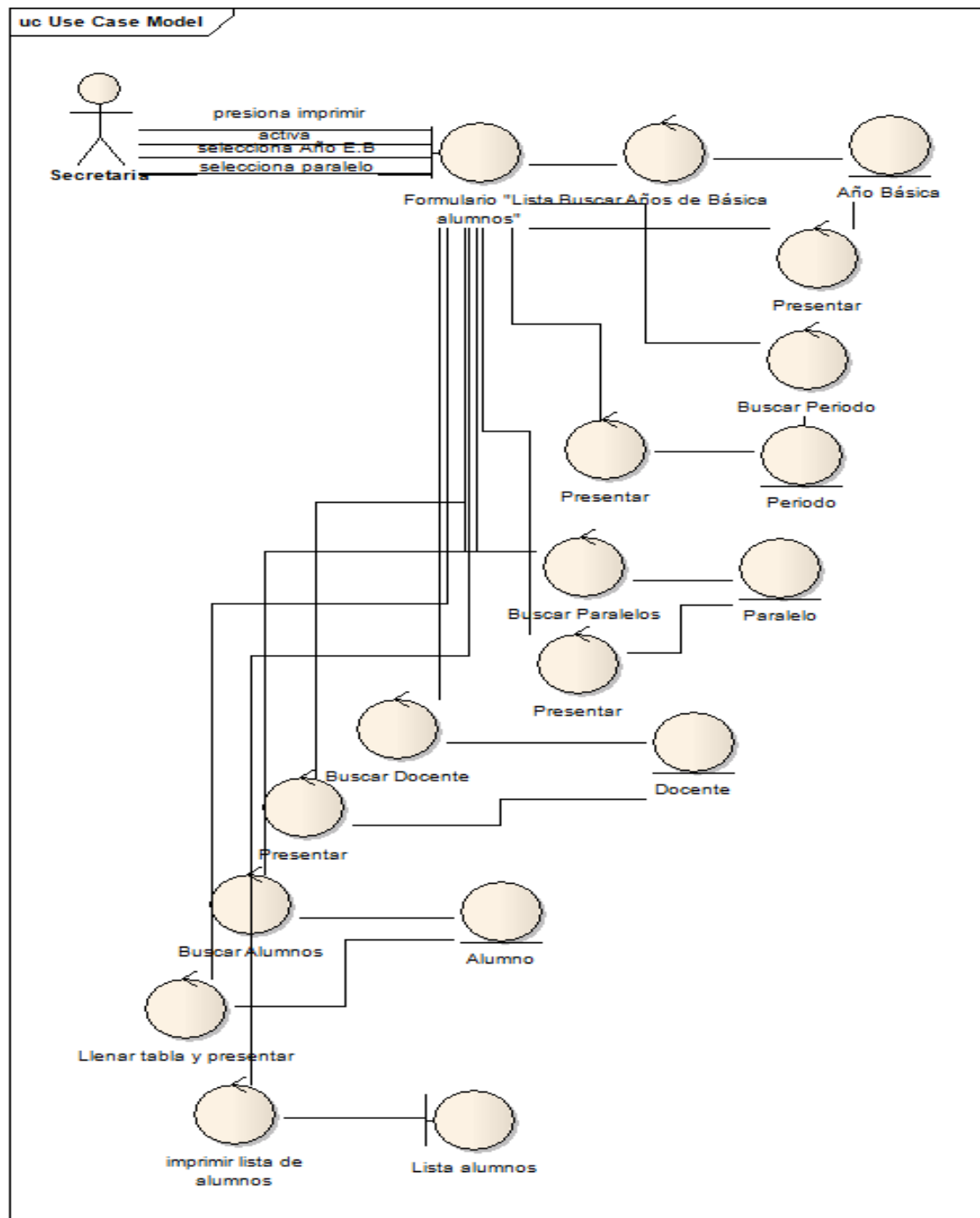


Fig. 8.3.11.4.1

### 8.3.11.4.2 Diagrama de secuencia

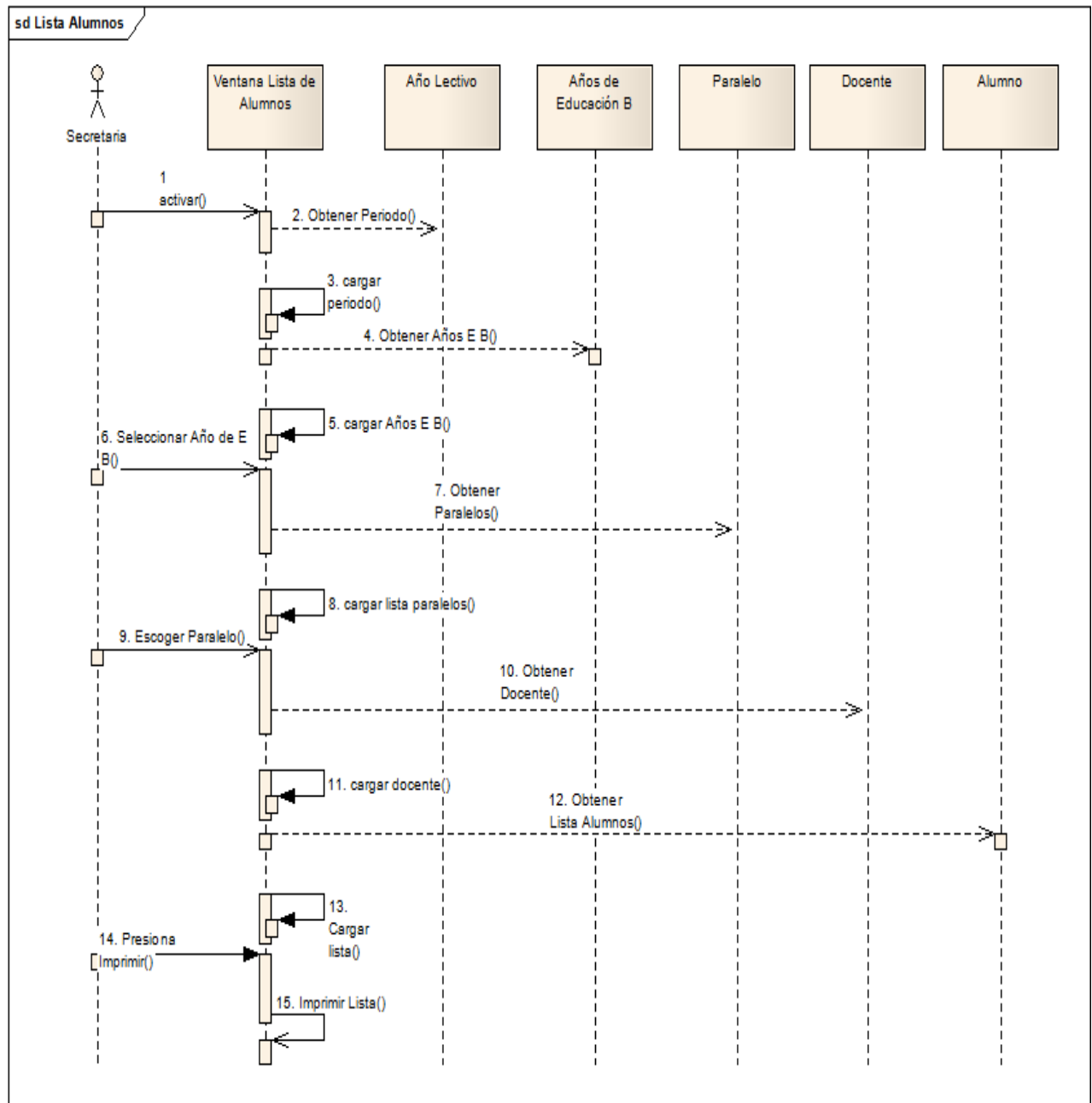


Fig. 8.3.11.4.2



### 8.3.11.4.3 Diagrama de colaboración

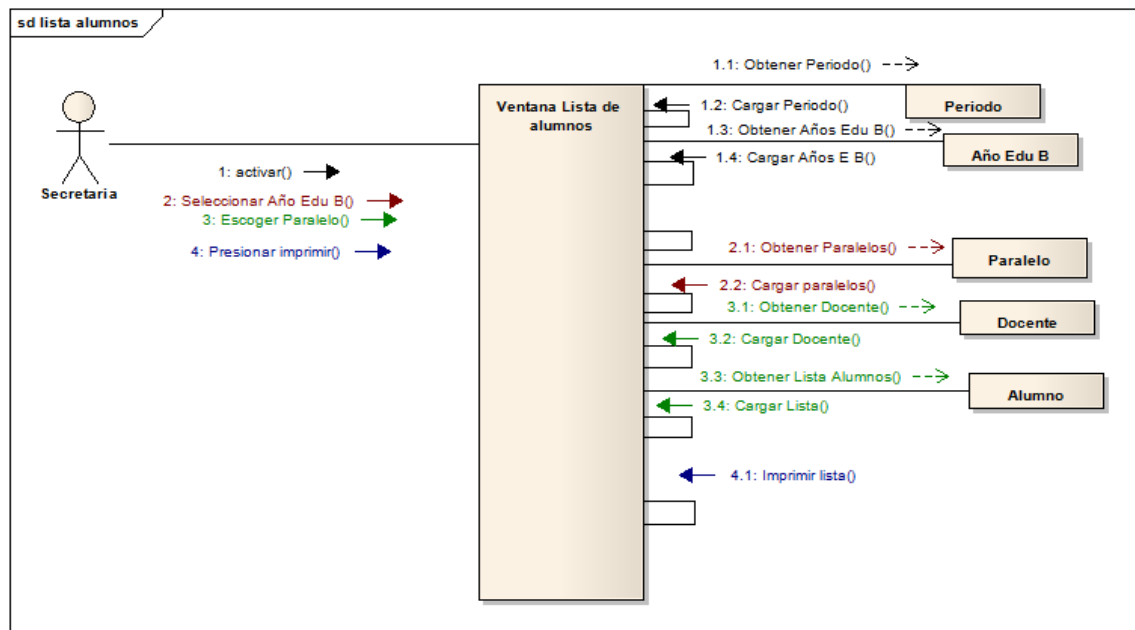


Fig. 8.3.11.4.3

### 8.3.12 Sección: Generar lista de representantes

Diseño previo [ListasRepresentantes]

**Escuela Fiscal Mixta Julio Maria Matovelle**

Año:  Paralelo:

Docente:

Año Lectivo:

Nº	Nombre			
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				

IMPRIMIR SALIR

Fig. 8.3.12

Acción del actor	Respuesta del Sistema
10. Empieza cuando la secretaria va a la ventana reportes al ítem lista de representantes	11. El Sistema muestra la ventana lista de representantes. 12. Muestra el año lectivo actual y en la lista despegable los años retornados de la base de datos correspondiente al año lectivo.
13. La secretaria escoge un año de básica.	14. El Sistema retorna de la base de datos los paralelos correspondientes al año de básica.
15. La secretaria escoge un paralelo	16. El sistema retorna desde la base de datos el docente a cargo y la lista de representantes con un número correspondiente.
17. La secretaria hace clic en el botón imprimir.	18. El sistema imprime la lista de representantes.

**Tabla 8.3.12**



### 8.3.11.5.2 Diagrama de secuencia

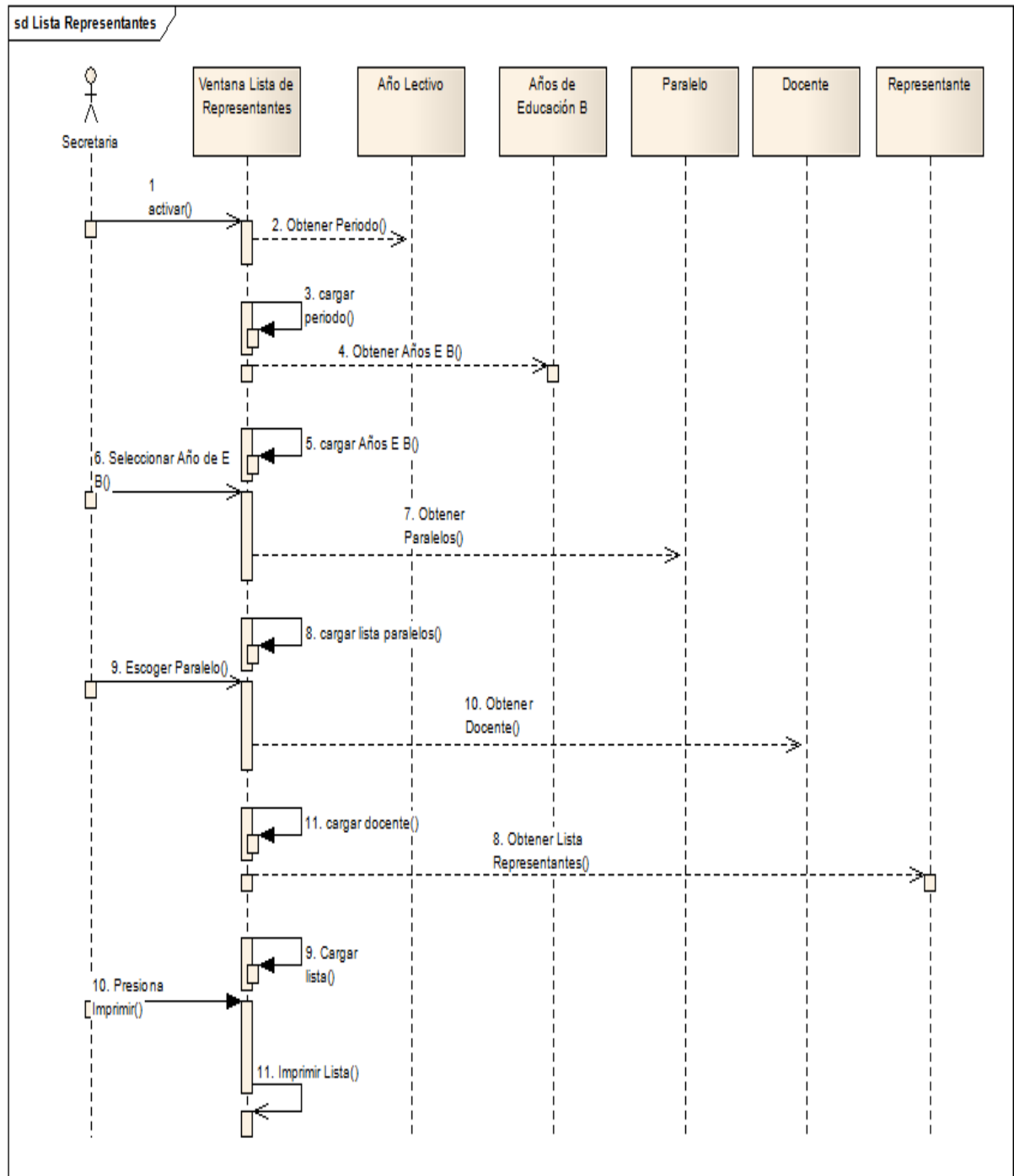


Fig. 8.3.11.5.2

### 8.3.12.3 Diagrama de colaboración

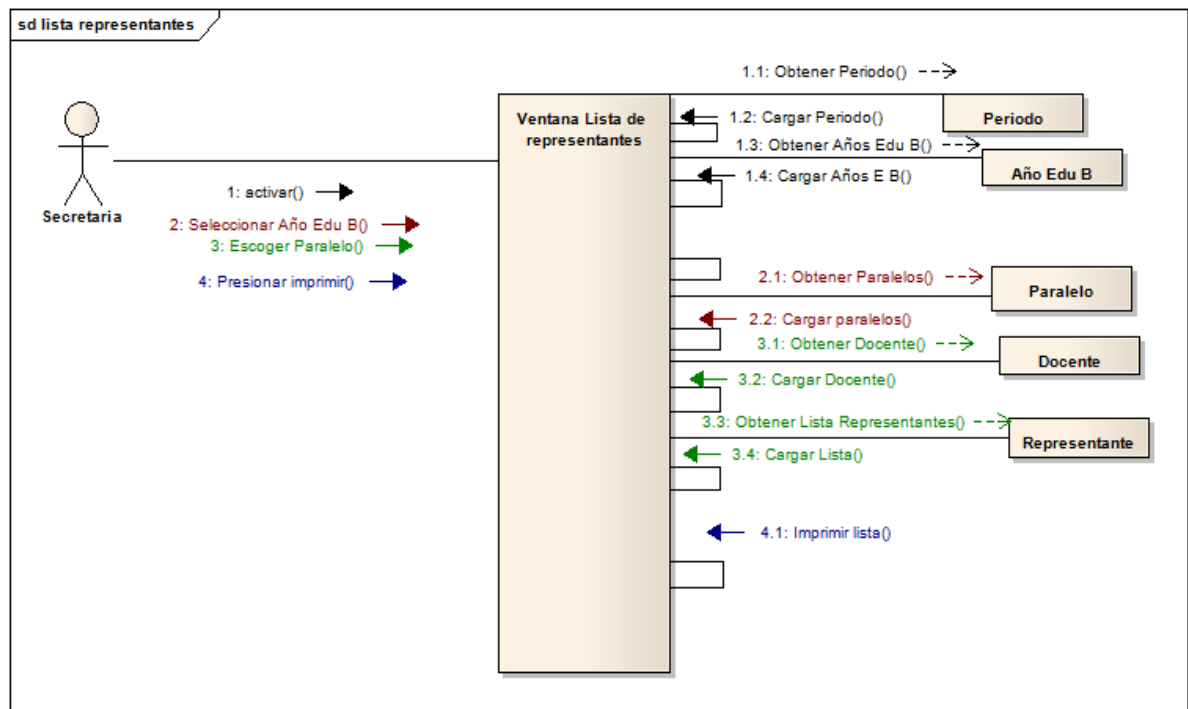


Fig. 8.3.12.3

### 8.3.11.5 Caso de Uso: Administrar Ciclo de Estudio



Fig. 8.3.11.5

<b>Caso de uso 003</b>	Administrar Ciclo de Estudio
<b>Actores</b>	Secretaria (iniciadora)
<b>Propósito</b>	Crear ciclos de estudio
<b>Resumen</b>	La Secretaria ingresa al sistema a la ventana administrar Ciclo de Estudio, ingresa, modifica, el número de ciclo de estudio.
<b>Tipo</b>	Primario y esencial
<b>Referencias cruzadas</b>	R 1.10

**Tabla 8.3.11.5**

### **CURSO NORMAL DE EVENTOS**

Acción del actor	Respuesta del sistema
8. Comienza cuando la secretaria desea agregar, modificar ciclos de estudio.	
9. La Directora va al menú administrar y selecciona el ítem Administrar Ciclo de Estudio.	10. El sistema presenta la ventana Administrar Ciclo de Estudio.
11. Si la secretaria desea agregar Ciclo de estudio ir a la sección Agregar Ciclo de Estudio	12. El sistema agrega ciclo de estudio.
13. Si la Secretaria desea modificar Ciclo de estudio va a la sección Modificar Ciclo de Estudio	14. El sistema modifica ciclo de estudio y actualiza.

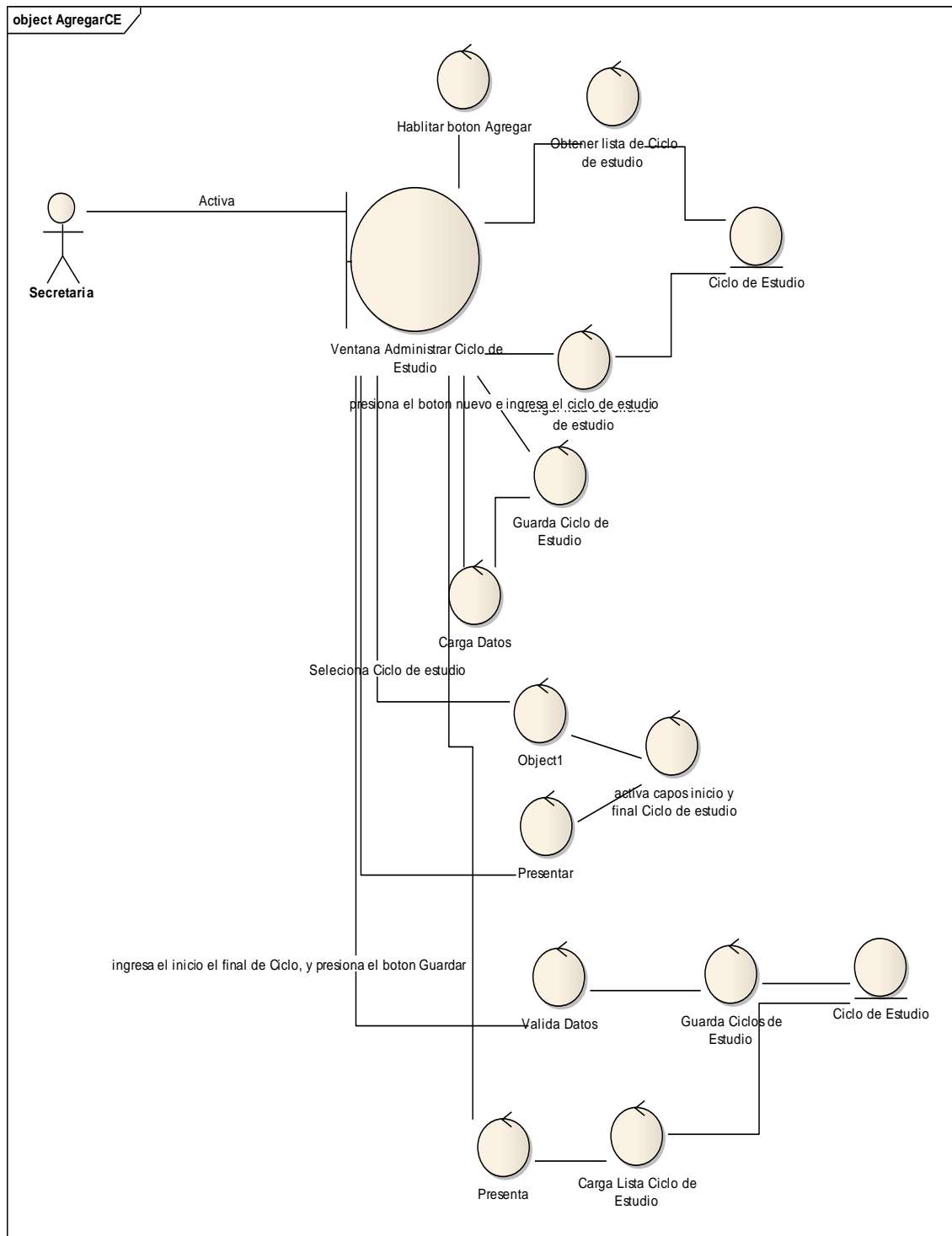
#### **8.3.1.12.1 Sección: Agregar Ciclo de Estudio**

Acción del actor	Respuesta del Sistema
1. Empieza cuando la Secretaria ingresa en la ventana Administrar	2. En la ventana Ciclo de estudio se cargan los Ciclos de estudio, y se

Ciclo de Estudio.	activa el botón nuevo.
3. La Secretaria presiona el botón nuevo, e ingresa el número de ciclos de estudio	4. El Sistema Carga un lista de ciclos de estudio.
5. Selecciona un ciclo de estudio	6. Activa el botón agregar
7. Presiona el botón agregar, ingresa el inicio, el final. Y presiona Aplicar.	8. El sistema valida los datos, los guarda los ciclos de estudio, 9. Carga la lista de ciclos de estudio.

**Tabla 8.3.1.12.1**

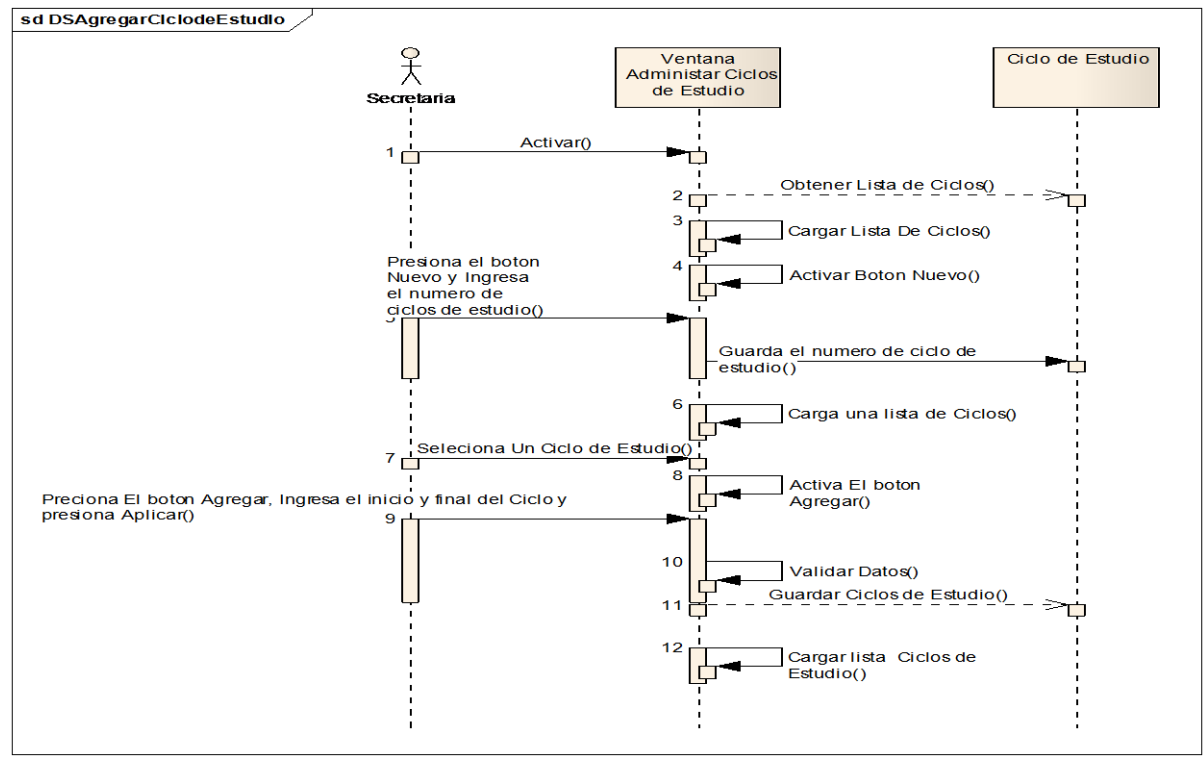
### 8.3.12.2.1 Diagrama de Robustez



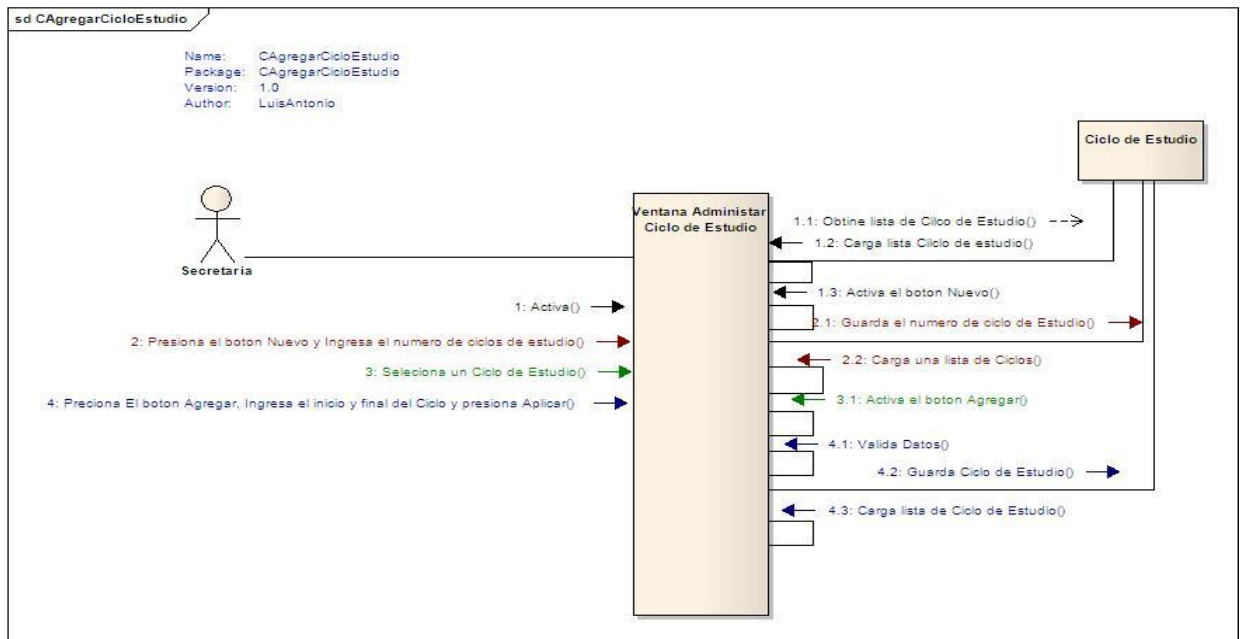
**Fig. 8.3.12.2.1**



### 8.3.12.2.2 Diagrama de secuencia



### 8.3.12.2.3 Diagrama de Colaboración

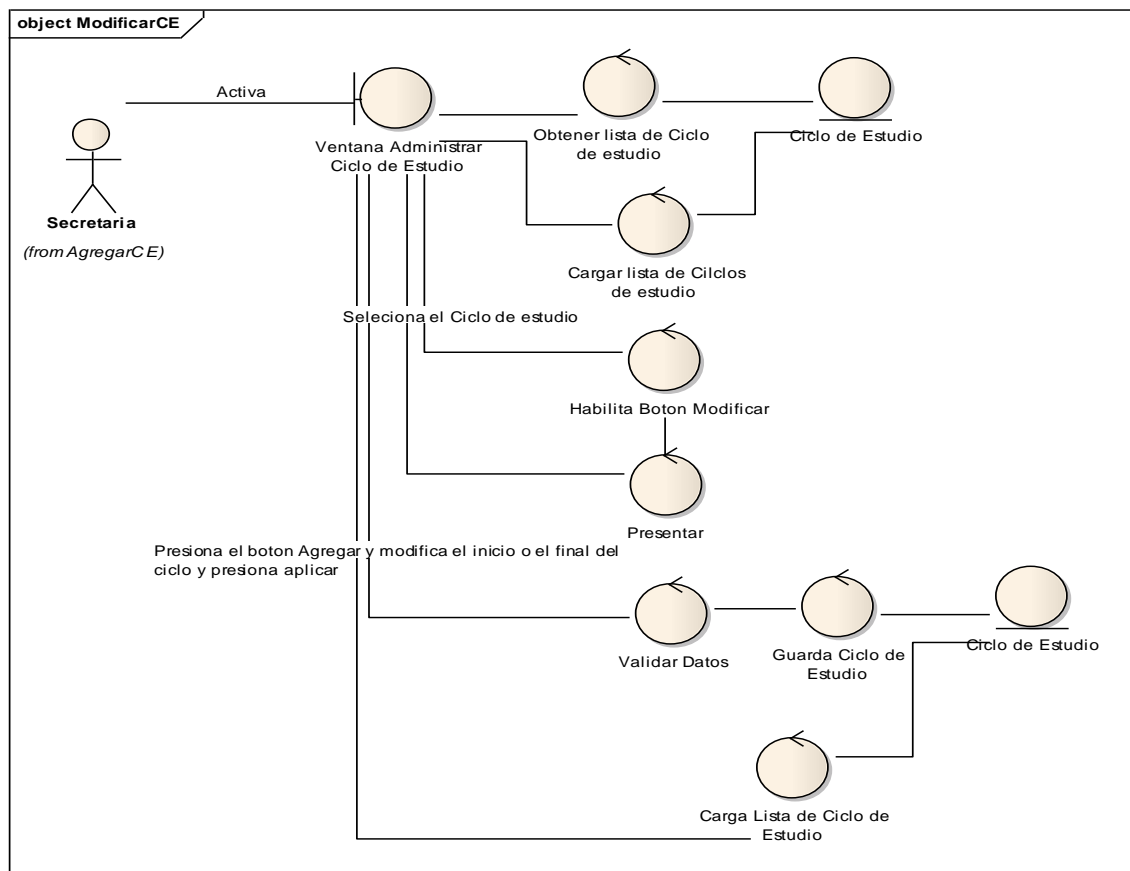


### 8.3.12.3 Sección: Modificar Ciclo de Estudio

Acción del actor	Respuesta del Sistema
1. Empieza cuando la Secretaria ingresa en la ventana Administrar Ciclo de Estudio.	2. En la ventana Ciclo de estudio se cargan los Ciclos de estudio, y se activa el botón nuevo.
3. La Secretaria selecciona el ciclo de estudio	4. Habilita el botón Agregar.
5. Presiona el botón Agregar y modifica el inicio o el final del ciclo y presiona aplicar	6. El sistema valida datos, y guarda el ciclo de estudio 7. Carga la lista de ciclos de estudio.

**Tabla. 8.3.12.3**

#### 8.3.12.3.1 Diagrama de Robustez



**Fig. 8.3.12.3.1**

### 8.3.12.3.2 Diagrama de secuencia

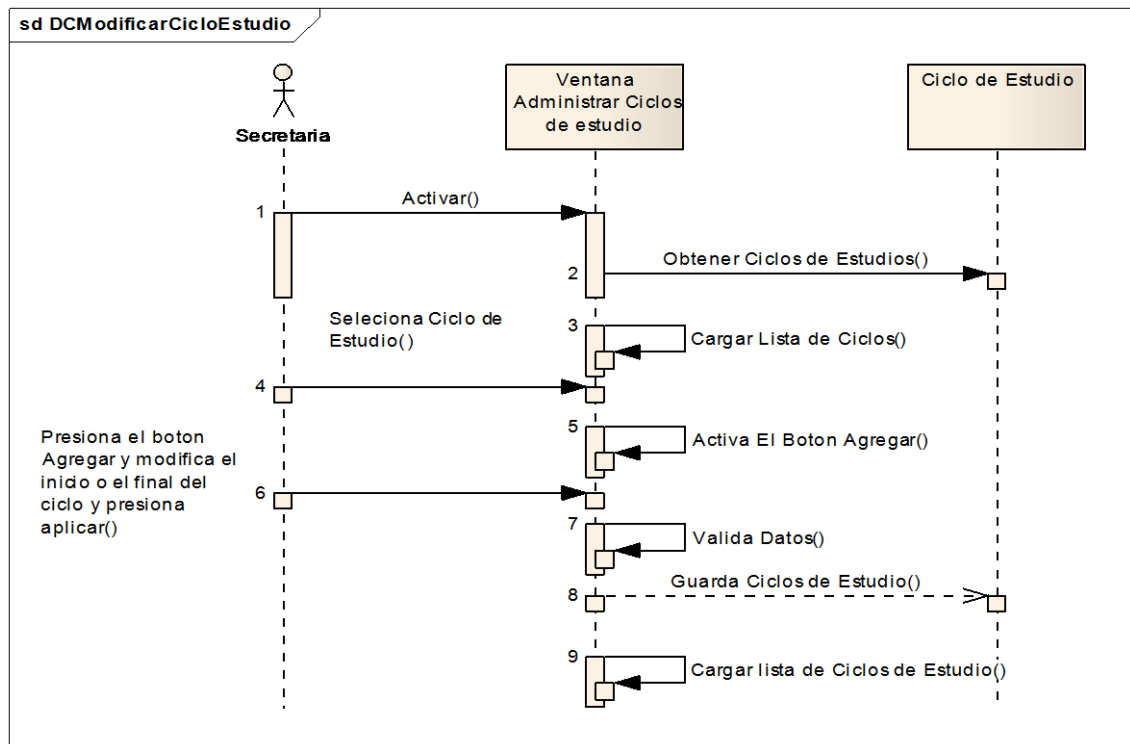


Fig. 8.3.12.3.2

### 8.3.12.3.3 Diagrama de Colaboración

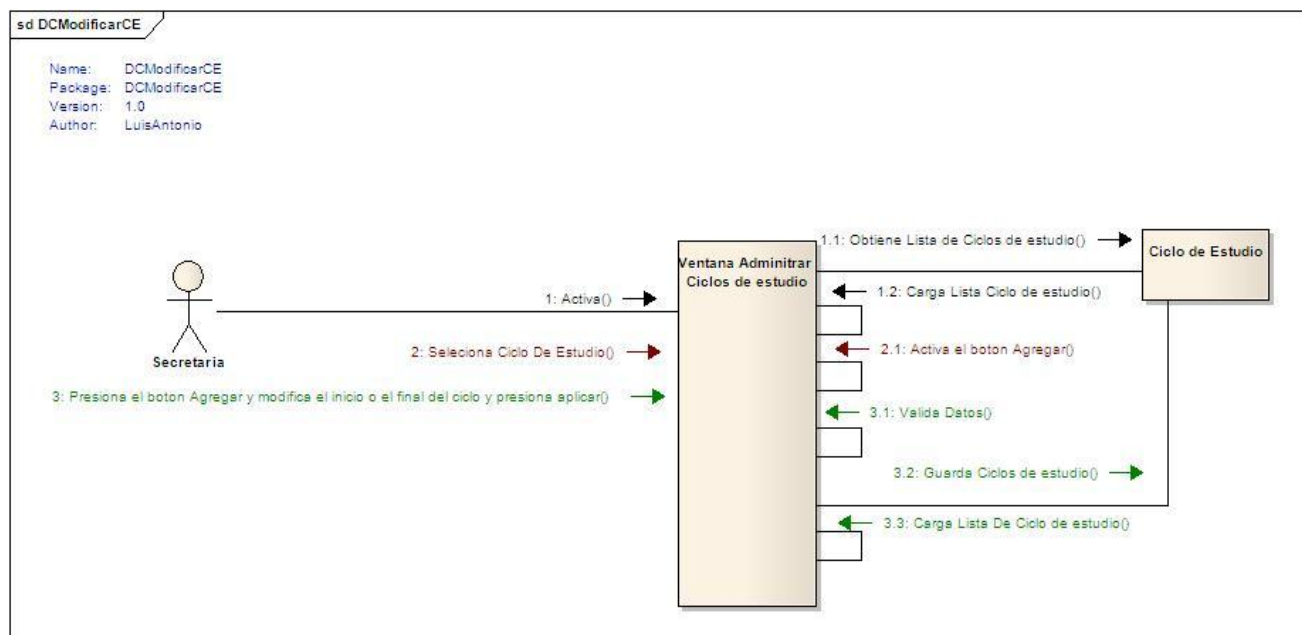


Fig. 8.3.12.3.1

## 8.4 Modelo de paquetes

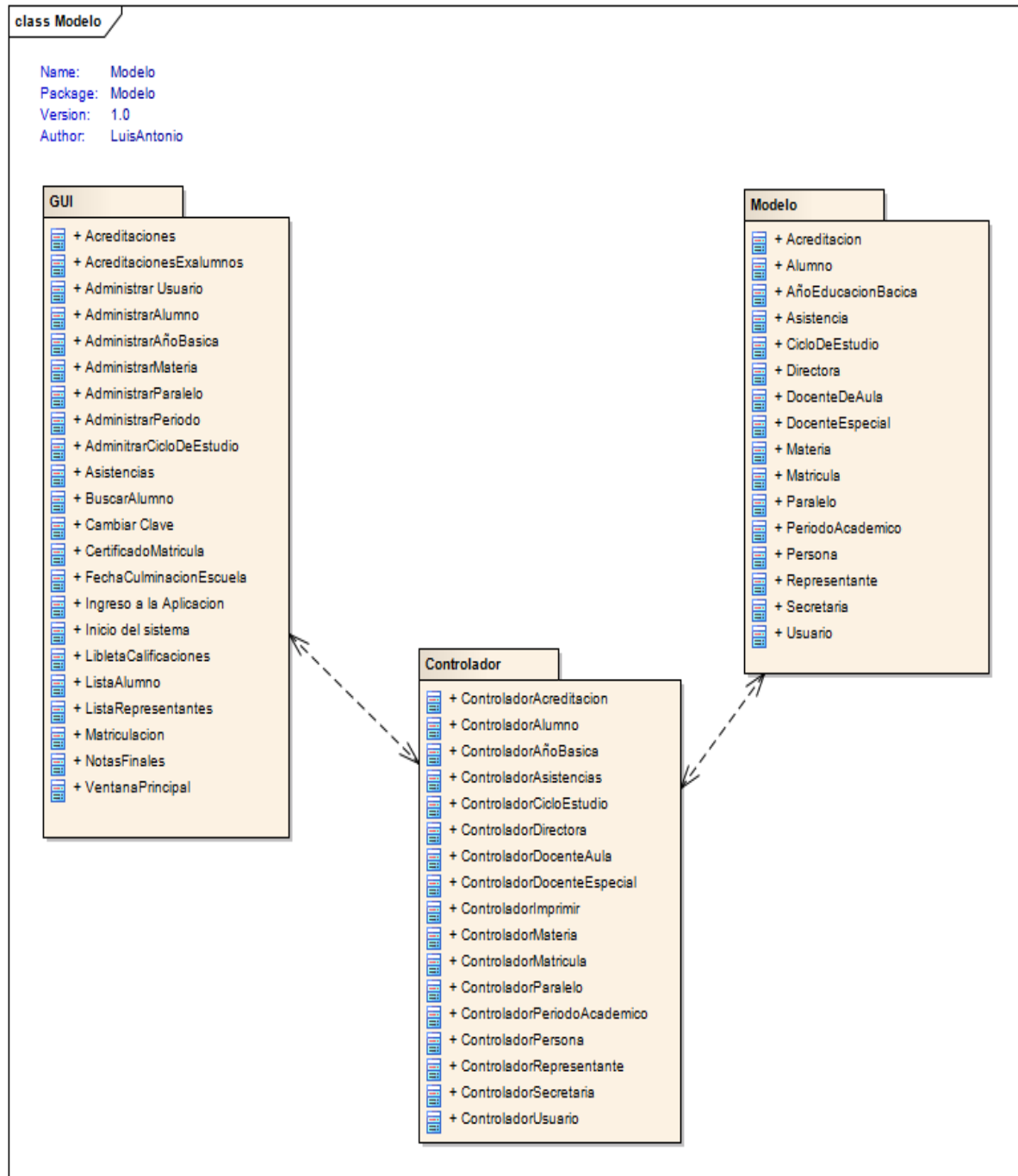


Fig.8.4

## 8.5 Diagrama de Ventanas

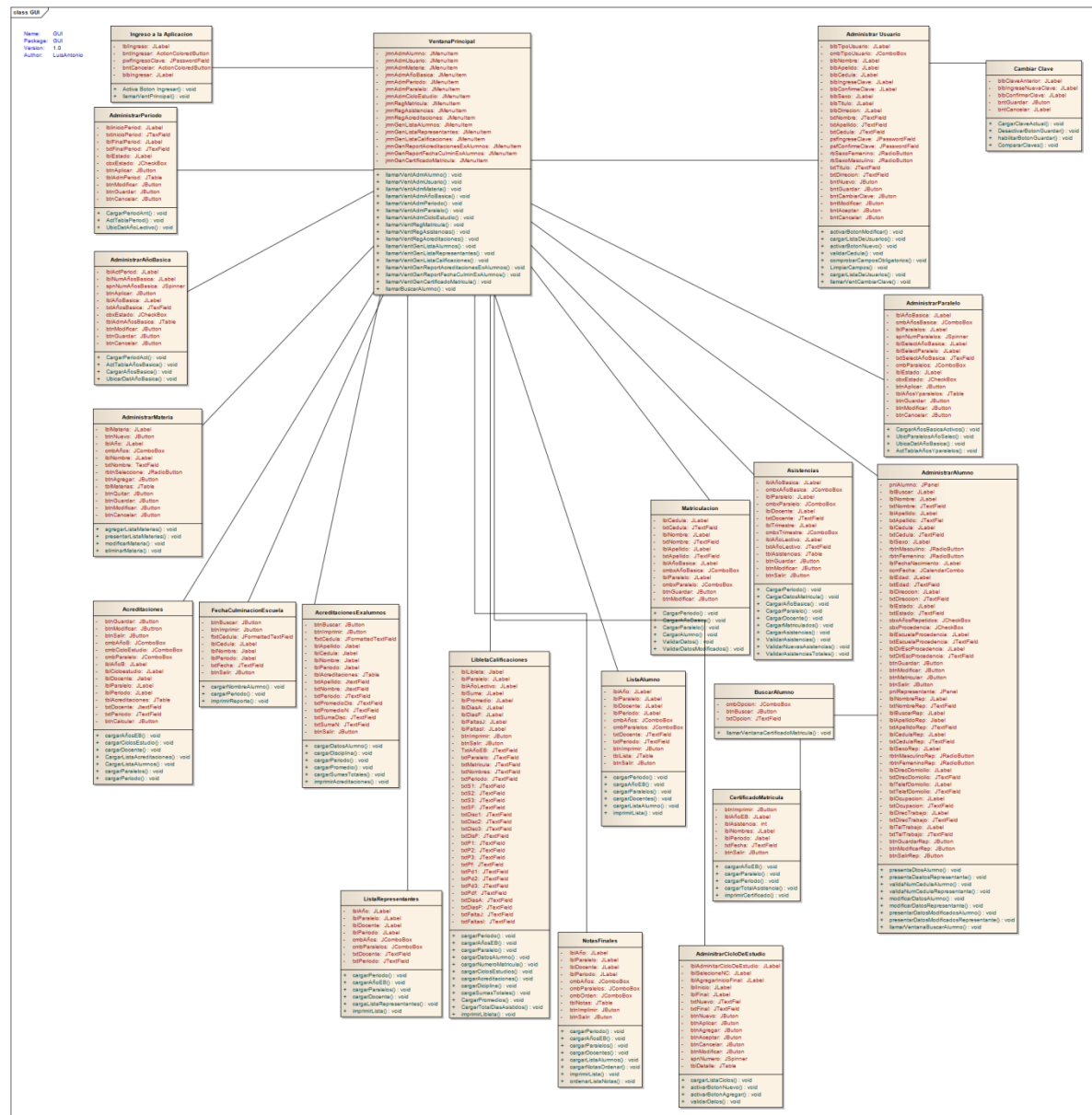


Fig. 8.5

## 8.6 Diagramas de Controladores

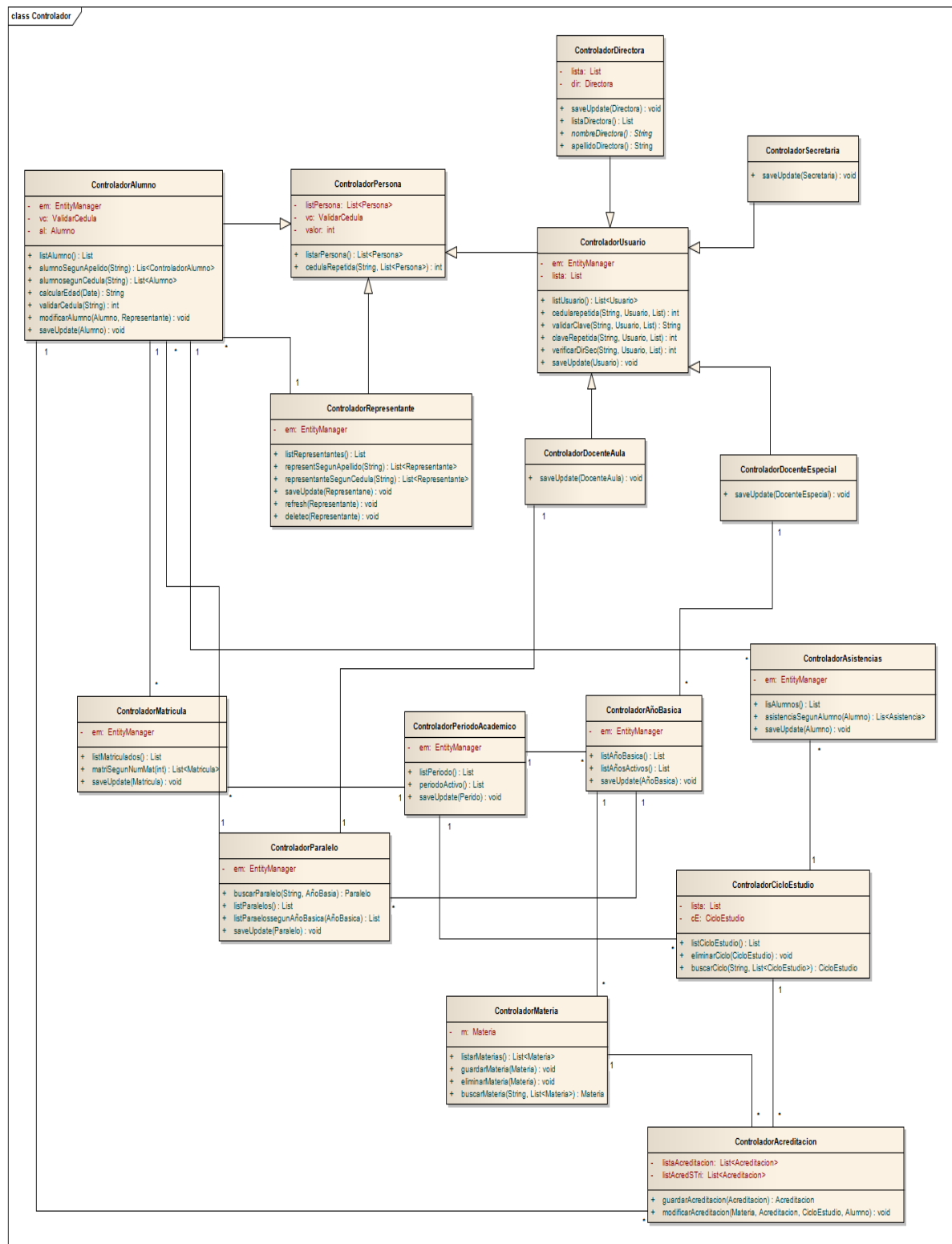


Fig. 8.6

## 8.7 Diagrama de Clases

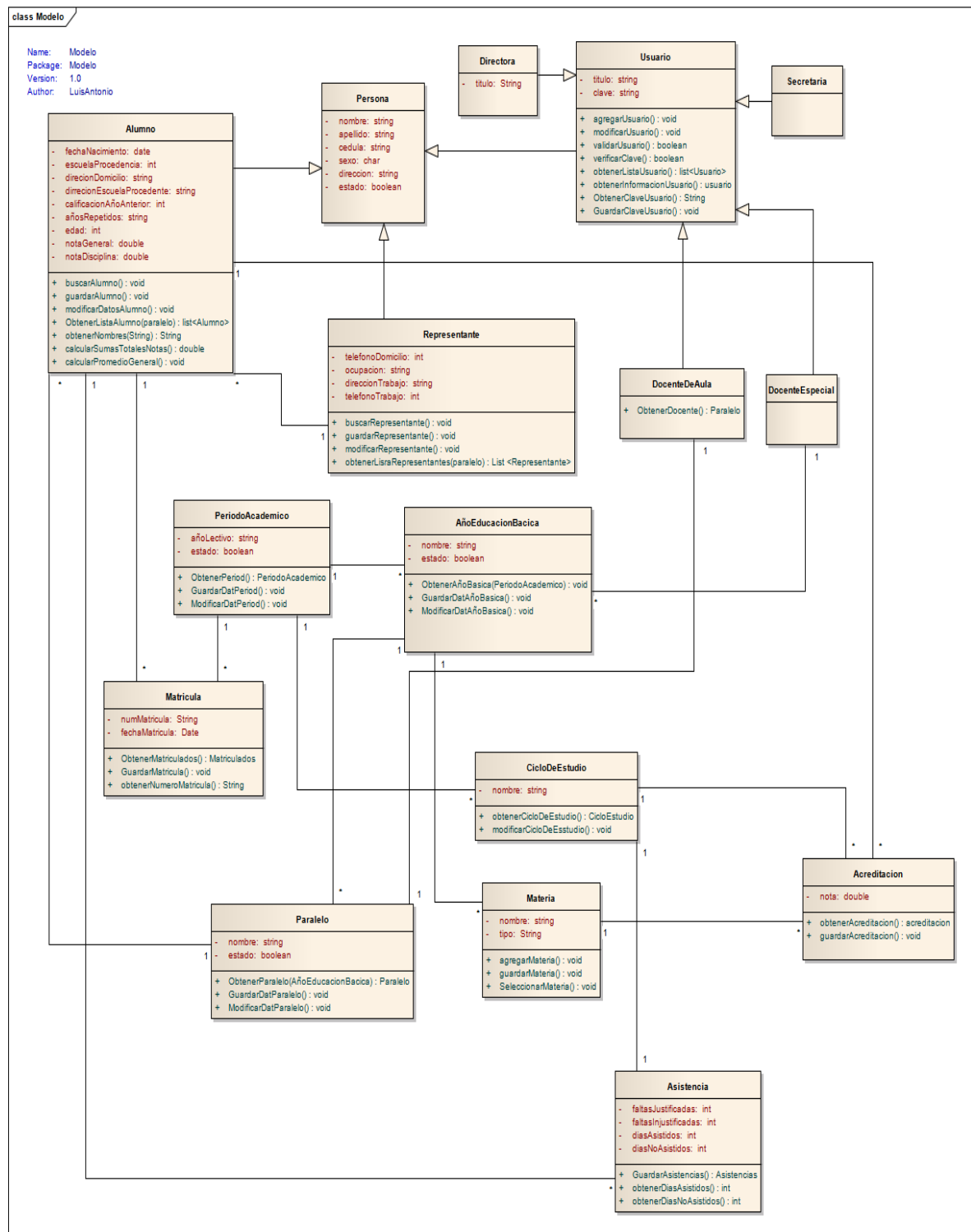


Fig. 8.7

## 8.8 Modelo de la Base de Datos

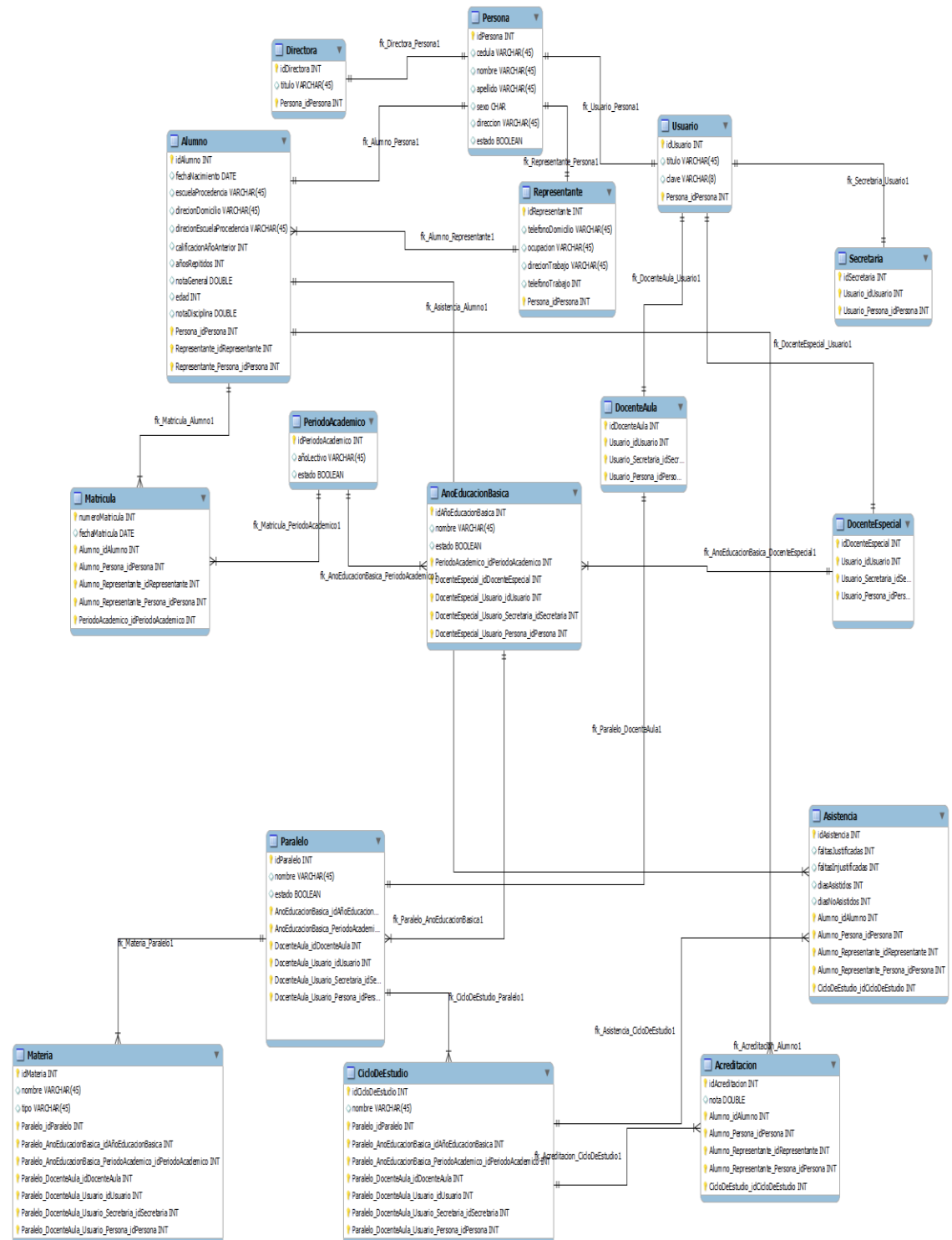


Fig. 8.8



## 9. CONCLUSIONES Y RECOMENDACIONES

### 9.1 CONCLUSIONES

Una vez realizado el diseño e implementación del sistema de información para la Escuela Fiscal Mixta Julio María Matovelle se ha llegado a las siguientes conclusiones:

- ❖ Los requerimientos de la Escuela Fiscal Mixta Julio María Matovelle, permitieron el diseño del modelo de dominio y los casos de uso.
- ❖ La metodología ICONIX se utilizó para las tareas de diseño e implementación del Sistema Informático.
- ❖ La elaboración de Casos de Usos es una etapa fundamental del análisis de Requerimientos para describir el funcionamiento del Sistema.
- ❖ Las herramientas CASE, ayudan a detallar la secuencia de todos los casos de uso
- ❖ Se realizó el mapeo de la base de datos y con esto se pudo crear un programa de fácil manejo el cual permite agilizar el proceso de Gestión Académica en la escuela “Julio María Matovelle”.
- ❖ La descripción de los casos uso extendidos permitieron describir los diagramas de robustez a partir de los cuales se desarrollaron los diagramas de interacción
- ❖ Los diagramas de clases y base de datos ayudan a comprender de mejor manera el sistema para la generación del código

## 9.2 RECOMENDACIONES

Como grupo de investigación recomendamos lo siguiente:

- ❖ Se recomienda el uso de herramientas Case para la creación del mapeo de la base de datos, así mismo para la creación de la base de datos.
- ❖ Se recomienda la elaboración de un manual de usuario que explique paso a paso el funcionamiento de sistema Académico, esto facilitara el correcto uso del sistema por parte de los usuarios.
- ❖ Usar una herramienta de diseño para la creación de casos de uso, modelo, diagramas.
- ❖ Realizar correctamente las relaciones entre las clases para que no presente inconsistencia en los datos.
- ❖ Utilizar el programa SQL Manager para facilitar la generación de las tablas mediante el Script.
- ❖ Usar una metodología ICONIX de manera que facilite el Diseño e implementación de un sistema informático.

## 10. BIBLIOGRAFÍA

- Kendal y Kendal. Análisis y Diseño de sistemas. Sexta Edición, 2005
- Larman Craig. UMLyPatrones. Introducción al Análisis y Diseño Orientado a Objetos
- <http://www.monografias.com/trabajos/anaydisis/anaydisis.shtml>
- [http://es.wikipedia.org/wiki/An%C3%A1lisis\\_de\\_sistemas](http://es.wikipedia.org/wiki/An%C3%A1lisis_de_sistemas)
- Larman Craig. UML.y.Patrones. Introducción al Análisis y Diseño Orientado a Objetos. Prentice.Hall.
- Fowler Martin, Kendall Scot. Uml Gota a Gota.Pearson Education.
- ROSENBERG D, SCOTT K, Use Case Driven Object Modeling with UML: a practical approach, Editorial AddisonWesley, USA.
- REED, Paul R. Jr., Developing Applications with Java and UML, Editorial AddisonWesley, 2002, USA.
- SCHNEIDER G, WINTERS J. P., Applying Use Cases a practical Guide, Editorial AddisonWesley Object Technology Series, USA.
- RUMBAUGH J, JACOBSON I, BOOCH G, The Unified Modeling Language Reference Manual, Editorial AddisonWesley Object



### Script de la base de datos

```

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

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1
COLLATE latin1_swedish_ci ;
SHOW WARNINGS;
USE `mydb`;

-- -----
-- Table `mydb`.`Persona`
-- -----
DROP TABLE IF EXISTS `mydb`.`Persona` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Persona` (
  `idPersona` INT NOT NULL ,
  `nombre` VARCHAR(25) NULL ,
  `apellido` VARCHAR(25) NULL ,
  `cedula` VARCHAR(10) NULL ,
  `edad` VARCHAR(15) NULL ,
  PRIMARY KEY (`idPersona`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`Matricula`
-- -----
DROP TABLE IF EXISTS `mydb`.`Matricula` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Matricula` (
  `idNumMatricula` INT NOT NULL ,
  `fechaMatricula` DATE NULL ,
  PRIMARY KEY (`idNumMatricula`) )
ENGINE = InnoDB;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`Acreditacion`
-- -----
DROP TABLE IF EXISTS `mydb`.`Acreditacion` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Acreditacion` (
  `idAcreditacion` INT NOT NULL ,
  `nota` INT NULL ,
  PRIMARY KEY (`idAcreditacion`) )
ENGINE = InnoDB;

```

```
SHOW WARNINGS;
```

```
-- -----  
-- Table `mydb`.`Asistencias`  
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`Asistencias` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `Asistencias` (  
  `idAsistencias` INT NOT NULL ,  
  `faltasJustificadas` INT NULL ,  
  `faltasNJustificadas` INT NULL ,  
  `diasAsistidos` INT NULL ,  
  `diasNAsistidos` INT NULL ,  
  PRIMARY KEY (`idAsistencias`) )  
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-- -----  
-- Table `mydb`.`Paralelo`  
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`Paralelo` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `Paralelo` (  
  `idParalelo` INT NOT NULL ,  
  `nombre` VARCHAR(5) NULL ,  
  `estado` TINYINT(1) NULL ,  
  PRIMARY KEY (`idParalelo`) )  
ENGINE = InnoDB;
```

```
SHOW WARNINGS;
```

```
-- -----  
-- Table `mydb`.`Alumno`  
-- -----
```

```
DROP TABLE IF EXISTS `mydb`.`Alumno` ;
```

```
SHOW WARNINGS;
```

```
CREATE TABLE IF NOT EXISTS `Alumno` (  
  `fechaNacimiento` DATE NULL ,  
  `escuelaProcedencia` VARCHAR(45) NULL ,  
  `direccionDomicilio` VARCHAR(45) NULL ,  
  `direccionEscuelaProdedencia` VARCHAR(45) NULL ,  
  `calificacionAñoAnterior` INT NULL ,  
  `añosRepetidos` VARCHAR(10) NULL ,  
  `notaGeneral` DOUBLE NULL ,  
  `notaDisciplina` DOUBLE NULL ,  
  `Persona_idPersona` INT NULL ,  
  `Matricula_idNumMatricula` INT NOT NULL ,  
  `Acreditacion_idAcreditacion` INT NOT NULL ,  
  `Asistencias_idAsistencias` INT NOT NULL ,  
  `Paralelo_idParalelo` INT NOT NULL ,
```

```

PRIMARY KEY (`Matricula_idNumMatricula`,
`Acreditacion_idAcreditacion`, `Asistencias_idAsistencias`,
`Paralelo_idParalelo`) ,
CONSTRAINT `fk_Alumno_Personal`
FOREIGN KEY (`Persona_idPersona` )
REFERENCES `Persona` (`idPersona` )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Alumno_Matriculal`
FOREIGN KEY (`Matricula_idNumMatricula` )
REFERENCES `Matricula` (`idNumMatricula` )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Alumno_Acreditacion1`
FOREIGN KEY (`Acreditacion_idAcreditacion` )
REFERENCES `Acreditacion` (`idAcreditacion` )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Alumno_Asistencias1`
FOREIGN KEY (`Asistencias_idAsistencias` )
REFERENCES `Asistencias` (`idAsistencias` )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Alumno_Paralel1`
FOREIGN KEY (`Paralelo_idParalelo` )
REFERENCES `Paralelo` (`idParalelo` )
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_Alumno_Personal` ON `Alumno`
(`Persona_idPersona` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Alumno_Matriculal` ON `Alumno`
(`Matricula_idNumMatricula` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Alumno_Acreditacion1` ON `Alumno`
(`Acreditacion_idAcreditacion` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Alumno_Asistencias1` ON `Alumno`
(`Asistencias_idAsistencias` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Alumno_Paralel1` ON `Alumno`
(`Paralelo_idParalelo` ASC) ;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`Representante`
-- -----
DROP TABLE IF EXISTS `mydb`.`Representante` ;

```

```

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Representante` (
  `ocupacion` VARCHAR(25) NULL ,
  `direccionTrabajo` VARCHAR(45) NULL ,
  `telefonoTrabajo` VARCHAR(25) NULL ,
  `Persona_idPersona` INT NULL ,
  `Alumno_Matricula_idNumMatricula` INT NOT NULL ,
  `Alumno_Acreditacion_idAcreditacion` INT NOT NULL ,
  `Alumno_Asistencias_idAsistencias` INT NOT NULL ,
  `Alumno_Paralelo_idParalelo` INT NOT NULL ,
  PRIMARY KEY (`Alumno_Matricula_idNumMatricula`,
`Alumno_Acreditacion_idAcreditacion`,
`Alumno_Asistencias_idAsistencias`,
`Alumno_Paralelo_idParalelo`) ,
  CONSTRAINT `fk_Representante_Personal`
    FOREIGN KEY (`Persona_idPersona`)
    REFERENCES `Persona` (`idPersona`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Representante_Alumno1`
    FOREIGN KEY (`Alumno_Matricula_idNumMatricula`,
`Alumno_Acreditacion_idAcreditacion`,
`Alumno_Asistencias_idAsistencias`,
`Alumno_Paralelo_idParalelo`)
    REFERENCES `Alumno` (`Matricula_idNumMatricula`,
`Acreditacion_idAcreditacion`, `Asistencias_idAsistencias`,
`Paralelo_idParalelo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SHOW WARNINGS;
CREATE INDEX `fk_Representante_Personal` ON `Representante`
(`Persona_idPersona` ASC) ;

```

```

SHOW WARNINGS;
CREATE INDEX `fk_Representante_Alumno1` ON `Representante`
(`Alumno_Matricula_idNumMatricula` ASC,
`Alumno_Acreditacion_idAcreditacion` ASC,
`Alumno_Asistencias_idAsistencias` ASC,
`Alumno_Paralelo_idParalelo` ASC) ;

```

```

SHOW WARNINGS;

```

```

-- -----
-- Table `mydb`.`Directora`
-- -----
DROP TABLE IF EXISTS `mydb`.`Directora` ;

```

```

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Directora` (
)
ENGINE = InnoDB;

```

```

SHOW WARNINGS;

```



```

-----
-- Table `mydb`.`DocenteDeAula`
-----
DROP TABLE IF EXISTS `mydb`.`DocenteDeAula` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `DocenteDeAula` (
  `Paralelo_idParalelo` INT NOT NULL ,
  PRIMARY KEY (`Paralelo_idParalelo`) ,
  CONSTRAINT `fk_DocenteDeAula_Paralelo1`
    FOREIGN KEY (`Paralelo_idParalelo`)
    REFERENCES `Paralelo` (`idParalelo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_DocenteDeAula_Paralelo1` ON `DocenteDeAula`
(`Paralelo_idParalelo` ASC) ;

SHOW WARNINGS;

-----
-- Table `mydb`.`Secretaria`
-----
DROP TABLE IF EXISTS `mydb`.`Secretaria` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Secretaria` (
)
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `mydb`.`DocenteEspecial`
-----
DROP TABLE IF EXISTS `mydb`.`DocenteEspecial` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `DocenteEspecial` (
)
ENGINE = InnoDB;

SHOW WARNINGS;

-----
-- Table `mydb`.`Usuario`
-----
DROP TABLE IF EXISTS `mydb`.`Usuario` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Usuario` (
  `clave` VARCHAR(25) NULL ,
  `cargo` VARCHAR(25) NULL ,

```

```

`titulo` VARCHAR(25) NULL ,
`Persona_idPersona` INT NULL ,
`DocenteDeAula_Paralelo_idParalelo` INT NULL ,
`Secretaria_idSecretaria` INT NULL ,
`Directora_idDirectora` INT NULL ,
`DocenteEspecial_idDocenteEspecial` INT NULL ,
CONSTRAINT `fk_Usuario_Personal`
    FOREIGN KEY (`Persona_idPersona` )
    REFERENCES `Persona` (`idPersona` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario_DocenteDeAula1`
    FOREIGN KEY (`DocenteDeAula_Paralelo_idParalelo` )
    REFERENCES `DocenteDeAula` (`Paralelo_idParalelo` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario_Secretarial`
    FOREIGN KEY (`Secretaria_idSecretaria` )
    REFERENCES `Secretaria` (`` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario_Directoral`
    FOREIGN KEY (`Directora_idDirectora` )
    REFERENCES `Directora` (`` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario_DocenteEspecial1`
    FOREIGN KEY (`DocenteEspecial_idDocenteEspecial` )
    REFERENCES `DocenteEspecial` (`idDocenteEspecial` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_Usuario_Personal` ON `Usuario`
(`Persona_idPersona` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Usuario_DocenteDeAula1` ON `Usuario`
(`DocenteDeAula_Paralelo_idParalelo` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Usuario_Secretarial` ON `Usuario`
(`Secretaria_idSecretaria` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Usuario_Directoral` ON `Usuario`
(`Directora_idDirectora` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_Usuario_DocenteEspecial1` ON `Usuario`
(`DocenteEspecial_idDocenteEspecial` ASC) ;

SHOW WARNINGS;

```

---

```
-- Table `mydb`.`Materia`
-- -----
DROP TABLE IF EXISTS `mydb`.`Materia` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `Materia` (
  `idMateria` INT NOT NULL ,
  `nombre` VARCHAR(25) NULL ,
  `tipo` VARCHAR(25) NULL ,
  `Acreditacion_idAcreditacion` INT NOT NULL ,
  PRIMARY KEY (`idMateria`, `Acreditacion_idAcreditacion`) ,
  CONSTRAINT `fk_Materia_Acreditacion1`
    FOREIGN KEY (`Acreditacion_idAcreditacion`)
    REFERENCES `Acreditacion` (`idAcreditacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_Materia_Acreditacion1` ON `Materia`
(`Acreditacion_idAcreditacion` ASC) ;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`AñoEducacionBasica`
-- -----
DROP TABLE IF EXISTS `mydb`.`AñoEducacionBasica` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `AñoEducacionBasica` (
  `idAñoEducacionBasica` INT NOT NULL ,
  `nombre` VARCHAR(20) NULL ,
  `estado` TINYINT(1) NULL ,
  `Materia_idMateria` INT NOT NULL ,
  `Paralelo_idParalelo` INT NOT NULL ,
  `DocenteEspecial_idDocenteEspecial` INT NOT NULL ,
  PRIMARY KEY (`idAñoEducacionBasica`, `Materia_idMateria`,
`Paralelo_idParalelo`, `DocenteEspecial_idDocenteEspecial`) ,
  CONSTRAINT `fk_AñoEducacionBasica_Material`
    FOREIGN KEY (`Materia_idMateria`)
    REFERENCES `Materia` (`idMateria`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_AñoEducacionBasica_Paralelo1`
    FOREIGN KEY (`Paralelo_idParalelo`)
    REFERENCES `Paralelo` (`idParalelo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_AñoEducacionBasica_DocenteEspecial1`
    FOREIGN KEY (`DocenteEspecial_idDocenteEspecial`)
    REFERENCES `DocenteEspecial` (`idDocenteEspecial`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```

SHOW WARNINGS;
CREATE INDEX `fk_AÃtoEducacionBasica_Material` ON
`AÃtoEducacionBasica` (`Materia_idMateria` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_AÃtoEducacionBasica_Paralelo1` ON
`AÃtoEducacionBasica` (`Paralelo_idParalelo` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_AÃtoEducacionBasica_DocenteEspecial1` ON
`AÃtoEducacionBasica` (`DocenteEspecial_idDocenteEspecial` ASC)
;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`CicloDeEstudio`
-- -----
DROP TABLE IF EXISTS `mydb`.`CicloDeEstudio` ;

SHOW WARNINGS;
CREATE TABLE IF NOT EXISTS `CicloDeEstudio` (
  `idCicloDeEstudio` INT NOT NULL ,
  `nombre` VARCHAR(25) NULL ,
  `Acreditacion_idAcreditacion` INT NOT NULL ,
  `Asistencias_idAsistencias` INT NOT NULL ,
  PRIMARY KEY (`idCicloDeEstudio`,
`Acreditacion_idAcreditacion`, `Asistencias_idAsistencias`) ,
  CONSTRAINT `fk_CicloDeEstudio_Acreditacion1`
    FOREIGN KEY (`Acreditacion_idAcreditacion` )
    REFERENCES `Acreditacion` (`idAcreditacion` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_CicloDeEstudio_Asistencias1`
    FOREIGN KEY (`Asistencias_idAsistencias` )
    REFERENCES `Asistencias` (`idAsistencias` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_CicloDeEstudio_Acreditacion1` ON
`CicloDeEstudio` (`Acreditacion_idAcreditacion` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_CicloDeEstudio_Asistencias1` ON
`CicloDeEstudio` (`Asistencias_idAsistencias` ASC) ;

SHOW WARNINGS;

-- -----
-- Table `mydb`.`PeriodoAcademico`
-- -----
DROP TABLE IF EXISTS `mydb`.`PeriodoAcademico` ;

SHOW WARNINGS;

```

```

CREATE TABLE IF NOT EXISTS `PeriodoAcademico` (
  `idPeriodoAcademico` INT NOT NULL ,
  `inicioPeriodo` DATE NULL ,
  `finPeriodo` DATE NULL ,
  `Matricula_idNumMatricula` INT NOT NULL ,
  `AÃtoEducacionBasica_idAÃtoEducacionBasica` INT NOT NULL ,
  `CicloDeEstudio_idCicloDeEstudio` INT NOT NULL ,
  PRIMARY KEY (`idPeriodoAcademico`, `Matricula_idNumMatricula`,
  `AÃtoEducacionBasica_idAÃtoEducacionBasica`,
  `CicloDeEstudio_idCicloDeEstudio`) ,
  CONSTRAINT `fk_PeriodoAcademico_Matriculal`
    FOREIGN KEY (`Matricula_idNumMatricula`)
    REFERENCES `Matricula` (`idNumMatricula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_PeriodoAcademico_AÃtoEducacionBasical`
    FOREIGN KEY (`AÃtoEducacionBasica_idAÃtoEducacionBasica`)
    REFERENCES `AÃtoEducacionBasica` (`idAÃtoEducacionBasica`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_PeriodoAcademico_CicloDeEstudio1`
    FOREIGN KEY (`CicloDeEstudio_idCicloDeEstudio`)
    REFERENCES `CicloDeEstudio` (`idCicloDeEstudio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SHOW WARNINGS;
CREATE INDEX `fk_PeriodoAcademico_Matriculal` ON
`PeriodoAcademico` (`Matricula_idNumMatricula` ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_PeriodoAcademico_AÃtoEducacionBasical` ON
`PeriodoAcademico` (`AÃtoEducacionBasica_idAÃtoEducacionBasica`
ASC) ;

SHOW WARNINGS;
CREATE INDEX `fk_PeriodoAcademico_CicloDeEstudio1` ON
`PeriodoAcademico` (`CicloDeEstudio_idCicloDeEstudio` ASC) ;

SHOW WARNINGS;

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

```

