

CAPÍTULO I

CAPÍTULO I: EL PROYECTO

1.1 PRESENTACIÓN DEL PROYECTO

1.1.1 Título del Proyecto

Mejoramiento de la Administración Académica del Instituto Comercial Superior (INCOS) Tarija.

1.1.2 Carrera/Unidad

Ingeniería Informática.

1.1.3 Facultad

Ciencias y Tecnología.

1.1.4 Duración del Proyecto

8 meses.

1.1.5 Área/línea de Investigación Priorizado

Tecnologías de la Información y Comunicación - Desarrollo de Sistemas Informáticos – Sistema Web

1.1.6 Responsable del Proyecto

Carrera de Ingeniería Informática – Taller III Grupo 1.

1.1.7 Entidad Asociada (s)

Instituto Comercial Superior INCOS Tarija

1.2 PERSONAL VINCULADO AL PROYECTO

1.2.1 Director del Proyecto

Narvaez	Michel	Wilson Luis	7158693 Tja.
Apellido Paterno	Apellido Materno	Nombre	C.I.
Estudiante Univ.	Ing. Informática	Ciencias y Tecnología	
Profesión	Carrera o Unidad	Facultad	
-----	76198345	wilsonthebad@gmail.com	
Telf. Oficina	Celular	Correo electrónico	Firma

Tabla 1. Director del Proyecto

1.2.2 Participantes del Equipo de Trabajo

Categoría	Nombres y Apellidos	Carrera/Profesión	C.I.	Firma
Director	Wilson Luis Narvaez Michel	Estudiante en Ingeniería Informática	7158693Tja	
Asesor	Ing. Roberto Salinas	Ing. De Sistemas		

Tabla 2. Participantes del Equipo de Trabajo

1.2.3 Actividades Previstas para los Integrantes del Equipo de

Investigación

Responsable	Actividades
Director	Como Jefe de Proyecto: - Planificar y controlar el cronograma del proyecto. - Asignar y gestionar recursos y prioridades a los distintos componentes y actividades del proyecto. - Coordinar las iteraciones entre el equipo de trabajo y los usuarios del proyecto. - Mantener al equipo del proyecto enfocado en los objetivos. - Establecer un conjunto de prácticas que aseguren la calidad e integridad del proyecto. - Supervisar el desarrollo del proyecto. Como Analista de Sistemas: - Capturar la especificación y validación de requisitos interactuando con los usuarios. - Elaborar el Análisis y Diseño del Sistema. Como Programador: • Realizar la Programación del Sistema. - Elaborar las pruebas funcionales del Sistema. Como Capacitador: - Formar al personal en el uso de las TIC para el manejo del producto final.

Asesor	- Asesoramiento en los aspectos tecnológicos para el desarrollo del Proyecto. - Asesoramiento en el uso de las Metodologías de desarrollo. - Evaluación del documento del proyecto.
---------------	---

Tabla 3. Actividades Previstas para los Integrantes del Equipo de Investigación

1.2.3.1 Unidades de Gestión: Organigrama del Equipo del Proyecto



Figura 1. Organigrama del Equipo del Proyecto

1.3 DESCRIPCIÓN DEL PROYECTO

1.3.1 Justificación

El Instituto Comercial Superior “INCOS” Tarija es una institución de la ciudad de Tarija encargada de:

Promover una educación superior integral y comunitaria, capaz de formar profesionales de excelencia con principios y valores éticos comprometidos con el desarrollo de la sociedad y el Estado.

La institución, tiene dificultad en dar a conocer información actualizada a la población, estudiantes, docentes y personal administrativo de la institución, debido a que actualmente maneja la información de(notas, carreras, materias) mediante comunicados escritos que son publicados dentro de sus instalaciones, lo que provoca que muchas personas, que no tienen acceso a la institución desconozcan información que es indispensable para la población o también los mismos miembros que no tienen la disponibilidad de tiempo para acercarse al Instituto Comercial Superior INCOS en cualquier momento para recibir la información de importancia para ellos.

En la actualidad, la manera de manipular la información de dicha institución es manual, debido a esta situación el proceso de obtener y administrar la información se torna de manera complicada y morosa, hay una inexistencia de validación de datos en el registro de información que se almacena sobre Docentes y Estudiantes y las diferentes actividades que estos efectúan en INCOS.

Para enfrentar este problema, el presente proyecto llamado “INCOS” que tiene la finalidad de fortalecer la actividad administrativa del Instituto Comercial Superior (INCOS) Tarija, con el propósito de Mejorar la Administración Académica de la información del Instituto Comercial Superior Tarija, el cual contará con los módulos usuarios, materias, reporte de notas y otros.

Este proyecto contempla un conjunto de elementos que interactuarán entre sí, con el fin de apoyar a las actividades acordes con las necesidades de Administración

Académica.

El Sistema genera Reportes de Notas de los Alumnos que son de utilidad tanto para docentes como para estudiantes de la institución.

Para los estudiantes como para docentes se facilita el acceso a la información del Instituto disponible vía internet. Para ello es de vital importancia, el desarrollo del sistema automatizado y la socialización del sistema y de las TIC's al personal de la institución; por el impacto que ha tenido el desarrollo de proyectos informáticos con el fin de automatizar la información para facilitar el trabajo, reducir el tiempo de acceso a la información, aumentar la eficiencia, etc. a tenido mucho éxito en los lugares donde fueron desarrollados y a los problemas identificados como pérdida de tiempo, acceso a la información, se vio conveniente desarrollar un proyecto informático para el fortalecimiento de la gestión de información.

1.3.1.1 Análisis de Causas del Problemas

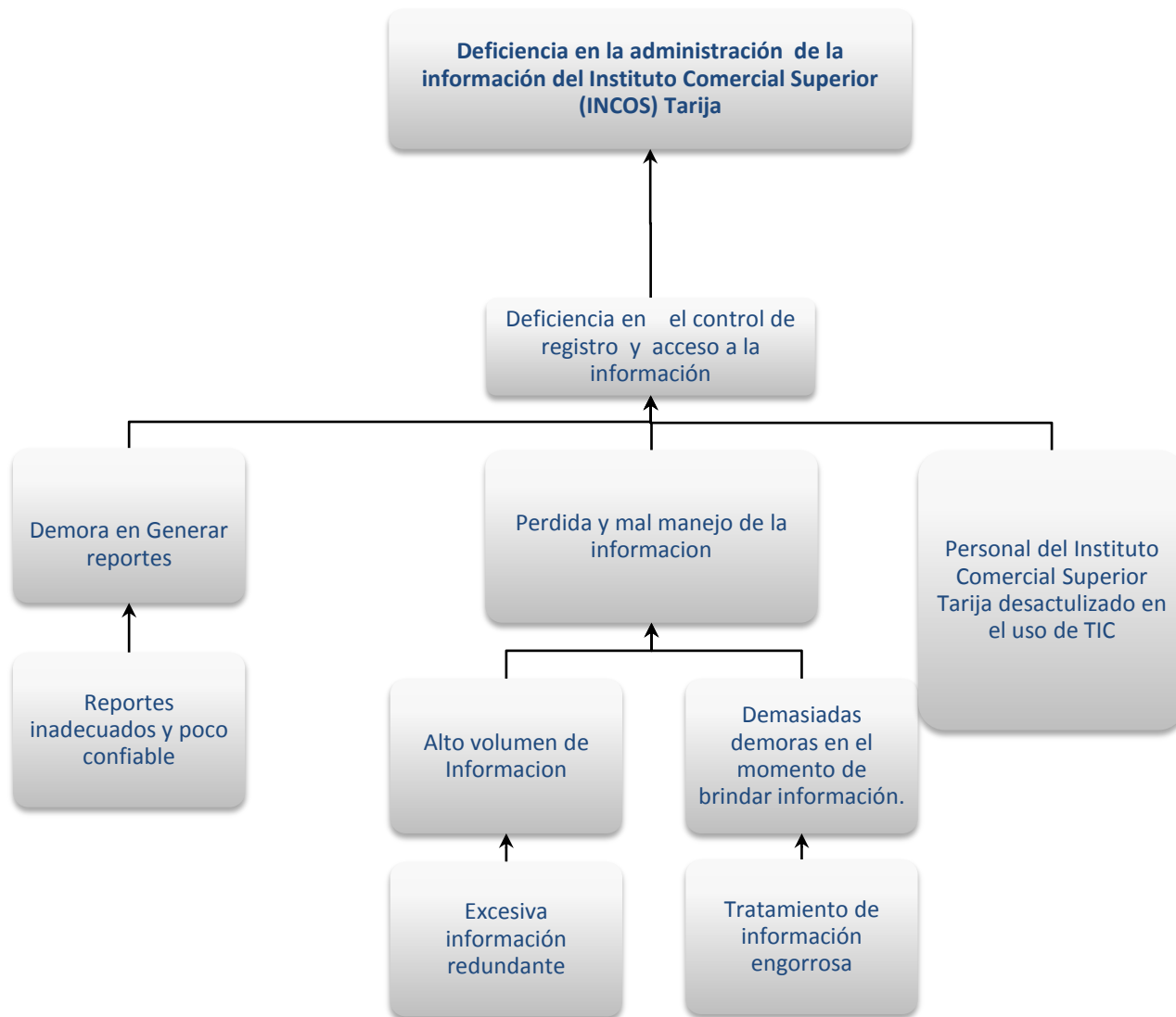


Figura 2. Árbol de Problemas

1.3.1.2 Análisis de Objetivos

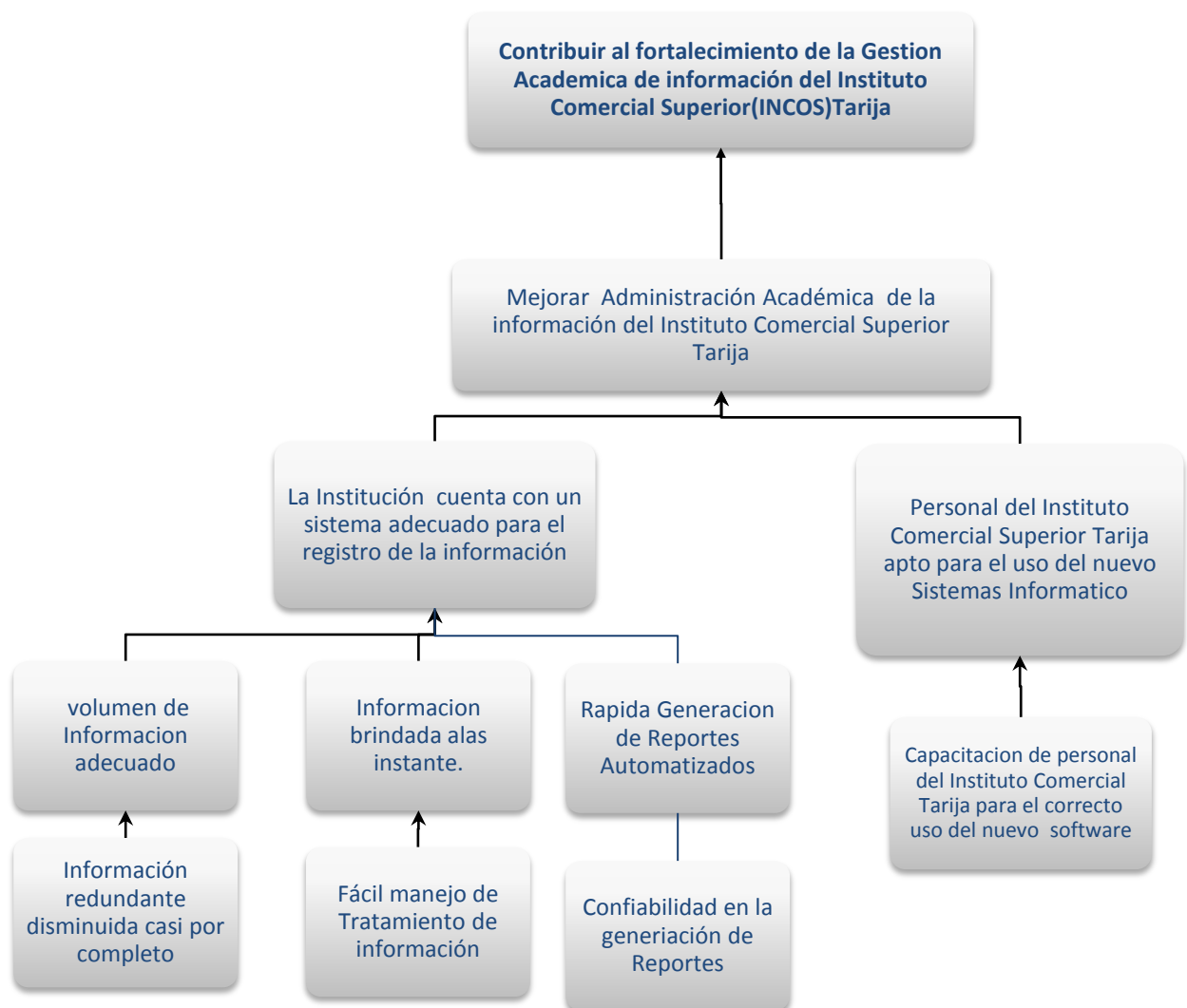


Figura 3. Árbol de Objetivos

1.3.2 Objetivos

1.3.2.1 Objetivo General

Mejorar la Administración Académica del Instituto Comercial Superior Tarija.

Objetivos Específicos

- Analizar la información sobre el manejo de datos académicos del Instituto Comercial Superior INCOS Tarija.
- Desarrollar un Sistema Informático para el apoyo administrativo del Instituto Comercial Superior Tarija
- Realizar un taller de capacitación para el personal administrativo del Instituto Comercial Superior Tarija sobre el uso del nuevo Sistemas Informático

La forma de trabajo de este proyecto está basado en:

1.3.2.2 Metodología para el desarrollo de las aplicaciones

Se utilizó la metodología **RUP (Racional Unified Process)**, que mejora considerablemente la calidad de desarrollo del sistema, ya que la misma utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software.

RUP es un proceso ágil de desarrollo que se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes.

El flujo de trabajo fundamental tiene los siguientes pasos:

- Requerimientos: Traslado las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado los requerimientos dentro de la arquitectura de software.
- Programación e Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que

todo lo solicitado está presente.

Requerimientos: En base a las entrevistas se obtuvo la información que refleja las necesidades de los involucrados para la determinación de requerimientos.

Análisis y diseño: En base a la determinación de requerimientos, se estructuró las diferentes vistas (Diagramas, base de datos, Pantallas) de la aplicación, tomando en cuenta metodologías de desarrollo de software.

Programación e Implementación: La programación fue de forma modular y orientada a objetos, se utilizaron tecnologías de punta contando con el apoyo de programadores experimentados (ingenieros informáticos) y programadores novatos (estudiantes), creando la aplicación informática que tenga el comportamiento deseado.

Pruebas y Validación: Antes de desarrollar las pruebas se procedió a la introducción de datos. Introducida esta información al sistema se dará inicio a la fase de pruebas de desarrollo que serán mediante casos de prueba tomados de cada módulo y se realizarán los ajustes necesarios para una correcta validación.

Este proceso se torna repetitivo, si se detectan inconsistencias en el sistema implicando el retorno de cualquiera de las fases anteriores para su corrección.

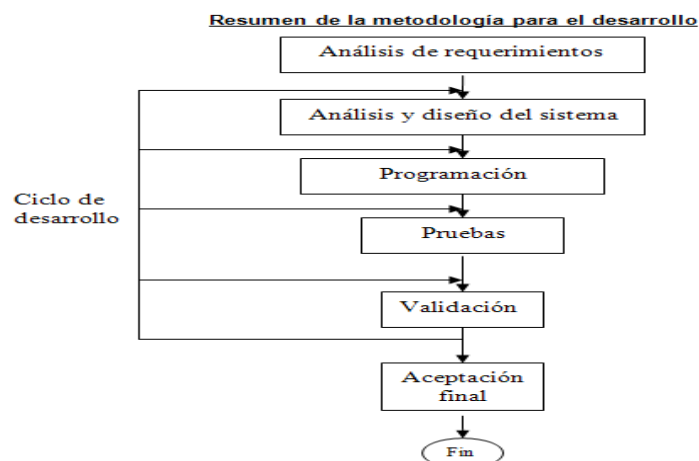


Figura 4. Resumen de la Metodología para el Desarrollo

1.3.3 Resultados Esperados

Con el Sistema de Gestión Académica para el INCOS se pueden realizar las operaciones de la mejor manera, donde los estudiante no esperen mucho tiempo para poder obtener información sobre los reportes de notas de sus distintas materias, como también los docentes podrán manejar de una mejor manera las notas de las distintas materias que dictan.

En la institución se pretende minimizar posibles errores de cálculo de las notas para esto se propone generar reportes generales del cálculo exacto de las mismas.

1.3.4 Transferencia de Resultados

1.3.4.1 Medios y estrategias para la transferencia de Resultados

Presentación final del sistema informático al jefe académico del Instituto Comercial Superior INCOS Tarija.

Entrega de instaladores del sistema informático y la documentación desarrollada en el proyecto.

Socialización del producto final con el personal involucrado.

1.3.5 Grupo de Beneficiarios de Resultados

Estudiantes

Son atendidos de manera Rápida y Eficaz, aparte de contar con el Sistema que permitirá acceder, informarse y consultar las notas de todas sus materias.

Jefe Académico

Le es más fácil el seguimiento día a día de la institución, dejando de usar tanto papeleo.

Docente

Pueden realizar el manejo de las notas de sus materias de manera fácil y sencilla, sin necesidad de realizar cálculos mentales y/o manuales evitando discrepancias.

1.3.6 Matriz de Marco Lógico

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir al fortalecimiento de la Gestión Académica de información del Instituto Comercial Superior(INCOS)Tarija	Al finalizar el proyecto se cuenta con información completa en el 66% de los procesos relacionados con Gestión Académica del Instituto Comercial Tarija. $\text{For} = (\text{procesos automatizados} * 100\%) / \text{Total Procesos}$ $\text{Formula} = (12 * 100\%) / 18 = 66,66\%$	Desarrollo de encuestas al personal para determinar el fortalecimiento en la actividad administrativa, avalado por el propietario de la empresa. Desarrollo de encuestas al personal para determinar el fortalecimiento de la actividad administrativa y la eficiencia y eficacia al momento de brindar información.	El apoyo del jefe Académico del Instituto Comercial Superior se mantiene brindando la suficiente información para la ejecución del Proyecto. Existe voluntad y recurso de sostenibilidad del proyecto a lo largo del tiempo

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Objetivo General (Propósito) Administración Académica del Instituto Comercial Superior Tarija mejorada.	Al finalizar el proyecto la Administración Académica del Instituto Comercial Superior Tarija con relación a gestiones pasadas mejoraron al menos en un 45% para los docentes(con criterio de mejora) Al finalizar el proyecto tanto estudiantes como docentes del Instituto Comercial Superior Tarija expresan su conformidad con la información disponible y los medios de acceso a la misma.	Informe avalado por el jefe académico que certifica que el 45% de los procesos han sido automatizados satisfactoriamente. Se cuenta con copias de reportes generados con el Sistema Informático avalando la mejora y la rapidez en un 45% en la obtención de la información.	Existe apoyo de Jefe Académico para brindar la información de forma confiable y oportuna. El personal del Instituto Comercial Superior Tarija apoya de forma constante al proyecto.

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Objetivos Específicos (Componentes) 1. Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija desarrollado 2. Capacitación del uso de los recursos TIC y el manejo del Sistema	Al finalizar el proyecto en la gestión 2013 El sistema Informático automatizo el 66% de las actividades de administración académica del Instituto Comercial Superior Tarija completando las fases de análisis diseño junto con el desarrollo del sistema desde el 15 Mayo hasta el 12 de Noviembre de 2013. El curso de capacitación y socialización para el manejo del Sistema Informático para	Certificado de conformidad con el Sistema Informático firmado por el Ing. Carlos Quiñones Jefe Académico del Instituto Comercial Superior Tarija Documentación de Análisis y Diseño del Sistema Informático revisado por los docentes de la materia de Taller III Certificado de conformidad del Instituto Comercial	Jefe Académico del Instituto Comercial Superior Tarija brindo información oportuna para la realización del sistema. Disponibilidad de recursos y herramientas de software y hardware para el desarrollo del Sistema Informático. Predisposición del personal del Instituto Comercial

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija desarrollado.	la administración Académica del Instituto Comercial Superior (INCOS)Tarija, se realizó el 10 de Noviembre del 2013	Superior (INCOS) Tarija de la participación en la capacitación sobre el manejo del Sistema Informático para la administración Académica. Guía de Capacitación (Manual de Usuario). Presentación del material didáctico usado en la capacitación y socialización al personal de la empresa. Planilla con firma de las personas capacitadas.	Superior en participar en la capacitación acerca del Sistema Informático. El personal es receptivo a la nueva tecnología incorporada
Actividades I. Fase de Análisis y	Se tienen presupuestado un	Información del	Desembolso efectuado de

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Diseño del Sistema Informático II. Fase de Desarrollo del Sistema Informático III. Fase de Pruebas del Sistema Informático IV. Presentación del Sistema V. Elaboración de material de apoyo para el uso del software elaborado (<i>Guías de usuario</i>) VI. Cursos de Capacitación	monto de 6290 \$us., para llevar a cabo el proyecto, el cual un 80 % es para el desarrollo del Sistema Académico Web y un 20% para realizar la capacitación del personal de Instituto Comercial Superior Tarija.	presupuesto de gastos del equipo proyectista.	acuerdo a lo planificado. Disponibilidad de herramientas para realizar el desarrollo del sistema. Apoyo por parte de los trabajadores de la empresa.

Tabla 4. Marco Lógico del Proyecto

CAPÍTULO II

CAPÍTULO II: COMPONENTES

2.1 COMPONENTE 1: SISTEMA INFORMÁTICO PARA EL MEJORAMIENTO DE LA ADMINISTRACIÓN ACADÉMICA INCOS.

2.1.1 Antecedentes

INCOS es el Instituto Líder en Formación Superior, inició sus actividades el 1 de enero de 1976 en la ciudad de Tarija, con la fe puesta en un país que le abría sus puertas y cuya necesidad más grande es la formación de profesionales técnicos en todas las áreas.

INCOS es una institución que cree firmemente que BOLIVIA necesita para su desarrollo "Técnicos" y por ello su misión es la formación de profesionales competitivos, prácticos y conscientes de que su esfuerzo contribuirá al progreso de todos.

INCOS forma las generaciones de HOY y no del mañana.....forma profesionales que podrán ejercer la carrera que estudiaron en muy corto plazo y en la mayor parte de los casos antes que concluyan su carrera.

Todos sabemos que la educación Técnica Superior es esencial para la existencia de todos los países del mundo. Sin este tipo de formación no podríamos llegar nunca a la madurez como nación, ni se podría entender el desarrollo, tampoco responder a los retos crecientes de la organización, la sociedad, la economía y la cultura.

La educación es mucho más que el aprendizaje o que la enseñanza, es creadora de cultura y tecnología. INCOS tiene la firme convicción de seguir sirviendo a nuestro país en la formación de profesionales dinámicos y competitivos con espíritu joven de una Bolivia dispuesta a ganar y a superarse cada día más. INCOS es un instituto a nivel superior con más de 35 años de experiencia, cuenta con unidades académicas en las ciudades de: La Paz, Tarija, Cochabamba y Santa Cruz.

- **Misión**

Formar profesionales técnicos y tecnólogos, que posean conocimientos, habilidades, actitudes y valores que les permita insertarse en un mercado laboral altamente competitivo con la menor dificultad posible y contribuyan al mejoramiento de las condiciones de vida de la región y el país.

- **Visión**

El INCOS TARIJA proyecta en su visión hacia el 2015 como una red de Instituciones de Educación Superior que forman profesionales para la vida con capacidad emprendedora y comprometida con el desarrollo tecnológico de su entorno.

- **Objetivo**

Consolidarnos como el mejor instituto de enseñanza en Tarija como una institución reconocida por la calidad, en la formación de nuevos profesionales brindarle así a los estudiantes más posibilidades de ejercer su profesión debido al prestigio de nuestra institución.

2.1.2 Objetivo General

Analizar, Diseñar y Desarrollar un sistema automatizado que ayude a controlar los registros de Notas del Instituto.

2.1.2.1 Objetivos Específicos

- Recoger la mayor cantidad de información, con la que podamos entender la manera de registro de las notas, que realiza la administración académica del Instituto Comercial Superior INCOS.
 - Analizar toda la información recolectada para adquirir conocimiento sobre la institución y así poder determinar los requerimientos que cubre el sistema.
 - Diseñar y estructurar el sistema de forma que este cumpla todos los requerimientos identificados.
 - Desarrollar el sistema tomando en cuenta las tecnologías que mejor se adapten a su propósito.
- Ejecutar pruebas que garanticen la calidad del sistema e identifiquen errores a corregir antes de su implementación.

2.1.2.2 Matriz Foda

El FODA se representa a través de una matriz de doble entrada, llamada matriz FODA, en la que en el nivel horizontal se analiza los factores positivos y los negativos.

En la lectura vertical se analizan los factores internos y por tanto controlables del programa o proyecto y los factores externos, considerados no controlables.

- Las Fortalezas son todos aquellos elementos internos y positivos que diferencian al programa o proyecto de otros de igual clase.
- Las Oportunidades son aquellas situaciones externas, positivas, que se generan en el entorno y que una vez identificadas pueden ser aprovechadas.
- Las Debilidades son problemas internos, que una vez identificadas, se desarrollan adecuadas estrategias, pueden desaparecer y luego eliminarse.

- Las Amenazas son situaciones negativas, externas al programa o proyecto, que pueden atentar contra éste, por lo que llegado al caso, puede ser necesario diseñar una estrategia adecuada para poder sortearla.



Figura 1. Matriz FODA

2.1.2.3 Diagnóstico Estratégico

El análisis realizado permite detectar que INCOS se encuentra compitiendo en un mercado complejo ya que la necesidad que satisface la institución es susceptible a diversos sustitutos, convirtiéndose esto en una falencia de la institución.

El análisis es útil para que la institución conozca qué características deba tomar en cuenta para afrontar estratégicamente y cuáles son sus principales fortalezas que motiven a la institución seguir adelante y mejorar, optimizar y estandarizar los servicios y productos que ofrece.

Es importante destacar que el ritmo de vida acelerado de las personas favorece a este tipo de institución que por la rapidez de conseguir una carrera a nivel técnico en corto tiempo aprovecharan el servicio que ofrece el INCOS.

Es imprescindible que la institución cuente con personal humano calificado que le permita desarrollar sus actividades de la mejor manera logrando así la satisfacción plena de quienes confían en los servicios que el INCOS ofrece.

Además que el Know-How de la institución para la elaboración de los servicios le permite crear una ventaja competitiva que se transforma en la percepción que las personas tiene de la institución.

2.1.3 Unidad De Gestión

Debido a la importancia que representa el sistema para el actual proyecto se utilizó las siguientes unidades para facilitar el desarrollo y garantizar la calidad del Sistema:

- **Planificación y control de etapas del proyecto:** Para realizar la planificación de fases se utiliza la herramienta:

- **Microsoft Project:** Para determinar el calendario que se seguirá durante el desarrollo del proyecto teniendo así un mayor control del tiempo en el avance del proyecto.

- **Determinación de requerimientos:** Para realizar la identificación de los requerimientos, del sistema se requirió hacer uso de metodologías en este caso se utilizarán entrevistas y cuestionarios.

Además se utilizará la herramienta **Enterprise Architect 7.5** para la representación de los casos de uso del sistema y sus funcionalidades.

- **Diseño:** Se requerirá hacer uso de herramientas como:
 - **Enterprise Architect 7.1** para la representación de los diagramas de: casos de uso, actividad, clases, secuencia.
 - **Bootstrap** como apoyo para el diseño web adaptable de las interfaces HTML.
 - **Macromedia Dreamweaver** para el diseño las interfaces HTML.
 Esto permitirá representar todos los procesos identificados en el análisis, que sean fáciles de entender para el usuario.

- **Programación:** Se utilizarán de las siguientes herramientas:
 - **Modelo Vista Template** como el framework de trabajo.
 - **Tornado Web Server** que será el servidor del sitio web.
 - **Sublime Text** será el IDE para poder editar código Python que será el lenguaje de programación utilizado para la Pagina Web.
 - **PostgreSQL 9.0** para la base de datos implementada.
 - **Adobe Dreamweaver** para editar código HTML.
 - **Weasyprint** para elaborar las plantillas que generarán los reportes.

También se utilizará diferentes librerías que permitirán obtener información de la base de datos, además también Ponyorm, para la parte del mapeo de las clases de python, con la base de datos.

2.1.4 Metodología de Desarrollo

2.1.4.1 Metodología RUP(Racional Unified Process)

a) Definición, **RUP** es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Es un proceso de Ingeniería de Software que captura las mejores prácticas del conocimiento de líderes en Ingeniería de Software y que provee un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo.

Los **procesos de RUP**, estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos por RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme.

Nos permite realizar un levantamiento exhaustivo de requerimientos.

Las actividades de RUP se centran en crear y mantener modelos, utilizando UML, en forma efectiva.

- Busca detectar defectos en las fases iniciales.

- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el Análisis y Diseño, tan completo como sea posible.
- Diseño genérico, intenta anticiparse a futuras necesidades.
- Las necesidades de clientes no son fáciles de discernir.
- Existe un contrato prefijado con los clientes.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.

b) Características esenciales, Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- **Está dirigido por los Casos de Uso:** Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

Los Casos de Uso, no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba.

- **Está Centrado en su Arquitectura:** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que

indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo del software.

- **Es Iterativo e Incremental:** Según el equilibrio correcto entre los Casos de Uso y la Arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y Arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales)

del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

c) Fases en el ciclo de Desarrollo, Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases que se describirán a continuación, la figura muestra las Fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cuál de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

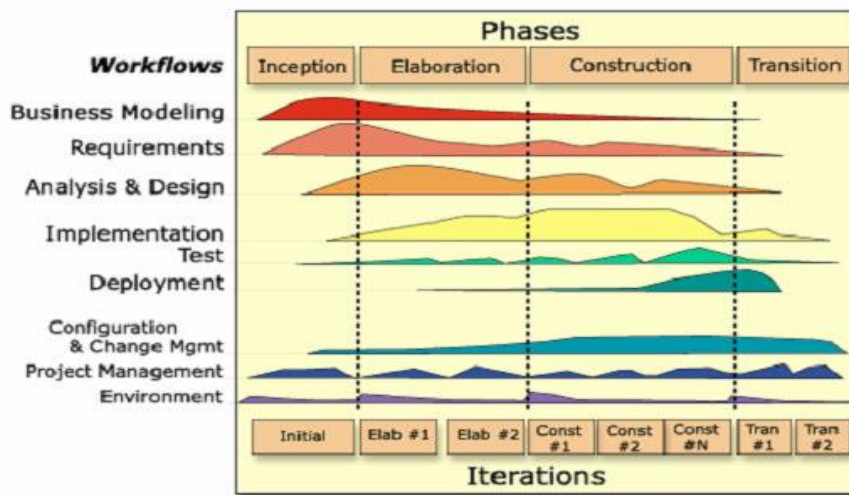


Figura 2. Fases en el Ciclo de Desarrollo de R.U.P.

- **Fase 1: Preparación Inicial (“Incepción”)**

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

El caso de negocio incluye criterios de éxito, la evaluación de riesgos, y la estimación de los recursos necesarios, y un plan de la fase que muestre las fechas previstas e hitos importantes.

- **Fase 2: Preparación Detallada (“Elaboración”)**

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

El resultado de la fase de elaboración es:

Un modelo de caso de uso (por lo menos 80% completo) - todos los casos de uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de casos de uso.

Requerimientos suplementarios que capturan los requerimientos no funcionales o cualquier requerimiento que no se asocie a un caso de uso específico.

- **Fase 3: Construcción (“Construcción”)**

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

- **Fase 4: Transición (“Transición”)**

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

Éste incluye:

- Operación en paralelo con un sistema anterior que el nuevo sistema esté sustituyendo.
- La conversión de las bases de datos operacionales.
- Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

2.1.4.2 UML (Lenguaje Unificado de Modelado).

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Este lenguaje, nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las Metodologías de desarrollo.

a) Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- 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.

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para

modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

b) Un modelo UML está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

c) UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

d) UML ofrece notación y semántica estándar:

UML prescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, es útil categorizarlos jerárquicamente.

Los **Diagramas de Estructura** enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases, representan la estructura estática en términos de clases y relaciones.
- Diagrama de componentes, representan los componentes físicos de una aplicación.
- Diagrama de objetos, representan los objetos y sus relaciones, corresponden a diagramas de colaboración simplificados sin la representación del envío de mensajes.
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue, representan el despliegue de los componentes sobre los dispositivos físicos.
- Diagrama de paquetes, muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones.

Los **Diagramas de Comportamiento** enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades, representan el comportamiento del sistema en términos de acciones.
- Diagrama de casos de uso, representan funcionalidad del sistema desde el punto de vista del usuario.
- Diagrama de estados, representan el comportamiento de una clase en término de estados.

Los **Diagramas de Interacción** son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia, son una representación temporal de los objetos y sus interacciones.

- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x), son una representación espacial de los objetos, enlaces e interacciones entre ellos.
- Diagrama de tiempos (UML 2.0).
- Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0).

2.1.4.2.1 Tipos de Diagramas Utilizados

2.1.4.2.1.1 Diagramas de Clases

Es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro.

2.1.4.2.1.2 Diagramas de Casos de Uso

Captura las interacciones de los casos de uso y los actores. Describe los requisitos funcionales del sistema, la forma en la que las cosas externas (actores) interactúan a través del límite del sistema y la respuesta del sistema.

2.1.4.2.1.3 Diagramas de Actividades

Que se usa para modelar el comportamiento de un sistema, y la manera en que este comportamiento está relacionado con un flujo global del sistema. Se usan los caminos lógicos que sigue un proceso basado en varias condiciones, concurrencia en el proceso, los datos de acceso, interrupciones y otras alternativas del camino lógico para construir un proceso, sistema o procedimiento.

2.1.4.2.1.4 Diagramas de Secuencias

Es una representación estructurada del comportamiento como una serie de pasos secuenciales a lo largo del tiempo. Se usa para representar el flujo de trabajo, el paso

de mensajes y cómo los elementos en general cooperan a lo largo del tiempo para lograr un resultado.

2.1.4.2.1.5 Diagrama de Componentes

Ilustra los fragmentos de software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de componentes tiene un nivel de abstracción más elevado que un diagrama de clase - usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como así eventualmente un componente puede comprender una gran porción de un sistema.

2.1.5 Herramientas de Construcción de Software

2.1.5.1 Modelo Vista Template

Básicamente los 3 componentes principales del framework tienen como función crear una comunicación entre ellos para extraer información vital de base de datos y presentarlas en el navegador

Model.- El modelo tiene como objetivo mapear a la base de datos de tal forma que crea una sincronización entre la base de datos y la aplicación, con el fin de mantener actualizada toda la información de las tablas, campos, y datos de nuestra base

Vista.- La vista tiene como objetivo recibir el requerimiento que es enviado por el navegador, procesar la información, si es necesario realizar una petición al modelo para extraer un valor de base de datos u otro de los casos podría ser que sólo muestre un mensaje enviando un requerimiento al template.

Template.- Es básicamente un archivo HTML que tiene como objetivo mostrar el pantalla de forma amigable la respuesta que es enviada por la vista

Url Dispatch.- Este es un componente adicional al framework pero a su vez uno de los más importantes porque es el encargado de recibir el requerimiento de una url escrita por el cliente y enviarla a su respectiva vista para procesar dicha información

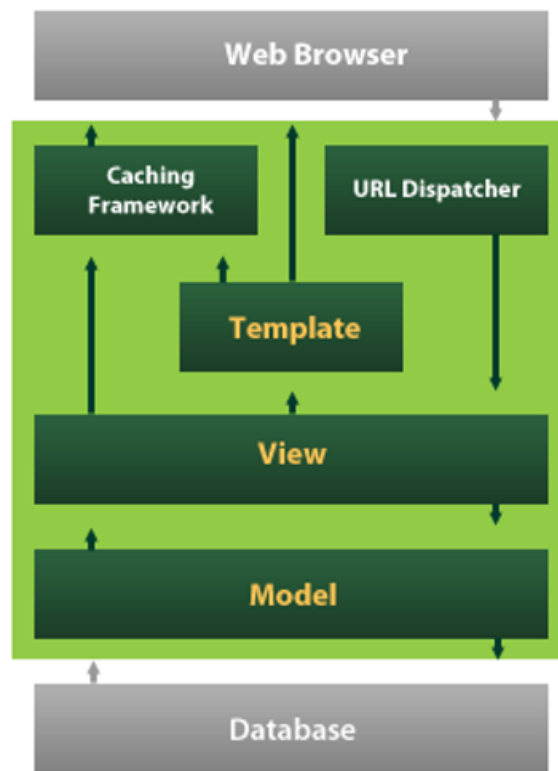


Figura 3. Modelo Vista Template

2.1.5.2 Sublime Text

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero él no disponer de ésta no genera ninguna limitación más allá de una alerta cada cierto tiempo.

Características

- **Minimapa:** consiste en una previsualización de la estructura del código, es muy útil para desplazarse por el archivo cuando se conoce bien la estructura de este.
- **Multi Selección:** Hace una selección múltiple de un término por diferentes partes del archivo.
- **Multi Cursor:** Crea cursores con los que podemos escribir texto de forma arbitraria en diferentes posiciones del archivo.
- **Multi Layout:** Trae siete configuraciones de plantilla podemos elegir editar en una sola ventana o hacer una división de hasta cuatro ventanas verticales o cuatro ventanas en cuadrícula.
- **Soporte nativo para infinidad de lenguajes:** Soporta de forma nativa 43 lenguajes de programación y texto plano.
- **Syntax Highlight** configurable: El remarcado de sintaxis es completamente configurable a través de archivos de configuración del usuario.
- **Búsqueda Dinámica:** Se puede hacer búsqueda de expresiones regulares o por archivos, proyectos, directorios, una conjunción de ellos o todo a la vez.
- **Auto completado y marcado de llaves:** Se puede ir a la llave que cierra o abre un bloque de una forma sencilla.
- **Soporte de Snippets y Plugins:** Los snippets son similares a las macros o los bundles además de la existencia de multitud de plugins.
- **Configuración total de Keybindings:** Todas las teclas pueden ser sobrescritas a nuestro gusto.
- **Acceso rápido a línea o archivo:** Se puede abrir un archivo utilizando el conjunto de teclas Cmd+P en Mac OS X o Ctrl+P en Windows y Linux y escribiendo el nombre del mismo o navegando por una lista. También se puede ir a una línea utilizando los dos puntos ":" y el número de línea.

- **Paleta de Comandos:** Un intérprete de Python diseñado solo para el programa con el cual se puede realizar infinidad de tareas.
- **Coloreado y envoltura de sintaxis:** Si se escribe en un lenguaje de programación o marcado, resalta las expresiones propias de la sintaxis de ese lenguaje para facilitar su lectura.
- **Pestañas:** Se pueden abrir varios documentos y organizarlos en pestañas.
- **Resaltado de paréntesis e indentación:** cuando el usuario coloca el cursor en un paréntesis, corchete o llave, resalta esta y el paréntesis, corchete o llave de cierre o apertura correspondiente.



Figura 4. Sublime Text

2.1.5.3 Hojas de estilo CSS3

El nombre hojas de estilo en cascada viene del inglés *Cascading Style Sheets*, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un

documento de su *presentación*.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo.

A graphic of a document with a folded corner, containing CSS code. Below the document is an orange label with the text 'CSS' in white.

```
body {  
  margin: 4px;  
  border: 3px dotted #  
  font-family: sans-serif;  
  color: #000000;  
  background-color: #FFFFFF;  
}  
  
h1 {  
  padding: 5px;  
  margin: 10px;  
  border: 1px solid #C0C0C0;  
  color: #FF0000;  
  background-color: #0000FF;  
}
```

Figura 5. Hojas de estilo CSS3

2.1.5.4 JQuery

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

- Seguridad

El código será más fácil de mantener, eliminando ambigüedades y permitiendo una escritura más clara y concisa.

- **Simplicidad**

El lenguaje será lo suficientemente intuitivo como para que el programador pueda escribir código sin tener que consultar el manual de referencia constantemente.

- **Performance**

El lenguaje habilitará a los programadores para escribir complejos programas que funcionen de forma más eficiente y responsable.

- **Compatibilidad**

Como dialecto de la especificación ECMAScript, el lenguaje estará provisto de una mayor compatibilidad con los estándares de la industria. Se pretende lograr un lenguaje más coherente y unificado.

2.1.5.5 Enterprise Architect for UML

Es una herramienta “modelado visual” de UML. La herramienta está diseñada para una gama amplia de usuarios como ser ingenieros de software, analistas de sistemas, analistas comerciales y arquitectos de sistemas o para cualquiera que está interesado en construir sistemas, software de gran potencia que usan un acercamiento orientado a objeto fiablemente.

Además, VP-UML apoya las últimas normas de anotación de UML.



Figura 6. Enterprise Architect 7.5 for UML

2.1.5.6 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Características y paradigmas

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.



Figura 7. Python

2.1.5.7 PostgreSQL

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como:

- Linux
- UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)
- Windows

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos del SQL 2008, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C ++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros.



Figura 8. PostgreSQL

2.1.6 Técnica

2.1.6.1 Tornado Web Server

Tornado es un framework web de Python y la biblioteca de red asincrónica, desarrollado originalmente en FriendFeed . Mediante el uso de la red no-bloqueo de E / S, Tornado puede escalar a decenas de miles de conexiones abiertas, por lo que es ideal para el sondeo de largo , WebSockets , y otras aplicaciones que requieren una conexión de larga duración para cada usuario.

Tornado se pueden dividir en cuatro componentes principales:

- Un framework web (incluyendo RequestHandler que es una subclase para crear aplicaciones web, y diversas clases de apoyo).
- Implementations cliente-y del lado del servidor de HTTP (HTTPServer y AsyncHTTPClient).
- Una biblioteca de red asíncrona (IOLoop y iostream), que sirven como bloques de construcción para los componentes de HTTP y también se puede utilizar para implementar otros protocolos.
- Una biblioteca de co-rutina (tornado.gen) que permite que el código asíncrono a ser escrito de una manera más sencilla que el encadenamiento de las devoluciones de llamada.

El servidor web framework y HTTP Tornado junto ofrecer una alternativa-stack completo a WSGI .Si bien es posible utilizar el framework web Tornado en un recipiente WSGI (WSGIAdapter), o utilizar el servidor HTTP Tornado como contenedor de otros marcos WSGI (WSGIContainer), cada una de estas combinaciones tiene limitaciones y sacar el máximo provecho de Tornado se quiere necesitará utilizar framework web del Tornado y el servidor HTTP juntos.



Figura 9. Tornado Servidor Web

2.1.6.2 Bootstrap

Bootstrap es una libre colección de herramientas para la creación de sitios web y aplicaciones web. Contiene HTML y CSS plantillas de diseño basadas en la tipografía, formas, botones, navegación y otros componentes de la interfaz, así como opcionales JavaScript extensiones.

Características

Bootstrap es compatible con las últimas versiones de los principales navegadores. Se degrada con gracia cuando se utiliza en navegadores antiguos como Internet Explorer 8.

Desde la versión 2.0 también es compatible con el diseño web de respuesta . Esto significa que el diseño de páginas web ajusta dinámicamente, teniendo en cuenta las características del dispositivo utilizado (de escritorio, tableta, teléfono móvil). Desde la versión 3.0, Bootstrap adoptó un primer móvil filosofía de diseño, haciendo hincapié en el diseño de respuesta por defecto.

Estructura y función

Bootstrap es modular y se compone esencialmente de una serie de hojas de estilo menos que implementan los diversos componentes de la caja de herramientas. Una

hoja de estilo llamada `bootstrap.less` incluye las hojas de estilo de componentes. Los desarrolladores pueden adaptar el Bootstrap propio archivo, la selección de los componentes que se desean utilizar en su proyecto.

Los ajustes son posibles de forma limitada a través de una hoja de estilos de configuración central. Más cambios profundos son posibles por las declaraciones LESS.

El uso de un lenguaje menos hoja de estilo permite el uso de variables, funciones y operadores, selectores anidados, así como los llamados mixins.

Desde la versión 2.0, la configuración de Bootstrap también tiene una opción especial "Personalizar" en la documentación. Por otra parte, el desarrollador elige en un formulario los componentes deseados y se ajusta, si es necesario, los valores de las distintas opciones a sus necesidades. El paquete generado posteriormente ya incluye la hoja de estilos CSS pre-construidos.

Sistema de red y el diseño de respuesta viene de serie con un diseño de 1170 píxeles de ancho, rejilla. Como alternativa, el desarrollador puede utilizar un diseño de anchura variable. Para ambos casos, el conjunto de herramientas tiene cuatro variaciones para hacer uso de diferentes resoluciones y tipos de dispositivos: teléfonos móviles, retrato y el paisaje, tablets y PCs con baja y alta resolución. Cada variación ajusta el ancho de las columnas.

La hoja de estilos CSS

Bootstrap proporciona un conjunto de hojas de estilo que proporcionan definiciones de estilo básicas para todos los componentes HTML clave. Estos proporcionan una apariencia uniforme y moderna para dar formato a texto, tablas y elementos de formulario.

Reutilizables componentes

Además de los elementos HTML regulares, Bootstrap contiene otros elementos de la interfaz de uso común. Estos incluyen botones con funciones avanzadas (por ejemplo, la agrupación de botones o teclas opción desplegable con, hacer y listas de navegación, pestañas horizontales y verticales, la navegación, la ruta de navegación, paginación, etc), etiquetas, capacidades tipográficas avanzadas, miniaturas, los mensajes de advertencia y una barra de progreso.

Componentes JavaScript

Bootstrap viene con varios componentes JavaScript en la forma de jQuery plugins. Proporcionan elementos adicionales de interfaz de usuario, tales como cuadros de diálogo, información sobre herramientas, y carruseles. También amplían la funcionalidad de algunos elementos de la interfaz existentes, incluyendo por ejemplo una función de autocompletado para los campos de entrada. En la versión 2.0, los siguientes complementos de JavaScript son compatibles: Modal, desplegable, Scrollspy, Tab, Tooltip, Popover, Alerta, Button, Collapse, Carousel y TYPEAHEAD.



Figura 10. Bootstrap

2.1.6.3 Diseño Web Adaptable

El diseño web adaptable o adaptativo, conocido por las siglas RWD (del inglés, Responsive Web Design) es una filosofía de diseño y desarrollo cuyo objetivo es

adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla. Hoy en día las páginas web se visualizan en multitud de tipos de dispositivos como tabletas, smartphones, libros electrónicos, portátiles, PCs,... Además, aún dentro de cada tipo, cada dispositivo tiene sus características concretas: Tamaño de pantalla, resolución, potencia de CPU, capacidad de memoria,... Esta tecnología pretende que con un solo diseño web tengamos una visualización adecuada en cualquier dispositivo.

El diseñador y autor norteamericano Ethan Marcotte creó y difundió esta técnica a partir de una serie de artículos en A List Apart, una publicación en línea especializada en diseño y desarrollo web, idea que luego extendería en su libro Responsive Web Design.

El uso de dispositivos móviles está creciendo a un ritmo increíble, dispositivos como tablets y smartphones han incrementado sus ventas en los últimos años y la navegación en Internet mediante estos dispositivos es cada vez más común. Ese es el motivo por el que el diseño web adaptable se ha vuelto tan popular, pues es una técnica que proporciona una solución web que puede manejar la visualización web tanto de escritorio como de dispositivos.

Con una sola versión en HTML y CSS se cubren todas las resoluciones de pantalla, es decir, el sitio web creado estará optimizado para todo tipo de dispositivos: PCs, tabletas, teléfonos móviles, etc. Esto mejora la experiencia de usuario a diferencia de lo que ocurre, por ejemplo, con sitios web de ancho fijo cuando se acceden desde dispositivos móviles.

De esta forma se reducen los costes de creación y mantenimiento cuando el diseño de las pantallas es similar entre dispositivos de distintos tamaños.

También se supone que evita tener que desarrollar aplicaciones ad-hoc para versiones móviles, o no, por ejemplo, una aplicación específica para iPhone, otra para móviles Android, etc., aunque hoy en día las webs para móviles todavía no pueden realizar las mismas funciones que las aplicaciones nativas.

Desde el punto de vista de la optimización de motores de búsqueda, sólo aparecería una URL en los resultados de búsqueda, con lo cual se ahorran redirecciones y los fallos que se derivan de éstas. También se evitarían errores al acceder al sitio web en concreto desde los llamados "social links", es decir, desde enlaces que los usuarios comparten en medios sociales tales como Facebook, Twitter, etc y que pueden acabar en error dependiendo de qué enlace se copió (desde qué dispositivo se copió) y desde qué dispositivo se accede.⁶

Funcionamiento del Diseño Web Adaptable

El Responsive Web Design se hace posible gracias a la introducción de las Media Queries en las propiedades de los Estilos CSS en su versión número 3. Las media queries son una serie de órdenes que se incluyen en la hoja de estilos que indica al documento HTML cómo debe comportarse en diferentes resoluciones de pantalla.

Para entenderlo mejor, los diseños de las Páginas Web, al igual que los periódicos y las revistas, están basados en columnas, entonces con la filosofía del Diseño Adaptativo, si una web a resolución de PC (1028x768 px) tiene 5 columnas, para una Tablet (800x600 px) necesitaría sólo 4, y en el caso de un smartphone cuyo ancho suele ser entre 320 px y 480 px las columnas usadas serían 3.

2.1.6.4 Weasyprint

WeasyPrint es un motor de renderizado visual de HTML y CSS que se pueden exportar a PDF. Su objetivo es apoyar los estándares web para la impresión usando Python como lenguaje de programación. También permite crear gráficos en bitmap y vectores.



Figura 11. Weasyprint

2.1.6.5 Pony ORM

El acrónimo ORM significa "mapeador objeto-relacional". Un ORM permite a los desarrolladores trabajar con el contenido de una base de datos en forma de objetos. Una base de datos relacional contiene filas que se almacenan en tablas. Sin embargo, al escribir un programa en un lenguaje orientado a objetos de alto nivel, es mucho más conveniente cuando los datos recuperados de la base de datos se puede acceder en forma de objetos. Pony ORM es una biblioteca para el lenguaje Python que te permite trabajar cómodamente con los objetos que se almacenan como filas de una base de datos relacional.

Tiene las siguientes ventajas:

- Una sintaxis excepcionalmente conveniente para escribir consultas
- La optimización automática de consultas
- Una solución elegante para el problema de N + 1
- Un editor gráfico de esquema de base de datos



Figura 12. Pony ORM

2.1.6.6 Base de Datos

2.1.6.6.1 Componentes principales de una Base de Datos

- Datos: Los datos son la Base de Datos propiamente dicha.
- Hardware: Se refiere a los dispositivos de almacenamiento en donde reside la base de datos así como los dispositivos periféricos (Unidad de Control, Canales de Comunicación, etc.) necesarios para su uso.
- Software: Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DBMS), manejando éste todas las solicitudes formuladas por los usuarios a la base de datos.
- Usuarios: Normalmente identificándose 3 tipos:
 - El programador de aplicaciones.
 - El usuario Final.

El Administrador de la Base de Datos quien se encarga del control general del Sistema de Base de Datos.

2.1.6.6.2 Herramienta PgAdmin para el manejo de la BD

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como PervasivePostgres, EnterpriseDB, MammothReplicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad.

2.1.6.7 El Internet

Algunos definen Internet como "La Red de Redes", y otros como "La Autopista de la Información".

Efectivamente, Internet es una Red de Redes porque está hecha a base de unir muchas redes locales de ordenadores, o sea de unos pocos ordenadores en un mismo edificio o empresa.

Por la Red Internet circulan constantemente cantidades increíbles de información. Por este motivo se le llama también La Autopista de la Información. Hay 50 millones de "Internautas", es decir, de personas que "navegan" por Internet en todo el Mundo. Se dice "navegar" porque es normal el ver información que proviene de muchas partes distintas del Mundo en una sola sesión.

Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras. Incluso podemos ver conectados a la Red cámaras de vídeo, robots, y máquinas de refrescos, etc.

2.1.6.7.1 La World Wide Web

La World Wide Web consiste en ofrecer una interface simple y consistente para acceder a la inmensidad de los recursos de Internet. Es la forma más moderna de ofrecer información. El medio más potente. La información se ofrece en forma de páginas electrónicas.

El World Wide Web o WWW o W3 o simplemente Web, permite saltar de un lugar a otro en pos de lo que no interesa. Lo más interesante es que con unas pocas ordenes se puede mover por toda la Internet.

La World Wide Web permite una manera más organizada de acceder a la información disponible en Internet, presentando una interfaz amigable con el usuario mediante navegadores como Netscape, Mosaic y Microsoft Internet Explorer, Mozilla FireFox.

El surgimiento de la World Wide Web ha ayudado a un crecimiento considerable de Internet en la actualidad. Compañías pequeñas, empresas grandes, ayuntamientos, estados, gobiernos de distintos países, universidades, bibliotecas, están presentes en Internet.

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

2.1.6.7.2 Sistemas de Información vía Web

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo del Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los

servicios existentes en Internet. Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Pudiésemos definir entonces como sistema de información al conjunto de elementos relacionados y ordenados, según ciertas reglas que aporta al sistema objeto, es decir, a la organización a la que sirve y que marca sus directrices de funcionamiento, la información necesaria para el cumplimiento de sus fines; para ello, debe recoger, procesar y almacenar datos, procedentes tanto de la organización como de fuentes externas, con el propósito de facilitar su recuperación, elaboración y presentación. Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información.

2.1.7 Plan De Desarrollo Del Software

2.1.7.1 Introducción

Este Plan de Desarrollo del Software es una versión corregida para ser incluida en la propuesta elaborada como respuesta al proyecto de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto fue desarrollado por el universitario: Wilson Luis Narvaez Michel basado en una metodología de Rational Unified Process (RUP) en la que se procederá a cumplir con las cuatro fases que marca la metodología. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración, adicionalmente se esbozarán las fases

posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque del desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

2.1.7.2 Propósito

El propósito del Plan de Desarrollo de Software es dar a conocer la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El Director del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.
- El docente para evaluar el cumplimiento del proyecto

2.1.7.3 Alcance

Con el Plan de Desarrollo del Software se pretende analizar y elaborar un proyecto de gran magnitud abarcando todas las fases requeridas en la terminología RUP. Dichas fases tendrán un cronograma de cada una de las actividades a realizar. También se especificará los detalles de construcción del proyecto para los distintos roles que cumplen los desarrolladores.

2.1.7.4 Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

- Vista General del Proyecto, proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.
- Organización del Proyecto, describe la estructura organizacional del equipo de desarrollo.
- Gestión del Proceso, explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.
- Planes y Guías de aplicación, proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

2.1.7.5 Vista General del Proyecto

2.1.7.5.1 Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones, entrevistas y charlas que se han realizado con los usuarios destino del Instituto Comercial Superior INCOS Tarija.

2.1.7.5.1.1 Propósito

Con el presente proyecto se pretende el mejoramiento de la Administración Académica del Instituto Comercial Superior (INCOS) Tarija.

2.1.7.5.1.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo de INCOS un Sistema Informático que está comprendido por los siguientes Módulos:

- Modulo Jefe Académico.
- Modulo Estudiante.
- Modulo Docente.
- Modulo Secretaria

Analizando factores predominantes dentro de la institución, podemos llegar a la conclusión de que la implementación del sistema automatizado dará beneficios claramente perceptibles, dando soluciones a problemas que arrastra; este sistema proyectará una solución a mediano plazo que beneficiará a dicha institución, pero lo más importante mejorará el sistema actual.

Esta propuesta de sistema (Software) contiene una serie de alternativas de mejoramiento para las expectativas futuras de la institución, las cuales se detallan a continuación:

- Brindar seguridad al sistema mediante una clave de ingreso, permitiendo el acceso al mismo sólo al personal autorizado.
- Desarrollar un manual de usuario y de instalación del sistema.

2.1.7.5.1.3 Objetivos

2.1.7.5.1.3.1 Objetivo General

Mejorar el registro de la información referente al manejo de notas por medio del Sistema Informático para la Administración Académica del Instituto Comercial Superior (INCOS) Tarija.

2.1.7.5.1.3.2 Objetivo Específicos

- Obtener un sistema para el uso en la Administración Académica de Notas y difusión de información.
- Crear una base de datos de acuerdo a las necesidades de la institución.
- Brindar información confiable a estudiantes como a docentes de la institución.

2.1.7.5.1.4 Suposiciones y Restricciones

Acontecimientos que deben ocurrir para que el proyecto sea ejecutado con éxito pero que están totalmente fuera del ámbito del control del equipo de proyecto.

2.1.7.5.1.5 Suposiciones

Suponemos que el Instituto Comercial Superior INCOS Tarija cuenta con:

- Un ambiente adecuado para la implementación del sistema.
- Equipo de computación. En el que incluye un computador, impresora.
- La disponibilidad necesaria para entregar información.
- Personal con conocimientos básicos de computación.
- Presupuesto suficiente para la implementación del Sistema.
- Interés de la institución para utilizar el Sistema.
- La formación del personal encargado se lleve en la fecha determinada.
- Un servidor Web para la implementación del sistema

2.1.7.5.1.6 Restricciones

Limitaciones generalmente fuera del ámbito de control del equipo de proyecto que pueden afectar negativamente a su alcance.

- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- Contar con un servidor Web que no presente las características mínimas para el correcto funcionamiento del sistema.
- El sistema será restringido, sólo usuarios privilegiados podrán acceder al sistema.
- Para la manipulación de la base de datos, sólo podrán acceder el personal autorizado.
- No contar con los fondos suficientes para realizar la socialización del Proyecto.

2.1.7.5.2 Entregables del proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados

y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

2.1.7.5.3 Plan de Desarrollo del Software

Es el presente documento.

2.1.7.5.4 Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior, consideramos los objetivos estratégicos de la organización, teniendo en cuenta que esos objetivos serán descompuestos en un conjunto de sub-objetivos más concretos, para la identificación de procesos de negocio. Se presentan los modelos definidos en RUP como modelo del negocio (modelo de casos de uso del negocio y de objetos del negocio).

2.1.7.5.4.1 Introducción

El Modelo de Caso de Uso del Negocio es un artefacto de la disciplina Requisitos en la metodología RUP la cual estamos implementando.

2.1.7.5.4.2 Propósito

Comprender la Estructura y la Dinámica de la Organización.

Comprender problemas actuales e identificar posibles mejoras.

2.1.7.5.4.3 Alcance

Describe los Procesos de Negocio y los Clientes.

Identifica y Describe los Procesos de Negocio según los Objetivos de la Organización.

Definir un caso de uso del negocio para cada proceso de negocio.

2.1.7.5.5 Diagramas de Caso de Uso del Negocio

2.1.7.5.5.1 Modelo de casos de Uso del Negocio Inscripción

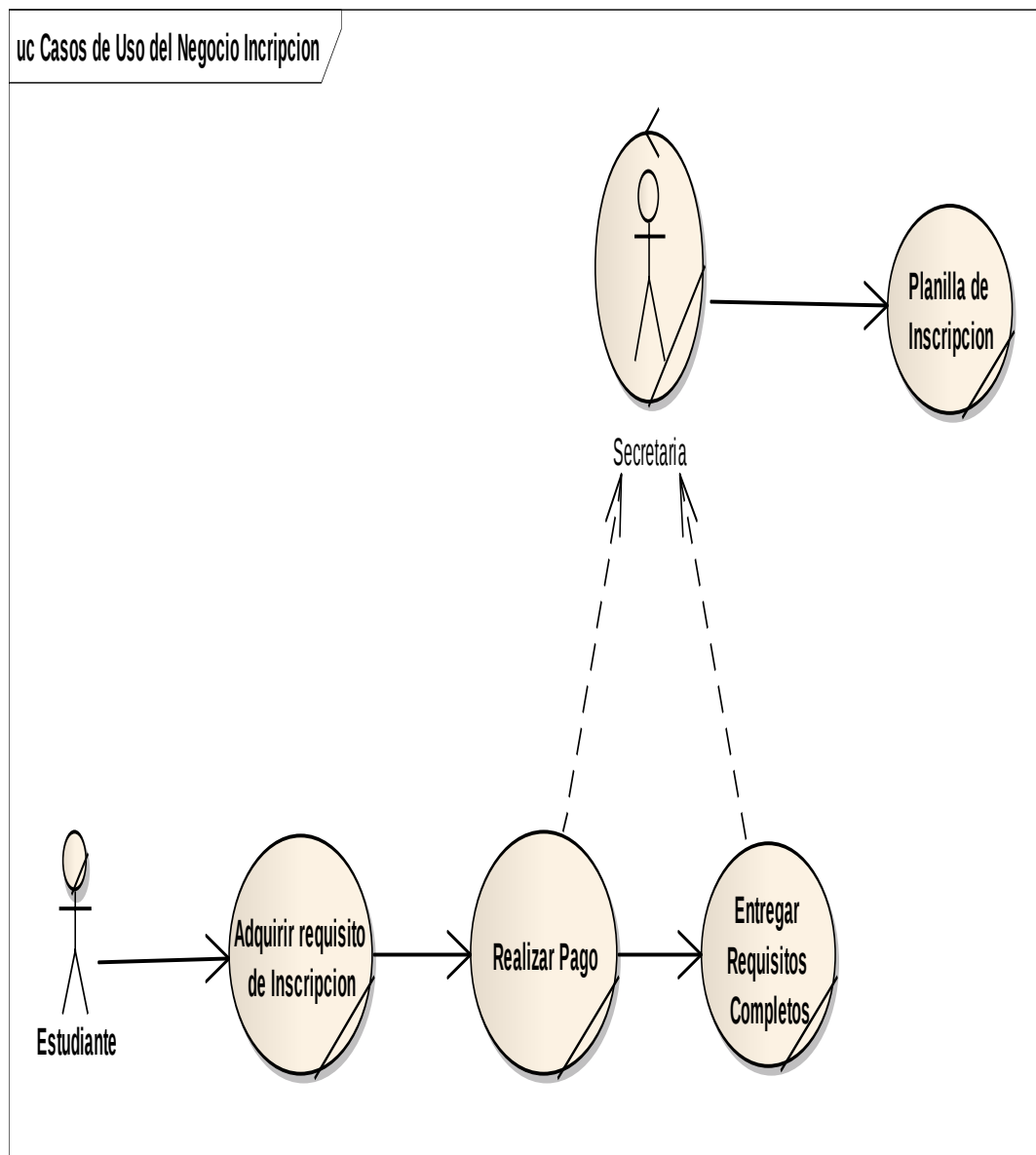


Figura 13. Caso de Uso del Negocio Inscripción

2.1.7.5.2 Modelo de casos de Uso del Negocio Registro de Notas

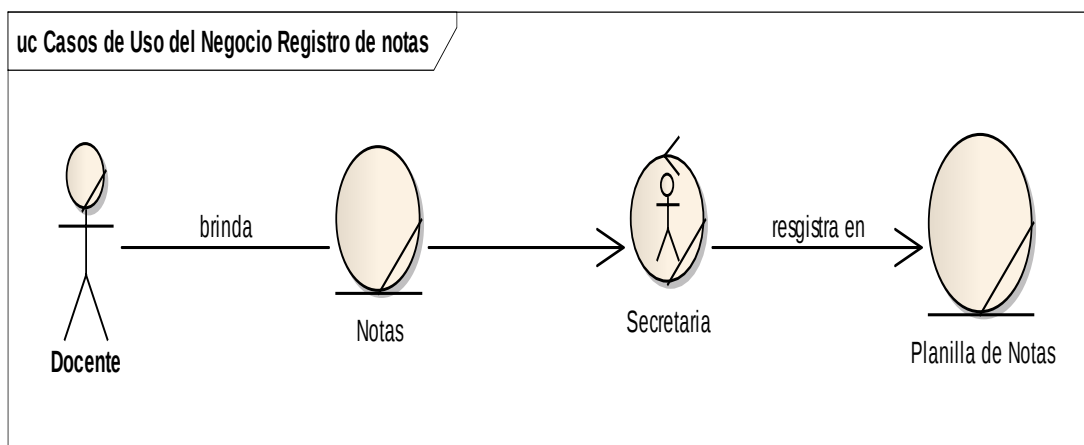


Figura 14. Caso de Uso del Registro de Notas

2.1.7.5.3 Modelo de casos de Uso del Negocio Reporte

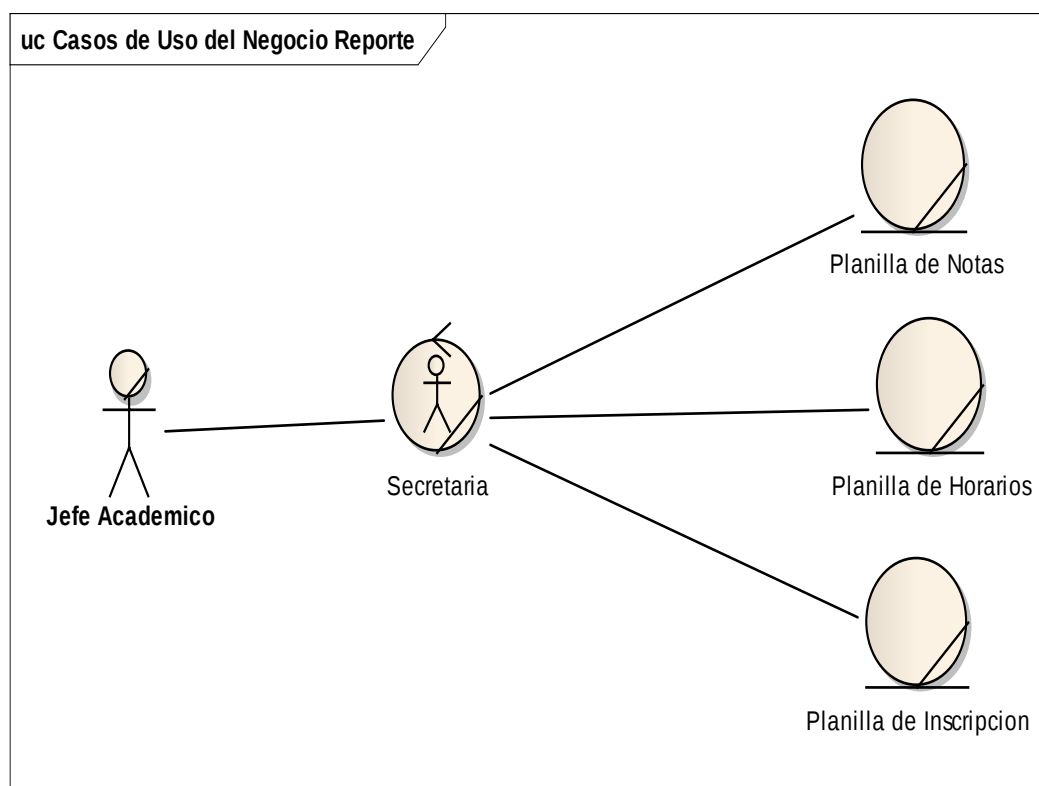


Figura 15. Caso de Uso del Negocio Reporte

2.1.7.5.4 Modelo de casos de Uso del Negocio Reporte

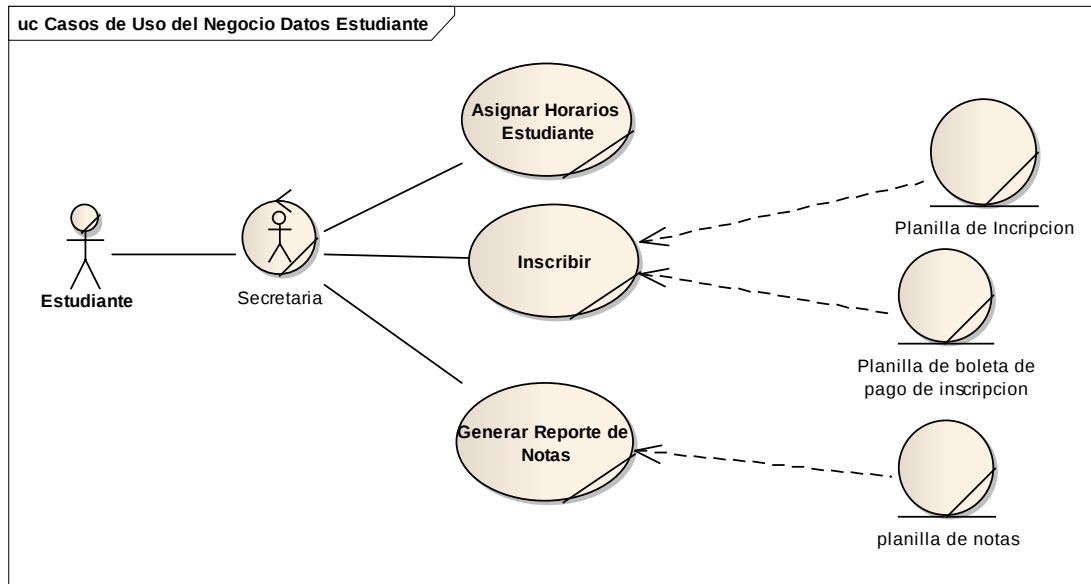


Figura 16. Caso de Uso del Negocio Reporte

2.1.7.5.6 Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio. Para la representación de este modelo se utilizan Diagramas de Colaboración (para mostrar actores externos, internos y las entidades (información) que manipulan, un Diagrama de Clases para mostrar gráficamente las entidades del sistema y sus relaciones, y Diagramas de Actividad para mostrar los flujos de trabajo.

2.1.7.5.6.1 Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

2.1.7.5.6.2 Introducción

El presente documento recoge los términos manejados durante la elaboración del proyecto de desarrollo del Sistema Informático para Mejoramiento de Afiliación. Se trata de un diccionario informal de datos y de definiciones de la nomenclatura que se maneja en la construcción del Sistema.

2.1.7.5.6.3 Propósito

El propósito del presente documento es definir la terminología manejada en el proyecto a desarrollar, también sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecidos del proyecto.

2.1.7.5.6.4 Alcance

El alcance del presente documento se extiende a todo el proyecto en desarrollo.

2.1.7.5.6.4.1 Organización del Glosario

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según ordenación alfabética tradicional.

2.1.7.5.7 Definiciones

2.1.7.5.7.1 Administrador

Es aquella persona que se dedica a mantener y operar un sistema de cómputo.

2.1.7.5.7.2 Backup (copia de respaldo, copia de seguridad)

Copia de ficheros o datos de forma que estén disponibles en caso de que un fallo produzca la pérdida de los originales. Esta sencilla acción evita numerosos, y a veces irremediables, problemas si se realiza de forma habitual y periódica.

2.1.7.5.7.3 Base de Datos

Se encarga de almacenar información acerca de administradores, usuarios, y todo referente a las actividades de los usuarios, con el ordenador y componentes del sistema

2.1.7.5.7.4 Usuario

Es la persona que podrá hacer uso del sistema en desarrollo, pudiendo realizar procesos de en el sistema de acuerdo al rol asignado.

2.1.7.5.7.5 Diagrama

Representación gráfica de un conjunto de elementos, representando la mayoría de las veces como un grafo conexo de nodos (elementos) y arcos (relaciones).

2.1.7.5.7.6 Foda

Es una herramienta de análisis estratégico, que permite analizar elementos internos o externos de programas y proyectos.

2.1.7.5.7.7 Interfaz de Usuario

La interfaz de usuario (IU) es uno de los componentes más importantes de cualquier sistema computacional, pues funciona como el vínculo entre el humano y la máquina. La interfaz de usuario es un conjunto de protocolos y técnicas para el intercambio de información entre una aplicación computacional y el usuario. La IU es responsable de solicitar comandos al usuario, y de desplegar los resultados de la aplicación de una manera comprensible. La IU no es responsable de los cálculos de la aplicación, ni del almacenamiento, recuperación y transmisión de la información

2.1.7.5.7.8 Modelo

Abstracción que describe el sistema bajo estudio. (Un modelo puede consistir en diagramas más los textos, notaciones o aclaraciones necesarias para entenderlos)

2.1.7.5.7.9 Negocio

Satisfacer las necesidades de alimentación y socialización ofreciendo diversos productos de comida rápida, diferenciados en cuanto al sabor e higiene, brindando un ambiente confortable y familiar a través de un sistema de atención personalizada mediante el asesoramiento al cliente..

2.1.7.5.7.10 Operador de Base de Datos

Persona con acceso, a las interioridades de la BD.

2.1.7.5.7.11 RUP

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

2.1.7.5.7.12 SQL

StructuredQueryLanguage - Lenguaje Estructurado de Consultas. Lenguaje estándar interactivo para la consulta y actualización de bases de datos.

2.1.7.5.7.13 Usuario

El *usuario* de un producto informático (bien sea hardware o software), es la persona a la que va destinada dicho producto una vez que ha superado las fases de desarrollo correspondientes.

Normalmente, el software se desarrolla pensando en la comodidad del *usuario final*, y por esto se presta especial interés y esfuerzo en conseguir una interfaz de usuario lo más clara y sencilla posible.

2.1.7.5.8 Modelo de Casos de Uso

2.1.7.5.8.1 Modelo de Caso de Uso General



Figura 17. Caso de Uso del General

2.1.7.5.8.2 Modelo de Caso de Uso del Jefe Académico

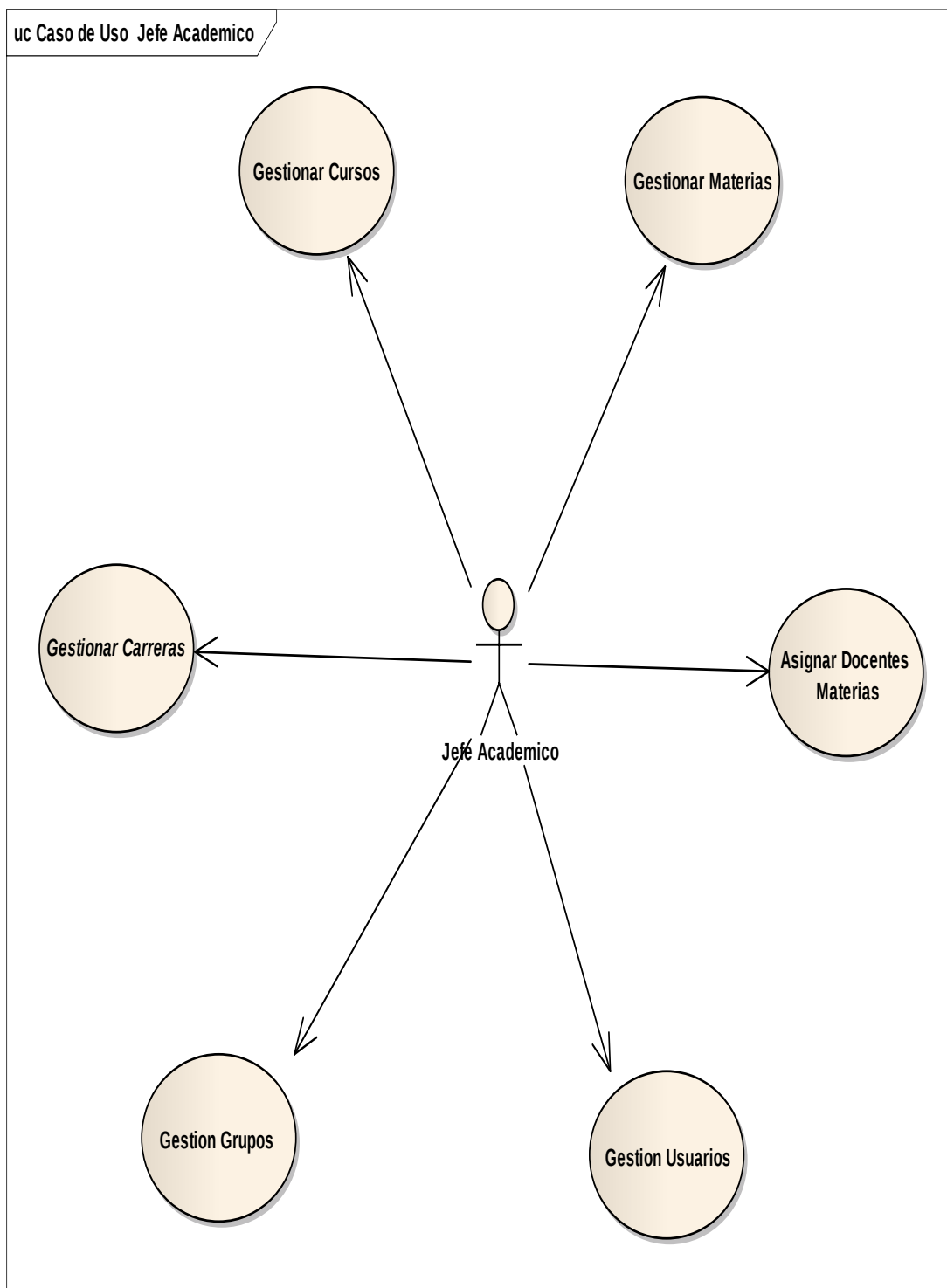


Figura 18. Caso de Uso del Módulo Jefe Académico

2.1.7.5.8.3 Modelo de Caso de Uso del Módulo Reporte

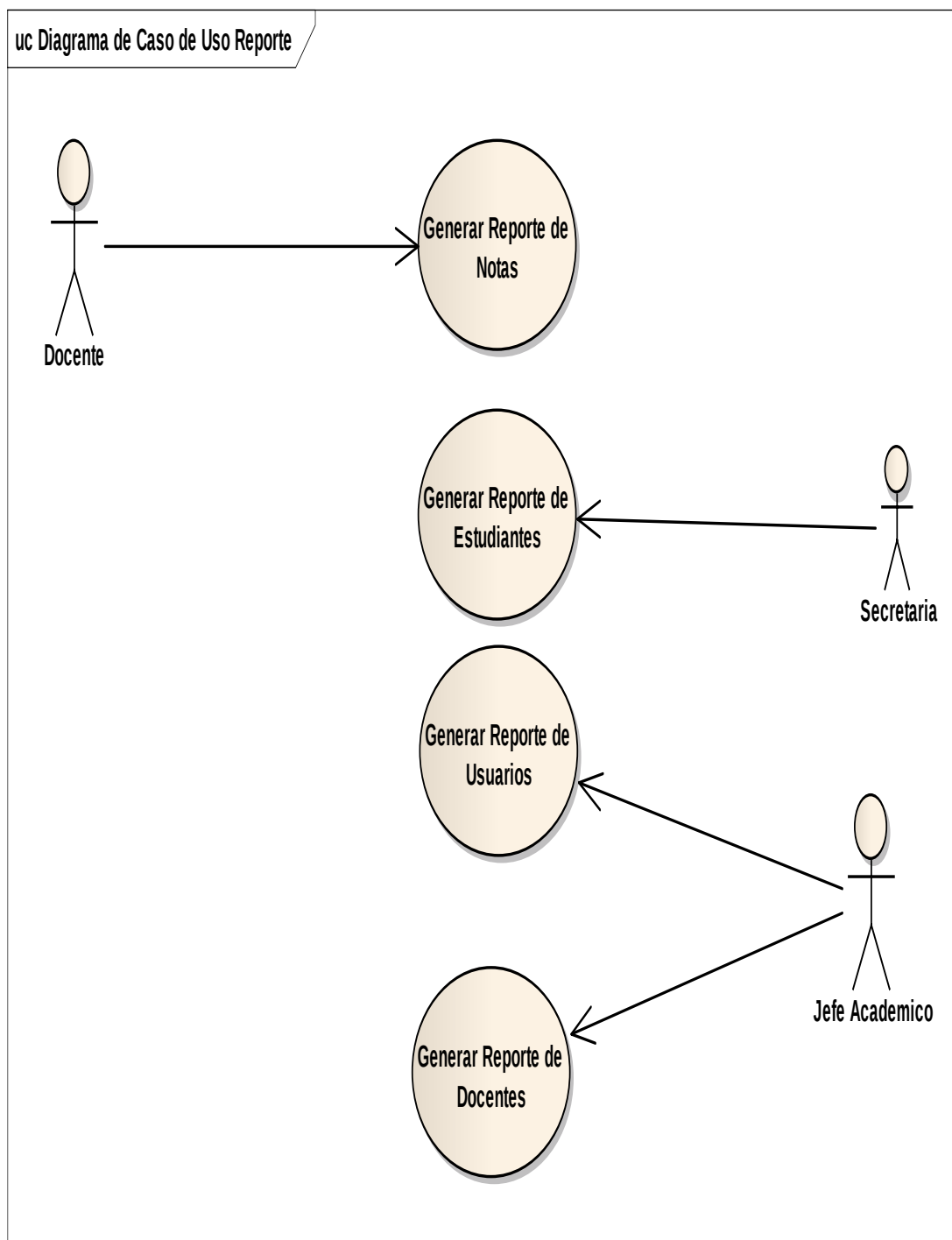


Figura 19. Caso de Uso del Módulo Reportes

2.1.7.5.8.4 Modelo de Caso de Uso Ingresar al Sistema

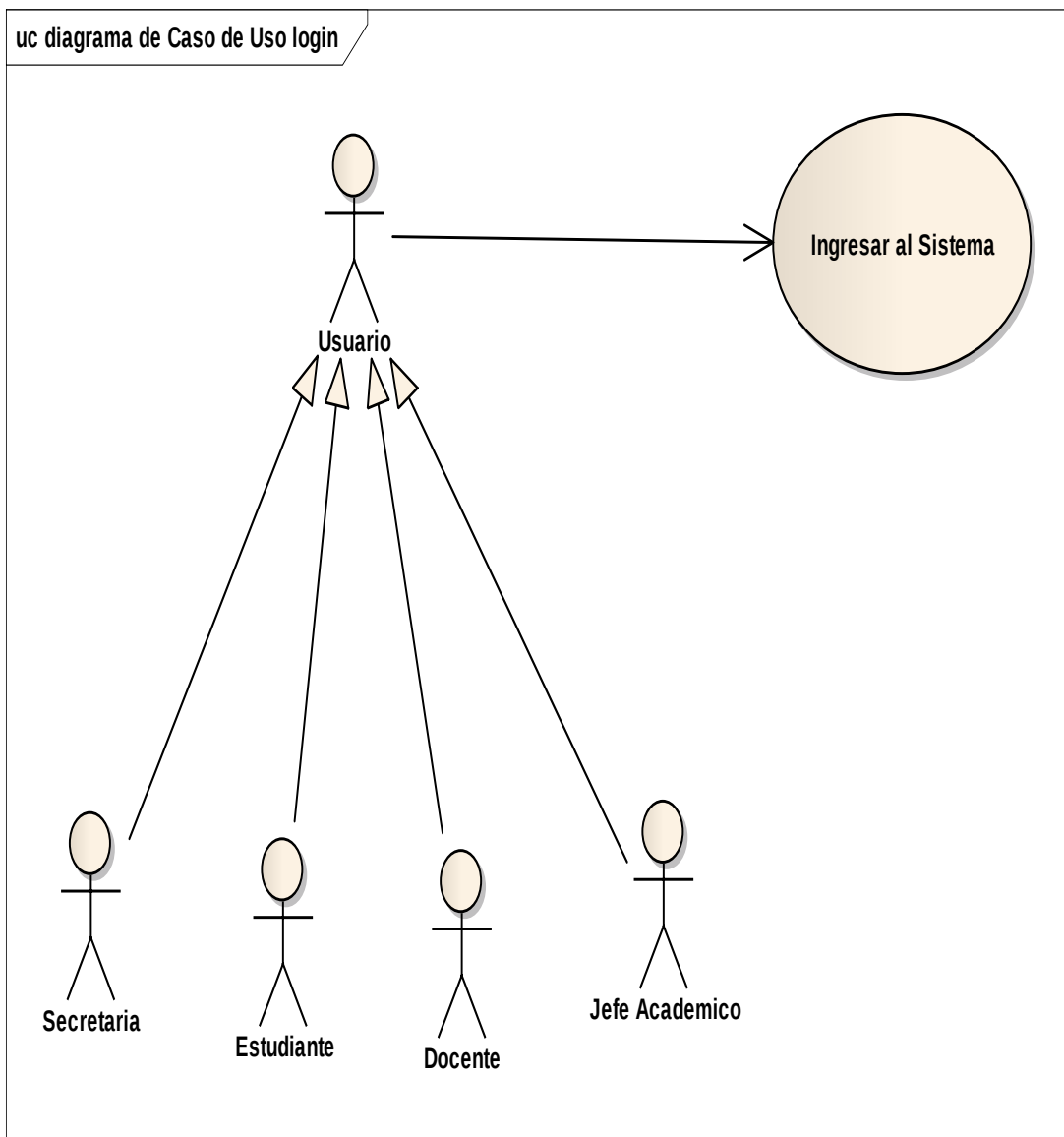


Figura 20. Caso de Uso Ingresar al Sistema

2.1.7.5.8.5 Modelo de Caso de Uso Gestionar Usuario

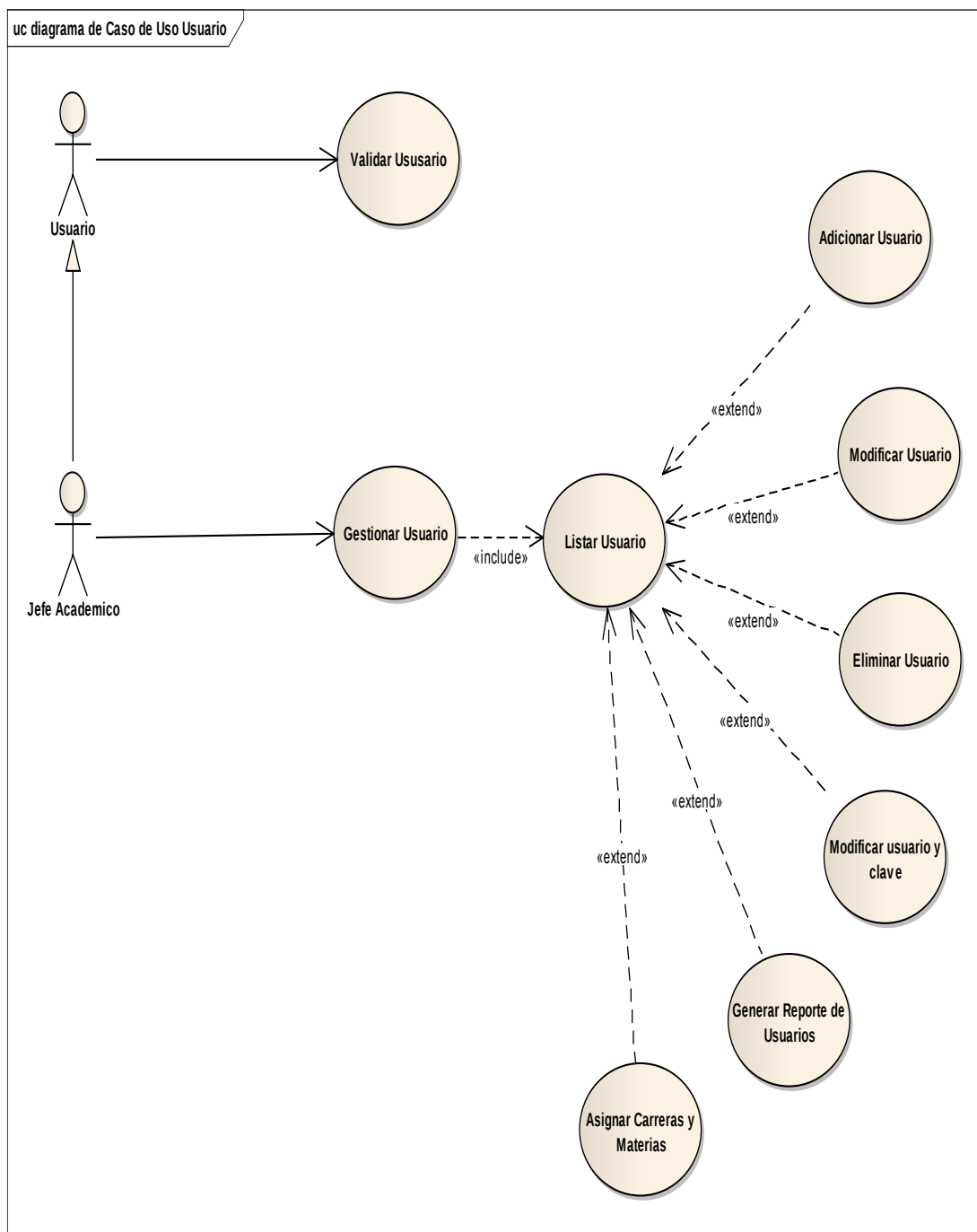


Figura 21. Caso de Uso Gestionar Usuario

2.1.7.5.8.6 Modelo de Caso de Uso Gestionar Evaluación

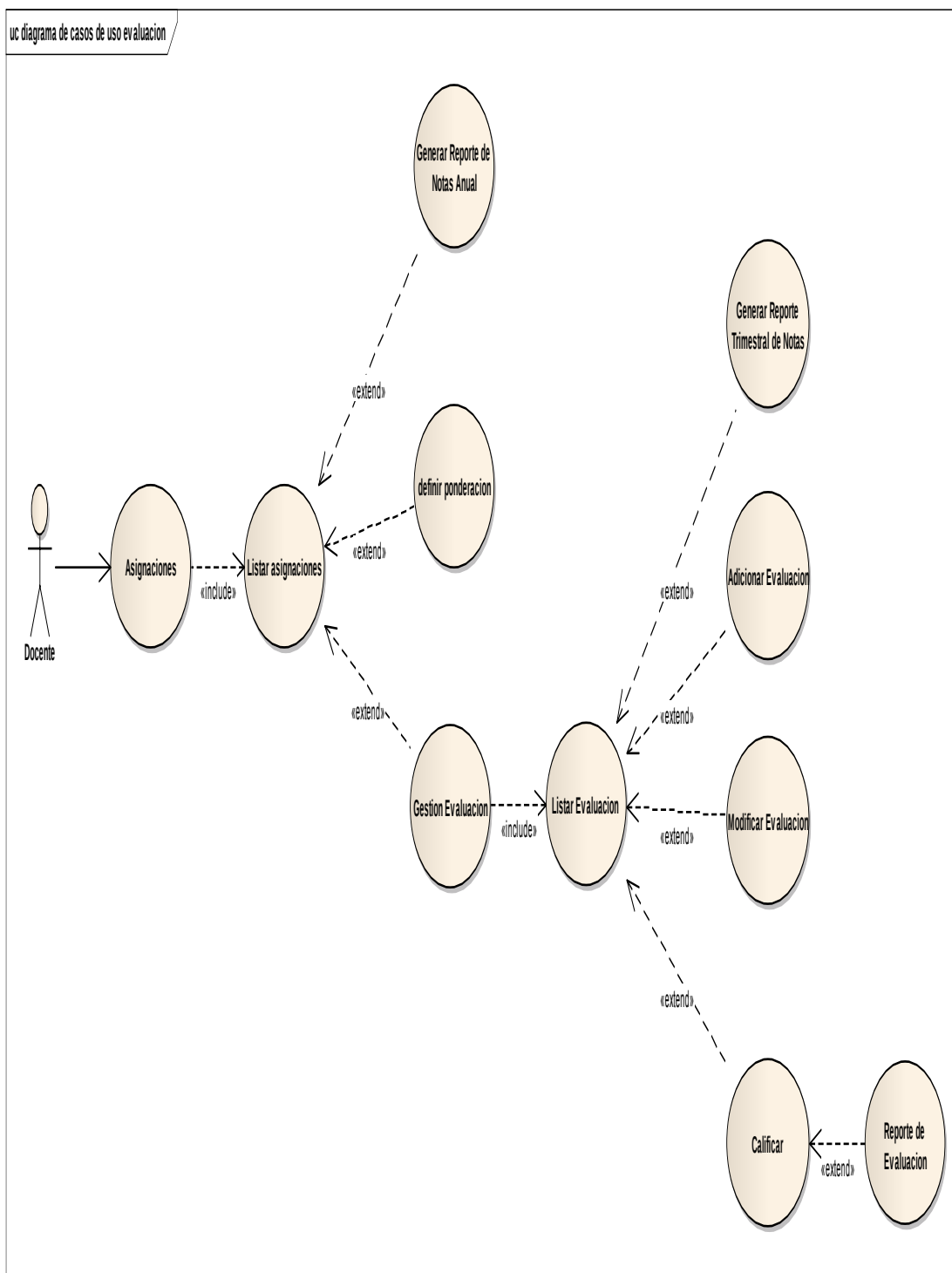


Figura 22. Caso de Uso Gestionar Evaluación

2.1.7.5.8.7 Modelo de Caso de Uso Gestionar Cursos

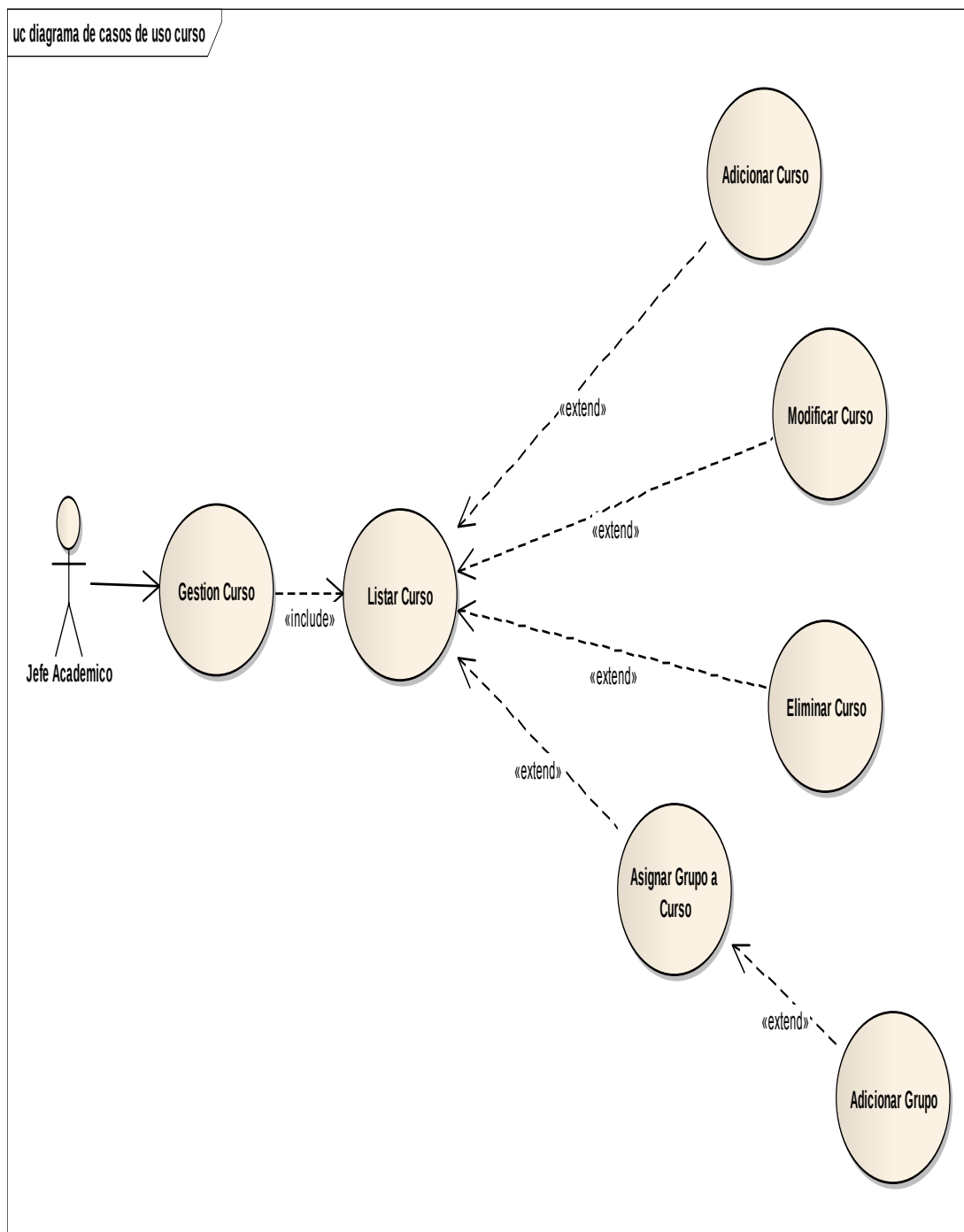


Figura 23. Caso de Uso Gestionar Curso

2.1.7.5.8.8 Modelo de Caso de Uso Gestionar Materias

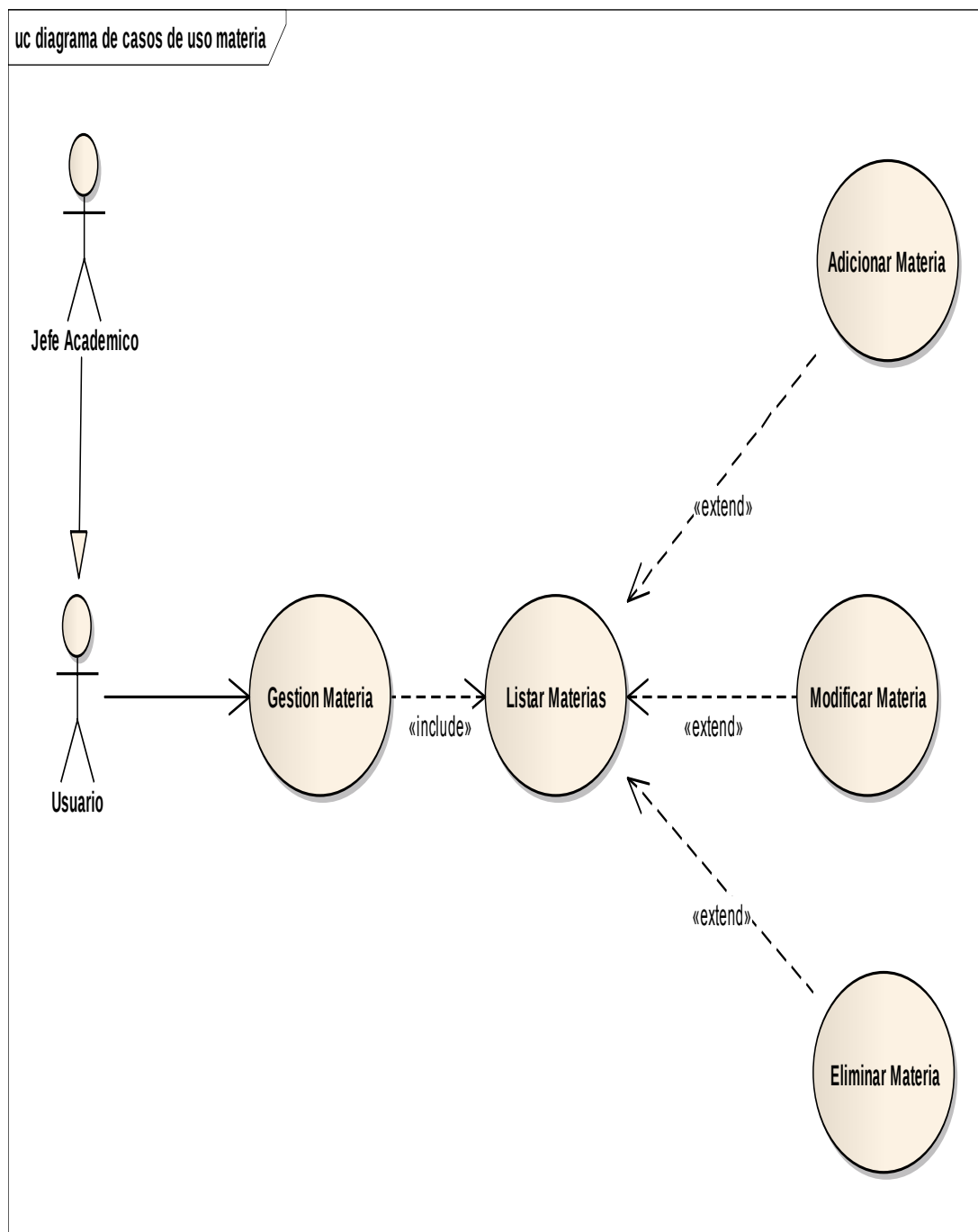


Figura 24. Caso de Uso Gestionar Materias

2.1.7.5.8.9 Modelo de Caso de Uso Gestionar Carreras

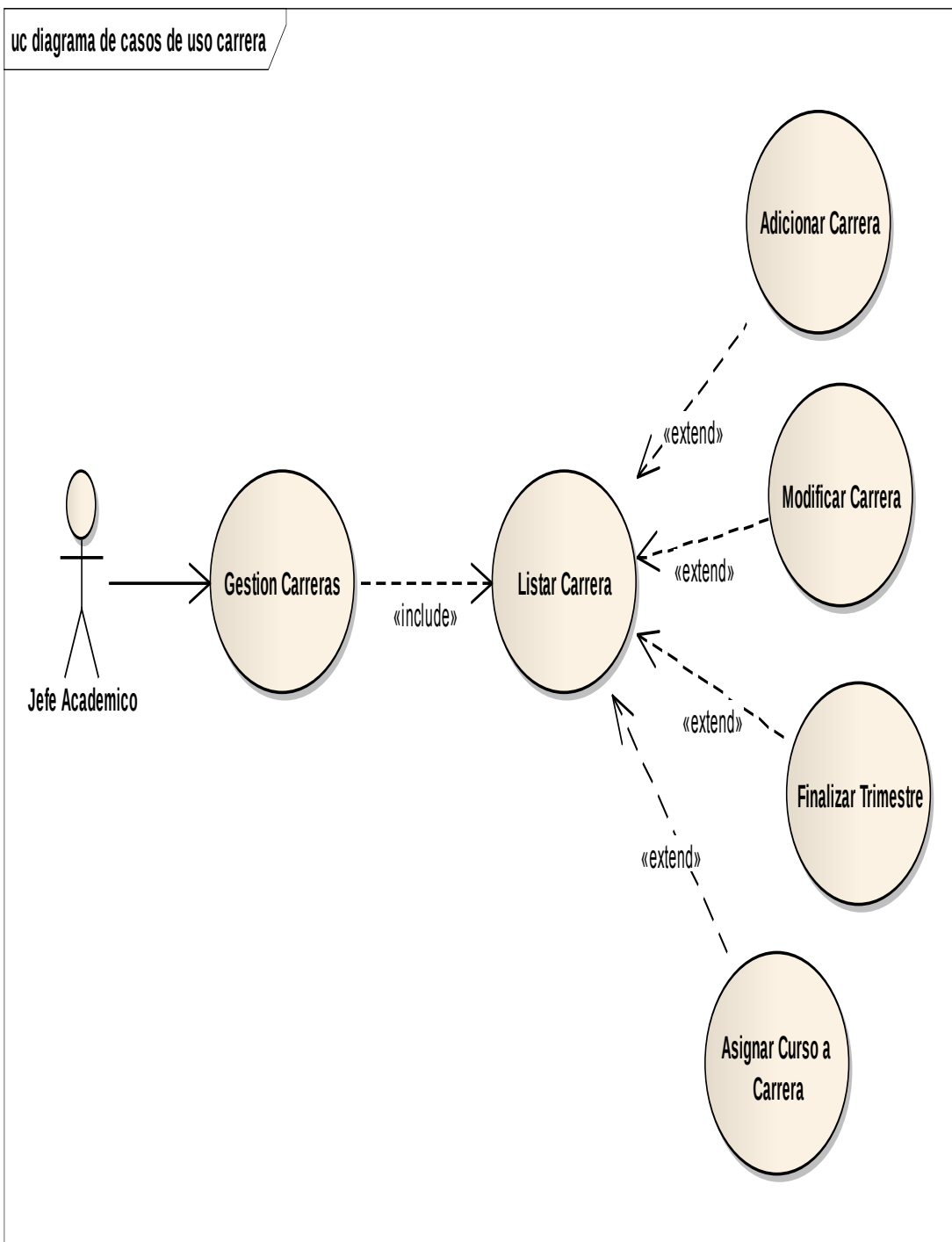


Figura 25. Caso de Uso Gestionar Carrera

2.1.7.5.8.10 Modelo de Caso de Uso Gestionar Estudiantes

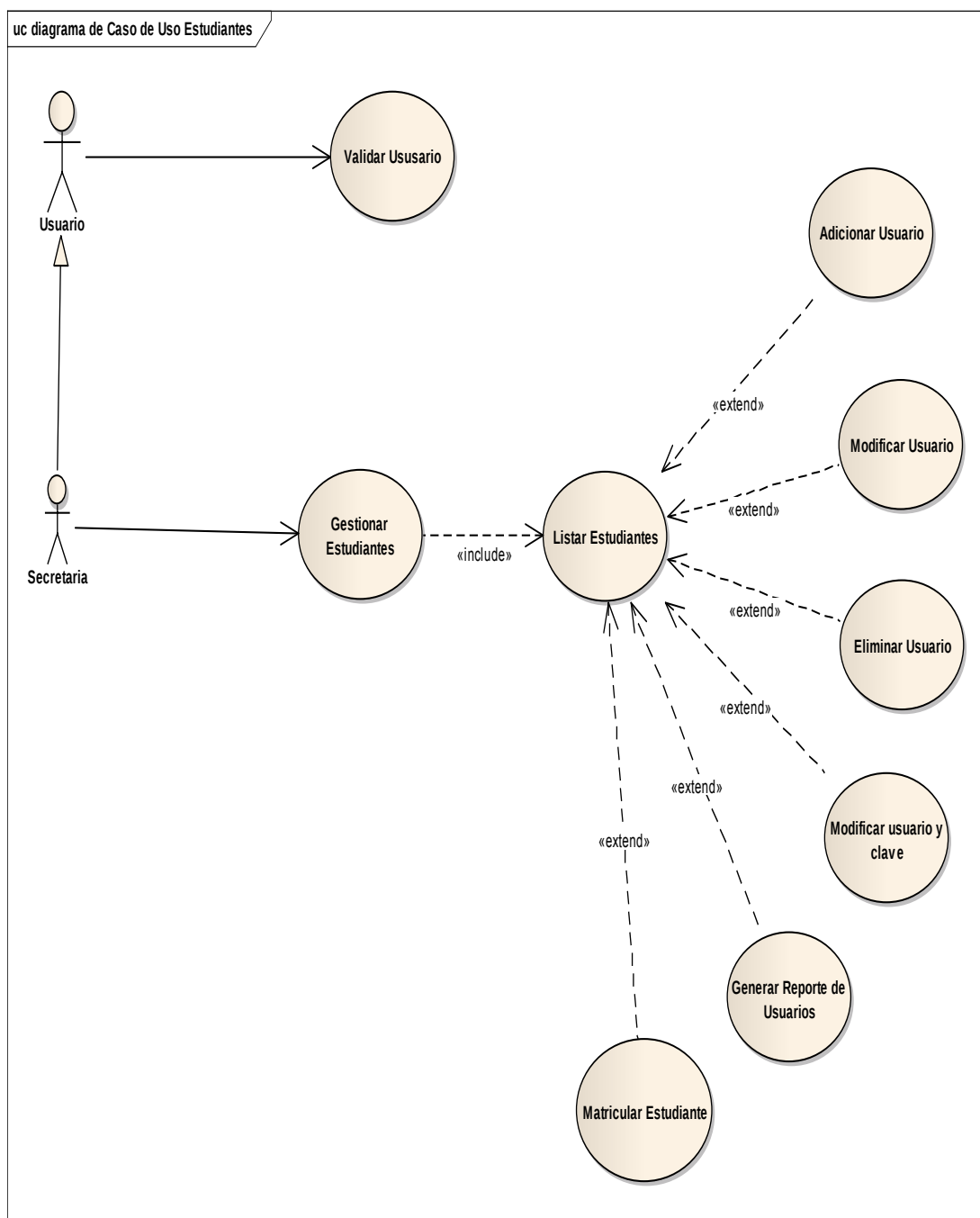


Figura 26. Caso de Uso Gestionar Estudiantes

2.1.7.5.8.11 Modelo de Caso de Uso Gestionar Matricula

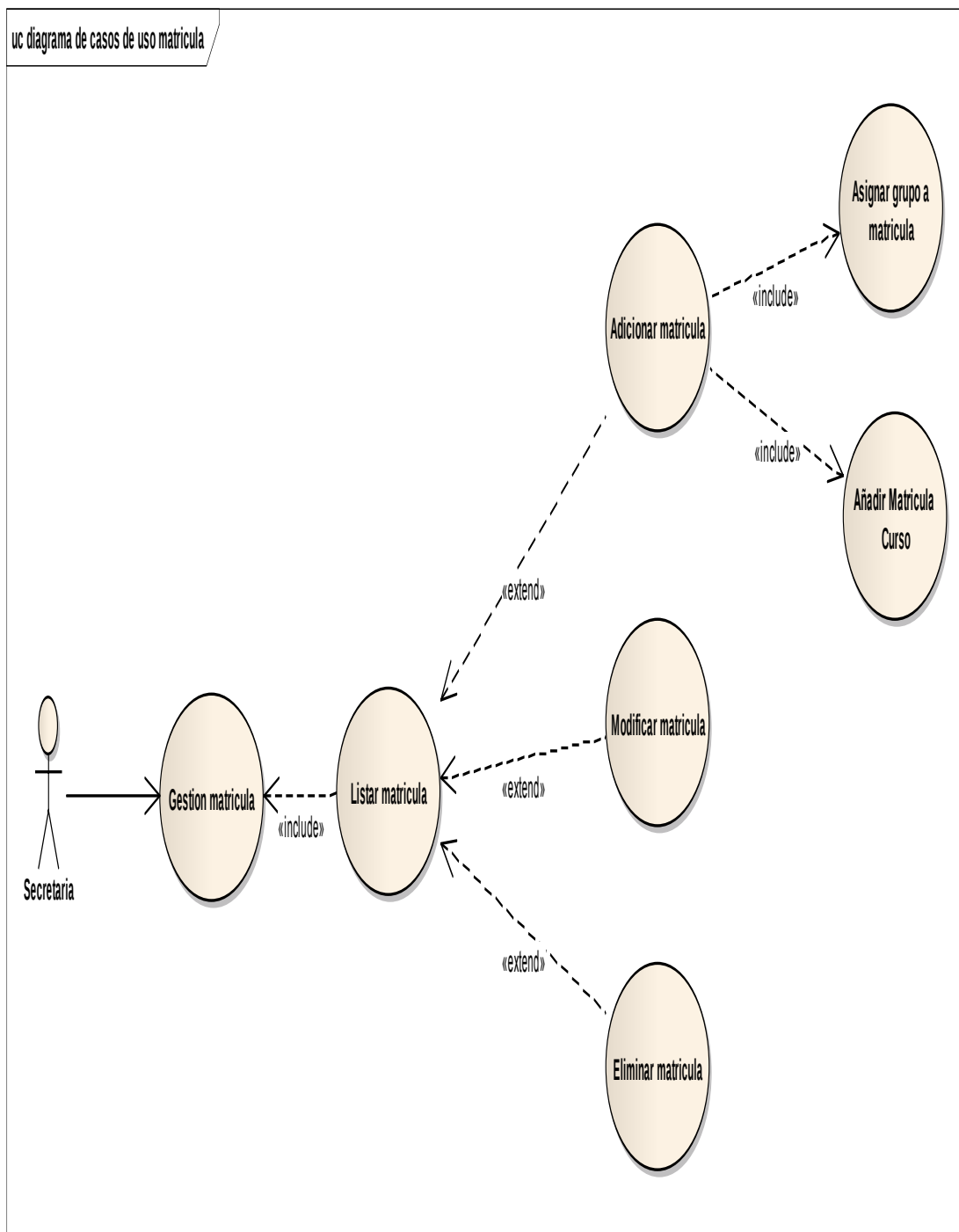


Figura 27. Caso de Uso Gestionar Matricula

2.1.7.5.8.12 Modelo de Caso de Uso Gestionar Grupo

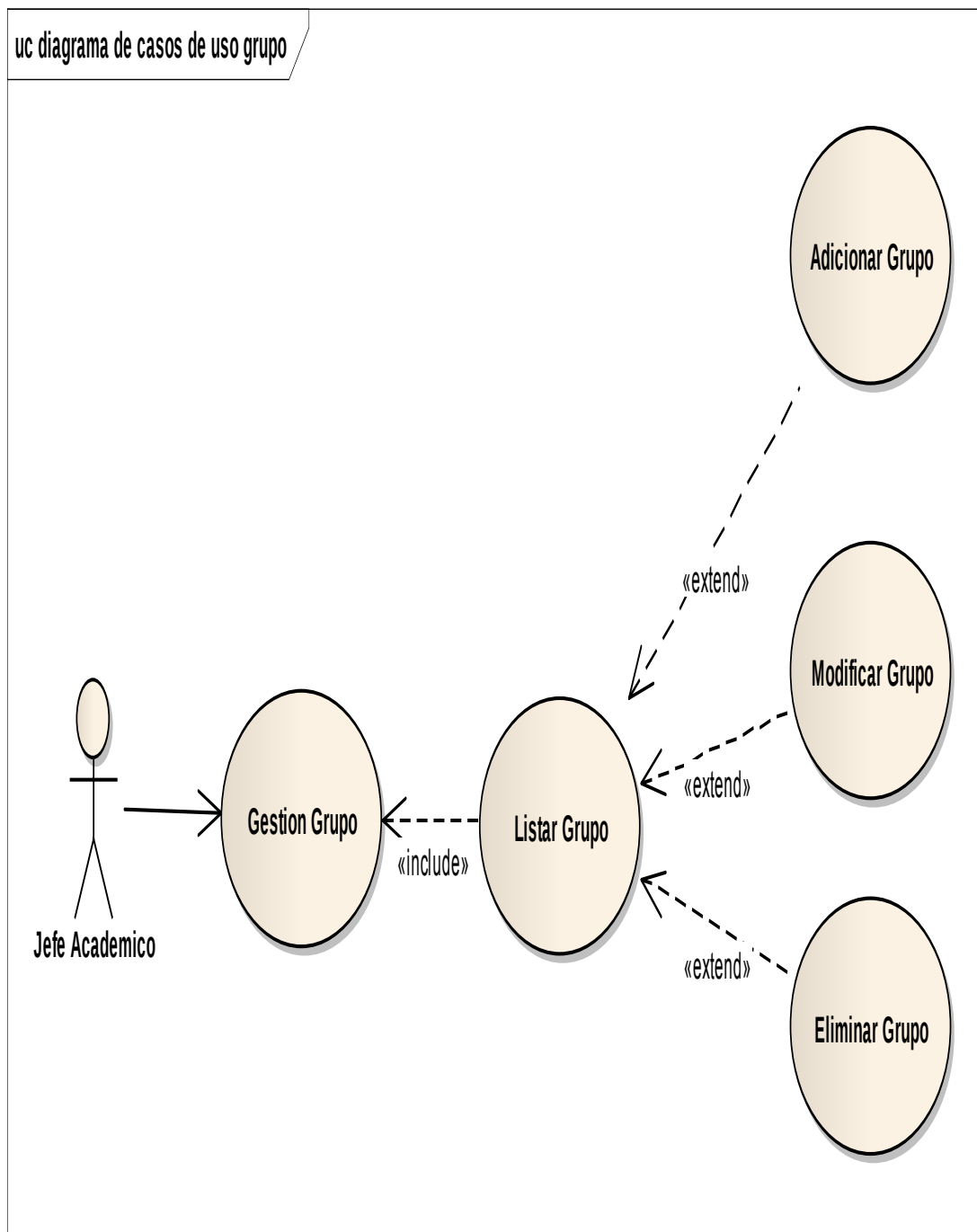


Figura 28. Caso de Uso Gestionar Grupo

2.1.7.5.9 Visión

2.1.7.5.9.1 Introducción

2.1.7.5.9.2 Propósito

El propósito de éste documento es recoger, definir y analizar las necesidades más importantes y las características del desarrollo del Sistema Informático encargado del mejoramiento de la administración académica Instituto Comercial Superior INCOS Tarija, con el objetivo de llevar a cabo un producto que tome en cuenta las estrategias necesarias para poder brindar información valedera para el Usuario final.

2.1.7.5.9.3 Alcance

El presente documento se ocupa de reunir todas las necesidades del Usuario, para así poder diseñar un Sistema que satisfaga a las mismas como ser:

- Contar con un medio de publicación, noticias y promociones.
- Contar con la información centralizada en una sola organización.

2.1.7.5.9.4 Definiciones, Acrónimos y Abreviaciones

RUP: Son las siglas de RationalUnifiedProcess. Se trata de una metodología para describir el proceso de desarrollo de software.

UML: Son las siglas de Lenguaje Unificado de Modelación. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

COCOMO: Acrónimo en inglés de Modelo Constructivo de Costes. Es un modelo que permite realizar estimaciones y planificaciones de proyectos de sistemas de información.

LAN: Las siglas de Red de Área Local.

SQA: Acrónimo en inglés de Aseguramiento de la Calidad del Software.

2.1.7.5.9.5 Posicionamiento

2.1.7.5.9.6 Perfil de los Stakeholders

2.1.7.5.9.6.1 Universidad Autónoma Juan Misael Saracho

Representante	Universidad Autónoma Juan Misael Saracho.
Descripción	Entidad que asesora el proyecto.
Tipo	Proveedor.
Responsabilidades	Seguimiento del desarrollo del proyecto. Aprueba requisitos y funcionalidades.
Grado de participación	Velar que el proyecto sea ejecutado.

Tabla 1. Juan Misael Saracho

2.1.7.5.9.6.2 Instituto Comercial Superior INCOS Tarija

Representante	Ing. Carlos Quiñones
Descripción	Jefe Académico del Instituto Comercial INCOS Tarija
Tipo	Administrador
Responsabilidades	Administrador del Sistema INCOS
Grado de participación	Velar por que se cumplan requisitos y funcionalidades, y que el proyecto sea ejecutado.

Tabla 2. Instituto Comercial Superior INCOS Tarija

2.1.7.5.9.6.3 Docente

Representante	Lic. Efraín Torrejón
Descripción	Ente regulador universitario.
Tipo	Guía de desarrollo.
Responsabilidades	Realiza un control paso a paso del desarrollo del proyecto. Aprueba los distintos puntos tratados en el proyecto. Analiza los distintos documentos presentados acerca del producto.
Grado de participación	Guiar y evaluar el análisis, diseño y desarrollo del sistema.

Tabla 3. Skateholder Docente

2.1.7.5.9.6.4 Perfil del Usuario

2.1.7.5.9.6.5 Jefe Académico

Representante	Jefe Académico
Descripción	Administrador de los recursos del sistema.
Tipo	Usuario.
Responsabilidades	Responsable del control de roles y acceso, usuarios del sistema; además es el responsable de administrar todos los componentes de la empresa
Grado de participación	Usuario principal del Sistema

Tabla 4. Skateholder Administrador

2.1.7.5.9.6.6 Usuario

Representante	Usuario.
Descripción	Usuario con ciertos atributos- permisos de ejecución de la aplicación.
Tipo	Usuario causal.
Responsabilidades	Es el encargado de realizar ventas y administrar los pedidos
Grado de participación	Usar el Sistema, buscar debilidades del mismo.

Tabla 5. Skateholder Usuario

2.1.7.5.9.7 Descripción Global del Proyecto

2.1.7.5.9.7.1 Descripción Global del Producto

Debido a la creciente demanda de las nuevas tecnologías de información y comunicación y el impacto sociocultural que causan en nuestro entorno, elaboramos el proyecto “SIMA” Sistema Informático para el Mejoramiento de Afiliación.

Los beneficios de la utilización de nuevas tecnologías de comunicación como ser el Internet nos aseguran la integridad, veracidad y la organización de nuestra información.

2.1.7.5.10 Especificación de los Casos de Uso

2.1.7.5.10.1 Caso de Uso Ingresar al Sistema

Caso de uso	Ingresar al Sistema
Actores	Jefe Académico, Usuario
Propósito	Capturar los datos que introdujo el administrador o encargado de operaciones y proporcionar un interfaz.
Resumen	Los datos introducidos por el administrador o encargado de operaciones son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el administrador del sistema.
Precondiciones	El administrador del sistema debe introducir sus datos de identificación en el sistema.
Flujo principal	1. El actor introduce datos de login y password en el sistema. 2. El actor presiona botón Ingresar. 3. El sistema valida sus datos. 4. El actor espera su validación. Si los datos son correctos muestra la Pantalla Principal de

	Administración.
Flujo alternativo	1. Si el actor no llena el formulario de acceso, se muestra un mensaje donde se informa que los campos son requeridos. 2. Si el usuario no existe se muestra mensaje de error y vuelve a mostrar la pantalla Principal.
Excepciones	1. Usuario no Registrado.

Tabla 6. Caso de uso Ingresar al Sistema

2.1.7.5.10.2 Caso de Uso Gestionar Grupo

Caso de uso	Gestionar Grupo
Actores	Jefe Académico
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a las Grupos.
Resumen	Los datos introducidos por el Jefe Académico son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el administrador del sistema.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de las Grupo registrados en el sistema. 2. El actor realiza la Búsqueda Proceso en el Listado. 3. El sistema muestra los Grupo que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo	1. El actor decide no confirmar selección y el sistema no

alternativo	realiza ninguna operación.
Excepciones	Ninguno

Tabla 7. Caso de uso Gestionar Grupo

2.1.7.5.10.3 Caso de Uso Adicionar Grupo

Caso de uso	Adicionar Grupo
Actores	Jefe Académico
Propósito	Adiciona una nueva Grupo
Resumen	Se introducen nuevos datos para adicionar un nuevo Grupo en la base de datos.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario. 3. El actor introduce datos del Grupo en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Grupo.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Proceso Guardada Satisfactoriamente

Tabla 8. Caso de uso Adicionar Categoría

2.1.7.5.10.4 Caso de Uso Eliminar Grupo

Caso de uso	Eliminar Grupo
Actores	Jefe Académico
Propósito	Eliminar un registro de Grupo

Resumen	Se escoge la opción eliminar del Proceso que se desee eliminar
Precondiciones	1. El Jefe Académico deberá escoger el Grupo que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación del Grupo El sistema cambiando el estado del mismo regresa a la pantalla de Listar Proceso
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Grupo eliminada Satisfactoriamente

Tabla 9. Caso de uso Eliminar Grupo

2.1.7.5.10.5 Caso de Uso Modificar Grupo

Caso de uso	Modificar Grupo
Actores	Jefe Académico
Propósito	Modificar un registro de Grupo
Resumen	Se escoge la opción modificar del Grupo que desee cambiar valores
Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Grupo
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos del Grupo 3. El actor realiza los cambios de datos del Grupo en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios

	estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Listar Grupo
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Proceso Modificada Satisfactoriamente 2. No se pudo Actualizar a la Grupo

Tabla 10. Caso de uso Modificar Grupo

2.1.7.5.10.6 Caso de Uso Gestionar Usuario

Caso de uso	Gestionar Usuario
Actores	Jefe Académico
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a los Usuarios.
Resumen	Los datos introducidos por el administrador son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el administrador del sistema.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de los Usuarios registrados en el sistema. 2. El actor realiza la Búsqueda del Usuario en el Listado. 3. El sistema muestra los Usuarios que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.

Excepciones	Ninguno
--------------------	---------

Tabla 11. Caso de uso Gestionar Usuarios

2.1.7.5.10.7 Caso de Uso Adicionar Usuario

Caso de uso	Adicionar Usuario
Actores	Jefe Académico
Propósito	Adiciona un nuevo Usuario
Resumen	Se introducen nuevos datos para adicionar un nuevo Usuario en la base de datos
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario. 3. El actor introduce datos del Usuario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Usuario
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Usuarios Guardado Satisfactoriamente

Tabla 12. Caso de uso Adicionar Usuario

2.1.7.5.10.8 Caso de Uso Eliminar Usuarios

Caso de uso	Eliminar Usuario
Actores	Jefe Académico
Propósito	Eliminar un registro de Usuario
Resumen	Se escoge la opción eliminar Usuario que se desee eliminar

Precondiciones	1. El Jefe Académico deberá escoger el Usuario que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación del Usuario cambiando el estado del mismo. 4. El sistema regresa a la pantalla de Listar Usuario
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Usuario eliminado Satisfactoriamente

Tabla 13. Caso de uso Eliminar Usuario

2.1.7.5.10.9 Caso de Uso Modificar Usuarios

Caso de uso	Modificar Usuario
Actores	Jefe Académico
Propósito	Modificar un registro de Usuario
Resumen	Se escoge la opción modificar del Usuario que desee cambiar valores
Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Usuario
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos del Usuario 3. El actor realiza los cambios de datos del Usuario en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario.

	7. El sistema regresa a la pantalla de Listar Usuario
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Usuario Modificado Satisfactoriamente 2. No se pudo Actualizar al Usuario

Tabla 14. Caso de uso Modificar Usuarios

2.1.7.5.10.10 Caso de Uso Generar Reporte de Usuarios

Caso de uso	Generar Reporte de Usuarios
Actores	Jefe Académico
Propósito	Generar un reporte con la documentación requerida
Resumen	Generar un Pdf
Precondiciones	1. Haber iniciado sesión
Flujo principal	1. El actor presiona en el menú la opción Reporte. 2. El sistema muestra el Reporte.
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Ninguna

Tabla 15. Caso de uso Modificar Usuarios

2.1.7.5.10.11 Caso de Uso Gestionar Materia

Caso de uso	Gestionar Materia
Actores	Jefe Académico
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a las Materia.
Resumen	Los datos introducidos por el Jefe Académico son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el

	administrador del sistema.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de las Materias registradas en el sistema. 2. El actor realiza la Búsqueda de Materias en el Listado. 3. El sistema muestra los Materias que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	Ninguno

Tabla 16. Caso de uso Gestionar Materia

2.1.7.5.10.12 Caso de Uso Adicionar Materia

Caso de uso	Adicionar Materia
Actores	Jefe Académico
Propósito	Adiciona una nueva Materia
Resumen	Se introducen nuevos datos para adicionar una nueva Materia en la base de datos
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario. 3. El actor introduce datos de la Materia en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Materia

Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Materia Guardado Satisfactoriamente

Tabla 17. Caso de uso Adicionar Materia

2.1.7.5.10.13 Caso de Uso Eliminar Materia

Caso de uso	Eliminar Materia
Actores	Jefe Académico
Propósito	Eliminar un registro de Materia
Resumen	Se escoge la opción eliminar de la Materia que se desee eliminar de la Base de Datos
Precondiciones	1. El administrador deberá escoger la Materia que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación cambiando el estado de la Materia. 4. El sistema regresa a la pantalla de Listar Materia
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Materia eliminado Satisfactoriamente

Tabla 18. Caso de uso Eliminar Materia

2.1.7.5.10.14 Caso de Uso Modificar Materia

Caso de uso	Modificar Egreso
Actores	Jefe Académico
Propósito	Modificar un registro de Materia
Resumen	Se escoge la opción modificar del Materia que desee cambiar valores

Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Materia
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos de la Materia 3. El actor realiza los cambios de datos de la Materia en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Listar Materia
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Materia Modificada Satisfactoriamente 2. No se pudo Actualizar a la Materia

Tabla 19. Caso de uso Modificar Materia

2.1.7.5.10.15 Caso de Uso Generar Reporte de Materia

Caso de uso	Generar Reporte de Materia
Actores	Jefe Académico
Propósito	Generar un reporte con la documentación requerida
Precondiciones	1. Haber iniciado sesión
Flujo principal	1. El actor presiona en el menú la opción Reporte. 2. El sistema muestra el Reporte.
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Ninguna

Tabla 20. Caso de uso Reporte de Materia

2.1.7.5.10.16 Caso de Caso de Gestionar Carrera

Caso de uso	Gestionar Carrera
Actores	Jefe Académico
Propósito	Permite Listar, Modificar y Eliminar a las Carreras.
Resumen	Los datos introducidos por el administrador o encargado de operaciones son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el administrador del sistema.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de las Carreras registradas en el sistema. 2. El actor realiza la Búsqueda de la Carrera en el Listado. 3. El sistema muestra los Carreras que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	Ninguno

Tabla 21. Caso de uso Gestionar Carreras

2.1.7.5.10.17 Caso de Uso Eliminar Carrera

Caso de uso	Eliminar Carreras
Actores	Jefe Académico
Propósito	Eliminar un registro de Carreras
Resumen	Se escoge la opción eliminar de la Carrera que se desee

	eliminar de la Base de Datos
Precondiciones	1. El administrador deberá escoger el Carrera que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación cambiando el estado de la Carrera 4. El sistema regresa a la pantalla de Listar Carreras
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Carreras eliminado Satisfactoriamente

Tabla 22. Caso de uso Eliminar Carreras

2.1.7.5.10.18 Caso de Uso Modificar Carreras

Caso de uso	Modificar Carreras
Actores	Jefe Académico
Propósito	Modificar un registro de Carrera
Resumen	Se escoge la opción modificar del Carreras que desee cambiar valores
Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Carrera
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos de la Carreras 3. El actor realiza los cambios de datos de la Carreras en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos.

	6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Listar Carreras
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Carreras Modificado Satisfactoriamente 2. No se puedo Actualizar al Carrera

Tabla 23. Caso de uso Modificar Carrera

2.1.7.5.10.19 Caso de Uso Generar Reporte de Carreras

Caso de uso	Generar Reporte de Carreras
Actores	Jefe Académico
Propósito	Generar un reporte con la documentación requerida
Resumen	Generar un Pdf.
Precondiciones	1. Haber iniciado sesión
Flujo principal	1. El actor presiona en el menú la opción Reporte. 2. El sistema muestra el Reporte.
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Ninguna

Tabla 24. Caso de uso Generar Reporte de Carrera

2.1.7.5.10.20 Caso de Uso Gestionar Curso

Caso de uso	Gestionar Curso
Actores	Jefe Académico
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a los Cursos.
Resumen	Los datos introducidos por el Jefe Académico son capturados, para luego ser evaluados y poder dar así una respuesta al Jefe Académico, de ser correctos los datos se

	abrirán una interfaz para el administrador del sistema.
Precondiciones	1. Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de los Cursos registrados en el sistema. 2. El actor realiza la Búsqueda de Cursos en el Listado. 3. El sistema muestra los Cursos que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	Ninguno

Tabla 25. Caso de uso Gestionar Curso

2.1.7.5.10.21 Caso de Uso Adicionar Curso

Caso de uso	Adicionar Curso
Actores	Jefe Académico
Propósito	Adiciona un nuevo Curso
Resumen	Se introducen nuevos datos para adicionar un nuevo Curso en la base de datos
Precondiciones	1. Acto de haber iniciado sesión 2. Entrar a la pantalla listar Curso
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario. 3. El actor introduce datos del Curso en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos.

	6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Curso
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Curso Guardado Satisfactoriamente

Tabla 26. Caso de uso Adicionar Curso

2.1.7.5.10.22 Caso de Uso Eliminar Curso

Caso de uso	Eliminar Curso
Actores	Jefe Académico
Propósito	Eliminar un registro de Curso
Resumen	Se escoge la opción eliminar del Curso que se desee eliminar de la Base de Datos
Precondiciones	1. El Jefe Académico deberá escoger el Curso que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación del Curso cambiando el estado. 4. El sistema regresa a la pantalla de Listar Curso
Flujo alternativo	2. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Curso eliminado Satisfactoriamente

Tabla 27. Caso de uso Eliminar Curso

2.1.7.5.10.23 Caso de Uso Modificar Curso

Caso de uso	Modificar Curso
Actores	Jefe Académico
Propósito	Modificar un registro de Curso
Resumen	Se escoge la opción modificar del Curso que desee cambiar valores
Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Curso
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos del Curso 3. El actor realiza los cambios de datos del Curso en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Listar Curso
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Curso Modificado Satisfactoriamente 2. No se pudo Actualizar al Curso

Tabla 28. Caso de uso Modificar Curso

2.1.7.5.10.24 Caso de Uso Gestionar Evaluación

Caso de uso	Gestionar Evaluación
Actores	Docente
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a las Evaluación.

Resumen	Los datos introducidos por el administrador o encargado de operaciones son capturados, para luego ser evaluados y poder dar así una respuesta al administrador o encargado de operaciones, de ser correctos los datos se abrirán una interfaz para el administrador del sistema.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de las Evaluaciones registradas en el sistema. 2. El actor realiza la Búsqueda de las Notas en el Listado. 3. El sistema muestra la Evaluación que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado. 5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	Ninguno

Tabla 29. Caso de uso Gestionar Evaluación

2.1.7.5.10.25 Caso de Uso Adicionar Evaluación

Caso de uso	Adicionar Evaluación
Actores	Docente
Propósito	Adicionar una nueva Evaluación
Resumen	Se introducen nuevos datos para adicionar una nueva Evaluación en la base de datos
Precondiciones	1. Acto de haber iniciado sesión 2. Entrar a la pantalla listar Evaluación
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario.

	3. El actor introduce datos de la Evaluación en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Evaluación
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Notas Guardado Satisfactoriamente 2. Datos Inválidos

Tabla 30. Caso de uso Adicionar Evaluación

2.1.7.5.10.26 Caso de Uso Eliminar Evaluación

Caso de uso	Eliminar Evaluación
Actores	Docente
Propósito	Eliminar un registro de Evaluación
Resumen	Se escoge la opción eliminar de la Evaluación que se desee eliminar de la Base de Datos
Precondiciones	1. El administrador deberá escoger la Evaluación que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación de la Evaluación. 4. El sistema regresa a la pantalla de Listar Evaluación
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Nota eliminada Satisfactoriamente

	2. No se pudo eliminar a la Evaluación
--	--

Tabla 31. Caso de uso Eliminar Evaluación

2.1.7.5.10.27 Caso de Uso Modificar Evaluación

Caso de uso	Modificar Evaluación
Actores	Docente
Propósito	Modificar un registro de Evaluación
Resumen	Se escoge la opción modificar de la Evaluación que desee cambiar valores
Precondiciones	1. Haber iniciado sesión 2. Haber entrado a la pantalla de listar Evaluación
Flujo principal	1. El actor presiona en el menú la opción Modificar. 2. El sistema le muestra una nueva ventana con un formulario con datos de las Evaluación 3. El actor realiza los cambios de datos de la Evaluación en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema actualiza los datos del formulario. 7. El sistema regresa a la pantalla de Listar Evaluación
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Evaluación Modificado Satisfactoriamente 2. No se pudo Actualizar la Evaluación

Tabla 32. Caso de uso Modificar Evaluación

2.1.7.5.10.28 Caso de Uso Generar Reporte de Evaluación

Caso de uso	Generar Reporte de Evaluación
Actores	Docente
Propósito	Generar un reporte con la documentación requerida
Resumen	Generar un Pdf.
Precondiciones	1. Haber iniciado sesión
Flujo principal	1. El actor presiona en el menú la opción Reporte. 2. El sistema muestra el Reporte.
Flujo alternativo	2. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Ninguna

Tabla 33. Caso de uso Generar Reporte de Evaluación

2.1.7.5.10.29 Caso de Uso Gestionar Matrícula

Caso de uso	Gestionar Matrícula
Actores	Secretaria
Propósito	Permite Listar, Adicionar, Modificar y Eliminar a los Matrícula.
Resumen	Los datos introducidos por la secretaria son capturados, para luego ser evaluados, de ser correctos los datos se abrirán una interfaz para la secretaria.
Precondiciones	Acto de haber iniciado sesión
Flujo principal	1. El sistema muestra Listado de las Matriculas registradas en el sistema. 2. El actor realiza la Búsqueda de Matriculas en el Listado. 3. El sistema muestra las Matrículas que cumple(n) con los Criterios de Búsqueda. 4. El actor selecciona una fila del Listado.

	5. El actor selecciona la opción que requiere.
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	Ninguno

Tabla 34. Caso de uso Gestionar Matrícula

2.1.7.5.10.30 Caso de Uso Adicionar Matrícula

Caso de uso	Adicionar Matrícula
Actores	Secretaria
Propósito	Adiciona un nueva Matricula
Resumen	Se introducen nuevos datos para adicionar una nueva Matricula en la base de datos
Precondiciones	1. Acto de haber iniciado sesión 2. Entrar a la pantalla listar Estudiantes 3. Entrar a la pantalla listar Matrícula
Flujo principal	1. El actor presiona botón Adicionar. 2. El sistema abre una nueva pantalla con un formulario. 3. El actor introduce datos del Matrícula en el formulario. 4. Selecciona botón Guardar. 5. Se valida datos del formulario que son obligatorios estén introducidos. 6. El sistema guarda datos del formulario. 7. El sistema regresa a la pantalla de listar Matrícula
Flujo alternativo	1. El actor decide no confirmar selección y el sistema no realiza ninguna operación.
Excepciones	1. Matricula Guardada Satisfactoriamente

	2. Datos Inválidos
--	--------------------

Tabla 35. Caso de uso Adicionar Matricula

2.1.7.5.10.31 Caso de Uso Eliminar Matrícula

Caso de uso	Eliminar Matrícula
Actores	Secretaria
Propósito	Eliminar un registro de Matricula
Resumen	Se escoge la opción eliminar de la Matricula que se desee eliminar de la Base de Datos
Precondiciones	1. El administrador deberá escoger el Matrícula que desea eliminar
Flujo principal	1. El actor presiona en el menú la opción Eliminar. 2. El sistema muestra el mensaje de confirmación. 3. Por Si, El sistema realiza la eliminación de la Matricula 4. El sistema regresa a la pantalla de Listar Matricula
Flujo alternativo	1. Por el No, el sistema no realiza ninguna operación
Excepciones	1. Matricula eliminado Satisfactoriamente 2. No se pudo eliminar la Matrícula

Tabla 36. Caso de uso Eliminar Matrícula

2.1.7.5.10.32 Caso de Uso Modificar Matrícula

Caso de uso	Modificar Matrícula
Actores	Secretaria
Propósito	Modificar un registro de Matrícula
Resumen	Se escoge la opción modificar de la Matricula que desee cambiar valores.

Precondiciones	3. Haber iniciado sesión 4. Haber entrado a la pantalla de listar Matrícula
Flujo principal	8. El actor presiona en el menú la opción Modificar. 9. El sistema le muestra una nueva ventana con un formulario con datos del Matrícula 10. El actor realiza los cambios de datos de la Matrícula en el formulario. 11. Selecciona botón Guardar. 12. Se valida datos del formulario que son obligatorios estén introducidos. 13. El sistema actualiza los datos del formulario. 14. El sistema regresa a la pantalla de Listar Matrícula
Flujo alternativo	2. Por el No, el sistema no realiza ninguna operación
Excepciones	3. Matrícula Modificado Satisfactoriamente

Tabla 37. Caso de uso Modificar Matrícula

2.1.7.5.11 Diseño de interfaces

2.1.7.5.11.1 Portal Web y Acceso al Sistema

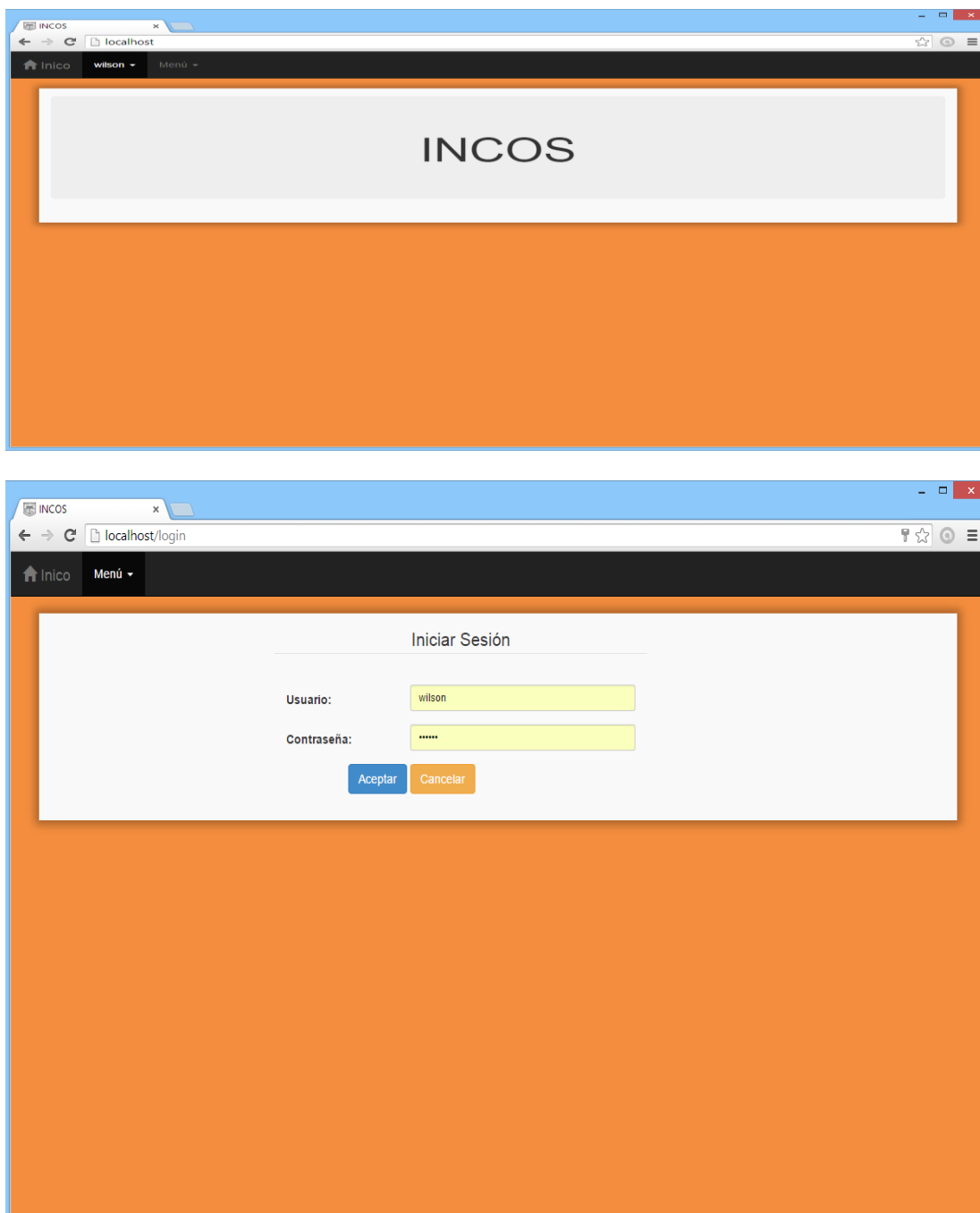


Figura 29. Portal Web y Acceso al Sistema

2.1.7.5.11.2 Interfaz del Jefe Académico

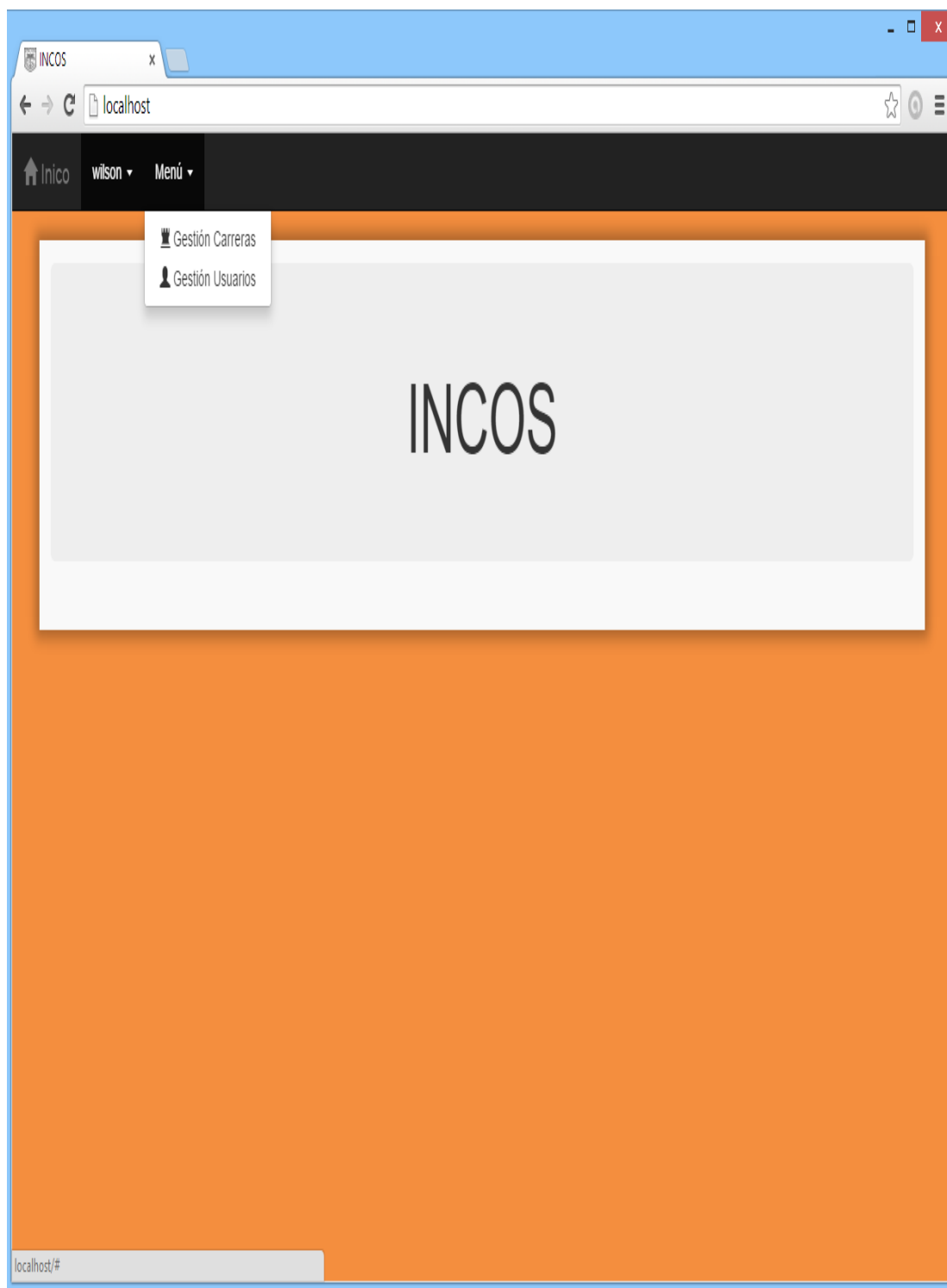


Figura 30. Interfaz del Administrador

2.1.7.5.11.3 Gestión de Usuarios

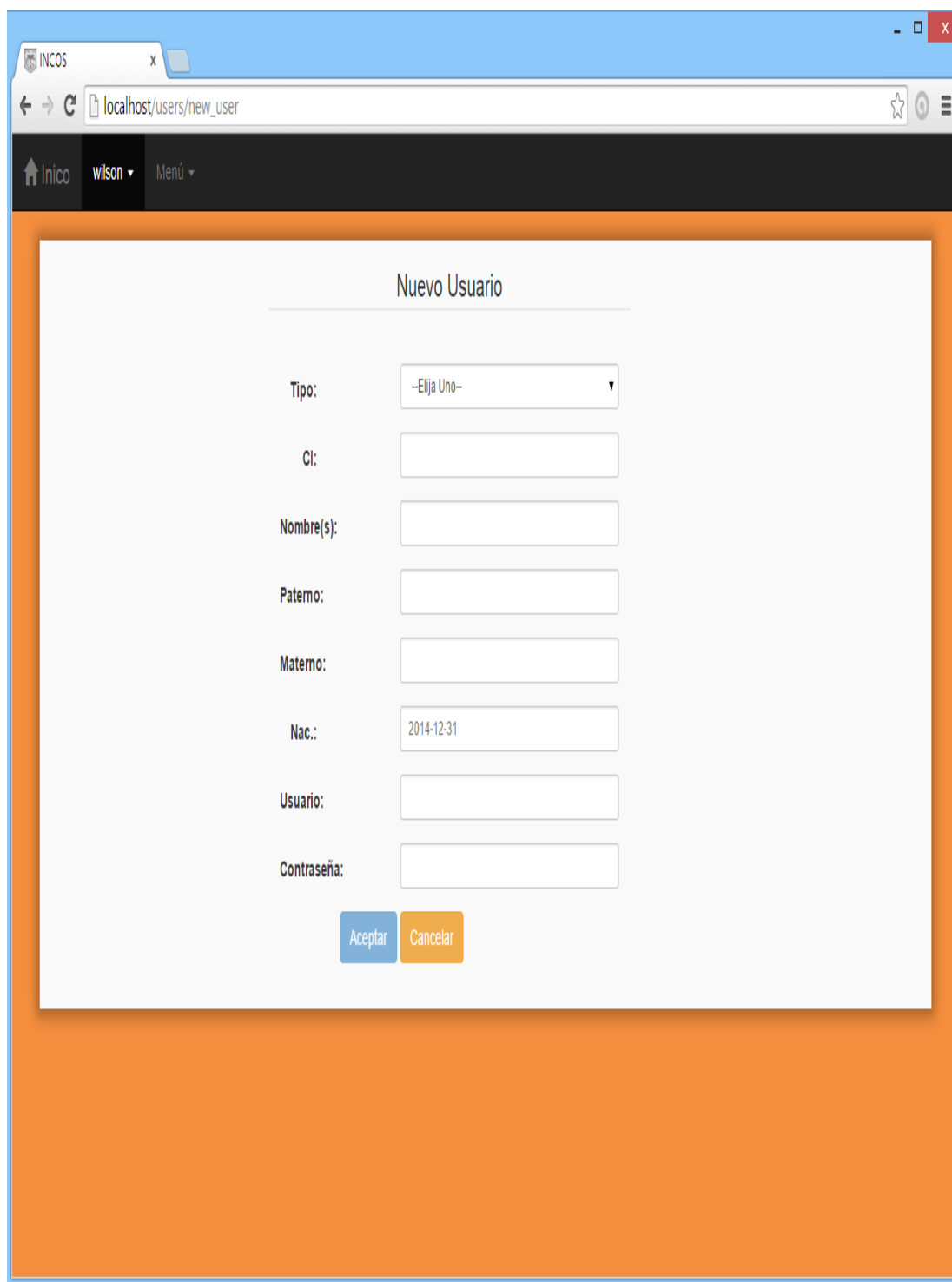
Mostrar 10 datos

#	Nombres y Apellidos	Tipo	Estado	
1	Wilson Luis Narvaez Michel	Jefe	Activo	
2	María René Castillo	Secretaria	Activo	
3	Nelson Rodríguez	Profesor	Activo	
4	Jorge Erazo	Profesor	Activo	
5	Mariela Gallardo Mamani	Profesor	Eliminado	
6	Martín Rosales Perales	Profesor	Activo	
7	Luis Aguilar Aranibar	Profesor	Activo	
8	Janeth Aguirre Cruz	Profesor	Activo	
9	Leonor Vedia Llanos	Profesor	Activo	

Mostrando 1 de 2 Anterior 1 2 Siguiente

Figura 31. Gestión de Usuarios

2.1.7.5.11.4 Nuevo Usuario



The screenshot shows a web browser window with the address bar displaying 'localhost/users/new_user'. The browser has a single tab titled 'INCOS'. The page features a dark navigation bar with a home icon, the text 'Inicio', a user profile 'wilson', and a 'Menú' dropdown. The main content area has an orange border and contains a form titled 'Nuevo Usuario'. The form includes the following fields: 'Tipo' (a dropdown menu with '-Elija Uno-' selected), 'Ci' (a text input), 'Nombre(s)' (a text input), 'Paterno' (a text input), 'Materno' (a text input), 'Nac.' (a date input showing '2014-12-31'), 'Usuario' (a text input), and 'Contraseña' (a text input). At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Nuevo Usuario

Tipo:

Ci:

Nombre(s):

Paterno:

Materno:

Nac.:

Usuario:

Contraseña:

Figura 32. Añadir de Usuario

2.1.7.5.11.5 Añadir Datos

The screenshot shows a web application interface for managing users. A modal form is open for adding a new user, titled "Nelson Rodríguez". The form contains two input fields: "Usuario:" with the value "nelsonr" and "Contraseña:" with masked characters "*****". Below the fields are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

The background interface shows a table of existing users. The table has columns for ID, Name, Role, Status, and Actions. The data is as follows:

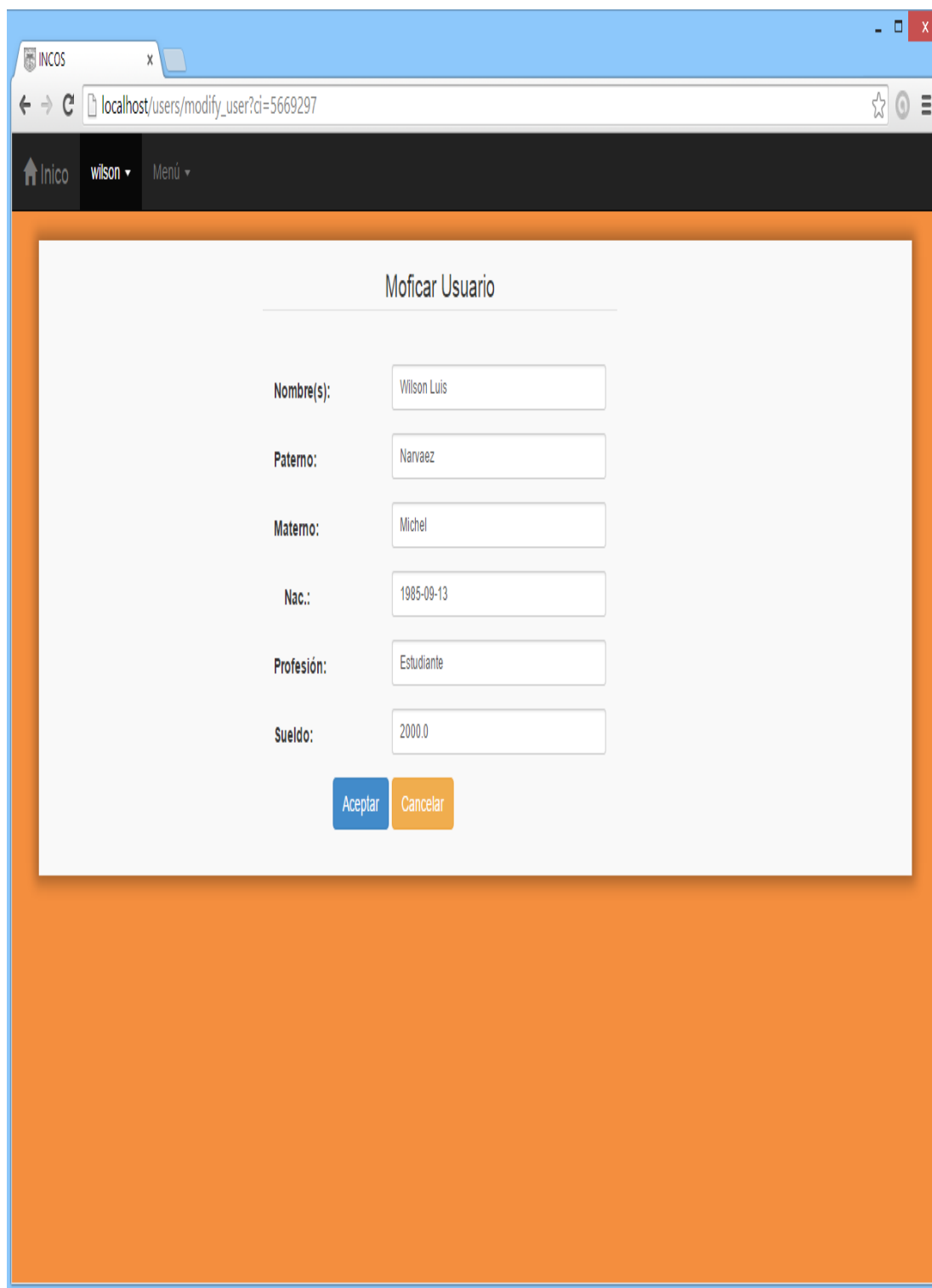
#	Nombre	Cargo	Estado	Acciones
1	Nelson Rodríguez	Profesor	Activo	[Edit] [Delete] [Details]
2	Jorge Erazo	Profesor	Activo	[Edit] [Delete] [Details]
3	Mariela Gallardo Mamani	Profesor	Eliminado	[Edit] [Delete] [Details]
4	Martin Rosales Perales	Profesor	Activo	[Edit] [Delete] [Details]
5	Luis Aguilar Aranibar	Profesor	Activo	[Edit] [Delete] [Details]
6	Janeth Aguirre Cruz	Profesor	Activo	[Edit] [Delete] [Details]
7	Leonor Vedia Llanos	Profesor	Activo	[Edit] [Delete] [Details]

Mostrando 1 de 2

Anterior 1 2 Siguiente

Figura 33. Añadir Datos

2.1.7.5.11.6 Modificar Usuario



The screenshot displays a web browser window with the INCOS logo and a navigation bar. The main content area features a form titled "Modificar Usuario". The form includes the following fields and values:

Field	Value
Nombre(s):	Wilson Luis
Paterno:	Narvaez
Materno:	Michel
Nac.:	1985-09-13
Profesión:	Estudiante
Sueldo:	2000.0

At the bottom of the form, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel).

Figura 34. Modificar Usuario

2.1.7.5.11.7 Eliminar Usuario

Mostrar 10 datos

Buscar

#	Nombres y Apellidos	Tipo	Estado	
1	Wilson Luis Narvaez Michel	Jefe	Activo	
2	María René Castillo	Secretaria	Activo	
3	Nelson Rodríguez	Profesor	Activo	
4	Jorge Erazo	Profesor	Activo	
5	Mariela Gallardo Mamani		Eliminado	
6	Martin Rosales Perales		Activo	
7	Luis Aguilar Aranibar		Activo	
8	Janeth Aguirre Cruz		Activo	
9	Leonor Vedia Llanos		Activo	

Mostrando 1 de 2

Anterior 1 2 Siguiente

Mensaje de la página localhost:

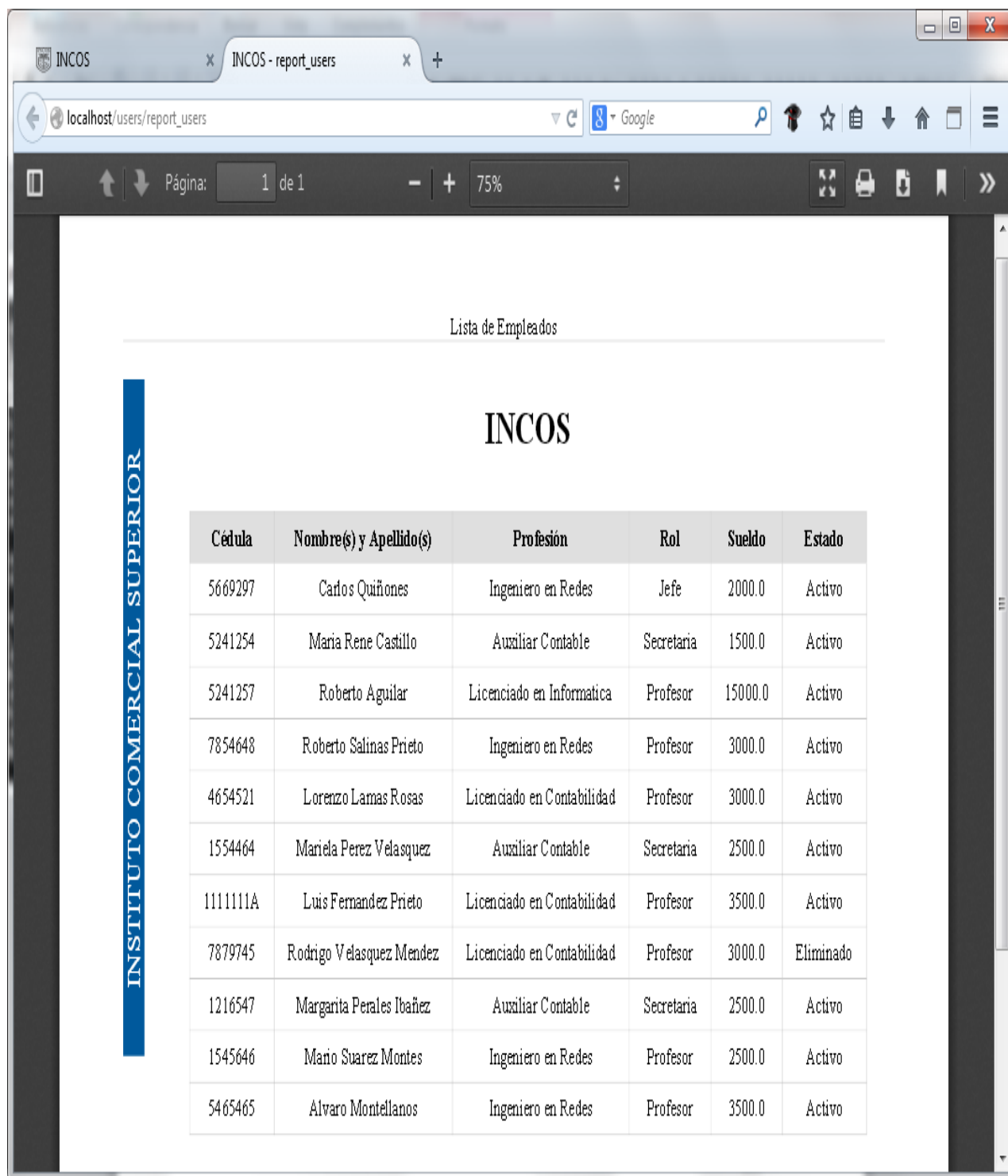
Confirmar Eliminar a Wilson Luis Narvaez?

Aceptar Cancelar

localhost/users/delete_user?ci=5669297

Figura 35. Eliminar Usuario

2.1.7.5.11.8 Reporte Usuarios



Lista de Empleados

INCOS

Cédula	Nombre(s) y Apellido(s)	Profesión	Rol	Sueldo	Estado
5669297	Carlos Quiñones	Ingeniero en Redes	Jefe	2000.0	Activo
5241254	Maria Rene Castillo	Auxiliar Contable	Secretaria	1500.0	Activo
5241257	Roberto Aguilar	Licenciado en Informatica	Profesor	15000.0	Activo
7854648	Roberto Salinas Prieto	Ingeniero en Redes	Profesor	3000.0	Activo
4654521	Lorenzo Lamas Rosas	Licenciado en Contabilidad	Profesor	3000.0	Activo
1554464	Mariela Perez Velasquez	Auxiliar Contable	Secretaria	2500.0	Activo
1111111A	Luis Fernandez Prieto	Licenciado en Contabilidad	Profesor	3500.0	Activo
7879745	Rodrigo Velasquez Mendez	Licenciado en Contabilidad	Profesor	3000.0	Eliminado
1216547	Margarita Perales Ibañez	Auxiliar Contable	Secretaria	2500.0	Activo
1545646	Manio Suarez Montes	Ingeniero en Redes	Profesor	2500.0	Activo
5465465	Alvaro Montellanos	Ingeniero en Redes	Profesor	3500.0	Activo

Figura 36. Reporte de Usuarios

2.1.7.5.11.9 Gestión de Materias

The screenshot displays a web application interface for managing first-year subjects. The browser address bar shows the URL `localhost/subjects/manage_subjects?id_cur=1`. The application header includes a home icon, the name 'Inico', a user dropdown 'wilson', and a 'Menú' dropdown. The main content area features a table titled 'Gestión Materias de Primero'.

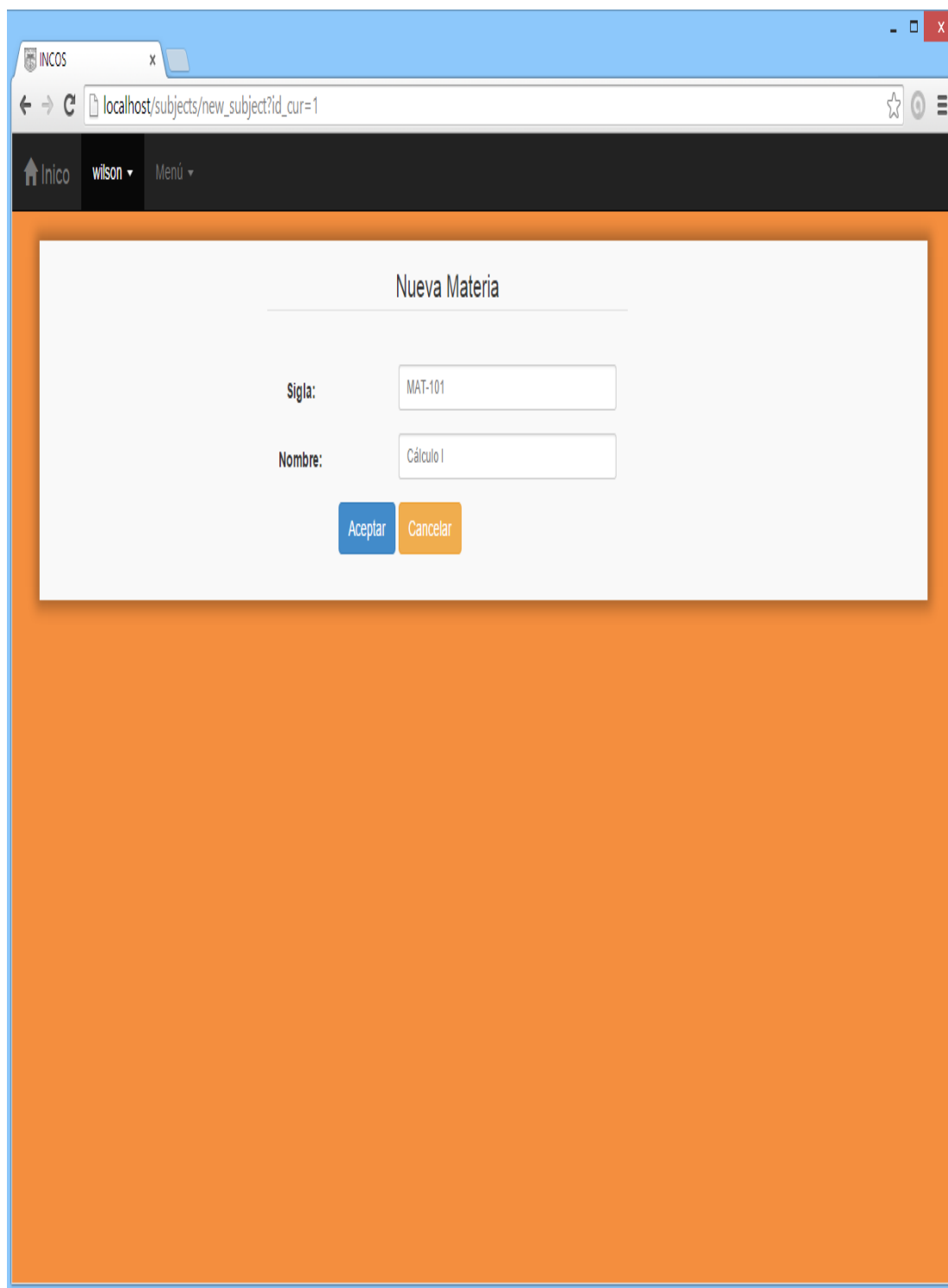
At the top of the table area, there is a 'Mostrar 10 datos' dropdown and a 'Buscar' search box. The table itself has the following structure:

Sigla	Nombre	Grupos	+ ↑
CON-101	Contabilidad I	3	
MAT-102	Matemática Aplicada	3	
IDO-103	Idioma Originario	3	
ICO-104	Informatica Contable	3	
HSM-105	Historia de Sociedades del Mundo	3	
AGL-106	Administración General	1	
DCM-107	Derecho Comercial y Documento Mercantiles	3	
RHP-108	Relaciones Humanas Públicas y Ética Profesional	3	
LEL-109	Legislación Laboral y Seguridad Social	3	

Below the table, there is a 'Mostrando 1 de 1' indicator and a pagination control with 'Anterior', '1' (selected), and 'Siguiente' buttons.

Figura 37. Gestión de Materias

2.1.7.5.11.10 Añadir Materia



The screenshot shows a web browser window with the address bar displaying `localhost/subjects/new_subject?id_cur=1`. The browser's address bar also shows the INCOS logo and a tab labeled 'x'. The page has a dark navigation bar with a home icon, the text 'Inicio', a dropdown menu labeled 'wilson', and another dropdown menu labeled 'Menú'. The main content area has an orange background. In the center, there is a white box titled 'Nueva Materia'. Inside this box, there are two input fields: 'Sigla:' with the value 'MAT-101' and 'Nombre:' with the value 'Cálculo I'. Below these fields are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Nueva Materia

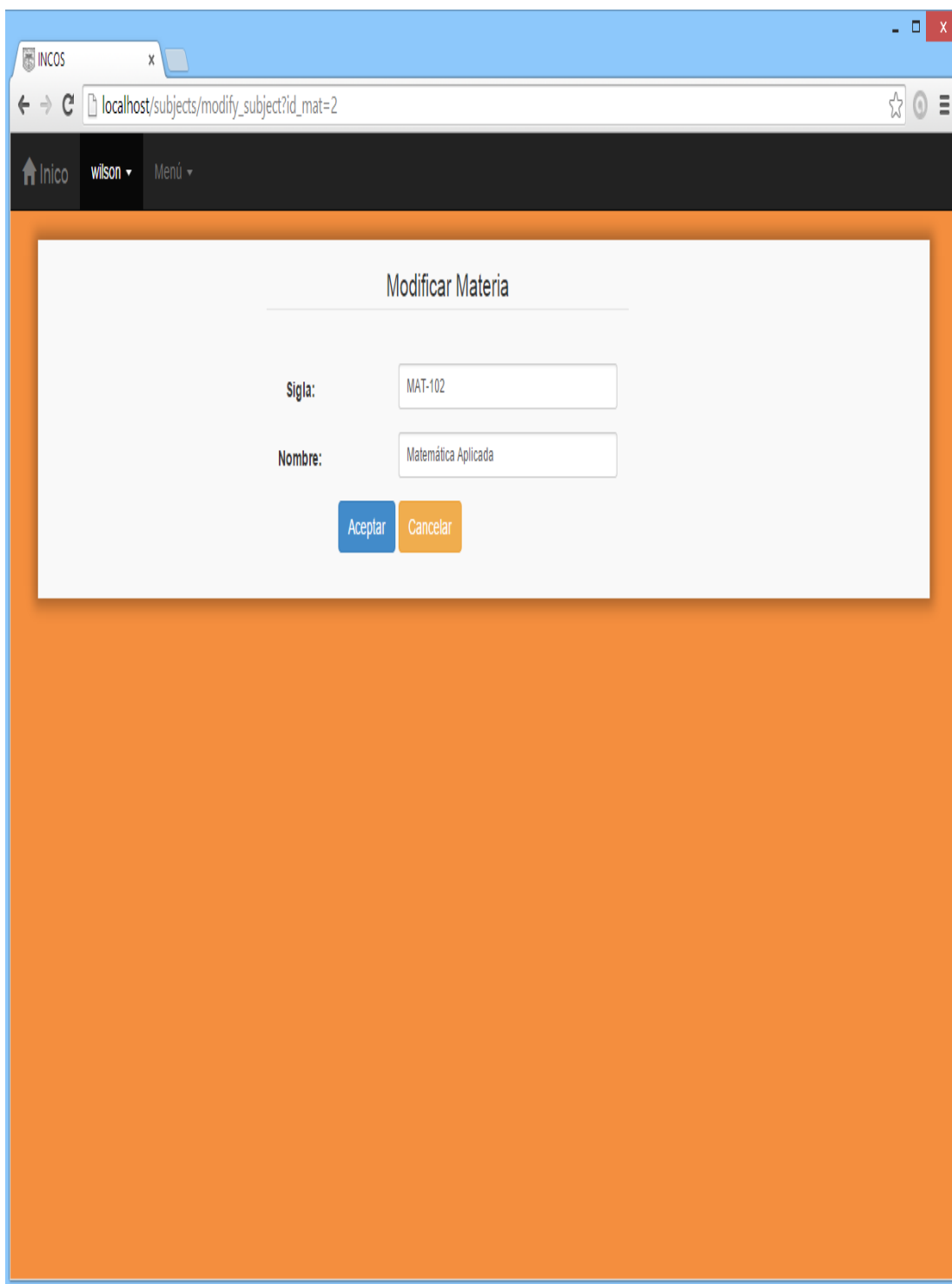
Sigla: MAT-101

Nombre: Cálculo I

Aceptar Cancelar

Figura 38. Añadir Materia

2.1.7.5.11.11 Modificar Materia



The screenshot shows a web browser window with the URL `localhost/subjects/modify_subject?id_mat=2`. The browser's address bar and tabs are visible. The page has a dark header with a home icon, the text 'Inicio', a user profile 'wilson', and a 'Menú' dropdown. The main content area has an orange background. A white modal box titled 'Modificar Materia' is centered on the screen. Inside the modal, there are two input fields: 'Sigla:' with the value 'MAT-102' and 'Nombre:' with the value 'Matemática Aplicada'. Below these fields are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

INCOS x

localhost/subjects/modify_subject?id_mat=2

Inicio wilson Menú

Modificar Materia

Sigla: MAT-102

Nombre: Matemática Aplicada

Aceptar Cancelar

Figura 39. Modificar Materia

2.1.7.5.11.12 Gestión de Carrera

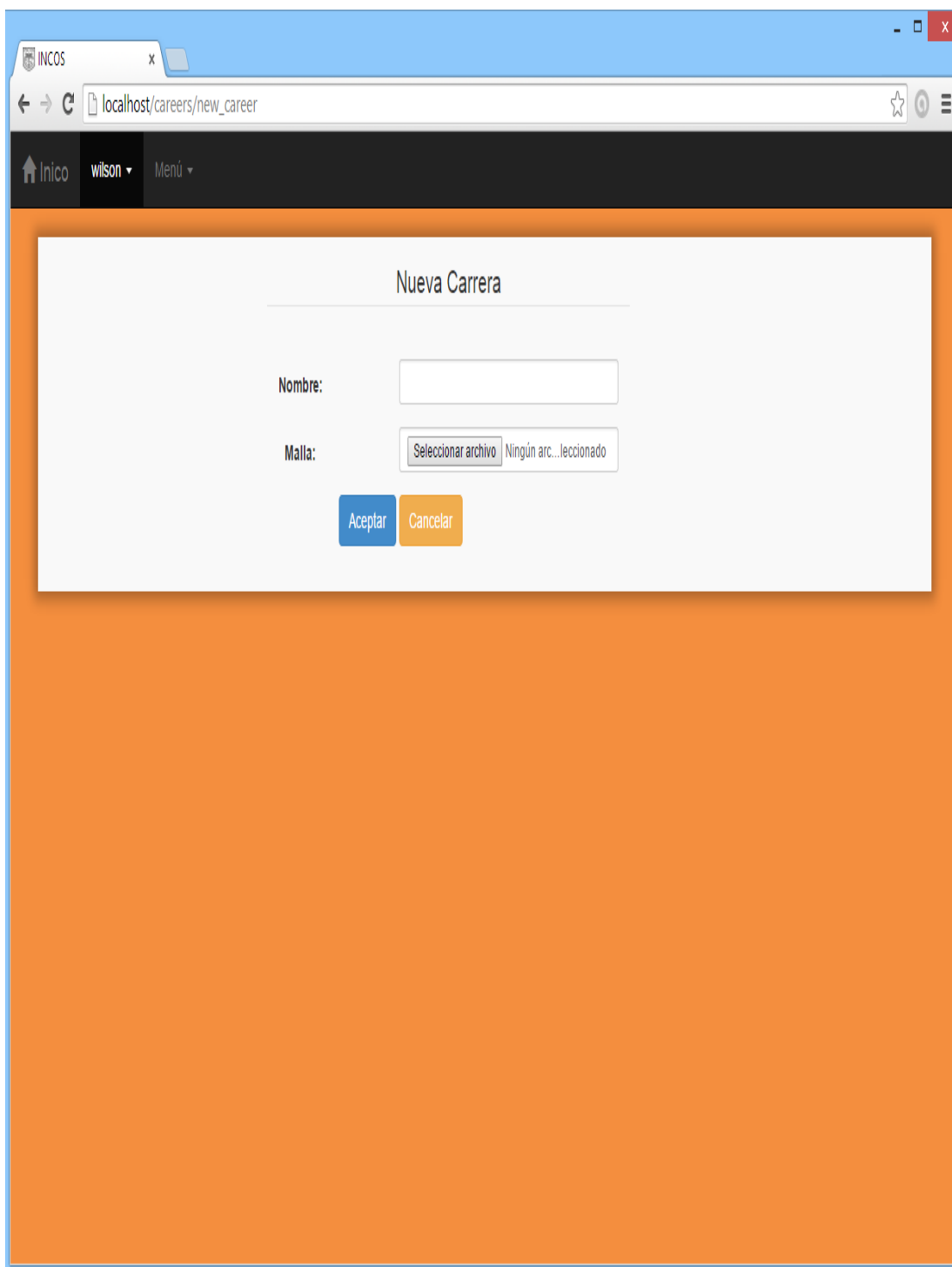
Mostrar 10 datos

Gestión Carreras			
Nombre	Cursos	Profesores	
Contabilidad	2	9	
Informática	1	2	

Mostrando 1 de 1 Anterior 1 Siguiente

Figura 40. Gestión de Carrera

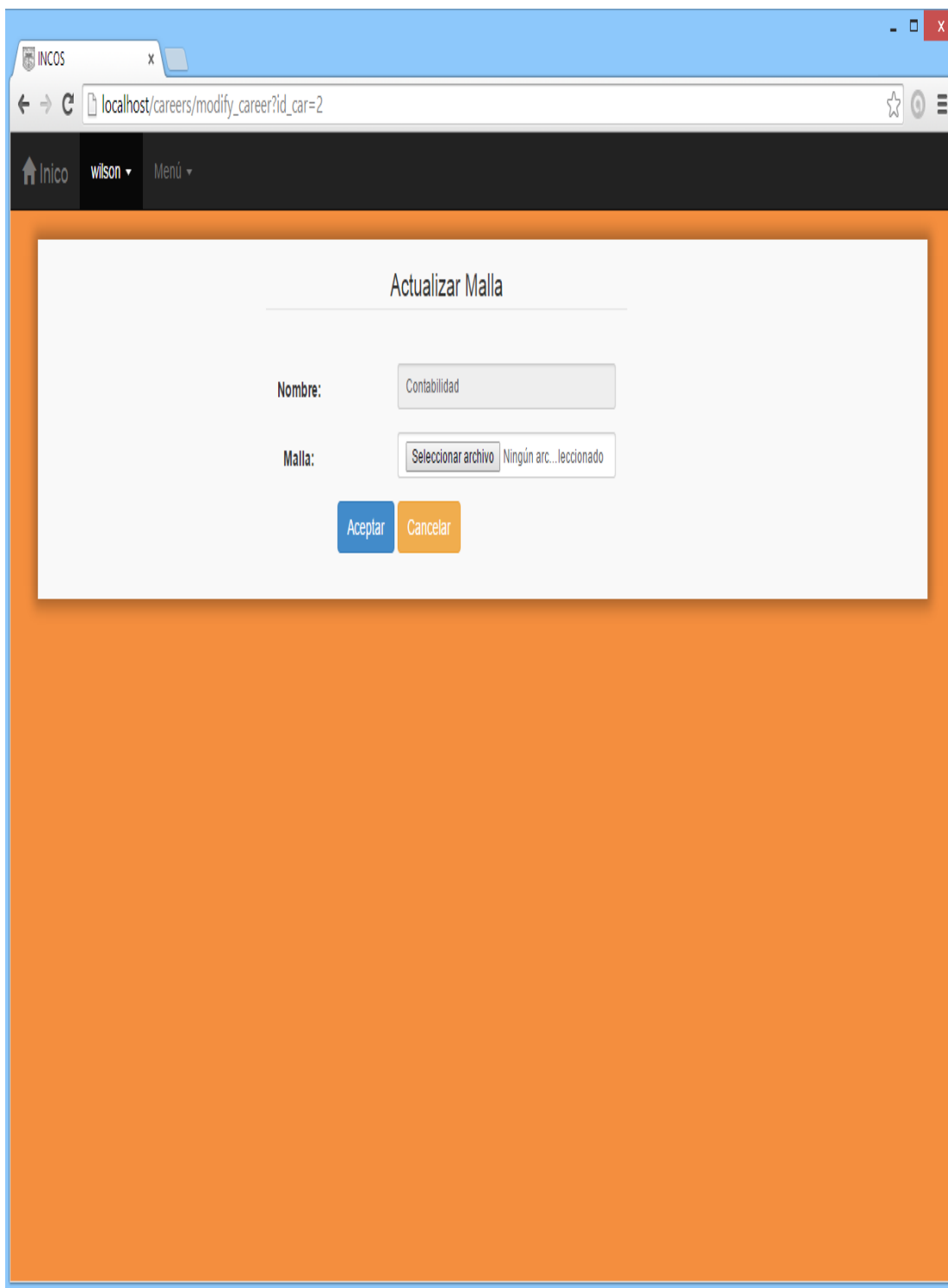
2.1.7.5.11.13 Añadir Carrera



The screenshot shows a web browser window with the address bar displaying `localhost/careers/new_career`. The browser's tab is labeled 'INCOS'. The page has a dark navigation bar at the top with a home icon, the text 'Inicio', a user profile 'wilson' with a dropdown arrow, and a 'Menú' dropdown. The main content area has an orange background. Centered on this background is a white modal box titled 'Nueva Carrera'. Inside the modal, there are two form fields: 'Nombre:' followed by a text input box, and 'Malla:' followed by a file selection button labeled 'Seleccionar archivo' and a text label 'Ningún arc...leccionado'. At the bottom of the modal are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 41. Añadir Carrera

2.1.7.5.11.14 Modificar Carrera



The screenshot shows a web browser window with the URL `localhost/careers/modify_career?id_car=2`. The browser's address bar and tabs are visible. The application's header includes a logo, the name 'Inico', a user profile 'wilson', and a 'Menú' dropdown. The main content area features a white box titled 'Actualizar Malla'. Inside this box, there are two form fields: 'Nombre' with the value 'Contabilidad' and 'Malla' with a dropdown menu showing 'Seleccionar archivo' and 'Ningún arc...leccionado'. At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Actualizar Malla

Nombre: Contabilidad

Malla: Seleccionar archivo Ningún arc...leccionado

Aceptar Cancelar

Figura 42. Modificar Carrera

2.1.7.5.11.15 Gestión Cursos

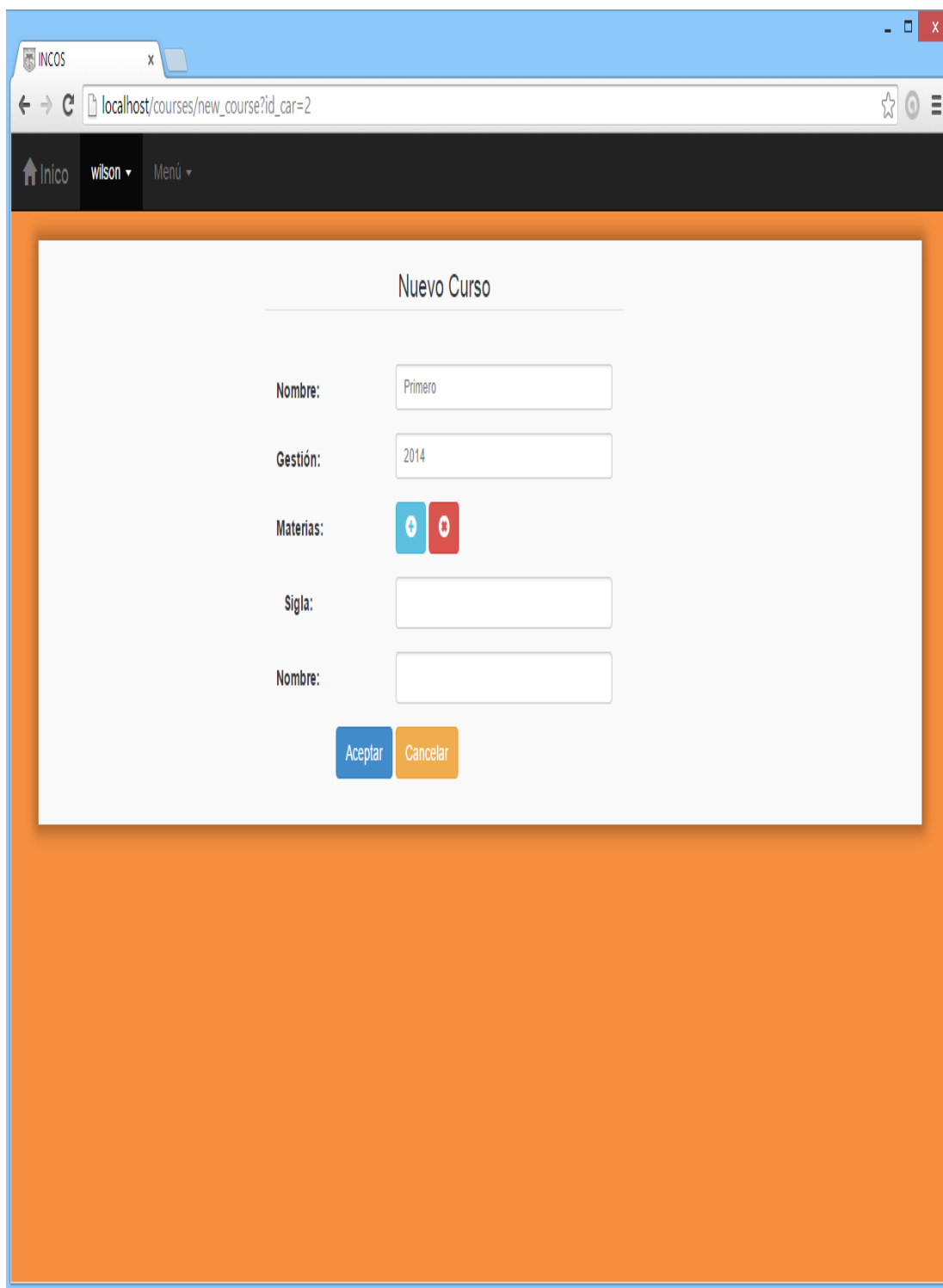
Mostrar 10 datos

Nombre	Gestión	Materias	Grupos	
Primero	2014	9	4	  
Segundo	2014	1	0	  

Mostrando 1 de 1 Anterior 1 Siguiente

Figura 43. Eliminar Curso

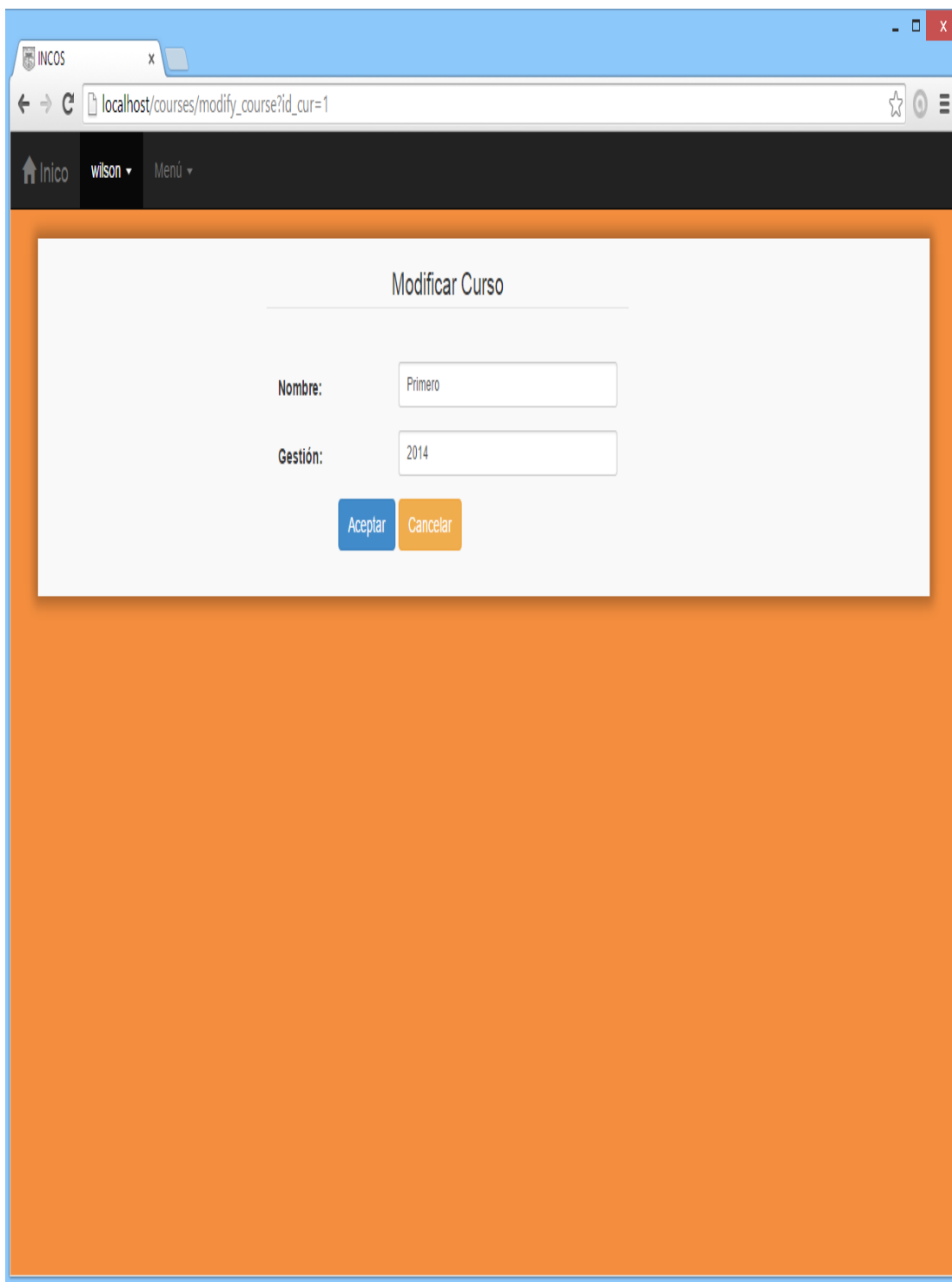
2.1.7.5.11.16 Añadir Curso



The screenshot shows a web browser window with the address bar displaying `localhost/courses/new_course?id_car=2`. The browser's tab is labeled 'INCOS'. The page has a dark navigation bar with a home icon, the text 'Inicio', a user profile 'wilson' with a dropdown arrow, and a 'Menú' dropdown. The main content area has an orange background. A white modal form titled 'Nuevo Curso' is centered on the screen. The form contains the following fields: 'Nombre:' with the value 'Primero', 'Gestión:' with the value '2014', 'Materias:' with two colored buttons (blue with a plus icon and red with a minus icon), 'Sigla:', and another 'Nombre:'. At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 44. Añadir Curso

2.1.7.5.11.17 Modificar Curso



The screenshot shows a web browser window with the address bar displaying `localhost/courses/modify_course?id_cur=1`. The browser's tab is labeled 'INCOS'. The page has a dark navigation bar with a home icon, the text 'Inicio', a user profile 'wilson', and a 'Menú' dropdown. The main content area has an orange background. Centered on this background is a white box titled 'Modificar Curso'. Inside this box, there are two form fields: 'Nombre:' with the value 'Primero' and 'Gestión:' with the value '2014'. Below these fields are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 45. Modificar Curso

2.1.7.5.11.18 Gestión Grupos

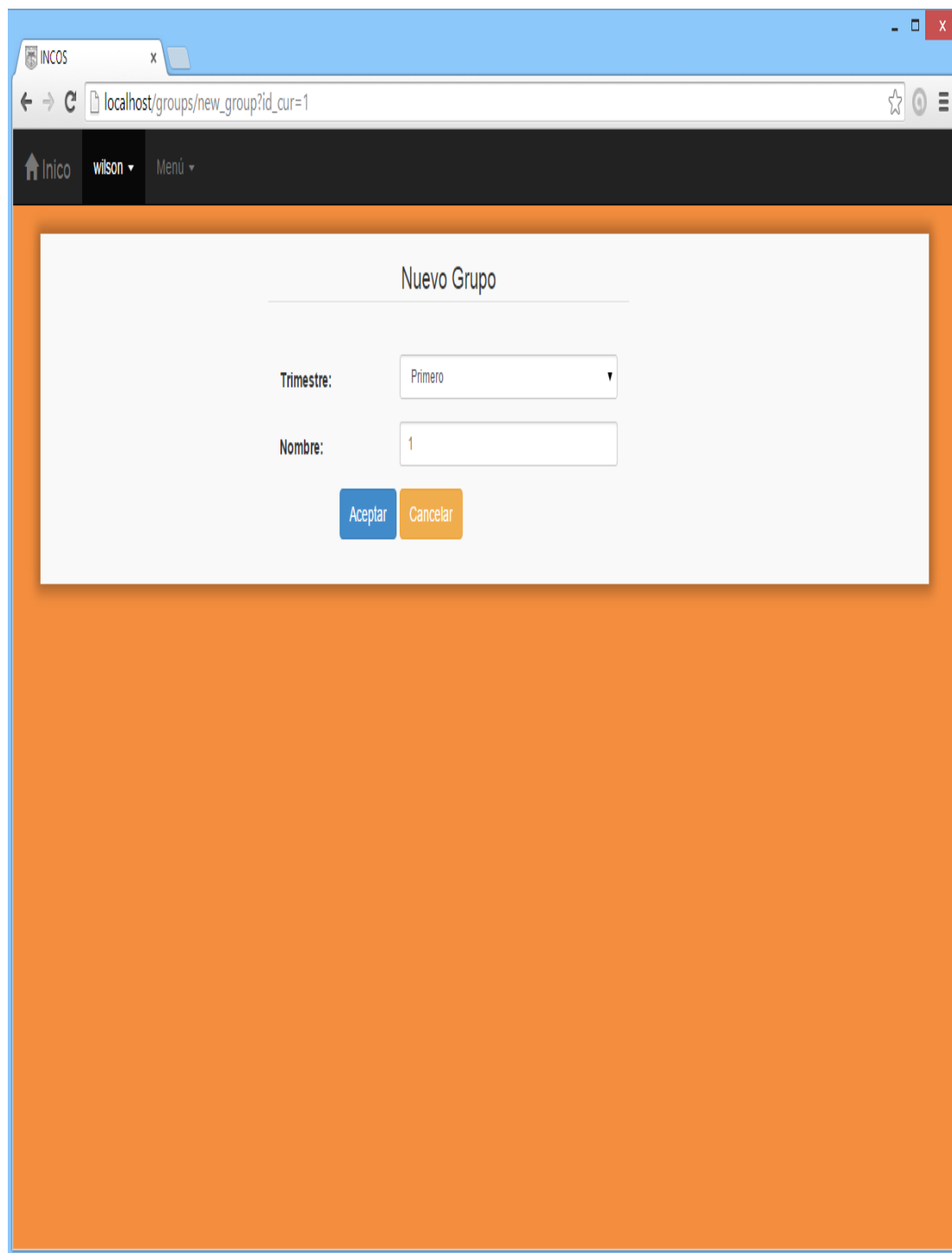
Mostrar 10 datos

Grupo	Trimestre	Gestión	Materias	Alumnos	
1	1	2014	9	10	 
1	2	2014	8	10	 
1	3	2014	8	10	 
2	1	2014	0	0	

Mostrando 1 de 1 Anterior 1 Siguiente

Figura 46. Gestión Grupos

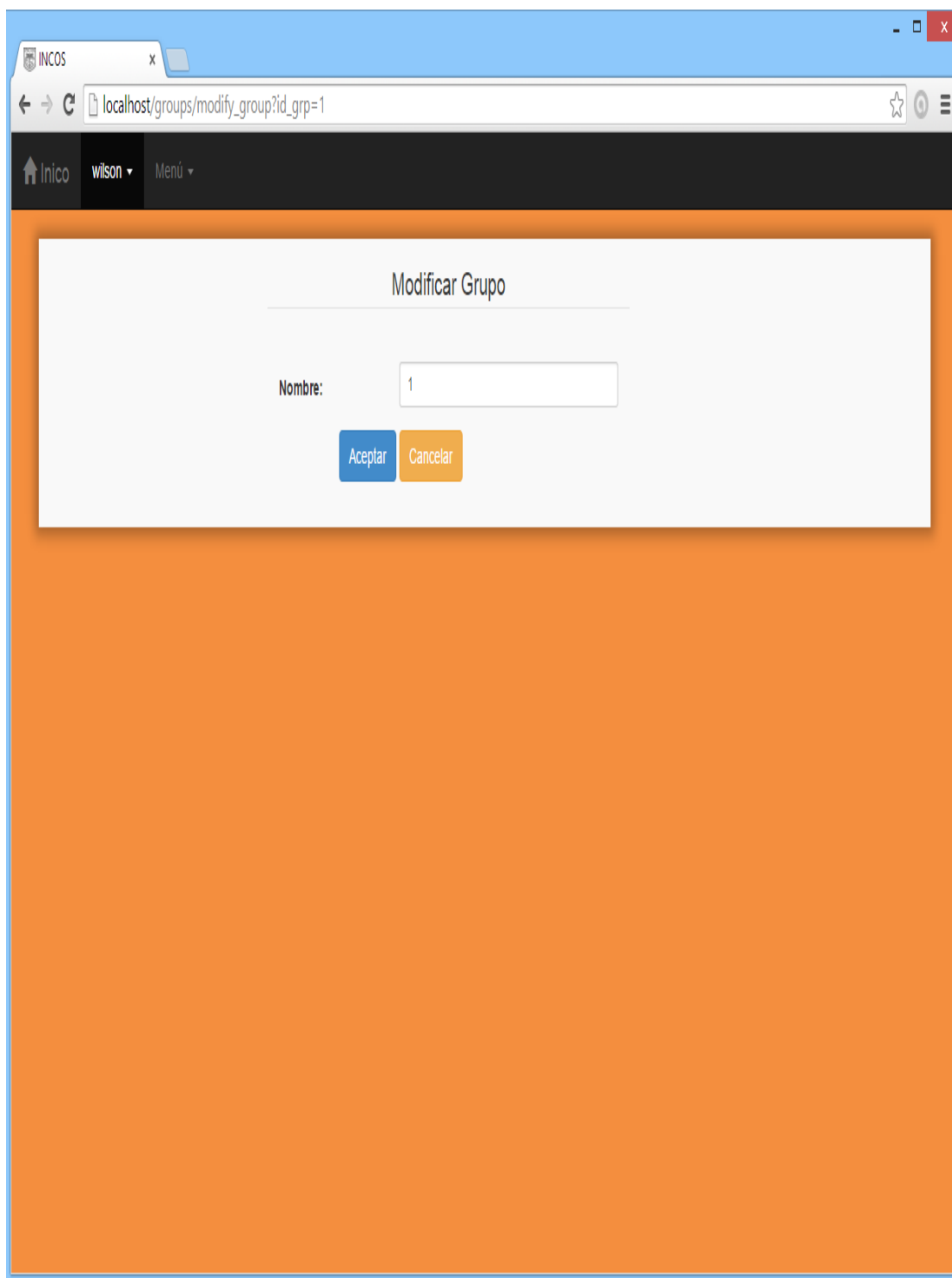
2.1.7.5.11.19 Añadir Grupos



The screenshot shows a web browser window with the title 'INCOS' and a single tab. The address bar displays 'localhost/groups/new_group?id_cur=1'. The browser's navigation bar includes a home icon, the text 'Inicio', a user profile 'wilson' with a dropdown arrow, and a 'Menú' dropdown. The main content area has an orange background. Centered on this background is a white rectangular form titled 'Nuevo Grupo'. The form contains two input fields: 'Trimestre' with a dropdown menu currently showing 'Primero', and 'Nombre' with a text input field containing the number '1'. At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 47. Añadir Grupos

2.1.7.5.11.20 Modificar Grupos



The screenshot shows a web browser window with the address bar displaying `localhost/groups/modify_group?id_grp=1`. The browser's address bar includes navigation icons (back, forward, refresh) and a search icon. The browser's tab bar shows a single tab titled 'INCOS'. The browser's window title bar shows standard OS controls (minimize, maximize, close). The browser's content area displays a web page with a dark blue header bar. The header bar contains a home icon, the text 'Inicio', a dropdown menu labeled 'wilson', and another dropdown menu labeled 'Menú'. The main content area of the browser is orange. Centered on the orange background is a white rectangular form titled 'Modificar Grupo'. The form has a label 'Nombre:' followed by a text input field containing the number '1'. Below the input field are two buttons: a blue button labeled 'Aceptar' and an orange button labeled 'Cancelar'.

INCOS x

localhost/groups/modify_group?id_grp=1

Inicio wilson Menú

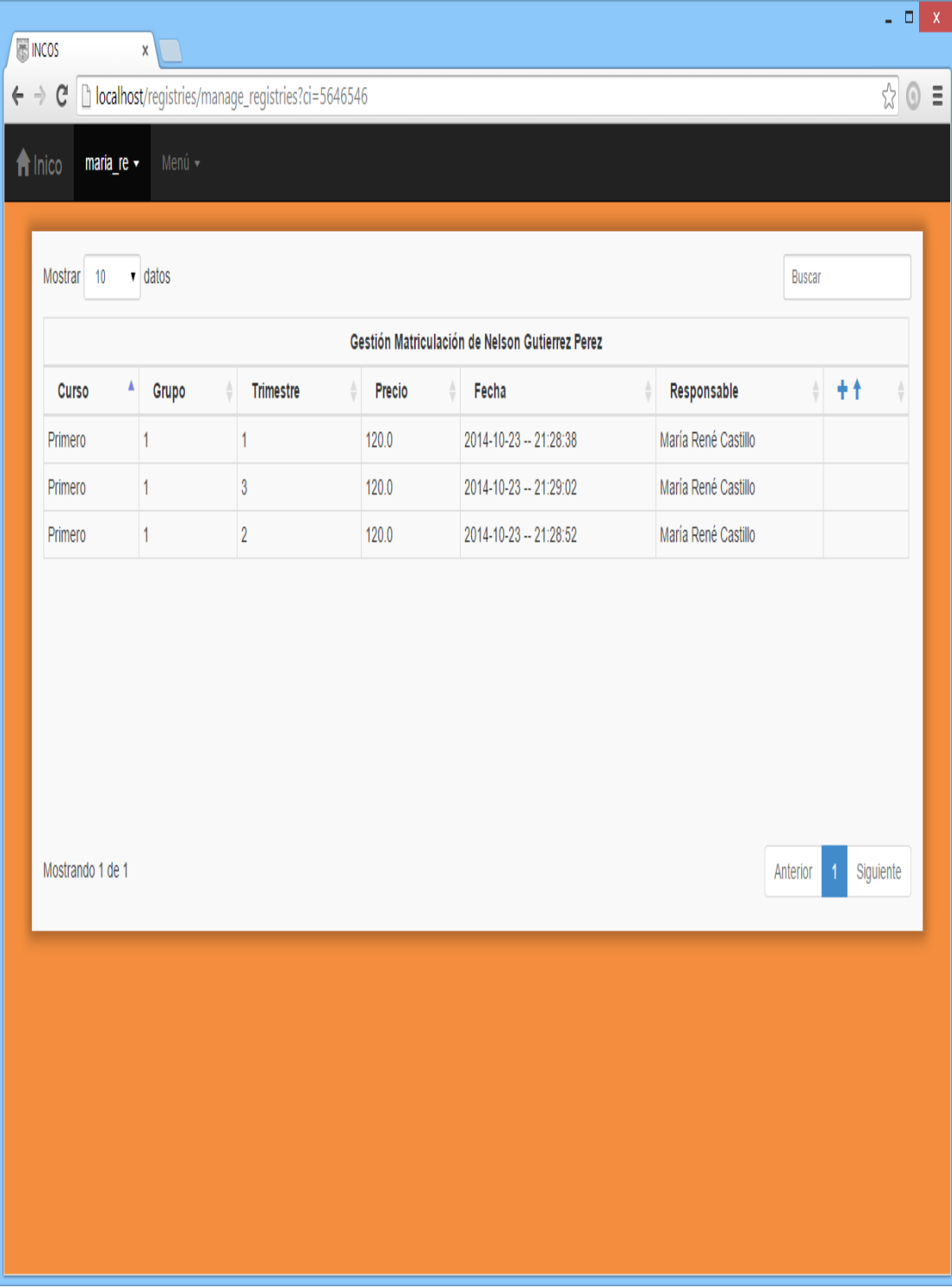
Modificar Grupo

Nombre: 1

Aceptar Cancelar

Figura 48. Modificar Grupos

2.1.7.5.11.21 Gestión Matricula

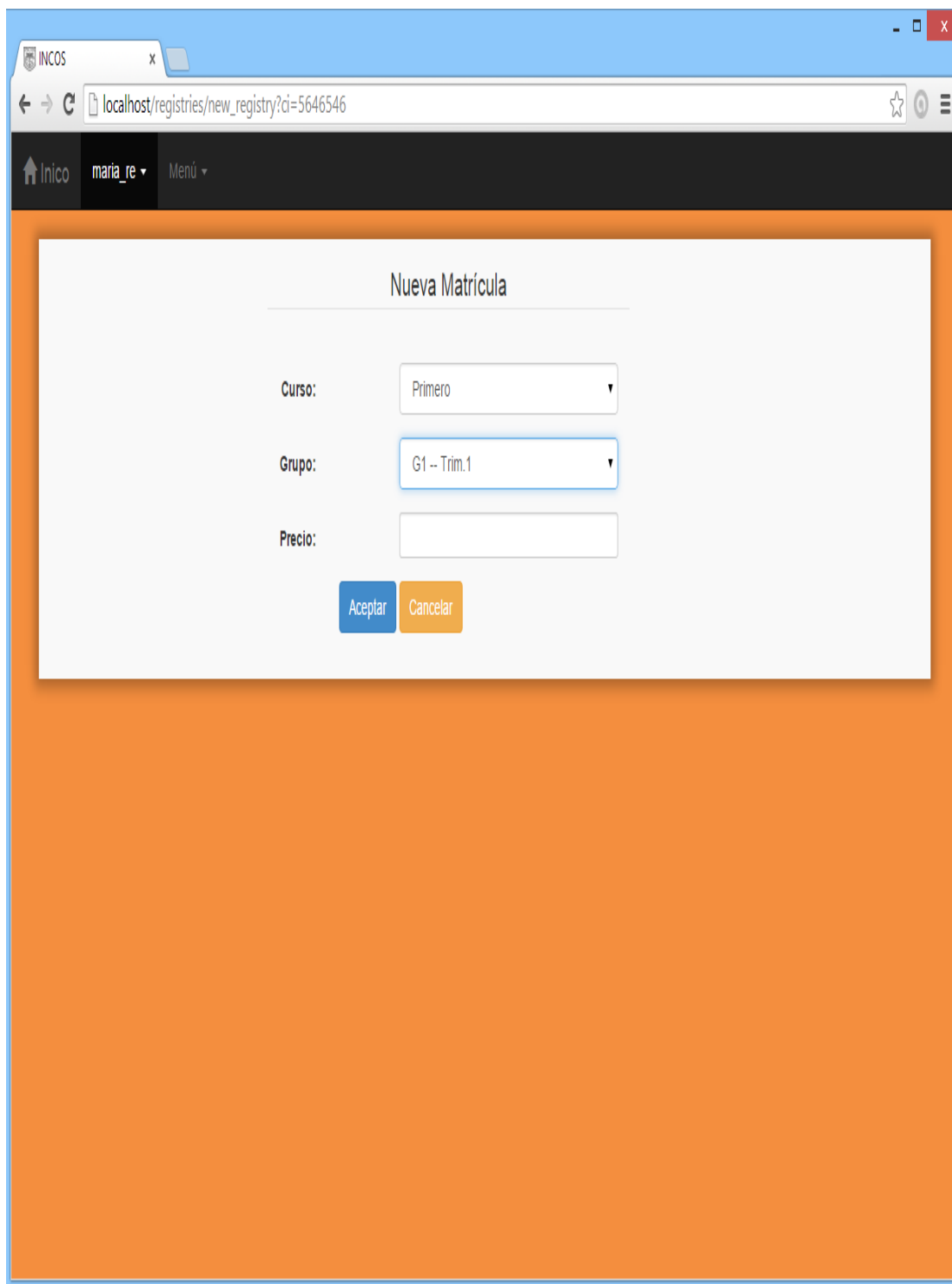


The screenshot shows a web browser window with the URL `localhost/registries/manage_registries?ci=5646546`. The application has a dark navigation bar with a home icon, the text "Inicio", a user profile "maria_re", and a "Menú" dropdown. The main content area has an orange background and contains a table titled "Gestión Matriculación de Nelson Gutierrez Perez". Above the table is a search bar labeled "Buscar" and a "Mostrar" dropdown set to "10" with a "datos" link. The table has columns for "Curso", "Grupo", "Trimestre", "Precio", "Fecha", and "Responsable", with sorting arrows. It displays three rows of enrollment data. At the bottom, it shows "Mostrando 1 de 1" and navigation buttons "Anterior", "1", and "Siguiete".

Curso	Grupo	Trimestre	Precio	Fecha	Responsable
Primero	1	1	120.0	2014-10-23 -- 21:28:38	María René Castillo
Primero	1	3	120.0	2014-10-23 -- 21:29:02	María René Castillo
Primero	1	2	120.0	2014-10-23 -- 21:28:52	María René Castillo

Figura 49. Gestión Matricula

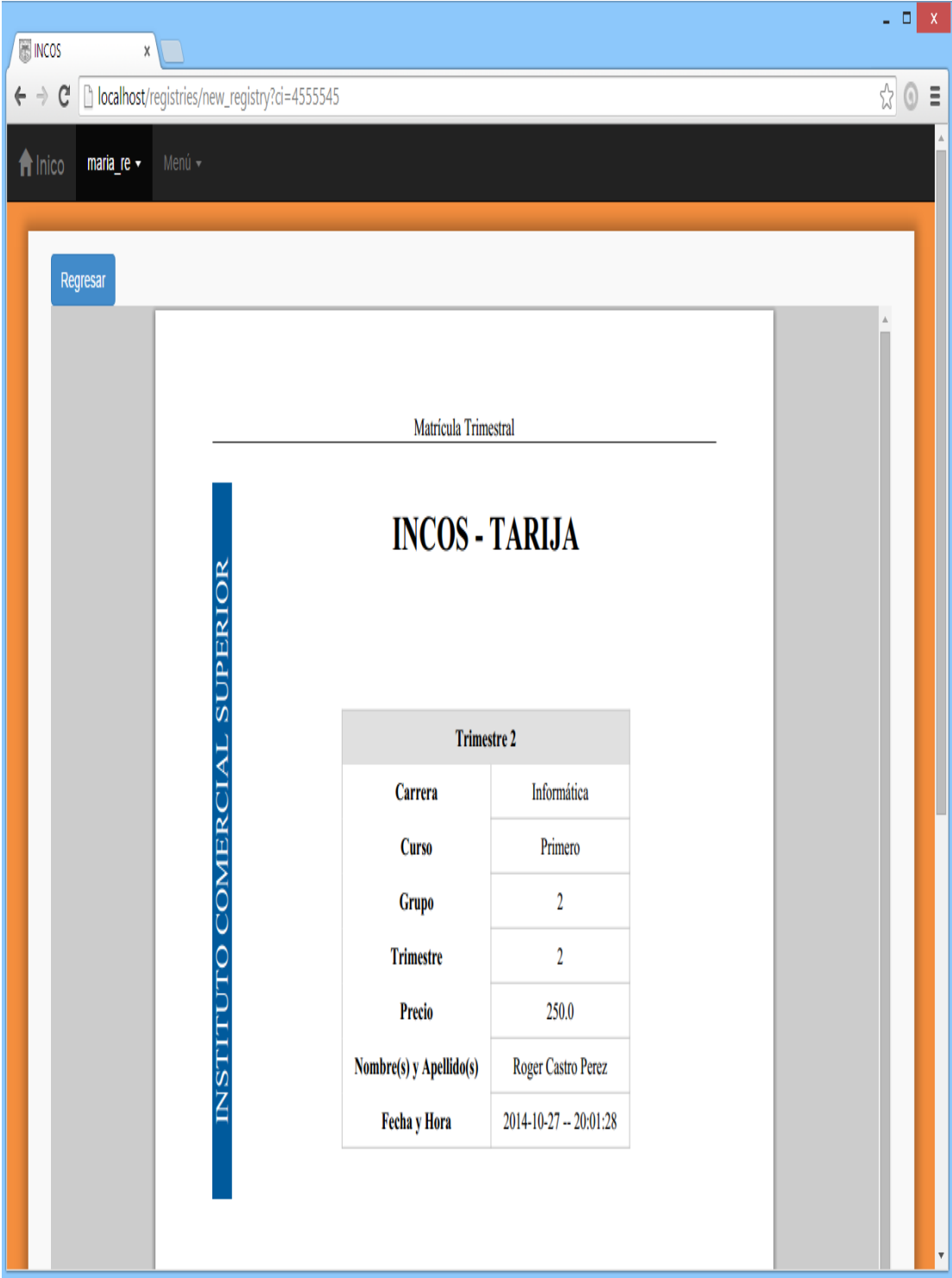
2.1.7.5.11.22 Añadir Matricula



The screenshot shows a web browser window with the URL `localhost/registries/new_registry?ci=5646546`. The browser's address bar and tabs are visible. The page has a dark navigation bar with a home icon, the text 'Inicio', a user profile 'maria_re' with a dropdown arrow, and a 'Menú' dropdown. The main content area has an orange background. A white modal box titled 'Nueva Matricula' is centered on the screen. Inside the modal, there are three form fields: 'Curso' with a dropdown menu showing 'Primero', 'Grupo' with a dropdown menu showing 'G1 -- Trim.1', and 'Precio' with an empty text input field. Below these fields are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 50. Añadir Matricula

2.1.7.5.11.23 Reporte de Matricula



Regresar

Matricula Trimestral

INCOS - TARIJA

INSTITUTO COMERCIAL SUPERIOR

Trimestre 2	
Carrera	Informática
Curso	Primero
Grupo	2
Trimestre	2
Precio	250.0
Nombre(s) y Apellido(s)	Roger Castro Perez
Fecha y Hora	2014-10-27 -- 20:01:28

Figura 51. Reporte de Matricula

2.1.7.5.11.24 Gestión Evaluación

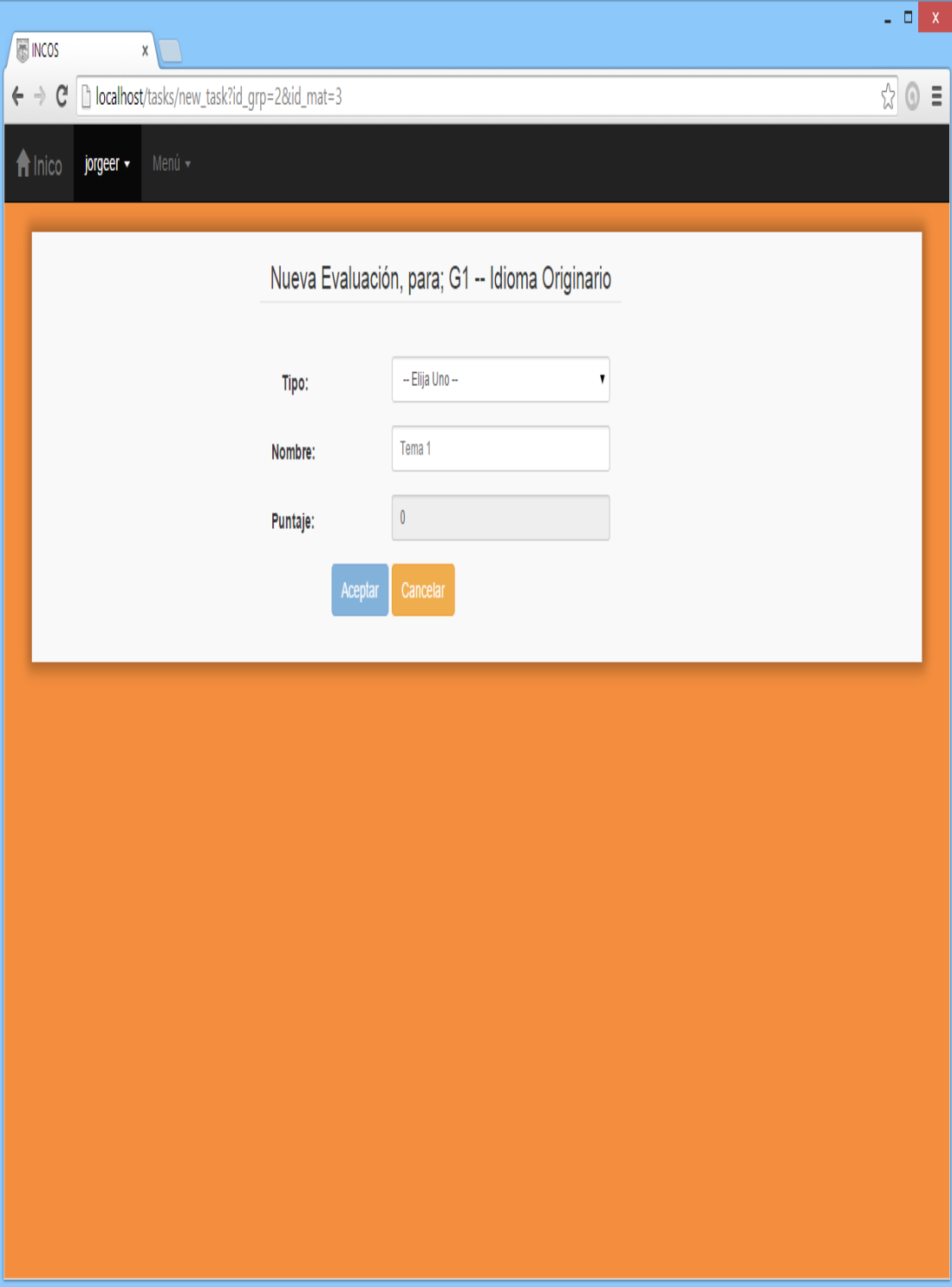
Mostrar 10 datos

Gestión Evaluación Continua de Contabilidad, Primero, G1, Trim.2 y Idioma Originario						
#	Nombre	Puntaje	Tipo	Fecha y Hora	Notas	+ 👤 ↑
1	Tema 1	15.0	Práctico	2014-10-24 11:32:29	10	👤
2	Examen 1	80.0	Examen	2014-10-25 10:23:04	10	👤
3	Limites	5.0	Proyecto	2014-10-25 10:23:22	10	👤

Mostrando 1 de 1 Anterior 1 Siguiente

Figura 52. Gestión Evaluación

2.1.7.5.11.25 Añadir Evaluación



INCOS x

localhost/tasks/new_task?id_grp=2&id_mat=3

Inico jorgeer Menú

Nueva Evaluación, para; G1 -- Idioma Originario

Tipo:

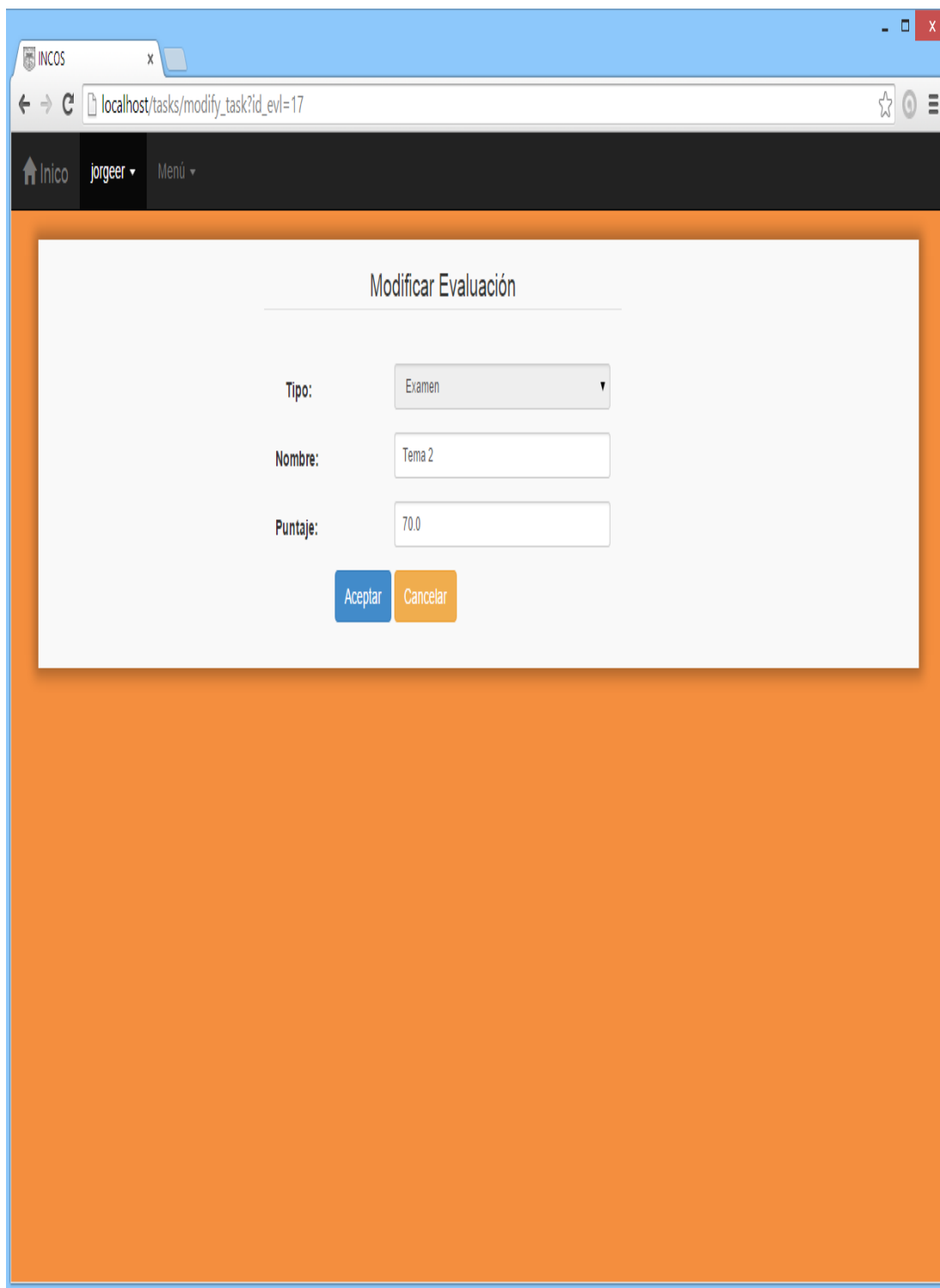
Nombre:

Puntaje:

Aceptar Cancelar

Figura 53. Añadir Evaluación

2.1.7.5.11.26 Modificar Evaluación



The screenshot shows a web browser window with the URL `localhost/tasks/modify_task?id_evl=17`. The browser's address bar and tabs are visible. The web application has a dark navigation bar with a home icon, the text 'Inicio', a user profile 'jorgeer', and a 'Menú' dropdown. The main content area has an orange background. A white modal form titled 'Modificar Evaluación' is centered on the screen. The form contains three input fields: 'Tipo' with a dropdown menu showing 'Examen', 'Nombre' with a text input containing 'Tema 2', and 'Puntaje' with a text input containing '70.0'. At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Modificar Evaluación

Tipo: Examen

Nombre: Tema 2

Puntaje: 70.0

Aceptar Cancelar

Figura 54. Modificar Evaluación

2.1.7.5.11.27 Pantalla Asignaciones

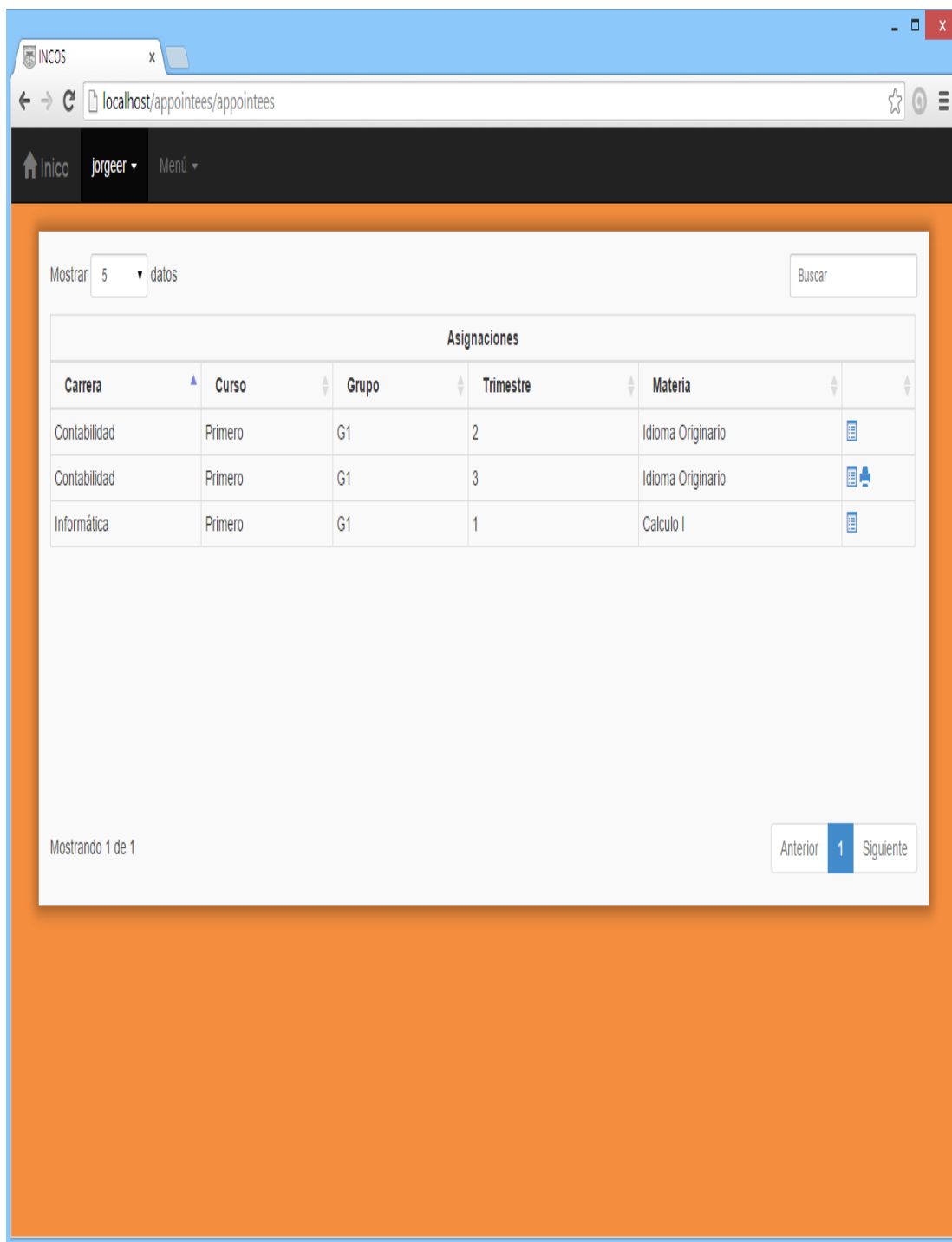


Figura 55. Pantalla Asignaciones

2.1.7.5.11.28 Asignar Docente

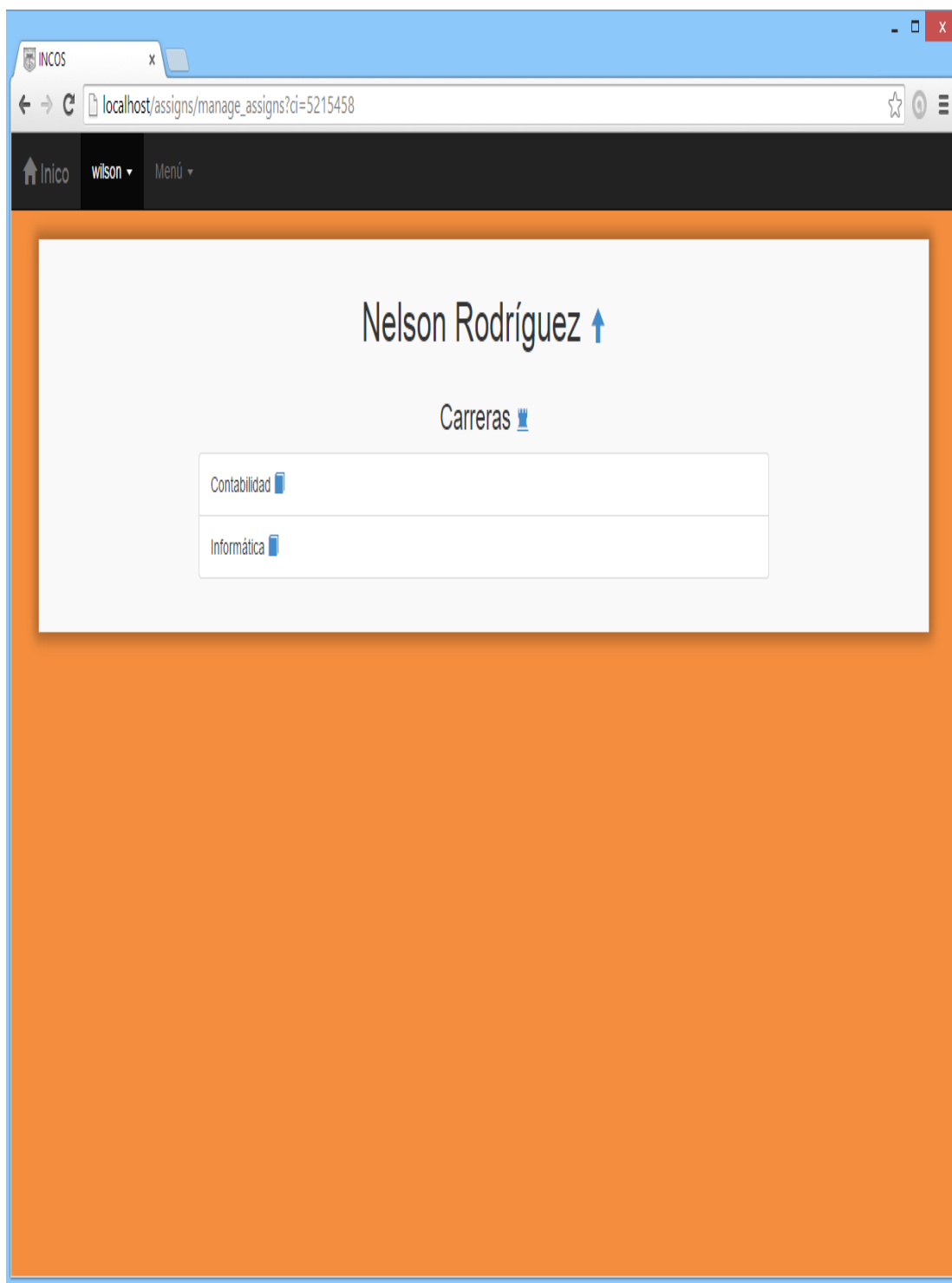


Figura 56. Asignar Docente

2.1.7.5.11.29 Asignar Carreras

Asignar Carreras

☒	Informática
☒	Contabilidad

Aceptar Cancelar

Figura 57. Asignar Carreras

2.1.7.5.11.30 Asignar Materias y Grupos

The screenshot shows a web browser window with the URL `localhost/assigns/assign_subjects?ci=5215458&id_car=2`. The page title is "Asignar Materias y Grupos en Contabilidad". The interface includes a course selection dropdown set to "Primero", a table of group and subject assignments, and "Aceptar" and "Cancelar" buttons.

Curso	
Primero	

☒	G1 trim.2	▼	Matemática Aplicada	▼
☒	G1 trim.3	▼	Matemática Aplicada	▼

Aceptar Cancelar

Figura 58. Asignar Materias y Grupos

2.1.7.5.11.31 Ponderación

The screenshot displays a web browser window with the URL `localhost/appointees/define_credits?id_grp=3&id_mat=9`. The browser's address bar shows the INCOS logo and a tab labeled 'x'. The page features a dark navigation bar with a home icon, the text 'Inicio', a dropdown menu labeled 'pedroloy', and another dropdown labeled 'Menú'. The main content area has an orange background and contains a white box with the title 'Ponderación de Legislación Laboral y Seguridad Social 100'. Below the title, there are three input fields labeled 'Examen:', 'Practico:', and 'Proyecto:'. At the bottom of the white box are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 59. Ponderación

2.1.7.5.11.32 Reporte de Notas Anual

Materia: Idioma Originario | Gestión: 2014

N°	Nombre(s) y Apellido(s)	1° TRIM	2° TRIM	3° TRIM	Final	Promedio	Observaciones
1	Weymar Borda Vargas	43.50	81.00	70.00	80.00	72.42	Aprobado
2	Aurelia Condori Condori	52.00	30.00	74.00	50.00	51.00	Aprobado
3	Sebastian Gallo Soruco	53.00	59.00	72.00	65.00	63.17	Aprobado
4	Eduardo Gudño Aguirre	77.00	59.00	93.00	52.00	64.17	Aprobado
5	Nelson Gutierrez Perez	81.00	67.00	95.00	35.00	58.00	Aprobado
6	Gabriela Heredia Gallardo	65.00	76.00	95.00	65.00	71.83	Aprobado
7	Ximena Martinez Marquez	73.00	80.00	84.00	54.00	66.50	Aprobado
8	Lidia Ramos	69.75	74.00	91.00	0.00	39.12	Reprobado
9	Roman Ricalde Apaza	34.50	76.00	71.00	54.00	57.25	Aprobado
10	Alberta Rodriguez Tejerina	32.00	61.00	88.00	65.00		

Figura 60. Reporte de Notas Anual

2.1.7.5.12 Modelo de Diagrama de Actividades

2.1.7.5.12.1 Introducción

El diagrama de Actividades es un artefacto de la disciplina Análisis de Sistemas en la metodología RUP la cual estamos implementando.

Los Diagramas de Actividades se Utilizan para modelar aspectos dinámicos de un Sistema, esto implica modelar los pasos secuenciales de un proceso.

2.1.7.5.12.2 Propósito

Comprender la estructura y la dinámica del sistema deseado.

Identificar posibles mejoras en el Sistema.

2.1.7.5.12.3 Alcance

Describir los procesos del sistema y los clientes.

Identificar y definir los procesos de los casos de uso según los objetivos de la organización.

Definir un diagrama de actividad para cada caso de uso del sistema.

2.1.7.5.13 Diagrama de Actividades

2.1.7.5.13.1 Diagrama de Actividad: Caso de uso Iniciar Sesión

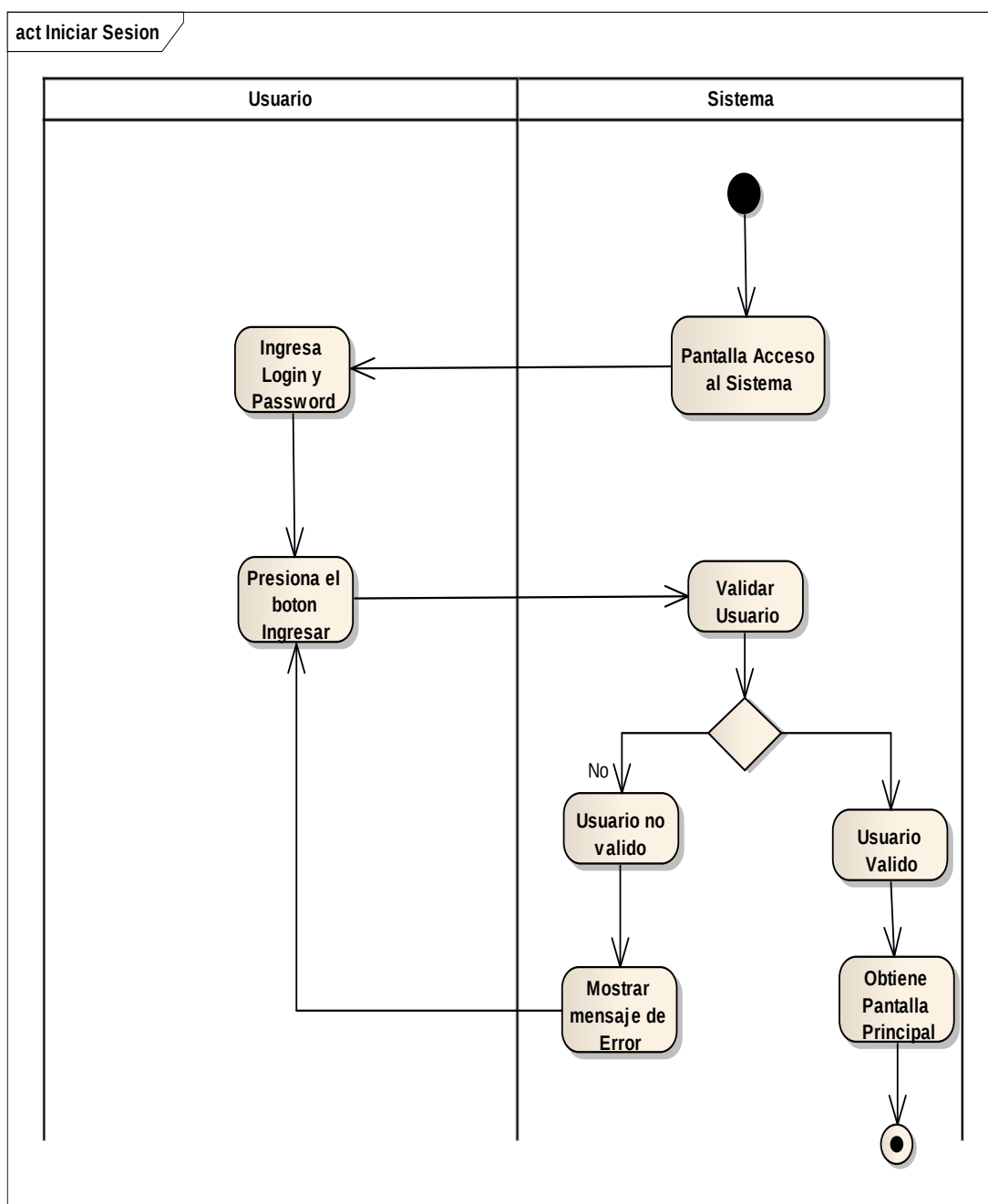


Figura 61. Diagrama de Actividad Iniciar Sesión

2.1.7.5.13.2 Diagrama de Actividad: Caso de Uso Gestionar Usuario

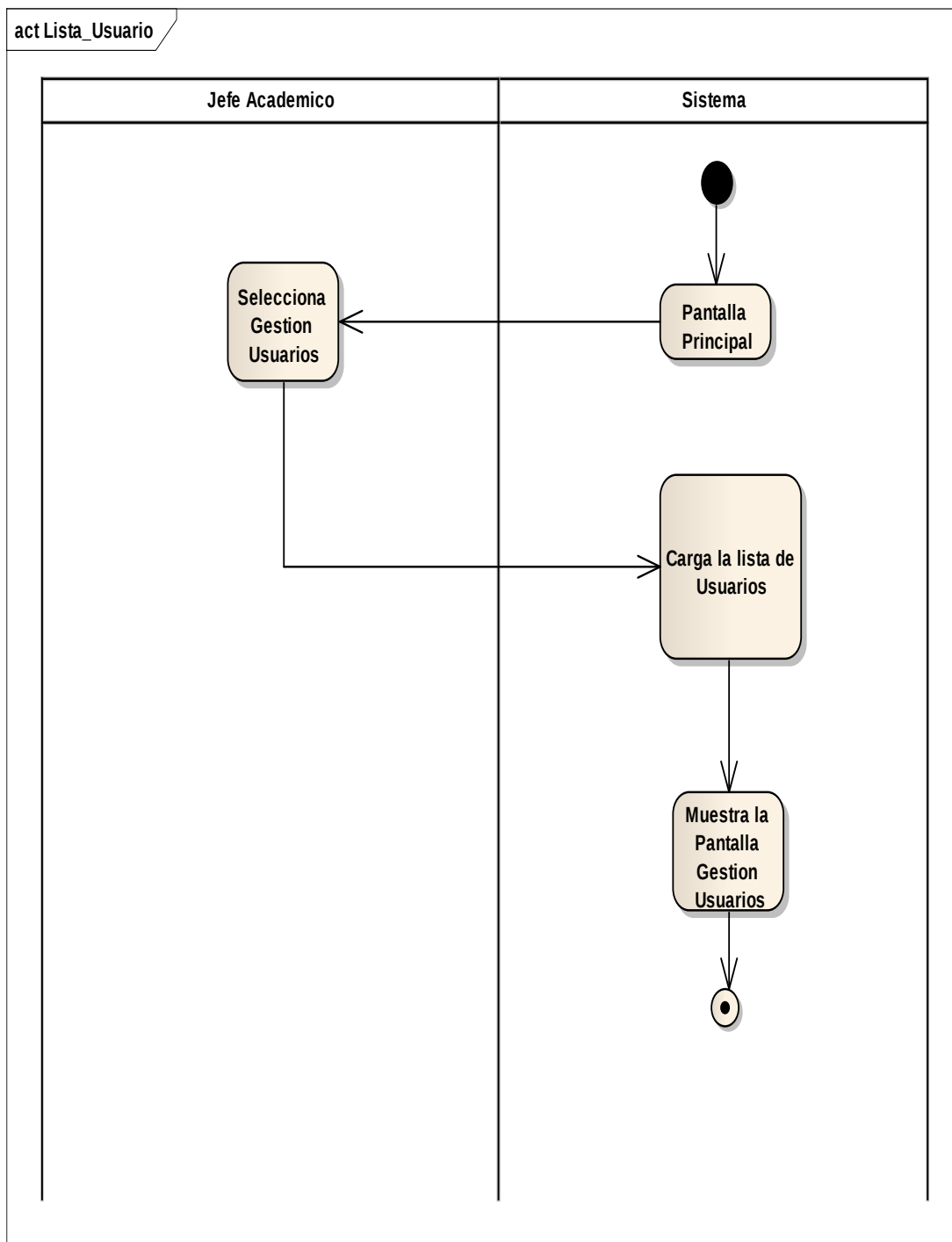


Figura 62. Diagrama de Actividad Gestionar Usuario

2.1.7.5.13.3 Diagrama de Actividad: Caso de Uso Adicionar Usuario

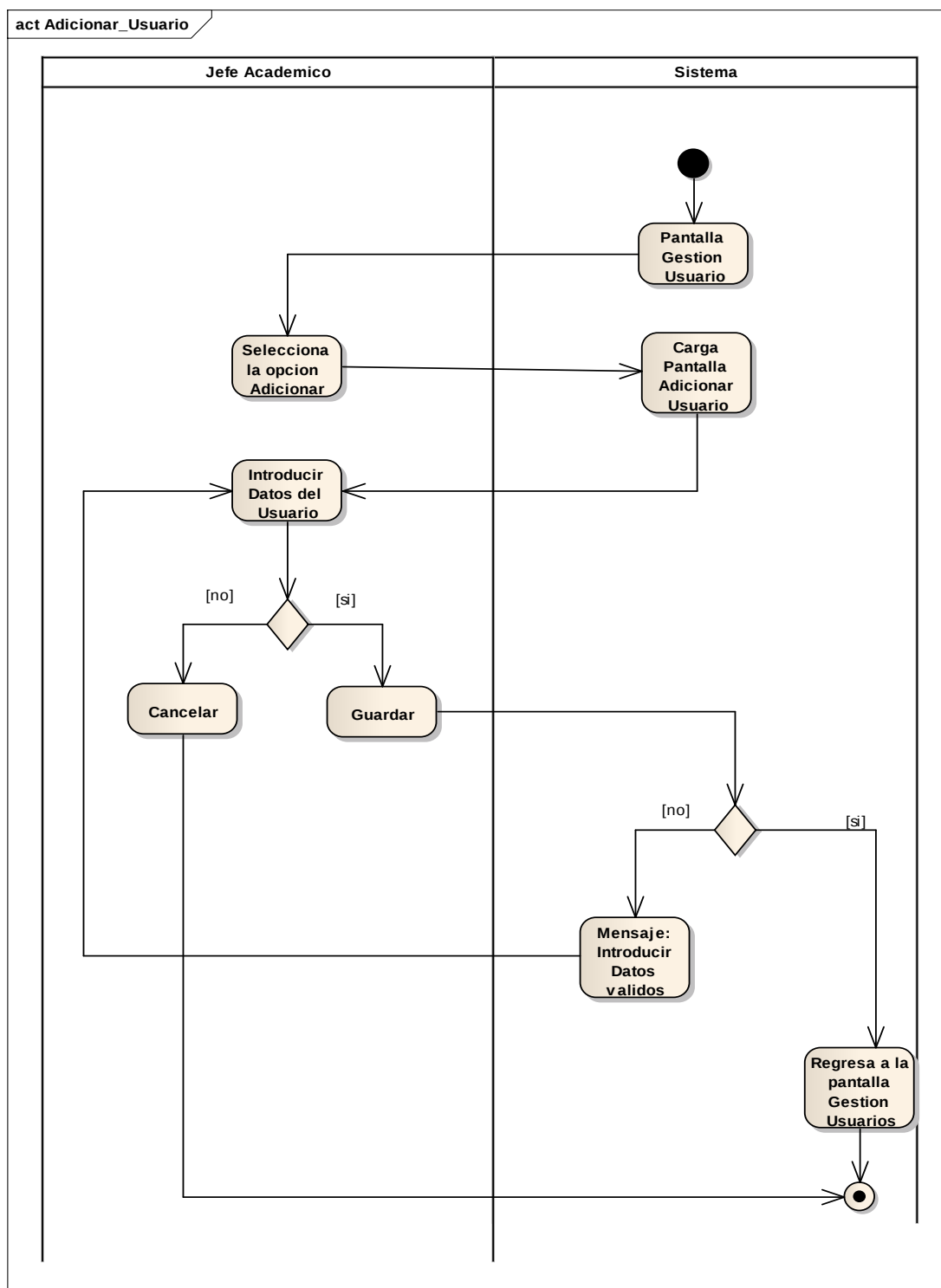


Figura 63. Diagrama de Actividad Adicionar Usuario

2.1.7.5.13.4 Diagrama de Actividad: Caso de Uso Modificar Usuario

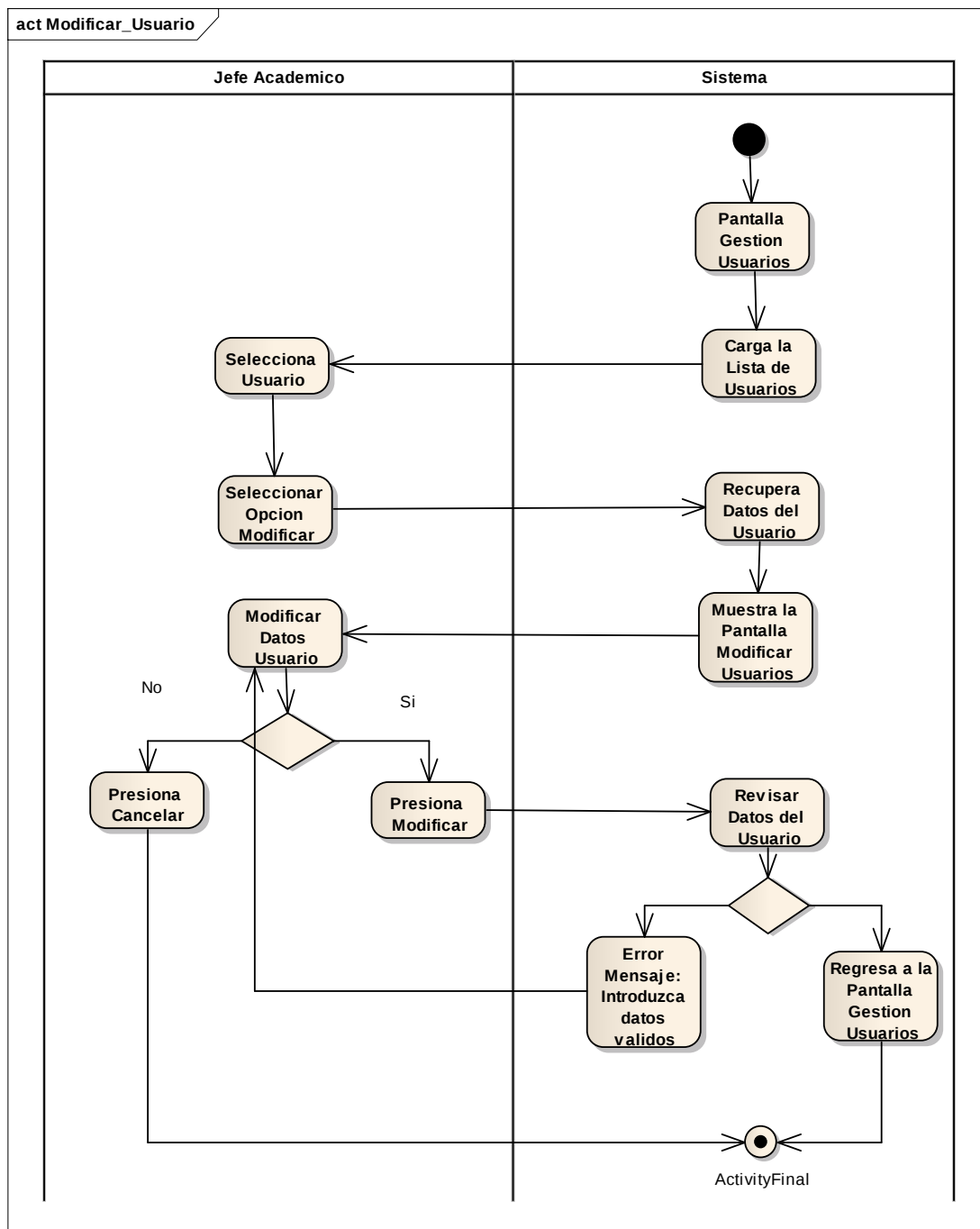


Figura 64. Diagrama de Actividad Modificar Usuario

2.1.7.5.13.5 Diagrama de Actividad: Caso de Uso Eliminar Usuario

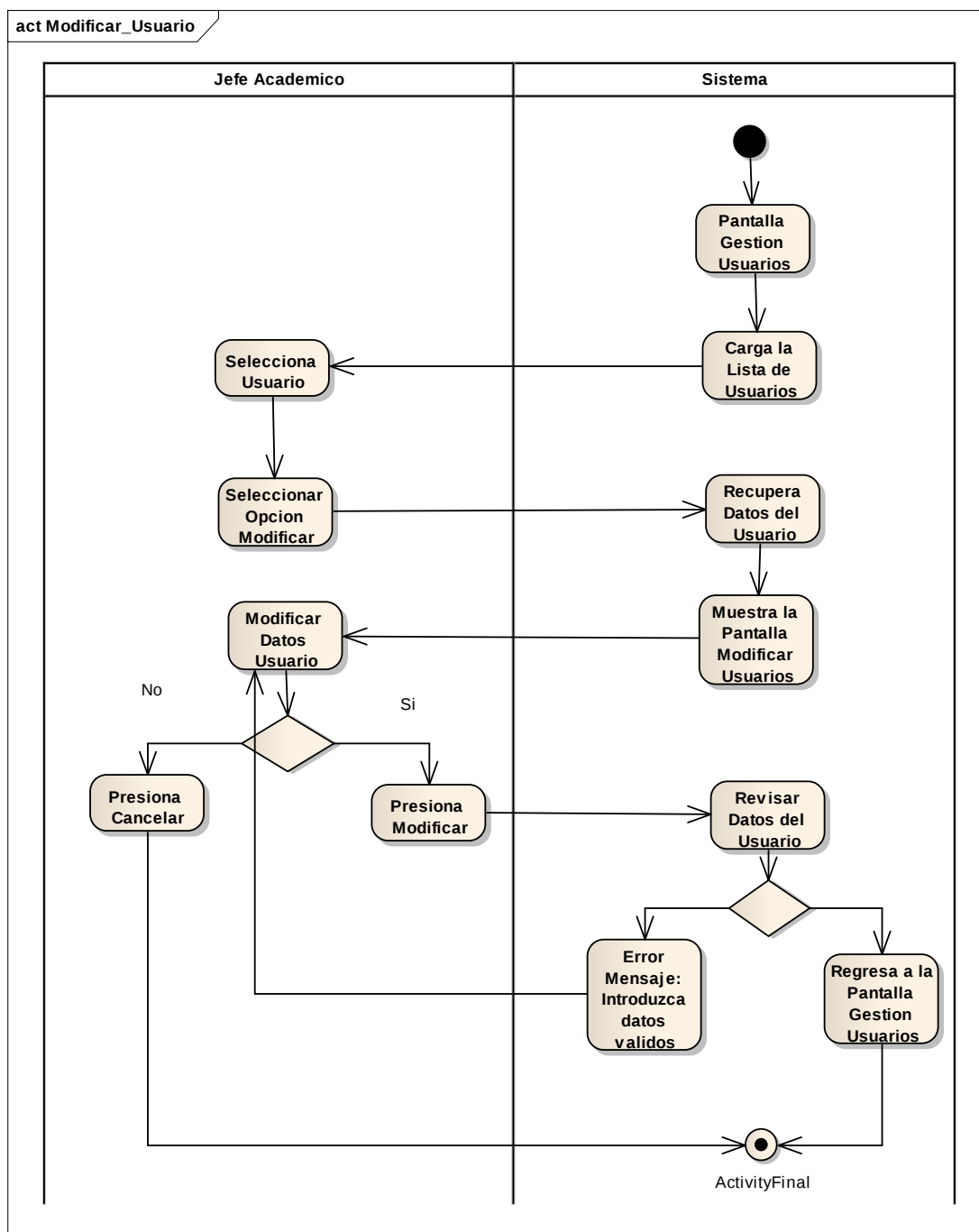


Figura 65. Diagrama de Actividad Eliminar Usuario

2.1.7.5.13.6 Diagrama de Actividad: Caso de Uso Gestionar Matricula

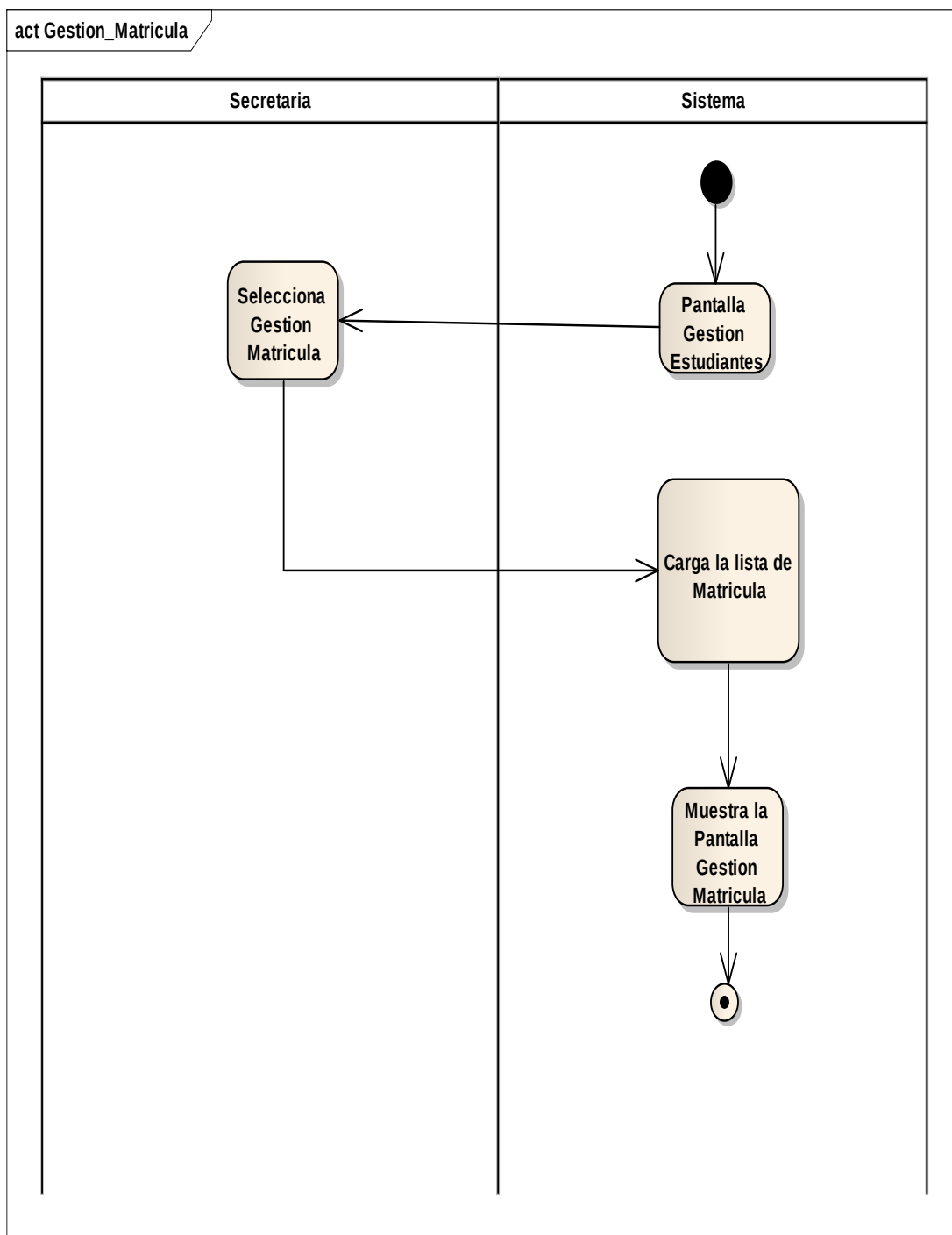


Figura 66. Diagrama de Actividad Gestionar Matricula

2.1.7.5.13.7 . Diagrama de Actividad Adicionar Matricula

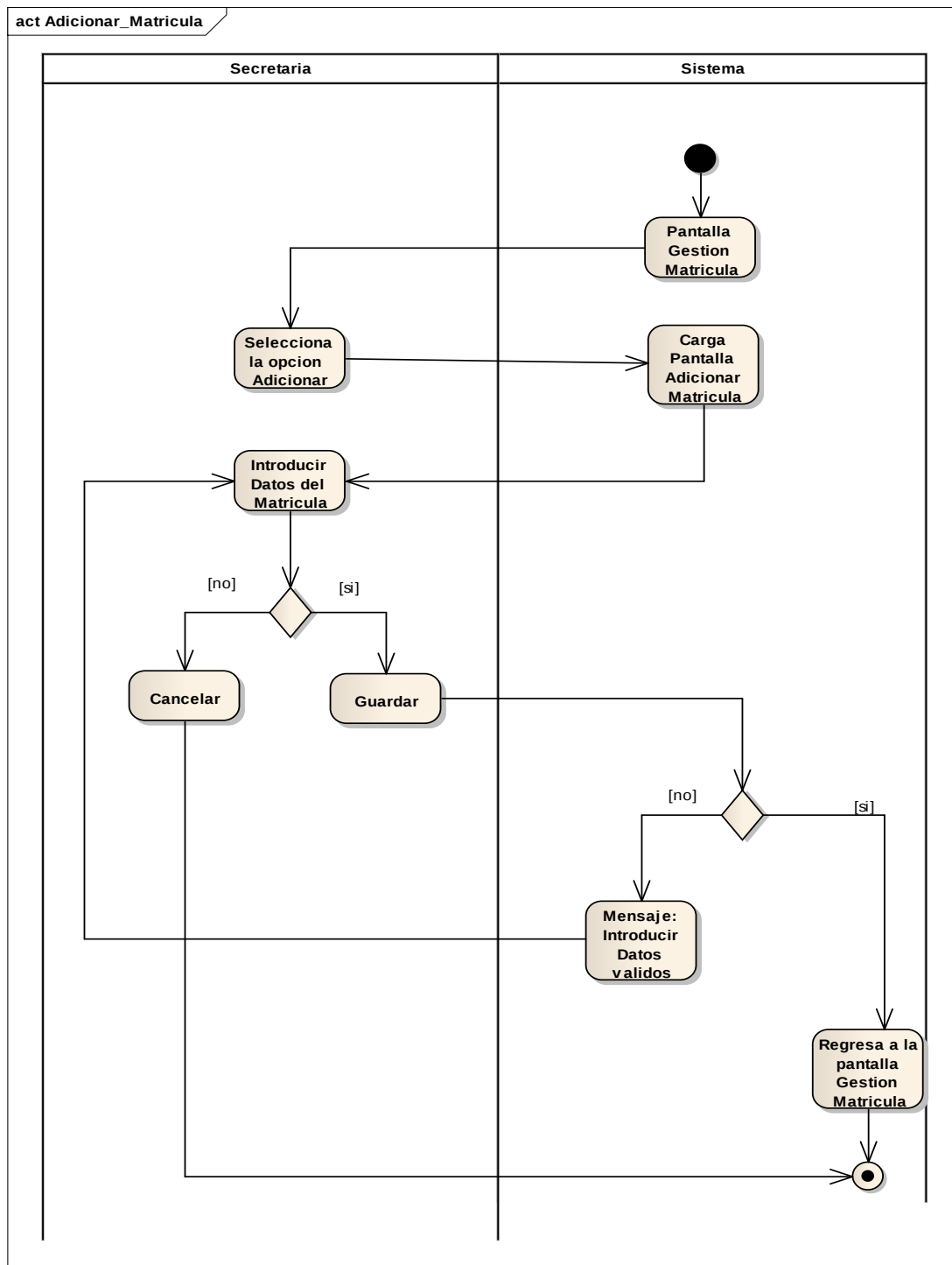


Figura 67. Diagrama De Actividad Adicionar Matricula

2.1.7.5.13.8 Diagrama de Actividad Modificar Matricula

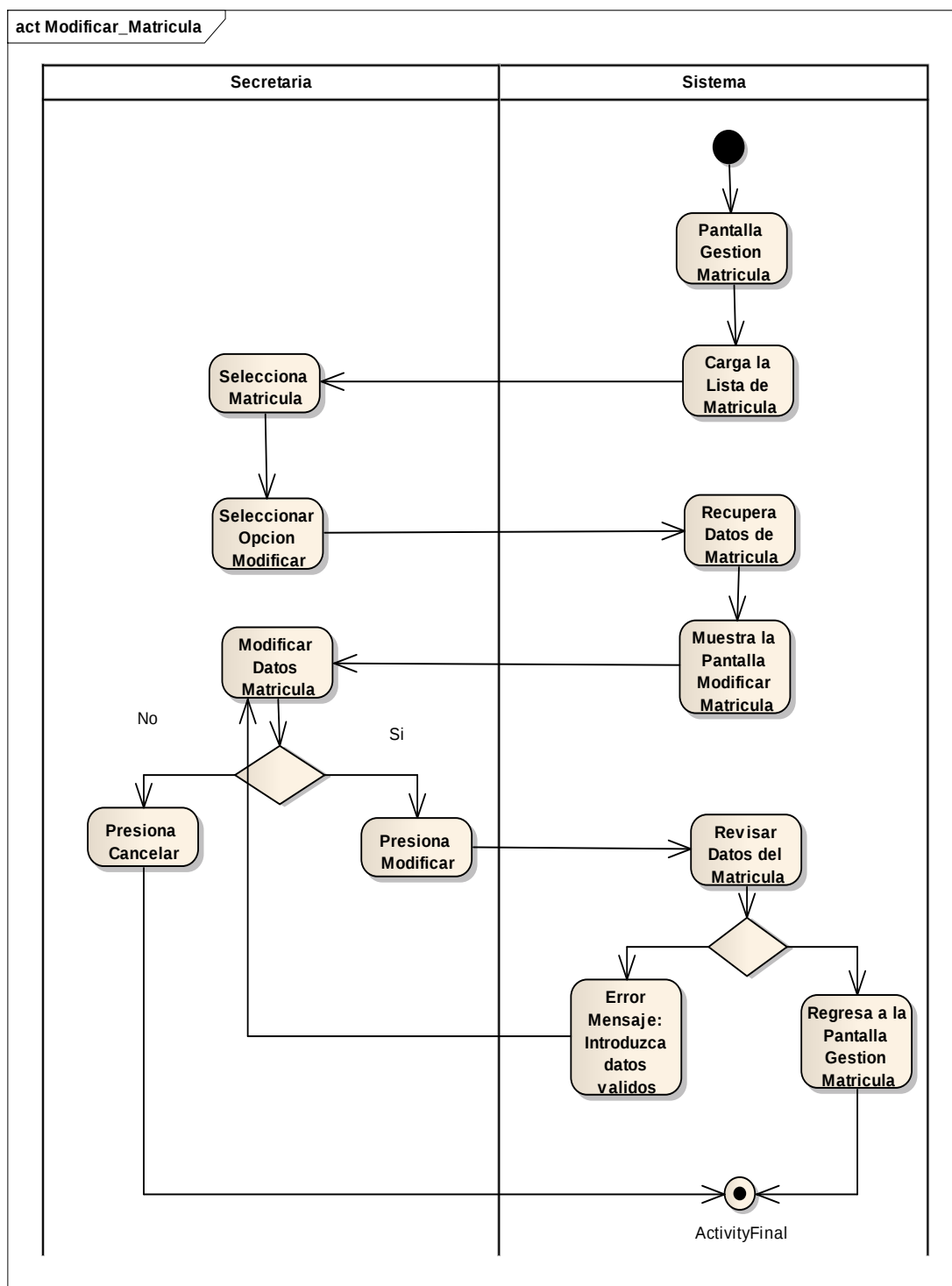


Figura 68. Diagrama De Actividad Modificar Matricula

2.1.7.5.13.9 Diagrama de Actividad Eliminar Matricula

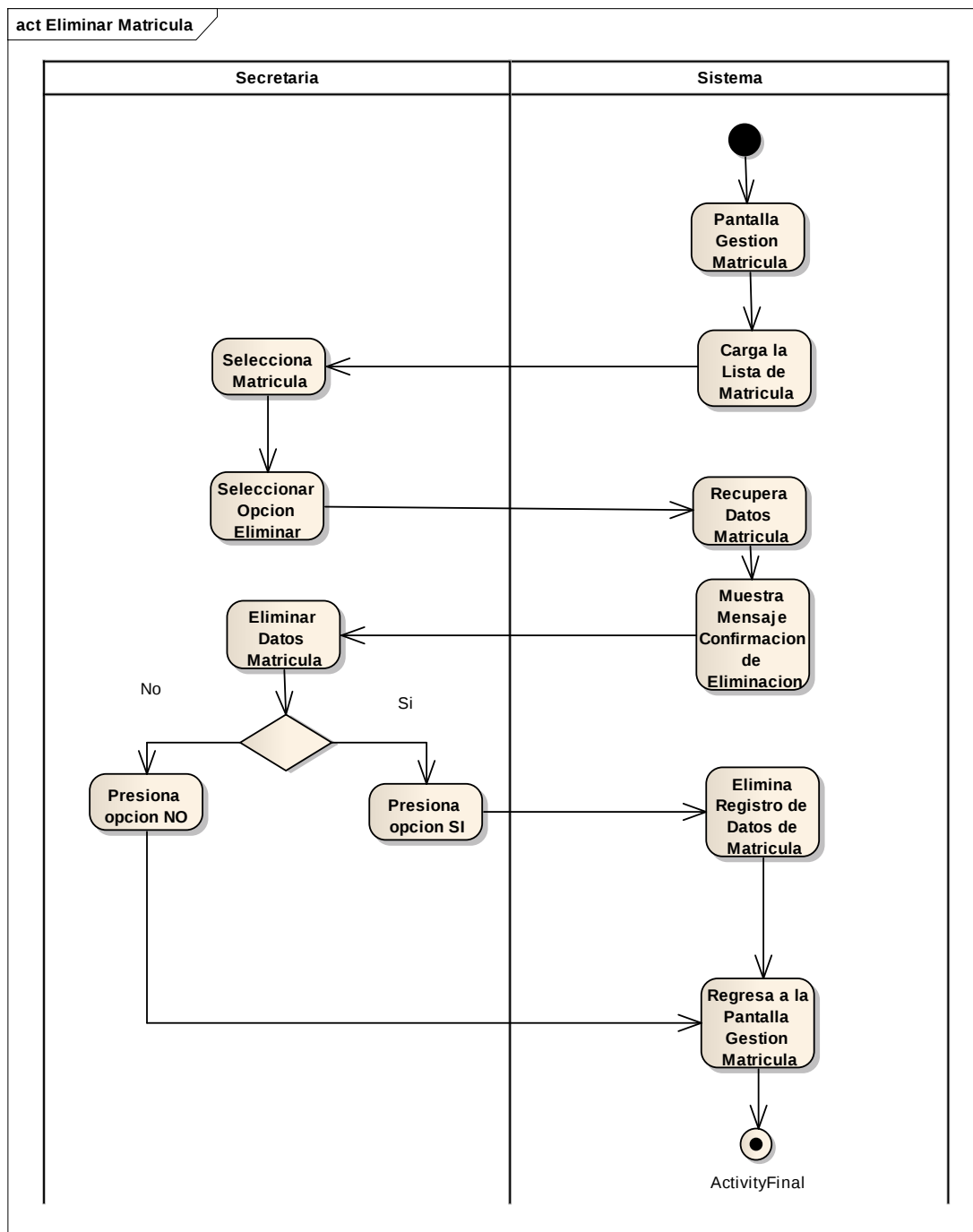


Figura 69. Diagrama De Actividad Eliminar Matricula

2.1.7.5.13.10 Diagrama de Actividad Gestión Carrera

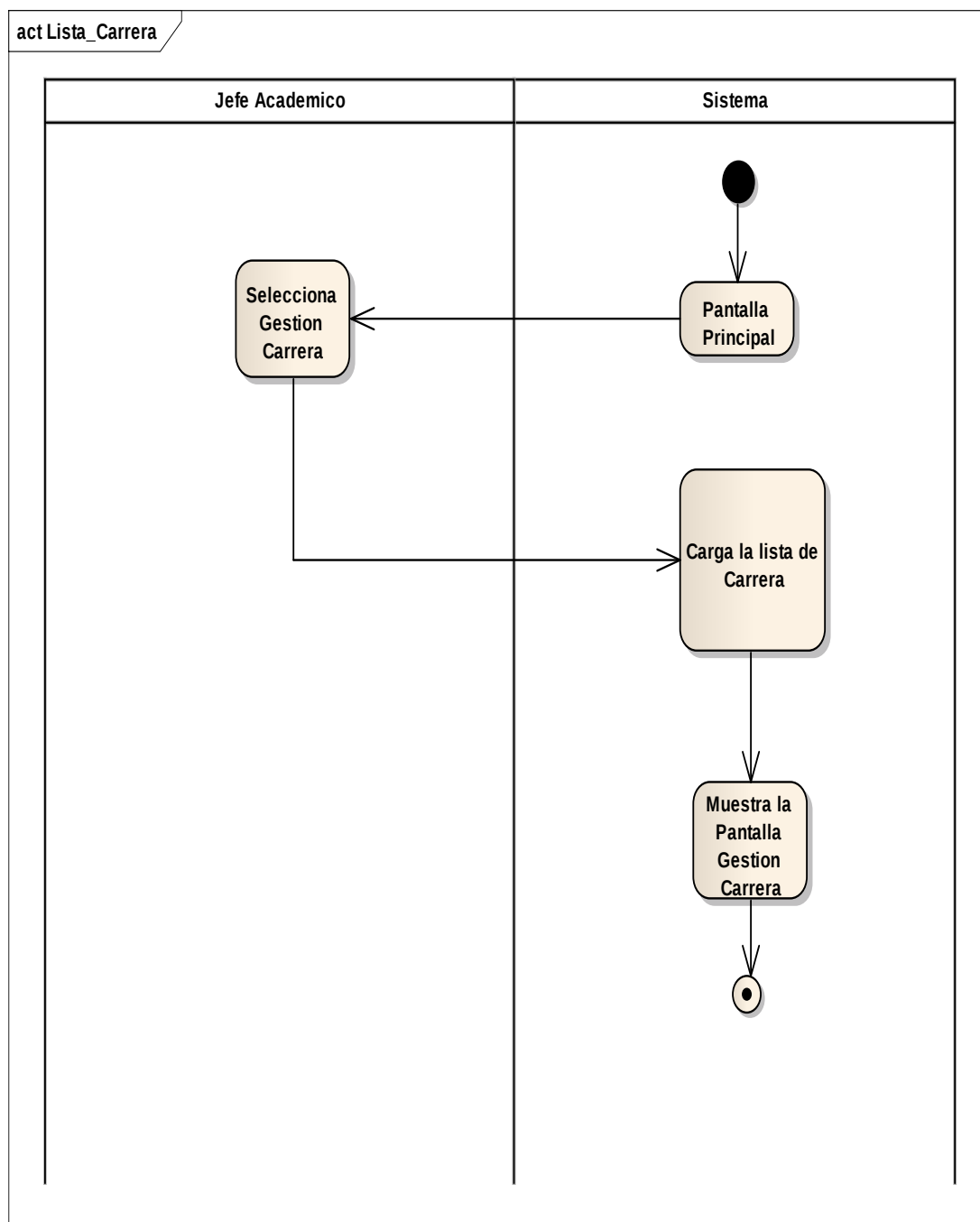


Figura 70. Diagrama De Actividad Gestión Carrera

2.1.7.5.13.11 Diagrama de Actividad Adicionar Carrera

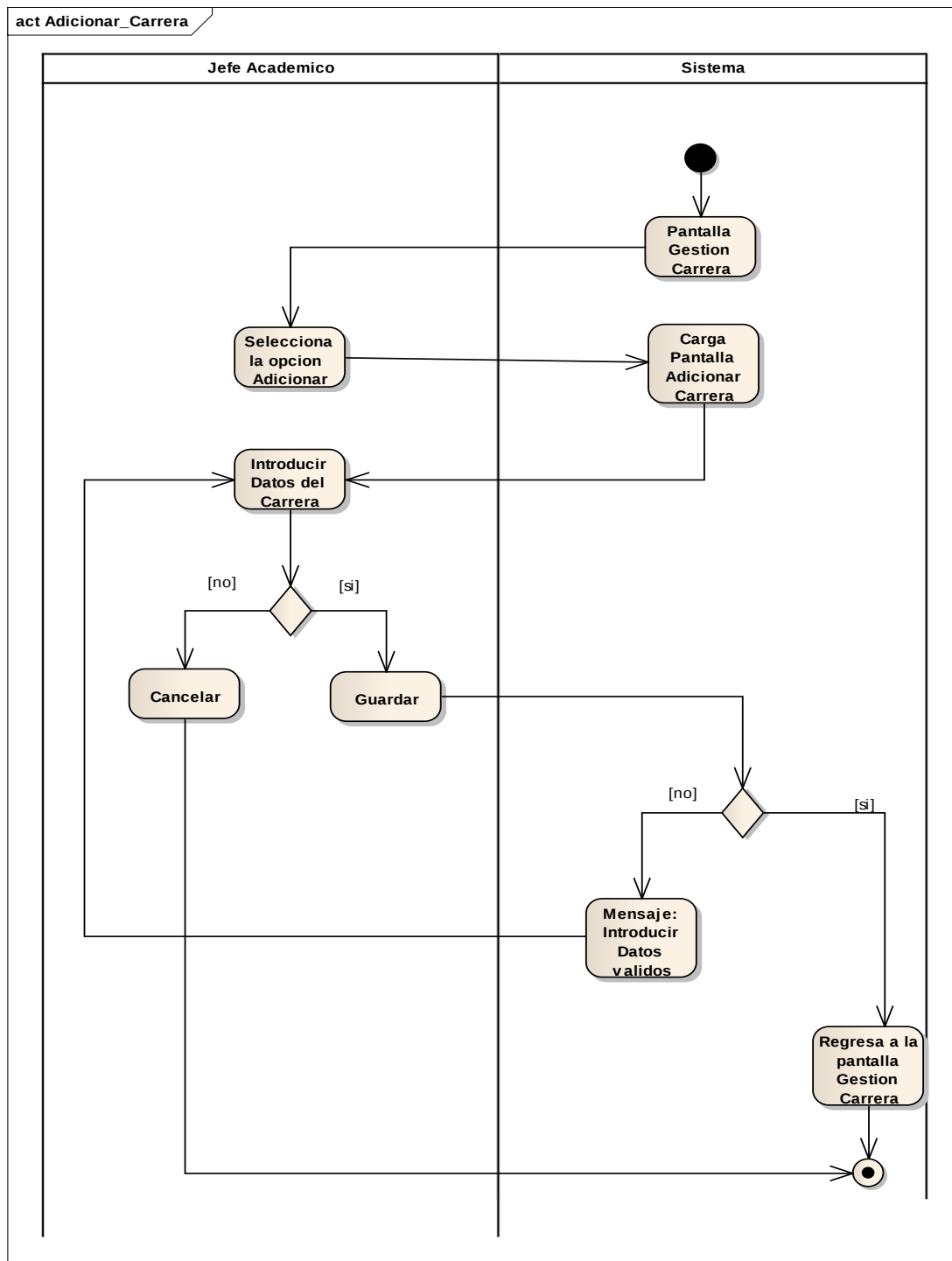


Figura 71. Diagrama De Actividad Adicionar Carrera

2.1.7.5.13.12 Diagrama de Actividad Modificar Carrera

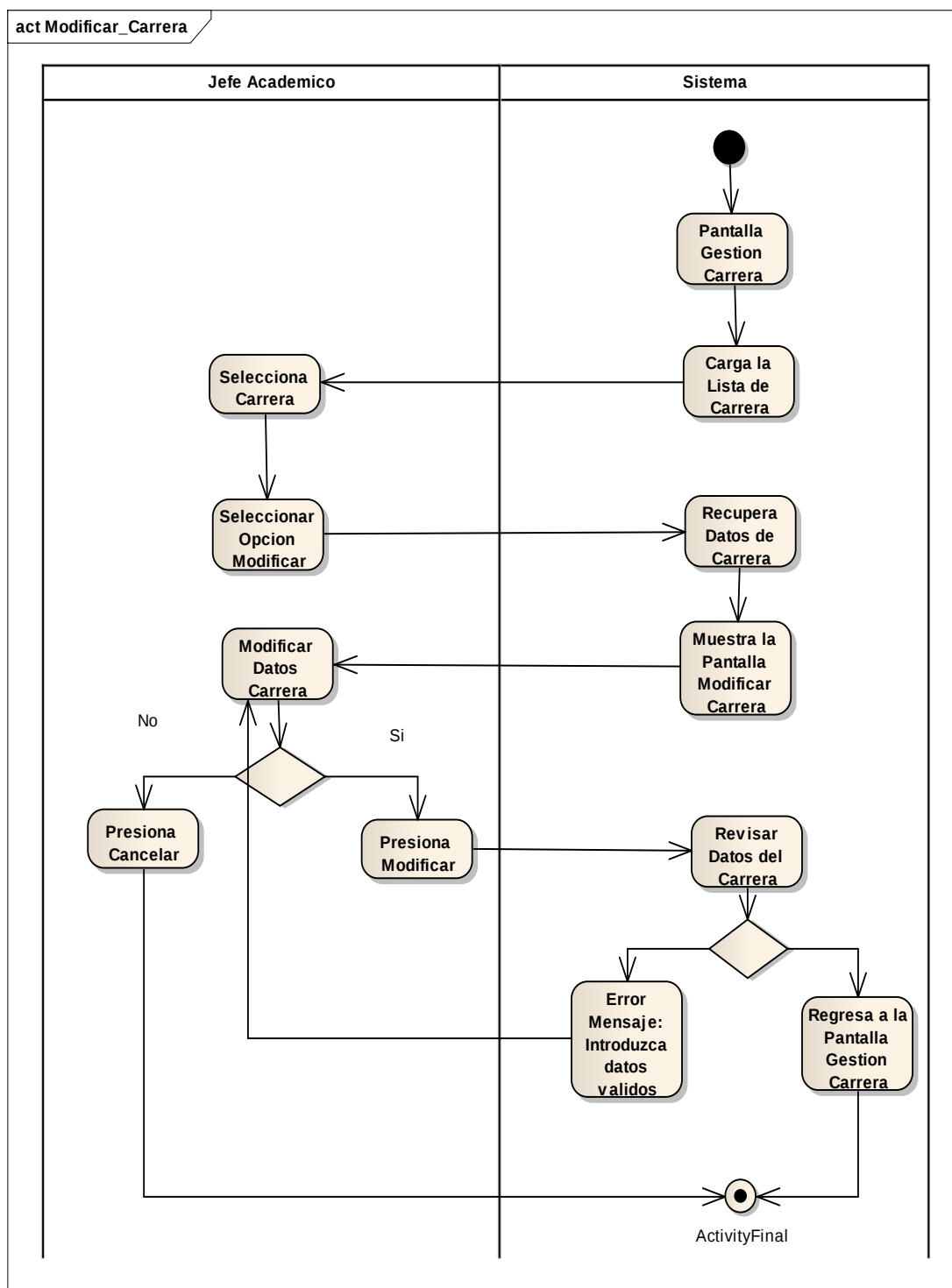


Figura 72. Diagrama De Actividad Modificar Carrera

2.1.7.5.13.13 Diagrama de Actividad Eliminar Carrera

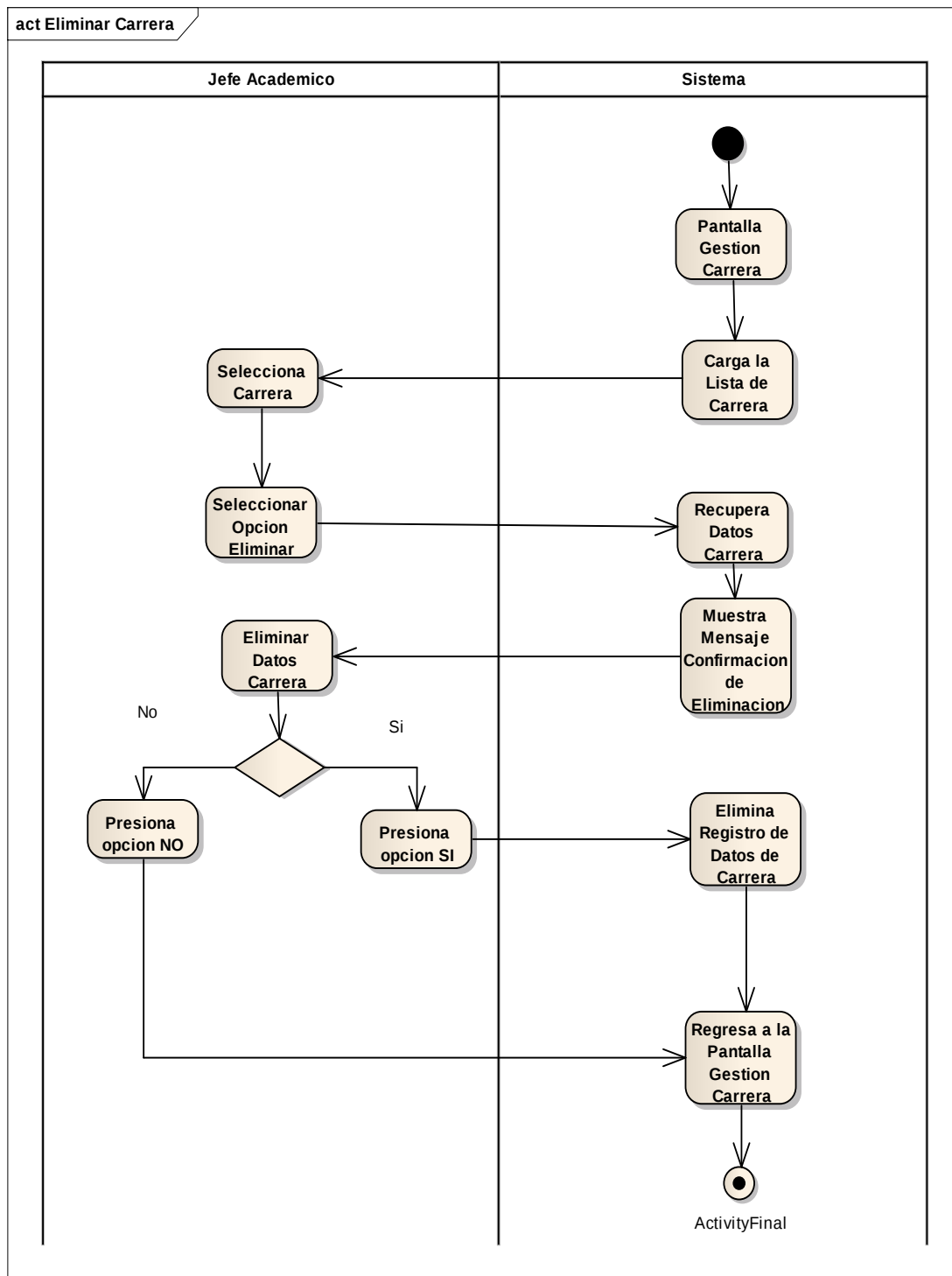


Figura 73. Diagrama De Actividad Eliminar Carrera

2.1.7.5.13.14 Diagrama de Actividad Gestionar Materias

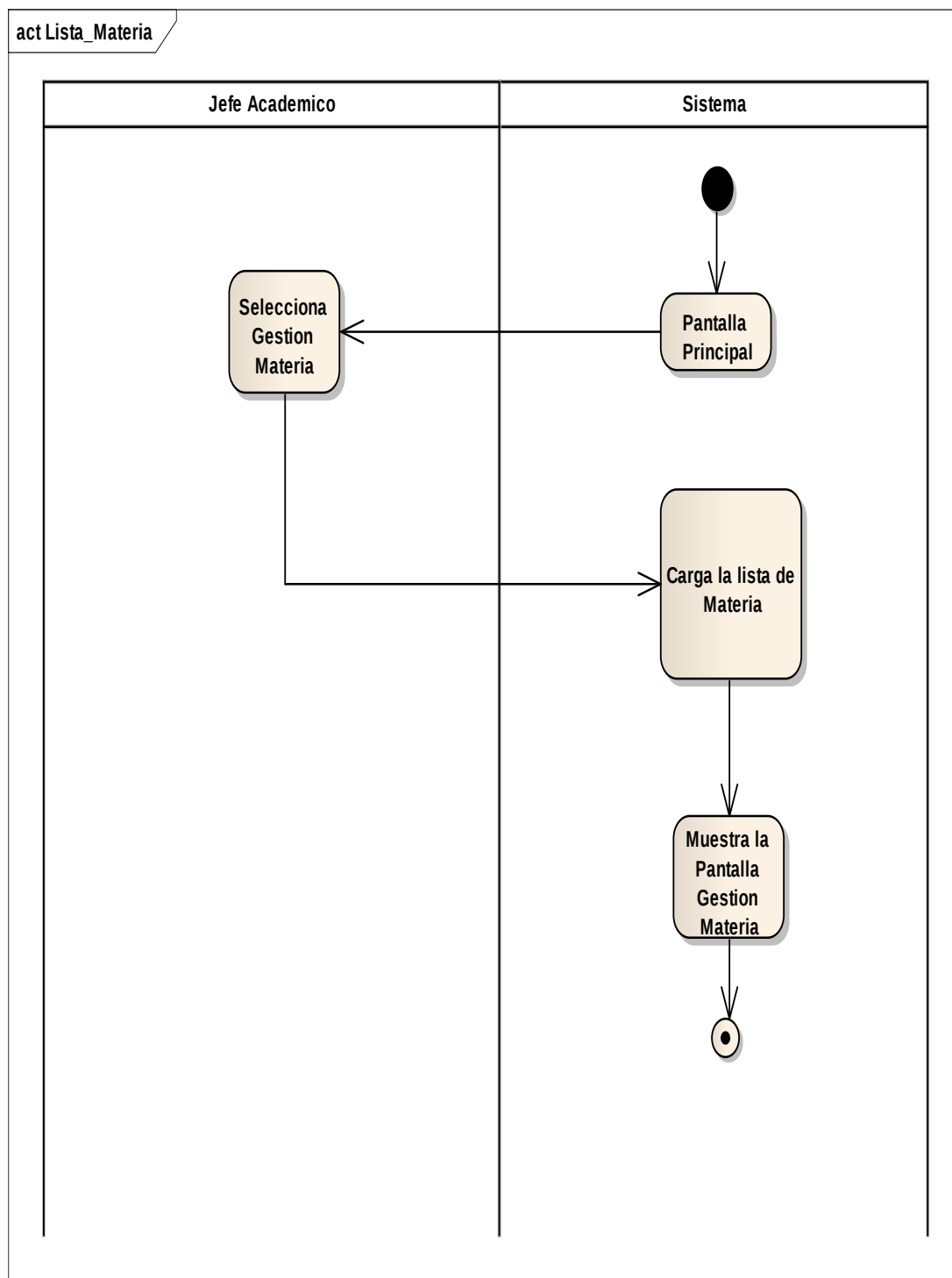


Figura 74. Diagrama De Actividad Gestión Materias

2.1.7.5.13.15 Diagrama de Actividad Adicionar Materia

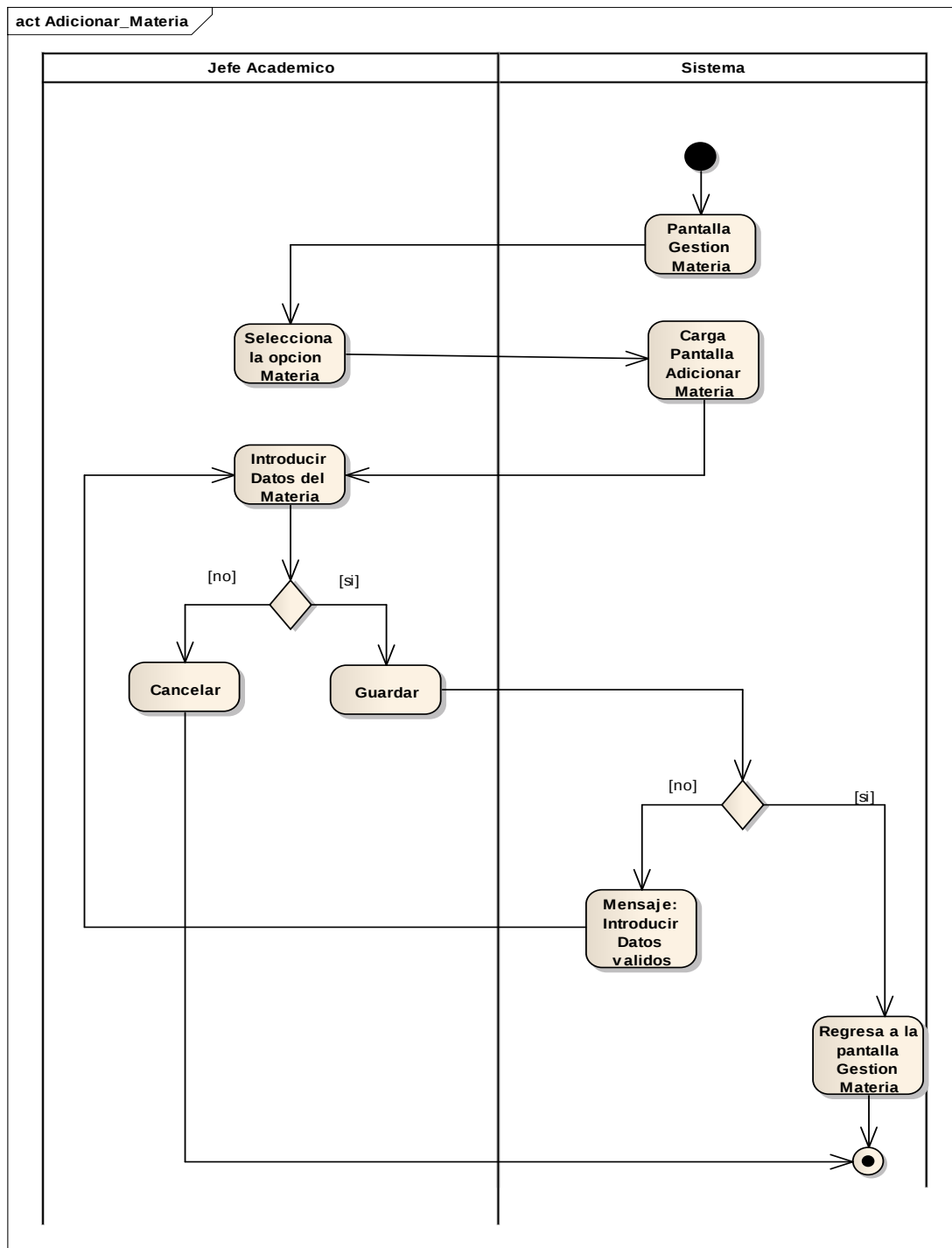


Figura 75. Diagrama De Actividad Adicionar Materia

2.1.7.5.13.16 Diagrama de Actividad Modificar Materia

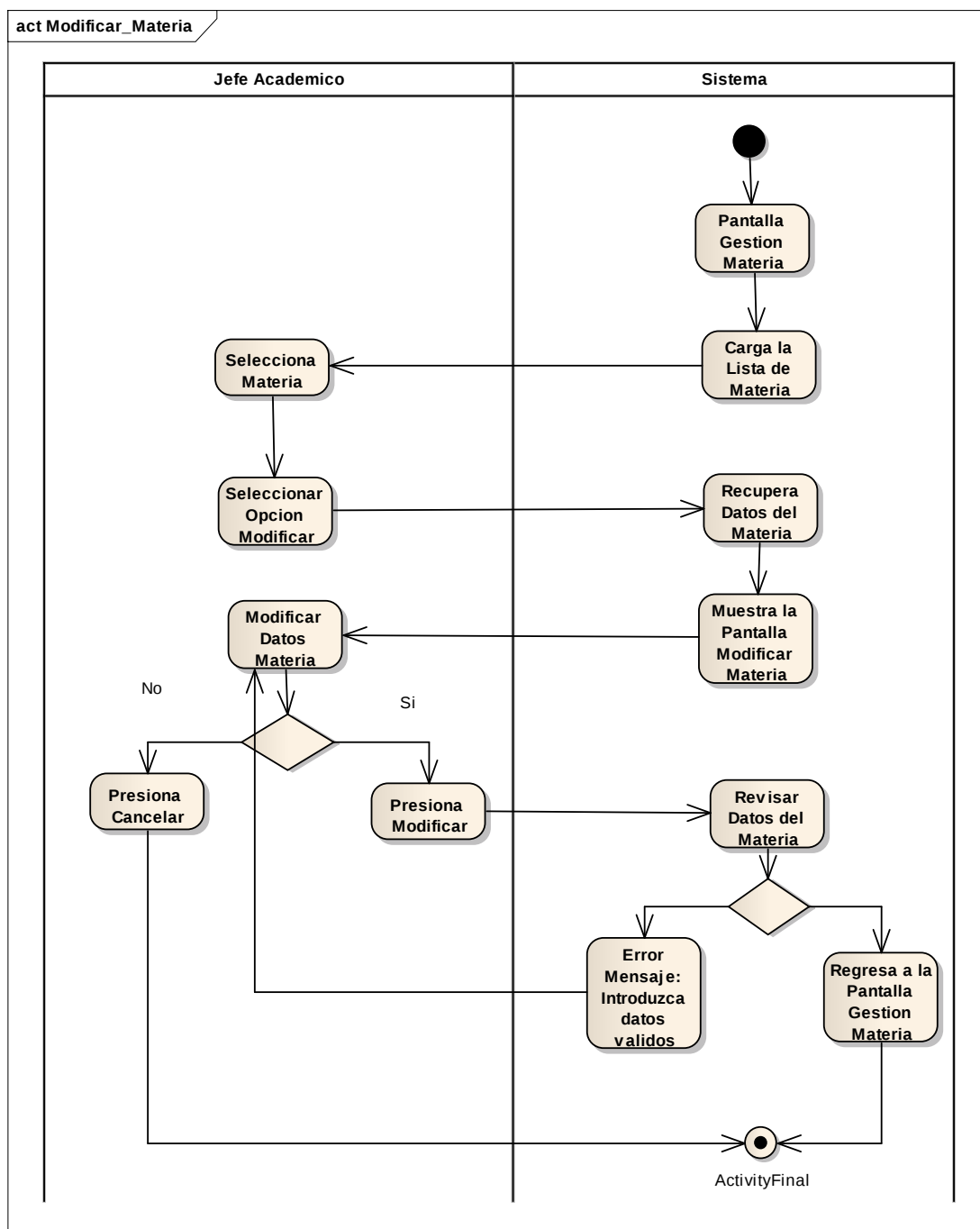


Figura 76. Diagrama De Actividad Modificar Materia

2.1.7.5.13.17 Diagrama de Actividad Eliminar Materia

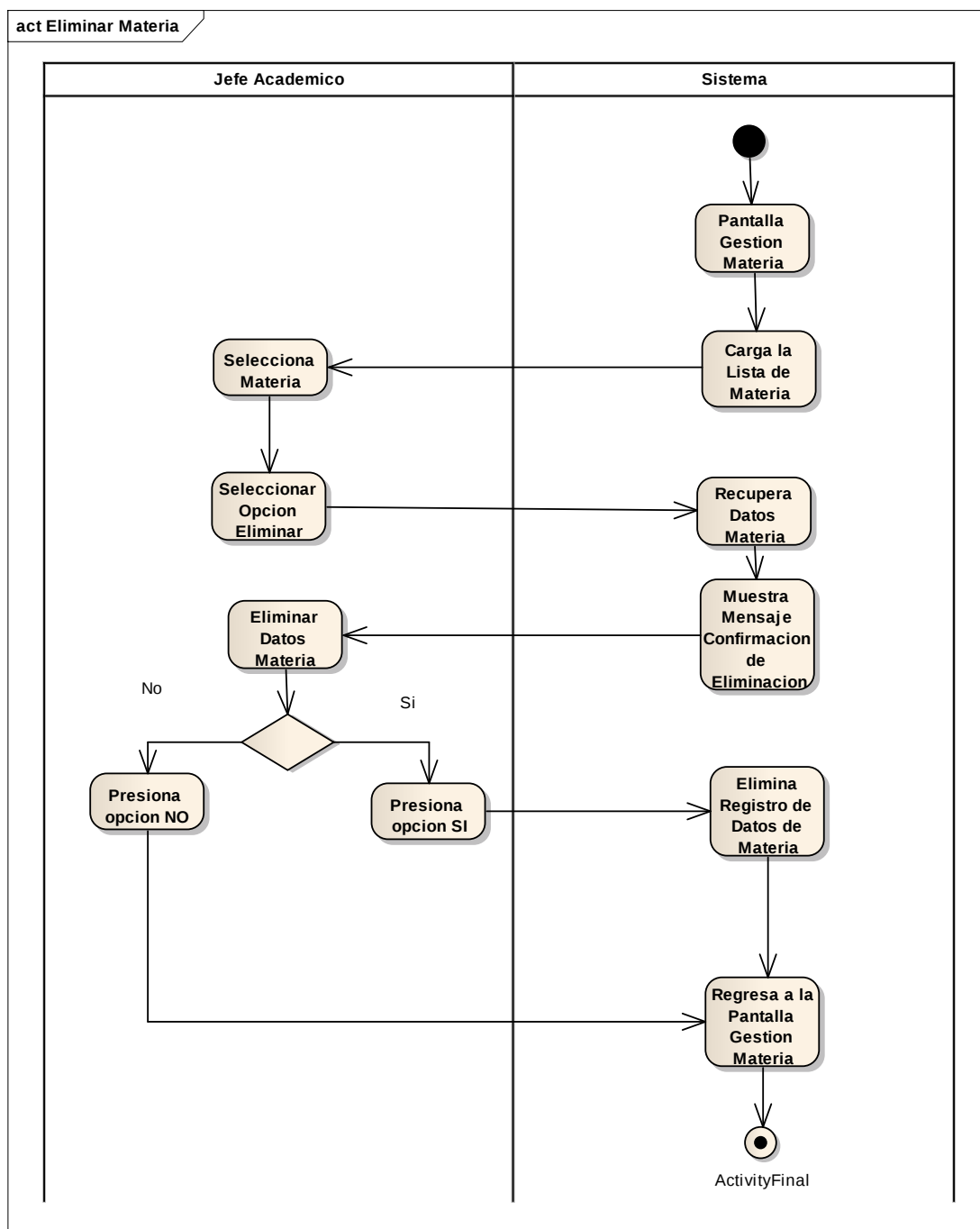


Figura 77. Diagrama de Actividad Eliminar Materia

2.1.7.5.13.18 Diagrama de Actividad Gestión Grupo

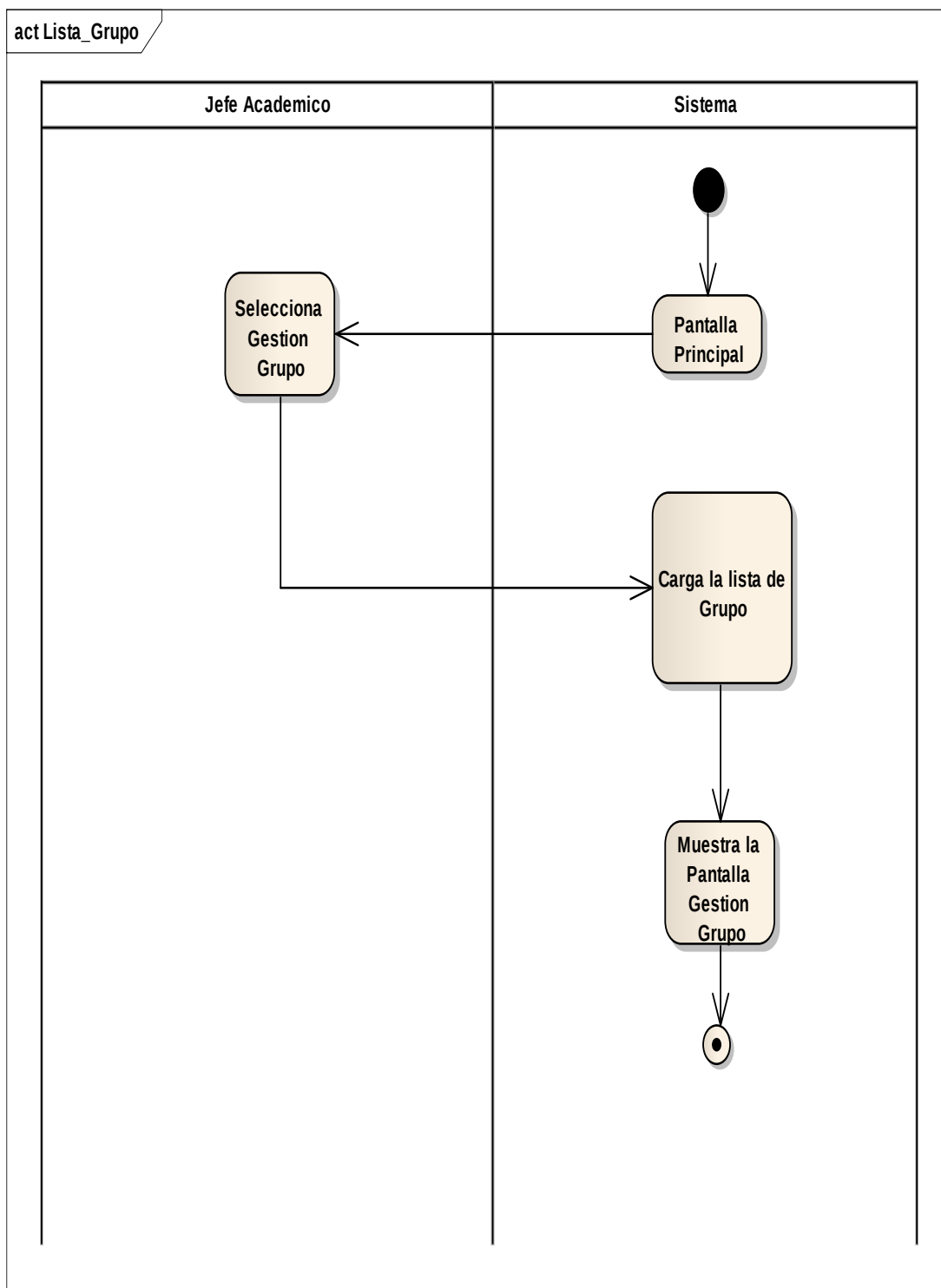


Figura 78. Diagrama De Actividad Gestión Grupo

2.1.7.5.13.19 Diagrama de Actividad Adicionar Grupo

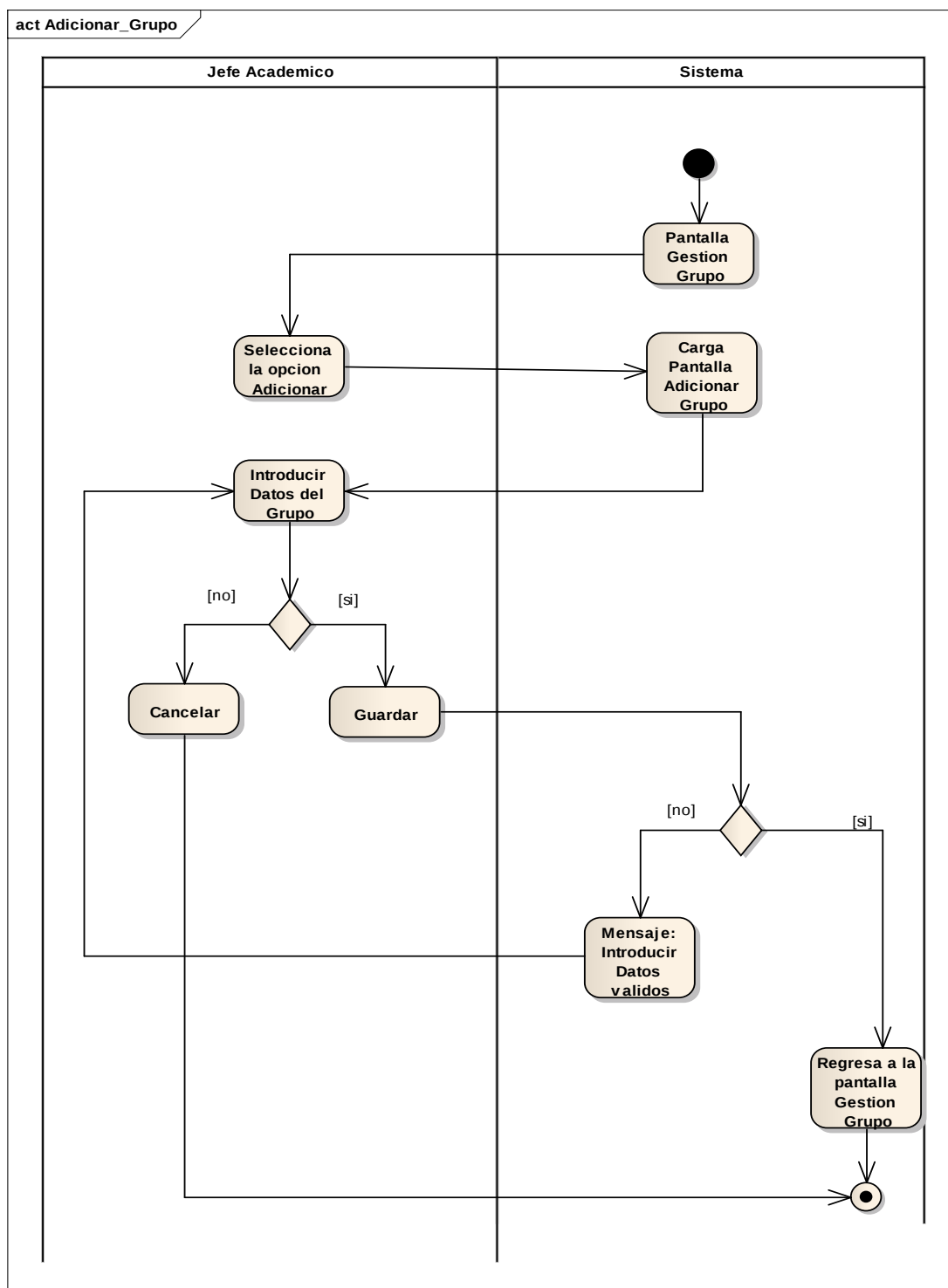


Figura 79. Diagrama De Actividad Adicionar Grupo

2.1.7.5.13.20 Diagrama de Actividad Modificar Grupo

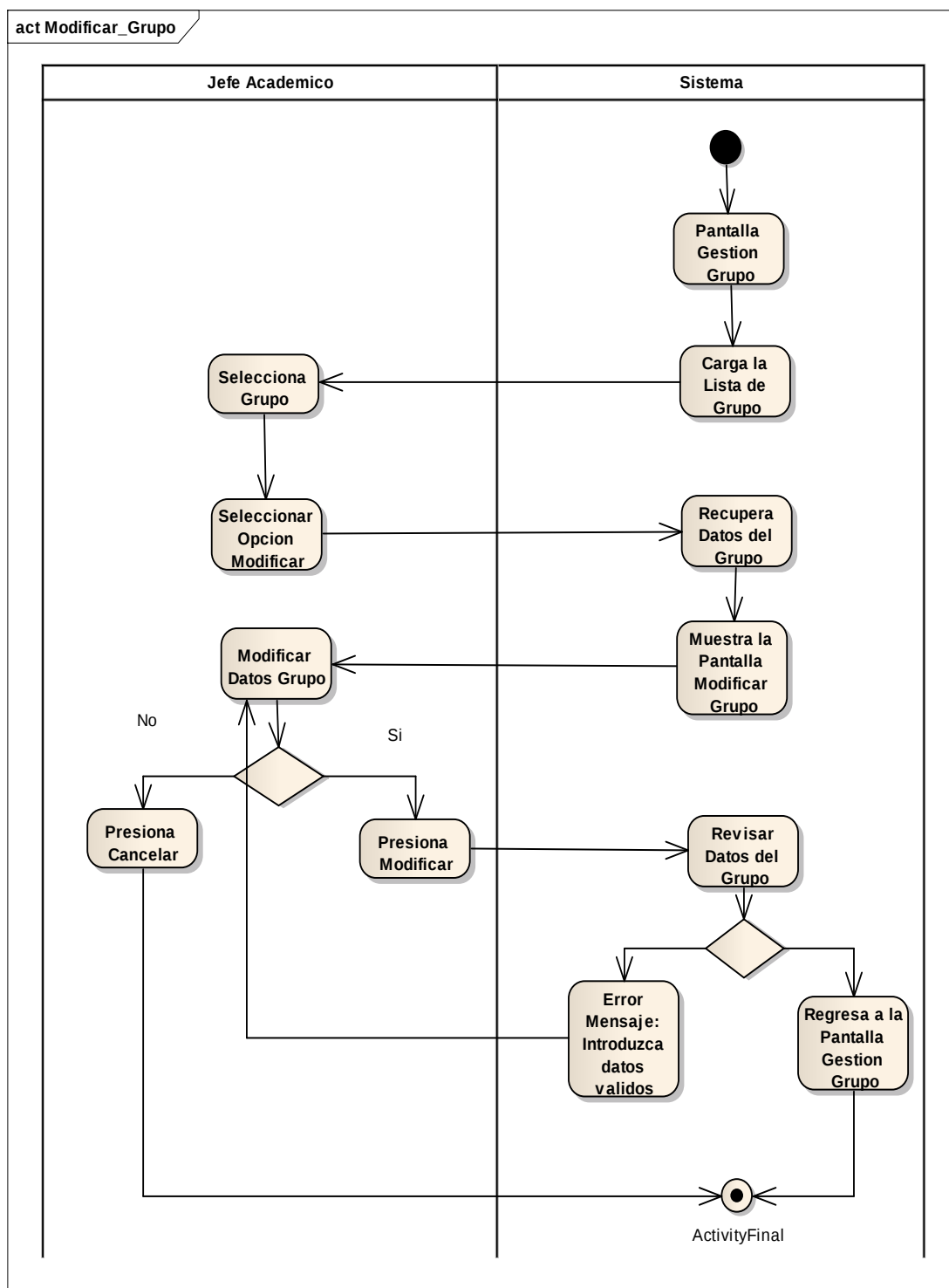


Figura 80. Diagrama De Actividad Modificar Grupo

2.1.7.5.13.21 Diagrama de Actividad Eliminar Grupo

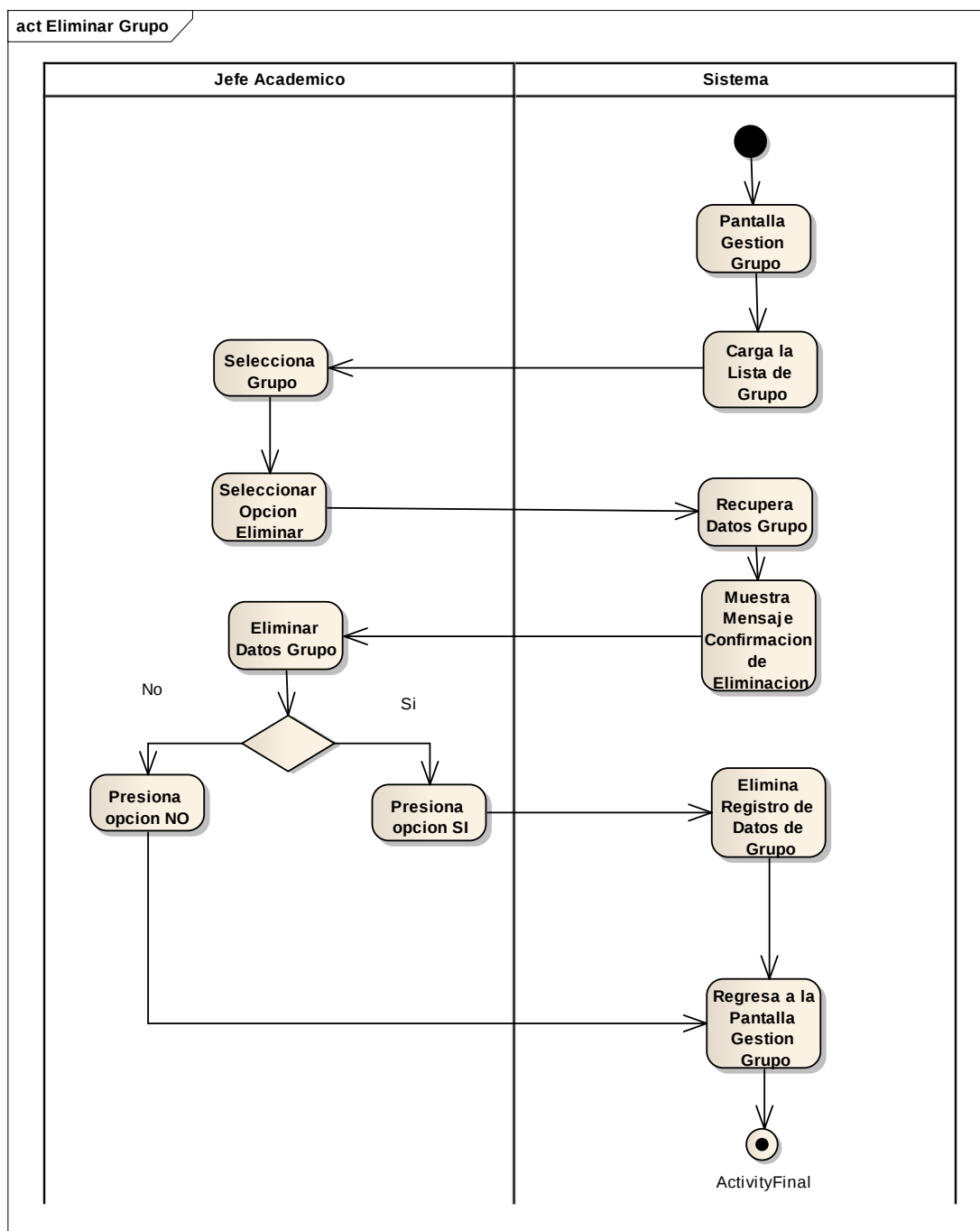


Figura 81. Diagrama De Actividad Eliminar Grupo

2.1.7.5.13.22 Diagrama de Actividad Gestión Curso

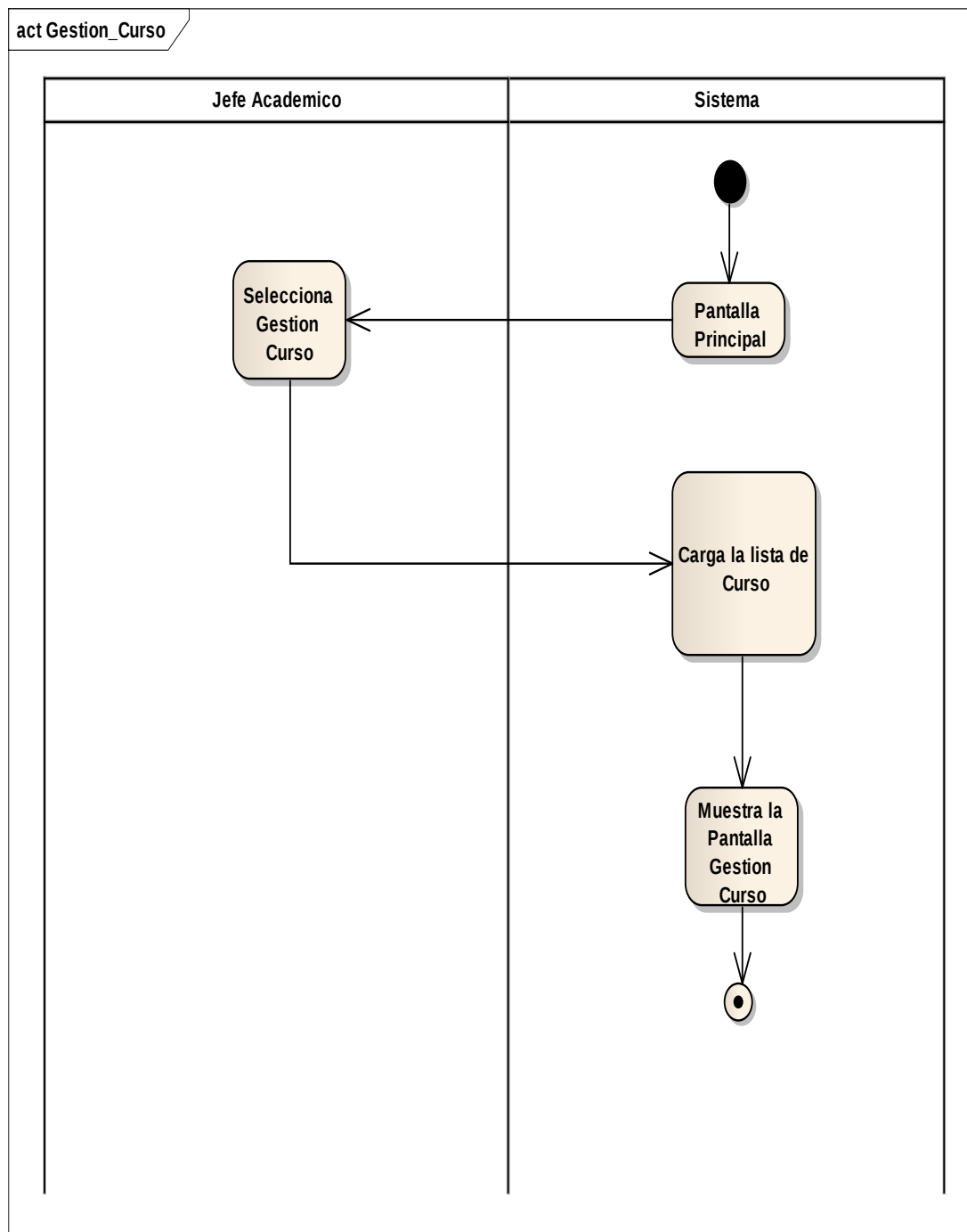


Figura 82. Diagrama De Actividad Gestión Curso

2.1.7.5.13.23 Diagrama de Actividad Adicionar Curso

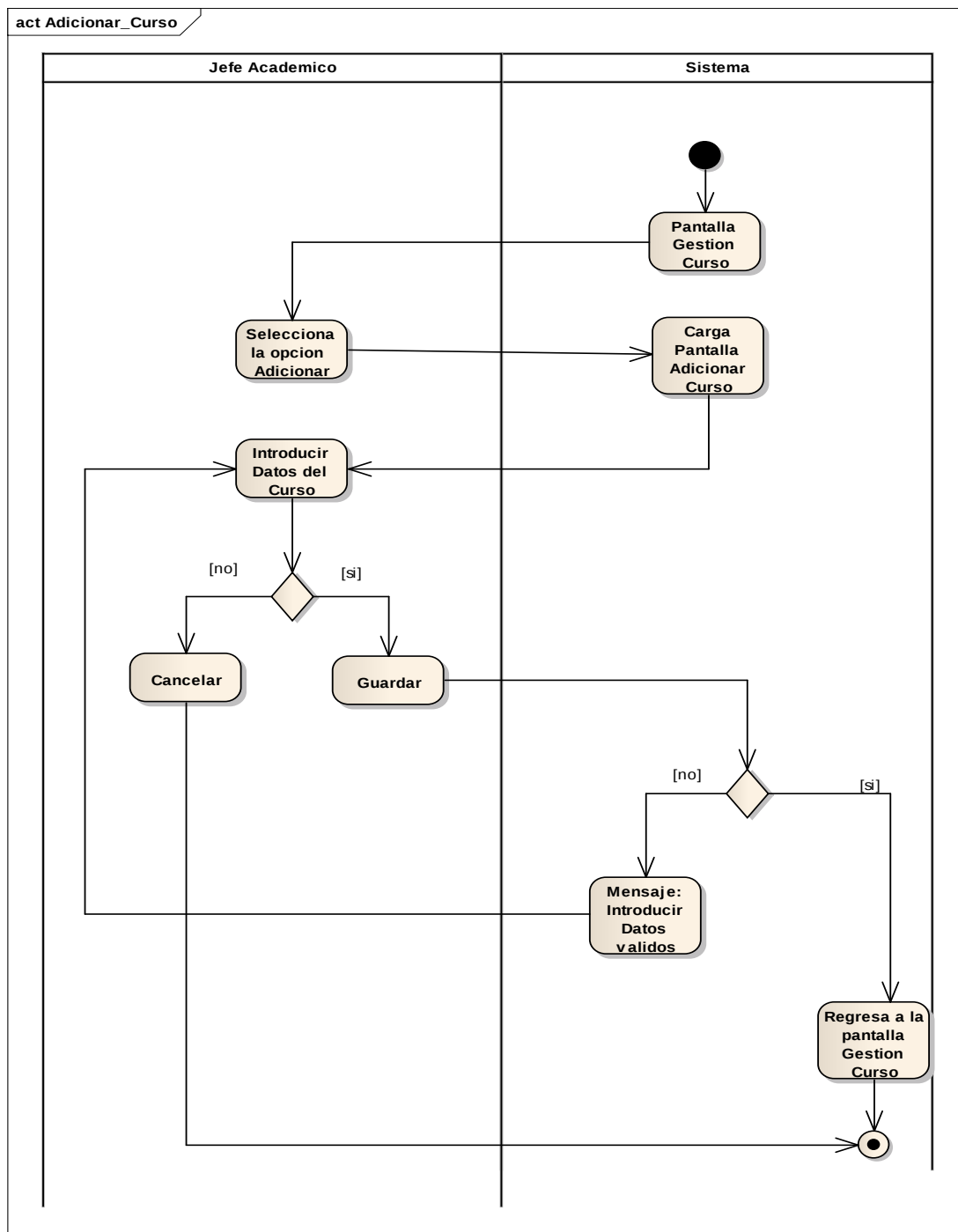


Figura 83. Diagrama De Actividad Adicionar Curso

2.1.7.5.13.24 Diagrama de Actividad Modificar Curso

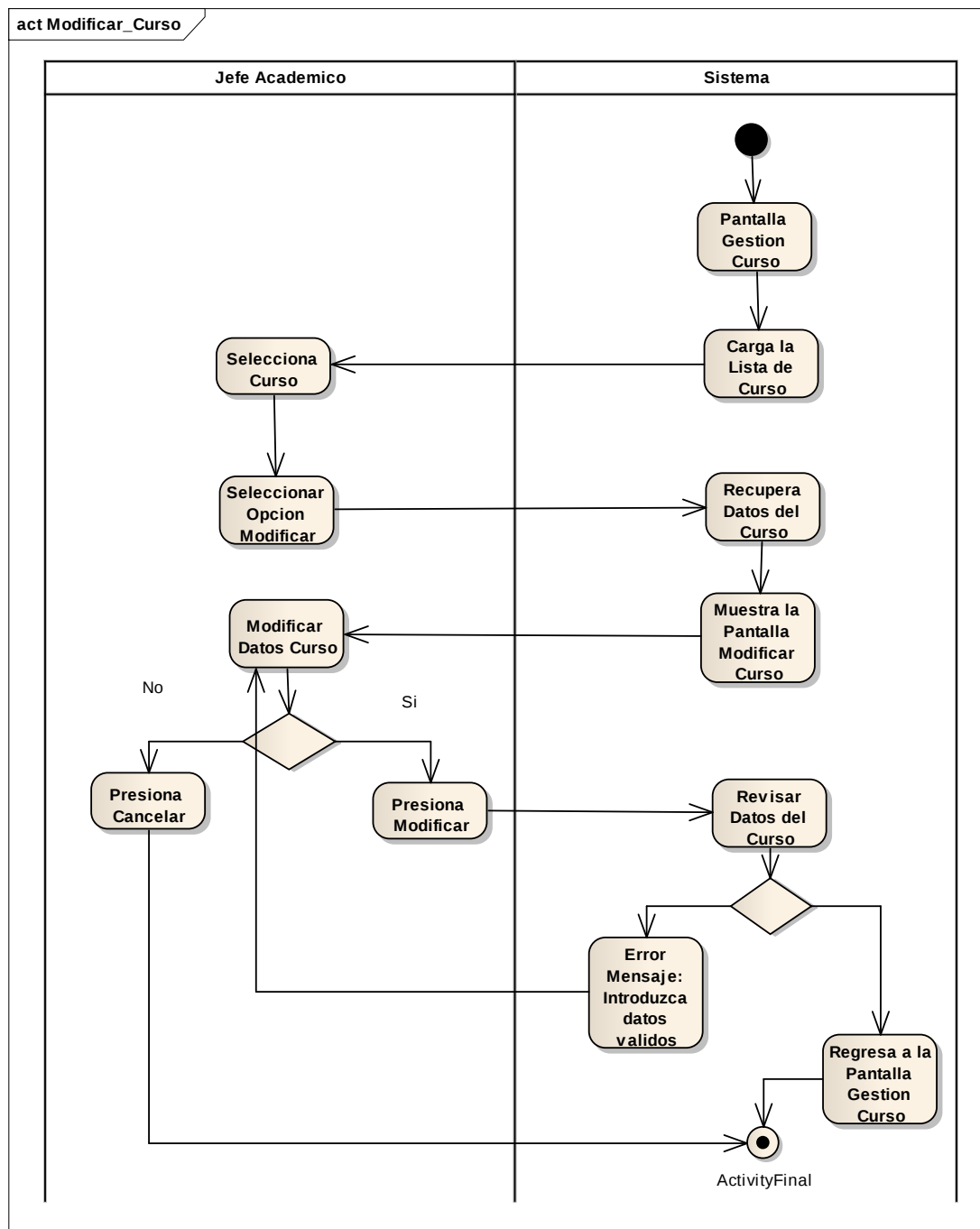


Figura 84. Diagrama De Actividad Modificar Curso

2.1.7.5.13.25 Diagrama de Actividad Eliminar Curso

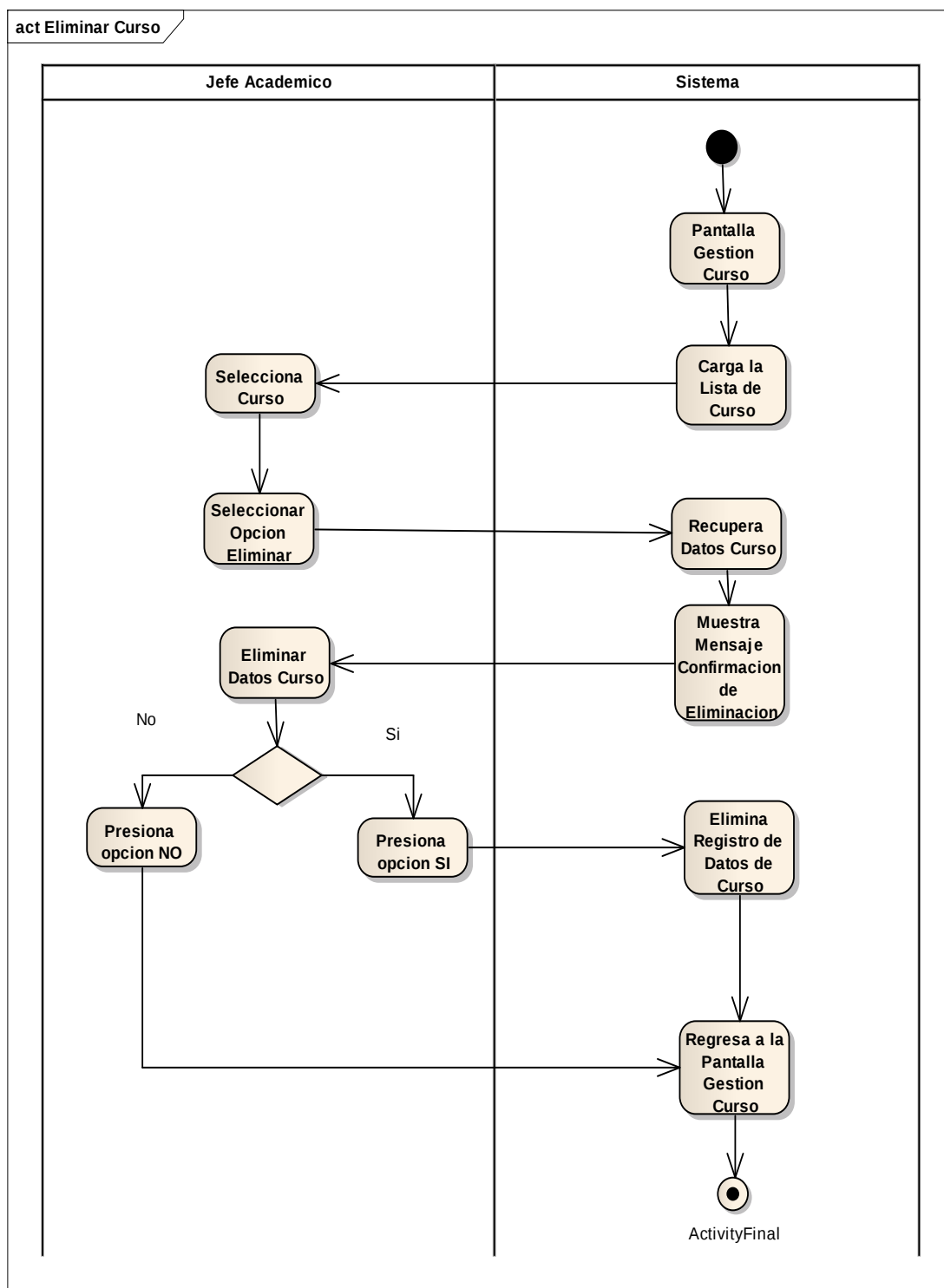


Figura 85. Diagrama De Actividad Eliminar Curso

2.1.7.5.13.26 Diagrama de Actividad Gestión Evaluación

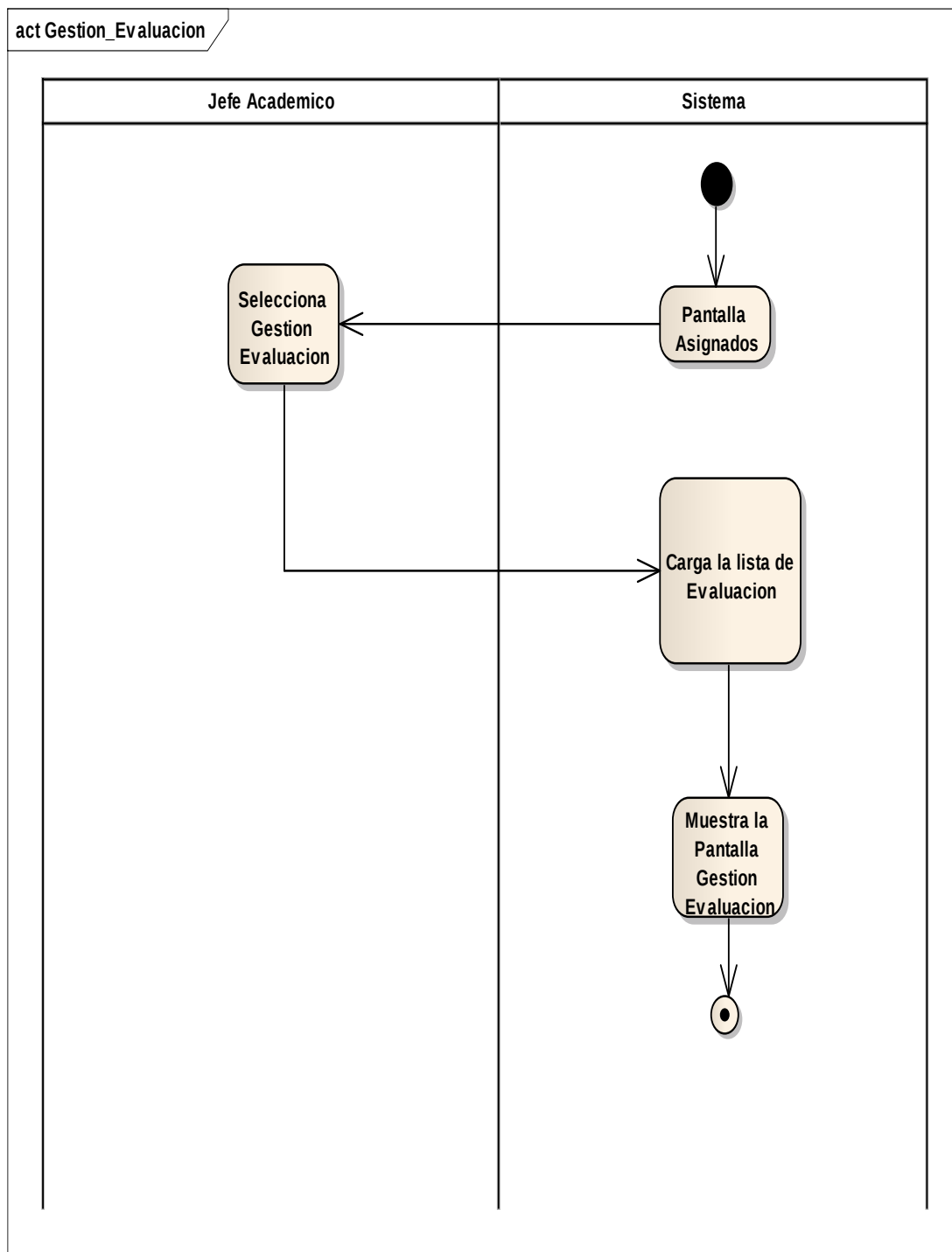


Figura 86. Diagrama De Actividad Gestión Evaluación

2.1.7.5.13.27 Diagrama de Actividad Adicionar Evaluación

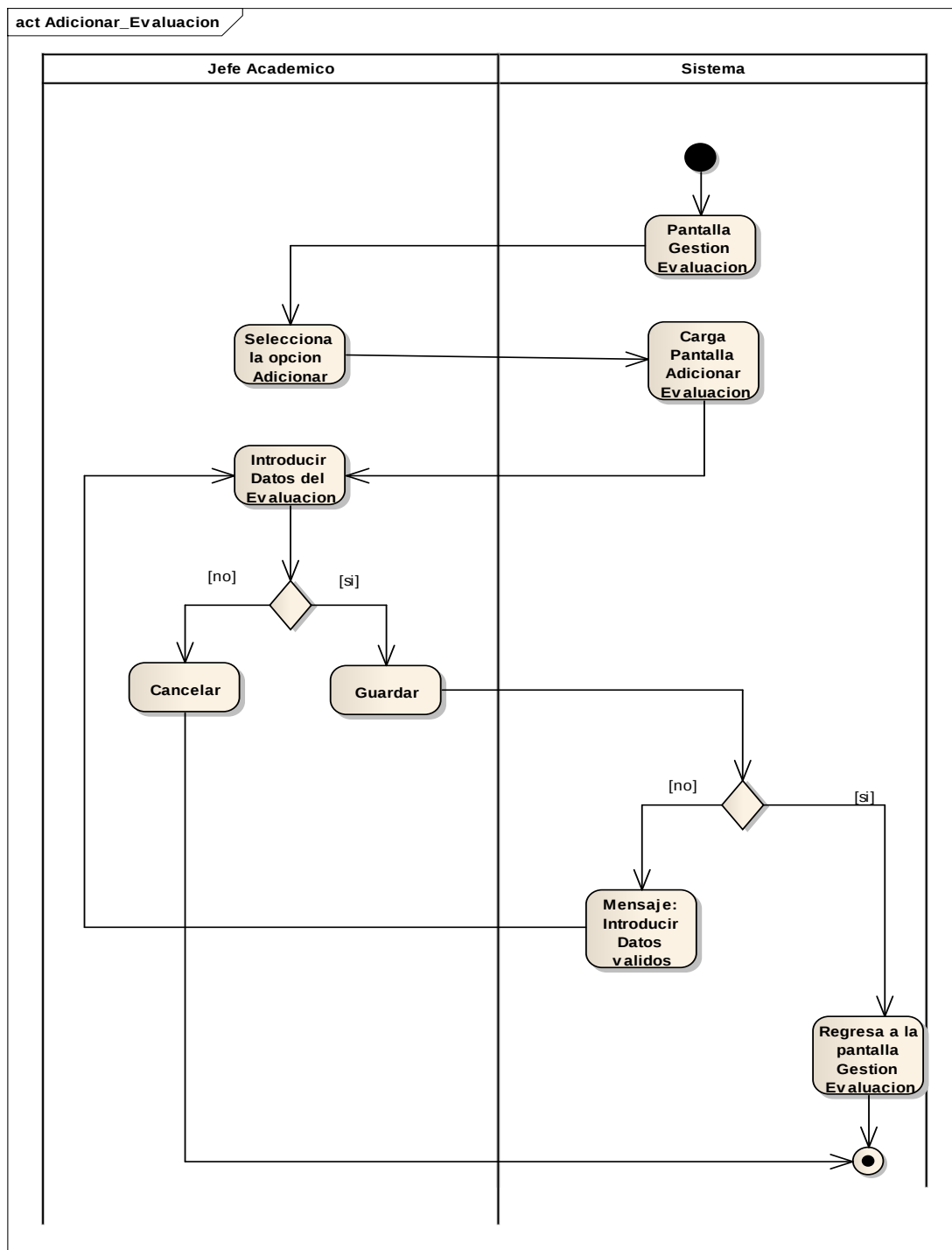


Figura 87. Diagrama De Actividad Adicionar Evaluación

2.1.7.5.13.28 Diagrama de Actividad Modificar Evaluación

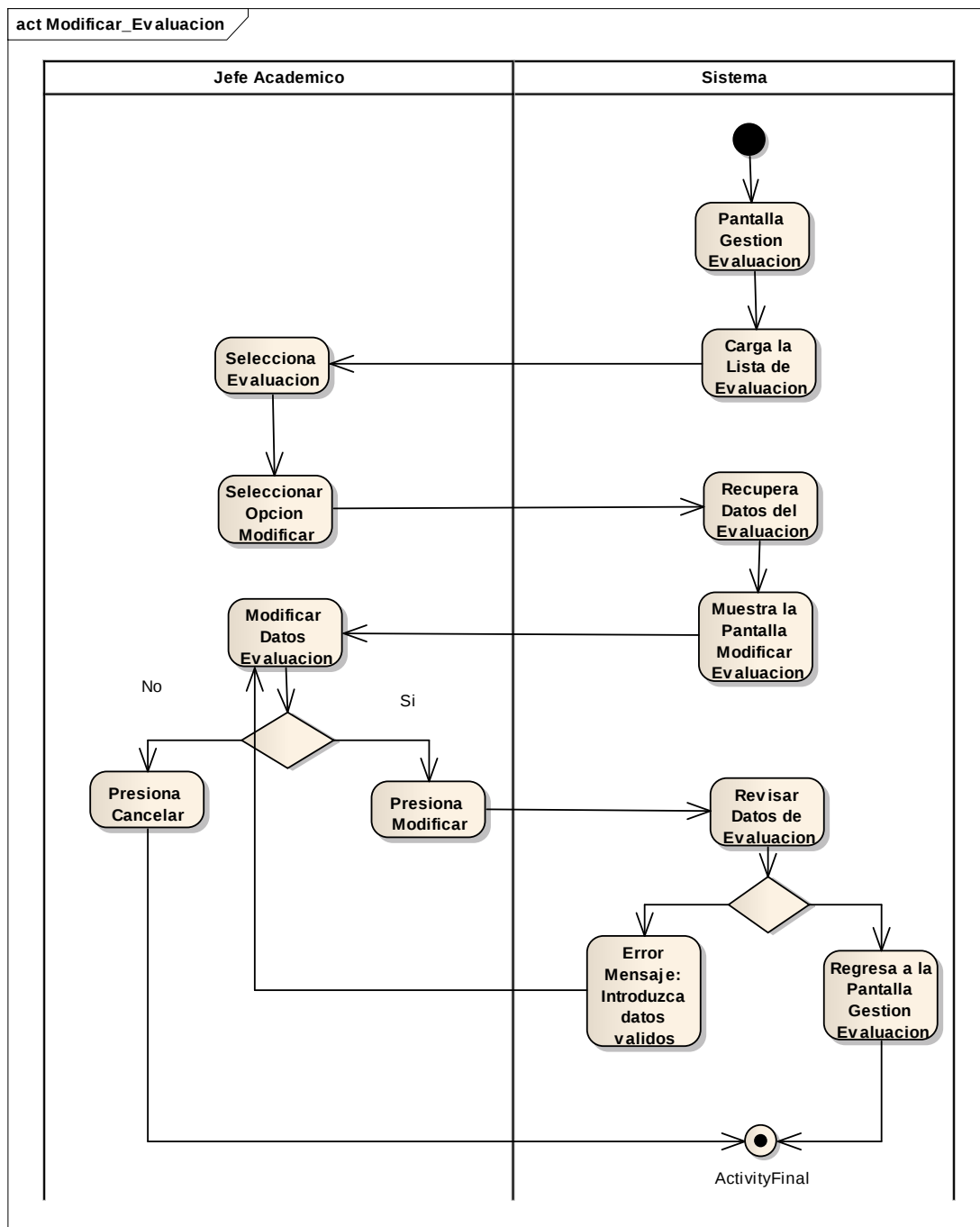


Figura 88. Diagrama De Actividad Modificar Evaluación

2.1.7.5.13.29 Diagrama de Actividad Asignaciones

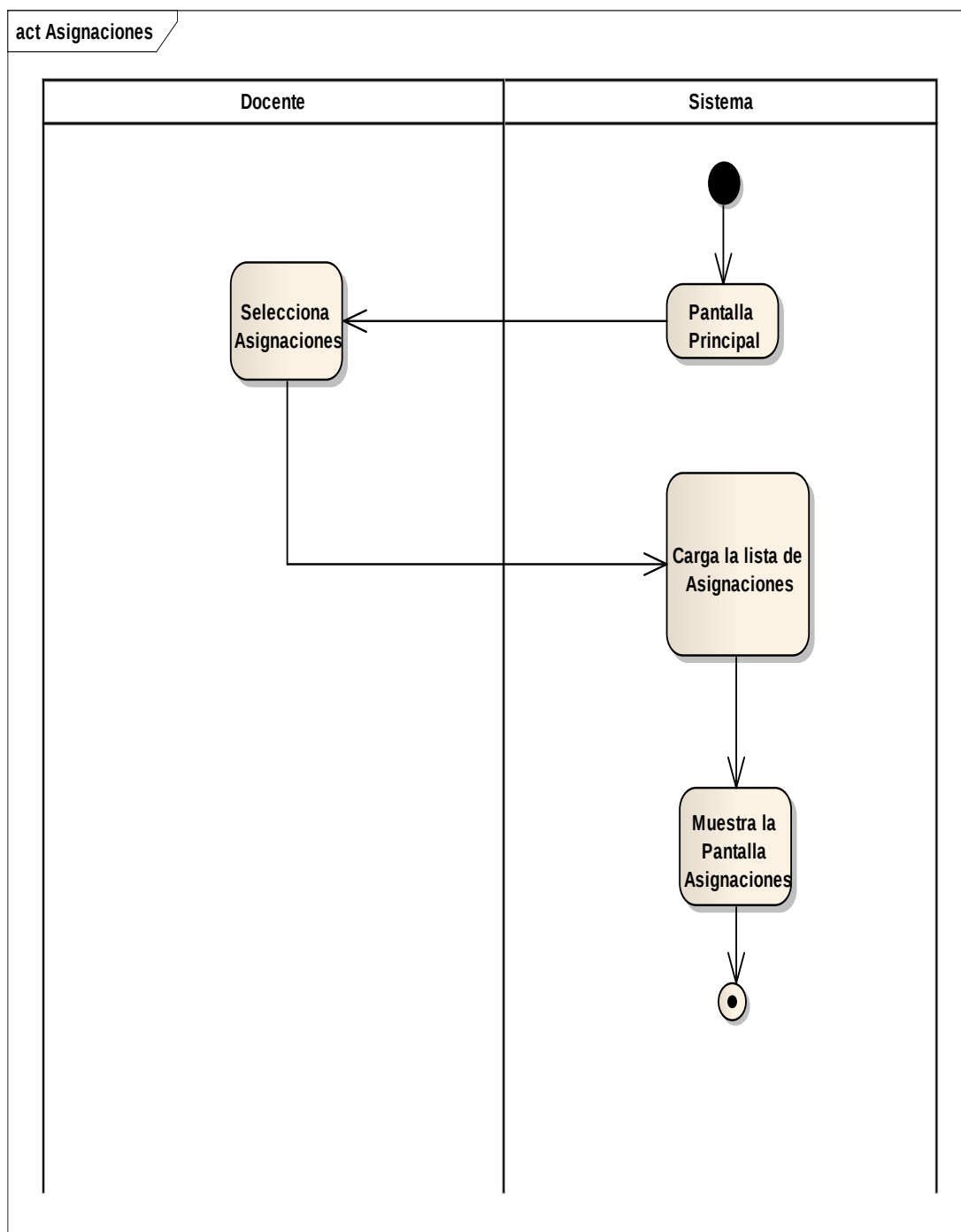


Figura 89. Diagrama De Actividad Asignaciones

2.1.7.5.13.30 Diagrama de Actividad Ponderación

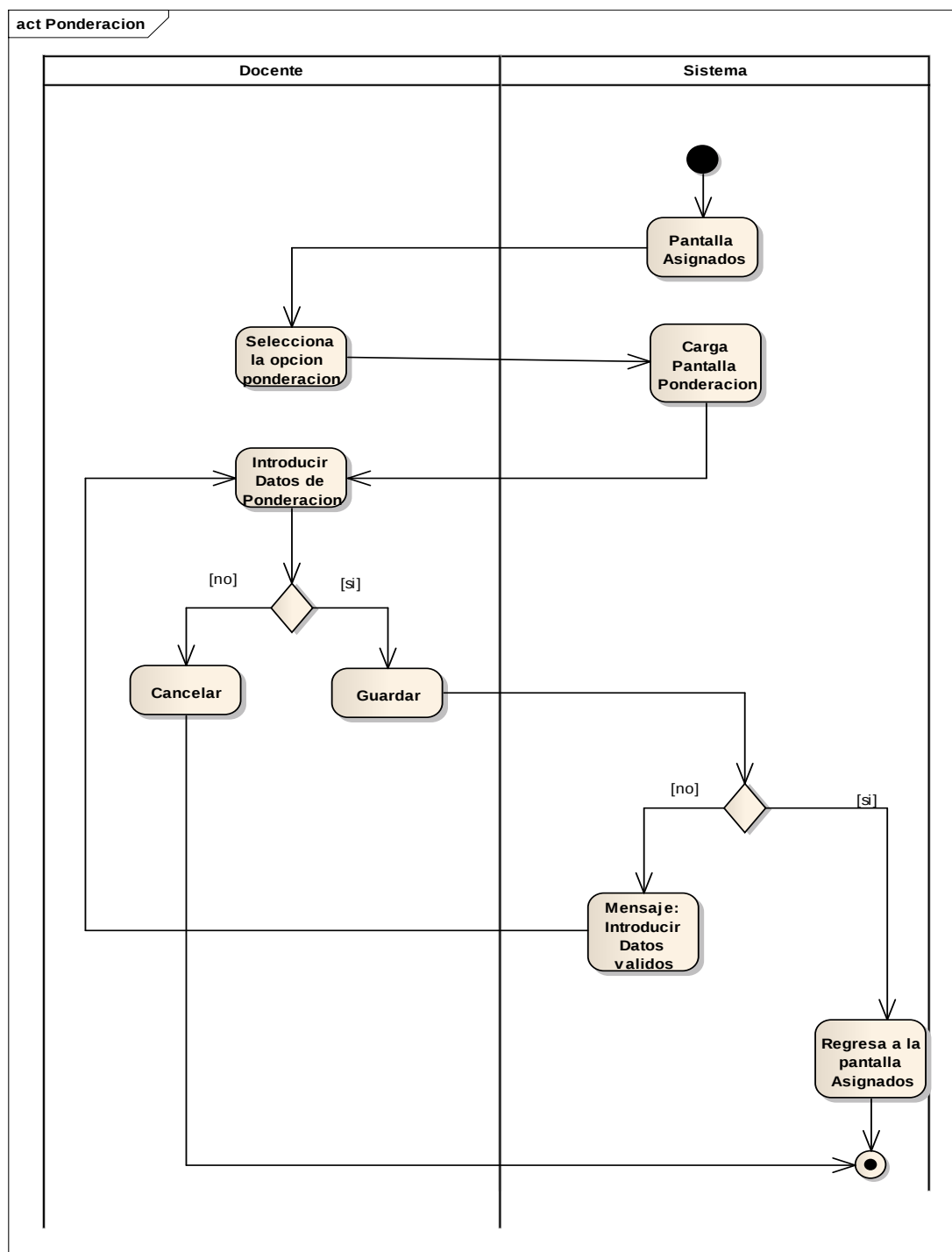


Figura 90. Diagrama De Actividad Ponderación

2.1.7.5.14 Modelo de Diagramas de Secuencias

2.1.7.5.14.1 Introducción

Un diagrama de secuencia del sistema es un artefacto creado de manera rápida y fácil que muestra los eventos de entrada y salida relacionados con el sistema que está estudiando.

UML incluye la notación de los diagramas de secuencia

Los diagramas de secuencia es un dibujo que muestra, para un escenario específico de un caso de uso, los eventos que generan los actores externos, el orden y los eventos entre los sistemas.

2.1.7.5.14.2 Propósito

- ❖ Comprender la estructura y la dinámica del sistema deseado para la organización.
- ❖ Comprender la interacción de los actores del sistema.

2.1.7.5.14.3 Alcance

- ❖ Describe un escenario específico de un caso de uso
- ❖ Representar las interacciones entre actores y operaciones que inician.
- ❖ Describe la interacción entre los objetos del sistema

2.1.7.5.15 Diagramas de Secuencia

2.1.7.5.15.1 Diagrama de Secuencia Gestión Usuarios

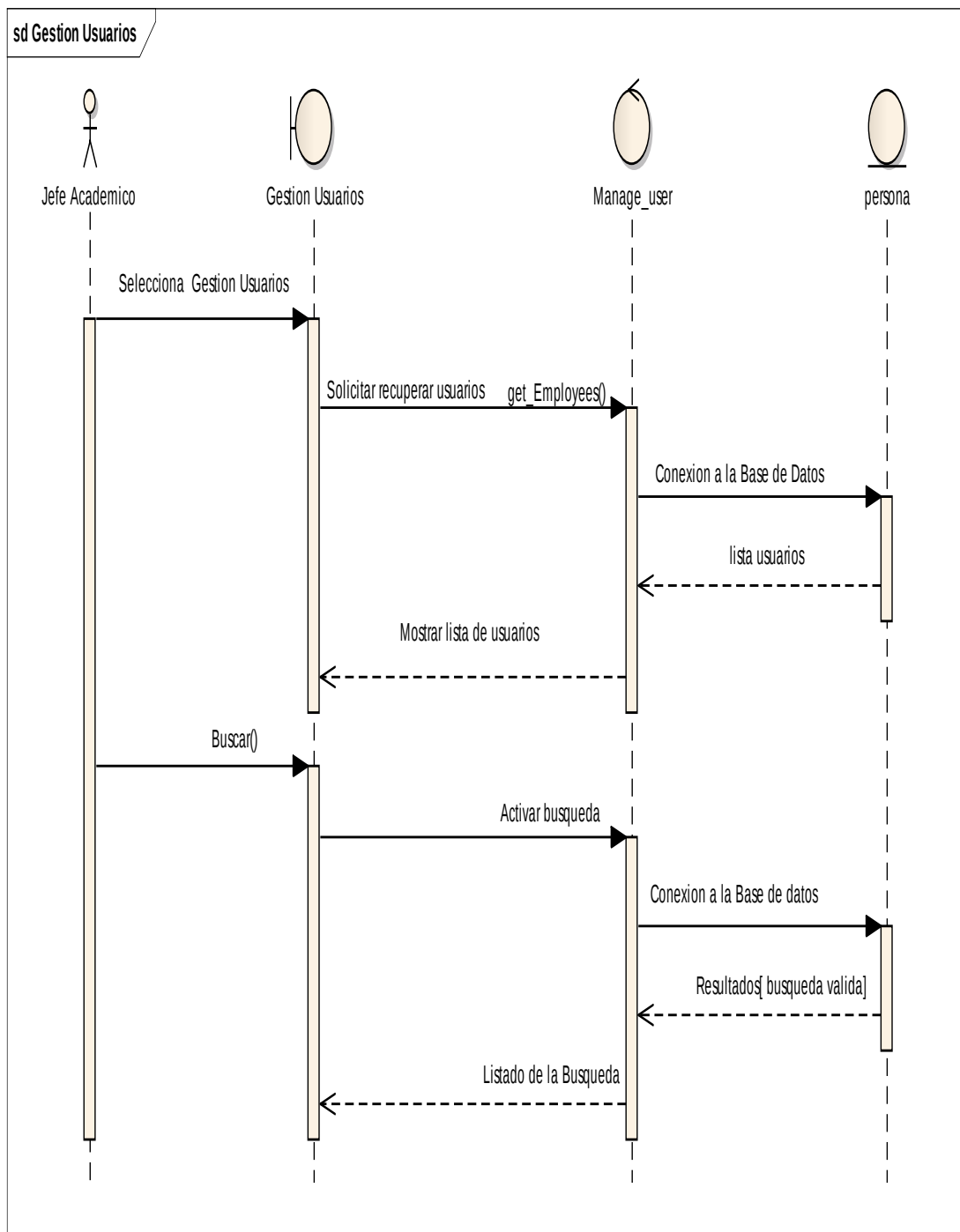


Figura 91. Diagrama de Secuencia Administrar Usuarios

2.1.7.5.15.2 Diagrama de Secuencia Adicionar Usuarios

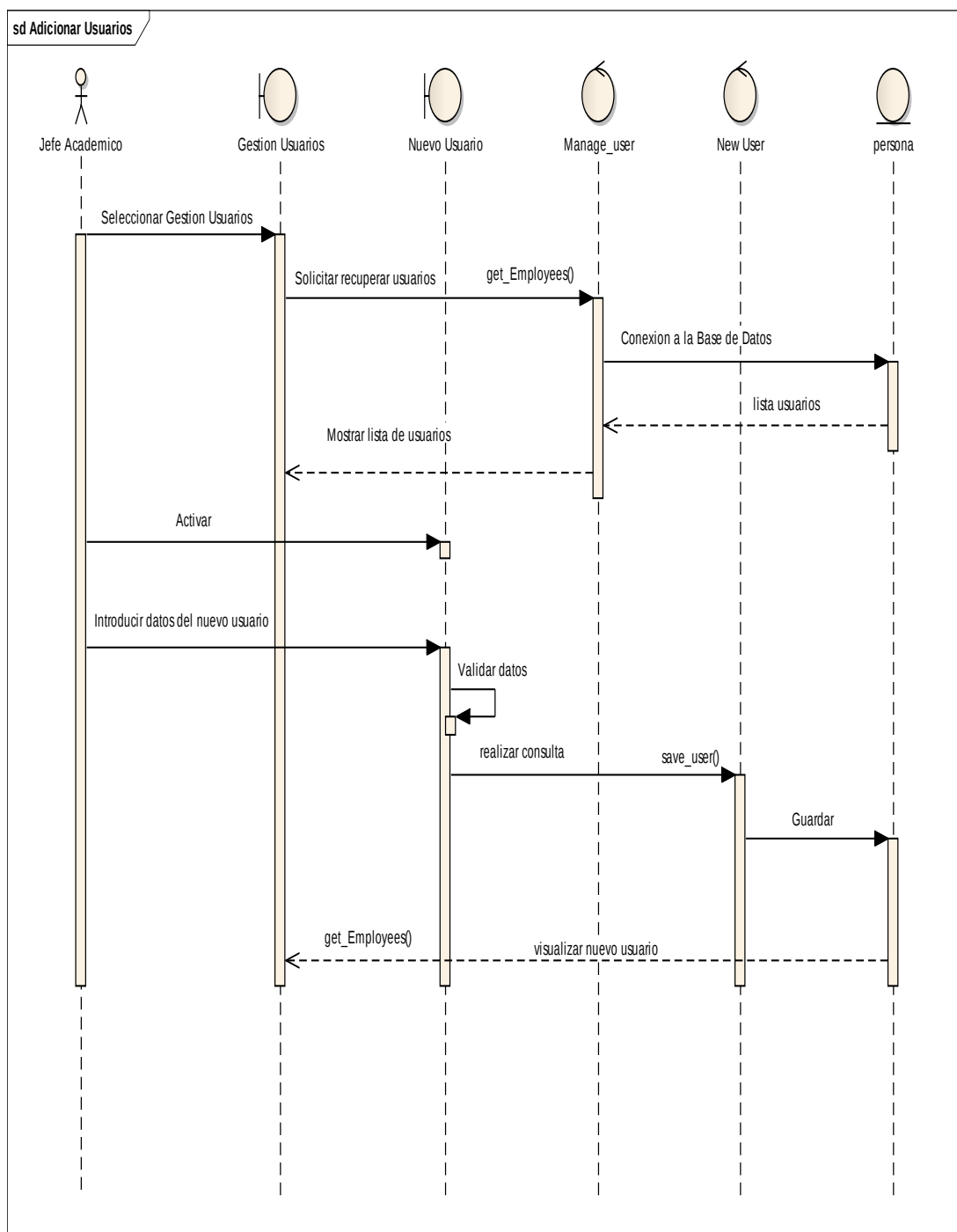


Figura 92. Diagrama de Secuencia Adicionar Usuarios

2.1.7.5.15.3

Diagrama de Secuencia Modificar Usuario

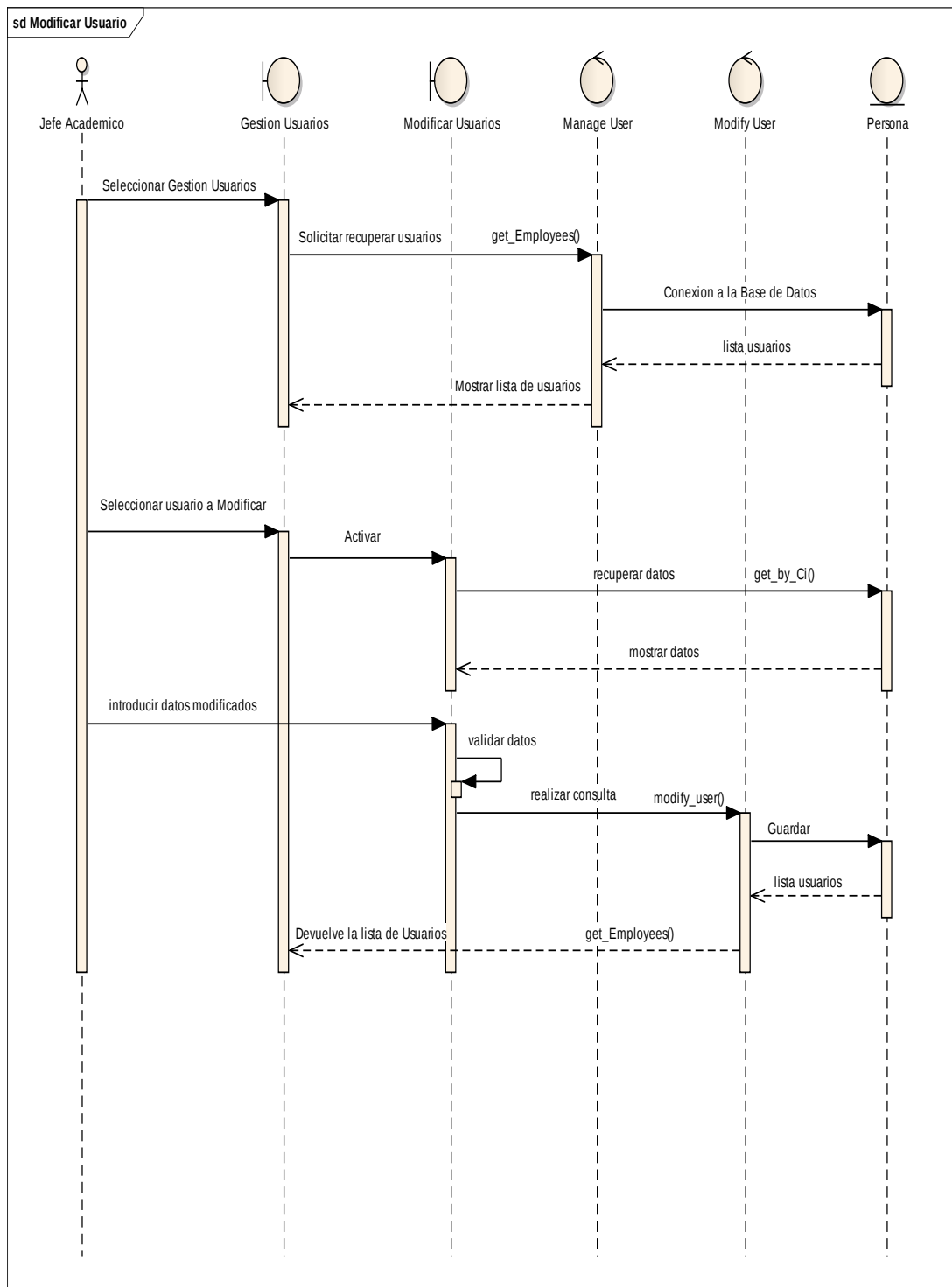


Figura 93. Diagrama de Secuencia Modificar Usuario

2.1.7.5.15.4 Diagrama de Secuencia Eliminar Usuario

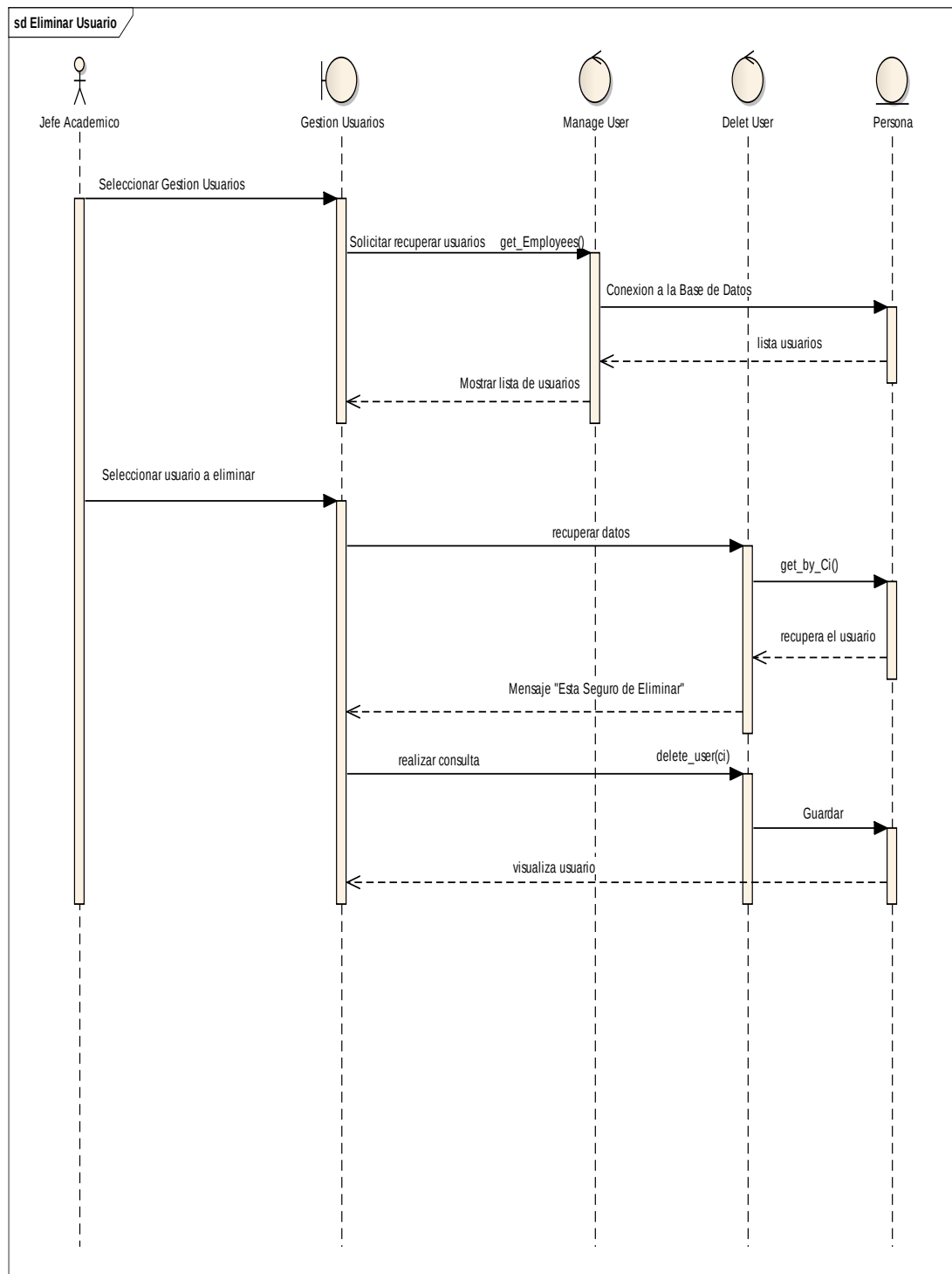


Figura 94. Diagrama De Secuencia Ver Usuario

2.1.7.5.15.5 Diagrama de Secuencia Adicionar Clave y Login

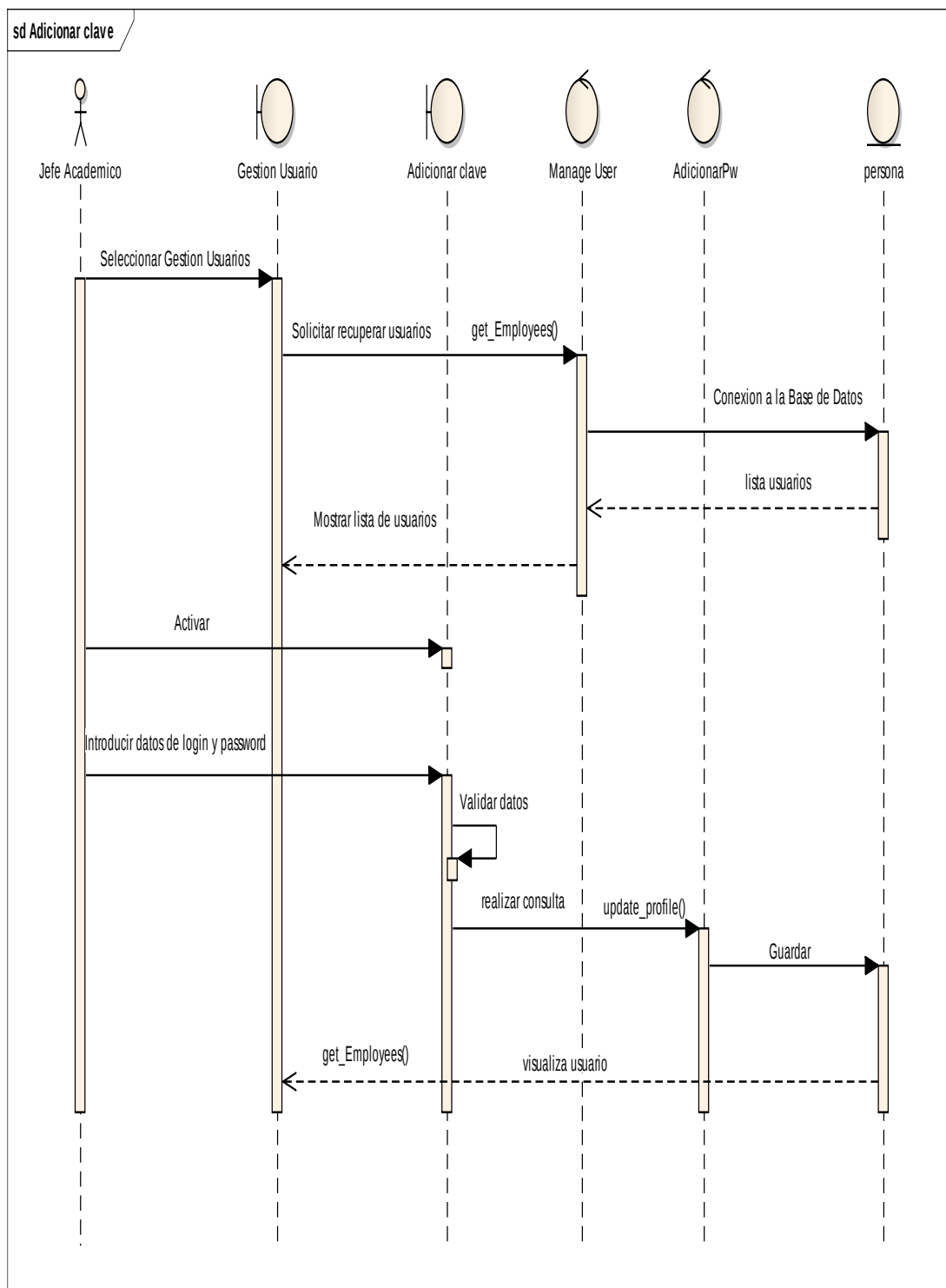


Figura 95. Diagrama De Secuencia Adicionar Clave Y Login

2.1.7.5.15.6 Diagrama de Secuencia Gestión Cursos

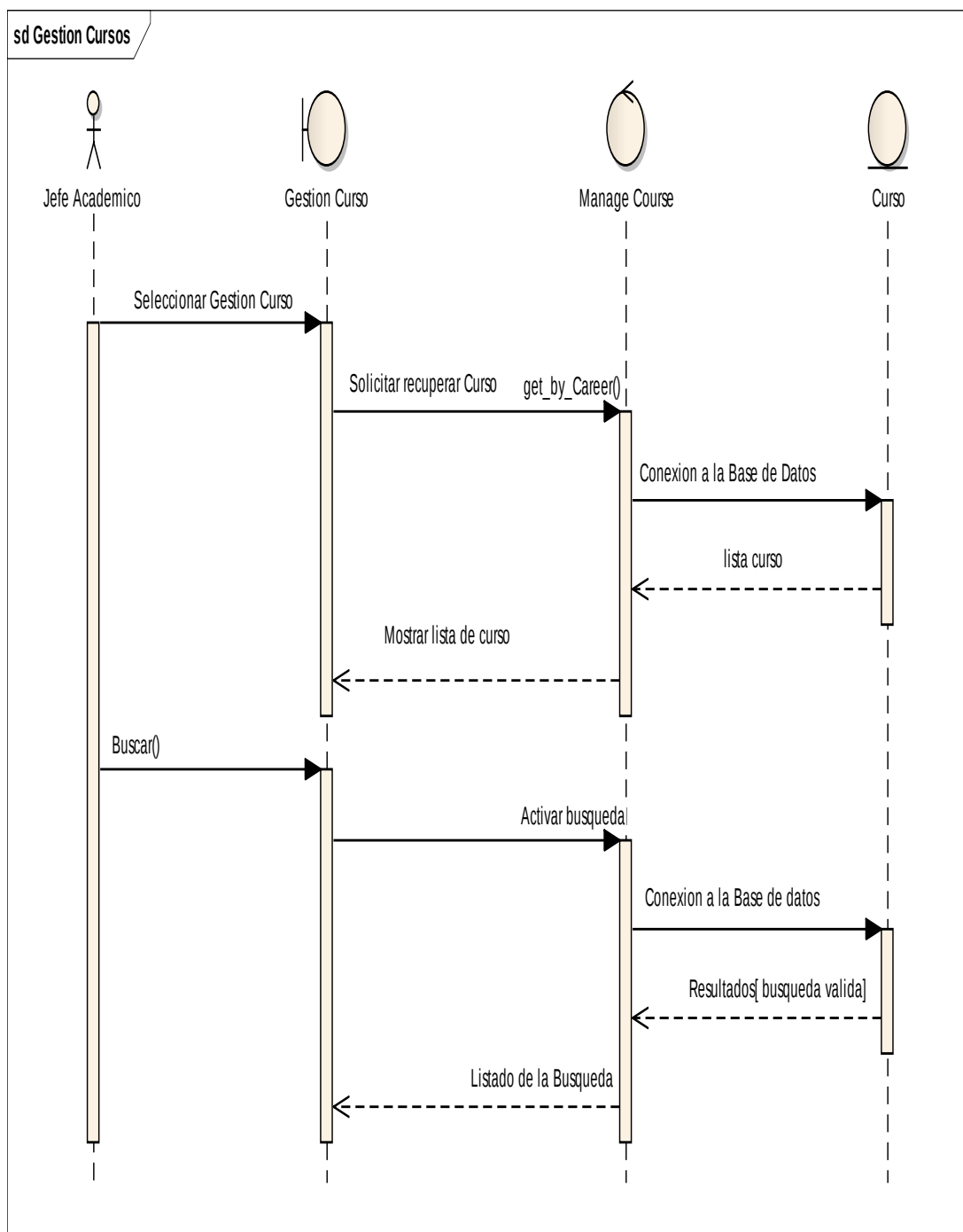


Figura 96. Diagrama De Secuencia Gestión Cursos

2.1.7.5.15.7 Diagrama de Secuencia Adicionar Curso

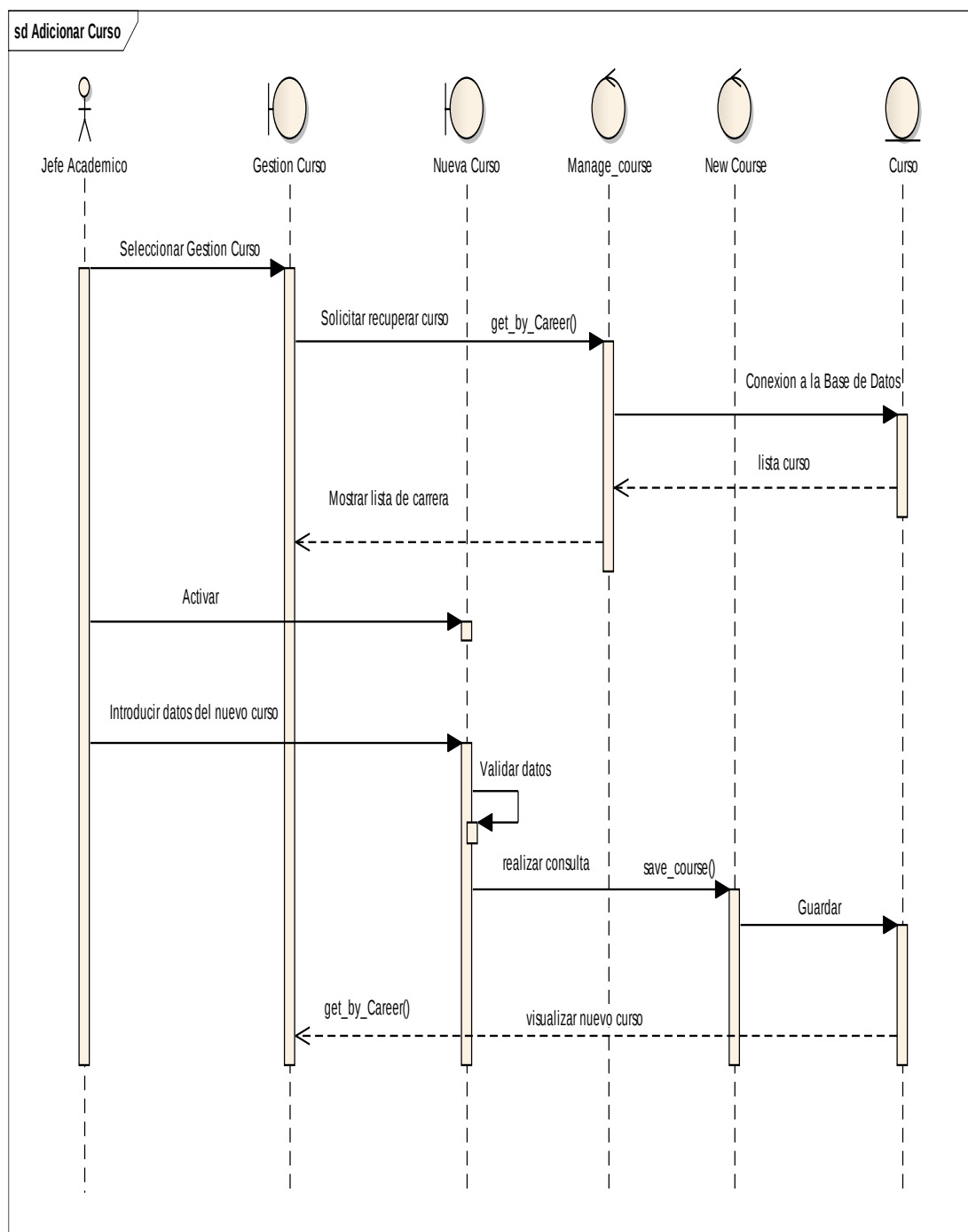


Figura 97. Diagrama De Secuencia Adicionar Curso

2.1.7.5.15.8 Diagrama de Secuencia Modificar Curso

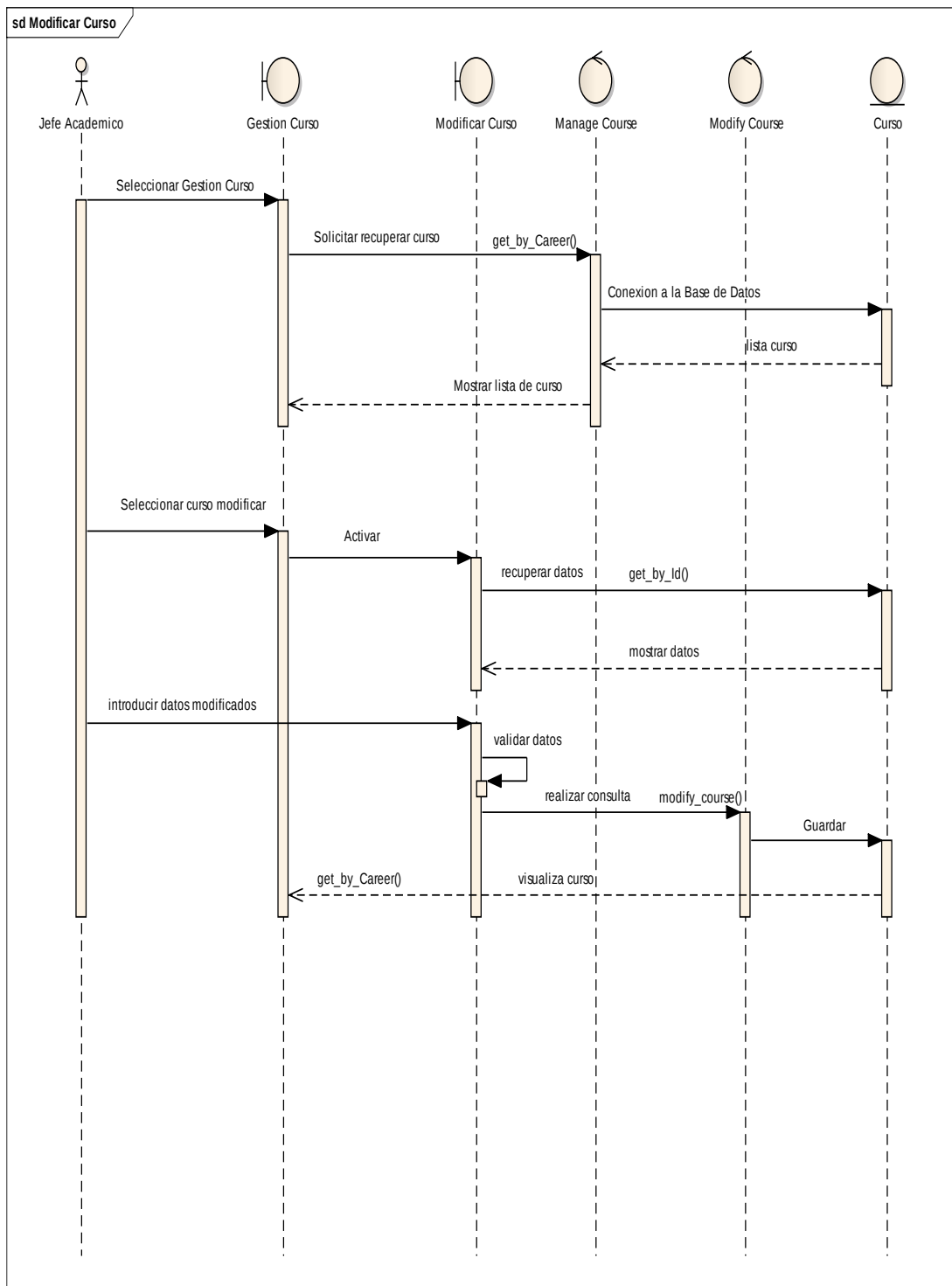


Figura 98. Diagrama De Secuencia Modificar Curso

2.1.7.5.15.9 Diagrama de Secuencia Eliminar Curso

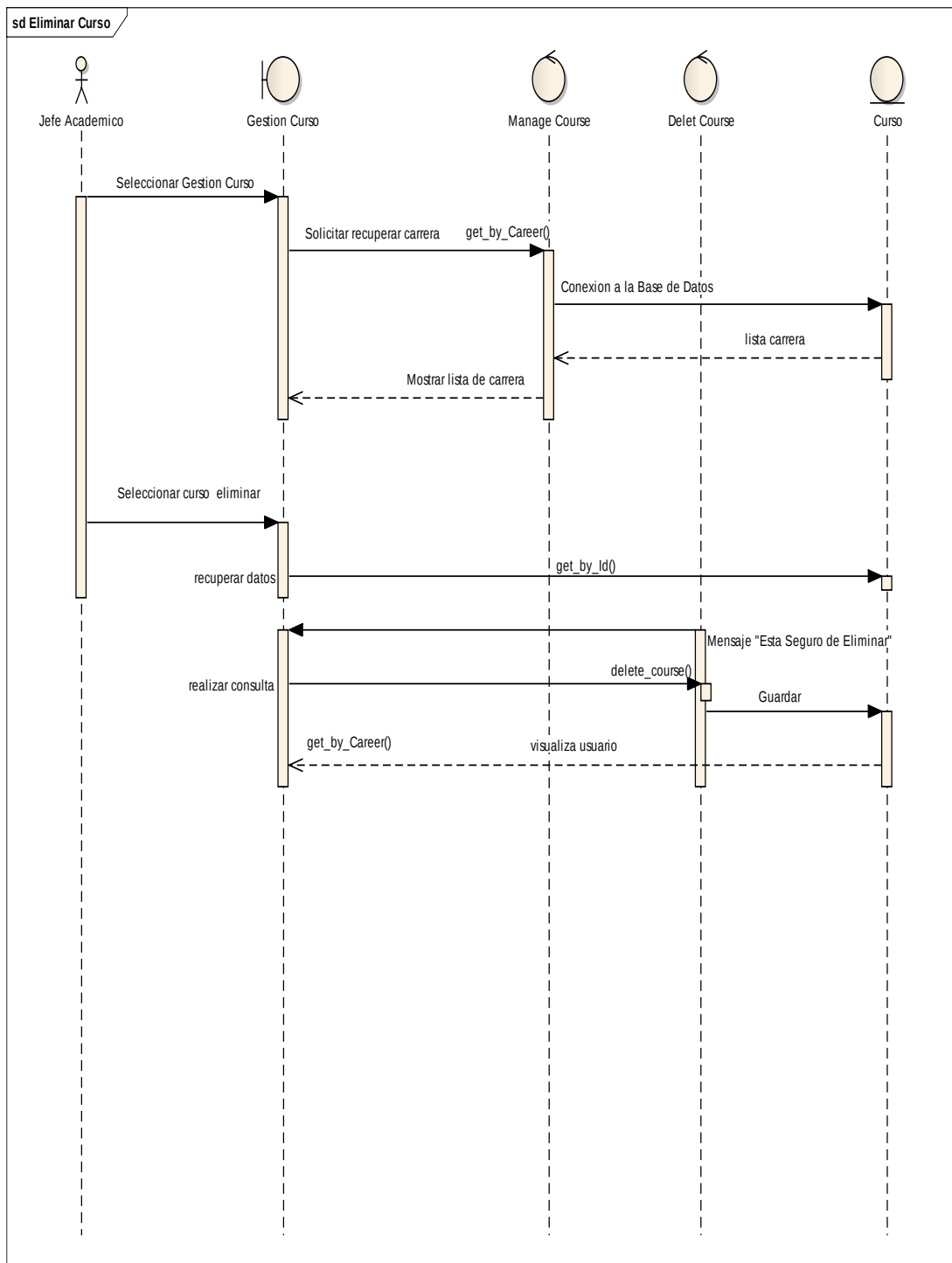


Figura 99. Diagrama De Secuencia Eliminar Curso

2.1.7.5.15.10 Diagrama de Secuencia Gestión Carrera

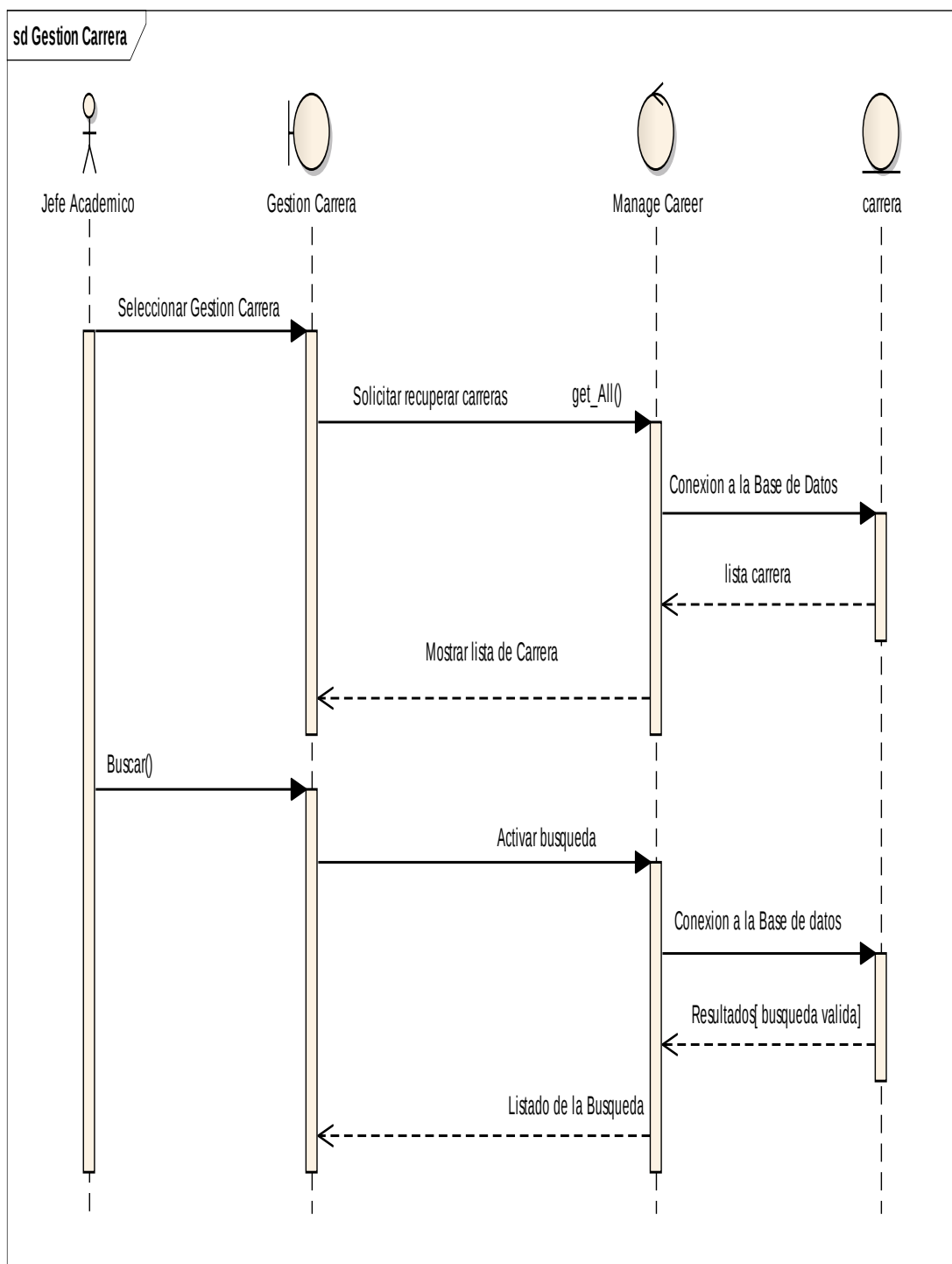


Figura 100. Diagrama De Secuencia Gestión Carrera

2.1.7.5.15.11 Diagramas de Secuencia Adicionar Carrera

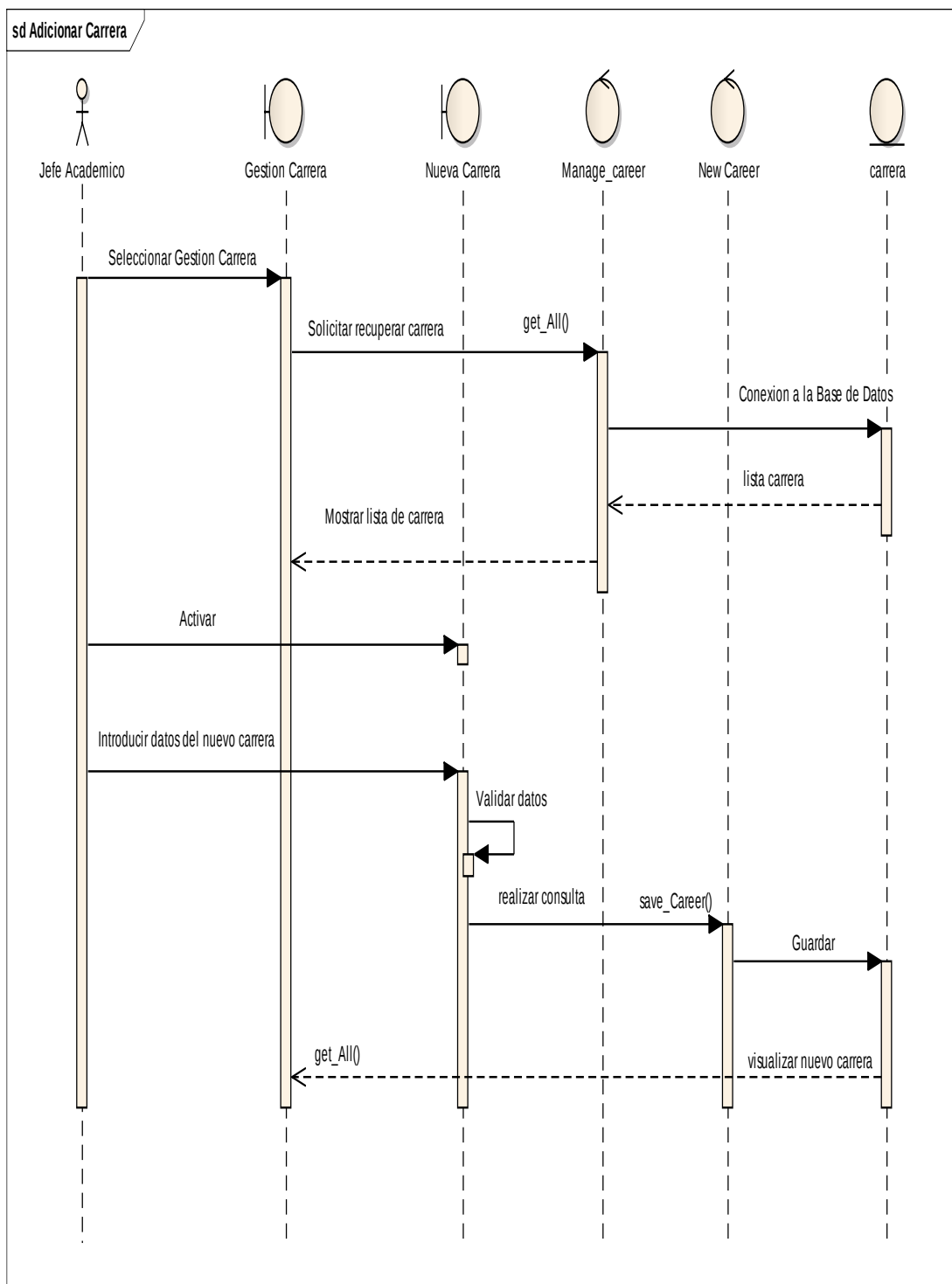


Figura 101. Diagramas De Secuencia Adicionar Carrera

2.1.7.5.15.12 Diagramas de Secuencia Modificar Carrera

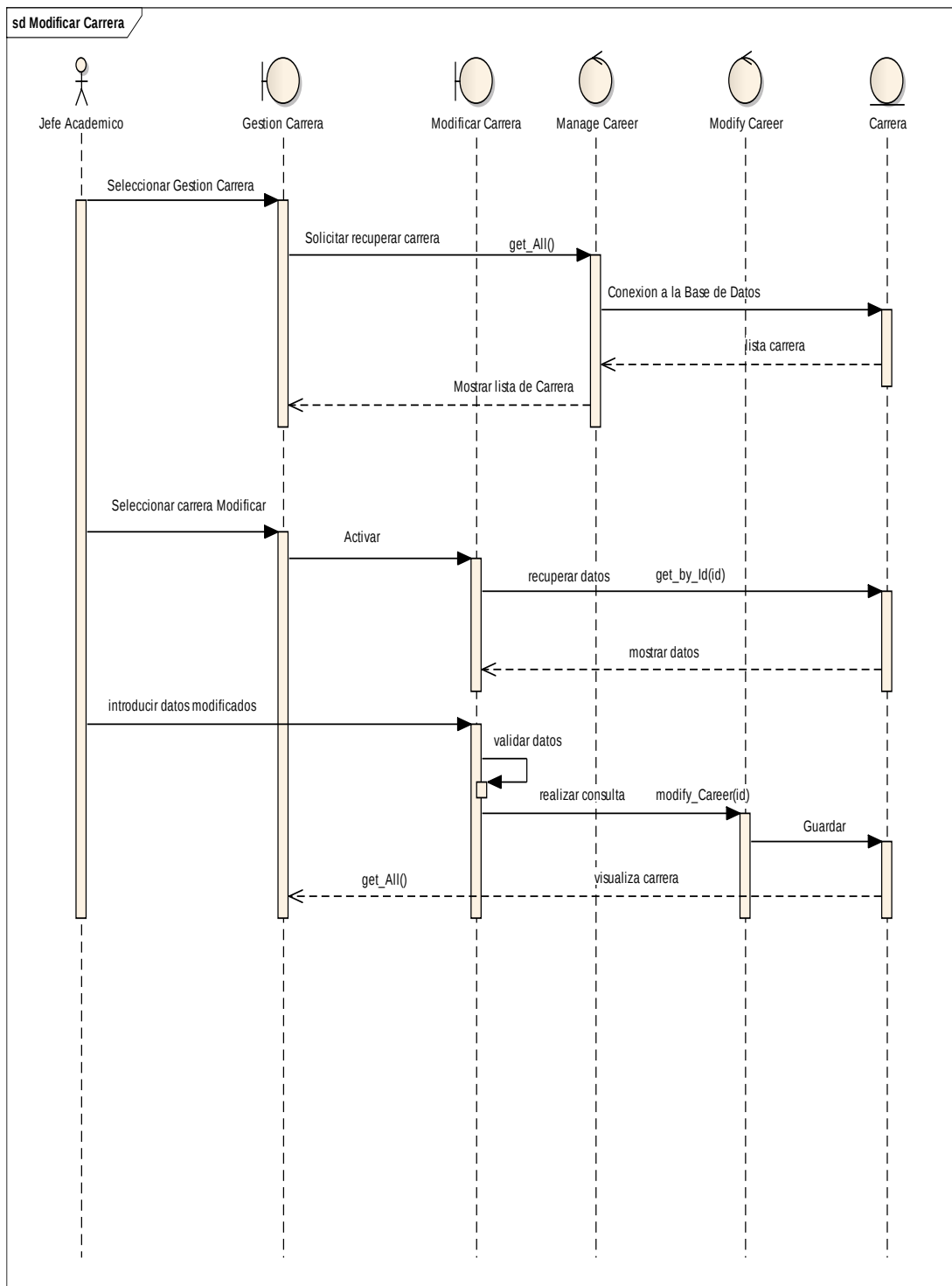


Figura 102. Diagramas De Secuencia Modificar Carrera

2.1.7.5.15.13 Diagramas de Secuencia Eliminar Carrera

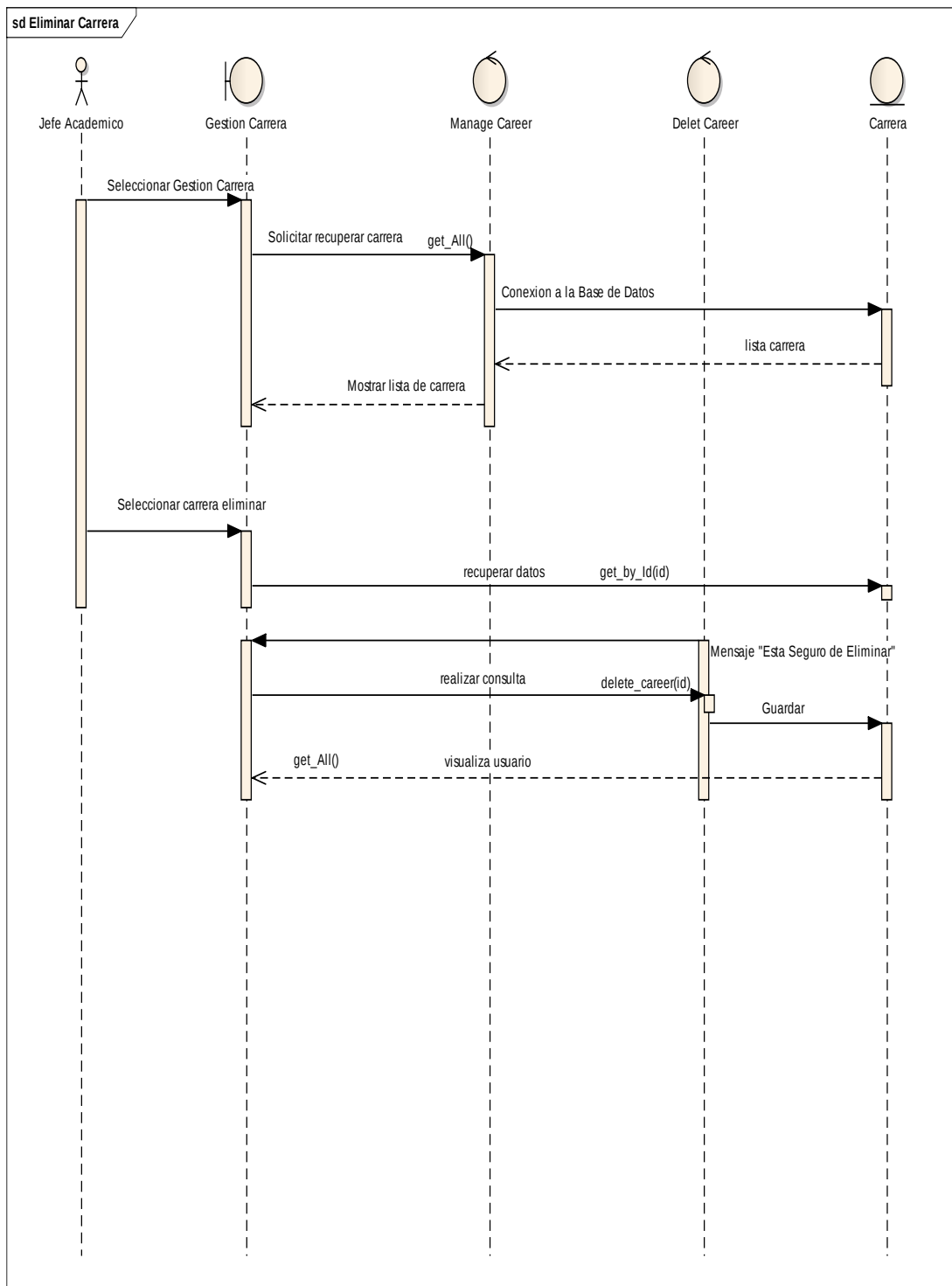


Figura 103. Diagramas De Secuencia Eliminar Carrera

2.1.7.5.15.14 Diagramas de Secuencia Gestión Evaluación

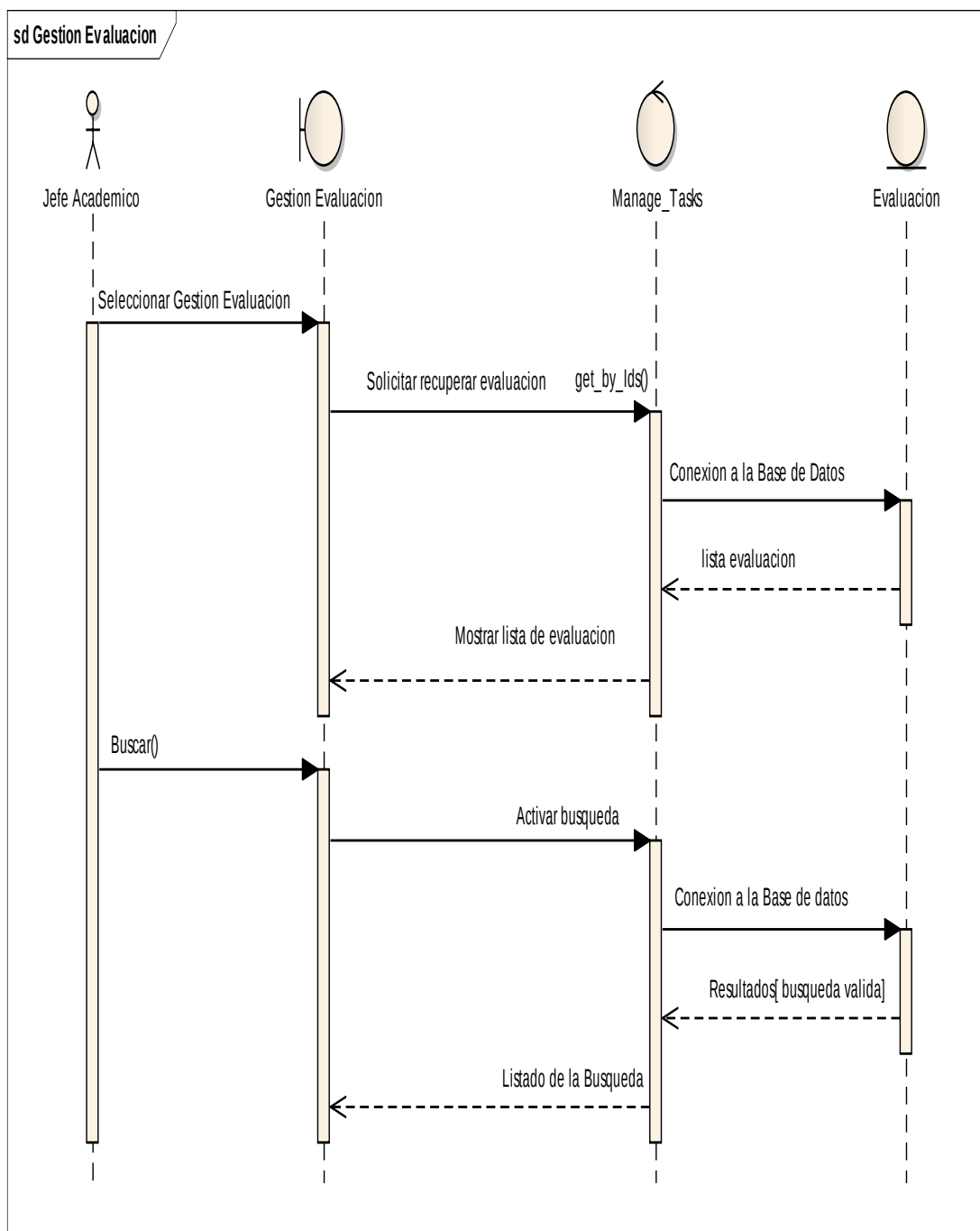


Figura 104. Diagramas De Secuencia Gestión Evaluación

2.1.7.5.15.15 Diagramas de Secuencia Adicionar Evaluación

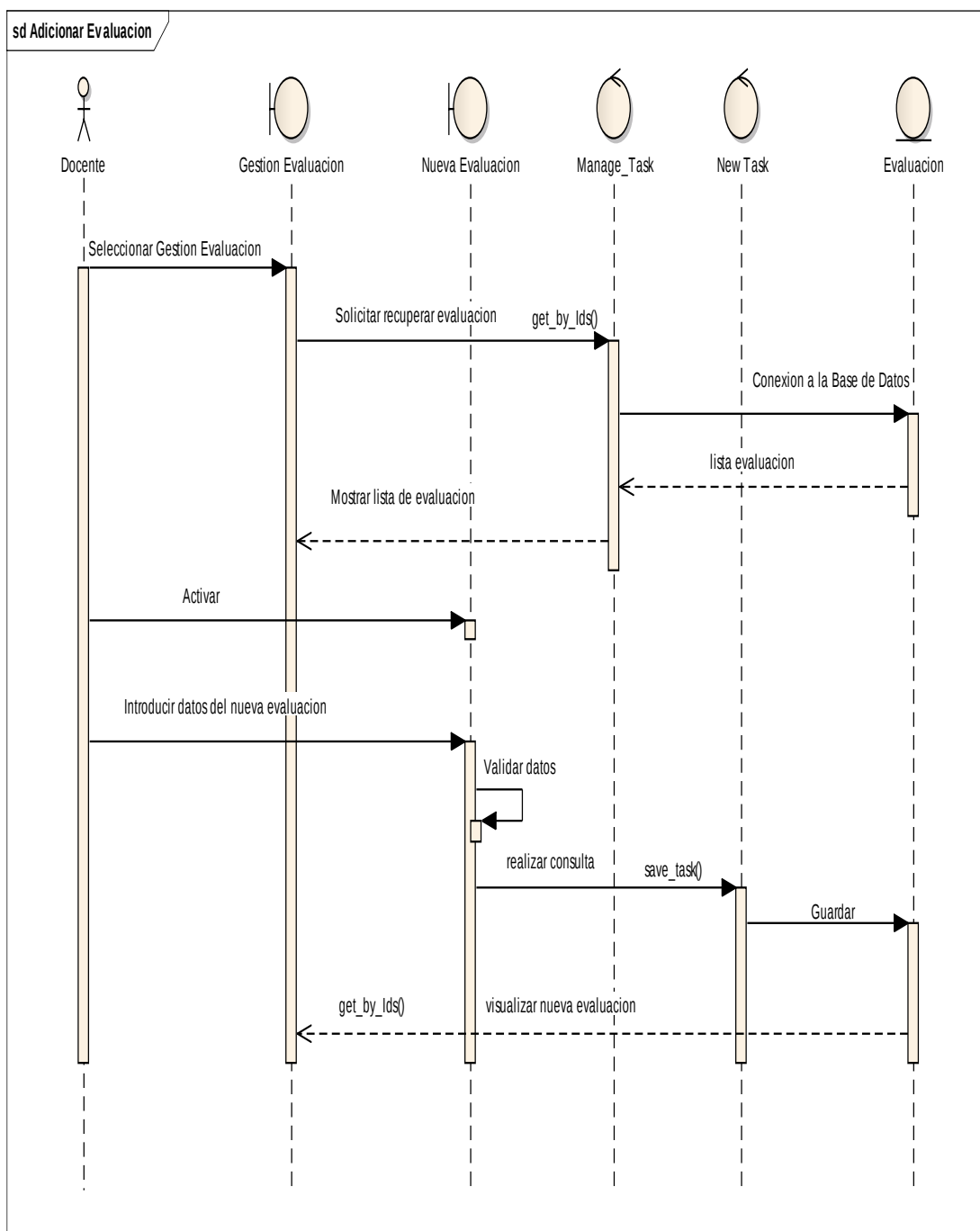


Figura 105. Diagramas De Secuencia Adicionar Evaluación

2.1.7.5.15.16 Diagramas de Secuencia Modificar Evaluación

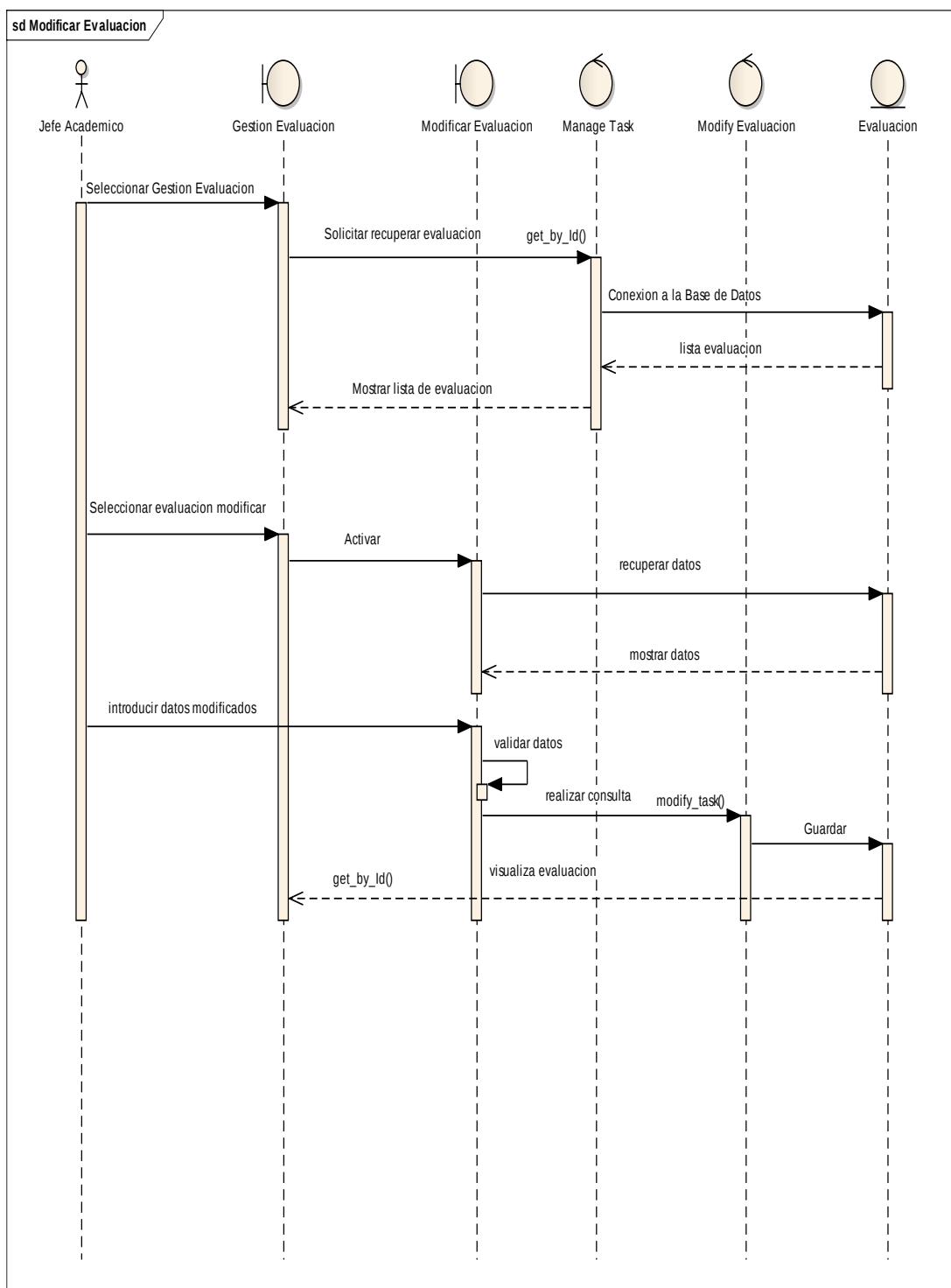


Figura 106. Diagramas De Secuencia Modificar Evaluación

2.1.7.5.15.17 Diagramas de Secuencia Eliminar Evaluación

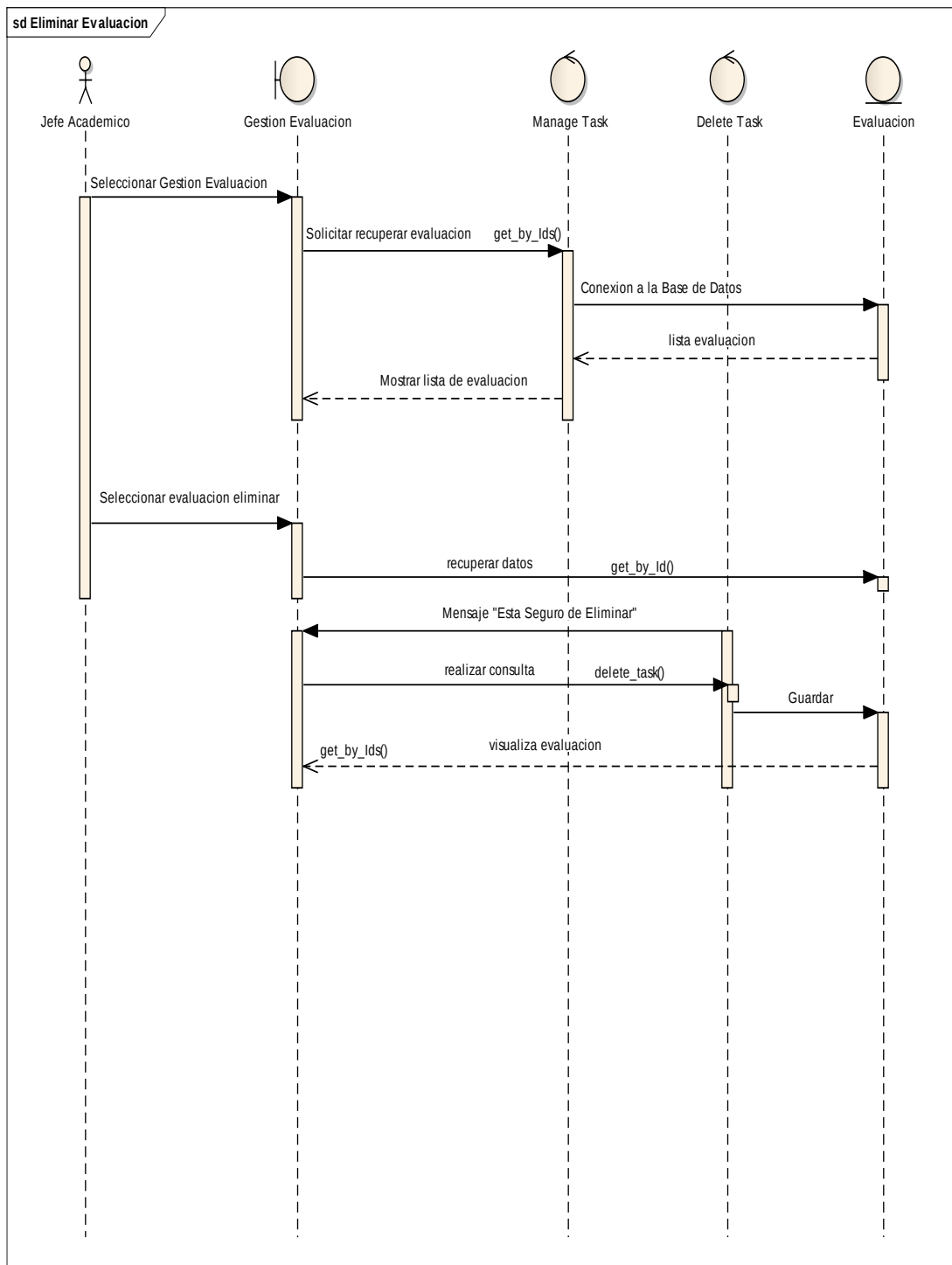


Figura 107. Diagramas de Secuencia Adicionar Requisitos

2.1.7.5.15.18 Diagramas de Secuencia Gestión Grupo

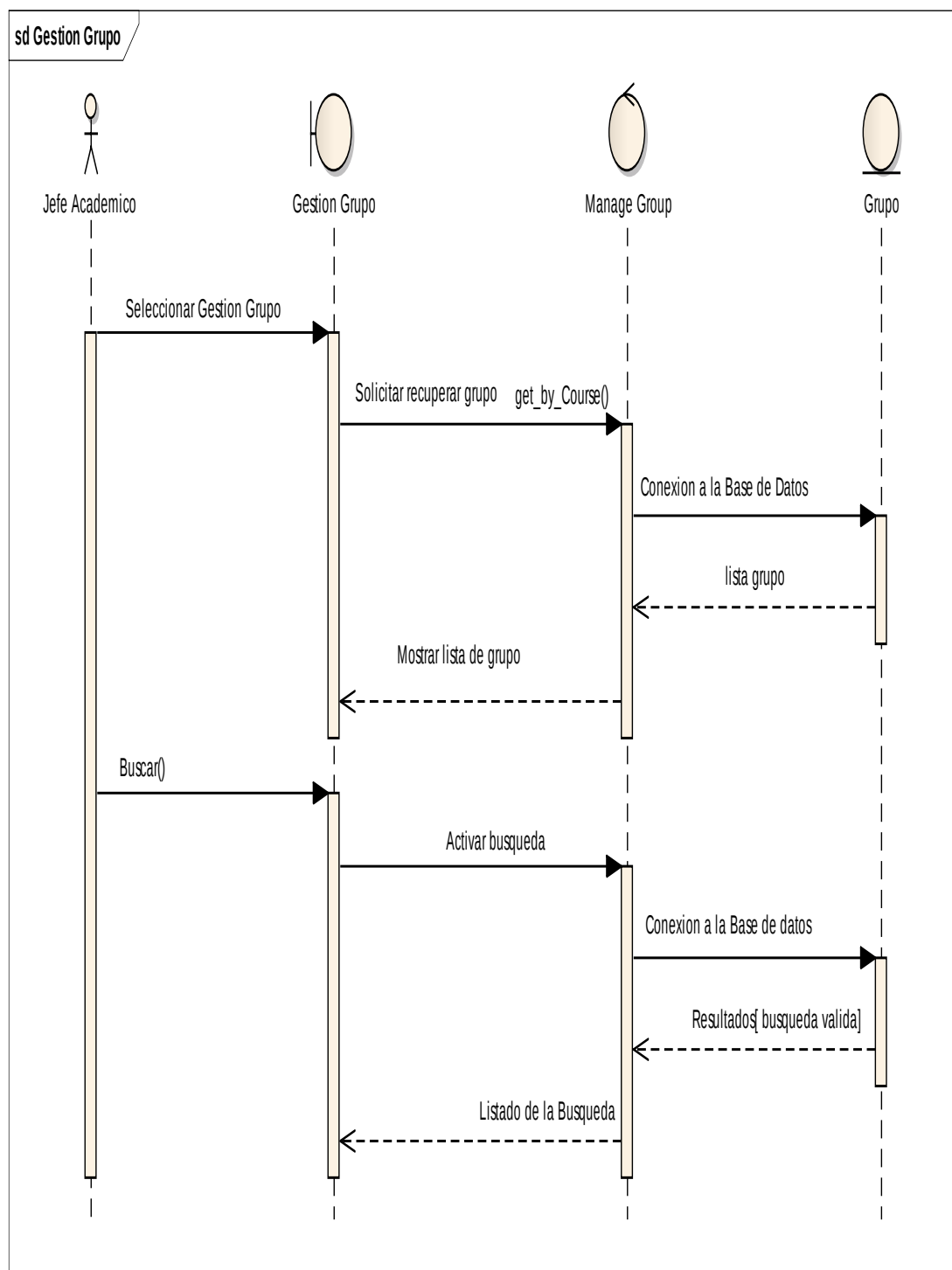


Figura 108. Diagramas De Secuencia Gestión Grupo

2.1.7.5.15.19 Diagramas de Secuencia Adicionar Grupo

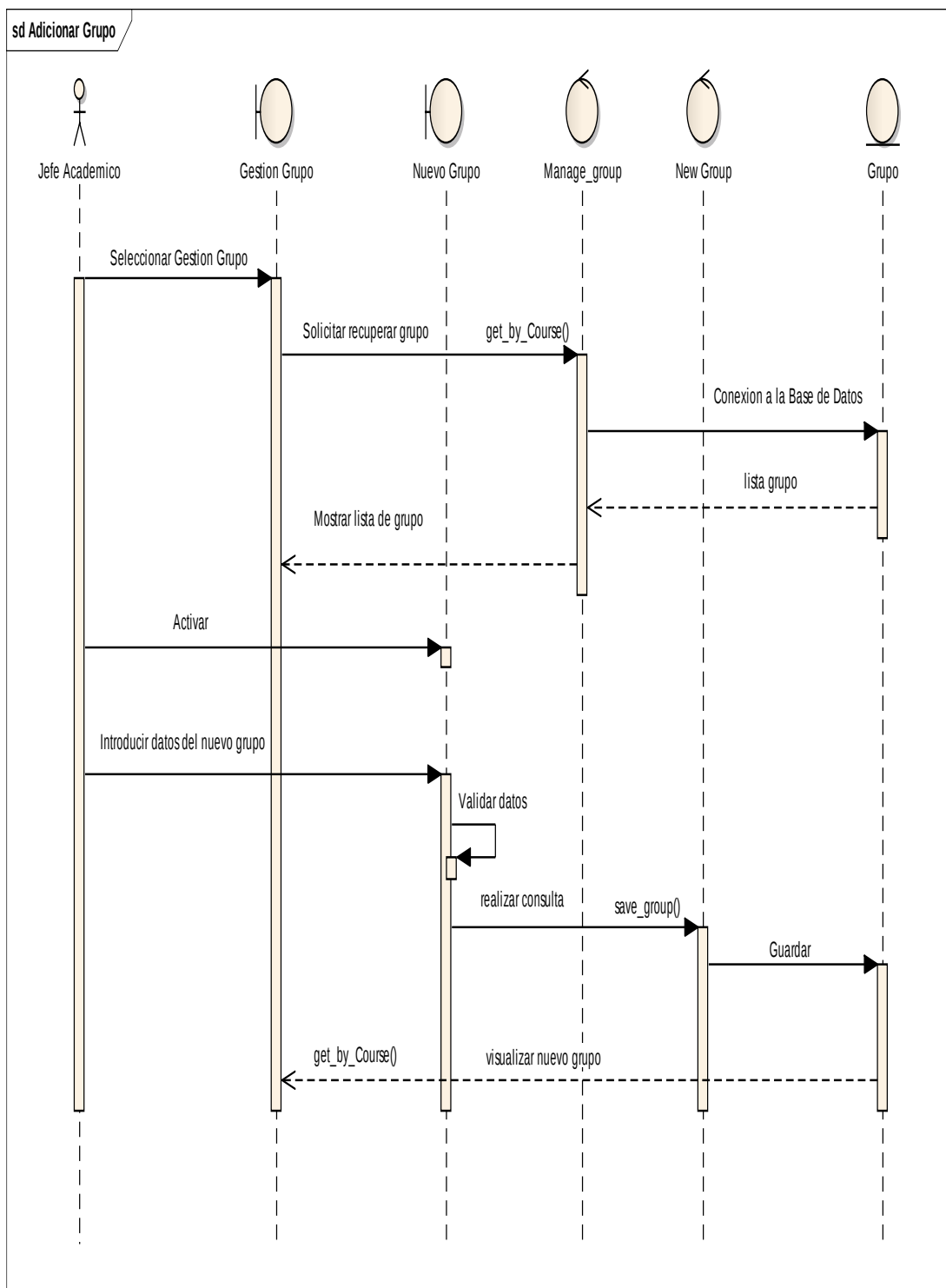


Figura 109. Diagramas De Secuencia Adicionar Grupo

2.1.7.5.15.20 Diagramas de Secuencia Modificar Grupo

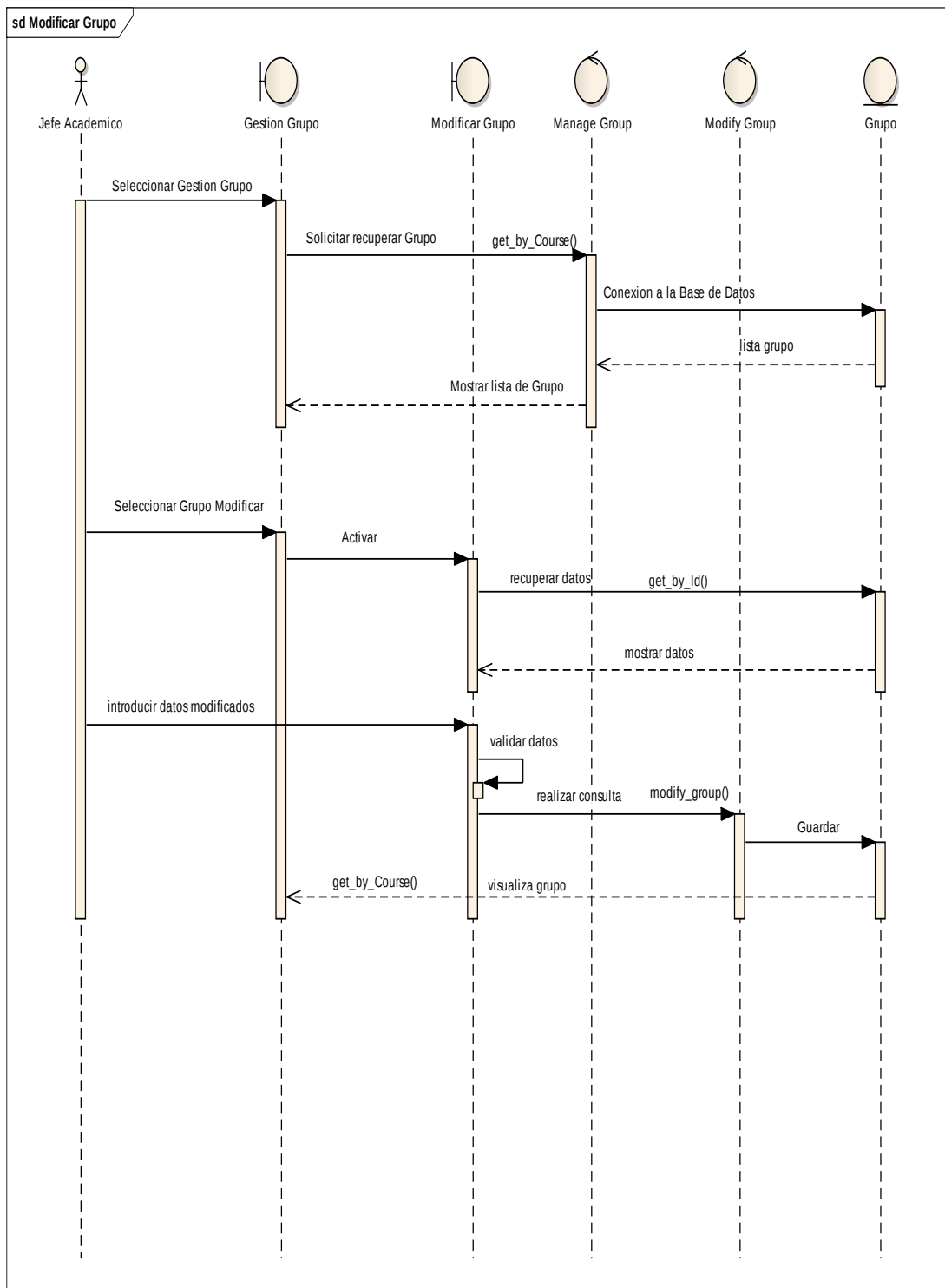


Figura 110. Diagramas de Secuencia Modificar Grupo

2.1.7.5.15.21 Diagramas de Secuencia Eliminar Grupo

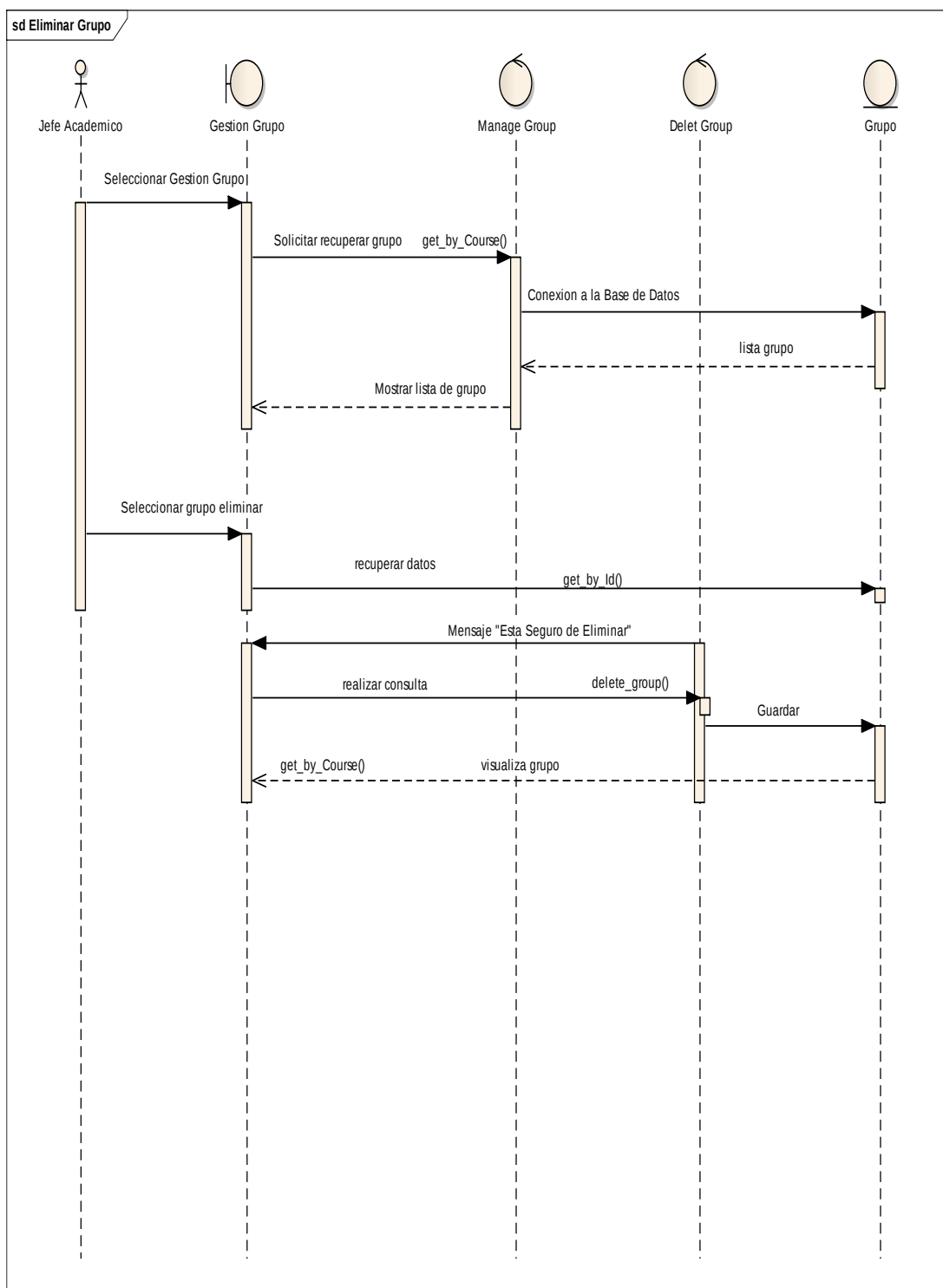


Figura 111. Diagramas De Secuencia Eliminar Grupo

2.1.7.5.15.22 Diagramas de Secuencia Asignaciones

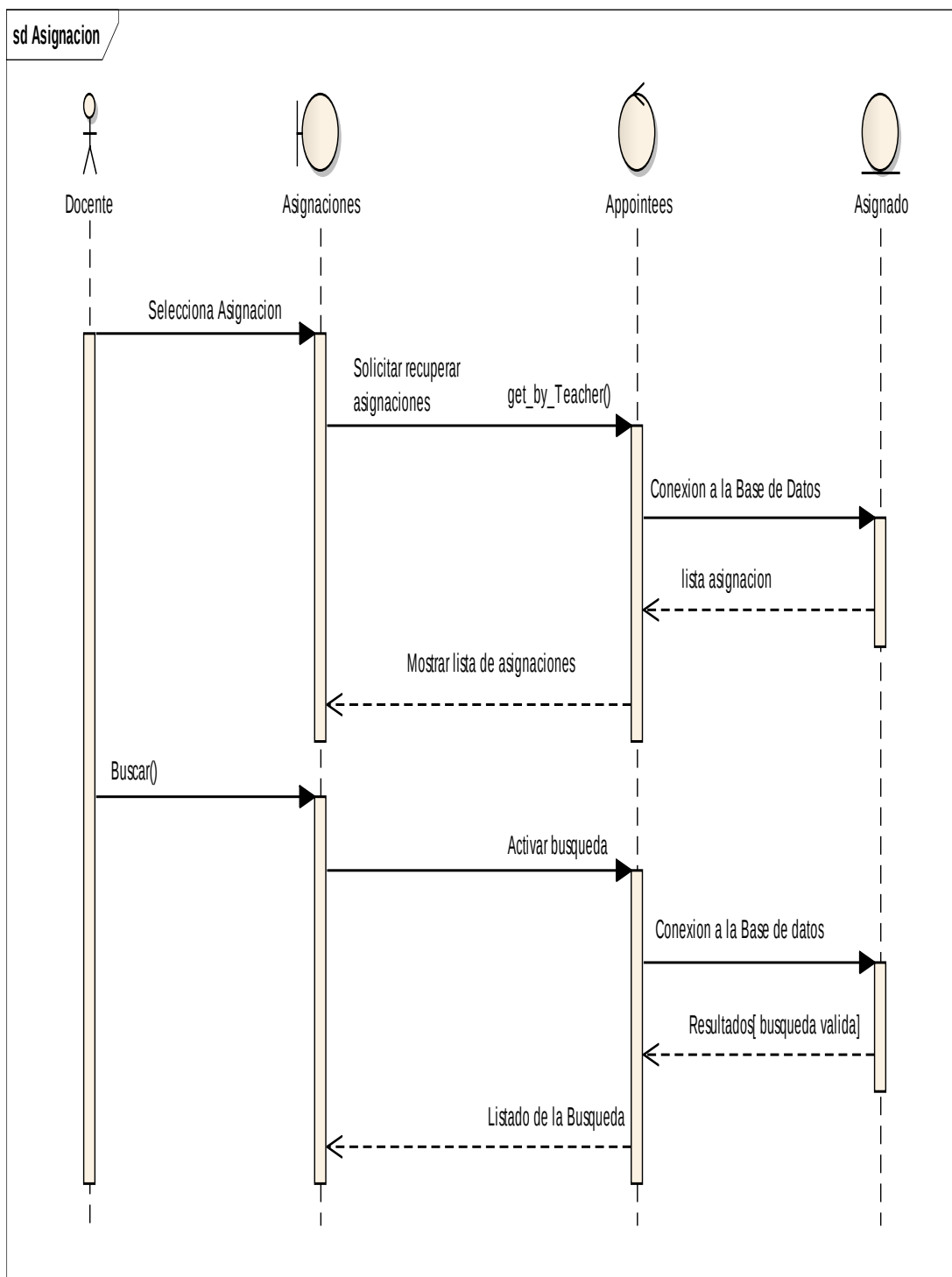


Figura 112. Diagramas De Secuencia Gestión Asignaciones

2.1.7.5.15.23 Diagramas de Secuencia Ponderación

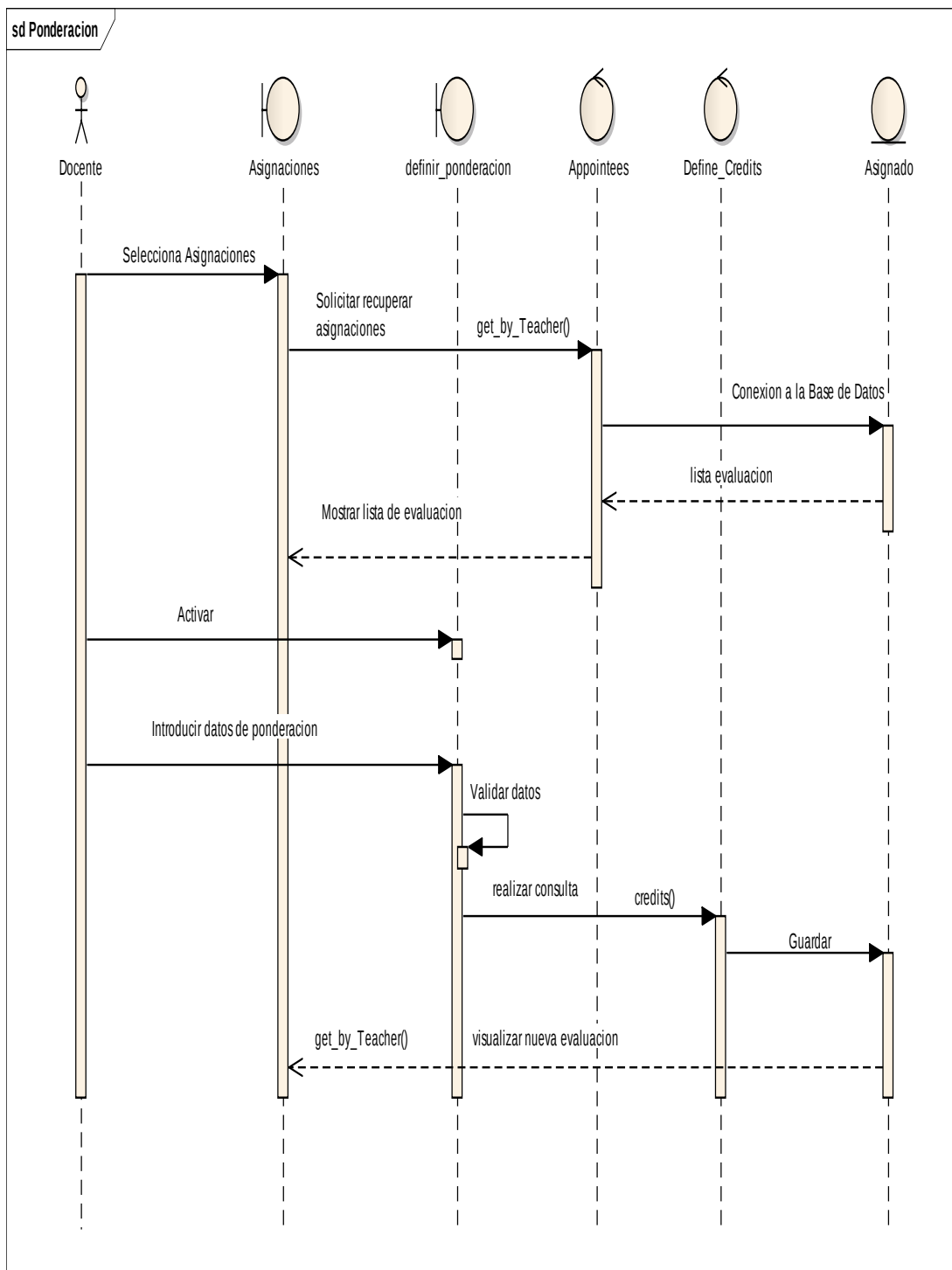


Figura 113. Diagramas De Secuencia Ponderación

2.1.7.5.15.24 Diagramas de Secuencia Gestión Materia

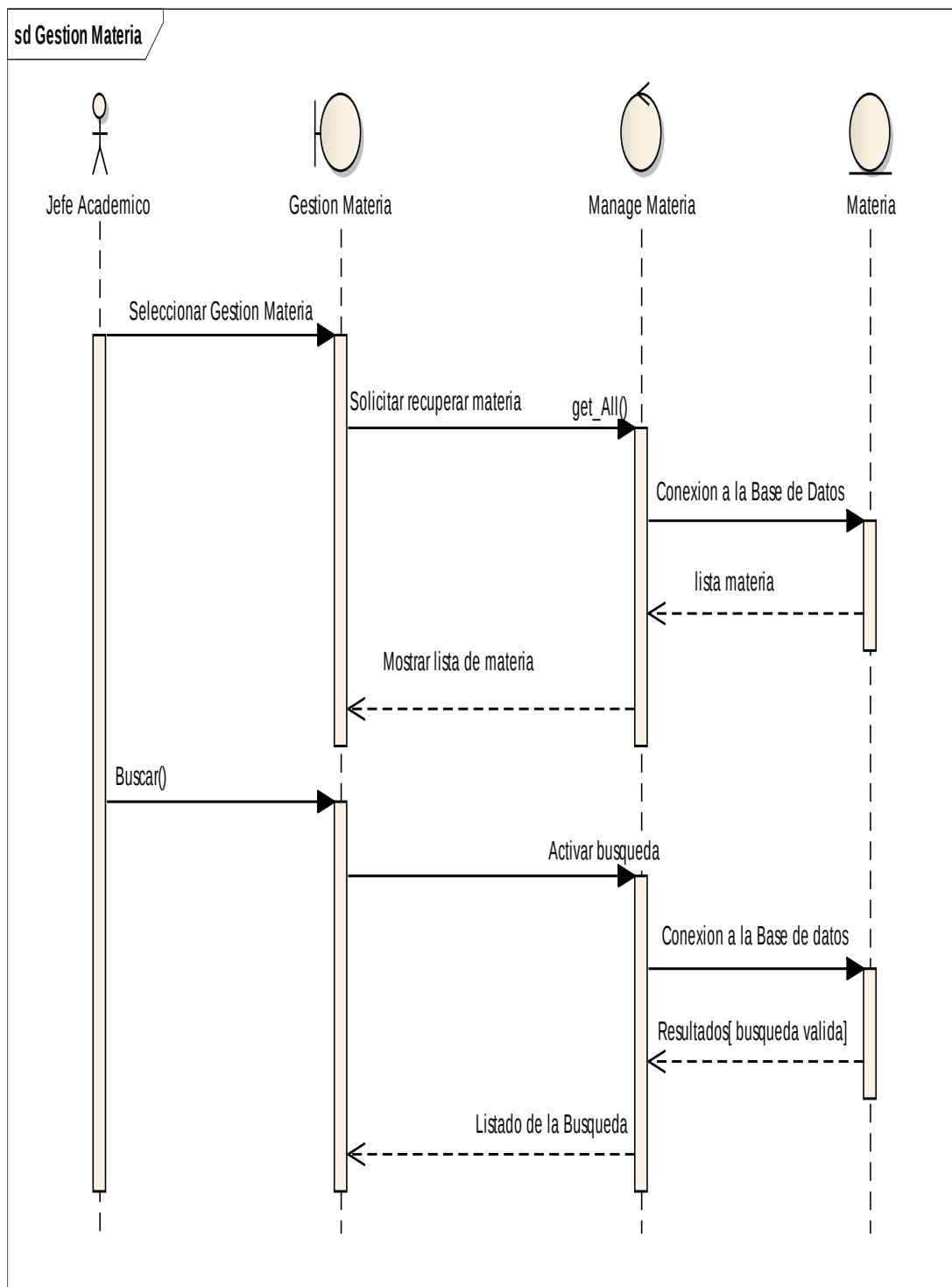


Figura 114. Diagramas De Secuencia Gestión Materia

2.1.7.5.15.25 Diagramas de Secuencia Adicionar Materia

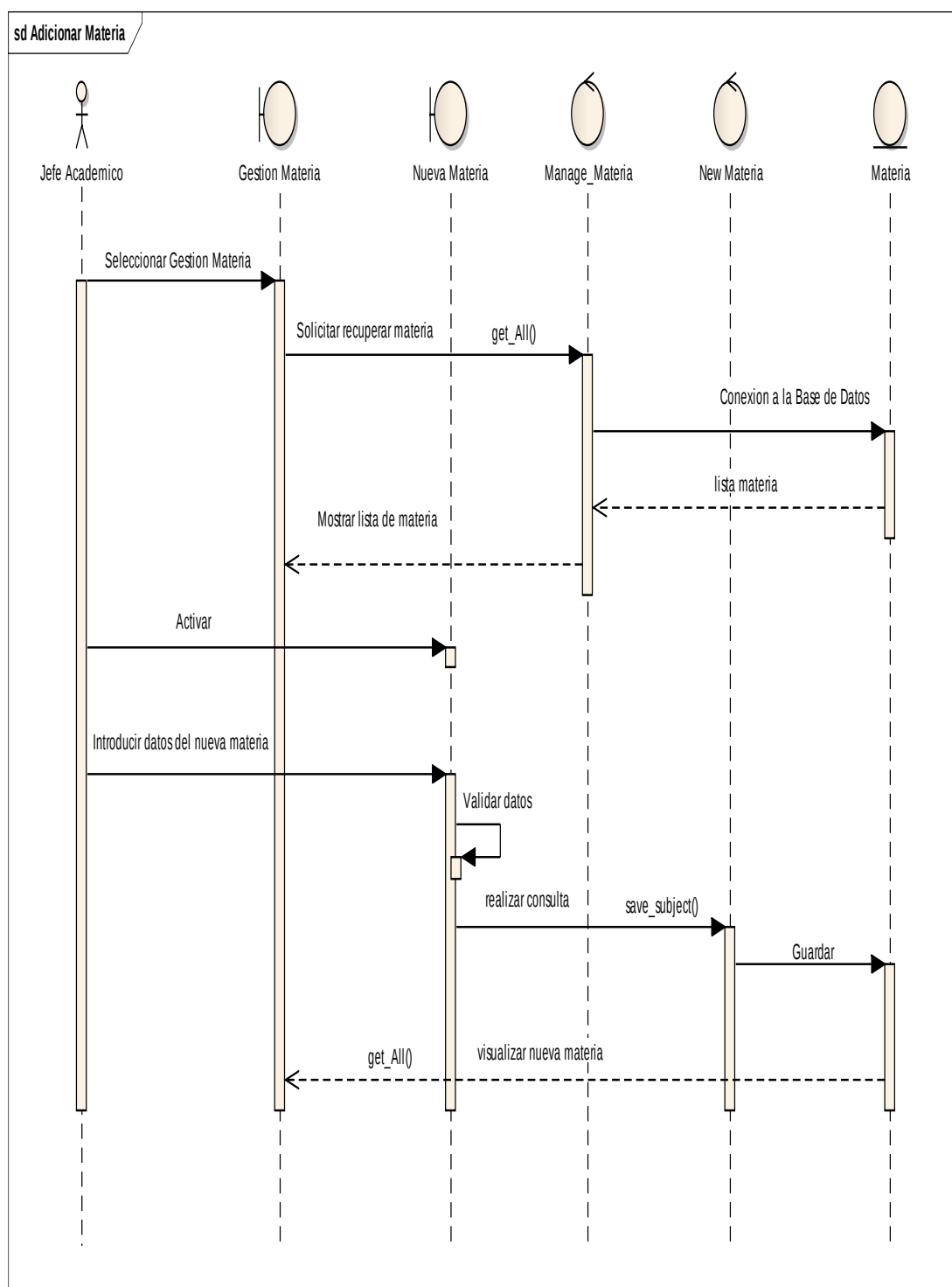


Figura 115. Diagramas De Secuencia Adicionar Materia

2.1.7.5.15.26 Diagramas de Secuencia Modificar Materia

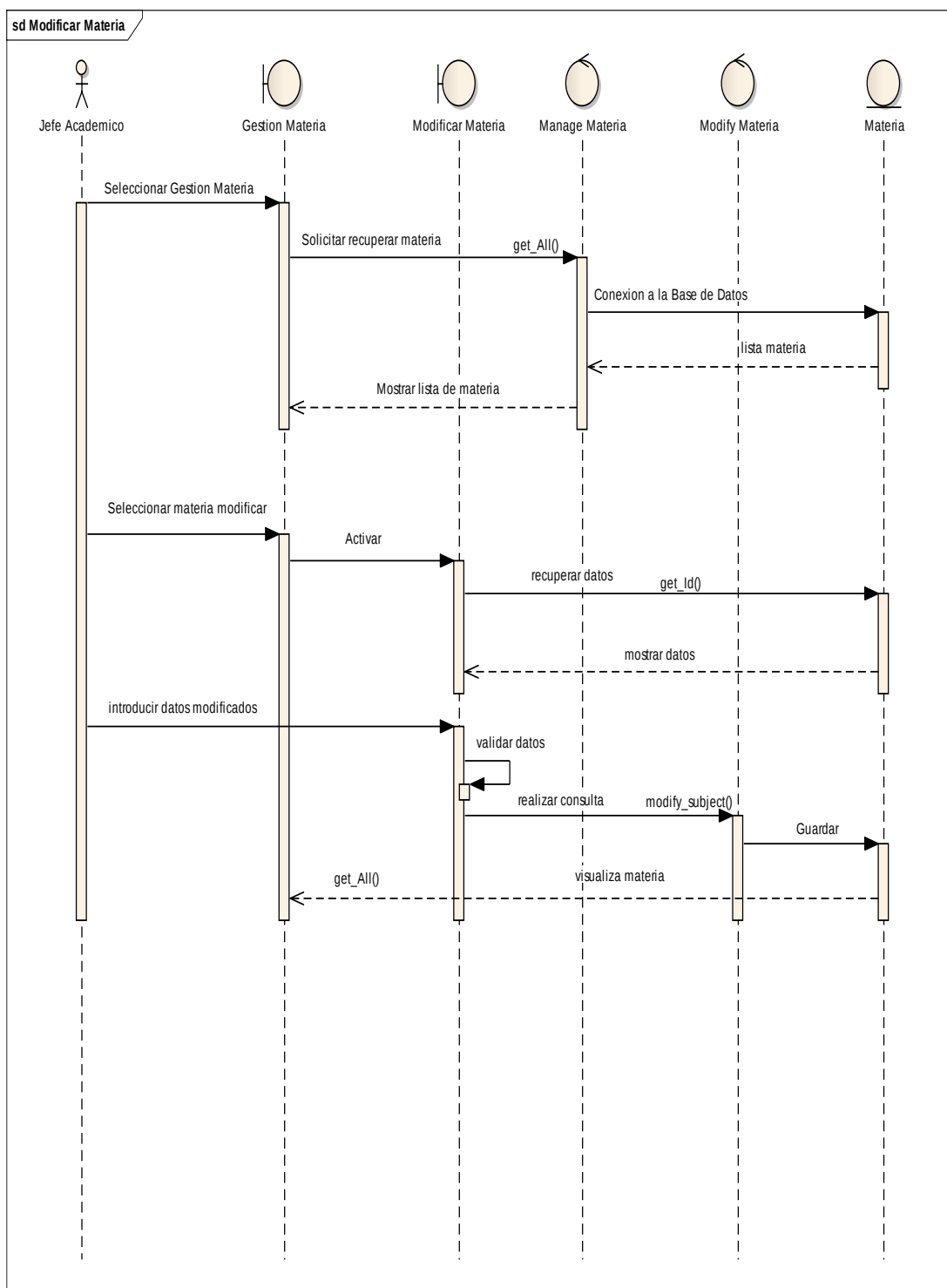


Figura 116. Diagramas De Secuencia Modificar Materia

2.1.7.5.15.27 Diagramas de Secuencia Eliminar Materia

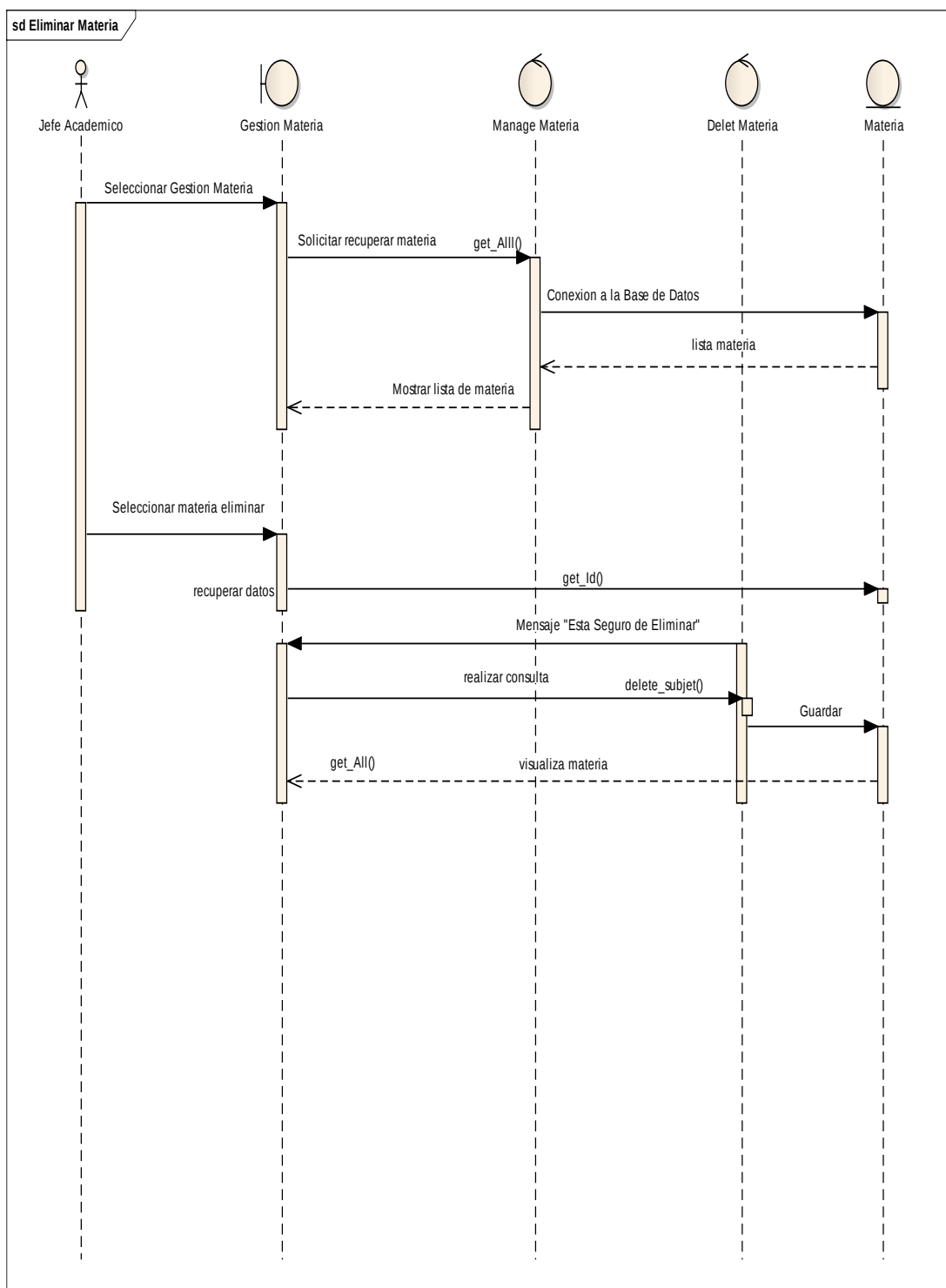


Figura 117. Diagramas De Secuencia Eliminar Materia

2.1.7.5.15.28 Diagramas de Secuencia Gestión Matricula

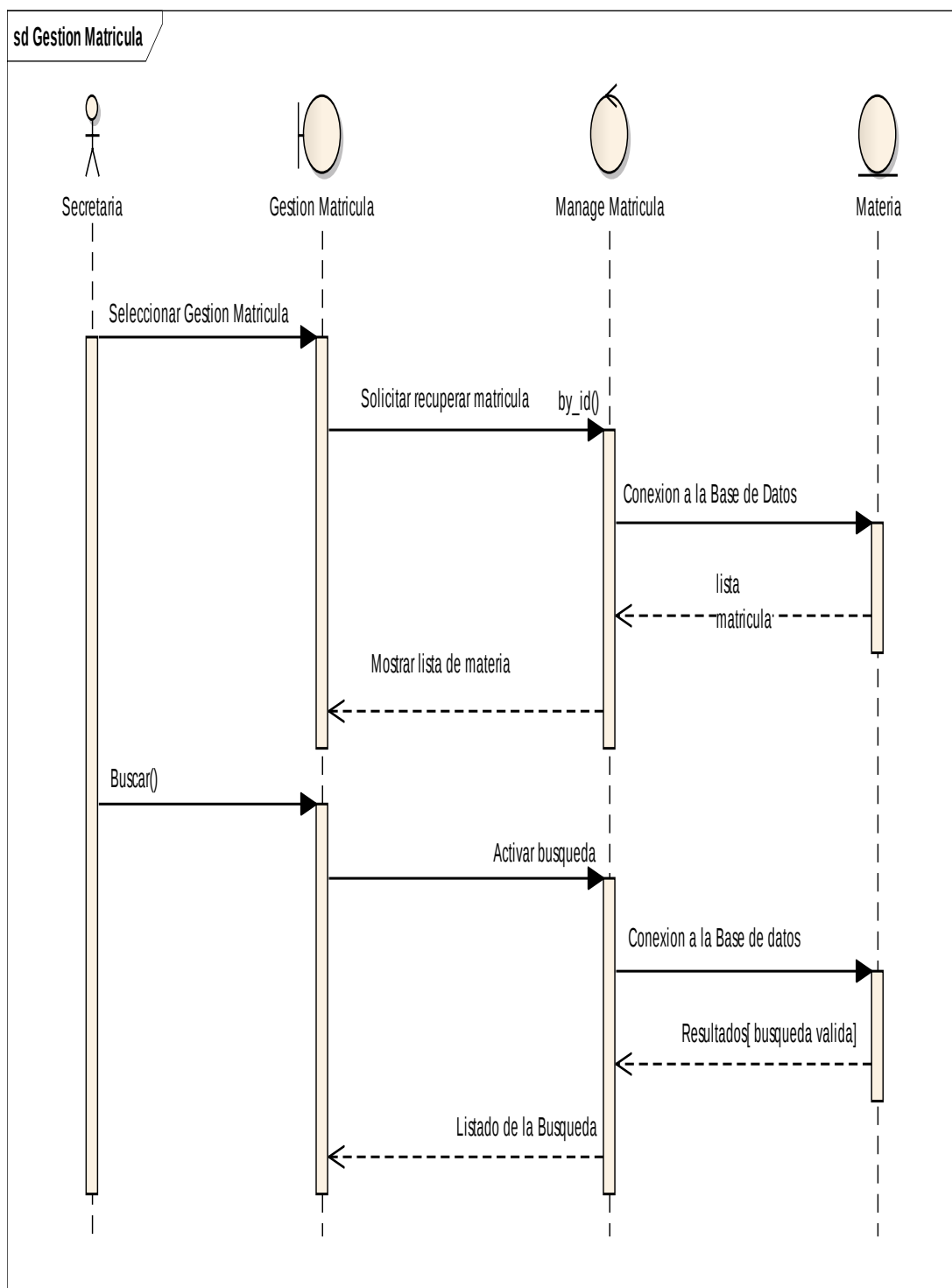


Figura 118. Diagramas De Secuencia Gestión Matricula

2.1.7.5.15.29 Diagramas de Secuencia Adicionar Matricula

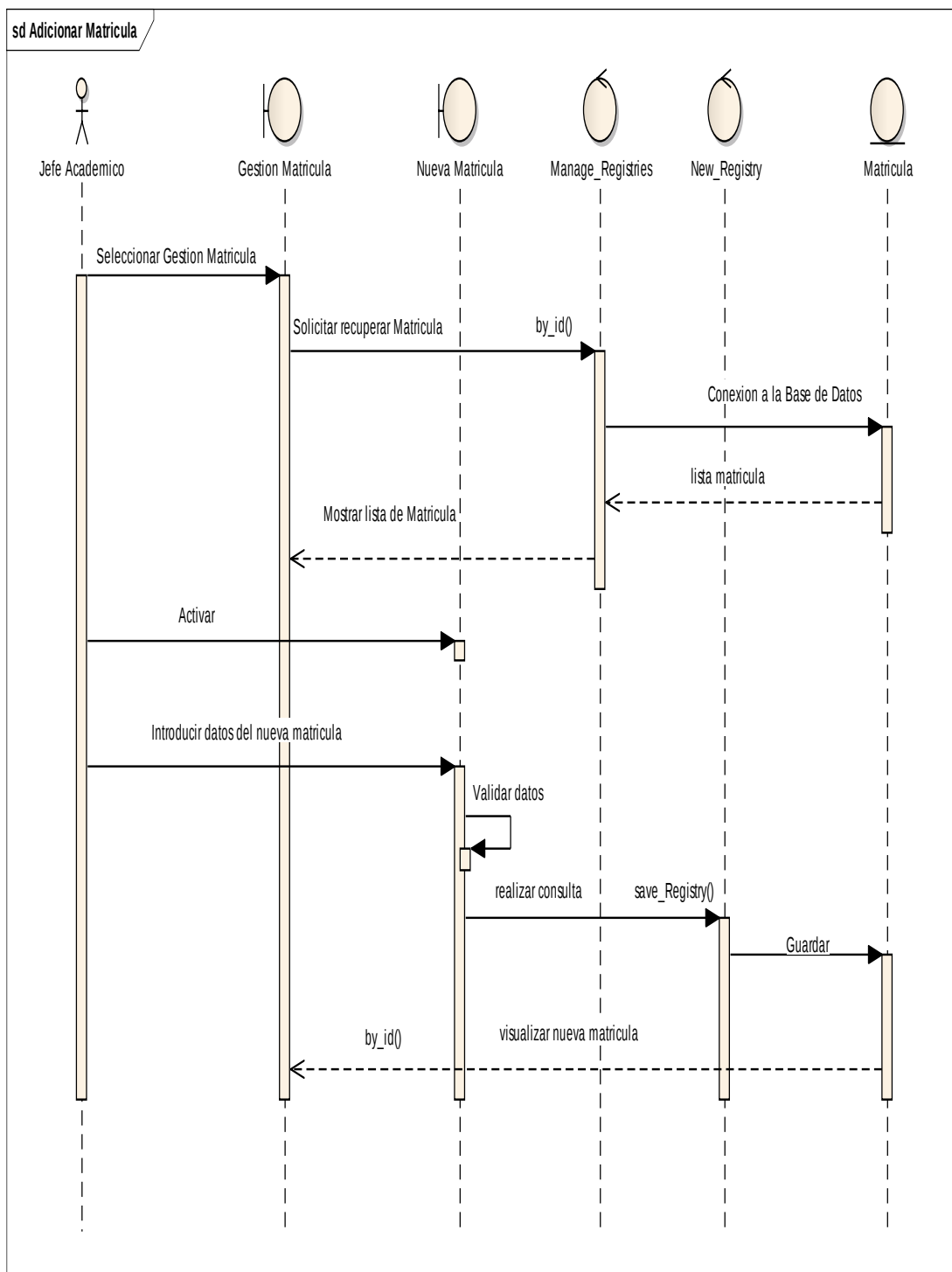


Figura 119. Diagramas De Secuencia Adicionar Matricula

2.1.7.5.15.30

Diagramas de Secuencia Modificar Matricula

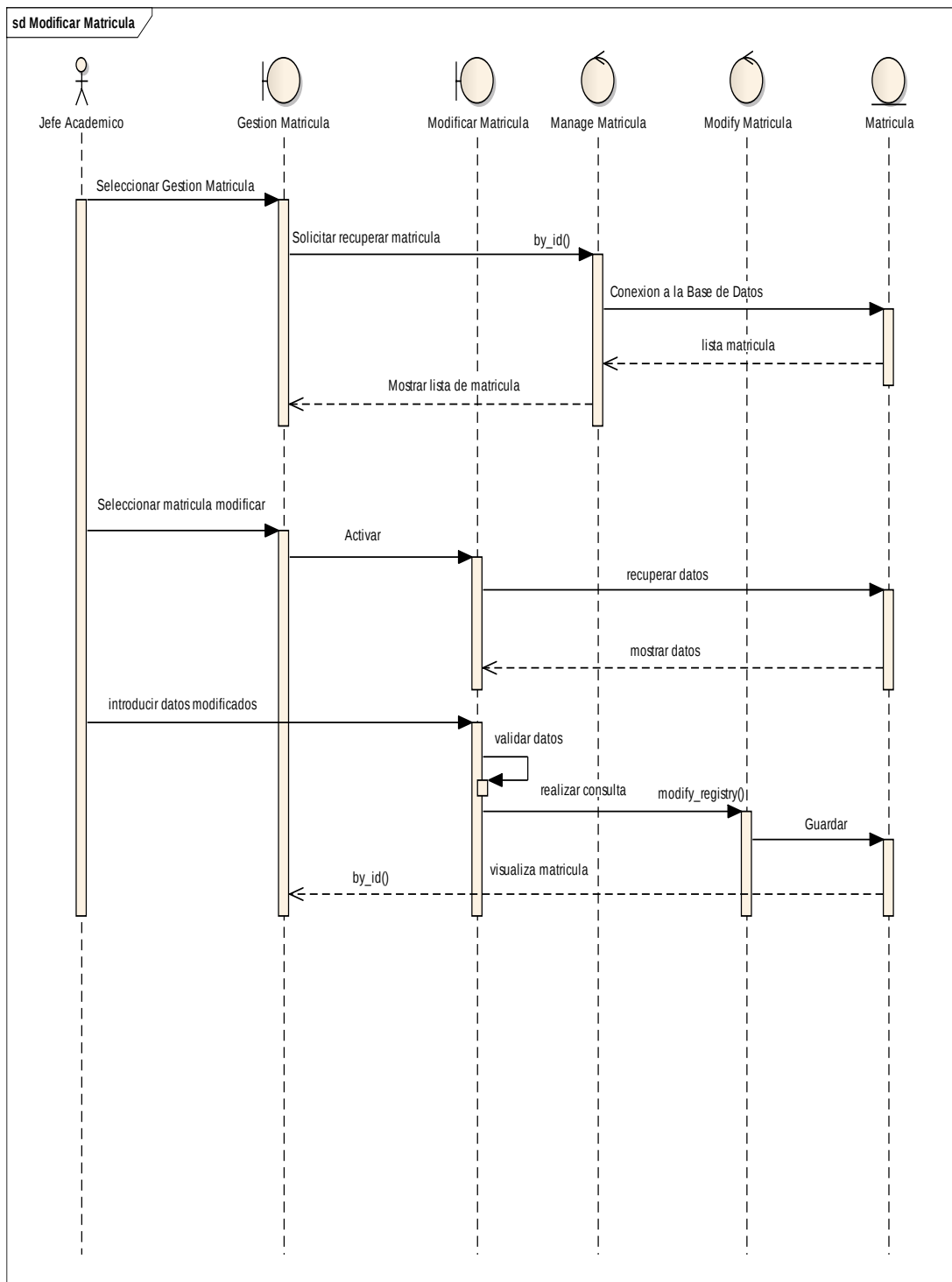


Figura 120. Diagramas De Secuencia Modificar Matricula

2.1.7.5.15.31 Diagramas de Secuencia Eliminar Matricula

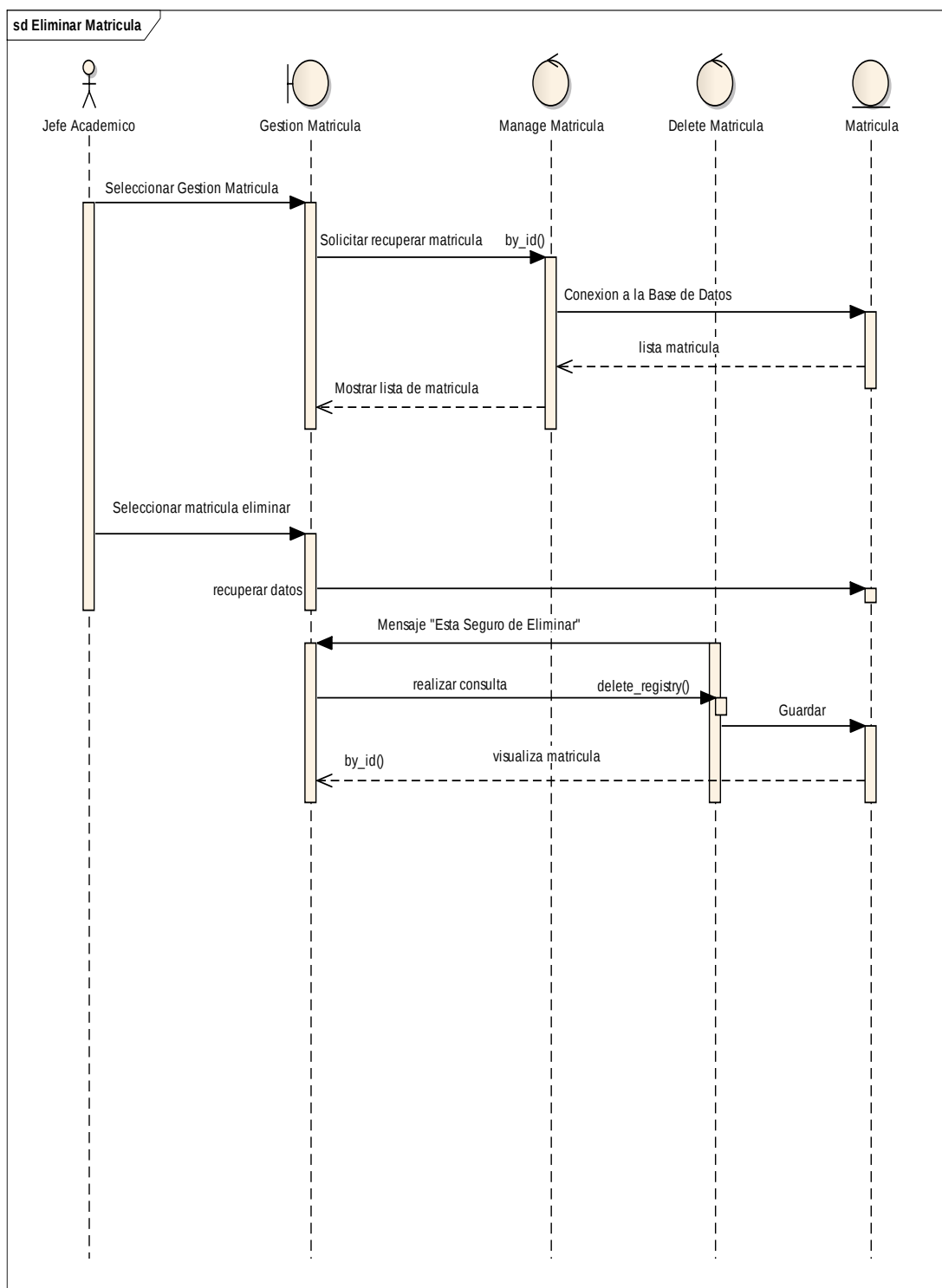


Figura 121. Diagramas De Secuencia Eliminar Matricula

2.1.7.6 Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación) de acuerdo al avance del proyecto.

2.1.7.6.1 Diagrama de Clases

2.1.7.6.2 Introducción

El diagrama de clases es el diagrama principal para el análisis y diseño. Un diagrama de clases representa las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos, y operaciones.

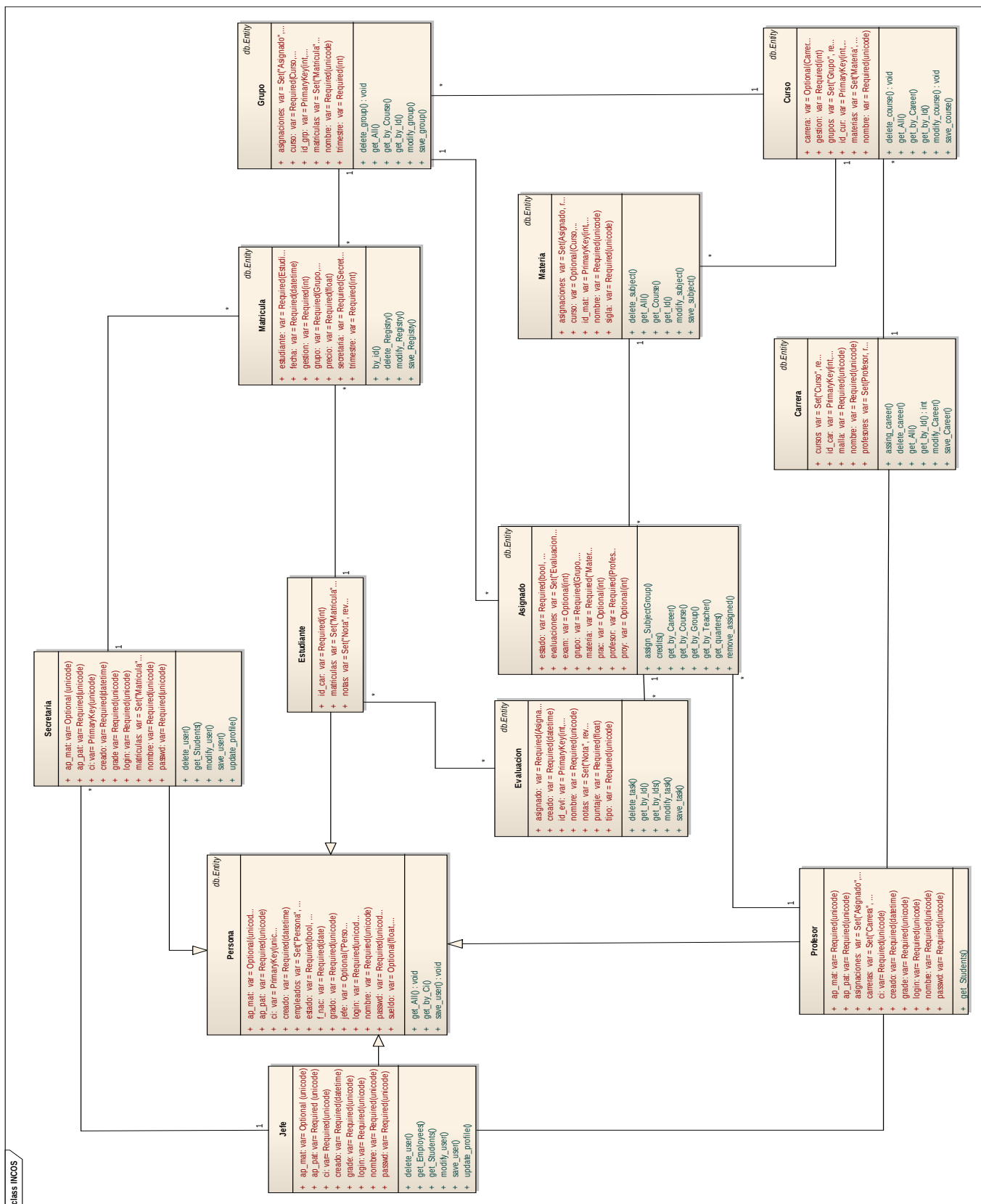
Cada clase se representa en un rectángulo con tres compartimientos:

- Nombre de la clase
- Atributos de la clase
- Operaciones de la clase

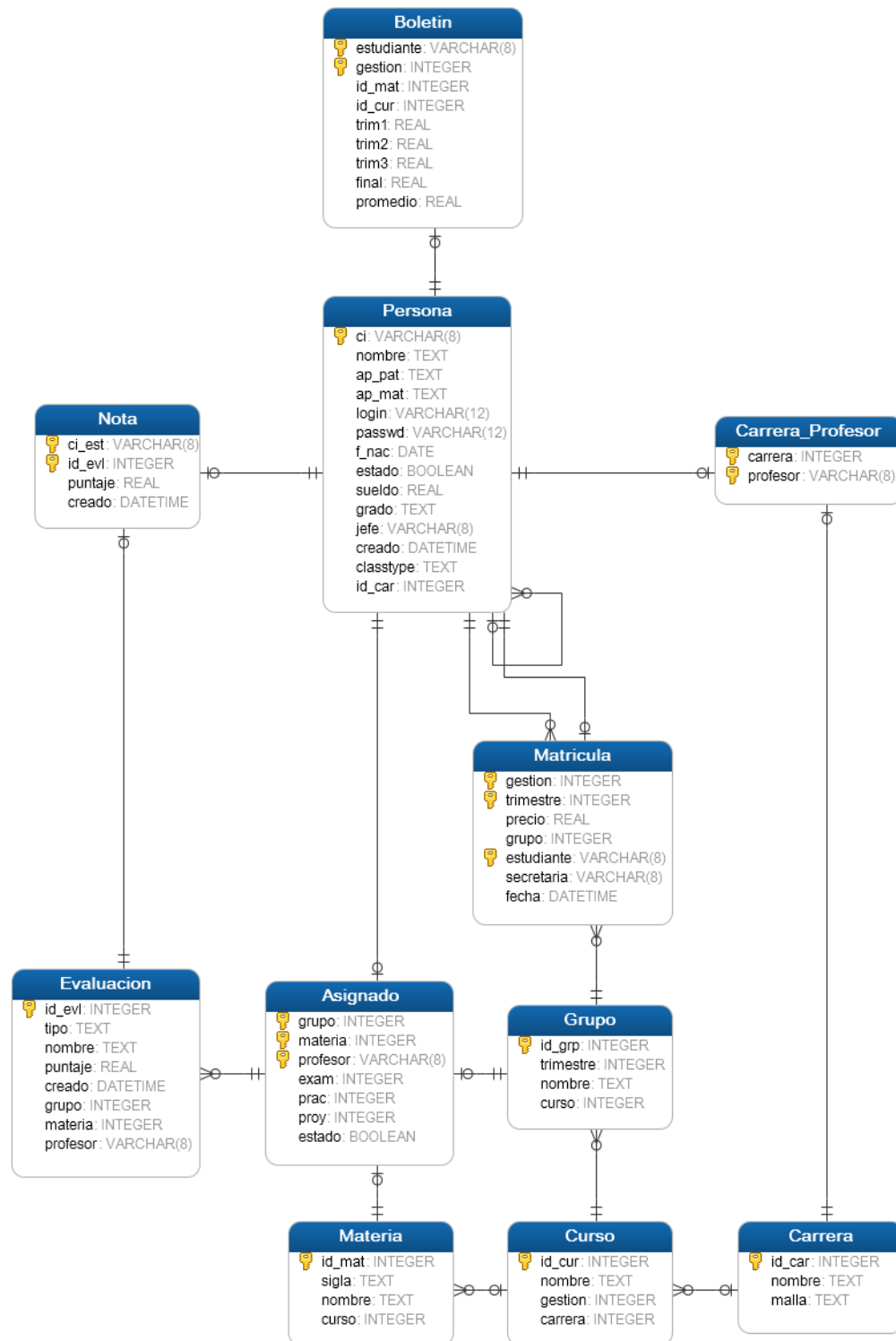
Los atributos de una clase no deberían ser manipulables directamente por el resto de objetos. Por esta razón niveles de visibilidad para los elementos que son:

- (-) Privados: Es el más fuerte. Esta parte es totalmente invisible.
- (#) Los atributos u operaciones protegidas estas visibles para las clases de herencia.
- (+) Los atributos u operaciones públicas son visibles desde otras clases y también por clases de herencia.

2.1.7.7 Diagrama de Clases



2.1.7.8 Diagrama Entidad Relación



2.1.7.9 Modelo de Datos.

2.1.7.9.1 Introducción

Base de Datos es el conjunto de datos almacenados con una estructura lógica. Es decir, tan importante como los datos, es la estructura conceptual con la que se relacionan entre ellos. En la práctica, podemos pensar esto como el conjunto de datos más los programas (o *software*) que hacen de ellos un conjunto consistente.

Si no tenemos los dos factores unidos, no podemos hablar de una base de datos, ya que ambos combinados dan la coherencia necesaria para poder trabajar con los datos de una manera sistemática.

2.1.7.9.2 Propósito

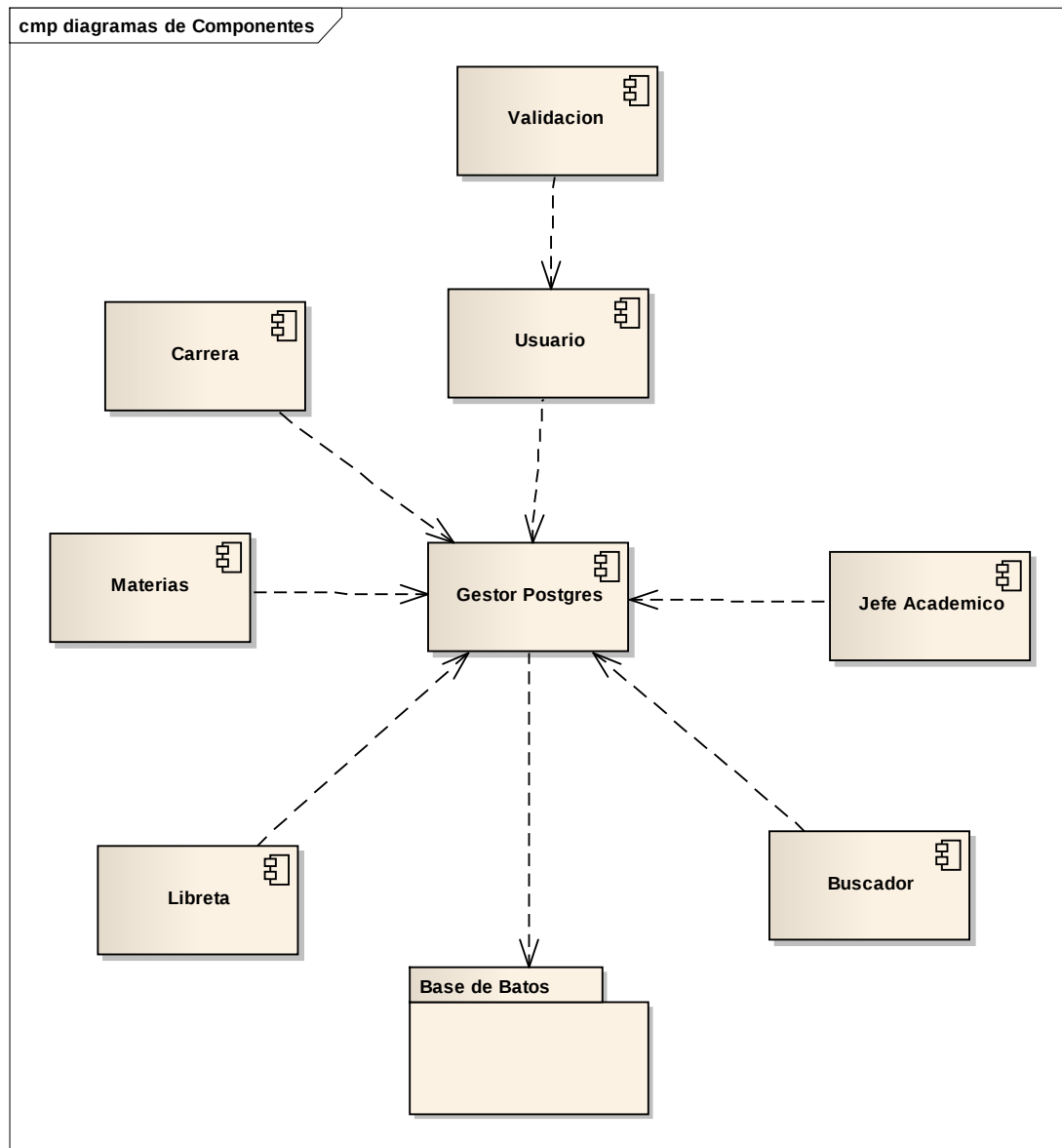
- Usar el lenguaje estándar.

2.1.7.9.3 Alcance

- Definir de la Base de Datos con el Lenguaje Estructurado

2.1.7.10 Modelo de Implementación

2.1.7.10.1 Diagrama de Componentes



2.1.7.11 Especificación De Tablas De La Base De Datos

2.1.7.11.1 Especificación de la Tabla: Persona

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
ci	character varying(8)	Si	Si		
nombre	text	Si	No		
ap_pat	text	Si	No		
ap_mat	text	No	No		
login	text	Si	No		
passwd	character varying(8)	Si	No		
f_nac	date	Si	No		
estado	boolean	No	No		
classtype	text	Si	No		
grado	text	No	No		
suelo	double precision	No	No		

Restricciones

Nombre	Tipo	Definición	Comentario
persona_pkey	Clave primaria	(ci)	
persona_login_key	Único	(login)	

2.1.7.11.2 Especificación de la Tabla: Asignado

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
grupo	integer	Si	Si		
materia	integer	Si	Si		
profesor	text	Si	Si		
exam	integer	No	No		
prac	integer	No	No		
proy	integer	No	No		
estado	bytea	Si	No		

Nombre	Tipo	Definición	Comentario
Asignado_pk ey	Clave primaria	(grupo, materia, profesor)	
fkey0	Clave ajena	(grupo) REFERENCES "Grupo" (id_grp) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fkey1	Clave ajena	(materia) REFERENCES "Materia" (id_mat) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fkey2	Clave ajena	(profesor) REFERENCES "Persona" (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.3 Especificación de la Tabla: Boletín

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
estudiante	text	Si	Si		
gestion	integer	Si	Si		
id_mat	integer	Si	No		
id_cur	integer	Si	No		
trim1	double precision	No	No		

trim2	double precision	No	No		
trim3	double precision	No	No		
final	double precision	No	No		
promedio	double precision	No	No		

Restricciones

Nombre	Tipo	Definición	Comentario
Boletin_pkey	Clave primaria	(estudiante, gestion)	
fk_Boletin_Persona_1	Clave ajena	(estudiante) REFERENCES "Persona" (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.4 Especificación de la Tabla: Matricula

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_mtl	integer	Si	Si	nextval('matricula_id_mtl_seq'::regclass)	
precio	double precision	Si	No		
fecha	timestamp without time zone	Si	No		
estudiante	character varying(8)	Si	No		
secretaria	character varying(8)	Si	No		
grupo	integer	Si	No		

Restricciones

Nombre	Tipo	Definición	Comentario
--------	------	------------	------------

matricula_pkey	Clave primaria	(id_mtl)	
fk_matricula_estudiante	Clave ajena	(estudiante) REFERENCES persona (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_matricula_grupo	Clave ajena	(grupo) REFERENCES grupo (id_grp) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_matricula_secretaria	Clave ajena	(secretaria) REFERENCES persona (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.5 Especificación de la Tabla: Materia Profesor

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
materia	integer	Si	Si		
profesor	character varying(8)	Si	Si		

Restricciones

Nombre	Tipo	Definición	Comentario
materia_profesor_pkey	Clave primaria	(materia, profesor)	
fk_materia_profesor_materia	Clave ajena	(materia) REFERENCES materia (id_mat) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_materia_profesor_profesor	Clave ajena	(profesor) REFERENCES persona (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.6 Especificación de la Tabla: Materia

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_mat	integer	Si	Si	nextval('materia_id_mat_seq'::regclass)	
nombre	text	Si	No		
sigla	text	Si	No		

Restricciones

Nombre	Tipo	Definición	Comentario
materia_pkey	Clave primaria	(id_mat)	

2.1.7.11.7 Especificación de la Tabla: Grupo_Materia

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
grupo	integer	Si	Si		
materia	integer	Si	Si		

Restricciones

Nombre	Tipo	Definición	Comentario
grupo_materia_pkey	Clave primaria	(grupo, materia)	
fk_grupo_materia_grupo	Clave ajena	(grupo) REFERENCES grupo (id_grp) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_grupo_materia_materia	Clave ajena	(materia) REFERENCES materia (id_mat) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.8 Especificación de la Tabla: Grupo

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_grp	integer	Si	Si	nextval('grupo_id_grp_seq'::regclass)	
nombre	text	Si	No		
turno	text	Si	No		
gestion	integer	Si	No		
curso	integer	Si	No		

Restricciones

Nombre	Tipo	Definición	Comentario
grupo_pk	Clave primaria	(id_grp)	
fk_grupo__curso	Clave ajena	(curso) REFERENCES curso (id_cur) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.9 Especificación de la Tabla: Evaluación

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_evl	integer	Si	Si	nextval('evaluacion_id_evl_seq'::regclass)	
tema	text	Si	No		
nombre	text	Si	No		
puntaje	double precision	Si	No		
materia	integer	Si	No		
profe	character	Si	No		

sor	varying(8)				
classtype	text	Si	No		
fe_ini	timestamp without time zone	No	No		
fe_fin	timestamp without time zone	No	No		
fecha	date	No	No		

Restricciones

Nombre	Tipo	Definición	Comentario
evaluacion_pkey	Clave primaria	(id_evl)	
fk_evaluacion_materia	Clave ajena	(materia) REFERENCES materia (id_mat) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_evaluacion_profesor	Clave ajena	(profesor) REFERENCES persona (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.10 Especificación de la Tabla: Curso

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_cur	integer	Si	Si	nextval('curso_id_cur_seq'::regclass)	
nombre	text	Si	No		
trimestre	integer	Si	No		
carrera	integer	Si	No		

Restricciones

Nombre	Tipo	Definición	Comentario
curso_pkey	Clave primaria	(id_cur)	

fk_curso__carrera	Clave ajena	(carrera) REFERENCES carrera (id_car) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
-------------------	-------------	--	--

2.1.7.11.11 Especificación de la Tabla: Carrera_profesor

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
carrera	integer	Si	Si		
profesor	character varying(8)	Si	Si		

Restricciones

Nombre	Tipo	Definición	Comentario
carrera_profesor_pkey	Clave primaria	(carrera, profesor)	
fk_carrera_profesor__carrera	Clave ajena	(carrera) REFERENCES carrera (id_car) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_carrera_profesor__profesor	Clave ajena	(profesor) REFERENCES persona (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.11.12 Especificación de la Tabla: Carrera

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
id_car	integer	Si	Si	nextval('carrera_id_car_seq'::regclass)	
nombre	text	Si	No		

Restricciones

Nombre	Tipo	Definición	Comentario
carrera_pkey	Clave primaria	(id_car)	

2.1.7.11.1 Especificación de la Tabla: Nota

Nombre	Tipo de Dato	¿No Nulo?	¿Clave Primaria?	Defecto	Comentario
ci_est	text	Si	Si		
id_evl	integer	Si	Si		
puntaje	double precision	Si	No		
creado	bytea	Si	No		

Restricciones

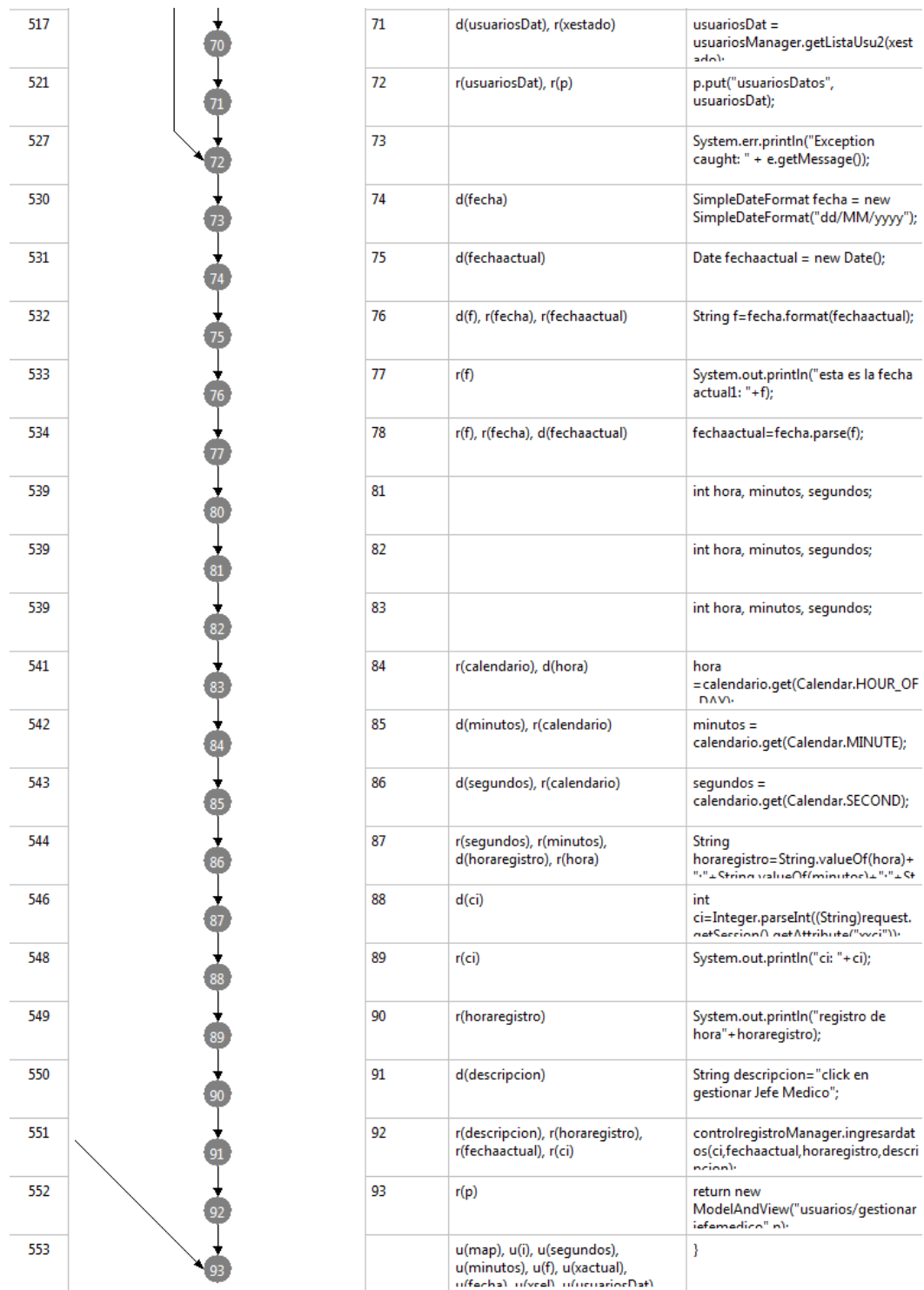
Nombre	Tipo	Definición	Comentario
Nota_pkey	Clave primaria	(ci_est, id_evl)	
fk_Nota_Evaluacion_1	Clave ajena	(id_evl) REFERENCES "Evaluacion" (id_evl) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	
fk_Nota_Persona_1	Clave ajena	(ci_est) REFERENCES "Persona" (ci) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION	

2.1.7.12 Casos de Prueba

2.1.7.12.1 Pruebas de Caja Blanca

2.1.7.12.1.1 Gestionar Usuarios

391	0	1	u(map), u(i), u(segundos), u(minutos), u(f), u(xactual), u(fecha), u(xcel), u(usuariosDat)	public ModelAndView gestionarfemedico(HttpServletRequestRe quest request, HttpServletResponse
391	1	2		public ModelAndView gestionarfemedico(HttpServletRequestRe quest request, HttpServletResponse
392	2	3	d(sesion)	HttpSession sesion= request.getSession(true);
396	4	5		System.out.println("entraaaaaaa");
397	5	6, 8		if((request.getSession().getAttribute ("xci")!=null)){ ...
399	7	93		return new ModelAndView("intro/iniciarsesion");
406	8	9		System.out.println("entra222222");
407	9	10	d(rol)	rol=Integer.parseInt(usuariosManag er.getRol(Integer.parseInt((String)re quest.getSession().getAttribute("vr
408	10	11	r(rol)	System.out.println("valor de rol: "+rol);
409	11	12, 14	r(rol)	if((rol==1)){
410	12	13	r(sesion)	System.out.println("ingresar:"+sesio n.getAttribute("ci"));
411	13	93		return new ModelAndView("intro/iniciarsesion");
422	14	15	d(p)	Map p = new HashMap();
424	15	16	d(xsel)	int xsel=0, cant=0, num=0, xestado=0;
424	16	17	d(cant)	int xsel=0, cant=0, num=0, xestado=0;
424	17	18	d(num)	int xsel=0, cant=0, num=0, xestado=0;
424	18	19	d(xestado)	int xsel=0, cant=0, num=0, xestado=0;
425	19	20	d(q)	String q, xinicio="0", xactual="1", xfiltro=null;
425	20	21	d(xinicio)	String q, xinicio="0", xactual="1", xfiltro=null;
425	21	22	d(xactual)	String q, xinicio="0", xactual="1", xfiltro=null;
425	22	23	d(xfiltro)	String q, xinicio="0", xactual="1", xfiltro=null;
426	23	24	d(usuariosDat)	List usuariosDat=null;
427	24	25	d(xcentro)	List xcentro=null;



2.1.7.12.1.2 Adicionar Usuarios

557	0	1	u(map), u(i), u(f), u(minutos), u(segundos), u(hora), u(fecha), u(fechaactual), u(calendar), u(n)	public ModelAndView addJefemedico(HttpServletRequest request, HttpServletResponse response)
557	1	2		public ModelAndView addJefemedico(HttpServletRequest request, HttpServletResponse response)
560	2	3	d(p)	Map p = new HashMap();
562	3	4	d(lugar)	List lugar = usuariosManager.getLugar();
563	4	5	r(p), r(lugar)	p.put("lugar", lugar);
564	5	6	d(roles)	List roles=rolesManager.getroles();
565	6	7	d(codr)	int codr=request.getParameter("codr") --null?Integer.parseInt(request.g
567	7	8	r(codr)	System.out.println("dddddd"+codr) ;
569	8	9	r(p), r(codr)	p.put("codr", codr);
572	9	10	d(xsel)	int xsel=0,cant=0,num=0,xestado=0;
572	10	11	d(cant)	int xsel=0,cant=0,num=0,xestado=0;
572	11	12	d(num)	int xsel=0,cant=0,num=0,xestado=0;
572	12	13	d(xestado)	int xsel=0,cant=0,num=0,xestado=0;
573	13	14	d(q)	String q, xinicio="0", xactual="1", xfiltro=null;
573	14	15	d(xinicio)	String q, xinicio="0", xactual="1", xfiltro=null;
573	15	16	d(xactual)	String q, xinicio="0", xactual="1", xfiltro=null;
573	16	17	d(xfiltro)	String q, xinicio="0", xactual="1", xfiltro=null;
574	17	18	d(usuariosDat)	List usuariosDat= null;
575	18	19	d(xcentro)	List xcentro= null;
578	19	20	d(zactual)	String zactual=request.getParameter("zact ual");
579	20	21	r(zactual)	System.out.println("q pasa por aquiii betmar: "+zactual);
580	21	22	d(xfiltro)	xfiltro=request.getParameter("xfiltro ");
582	22	23, 24	r(xfiltro)	if (xfiltro== null){
583	23	26	d(xfiltro)	xfiltro="";
587	24	25	d(xfiltro), r(xfiltro)	xfiltro=xfiltro;

588	25		
590	26		
591	27		
593	28		
594	29		
596	30		
597	31		
598	32		
600	33		
601	34		
602	35		
604	36		
610	37		
611	38		
616	39		
617	40		
619	41		
623	42		
624	43		
626	44		
627	45		
629	46		
632	47		
632	48		
632	49		

26	r(xfiltro)	request.getSession().setAttribute("s_filtro", xfiltro);
27, 28	r(zactual)	if (zactual == null){
37	d(xactual)	xactual="1";
29	d(xactual), r(zactual)	xactual=zactual;
30	r(zactual)	System.out.println("q pasa por aqui mario: "+zactual);
31	r(zactual), d(xcant)	int xcant= (Integer.parseInt(zactual) * 10) - 10;
32	r(xcant)	System.out.println("q pasa por aqui mario: "+xcant);
33	d(xinicio), r(xcant)	xinicio = String.valueOf(xcant);
34	d(cdd)	String cdd=(String) request.getSession().getAttribute("s_filtro");
35, 36	r(cdd)	if (cdd == null){
37	d(xfiltro)	xfiltro="";
37	d(xfiltro), r(cdd)	xfiltro=cdd;
38, 39	r(xfiltro), r(zactual)	if ((xfiltro== null)&(zactual == null)){
39		request.getSession().removeAttribute("s_filtro");
40	d(lista)	ArrayList lista= new ArrayList();
41	d(lista2)	ArrayList lista2= new ArrayList();
42	d(usuariosDat), r(xfiltro), r(xinicio), r(codr)	usuariosDat = usuariosManager.getListUsuariosro (xfiltro, inicio, codr);
43	d(xcant)	int xcant=0;
44	d(xcant), r(xfiltro), r(codr)	xcant = usuariosManager.getcantUsuariosro (xfiltro, codr);
45, 46	r(xcant)	if ((xcant % 10)!=0){
47	d(xcant), r(xcant)	xcant = xcant / 10;
47	d(xcant), r(xcant)	xcant = (xcant / 10)+1;
48	d(i)	for (int i=1; i<=xcant; i++){
50, 53	r(i), r(xcant)	for (int i=1; i<=xcant; i++){
48	d(i)	for (int i=1; i<=xcant; i++){

633			
634			
635			
638			
644			
645			
646			
648			
649			
666			
669			
678			
681			
682			
683			
684			
685			
686			
689			
690			
690			
690			
692			
693			
694			

```

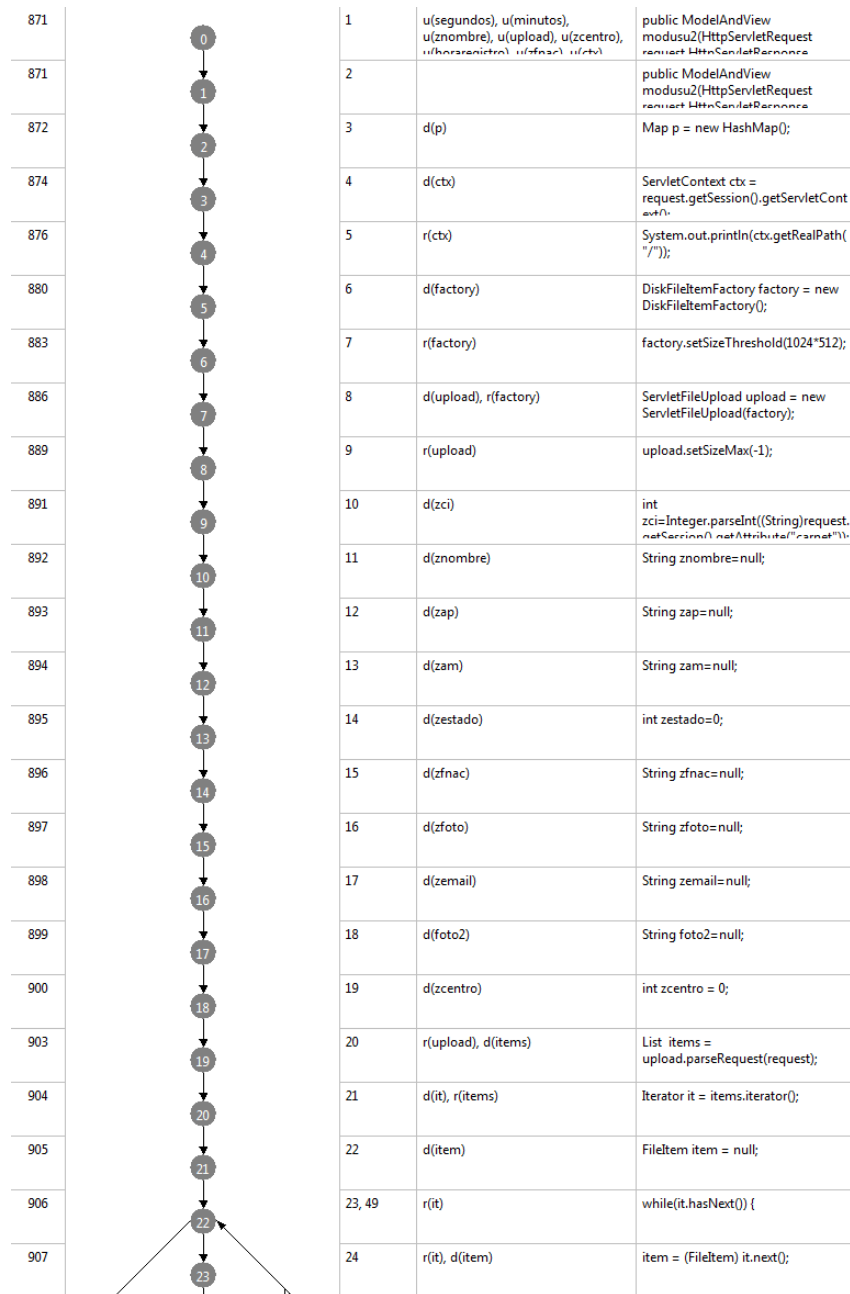
graph TD
    50((50)) -- loop --> 50
    50 --> 51((51))
    51 --> 52((52))
    52 --> 53((53))
    53 --> 54((54))
    54 --> 55((55))
    55 --> 56((56))
    56 --> 57((57))
    57 --> 58((58))
    58 --> 59((59))
    59 --> 60((60))
    60 --> 61((61))
    61 --> 62((62))
    62 --> 63((63))
    63 --> 64((64))
    64 --> 65((65))
    65 --> 66((66))
    66 --> 67((67))
    67 --> 68((68))
    68 --> 69((69))
    69 --> 70((70))
    70 --> 71((71))
    71 --> 72((72))
    72 --> 73((73))
    73 --> 74((74))
    50 --> 53
    53 --> 50

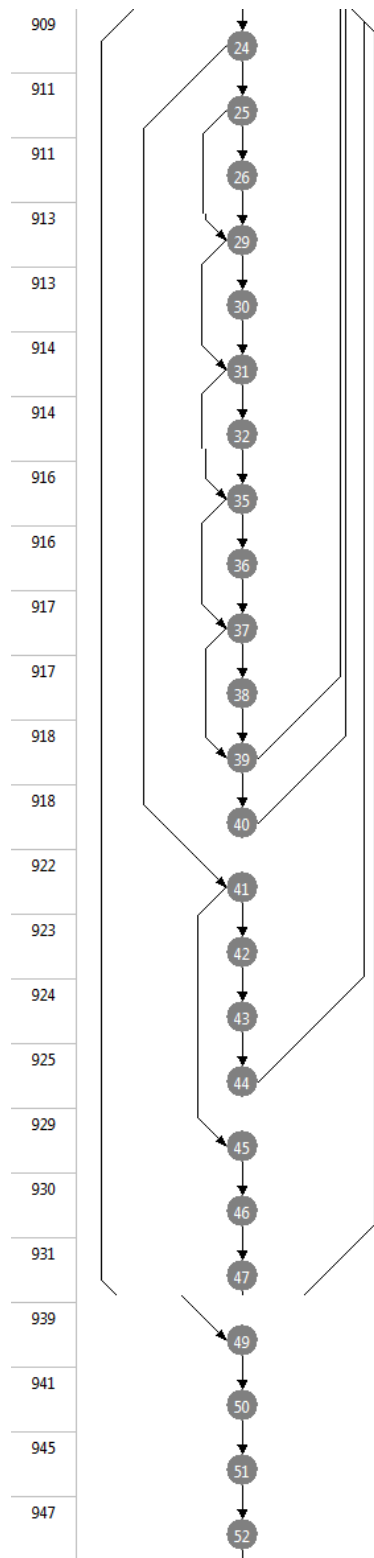
```

51	d(map)	Map map = new HashMap();
52	r(map), r(i)	map.put("xcont", String.valueOf(i));
49	r(map), r(lista2)	lista2.add(map);
54	d(xdatos)	List xdatos= datosManager.getDatos();
55	r(p), r(xfiltro)	p.put("xfiltro", xfiltro);
56	r(p), r(xactual)	p.put("xactual", xactual);
57	r(p), r(lista2)	p.put("xcantpaginas", lista2);
58	r(p), r(usuariosDat)	p.put("usuariosDatos", usuariosDat);
59	r(p), r(xdatos)	p.put("datos", xdatos);
60		System.err.println("Exception caught: " + e.getMessage());
61	r(p), r(roles)	p.put("roles", roles);
62		e.printStackTrace();
63	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
64	d(fechaactual)	Date fechaactual = new Date();
65	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
66	r(f)	System.out.println("esta es la fecha actual1: "+f);
67	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
68	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
69	d(calendario)	Calendar calendario = new GregorianCalendar();
70		int hora, minutos, segundos;
71		int hora, minutos, segundos;
72		int hora, minutos, segundos;
73	d(hora), r(calendario)	hora = calendario.get(Calendar.HOUR_OF_DAY);
74	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
75	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);

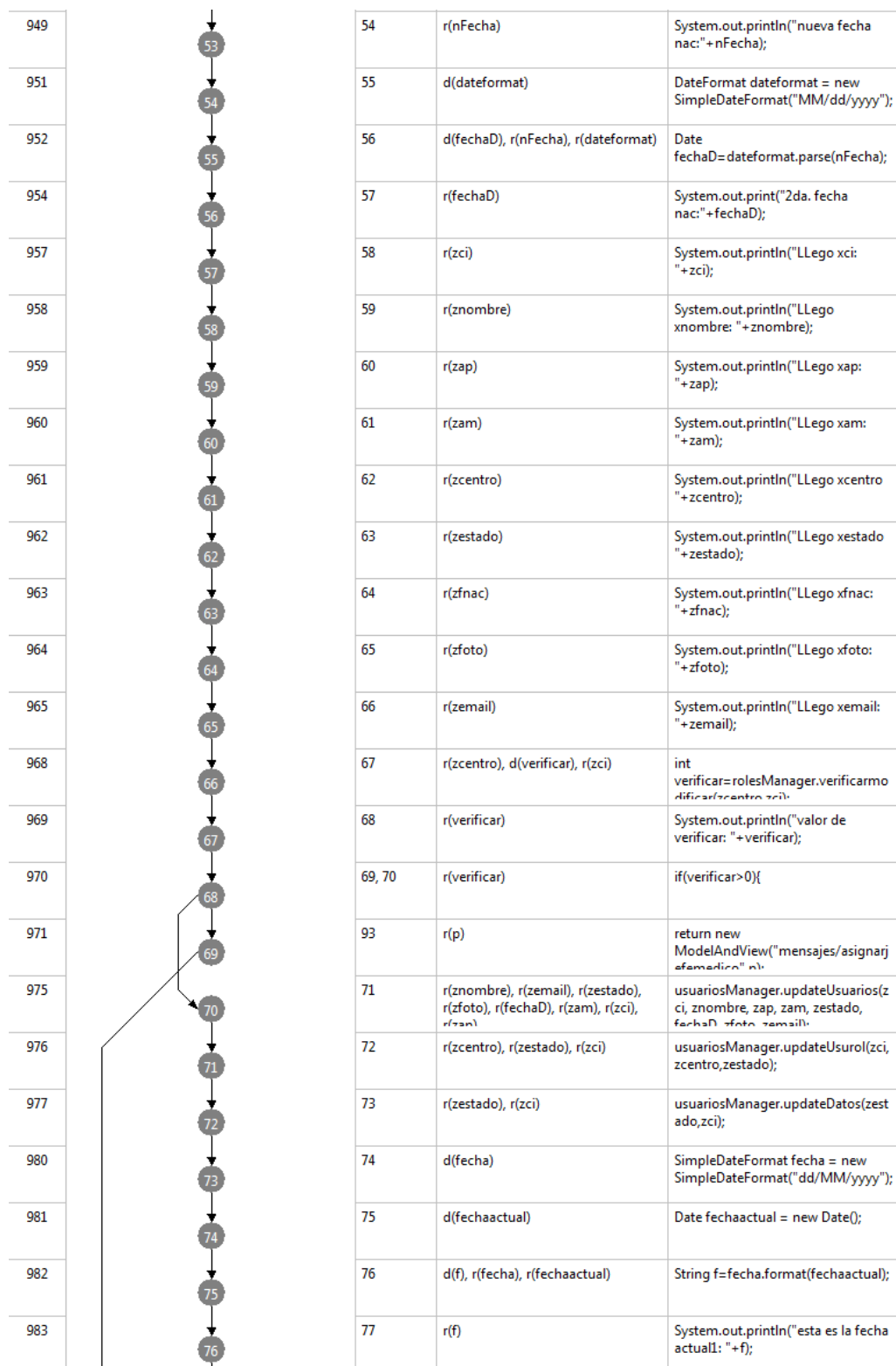
693	73	74	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
694	74	75	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
695	75	76	r(minutos), r(segundos), r(hora), d(horaregistro)	String horaregistro=String.valueOf(hora)+ "+"+String.valueOf(minutos)+" "+St
697	76	77	d(ci)	int ci=Integer.parseInt((String)request. getSession().getAttribute("vci"));
699	77	78	r(ci)	System.out.println("ci: "+ci);
700	78	79	r(horaregistro)	System.out.println("registro de hora"+ horaregistro);
701	79	80	d(descripcion)	String descripcion="click en agregar jefe medico";
702	80	81	r(fechaactual), r(ci), r(descripcion), r(horaregistro)	controlRegistroManager.ingresardat os(ci,fechaactual,horaregistro,descri pcion);
710	81	82	r(p)	return new ModelAndView("usuarios/addjefem edico");
711	82		u(map), u(i), u(f), u(minutos), u(segundos), u(hora), u(fecha), u(fechaactual), u(calendario), u(n)	}

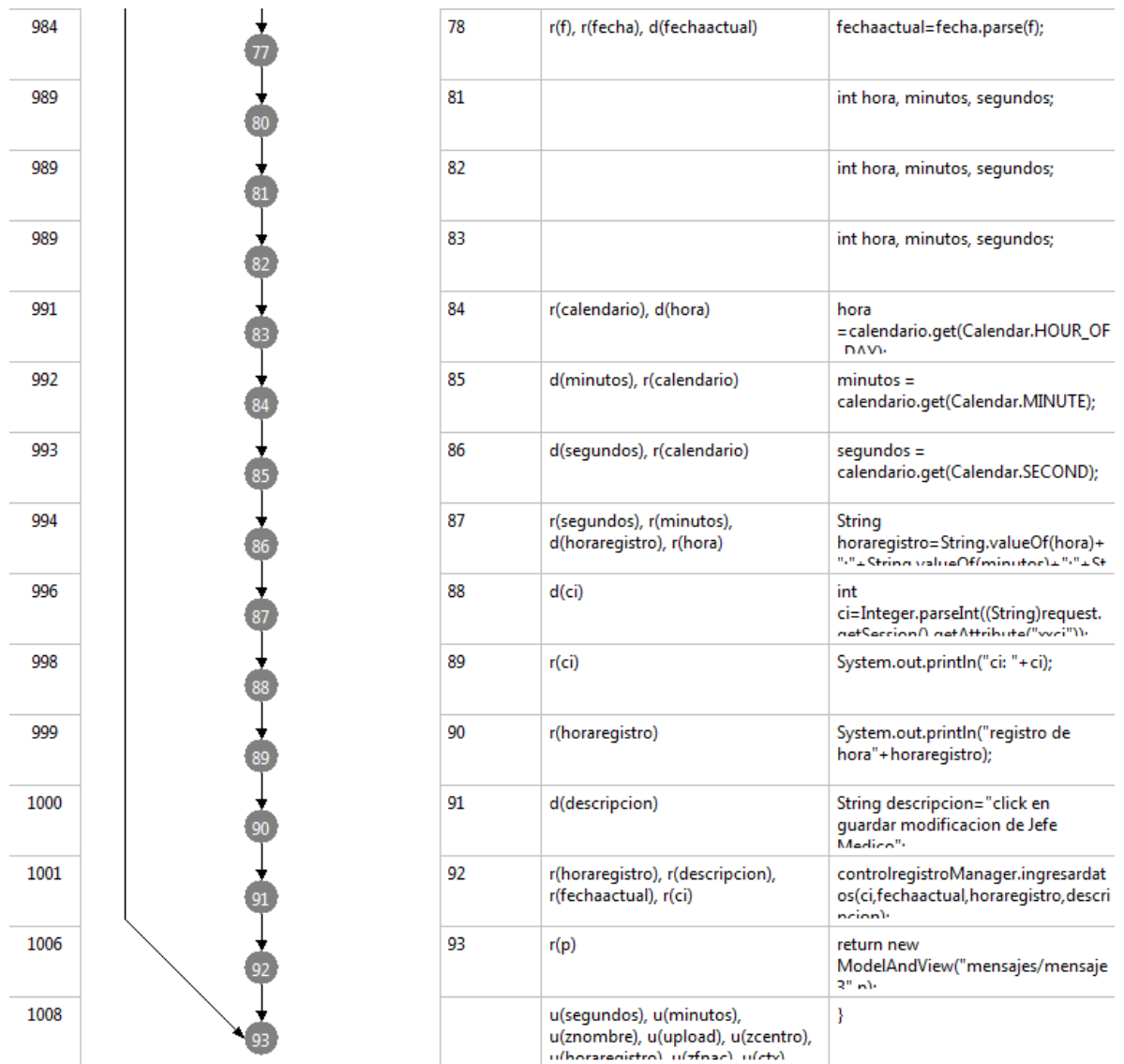
2.1.7.12.1.3 Modificar Usuario





25, 41	r(item)	if(item.isFormField()){
26, 27	r(item)	if (item.getFieldName().equals("xnom bre")){
27	d(znombre), r(item)	if (item.getFieldName().equals("xnom bre")){
30, 31	r(item)	if (item.getFieldName().equals("xam")){
31	d(zam), r(item)	if (item.getFieldName().equals("xam")){
32, 33	r(item)	if (item.getFieldName().equals("xcent ro")){
33	d(zcentro), r(item)	if (item.getFieldName().equals("xcent ro")){
36, 37	r(item)	if (item.getFieldName().equals("xfnac ")){
37	d(zfnac), r(item)	if (item.getFieldName().equals("xfnac ")){
38, 39	r(item)	if (item.getFieldName().equals("xemai l")){
39	d(zemail), r(item)	if (item.getFieldName().equals("xemai l")){
40, 22	r(item)	if (item.getFieldName().equals("foto2 ")){
22	d(foto2), r(item)	if (item.getFieldName().equals("foto2 ")){
42, 45	r(item)	if(item.getName().equals("")){
43	r(foto2)	System.out.println("entro al if de la foto : "+foto2);
44	r(foto2), d(zfoto)	zfoto=foto2;
22	r(zfoto)	System.out.println("zfoto if: "+zfoto);
46	r(ctb), d(file), r(item)	File file = new File(ctb.getRealPath("/")+"uploads/ "+ item.getName());
47	r(file), r(item)	item.write(file);
48	d(zfoto), r(item)	zfoto=item.getName();
50		e.printStackTrace();
51		e.printStackTrace();
52	r(zfnac)	System.out.println("fecha nac: "+zfnac);
53	r(zfnac), r(zfnac), r(zfnac), d(nFecha)	String nFecha=zfnac.substring(5,7)+"-"+zf nac.substring(9,10)+"-"+zfnac.subst

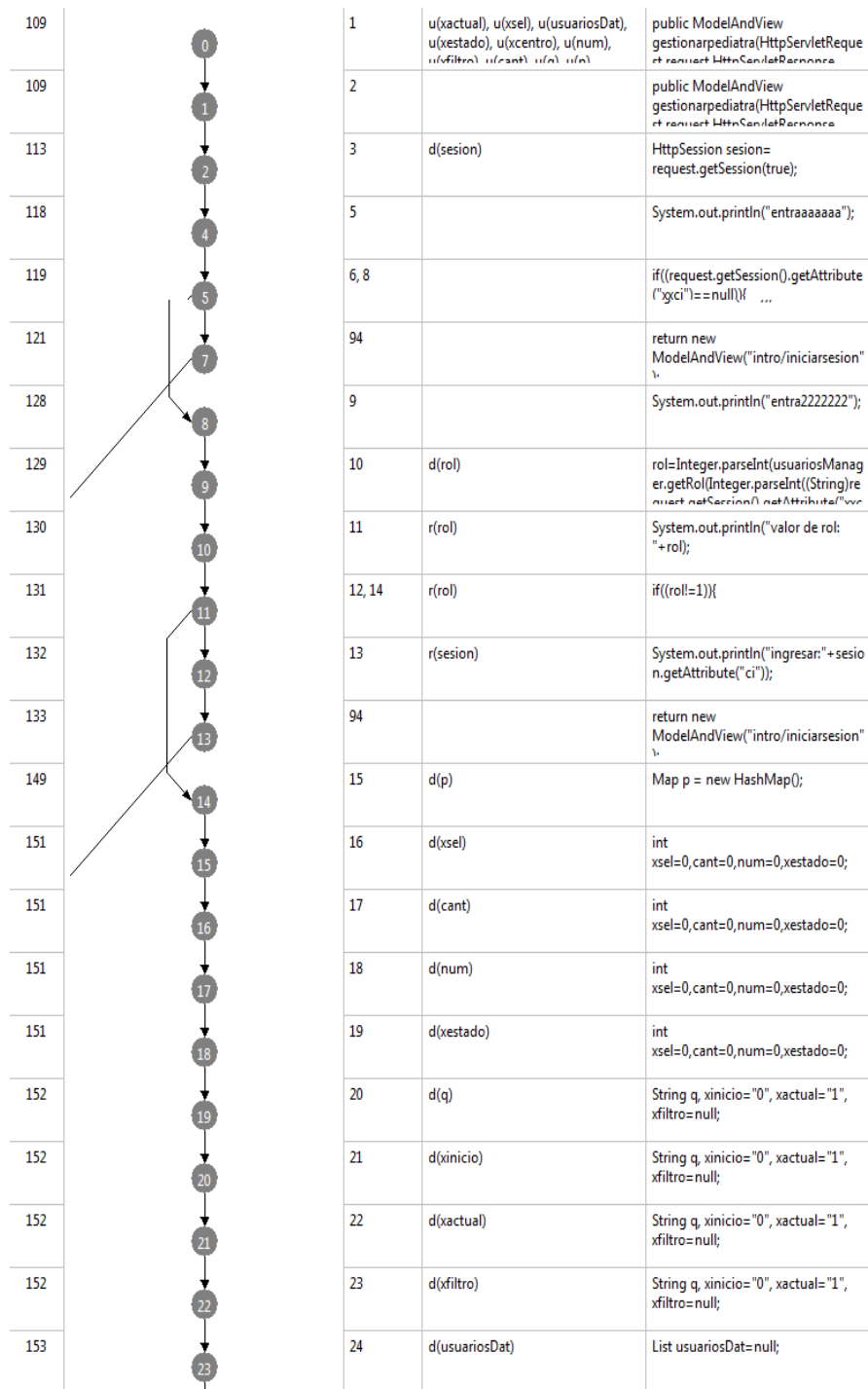


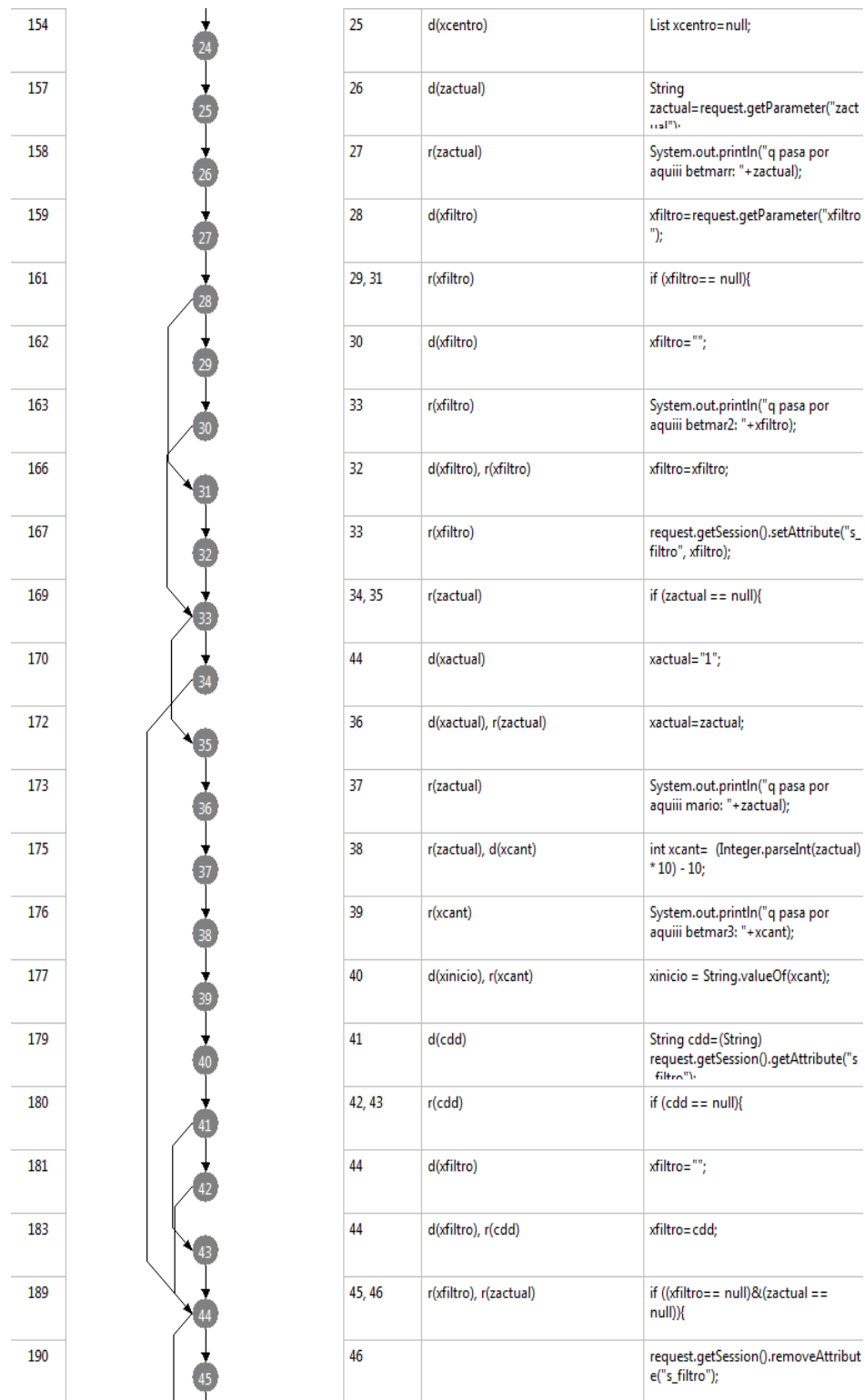


2.1.7.12.1.4 Eliminar Usuario

1193	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	public ModelAndView verusuario(HttpServletRequest request, HttpServletResponse response)
1193	1	2		public ModelAndView verusuario(HttpServletRequest request, HttpServletResponse response)
1194	2	3	d(p)	Map p = new HashMap();
1196	3	4	d(xci)	int xci= Integer.parseInt((request.getParameter("xcid")));
1197	4	5	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getUsuario(xci);
1199	5	6	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
1201	6	7	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
1202	7	8	d(fechaactual)	Date fechaactual = new Date();
1203	8	9	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
1204	9	10	r(f)	System.out.println("esta es la fecha actual: "+f);
1205	10	11	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
1206	11	12	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
1209	12	13	d(calendario)	Calendar calendario = new GregorianCalendar();
1210	13	14		int hora, minutos, segundos;
1210	14	15		int hora, minutos, segundos;
1210	15	16		int hora, minutos, segundos;
1212	16	17	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF_DAY);
1213	17	18	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
1214	18	19	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
1215	19	20	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+"_"+String.valueOf(minutos)+"_"+String.valueOf(segundos);
1217	20	21	d(ci)	int ci=Integer.parseInt((String)request.getAttribute("ci"));
1219	21	22	r(ci)	System.out.println("ci: "+ci);
1220	22	23	r(horaregistro)	System.out.println("registro de hora "+horaregistro);
1221	23	24	d(descripcion)	String descripcion="click ingreso a ver datos";
1222	24	25	r(descripcion), r(horaregistro), r(fechaactual), r(ci)	controlregistroManager.ingresardatos(ci,fechaactual,horaregistro,descripcion);
1228	25	26	r(p)	return new ModelAndView("usuarios/verusuario");
1229	26		u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	}

2.1.7.12.1.5 Gestionar Estudiante





195		46	47	d(lista)	ArrayList lista= new ArrayList();
196		47	48	d(lista2)	ArrayList lista2= new ArrayList();
203		48	49	d(usuariosDat), r(xfiltro), r(xinicio)	usuariosDat = usuariosManager.getListaPediatria(xfiltro, xinicio);
207		49	50	d(xcant)	int xcant=0;
208		50	51	r(xfiltro), d(xcant)	xcant = usuariosManager.getcantPediatria(xfiltro, xcant);
211		52	54	d(xcant), r(xcant)	xcant = xcant / 10;
213		53	54	d(xcant), r(xcant)	xcant = (xcant / 10)+1;
216		54	55	d(i)	for (int i=1; i<=xcant; i++){
216		55	57, 60	r(i), r(xcant)	for (int i=1; i<=xcant; i++){
216		56	55	d(i)	for (int i=1; i<=xcant; i++){
217		57	58	d(map)	Map map =new HashMap();
218		58	59	r(map), r(i)	map.put("xcant", String.valueOf(i));
219		59	56	r(map), r(lista2)	lista2.add(map);
222		60	61	d(xdatos)	List xdatos= datosManager.getDatos();
224		61	62	d(xcentro)	xcentro= centroManager.getCentro();
228		62	63	r(xfiltro), r(p)	p.put("xfiltro", xfiltro);
229		63	64	r(xactual), r(p)	p.put("xactual", xactual);
230		64	65	r(p), r(lista2)	p.put("xcantpaginas", lista2);
231		65	66	r(xcentro), r(p)	p.put("centro", xcentro);
232		66	67	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
233		67	68	r(p), r(xdatos)	p.put("datos", xdatos);
240		68	69, 73		if(request.getParameter("xestado")!=null)

242	69	70	d(xestado)	xestado=Integer.parseInt(request.getParameter("xestado"));
243	70	71	r(xestado)	System.out.println("llego estado:"+xestado);
244	71	72	d(usuariosDat), r(xestado)	usuariosDat = usuariosManager.getListaUs2pediatras(xestado);
248	72	73	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
254	73	74	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
255	74	75	d(fechaactual)	Date fechaactual = new Date();
256	75	76	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
257	76	77	r(f)	System.out.println("esta es la fecha actual1: "+f);
258	77	81	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
263	80	82		int hora, minutos, segundos;
263	81	83		int hora, minutos, segundos;
263	82	84	r(calendario), d(hora)	hora = calendario.get(Calendar.HOUR_OF_DAY);
265	83	85	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
266	84	86	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
267	85	87	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+"_"+String.valueOf(minutos)+"_"+String.valueOf(segundos);
268	86	88	d(ci)	int ci=Integer.parseInt((String)request.getSession().getAttribute("ci"));
270	87	89	r(ci)	System.out.println("ci1: "+ci);
272	88	90	r(horaregistro)	System.out.println("registro de hora"+horaregistro);
273	89	91	d(descripcion)	String descripcion="click entro a ver listado de pediatras";
274	90	92	r(descripcion), r(horaregistro), r(fechaactual), r(ci)	controlregistroManager.ingresardatos(ci, fechaactual, horaregistro, descripcion);
275	91	93		System.err.println("Exception caught: " + e.getMessage());
280	92	94	r(p)	return new ModelAndView("usuarios/gestionar medicopediatras");
283	93			
284	94		u(xactual), u(xsel), u(usuariosDat), u(xestado), u(xcentro), u(num), u(filtro), u(cont), u(n), u(n)	}

2.1.7.12.1.6 Agregar Estudiante

328	0	1	u(f), u(minutos), u(segundos), u(hora), u(fecha), u(fechaactual), u(calendario), u(lugar), u(ci)	public ModelAndView agregarpediatra(HttpServletRequest request, HttpServletResponse response)
328	1	2		public ModelAndView agregarpediatra(HttpServletRequest request, HttpServletResponse response)
329	2	3	d(p)	Map p = new HashMap();
331	3	4	d(lugar)	List lugar = usuariosManager.getLugar();
332	4	5	r(lugar), r(p)	p.put("lugar", lugar);
335	5	6	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
336	6	7	d(fechaactual)	Date fechaactual = new Date();
337	7	8	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
338	8	9	r(f)	System.out.println("esta es la fecha actual: "+f);
339	9	10	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
340	10	11	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
343	11	12	d(calendario)	Calendar calendario = new GregorianCalendar();
344	12	13		int hora, minutos, segundos;
344	13	14		int hora, minutos, segundos;
344	14	15		int hora, minutos, segundos;
346	15	16	d(hora), r(calendario)	hora = calendario.get(Calendar.HOUR_OF DAY);
347	16	17	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
348	17	18	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
349	18	19	r(minutos), r(segundos), r(hora), d(horaregistro)	String horaregistro=String.valueOf(hora)+ ":"+String.valueOf(minutos)+":"+String. valueOf(segundos);
351	19	20	d(ci)	int ci=Integer.parseInt((String)request. getAttribute("ci"));
353	20	21	r(ci)	System.out.println("ci: "+ci);
354	21	22	r(horaregistro)	System.out.println("registro de hora "+horaregistro);
355	22	23	d(descripcion)	String descripcion="click entro a agregar nuevo pediatra";
356	23	24	r(fechaactual), r(ci), r(descripcion), r(horaregistro)	controlregistroManager.ingresardatos(ci,fechaactual,horaregistro,descripcion);
359	24	25		e.printStackTrace();
362	25	26	r(p)	return new ModelAndView("usuarios/addmedi conpediatra");
363	26		u(f), u(minutos), u(segundos), u(hora), u(fecha), u(fechaactual), u(calendario), u(lugar), u(ci)	}

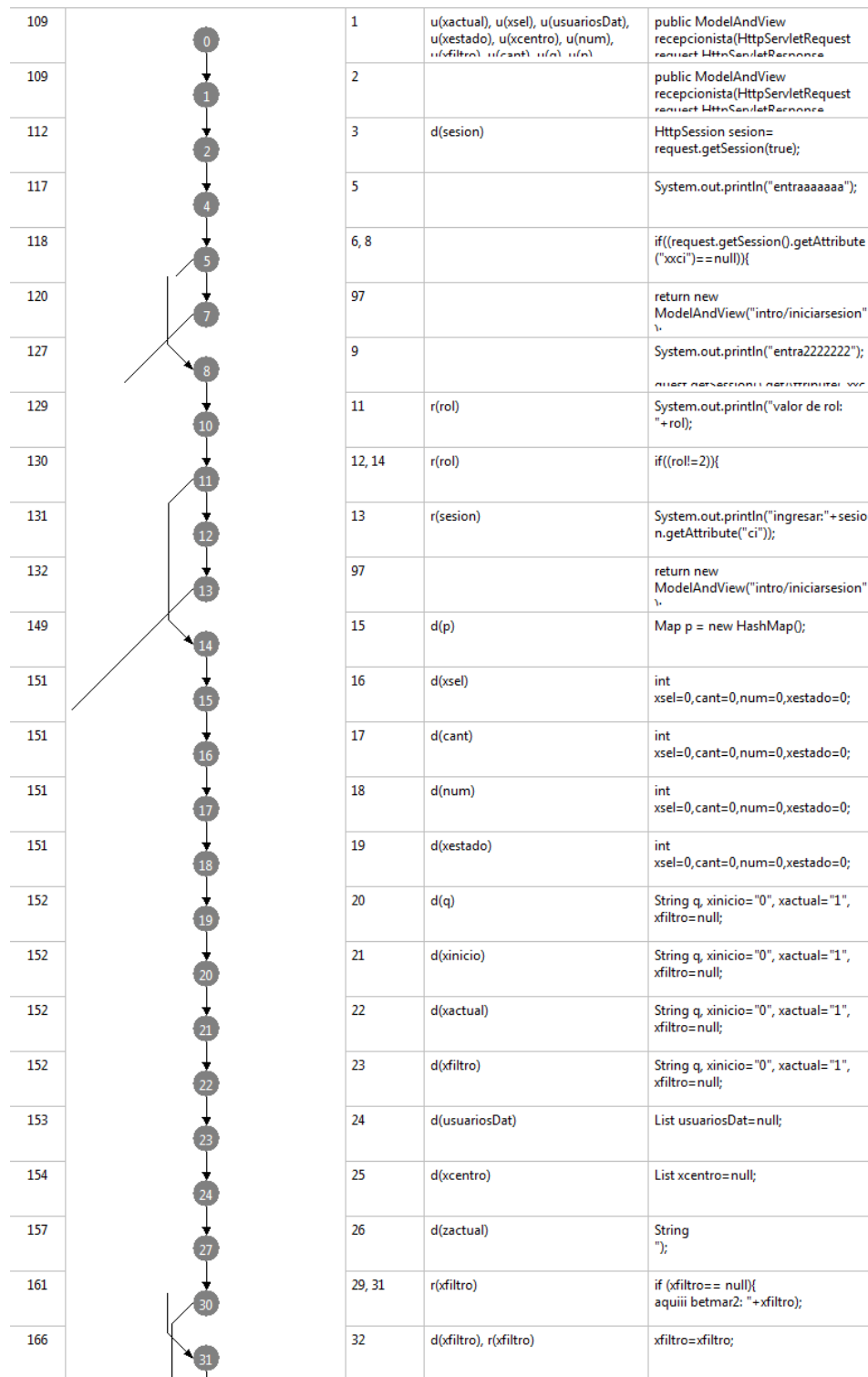
2.1.7.12.1.7 Ver Estudiante

292	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(landario), u(descripcion)	public ModelAndView verpediatra(HttpServletRequest request, HttpServletResponse response)
292	1	2		public ModelAndView verpediatra(HttpServletRequest request, HttpServletResponse response)
293	2	3	d(p)	Map p = new HashMap();
295	3	4	d(xci)	int xci= Integer.parseInt((request.getParame ter("xci")));
298	4	5	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getPediatra(xci);
300	5	6	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
303	6	7	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
304	7	8	d(fechaactual)	Date fechaactual = new Date();
305	8	9	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
306	9	10	r(f)	System.out.println("esta es la fecha actual: "+f);
307	10	11	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
308	11	12	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
311	12	13	d(calendario)	Calendar calendario = new GregorianCalendar();
312	13	14		int hora, minutos, segundos;
312	14	15		int hora, minutos, segundos;
312	15	16		int hora, minutos, segundos;
314	16	17	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF DAY);
315	17	18	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
316	18	19	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
317	19	20	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+ ":"+String.valueOf(minutos)+":"+St
319	20	21	d(ci)	int ci=Integer.parseInt((String)request. getParameter("ci"));
321	21	22	r(ci)	System.out.println("ci: "+ci);
322	22	23	r(horaregistro)	System.out.println("registro de hora "+ horaregistro);
323	23	24	d(descripcion)	String descripcion="click entro a ver datos del pediatra";
324	24	25	r(descripcion), r(horaregistro), r(fechaactual), r(ci)	controlregistroManager.ingresardat os(ci,fechaactual,horaregistro,descri pcion);
326	25	26	r(p)	return new ModelAndView("usuarios/verpediat ra");
327	26		u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(landario), u(descripcion)	}

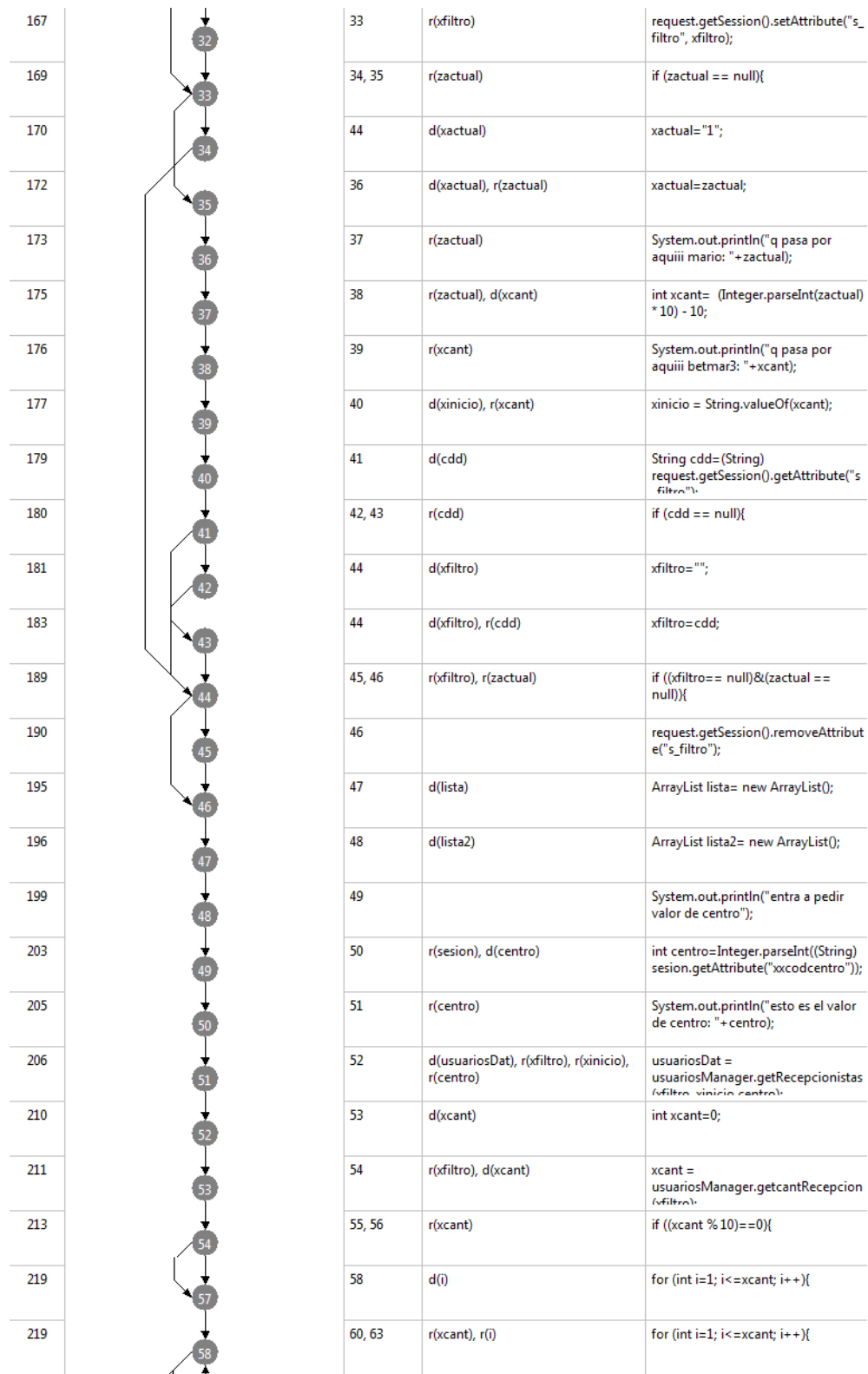
2.1.7.12.1.8 Modificar Estudiante

556	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(centro), u(usuariosDat), u(usuariosDat)	public ModelAndView modificar1(HttpServletRequest request, HttpServletResponse response)
556	1	2		public ModelAndView modificar1(HttpServletRequest request, HttpServletResponse response)
557	2	3	d(p)	Map p = new HashMap();
559	3	4	d(xci)	int xci= Integer.parseInt(request.getParameter("xci"));
561	4	5	r(xci), d(ci)	String ci=String.valueOf(xci);
562	5	6	r(ci)	request.getSession().setAttribute("c arret", ci);
564	6	7	d(lugar)	List lugar = usuariosManager.getLugar();
565	7	8	r(lugar), r(p)	p.put("lugar", lugar);
566	8	9	r(xci), d(centro)	List centro=usuariosManager.getNomb reCentro(xci);
567	9	10	r(p), r(centro)	p.put("nombrecentro", centro);
570	10	11	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getUsuario(xci);
572	11	12	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
574	12	13	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
575	13	14	d(fechaactual)	Date fechaactual = new Date();
576	14	15	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
577	15	16	r(f)	System.out.println("esta es la fecha actual: "+f);
578	16	17	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
579	17	18	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
582	18	19	d(calendario)	Calendar calendario = new GregorianCalendar();
583	19	20		int hora, minutos, segundos;
583	20	21		int hora, minutos, segundos;
583	21	22		int hora, minutos, segundos;
585	22	23	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF DAY);
586	23	24	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
587	24	25	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);

2.1.7.12.1.9 Gestionar Docente



219	58	r(xcant), r(i)	for (int i=1; i<=xcant; i++){
219	59	d(i)	for (int i=1; i<=xcant; i++){
220	60	d(map)	Map map = new HashMap();
221	61	r(map), r(i)	map.put("xcant", String.valueOf(i));
222	62	r(lista2), r(map)	lista2.add(map);
225	63	d(xdatos)	List xdatos= datosManager.getDatos();
227	64	d(xcentro)	xcentro= centroManager.getCentro();
231	65	r(xfiltro), r(p)	p.put("xfiltro", xfiltro);
232	66	r(xactual), r(p)	p.put("xactual", xactual);
233	67	r(p), r(lista2)	p.put("xcantpaginas", lista2);
234	68	r(xcentro), r(p)	p.put("centro", xcentro);
235	69	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
236	70	r(p), r(xdatos)	p.put("datos", xdatos);
243	71		if(request.getParameter("xestado")!=null)
245	72	d(xestado)	xestado=Integer.parseInt(request.getParameter("xestado"));
246	73	r(xestado)	System.out.println("llego estado: "+xestado);
247	74	d(usuariosDat), r(xestado), r(centro)	usuariosDat = usuariosManager.getListarecepcion ?(xestado, centro);
251	75	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
256	76	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
257	77	d(fechaactual)	Date fechaactual = new Date();
258	78	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
259	79	r(f)	System.out.println("esta es la fecha actual: "+f);
260	80	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
261	83	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
265	84		int hora, minutos, segundos;



265	85		int hora, minutos, segundos;
267	86	r(calendario), d(hora)	hora = calendario.get(Calendar.HOUR_OF_DAY);
268	87	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
269	88	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
270	89	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+":"+String.valueOf(minutos)+":"+String.valueOf(segundos);
272	90	d(ci)	int ci=Integer.parseInt((String)request.getSession().getAttribute("vci"));
274	91	r(ci)	System.out.println("ci: "+ci);
275	92	r(horegistro)	System.out.println("registro de hora "+horegistro);
276	93	d(descripcion)	String descripcion="click entro a ver recepcionistas del centro de salud";
277	94	r(descripcion), r(horegistro), r(fechaactual), r(ci)	controlregistroManager.ingresardatos(ci,fechaactual,horegistro,descripcion);
283	95		System.err.println("Exception caught: " + e.getMessage());
286	96	r(p)	return new ModelAndView("recepcionistas/recepcionista");
287	97	u(xactual), u(xsel), u(usuariosDat), u(xestado), u(xcentro), u(num), u(xfiltro), u(xent), u(n), u(n)	}

2.1.7.12.1.10 Eliminar Docente

295	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario) u(descripcion)	public ModelAndView verrepcionistas(HttpServletRequest request HttpServletResponse)
295	1	2		public ModelAndView verrepcionistas(HttpServletRequest request HttpServletResponse)
296	2	3	d(p)	Map p = new HashMap();
298	3	4	d(xci)	int xci= Integer.parseInt((request.getPara meter("radiola")) 16);
300	4	5	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getRecepcion(xci) ;
302	5	6	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
303	6	7	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
304	7	8	d(fechaactual)	Date fechaactual = new Date();
305	8	9	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
306	9	10	r(f)	System.out.println("esta es la fecha actual: "+f);
307	10	11	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
308	11	12	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
311	12	13	d(calendario)	Calendar calendario = new GregorianCalendar();
312	13	14		int hora, minutos, segundos;
312	14	15		int hora, minutos, segundos;
312	15	16		int hora, minutos, segundos;
314	16	17	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF DAY);
315	17	18	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
316	18	19	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
317	19	20	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+ ":"+String.valueOf(minutos)+":"+St
319	20	21	d(ci)	int ci=Integer.parseInt((String)request. getParameter("radiola"));
321	21	22	r(ci)	System.out.println("ci: "+ci);
322	22	23	r(horaregistro)	System.out.println("registro de hora "+ horaregistro);
323	23	24	d(descripcion)	String descripcion="click ver datos del recepcionista del centro de salud";
324	24	25	r(descripcion), r(horaregistro), r(fechaactual), r(ci)	controlregistroManager.ingresardat os(ci,fechaactual,horaregistro,descri pcion);
326	25	26	r(p)	return new ModelAndView("repcionistas/ver repcionistas");
327	26		u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario) u(descripcion)	}

2.1.7.12.1.11 Agregar Docente

295	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	public ModelAndView verrepcionistas(HttpServletRequest request, HttpServletResponse response)
295	1	2		public ModelAndView verrepcionistas(HttpServletRequest request, HttpServletResponse response)
296	2	3	d(p)	Map p = new HashMap();
298	3	4	d(xci)	int xci= Integer.parseInt((request.getParame ter("xci")));
300	4	5	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getRecepcion(xci);
302	5	6	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
303	6	7	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
304	7	8	d(fechaactual)	Date fechaactual = new Date();
305	8	9	d(f), r(fecha), r(fechaactual)	String fs=fecha.format(fechaactual);
306	9	10	r(f)	System.out.println("esta es la fecha actual: "+f);
307	10	11	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
308	11	12	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
311	12	13	d(calendario)	Calendar calendario = new GregorianCalendar();
312	13	14		int hora, minutos, segundos;
312	14	15		int hora, minutos, segundos;
312	15	16		int hora, minutos, segundos;
314	16	17	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF DAY);
315	17	18	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
316	18	19	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
317	19	20	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+ ":"+String.valueOf(minutos)+":"+St
319	20	21	d(ci)	int ci=Integer.parseInt((String)request. getParameter("ci"));
321	21	22	r(ci)	System.out.println("ci: "+ci);
322	22	23	r(horaregistro)	System.out.println("registro de hora "+horaregistro);
323	23	24	d(descripcion)	String descripcion="click ver datos del recepcionista del centro de salud";
324	24	25	r(descripcion), r(horaregistro), r(fechaactual), r(ci)	controlregistroManager.ingresardat os(ci,fechaactual,horaregistro,descri pcion);
326	25	26	r(p)	return new ModelAndView("recepcionistas/ver repcionistas");
327	26		u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	}

2.1.7.12.1.12 Modificar Docente

565	0	1	u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	public ModelAndView modRecepcionistas1(HttpServletRequestReq uest request, HttpServletResponse
565	1	2		public ModelAndView modRecepcionistas1(HttpServletRequestReq uest request, HttpServletResponse
566	2	3	d(p)	Map p = new HashMap();
568	3	4	d(xci)	int xci= Integer.parseInt(request.getParame ter("xci"));
570	4	5	r(xci), d(ci)	String ci=String.valueOf(xci);
571	5	6	r(ci)	request.getSession().setAttribute("c ariet", ci);
576	6	7	d(usuariosDat), r(xci)	List usuariosDat = usuariosManager.getRecepcion(xci)
578	7	8	r(usuariosDat), r(p)	p.put("usuariosDatos", usuariosDat);
581	8	9	d(fecha)	SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
582	9	10	d(fechaactual)	Date fechaactual = new Date();
583	10	11	d(f), r(fecha), r(fechaactual)	String f=fecha.format(fechaactual);
584	11	12	r(f)	System.out.println("esta es la fecha actual: "+f);
585	12	13	r(f), r(fecha), d(fechaactual)	fechaactual=fecha.parse(f);
586	13	14	r(fechaactual)	System.out.println("esta es la fecha actual: "+fechaactual);
589	14	15	d(calendario)	Calendar calendario = new GregorianCalendar();
590	15	16		int hora, minutos, segundos;
590	16	17		int hora, minutos, segundos;
590	17	18		int hora, minutos, segundos;
592	18	19	r(calendario), d(hora)	hora =calendario.get(Calendar.HOUR_OF DAY);
593	19	20	d(minutos), r(calendario)	minutos = calendario.get(Calendar.MINUTE);
594	20	21	d(segundos), r(calendario)	segundos = calendario.get(Calendar.SECOND);
595	21	22	r(minutos), r(segundos), d(horaregistro), r(hora)	String horaregistro=String.valueOf(hora)+ ":"+String.valueOf(minutos)+":"+String
597	22	23	d(cci)	int cci=Integer.parseInt((String)request. getSession().getAttribute("cci"));
599	23	24	r(ci)	System.out.println("ci: "+ci);
600	24	25	r(horaregistro)	System.out.println("registro de hora "+horaregistro);
601	25	26	d(descripcion)	String descripcion="click entro a modificar datos del recepcionista del centro de salud";
602	26	27	r(descripcion), r(horaregistro), r(cci), r(fechaactual)	controlregistroManager.ingresardat os(cci,fechaactual,horaregistro,desc ripcion);
604	27	28	r(p)	return new ModelAndView("recepcionistas/mo dificarRecepcionistas1");
605	28		u(f), u(minutos), u(segundos), u(fecha), u(usuariosDat), u(calendario), u(descripcion)	}

2.1.7.12.2 Pruebas de Caja Negra

2.1.7.12.2.1 Introducción

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la misma, y los resultados esperados. Estos casos de prueba son aplicados como pruebas (comprobación) de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de ensayo con las instrucciones para realizar la prueba.

2.1.7.12.2.2 Propósito

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error
- Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si se descubre un error no detectado hasta entonces.

2.1.7.12.2.3 Alcance

- Describir los casos de prueba de los formularios principales del sistema
- Identificar y definir estos casos de prueba para evitar que existan errores y para obtener una salida correcta.

2.1.7.12.2.3.1 . Modelos de Casos de Prueba

Formulario: Ingresar al Sistema como usuario

The screenshot shows a web browser window with the title 'INCOS'. The address bar displays 'localhost:8888/users/manage_users'. The page content features a login form for 'Carlos Quiñones'. The form has two input fields: 'Usuario:' with the text 'Carlos' and 'Contraseña:' with masked characters '.....'. Below the fields are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange). In the background, a table of users is visible, partially obscured by the login form.

Nombre	Cargo	Estado	Acciones
Lorenzo Lamas Rosas	Profesor	Activo	[Edit] [Delete] [Details]
Mariela Perez Velasquez	Secretaria	Activo	[Edit] [Delete] [Details]
Luis Fernandez Prieto	Profesor	Activo	[Edit] [Delete] [Details]
Rodrigo Velasquez Mendez	Profesor	Eliminado	[Edit] [Delete] [Details]
Margarita Perales Ibañez	Secretaria	Activo	[Edit] [Delete] [Details]
Mario Suarez Montes	Profesor	Activo	[Edit] [Delete] [Details]
Alvaro Montellanos	Profesor	Activo	[Edit] [Delete] [Details]

Figura 122. Ingresar al sistema al Sistema Como Usuario

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
Valor de entrada de Usuario	1. Alfanumérico	2. Otro Valor
Tamaño de Usuario	3. $6 \leq \text{Valor} \leq 20$	4. Valor < 6 5. Valor > 20
Valor de entrada de Clave	6. Alfanumérico	7. Otro Valor
Tamaño de Clave	8. $0 \leq \text{Valor} \leq 20$	9. Valor < 1 10. Valor > 20

Tabla 56 Formulario Acceso al Sistema Como Usuario

FORMULARIO: Adicionar Estudiante

The screenshot shows a web browser window with the address bar displaying 'localhost:8888/users/new_user'. The page title is 'Nuevo Usuario'. The form contains the following fields and controls:

- Tipo:** A dropdown menu with the selected value '--Elija Uno--'.
- Carrera:** A dropdown menu with the selected value 'Informatica'.
- CI:** A text input field.
- Nombre(s):** A text input field.
- Paterno:** A text input field.
- Materno:** A text input field.
- Nac.:** A text input field.
- Usuario:** A text input field.
- Contraseña:** A text input field.
- Buttons:** 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 123. Formulario Adicionar Estudiante

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
Valor de entrada de CI.	1. Numérico	2. Otro Valor
Valor de entrada de	3. Alfanumérico	4. Otro Valor

Nombre		
Tamaño de Nombre	5. $5 \leq \text{Valor} \leq 50$	6. Valor < 5 7. Valor > 50
Valor de entrada de Apellido Paterno	8. Alfanumérico	9. Otro Valor
Tamaño de Apellido Paterno	10. $0 \leq \text{Valor} \leq 50$	11. Valor < 1 12. Valor > 50
Valor de entrada de Apellido Materno	13. Alfanumérico	14. Otro Valor
Tamaño de Apellido Materno	15. $1 \leq \text{Valor} \leq 50$	16. Valor < 1 17. Valor > 60
Valor de entrada de dirección	33. Alfanumérico	34. Otro Valor
Tamaño de dirección	35. $0 \leq \text{Valor} \leq 100$	36. Valor < 0 37. Valor > 100
Tamaño de Teléfono	38. $0 \leq \text{Valor} \leq 10$	39. Valor < 0 40. Valor > 10

Tabla 57. Formulario Adicionar Estudiante

FORMULARIO: Formulario Adicionar Usuario

The screenshot shows a web browser window with the INCOS application running on localhost:8888/users/new_user. The page has a dark navigation bar with 'Inicio', 'maria_re', and 'Menú'. The main content area is titled 'Nuevo Usuario' and contains the following form fields:

- Tipo:** A dropdown menu with the selected value '--Elija Uno--'.
- Carrera:** A dropdown menu with the selected value 'Informatica'.
- CI:** A text input field.
- Nombre(s):** A text input field.
- Paterno:** A text input field.
- Materno:** A text input field.
- Nac.:** A text input field.
- Usuario:** A text input field.
- Contraseña:** A text input field.

At the bottom of the form are two buttons: 'Aceptar' (blue) and 'Cancelar' (orange).

Figura 124. Formulario Adicionar Usuario

Condiciones de Entrada	Condiciones Válidas	Condiciones Inválidas
ci	1. Numérico	2. Otro Valor

Tamaño de ci	3. $1 \leq \text{Valor} \leq 7$	4. Valor < 1 5. Valor > 7
Valor de entrada de Nombre	6. Alfanumérico	7. Otro Valor
Tamaño de Nombre	8. $5 \leq \text{Valor} \leq 50$	9. Valor < 5 10. Valor > 50
Valor de entrada Apellido Paterno	11. Alfanumérico	12. Otro
Tamaño de Apellido Paterno	12. $0 \leq \text{Valor} \leq 50$	13. Valor < 1 14. Valor > 50
Valor de entrada Apellido Materno	15. Alfanumérico	16. Otro
Tamaño de Apellido Materno	17. $1 \leq \text{Valor} \leq 50$	18. Valor < 1 19. Valor > 60
Valor de entrada tipo usuario	20. Alfanumérico	21. Otro
Tamaño de tipo de usuario	22. $1 \leq \text{Valor} \leq 50$	23. Valor < 1 24. Valor > 60
Valor de entrada Usuario	25. Alfanumérico	26. Otro

Tamaño de usuario	27. $3 \leq \text{Valor} \leq 10$	28. Valor < 3 29. Valor > 10
Valor de entrada Clave	30. Alfanumérico	31. Otro
Tamaño de clave	32. $0 \leq \text{Valor} \leq 10$	33. Valor < 0 34. Valor > 10

Tabla 58. Formulario Adicionar Usuario

1. Caso de Prueba “Ingresar al Sistema”

1.1.Descripción del Caso de Prueba

En esta parte del documento sólo se cubrirá la lista de pruebas realizadas sobre el Caso de Uso “Ingresar al Sistema”.

Las pruebas realizadas a este caso de uso son:

d

- A. Introducir Datos Correctos de Ingreso.
- B. Introducir Datos Incorrectos de Ingreso.

1.2.Condición de ejecución

A. Introducir Datos Correctos.

Escribirnos la Url de la página de inicio de sesión para que nos muestre el formulario por el cual podemos acceder al Sistema.

Precondiciones.

Para ingresar al Sistema el usuario tiene que estar registrado en el Sistema.

Valor de entrada.

A continuación se muestra los datos correctos con los cuales se pueden acceder al Sistema.

- ❖ Introducimos: “Lamas” en el campo Usuario.

- ❖ Introducimos: “lamas123” en el campo Clave.
- ❖ Pulsamos el botón “Aceptar”.
- ❖ El sistema nos muestra la pantalla Principal del Administrar con los privilegios de acuerdo al rol que lo pertenece.

Observaciones

Con estos datos de usuario y clave no se muestra ningún error.

Resultados esperados

El proceso termina y el usuario ingresa al sistema sin ningún problema.

B. Introducir Datos Incorrectos de Ingreso.

Escribirnos la Url de la página de inicio de sesión para que nos muestre el formulario por el cual podemos acceder al sistema.

Precondiciones.

La condición que se debe dar en el Sistema antes de ingresar al mismo es que el usuario introduzca datos erróneos o que no esté registrado como usuario del sistema.

Valor de entrada.

A continuación se muestra los datos con los cuales el sistema nos muestra el mensaje de error.

- ❖ Introducimos: “uno” en el campo Usuario.
- ❖ Introducimos: “dos” en el campo Clave.
- ❖ Pulsamos el botón “Aceptar”.
- ❖ El Sistema nos muestra un mensaje de error.

Observaciones

Los errores se muestran en al lado de los datos de entrada del formulario.

Resultados esperados

El proceso termina, porque se muestra un mensaje de error.

2.1.7.13 MEDIOS DE VERIFICACIÓN

Medios de Verificación del Componente 1 Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija.

Informe otorgado por el Jefe Académico del Instituto Comercial Superior sobre la Culminación de la Actividad denominada “Determinación de Requerimiento”



Instituto Comercial Superior (INCOS) Tarija.

Tarija, lunes 4 de Agosto de 2014

Señores

Docentes de Taller III

REF.: PROYECTO “MEJORAMIENTO DE LA ADMINISTRACIÓN ACADÉMICA DEL INSTITUTO COMERCIAL SUPERIOR (INCOS) TARIJA.”

De mi mayor consideración:

El motivo de la presente, es para hacerlo conocer que como Jefe Académico del Instituto Comercial Superior. – Tarija doy por aprobado la etapa de **Especificación de Requerimientos** del proyecto titulado “**Mejoramiento De La Administración Académica Del Instituto Comercial Superior (INCOS) Tarija.**” elaborado por el universitario Wilson Luis Narvaez Michel o que viene trabajando desde Abril en esta tarea y al cual se le otorgo toda la información necesaria.

Habiendo revisado el documento titulado **Especificación de Requisitos de Software** bajo la **norma IEEE-830**, en compañía del tesista Doy por aprobado los requerimientos que serán cumplidos por el sistema, que en mi opinión son acertados a las necesidades de la Institución, pero sin dejar de lado futuras modificaciones que mejoren mi trabajo.

Sin otra particular, me despido de ustedes con las consideraciones más distinguidas.

Atentamente:

Ing. Carlos Quiñones

(Jefe Académico del Instituto Comercial Superior Tarija)



CAPÍTULO III

III. CAPITULO

3.1 COMPONENTE II: CAPACITACION AL PERSONAL INVOLUCRADO

3.1.1 Introducción

El objetivo de este componente es capacitar a los usuarios en el uso del sistema web para el Mejoramiento De La Administración Académica Del Instituto Comercial Superior (INCOS) Tarija. Según el nivel de los mismos empleando métodos y medios de enseñanza-aprendizaje adecuado. Con lo siguiente se pretende evitar errores y riesgos en el manejo del sistema y así poder sacar el máximo beneficio a dicho sistema.

El propósito del proyecto es el:” Mejoramiento De La Administración Académica Del Instituto Comercial Superior (INCOS) Tarija; la capacitación en el uso del sistema informático al personal afectado por el proyecto se convierte en un componente fundamental para el logro del mismo.

El componente capacitación, se encamina hacia el siguiente objetivo: usar adecuadamente el Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija explotando las fortalezas del mismo.

La Capacitación será presencial dada la corta duración de la misma, la disponibilidad de ambientes, de materiales didácticos y la importancia de posibilitar que el alumno (usuario) reciba asesoramiento oportuno ante cualquier consulta.

3.1.2 Contexto

La Capacitación se desarrollará en dos partes: la primera parte tiene como objetivo que el personal a capacitar conozca en forma global los alcances y beneficios que el sistema informático “Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija” aporta a la institución “Instituto Comercial Superior (INCOS) Tarija” así como los cambios positivos y responsabilidades que esto implica para la institución.

Se realizarán actividades de capacitación personalizadas de acuerdo al rol que a cada uno le compete.

En este contexto el Capacitador confeccionó la Guía para Capacitación tomando en cuenta los diferentes niveles de preparación del usuario final.

El rol del capacitador estará en función a las categorías de los usuarios según el siguiente detalle:

Nivel ejecutivo:

- Se mostrará la importancia de la capacitación, objetivos y participación del personal seleccionado.

Personal Técnico

Se realizarán actividades de capacitación acorde a las siguientes categorías:

- ✓ Personal de soporte técnico al usuario final (si corresponde)
- ✓ Personal de administración de servicios (si corresponde)
- ✓ Personal de desarrollo (si corresponde)

Usuarios Finales

3.1.3 Propuesta Pedagógica

La propuesta pedagógica a utilizar dada las características de los usuarios del Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija tendrá en cuenta sus particularidades, el rol que juega dentro de la organización y niveles de conocimiento.

Los métodos de enseñanza a utilizar pondrán su énfasis principalmente en tres teorías de aprendizajes: la cognitiva, con su máximo exponente en el constructivismo, la colaborativa, fundamentalmente para ser explotada con intensidad en la formación del personal técnico y finalmente la significativa aunque también estará presente en la formación del personal de las categorías de nivel ejecutivo y de usuarios finales.

El aprendizaje colaborativo se entiende como el proceso en el que los alumnos aprenden mientras proponen y comparten ideas para resolver una tarea, favoreciéndose con el diálogo y la reflexión sobre las propuestas propias y las de sus compañeros.

Este punto sí podría enriquecerse con el uso de las tecnologías. Se trata por tanto de construir sistemas muy adaptables a los diferentes grupos de usuarios, en donde el soporte y la intervención no impongan un comportamiento prescriptivo. Desde el campo de la psicología, algunos autores, especialmente ligados a lo que se ha llamado la psicología socio-cultural, postulaban que aprender es una experiencia de carácter fundamentalmente social, en dónde el lenguaje juega un papel básico como herramienta de mediación no sólo entre profesor y alumno sino también entre compañeros.

Finalmente se pone de manifiesto el aprendizaje significativo porque el alumno tiene que incorporar los nuevos conocimientos en forma sustantiva en su estructura cognitiva. Esto se logra cuando el alumno relaciona los nuevos conocimientos con los anteriormente adquiridos; pero también es necesario que el alumno se interese por aprender lo que se le está mostrando. De esta forma el alumno no solo obtendrá

resultados satisfactorios en un trabajo final, sino que será capaz de enfrentarse a diversas situaciones donde podrá aplicar los conocimientos adquiridos.

3.1.4 CONTENIDOS DE LA CAPACITACION

(Se listan los contenidos detallados estructurados por temas o unidades)

3.1.5 Plan de Clases

Nro.	CONTENIDO	OBJETIVO	Fecha	DURACION (horas)	MATERIAL DIDÁCTICO	MEDIOS DE ENSEÑANZA - APRENDIZAJE	DESTINATARIO
1	Lección 1: Ingresar al sistema	Que el usuario se identifique y pueda entrar al sistema de acuerdo a su rol.	11/11/2013	1	Guía de laboratorio y diapositivas	Data display, computadora, pizarra.	Jefe Académico, Docente y Alumnos.
2	Lección 2: Gestionar Materias	Que el usuario pueda operar en el sistema realizando el registro de las materias.	11/11/2013	1	Guía de laboratorio y diapositivas	Data display, computadora, pizarra.	Jefe Académico, Docente.

3	Lección 3: Gestionar Carreras	Que el usuario pueda realizar las operaciones de manejo adecuado a su rol.	11/11/2013	1	Guía de laboratorio y diapositivas	Data display, computadora, pizarra.	Jefe Académico
4	Lección 4: Gestión de Usuarios	Que el usuario pueda operar en el sistema realizando el registro de usuarios.	12/11/2013	1,5	Guía de laboratorio y diapositivas	Data display, computadora, pizarra.	Jefe Académico
5	Lección 5: Gestión de Notas	Que el usuario pueda operar en el sistema realizando el registro de notas.	12/11/2013	1,5	Guía de laboratorio y diapositivas	Data display, computadora, pizarra.	Jefe Académico, Docentes, Alumnos

3.1.6 Cronograma

	Modo de	Nombre de tarea	Dura	Comienzo	Fin
1					
2		Fase de Análisis y Diseño del Sistema Informático	60 días	mié 06/02/13	mar 30/04/13
3		Fase de Desarrollo del Sistema Informático	90 días	mié 01/05/13	mar 03/09/13
4		Fase de Pruebas del Sistema Informático	30 días	mié 04/09/13	mar 15/10/13
5		Presentación del Sistema	15 días	mié 16/10/13	mar 05/11/13
6		Elaboración de material de apoyo para el uso del software elaborado (Guías de usuario)	15 días	mié 06/11/13	mar 26/11/13
7		Cursos de Capacitación	2 días	mié 27/11/13	jue 28/11/13

12	04 feb '13			18 mar '13			29 abr '13			10 jun '13			22 jul '13			02 sep '13			14 oct '13			25 nov '13			06 ene '14	
1	29	16	06	24	11	29	17	04	22	10	28	15	02	20	08	26	13	01	19	06	24					



3.1.7 Resultados esperados

En noviembre de 2013 los encargados usuarios del sistema son capacitados en un 80% en el uso del Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija, habiendo realizado 2 jornadas de capacitación.

3.1.8 MEDIOS DE VERIFICACIÓN

3.1.8.1 Medios de verificación de componente 2: “Capacitación del personal

involucrado”

- 1. Informe otorgado por la Secretaria del Departamento Informática y Sistemas para Instituto Comercial Superior (INCOS) Tarija con el motivo de brindarle información, al Director del Proyecto “Mejoramiento de la Administración Académica del Instituto Comercial Superior (INCOS) Tarija.”**
- ✓ Informe otorgado por el Jefe Académico Instituto Comercial Superior (INCOS) Tarija del sobre la “Capacitación”.**
- ✓ Documento impreso de Manual de Usuario**

- 1. Informe otorgado por la Secretaria del Departamento Informática y Sistemas para Instituto Comercial Superior (INCOS) Tarija con el motivo de brindarle información, al Director del Proyecto “Mejoramiento de la Administración Académica del Instituto Comercial Superior (INCOS) Tarija.”**



Departamento de
Informática y Sistemas

UNIVERSIDAD AUTONOMA JUAN MISAEL SARACHO
FACULTAD DE CIENCIAS Y TECNOLOGIA
DEPARTAMENTO DE INFORMATICA Y SISTEMAS
DIRECCION DE DEPARTAMENTO

Campus Universitario "El Tejar" - Edificio FCYT, 3er. Piso - Tel. 591 466 40265 Fax: 591 466 40265
Email: dis@uajms.edu.bo - web: dis.uajms.edu.bo - Casilla de correo #53 - Tarija - Bolivia

Tarija, 4 de Abril de 2012

FAC. CS. Y TEC, DPTO. INF. Y SIST, OF. N° 64/12

Señor

Ing. Carlos Quiñones

JEFE ACADEMICO INCOS TARIJA

Presente

De mi mayor consideración:

En mi calidad de Director del Dpto. de Informática y Sistemas, me cabe informarle que el alumno **WILSON LUIS NARVAEZ MICHEL** que cursa la materia de Taller III y como actividad principal de la misma debe diseñar y ejecutar un proyecto informático en una institución del departamento.

En éste sentido, le solicito en lo posible que se le preste la colaboración necesaria que principalmente se concentrará en proporcionarle la información que ella solicite de manera oportuna y completa. De ésta manera el estudiante podrá apoyarle efectivamente a su Institución.

Agradecido de antemano por su colaboración, reciba usted mis consideraciones más distinguidas.

Atentamente


Lic. S. Efraín Torrejón Tejerina
DIRECTOR DPTO. INFORMATICA Y SISTEMAS
FACULTAD DE CIENCIAS Y TECNOLOGIA



Cc arch.



Sr.

Ing. Ing. Carlos Quiñones

Jefe Académico del Instituto Comercial Superior Tarija

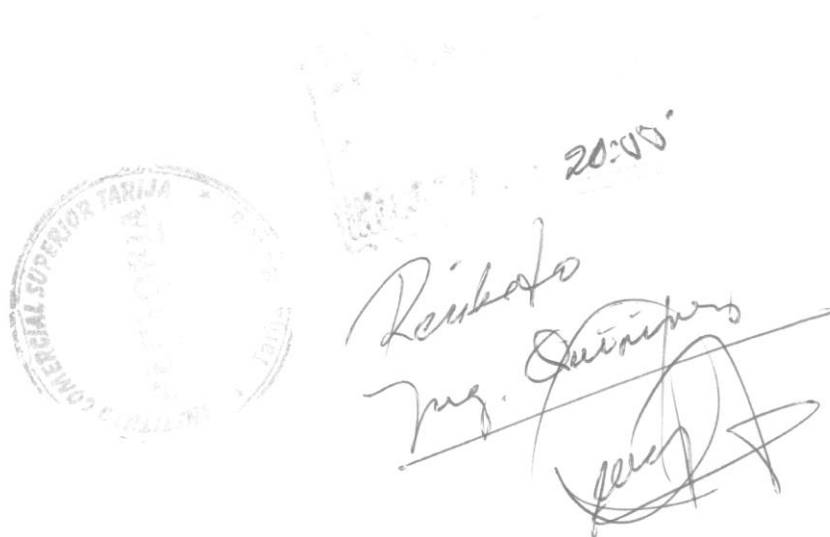
Ref.: solicitud para capacitación

Estimado Sr.

Con el debido respeto me dirijo a usted como el director del proyecto titulado “Mejoramiento de la Administración Académica del Instituto Comercial Superior (INCOS) Tarija”. Haciéndole llegar el presente, solicitándole que se me permita capacitar al personal que desempeña sus labores en su empresa y que están involucrados con el proyecto ya mencionado. Como es de su conocimiento el sistema ya fue terminado, por lo tanto considero necesaria dicha capacitación. Por parte del personal es necesario e importante el entusiasmo en esta actividad.

Me despido esperando una respuesta positiva y deseándole éxitos en sus labores que desempeña.

Atentamente:

A handwritten signature in dark ink, which appears to read "Carlos Quiñones", is written over a circular official stamp. The stamp contains the text "INSTITUTO COMERCIAL SUPERIOR TARIJA" around its perimeter. Above the signature, there is a handwritten date "20/05".

Univ. Wilson Luis Narvaez Michel
(Director del proyecto)



Instituto Comercial Superior (INCOS) Tarija.

Tarija, lunes 4 de Agosto de 2014

Lic. Efraín Torrejón Tejerina

(Docente de la materia de Taller III)

REF.: PROYECTO “MEJORAMIENTO DE LA ADMINISTRACIÓN ACADÉMICA DEL INSTITUTO COMERCIAL SUPERIOR (INCOS) TARIJA.”

De mi mayor consideración:

El motivo de la presente, es para darle a conocer que el universitario Wilson Luis Narvaez Michel ha cumplido con el proyecto “Mejoramiento De La Administración Académica Del Instituto Comercial Superior (INCOS) Tarija.”. Planteado anteriormente.

Expresando así mi conformidad, me complace comunicarles que el Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija cumple con todos los requisitos funcionales para la eficiente administración académica del instituto.

La capacitación a los usuarios sobre la administración y usabilidad del sistema WEB fue realizado con éxito a las personas involucradas.

Sin otra particular, me despido de usted con las consideraciones más distinguidas.

Atentamente:

Ing. Carlos Quiñones

(Jefe Académico del Instituto Comercial Superior Tarija)



3.1.9 Conclusiones

La capacitación realizada no fue tan óptima como se esperaba, porque el personal involucrado no respetó los horarios y la asistencia, Pese a lo antes mencionado se realizó los talleres cumpliendo con el calendario.

El material didáctico preparado fue de gran utilidad, porque el personal no tenía mucho conocimiento en cuanto a sistemas informáticos y el hardware necesario.

CONCLUSIONES Y RECOMENDACIONES

III. CONCLUSIONES Y RECOMENDACIONES

4.2 Conclusiones

Una buena forma de abordar el proyecto fue la matriz de marco lógico (MML), porque resume de manera simple y concisa la estructura del proyecto contando con los indicadores, medios de verificación y supuestos. El indicador del propósito es muy importante ya que nos muestra el efecto directo logrado al término de la ejecución del proyecto.

Después de haber finalizado el proyecto que la Administración Académica del Instituto Comercial Tarija ha sido mejorada, evitando demoras al momento de consultar por información sobre las notas de los estudiantes, como también es de mucha ayuda para los docentes ya que ellos pueden realizar los registros de notas de una manera más segura.

Al haber finalizado el presente proyecto de Taller III se concluye que todas las materias cursadas a lo largo de la carrera fueron de utilidad para poder llevar a finalizar con éxito el presente proyecto ya que sin los conocimientos previamente adquiridos no podría haber analizado y desarrollado el componente Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS) Tarija que es de vital importancia para poder cumplir el objetivo General del Proyecto.

Al final de la ejecución del proyecto, los procesos que involucra el manejo de información con referencia a la administración Académica del Instituto Comercial Superior (INCOS) Tarija, han sido automatizados.

También el cumplimiento de los supuestos es importante ya que tienen que ocurrir, junto con el logro del Propósito, componentes y actividades, para contribuir de manera significativa al Fin del proyecto.

Los componentes del proyecto finalizaron de manera exitosa de acuerdo al calendario. El componente Sistema Informático para la administración Académica del

Instituto Comercial Superior (INCOS) Tarija, debido a su gran importancia se le dio inicio con anterioridad, sabiendo que sin este no se podría ejecutar el segundo componente “Capacitación del uso de los recursos TIC y el manejo del Sistema Informático para la administración Académica del Instituto Comercial Superior (INCOS)Tarija desarrollado”. El cual fue indispensable para los usuarios del sistema.

Con el sistema actual se observa que las tareas manuales en el manejo de notas, el control de los usuarios, estudiantes, docentes y la información académica se optimizan de una manera que ayuda a automatizar los datos y tenerlos actualizados, evitando las demoras.

Se logró generar reportes que servirán a los estudiantes para poder informarse del acerca de las calificaciones que poseen en sus distintas materias de una forma más rápida y confiable.

La metodología usada fue RUP, la metodología estándar más utilizada para el desarrollo de la documentación, asegurando la producción de un software de calidad dentro de plazos y presupuestos predecibles.

4.3 Recomendaciones

El fin del proyecto es Contribuir al fortalecimiento de la Gestión Académica de información del Instituto Comercial Superior (INCOS) Tarija, el sistema y el programa de capacitación, son parte sustancial para contribuir a ello. Sin embargo el sistema informático debería ser implementado y alojado en un servidor web con sus dominios respectivos, para esto también se necesitaría diseñar una red, por la estructura de la institución debería ser de tipo LAN, por ser de corto alcance y tamaño.

Para elaborar un proyecto posterior se recomienda esta metodología a los desarrolladores, ya que esta de acorde a lo requerido, siendo una forma más rápida de avance.

Es importante el conocimiento que adquiere el personal involucrado ya que ellos serán los que interactúen con el sistema, se recomienda controlar la asistencia, disponer de material didáctico y brindar certificados para regenerar mayor interés en los capacitados.

También es importante que para la implantación y migración de datos de la empresa al nuevo sistema, se debería contar con personal calificado en el área de Tecnologías de la Computación o el Administrador (jefe del proyecto), para no cometer errores que afectarían al sistema en un futuro.

Se aconseja manejar los navegadores como ser: Mozilla Firefox, Google Chrome, Opera.