

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **MÓDULO DE REGISTRO ESTUDIANTIL DEL SAE EN PLATAFORMA DE LIBRE DIFUSIÓN.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**VELASCO MALDONADO SYLVIA XIMENA**

**VILCA CHILQUINGA PAOLA JESSENIA**

**DIRECTOR: ING. NIDIA GUAYAQUIL**

**Quito, Abril 2007**

## DECLARACIÓN

Nosotros, Velasco Maldonado Sylvia Ximena y Vilca Chiliquinga Paola Jessenia, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Velasco Maldonado Sylvia Ximena

---

Vilca Chiliquinga Paola Jessenia

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Velasco Maldonado Sylvia Ximena y Vilca Chiliquina Paola Jessenia, bajo mi supervisión.

---

**Ing. Nidia Guayaquil**  
**DIRECTOR DE PROYECTO**

## RESUMEN

El presente trabajo muestra el desarrollo de un prototipo del Modulo de Registro Estudiantil de SAE (Sistema de Administración Académica) en plataforma de libre difusión, dicho modulo tiene como objetivo principal permitir que los estudiantes de la EPN puedan matricularse Vía Web acelerando y optimizando el proceso de inscripción y matriculación, además de realizar ciertas consultas de carácter informativo.

Los antecedentes del SAE forman parte del fundamento teórico, así como la información proporcionada acerca de Software Libre y Reingeniería de Software.

En base a la teoría expuesta acerca de Reingeniería de Software para desarrollo del sistema, se realiza el inventario del modulo, ingeniería inversa, ingeniería progresiva en la cual se aplica el siguiente proceso: definición de requerimientos, análisis, diseño, implementación y pruebas del mismo. Finalmente se realiza la evaluación de los resultados obtenidos en las pruebas efectuadas.

## CONTENIDO

CAPITULO 1: MARCO TEORICO .....	1
1.1 ANTECEDENTES DEL SAE.....	1
1.1.1 RESEÑA HISTORICA .....	1
1.1.2 GENERALIDADES DEL SAE2000 .....	3
1.1.3 ALCANCE DEL SISTEMA .....	3
1.2 SOFTWARE LIBRE .....	6
1.2.1 INTRODUCCIÓN.....	6
1.2.2 ¿QUÉ ES SOFTWARE LIBRE?.....	7
1.2.2.1 Licencias Utilizadas en el Software Libre.....	9
1.2.3 ¿QUÉ ES SOFTWARE DE CÓDIGO CERRADO O SOFTWARE COMERCIAL? .....	12
1.2.4 SOTFWARE LIBRE VS SOFTWARE COMERCIAL O PROPIETARIO.....	13
1.2.5 COSTO DEL CAMBIO AL UTILIZAR SOFTWARE LIBRE .....	14
1.3 REINGENIERÍA .....	14
1.3.1 MODELO DE PROCESO DE REINGENIERIA DE SOFTWARE .....	16
CAPÍTULO 2: REINGENIERIA DEL MÓDULO PRINCIPAL DEL SISTEMA SAE....	20
2.1 INVENTARIO .....	20
2.1.1 NOMBRE Y OBJETIVO .....	20
2.1.2 FUNCIONALIDAD .....	21
2.1.3 PLATAFORMA Y METODOLOGÍA.....	22
2.1.4 DOCUMENTACIÓN.....	23
2.1.5 EQUIPO DE DESARROLLO.....	23
2.1.6 PROBLEMAS .....	24
2.2 INGENIERÍA INVERSA.....	29
2.3 INGENIERÍA PROGRESIVA .....	29
2.3.1 REQUERIMIENTOS .....	29
2.3.2 ANÁLISIS .....	57
2.3.3 DISEÑO .....	80
CAPITULO 3: CONSTRUCCIÓN Y PRUEBAS .....	108
3.1 SELECCIÓN DE LAS HERRAMIENTAS.....	108
3.1.1 HERRAMIENTAS DE BACK – END .....	108
3.1.2 HERRAMIENTAS DE FRONT – END .....	118
3.1.3 JUSTIFICACIÓN DE LA SELECCIÓN DE HERRAMIENTAS .....	128
3.2 CONSTRUCCIÓN.....	129
3.2.1 CONSIDERACIONES PARA LA CONSTRUCCIÓN.....	129
3.2.2 PRUEBAS .....	135
CAPITULO 4: EVALUACIÓN .....	2
4.1 AMBIENTE .....	2
4.2 INTEGRIDAD .....	2
4.3 ANÁLISIS DE RESULTADOS.....	3
CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES .....	5
5.1 CONCLUSIONES.....	5
5.2 RECOMENDACIONES: .....	6
BIBLIOGRAFÍA: .....	8

## INDICE DE FIGURAS

CAPITULO 1: MARCO TEORICO .....	1
Figura 1.1 Procesos en reingeniería de software .....	16
CAPÍTULO 2: REINGENIERIA DEL MÓDULO PRINCIPAL DEL SISTEMA SAE....	20
Figura 2.1: Modelo de Caso de Uso del Negocio.....	29
Figura 2.1: Modelo de Caso de Uso del Negocio.....	30
Figura 2.2: Casos de Uso del Modulo de Registro Estudiantil.....	31
Figura 2.3: Diagrama de colaboración CU – 01 Solicitar información estudiante. ....	59
Figura 2.4: Diagrama de colaboración CU – 02 Generar Inscripción. ....	61
Figura 2.5: Diagrama de colaboración CU – 03 Reinscribir. ....	62
Figura 2.6: Diagrama de colaboración CU – 04 Anular matricula.....	63
Figura 2.7: Diagrama de colaboración CU – 05 Enviar al banco.....	64
Figura 2.8: Diagrama de colaboración CU – 06 Pagar en tesorería. ....	65
Figura 2.9: Diagrama de colaboración CU – 07 Legalizar matricula.....	66
Figura 2.10: Diagrama de colaboración CU – 08 Autorizar reinscripciones. ....	67
Figura 2.11: Diagrama de colaboración CU – 09 Autorizar reingresos. ....	68
Figura 2.12: Diagrama de colaboración CU – 10 Autorizar matriculas extraordinarias. ....	69
Figura 2.13: Diagrama de colaboración CU – 11 Autorizar matriculas extemporáneas. ....	70
Figura 2.14: Diagrama de colaboración CU – 12 Autorizar cambios de carrera. ....	71
Figura 2.15: Diagrama de colaboración CU – 13 Registro de impedimentos de matricula. .....	72
Figura 2.16: Diagrama de colaboración CU – 14 Registrar aspirante.....	73
Figura 2.17: Diagrama de colaboración CU – 15 Registrar rubros adicionales.....	74
Figura 2.18: Diagrama de colaboración CU – 16 Autorizar pagos por partes. ....	75
Figura 2.19: Diagrama de colaboración CU – 17 Autorizar pagos con IECE.....	76
Figura 2.20: Diagrama de colaboración CU – 18 Registrar IPF. ....	77
Figura 2.21: Diagrama de colaboración CU – 19 Registrar descuentos.....	78
Figura 2.22: Diagrama de colaboración CU – 20 Registrar becas. ....	79
Figura 2.23: Diagrama de Modelo de Casos de Uso de Diseño CU – 01 Solicitar información.....	82
Figura 2.24: Diagrama de Modelo de Casos de Uso de Diseño CU – 02 Generar Inscripción. ....	83
Figura 2.25: Diagrama de Modelo de Casos de Uso de Diseño CU – 03 Reinscribir....	83
Figura 2.26: Diagrama de Modelo de Casos de Uso de Diseño CU – 04 Anular matricula. ....	84
Figura 2.27: Diagrama de Modelo de Casos de Uso de Diseño CU – 05 Enviar al banco. .....	84
Figura 2.28: Diagrama de Modelo de Casos de Uso de Diseño CU – 06 Pagar en tesorería. ....	85
Figura 2.29: Diagrama de Modelo de Casos de Uso de Diseño CU – 07 Legalizar matricula. ....	85
Figura 2.30: Diagrama de Modelo de Casos de Uso de Diseño CU – 08 Autorizar reinscripciones. ....	86
Figura 2.31: Diagrama de Modelo de Casos de Uso de Diseño CU – 09 Autorizar reingresos.....	86
Figura 2.32: Diagrama de Modelo de Casos de Uso de Diseño CU – 10 Autorizar matriculas extraordinarias.....	87

Figura 2.33: Diagrama de Modelo de Casos de Uso de Diseño CU – 11 Autorizar matriculas extemporáneas.....	87
Figura 2.34: Diagrama de Modelo de Casos de Uso de Diseño CU – 12 Autorizar cambios de carrera. ....	88
Figura 2.35: Diagrama de Modelo de Casos de Uso de Diseño CU – 13 Registro de impedimentos de matrícula.....	88
Figura 2.36: Diagrama de Modelo de Casos de Uso de Diseño CU – 14 Registrar aspirante.....	89
Figura 2.37: Diagrama de Modelo de Casos de Uso de Diseño CU – 15 Registrar rubros adicionales. ....	89
Figura 2.38: Diagrama de Modelo de Casos de Uso de Diseño CU – 16 Autorizar pagos por partes. ....	90
Figura 2.39: Diagrama de Modelo de Casos de Uso de Diseño CU – 17 Autorizar pagos con IECE. ....	90
Figura 2.40: Diagrama de Modelo de Casos de Uso de Diseño CU – 18 Registrar IPF.	91
Figura 2.41: Diagrama de Modelo de Casos de Uso de Diseño CU – 19 Registrar descuentos.....	91
Figura 2.42: Diagrama de Modelo de Casos de Uso de Diseño CU – 20 Registrar becas. ....	92
Figura 2.43: Diagrama de Secuencia CU – 01 Solicitar información. ....	95
Figura 2.44: Diagrama de Componentes. ....	96
Figura 2.45: Diagrama de Arquitectura. ....	96
<b>CAPITULO 3: CONSTRUCCIÓN Y PRUEBAS</b> .....	<b>108</b>
<b>CAPITULO 4: EVALUACIÓN</b> .....	<b>2</b>
<b>CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>5</b>
<b>BIBLIOGRAFÍA:</b> .....	<b>8</b>

## INDICE DE TABLAS

CAPITULO 1: MARCO TEORICO .....	1
Tabla 1.1: Descripción del alcance por módulos del proyecto SAE2000 .....	5
Tabla 1.2: Cuadro comparativo software libre vs. Software propietario.....	14
Tabla 1.3: Metodología de desarrollo de software. ....	19
CAPÍTULO 2: REINGENIERIA DEL MÓDULO PRINCIPAL DEL SISTEMA SAE....	20
Tabla 2.1: Factor Interno .....	28
Tabla 2.2: Lista y descripción de actores. ....	33
Tabla 2.3: Lista y funciones de los casos de uso .....	35
Tabla 2.4: CU – 01 Solicitar información estudiante. ....	37
Tabla 2.5: CU – 02 Generar Inscripción.....	39
Tabla 2.6: CU – 03 Reinscribir.....	40
Tabla 2.7: CU – 04 Anular matrícula. ....	41
Tabla 2.8: CU – 05 Enviar al banco. ....	41
Tabla 2.9: CU – 06 Pagar en tesorería. ....	42
Tabla 2.10: CU – 07 Legalizar matrícula. ....	43
Tabla 2.11: CU – 08 Autorizar reinscripciones. ....	43
Tabla 2.12: CU – 09 Autorizar reingresos.....	44
Tabla 2.13: CU – 10 Autorizar matrículas extraordinarias. ....	44
Tabla 2.14: CU – 11 Autorizar matrículas extemporáneas. ....	45
Tabla 2.15: CU – 12 Autorizar cambios de carrera. ....	45
Tabla 2.16: CU – 13 Registrar impedimentos de matrícula. ....	46
Tabla 2.17: CU – 14 Registrar aspirantes.....	47
Tabla 2.18: CU – 15 Gestionar rubros adicionales.....	47
Tabla 2.19: CU – 16 Autorizar pagos en partes. ....	48
Tabla 2.20: CU – 17 Autorizar pagos con IECE. ....	49
Tabla 2.21: CU – 18 Registrar IPF. ....	49
Tabla 2.22: CU – 19 Registrar documentos. ....	50
Tabla 2.23: CU – 20 Registrar becas.....	51
Tabla 2.24: Casos de Prueba: Ingreso al Sistema.....	104
Tabla 2.25: Casos de Prueba: Información del Estudiante.....	104
CAPITULO 3: CONSTRUCCIÓN Y PRUEBAS .....	108
Tabla 3.1: Cuadro comparativo de Bases de Datos Libres.....	108
Tabla 3.2: Cuadro Comparativo de Herramientas de Desarrollo .....	120
Tabla 3.3: Resumen de estándar utilizado en la plataforma J2ee .....	135
CAPITULO 4: EVALUACIÓN .....	2
CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES .....	5
BIBLIOGRAFÍA: .....	8

## **CAPITULO 1: MARCO TEORICO**

### **1.1 ANTECEDENTES DEL SAE<sup>1</sup>**

#### **1.1.1 RESEÑA HISTORICA**

Desde el año 1976 se da el primer paso para la automatización de la Administración Estudiantil de la Escuela Politécnica Nacional, el cual inicia con el proceso de Matrículas por lotes (batch), donde se obtienen listados por cursos y por facultad, en un computador Mainframe IBM 370, el cual era el más moderno para aquel entonces; dicho computador poseía como medios de entrada tarjetas perforadas y cintas con el lenguaje de programación COBOL.

En 1986 se conforma un equipo de desarrollo cuyo líder fue el Ing. Cristóbal Trujillo, más un programador con el fin de modernizar el antiguo proceso Bach e incluir nuevos requerimientos orientados a un proceso en línea y con las facilidades de los PC's, que ya tomaban auge. Posteriormente se une a este equipo la Ing. Nidia Guayaquil, ya que los dos ingenieros formaban parte del Centro de Cómputo de la Escuela Politécnica Nacional.

La primera versión del SAE en DBASE III se la obtuvo en 1987 y se la utilizaba en Ingeniería Civil. Como todo sistema que lanza sus versiones de mejora este tenía ciertos problemas inherentes al DBSAE III, como el de no producir programas ejecutables y el dejar abierto el manejo de los datos sin seguridad alguna.

A partir del año 1988 ya se contaba con una versión operativa del SAE, aunque no era el sistema oficial de la EPN, comenzó a ser implantado en otras facultades como la de Sistemas, Mecánica, Eléctrica entre otras, Dicha implantación es llevada a cabo por la Ing. Nidia Guayaquil en estaciones de trabajo, bajo el sistema operativo DOS que era la plataforma que se utilizaba en ese momento, y posteriormente se la instaló sobre la plataforma Novell para dar soporte a estaciones en red. Para el uso del nuevo sistema se programó en forma

---

<sup>1</sup> Capítulo 2; Págs. 77-79; AUDITORIA DEL DESEMPEÑO DEL SAE; ESPINOZA TORRES ENNA DEL CARMEN; Quito- Junio 2003

sistemática la capacitación a secretarias, subdecanos, y se atendió con el soporte logístico para que el proceso funcione, a esta versión del SAE en 1990 se la elige como un estándar en el manejo de la Información Estudiantil de la Escuela, razón por la cual su administración pasa a estar bajo el manejo del Centro de Cómputo de la Escuela Politécnica Nacional, la cual realiza desde entonces la implantación de SAE en todas las Facultades e Institutos.

En el año de 1993 se empieza con la implantación del SAE2 el mismo que incluía nuevos requerimientos debido a los cambios administrativos de la Institución y por las experiencias recogidas a lo largo de su funcionamiento; Esta versión incluye cambios por parte de sus creadores como el de ser dividido en módulos independientes como el de Administración del Sistema, el de Utilitarios y el de Administración Estudiantil que a la larga llagaron en conjunto a ser el SAE2 o de una manera formal, el Sistema de Administración Universitaria. Con SAE2 se incorporó la utilización de herramientas CASE las mismas que permitieron concebir la documentación.

En el año 1997, se realiza la planificación del desarrollo de una nueva versión del SAE con la finalidad de acoger una nueva tecnología que permita llevar una mejor consistencia y seguridad de la información, con un motor de base de datos adecuado y además un entorno seguro y amigable.

Este proyecto se concretó en 1999, por gestión administrativa del Rector de la Escuela de aquella época y del Decano de la Facultad de Ingeniería de Sistemas de la Escuela Politécnica Nacional, la versión que se planificó por parte del Ing. Trujillo, se la llamo SAE2000, en la que se planteo la expectativa de hacer una migración de plataforma .

### **1.1.2 GENERALIDADES DEL SAE2000**

SAE2000 fue realizado con herramientas y en plataforma de desarrollo actuales a la fecha en el ámbito informático como era Visual Studio 6.0 y sobre plataforma Windows 2000 Server.

El modelo de Administración Estudiantil del SAE2000, realizan varias funciones como consultas, actualizaciones y emisión de listados, agregando a estas funciones nuevas características operativas.

La nueva versión "SAE2000", envuelve nuevos requerimientos administrativos, que el sistema SAE2 no los operaba tomando en cuenta los requerimientos técnicos, aplicando diferentes paradigmas en el desarrollo de sistemas, y los cambios administrativos y académicos de la Institución.

Los requerimientos técnicos que se plantean con esta versión son, el que sea un sistema certificado para el año 2000, que sea en un ambiente cliente-servidor, con una interfase gráfica amigable, un adecuado administrador de base de datos y además uno de los objetivos es el que éste sistema, sea un sistema de gestión universitaria tal, que en lo posterior sea comercializado para Instituciones afines.

### **1.1.3 ALCANCE DEL SISTEMA**

Actualmente el área de desarrollo tiene bajo su responsabilidad el proyecto SAE2000, el cual tiene como objetivo brindar al servicio de la EPN, una nueva versión del sistema de Administración Estudiantil SAE, que opera por aproximadamente más de dos décadas.

El alcance del proyecto SAE2000 en sus inicios contemplaba los siguientes módulos o subsistemas los cuales se marcan los que están funcionando actualmente.

## ALCANCE DEL PROYECTO SAE2000

MÓDULOS	REALIZADOS
<b>1. PLANIFICACIÓN ACADÉMICA</b>	
Planificar materias, horarios, paralelos, aulas	Si
Planificar carga académica de profesores	Si
Asignar tutores	Si
Planificar pènsum de estudios	Si
<b>2. ADMINISTRACIÓN ESTUDIANTIL</b>	
Administración de matrículas y registro de matriculas	Si
Administración de calificaciones	Si
Emisión de listas	Si
Emisión de certificados	Si
<b>3. EVALUACION ACADÉMICA</b>	
Chequear cumplimiento de horarios, prerrequisitos, y créditos	Si
Evaluar currículum`s	Si
Controlar el cumplimiento académico, registros y evaluación	No
Acumular históricos	Si
<b>4. EVALUACIÓN DOCENTE</b>	
Registro de datos de profesores	Si
Planificar las actividades docentes	No
Hacer el seguimiento y evaluación de desempeño docente	No
Administrar la promoción de los profesores	No
<b>5. CONTROL DE TESIS</b>	
Administrar temas	No
Controlar el avance y rendimiento	No
Administrar graduados	No
<b>6. INFORMACIÓN GERENCIAL ACADEMICA</b>	
Información global	No
Estadísticas	No
Análisis y comparaciones	No
Proyecciones	No

<b>7. INFORMACIÓN ESTUDIANTIL (SIEWEB2000)</b>	
<b>Información disponible para el estudiante sobre:</b>	
Registro de materias y calificaciones	No
Vital	Si
Información de temas de tesis	Si
Materias que se ofrece	Si
Temas de tesis	No
<b>8. INFORMACIÓN DOCENTE</b>	
<b>Informar para que el profesor pueda ingresar:</b>	
Informe semestral	No
Calificaciones	No
<b>Información disponible para el profesor:</b>	
Carga Académica	No
Listas de estudiantes	No
Evaluación semestral	No
Ascensos	No

Tabla 1.1: Descripción del alcance por módulos del proyecto SAE2000

Según el detalle de la tabla del alcance por módulos del proyecto SAE2000 no todos los módulos han sido desarrollados en su totalidad, pero a partir del 2003 se ve la necesidad de integrar la información que se maneja en la E.P.N ya sea esta de los estudiantes, del personal docente, materias, etc.; para lograr esto se inicia el diseño de la versión SAE-MULTICARRERA del cual se ha desarrollado hasta la fecha el Módulo "Control e Proyectos de Titulación y Tesis", y para continuar con esta meta se ha planteado y se está iniciando un nuevo proyecto en el cual se desarrollará una nueva versión integrada del SAE en plataforma WEB que opere sobre la POLIRED.

## 1.2 SOFTWARE LIBRE

### 1.2.1 INTRODUCCIÓN

Desde hace más de 30 años nos hemos acostumbrado a que quien comercializa un programa pueda imponer las condiciones bajo las que puede usarse, esto es prohibir que se preste a tercero.

Con la llegada del Software Libre los usuarios no son solo *consumidores* de información, sino *prosumidores*, pues al mismo tiempo que consumen también son productores, pues generan una proporción muy significativa de los mejores programas que se tienen actualmente.

GNU aseguran que los usuarios posean la libertad de distribuir copias de Software Libre, que reciban el código fuente o que tengan la posibilidad de recibirlo, si así lo desea, les permite que puedan cambiar o modificar el software o utilicen sólo partes del mismo en nuevos paquetes de Software Libre; y finalmente que tengan pleno conocimiento de estas facultades.

La protección que otorga la licencia GNU se hace de dos maneras simultáneas:

- Se otorga protección al software bajo la ley de copyright, y
- Se ofrece la protección bajo esta licencia, la cual otorga permiso legal para copiar, distribuir y/o modificar el software.

La GNU evita que los redistribuidores de software libre puedan obtener una patente y así poder obtener programas de software propietario. Para lograr esto, queda totalmente claro que cualquier patente debe otorgar licencias que permitan el uso libre del programa para todos o no otorgar licencia alguna.

## 1.2.2 ¿QUÉ ES SOFTWARE LIBRE?

Para entender lo que es El Software Libre se debe tener claro que es cuestión de libertad más no a precios y al decir libertad nos referimos a libertad de usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Todo esto se relaciona con cuatro clases de libertad para los usuarios de software que son:

- Libertad 0: La libertad de ejecutarlo en cualquier propósito.
- Libertad 1: La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- Libertad 2: La libertad de distribuir copias.
- Libertad 3: La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

El Software Libre presenta algunas características, entre ellas tenemos:

- El Software Libre es siempre legal, salvo cuando compilemos el código fuente y lo vendamos como propietario. Por lo tanto, al utilizar este tipo de software estaremos siempre "por derecha", por lo que no seremos pasibles de multas y/o prisión.
- El software libre es generado por expertos programadores voluntarios, empresas, administraciones y otros tipos de organizaciones que ofrecen las soluciones desarrolladas al resto de la comunidad para que se utilicen de forma 'libre'
- El Software Libre centra más sus fuentes de ingresos en los servicios de valor añadido para los clientes como son personalización, integración y soporte.
- El Software Libre es transparente pues entrega el código fuente que puede ser revisado exhaustivamente para encontrar algún problema de seguridad o código malicioso.

- El Software Libre permite el ahorro en costes, evidentemente su libertad de uso y copia de un mismo programa cuantas veces se desee, trae un ahorro de costes considerable. Una vez que los costes de cambio se han realizado, los costes de mantenimiento y de mejora son bastante menores en relación a un software comercial.

El Software Libre tiene algunas clasificaciones las mismas que son<sup>2</sup>:

- **De acuerdo al costo de adquisición:** Puede ser de dos clases, es decir, de costo cero o de costo mayor que cero. Lo que lo diferencia del Software Propietario es que su costo es independiente del número de computadoras que se poseen.
- **De acuerdo a la apertura del código fuente:** Siempre es "open source", es decir, que su código fuente es abierto pues es necesario para cumplir con las libertades 1 y 3 que mencionamos en puntos anteriores
- **De acuerdo a su protección:** Siempre se protege con licencias de copyleft, no con las de dominio público ya que de ese modo cualquiera se adueña del software, tampoco con Copyright ya que le pueden agregar restricciones y no seguir siendo software libre.
- **De acuerdo a su legalidad:** Siempre es legal, porque al usarlo, estudiarlo, modificarlo, adaptarlo y/o mejorarlo no se viola ninguna norma ya que este tipo de software permite hacerlo, con la única excepción de no poder agregarle ninguna restricción adicional cuando lo transfiera a otra persona.

A continuación presentaremos la definición de ciertos términos que se relacionan o que tienden a confundirse con el Software Libre:

FSF.- FSF son las siglas de Free Software Foundation, entidad que promueve el uso y desarrollo de software de este tipo.

---

<sup>2</sup> <http://www.monografias.com/trabajos12/elsofrib.shtml.html>; EL SOFTWARE LIBRE; Morea Lucas; 1997

**FREEWARE.-** Programas gratuitos. Normalmente se ceden en binario y con derechos usar, copiar, y distribuir libremente. Se diferencia del Open Source por que el autor siempre es dueño de los derechos, o sea que el usuario no puede realizar algo que no esté expresamente autorizado por el autor del programa, como modificarlo o venderlo.

**SHAREWARE.-** No es siquiera software gratis, sino un método de distribución, ya que los programas generalmente sin fuentes se pueden copiar libremente, pero no usar continuamente sin pagarlos. La exigencia a de pago puede estar incentivada por funcionalidad limitada o mensajes molestos, o una simple apelación a la moral del usuario, además de que las estipulaciones legales de la licencia podrían utilizarse en contra del infractor.

**OPEN SOURCE.-** Es el software que permite tener acceso a su código fuente a través de cualquier medio (ya sea acompañado con el programa ejecutable, a través de Internet, a través del abono de una suma de dinero, etc.) razón por la cual se comparte abiertamente entre desarrolladores y usuarios finales de tal forma que todos aprendan de todos.

### **1.2.2.1 Licencias Utilizadas en el Software Libre**

#### **Licencias Permisivas<sup>3</sup>**

Las licencias permisivas, a veces también llamadas liberales, son aquellas que vienen desde el autor con autorización de redistribuir y modificar así como para añadirle restricciones adicionales, incluso hacerlo no libre o propietario. Las licencias permisivas son bastante populares, y entre las más conocidas tenemos a las siguientes:

---

<sup>3</sup> Págs, 5-6; Software libre: licencias y propiedad intelectual; <http://sinetgy.org/~jgb/articulos/soft-libre-licencias-2004/soft-libre-licencias-2004.pdf>; Julio 2004

### **Licencia BSD (BERKELEY SOFTWARE DISTRIBUTION)**

Esta licencia exige como única obligación dar crédito a los autores, mientras que permite tanto la redistribución binaria y de los fuentes, aunque no obliga a ninguna de las dos en ningún caso. Así mismo se da permiso para realizar modificaciones y ser integrada con otros programas casi sin restricciones.

### **ZPL (ZOPE PUBLIC LICENSE 2.0 )**

Esta licencia es usada para la distribución de Zope (un servidor de aplicaciones) y otros productos relacionados. Es una licencia similar a la BSD, con el interesante detalle de prohibir expresamente el uso de marcas registradas por Zope Corporation.

### **Licencia de Apache**

Licencia bajo la que se distribuyen la mayor parte de los programas producidos por el proyecto Apache. Es similar a la licencia BSD.

### **Licencia de XWINDOW versión 11 (X11)**

Es una licencia muy similar a la licencia BSD, que permite redistribución, uso y modificación prácticamente sin restricciones. A veces, esta licencia es llamada "licencia MIT" (con poca precisión, porque el MIT ha usado otros tipos de licencias). Bajo esta licencia se distribuyen también trabajos derivados de X Windows, como XFree86.

## Licencia no Permisivas (ROBUSTAS)<sup>4</sup>

Las licencias robustas, llamadas habitualmente licencias copyleft tratan de garantizar las libertades que otorga el autor no sólo a quien recibe el programa directamente de él, sino también a quienes los reciben, más adelante después de la primera redistribución. Entre estas tenemos:

### GNU/GPL (LICENCIA PÚBLICA GENERAL GNU)

Esta es impulsada por la FSF la misma que otorga la cuatro libertades y estipula que al momento de existir una redistribución esta también estará en términos de la misma licencia es decir bajo licencia GNU/GLP, permitiendo que la redistribución continúe siendo modificada sin limitar ninguna de las cuatro libertades básicas de los usuarios de Software Libre.

Las principales características de esta licencia son las siguientes<sup>5</sup>:

- Se hace un uso muy interesante de la legislación sobre copyright es decir en lugar de limitar los derechos de los usuarios, los garantiza.
- Permite la redistribución binaria.
- Permite la redistribución fuente (obliga a ella en caso de redistribución binaria).
- Permite las modificaciones sin restricciones.
- Integración completa sólo con software cubierto por la GLP

La idea básica de esta licencia es utilizar la legislación actual de *copyright* para obligar a los que reciben un programa a cumplir ciertas condiciones. Esas condiciones están cuidadosamente diseñadas para conseguir que ese programa, o cualquier trabajo derivado de él, sigan siendo software libre. Desde este punto de vista, la GPL no sólo permite distribuir software

---

<sup>4</sup> Págs, 5-6; Software libre: licencias y propiedad intelectual; <http://sinetgy.org/~jgb/articulos/soft-libre-licencias-2004/soft-libre-licencias-2004.pdf>; Julio 2004.

<sup>5</sup> <http://pulsar.unizar.es/gluz/manual-sl/x490.html>; Historia y cultura del Software Libre.

libre, sino que se esfuerza para asegurarse de que a partir de él nunca se derivará software propietario.

### **LGPL (LICENCIA PÚBLICA GENERAL MENOR DE LA GNU)**

La LGPL permite el uso de programas libres con software propietario. El programa en sí se redistribuye como si estuviera bajo la licencia GPL, pero se permite la integración con cualquier otro software sin prácticamente limitaciones.

Sus características principales son las siguientes<sup>6</sup>:

- Está pensada para permitir el uso de bibliotecas libres con software propietario (por ejemplo, en el caso de un compilador).
- Se comporta como la GPL cuando se redistribuye la biblioteca como tal.
- Permite la integración con cualquier otro software. En este caso, no hay prácticamente limitaciones.

### **1.2.3 ¿QUÉ ES SOFTWARE DE CÓDIGO CERRADO O SOFTWARE COMERCIAL?**

Es aquel programa (o grupo de programas) que es entregado en formato binario o ejecutable. No se entrega el código fuente original, por tanto no es posible su modificación o mejora. Su copia y distribución está limitada por la licencia del fabricante, que usualmente tiene un costo monetario. El único derecho disponible para el usuario final es el de uso o ejecución del programa<sup>7</sup>.

Así como en el Software libre detallamos ciertos términos aquí también determinaremos algunos<sup>8</sup>:

---

<sup>6</sup> <http://pulsar.unizar.es/gluz/manual-sl/x505.html>; Historia y cultura del Software Libre

<sup>7</sup> <http://www.linux.org.pe/softwarelibre/pronuncia.php>; Pronunciamiento PLUG sobre el Proyecto de Ley de Software Libre ; Mayo 2002

<sup>8</sup> <http://www.wilkinsonpc.com.co/free/articulos.html>; Que es, formas y tipos de licenciamiento de software; Abril 2004

LICENCIA.- Es una especie de contrato, en donde se especifican todas las normas y cláusulas que rigen el uso de un determinado programa, principalmente se estipulan los alcances de uso, instalación, reproducción y copia de estos productos.

LICENCIAR UN SOFTWARE.- "El procedimiento de conceder a otra persona o entidad el derecho de usar un software con fines industriales, comerciales o personales, de acuerdo a las cláusulas que en ella aparecen."

PROTECCIÓN COPYRIGHT.- Esto hace referencia a los derechos de autor (o de copia). El usuario no puede adquirirlo para usarlo y luego vender copias (salvo con la autorización de su creador).

#### 1.2.4 SOFTWARE LIBRE VS SOFTWARE COMERCIAL O PROPIETARIO

En este punto presentaremos un cuadro comparativo, el mismo que nos permitirá ver las debilidades y fortalezas del Software Libre frente al Software Propietario.

Aspectos por Considerar	Software Propietario	Software Libre (GPL)
Acceso al código fuente	Prohibido por licencia	Si, Garantizado
Corrección de errores en el programa por el cliente/usuario	No	Si
Duplicación del software	Prohibido	Posible y Recomendada
Libertad de Competencia para el mantenimiento	No, depende del fabricante	Si imposible limitarla
Posibilidad de examinar el código del producto	Prohibido, salvo permiso fabricante	Si
Respeto a estándares globales	En función del fabricante	En la mayoría de los casos
Adaptaciones al cliente	En función del fabricante	Disponible
Virus, gusanos	Frecuente	Poco frecuente

Costos de licencias	Altos	Bajos
Garantía proveniente del autor	Si	No

Tabla 1.2: Cuadro comparativo software libre vs. Software propietario.

### 1.2.5 COSTO DEL CAMBIO AL UTILIZAR SOFTWARE LIBRE

El realizar un cambio de un esquema de Software Comercial a uno de Software Libre no quiere decir que implique un costo cero. Debemos tener en cuenta los siguientes aspectos para llevar a cabo un cambio:

- a. Capacitación: En este punto los costos crecen ya que se debe dedicar un tiempo considerablemente largo para el aprendizaje de los diferentes usuarios (desarrolladores, administradores, usuarios finales )
- b. Licencias: Ya que el Software Libre otorga la licencia de distribuir copias se puede comprar una sola licencia y no tantas como computadoras se posee, esto hace que los costos reduzcan más no quiere decir que se hagan cero.
- c. Soporte: Debido a que el soporte de las aplicaciones realizadas en Software Libre las darán algunos de los usuarios que recibieron la capacitación de aprendizaje los costes de soporte disminuyen significativamente.

## 1.3 REINGENIERÍA

Reingeniería de Software es una forma de modernización para mejorar las capacidades y/o mantenibilidad de los sistemas de información heredados mediante la aplicación de tecnologías y prácticas modernas. La Reingeniería de Software ofrece una disciplina de preparación para migrar un sistema de

información heredado hacia un sistema evolucionable. El proceso aplica principios de ingeniería para un sistema existente para encontrar nuevos requerimientos.

El propósito de la reingeniería es que los sistemas existentes tomen ventajas de las nuevas tecnologías y habilitar el nuevo esfuerzo de desarrollo para que aproveche las ventajas de reutilizar sistemas existentes. La reingeniería tiene el potencial de mejorar la productividad y calidad del software a través de todo el ciclo de vida.

Para llevar a cabo una Reingeniería debemos tener en cuenta ciertos puntos para el estudio de una aplicación los cuales son:

- Tener en cuenta si es aconsejable invertir tiempo en un sistema antiguo, ya que este puede ser muy confuso, desorganizado, etc. y sea preferible hacer uno nuevo.
- Si se va a tener una sustitución inmediata del sistema se debe de valorar el grado de esfuerzo adecuado.
- Considerar un método adecuado para no restaurar una aplicación cometiendo los mismos errores (falta de documentación, mala arquitectura, etc.).
- Puede reducir los riesgos evolutivos de una organización
- Puede ayudar a las organizaciones a recuperar sus inversiones en software

Es recomendable antes de llevar a cabo una reconstrucción de un sistema analizar las siguientes alternativas disponibles y escoger la que mejor relación coste/beneficio ofrezca:

- Dejar el producto como está.
- Adquirir uno en el mercado que realice la misma función.
- Reconstruirlo.

### 1.3.1 MODELO DE PROCESO DE REINGENIERIA DE SOFTWARE<sup>9</sup>

Para aplicar un proceso de reingeniería es necesario comprender perfectamente como funciona el sistema, para esto se define seis actividades (Figura 1.1) estructuradas dentro de un modelo circular las mismas que se pueden visitar en repetidas ocasiones pudiendo finalizar en cualquiera de ellas.

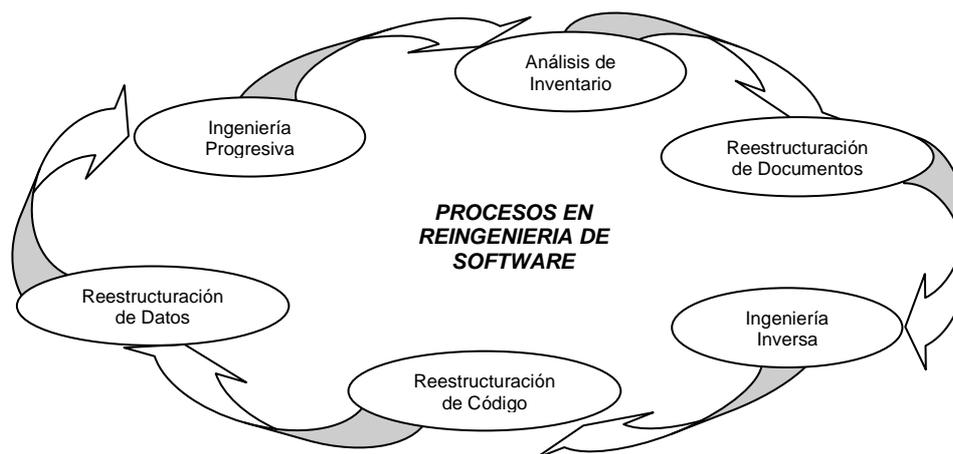


Figura 1.1 Procesos en reingeniería de software

#### 1.3.1.1 Análisis de Inventario

En esta primera actividad se realiza un inventario de todas las aplicaciones disponibles ordenando la información de acuerdo a su antigüedad, importancia en el negocio, mantenibilidad actual y otros criterios.

En el inventario se pueden considerar los siguientes datos: Nombre de la aplicación, Año en que se creó, Cambios efectuados, fecha y valoración del esfuerzo, Lenguaje y posible sistema donde instalarlo, Aplicaciones con las cuales tiene relación, Errores detectados, Valoración de la complejidad, Arquitectura del programa, Código, Documentación, Calidad de la documentación, Longevidad del

<sup>9</sup> Págs 4-5; <http://www.elrinconcito.com/articulos/Reingenieria/ReingSoftySistemas.pdf>; Reingeniería de Software, Reingeniería de Sistemas; Romero Danille;

proyecto, Número estimado de cambios, Tiempo estimado de los cambios, entre otras.

Es aconsejable que toda esta información sea correcta y actualizada en todo momento.

#### **1.3.1.2 Reestructuración de Documentos**

Esta actividad se enfoca a documentar todo el sistema ya que este es fundamental para el negocio pero siempre reduciendo al mínimo la documentación y además se debe tener presente que dicha documentación debe estar actualizada al realizar cambios.

#### **1.3.1.3 Ingeniería Inversa**

En esta actividad se tiene como objetivo el proceso de recuperación de diseño; se extrae información acerca de los datos de arquitectura y diseño de procedimientos del programa existente.

#### **1.3.1.4 Reestructuración del Código**

Esta actividad consiste en realizar un análisis del código fuente ya sea de una manera estática o dinámica, en ambas se podrá detectar si tiene defectos o errores el código. Este análisis también deberá ser documentado esto con el fin de mejorar la calidad del software.

#### **1.3.1.5 Reestructuración de Datos**

Para llevar a cabo esta actividad en la mayoría de casos es necesario un proceso previo de ingeniería inversa. El objetivo de esta actividad es realizar un análisis para extraer elementos y objetos de datos, para obtener información acerca del flujo de datos, así como comprender las estructuras de datos ya existentes. Para ello es necesario estudiar el código fuente analizando las sentencias que

contengan definiciones de datos, descripciones de archivos, estructuras de I/O y descripciones de interfaz. Luego de esto se debe clarificar las definiciones ajustándolas a los estándares de la organización para obtener consistencia en los nombres, formatos de registros, ficheros, etc.

### 1.3.1.6 Ingeniería Progresiva

La ingeniería progresiva utiliza la información recuperada en la ingeniería inversa para reconstruir el sistema existente en esfuerzo con vistas a la calidad, incorporándose, generalmente, nuevas funcionalidades al sistema.

## 1.4 METODOLOGIA DE DESARROLLO DE SOFTWARE

Para el presente proyecto se utilizará el ciclo de Reingeniería propuesto por Rogger Pressman y para el proceso de Ingeniería Progresiva se utilizará el UP lo que se resume en el siguiente cuadro:

<b>Fases</b>	<b>Actividades</b>
Inventario	Recopilar Información de la versión actual Evaluación de la Información
Reconstrucción de Documentos	En base a la documentación existente
Ingeniería Inversa	Para comprender los datos
	Para comprender el proceso
	De Interfaces de usuarios
Reestructuración de Código	Análisis del Código (obtener algoritmos ya establecidos)
Reestructuración de Datos	Clarificar las definiciones ajustando a estándares

Ingeniería Progresiva (De acuerdo a UP)	Flujo de Trabajo	Actividad	Entregable
	Requisitos	Modelo del Negocio	✓
		Modelo del Dominio	
		Modelo de Casos de Uso	✓
	Análisis	Modelo de Clases	
		Modelo de Colaboración	✓
	Diseño	Modelo de Casos de Uso de Diseño	✓
		Modelo de Secuencia	✓
		Modelo de Componentes	✓
	Construcción y Pruebas	Codificación y Pruebas	✓
Manual de Programación		✓	
Manual de Usuario		✓	

Tabla 1.3: Metodología de desarrollo de software.

## **CAPÍTULO 2: REINGENIERIA DEL MÓDULO PRINCIPAL DEL SISTEMA SAE**

### **2.1 INVENTARIO**

Para la elaboración del inventario se tuvo como fuente de información entrevistas realizadas a personas relacionadas con el desarrollo y gestión del sistema SAE, de lo que se obtuvo los siguientes datos:

#### **2.1.1 NOMBRE Y OBJETIVO**

Sistema de Administración de Estudiantes “SAE”

El SAE tiene sus primeros pasos desde el año 1976, pero fue en 1986 que se conforma un grupo de desarrollo cuyo líder fue el Ing. Cristóbal Trujillo, un programador y la Ing. Nidia Guayaquil ellos fueron los pioneros de dicho sistema que ha tenido varias versiones desde 1987 hasta 1999 que se obtuvo a la versión SAE2000 que es la actual aplicación que se utiliza en la EPN.

El objetivo de dicha aplicación era la de ser un sistema certificado para el 2000 en un ambiente cliente-servidor con interfaz gráfica amigable, un adecuado administrador de base de datos y además ser un sistema de gestión universitaria para ser comercializado a otras instituciones afines.

En Abril de 2006 fue aprobada la construcción e implementación del SAEIntegrado el cual tiene como líderes de proyecto el Ing. Cristóbal Trujillo y la Ing. Nidia Guayaquil, el objetivo de la aplicación es integrar la base de datos de las carreras de la EPN y cambiar el ambiente de la aplicación a un entorno Web el cual está en un proceso de desarrollo e implementación con herramientas comerciales.

### **2.1.2 FUNCIONALIDAD**

El SAE es un sistema encargado de realizar la administración de la información académica de los estudiantes de la EPN.

El SAE2000 posee vínculos con otros sistemas o pequeñas aplicaciones que le permite cargar información importante para llevar a cabo sus procesos.

El SAE2000 opera de una manera descentralizada ya que cada carrera posee su propia Base de Datos lo que es una gran desventaja debido a que la información no se maneja de forma integrada lo cual puede causar inconsistencias al momento de obtener información.

El SAE2000 consta de los siguientes módulos:

- Planificación Académica
- Administración Estudiantil
- Evaluación Académica
- Evaluación Docente
- Control de Tesis
- Información Gerencial Académica
- Información Estudiantil
- Información Docente

De los módulos que conforman el SAE2000 2 de ellos están completamente terminados, 3 están incompletos y 3 falta de realizar por esta razón la institución ha decidido hacer un nuevo proyecto el cual es el SAEIntegrado.

El SAEIntegrado es un proyecto en ejecución que permitirá la integración de las bases de datos de 67 carreras de la Escuela Politécnica Nacional el proyecto se ejecutará en 3 semestres de la siguiente manera:

1er Semestre: Los módulos mencionados a continuación se encuentran en una etapa de desarrollo y pruebas.

- SAE-ACADÉMICO:
  - Planificación
  - Registro y Matrícula
  - Evaluación Estudiantil
- SAE-ADMISIÓN
- SAE-BIENESTAR ESTUDIANTIL

2do y 3er Semestre

- SAE-TESIS Y PROYECTOS DE TITULACIÓN
- SAE-EVALUACIÓN Y ESCALAFÓN DOCENTE:
  - Evaluación del Desempeño Académico
  - Control de Proyectos
  - Evaluación y ascensos
- SAE-AUDITOR
  - Auditoria de Datos
  - Auditoria Académica
- SAE-GERENCIAL
- SAE-SERVICIOS DE INFORMACIÓN
- SAE-GESTIÓN DE USUARIOS y CONFIGURACIÓN

### **2.1.3 PLATAFORMA Y METODOLOGÍA**

El SAE2000 se encuentra realizado en Visual Basic 6.0 y SQL Server 7.0 sobre plataforma Windows 2000 Server y metodología es orientada objetos.

El SAEIntegrado esta siendo desarrollado en Visual .NET y SQL Server 2005 sobre plataforma Windows y metodología orientada a objetos.

#### **2.1.4 DOCUMENTACIÓN**

En base a la información obtenida acerca del SAE2000 se llegó a determinar que no existe documentación del sistema a excepción del modelo entidad relación de la base de datos que se encuentra realizado en Power Designer.

El SAEIntegrado posee información acerca de los requerimientos que son implementados en los módulos:

- SAEPlanificación
- SAEAdmisión
- SAEMatrículas
- SAECalificación

Los módulos mencionados son aquellos establecidos en el 1er semestre los cuales se encuentran en etapa de desarrollo y pruebas.

#### **2.1.5 EQUIPO DE DESARROLLO**

Equipo de desarrollo SAE2000

Líder: Ing. Cristóbal Trujillo  
Ing. Nidia Guayaquil  
Programador: Eduardo Romero

Equipo de desarrollo SAEIntegrado

Líderes: Ing. Cristóbal Trujillo y Ing. Nidia Guayaquil  
Grupo de programadores contratados

## 2.1.6 PROBLEMAS

Problemas SAE2000:

- Uno de los problemas significativos que tiene el SAE2000 es que la base de datos no esta centralizada para la Escuela Politécnica Nacional ya que cada carrera maneja su propia base de datos.
- Separación y no coordinación del grupo de desarrollo y el grupo de gestión del sistema SAE2000.

Problemas SAEIntegrado:

- En la actualidad no se puede establecer si existe problemas en el sistema SAEIntegrado ya que los módulos especificados en el 1er semestre se encuentra en etapa de desarrollo y pruebas, además de la integración de la base de datos.

Se realizó una evaluación de los factores internos y externos del SAE2000 utilizando ciertos índices.

- **Factor Externo**

Hardware

El hardware se encuentra disperso en las instalaciones de cada una de las 67 carrera de la EPN por esta razón tiene características diferentes los equipos utilizados depende de la administración de la carrera.

Software

El software también depende de la administración de cada carrera no se maneja estándares con respecto a éste los equipos tienen como

software mínimo el software utilizado por la aplicación SAE2000 que es Visual Basic 6.0 y SQL Server 7.0 sobre plataforma Windows.

- **Factor Interno**

Sistema SAE2000

Aspecto	Puntaje(1- 4)	Referencia	Razón
Tamaño (# líneas de código)	2	<ol style="list-style-type: none"> <li>1. Muy Grande</li> <li>2. Grande</li> <li>3. Mediano</li> <li>4. Pequeño</li> </ol>	El SAE2000 maneja dos módulos de los cuales tiene aproximadamente 50000 líneas de código.
Complejidad	3	<ol style="list-style-type: none"> <li>1. Muy Alta</li> <li>2. Alta</li> <li>3. Mediana</li> <li>4. Simple</li> </ol>	No es necesario mayor capacitación ya que la aplicación es utilizada por personas con experiencia en los procesos implementados en el sistema
Documentación	2	<ol style="list-style-type: none"> <li>1. No existe</li> <li>2. Mínima</li> <li>3. Adecuada</li> <li>4. Buena</li> </ol>	No existe ningún tipo de documentación acerca del sistema SAE2000 a excepción del modelo de datos en PowerDesigner.
Datos (Confiabilidad de la BDD)	3	<ol style="list-style-type: none"> <li>1. Nada confiable</li> <li>2. Poco confiable</li> <li>3. Confiable</li> <li>4. Muy confiable</li> </ol>	A pesar de que la información se encuentra separada en bases de datos diferentes existen administradores de base de datos encargados para las 67 carreras de la EPN.
Datos (Consistencia de	1	<ol style="list-style-type: none"> <li>1. Nada confiable</li> <li>2. Poco confiable</li> </ol>	Por la gestión de datos que se hace en cada carrera.

información desplegada)		<ol style="list-style-type: none"> <li>3. Confiable</li> <li>4. Muy confiable</li> </ol>	
Ambiente de Pruebas	3	<ol style="list-style-type: none"> <li>1. No existe</li> <li>2. Realizada considerando errores leves</li> <li>3. Realizada considerando errores principales</li> <li>4. Realizada considerando todos los errores en la aplicación</li> </ol>	Se realizaron pruebas al SAE2000 y se puso énfasis en la corrección de los errores principales dentro de la aplicación.
Registro de Mantenimiento	1	<ol style="list-style-type: none"> <li>1. No existe</li> <li>2. Poco frecuente</li> <li>3. Frecuente</li> <li>4. Muy frecuente</li> </ol>	No se lleva un registro de las actualizaciones realizadas ya que las modificaciones en el sistema dependen de la persona en cargada en cada carrera.
Requiere soporte técnico	3	<ol style="list-style-type: none"> <li>1. Siempre</li> <li>2. Casi siempre</li> <li>3. Rara vez</li> <li>4. Nunca</li> </ol>	Ya que el usuario no puede crear ni cerrar semestres.
Desempeño	3	<ol style="list-style-type: none"> <li>1. Quejas Frecuentes</li> <li>2. Quejas Regulares</li> <li>3. Quejas Ocasionales</li> <li>4. No hay Quejas</li> </ol>	Ya que los coordinadores de carrera solicitan requerimientos personalizados.
Seguridad	2	<ol style="list-style-type: none"> <li>1. Mala</li> <li>2. Regular</li> <li>3. Buena</li> <li>4. Muy buena</li> </ol>	El modulo de administración de perfiles no fue desarrollado en su totalidad.
Interfaces	2	<ol style="list-style-type: none"> <li>1. Nada amigable</li> <li>2. Poco amigable</li> <li>3. Amigable</li> <li>4. Muy amigable</li> </ol>	Poco intuitiva para personas que no requieren conocimiento de los procesos que se realizan en

			la aplicación.
Susceptible al Cambio (Código)	3	<ol style="list-style-type: none"> <li>1. No susceptible</li> <li>2. Poco susceptible</li> <li>3. Susceptible</li> <li>4. Muy susceptible</li> </ol>	Se utiliza procedimientos almacenados a nivel de base de datos y librerías a nivel de código fuente los cuales se han ido depurando con la evolución de la aplicación.
Tiempo de respuestas	2	<ol style="list-style-type: none"> <li>1. Malos</li> <li>2. Regulares</li> <li>3. Buenos</li> <li>4. Excelente</li> </ol>	La aplicación en el periodo que es más utilizada los tiempos de respuestas son muy lentos, aun que depende del Hw de cada carrera.
Errores	3	<ol style="list-style-type: none"> <li>1. Frecuente</li> <li>2. No tan frecuente</li> <li>3. Rara vez</li> <li>4. Ninguno</li> </ol>	Presenta errores leves los cuales dependen de la utilización y administración de la aplicación en cada carrera.
Integración con otros Sistemas	1	<ol style="list-style-type: none"> <li>1. Ninguno sistema integrado</li> <li>2. Muy pocos sistemas integrados</li> <li>3. Pocos sistemas integrados</li> <li>4. Muchos sistemas integrados</li> </ol>	La aplicación SAE2000 utiliza otros sistemas relacionados para cálculos de costos y pagos de matricula pero no se encuentran integrados a la aplicación.
Importancia en el negocio	4	<ol style="list-style-type: none"> <li>1. Nada importante</li> <li>2. Poco importante</li> <li>3. Considerable</li> <li>4. Altamente importante</li> </ol>	Administración de información indispensable para la gestión académica de la EPN durante todos los años de funcionamiento.
Respaldos de	3	<ol style="list-style-type: none"> <li>1. Ninguno</li> </ol>	Se obtiene respaldos cuando

información		<ol style="list-style-type: none"> <li>2. Regular</li> <li>3. No tan frecuente</li> <li>4. Frecuente</li> </ol>	<p>existe gran cantidad de procesamiento de información (Periodos picos de utilización de la aplicación) lo que es responsabilidad de las personas encargadas de la administración de la base de datos en UGI.</p>
Acceso a la aplicación	2	<ol style="list-style-type: none"> <li>1. Muy compleja</li> <li>2. Compleja</li> <li>3. Mediana</li> <li>4. Simple</li> </ol>	<p>Ya que a la aplicación se tiene acceso mediante un ambiente cliente/servidor en una intranet es decir solo en el Campus de la EPN.</p>
<b>Máximo</b>	<b>76</b>		
<b>Mínimo</b>	<b>38</b>		
<b>Total</b>	<b>43/76</b>		

Tabla 2.1: Factor Interno

Después de la evaluación realizada al factor interno de la aplicación se puede observar que el puntaje obtenido es bajo de lo que se puede determinar que la aplicación no esta cumpliendo con las expectativas del usuario, esto puede justificarse con el hecho de que el SAE ha ido cambiando a distintas versiones en las cuales no se mantuvo los requerimientos principales planteados al inicio o que se dieron enfoques distintos durante los cambios de la aplicación además de que la aplicación ha ido cambiando de acuerdo a los requerimientos personalizados en cada carrera.

De lo que se concluye que la aplicación SAE requiere Reingeniería en cuanto a la documentación, interfaz y al proceso de datos, con el fin de obtener una aplicación que pueda ser mantenible, que tenga un buen desempeño, que la información y la aplicación sea administrada de forma centralizada, que el acceso

a la aplicación pueda realizarse de manera remota es decir poner la aplicación sobre un ambiente Web.

## 2.2 INGENIERÍA INVERSA

Para nuestro caso no se llevo a cabo la ingeniería inversa debido que se realizo el Modulo de Registro Estudiantil en otro lenguaje de programación (JAVA).

## 2.3 INGENIERÍA PROGRESIVA

### 2.3.1 REQUERIMIENTOS

Los requerimientos fueron tomados del proyecto SAEIntegrado el cual fue aprobado en Abril de 2006.

#### 2.3.1.1 Modelo de Caso de Uso del Negocio

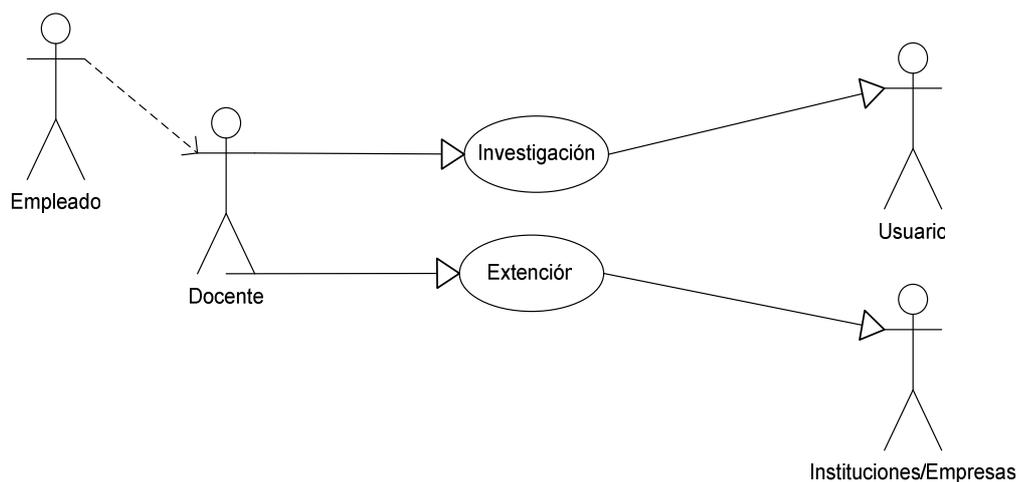


Figura 2.1: Modelo de Caso de Uso del Negocio.

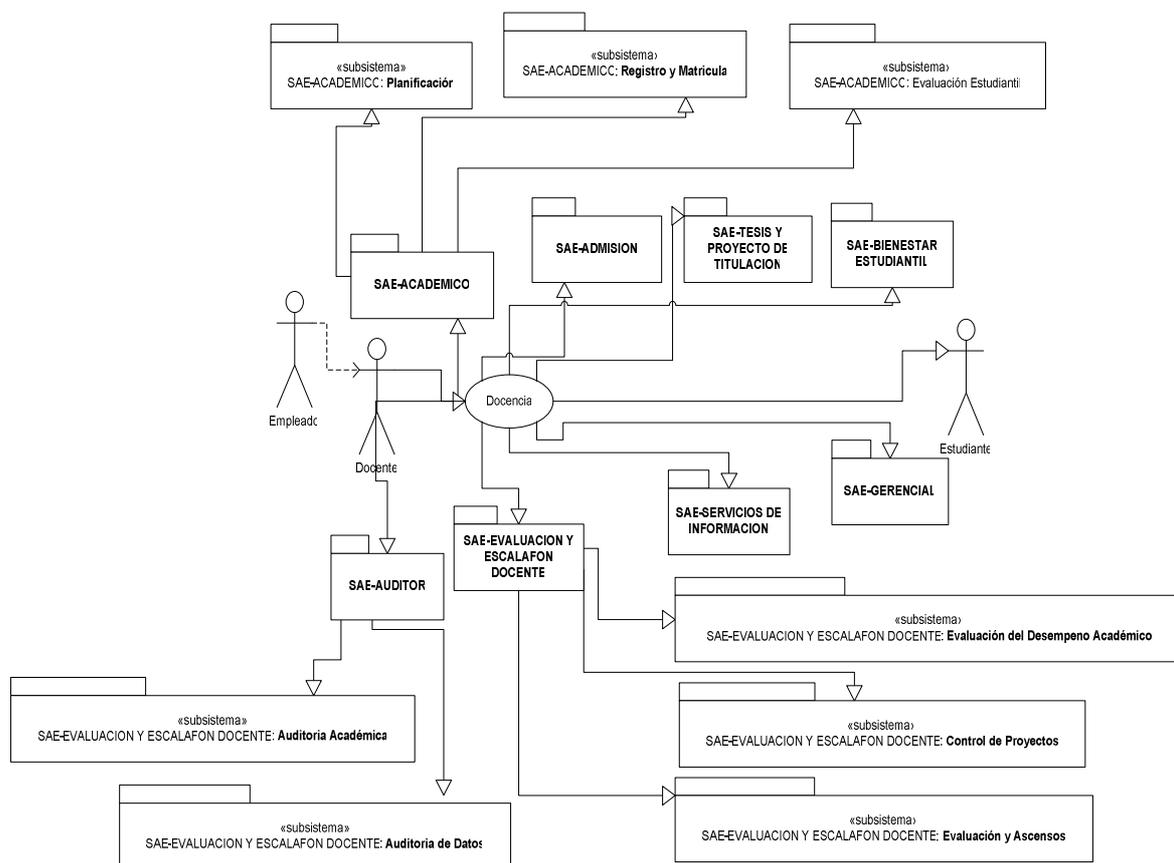


Figura 2.1: Modelo de Caso de Uso del Negocio.

### 2.3.1.2 Casos de Uso

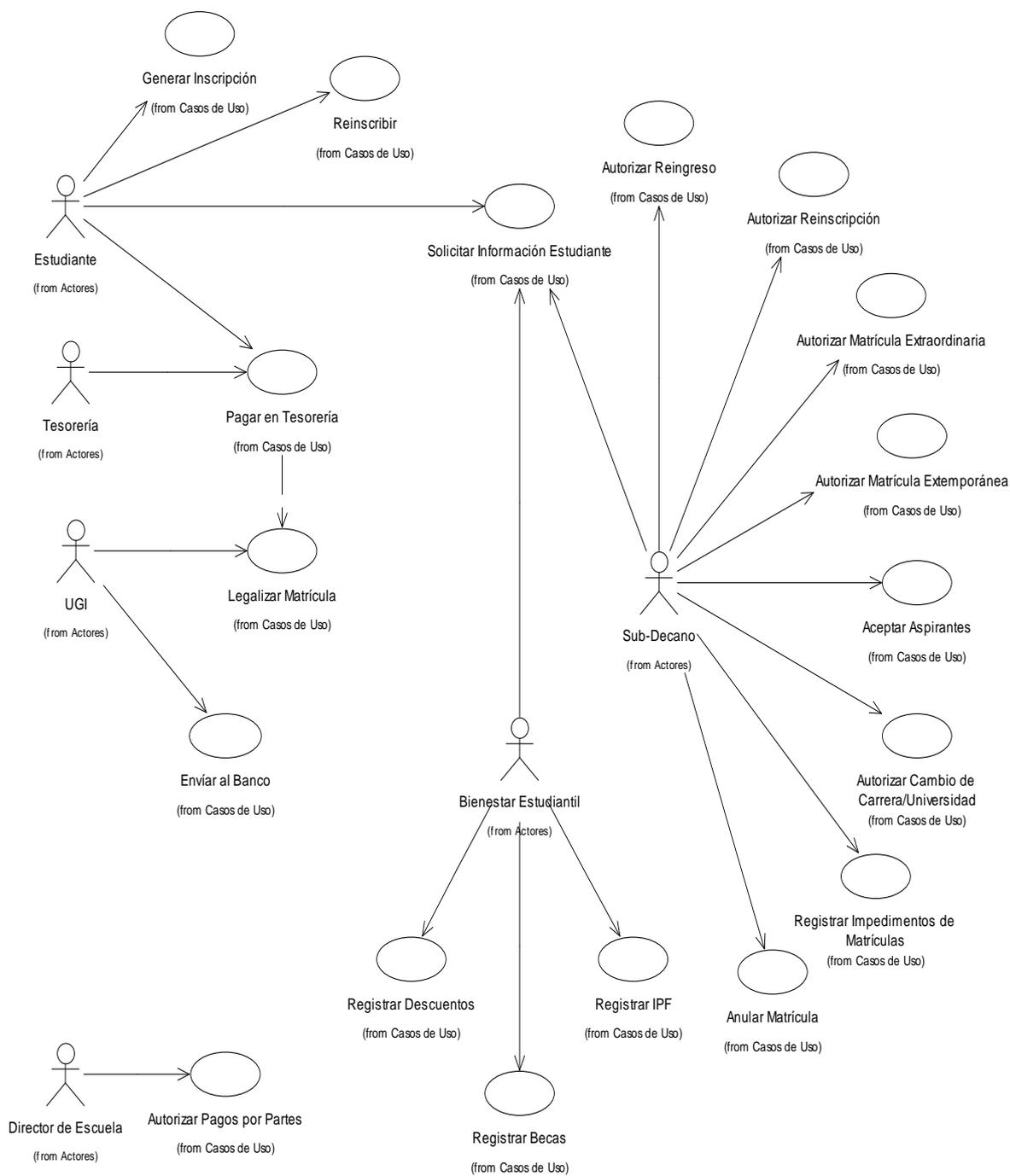


Figura 2.2: Casos de Uso del Modulo de Registro Estudiantil.

2.3.1.2.1 *Lista y descripción de actores*

<b>Actor</b>	<b>Descripción</b>
<b>Subdecano Carrera</b>	<ul style="list-style-type: none"> <li>• Autorizar reinscripciones, reingresos, cambios de carrera, matrículas extemporáneas y extraordinarias.</li> <li>• Aceptar aspirantes.</li> <li>• Ingresar los rubros adicionales.</li> <li>• Actualizar el currículum académico de un estudiante para revalidaciones, equiparaciones de materias y aprobación de inglés.</li> <li>• Ingresar los impedimentos.</li> <li>• Realizar la reinscripción de materias.</li> </ul>
<b>Estudiante</b>	<ul style="list-style-type: none"> <li>• Visualizar la información personal y académica, además podrá modificar algunos campos personales (dirección, teléfono, estado civil, e-mail).</li> <li>• Realizar el proceso de inscripción y anulación de la matrícula.</li> </ul>
<b>Bienestar Estudiantil</b>	<ul style="list-style-type: none"> <li>• Visualizar información personal y académica del estudiante.</li> <li>• Registrar o actualizar datos socioeconómicos.</li> <li>• Registrar descuentos del estudiante.</li> <li>• Registrar becas del estudiante.</li> <li>• Registrar los pagos por rol.</li> <li>• Registrar IPF.</li> </ul>
<b>Unidad de Gestión de Información (UGI)</b>	<ul style="list-style-type: none"> <li>• Enviar el archivo de inscripción de estudiantes con el respectivo valor a pagar por matrícula al banco.</li> <li>• Realizar la legalización de las matrículas.</li> </ul>
<b>Director de Escuela</b>	<ul style="list-style-type: none"> <li>• Autorizar los pagos en partes.</li> <li>• Autorizar los pagos con crédito IECE.</li> </ul>

<b>Tesorería</b>	<ul style="list-style-type: none"> <li>• Registrar y legalizar el pago de la matrícula de los estudiantes que realizan su pago por: cheque, tarjeta de crédito, o IECE.</li> <li>• Emitir comprobantes de pago de matrículas.</li> </ul>
------------------	--

Tabla 2.2: Lista y descripción de actores.

### 2.3.1.2.2 Lista y funciones de los casos de uso

<b>Nro</b>	<b>Caso de Uso</b>	<b>Función</b>
CU - 01	Solicitar Información Estudiante	<ul style="list-style-type: none"> <li>• Despliega Datos Personales</li> <li>• Despliega Currículum Académico</li> <li>• Despliega Créditos por Categoría</li> <li>• Despliega Malla Curricular</li> <li>• Despliega Horarios</li> <li>• Despliega Calificaciones</li> <li>• Despliega Períodos Matriculados.</li> </ul>
CU - 02	Generar inscripción	<ul style="list-style-type: none"> <li>• Permitir al estudiante inscribirse en las materias que desee tomar durante un período.</li> </ul>
CU - 03	Reinscribir	<ul style="list-style-type: none"> <li>• Permite que el estudiante añada y retire materias.</li> </ul>
CU - 04	Anular matrícula	<ul style="list-style-type: none"> <li>• Realizar la anulación de la matrícula de un estudiante.</li> </ul>
CU - 05	Enviar al banco	<ul style="list-style-type: none"> <li>• Generar un archivo plano, con el detalle del valor a ser cobrado a los estudiantes por concepto de su matrícula.</li> </ul>
CU - 06	Pagar en tesorería	<ul style="list-style-type: none"> <li>• Realizar el pago y la legalización de la matrícula de los estudiantes que hayan especificado en su inscripción o reinscripción las siguientes formas de</li> </ul>

		pago: cheque, tarjeta de crédito o IECE.
CU - 07	Legalizar matrícula	<ul style="list-style-type: none"> <li>• Legalizar la matrícula de los estudiantes que hayan cancelado el valor de su matrícula en el Banco.</li> </ul>
CU - 08	Autorizar reinscripción	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante para la reinscripción de materias.</li> </ul>
CU - 09	Autorizar reingresos	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante para que reingrese a la Carrera.</li> </ul>
CU - 10	Autorizar matrículas extraordinarias	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante para que se inscriba en las materias que desee tomar.</li> </ul>
CU - 11	Autorizar matrículas Extemporáneas	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante para que se inscriba en matrículas extemporáneas.</li> </ul>
CU - 12	Autorizar cambios de Carrera	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante de otra Carrera de la EPN, para que forme parte de la Carrera.</li> </ul>
CU - 13	Registrar impedimentos de matrículas	<ul style="list-style-type: none"> <li>• Registrar las sanciones a los estudiantes que tienen algún impedimento para matricularse.</li> </ul>
CU - 14	Registrar aspirantes	<ul style="list-style-type: none"> <li>• Registrar a los estudiantes que hayan aprobado propedéutico para que sean aspirantes a la carrera.</li> </ul>
CU - 15	Gestionar rubros adicionales	<ul style="list-style-type: none"> <li>• Registrar rubros adicionales que deben pagar los estudiantes en la matrícula.</li> </ul>
CU - 16	Autorizar pagos por partes	<ul style="list-style-type: none"> <li>• Autorizar a los estudiantes para que realicen su pago de matrícula en partes.</li> </ul>
CU - 17	Autorizar pagos con IECE	<ul style="list-style-type: none"> <li>• Autorizar a los estudiantes para que realicen su pago de matrícula con crédito IECE.</li> </ul>
CU - 18	Registrar IPF	<ul style="list-style-type: none"> <li>• Registrar y Actualizar los valores de</li> </ul>

		IPF (Ingreso Percapital Familiar).
CU - 19	Registrar descuentos	<ul style="list-style-type: none"> <li>Registrar el porcentaje de descuento que tiene un estudiante por los siguientes conceptos: hermanos, cónyuge, e hijos de empleados de la EPN.</li> </ul>
CU - 20	Registrar becas	<ul style="list-style-type: none"> <li>Registrar el porcentaje de descuento que tiene por Beca un estudiante por merito académico, deportivo o por situación económica.</li> </ul>

Tabla 2.3: Lista y funciones de los casos de uso

### 2.3.1.2.3 Descripción de casos de uso

<b># Caso de Uso:</b>	CU – 01
<b>Nombre:</b>	Solicitar Información Estudiante
<b>Función:</b>	El sistema le permitirá al Estudiante, Subdecano de Carrera o Unidad de Bienestar Estudiantil visualizar información personal o académica de un estudiante.
<b>Actor:</b>	Estudiante, Subdecano, Bienestar Estudiantil.
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Escenario Nro1.</b>	Datos Personales
<b>Eventos:</b>	
<b>Camino Básico:</b>	1. El sistema presenta los datos personales del estudiante que consiste en: nombre, código único, cédula, cédula militar, estado civil, sexo, fecha de nacimiento, fecha de inscripción, número de

	matrícula, estado académico, datos de procedencia como país, provincia, ciudad, datos de procedencia educativa como institución educativa, descripción del colegio, país, ciudad, provincia, título y contactos como dirección, teléfono, e-mail.
<b>Escenario Nro2.</b>	Currículum Académico
<b>Camino Básico:</b>	<ol style="list-style-type: none"> <li>1. El Sistema presenta el currículum académico del estudiante que consiste de: Período, semestre, código materia, nombre de materia, número de matrícula, calificación, numero de créditos de cada materia, estado de aprobación, nombre de la categoría, total de materias aprobadas, créditos acumulados, promedio, IRA y porcentaje de carrera y de materias aprobadas.</li> <li>2. El Subdecano de Carrera puede actualizar el currículum académico del estudiante para revalidaciones, equiparaciones de materias y aprobación de inglés.</li> <li>3. El sistema verificará que los campos modificables no queden vacíos y que cumplan con el estatuto académico.</li> <li>4. El sistema guardará la información.</li> </ol>
<b>Escenario Nro3.</b>	Créditos por Categoría
<b>Camino Básico:</b>	<ol style="list-style-type: none"> <li>1. El sistema presentará una lista que contiene: información general como carrera, pensum, fecha de ingreso, última matrícula y además una tabla con la categoría, referencia, número de créditos aprobados, número de créditos por aprobar y el porcentaje total de créditos aprobados.</li> </ol>
<b>Escenario Nro4.</b>	Malla Curricular
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema presenta la malla curricular considerando el pensum, en la cual se resaltará las materias</li> </ol>

	aprobadas de las no aprobadas
<b>Escenario Nro5.</b>	Horarios
<b>Camino Básico:</b>	1. El sistema presenta los horarios de las materias del periodo actual matriculado, esta información consiste de número de materia, código, paralelo, nombre de la materia, número de matrícula, créditos de la materia, y horario para cada uno de los días laborables y el día sábado.
<b>Escenario Nro6.</b>	Calificaciones
<b>Camino Básico:</b>	1. El sistema presenta al estudiante la información de las calificaciones parciales y totales de cada una de las materias del período actual. Las calificaciones se encuentran compuestas por código materia, nombre de materia, paralelo, número de matrícula, nota1, nota2, nota3, total y estado de aprobación. (aprobado, exonerado, supletorio).
<b>Escenario Nro7.</b>	Periodos Matriculados
<b>Camino Básico</b>	1. El sistema presenta una lista de los periodos matriculados por el estudiante en la carrera.
<b>Poscondiciones:</b>	<ul style="list-style-type: none"> <li>• La información estará disponible para nuevas consultas.</li> </ul>

Tabla 2.4: CU – 01 Solicitar información estudiante.

<b>#Caso de Uso:</b>	CU – 02
<b>Nombre:</b>	Generar Inscripción
<b>Función:</b>	Permitir al estudiante inscribirse en las materias que desee tomar durante un período.
<b>Actor:</b>	Estudiante
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El estudiante debe estar matriculado el período inmediato anterior o debe tener la autorización de reingreso</li> </ul>

	<ul style="list-style-type: none"> <li>• No poseer ningún tipo de impedimentos.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta al estudiante la lista de materias que podrá tomar de acuerdo a las materias aprobados y el pensum.</li> <li>2. El estudiante debe seleccionar la materia que se desea tomar.</li> <li>3. El sistema despliega la lista de materia.</li> <li>4. El estudiante debe seleccionar el paralelo que tenga el horario preferido.</li> <li>5. El sistema validará, cruce de horarios entre las materias inscritas y el número mínimo y máximo de créditos a tomar (tomando en cuenta 2 y 3 matrícula).</li> <li>6. Si el estudiante desea eliminar una materia o cambiar el grupo se deberá realizar las siguientes acciones: <ol style="list-style-type: none"> <li>6.1. Eliminar materia <ol style="list-style-type: none"> <li>6.1.1. Seleccionar la materia que desea eliminar, de la lista de materias inscritas.</li> <li>6.1.2. El sistema advertirá al estudiante que se eliminará los créditos de la materia antes seleccionada.</li> <li>6.1.3. El estudiante debe aceptar la advertencia.</li> <li>6.1.4. El sistema borra la materia antes seleccionada de la lista de materias inscritas.</li> <li>6.1.5. En caso de querer anular más materias regresar a 6.1.1.</li> </ol> </li> </ol> </li> <li>7. El estudiante cierra la inscripción.</li> <li>8. El sistema le presenta al estudiante el detalle de las materias inscritas (código, grupo, paralelo, número de créditos, aulas y horarios de clases), y el detalle del valor de la matrícula (valor matrícula, valor por créditos, saldos a favor, saldos en contra,</li> </ol>

	<p>descuentos, rubros adicionales. En caso de matrículas extraordinarias o extemporáneas, el recargo se verá reflejado en el valor de la matrícula y créditos.</p> <p>9. El estudiante selecciona la forma de pago y cerrará la inscripción.</p> <p>10. El sistema despliega un mensaje advirtiéndole al estudiante que los datos no podrán ser modificados hasta el período de reinscripciones en caso de matrícula ordinaria, no siendo así en caso de matrículas extraordinarias o extemporáneas.</p>
<b>Camino Alternativo:</b>	<p>En 1.- Si el estudiante no cumple con alguno de los prerrequisitos antes descritos, el sistema presentará un mensaje de impedimento de matrícula, pidiéndole que se acerque a la escuela.</p> <p>En 5.- Si hay conflicto en cruce de horario o crédito, el sistema dará un mensaje de advertencia y continuará en 2.</p>
<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>• La información no podrá ser modificada, una vez guardada.</li> </ul>

Tabla 2.5: CU – 02 Generar Inscripción.

<b>#Caso de Uso:</b>	CU – 03
<b>Nombre:</b>	Reinscribir
<b>Función:</b>	Realizar el proceso de reinscripción (cambios de grupo, anulación e inscripción de nuevas materias).
<b>Actor:</b>	Estudiante
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar matriculado en el período actual.</li> <li>• Debe tener la autorización correspondiente para este proceso.</li> </ul>
<b>Eventos:</b>	

<b>Camino Básico</b>	El camino básico para el proceso de reinscripción es similar al camino básico del caso de uso CU - 02.
<b>Camino Alternativo:</b>	El camino alternativo para el proceso de reinscripción es similar al camino alternativo del caso de uso CU - 02.
<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>La información no podrá ser modificada, una vez guardada.</li> </ul>

Tabla 2.6: CU – 03 Reinscribir.

<b>#Caso de Uso:</b>	CU – 04
<b>Nombre:</b>	Anular Matrícula
<b>Función:</b>	Realizar la anulación de la matrícula de un estudiante.
<b>Actor:</b>	Subdecano
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>Debe estar matriculado el estudiante en el período actual.</li> <li>Se debe estar dentro del período programado para la anulación de la matrícula (calendario académico).</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>El sistema le presenta al subdecano el detalle de la matrícula (carrera, fecha y hora del sistema, pensum, número de materias, código, nombre, paralelo, número de matrícula, créditos, horario de clases)</li> <li>El sistema despliega un mensaje de advertencia de la anulación de la matrícula.</li> <li>El subdecano aceptará la anulación.</li> <li>El sistema anula la matrícula del estudiante, almacenando el valor total de su matrícula como un saldo a favor del mismo para futuras matrículas.</li> </ol>

<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>La información no podrá ser modificada, una vez guardada.</li> </ul>
----------------------	---

Tabla 2.7: CU – 04 Anular matrícula.

<b>#Caso de Uso:</b>	CU – 05
<b>Nombre:</b>	Enviar al banco
<b>Función:</b>	Genera un archivo plano, con el detalle del valor a ser cobrado a los estudiantes por concepto de matrícula.
<b>Actor:</b>	UGI
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>Tener el registro de los estudiantes inscritos.</li> <li>Debe finalizar el período de inscripciones o reinscripciones.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>El sistema le presenta a la UGI la siguiente información: período, fecha y total de estudiantes inscritos.</li> <li>La UGI escoge la opción generar archivo.</li> <li>El sistema genera el archivo excel, que será necesario para enviar al banco.</li> </ol>
<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>Una vez generado el archivo excel, no podrá ser creado nuevamente hasta que se termine el proceso de reinscripciones.</li> </ul>

Tabla 2.8: CU – 05 Enviar al banco.

<b>#Caso de Uso:</b>	CU – 06
<b>Nombre:</b>	Pagar en Tesorería
<b>Función:</b>	Realizar el pago y la legalización de la matrícula de los estudiantes que hayan especificado en su inscripción o reinscripción las siguientes formas de pago: cheque, tarjeta de crédito o IECE.
<b>Actor:</b>	Tesorería

<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>Tener la lista de estudiantes inscritos que hayan especificado su forma de pago como cheque, tarjeta de crédito o crédito IECE.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>El sistema presenta una pantalla de búsqueda de estudiantes.</li> <li>La tesorería ingresa el apellido o parte del mismo, del estudiante a buscar.</li> <li>El sistema despliega los registros que cumplen con el criterio antes ingresado.</li> <li>La tesorería selecciona al estudiante buscado.</li> <li>La tesorería emite la factura correspondiente al pago de la matrícula y automáticamente se legaliza la matrícula del estudiante.</li> </ol>
<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>Registrar al estudiante como matriculado.</li> </ul>

Tabla 2.9: CU – 06 Pagar en tesorería.

<b>#Caso de Uso:</b>	CU – 07
<b>Nombre:</b>	Legalizar matrícula
<b>Función:</b>	Legalizar la matrícula de los estudiantes que hayan cancelado el valor de su matrícula en el Banco.
<b>Actor:</b>	UGI
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>Tener el archivo excel que el banco obtiene luego del proceso de recaudación.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>La UGI debe seleccionar la opción Legalizar matrícula.</li> <li>Escoger el archivo Excel para subirlo y luego ejecutar el proceso de legalización de la matrícula de los estudiantes que constan en el archivo.</li> </ol>
<b>Poscondición:</b>	<ul style="list-style-type: none"> <li>El sistema registra al estudiante como</li> </ul>

	matriculado.
--	--------------

Tabla 2.10: CU – 07 Legalizar matricula.

<b>#Caso de Uso:</b>	CU – 08
<b>Nombre:</b>	Autorizar Reinscripciones
<b>Función:</b>	Autorizar a un estudiante para la reinscripción de materias.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>El estudiante debe estar matriculado en el periodo inmediato anterior.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>El sistema presenta un listado de los estudiantes que se matricularon en el periodo actual.</li> <li>Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>El Subdecano puede aceptar o no la autorización.</li> <li>El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.11: CU – 08 Autorizar reinscripciones.

<b>#Caso de Uso:</b>	CU – 09
<b>Nombre:</b>	Autorizar Reingresos
<b>Función:</b>	Autorizar a un estudiante para que reingrese a la Carrera.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>El estudiante no debe estar matriculado en el periodo inmediato anterior.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>El sistema le presenta un listado de los estudiantes que no están registrados en el periodo inmediato anterior.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Subdecano puede aceptar o no la autorización.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.12: CU – 09 Autorizar reingresos.

<b>#Caso de Uso:</b>	CU – 10
<b>Nombre:</b>	Autorizar matrículas extraordinarias
<b>Función:</b>	Autorizar a un estudiante para que se matricula en las materias que desee.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El estudiante debe estar matriculado en el periodo inmediato anterior.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta un listado de los estudiantes que no están registrados en el periodo actual.</li> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Subdecano puede aceptar o no la autorización.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.13: CU – 10 Autorizar matriculas extraordinarias.

<b>#Caso de Uso:</b>	CU – 11
<b>Nombre:</b>	Autorizar matrículas extemporáneas
<b>Función:</b>	Autorizar a un estudiante para que se pueda inscribir en la materia de proyecto de titulación.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El estudiante debe ser egresado de una carrera.</li> </ul>
<b>Eventos:</b>	

<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta un listado de los egresados que no están registrados en el periodo actual.</li> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Subdecano puede aceptar o no la autorización.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.14: CU – 11 Autorizar matriculas extemporáneas.

<b>#Caso de Uso:</b>	CU – 12
<b>Nombre:</b>	Autorizar Cambios de Carrera
<b>Función:</b>	<ul style="list-style-type: none"> <li>• Autorizar a un estudiante otra Carrera de la EPN, sea aceptado en la Carrera.</li> <li>• Autorizar a un estudiante de otra Universidad para que ingrese a una Carrera.</li> </ul>
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El estudiante debe estar registrado como alumno de alguna carrera de la EPN.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta un listado de los estudiantes que están registrados en la carrera actual.</li> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Subdecano puede aceptar o no la autorización.</li> <li>4. El sistema mostrará el currículum académico del estudiante para que el Subdecano lo actualice considerando revalidaciones y equiparaciones de materias.</li> <li>5. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.15: CU – 12 Autorizar cambios de carrera.

<b>#Caso de Uso:</b>	CU – 13
<b>Nombre:</b>	Impedimentos de Matrícula
<b>Función:</b>	Registrar a los estudiantes que tienen algún impedimento para matricularse.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta un listado de los estudiantes que no están registrados en el periodo inmediato anterior.</li> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Subdecano selecciona el tipo de impedimento que tiene el estudiante.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.16: CU – 13 Registrar impedimentos de matricula.

<b>#Caso de Uso:</b>	CU – 14
<b>Nombre:</b>	Registrar Aspirantes
<b>Función:</b>	Registrar a los estudiantes que hayan aprobado propedéutico para que sean aspirantes a la carrera.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• El estudiante de haber aprobado PREPO.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta al Subdecano el formato de aceptación de aspirantes a la carrera que consiste de un listado que contiene los siguientes campos número, código, nombre, promedio, IRA. Si es un periodo nuevo esta lista se encontrará vacía.</li> </ol>

	<ol style="list-style-type: none"> <li>2. El Subdecano selecciona buscar y el sistema mostrara un cuadro de texto donde se ingresará el nombre del estudiante.</li> <li>3. El Subdecano selecciona al estudiante y los datos son cargados en el formato de aceptación de aspirantes. El sistema le permitirá eliminar algún estudiante de la lista.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.17: CU – 14 Registrar aspirantes.

<b>#Caso de Uso:</b>	CU – 15
<b>Nombre:</b>	Gestionar Rubros Adicionales
<b>Función:</b>	Registrar rubros adicionales que debe realizar el estudiante en la matrícula, considerando las necesidades de la Carrera.
<b>Actor:</b>	Subdecano de Carrera
<b>Precondiciones:</b>	
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. Se presenta al Subdecano un formato de registro de rubros adicionales que consiste de los siguientes campos: periodo, fecha, y un listado que contiene número, descripción del rubro y valor. Si es un periodo nuevo esta lista se encontrará vacía.</li> <li>2. Si el Subdecano desea registrar un rubro selecciona una fila en la tabla e ingresará la descripción del rubro y el valor. El sistema le permitirá retirar la información de un rubro al Subdecano.</li> <li>3. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.18: CU – 15 Gestionar rubros adicionales.

<b>#Caso de Uso:</b>	CU – 16
<b>Nombre:</b>	Autorizar Pagos en Partes
<b>Función:</b>	Autorizar a los estudiantes para que realicen su pago de matrícula en partes.
<b>Actor:</b>	Director de Escuela.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema le presenta un listado de los estudiantes que están registrados en la carrera.</li> <li>2. Esta lista puede reducirse con filtros como: nombre o apellido.</li> <li>3. El Director de Escuela selecciona la aceptación o no de la autorización.</li> <li>4. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.19: CU – 16 Autorizar pagos en partes.

<b>#Caso de Uso:</b>	CU – 17
<b>Nombre:</b>	Autorizar Pagos con IECE
<b>Función:</b>	Autorizar a los estudiantes para que realicen su pago de matrícula con crédito IECE.
<b>Actor:</b>	Director de Escuela.
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. Se presenta al Director de Escuela el formato para autorización de pago de matrícula con crédito IECE que consiste de los siguientes campos: nombre, cédula, carrera y pensum.</li> <li>2. El Director de Escuela selecciona buscar y el</li> </ol>

	<p>sistema mostrará un cuadro de texto donde se ingresará el nombre del estudiante.</p> <ol style="list-style-type: none"> <li>3. El Director de Escuela selecciona al estudiante y los datos son cargados en el formato de autorización de pago con crédito IECE.</li> <li>4. El Director de Escuela selecciona la aceptación o no de la autorización.</li> <li>5. El sistema guardará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.20: CU – 17 Autorizar pagos con IECE.

<b>#Caso de Uso:</b>	CU – 18
<b>Nombre:</b>	Registrar IPF
<b>Función:</b>	Registrar y Actualizar los valores de IPF (Ingreso Per cápita Familiar).
<b>Actor:</b>	Unidad de Bienestar Estudiantil o su Delegado
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema mostrara un cuadro de texto donde la Unidad de Bienestar Estudiantil ingresará parte del nombre del estudiante y luego selecciona buscar.</li> <li>2. El sistema presentará un listado de los estudiantes cuyos nombres concuerden con la información ingresada por la Unidad de Bienestar Estudiantil.</li> <li>3. La Unidad de Bienestar Estudiantil selecciona al estudiante y procede a ingresar los datos en caso de ser un aspirante caso contrario los actualiza.</li> <li>4. El sistema actualizará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.21: CU – 18 Registrar IPF.

<b>#Caso de Uso:</b>	CU – 19
<b>Nombre:</b>	Registrar Descuentos
<b>Función:</b>	Registrar el porcentaje de descuento que tiene un estudiante por hermanos, cónyuge, e hijos de empleados de la EPN.
<b>Actor:</b>	Unidad de Bienestar Estudiantil.
<b>Precondiciones:</b>	
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema mostrara un cuadro de texto donde la Unidad de Bienestar Estudiantil ingresará parte del nombre del estudiante y luego selecciona buscar</li> <li>2. El sistema presentará un listado de los estudiantes cuyos nombres concuerden con la información ingresada por la Unidad de Bienestar Estudiantil</li> <li>3. La Unidad de Bienestar Estudiantil selecciona el tipo de descuento y asigna el porcentaje.</li> <li>4. El sistema actualizará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.22: CU – 19 Registrar documentos.

<b>#Caso de Uso:</b>	CU – 20
<b>Nombre:</b>	Registrar Becas
<b>Función:</b>	Registrar el porcentaje de descuento que tiene por Beca un estudiante por merito académico, merito deportivo o por situación económica.
<b>Actor:</b>	Unidad de Bienestar Estudiantil
<b>Precondiciones:</b>	<ul style="list-style-type: none"> <li>• Debe estar registrada la información del estudiante.</li> </ul>
<b>Eventos:</b>	
<b>Camino Básico</b>	<ol style="list-style-type: none"> <li>1. El sistema mostrara un cuadro de texto donde la Unidad de Bienestar Estudiantil ingresará parte del</li> </ol>

	<p>nombre del estudiante y luego selecciona buscar</p> <ol style="list-style-type: none"> <li>2. El sistema presentará un listado de los estudiantes cuyos nombres concuerden con la información ingresada por la Unidad de Bienestar Estudiantil.</li> <li>3. La Unidad de Bienestar Estudiantil selecciona el tipo de descuento y asigna el porcentaje.</li> <li>4. El sistema actualizará los datos.</li> </ol>
<b>Poscondición:</b>	

Tabla 2.23: CU – 20 Registrar becas.

**1. Solicitar Información.-** El Sistema mostrará la información personal y académica del estudiante a través de los siguientes servicios:

**1.1. Datos Personales.-** El Sistema mostrará al estudiante su información personal, y podrá realizar modificaciones en ciertos campos como: dirección, teléfono, e-mail y estado civil.

El sistema podrá permitir modificaciones en algunos campos que no pueden ser modificados por el estudiante como tipo de colegio, nombre del colegio entre otra información a un usuario especial para un proceso de actualización de datos.

**1.2. Currículum Académico.-** El Sistema presentará información del currículum académico del estudiante, el cual contendrá: carrera, fecha de ingreso, pensum, última matrícula y la información de currículum como número de materias, año, periodo, código, nombre de materia, número de matrícula, calificación, número de créditos de cada materia, estado de aprobación (aprobado A, fallido F, exonerado E).

El coordinador podrá realizar modificaciones en el currículum académico del estudiante para equiparación y revalidación de materias para cambios de carrera o de universidad y el registro de aprobación de inglés.

- 1.3. Créditos por Categoría.-** El Sistema presentará una tabla en la cual se encontrará el porcentaje de créditos aprobados por categoría, considerando el pensum del estudiante.
- 1.4. Malla Curricular.-** El Sistema mostrará la malla curricular del estudiante considerando el pensum de estudios correspondiente, resaltando las materias aprobadas de las no aprobadas.
- 1.5. Horarios.-** El Sistema mostrará al estudiante el listado de las materias en las que se encuentra matriculado, con el respectivo horario de clases.
- 1.6. Calificaciones.-** El Sistema presentará información de las calificaciones parciales y totales del estudiante de cada una de las materias tomadas en el período actual.
- 1.7. Períodos Matriculados.-** El Sistema presentará una tabla en la cual se encontrará el detalle de los periodos en los que se ha matriculado el estudiante en la Carrera.
- 2. Generar Inscripción.-** El Sistema permitirá que el estudiante se inscriba en las materias tanto para matrículas ordinarias, extraordinarias y extemporáneas, para lo cual se presentará un listado de las materias que puede tomar considerando los prerrequisitos/correquisitos de acuerdo a la malla curricular y materias aprobadas.

Para el caso de estudiantes que estén realizando la tesis y que se haya vencido el plazo de proyecto de titulación, deberán tomar el curso de actualización, que consta de un paquete de materias para cada seminario. Además se mostrará la información detallada de tipo de matrícula, grupos, paralelos, cupos, horarios y profesores para cada materia. El Sistema validará los cruces de horarios, impedimentos, reingresos previa autorización y límites de créditos en la matrícula del estudiante.

Al finalizar con el proceso de inscripción de materias se mostrará el detalle del valor total a pagar de la matrícula considerando el número de pagos, valor de la matrícula, créditos, descuentos, saldos a favor, saldos en contra y rubros adicionales de la institución y de la carrera. Para el curso de actualización (seminarios) tendrá un valor fijo por matrícula y por crédito según el valor Pregrado (200, 20) y Postgrado (300,20). El usuario deberá escoger la forma de pago: banco, tarjeta de crédito, previa autorización para crédito IECE, pago por rol y pago en partes. Una vez realizada la inscripción el sistema no admitirá cambios.

El sistema podrá permitir al coordinador de carrera o su delegado la inscripción de materias a aquellos estudiantes que hayan sido previamente autorizados para matrículas extraordinarias y extemporáneas.

- 3. Reinscribir.-** El Sistema le permitirá realizar la reinscripción de matrícula a todos los estudiantes que se les haya previamente autorizado este proceso. La inscripción y retiro de las materias se lo efectuará de la misma manera que el proceso de inscripción (2); considerando que el sistema validará los cruces de horarios, límites de créditos e impedimentos en la matrícula del estudiante. Una vez realizada la reinscripción el sistema no admitirá cambios
- 4. Anular Matrícula.-** El Sistema permitirá al estudiante anular su matrícula siempre que se encuentre dentro del plazo permitido (calendario académico).
- 5. Enviar al Banco<sup>10</sup>.-** Una vez terminado el período de: matrículas ordinarias, extraordinarias y una vez terminado el plazo para pagos en dos partes, el sistema permitirá a la UGI generar un archivo plano que será enviado al banco para el pago de matrículas de los estudiantes.
- 6. Pagar en Tesorería.-** Una vez que el estudiante a indicado la forma de pago en las matrículas ordinarias, extraordinarias, y extemporáneas,

---

<sup>10</sup> Este proceso es opcional para este módulo

Tesorería podrá recuperar, la información de los estudiantes que solicitaron su forma de pago por crédito IECE (previa una autorización 14), tarjeta de crédito o pago con roles.

Cuando el estudiante realice su pago el sistema emitirá el comprobante de pago como constancia de registro de pago, quedando automáticamente legalizada la matrícula del estudiante. Para los estudiantes que realicen que tramiten su forma de pago con crédito IECE, quedarán provisionalmente matriculados debiendo realizar el pago de su matrícula hasta la fecha máxima establecida para pagos IECE. Si hasta la fecha máxima autorizada para este tipo de pago, el estudiante no realizará el pago, deberá cancelar su matrícula con cualquier otra forma de pago caso contrario se le anulará la matrícula.

Al legalizar la matrícula por esta forma de pago automáticamente se creará el número de matrícula para cada estudiante.

- 7. Legalizar Matrícula<sup>11</sup>.**- Una vez terminado el periodo de recaudación, el Banco enviará un archivo con los valores recaudados. El sistema tomará ese archivo y le permitirá a la Unidad de Gestión de Información (UGI), realizar el proceso de legalización de las matrículas de los diferentes estudiantes que realizaron el pago de su matrícula en el Banco.

Al legalizar la matrícula por esta forma de pago automáticamente se creará el número de matrícula para cada estudiante.

- 8. Autorizar Reinscripciones.**- El sistema permitirá al Coordinador de Carrera autorizar a un estudiante para que pueda reinscribirse en sus materias. La autorización de reinscripción permitirá al estudiante según sus necesidades cambiar el horario de clases, inscribirse en una nueva (as) materia (as) y anular una materia (as).

---

<sup>11</sup> La legalización de la matrícula permitirá inscribir de forma definitiva a un estudiante.

9. **Autorizar Reingresos.-** El sistema permitirá al Coordinador de Carrera, autorizar el reingreso de un estudiante a la carrera, que no se haya matriculado en el período inmediato anterior (es).
10. **Autorizar Matrículas Extraordinarias.-** El sistema permitirá al Coordinador de Carrera autorizar a un estudiante para que se inscriba en matrículas extraordinarias.
11. **Autorizar Matrículas Extemporáneas.-** El sistema permitirá al Coordinador de Carrera autorizar a un estudiante (tesita) para que se inscriba en tesis.
12. **Autorizar Cambios de Carrera.-** El sistema permitirá al Coordinador de Carrera, aceptar a un estudiante de otra carrera de la EPN para que forme parte de la carrera a la que representa, para lo cual se ingresará las revalidaciones y equiparaciones de los pensum de las diferentes materias.
13. **Registrar Impedimentos de Matrícula.-** El sistema permitirá al Coordinador de Carrera registrar a los estudiantes que hayan cometido algún tipo de falta, registrando la sanción, que les impedirá realizar su matrícula.
14. **Aceptar Aspirantes.-** El Sistema permitirá al Coordinador de Carrera aceptar a estudiantes que hayan aprobado previamente propedéutico para que sean aspirantes a la carrera.
15. **Registrar Rubros Adicionales.-** El sistema permitirá al Coordinador de Carrera registrar rubros adicionales que deben realizar los estudiantes en la matrícula. Estos valores dependerán de las necesidades de cada una de las carreras.
16. **Autorizar Pagos por Partes.-** El sistema permitirá al Director de Escuela o su delegado, autorizar a los estudiantes que puedan realizar el pago de matrícula en partes.

**17. Autorizar Pagos con IECE.-** El sistema permitirá al Director de Escuela o su delegado, autorizar a los estudiantes para que realicen el pago de la matrícula con crédito de IECE.

**18. Registrar Situación Socioeconómica IPF (Ingreso Per cápita Familiar).-**

El sistema permitirá a la Unidad de Bienestar Estudiantil ingresar o actualizar los datos socioeconómicos previos a un análisis de ingresos familiares para cada estudiante. Este valor es usado para el cálculo de costos del valor de la matrícula de los estudiantes.

**19. Registros de Descuentos.-** El sistema permitirá a la Unidad de Bienestar Estudiantil ingresar el porcentaje de descuentos que se le puede conceder a un estudiante por diversos motivos como hermanos, cónyuges, e hijos de los empleados de la EPN. Para el caso de hijos de empleados de la EPN, se procederá a registrar la deuda por rol en caso de que se desee pagar por este medio.

La EPN otorga descuentos especiales del 50% en el pago de aranceles de matrículas y créditos, cursos de inglés y demás programas o cursos académicos excepto los de postgrado, a los hijos de profesores titulares, personal administrativo y de servicios de la institución; y, del 10% en el pago de matrículas y créditos, cursos de inglés y demás programas o cursos académicos, excepto los de postgrado, a los estudiantes, hermanos o cónyuges de la EPN, miembros de un mismo núcleo familiar. Los descuentos no son acumulativos, y son aplicados para los valores de matrícula y créditos. El pago por otros conceptos deberá ser realizado por el estudiante en las fechas establecidas para pagos.

**20. Registrar Becas.-** El sistema permitirá a la Unidad de Bienestar Estudiantil ingresar el porcentaje de descuentos que se puede otorgar a los estudiantes, por los siguientes conceptos: merecimiento académico, deportivo o por situación económica. Las becas son aplicadas para los

valores de matrícula y créditos. El pago por otros conceptos deberá ser realizado por el estudiante en las fechas establecidas para pagos.

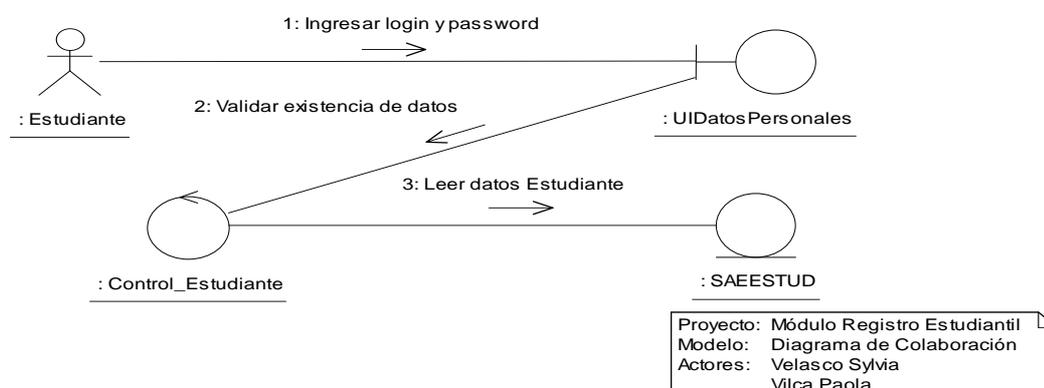
## 2.3.2 ANÁLISIS

En esta fase se estudia de una manera más detallada la información presentada anteriormente, pero esta vez desde el punto de vista del programador (vista interna del sistema) y no desde el de un usuario (vista externa del sistema), esto se realizó utilizando el siguiente diagrama:

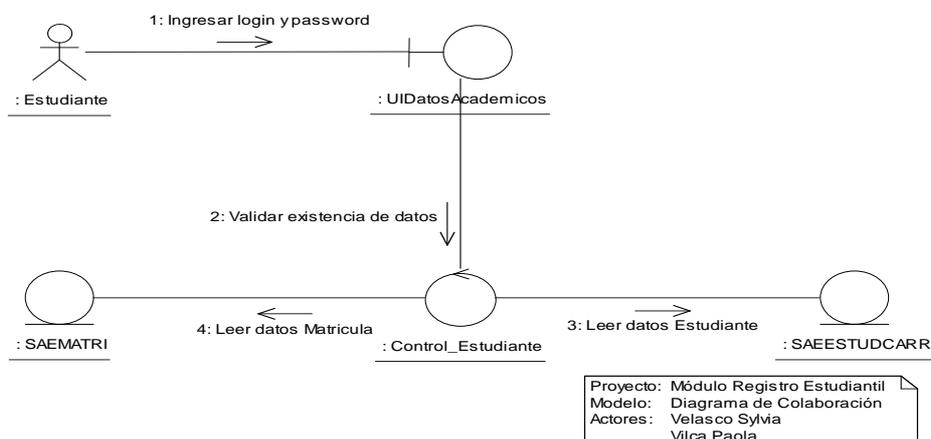
### 2.3.2.1 Diagrama de colaboración

#### CU – 01 SOLICITAR INFORMACION ESTUDIANTE

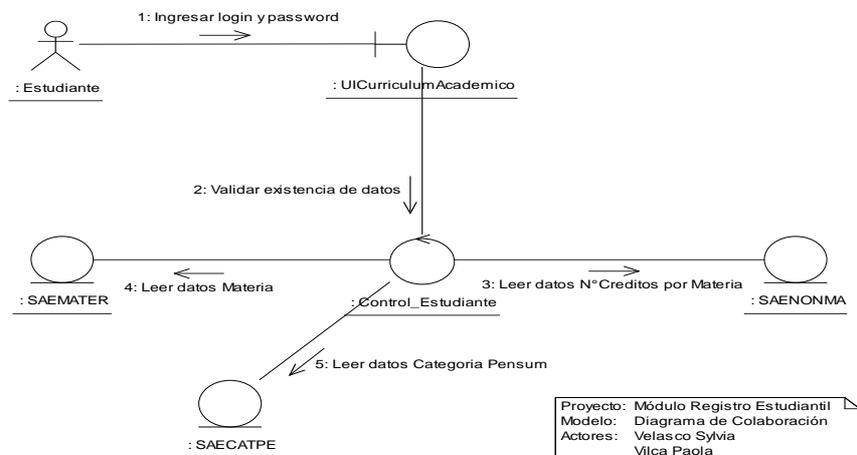
Datos Personales:



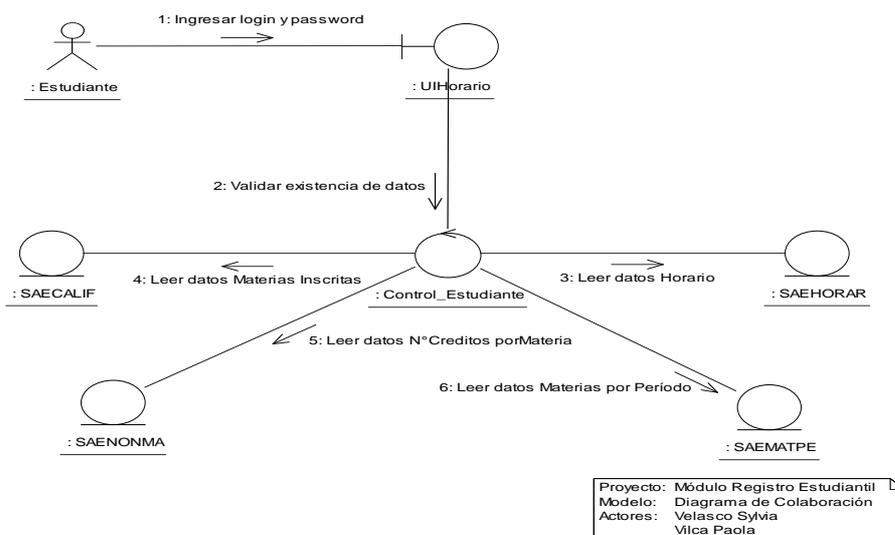
Datos Académicos:



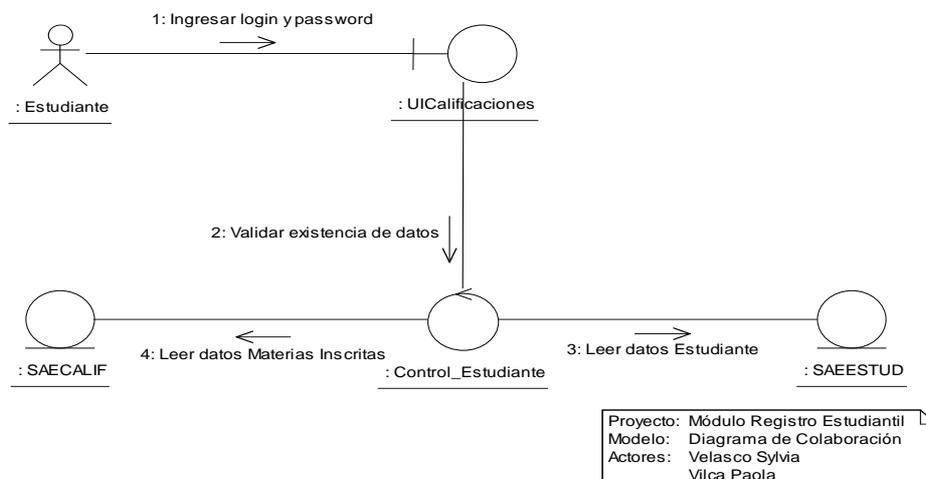
## Currículum Académico:



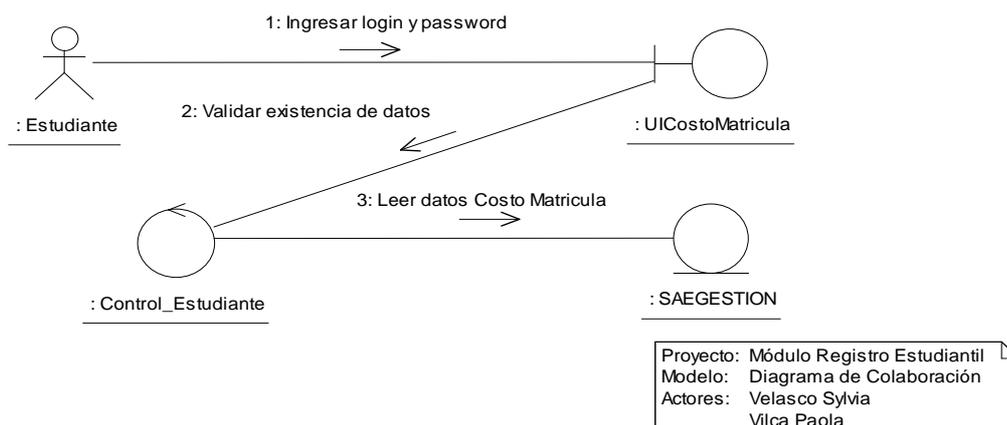
## Consultar Horario:



### Consultar Calificaciones:



### Consultar Costo Matricula:



### Consultar Porcentaje de Materias Aprobadas por Categoría:

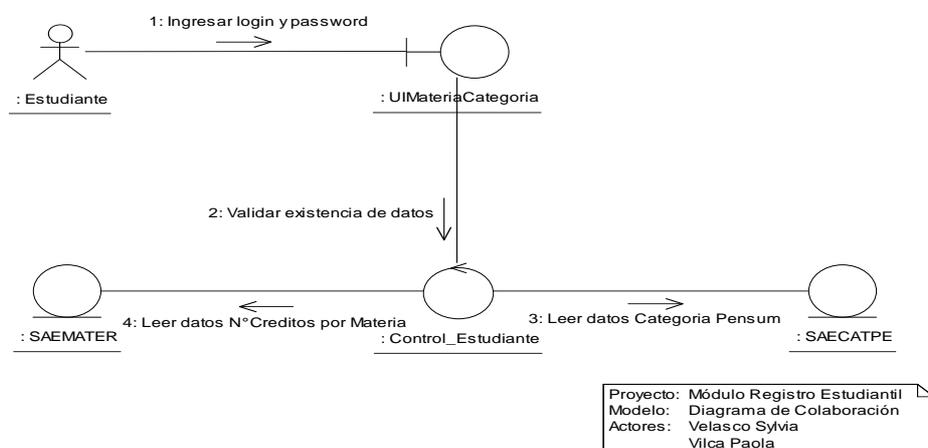


Figura 2.3: Diagrama de colaboración CU – 01 Solicitar información estudiante.

En el caso de uso solicitar información los actores participantes podrán ser:

- Estudiante
- Subdecano
- Bienestar estudiantil

Descripción del proceso:

El estudiante puede realizar consultas de la siguiente información:

- ✓ Datos Personales: Presenta al estudiante información como: nombres, apellidos, número único, fecha de nacimiento, CI, datos del colegio, etc.
- ✓ Datos Académicos: Presenta información de pensum, periodo de ingreso, ultimo periodo matriculado.
- ✓ Curriculum Académico: Presenta información acerca de: materias por periodo, porcentaje de materias aprobadas por categoría, promedio, IRA.
- ✓ Crédito por Categoría: Presenta información acerca de: materias aprobadas por categoría.
- ✓ Malla Curricular: Presenta un archivo en formato .pdf con la información respectiva.
- ✓ Consultar Horarios: Presenta información del horario del estudiante en el periodo actual.
- ✓ Consultar Calificaciones: Presenta información de calificaciones del estudiante en el periodo actual.
- ✓ Consultar Costo de Matricula: Presenta información en detalle del costo de matricula del periodo actual.

- ✓ Periodos matriculados: Presenta información de los periodos en el que el estudiante se ha matriculado.

## CU – 02 GENERAR INSCRIPCION

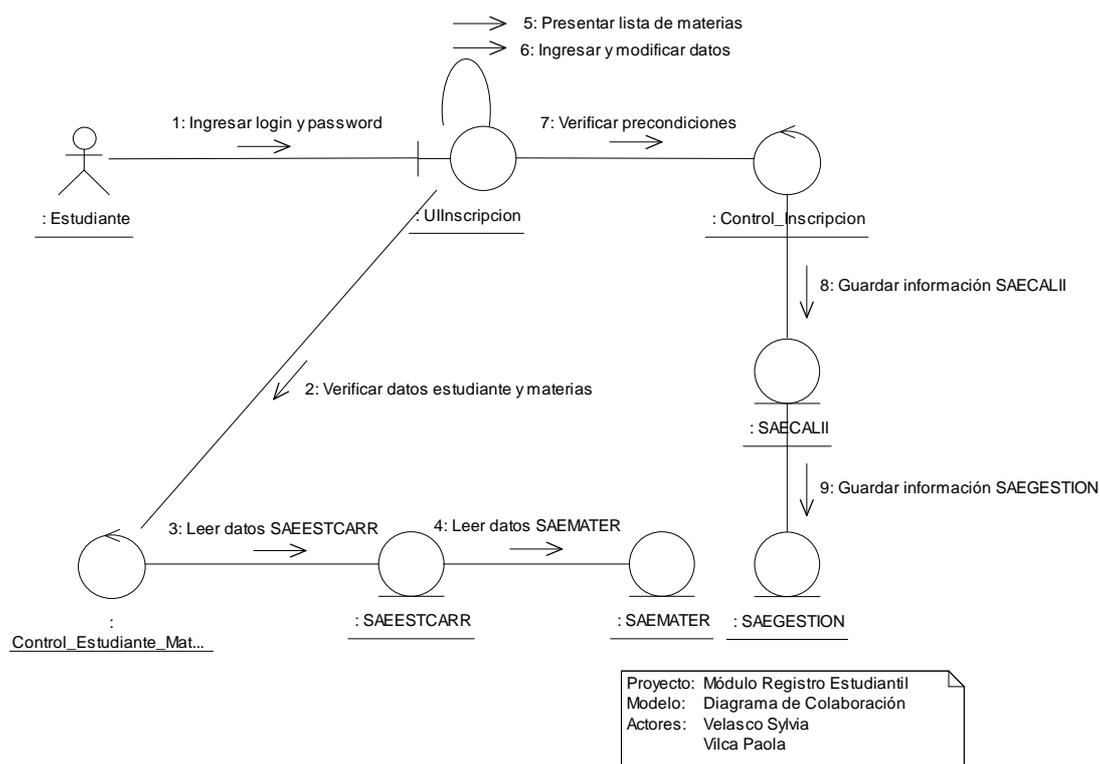


Figura 2.4: Diagrama de colaboración CU – 02 Generar Inscripción.

En el caso de uso solicitar información los actores participantes podrán ser:

- Estudiante

Descripción del proceso:

Proceso mediante el cual el estudiante puede inscribirse, al estudiante se le presentara información de las posibles materias que puede tomar el presente periodo, de las cuales deberá seleccionar las de su preferencia y las tomara en el

horario mas conveniente una vez terminado este paso se guardara la inscripción y el sistema calculara los costos de la matricula, elegir la forma de pago y finalmente terminar el proceso.

Dependiendo de la forma de pago al estudiante se le presentara información acerca del lugar de pago y las fechas en las cuales se puede cancelar.

### CU – 03 REINSCRIBIR

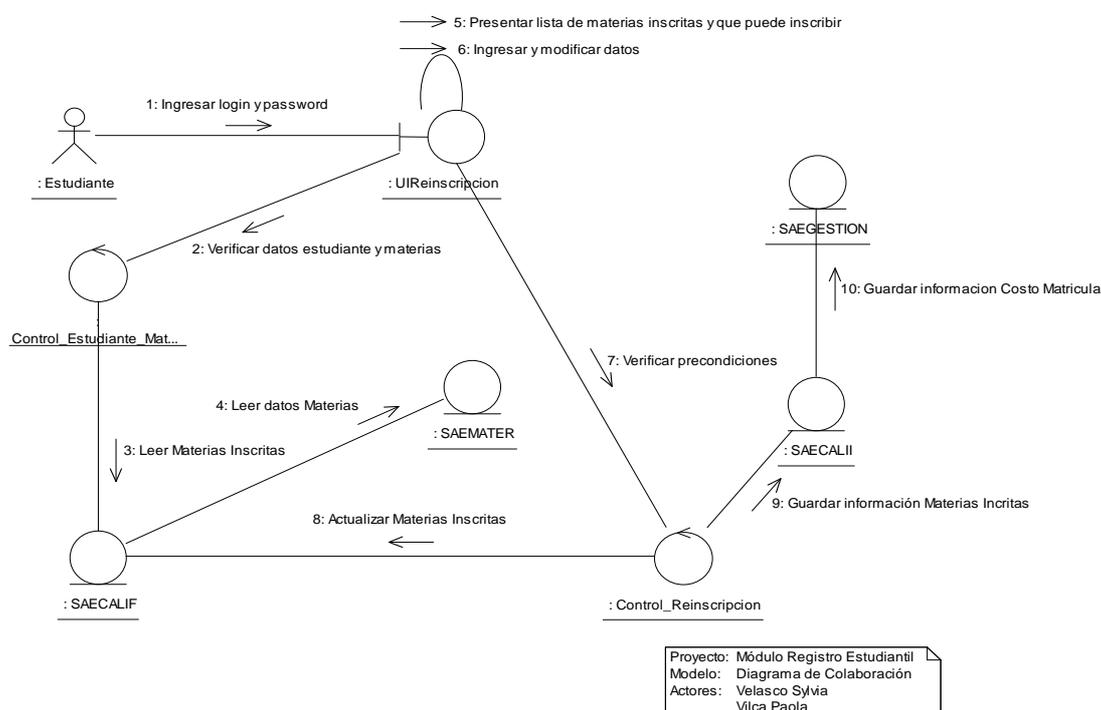


Figura 2.5: Diagrama de colaboración CU – 03 Reinscribir.

En el caso de uso solicitar información los actores participantes podrán ser:

- Estudiante

Descripción del proceso:

Proceso mediante el cual el estudiante puede reinscribirse, al estudiante se le presentara información de las materias que registro en el proceso de inscripción, además, de un lista con las posibles materias que puede tomar el presente

periodo, de las cuales deberá seleccionar las de su preferencia y las tomara en el horario mas conveniente una vez terminado este paso se guardara la reinscripción y el sistema calculara los costos de la matricula, elegir la forma de pago y finamente terminar el proceso.

Dependiendo de la forma de pago al estudiante se le presentara información acerca del lugar de pago y las fechas en las cuales se puede cancelar.

#### CU – 04 ANULAR MATRICULA

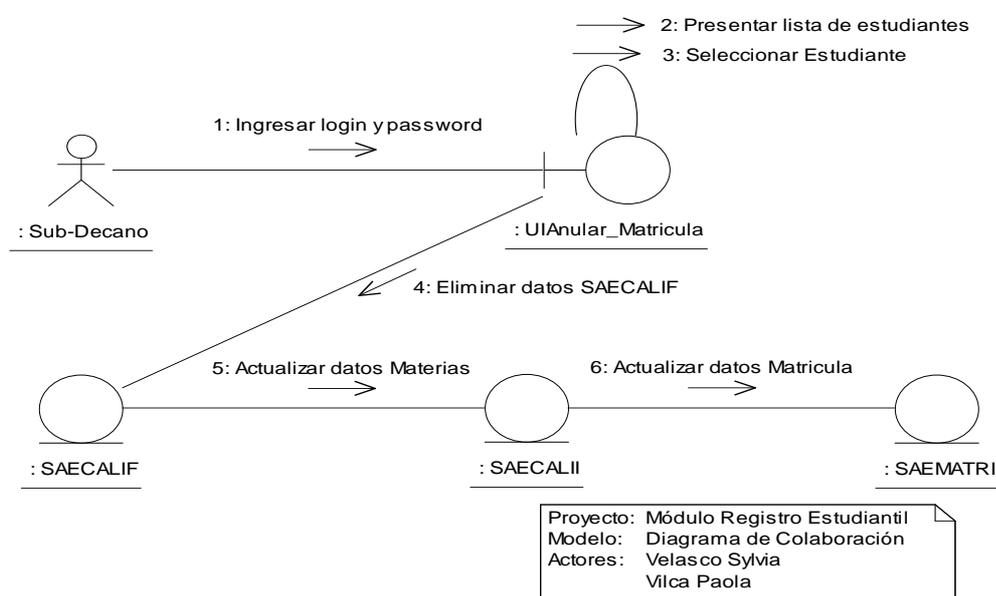


Figura 2.6: Diagrama de colaboración CU – 04 Anular matrícula.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

En el siguiente proceso el subdecano de cada facultad podrá anular la matrícula de un estudiante si así fuese solicitado, se presentara una lista de los estudiantes matriculados en el periodo actual, de la misma se seleccionara o se buscara al estudiante solicitante y se procederá a realizar la anulación de la matrícula.

## CU – 05 ENVIAR AL BANCO

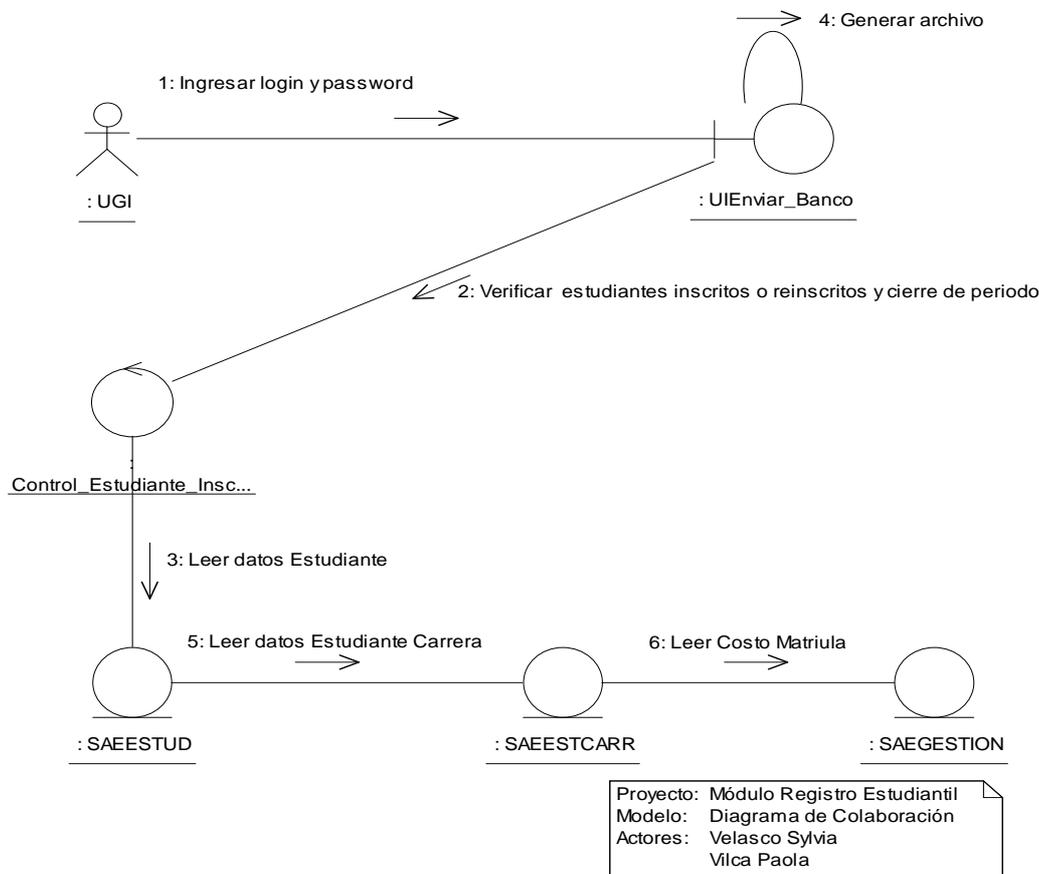


Figura 2.7: Diagrama de colaboración CU – 05 Enviar al banco.

En el caso de uso solicitar información los actores participantes podrán ser:

- UGI (Unidad de Gestión de información)

Descripción del proceso:

En este proceso el sistema generara un archivo en Excel para la forma de pago Banco con información de los estudiantes con el valor de la matrícula a cancelar, este archivo dependerá del periodo de matriculación en el que se encuentre la institución por ejemplo si fuesen matrículas ordinarias, extraordinarias o reinscripciones el cual será enviado al banco para los respectivos cobros.

## CU – 06 PAGAR EN TESORERIA

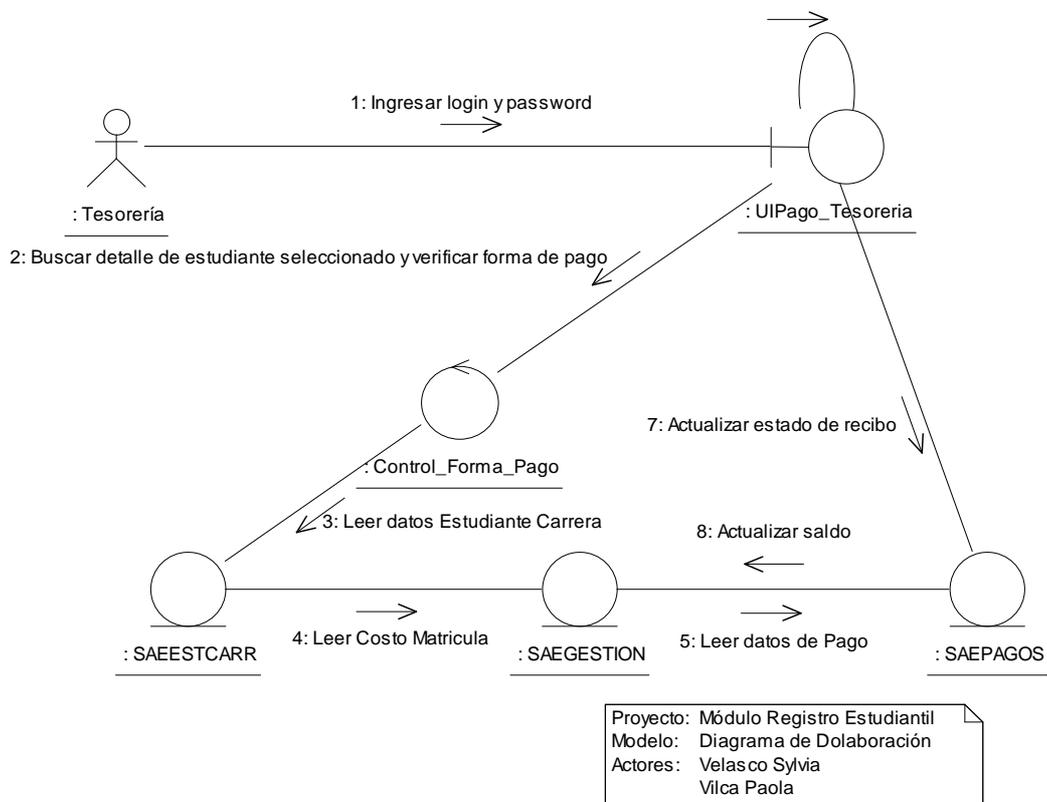


Figura 2.8: Diagrama de colaboración CU – 06 Pagar en tesorería.

En el caso de uso solicitar información los actores participantes podrán ser:

- Tesorería

Descripción del proceso:

Terminado el proceso de matriculación si el estudiante ha seleccionado la forma de pago "IECE" o "Tarjeta de Crédito" podrá realizar el pago en tesorería.

Cuando el estudiante realice su pago el sistema emitirá el comprobante de pago como constancia de registro de pago, quedando automáticamente legalizada la matrícula del estudiante.

Los estudiantes que realicen su pago con crédito IECE, quedarán provisionalmente matriculados debiendo realizar el pago de su matrícula hasta la fecha máxima establecida para pagos IECE. Si hasta la fecha máxima autorizada para este tipo de pago, el estudiante no lo ha realizado, deberá cancelar su matrícula con cualquier otra forma de pago caso contrario se le anulará la matrícula.

## CU – 07 LEGALIZAR MATRICULA

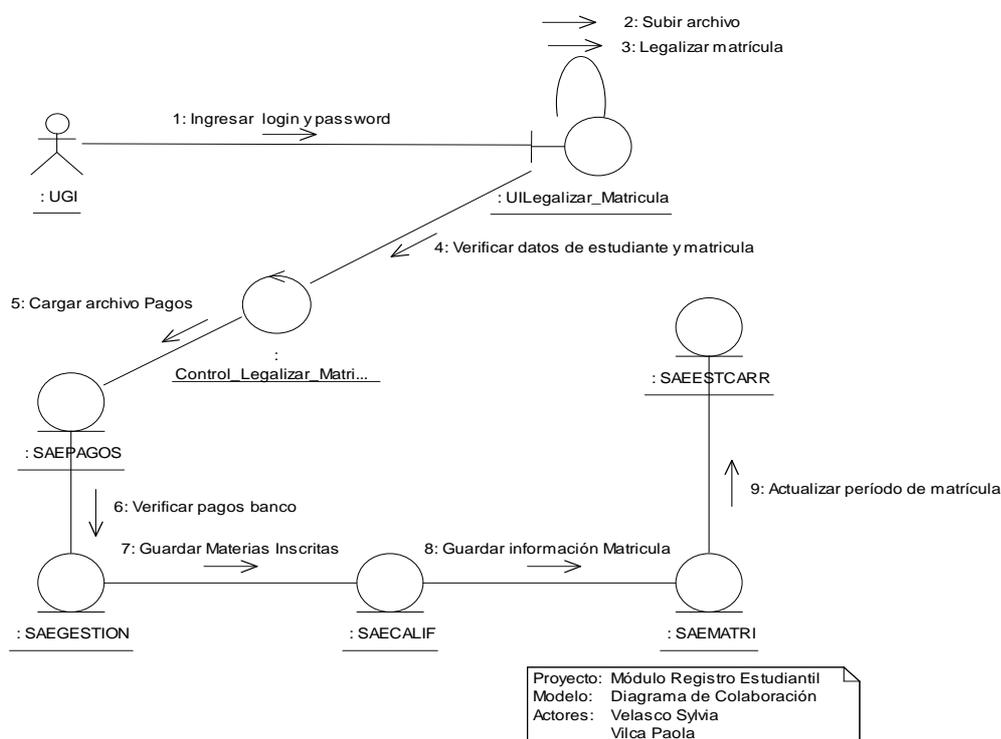


Figura 2.9: Diagrama de colaboración CU – 07 Legalizar matrícula.

En el caso de uso solicitar información los actores participantes podrán ser:

- UGI (Unidad de Gestión de Información)

Descripción del proceso:

Este proceso permitirá enviar un archivo al Banco con los valores a recaudar de los estudiantes matriculados, una vez terminado el proceso de pagos el banco

enviara un archivo a la EPN el cual permitirá a la Unidad de Gestión de Información (UGI), realizar el proceso de legalización de las matrículas.

## CU – 08 AUTORIZAR REINSCRIPCIONES

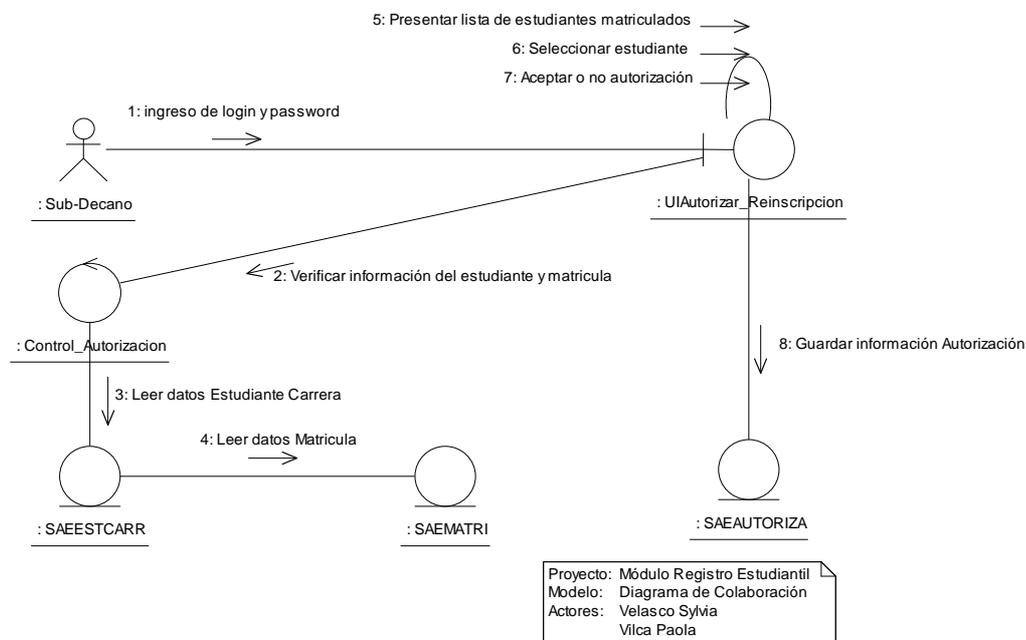


Figura 2.10: Diagrama de colaboración CU – 08 Autorizar reinscripciones.

En el caso de uso solicitar información los actores participantes podrán ser:

- Sub-Decano

Descripción del proceso:

Permite al Sub-Decano, autorizar a un estudiante para que pueda reinscribirse en materias adicionales a las ya tomadas en el proceso de inscripción.

La autorización de reinscripción permitirá al estudiante cambiar el horario de clases, inscribirse en una nueva (as) materia (as) y anular una materia (as).

## CU – 09 AUTORIZAR REINGRESOS

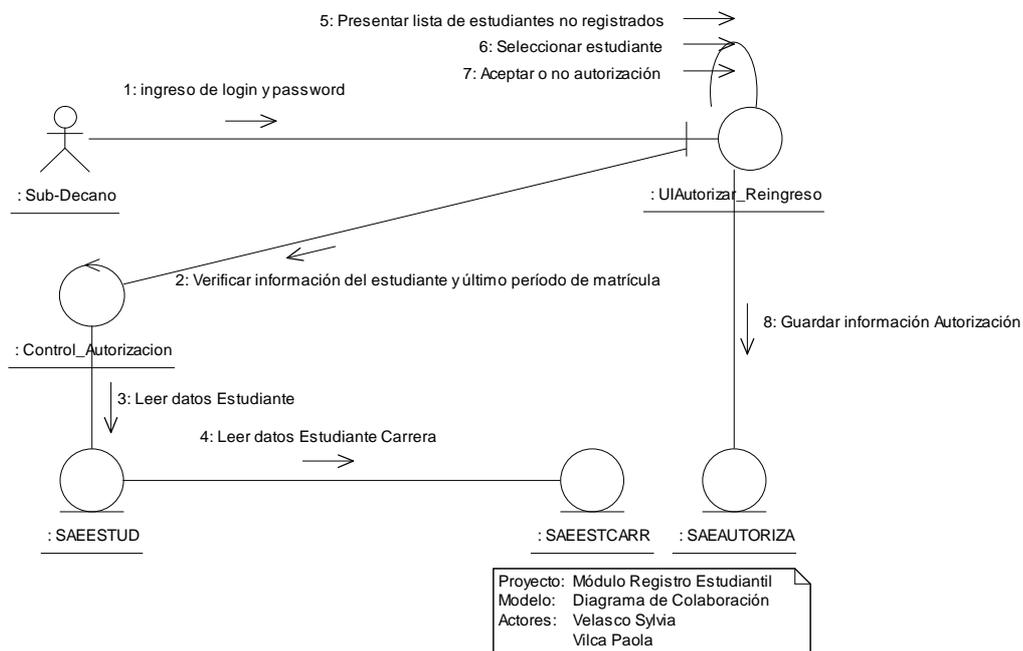


Figura 2.11: Diagrama de colaboración CU – 09 Autorizar reingresos.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permite al Subdecano, autorizar el reingreso de un estudiante a la carrera, que no se haya matriculado en el período inmediato anterior (es).

## CU – 10 AUTORIZAR MATRICULAS EXTRAORDINARIAS

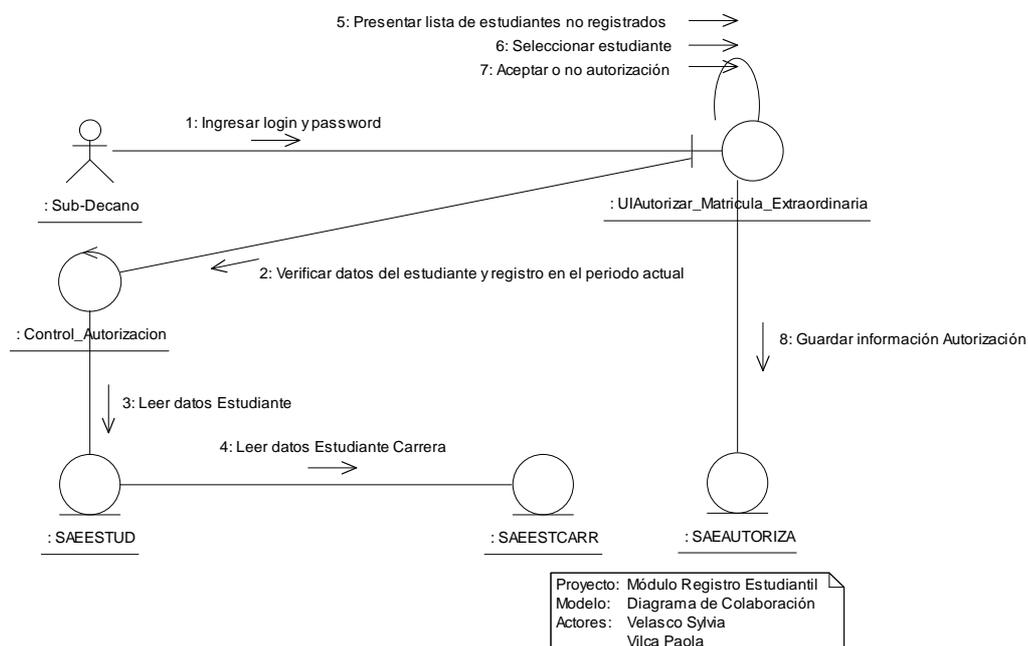


Figura 2.12: Diagrama de colaboración CU – 10 Autorizar matriculas extraordinarias.  
 En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permite al Sub-Decano autorizar a un estudiante para realizar el proceso de inscripción en el periodo de matrículas extraordinarias.

## CU -11 AUTORIZAR MATRICULAS EXTEMPORANEAS

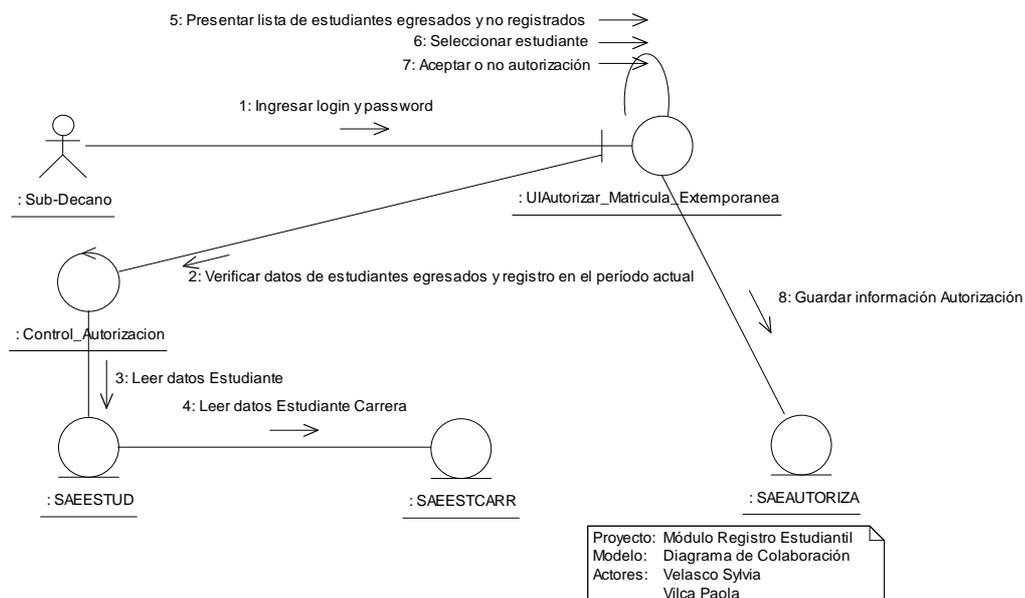


Figura 2.13: Diagrama de colaboración CU – 11 Autorizar matriculas extemporaneas.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permite al Subdecano autorizar a un estudiante (tesista) para que realice el proceso de inscripción en "Proyecto de Titulación".

## CU – 12 AUTORIZAR CAMBIOS DE CARRERA

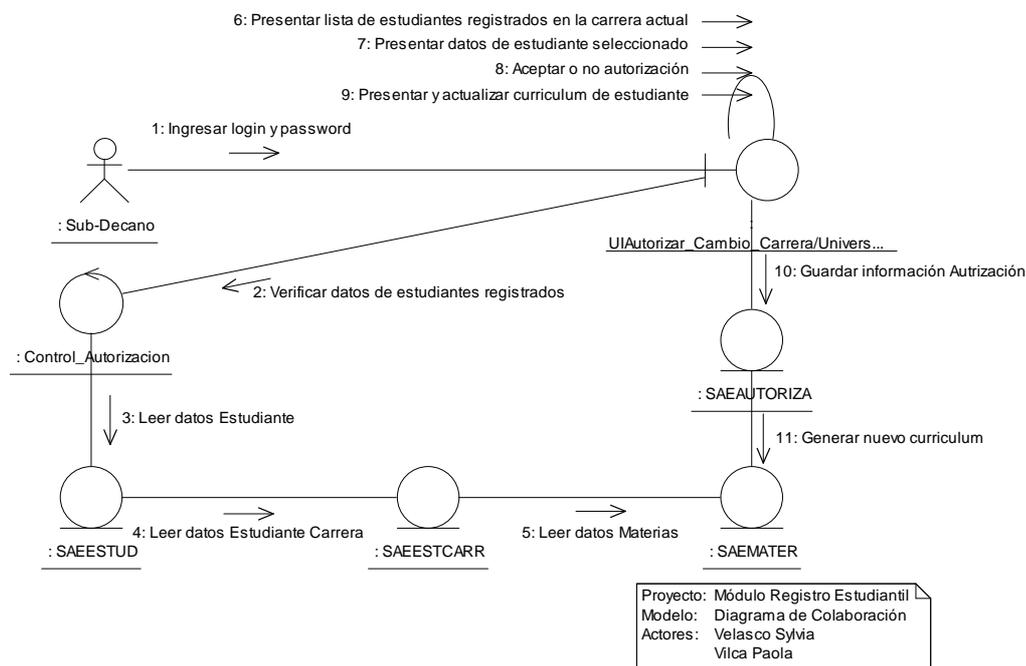


Figura 2.14: Diagrama de colaboración CU – 12 Autorizar cambios de carrera.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permite al Subdecano, aceptar a un estudiante de otra carrera de la EPN para que forme parte de la carrera a la que representa, en este proceso se ingresará las revalidaciones y equiparaciones de los pensum de las diferentes materias.

## CU – 13 REGISTRO DE IMPEDIMENTOS DE MATRICULA

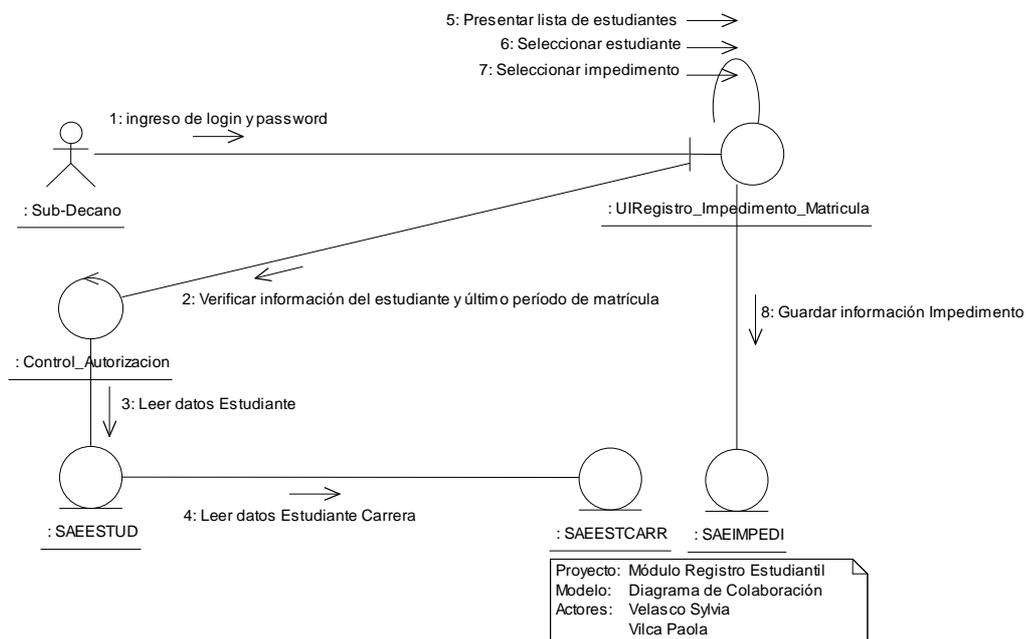


Figura 2.15: Diagrama de colaboración CU – 13 Registro de impedimentos de matrícula.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permitirá al Subdecano registrar a los estudiantes que hayan cometido algún tipo de falta se registrara la sanción la cual impedirá realizar el proceso de inscripción.

## CU – 14 REGISTRAR ASPIRANTE

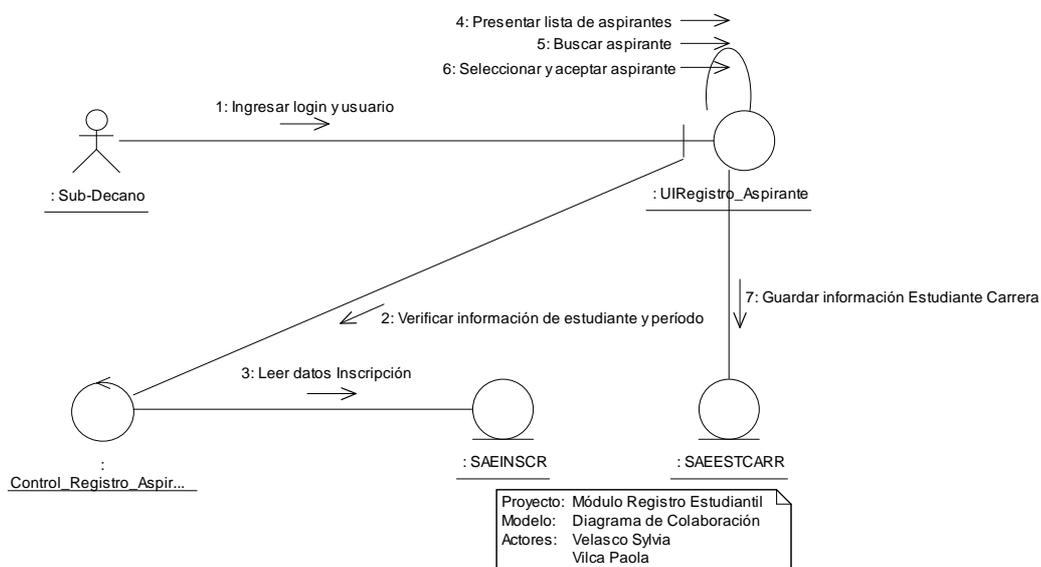


Figura 2.16: Diagrama de colaboración CU – 14 Registrar aspirante.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permitirá al Subdecano aceptar a estudiantes que hayan aprobado previamente propedéutico para que sean aspirantes a la carrera.

## CU – 15 GESTINAR RUBROS ADICIONALES

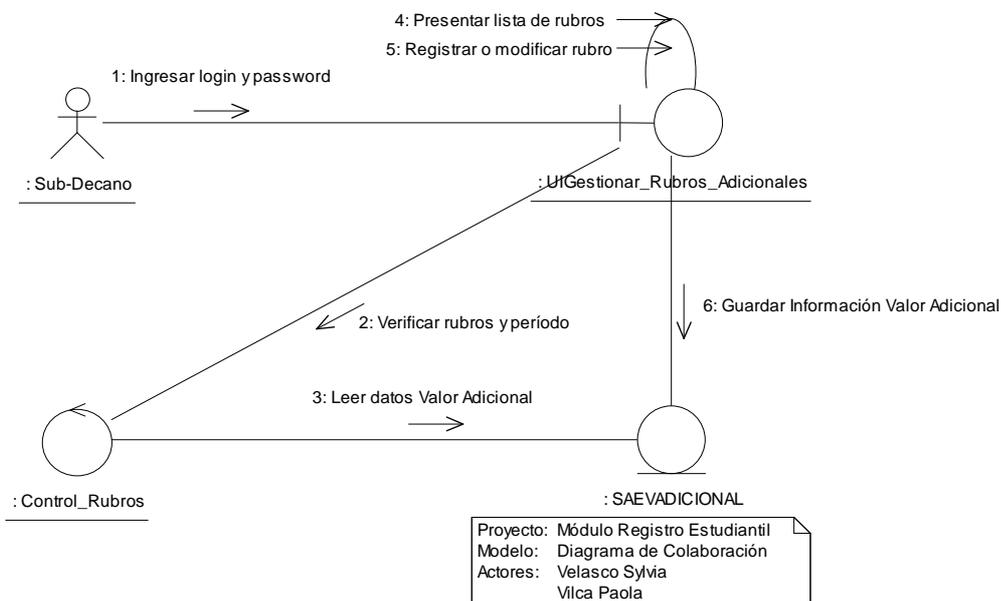


Figura 2.17: Diagrama de colaboración CU – 15 Registrar rubros adicionales.

En el caso de uso solicitar información los actores participantes podrán ser:

- Subdecano

Descripción del proceso:

Permitirá al Subdecano registrar rubros adicionales que deben cancelar los estudiantes en el proceso de matrícula. Estos valores dependerán de las necesidades de cada una de las carreras.

## CU – 16 AUTORIZAR PAGOS POR PARTES

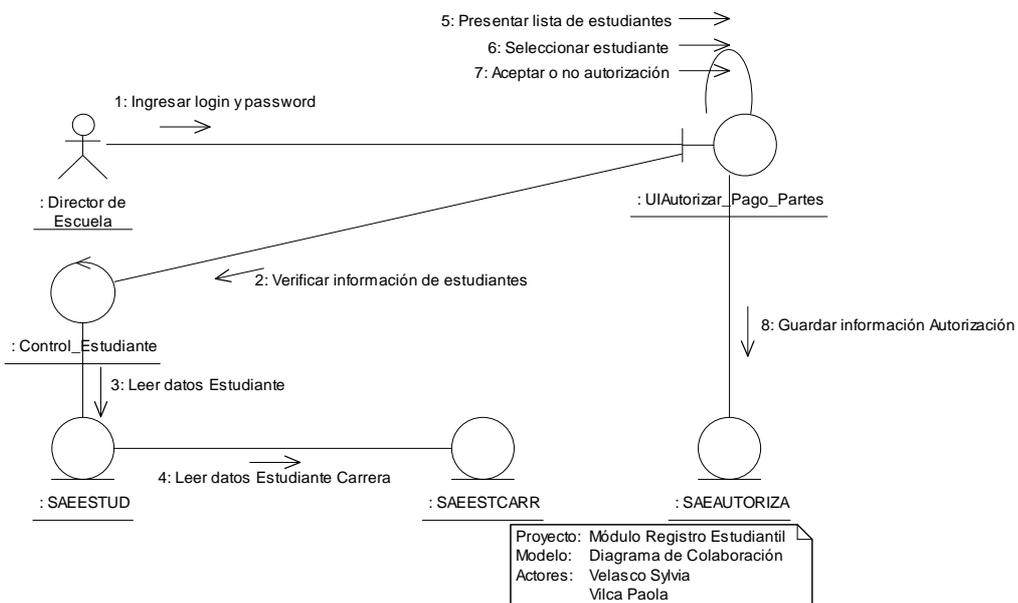


Figura 2.18: Diagrama de colaboración CU – 16 Autorizar pagos por partes.

En el caso de uso solicitar información los actores participantes podrán ser:

- Director de Escuela

Descripción del proceso:

Permite al Director de Escuela, autorizar a los estudiantes que puedan realizar el pago de matrícula en partes.

## CU – 17 AUTORIZAR PAGOS CON IECE

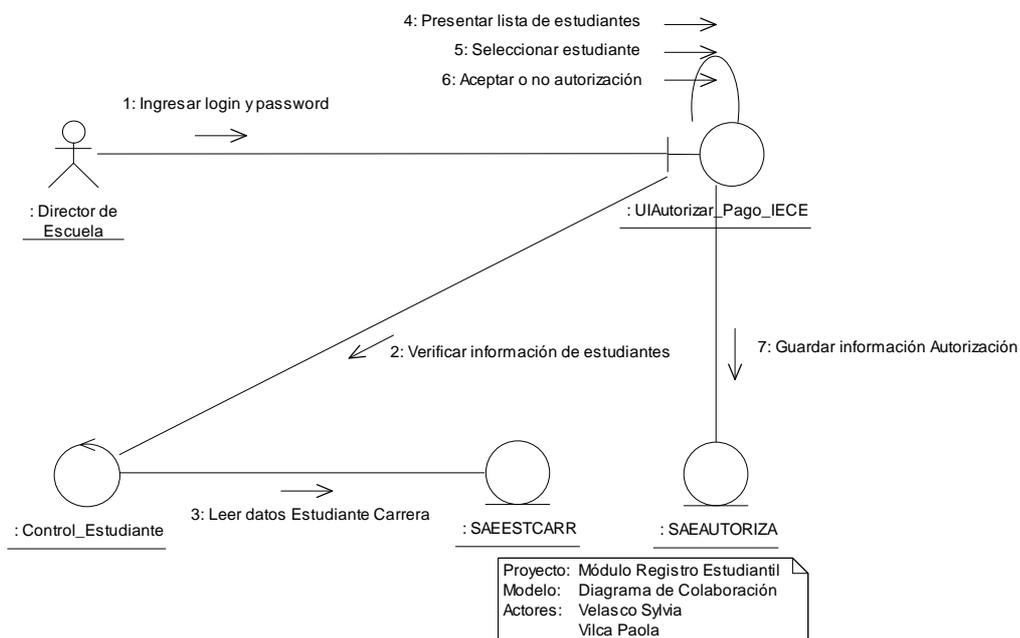


Figura 2.19: Diagrama de colaboración CU – 17 Autorizar pagos con IECE.

En el caso de uso solicitar información los actores participantes podrán ser:

- Director de Escuela

Descripción del proceso:

Permite al Director de Escuela, autorizar a los estudiantes para que realicen el pago de la matrícula con crédito de IECE.

## CU – 18 REGISTRAR IPF

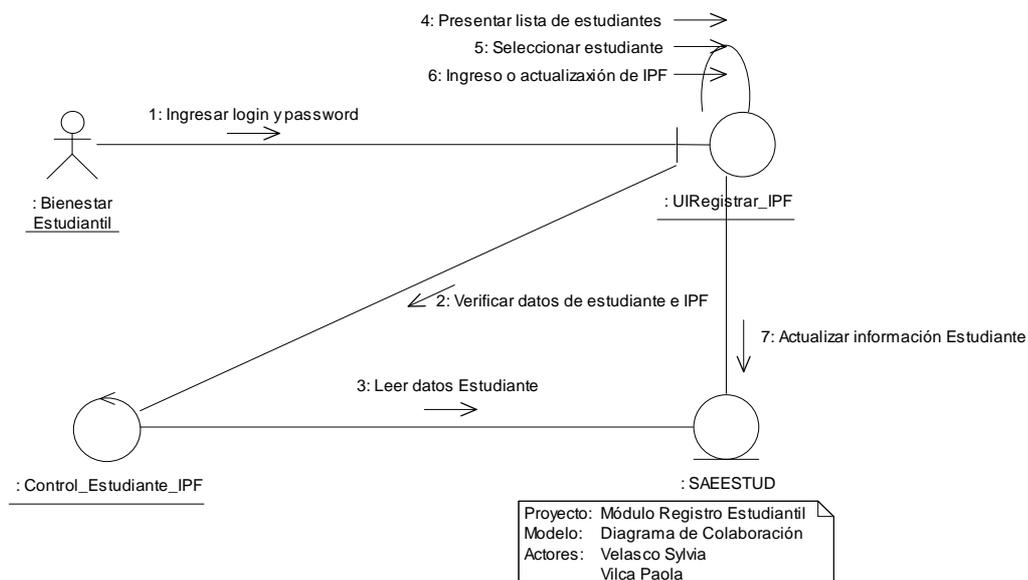


Figura 2.20: Diagrama de colaboración CU – 18 Registrar IPF.

En el caso de uso solicitar información los actores participantes podrán ser:

- Bienestar Estudiantil

Descripción del proceso:

Permite a la Unidad de Bienestar Estudiantil ingresar o actualizar los datos socioeconómicos de un estudiante previo a un análisis de ingresos familiares. Este valor es usado para el cálculo de costos del valor de la matrícula.

## CU – 19 REGISTRAR DESCUENTOS

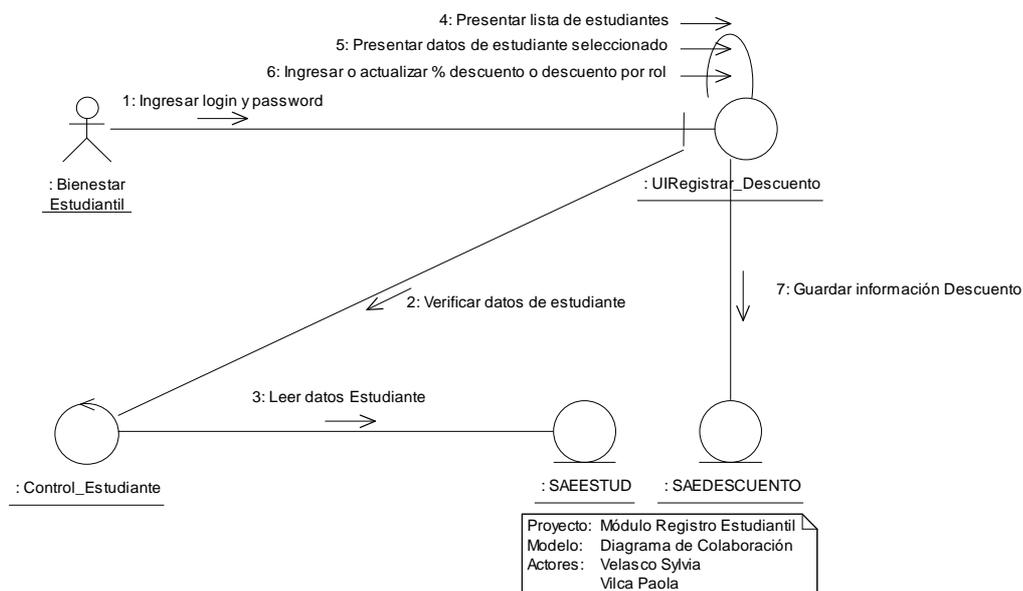


Figura 2.21: Diagrama de colaboración CU – 19 Registrar descuentos.

En el caso de uso solicitar información los actores participantes podrán ser:

- Bienestar Estudiantil

Descripción del proceso:

Permite a la Unidad de Bienestar Estudiantil ingresar el porcentaje de descuentos que se le puede conceder a un estudiante por diversos motivos como hermanos, cónyuges, o hijos de los empleados de la EPN. Para el caso de hijos de empleados de la EPN, se procederá a registrar la deuda por rol si el empleado así lo solicita.

La EPN otorga descuentos especiales del 50% en el pago de aranceles de matrículas y créditos, cursos de inglés y demás programas o cursos académicos excepto los de postgrado, a los hijos de profesores titulares, personal administrativo y de servicios de la institución; y, del 10% en el pago de matrículas y créditos, cursos de inglés y demás programas o cursos académicos, excepto los de postrado, a los estudiantes, hermanos o cónyuges de la EPN, miembros de un

mismo núcleo familiar. Los descuentos no son acumulativos, y son aplicados para los valores de matrícula y créditos. El pago por otros conceptos deberá ser realizado por el estudiante en las fechas establecidas para pagos.

## CU – 20 REGISTRAR BECAS

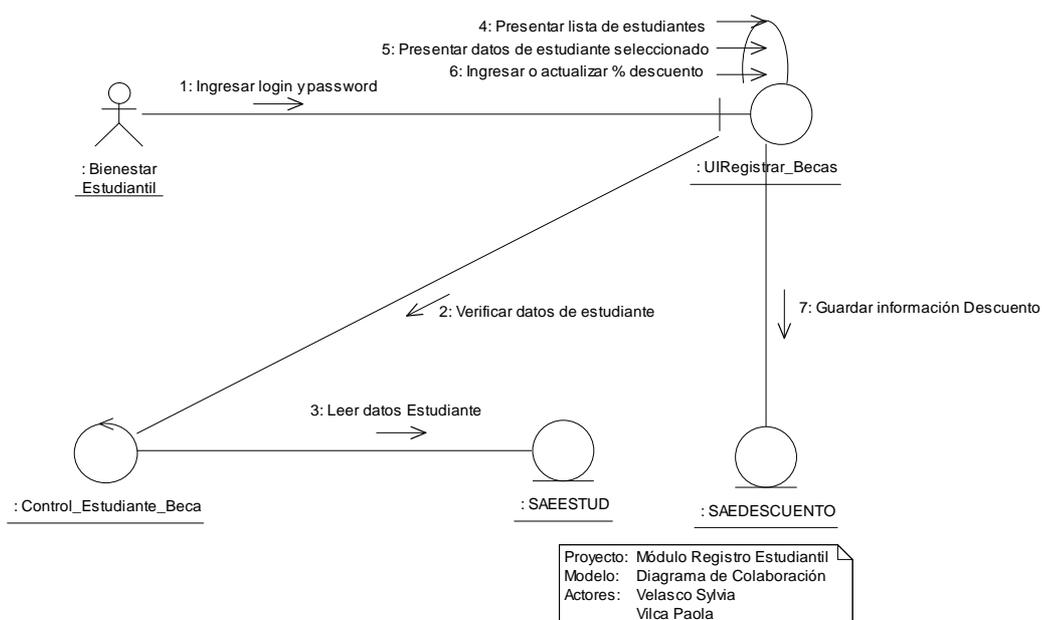


Figura 2.22: Diagrama de colaboración CU – 20 Registrar becas.

En el caso de uso solicitar información los actores participantes podrán ser:

- Bienestar Estudiantil

Descripción del proceso:

Permite a la Unidad de Bienestar Estudiantil ingresar el porcentaje de descuentos que se puede otorgar a los estudiantes, por los siguientes conceptos: merecimiento académico, deportivo o por situación económica.

Las becas son aplicadas para los valores de matrícula y créditos. El pago por otros conceptos deberá ser realizado por el estudiante en las fechas establecidas para pagos.

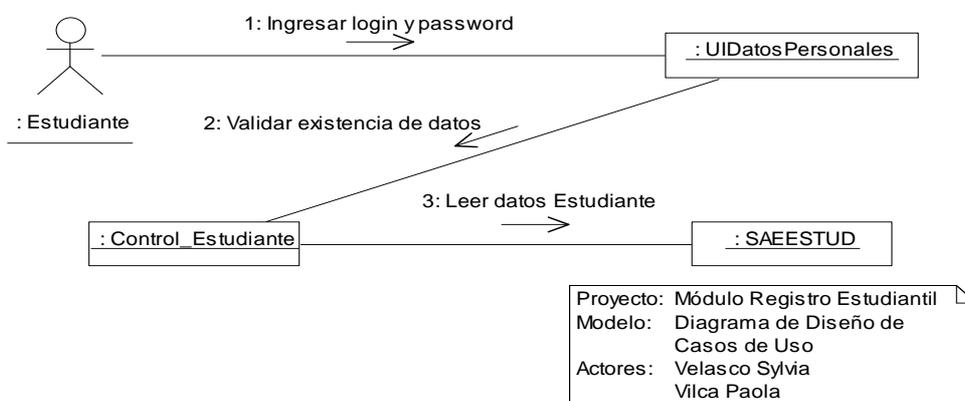
### 2.3.3 DISEÑO

En esta fase presentamos el modelo Lógico del sistema a implementar con ayuda de los siguientes diagramas:

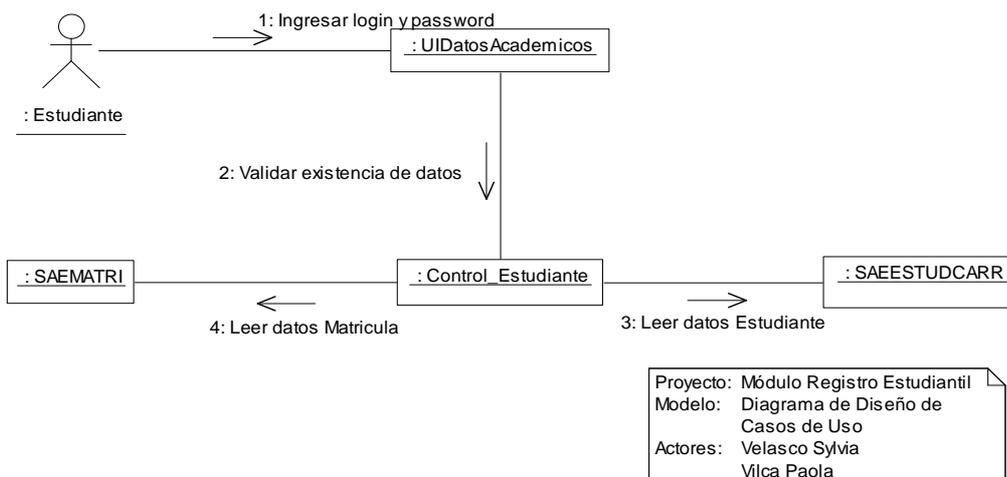
#### 2.3.3.1 Diagrama de Modelo de Casos de Uso de Diseño

##### CU – 01 SOLICITAR INFORMACION

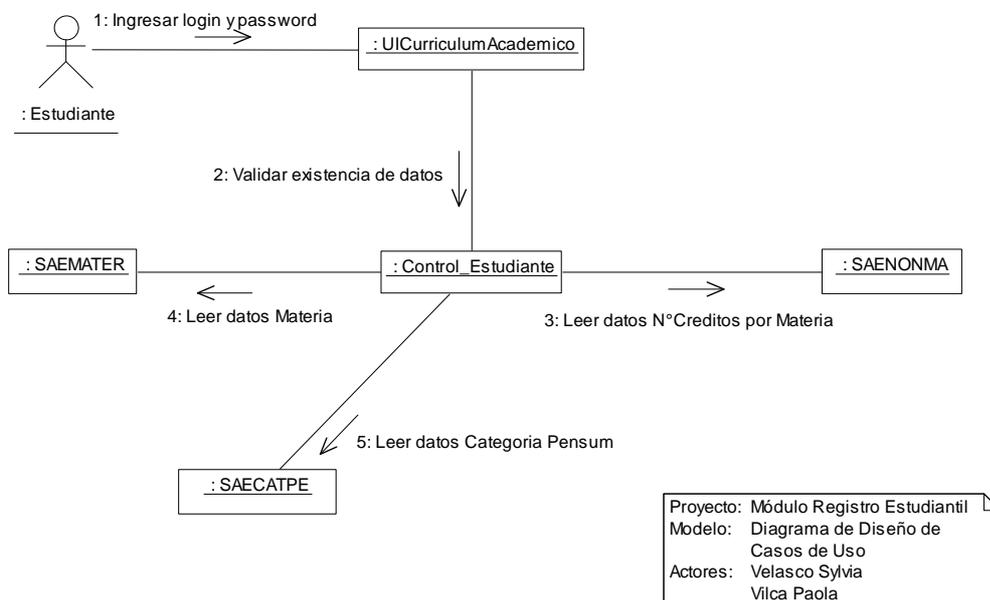
Datos Personales:



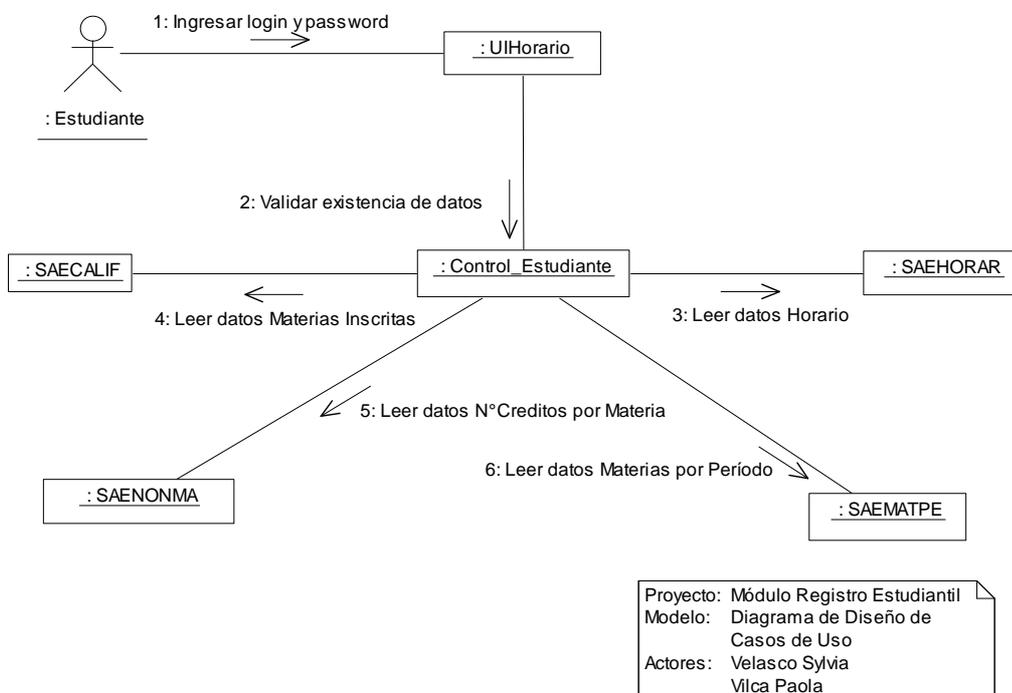
Datos Académicos:



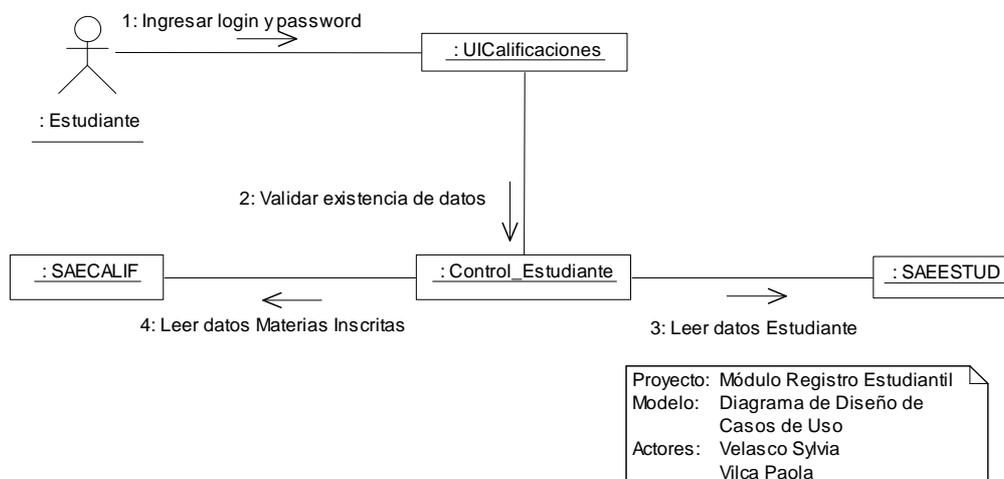
## Currículum Académico:



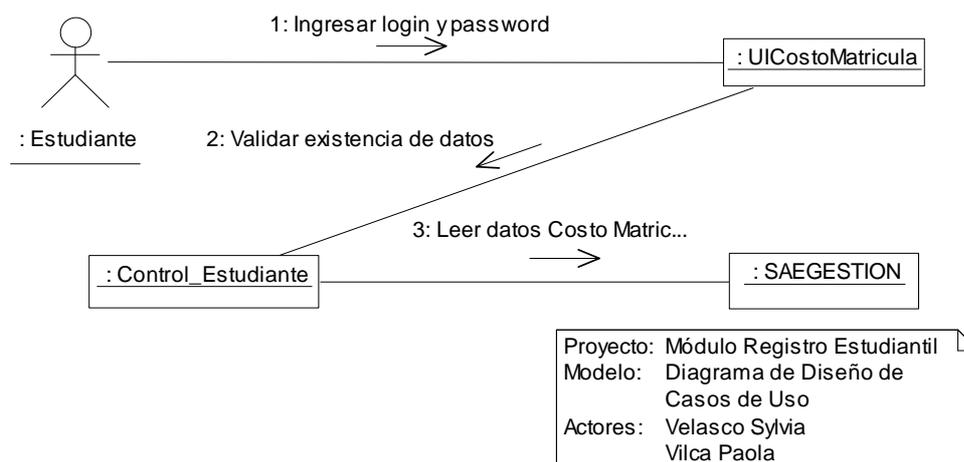
## Consultar Horario:



### Consultar Calificaciones:



### Consultar Costo Matricula:



### Consultar Porcentaje de Materias Aprobadas por Categoría:

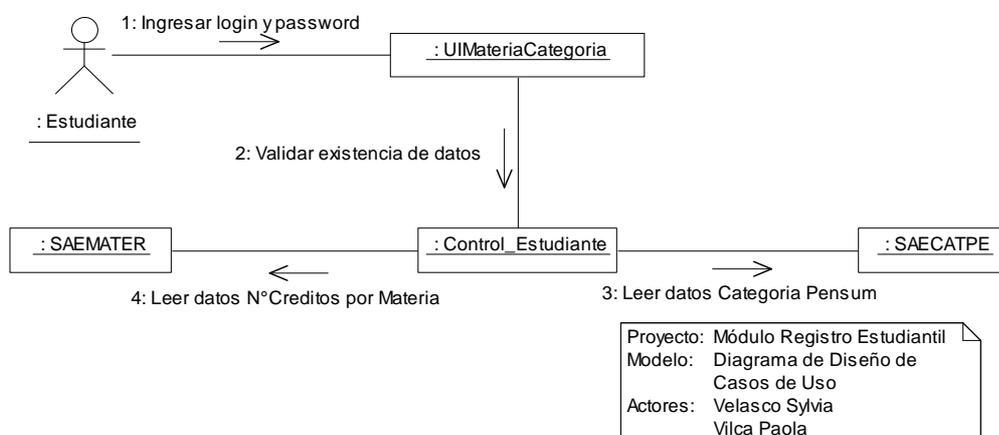


Figura 2.23: Diagrama de Modelo de Casos de Uso de Diseño CU – 01 Solicitar información.

## CU – 02 GENERAR INSCRIPCION

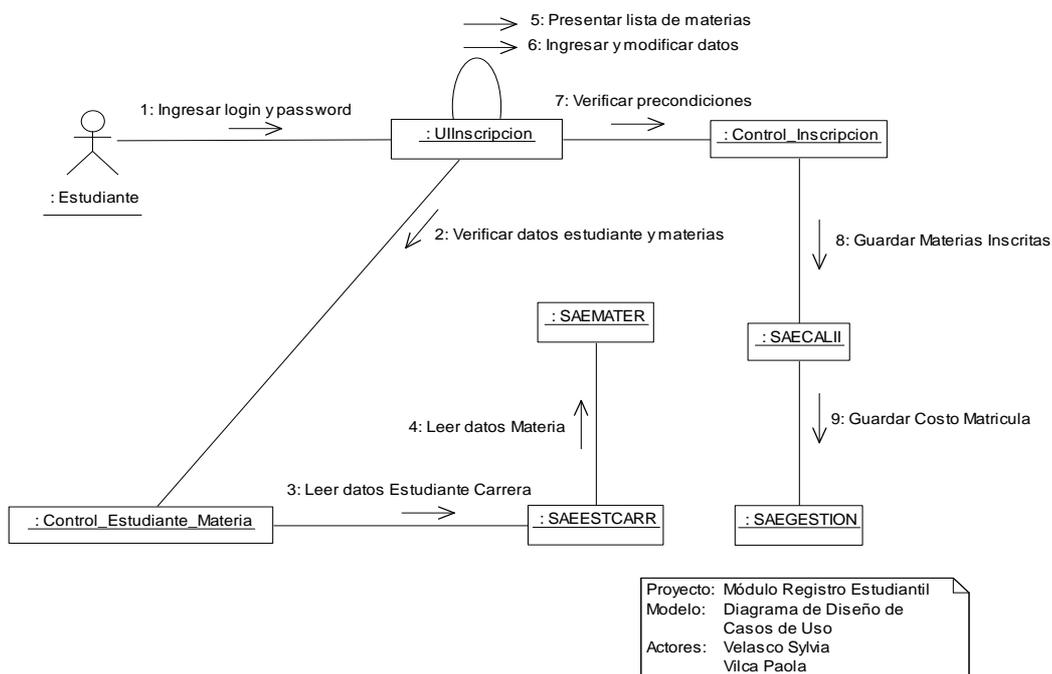


Figura 2.24: Diagrama de Modelo de Casos de Uso de Diseño CU – 02 Generar Inscripción.

## CU – 03 REINSCRIBIR

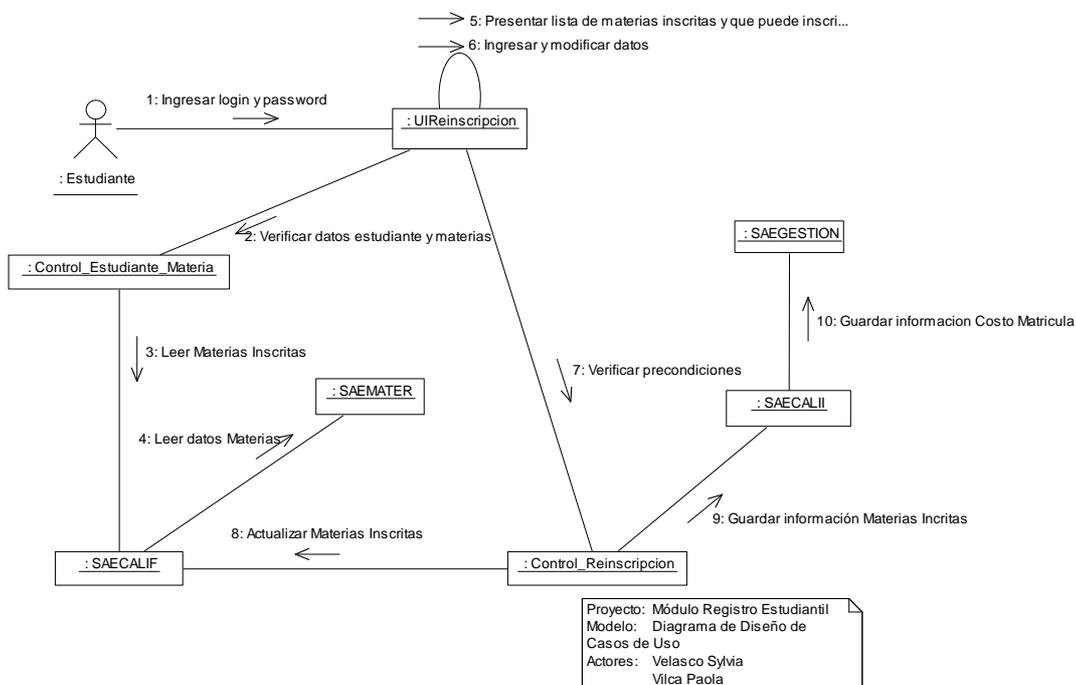


Figura 2.25: Diagrama de Modelo de Casos de Uso de Diseño CU – 03 Reinscribir.

## CU – 04 ANULAR MATRICULA

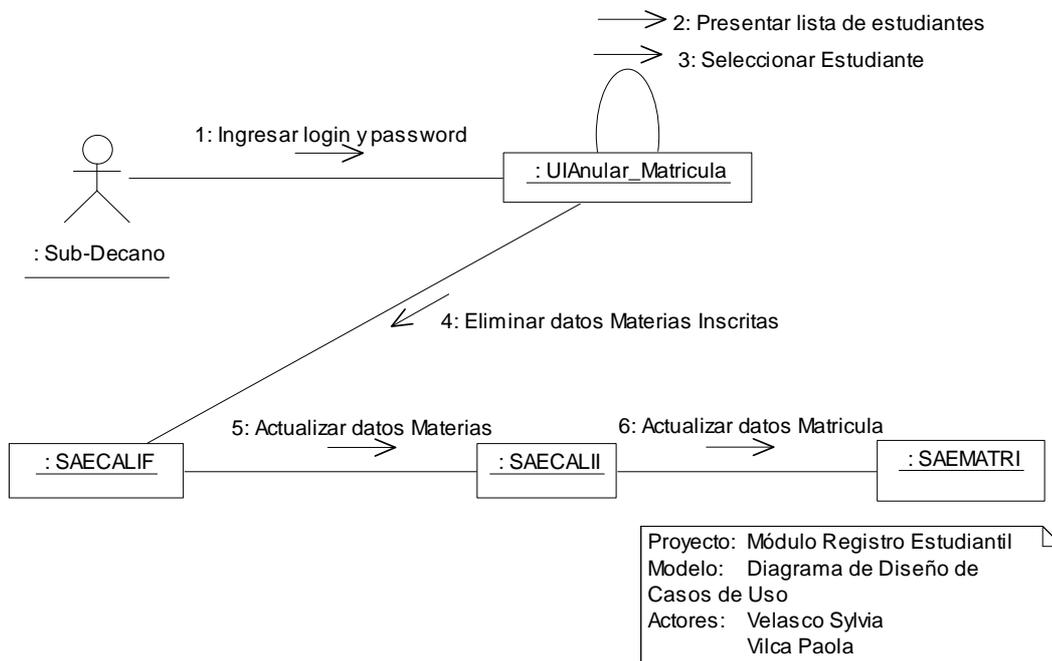


Figura 2.26: Diagrama de Modelo de Casos de Uso de Diseño CU – 04 Anular matricula.

## CU – 05 ENVIAR AL BANCO

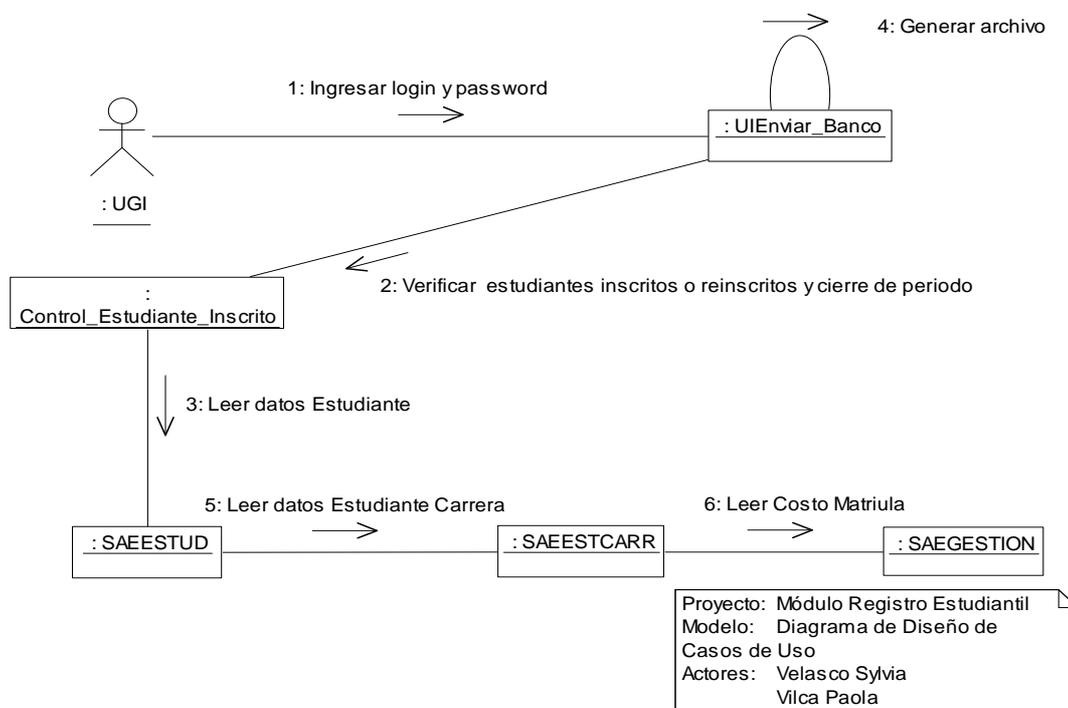


Figura 2.27: Diagrama de Modelo de Casos de Uso de Diseño CU – 05 Enviar al banco.

## CU – 06 PAGAR EN TESORERÍA

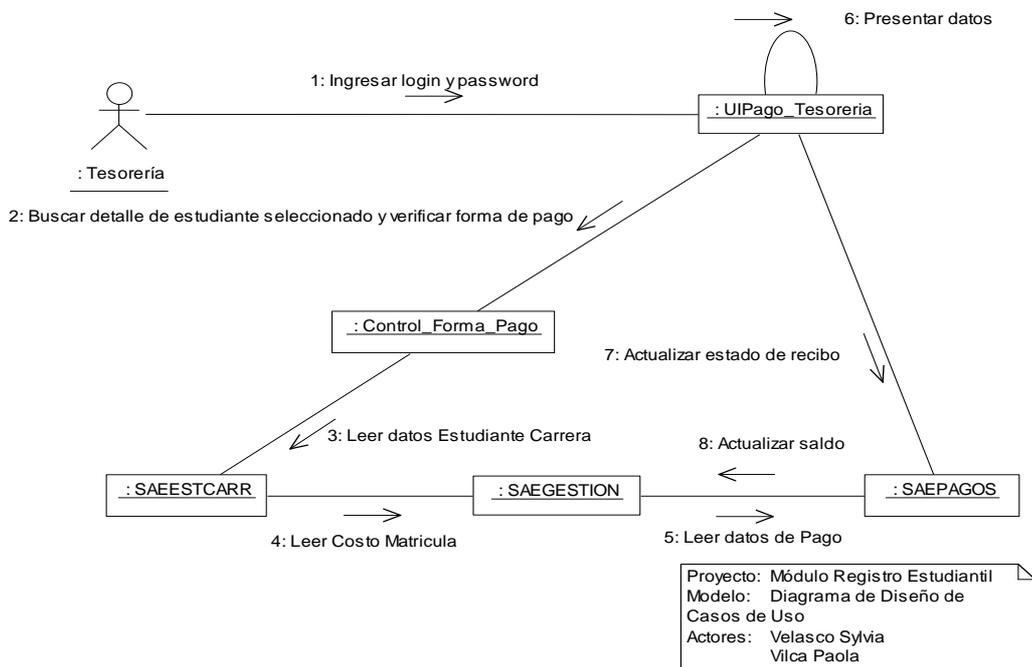


Figura 2.28: Diagrama de Modelo de Casos de Uso de Diseño CU – 06 Pagar en tesorería.

## CU – 07 LEGALIZAR MATRICULA

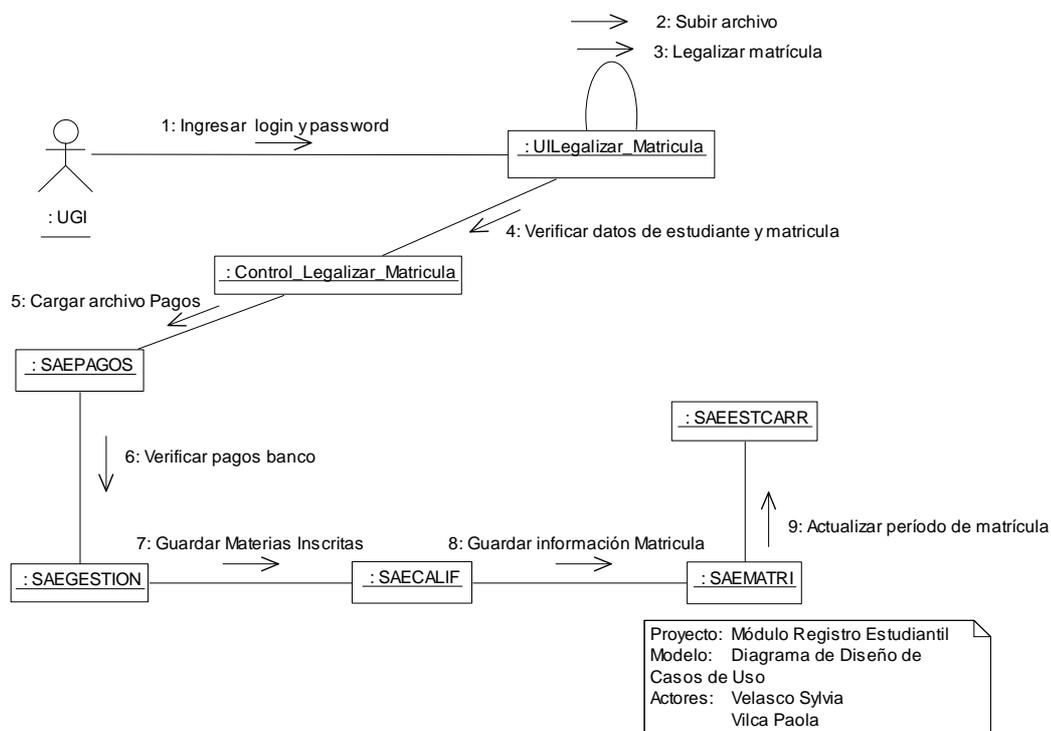


Figura 2.29: Diagrama de Modelo de Casos de Uso de Diseño CU – 07 Legalizar matricula.

## CU – 08 AUTORIZAR REINSCRIPCIONES

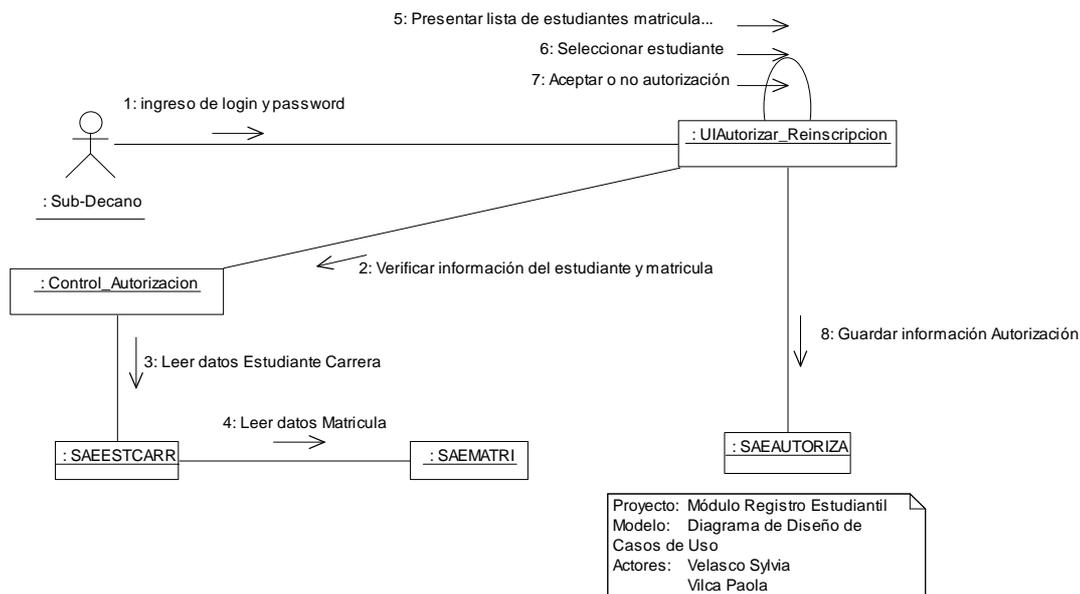


Figura 2.30: Diagrama de Modelo de Casos de Uso de Diseño CU – 08 Autorizar reinscripciones.

## CU – 09 AUTORIZAR REINGRESOS

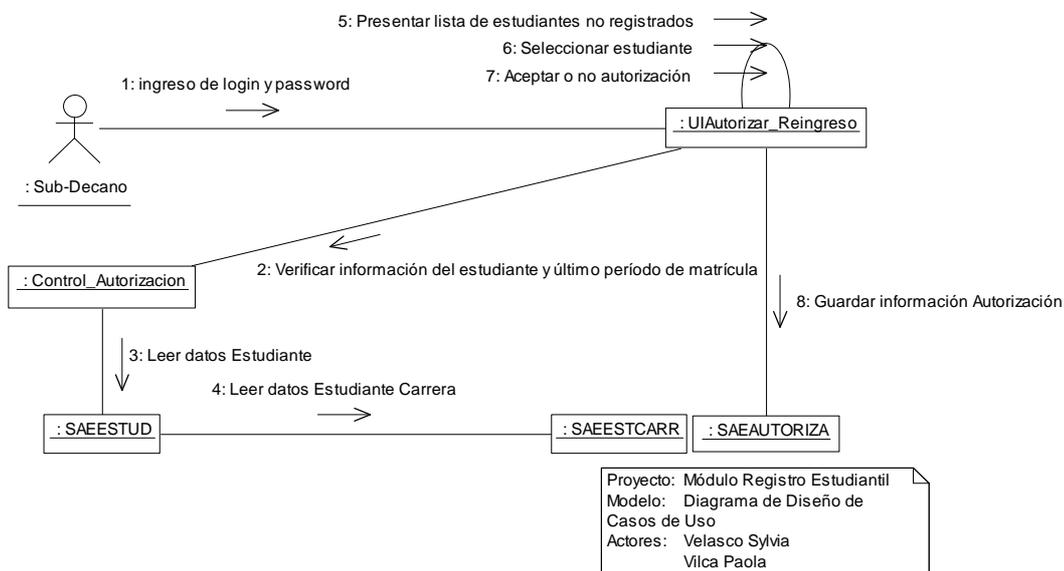


Figura 2.31: Diagrama de Modelo de Casos de Uso de Diseño CU – 09 Autorizar reingresos.

## CU – 10 AUTORIZAR MATRICULAS EXTRAORDINARIAS

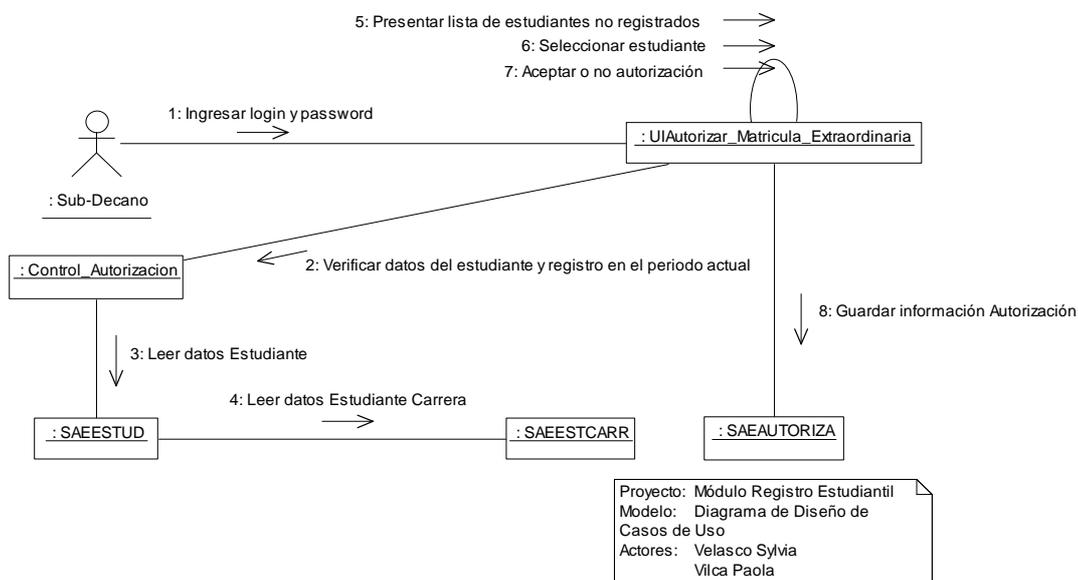


Figura 2.32: Diagrama de Modelo de Casos de Uso de Diseño CU – 10 Autorizar matriculas extraordinarias.

## CU -11 AUTORIZAR MATRICULAS EXTEMPORANEAS

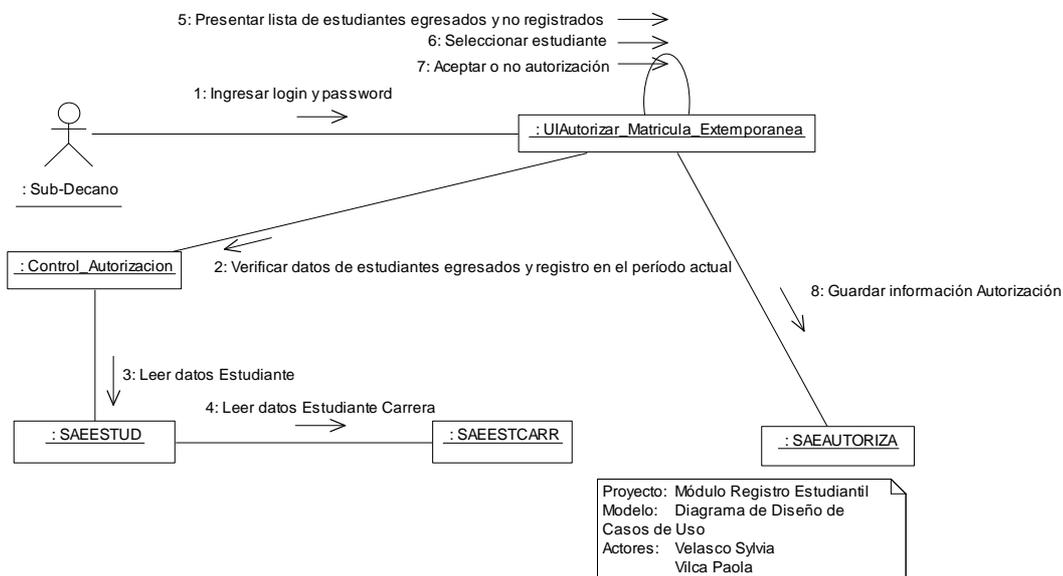


Figura 2.33: Diagrama de Modelo de Casos de Uso de Diseño CU – 11 Autorizar matriculas extemporáneas.

## CU – 12 AUTORIZAR CAMBIOS DE CARRERA/UNIVERSIDAD

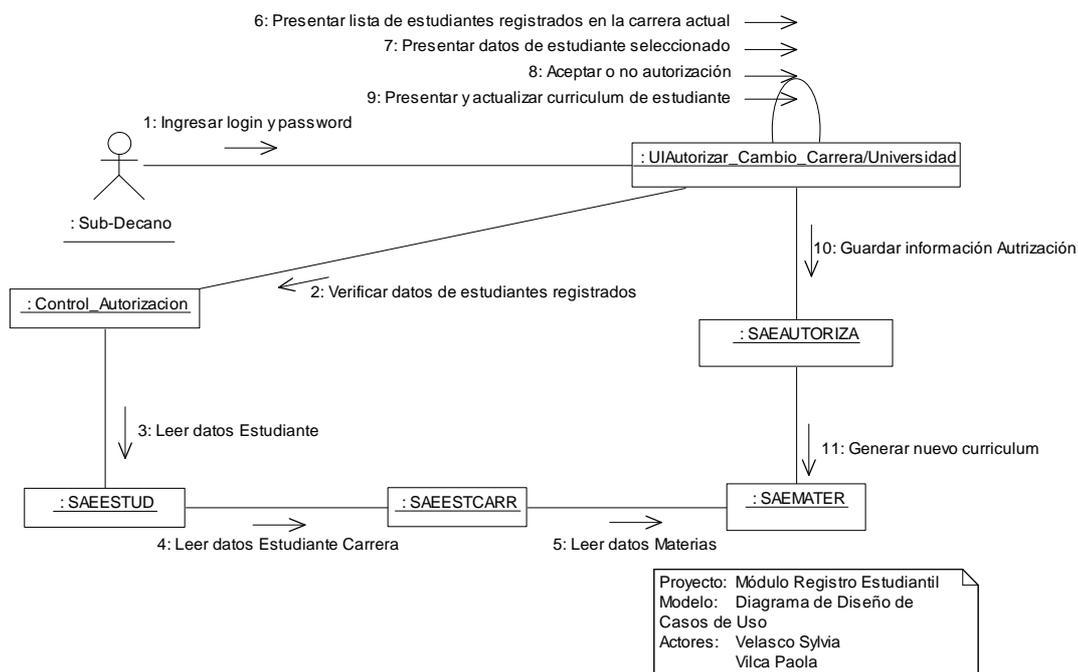


Figura 2.34: Diagrama de Modelo de Casos de Uso de Diseño CU – 12 Autorizar cambios de carrera.

## CU – 13 REGISTRO DE IMPEDIMENTOS DE MATRICULA

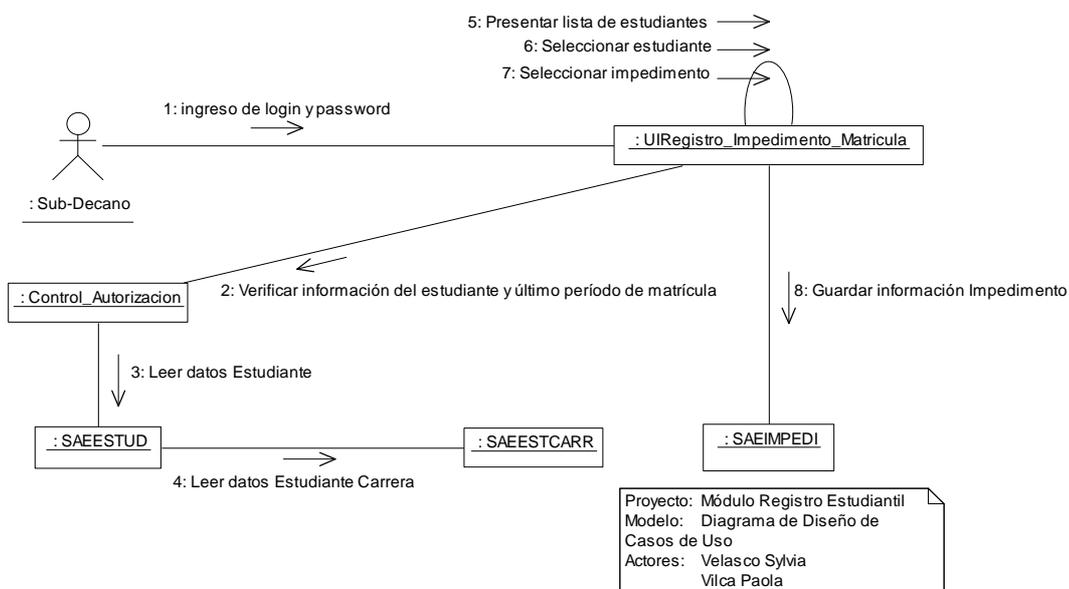


Figura 2.35: Diagrama de Modelo de Casos de Uso de Diseño CU – 13 Registro de impedimentos de matrícula.

## CU – 14 REGISTRAR ASPIRANTE

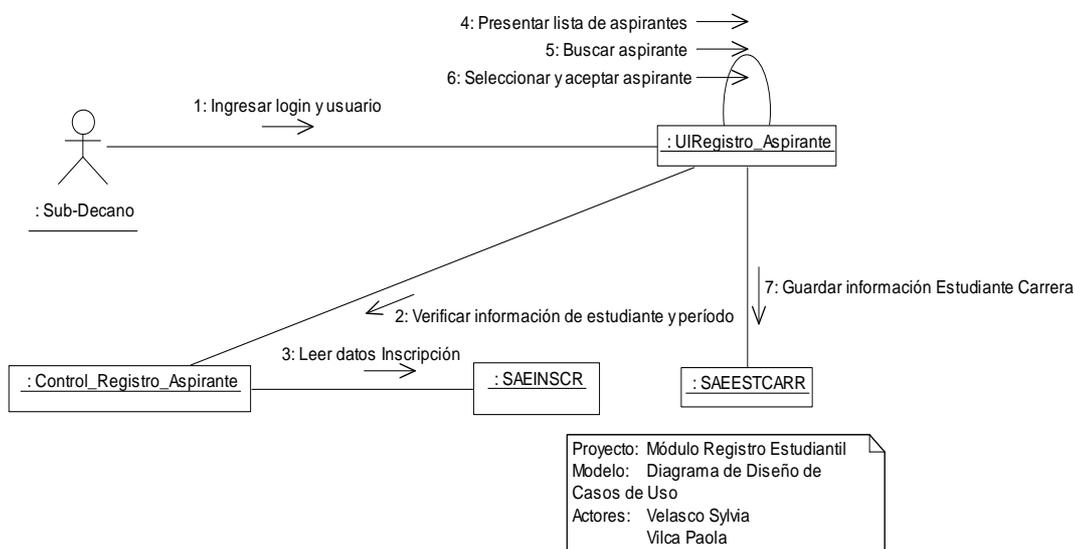


Figura 2.36: Diagrama de Modelo de Casos de Uso de Diseño CU – 14 Registrar aspirante.

## CU – 15 GESTINAR RUBROS ADICIONALES

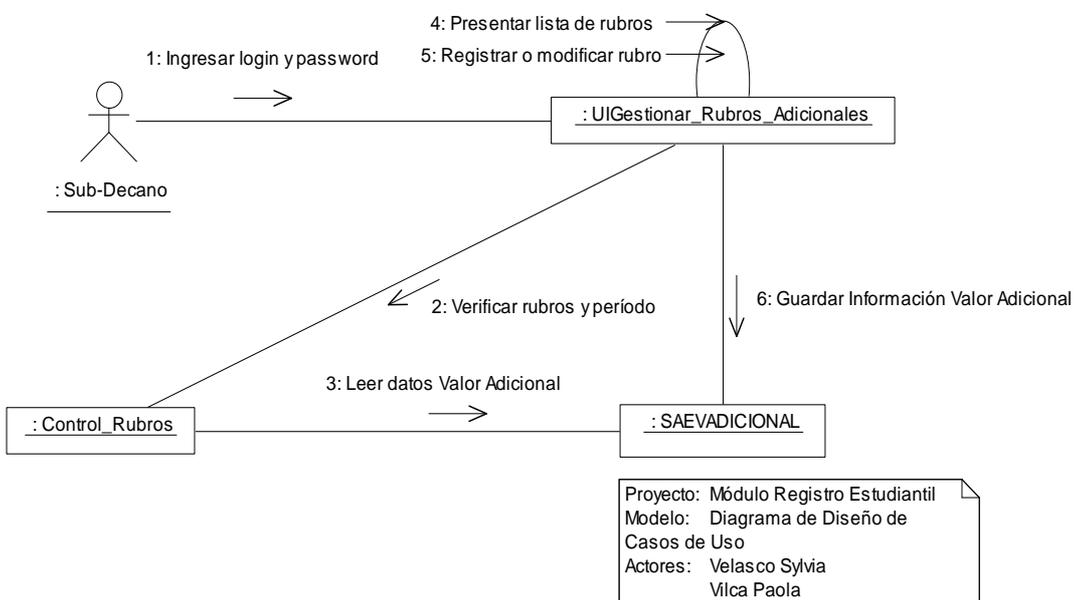


Figura 2.37: Diagrama de Modelo de Casos de Uso de Diseño CU – 15 Registrar rubros adicionales.

## CU – 16 AUTORIZAR PAGOS POR PARTES

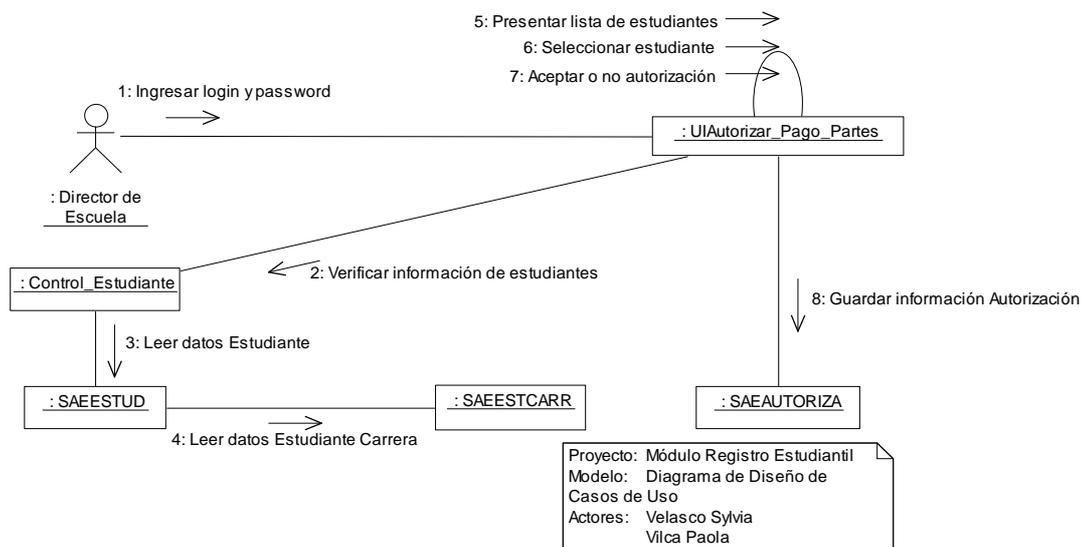


Figura 2.38: Diagrama de Modelo de Casos de Uso de Diseño CU – 16 Autorizar pagos por partes.

## CU – 17 AUTORIZAR PAGOS CON IECE

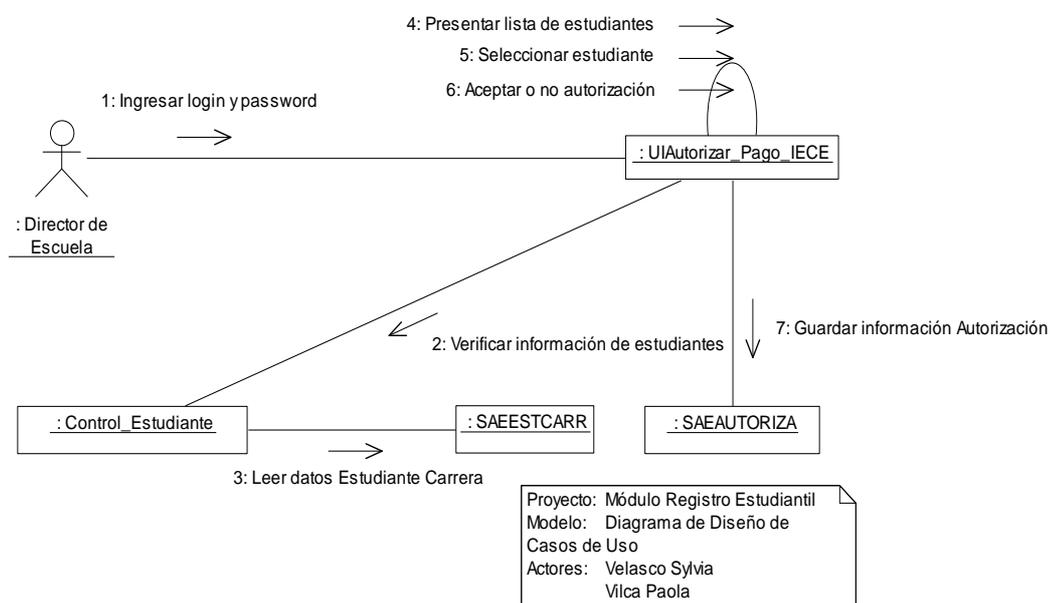


Figura 2.39: Diagrama de Modelo de Casos de Uso de Diseño CU – 17 Autorizar pagos con IECE.

## CU – 18 REGISTRAR IPF

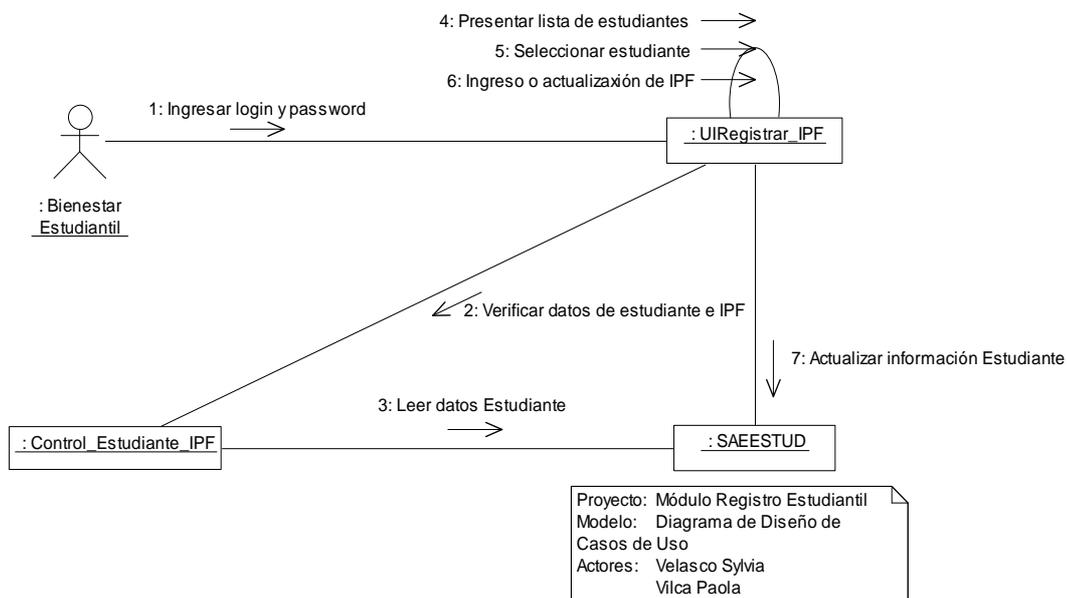


Figura 2.40: Diagrama de Modelo de Casos de Uso de Diseño CU – 18 Registrar IPF.

## CU – 19 REGISTRAR DESCUENTOS

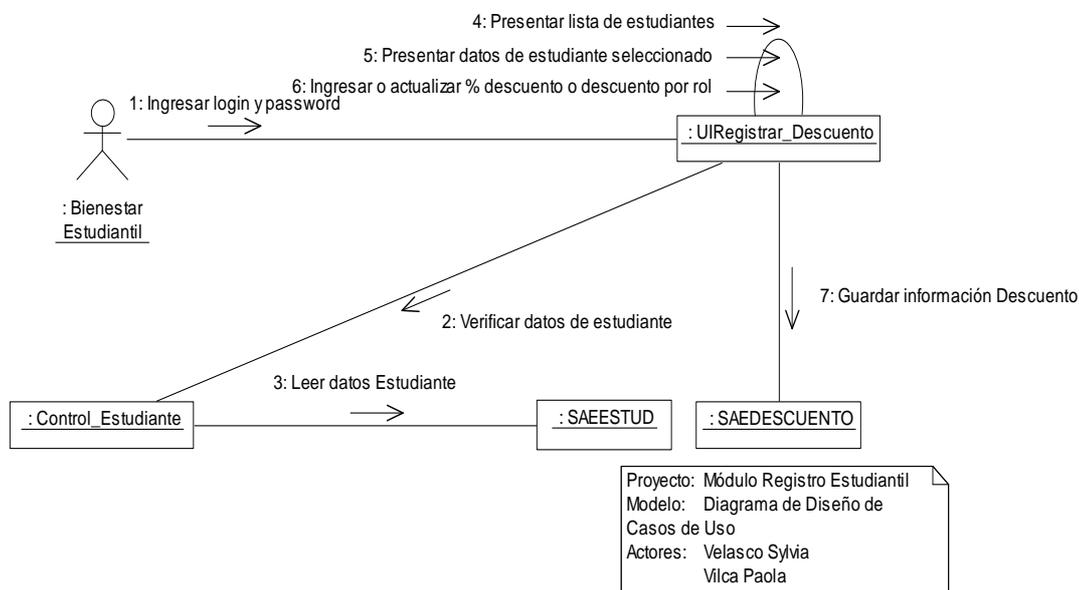


Figura 2.41: Diagrama de Modelo de Casos de Uso de Diseño CU – 19 Registrar descuentos.

## CU – 20 REGISTRAR BECAS

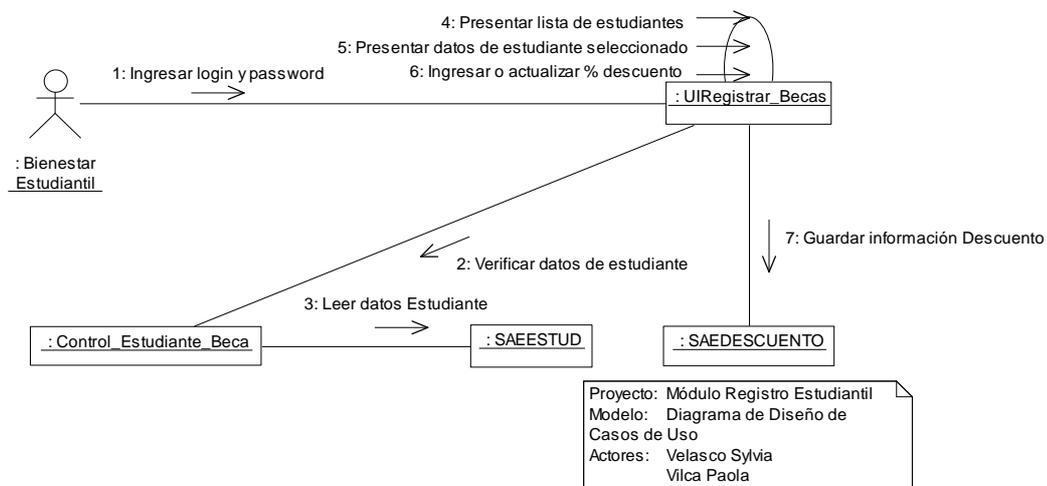
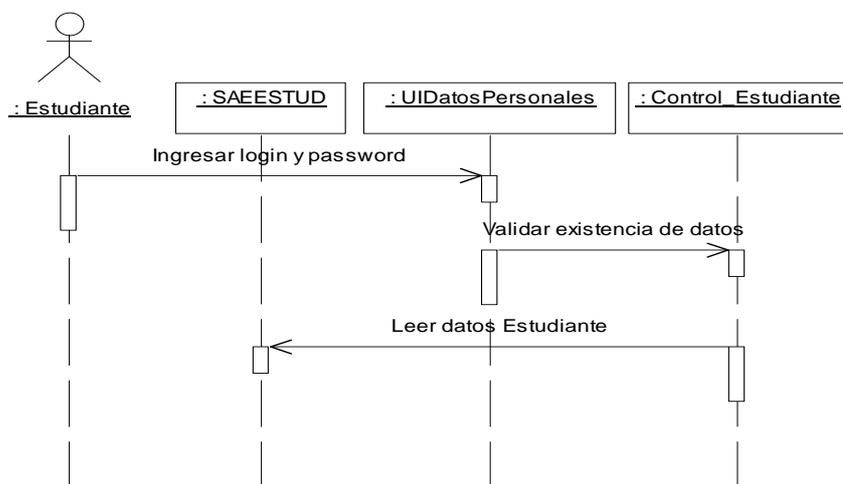


Figura 2.42: Diagrama de Modelo de Casos de Uso de Diseño CU – 20 Registrar becas.

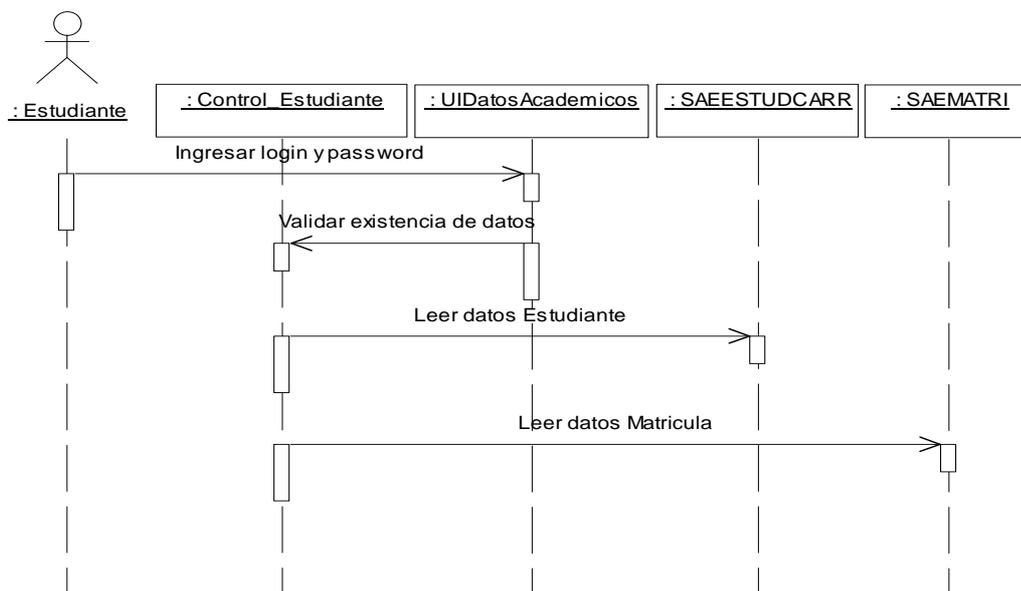
## 2.3.3.2 Diagrama de Secuencia

## CU – 01 SOLICITAR INFORMACION

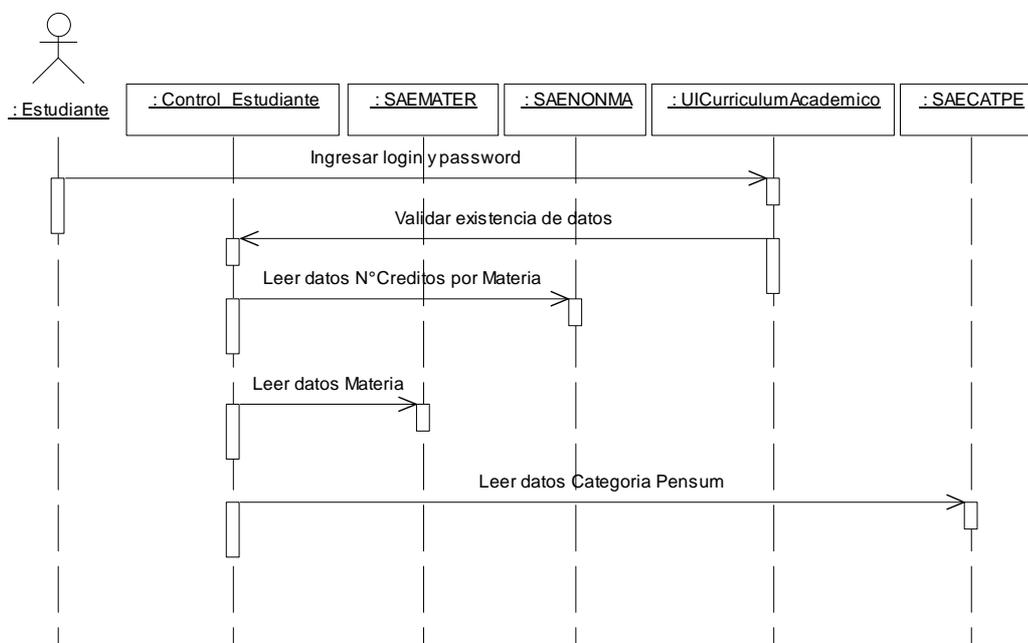
## Datos Personales



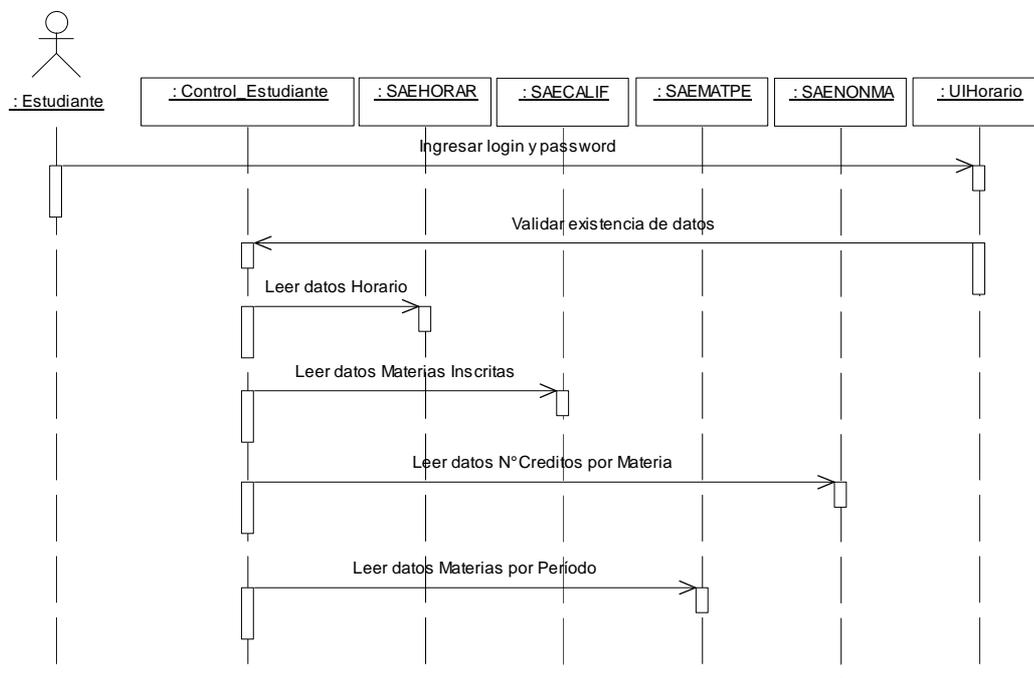
## Datos Académicos



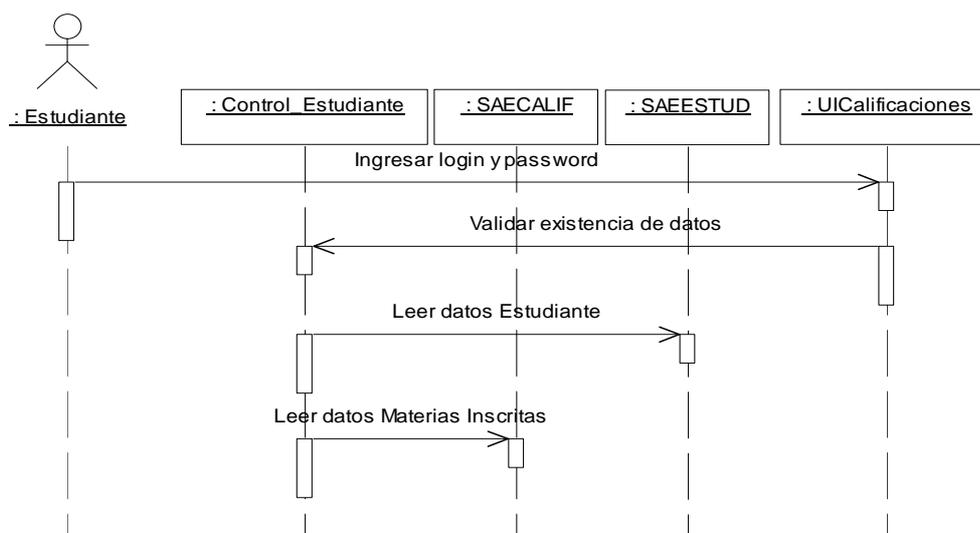
## Currículo Académico



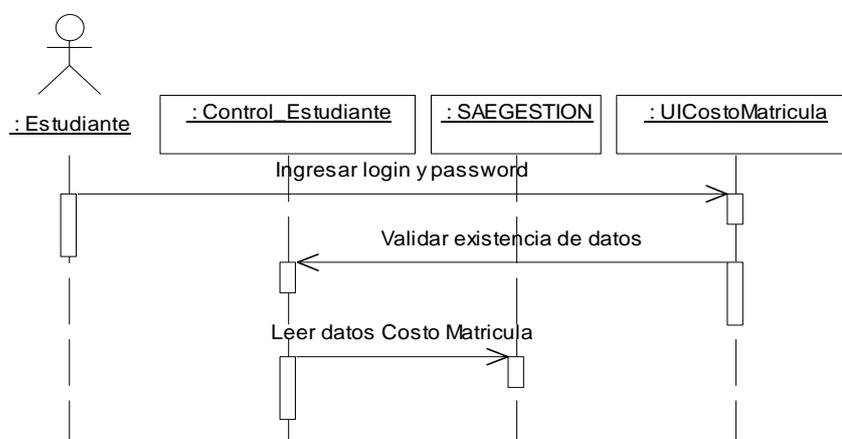
## Consultar Horario



## Consultar Calificaciones



## Consultar Costo Matricula



## Consultar Materias por Categoría

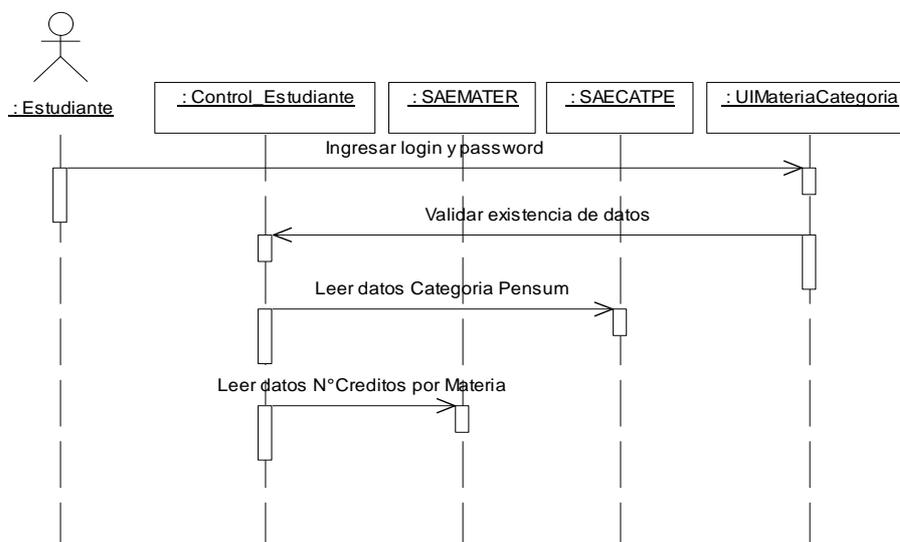


Figura 2.43: Diagrama de Secuencia CU – 01 Solicitar información.

Ver detalle de diagrama de secuencia en *Anexo I*.

### 2.3.3.3 Diagrama de Componentes

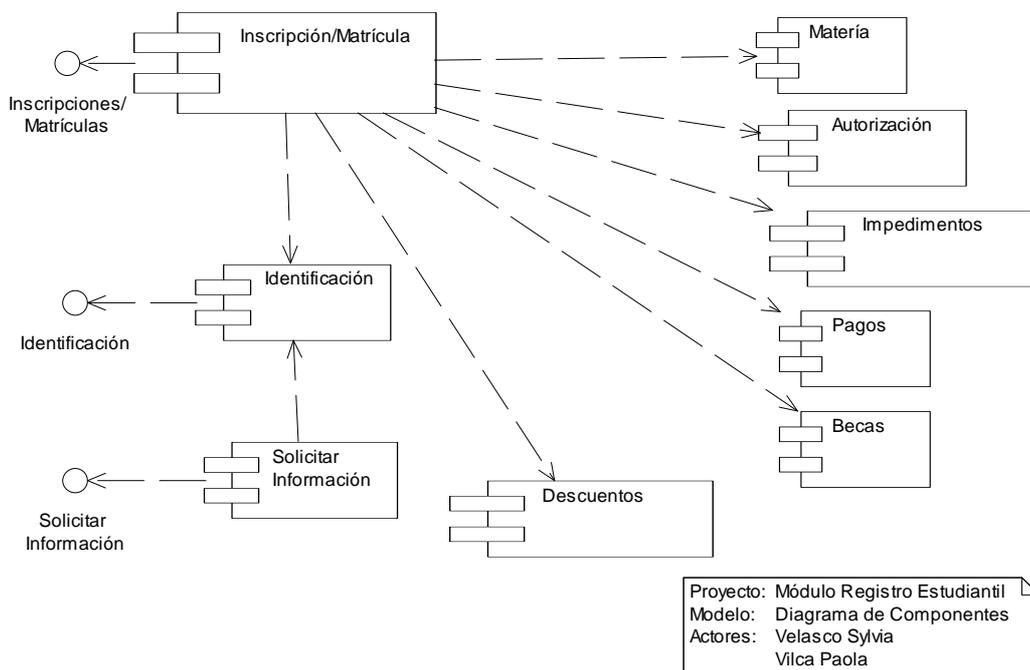


Figura 2.44: Diagrama de Componentes.

### 2.3.3.4 Diagrama de Arquitectura

#### Nivel de Presentación

#### Nivel de Negocio

#### Nivel de Datos

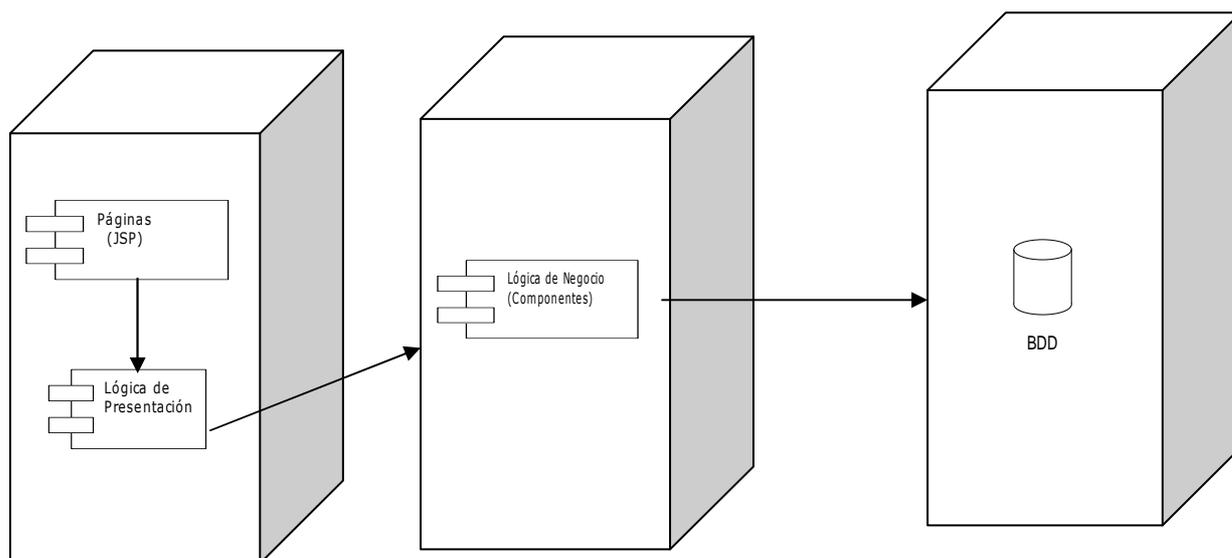


Figura 2.45: Diagrama de Arquitectura.

### 2.3.3.5 Diseño de Interfaces



**Botón “Aceptar”:** Al hacer clic en este link se llama al método validaUsu() en el cual se llama al método del SessionBean llamado valLoginSB(perfil, ci, clave) para validar al usuario y de esta manera presentarle el menú según el perfil.

### UI: Datos Personales



**Link “Datos Personales”:** Al hacer clic en este link se llama al método consultaDatos() en el cual se llama al método del SessionBean llamado datosPersonalesSB(Opcion, Facultad,CodigoCarrera,CodigoEstudiante) para poder presentar los datos del estudiante.

**Botón “Imprimir”:** Botón “Imprimir”: Al hacer clic en el botón se llama al método reporte() en el cual se utiliza un objeto tipo JasperReport de jasperreports-131.jar para setear la ubicación del reporte, además se utiliza el método fillReport(ubicacionReporte,paramReporte,conexionBDD) para efectuar la consulta y almacenar los datos en el objeto de tipo JasperPrint el mismo que será parámetro del método viewReport(print,false) el mismo que permite visualizar los reportes al usuario.

## .UI: Datos Académico



**Link “Datos Académicos”:** Al hacer clic en el link se llama al método `consultarDatosAcademicos()` en el cual se llama al método del SessionBean llamado

`consultarDatosAcademicosSB(CodigoCarrera,CodigoEst,ultimoPeriodoMatri,ultimoSemestreMatri)` el mismo que recibe los parámetros correspondientes para poder presentar los datos académicos del estudiante como fechas de ingreso y del último periodo matriculado.

**Botón “Imprimir”:** Botón “Imprimir”: Al hacer clic en el botón se llama al método `reporteDatosAcademicos()` en el cual se utiliza un objeto tipo JasperReport de `jasperreports-131.jar` para setear la ubicación del reporte, además se utiliza el método `fillReport(ubicacionReporte,paramReporte,conexionBDD)` para efectuar la consulta y almacenar los datos en el objeto de tipo JasperPrint el mismo que será parámetro del método `viewReport(print,false)` el mismo que permite visualizar los reportes al usuario.

## UI: Currículum Académico

				MATERIA	No. MATR.	CALIF.	CREDS.	APROB.	VID	CATEGORIA
2002	1	IPN020		ALGEBRA	1	9.00	6.00	F		PROPEDEUTICO
2002	1	IPN030		GEOMETRIA	1	13.00	6.00	F		PROPEDEUTICO
2002	1	IPN040		FISICA ELEMENTAL	1	24.00	5.00	A		PROPEDEUTICO
2002	1	IPN050		QUIMICA ELEMENTAL	1	31.00	2.00	A		PROPEDEUTICO
2002	1	IPN060		EDUCACION FISICA	1	30.00	2.00	A		PROPEDEUTICO
2002	2	IPN010		ALGEBRA	2	29.00	6.00	E		PROPEDEUTICO
2002	2	IPN020		GEOMETRIA	2	31.00	6.00	E		PROPEDEUTICO
2002	2	IPN030		FISICA ELEMENTAL	2	29.00	7.00	E		PROPEDEUTICO
2003	1	IS1114		TECNOLOGIA DE LA INFORMACION	1	36.00	4.00	E		INFORMATICA Y COMUNICACION
2003	1	IS1122		EXPREISION ORAL Y ESCRITA	1	28.00	2.00	E		INFORMATICA Y COMUNICACION
2003	1	IS1135		ALGEBRA LINEAL	1	12.00	5.00	F		BASICAS
2003	1	IS1146		CALCULO	1	4.00	6.00	F		BASICAS
2003	1	IS1156		FISICA I	1	25.00	6.00	A		BASICAS
2003	1	IS1164		QUIMICA GENERAL	1	28.00	4.00	E		BASICAS
2003	2	IDE100		IDIOMA EXTRANJERO	1	0.00	0.00	A		REQUISITO
2003	2	IS1135		ALGEBRA LINEAL	2	28.00	5.00	E		BASICAS

**Link “Curriculum Académico”:** Al hacer clic en el link se llama al método consultarCurriculumAcademico() en el cual se llama al método que carga el como de los periodos cargaComboPeriodosMatriculados() el mismo que llama al SessionBean llamado periodosMatriculadosSB (Opcion,CodigoCarr,CodigoEst) luego de cargar los periodos se procede a presentar el resto de datos del currículo por medio del método consultarCurriculumSB(Opcion,CodigoCarr,CodigoEst,filtroPeriodo,filtroSemestre) que también es parte del SessionBean para poder presentar los datos del currículo académico de cada estudiante.

**Combo “Periodo”:** Al hacer clic en el combo se llama al método periodoSeleccionado() el cual filtra la información a presentar.

**Botón “Imprimir”:** Botón “Imprimir”: Al hacer clic en el botón se llama al método reporteCurriculumAcademico() en el cual se utiliza un objeto tipo JasperReport de jasperreports-131.jar para setear la ubicación del reporte, además se utiliza el método fillReport(ubicacionReporte,paramReporte,conexionBDD) para efectuar la consulta y almacenar los datos en el objeto de tipo JasperPrint el mismo que será parámetro del metodo viewReport(print,false) el mismo que permite visualizar los reportes al usuario.

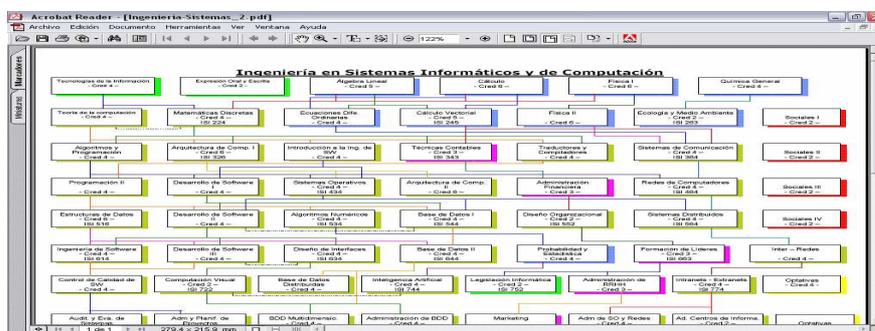
## UI: Créditos por Categoría

CATEGORÍA	REFERENCIA	APROBADOS	POR APROBAR	APROBADO
BÁSICAS	42	40.00	2.00	95%
COMUNICACIÓN Y COMUNICACION	8	8.00	0.00	100%
OBLIGATORIAS	138	94.00	44.00	68%
OPORTATIVAS	7	7.00	0.00	100%
ADMINISTRACIÓN, FINANZAS Y ECONOMÍA	15	12.00	3.00	80%
SOCIALES Y HUMANÍSTICAS	8	8.00	0.00	100%
PROYECTO TITULACIÓN	20	0.00	20.00	0%

**Link “Créditos por Categoría”:** Al hacer clic en el link se llama al método consultarCreditoCategoria () en el cual se llama al método del SessionBean llamado porcentajeMateriasAprobadasSB(Opcion,CodigoCarr,CodigoEst) el cual recibe los respectivos parámetros para presentar los porcentajes de las categorías de materias aprobadas.

**Boton “Imprimir”**: Botón “Imprimir”: Al hacer clic en el botón se llama al método `reporteCreditosAprobados()` en el cual se utiliza un objeto tipo `JasperReport` de `jasperreports-131.jar` para setear la ubicación del reporte, además se utiliza el método `fillReport(ubicacionReporte,paramReporte,conexionBDD)` para efectuar la consulta y almacenar los datos en el objeto de tipo `JasperPrint` el mismo que será parámetro del método `viewReport(print,false)` el mismo que permite visualizar los reportes al usuario.

## UI: Solicitar Malla Curricular



**Link “Malla Curricular”**: Al hacer clic en este link se llama a un método `verMallaCurricular()` en JavaScript el mismo que permite ver en formato .pdf las mallas de las distintas carreras de la EPN.

## UI: Horarios

The image shows a screenshot of the "SISTEMA DE ADMINISTRACIÓN ESTUDIANTIL" web application. The header includes the logo of the "ESCUELA POLITÉCNICA NACIONAL" and the date "20/10/2007". The main content area displays the "HORARIO DE CLASES : PERIODO ACTUAL" for the "INGENIERIA EN SISTEMAS INFORMATICOS Y DE COMPUTACION" program. A table shows the schedule for "PARALELO GR.1" with columns for "CREDITOS", "MATRICULA", "LUNES", "MARTES", "MIÉRCOLES", "JUEVES", and "VIERNES". The table shows 2.00 credits and 1 matriculation. A "Sala" button is visible in the top right corner, and a "Imprimir" button is visible in the top right corner of the table area. A navigation menu on the left includes options like "Inscripción y Matriculación", "Datos Personales", "Datos Académicos", "Currículum Académico", "Creditos por Categoría", "Malla Curricular", "Calificaciones", "Perfiles Matriculados", and "Costo de Matrícula".

**Link “Horario”**: Al hacer clic en el link se llama al método del `SessionBean` llamado `horarioSB(Opcion,nombreFacu,CodigoCarr,CodigoEst,periodo,semestre)` el mismo que recibe los parámetros necesarios para poder presentar el horario de las materias inscritas.

**Botón “Imprimir”**: Botón “Imprimir”: Al hacer clic en el botón se llama al método `reporteHorario()` en el cual se utiliza un objeto tipo `JasperReport` de `jasperreports-131.jar` para setear la ubicación del reporte, además se utiliza el método `fillReport(ubicacionReporte,paramReporte,conexionBDD)` para efectuar la consulta y almacenar los datos en el objeto de tipo `JasperPrint` el mismo que será parámetro del método `viewReport(print,false)` el mismo que permite visualizar los reportes al usuario.

## UI: Calificaciones



**Link “Calificaciones”**: Al hacer clic en el link se llama al método del `SessionBean` `consultarCalificacionesSB(Opcion,nombreFacu,CodigoCarr,CodigoEst,periodo,se mestre)` el mismo que recibe los parámetros necesarios para poder presentar el detalle del horario de las materias inscritas.

## UI: Períodos Matriculados



**Link “Periodos Matriculados”:** Al hacer clic en el link se llama al método del SessionBean llamado `periodosMatriculadosSB(Opcion,CodigoCarr,CodigoEst)` el mismo que recibe los parámetros necesarios para poder presentar los períodos matriculados.

**Botón “Imprimir”:** Botón “Imprimir”: Al hacer clic en el botón se llama al método `reportePeriodosMatriculados()` en el cual se utiliza un objeto tipo `JasperReport` de `jasperreports-131.jar` para setear la ubicación del reporte, además se utiliza el método `fillReport(ubicacionReporte,paramReporte,conexionBDD)` para efectuar la consulta y almacenar los datos en el objeto de tipo `JasperPrint` el mismo que será parámetro del método `viewReport(print,false)` el mismo que permite visualizar los reportes al usuario.

## UI: Costo Matricula



**Link “Costo Matricula”:** Al hacer clic en el link se llama al método del SessionBean llamado `consultarFechasMatriculasSB(Opcion,CodigoCarr,periodo,semestre)` el mismo que recibe los parámetros necesarios y filtrando los datos a presentar por un rango de fechas.

Los diseños faltantes *Ver Anexo 2.*

### 2.3.3.6 Diseño de Pruebas

#### Meta:

La meta principal de las pruebas por realizarse es la aceptación del sistema desarrollado por parte de los usuarios.

#### Objetivo:

- Comprobar la correcta implementación del sistema por parte de los desarrolladores.
- Verificar que el sistema cumpla con todos los requerimientos y expectativas del usuario final.
- Verificar la funcionalidad completa del sistema.
- Verificar rendimiento del sistema.

Tipos de pruebas por realizar<sup>12</sup>:

- Pruebas de unidad.- Este tipo de pruebas son realizadas típicamente por los programadores y no por los equipos de pruebas independientes, ya que este requiere un conocimiento detallado del diseño del producto de software.

Casos de Prueba: Ingreso al Sistema

<b>Fecha:</b>	10-04-2007
<b>Responsable:</b>	Desarrollador de la Aplicación
<b>Tarea:</b>	Verificar el método valLogin(string opcion, string cedula, string clave)
<b>Entrada:</b>	Caso 1 : Datos Correctos Perfil: Estudiante C.I: '0201666765' Clave: '1602'

<sup>12</sup> Capítulo 3: Pag 46-47; MODELO PARA PRUEBAS DE SOFTWARE; GUERRA BRAVO MONICA PAOLA; Mayo 2001.

<b>Resultados Esperado:</b>	Mensaje "Usuario Permitido"
-----------------------------	-----------------------------

Tabla 2.24: Casos de Prueba: Ingreso al Sistema.

## Caso de Prueba: Información del Estudiante

<b>Fecha:</b>	10-04-2007
<b>Responsable:</b>	Desarrolladores de la Aplicación
<b>Tarea:</b>	Verificar el método datosPersonales(string opcion, string facultad,string codCarr,string codEst)
<b>Entrada:</b>	opcion= 'C' facultad="" codCarr: '07' codEst: '9910878'
<b>Resultados Esperado:</b>	Devuelve un objeto Estudiante que permite presentar los datos del Estudiante

Tabla 2.25: Casos de Prueba: Información del Estudiante.

## Caso de Prueba: Generar Inscripción

<b>Fecha:</b>	10-04-2007
<b>Responsable:</b>	Desarrollador de la Aplicación
<b>Tarea:</b>	Verificar el método impedimentos(string opcion, string nomFacu, string codCarr, string codEst, string op)
<b>Entrada:</b>	Opcion= " nomFacu="" codCarr: '07' codEst: '0210249' op='I'

<b>Resultados Esperado:</b>	Devolver cadena vacía
-----------------------------	-----------------------

Tabla 2.26: Casos de Prueba: Generar Inscripción.

Los casos de prueba faltante *Ver Anexo 3*.

Para la realización de pruebas unitarias se utilizará el formulario PUNI – 01.

*Ver Anexo 4*

- Pruebas de sistema (aplicación).- Pruebas del software desarrollado completo en un ambiente lo mas parecido posible al ambiente de producción, por personas diferentes a aquellas que construyeron el programa. Estas pruebas están orientadas a verificar que el software desarrollado cumpla con todas las especificaciones de requerimientos del sistema establecidas para el producto.

#### Caso de Prueba: Información de Estudiante

<b>Fecha:</b>	11-04-2007
<b>Responsable:</b>	Equipo de Prueba – Desarrolladores
<b>Tarea:</b>	Asegurar el procesamiento y recuperación de datos.
<b>Entrada:</b>	Estudiante: 0220195
<b>Resultados Esperado:</b>	Presentar datos e Impresión

Tabla 2.27: Casos de Prueba: Información del Estudiante

#### Caso de Prueba: Generar Inscripción

<b>Fecha:</b>	11-04-2007
<b>Responsable:</b>	Equipo de Prueba – Desarrolladores
<b>Tarea:</b>	Asegurar el procesamiento y recuperación de datos.
<b>Entrada:</b>	Estudiante: 0220195

<b>Resultados Esperado:</b>	Terminar proceso de inscripción , presentando datos correctos.
-----------------------------	---

Tabla 2.28: Casos de Prueba: Generar Inscripción

Los casos de prueba faltante *Ver Anexo 3.*

Para la realización de pruebas de aplicación se utilizará el formulario PAPLI – 01. *Ver Anexo 4.*

- Pruebas de aceptación (con usuarios).- Es la ejecución de pruebas finales de software desarrollado bajo condiciones acordadas por el cliente para demostrar que el software satisface la especificación de requerimientos del cliente sobre un periodo limitado de tiempo y en el ambiente de explotación definido por el cliente.

#### Caso de Prueba: Información Estudiante

<b>Fecha:</b>	11-04-2007
<b>Responsable:</b>	Usuarios (Estudiantes, Subdecano, UGI)
<b>Tarea:</b>	Asegurar la navegación a través de la aplicación, movimientos del ratón y teclas de función. Las ventanas y sus características, como menús, tamaño, posición y estado cumplen los estándares
<b>Entrada:</b>	Menú del Estudiante
<b>Resultados Esperado:</b>	Aceptación del diseño.

Tabla 2.29: Casos de Prueba: Información del Estudiante

## Caso de Prueba: Generar Inscripción

<b>Fecha:</b>	11-04-2007
<b>Responsable:</b>	Usuarios (Estudiantes, Subdecano, UGI)
<b>Tarea:</b>	Asegurar la navegación a través de la aplicación, movimientos del ratón y teclas de función. Las ventanas y sus características, como menús, tamaño, posición y estado cumplen los estándares
<b>Entrada:</b>	Pantalla Inscripción
<b>Resultados Esperado:</b>	Aceptación del diseño.

Tabla 2.30: Casos de Prueba: Información del Estudiante

Los casos de prueba faltante *Ver Anexo 3*.

Para la realización de pruebas con usuarios se utilizará el formulario PUSU – 01. *Ver Anexo 4*.

- Pruebas de carga y estrés.- Esta basada en aplicaciones bajo cargas reales y en horas pico donde se manejen grandes cantidades de datos y existan altos índices de concurrencia, para determinar las degradaciones del sistema.

Este tipo de pruebas no serán realizadas para el desarrollo de este prototipo ya que este tipo de pruebas se debe realizar en un ambiente de producción el cual no es accesible, a demás que se necesita la participación de terceros.

## CAPITULO 3: CONSTRUCCIÓN Y PRUEBAS

### 3.1 SELECCIÓN DE LAS HERRAMIENTAS

#### 3.1.1 HERRAMIENTAS DE BACK – END

- a) La selección se realizará en base a la información de la *Tabla 3.1* en la cual se presenta una comparación entre las herramientas de Back – End.

MySQL	PostgreSQL	Firebird
Multiplataforma	Multiplataforma	Multiplataforma
Fácil instalación	Compleja instalación	Compleja instalación
Fácil de utilizar y administrar	Fácil de utilizar y administrar	Fácil de utilizar y administrar
Difusión alta	Difusión baja	Difusión baja
Manejo de triggers	Manejo de triggers	Manejo de triggers
Manejo de procedimientos almacenados	Manejo de procedimientos almacenados	Manejo de procedimientos almacenados
Tiempo de respuesta alta	Tiempo de respuesta baja	Tiempo de repuesta media
Seguridades media	Seguridades media	Seguridades alta

Tabla 3.1: Cuadro comparativo de Bases de Datos Libres

- b) A continuación se detalla información de los motores de bases de datos libres:

#### MySql

Es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GPL,

pero, empresas que quieran incorporarlo en productos propietarios puede comprar a la empresa una licencia más permisiva que les permita ese uso.

MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias propietarias, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Mysql es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. Mysql compite con sistemas RDBMS propietarios como Oracle, Sql Server y Db2.

Mysql incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído hablar nunca. Mysql utiliza el lenguaje de consulta estructurado (SQL).

Antes Mysql se consideraba como la opción ideal de sitios web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. Mysql es una base de datos robusta que se la puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, incluye un motor de almacenamiento InnoDB compatible con ACID, además dispone de store procedures, triggers, vistas.

Mysql es rápido, y una solución accesible para administrar correctamente los datos de una empresa. MysqlAB es la compañía responsable del desarrollo de Mysql, dispone de un sistema de asistencia eficiente y a un precio razonable, y, como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web.

### *Características de MySQL*

- Escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes. Funciona en diferentes plataformas. Se ha utilizado GNU Autoconfig, de modo que es posible portar MySQL a todos los sistemas modernos que tengan un compilador de C++ y una implementación funcional de subprocesos (threads) POSIX. (El soporte de subprocesos es necesario para el servidor. Para compilar únicamente el código del cliente, no se requiere más que el compilador de C++), se desarrolla y utiliza el software en Linux (SuSE y Red Hat), FreeBSD, y Sun Solaris (Versiones 8 y 9),
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente multiple CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Seguridad cuenta con un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Soporte a grandes bases de datos. Se usa MySQL Server con bases de datos que contienen 50 millones de registros. También existen

usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000 de registros.

- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna **CHAR**, **VARCHAR**, **BLOB**, o **TEXT**.

#### *Ventajas:*

- Es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- MysqlAB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad Mysql.
- Es mucho más rápido que la mayoría de sus rivales.
- Dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- Se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- Resulta fácil de utilizar y de administrar. Las herramientas de Mysql son potentes y flexibles, sin sacrificar su capacidad de uso.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, es decir, qué usuarios tienen acceso a que tablas y con qué permisos.
- MySQL se comporta mejor que otros motores de BD a la hora de modificar ó añadir campos a una tabla.

### *Desventajas:*

- En cuanto a seguridades MySql no realiza el manejo de privilegios de acceso en grupo y roles de usuario.
- No realiza bloqueos a nivel de campo.
- No se puede crear nuevos tipos de datos.

## **PostgresSQL**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado *Ingres* en la Universidad Berkeley de California. *Ingres* fue más tarde desarrollado comercialmente por la *Relational Technologies/Ingres Corporation*.

En 1986 otro equipo dirigido por *Michael Stonebraker* de Berkeley continuó el desarrollo del código de *Ingres* para crear un sistema de bases de datos objeto-relacionales llamado *Postgres*. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, *Postgres* fue renombrado a *PostgreSQL*, tras un breve periplo como *Postgres95*. El proyecto *PostgreSQL* sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

### *Características de PostgreSQL*

- Tiene una alta concurrencia ya que mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo *commit*. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

- DBMS Objeto-Relacional aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.
- Altamente\_Extensible soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.
- Soporte\_SQL\_Completo soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- Integridad Referencial soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- La flexibilidad del API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.
- Lenguajes Procedurales tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.
- MVCC o Control de Concurrencia Multi-Versión (Multi-Version Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.
- Cliente/Servidor usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

- Write Ahead Logging (WAL) característica de PostgreSQL la cual incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

#### *Ventajas:*

- PostgreSQL intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase.
- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.

#### *Desventajas:*

- Consume gran cantidad de recursos.

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- Es de 2 a 3 veces más lento que MySQL.

### **Firebird/Interbase**

“**Interbase** fue creado por un grupo de exempleados de la Digital Equipment Corporation (DEC). Comenzó en 1985 con el nombre de Groton Database Systems, siendo renombrado poco después como Interbase. Ashton Tate adquirió el producto en 1991, y Borland lo adquirió a su vez en 1992 como parte de la compra de Ashton Tate.

En enero del 2000, la empresa Borland, dueña de Interbase Software Corporation, anuncio que la base de datos seria liberada bajo una licencia similar a la MPL de Netscape, bajo el esquema de código abierto. Se liberó la primera versión de código abierto de Interbase versión 6.0 en julio del 2000”<sup>13</sup>.

En la actualidad existen dos desarrollos independientes derivados de Interbase 6.0, debido principalmente a que Borland volvió a su esquema de licencias comerciales, mientras que otros desarrolladores crearon el proyecto Firebird que es un sistema de administración de base de datos (o RDBMS) de código abierto, basado en la versión 6.0 de Interbase. Su código fue reescrito de C a C++.

La tecnología de Firebird lleva 20 años funcionando, esto hace que sea un producto muy maduro y estable.

#### *Características de Firebird*

Existen dos tipos de servidor Firebird para ser instalados: **Classic** y **Super Server**:

---

<sup>13</sup> [www.interbase.com.mx/articulos/porque\\_ib-2.php](http://www.interbase.com.mx/articulos/porque_ib-2.php), Porqué utilizar Firebird-Interbase, Ricaño Alfonso.

**Super Server** maneja hilos de ejecución individuales para cada conexión. Por lo tanto para un número reducido de conexiones el recomendado sería el classic por que consumirá menor cantidad de recursos.

En caso de arquitecturas SMP (*Symmetric Multi-Processing*), se debe utilizar el servidor Classic porque el Super Sever no tiene soporte para este tipo de arquitectura.

Los propios desarrolladores de Firebird recomiendan lo siguiente a la hora de decidirse por uno de estos servidores:

- En plataformas Windows seleccionar el Superserver.
- En Linux simplemente elegir cualquiera, según las conexiones estimadas. En la mayoría de las situaciones no se notará diferencias en la ejecución.

Podría considerarse un tercer tipo, el **Embebed**. Este consiste en una única DLL (*Dynamic Linking Library*) (de unos 2 MB de tamaño) que contiene a todo el servidor. De esta forma se puede tener un DBMS completo disponible y distribuible junto con aplicaciones de usuario.

Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.

Los requisitos de administración son bajos, por esta razón es considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad y restauraciones periódicas.

Cuenta con una de las implementaciones de disparadores (TRIGGERS) más completas comparada con otras bases de datos, y procedimientos almacenados (Stored Procedures) que funcionan de manera similar a los triggers, con la diferencia de que pueden ser ejecutados de manera independiente. Adicionalmente puedes regresar datos y usarlo en sentencias SELECT.

Permite establecer reglas de integridad entre tablas, para que no violen los principios de las relaciones entre tablas.

“Integridad Referencial, permite establecer reglas de integridad entre tablas, para que no violen los principios de las relaciones entre tablas maestro-detalle.

Lenguaje SQL compatible con SQL 92, la implementación de SQL en Firebird es una de las más completas, incluso mejor que algunos de sus competidores de código abierto, como Postgres y MySQL, lo que asegura que se pueden realizar complejas consultas anidadas, y utilizar funciones de conversión como CAST o extracción de las partes de las fechas, con EXTRACT”<sup>13</sup>.

Seguridad: Firebird maneja *alias de Bases de Datos* los cuales esconden al cliente la ubicación física de las bases de datos. Usando alias, un cliente puede por ejemplo conectar a “frodo:zappa” sin necesidad de saber que la ubicación real es `frodo:/var/firebird/music/underground/mothers_of_invention.fdb`. Los alias también le permiten reubicar bases de datos manteniendo la misma cadena de conexión en los clientes.

Se puede manejar permisos independientes de acceso, modificación, inserción y eliminación por tabla y por usuario.

#### *Ventajas:*

- Es medianamente escalable.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Buena seguridad basada en usuarios/roles.
- Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como cd-roms.
- Existencia de controladores ODBC, OLEDB y JDBC.
- Controlador nativo para PHP.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).

- Soporte de transacciones y claves foráneas.
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Capacidad de almacenar elementos BLOB (*binary large objects*).
- Escritura segura – recuperación rápida sin necesidad de logs de transacciones.

#### *Desventajas:*

Una desventaja importante es la difícil instalación del motor de base de datos ya que al ser complejo permite que tan solo aquellos usuarios expertos puedan instalar este software mediante comandos y siguiendo indicaciones específicas. No dispone de un asistente de instalación y este proceso se lo realiza a bajo nivel.

### 3.1.2 HERRAMIENTAS DE FRONT – END

- a) La selección se realizará en base a la información de la *Tabla 3.2* en la cual se presenta una comparación entre las herramientas de Front – End.

PHP	Java	PERL	RUBY	PYTHON
Curva de aprendizaje media-corta	Curva de aprendizaje media-corta	Curva de aprendizaje corta	Curva de aprendizaje corta	Curva de aprendizaje alta en aplicaciones web
Robusto	Muy robusto	Robusto	Robusto	Robusto
Sintaxis sencilla, sucia, no estricta	Sintaxis compleja, limpia, estricta	Sintaxis sencilla	Sintaxis sencilla	Sintaxis sencilla, limpia, estricta
Cantidad de	Cantidad de	Cantidad de	Cantidad de	Cantidad de

APIs disponible alta	APIs disponibles muy alta	APIs disponible alta pero no son actualizadas tan seguido	APIs disponibles alta	APIs disponible alta
Mucho código fuente libre	Mucho código fuente libre	Poco código fuente libre	Poco código fuente libre	Poco código fuente libre
Desarrollo medio-corto	Desarrollo medio-largo	Desarrollo rápido	Desarrollo rápido	Desarrollo rápido
Ejecución muy rápida	Ejecución muy lenta pues el código debe ser interpretado	Ejecución lenta	Ejecución lenta	Ejecución muy lenta
Permite mayor integración con la mayoría de base de datos.	Permite integración con base de datos	Permite integración con ciertas bases de datos.	Permite integración solo con las más populares de las base de datos.	Permite integración con base de datos.
Gran madurez	Alta madurez	Tiene madurez	Tiene madurez	Tiene madurez
Difusión alta	Difusión media-alta	Difusión media	Difusión baja	Difusión baja
Orientación a objetos	Mucha orientación a objetos	Orientación a objetos, Procedural	Orientación a Objetos pura	Orientada a Objetos, Procedural
Aplicación orientada a	Toda clase de aplicación	Toda clase de aplicaciones	Aplicaciones Web simples	Toda clase de aplicaciones

Web simples y complejas				
----------------------------	--	--	--	--

Tabla 3.2: Cuadro Comparativo de Herramientas de Desarrollo

- b)** A continuación se presenta información detallada de las herramientas de Front – End a seleccionar para el desarrollo del prototipo del Módulo de Registro Estudiantil:

### **Php**

Un lenguaje de programación utilizado mayormente para desarrollar servicios web. PHP es un lenguaje de fácil aprendizaje, distribuido en forma gratuita, que permite interactuar con muchos sistemas de gestión de bases de datos

El principal objetivo de PHP5 ha sido mejorar los mecanismos de POO para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes.

#### *Ventajas:*

- Debido a que no requiere demasiados recursos del sistema el PHP es muy rápido.
- Es software libre.
- Código fuente disponible.
- Diseñado para la web.
- PHP es un lenguaje multiplataforma, lo que significa que está preparado para trabajar sobre distintos sistemas operativos.
- Multisistema operativo.
- Soporte para varios servidores web.
- PHP es moduable.
- Al ser muy difundido tiene buena documentación.

- Se caracteriza por ser un lenguaje cuyo aprendizaje es bastante sencillo e parecer y utilizar.
- Seguro (ya que evidentemente tiene errores pero se solucionan mucho antes que otros sistemas propietarios).
- PHP utiliza su propio sistema de administración de recursos y dispone de un satisfactorio método de manejo de variables, conformando un sistema robusto y estable.
- Similar en sintaxis a C y Perl.
- Soporta en cierta medida la Programación Orientada a Objetos.
- PHP provee diferentes niveles de seguridad los cuales pueden ser configurados desde un archivo de configuración .ini.
- Permite integración con la mayoría de base de datos.
- PHP soporta la utilización de otros protocolos como IMAP, SNMP, NNTP, POP3 o HTTP a nivel de socket.
- PHP no soporta directamente punteros, como en el C, de forma que no existen los problemas de depuración provocados por estos.
- Se puede hacer grandes cosas con pocas líneas de código.
- El código PHP es mucho más legible que el PERL.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF, etc).
- “PHP 5 ofrece mejoras significativas con respecto a versiones anteriores, con una orientación a objetos similar a la de Java, y un rendimiento del nivel de las plataformas J2EE y .NET”<sup>14</sup>.
- Posee una amplia documentación en su página oficial.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- procesar una gran cantidad de texto o datos binarios que deben ser analizados y reducidos a reportes.

---

<sup>14</sup> [www.formaselect.com/curso/Programacion-en-PHP-5/Programacion-en-PHP-5.pdf](http://www.formaselect.com/curso/Programacion-en-PHP-5/Programacion-en-PHP-5.pdf), ” Programación en PHP5”, FormaSelect Grupo Empresarial

*Desventajas:*

- Los errores no se muestran en tiempo de compilación sino en tiempo de ejecución.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y php.
- La transición de versiones no son compatibles en un 100%.
- No es un lenguaje scripting 100% orientado a objetos.

**Java**

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

*Ventajas:*

- Es orientado a objetos.
- Es multiplataforma.
- La arquitectura esta basada en capas separadas lo que permite hacer modificaciones sin que afecten a las demás.
- Permite crear arquitecturas completas únicas y exclusivas en productos de software libre.
- El uso de Java y J2EE simplifica al software a través del uso de componentes modulares y estándar.
- El proceso de seguridad en la ejecución de código en java es por medio de un verificador de "bytecodes" y aplica un probador de teoremas para detectar fragmentos de código ilegal.
- El uso de herramientas como Enterprise JavaBeans, Servlets, páginas JSPs y XML permite crear aplicaciones escalables, confiables e integradas con los sistemas existentes en tiempos cada vez más cortos.

- Existe un garbage collector que se encarga de reasignar la memoria de objetos que ya no estén siendo referenciados.
- El manejo de las bases de datos es uniforme, es decir transparente y simple.
- Tiene un recolector de basura de tal manera que al programador no se le permite asignar memoria durante tiempo de ejecución.
- Es bueno para crear aplicaciones con arquitectura de n capas.
- En cuanto a entorno de ejecución, se han perfeccionado las JVM's y existen diversos proveedores, quienes han ido compitiendo para proveer JVM's cada vez más rápidas y eficientes.

#### *Desventajas:*

- Para realizar proyectos en Java y sobretodo enfocado a la arquitectura J2ee se requiere de programadores o desarrolladores con experiencia.
- Es lento ya que su código debe ser interpretado y no ejecutado.
- No es modulable ya que solo depende de un solo lenguaje.
- El soporte de Java para patrones de texto y la manipulación de éste no es tan potente como en lenguajes como Perl, Ruby o PHP.
- El código Java puede ser a veces redundante en comparación con otros lenguajes. Esto es en parte debido a las frecuentes declaraciones de tipos y conversiones de tipo manual (casting).
- En cuanto a la portabilidad en el manejo a bajo nivel deben usarse métodos nativos.
- En cuanto al manejo de diseño de interfaces gráficas con awt y swing no es simple.

#### **Perl**

Perl (*Practical Extraction and Report Language*) fue creado en octubre de 1987 por Larry Wall. Su propio autor señala que combina algunas de las mejores capacidades de C, sed, awk y sh, por lo que programadores familiarizados con

estos lenguajes tendrán gran facilidad para trabajar con él sobretodo en administración de sistemas.

Perl tiene hoy en día una gama muy amplia de usos, en una gran variedad de arquitecturas, bajo varios sistemas operativos. Es con seguridad uno de los lenguajes más utilizados para tareas pequeñas y de tamaño mediano, dada la rapidez de aprendizaje, si bien su capacidad para modularización lo hace también un candidato factible para tareas de mayor envergadura.

*Ventajas:*

- Es software libre (GPL).
- Es portable (multiplataforma).
- Soporta la programación estructurada, programación orientada a objetos, programación funcional.
- Es relativamente rápido para un lenguaje tipo "script".
- El desarrollo de aplicaciones es muy rápido.
- Para ejecutar código Perl no es necesaria gran potencia de CPU ni memoria.
- En general el código Perl no necesita traducción explícita, aunque puede hacerse. Se paga un costo en tiempo de compilación cada vez que se ejecuta el programa, aunque usualmente es bajo.
- La mantención de sistemas en Perl es en general barata. Más aún, un código no ofuscado, bien escrito y bien comentado es en principio fácil de mantener.
- No hay mayor discrepancia entre versiones, y se logra una compatibilidad hacia atrás y hacia adelante en general bastante alta.
- Posee extensiones para ODBC que permiten construir aplicaciones que permitan acceder diversas bases de datos.
- Reduce el ciclo de programación. No tiene que compilar su aplicación, PERL es interpretado y por ello, sus programas pueden ser corridos en muchas plataformas sin necesidad de ser recompilado.

- Existe la posibilidad de incrustar código Perl en otros lenguajes y viceversa, así como, usar módulos externos como si fuesen construidos en Perl, gracias al mecanismo de importación.
- Permite realizar directamente llamadas al sistema operativo.
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares.
- Retroalimentación inmediata durante el proceso de desarrollo.
- La gestión de la memoria es automática.
- Perl es un lenguaje interpretado optimizado para el procesamiento de archivos de texto plano, extrayendo información de ellos e imprimiendo reportes en base a esa información.

#### *Desventajas:*

- La transición de versiones no es compatible en un 100%.
- Perl puede ser más lento que otros lenguajes haciendo lo mismo porque tiene que compilar el fuente cada vez que corre el programa.
- Perl es difícil traducirlo a otros lenguajes con absoluta fidelidad.
- El mantenimiento del intérprete Perl, a lo largo de los años, se ha vuelto cada vez más difícil.
- El tamaño y complejidad del intérprete es una barrera para los desarrolladores que desean trabajar en él.
- Es lento para algunas aplicaciones, como programación a bajo nivel, escribiendo un "driver" para una aplicación o corriendo modelos numéricos de cálculo intensivo.
- Utiliza muchos recursos de la máquina. Esto significa que no es tan ligero como un programa en C, pero en la práctica es ligero comparado con la potencia de computación de los ordenadores actuales.
- En general tiene soporte en los servidores pero en modo "CGI"<sup>15</sup> no como módulo de Apache esto implica que el rendimiento sea menor.

---

<sup>15</sup> [www.maestrosdewlweb.com/editorial/cgiintro/](http://www.maestrosdewlweb.com/editorial/cgiintro/), "Qué es el CGI", Christian Van Der Henst S., 23 de Marzo de 2001, Common Gateway Interface, método para transformar información hacia un compilador instalado en el servidor. Su

- La libertad que se le otorga al programador puede significar que el resultado sea un programa ilegible. Si no se escribe con cuidado puede llegar a ser difícil de leer.
- las versiones de Perl para Windows no son demasiado prácticas ni, sobre todo, rápidas.

## **Ruby**

Ruby es el lenguaje de scripts para una programación orientada a objetos rápida y sencilla. Tiene muchas características para el procesado de ficheros de texto y para realizar el mantenimiento del sistema (al igual que Perl).

### *Ventajas:*

- Software libre.
- Dispone de muchas posibilidades para procesar ficheros de texto y para realizar tareas de administración del sistema (como en Perl).
- Es simple, directo y extensible.
- Lenguaje orientado a objetos.
- Tiene un Garbage Collection.
- Altamente portable.
- Es multiplataforma.
- Tiene librerías para casi todo lo imaginable.
- Tiene integración con bases de datos.
- Es modulable.

### *Desventajas:*

- Poca información.
- Poco alojamiento.

- No son tan compatibles las diferentes versiones.
- Es un buen competidor de Perl pero inferior.

## **Pitón**

Python es un lenguaje de programación interpretado e interactivo, capaz de ejecutarse en una gran cantidad de plataformas.

### *Ventajas:*

- Software Libre.
- Se pueden crear todo tipo de programas.
- Python es un lenguaje interpretado pues no es necesario compilar.
- Fácil y rápido.
- Gran capacidad orientada a objetos.
- Multiplataforma.
- Es bueno para llevar a cabo prototipos de sistema.
- Es bueno para desarrollo Web y de sistemas distribuidos (cliente servidor).
- Es bueno para el desarrollo de tareas científicas, en los que hay que simular y prototipar rápidamente.
- El código es más legible y mantenible.
- Ahorra un tiempo considerable en el desarrollo del programa.

### *Desventajas:*

- Poca información.
- No es bueno para Programación de bajo nivel como programación de drivers y de kernel.
- Poco alojamiento.
- Es un buen competidor de Perl pero inferior.

Se cree necesario mencionar que las propiedades de Ruby y Python son parecidas a las de Perl, por esta razón no se hace mucha referencia a estas.

### **3.1.3 JUSTIFICACIÓN DE LA SELECCIÓN DE HERRAMIENTAS**

Después del análisis realizados tanto a los motores de base de datos libres como a las herramientas de desarrollo, se ha llegado a la conclusión de que una de las mejores opciones es la utilización de la base de datos firebird y de la herramienta de desarrollo java utilizando la plataforma j2ee.

Con respecto a la herramienta de desarrollo se eligió java ya que es un lenguaje muy seguro cuando se crea aplicaciones multiusuario, además ya que la aplicación es orientada a web se realizara bajo plataforma j2ee utilizando javaScript y ciertas librerías (jar) como por ejemplo Ajax, myFaces que nos ayudaran a mejorar la interfaz en nuestras paginas JSF; La decisión fue tomada debido al alcance del SAE (Sistema de Administración Estudiantil) ya que es una aplicación compleja debido al manejo de varios módulos, a pesar de que el alcance de esta tesis no se enfoca al desarrollo de todos los módulos que maneja el SAE se ha considerado que es una buena opción para el desarrollo futuro de la aplicación.

En cuanto a la selección del motor de base de datos se elige Firebird por el manejo de seguridades de datos ya que la información manejada por el sistema SAE es de gran importancia, además de brindar muchas opciones que maneja una base de datos grande como: los triggers, la integridad relacional, los procedimientos almacenados, etc; Pero para este caso se mantendrá el motor de base de datos es decir SQL2005 ya que la migración de la información implica un proceso complejo y puede tomar mucho tiempo y esto no es objetivo de dicha tesis.

## 3.2 CONSTRUCCIÓN.

La construcción del prototipo será realizado basándose en los requerimientos tomados en la fecha Mayo 29 del 2006 considerando que la implementación será realizada únicamente para el perfil estudiante, por lo que en la etapa de construcción del prototipo no se lo realizarán todos los casos de uso establecidos como requerimientos *Ver sección 2.3.1.3*. Con respecto a la base de datos utilizada para las pruebas será la provista por el grupo SAE con información del cierre de semestre 1 del 2007.

El prototipo del Módulo de Registro Estudiantil del SAE se desarrollará con el lenguaje anteriormente seleccionado, sin embargo, el motor de base de datos utilizado será SQLServer 2005, motor en el cual se encuentra actualmente el Sistema Académico Estudiantil (SAE), la decisión fue tomada ya que para la migración de la base de datos al motor seleccionado debería realizarse un análisis mas profundo y comparativo, por ejemplo tipos de datos, manejo de sintaxis para procedimientos almacenados, triggers, etc, que se maneja SQLServer 2005 y Firebird.

### 3.2.1 CONSIDERACIONES PARA LA CONSTRUCCIÓN

La plataforma en la que se desarrollará nuestro prototipo es J2EE basada en lenguaje java (lenguaje de desarrollo seleccionado 3.1.4), la estructura de capas a utilizar es la siguiente:

1. **Cliente:** En esta capa podemos encontrarnos con distintos tipos de cliente: Aplicaciones de escritorio, Navegadores web, Aplicaciones para dispositivos móviles.

Para nuestro caso deben ser a través de navegadores web.

- 2. Presentación:** Esta capa deberá proveer de los componentes adecuados para permitir la interacción del cliente hacia la capa de negocio que ofrece una aplicación.

El estándar establecido para esta capa es el patrón de diseño arquitectónico MVC (Model, View, Controller), existen distintos frameworks que implementan este patrón definiendo las distintas partes:

- **Modelo:** Componente en el cual se almacenará la información sobre los datos obtenidos del cliente.
- **Vista:** Representación visual de datos del modelo y de las operaciones realizadas en la capa de negocio.
- **Controlador:** Encargado de gestionar la interacción entre el modelo y las vistas.

Existen diferentes frameworks y Api's que implementan el patrón MVC. Las posibilidades que tenemos son varias: Strut, Servlets, Jsp, JSF.

En esta capa se escogió implementar JSF.

- 3. Lógica de Negocio:** Esta capa es el núcleo de nuestra aplicación, en el cual se representan nuestras entidades, relaciones y reglas que implementarán nuestros procesos de negocio. Las posibilidades que tenemos para implementar la lógica de negocio son variadas:

**3.1 POJO's:** "Plain Old Java Objects", se trata de objetos normales Java que contienen procesos de negocio o bien representan entidades, esto implica que podemos incluirlos en la misma capa. Las principales ventajas que conllevan el uso de esta alternativa, es que estos objetos son muy ligeros, no añaden sobrecarga (seguridad, transacciones, sesiones, etc.), y su implementación es sencilla y rápida (productividad alta de desarrolladores), por otro lado no ofrecen tanta funcionalidad

como los Ejb's (escalabilidad, seguridad, etc.), no obstante como siempre deberemos situarnos en el contexto de nuestra aplicación.

**3.2 EJB'S:** Es una componente de software, escrita en Java, que se ejecuta en el entorno de un servidor de aplicaciones J2EE. Las principales ventajas que conllevan el uso de esa alternativa son:

- Es un objeto distribuido.
- Puede ser invocado desde un sistema remoto.
- Encapsula la lógica de negocios de una aplicación.
- Es *altamente portable y reusable*.
- Manejo de transacciones asociadas a las llamadas a los métodos del bean.
- Llamada simultánea a un mismo bean desde múltiples clientes "Concurrencia".
- Sincronización entre los datos del bean y tablas de una base de datos llamada "Persistencia"
- Gestión de mensajes: manejo de Java Message Service.

Los tipos de Ejb que podemos utilizar son los siguientes:<sup>16</sup>

**3.2.1 Los beans de sesión:** Son de dos tipos:

Los ***beans de sesión sin estado*** se usan en general para encapsular procesos de negocio, más que datos de negocio, también puede usarse un bean de sesión sin estado como un puente de acceso a una base de datos o a un bean de entidad.

Los ***bean de sesión con estado*** las *variables de instancia* del bean almacenan datos específicos obtenidos durante la conexión con el cliente. Cada bean de sesión con estado, por tanto, almacena el estado conversacional de un cliente que interactúa con el bean.

---

<sup>16</sup> <http://www.jtech.ua.es/j2ee/ejemplos/ejb/sesion01-apuntes.htm>, Introducción a la tecnología EJB

**3.2.2** Los *beans de entidad* Bean Managed Persistence o Container Manager Persistence, dependiendo de si la persistencia de entidades es manejada por el bean o por el contenedor, es decir, se encargan la persistencia de las entidades de negocio, se utilizan en la capa de integración.

**3.2.3** Los *beans dirigidos por mensajes* se utilizan para servicios de mensajería, probablemente más conocidas como colas de mensajes.

Los principales problemas que presentan el uso de Ejb's vienen en general descritos por estos aspectos:

- Solución compleja.
- Es necesario definir un nombre y las operaciones que realizan para su despliegue en un contenedor.
- Se deben generar fachadas de acceso remoto y locales.
- La activación de los Ejb's varía de un contenedor a otro.

Los Session Bean/POJO's es nuestra opción para esta capa.

**4. Integración:** La principal preocupación viene determinada por la tecnología a utilizar para manejar la persistencia, en J2EE tenemos varias posibilidades:

**4.1 Jdbc:** Es una opción bastante sencilla y adoptada para manejar la persistencia, se la utiliza cuando planteamos en nuestra aplicación el uso Pojo's o Beans de entidad con persistencia manejada por el bean. Es una opción muy ligera, además es bastante conocida por muchos desarrolladores, la dificultad se presenta al tener que acceder a los datos, lo que se debe hacer de forma manual, pero este problema se puede solventar con un buen diseño basado en el patrón DAO (DATA

ACCESS OBJECT'S) lo que permitirá un acceso a datos transparente independientemente de la fuente del recurso.

**4.2 Motores de persistencia:** Frameworks de mapeo de objetos, son útiles y bastante utilizados. La ventaja reside en la posibilidad de definir el esquema de persistencia mediante XML y por lo tanto puede ser procesado de manera automática, aunque la gestión de estos ficheros no es sencilla y puede resultar problemática, si se realiza manualmente, en el momento en el que la aplicación crezca. Uno de los motores utilizados, es Hibernate, aunque existen otros como Castor, TopLink, etc.

**4.3 Entity Beans:** BMP, la persistencia la maneja el Bean de entidad y es el desarrollador quien implementa las operaciones CRUD (create, retrieve, update, delete) de las entidades. CMP, las operaciones CRUD son manejadas por el contenedor,

Para la integración se utilizarán las JDBC.

**5. Sistemas de información:** Son las fuentes de información: bases de datos, ficheros, etc. El principal problema reside en la conexión a estas fuentes de una forma homogénea de ahí la necesidad de una capa de integración.

Dentro de esta capa se utilizará SQLServer 2005 motor de base de datos en el cual reside actualmente la información de SAE.

Debemos considerar que tanto los servidores web como los servidores de aplicación son fundamentales para el desarrollo de una aplicación J2EE por lo que mencionaremos algunos de ellos.

Entre los principales servidores web podemos mencionar: Apache, Tomcat. Y respecto a servidores de aplicaciones tenemos al JBOSS.

Se debe tomar en cuenta que de acuerdo a las necesidades y complejidad de la aplicación se puede variar la estructura de programación.

### **Consideraciones adicionales para la implantación del prototipo:**

#### **Seguridad:**

Las aplicaciones implementadas con arquitectura J2EE y especialmente la subarquitectura especificada para este prototipo. *Ver Tabla 3.3;* presentan grandes ventajas en cuanto a la seguridad de la misma, entre estas podemos mencionar las siguientes:

- Modularidad, se identifican bien cada capa.
- La capa de presentación se utilizará MVC por medio de JSF lo que permite separar la lógica de la capa antes mencionada.
- Utilizar Stateless Session Bean`s localmente nos brinda una gran seguridad ya que de esta forma no permitirá el acceso a ninguna aplicación externa al sistema SAE
- Otro tipo de seguridad es utilizar la configuración de firma digital en el servidor de aplicaciones en nuestro caso el JBOSS.
- Tener una buena arquitectura en red es otra opción para mantener la seguridad en las aplicaciones J2EE.

#### ***Migración de BDD:***

Se realizara la migración de la base de datos de SQLServer 2005 a Firebird solamente la estructura y datos como prueba del motor seleccionado, para la migración de procedimientos almacenados se ha determinado que se debe realizar un análisis más a fondo ya que los motores de bases de datos no manejan un mismo lenguaje de implementación.

### Resumen de estándar utilizado en la plataforma J2ee del prototipo.

Capa	Selección
Presentación	Considerar que los clientes son Navegadores Web. La aplicación web se desarrollará bajo tecnología JSF, empleando patrones MVC.
Negocio	SessionBeans, POJO's
Integración	Se utilizará JDBC.
Sistema de información	Se utilizará la bdd en donde actualmente reside la información del SAE.

Tabla 3.3: Resumen de estándar utilizado en la plataforma J2ee

El código fuente y documentación de programación (catalogo de unidades programadas) mapa de llamadas y conexiones se incluirá en el *Anexo 5*.

### 3.2.2 PRUEBAS

El objetivo de esta etapa es verificar que el prototipo del módulo de Registro Estudiantil cumpla con todos los requerimientos funcionales establecidos y especificados en los casos de uso, *Ver Sección 2.3.1.3* de los cuales se implementarán los especificados en la etapa de construcción.

## Formulario PUNI\_01

**Proyecto:** SAE**Responsable:** Desarrolladores de la aplicación.**Tipo de Revisión:** Prueba Unitaria.**Carrera:** Ingeniería en Sistemas y de Computación.**Caso de Prueba:** Ingreso al sistema.

Fecha	Objetivo de la Prueba	Datos de entrada	Resultados Esperados	Resultado Obtenidos	Observación
10-04-2007	Validar el logeo del sistema con datos correctos	Perfil: ESTUDIANTE Cédula: 0802320911 Código único: paola12	Usuario Permitido	Usuario Permitido	
10-04-2007	Validar el logeo del sistema con datos incorrectos	Perfil: SUBDECANO Cédula: 0802320911 Código único: paola1218	Usuario No Permitido	Usuario No Permitido	Se debe validar que el campo perfil sea el que tiene asignado el usuario.
10-04-2007	Validar el logeo del sistema con datos incorrectos	Perfil: ESTUDIANTE Cédula: 0802320911 Código único:	Usuario No Permitido	Usuario No Permitido	Se debe validar que los campos no estén vacíos.

## Formulario PUNI\_01

**Proyecto:** SAE .**Responsable:** Desarrolladores de la aplicación.**Tipo de Revisión:** Prueba Unitaria.**Carrera:** Ingeniería en Sistemas y de Computación.**Caso de Prueba:** Generar Inscripción

Fecha	Objetivo de la Prueba	Datos de entrada	Resultados Esperados	Resultado Obtenidos	Observación
10-04-2007	Validar impedimentos del estudiante con datos correctos	opcion= " nomFacu= codCarr: '07' codEst: '9910878' op='1'	Cadena vacía	Cadena vacía	
10-04-2007	Validar impedimentos del estudiante con datos incorrectos	opcion= " nomFacu= codCarr: '07' codEst: '9910878' op="	Cadena vacía	Cadena con mensaje: Tiene impedimentos/ No inscrito	Se debe validar que los campos sean enviados correctamente.
10-04-2007	Generar lista de materias con datos correctos	opcion= 'L' nomFacu= codCarr='07' codEst = '0210249' codMatBean= nomMatBean= " op='1'	Lista de posibles materias.	Lista de posibles materias.	

10-04-2007	Generar lista de materias con capos de filtro	opcion= 'L' nomFacu="" codCarr='07' codEst = '0210249' codMatBean="" nomMatBean= 'A%' op='I'	Lista de posibles materias que empiecen con la letra 'A'	Lista llena.	Verificar que el filtro funcione bien.
10-04-2007	Chequear cruce de materias.	opcionCruce= 'I' codCarr: '07' codEst: '0210249' codMat: 'ISO378' codParalelo: 'GR.1'	Cadena vacía	Cadena vacía	

Documentación de las restantes pruebas. *Ver Anexo 6.*

### **Análisis de los resultados de las pruebas de unidad.**

Luego de aplicar las pruebas de unidad a la aplicación se realizaron las correcciones detectadas en los procesos con el fin de que el sistema se encuentre acorde a los requerimientos solicitados por los usuarios.

## **CAPITULO 4: EVALUACIÓN**

### **4.1 AMBIENTE**

#### **Recursos físicos:**

- PC (Mobile AMD Sempron(tm) 3100+, velocidad 789 MHZ, 512 MB de RAM).
- PC (Intel(R) Pentium(R) M, Procesador 1.73 GHz, 512 MB de RAM).

Para la realización de las pruebas no se capacitó recurso humano ya que los usuarios tienen un alto conocimiento de los procesos realizados en el modulo de registro estudiantil.

Para la evaluación de la aplicación se utilizará un respaldo de la BDD del SAE provista por el grupo SAE con información del cierre de semestre 1 del 2007, la misma que se encuentra sobre el motor SQL Server 2005.

Las pruebas del prototipo fueron realizadas en un periodo posterior al de matriculas extraordinarias con una muestra al 3% de estudiantes tomados de la carrera de ingeniería en sistemas y de computación.

Los datos de planificación, fueron modificados para proveer números limitados de cupos, habitualmente se manejan valores de 30 por paralelo, en este caso, se utilizó 5 por el número reducido de casos por probar.

### **4.2 INTEGRIDAD**

Se evaluarán los resultados que obtengan los estudiantes durante el manejo o utilización del prototipo del modulo de registro estudiantil, los procesos a evaluar serán los casos de uso que estén directamente relacionados con el usuario estudiante.

La evaluación de dichos resultados nos ayudará a verificar la legitimidad y certeza de los procesos implementados en el prototipo.

### 4.3 ANÁLISIS DE RESULTADOS

Luego de la ejecución de las pruebas realizadas al prototipo del módulo de registro estudiantil se realizó el análisis de resultados obtenidos. *Ver Anexo 5.*

Según la información que se pueden observar en el anexo antes mencionado, se presenta el siguiente resumen de resultados:

#### Análisis de resultados de las pruebas de sistema:

<b>Análisis de resultados</b>			
<b>Resumen</b>			
<b>Cantidad de pruebas realizadas</b>		30 Formularios de Inscripción.	<b>% pruebas satisfactorias</b>
Número de Pruebas	<b>Aceptadas</b>	10	33,3%
	<b>Rechazadas</b>	15	50,0%
	<b>Modificadas</b>	5	16,7%
Comentarios: Los errores provocados durante las pruebas son por el mal manejo o duplicación de información en la base de datos más no por errores de programación en la implementación de la aplicación.			

Después del resumen presentado se realizó las verificaciones y correcciones necesarias para que la aplicación cumpla con los requerimientos establecidos, y para la realización de las pruebas con usuarios.

#### Análisis de resultados de las pruebas con usuarios:

<b>Análisis de resultados</b>			
<b>Resumen</b>			
<b>Cantidad de pruebas realizadas</b>		80 Usuarios.	<b>% pruebas satisfactorias</b>
Número de Pruebas	<b>Aceptadas</b>	35	43,75%
	<b>Rechazadas</b>	25	31,25%
	<b>Modificadas</b>	20	25,00%
Comentarios: Del resumen obtenido se puede concluir que el sistema tiene un buen porcentaje de pruebas satisfactorias sin embargo se debería realizar un buen análisis de la información que se maneja en la base de datos para poder			

realizar las correcciones respectivas con el fin de que la aplicación funcione y presente la información de acuerdo a los requerimientos establecidos por los usuarios.

### **Resumen de Resultados:**

Después del análisis de resultados de las pruebas realizadas al sistema se puede decir que un 68,75% de la muestra tomada obtuvieron resultados satisfactorios, acorde a los requerimientos establecidos.

El 31,25% de la muestra tomada obtuvo resultados no satisfactorios ya que la aplicación presento resultados erróneos como:

- Casos de cruces de horarios, producidos por intersección en paralelos de materias de sociales.
- Falta de cupos para estudiantes que tenían derecho, pero ya fueron ocupados por otros estudiantes de menor número de créditos.
- Diferencias en valores de matrícula por pagar; ya que no se registro el descuento por hermanos.

## CAPITULO 5: CONCLUSIONES Y RECOMENDACIONES

### 5.1 CONCLUSIONES

- Para que un proyecto de reingeniería de sistemas tenga éxito se requiere establecer una metodología ordenada, en la cual tanto el plan de reingeniería como la documentación que se genera a medida que se avanza en el trabajo deben tener un papel importante, puesto que ayudará a mantener orden y control sobre el proceso.
- Los costos de emprender un proceso de reingeniería depende del alcance de la misma, ya que en ciertas ocasiones el realizar un nuevo proyecto podría tener un costo inferior al de aplicar la reingeniería.
- Debe plantearse la reingeniería como una solución siempre y cuando el sistema o proceso sea de alto valor para el negocio y la versión en operación de conflictos y no cumpla los requisitos. Podría darse el caso que se aplique reingeniería por módulos o procesos determinados, de tal forma que se ahorre dinero y riesgos al cliente.
- Al aplicar reingeniería a un sistema heredado, no se busca el obtener una nueva aplicación lo que se pretende y se logra es un producto mejorado, óptimo, capaz de evolucionar a bajo costo y con un esfuerzo mínimo de recursos, en nuestro caso la reingeniería se enfocó a reestructurar la documentación del modulo de registro estudiantil ya que en relación a la aplicación se implemento una nueva versión en la cual el código no pudo ser reutilizado.
- La adquisición del Software libre no siempre es de costo cero, sin embargo se debe tener en cuenta que es mucho mas económico que el software propietario equivalente.
- Según estadísticas el Software Libre junto a la plataforma J2EE ha venido dando aplicaciones con buenas características de: rentabilidad, seguridad, escalabilidad, etc. lo que incrementa la confiabilidad en un sistema implementado con este tipo de herramientas.
- En inicio se selecciono un motor de base de datos libre FireBird considerado como una de las mejores opciones, la migración de

procedimientos no se la realizo ya que la estructura de dichos procedimientos es propia de cada motor de BDD razón por la cual no se continuo con Firebird.

- El prototipo se conecta con una imagen de la base de datos SAE en SQL Server 2005 motor de base de datos donde reside actualmente la información manejada por el sistema.
- Se debe tomar en cuenta que el alcance del proyecto de titulación es realizar un prototipo del módulo de registro estudiantil en herramientas de libre difusión, más no la migración de la base de datos.
- Este trabajo podemos considerarlo como una ayuda referencial para desarrolladores que quieran explorar el ambiente JEE.
- Existen múltiples herramientas y tecnologías para construir aplicaciones J2EE, tenemos que saber elegir aquellas que necesitemos, potenciando la productividad de los desarrolladores.
- Pese que si existe documentación accesible no hay soporte de recurso humano con pericia en el desarrollo de aplicaciones con J2ee para consultar.
- La especificación J2EE posee una complejidad que dificulta su adopción por los desarrolladores menos experimentados.

## **5.2 RECOMENDACIONES:**

- Al comenzar un proyecto de reingeniería, se recomienda total interés en la etapa de estudio del sistema heredado, ya que el correcto entendimiento del sistema disminuirá la ocurrencia de errores futuros que puedan ocasionar inconvenientes en el proceso.
- Se recomienda analizar todas las posibles fuentes de información, ya que cualquier criterio permitirá fortalecer el entendimiento del sistema.
- Durante todo el proceso de reingeniería se recomienda mantener una estrecha comunicación con los usuarios pero sin dejarse influenciar al 100% por la versión anterior con la finalidad de buscar en la reingeniería mejoras de procesos y de datos.

- En el caso de escoger J2EE, las empresas, instituciones deben informarse sobre las facilidades que prestan las herramientas Libres y las que necesitan licencia, ya que las últimas pueden ahorrar considerablemente el tiempo de implementación de las aplicaciones por facilidad de manejo y por el soporte técnico.
- Se recomienda la implementación de aplicaciones J2EE ya que se obtiene mejoras de seguridad tanto a nivel de autenticación, como de autorización y transportes.
- Se recomienda conocer técnicas avanzadas de análisis, diseño y desarrollo para construir aplicaciones J2EE, de un modo profesional.
- En caso de migrar la base de datos a un motor libre se debería realizar un análisis del tipo de dato soportado por el motor actual (SQL Server 2005) y el seleccionado, ya que cada motor de bases de datos posee una estructura específica para la creación de procedimientos almacenados.
- Teniendo en cuenta que en ciertas empresas la reingeniería es una actividad común se recomienda que en la carrera se profundice este tema en seminarios o cursos formales para que nosotros los estudiantes estemos mejor preparados para este tipo de retos.
- Se recomienda dar un mayor interés a todo lo que se refiere software libre ya que la tendencia es llegar a cambiar todo lo que se tiene actualmente en herramientas comerciales tanto en Empresas privadas e instituciones del Estado, en este momento inclusive es una propuesta actual.
- Se recomienda que se mantenga una adecuada documentación de todos los cambios que se realicen en el SAE para un buen manteniendo del mismo.

## **BIBLIOGRAFÍA:**

- <http://www.novell.com/officesemeaspainnewspr112703.html>; Acerca de Mono, breve resumen de la creación de la herramienta de desarrollo; 2005.
- <http://www.monohispano.org/tutorialesmono-puf&c944.html>; Introducción a Mono; 2005.
- <http://www.mono-project.com/MySQL>; MySQL y Mono; Marzo 2005.
- <http://pontonetpt.com/Artigos/202.aspx>; Como Instalar Mono; 2003-2004.
- <http://www.mono-project.com/Downloads>; Descargar archivos de instalación de la Herramienta de desarrollo; Mayo 2005.
- <http://www.ciberteca.net/webmaster/php/>; Introduction a PHP; Octubre 2002.
- <http://www.ciberteca.net/webmaster/php/index5.asp>; Ejemplos de PHP; Diciembre 2003.
- <http://www.programatium.com/php2.htm>; Web's dinámicos con PHP; 2001.
- <http://www.rational.com/products/rup/prodinfo/whitepapers/dymanic.jtmpl>; Breve introducción del RUP y sus mejores prácticas para los equipos de desarrollo de Software; Nov 1999.
- <http://www.monografias.com/trabajos5/inso/inso2.shtml>; Información sobre definiciones de Reingeniería e Ingeniería Inversa y además sobre las situaciones que se deben tener presentes para aplicar la Reingeniería; 1997.
- <http://www.wilkinsonpc.com.co/free/articulos.html>; Artículo que hace referencia a lo que es software libre, formas y tipos de licenciamiento de software; 1999.
- <http://www.gnu.org/philosophy/free-sw.es.html>; Información sobre los que es el Software Libre; 2000.
- <http://www.monografias.com/trabajos12/elsoflib.shtml>; Definición del Software Libre, Tipos de Licencias y Beneficios; 2000.

- <http://calisto.sip.ucm.es/people/pablo/teaching/tp0304/09%20Proceso%20Unificado%20Rational.pdf>; El Proceso Unificado de Desarrollo. Pablo Gervás; Diciembre 2003.
- [http://weblogs.javahispano.org/page/jlmonteagudo?entry=integracion\\_de\\_ejb3\\_y\\_jsf](http://weblogs.javahispano.org/page/jlmonteagudo?entry=integracion_de_ejb3_y_jsf); Integración de EJB3 y JFSF. Julio Monteagudo; Julio 2006.
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=entornosJSF>; Creando JSF. Beatriz Bonilla.
- <http://www.programacion.com/tutorial/j2ee/>; Introducción a J2EE. Juan Antonio Palos.

#### Libros:

- El Proceso Unificado de Desarrollo de Software. Ivar Jacobson, Grady Booch, James Rumbaugh, Madrid-España, Editorial Pearson Education S.A., 2000, pág. 563.
- UML y Patrones: Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado, Madrid-España, 4ª Edición, Editorial Pearson Education S.A. , 2003, pág. 624.
- Ingeniería de Software Orientada a Objetos, Bruegge, Bernd y Dutoit, Hallen H., México, Editorial Pearson Education S.A. , 2002, pág. 576.
- El Lenguaje Unificado de Modelado. Ivar Jacobson, Grady Booch, James Rumbaugh, Editorial Addison Wesley, 1999.