

**UNIVERSIDAD NACIONAL DE PIURA
FACULTAD DE INGENIERÍA INDUSTRIAL
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA**



TRABAJO DE INVESTIGACION:

**“DESARROLLO DEL SISTEMA DE INFORMACIÓN ACADEMICO DEL
INSTITUTO SUPERIOR TECNOLÓGICO MANUEL YARLEQUE ESPINOZA
DE CATACAOS, UTILIZANDO LA METODOLOGIA UML”.**

**AUTORAS:
CASTRO GUTIERREZ VANINA SUHAIL
MAZA ANTON GINA LIZBETH**

**PIURA - PERU
2010.**

OBJETIVOS DE LA INVESTIGACIÓN

OBJETIVO GENERAL

Desarrollar un Sistema de Información Académico en el Instituto Superior Tecnológico Manuel Yarleque Espinoza de Catacaos.

OBJETIVOS ESPECIFICOS

- Analizar los requerimientos de información académica de la institución educativa Manuel Yarleque Espinoza de Catacaos, utilizando una metodología para el desarrollo del sistema orientado a objetos: UML
- Realizar los diagramas que la metodología exige, que nos permitirá modelar el sistema de información y brinde los alcances en su implementación.
- Estudiar el desarrollo de un Sistema de Información Académico empleando el desarrollo iterativo, característica primordial de la metodología UML, con la finalidad de elaboración el diagrama de clases, siendo éste permitirá la solución lógica del software.
- Como toda aplicación necesita un almacenamiento, se ha desarrollado una base de datos que podrá almacenar y procesar la información captada en los procesos académicos.

DEDICATORIA

A Dios, Por ser nuestro gran maestro, que nos enseña cada día, ha abstraer de las experiencias, conocimientos de la vida.

A nuestros padres, Por ser tan comprensivos y ser la fuerza motriz de nuestras metas, a Uds. padres porque siempre están allí en los momentos de desmayo.

A nuestros hermanos, Para que en el afán de superación, sea el precursor del éxito.

INTRODUCCIÓN

En los últimos años los sistemas computacionales y de información han tenido un gran desarrollo, tanto así, que actualmente se han vuelto indispensables en las actividades de las instituciones; y orientado a instituciones educativas cumplen un papel importante al permitir brindar un mejor servicio a su comunidad educativa, y permitir a la Gestión brindar información oportuna en la toma de decisiones.

Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico, que indujo a una falta de estandarización. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente de diseño. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de despliegue.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones en Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender. Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción. Construir: a partir de los modelos especificados se pueden construir los sistemas diseñados. Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

El presente trabajo de investigación se logro modelar un sistema de información académico, para el instituto Manuel Yarleque Espinoza, cuyo desarrollo se realizo con la ayuda del lenguaje Unificado de modelado UML, el mismo que nos permitirá desarrollar, documentar e intercambiar modelos de información significativa del sistema de información académico, para el instituto Manuel Yarleque Espinoza.

RESUMEN

El trabajo de tesis: “**DESARROLLO DEL SISTEMA DE INFORMACIÓN ACADEMICO DEL INSTITUTO SUPERIOR TECNOLÓGICO MANUEL YARLEQUE ESPINOZA DE CATACAOS**”, materia del presente estudio se realiza haciendo un análisis de la situación académica actual del instituto, de tal forma que ha permitido conocer que el instituto no cuenta con sistema de información académico y mucho menos con una base de datos que pueda almacenar la información captada en el proceso académico. En miras de brindar un mejor servicio a la comunidad del instituto, se ha propuesto el desarrollo de un sistema de información académico, empleando la tecnología de objetos mediante la metodología UML.

El estudio inicio, obteniendo información general sobre la estructura orgánica y funcional del instituto, con la finalidad de estudiar sus procesos académicos, y recoger los requerimientos de información; así mismo se da un panorama de las metodologías orientados a objetos para el desarrollo del sistema.

En base a las necesidades de información, se propone un modelo del sistema, constituido por la definición de requisitos, Diagramas y Casos de Uso; que son la base para implementar la funcionalidad de la base de datos y sistema de información.

En lo referente a la implementación del sistema, se propone la arquitectura cliente/servidor; desarrollado en el sistema de administración de base de datos Microsoft SQL Server.

Finalmente el estudio realiza un análisis económico del sistema de información, evaluando sus costos y beneficios del, en términos de costos, tiempo

INDICE

CAPITULO I: MARCO TEORICO

1.1.	Datos generales de la institución	1
1.1.1.	Reseña histórica	1
1.1.2.	Estructura orgánica	2
1.1.3.	Estructura funcional	4
1.2.	Metodología para el desarrollo del sistema	7
1.2.1.	Análisis y diseño orientado a objetos	7
	Metodología OMT	7
	Metodología de BOOCH	9
	Metodología UML	10
1.2.2.	Comparación entre el análisis y diseño orientados a objetos y el análisis y diseño estructurado	14
		15
1.2.3.	Fundamento de la selección de la metodología	

CAPITULO II: ANÁLISIS DE LA SITUACIÓN ACTUAL

2.1.	Formulación del problema	16
2.2.	Descripción del sistema actual	17
	Programación académica	17
	Proceso de matricula	17
	Evaluación académica	18
2.3.	Alcance del sistema propuesto	18

CAPITULO III: MODELADO DEL SISTEMA

3.1.	Análisis	19
3.1.1.	Definir requisitos.	19
	Metas o propósitos	19
	Funciones del sistema	19
3.1.2.	Casos esenciales de uso	23
3.1.3.	Crear diagramas de casos de uso	33
3.1.4.	Crear modelo conceptual	34
3.1.5.	Diagramas de secuencia	35
3.1.6.	Definir contratos	37
	Contratos del sistema	37
	Diagramas de estados	48

3.2.	Diseño	52
3.2.1.	Casos de Uso reales	52
3.2.2.	Diagramas de Colaboración	63
3.2.3.	Diagrama de clases	66

CAPITULO IV: DESARROLLO DEL SISTEMA

4.1.	Arquitectura de tecnologías de información	67
4.1.1.	Justificación de la utilización de la arquitectura cliente-servidor	67
4.1.2.	Arquitectura cliente-servidor	68
	Arquitectura de dos capas	68
	Arquitectura de tres capas	70
4.2.	Implementación del diseño de base de datos	72
4.2.1.	Conversión del diagrama de clases a una base de datos relacional	72
4.2.2.	Creación de la base de datos	77
4.3.	Implementación de la aplicación	79
4.3.1.	Creación de clases	79
4.3.2.	Ventanas	81
4.3.3.	Reportes	83
4.4.	Seguridad del sistema	83
	Seguridad de SQL Server	84
	Usuarios y Funciones	85

CAPITULO V: EVALUACIÓN DEL SISTEMA

5.1.	Costos y beneficios del sistema propuesto	88
5.1.1.	Costos del desarrollo del sistema.	88
5.1.2.	Costos de mantenimiento del sistema	89
5.1.3.	Beneficios	89
	De tiempo	91
	Económicos	93
	Intangibles	94
5.2.	Análisis económico	95

ANEXOS

ANEXO I: DIAGRAMA DE SECUENCIA DEL SISTEMA

CAPITULO I: MARCO TEORICO

1.1. DATOS GENERALES DE LA INSTITUCION

1.1.1 Reseña Histórica

El Instituto Superior Tecnológico “Manuel Yarleque Espinoza” está ubicado en el Distrito de Catacaos, Provincia y Departamento de Piura, en el pasaje N° 15 de la I Etapa del A.A.H.H. Nuevo Catacaos.

Según consta en el libro de Actas de Sesiones Ordinarias (Sesión Ordinaria N° 002/91), los profesores Matías Cruz Sandoval y Máximo Ordóñez Córdova, expusieron ante los miembros del Concejo Municipal de Catacaos el Proyecto de creación de un Instituto Tecnológico a Nivel Estatal para Catacaos; siendo así que el Señor Alcalde mostró interés en un proyecto que beneficiaría a la juventud estudiosa de escasos recursos, por lo que en unanimidad con los Miembros del Concejo Municipal de Catacaos acordaron aprobar la propuesta presentada y acordaron sacar adelante el funcionamiento del Instituto Superior Tecnológico Estatal en esta ciudad.

Según el expediente N° 05430 del 19 de Febrero de 1991, presentado por el señor Alcalde Distrital de Catacaos, en calidad de representante de la entidad promotora, para la creación del Instituto Superior Tecnológico “Manuel Yarleque Espinoza” de Catacaos, con el voto aprobatorio del Consejo Regional de la Región Grau, en Sesión Ordinaria N° 15 del 22 de abril de 1991; y mediante RESOLUCIÓN EJECUTIVA REGIONAL N° 210, se resuelve:

- Aprobar a partir de la fecha la creación del Instituto Superior Tecnológico “Manuel Yarleque Espinoza”, del Distrito de Catacaos, Provincia de Piura, Sub-Región Piura, Región Grau.
- Autorizar al citado Instituto ofertar las carreras profesionales de Agropecuaria, Mecánica Automotriz y Administración.
- La Secretaria Regional de Asuntos Sociales de la Región Grau y la Dirección Departamental de Educación de Piura, expedirán las disposiciones complementarias correspondientes en las instancias de su competencia, para el cumplimiento del Decreto Regional, dado a nivel de la Región Grau.

Con el expediente N° 35311-99, presentado por el Director del Instituto Superior Tecnológico “Manuel Yarleque Espinoza”, se solicita autorización para desarrollar los estudios conducentes al otorgamiento de nuevos títulos de Profesional Técnico. El Ministerio de Educación mediante Resolución Directoral N° 211-2000-ED, el 6 de Marzo del 2000, resuelve autorizar, al Instituto Superior Tecnológico Público “Manuel Yarleque Espinoza”, a desarrollar los estudios conducentes al otorgamiento de los Títulos de:

- Profesional Técnico en Computación e Informática.
- Profesional Técnico en Enfermería Técnica.

A partir del I Semestre Académico del Año 2000, en un solo turno, con una meta de 40 alumnos por cada título y con una duración de seis semestres académicos.

1.1.2. Estructura Orgánica¹

El análisis de la estructura orgánica permite definir las relaciones entre las dependencias relacionadas con el sistema. Las labores relacionadas con el Sistema de Información Académico se rigen por el Reglamento Interno del Instituto Superior Tecnológico “Manuel Yarleque Espinoza” y el Reglamento General de Institutos y Escuelas Superiores Públicos y Privados y son efectuadas al momento de la ejecución del presente trabajo. La Estructura Orgánica del Instituto Superior Tecnológico “Manuel Yarleque Espinoza” de Catacaos, es la siguiente:

- **Dirección.-** Órgano responsable de la conducción académica y administrativa del Instituto en concordancia con las normas y lineamientos de políticas educativas. Este órgano lo integran el Director y la secretaria de dirección.
- **Secretaría General.-** Órgano responsable de planear, organizar, dirigir y controlar las acciones de registro y certificación académica; así como, las administrativas. El Secretario Docente es el encargado de la organización y funcionamiento de este órgano.

¹“Manual de Organización del Instituto Superior Tecnológico Manuel Yarleque Espinoza”

CAPITULO I: MARCO TEORICO

- **Área de Administración.**- El área de Administración esta integrada por el Jefe de Área, Oficinista y personal de servicio, limpieza y guardianía.
- **Departamento de Formación General y Profesional.**- Responsable de implementar, supervisar y evaluar el currículum en las líneas de humanidades, arte, ciencias básicas y actividades. Así mismo como las estructuras curriculares específicas de cada una de las carreras que oferta la institución. Esta integrado por el Jefe de Departamento y Docentes estables y/o contratados.



Figura I.1. Estructura Orgánica del I.S.T “Manuel Yarleque Espinoza” - Catacaos

1.1.3. Estructura Funcional²

Funciones:

Estas sirven para ayudar al análisis del sistema actual en lo que se requiere a como deben desenvolverse las funciones dentro de la estructura orgánica.

Funciones de Dirección

Funciones del Director

- Planifica, organiza, dirige y evalúa las actividades del Instituto, así como proyectar su desarrollo.
- Convoca y preside reuniones Técnico-Pedagógicas y administrativas.
- Aprueba el Plan Anual de Trabajo, el cuadro de necesidades; el Presupuesto de Operación, el Cuadro de Distribución de Horas Académicas.
- Dirige el proceso de Admisión, aprueba matrícula y autoriza traslados, exoneraciones y convalidaciones.
- Otras funciones que le encomienda la autoridad de acuerdo a ley.

Funciones de la Secretaria de Dirección

Depende del Director del IST “MYE” y desempeña las siguientes funciones:

- Recepciona y registra la documentación que ingresa.
- Atiende los servicios de comunicación.
- Mantiene actualizado el archivo de la Dirección.
- Realiza otras funciones que le asigne el director del instituto.

Funciones de Secretaría General

Funciones del Secretario Docente

- Prepara la correspondencia oficial del instituto, que deben ser autorizadas por el director.
- Registra y archiva actas de evaluación, documentos académicos – administrativos.

²“Manual de Organización del Instituto Superior Tecnológico “Manuel Yarleque Espinoza” de Catacaos

- Expide certificados de estudio, diplomas, títulos y otros documentos oficiales.
- Ejecuta el proceso de matricula de los educandos.
- Prepara información cuantitativa de matricula por carrera profesional, semestre académico; deserción estudiantil, alumnos, egresados, titulados y otros.
- Interpreta y adecua las normas sobre matricula, evaluación, certificación y títulos de los educandos.

Funciones del Departamento de Formación General y Profesional

Funciones del Jefe

- Planifica, organiza, dirige y evalúa las actividades del Departamento.
- Asesora y apoya la actualización permanente de los docentes.
- Dirige, orienta, supervisa y evalúa los procesos de programación, ejecución y evaluación curricular.
- Verifica y visa las actas y otros documentos de evaluación académica, debiendo alcanzar oportunamente a secretaría general.
- Ejerce tareas de control académico y de las actividades durante la permanencia de los docentes en el cumplimiento de la jornada establecida.
- Realiza las demás funciones que el Director le asigna.

Funciones de Docentes de Formación General y Profesional

- Organizan, coordinan, ejecutan y evalúan la programación curricular de las asignaturas a su cargo, adecuando el currículo a realidad institucional y socio-económica de la zona.
- Programan, dirigen y evalúan el proceso enseñanza-aprendizaje en el aula y el trabajo independiente de los educandos.
- Participan en la formulación y ejecución y evaluación del plan de trabajo del Departamento de Formación General y Profesional.
- Realizan además funciones que les asigne el jefe del Departamento de Formación General y Profesional.

Funciones del Área de Administración

Funciones del Jefe de Área

- Verifica los depósitos de los ingresos propios y efectúa la conciliación bancaria.
- Verifica y visa los presupuestos de abastecimiento, compras y los inventarios de bienes muebles e inmuebles de la institución.
- Verifica el registro de las operaciones contables del instituto.

Funciones de La Secretaria (Funciones Adicionales de Tesorería)

- Emite y forma los recibos de ingresos y egresos efectuados.
 - Efectúa los depósitos de los ingresos captados por la institución
 - Dispone y controla el fondo de caja chica.
- Efectúa el registro de las operaciones en la documentación preliminar para su posterior anotación en los libros Contables por parte del contador.

1.2. METODOLOGÍA PARA EL DESARROLLO DEL SISTEMA

1.2.1. Análisis y Diseño Orientado a Objetos

El análisis y diseño orientado a objetos constituye una nueva forma de pensar acerca de problemas empleando modelos que son útiles para comunicarse con expertos en esa aplicación, modelar empresas, preparar documentación, diseñar programas y bases de datos.

La esencia del análisis y diseño orientado a objetos es la identificación y organización de conceptos del dominio de la aplicación, y no de su presentación final en un lenguaje de programación, es decir, es un proceso conceptual independiente de sí el lenguaje es orientado a objetos.

El uso del análisis y diseño orientado a objetos puede facilitar mucho la creación de prototipos, y las técnicas de desarrollo evolutivo de software. Los objetos son inherentemente reutilizables, y se puede crear un catálogo de objetos que podemos usar en sucesivas aplicaciones. De esta forma, podemos obtener rápidamente un prototipo del sistema, que pueda ser evaluado por el cliente, a partir de objetos analizables, diseñados e implementados en aplicaciones anteriores. Y lo que es más importante, dada la facilidad de reutilización de estos objetos, el prototipo puede ir evolucionado hacia convertirse en el sistema final, según vamos refinando los objetos de acuerdo a un proceso de especificación incremental.

En conclusión: “El análisis y diseño orientado a objetos es un proceso continuo donde no existe una división clara entre el análisis y el diseño, además, está centrada en un concepto de modelado y no en técnicas de implementación por lo que es independiente de todo lenguaje de programación hasta las etapas finales³”.

METODOLOGÍA OMT⁴

La Técnica de Modelado de Objetos (OMT) fue desarrollada por James Rumbaugh, es una metodología orientada a objetos muy difundida que se hace cargo de todo el ciclo de vida del software. Utiliza los mismos conceptos y la misma notación a lo largo de todo el ciclo de vida.

³“Análisis y Diseño Orientado a Objetos” de James Martín - Prentice Hall Hispanoamericana, Primera Edición , 1994.

⁴“Modelado y Diseño Orientado de Objetos – Metodología OMT” de James Rumbaugh - Primera Edición PRENTICE HALL, 1996.

Divide el ciclo de vida del software en cuatro fases consecutivas: análisis de objetos, diseño del sistema, diseño de objetos e implementación (Fig. I.2)

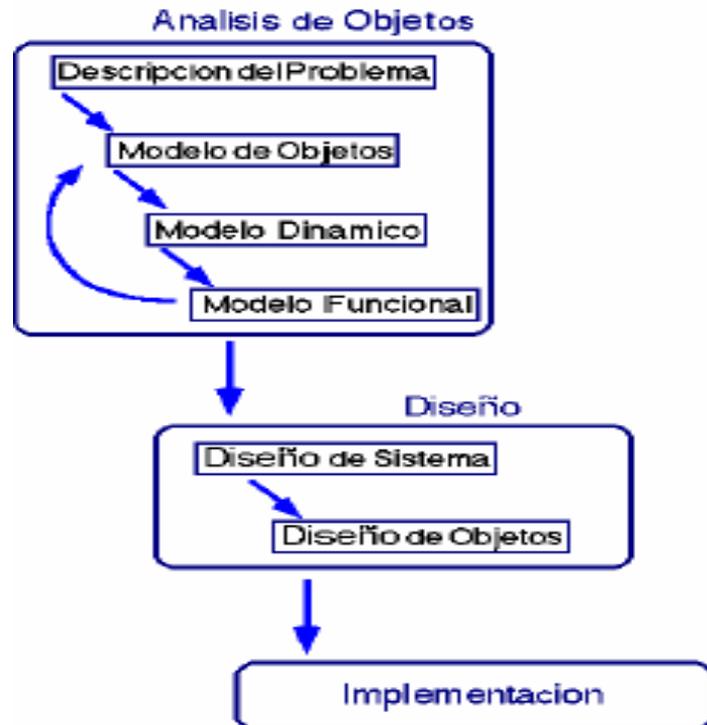


Figura I.2

Análisis de Objetos

Esta fase tiene por finalidad la descripción del problema en un modelo de análisis.

Diseño de Sistemas

Comprende la arquitectura básica. En esta fase se tomarán las decisiones estratégicas de diseño.

Diseño de objetos

Es el último paso antes de implementar, y sirve para definir completamente todas las características de los objetos. Se detallan los tres modelos ya descritos en el análisis de objetos para poder implementarlo y optimizar el programa.

Implementación

Se codifica el diseño en un lenguaje específico.

METODOLOGÍA DE BOOCH⁵

Esta metodología es también muy popular, detalla un método ofreciendo también análisis y diseño ‘iterativo’, centrándose en el lado del diseño. Las etapas de esta metodología son: Análisis de Requerimientos, Análisis de Dominio y Diseño.

Análisis de Requerimientos

En esta etapa se define qué quiere el usuario del sistema. Es una etapa de alto nivel que identifica las funciones principales del sistema, el alcance del modelamiento del mundo y documenta los procesos principales y las políticas que el sistema va a soportar. No se definen pasos formales, ya que éstos dependen de qué tan nuevo es el proyecto, la disponibilidad de expertos y usuarios y la disponibilidad de documentos adicionales.

Análisis de Dominio

Es el proceso de definir de una manera concisa, precisa y orientado a objetos la parte del modelo del mundo del sistema. Las siguientes actividades son parte de esta etapa:

- **Diagrama de clases**, clases del dominio claves y sus relaciones.
- **Especificación de las clases**
- **Vistas de herencia**. Diagramas de clases con este tipo de relaciones.
- **Diagramas de escenarios de objetos**
- **Especificación de objetos**, que relacionan objetos y sus clases.

Diseño

Es el proceso de determinar una implementación efectiva y eficiente que realice las funciones y tenga la información del análisis de dominio. Las siguientes actividades se plantean en esta etapa:

⁵“Análisis y Diseño Orientado a Objetos” de G. Booch. Benjamin/Cummings, (1994)

- **Descripción de arquitectura**, describe las decisiones más importantes de diseño como son el conjunto de procesos, manejadores de bases de datos, sistemas operativos, lenguajes, etc.
- **Descripciones de prototipo**, describen las metas y contenido de las implementaciones sucesivas de prototipos, su proceso de desarrollo y la forma de probar requerimientos.
- **Diagramas de Categorías**
- **Diagramas de clases en diseño**, detallan las abstracciones de análisis con características de implementación.
- **Diagramas de objetos en diseño**, muestran las operaciones necesarias para desarrollar una operación.
- **Nuevas especificaciones**
- **Especificaciones de clases corregidas**, muestra la especificación completa de los métodos con algoritmos complicados, la implementación de relaciones y el tipo de atributos.

METODOLOGÍA UML⁶

Esta metodología surge cuando J. Rumbaugh y G. Booch en el año 1994 se unen en una empresa común (de objetivos y de negocio) Rational Software Corporation donde unifican sus dos métodos y como fruto de dicha unión aparece en octubre de 1995 **UML 0.8** (*Unified Modeling Language*). A finales de ese mismo año se une al grupo I. Jacobson, creador de OOSE (Object Oriented Software Engineer) para finalmente conseguir un modelo totalmente unificado.

En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas. La metodología ofrece los siguientes diagramas a modelar sistemas.

⁶“El Lenguaje Unificado de Modelado” de G. Booch, J. Rumbaugh, I. Jacobson - Addison Wesley Iberoamericana, 1999.

- **Diagramas de Estructura Estática**
 - **Diagrama de Casos de Uso:** Representa un conjunto de casos de uso, actores y sus relaciones. Vista de caso de uso estática, para organizar y modelar el comportamiento del sistema.
 - **Diagrama de Clases:** Para modelar la estructura estática de las clases del sistema.
 - **Diagrama de Objetos:** Para modelar la estructura estática de los objetos en el sistema.

- **Diagramas Dinámicos**
 - **Diagramas de Secuencia:** Para modelar el paso de mensajes entre objetos.
 - **Diagramas de Colaboración:** Para modelar interacciones entre objetos.
 - **Diagramas de Estado:** Representa la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

Teniendo en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación. En este trabajo de tesis se sigue el método propuesto por Craig Larman⁷ que se ajusta a un ciclo de vida iterativo e incremental dirigido por casos de uso.

Un proceso de desarrollo es el que se ocupa de plantear cómo se realiza el análisis y el diseño, y cómo se relacionan los productos de ambos, entonces la construcción de sistemas de software va a poder ser planificable y repetible, y la probabilidad de obtener un sistema de mejor calidad al final del proceso aumenta considerablemente. Este proceso define una serie de actividades que pueden realizarse en cada fase, las cuales deben adaptarse según las condiciones del proyecto que se esté llevando a cabo. Se ha escogido seguir este proceso debido a que aplican los últimos avances en Ingeniería del Software, y a que adopta un enfoque eminentemente práctico.

⁷“Desarrollo Orientado a Objetos con UML” de Xavier Ferré Graú, María Isabel Sánchez Segura basado en UML y Patrones de C. Larman. Prentice Hall, 1999.

Las tres fases al nivel más alto son las siguientes:

- **Planificación:** Planificación, y construcción de prototipos.
- **Construcción.** La creación del sistema. Las fases dentro de esta etapa son las siguientes:
 - **Análisis:** Se analiza el problema a resolver desde la perspectiva del usuario y de las entidades externas que van a solicitar servicios al sistema.
 - **Diseño:** El sistema se especifica en detalle, describiendo como va a funcionar internamente para satisfacer lo especificado en el análisis.
 - **Implementación:** Se lleva lo especificado en el diseño en un lenguaje de programación.
 - **Pruebas:** Se lleva a cabo una serie de pruebas para corroborar que el software funciona correctamente y que satisface lo especificado en la etapa de Planificación.
- **Instalación:** La puesta en marcha del sistema en el entorno previsto de uso.

La mayor parte del esfuerzo y del tiempo en un proyecto de desarrollo se basa en la etapa de Construcción. Esta etapa se lleva a cabo tomando en cada iteración un subconjunto de los requisitos que llevados a través del análisis y diseño hasta la implementación y pruebas (ver Figura I.3) permite que el sistema vaya creciendo incrementalmente en cada ciclo.

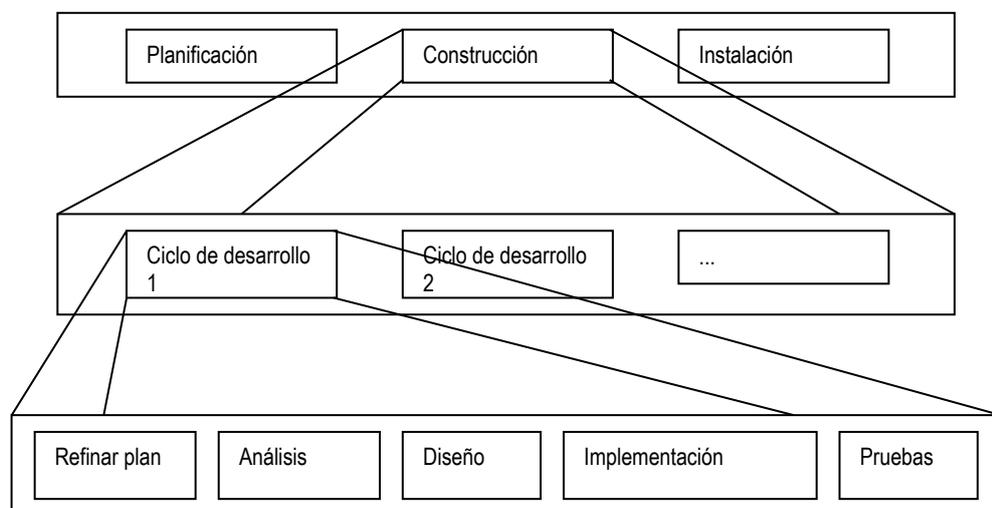


Figura I.3. Desarrollo iterativo en la construcción

Análisis

En la fase de análisis de un ciclo de desarrollo se investiga el problema, sobre los conceptos relacionados con el subconjunto de casos de uso que se esté tratando. Se intenta llegar a una buena comprensión del problema por parte del equipo de desarrollo, sin entrar a como va a ser la solución en cuanto a detalles de implementación.

Las actividades de esta fase son las siguientes:

1. **Definir requisitos:** Es la descripción de necesidades o aspiraciones respecto a un producto.
2. **Definir casos esenciales de uso:** Un caso de uso es un documento narrativo que describe la secuencia de eventos de un actor que usa un sistema para completar un proceso. Los casos esenciales de uso se describen a un nivel más detallado.
3. **Crear el diagrama de casos de uso.**
4. **Crear el Modelo Conceptual:** El Modelo Conceptual es el diagrama de estructura estática de UML que permite una representación de conceptos del mundo real, no de componentes de software.
5. **Diagramas de Secuencia del sistema.**
6. **Definir contratos de operación:** Un contrato de operación del sistema describe cambios en el estado del sistema cuando una operación del sistema es invocada.
7. **Diagramas de Estados.**

Diseño

En la fase de diseño se crea una solución a nivel lógico para satisfacer los requisitos, basándose en el conocimiento reunido en la fase de análisis.

Las actividades de esta fase son las siguientes:

1. **Definir los casos reales de uso:** Los casos reales de uso describen el diseño real del caso de uso según una tecnología concreta de entrada y salida y su implementación.
2. **Definir Diagramas de Interacción:** Los Diagramas de Interacción o Colaboración muestran el intercambio de mensajes entre instancias

del modelo de clases para cumplir las post-condiciones establecidas en un contrato.

3. **Definir el Diagrama de Clases:** Un diagrama de clases muestra la especificación para las clases software de una aplicación.
4. **Definir el esquema de base de datos.**

1.2.2. Comparación entre el Análisis y Diseño Orientados a Objetos y el Análisis y Diseño Estructurado⁸

Tabla I.1.

Análisis y Diseño Estructurado	Análisis y Diseño Orientados a Objetos
<ul style="list-style-type: none"> • El análisis y diseño estructurado se basa fundamentalmente en la descomposición funcional del sistema que queremos construir. Esta descomposición funcional requiere traducir el dominio del problema en una serie de funciones y subfunciones. Lo que puede parecer la forma más directa de implementar un objetivo deseado, pero el sistema resultante puede ser frágil. • Los cambios en los requisitos afectan notablemente a la funcionalidad del sistema, por lo que afectan mucho al software desarrollado, por lo que puede ser necesaria una importante reestructuración. 	<ul style="list-style-type: none"> • El análisis y diseño orientado a objetos buscan ante todo descomponer un espacio de problema por objetos y esto permite analizar mejor el dominio del problema. El concepto orientado a objetos es más simple y permite una mejor comunicación entre el analista y el experto en el dominio del problema (es decir, el cliente). • Las evoluciones de los requisitos afectan en mucha menor medida a los objetos que componen o manejan el sistema, que son mucho más estables.

⁸“Modelado y Diseño Orientado a Objetos” de J. Rumbaugh Pág.26 – Prentice Hall,1996.

1.2.3. Fundamento de la selección de la metodología

UML ha sido desarrollado con el propósito de ser útil para modelar diferentes sistemas de información, técnicos (telecomunicaciones, industria), y no sólo es útil para la programación sino también para modelar negocios, es decir, los procesos y procedimientos que establecen el funcionamiento de una empresa. Por lo tanto existe suficiente razón para que pueda ser utilizado en la mayoría de los proyectos de software. Este nuevo enfoque “orientado a objetos” trajo consigo muchas metodologías usadas en el desarrollo de software orientado a objetos, pero en 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de un metamodelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0 fue propuesto y aprobado por el OMG en noviembre de 1997.

El Lenguaje Unificado de Modelado (UML) es una metodología que tiene como objetivo conseguir un **modelo unificado**, abierto, que siga evolucionando en conjunto y no por separado tal como estaba ocurriendo hasta ese entonces, comprensible por el hombre, utilizable por la máquina y fundamentalmente con la capacidad de unificar las perspectivas de diferentes sistemas. Asimismo, esta dirigido por los casos de uso, centrado en la arquitectura y es iterativo e incremental, permite guiar a los desarrolladores en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los clientes. Además, prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

CAPITULO II: ANÁLISIS DE LA SITUACIÓN ACTUAL

CAPITULO II: ANÁLISIS DE LA SITUACIÓN ACTUAL

2.1. FORMULACION DEL PROBLEMA

Actualmente el Instituto Superior Tecnológico “Manuel Yarleque Espinoza” de Catacaos no cuenta con un sistema de información que le permite automatizar los procesos y actividades académicas que se desarrollan en la institución. El desarrollo del proceso académico se realiza de forma manual con ayuda de algunos programas (Excel, Word) en los que se desempeñan funciones específicas, pero frente a las muchas limitaciones y sobre todo a la necesidad de tener la información académica oportuna y disponible con el fin de procesar información en forma rápida, precisa y segura; es por ello que la institución académica ha creído conveniente el desarrollo del proyecto: “Desarrollo del Sistema de Información Académico para el Instituto Superior Tecnológico Manuel Yarleque Espinoza”.

El Departamento de Formación General y Profesional es el órgano mayor involucrado en el proceso académico que en coordinación con Secretaría General y el Área de Administración trabajan en el proceso académico.

Secretaría General es el órgano encargado de crear un código único por alumno, el que le es asignado desde que inicia su carrera hasta que la culmina. Sin embargo, esta codificación manual ha duplicado en muchas ocasiones los códigos asignados generando problemas.

El Departamento de Formación General y Profesional que es el órgano encargado de la Programación Curricular y de los planes curriculares de cada carrera profesional; no cuenta aún con una codificación de los cursos teniendo la necesidad de asignar un código único por cada curso. El Área de Administración es el área encargada de todos los procesos de pago, específicamente de captar el pago por derecho de matrícula.

En este proyecto se desarrolla todo el proceso académico, empezando por el proceso de Programación Académica y siguiendo con la Matrícula de ingresantes, Matrícula de regulares, Inscripción de cursos y el registro de notas. Son estos procesos los que permiten tener toda la historia académica de un alumno y a la vez dar lugar a un mejor servicio, mayor rapidez y seguridad en el tratamiento de la información.

CAPITULO II: ANÁLISIS DE LA SITUACIÓN ACTUAL

2.2. DESCRIPCIÓN DEL SISTEMA ACTUAL

Las actividades académicas anuales del IST “MYE” se desarrollan en dos períodos semestrales. Cada año, se da el proceso de admisión con la finalidad de captar nuevos alumnos al IST “MYE”, para el cual se tiene un sistema de calificación mediante los exámenes de admisión, que genera un registro de ingresantes, pero que en el presente estudio no forman parte de nuestro objetivo. El semestre académico tiene una serie de procesos que serán descritos a continuación:

PROGRAMACIÓN ACADÉMICA

El desarrollo de un semestre académico se inicia con la elaboración de la programación académica. El Departamento de Formación General y Profesional de acuerdo al Reglamento Interno del Instituto, el Plan de Trabajo Anual y normas vigentes, inicia este proceso, verifica los resultados del semestre anterior, estima las asignaturas que han de ser programadas en el semestre. La relación de asignaturas a ofrecer es enviada al Consejo de Docentes de Formación General y Profesional, el cual distribuye internamente la carga lectiva, designando la asignatura al docente; luego en unanimidad con el Jefe del Departamento de Formación General y Profesional determina los horarios de clase para cada asignatura.

PROCESO DE MATRICULA

Dirección solicita al Área de Administración un reporte con los deudores del semestre anterior por concepto de matricula y posteriormente coordina con los órganos de la institución y definen un cronograma para tal proceso.

El área de administración ejecuta la matricula y verifica el estado de deudas: inicialmente el alumno paga un derecho de matricula y recibe una boleta de pago.

La oficina de la Secretaría General es la encargada de recepcionar la copia de la boleta de pago, asimismo verifica si se trata de un alumno nuevo o regular. Al tratarse de un nuevo alumno el Secretario Docente tiene la responsabilidad de asignar un código único al alumno y proseguir con el proceso normal, de entregar al alumno los formatos de la ficha de matricula por duplicado.

El alumno llena la ficha de matricula y terminada ésta es verificada por el Secretario Docente y el Jefe Administrativo los cuales la firman y dan el Vo Bo, incluido la firma del alumno.

CAPITULO II: ANÁLISIS DE LA SITUACIÓN ACTUAL

El Secretario Docente recibe las fichas de matrícula inscritas, lo que resume para luego emitir los Registros de Alumnos al Departamento de Formación General y Profesional, el cual se encarga de hacerlos llegar a los docentes, para controlar la asistencia y avance del alumno.

Antes de terminar el semestre, la oficina de Secretaría General emite las actas al Departamento de Formación General y Profesional. Este órgano después de haber verificado los registros de evaluaciones realiza el traspaso de notas (nota promocional del curso) al acta para posteriormente enviarlas a Secretaría General y luego al Director quien da el Vo Bo con una firma y sello.

EVALUACIÓN ACADEMICA

El profesor evalúa al alumno a través de cuatro unidades funcionales, el que es supervisado por el Departamento de Formación General y Profesional de manera interna, este proceso dejará como constancia la nota promocional del curso.

Secretaría General procesa la información académica del semestre; manualmente, anota las notas de las actas promocionales en las boletas de notas del alumno. Finalmente la boleta de notas o constancia de notas es verificada y sellada por Dirección para luego ser entregadas al alumno. Asimismo, los alumnos solicitan documentos como: constancias, certificados de estudio, informes académicos, etc.

2.3. ALCANCE DEL SISTEMA PROPUESTO

En el presente estudio se pretende instaurar un sistema que permita a la institución manejar información académica de manera actualizada y oportuna. De esta manera nuestra propuesta contribuye con los objetivos de la institución, además de considerar las limitaciones que se presentan en el IST-MYE.

Por lo que estudiando las alternativas, se ha considerado: Un sistema de información centralizado por la oficina de Secretaría General con una terminal en la oficina de la Dirección, la que operaría de acuerdo a sus requerimientos.

Esta alternativa pese a centralizar la información académica permitirá mejorar la planificación de las actividades académicas, brindar mejor servicio y reflejar eficiencia en los servicios académicos. Además, aminora el número de personas en el manejo del sistema.

3.1. ANÁLISIS

3.1.1. Definir requisitos.

Al determinar los requisitos se utilizaron diversas fuentes, tales como: entrevistas, cuestionarios, entre otros; información, que se puede traducir en un formato a seguir para la especificación de requisitos. Este formato de especificación de requisitos no está definido en UML, pero se incluye este punto para resaltar esta actividad que es un paso clave en la creación de software. El formato incluye los siguientes puntos:

- Metas o propósitos
- Funciones del Sistema

Metas o propósitos

En términos generales, la meta es una mayor automatización de los datos académicos de un alumno, de un curso, y asimismo dar soporte a servicios más rápidos y mejores. Específicamente la meta incluye:

- Control sobre el número de alumnos matriculados por especialidad.
- Control del dinero ingresado por concepto de matriculas.
- Control de los alumnos inscritos en un curso.
- Control sobre la programación académica de un semestre.
- Control sobre las notas de un alumno en cada semestre para generar reportes sobre el record académico y boletas de notas.
- Control del número de aprobados y desaprobados de un curso.
- Control sobre el número de alumnos con asignaturas a cargo y/o repitencia.
- Control automático de deudas de los alumnos por concepto de matrícula.

Funciones del Sistema

El objetivo de describir las funciones del sistema es entender los detalles del análisis y del diseño, no el funcionamiento de la institución académica. Las funciones pueden clasificarse en: Evidente (el usuario debe saber que se ha realizado) y Oculta (debe realizarse, aunque no es visible para el usuario).

A continuación el sistema presenta las siguientes funciones:

Tabla III.1: Funciones del sistema.

Ref. #	Función	Categoría
R1.1	El operador ingresa una identificación y una contraseña para utilizar el sistema.	Evidente
R1.2	Registra la programación de carreras de un semestre: carrera, sección, turno y aula.	Evidente
Ref. #	Función	Categoría
R1.3	Captura la información sobre la carrera profesional programada haciendo una selección sobre la lista de carreras que ofrece la institución.	Evidente
R1.4	Captura la información sobre el aula asignada a una carrera en un semestre haciendo una selección sobre la lista de aulas disponibles de la institución.	Evidente
R1.5	Se registran las carreras programas en el semestre ya efectuadas.	Ocultas
R1.6	Registra la distribución académica por carrera programada en un semestre: las asignaturas programadas.	Evidente
R1.7	Captura la información sobre la asignatura programada haciendo una selección sobre la lista de asignaturas correspondientes a la carrera y sección.	Evidente
R1.8	Muestra el código, número de horas y tipo del curso seleccionado.	Evidente
Ref. #	Función	Categoría
R1.9	Captura la información sobre la asignatura programada haciendo una selección sobre la lista de asignaturas correspondientes a la carrera y sección.	Evidente
R1.10	Muestra el código, número de horas, tipo y encargado del curso seleccionado.	Evidente
R1.11	Ofrece un mecanismo persistente.	Ocultas
R1.12	Registra la matrícula: código de alumno y datos del recibo de ingreso.	Evidente
R1.13	Captura la información sobre el alumno matriculado usando su código.	Evidente
R1.14	Muestra el nombre, carrera, turno y sección del alumno registrado.	Evidente
R1.15	Muestra el costo de la matrícula.	Evidente
R1.16	Se registran las matrículas efectuadas y su correspondiente pago.	Ocultas
R1.17	Registra la inscripción de una asignatura: código de alumno y nombre de asignatura.	Evidente
R1.18	Captura la información sobre el alumno matriculado usando su código.	Evidente
R1.19	Muestra información de alumno matriculado: nombre, especialidad y ciclo.	Evidente

CAPITULO III: MODELADO DEL SISTEMA

Ref. #	Función	Categoría
R1.20	Muestra una lista los posibles cursos a inscribirse de acuerdo a la carrera, ciclo y cursos programados en un semestre.	Evidente
R1.21	Captura la información sobre la asignatura seleccionada.	Evidente
R1.22	Muestra código, número de horas y docente del curso seleccionado.	Evidente
R1.23	Incrementa la lista de cursos llevados por un alumno.	Oculto
R1.24	Se registran las inscripciones de cursos efectuadas.	Oculto
R1.25	Ofrece mecanismos de comunicación entre los procesos de distribución académica, matrícula e inscripción de cursos.	Oculto
R1.26	Registra nota promocional de una asignatura: nombre del docente, nombre de la asignatura, código del alumno y nota obtenida.	Evidente
R1.27	Captura la información sobre el docente encargado de la asignatura usando las primeras letras del apellido para realizar una búsqueda y seleccionar el docente respectivo.	Evidente
Ref. #	Función	Categoría
R1.28	Muestra una lista de las asignaturas a cargo en un semestre.	Evidente
R1.29	Captura la información sobre la asignatura seleccionada.	Evidente
R1.30	Captura la información sobre el alumno inscrito en la asignatura seleccionada y muestra nombre del alumno.	Evidente
R1.31	Captura y añade información al curso llevado por un alumno.	Oculto
R1.32	Se registran las evaluaciones ya efectuadas.	Oculto
R1.33	Registra el pago de una matrícula (en caso de tratarse de una matrícula de pago fraccionado): código de alumno, número de boleta, fecha y monto.	Evidente
R1.34	Actualiza el saldo de una matrícula	Oculto
R1.35	Relación de matrículas de un determinado semestre para obtener relación de deudores.	Evidente
R1.36	Registra inscripción de curso a cargo: código de alumno y nombre de la asignatura a cargo.	Evidente
R1.37	Captura información sobre el alumno matriculado usando su código.	evidente
R1.38	Captura información sobre la asignatura a cargo.	Evidente
Ref. #	Función	Categoría
R1.39	Valida que el nombre ingresado de la asignatura sea una asignatura a cargo del alumno matriculado.	Oculto
R1.40	Muestra el nombre y código de la asignatura.	Evidente

CAPITULO III: MODELADO DEL SISTEMA

R1.41	Se registran las inscripciones a cargo efectuadas.	Ocultas
R1.42	Registra nuevo alumno: nombres, apellidos, sexo, fecha de nacimiento, lugar de nacimiento, fecha de ingreso, etc. Además de la carrera profesional y turno a la que pertenece.	Evidente
R1.43	Genera código del alumno basándose en el año de ingreso y código de la carrera profesional a la que pertenece.	Ocultas
R1.44	Muestra código único del alumno.	Evidente
R1.45	Registra nuevo docente: nombres, apellidos, sexo, fecha de nacimiento, lugar de nacimiento, etc. Además de los títulos profesionales obtenidos por el docente.	Evidente
R1.46	Registra nuevo plan curricular: carrera, fecha de creación, año de vigencia, número de horas total y número de horas de practica profesionales, etc.	Evidente
Ref. #	Función	Categoría
R1.47	Registra los cursos de un plan curricular: nombre, descripción, nivel, ciclo y número de horas.	Evidente
R1.48	Captura la información sobre el curso añadido a un plan curricular.	Evidente
R1.49	Se registran los cursos añadidos a un plan curricular.	Ocultas
R1.50	Calcula el total de total de número horas de un plan curricular.	Ocultas
R1.51	Ofrece un mecanismo de almacenamiento persistente.	Ocultas
R1.52	Registra una nueva aula: nombre, estado y capacidad.	Ocultas
R1.53	Registra una nueva carrera: nombre, fecha de creación, número de resolución, título a otorgar, número de ciclos, etc.	Evidente
R1.54	Registra horario por carrera programada: curso, día y hora.	Evidente
R1.55	Captura información sobre asignatura programada en una carrera y sección.	Evidente

3.1.2. Casos Esenciales de Uso

Un caso de uso es un documento narrativo que describe las secuencias de eventos de un actor (agente externo) y que constituyen un paso muy útil en las especificaciones del sistema. Dado que estos se clasifican en casos de uso de alto nivel y casos de uso expandidos; primero se ha iniciado por describir los casos de uso de alto nivel para lograr un rápido entendimiento de los procesos principales.

A continuación se describen los casos de uso del sistema en desarrollo:

Tabla III.2

Casos de Uso	Registrar Programación de Carreras.
Actores	Jefe del DFGP(iniciador), Operador
Tipo	Primario
Descripción	El Jefe del DFGP envía al operador el cuadro de la programación de carreras del semestre en curso. El operador registra las aulas asignadas a una carrera. Al terminar la transacción, el operador remite el cuadro de programación de carreras al Jefe del DFGP.

Tabla III.3

Casos de Uso	Registrar Programación Académica.
Actores	Jefe del DFGP(iniciador), Operador
Tipo	Primario
Descripción	El Jefe del DFGP aprueba y envía al operador el cuadro de distribución académica del semestre en curso. El operador registra las asignaturas que le han sido asignadas a un docente. Al terminar la transacción, el operador remite un informe del cuadro de distribución de horas del semestre al Jefe del DFGP.

Tabla III.4

Casos de Uso	Registrar horario
Actores	Jefe del DFGP (iniciador), Operador del sistema.
Tipo	Secundario
Descripción	El Jefe del DFGP envía al operador el horario del semestre en curso de una carrera programada. El operador registra las asignaturas y su respectivo horario.

CAPITULO III: MODELADO DEL SISTEMA

Tabla III.5

Casos de Uso	Obtener deudas de un semestre.
Actores	Operador del sistema (iniciador).
Tipo	Primario
Descripción	El operador solicita al sistema un reporte de deudas por concepto de matricula del semestre indicado. El sistema comprueba la existencia de deudores y lo muestra, entonces el operador solicita la impresión del reporte para luego hacerlo llegar a cada oficina que sea necesaria.

Tabla III.6.

Casos de Uso	Registrar Pago de Matricula.
Actores	Secretario Docente (iniciador), Operador.
Tipo	Primario
Descripción	El Secretario Docente entrega la boleta de pago por derecho de matricula de un alumno al operador. El operador registra los datos de la boleta de pago y el monto efectuado. Al terminar la transacción, el operador remite la boleta de pago del alumno al Secretario Docente.

Tabla III.7.

Casos de Uso	Registrar Inscripción de cursos
Actores	Secretario Docente (iniciador), Operador.
Tipo	Primario
Descripción	El Secretario Docente entrega al operador la copia de la ficha de matricula de un alumno. El operador registra los cursos en los que se ha inscrito un alumno. Al terminar el operador da el visto bueno a la ficha y lo remite al Secretario Docente.

Tabla III.8.

Casos de Uso	Registrar nota promocional de asignatura.
Actores	Secretario Docente (iniciador), Operador.
Tipo	Primario
Descripción	El Secretario Docente entrega al operador el acta de notas de una asignatura. El operador registra las notas obtenidas por los alumnos inscritos en una asignatura. Al terminar el operador remite el acta al Secretario Docente.

CAPITULO III: MODELADO DEL SISTEMA

Tabla III.9.

Casos de Uso	Registrar nuevo plan curricular
Actores	Jefe del DFGP(iniciador), Operador del sistema.
Tipo	Secundario
Descripción	El Jefe del DFGP envía al operador el nuevo plan curricular de una carrera con los cursos contenidos en ese plan. El operador registra cada uno de los cursos contenidos en el nuevo plan. Al terminar el operador remite el plan curricular al Departamento de Formación General y Profesional.

Tabla III.10.

Casos de Uso	Registrar nuevo docente.
Actores	Director (iniciador), operador.
Tipo	Primario
Descripción	El director autoriza y hace llegar al operador el expediente del nuevo docente. El operador registra los datos del docente. Al terminar el operador remite el expediente.

Tabla III.11.

Casos de Uso	Control Acceso
Actores	Operador del sistema.
Tipo	Primario
Descripción	El operador ingresa la clave y contraseña para acceder al sistema. El sistema verifica e informa la disponibilidad del acceso solicitado y permite el acceso al sistema.

Una vez comprendido los procesos principales, como segundo punto de esta sección se procederá a detallar profundamente los casos de uso más relevantes y que interactúan directamente con el sistema en desarrollo.

A continuación se describen los casos **expandidos** de uso del sistema en desarrollo:

Tabla III.12.

Casos de Uso	Registrar Programación de Carreras
Actores	Jefe del DFGP (iniciador), Operador del sistema.
Propósito	Capturar la programación de carreras y las aulas asignadas a una carrera en un semestre.
Tipo	Primario y esencial.
Referencias	Funciones: R1.2, R1.3, R1.4, R1.5, R1,52.
Cruzadas	Casos de uso: Control de Acceso, Registrar aula.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el Jefe del DFGP envía al operador del SIA la programación de carreras del semestre actual.	
2. El operador selecciona la carrera turno y ciclo programado en el semestre.	3. Determina el código programado. Muestra una lista de aulas disponibles.
4. El operador selecciona un aula.	5. Muestra el código, estado y ubicación del aula.
6. Terminada dicha operación, el operador indica al SIA que capturé dicha programación.	7. Captura la programación de la carrera en dicho semestre.

Cursos Alternos:

- Línea 2: Si ya existe la programación de la carrera y sección se cancela dicha operación.

Tabla III.13.

Casos de Uso	Registrar Programación Académica.
Actores	Jefe del DFGP (iniciador), Operador
Propósito	Capturar los cursos programados en un semestre y asignar un docente encargado.
Tipo	Primario y esencial.
Referencias	Funciones: R1.6, R1.7, R1.8, R1.9, R1.10, R1.11.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el Jefe del DFGP envía al operador del SIA la distribución académica del semestre en curso.	
2. El operador selecciona la carrera, turno y sección programada en el semestre.	3. Determina una lista de las posibles asignaturas a programar de acuerdo a la carrera y sección.
4. El operador selecciona el nombre	5. Determina el código, tipo y número

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
de la asignatura a programar.	de horas de la asignatura.
6. El operador ingresa el nombre del docente a cargo de la asignatura	7. Determina el código del docente.
8. Al terminar de registrar la asignatura programada y el respectivo docente, el operador indica que se concluyó la captura de la asignatura.	9. Se registra la programación de la asignatura. Calcula y presenta el total de horas programadas a la carrera.

Cursos Alternos:

- Línea 6: Invalidar búsqueda de nombre indicado.

Tabla III.14.

Casos de Uso	Registrar horario.
Actores	Jefe del DFGP (iniciador), Operador del sistema.
Propósito	Capturar la distribución de horas de clase de las asignaturas programadas para una carrera – sección en un semestre.
Tipo	Secundario y esencial.
Referencias	Funciones: R1.54, R1.55.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el Jefe del DFGP envía al operador del SIA el horario de las asignaturas de una determinada carrera - sección en un semestre.	
2. El operador selecciona la carrera, turno y sección.	3. Procesa una consulta de todas las asignaturas programadas en el semestre por carrera-sección y los muestra.
4. Por cada asignatura, el operador selecciona una asignatura e introduce el día y la hora respectiva.	5. Determina el código, número de horas de la asignatura y aula específica donde se dictará la asignatura.
6. Al terminar de registrar el día y hora en	7. Registra horario de la

CAPITULO III: MODELADO DEL SISTEMA

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
que se dictará la asignatura, el operador le indica al SIA que concluyó la captura.	asignatura.
8. El operador remite al Departamento de Formación General y Profesional un informe del cuadro de distribución de Horas de Clase de una carrera – sección e indica al DFGP que tiene disponibles los horarios por docente.	
9. El Jefe del DFGP solicita un reporte de los horarios por docente.	
10. El operador solicita al sistema un informe de horarios por docente.	11. Genera un reporte de horarios por docente.

Cursos Alternos:

- Línea 4: Día u hora no son válidos. Indicar error.

Tabla III.15.

Casos de Uso	Obtener deudas de un semestre.
Actores	Director (iniciador), Operador del sistema.
Propósito	Capturar la información de deudores por concepto de matrícula de un determinado semestre para luego visualizarla y obtener un reporte.
Tipo	Primario y esencial.
Referencias	Funciones: R1.35.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el operador solicita al SIA los deudores de un determinado semestre.	
2. El operador introduce el año y semestre.	3. Genera un informe de alumnos deudores clasificados por carrera profesional, indicando su saldo por concepto de matrícula.
4. Indica al sistema que imprima dicho reporte.	5. Imprime reporte.

CAPITULO III: MODELADO DEL SISTEMA

6. El operador envía los reportes a Dirección para su uso correspondiente.

Cursos Alternos:

- Línea 2: Año y semestre no válidos. Indicar error.

Tabla III.16.

Casos de Uso	Registrar pago de matricula.
Actores	Secretario Docente (iniciador), Operador del sistema.
Propósito	Capturar una matricula y su correspondiente pago.
Tipo	Primario y esencial.
Referencias Cruzadas	Funciones: R1.12, R1.13, R1.14, R1.15, R1.16, R1.33, R1.34, R1.42, R1.43, R1.44. Casos de uso: Control de Acceso, Registrar nuevo alumno.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el Secretario Docente entrega al operador del SIA la copia de la boleta de pago por derecho de matricula.
2. El operador verifica la copia de la boleta de pago: Si la boleta tiene adjunto el expediente de un nuevo alumno, iniciar **Registrar nuevo alumno**. De lo contrario, ingresa el código del alumno.
3. mina el nombre, apellidos, carrera y ciclo al que pertenece el alumno. Muestra el costo de la matricula.
4. El operador ingresa el número de la boleta, fecha y el pago respectivo.
5. Actualiza y muestra el saldo de la matricula.
6. Al terminar de registrar el pago, el operador indica al SIA capturar la matricula efectuada.
7. Registra la matricula.
8. El operador retorna la boleta de pago al

Secretario
Docente.

Cursos alternos:

- Línea 2: El código es incorrecto. Se indica el error y se cancela la operación.
- Línea 4: Número de boleta ya existe. Se indica el error y se cancela la operación.

CAPITULO III: MODELADO DEL SISTEMA

Pago es mayor que el costo de matricula. Desplegar mensaje: Monto excedido.

Caso de uso relacionado: Registrar Nuevo Alumno CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el operador al verificar la copia de la boleta de pago por derecho de matricula tiene adjunto el expediente de un nuevo alumno.	
2. El operador escoge la opción nuevo alumno. Y luego registra los datos del nuevo alumno.	3. Muestra ventana que permite ingreso de datos del alumno. Genera el código del nuevo alumno y lo muestra. Captura los datos del nuevo alumno.
4. El operador indica al SIA retornar a la pantalla anterior y prosigue.	5. Muestra pantalla anterior.

Tabla III.17.

Casos de Uso	Registrar Inscripción de Curso.
Actores	Secretario Docente (iniciador), Operador del sistema.
Propósito	Capturar las asignaturas en las que se ha inscrito un alumno.
Tipo	Primario y esencial.
Referencias Cruzadas	Funciones: R1.17, R1.18, R1.19, R1.20, R1.21, R1.22, R1.23, R1.24, R1.25, R1.36, R1.37, R1.38, R1.39, R1.40, R1.41. Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el Secretario Docente entrega al Operador del SIA la ficha de matricula con las asignaturas en las que se ha inscrito un alumno.	
2. El operador ingresa el código del alumno.	3. Determina y muestra el nombre, carrera y ciclo correspondiente al alumno. Muestra una lista de las posibles asignaturas programadas para la especialidad y ciclo en que el alumno se ha matriculado.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

- Determina y muestra las asignaturas de cargo y/o repitencia que el alumno tuviere.
4. El operador verifica las secciones de las asignaturas:
Por cada asignatura que pertenece al semestre, el operador selecciona la asignatura respectiva.
5. Determina y muestra el código, número de horas de la asignatura.
Muestra el nombre del docente que estará a cargo de la asignatura.
6. Al terminar de registrar la asignatura, el operador indica al SIA que añada el curso a la inscripción de cursos.
7. Captura la inscripción de la asignatura.
Muestra el total de horas que hasta ese momento el alumno lleva.
8. Al terminar de registrar los cursos, el operador remite la copia de la ficha de matricula al Secretario Docente.

Cursos Alternos:

- Línea 2: El código es incorrecto. Desplegar error y cancelar la operación.

Tabla III.18.

Casos de Uso	Registrar Nuevo Docente.
Actores	Director (iniciador), Operador del Sistema.
Propósito	Capturar la información del nuevo docente.
Tipo	Secundario y esencial.
Referencias	Funciones: R1.45.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el Director autoriza y firma la Ficha de Datos del nuevo docente, para posteriormente enviarla al operador del sistema.
2. El operador registra la información del docente.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

3. Al terminar el operador indica al SIA que ha concluido el ingreso de datos.
4. Registra el nuevo docente.
5. El operador retorna la ficha de datos del docente.

Tabla III.19.

Casos de Uso	Control Acceso.
Actores	Operador del Sistema (iniciador).
Propósito	Capturar la clave y contraseña para iniciar el sistema.
Tipo	Primario y esencial.
Referencias Cruzadas	Funciones: R1.1.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el operador del sistema enciende el equipo de cómputo para utilizar el sistema.
2. El operador hace clic en el icono del Sistema de Información Académica.
3. Presenta el formulario Acceso al Sistema.
4. El operador ingresa la clave y contraseña de usuario.
5. Si clave y contraseña son correctas ingresa al Menú Principal.
De lo contrario muestra un mensaje de error "Ingrese clave y contraseña correcta".
6. El operador tiene la disponibilidad de usar el sistema para la operación que quiera ejecutar.

Cursos Alternos:

- Línea 4: Error clave y contraseña no son válidas. Indicar error y cancelar operación.

3.1.3. Crear Diagramas de casos de Uso.

La figura III.1 describe el Sistema de Información Académico del ISTMYE, el sistema tiene cuatro actores: Jefe del DFGP, Secretario Docente, Director y Operador del Sistema (Asistente del Secretario Docente). Algunos de ellos no interactúan directamente con el sistema, pero se les incluye para darle mayor claridad a la descripción de su funcionalidad (este criterio se ha basado en el método de desarrollo orientado a objetos que propone Craig Larman en su publicación “UML Y PATRONES”; a su vez Xavier Ferré Grau y María Isabel Sánchez Segura de la U.P.M en su publicación: “Desarrollo Orientado a Objetos con UML”).

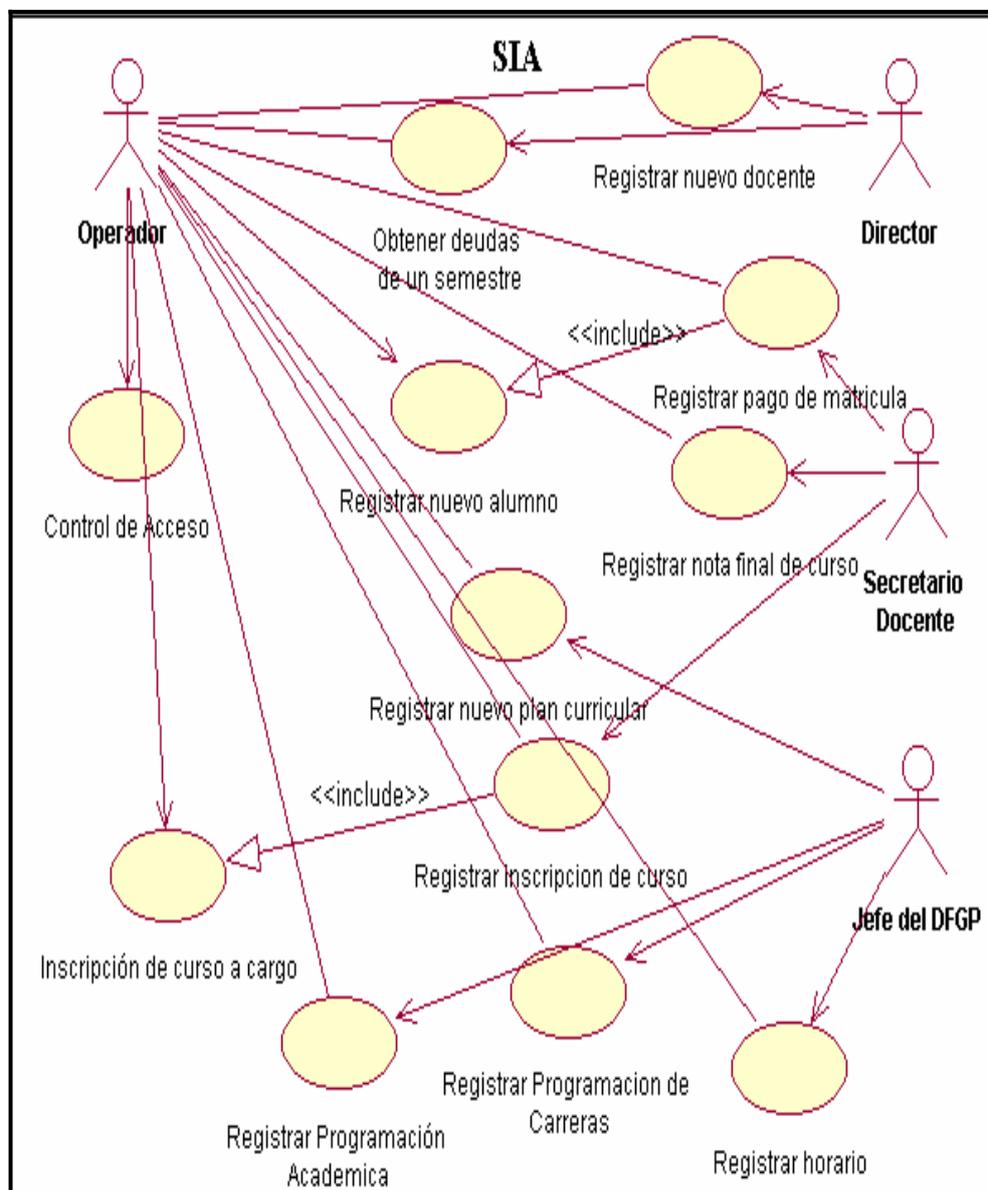


Fig. III.1: Diagrama de Casos de Uso del Sistema

3.1.4. Crear Modelo Conceptual

El objetivo de la creación de un Modelo Conceptual es aumentar la comprensión del problema. En UML este modelo, se ilustra con un grupo de diagramas de estructura estática, donde no se define ninguna operación.

En esta representación es mejor crear un modelo con muchos conceptos que quedarse corto y olvidar algún concepto importante.

Para identificar conceptos hay que basarse en la especificación de requisitos y en el conocimiento general acerca del dominio del problema. Para crear el Modelo Conceptual se siguen los siguientes pasos:

1. Hacer una lista de conceptos idóneos (ver Tabla III.22).
2. Representarlos en un diagrama.
3. Añadir las asociaciones necesarias para ilustrar las relaciones entre conceptos que es necesario conocer.
4. Añadir los atributos necesarios para contener toda la información que se necesite conocer de cada concepto.

Tabla III.22. Lista de conceptos.

<ul style="list-style-type: none">• SIA• Alumno• Matricula• DocumentodePago• PlanCurricular	<ul style="list-style-type: none">• IST• Docente• EspecificacióndeMatricula• InscripcióndeCurso• NotaPromocional
<ul style="list-style-type: none">• EspecificacióndePlan• Asignatura• CursoElemental• Actividad• ProgramacióndeCarrera• Turno• BloquedeHoras• TUPA• Persona	<ul style="list-style-type: none">• Aula• AsignaciónAcadémica• AsignacióndeCursos• Horario• Cargo• EspecificacióndeCargo• Rubro• Carrera• Operador

3.1.5. Diagramas de Secuencia.

Los diagramas de secuencia reflejan el comportamiento del sistema, porque dan una descripción gráfica de las interacciones del actor y de las operaciones a que dan origen. A continuación se identifican los eventos para los *casos de uso del sistema*:

Registrar Programación de Carreras, los eventos son los siguientes:

- especificarCarrera (nombre, sección, turno)
- asignarAula(nombreAula)

Registrar Asignación Académica, los eventos son los siguientes:

- especificarCarrera (nombre, sección, turno)
- seleccionarCurso(nombre)
- asignarDocente (nombre de docente)

Registrar horario, los eventos son los siguientes:

- especificarCarrera (nombre, sección, turno)
- seleccionarCurso(nombre)
- agregarHorario(día, hora)
- verReporte()

Obtener deudas de un semestre, los eventos son los siguientes:

- verDeudores(año, semestre)
- imprimir()

Registrar pago de matrícula, los eventos son los siguientes:

- ingresarAlumno(código, condición)
- capturarBoleta(número, fecha, monto)

Registrar inscripción de curso, los eventos son los siguientes:

- ingresarAlumno(código)
- agregarCurso(nombre)

CAPITULO III: MODELADO DEL SISTEMA

Registrar nota promocional, los eventos son los siguientes:

- especificarCarrera (nombre, sección, turno)
- mostrarCursos()
- seleccionarCurso(nombre)
- capturarEvaluacion(CUA, nota)

Registrar nuevo docente, los eventos son los siguientes:

- introducirDocente(Nombre, Apellidos, Dirección, codigoModular, códigoPlanillas, otros.)
- grabarDatos()

Registrar nuevo plan curricular, los eventos son los siguientes:

- ingresarCarrera(nombreCarrera, fechaCreacion, nroResolucion, semVigencia, horasTotal)
- agregarCurso(nombreCurso, sección)
- verReporte()

Control de Acceso, los eventos son los siguientes:

- ingresarIdentificacion(clave, contraseña)
- solicitarAcceso()

Una vez identificados los eventos se muestran, en el Anexo I los diagramas de secuencias para cada caso de uso

3.1.6. Definir Contratos.

Los **contratos** son documentos útiles que describen el comportamiento del sistema a partir de cómo cambia el estado del sistema cuando se llama una operación suya, es decir, un contrato describe los cambios del estado del sistema total cuando se llama a sus operaciones. A partir de los diagramas de secuencia se pueden identificar las operaciones, es decir, por cada evento existe una operación.

A continuación se describe algunas secciones de un contrato. No todas las secciones son necesarias, pero se recomienda las de Responsabilidades y Postcondiciones.

CONTRATOS DE OPERACIONES DEL SISTEMA

Contratos para el caso de uso: Registrar Programación de Carreras

Contrato

Nombre seleccionarCarrera(carrera: texto, turno: texto, sección: texto)

Responsabilidades Especificar la carrera, turno y sección de una determinada carrera en un semestre.

Tipo Sistema.

Referencias Funciones del Sistema: R1.2, R1.3, R1.4, R1.5.

Cruzadas Casos de uso: Control acceso, Registrar aula.

Notas Utilice el acceso superrápido a la base de datos.

Precondiciones El sistema conoce la carrera profesional.

El sistema conoce la sección y el turno.

Postcondiciones

- Si se trata de una nueva programación de carreras, una programación de carrera fue creada (creación de instancias).
- *ProgramaciondeCarreras* fue asociada a *Carrera*.
- *ProgramaciondeCarreras* fue asociada a *Turno*.
- Se estableció *ProgramaciondeCarreras.seccion* con el valor de la sección (modificación de atributo).

Contrato

Nombre asignarAula(nombreAula: texto).

Responsabilidades Asignar un aula a una carrera programada en un semestre.

Tipo Sistema.

Referencias Funciones del sistema: R1.

Cruzadas Casos de uso: Control acceso.

Notas Utilice el acceso superrápido a la base de datos.

Precondiciones El Sistema conoce el aula.

Postcondiciones • *ProgramaciondeCarreras* fue asociada a *Aula*.

Contratos para el caso de uso: Registrar Asignación Académica

Contrato

Nombre especificarCarrera(nombre: texto, turno: texto, sección: texto).

Responsabilidades Mostrar una lista de los posibles cursos que serían programados para la carrera – sección.

Tipo Sistema.

Referencias Funciones del Sistema: R1.6, R1.7, R1.8, R1.9,

Cruzadas R1.10, R1.11.

Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Si la carrera no aparece en la lista, verificar la relación de carreras programadas en un semestre.

Precondiciones El Sistema conoce el plan de estudios de la carrera.

Postcondiciones

- Si es una nueva programación académica, se crea *AsignacionDeCurso* (creación de instancia).
- *AsignacionDeCurso* se ha asociado con *PlanCurricular*.

Contrato

Nombre seleccionarCurso (nombre: texto).

Responsabilidades Capturar información del curso programado.

Tipo Sistema.

Referencias Funciones del Sistema: R1.6, R1.7, R1.8, R1.9,

Cruzadas R1.10, R1.11.

Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Si el curso no aparece en la lista, verificar plan curricular de la carrera.

Precondiciones El Sistema conoce las asignaturas que una carrera debe llevar.

Postcondiciones

- *PlanCurricular* se ha asociado a la instancia *Asignatura*.

Contrato

Nombre asignarDocente(nombre: texto).

Responsabilidades Capturar información del docente encargado del curso.

Tipo Sistema.

Referencias Funciones del Sistema: R1.6, R1.7, R1.8, R1.9,

Cruzadas R1.10, R1.11.

Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Si el nombre del docente no se encuentra en la búsqueda, verificar lista de docentes. Cancelar operación.

Precondiciones El Sistema conoce información sobre el docente.

Postcondiciones

- *AsignacionDeCurso* se ha asociado a la instancia *Docente*.
- Se estableció *AsignacionDeCurso.turno* con el valor del turno (modificación de atributo).

Contratos para el caso de uso: Registrar horario

Contrato

Nombre especificarCarrera(nombre: texto, turno: texto, sección: texto)

Responsabilidades Mostrar la lista cursos de cursos programados a la carrera en un semestre.

Tipo Sistema.

Referencias Funciones del Sistema: R1.54, R1.55.

Cruzadas Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Precondiciones El Sistema conoce las asignaturas programas para una carrera.

Postcondiciones

- Si es un nuevo horario, se crea *Horario* (creación de instancia).
- La instancia *Horario* se ha asociado a *AsignacionDeCurso*, basado esto en la correspondencia que existe entre las instancias.

Contrato

Nombre seleccionarCurso (nombre: texto).

Responsabilidades Capturar información del curso programado.
Mostrar el encargado del curso y el aula donde se dictará el curso.

Tipo Sistema.

Referencias Funciones del Sistema: R1.54, R1.55.

Cruzadas Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Precondiciones El sistema conoce las asignaturas programadas a una carrera en un semestre.

Poscondiciones

- La instancia *AsignacionDeCurso* se ha asociado a la instancia *Asignatura*.

Contrato

Nombre agregarHorario (día: texto, hora: texto).

Responsabilidades Registrar el horario de un curso programado en un semestre.

Tipo Sistema.

Referencias Funciones del Sistema: R1.54, R1.55.

Cruzadas Casos de uso: Control acceso

Notas El sistema conoce el bloque de horas de un turno.

Poscondiciones

- Se asignó a la instancia *Horario.día* un valor.
- Se asignó a la instancia *Horario.hora* un.
- Se asocia la instancia *Horario* a *BloquedeHoras*.

Contratos para el caso de uso: Obtener deudas de un semestre

Contrato

Nombre verDeudores(año: número, semestre: texto)

Responsabilidades Consultar los deudores por concepto de matricula de un año y semestre.

Tipo Sistema.

Referencias Funciones del Sistema: R1.33, R1.34.

Cruzadas Casos de uso: Control acceso.

Precondiciones El sistema conoce las matriculas de un semestre.

Poscondiciones

- Se asocia *Matricula* a la instancia *DocumnetoDePago*.
- Se asocia *Matricula* a la instancia *Alumno*.
- Se realiza una consulta a *DocumnetoDePago*.

Contrato

Nombre imprimir Reporte().

Responsabilidades Imprimir un reporte de acuerdo a la opción.

Tipo Sistema.

Referencias Funciones del Sistema: R1.33, R1.34.

Cruzadas Casos de uso: Control acceso

Precondiciones Especificar año y semestre.

Contratos para el caso de uso: Registrar inscripción de curso

Contrato

Nombre ingresarAlumno(código: número)

Responsabilidades Ingresar el código de un alumno matriculado.

Muestra la carrera, sección y turno de un alumno matriculado en un semestre.

Tipo Sistema.

Referencias Funciones del sistema: R1.17, R1.18, R1.19, R1.20,

Cruzadas R1.21, R1.22, R1.23, R1.24, R1.25, R1.36, R1.37, R1.38, R1.39, R1.40, R1.41.

Casos de uso: Control acceso.

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Si el código no es válido, indique error.

Precondiciones El sistema conoce el código del alumno.

Postcondiciones

- Se creó una instancia *IncripciondeCurso*.
- Se asocio *IncripciondeCurso* a *Matricula*.
- La instancia *Matricula* fue asociada a *EspecificaciondeEstudio* y esta a *Alumno*, basado esto en el código único del alumno.

Contrato

Nombre agregarCurso(nombre: texto).

Responsabilidades Registrar la inscripción de un curso y añadirlo al total de horas llevadas por un alumno en un semestre.

Muestra el código, tipo, número de horas y docente encargado del curso.

Tipo Sistema.

Referencias Funciones del sistema: R1.21

Cruzadas Casos de uso: Control Acceso.

Notas Utilice el acceso superrápido de la base de datos.

Precondiciones El sistema conoce el curso.

Postcondiciones

CAPITULO III: MODELADO DEL SISTEMA

- La instancia *IncripciondeCurso* fue asociada a *EspecificacióndePlan*, basado esto en el nombre del curso.

Contratos para el caso de uso: Registrar nota promocional de asignatura

Contrato

Nombre especificarCarrera(nombre: texto, turno: texto, sección: texto)

Responsabilidades Ingresar el nombre de la carrera, turno y sección para especificar los datos de la asignatura a buscar. Mostrar una lista de asignaturas programadas para la carrera y ciclo.

Tipo Sistema.

Referencias Funciones del sistema: R1. 26, R1.27, R1.28,

Cruzadas R1.29, R1.30, R1.31, R1.32.

Caso de Uso: Control Acceso

Nota Utilice el acceso superrápido a la base de datos.

Excepciones Si el nombre de la especialidad no se encuentra en lista de carreras programadas, indique que se cometió un error.

Precondiciones El sistema conoce el nombre de la carrera profesional.

Postcondiciones

- Se asocia *Carrera* a *ProgramacionDeCarrera*, basado esto en el nombre de la especialidad.

Contrato

Nombre mostrarCursos()

Responsabilidades Desplegar una lista de todas las asignaturas de un alumno matriculado en una determinada carrera, sección y turno.

Tipo Sistema.

Referencias Funciones del sistema: R1. 26, R1.27, R1.28,

Cruzadas R1.29, R1.30, R1.31, R1.32.

Caso de Uso: Control Acceso

CAPITULO III: MODELADO DEL SISTEMA

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Carrera profesional no tiene programada una carga académica.

Postcondiciones

- Se asocia *AsignacionDeCurso* a *SIA*.

Contrato

Nombre seleccionarCurso(nombre: texto)

Responsabilidades Registrar asignatura. Mostrar datos de la asignatura asignado a un docente en un semestre.

Tipo Sistema.

Referencias Funciones del sistema: R1. 26, R1.27, R1.28, R1.29,

Cruzadas R1.30, R1.31, R1.32.

Caso de Uso: Control Acceso

Notas Utilice el acceso superrápido a la base de datos.

Postcondiciones

- Se asocia *AsignacionDeCurso* a *PlanCurricular*.
- La instancia *PlanCurricular* se asocia a la *EspecificacionDePlan*.

Contrato

Nombre capturarEvaluación(CUA: número, nota: número).

Responsabilidades Registrar nota de un alumno inscrito en un curso.
Mostrar nombre del alumno.

Tipo Sistema.

Referencias Funciones del sistema: R1. 26, R1.27, R1.28, R1.29,

Cruzadas R1.30, R1.31, R1.32.

Caso de uso: Control acceso.

Precondiciones El sistema conoce el CUA.

Postcondiciones

- Si se trata de una nueva nota, una nota fue creada.
- Si se trata de una nueva nota, la *Nota* fue asociada a *InscripcióndeCurso*.
- La instancia *InscripcióndeCurso* se asocia a *Matricula*.

CAPITULO III: MODELADO DEL SISTEMA

- *Matricula* se asocia a *Alumno*, basado esto en el código del alumno.
- Se asignó a *Nota.nota* el valor de la nota.

Contratos para el caso de uso: Registrar nuevo plan curricular

Contrato

Nombre ingresarCarrera (nombreCarrera: texto, fecha: date, numResolucion: número, semVigencia: número, horasTotal: número).

Responsabilidades Ingresar los datos del plan curricular de una determinada carrera. Muestra el número del plan.

Tipo Sistema.

Referencias Funciones del Sistema: R1.47, R1.48, R1.49, R1.50.

Cruzadas Casos de uso: Control acceso

Precondiciones El sistema conoce el código único de especialidad.

Postcondiciones

- La instancia *PlanCurricular* se asoció con *Carrera*.
- Se asignó a *PlanCurricular.numPlan* el valor del número del plan.
- Se asignó a *PlanCurricular.numResolución* el valor del número de resolución.
- Se asignó a *PlanCurricular.fechaCreación* el valor del N del plan.
- Se asignó a *PlanCurricular.numHrasTotal* el valor del N total de horas.
- Se asignó a *PlanCurricular.HrasPractProf* el valor del N total de horas.
- Se asignó a *PlanCurricular.semVigencia* el valor del año y semestre en que empezará a regir.

Contrato

Nombre agregarCurso(nombre: texto, sección: texto, númeroHoras: número)

Responsabilidades Registrar un curso en un plan curricular que pertenece.

Tipo Sistema.

Referencias Funciones del Sistema: R1.47, R1.48, R1.49, R1.50.

Cruzadas Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

CAPITULO III: MODELADO DEL SISTEMA

Excepciones Si el nombre del curso no es válido indique que se cometió un error.

Precondiciones El sistema conoce el nombre del curso.

Postcondiciones

- Si se trata de un nuevo plan curricular, un *PlanCurricular* fue creado.
- Si se trata de una nuevo plan curricular, el nuevo *Plan* fue asociado al *SIA*.
- Se creó una instancia *EspecificacióndePlan*.
- La instancia *EspecificacióndePlan* se asocio a *PlanCurricular*.
- Se asocio *EspecificacióndePlan* a *Asignatura*, basado esto en el nombre del curso.
- Se asignó a *EspecificacióndePlan.sección* el valor de la sección.
- Se asignó a *EspecificacióndePlan.nroHoras* el valor del N de horas.

Contrato

Nombre grabarDocente().

Responsabilidades Capturar los datos de un nuevo docente.

Tipo Sistema.

Referencias Casos de uso: Control acceso

Cruzadas

Postcondiciones

- Si se trata de un nuevo docente, un *Docente* fue creado.
- Si se trata de un nuevo docente, el nuevo *Docente* fue asociado al *SIA*.

Contratos para el caso de uso: Control de Acceso

Contrato

Nombre ingresarIdentificacion(Clave:código, Contraseña: texto).

Responsabilidades Ingresar el código del operador y leer su correspondiente contraseña.

Tipo Sistema.

Referencias Funciones del Sistema: R1.1.

Cruzadas Casos de uso: Control acceso

Postcondiciones

- Leer el valor de Operador.código.
- Leer el valor de Operador.contraseña.

Contrato

Nombre aceptarUsuario ().

Responsabilidades Permitir inicialización del sistema.

Tipo Sistema.

Referencias Funciones del Sistema: R1.1.

Cruzadas Casos de uso: Control acceso

Notas Utilice el acceso superrápido a la base de datos.

Excepciones Si código y contraseña son incorrectos, indique que se cometió un error.

Precondiciones El sistema conoce el código y contraseña del operador.

Postcondiciones

- Se creó una instancia IST, SIA, PlanCurricular, EspecificaciondePlan, TUPA, Curso, CursoElemental, CursoActividad, DetalledePago, Alumno, Docente, Operador, InscripcióndeCurso, Carrera, Turno, etc. (creación de instancias).
- Se asoció *PlanCurricular* a *EspecificaciondePlan*.
- Se asoció *EspecificaciondePlan* a *Curso*.
- Se asoció *PlanCurricular* a *Carrera*.
- Se asoció *PlanCurricular* a SIA.
- Se asoció *TUPA* a *Matricula*.
- Se asoció *Matricula* a SIA.
- Se asoció *Matricula* a InscripcióndeCurso.
- Se asoció *ProgramaciónAcadémica* a SIA.
- Se asoció *Operador* a SIA.
- Se asoció *Docente* a SIA.
- Se asoció Alumno a SIA.
- Se asoció Alumno a EspecificaciondeEstudio.

3.1.7. Diagramas de Estados

Este tipo de diagramas muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida.

En cuanto a la representación, un diagrama de estados es un gráfico cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos. Un estado se representa mediante una caja redondeada con el nombre del estado en el interior y una transición se representa mediante una flecha desde el estado origen al estado destino.

Un conjunto de diagramas de estado para los casos de uso, permitirá al diseñador desarrollar metódicamente un diseño que le garantice el orden correcto de los eventos del sistema. A continuación se muestran los diagramas de estado de algunos casos de uso del sistema:

Registrar Programación de Carreras

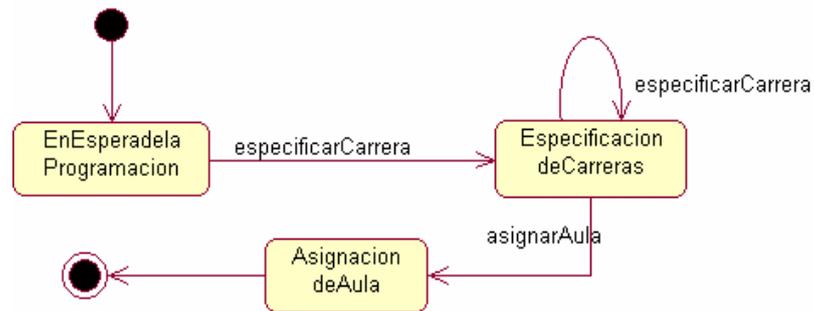


Figura III.14. Diagrama de Estados: Registrar programación de carreras.

Registrar Asignación Académica

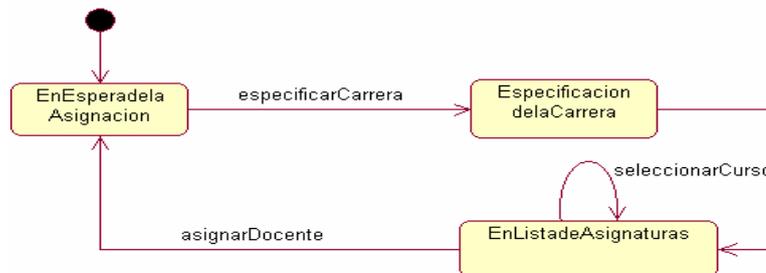


Figura III.15. Diagrama de Estados: Registrar asignación académica.

Registrar Pago de Matricula

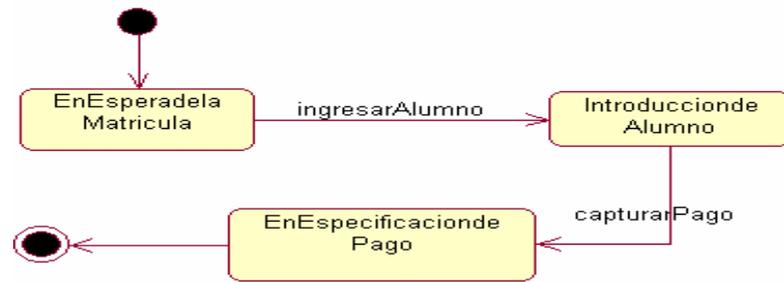


Figura III.17. Diagrama de Estados: Registrar pago de matricula.

Registrar Inscripcion de Asignaturas



Figura III.18. Diagrama de Estados: Registrar inscripción de asignaturas.

Registrar Nota Promocional De Asinatura



Figura III.19. Diagrama de Estado: Registrar nota promocional de asignatura.

CAPITULO III: MODELADO DEL SISTEMA

Además, a continuación se ilustran algunos de los diagramas de estados que representan el comportamiento de algunos de los objetos del sistema.

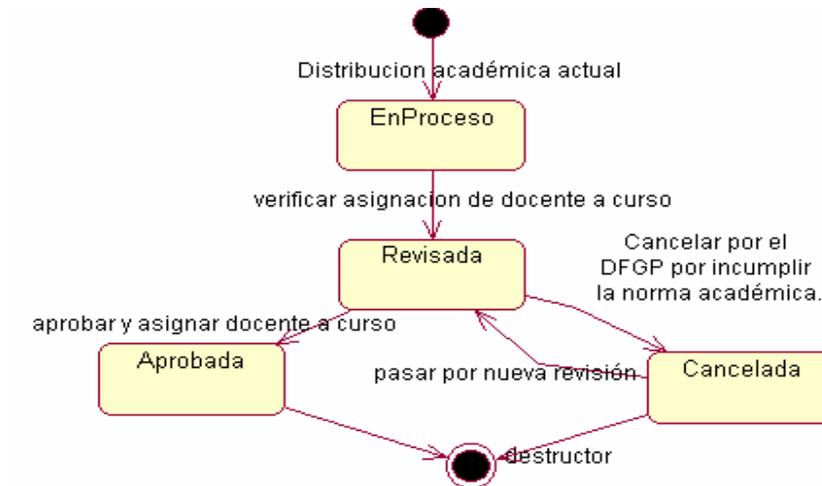


Figura III.21: Diagrama de estados del objeto Asignación Académica

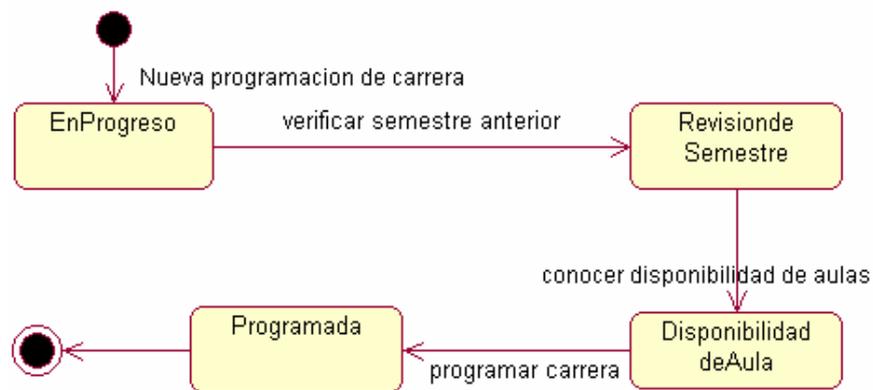


Figura III.20: Diagrama de estados del objeto Programación de Carrera

CAPITULO III: MODELADO DEL SISTEMA

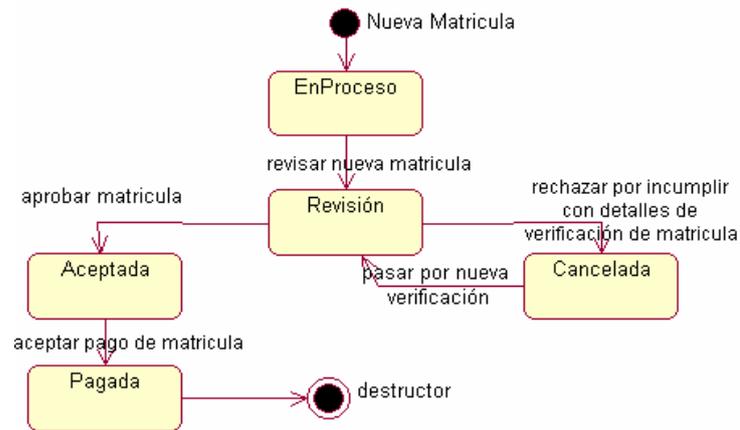


Figura III.23: Diagrama de estados del objeto Matricula

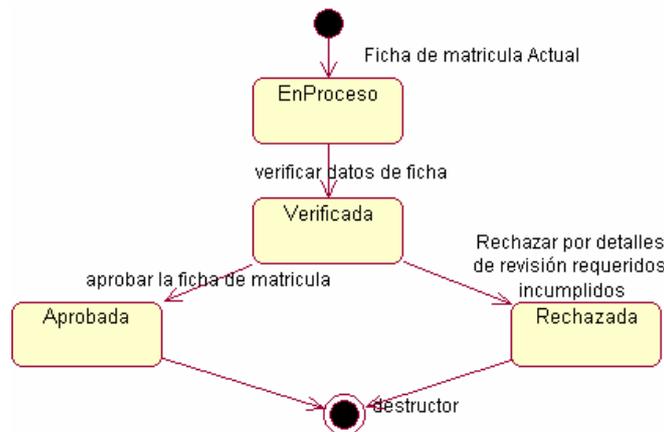


Figura III.24: Diagrama de estados del objeto Inscripción de Curso

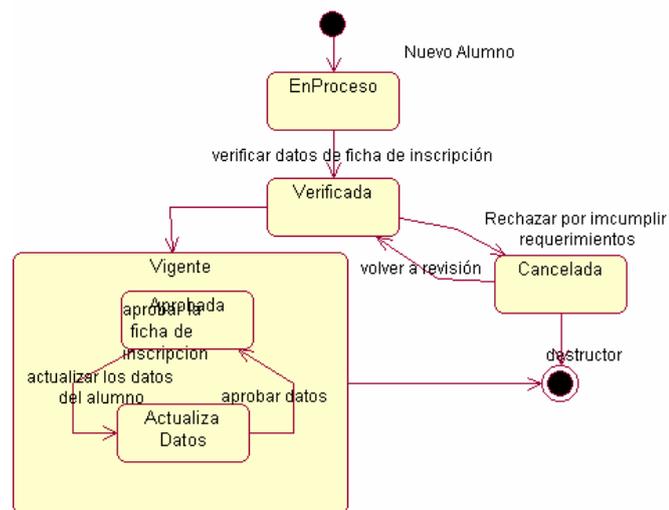


Figura III.25: Diagrama de estados del objeto Alumno

3.2. DISEÑO

3.2.1. Casos de Uso Reales

Un caso de uso real describe concretamente el proceso en términos del diseño real, de la solución específica que se va a llevar a cabo. Se ajusta a un tipo de interfaz específica, así como se detalla a partir de una tecnología particular de entrada y salida, su implementación global.

A continuación se muestran los casos de uso real:

Tabla III.23.

Casos de Uso: Registrar Programación de Carreras.	
Actores	Jefe del DFGP (iniciador), Operador del sistema.
Propósito	Capturar la programación de carreras y el aula asignada a una carrera en un semestre.
Resumen	El Jefe del DFGP envía al operador el cuadro de carreras programadas en un semestre. El operador registra cada una de las carreras programadas en el semestre y el aula que le ha sido asignada. Al terminar la transacción, el operador envía un reporte al Jefe del DFGP.
Tipo	Primario y real.
Referencias	Funciones: R1.2, R1.3, R1.4, R1.5.
Cruzadas	Casos de uso: Control de Acceso, Registrar aula.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

- | | |
|---|--|
| 1. Este caso de uso comienza cuando el Jefe del DFGP envía al operador del SIA la programación de carreras del semestre actual. El operador elige del menú principal la opción Registrar Programación de Carrera. | 2. Presenta el formulario Registrar Programación de Carrera de la figura III.26. Muestra el semestre actual. |
| 3. Con cada programación de carrera, el operador selecciona el nombre de la carrera, sección y turno en A, B y C. Se selecciona el nombre del aula asignada | 4. Determina y muestra el código, estado y capacidad del aula en E, F y G. Agrega la información |

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

en D y se oprime el botón Aceptar después de capturar a cada programación de carrera.

Respuesta del Sistema

sobre el aula a la actual programación.

5. Al terminar de capturar la programación de carreras, el operador cierra el formulario.

Cursos Alternos:

- Línea 3: Programación de carrera, turno y sección no es válida. Desplegar mensaje informando que programación ya existe.



Figura III.26

Tabla III.24.

Casos de Uso	Registrar horario.
Actores	Jefe del DFGP (iniciador), Operador del sistema.
Propósito	Capturar la distribución de horas de clase de los cursos programados a una carrera.
Resumen	El Jefe del DFGP envía al operador el horario de distribución de horas de clase de una carrera en un semestre. El operador registra los horarios de cada uno de los cursos programados a una carrera. Al terminar la operación, el operador solicita un reporte de horarios por docente para enviarlo al Departamento de Formación General y Profesional.

CAPITULO III: MODELADO DEL SISTEMA

Tipo	Primario y real.
Referencias Cruzadas	Funciones: R1.54, R1.55.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el Jefe del DFGP envía al operador del SIA el horario de clases de una carrera. El operador elige del menú principal la opción Registrar Horario.
2. El menú principal muestra el formulario Registrar horario (figura III.28), que solicita los datos de una carrera.
Muestra el actual semestre.
3. El operador selecciona la carrera, turno y sección en A, B y C. Después se oprime el botón D.
4. Determina y muestra en E la lista de asignaturas programadas a la carrera de acuerdo a la sección.
5. Por cada asignatura, el operador selecciona el nombre de la asignatura en E y selecciona en J y K, el día y la hora. Después de registrar el horario oprime el botón L
6. Agrega la información del curso en la actual transacción. El código y número de horas de la signatura se muestran en F y G. El aula donde se dictará el curso se muestra en H, y el nombre del profesor se muestra en I.
7. Al terminar de capturar el horario de todos los cursos de una carrera, el operador cierra la ventana y escoge una de las opciones del módulo reporte según los requerimientos. El operador indica al SIA un informe de los horarios por docente.
8. Genera reportes.
9. El operador envía los reportes para ser entregados a cada uno de los interesados.

CAPITULO III: MODELADO DEL SISTEMA

Cursos Alternos:

- Línea 3: Lista de cursos no existe. Verificar que programación académica de carrera se haya efectuado.
- Línea 5: Día y hora de horario no son válidos. Desplegar mensaje informando que ese horario ya fue ocupado.

Figura III.28

Tabla III.25.

Casos de Uso	Obtener deudas de un semestre.
Actores	Director (iniciador), Operador del sistema.
Propósito	Capturar la información de deudores por concepto de matrícula de un determinado año y semestre.
Resumen	El Director solicita al operador un informe de todos los alumnos deudores por concepto de matrícula. El operador solicita al SIA un reporte de deudores del semestre. Al terminar la transacción, el operador envía a Dirección el informe.
Tipo	Primario y real.
Referencias	Funciones: R1.33, R1.34.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el operador solicita al SIA la relación de deudores por concepto de matricula de un año y semestre clasificados por carrera profesional. El operador elige del menú principal el reporte.	2. Genera un Informe Deudas de Matricula, por especialidad.
3. El operador solicita al SIA imprimir reporte.	4. Imprime reporte.
5. El operador envía el reporte para ser publicado.	

Tabla III.26.

Casos de Uso	Registrar pago de matricula.
Actores	Secretario Docente (iniciador), Operador del sistema.
Propósito	Capturar una matricula y su correspondiente pago.
Resumen	El Secretario Docente entrega al operador la copia de la boleta de pago por derecho de matricula. El operador registra el CUA (código único del alumno), el número, fecha y monto de la boleta. Al terminar la transacción la transacción, el operador remite la copia al Secretario Docente.
Tipo	Primario y real.
Referencias	Funciones: R1.12, R1.13, R1.14, R1.15, R1.16, R1.33, R1.34, R1.42, R1.43, R1.44.
Cruzadas	Casos de uso: Control de Acceso, Registrar nuevo alumno.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores	Respuesta del Sistema
1. Este caso de uso comienza cuando el Secretario Docente entrega al operador del SIA, copia del pago por derecho de matricula. El operador elige en el menú principal la opción registrar matricula.	2. Presenta el formulario Registrar Matricula de la figura III.29.
3. El operador oprime el botón para indicar que se trata de una	A 4. Agrega la información sobre el alumno a la actual transacción.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

nueva matricula. Por defecto en B escoge la opción Alumno Regular y tecllea el código del Alumno, pero de tratarse de un nuevo Alumno ver sección **Registrar nuevo alumno.**

Respuesta del Sistema

El nombre del alumno, la carrera profesional, turno y sección se muestran en D, E, F y G.

El costo de la matricula se muestra en M.

- 5. Al verificar la boleta de ingreso o solicitud del alumno, el operador escoge la condición de matricula y tecllea el número, fecha y pago de la boleta en I, J y K. Se oprime el botón Grabar.
- 6. Agrega la información sobre el pago de matricula a la actual transacción. Actualiza el saldo de matricula. El saldo de matricula se muestra en L.
- 7. Registrada la matricula, el operador retorna los documentos.

Cursos alternos

- Línea 3: código de alumno es incorrecto. Despliega mensaje informando que el código del alumno no existe.
- Línea 5: N de boleta es incorrecta. Se despliega mensaje informando que el número no es válido. Si el pago es mayor que el costo de matricula. Se despliega un mensaje informando que el monto no es válido.



Figura III.29

Tabla III.27.

Casos de Uso	Registrar Inscripción de Curso.
Actores	Secretario Docente (iniciador), Operador del sistema.
Propósito	Capturar las asignaturas en las que se ha inscrito un alumno.
Resumen	El Secretario Docente entrega al operador la copia de la ficha de matricula de un alumno. El operador registra los cursos en los que se ha inscrito un alumno. Al terminar la transacción, el operador retorna la copia de la ficha de matricula.
Tipo	Primario y real.
Referencias	Funciones: R1.17, R1.18, R1.19, R1.20, R1.21, R1.22,
Cruzadas	R1.23, R1.24, R1.25, R1.36, R1.37, R1.38, R1.39, R1.40, R1.41. Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el Secretario Docente entrega al Operador del SIA la copia de la ficha de matricula de un alumno con los cursos en los que se ha inscrito. El operador elige del menú principal la opción Inscripción de Curso.
2. Presenta el formulario Inscripción de Cursos de la figura III.30.
3. El operador teclea el código del alumno en A.
4. Agrega la información sobre el alumno a la actual inscripción. Determina y muestra en B, C, D y E, el nombre, especialidad, turno y sección del alumno matriculado. Muestra en F una lista de todas las asignaturas programadas a la carrera y sección donde el alumno se ha matriculado.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

Cursos Alternos

5. Por cada asignatura, el operador selecciona el nombre de la asignatura en F. Se oprime el botón Inscribir después de capturar cada asignatura.

6. Muestra la información sobre la asignatura a la actual inscripción. El código, tipo, número de horas y encargado del curso se muestran en G, H, I y J.

Calcula y presenta el total de horas inscritas en la pestaña Cursos Inscritos.

7. Al terminar de capturar los cursos, el operador retorna la copia de la ficha de la matrícula.

8. Genera un reporte de las asignaturas inscritas.

- Línea 3: El código del alumno es incorrecto. Desplegar mensaje informando que el código del alumno no es válido.

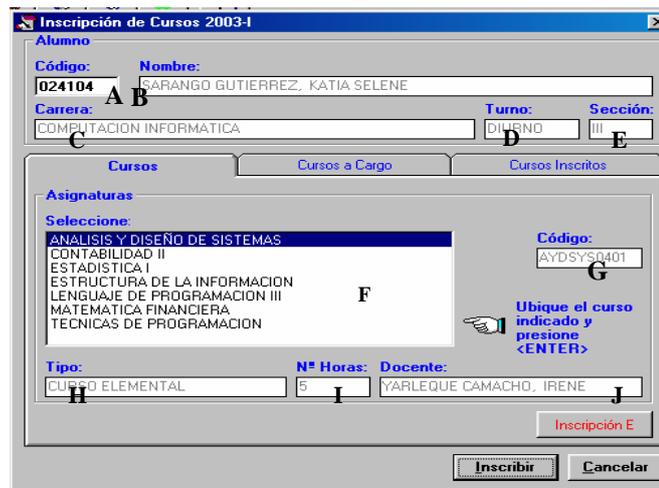


Figura III.30

Tabla III.28.

Casos de Uso	Registrar nota promocional de asignatura.
Actores	Secretario Docente (iniciador), Operador del sistema.
Propósito	Capturar las notas promocionales obtenidas por los alumnos inscritos en una asignatura.
Resumen	El Secretario Docente entrega al operador el acta de notas de una asignatura. El operador registra las notas de los

CAPITULO III: MODELADO DEL SISTEMA

	alumnos inscritos en una asignatura. Al terminar la transacción, el operador retorna el acta al Secretario Docente.
Tipo	Primario y real.
Referencias	Funciones: R1.26, R1.27, R1.28, R1.29, R1.30, R1.31, R1.32.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

1. Este caso de uso comienza cuando el Secretario Docente entrega al operador del SIA el acta con las notas obtenidas por los alumnos inscritos en un curso. El operador elige del menú principal la opción Registrar Nota.
2. Presenta el formulario de la figura III.31.
3. El operador selecciona la especialidad, turno y sección en A, B y C, luego presiona el botón D.
4. Determina y muestra en E una lista de todas asignaturas programadas a una carrera, turno y sección.
5. El operador selecciona el nombre de la asignatura en E y por cada alumno inscrito en la asignatura, el operador ingresa el código del alumno en H y en J el valor de la nota obtenida. Se oprime el botón Grabar después de capturar la evaluación del alumno.
6. Agrega la información del alumno a la actual transacción. Agrega la información del curso a la actual transacción. El código clave de la asignatura, así como el nombre del docente se muestran en F y G. El nombre del alumno se muestra en I. La observación de la evaluación se muestra en K.
7. Al terminar de capturar las notas de los alumnos inscritos en un curso, el operador remite el acta.
8. En la pestaña Relación de Notas se muestra el número de aprobados y desaprobados.

Cursos Alternos

CAPITULO III: MODELADO DEL SISTEMA

- Línea 6: La nota no es valida. Desplegar un mensaje informando que la nota no es válida.

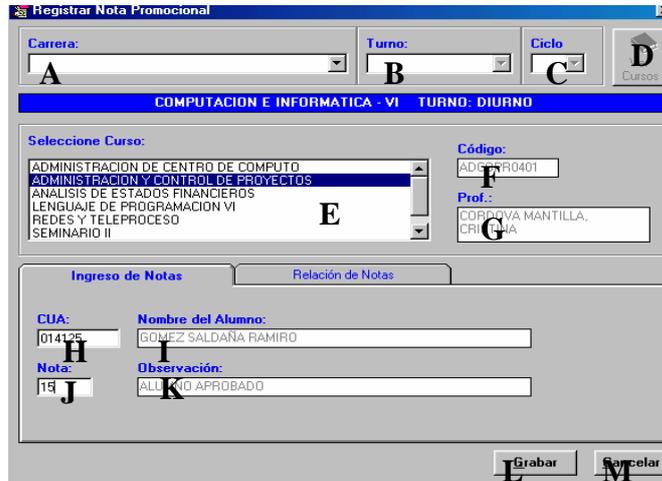


Figura III.31

Tabla III.29.

Casos de Uso	Control Acceso.
Actores	Operador del sistema (iniciador).
Propósito	Capturar la clave y contraseña para iniciar el sistema.
Resumen	El operador registra la clave y contraseña para inicializar el sistema.
Tipo	Primario y real.
Referencias	Funciones: R1.1.
Cruzadas	Casos de uso: Control de Acceso.

CURSO NORMAL DE LOS EVENTOS

Acción de los Actores

Respuesta del Sistema

- Este caso de uso comienza cuando el operador del sistema enciende el equipo de cómputo para utilizar el sistema. Y presiona el icono del SIA para utilizar el sistema.
- Presenta el formulario Control de Acceso (Figura III.32) Muestra en A el nombre de la institución académica. En B y C muestra el semestre actual.
- El operador teclea la clave y contraseña en C y D. Se oprime el botón Aceptar.
- Verifica y presenta el menú principal.
- El operador tiene la disponibilidad de usar el sistema.

Cursos Alternos:

- Línea 2: Clave y contraseña no válidas. Desplegar mensaje: clave y contraseña no son válidas.

Control de Acceso al SIA [Versión 1.0]

Para inicializar el sistema debe especificar su función y la contraseña que utilizará para iniciar la aplicación.

Institución: I.S.T. MANUEL YARLEQUE ESPINOZA - CATACAOS
A

Año Acad.: B 2003 - C II

Usuario: D_SG

Contraseña: E

Aceptar Cancelar

Figura III.32

3.2.2. Diagramas de Colaboración

Los diagramas de colaboración explican gráficamente la interacción entre las instancias del modelo (objetos). La creación de estos es una de las actividades más importantes en el desarrollo del sistema, pues contribuyen a decisiones clave en el funcionamiento del sistema. En los diagramas de secuencia se identificaron los eventos del sistema que a su vez generaron las operaciones del sistema, por lo que se construirá un diagrama de colaboración por cada operación. A continuación se muestran los principales diagramas de colaboración de los casos de uso:

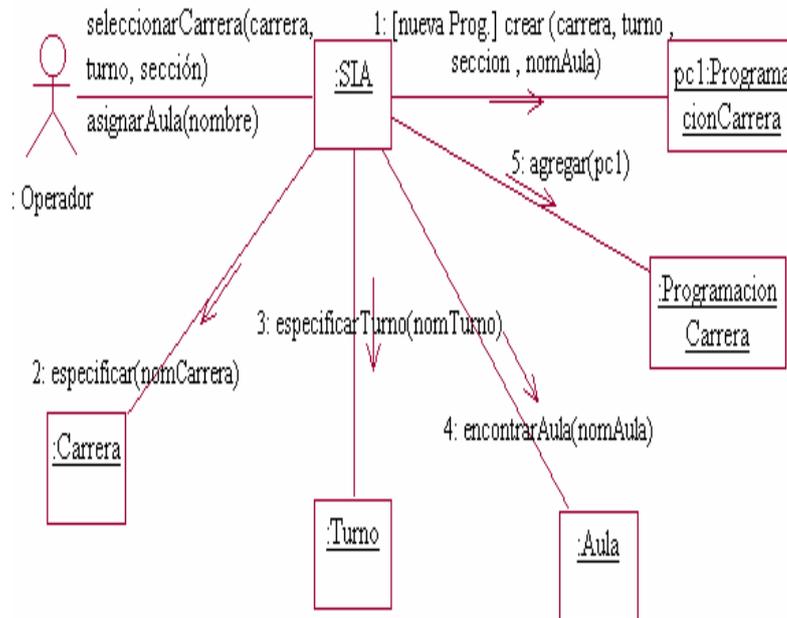


Figura III.33. Diagrama de Colaboración: Registrar Programación de Carreras

CAPITULO III: MODELADO DEL SISTEMA

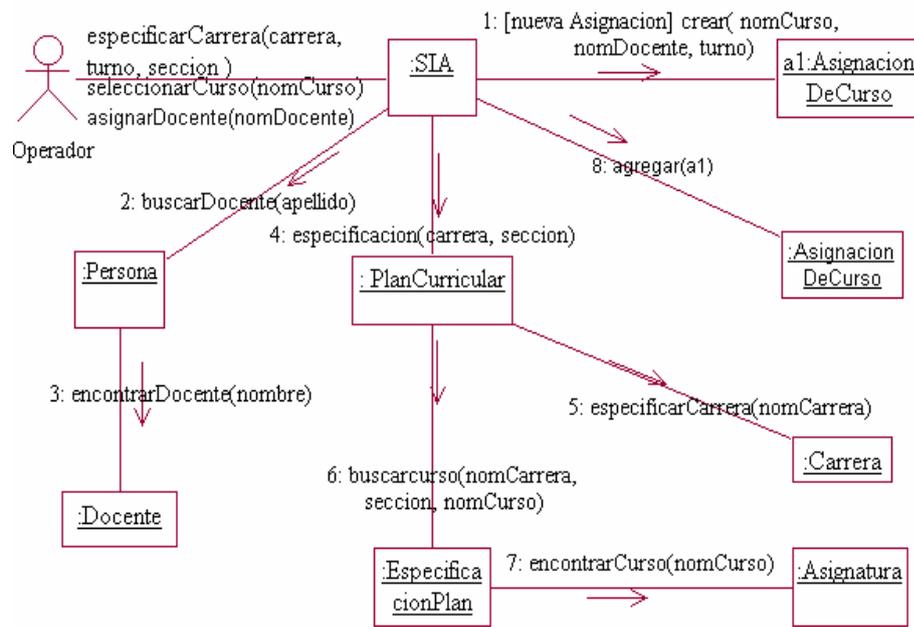


Figura III.34. Diagrama de Colaboración: Registrar asignación académica

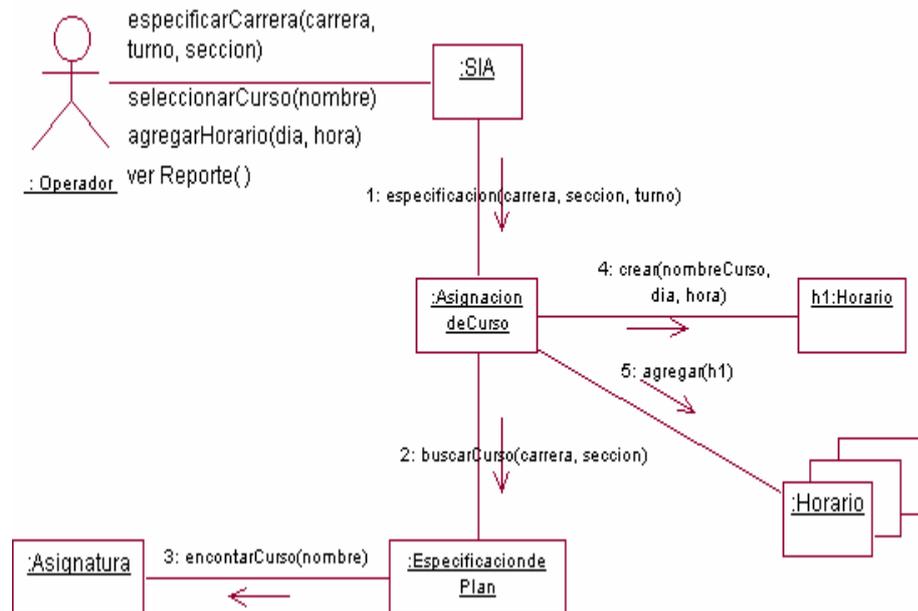


Figura III.35. Diagrama de Colaboración: Registrar Horario

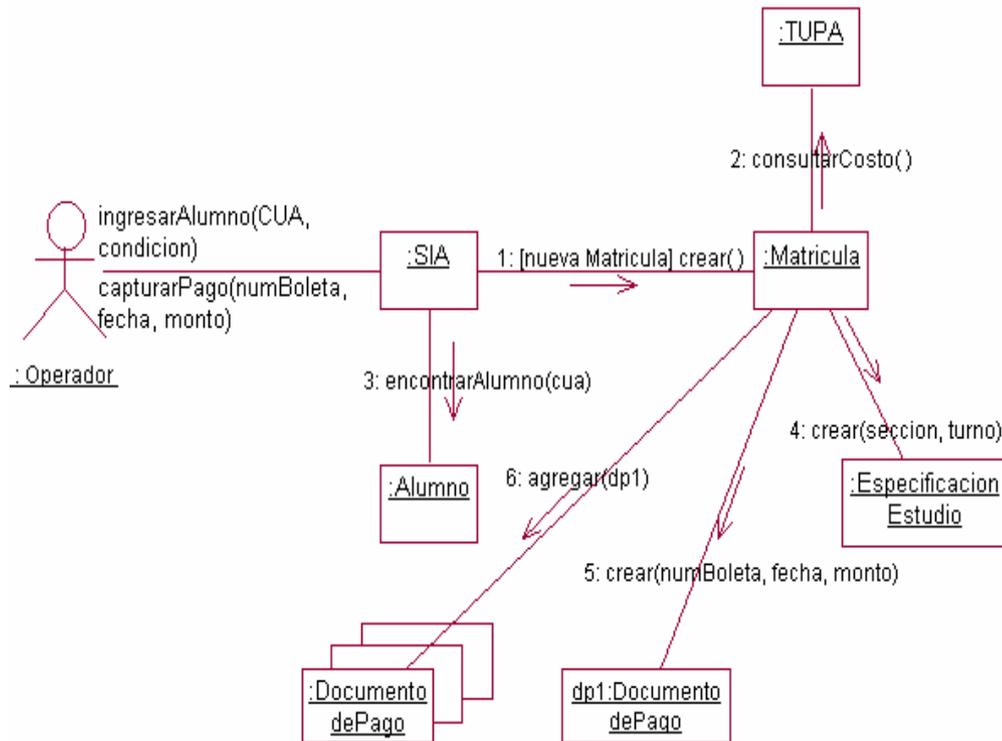


Figura III.36. Diagrama de Colaboración: Registrar pago de matricula

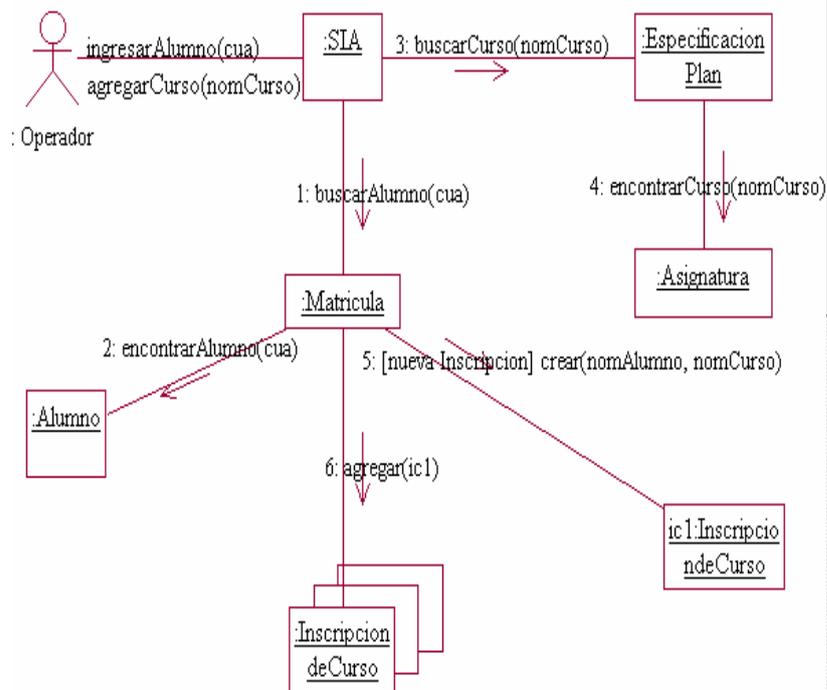


Figura III.37. Diagrama de Colaboración: Registrar inscripción de asignatura

3.2.3. Diagrama de Clases

El diagrama de clases expresa para el sistema computarizado la definición de clases como software, siendo este el diagrama principal de análisis y diseño para un sistema.

Durante el análisis del sistema se han diseñado diversos diagramas que han corroborado con información que ha ido complementándose con los diagramas desarrollados durante el diseño, hasta obtener un diagrama de clases que se especifica, con relaciones entre clases y estructuras de herencia. El Diagrama de Clases del sistema en desarrollo, se ilustran las clases y sus atributos provenientes de los conceptos asociados al modelo conceptual. La agregación de los métodos ha sido posible al analizar los mensajes de los diagramas de colaboración.

CAPITULO IV: DESARROLLO DEL SISTEMA

CAPITULO IV: DESARROLLO DEL SISTEMA

4.1. ARQUITECTURA DE TECNOLOGÍAS DE INFORMACIÓN

El desarrollo de un sistema de información es una tarea que toma tiempo conceptualizar, diseñar, programar, probar e implementar un sistema. En esta parte del proyecto analizaremos la arquitectura tecnológica más apropiada para el desarrollo del sistema, es decir, la tecnología cliente-servidor.

4.1.1. Justificación de la Utilización de la Arquitectura Cliente-Servidor

Se ha estado utilizando activamente el desarrollo de aplicaciones cliente servidor ya que pueden cubrir una gran cantidad de arquitecturas y asimismo sus componentes se pueden crear mediante numerosas herramientas independientes relacionadas con los lenguajes de programación. El utilizar una arquitectura cliente-servidor tiene sus ventajas como:

- Cuando un servidor de bases de datos procesa una consulta, la respuesta a esta petición dependerá del proceso del servidor y no del cliente.
- El proceso servidor activo devuelve sólo la información solicitada.
- Un proceso servidor activo puede asegurar más eficazmente la integridad de los datos.

Además, se ha considerado el uso de esta arquitectura por las siguientes características:

- El requerimiento de la institución de tener los datos almacenados en una ubicación central en donde los usuarios puedan acceder a ellos.
- El uso de Microsoft SQL Server como motor de base de datos relacional, cuyas características revolucionan el concepto de base de datos para la empresa.
- La institución cuenta con un personal muy reducido.

Todas estas características nos inicia en un proceso de desarrollo de la aplicación de tal forma que nos llevan a estudiar la arquitectura cliente-servidor.

CAPITULO IV: DESARROLLO DEL SISTEMA

4.1.2 Arquitectura Cliente-Servidor

Este término se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular.

Cualquier arquitectura cliente-servidor separa los lados del cliente y del servidor estableciéndose así las funciones que le corresponden al cliente y las funciones que le competen al servidor, por lo que esto nos conduce a especificar mejor la arquitectura a usar. A continuación mencionaremos las siguientes propuestas de arquitectura de capas que existen para los sistemas de información:

- **Arquitectura de dos capas**

La mayoría de las aplicaciones Cliente-Servidor funcionan bajo una arquitectura de dos capas en lenguajes de cuarta generación. Esta arquitectura consiste en dos casos típicos: 1) Cuando la base de datos se encuentra en el servidor conectado a una red, mientras que la aplicación, se encuentra en el proceso cliente y es quien se encarga de la ejecución de la lógica de la aplicación (ver Figura IV.1). 2) La lógica de la aplicación se codifica en el servidor como procedimientos almacenados (ver Figura IV.2).

CAPITULO IV: DESARROLLO DEL SISTEMA

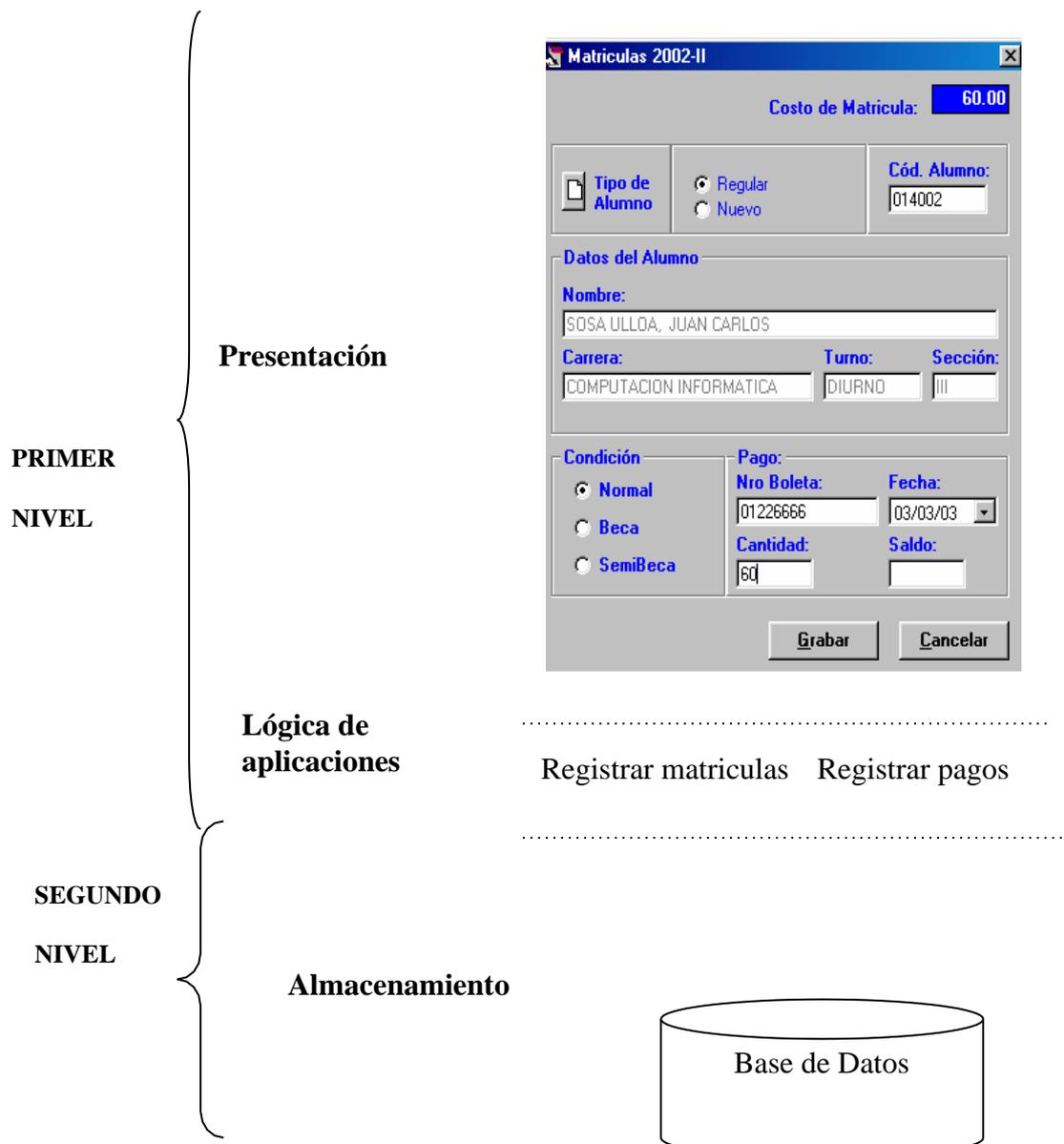


Figura IV.1. Arquitectura de Dos Capas – Caso 1.

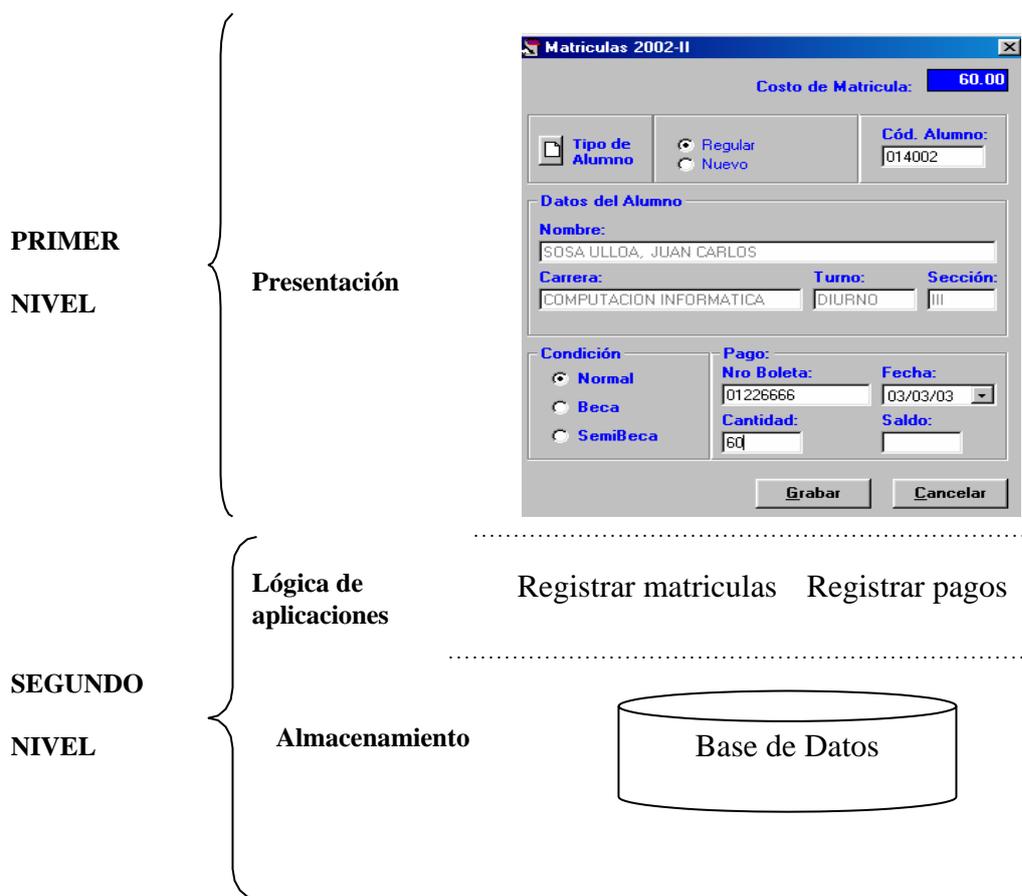


Figura IV.2. Arquitectura de Dos Capas – Caso 2.

- **Arquitectura de tres capas**

En una aplicación de tres capas (ver figura IV.3) cada capa trabaja en procesos separados, es decir, la capa de la lógica de aplicación se convierte en una capa intermedia bien definida y lógica de software.

CAPITULO IV: DESARROLLO DEL SISTEMA

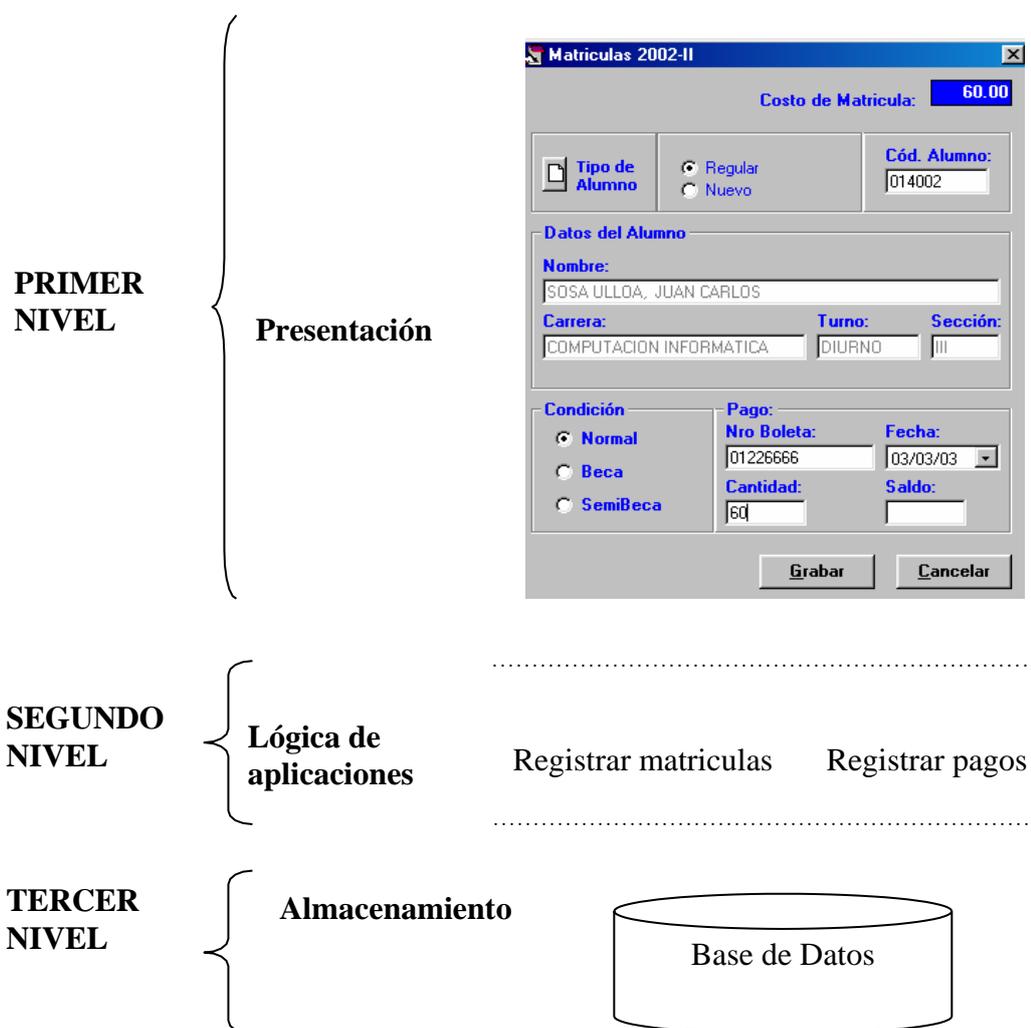


Figura IV.3. Arquitectura de tres capas.

Vistos los modelos de la arquitectura cliente-servidor y aunque es obvio que el modelo de dos capas contrasta con el modelo de tres capas. Indiscutiblemente el modelo de tres capas ofrece muchas ventajas frente al modelo de dos capas, pero hay que tener en cuenta que esto requiere pagar un precio, lo cual significa una mayor complejidad y administración. Por lo tanto desde nuestro punto de vista es mucho más ventajoso utilizar el modelo de dos capas caso 2, ya que de acuerdo a las necesidades y limitaciones de la institución se podrá decir que es el modelo que más conviene usar.

4.2. IMPLEMENTACION DEL DISEÑO DE BASE DE DATOS

4.2.1 Conversión del Diagrama de Clases a una Base de Datos Relacional

Los diseños orientados a objetos son eficientes, coherentes y menos proclives los problemas de actualización que aquejan a muchas otras técnicas de diseño de bases de datos. Pero, el modelo relacional ha ganado popularidad, por lo que se han incrementado sus ventajas en lo tocante a funcionalidad y flexibilidad. Además, a pesar de que las bases de datos orientadas a objetos tienen un aspecto prometedor, todavía no han alcanzado una amplia aceptación masiva por parte del mercado.

El diagrama de clases se puede usar para implementar un diseño de una base de datos relacional, pero para traducir un diagrama de clases a tablas ideales, hay que proporcionar detalles que faltan en el diagrama, como la clave primaria y los candidatos a clave para cada tabla; así como si un atributo puede ser o no ser nulo, además de asignarle a cada atributo un dominio.

Transformación del Diagrama de Clases en Modelos de Tablas

Esta sección se centra en la transformación de clases a tablas, donde se aplicará las reglas de correspondencia, porque el modelado de objetos no introduce nada nuevo en este sentido. La conversión es la misma para las tablas derivadas de objetos que la correspondencia a las que se derivan del enfoque tradicional. A continuación para mostrar las reglas de correspondencia que se ha seguido, se muestran algunas clases como ejemplo.

- **Correspondencia entre clases y tablas**

Toda clase se corresponde con una o más tablas; de igual manera que una clase tiene atributos, esos atributos pasan a ser atributos de la tabla, añadiendo el ID del objeto y detalles como parte de la formulación del modelo de tablas, especificando que atributos pueden ser o no nulos y asignando un dominio a cada atributo.

La figura VI.4 muestra como una clase del diagrama de clases es transformada en tabla.

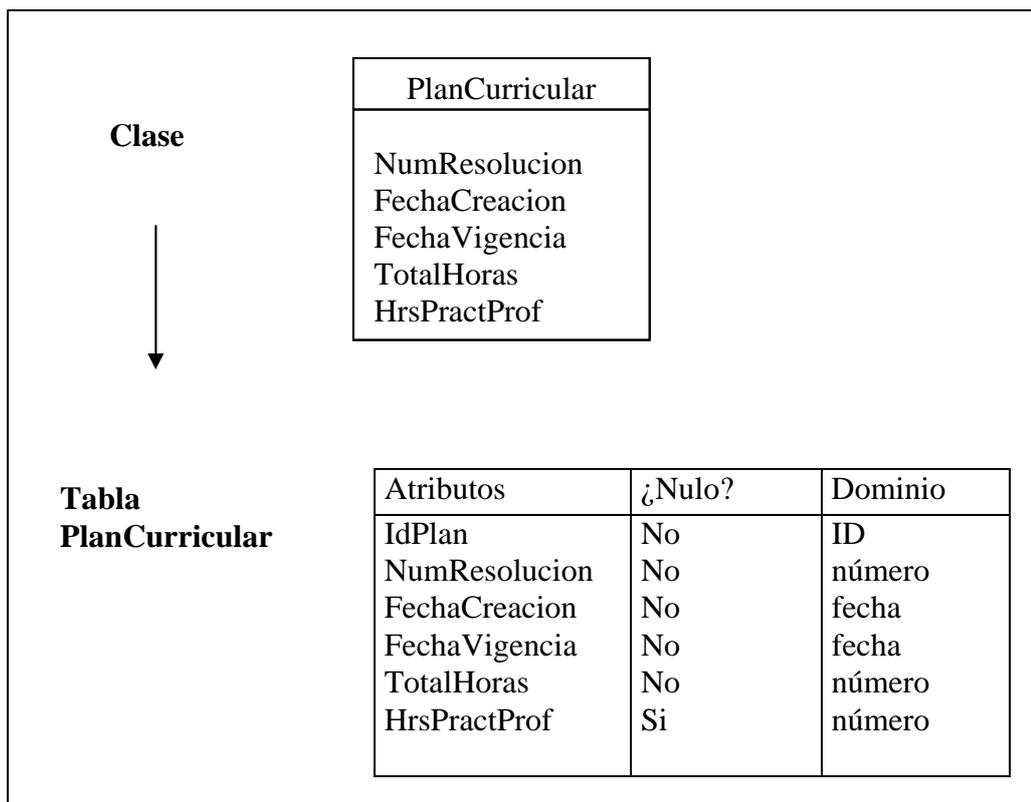


Figura IV.4: Correspondencia entre clases y tablas

- **Correspondencia entre asociaciones y tablas**

En términos generales, una asociación puede o no corresponderse con una tabla, ya que esto depende del tipo y multiplicidad de la asociación; y del criterio de quien diseña la base de datos.

La figura IV.5 muestra la asociación uno-a-muchos, donde la asociación incluye una clave externa en la tabla para la clase “muchos”.

La figura IV.7 muestra una asociación de muchos-a-muchos, extraída del diagrama de clases. La asociación muchos-a-muchos siempre se corresponde con una tabla concreta, satisfaciendo así la tercera forma normal. Las claves primarias tanto para las clases relacionadas como para los atributos de enlace pasan a ser atributos de la tabla asociación.

La figura IV.8 muestra como la asociación uno-a-uno se puede fundir y almacenar ambos objetos y asociación en una sola tabla; y al fusionarse se mejora el rendimiento, y se reduce el espacio de la base de datos y la posible violación de la tercera forma normal.

CAPITULO IV: DESARROLLO DEL SISTEMA

- **Correspondencia de las generalizaciones a tablas.**

Las figuras IV.9 y IV.10 muestran un modelo de clases con herencia simple. Sin embargo, la correspondencia que muestra la figura IV.9 es la de eliminar la tabla de superclase y los atributos de la superclase se duplican para cada subclase, esto permite eliminar la navegación de superclase a subclases y así acelerar el proceso, manteniendo la tercera forma normal.

En la figura IV.10, se elevan los atributos de las subclases al nivel de la superclase, esta regla es útil si hay dos o tres subclases con pocos atributos.

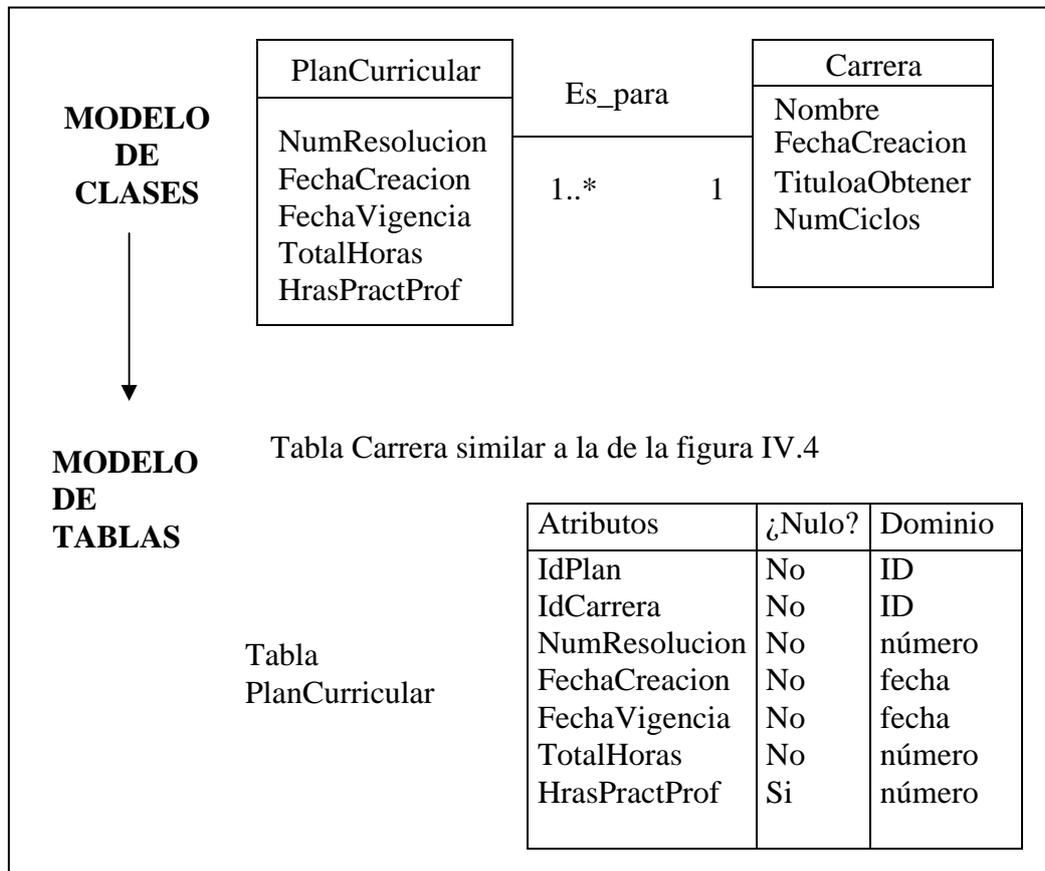
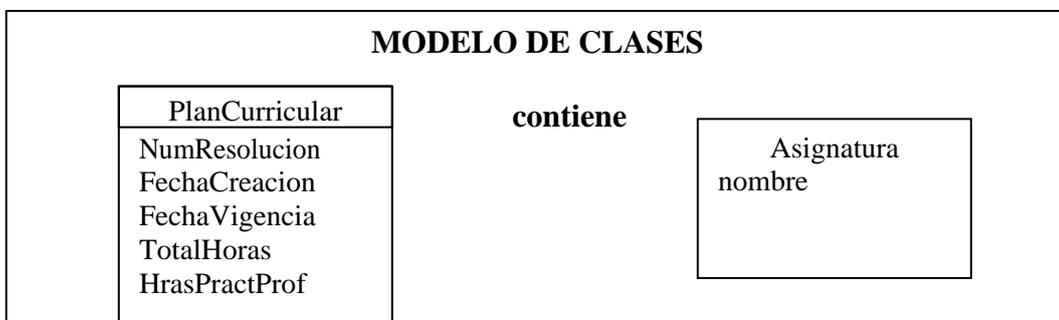


Figura IV.5. Correspondencia entre asociaciones uno a muchos y tablas



CAPITULO IV: DESARROLLO DEL SISTEMA

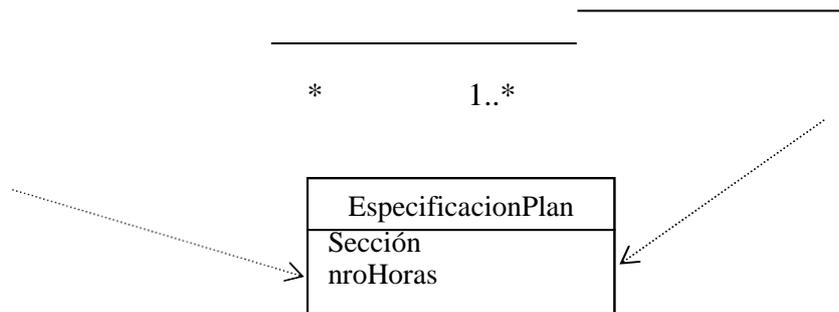


Figura IV.7: Correspondencia entre asociaciones muchos a muchos y tablas

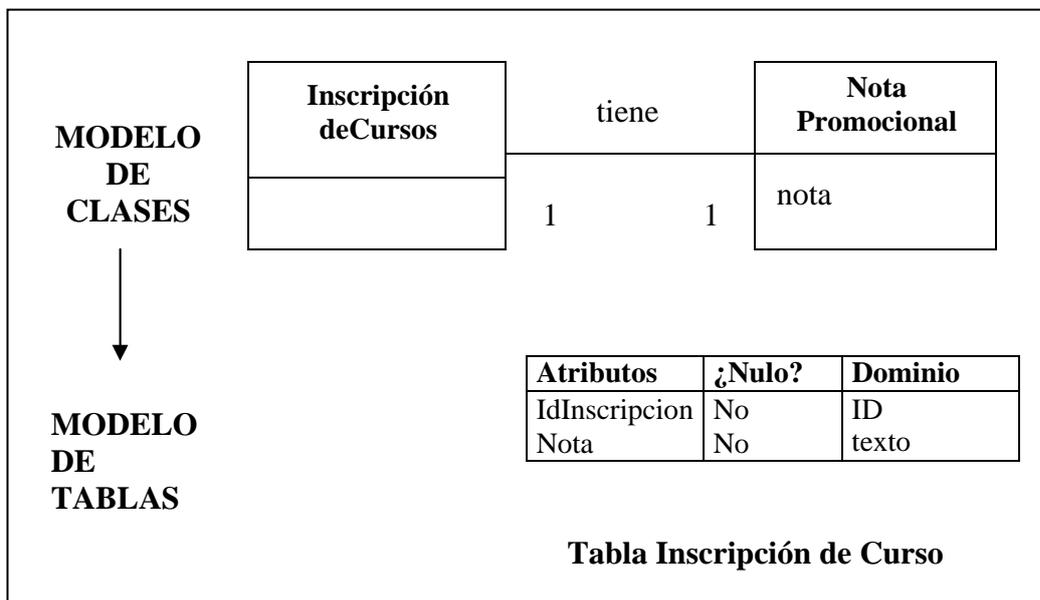


Figura IV.8: Correspondencia entre asociaciones uno a uno y tablas

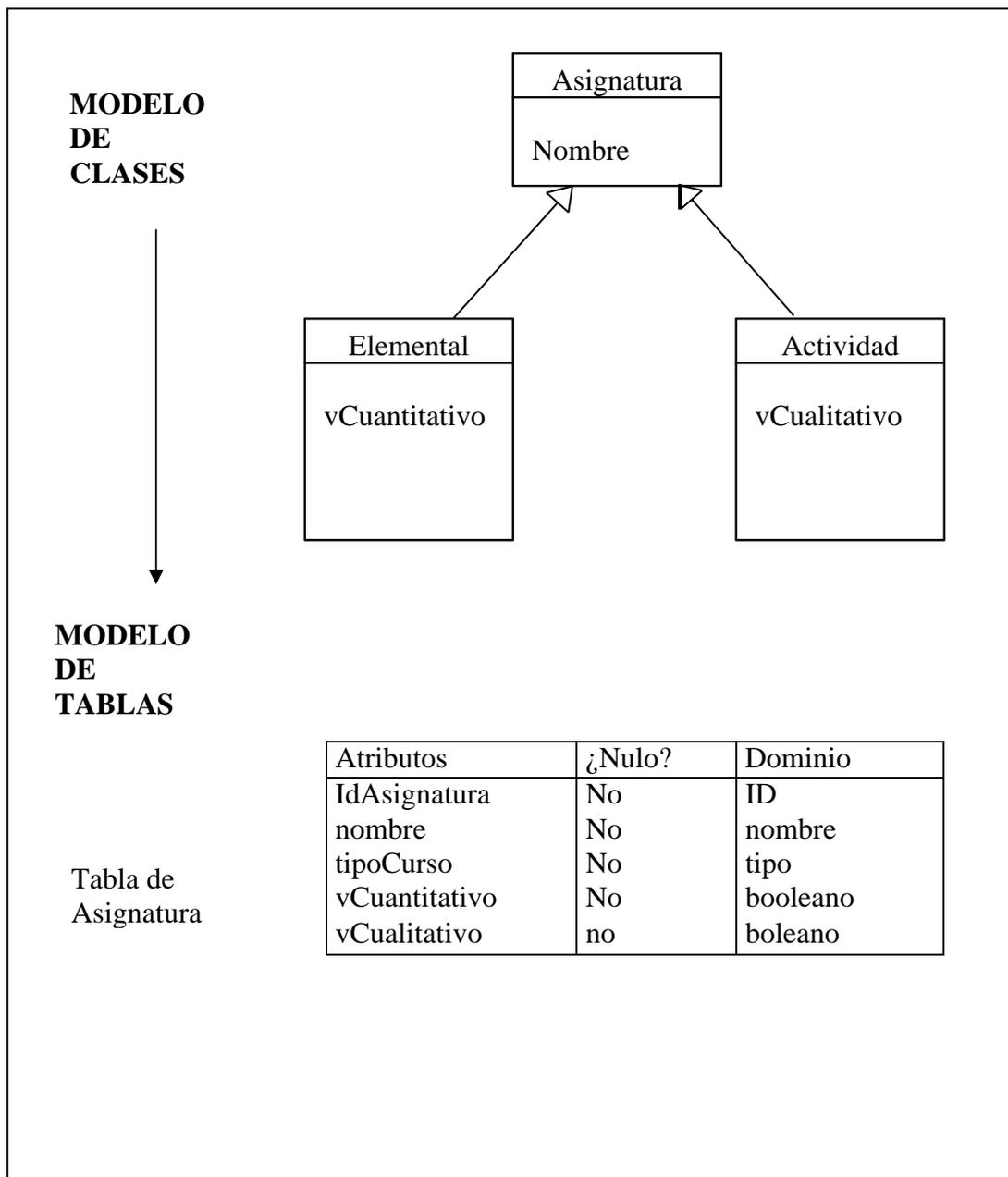


Figura IV.10: Modelo de tablas para la generalización: una tabla de superclase

CAPITULO IV: DESARROLLO DEL SISTEMA

En esta parte del proyecto, se implementa el diseño de la base de datos. Además, de trabajar con los diferentes tipos de objetos que ofrece Microsoft SQL Server: tablas, vistas, procedimientos almacenados, disparadores, etc.

Creación de Procedimientos Almacenados

Un procedimiento almacenado de SQL Server es similar a un procedimiento de otro lenguaje de programación y él crearlos representa las principales ventajas:

- Permiten una ejecución más rápida y pueden reducir el tráfico de red.
- Pueden utilizarse como mecanismo de seguridad, ya que es posible conceder permisos a los usuarios para ejecutar un procedimiento almacenado.

La creación de los siguientes procedimientos almacenados representa las operaciones o métodos que realiza cada clase en el Diagrama de clases.

A continuación se muestra un procedimiento almacenado del sistema:

1. Procedimiento Almacenado que permite insertar una fila en la tabla TblMatricula, especificando cada uno de los parámetros de entrada.

```
CREATE PROCEDURE proced_Insert_Matric @CodMatricula nchar(10), @Fecha smalldatetime,
@Seccion nchar (3), @Turno nchar (10), @Saldo smallmoney, @CodTupa nchar(2), @CodAlumno
nchar(6)
AS
IF NOT EXISTS(SELECT IdMatricula FROM TblMatricula WHERE IdMatricula =@CodMatricula)
BEGIN
    BEGIN TRANSACTION
    INSERT INTO TblMatricula (IdMatricula, Fecha, Seccion, Turno, Saldo, IdTupa,
    IdAlumno)
    VALUES(@CodMatricula,@Fecha,@Seccion,@Turno,@Saldo,@CodTupa,@CodAlumno)
END
ELSE
    Return -2
IF @@Error <> 0
BEGIN
    ROLLBACK TRANSACTION
    RETURN -1
END
COMMIT TRANSACTION
RETURN 0
```

CAPITULO IV: DESARROLLO DEL SISTEMA

GO

Creación de desencadenantes

Un desencadenador es una clase especial de procedimiento almacenado que actúa automáticamente cuando un usuario intenta una instrucción específica de modificación de datos en una tabla.

En SQL Server se creó varios desencadenadores, como el que se muestra:

1. Disparador en la tabla TblMatricula para eliminar una matricula mientras no exista inscripción de cursos en esa matricula:

```
CREATE TRIGGER DEL_MATRICULA ON dbo.TblMatricula
FOR DELETE
AS
IF EXISTS (SELECT * FROM TblInscripcion, Deleted WHERE
TblInscripcion.IdMatricula=Deleted.Idmatricula)
BEGIN
    ROLLBACK
END
ELSE
    DELETE TblDocumentoPago FROM TblDocumentoPago,
    Deleted
    WHERE TblDocumentoPago.IdMatricula=Deleted.IdMatricula
```

Creación de vistas

Una vista suele utilizarse para simplificar y personalizar la percepción de la base de datos para cada usuario. Además, puede emplearse como mecanismo de seguridad, ya que permite a los usuarios tener acceso a los datos por medio de la vista.

A continuación se presentan una vistas creadas en SQL Server:

CAPITULO IV: DESARROLLO DEL SISTEMA

1. Mostrar todos las asignaturas: código, nombre y tipo:

```
CREATE VIEW vCursos
AS
SELECT A.IdAsignatura AS Codigo, A.Nombre AS Nom,
        TipoCurso = case A.Tipo
        WHEN 'E' THEN 'CURSO ELEMENTAL'
        WHEN 'A' THEN 'ACTIVIDAD' END
FROM TblAsignatura A
```

4.3. IMPLEMENTACION DE LA APLICACIÓN

Como parte de la implementación del sistema, existe el proceso de decisión relacionado con la interfaz a utilizar. En este proyecto se ha seleccionado el lenguaje de programación Visual Basic 6.0 que soporta la abstracción, la encapsulación, el polimorfismo y la reutilización del código. Los objetos de Visual Basic están encapsulados, es decir, contienen sus propias propiedades, métodos y eventos. La mayoría de los lenguajes orientados a objetos proporcionan polimorfismo mediante la herencia. Visual Basic utiliza la técnica de interfaz múltiple del modelo de objetos componente que permite que los programas evolucionen con el tiempo, agregando nueva funcionalidad sin afectar al código existente.

4.3.1. Creación de Clases

El diagrama de clases se usa para generar una estructura base del código en el lenguaje escogido. Para la creación de clases, básicamente se ha ejecutado los siguientes pasos:

1. Crear un módulo de clase y establecer sus propiedades.
2. Agregar las propiedades de la clase.
3. Agregar los métodos.

Siguiendo lo expuesto, se muestran algunas de las clases Creadas:

'CLASE MATRICULA

'Propiedades o Atributos

Public codMat As String

CAPITULO IV: DESARROLLO DEL SISTEMA

Public fecha As Date
Public Sec As String
Public Turno As String
Public saldo As Currency
Public codTupa As String
Public CUA As String

'Métodos

```
'  
.....  
Public Function AddMatricula(CUM As String, fecha As Date, _  
Sec As String, Turno As String, saldo As Currency, codTupa As String, CUA  
As String)  
    'codigo  
End Function  
Public Function CostoMatricula(CUA) As Currency  
    'codigo  
End Function  
Public Function DeudaAnterior(CUA as string) As Currency  
    'codigo  
End Function
```

'CLASE INSCRIPCIONDECURSO

'Propiedades o atributos

Public CodInscrip As String
Public codMat As String
Public CUC As String
Public Cond As String

'Métodos

```
'  
.....  
Public Function AddInscripcion(CodInscrip As String, _  
codMat As String, CUC As String, Cond As String) As Integer
```

CAPITULO IV: DESARROLLO DEL SISTEMA

‘codigo

End Function

Public Function AddNota(CodInscrip As String, NotaProm As String) As

Integer

‘codigo

End Function

Public Function DelInscripcion(codInscripcion As String) As Integer

‘codigo

End Function

4.3.2 Ventanas

El diseño de las ventanas o pantallas responde a los requerimientos del sistema, ya que dentro de los atributos del sistema se encuentra la metáfora de interfaz, es decir, detallar las ventanas que faciliten el trabajo al usuario final.

El proceso de desarrollo de la aplicación se ha basado en este caso en los procesos académicos y en las actividades de los usuarios finales. Siendo los formularios la base para crear la interaz de una aplicación, se ha creado como pantalla principal una ventana (ver Figura IV.12) en la que podemos apreciar la estructura del sistema.



Figura IV.12. Menu Principal

CAPITULO IV: DESARROLLO DEL SISTEMA

A continuación se detallan las funciones del menú principal:

- **Generalidades:** Proporcionar información sobre Alumnos, Docentes, Cursos, Planes de Estudios, Carreras Profesionales, es decir, información elemental.
- **Aplicaciones:** Realizar los procesos académicos del sistema.
- **Consultas:** Proporcionar la información oportuna.
- **Reportes:** Emitir reportes de los procesos llevados cabo.
- **Utilitarios:** Opción que permite realizar operaciones de cálculo e informarse de la fecha, etc.
- **Ayuda:** Ventana que permite obtener información sobre el sistema.

Todas estas funciones del menú principal nos han llevado a la necesidad de contar con las siguientes ventanas:

- Ventana de acceso al sistema.
- Ventana del menu principal.
- Ventana para el Directorio Alumnos
- Ventana para el Directorio de Docentes
- Ventana para el Directorio de Asignaturas
- Ventana de Planes de Estudio
- Ventana para el Directorio de Carreras Profesionales.
- Ventana para el Directorio de Aulas.
- Ventana para la Programación de Carreras.
- Ventana para la Programación Académica
- Ventana para registrar los horarios.
- Ventana para el Proceso de Matricula.
- Ventana para la inscripción de asignaturas.
- Ventana para la carga de notas.
- Ventana para la recuperación de una asignatura.
- Ventana para la convalidación de asignaturas.
- Ventana para la actualización del pago de matricula.
- Ventana para la consulta de la lista de carreras programadas.
- Ventana para la consulta de la Distribución Académica.

CAPITULO IV: DESARROLLO DEL SISTEMA

- Ventana para la consulta de la carga académica de un determinado docente.
- Ventana para la consulta del horario de una determinada carrera.
- Ventana para la consulta de matriculas en general.
- Ventana para la consulta de matriculas por carrera profesional.
- Ventana para la consulta del estado de pagos de un alumno.
- Ventana para la consulta de cursos inscritos de un determinado alumno.
- Ventana para reportes de un Plan de Estudio.
- Ventana para reportes de la Distribución Académica.
- Ventana para reportes de la carga académica de un determinado docente.
- Ventana para reportes de Matriculas.
- Ventana para reportes de Deudores de Matricula.
- Ventana para reportes de Orden de Merito.
- Ventana para reportes de Alumnos con cursos a cargo y/o repitencia.
- Ventana para reportes de Boletas de Notas de un alumno.
- Ventana para reportes del Historial Académico de un alumno.
- Ventana para reportes de Certificados de Estudios.
- Ventanas diversas utilizadas en la opción Utilitarios.

4.3. Reportes

En la sección anterior se ha tratado el diseño de ventanas, técnica que permite la presentación y modificación de los datos. Sin embargo, esta presentación se ha logrado en formularios lo cual no constituye una mejor manera de imprimir los datos de una información.

La elaboración de los informes o reportes constituyen la mejor manera de crear una copia impresa de la información extraída o calculada a partir de la base de datos.

Los reportes del sistema se aprecian en el anexo E.

4.4. Seguridad del Sistema

En cuanto a la seguridad del sistema se ha tenido en cuenta los requerimientos de seguridad del Sistema de Información Académico del Instituto Superior Tecnológico Manuel Yarleque Espinoza, el cual esta basado en la protección de los datos y

CAPITULO IV: DESARROLLO DEL SISTEMA

seguridad de la información. Por ello se utilizarán técnicas que permitan que las personas autorizadas tengan acceso a los datos y entre ellas tenemos:

Seguridad de SQL Server.

Seguridad del Menú Principal de Visual Basic.

Seguridad de SQL Server

La seguridad que ofrece SQL Server pasa por dos fases:

- Autenticación: Proceso por el cual un servidor se registra en el sistema y este lo valida.
- Validación de Permisos: Controla las actividades que puede desarrollar el usuario tras su autenticación.

Fase 1: Autenticación

Existen dos formas de autenticar a los usuarios:

- Modo de Autenticación de Windows NT.
- Modo de Autenticación de SQL Server.

Modo de Autenticación de SQL Server

En nuestro caso el modo de autenticación de SQL Server es necesaria cuando se trabaja con Windows 95/98, ya que es el único sistema operativo que puede trabajar con dicho modo.

El modo de autenticación de SQL Server permite que un usuario que intenta conectarse especifique su nombre y contraseña, el servidor comprueba si la cuenta está configurada en SQL Server y si la contraseña coincide con la registrada. Si la cuenta del usuario es la correcta tendrá acceso, de lo contrario se producirá un error.

Fase 2: Validación de permisos

Independientemente del modelo de autenticación que se utilice hay que asignar cuentas para el usuario, ya que cada usuario realiza alguna tarea dentro del sistema y uno de los mecanismos de seguridad de Microsoft SQL Server es aplicar los permisos de este servidor a los usuarios. Los permisos nos ayudarán a controlar todas las actividades que puede desarrollar el usuario.

CAPITULO IV: DESARROLLO DEL SISTEMA

Los permisos serán asignados de acuerdo a las funciones de cada órgano de la institución.

Usuarios y Funciones

La determinación de usuarios es parte de la seguridad del Sistema de Información Académico, el mismo que se implementa con SQL Server, para lo cual se trabaja con funciones y usuarios.

Todos los usuarios que se conectan con SQL Server se tienen que identificar mediante un Id de inicio de sesión, el cual pedirá un nombre y una contraseña. A continuación la lista de Usuarios:

UADS: Usuario de la Administración del Sistema

UOSG: Usuario de la Oficina de Secretaría General.

UODG: Usuario de la Oficina de Dirección General.

Las funciones se han definido con la finalidad de poner restricciones no sólo a nivel de acciones (seleccionar, insertar, actualizar, eliminar) sino también para la restricción de datos. A continuación la definición de funciones (ver Tabla IV.1):

Tabla IV.1. Definición de funciones.

Nombre de las Funciones	Definición
Fx_Administración	Controlar toda la administración del Sistema Académico.
Fx_OSG	Contendrá las funciones que pueden realizar todos los procesos académicos.
Fx_ODG	Controlar las acciones que puede realizar la Secretaría General.

Es simple deducir los usuarios asignados a las funciones por lo que la tabla IV.2 permite observar esas asignaciones.

Tabla IV.2. Asignación de Funciones

Usuarios	Funciones		
	Fx_Administración	Fx_OSG	Fx_ODG

CAPITULO IV: DESARROLLO DEL SISTEMA

UADS	X		
UOSG		X	
UODG			X

Los permisos (término anteriormente mencionado en la validación de permisos) son todas las acciones que se pueden conceder a los usuarios, por lo tanto las funciones dependen de los permisos. En la Tabla IV.3. podemos observar las acciones asignadas a cada función y de esta manera se puede controlar el acceso de los usuarios.

Tabla IV.3. Asignación de Acciones.

Funciones	Acciones					
	Leer	Insertar	Actualizar	Eliminar	Bk Restauración B.D	Administración
Fx_Administración	X	X	X	X	X	X
Fx_OSG	X	X	X	X	X	
Fx_ODG	X					

Backup y recuperación de datos

Como parte de la seguridad del sistema existe la creación de copias de seguridad ya que la pérdida de datos se puede dar por muchas razones: virus, usuarios incompetentes o maliciosos, dispositivos de discos dañados o robo. Por lo tanto frente a estas situaciones hay que establecer un plan para proteger los datos de la institución.

Las copias de seguridad se realizarán mediante el Administrador Corporativo de SQL Server contando con la autorización de la Oficina de Secretaría General, ya que el plan que se establezca depende de la cantidad de transacciones que se estén efectuando. Si se registran gran cantidad de transacciones en un día es posible que se tome la decisión de hacer una copia de seguridad todos los días, pero si no se modifican los datos con frecuencia, bastara con hacer un backup semanal.

CAPITULO V: EVALUACIÓN DEL SISTEMA

CAPITULO V: EVALUACIÓN DEL SISTEMA

5.1. COSTOS Y BENEFICIOS DEL SISTEMA PROPUESTO

En esta parte del proyecto se consideran los costos de desarrollo del sistema, dado que no podríamos hablar de los costos de equipos para la implementación del sistema ya que la institución cuenta con una tecnología que le permite satisfacer los requerimientos de software y hardware.

Tabla V.1. Equipos disponibles de la institución en la oficina de Secretaría General.

Equipos	Características
1 Computador personal	Procesador: Pentium III Disco Duro: 30 Gb. Memoria: 128 Mb de RAM. Monitor: 14" VGA Mouse: De dos botones, Teclado: Estándar de 101/102 teclas. CD-ROM
Impresora LX 300+	Tipo matricial

5.1.1. Costos del Desarrollo del Sistema.

Los costos del desarrollo del sistema se dividen en: Costos de desarrollo del software, costos de equipo para el desarrollo, y otros costos adicionales.

- **Costos de desarrollo del software.** Este costo lo constituyen el monto fijado por el personal especialista en el desarrollo del sistema.

Costo (S/.)

- Análisis, diseño, programación e implementación

(S/. 1000.00 mes * 6 meses)

S/. 6,000.00

- **Costos de equipo para el desarrollo.** Son todos los costos incurridos en el uso del equipo necesario para el desarrollo del sistema en un total de 6 meses.

Costo (S/.)

- Computador

(s/. 1.00 hora * 8horas/día * 6 días/semana *

CAPITULO V: EVALUACIÓN DEL SISTEMA

4semanas/mes * 6 meses)	S/. 1,152.00
- Impresora (Pruebas y Reportes)	
(s/.0.50 hoja * 100 hojas máximo)	<u>S/. 50.00</u>
Total	S./1,202.00

- **Otros costos.** Estos son todos los costos que incluyen materiales de oficina y algunos imprevistos.

	Costo (S/.)
- Materiales de Oficina	S/. 200.00
- Imprevistos	<u>S/. 100.00</u>
	S/. 300.00

La siguiente tabla es un resumen de los costos del desarrollo del sistema.

Tabla V.2. Costos del Desarrollo del sistema

COSTOS	TOTAL(S/.)
Costos de desarrollo del software	6,600.00
Costos de equipo para el desarrollo	1,202.00
Otros Costos	300.00
TOTAL	8,102.00

5.1.2. Costos de Mantenimiento del Sistema

Los costos de mantenimiento del sistema involucran la administración diaria del sistema, la que se encontrará a cargo del administrador (rol a cargo del jefe del laboratorio de cómputo de la institución) y usuario del sistema (rol desempeñado por el asistente del Secretario Docente). Estos costos están asociados a la organización interna de la institución.

5.1.3. Beneficios

Los beneficios del sistema de información se manifiestan de muchas formas, entre ellos tenemos los beneficios tangibles e intangibles. El nuevo sistema de información se ha proyectado para una vida útil de 5 años, teniendo como base los siguientes criterios:

CAPITULO V: EVALUACIÓN DEL SISTEMA

- En todos los años de fundación de la institución, no se han realizado cambios relevantes en la forma de llevar a cabo el proceso, por lo que considerando el historial académico de la organización se estima que la posibilidad de realizar cambios en el proceso académico sea aproximadamente de 5 años.
- Dirección ha requerido operar el sistema de información con datos históricos (3 años antes a partir del funcionamiento del sistema).
- El nuevo sistema de información está desarrollado en una metodología y una tecnología que permitirá incorporar las nuevas demandas de información por parte del personal académico.

Beneficios Tangibles

En nuestro caso, son aquellos beneficios que se pueden comprobar en términos de tiempo y economía.

- **Beneficios obtenidos del ahorro de tiempo**

En la tabla V.3, se compara los tiempos efectuados en una actividad académica durante el sistema manual y el sistema de información, en la cual podemos apreciar que el sistema de información frente al sistema manual, ofrece significativamente un ahorro de tiempo en cada una de las actividades mencionadas. Las comparaciones se han hecho considerando como base que el personal académico tiene una jornada de 8 horas diarias.

CAPITULO V: EVALUACIÓN DEL SISTEMA

Tabla V.3. Beneficios de tiempo.

Actividades frecuentes	Sistemas		Beneficio	Tiempo Total de Ahorro
	Manual	Mecanizado		
	Tiempo. (min.)	Tiempo. (min.)		
a) Obtener información personal de un alumno.	20	1	En un día aproximadamente se busca información de 5 alumnos, lo que implica usar 100 min. diarios en el sistema manual, mientras que con el nuevo sistema 6 min. diarios.	94 min. por día
b) Obtener y elaborar un informe del cuadro de Distribución Académica.	30	5	Una vez ingresados los datos acerca de la Programación académica, el sistema de información puede mostrar un informe general o generar un informe de por docente en el momento que lo requiera.	25 min.
c) Obtener una relación de todos los deudores por concepto de matricula.	30	1	Información rápida y segura en el momento necesario.	29 min.
d) Matricula de un alumno.	20	4	Si el sistema manual implica usar 20 min. por alumno, eso implica atender 24 alumnos por día mientras que el otro implica atender 120 alumnos por día.	384 min. por día.
e) Inscripción de cursos por alumno.	30	15	La inscripción de cursos de un alumno se lleva a cabo aproximadamente en 30 min. en el sistema manual esto implica atender 16 alumnos en un día, mientras que el otro implica atender a 32 alumnos.	240 min. por día.
f) Ingreso de notas por acta.	30	5	El ingreso de notas para la elaboración de actas en el sistema manual implica elaborar 16 actas en un día.	400 min.
g) Obtener el historial académico de un alumno.	30	2	En un día se podrá atender a más de 16 alumnos.	448 min. por día.
h) Obtener boleta de notas semestrales por alumno.	10	2	El sistema manual podrá atender en un día 48 alumnos, mientras que el sistema de información atenderá a 240 alumnos en un día.	384 min.

CAPITULO V: EVALUACIÓN DEL SISTEMA

Actividades frecuentes	Sistemas		Beneficio	Tiempo Total de Ahorro
	Manual	Mecanizado		
	Tiempo. (min.)	Tiempo. (min.)		
i) Obtener informe de alumnos con asignaturas a cargo y/o repitencia.	30	2	Se podrán realizar otras actividades.	28 min. en un día.
j) Elaborar cuadros de merito.	240	2	El elaborar un cuadro de meritos le lleva aproximadamente 4 horas al sistema manual.	238 min.
k) Servicios académicos	240	2	El nuevo sistema puede elaborar informes (certificados, horarios por alumno) en un mínimo tiempo de aproximadamente 2 min.	238 min.

CAPITULO V: EVALUACIÓN DEL SISTEMA

- **Beneficios económicos**

Teniendo en cuenta que el objetivo primordial de la institución al aprobar el presente proyecto, es lograr un considerable ahorro de tiempo, el que se expresará en términos económicos, es por ello que considerando que el sueldo neto del personal académico es de S/.800.00 mensuales en un jornada de 8 horas diarias de trabajo. Hemos calculado que el costo del personal académico es de S/.0.067 por minuto ($S/.800.00/\text{mes} * 1 \text{ mes}/5\text{semanas} * 1 \text{ semana}/5\text{días} * 1\text{día}/480\text{minutos}$). Asimismo, se ha considerado que el personal académico opera 300 días al año ($12 \text{ meses/año} * 5 \text{ semanas/mes} * 5 \text{ días/semana}$). Estos aspectos se utilizarán como base para hallar el beneficio obtenido en el desarrollo de algunas actividades en un lapso de un año (ver tabla V.4).

Tabla V.4. Beneficios obtenidos en el ahorro de tiempo al efectuar una actividad.

Actividades	Tiempo por actividad		Costo Anual por Actividad (S/.)		Beneficio Anual (S/.) (I) - (II)
	Sist. Manual	Sist. Inform.	Sist. Manual ☺ * (300)*(0.067)	Sist. Inform. ☹ * (300)*(0.067)	
	☺	☹	(I)	(II)	
a)	20	1	402	20.1	
b)	30	5	603	100.5	
c)	30	1	603	20.1	
d)	20	4	402	80.4	
e)	30	15	603	301.5	
f)	30	5	603	100.5	
g)	30	2	603	40.2	
h)	10	2	201	40.2	
i)	30	2	603	40.2	
j)	240	2	4 824	40.2	
k)	240	2	4 824	40.2	
Total	710	41	14 271.00	824.10	13 446.90

CAPITULO V: EVALUACIÓN DEL SISTEMA

A estos beneficios se añaden los siguientes beneficios:

- **Ahorro de energía.**- Aunque las horas extras no implican pago de personal ya que mayormente la institución requiere de personal de apoyo (alumnos practicantes), pero por cada hora extra en el sistema manual hay un consumo extra de energía, lo que con el sistema de información ya no sería necesario, proporcionado un ahorro anual de S/.80.00.
- **Papel e impresiones.**- El costo del sistema anterior significativamente se ha convertido en un ahorro anual de S/.100.00.

Tabla V.5. Resumen de los beneficios económicos

Beneficios	Total (S/.)
Actividades	13 446.90
Ahorro de energía	80.00
Papel impresiones	100.00
Total (S/.)	13 626.90

Beneficios Intangibles

- El IST “MYE”, cuenta con una información organizada, que le permitirá contar con información oportuna en forma más rápida y segura.
- Secretaría General tiene la posibilidad de ofrecer a los alumnos un eficiente servicio académico.
- Garantizar seguridad a la información a través de los usuarios.
- Automatizar y llevar un mejor control de los procesos académicos principales.
- Reducción de pérdidas de información y facilidad para realizar cambios.

CAPITULO V: EVALUACIÓN DEL SISTEMA

5.2. ANÁLISIS ECONÓMICO

El análisis económico consiste en evaluar los costos estimados y los beneficios obtenidos, con el fin de determinar si el proyecto es rentable. Es evidente, que el mayor beneficio obtenido será cuando el sistema comience a funcionar, ya que el objetivo principal es el ahorro de tiempo.

Para determinar si el proyecto es rentable, consideraremos que el sistema de información tiene 5 años de vida útil por los criterios antes mencionados. En la tabla V.6 se muestran los costos calculados para 5 años de vida útil.

Tabla V.6: Costos de los Sistemas en proyección de 5 años.

$$Va = C_{S.M} - C_{S.I}$$

Donde: **Va** = Valor del ahorro

$$C_{S.M} = S/. 71 355.00$$

$C_{S.M}$ = Costo del Sistema Manual.

Años	0	1	2	3	4	5	Total S/.
Sist. Manual:							
• Actividades		14 271	14 271	14 271	14 271	14 271	71 355
Sist. Inform.:							
• Desarrollo de Software	6 600	-	-	-	-	-	6 600
• Equipos	1 202	-	-	-	-	-	1 202
• Mantenimiento (Actividades)	-	824.10	824.10	824.10	824.10	824.10	824.10
							8 626.10

$C_{S.I}$ = Costo del Sist. de Inform.

$$Va = S/. 62 728.90$$

Puesto que $Va > 0$, concluimos que el sistema de información es rentable para el tiempo de vida útil de 5 años.

ANEXO I: Diagrama de Secuencia del Sistema

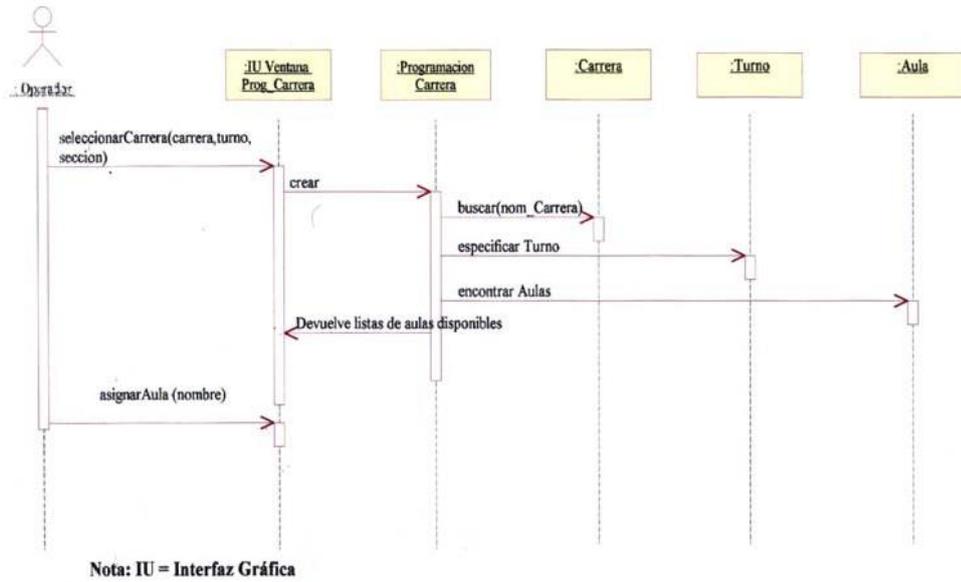


Figura III.3. Diagrama de Secuencias: Registrar Programacion de Carrera

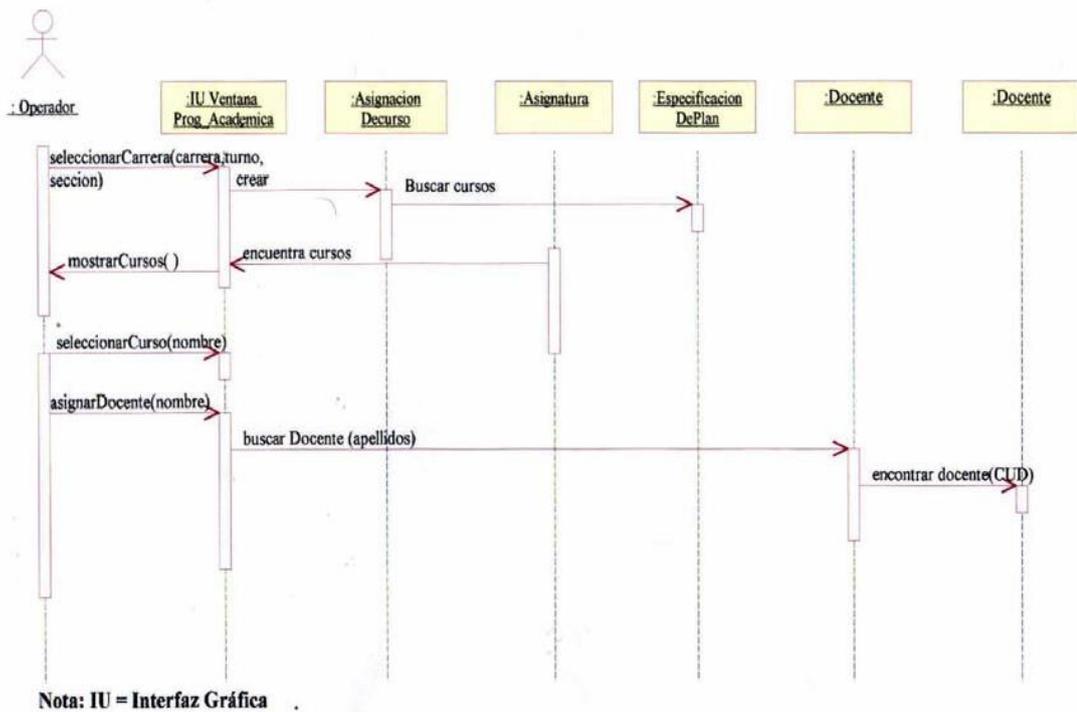


Figura III.4. Diagrama de Secuencias: Registrar Asignación Académica

ANEXO I: Diagrama de casos de Uso

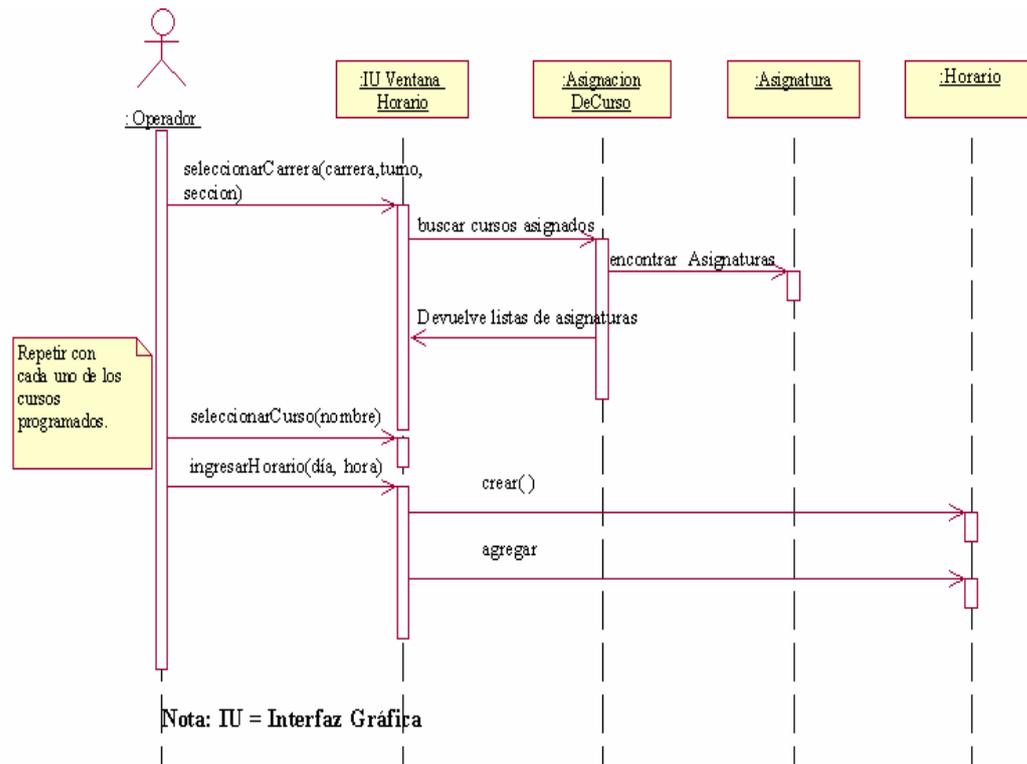


Figura III.5. Diagrama de Secuencias: Registrar Horario

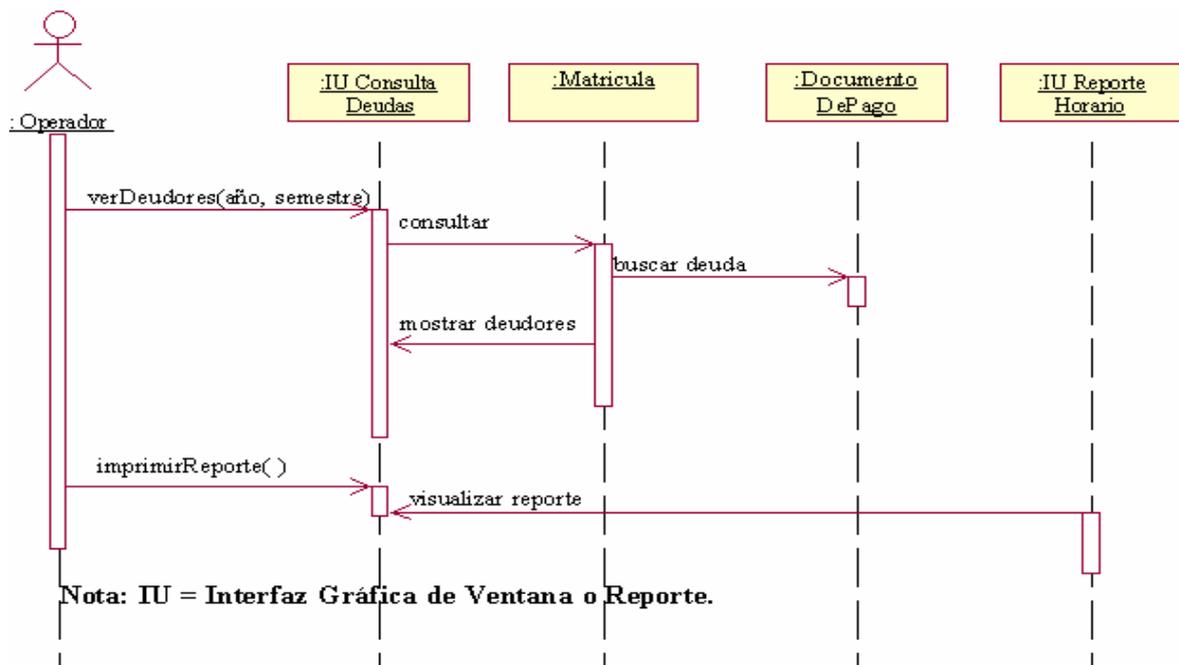


Figura III.6. Diagrama de Secuencias: Obtener Deudores

ANEXO I: Diagrama de casos de Uso

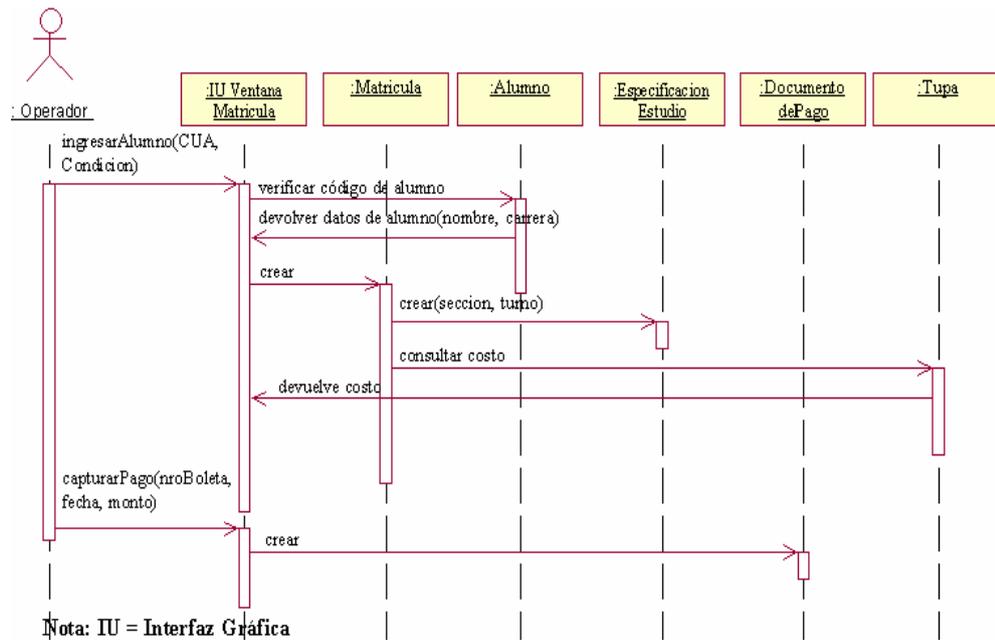


Figura III.7. Diagrama de Secuencias: Registrar pago de matricula

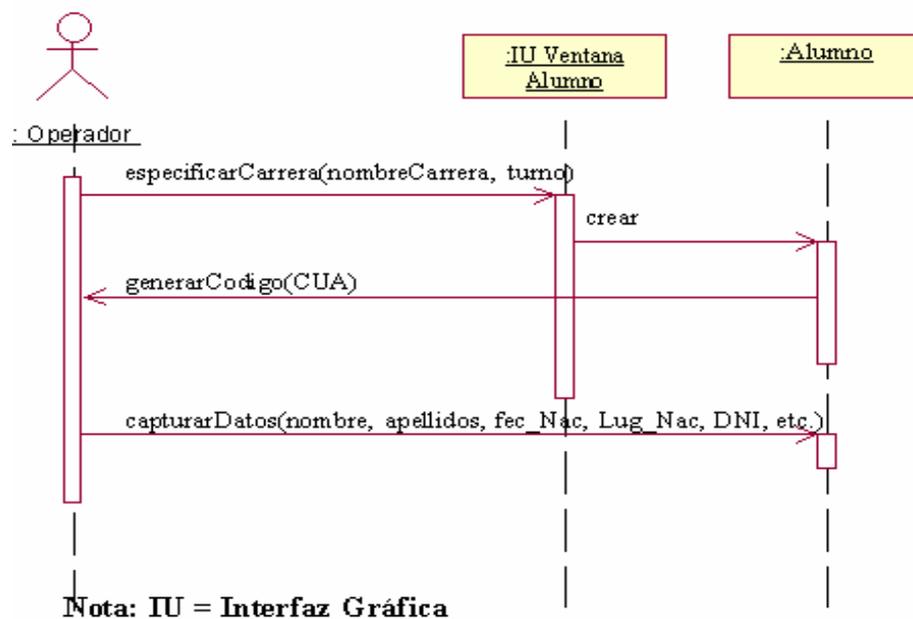


Figura III.8. Diagrama de Secuencias: Registrar nuevo alumno

ANEXO I: Diagrama de casos de Uso

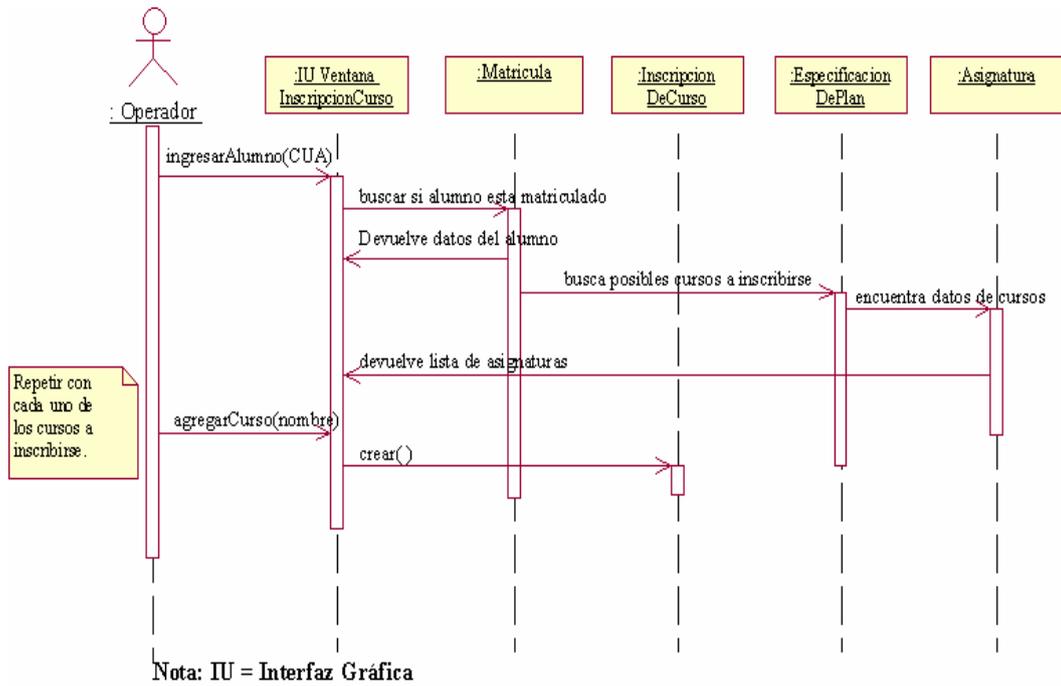


Figura III.9. Diagrama de Secuencias: Registrar inscripción de curso

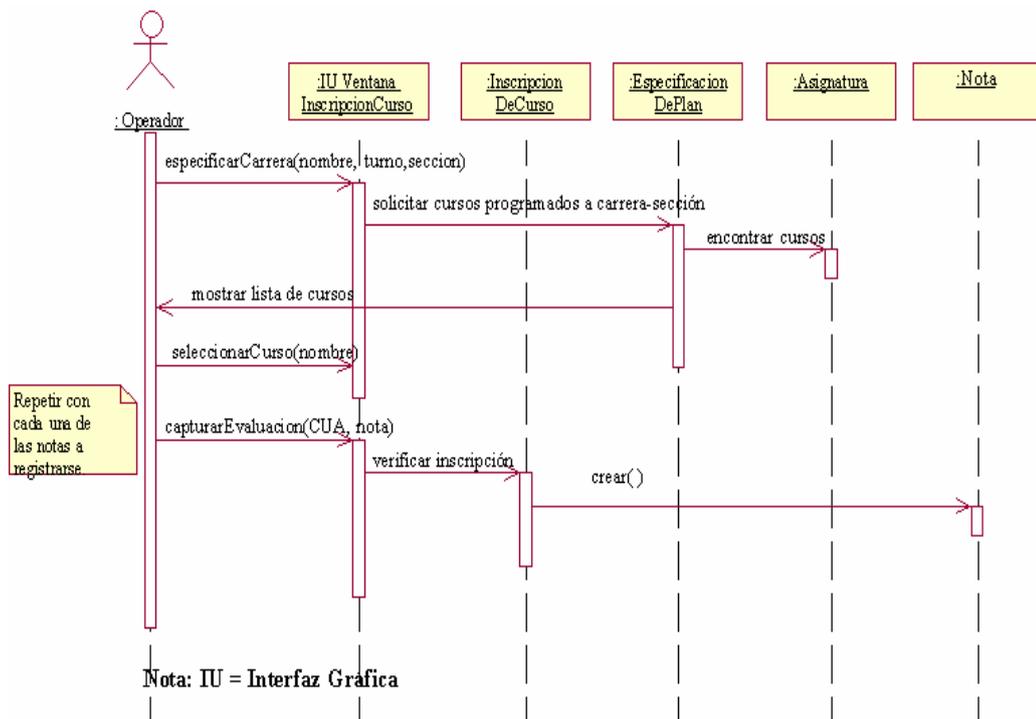


Figura III.10. Diagrama de Secuencias: Registrar nota promocional de asignatura

ANEXO I: Diagrama de casos de Uso

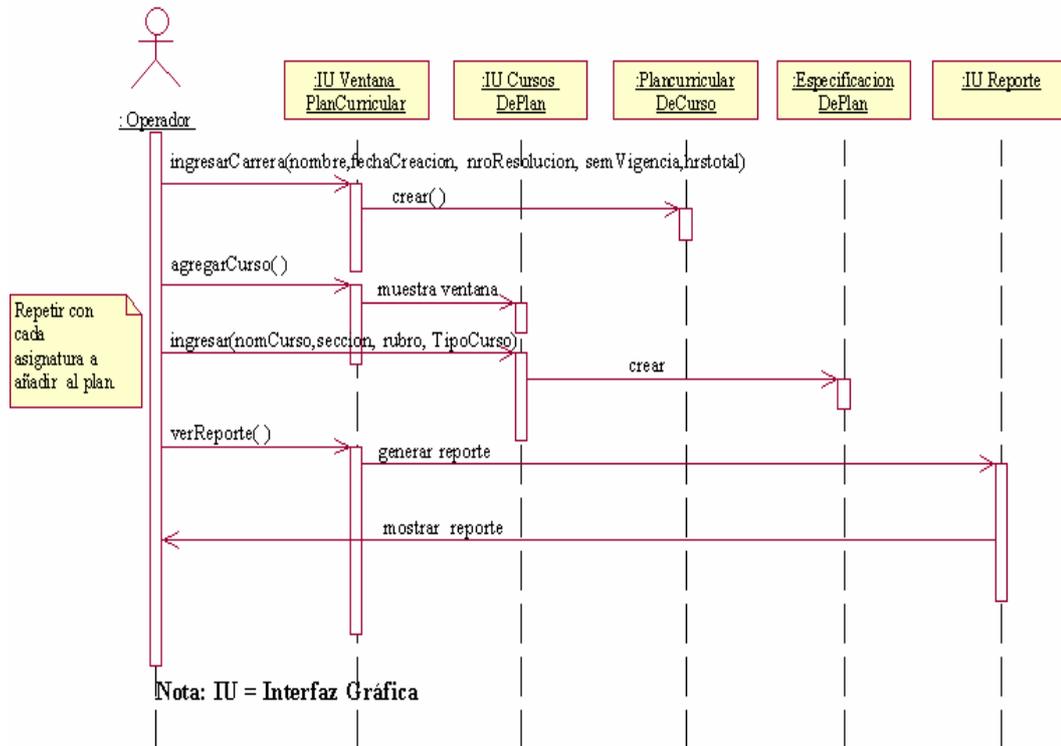


Figura III.11. Diagrama de Secuencias: Registrar plan curricular

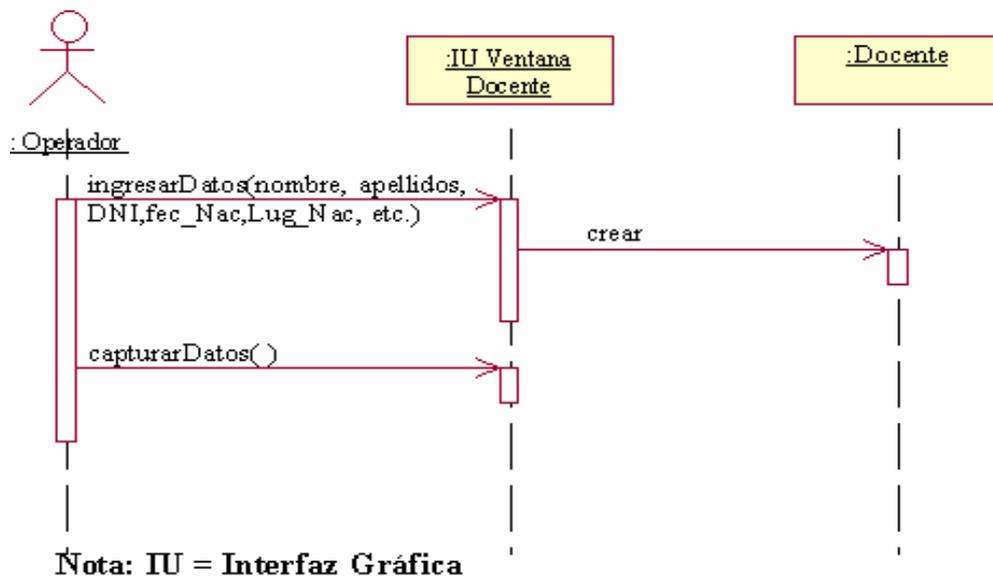


Figura III.12. Diagrama de Secuencias: Registrar nuevo docente

ANEXO I: Diagrama de casos de Uso

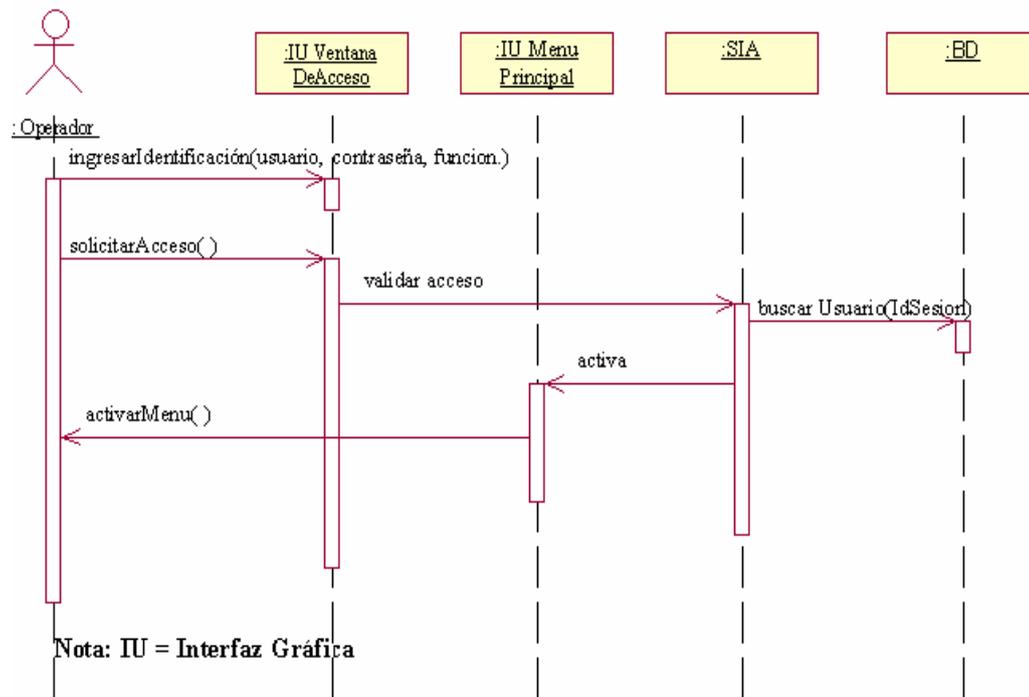


Figura III.13. Diagrama de Secuencias: Control de Acceso