

CAPITULO I

EL PROYECTO

1 Capítulo I : El Proyecto

1.1 Presentación del Proyecto

1.1.1 Título del Proyecto

El título del siguiente proyecto es: Mejoramiento en el Proceso de Administración de las inscripciones del Colegio Fe y Alegría “Humberto Portocarrero I”.

1.1.2 Carrera

Ingeniería Informática

1.1.3 Facultad

Ciencias y Tecnologías

1.1.4 Duración del Proyecto

8 meses

1.1.5 Área / línea de investigación priorizada

Desarrollo de Sistemas de Información

1.2 Personal Vinculado al Proyecto

Director del Proyecto

Cuenca Apellido Paterno	Aguado ApellidoMaterno	Angelo Nombre	711527 9 C.I.
Estudiante	Ing. Informática	Ciencias Y Tecnología	

Profesión	Carrera	Facultad:	
Telf. Oficina	76191547 Celular	Angelo.cueca22@gmail.com. Correo electrónico	Firma

Tabla 1 Personal vinculado al proyecto

1.2.1 Participantes del equipo de trabajo

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
Investigador	Univ. Angelo Cuenca Aguado	Ingeniería Informática	7115279	
Asesor	Ing. Osmar Ugarte Gandarillas	Ingeniero en Sistemas	5513035	

Tabla 2 Participantes del equipo de trabajo

1.2.2 Equipo de trabajo de: Empresas / Instituciones / Organizaciones / Participantes / Cooperantes.

Nombre: Colegio Fe y Alegría “Humberto Portocarrero I”			
Dirección: B/ Loma de San Juan C/ Cochabamba		Telef. Oficina:	
Nombre y Apellidos	Cargo	C.I.	Firma
Lic.: Marco Antonio Uriburu Cortez	Director		

Tec. Mario Perales Ortega	Técnico encargado de las TIC's		
----------------------------------	--------------------------------	--	--

Tabla 3 Participantes del equipo de trabajo

1.2.3 Actividades previstas para los integrantes del equipo de investigación

Responsable	Actividades
Director	• Planificar un cronograma de actividades • llevar a cabo un seguimiento minucioso al cronograma de actividades y a las etapas del proyecto • establecer una comunicación con el equipo de trabajo para una mejor organización • establecer las prioridades del proyecto para asignar recursos • realizar prácticas que aseguren la calidad de los componentes del proyecto • llevar a cabo un informe de la investigación del proyecto • desarrollar el diseño del proyecto • desarrollar los componentes del sistema • realizar un periodo de pruebas para establecer deficiencias en el proyecto • ejecutar la capacitación a los usuarios del proyecto • documentar el proyecto • realizar la presentación final

Tabla 4 Actividades previstas para los integrantes del equipo de investigación

1.3 Descripción del proyecto

1.3.1.1 Resumen ejecutivo

El proyecto tiene como fin que una vez puesto en marcha pasado un tiempo este contribuya al mejoramiento de La administración y control de registro de inscripción del Colegio Fe y Alegría “Humberto Porto carrero I”. Colaborando al mejor manejo de la información ya que con los avances tecnológicos es de gran ayuda unirse para no quedar excluidos de los mismos, y hoy en día sistemas de gestión han contribuido y beneficiado de gran manera a nuestra sociedad y por ende a la institución misma, es por eso que un sistema de gestión de inscripciones para el colegio Fe y Alegría sería de una gran ayuda.

El sistema de gestión estará compuesto por componentes que ayudaran a que el sistema de gestión se lleve a cabo con éxito y de la manera mejor planificada basándose en metodologías de desarrollo y estando dentro de normas universales como son la IEEE que norman y establecen ciertos requerimientos que debe cumplir todo sistema de gestión para el mejor uso de los usuarios.

Para colaborar a que el sistema no parezca de uso difícil se establecerá socializaciones y talleres prácticos para mostrar al usuario que son de sencillez y entendibles en su navegación, es decir el taller estará basado en la usabilidad del mismo.

1.3.2 Descripción, Fundamentación y Descripción del proyecto

El proyecto surge a causa de la falta de tener un sistema de información automatizado, que facilite, mejore y optimice el control de registro de inscripciones del colegio Fe y Alegría. Por otro lado un sistema informático mejorara la imagen proyectada de la institución hacia la sociedad y a la atención de los usuarios.

El proyecto será una colaboración al manejo de la información de las inscripciones, ya que la información que se maneja implica el manejo de varios documentos los cuales muchas veces hace que el proceso de las inscripciones se retrase en días, la

búsqueda de la información en la documentación es un factor muy importante en situaciones de urgencia con lo cual la institución este pasando por muchos problemas a causa de este factor.

El proceso de las inscripciones es algo que se debe de realizar con mayor fiabilidad y rapidez ya que es evidente las largas filas que se registran año tras año en nuestro medio, día tras día para los padres de familia se hace muy incómodo esta situación es por eso que el sistema de gestión ayudara en el proceso de registro de alumnos en el Colegio Fe y Alegría “Humberto Portocarrero I”

El proyecto estará en marcado en dos componentes muy importantes, el primer componente estará referido al desarrollo de un sistema informático, y el segundo componente referido a la socialización, capacitación y manejo practico del sistema desarrollado introduciendo una práctica a lo que son las TIC en la institución Fe y Alegría.

El primer componente donde se desarrolla el sistema informático para la administración de los datos, gestión y control de los mismos en el contexto de inscripción de los alumnos estos ya sean nuevos o antiguos, brindando un servicio de mayor eficacia, de donde se obtendrá reportes en instantes y búsquedas de la misma forma.

El segundo componente está enmarcado en la socialización de las TIC en el colegio Fe y Alegría, dando a conocer la importancia que tienen las TIC en la sociedad y en este caso la importancia que tiene en nuestro contexto donde los beneficiarios serán el plantel administrativo que va desde la cabeza de la institución que es el director, seguido de la secretaria, plantel docente, los alumnos inscritos en la institución. Y así a otras personas como la junta escolar, dándoles a conocer el manejo de las herramientas TIC. El objetivo del desarrollo del sistema es el guardado de los datos de manera eficaz y de manera segura teniendo en cuenta los estándares de la norma

IEEE. Donde el sistema permitirá el llenado de datos la modificación de los mismos y de ser necesario el eliminado de los mismos pero un eliminado parcial no total.

El fin del proyecto se lo vera a largo plazo es decir pasado 3 años desde la ejecución del proyecto. Donde se verá el mejoramiento de inscripción y el control de los mismos donde se obtendrá resultados favorables a la institución y a los usuarios.

1.3.2.1 Análisis de la causa del problema

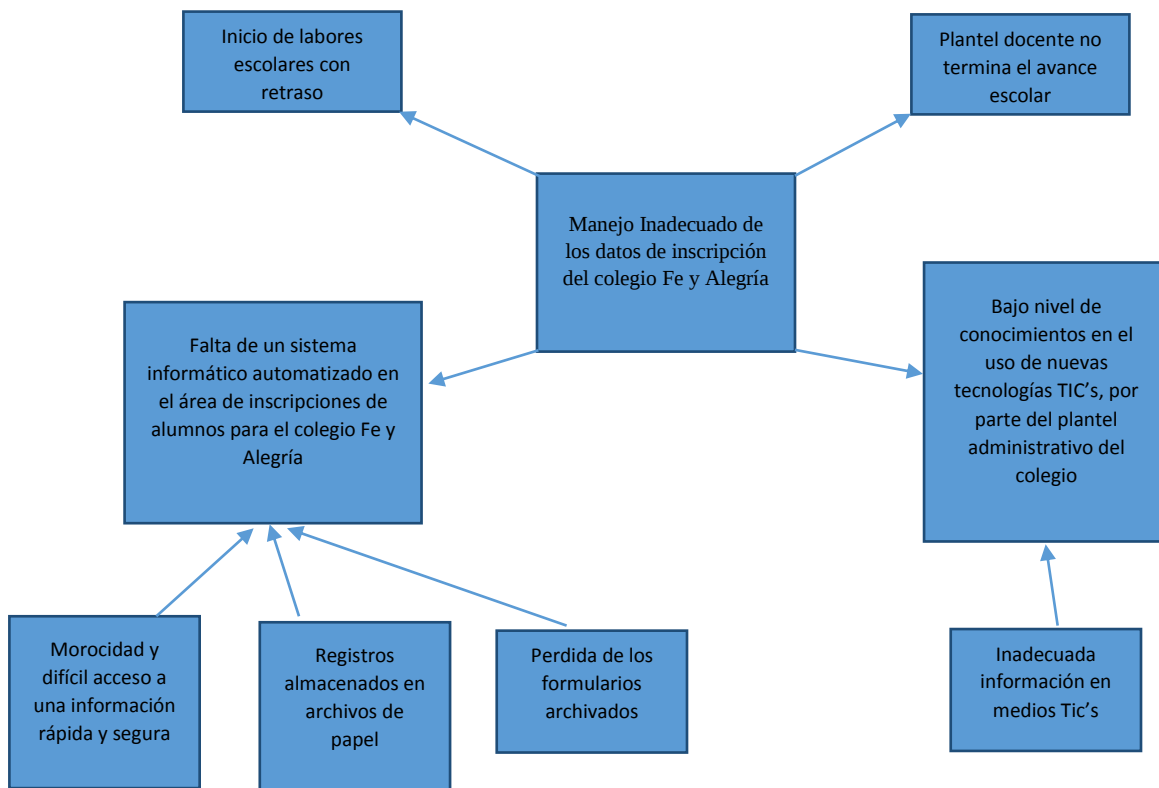
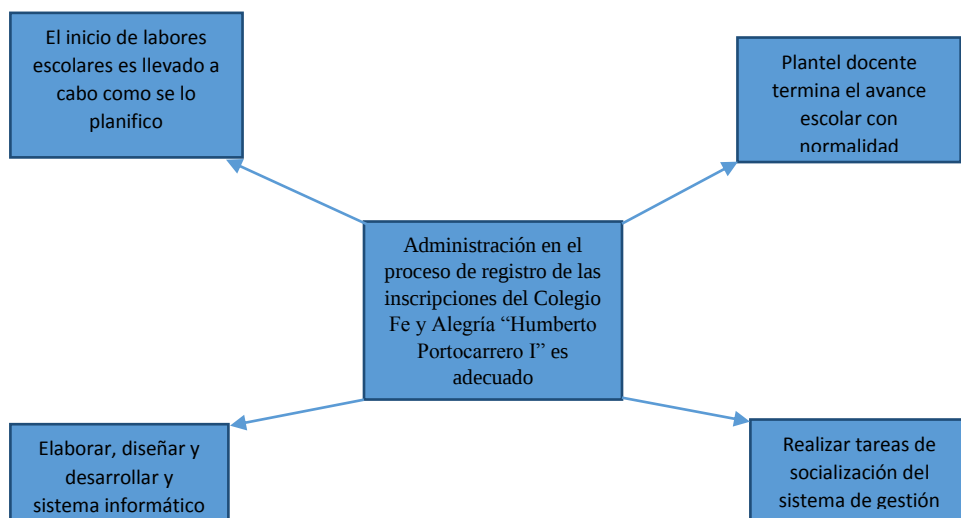


Tabla 5 Arbol de problemas

1.3.2.2 Analisis de objetivos



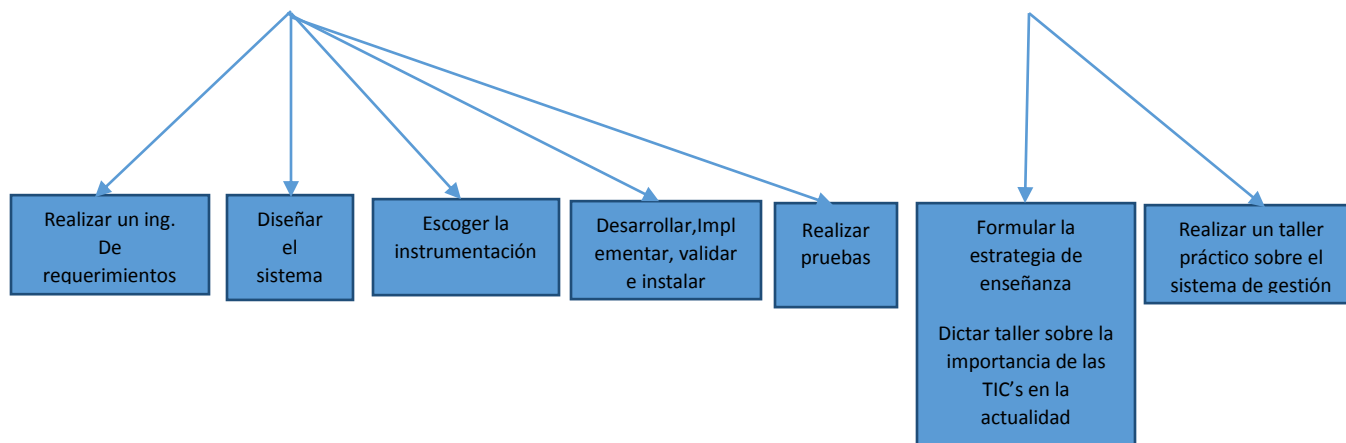


Tabla 6 Arbol de objetivos

1.3.3 Objetivos

1.3.3.1 Objetivo general

Mejoramiento en el Proceso de Administración de las inscripciones del Colegio Fe y Alegría “Humberto Portocarrero I”

1.3.3.2 Objetivos específicos

- Elaborar, diseñar y desarrollar un sistema informático de acuerdo a las necesidades del Colegio Fe y Alegría “Humberto Porto carrero I”
- Realizar tareas de socialización sobres las TIC

1.3.4 Marco lógico del proyecto

Resumen narrativo del proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a mejorar la atención en el registro de alumnos y administración de datos del Colegio Fe y Alegría “Humberto Portocarrero I”.	Luego de dos años de ejecutado el proyecto, al menos un 80% de los alumnos califican como “muy buena” la atención en lo referente al proceso de inscripciones.	Tiempo que toma inscribir a alumnos de un nivel disminuye a comparación del tiempo que toma inscribir a alumnos del mismo nivel en el año base. Resultados de la encuesta anual realizada por la secretaria del colegio.	El presupuesto previsto para la implementación de tecnologías es sostenible para mejoras en las tendencias de las TIC’s.
Objetivo general (propósito) Mejoramiento en el Proceso de Administración de las inscripciones del Colegio Fe y Alegría “Humberto Portocarrero I”.	Al finalizar el proyecto, el proceso de inscripción de alumnos del Colegio Fe y Alegría “Humberto Portocarrero I”, es al menos un 40 % más rápido que las inscripciones del	Cuadro comparativo de la cantidad de inscritos en un día vs. Cantidad de alumnos inscritos vía proceso automatizado avalado por el Director del	1. Los trabajadores de la institución se sienten a gusto con la elaboración del proyecto. 2. El ambiente del colegio es propicio

	año base 2013.	Colegio. Carta de aprobación del docente de taller III.	para el equipamiento. 3. Se dispone recursos económicos y humanos en la elaboración del proyecto.
Objetivos específicos (componentes) 1. Sistema automatizado para las inscripciones del Colegio Fe y Alegría “Humberto Portocarrero I”, desarrollado. 2. Estrategias de socialización del proyecto, implementadas	Al finalizar el proyecto se ha concluido el desarrollo del Sistema informático del Colegio Fe y Alegría de acuerdo a las especificaciones de requerimiento cumpliendo con la norma 830 de IEEE. Al finalizar el proyecto, al menos un 80% del plantel administrativo han participado del programa de capacitación sobre el uso del sistema.	Documentación del Sistema Automatizado en base a la especificación de requisitos de la norma IEEE 830 aprobada por los Docentes de Taller III. Carta de conformidad obtenida por parte del Director del Colegio. Informe de listas que se tomaron durante las sesiones que se hicieron para realizar la capacitación.	1. El personal brinda el apoyo necesario colaborando con los datos que se manejan en la institución. 2. La institución tiene el interés necesario para introducir las TIC's en la institución. 3. El grupo que será capacitado tiene interés en la capacitación. 4. Las actividades se llevan con normalidad sin existir bloqueos.
Actividades			1. Disponibilidad de

2.2.Dictar un taller sobre las ventajas de las TIC's en la actualidad. 2.3.Definir los medios para dictar los cursos del taller. 2.4.Realizar un taller pactico sobre las TIC's con respecto al sistema desarrollado. 2.5.Expender certificados de participación de los cursos sobre las TIC's	TEM	RUBROS	TOTAL (Bs.)	del Director del Colegio.	Fotografías de la capacitación.
		12000 Empleados no Permanentes	11500.-		
		Sub total rubro	11500.-		
	20000	SERVICIOS PERSONALES NO	650.-		
		21000. Servicios Básicos	600.-		
		22000. Servicios de transporte	360.-		
		23000. Alquileres			
		24000. Mantenimiento y reparación	240.-		
		25000. Servicios Profesionales y Comerciales	2000.-		
		Sub total rubro	2310.-		
	30000	MATERIALES SUMINISTROS Y	500.-		
		32000. Productos de Papel, Cartón e Impresos	300		
		Sub total rubro	800.-		
		TOTAL	12850.-		

1.3.5 Metodología

Componente I

Para el desarrollo del siguiente proyecto se utilizara la siguiente metodología.

El proceso Unificado Racional, 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.

- Forma disciplinada de asignar tareas y responsabilidades (quien hace que, cuando y como).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

Fase 1: Preparación inicial (“Incepción”):El objetivo principal es establecer los objetivos para el ciclo de vida del proyecto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber que se cubrirá y delimitar el alcance del proyecto.

- El caso de negocio incluye criterios de éxito, la evaluación de riesgos, 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 ha 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 sus sucesivas iteraciones e incrementos se desarrolla un producto de Software.

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 requeridas.

- Esto incluye:
 - ✓ Operación en paralelo con un sistema anterior que el nuevo sistema este sustituyendo
 - ✓ La conversión de la base de datos operacionales.
 - ✓ Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

El lenguaje unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores solo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos business.

- Diagramas de Secuencia para modelar el paso de mensaje entre objetos.
- Diagramas de Colaboración para modelar iteraciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de actividad para modelar el comportamiento de los casos de uso objetos u operaciones.
- Diagrama de clases para modelar la estructura estática de las clases en el sistema.
- Diagrama de componentes para modelar componentes.
- Diagramas de despliegue.

Componente II

Para el Desarrollo del componente del “Capacitación para el manejo adecuado del Sistema” se tienen como objetivos específicos lo siguiente:

Objetivos Específicos.

En las sesiones se priorizará esencialmente la combinación del método deductivo, inductivo y analítico (DIA) como forma de razonamiento, promoviendo en todo momento la participación activa de los participantes a través del diálogo y la discusión, la técnica expositiva.

Aplicar y Describir Diversas Herramientas de las Tecnologías de Información y

Comunicación (TIC) complementarias para el uso del sistema.

Se hará una descripción completa sobre el manejo óptimo del sistema de informático.

Se aplicará un permanente seguimiento a los participantes para mantener una Participación activa en el programa de capacitación.

Se levantará información de Internet sobre temas relacionados con el desarrollo del uso de las Tics en la Educación.

Se desarrollarán recursos multimedia (videos, imágenes, diapositivas, etc.) según los Contenidos usando las TICs para el proceso de E-A.

Duración.

Duración: 4 horas académicas de 45 minutos la hora académica.

Inicia: el 3 de marzo del 2014.

Horarios.

Horarios: 2 sesiones en la semana, en total 2 sesiones en 1 semana. Cada sesión de 2 horas académicas. Los horarios se coordinan según requerimientos.

Dirigido a: Personal Administrativo del colegio de todas las especialidades.

1.3.6 Cronograma de actividades

Nº	Actividad	Nº Días	Febrero -Marzo	Marzo -Abril	Abril- Mayo - Junio	Junio- Julio- Agosto- Septiembre	Septiembre -octubre- Noviembre - Diciembre	Enero- Febrero -Marzo- Abril
1	Realizar un análisis a la Ing. de requerimientos.	26	✓					
2	Emplear la metodología de UML	24	✓					
3	elaborar un diagrama de clases con los atributos necesarios	12	✓	✓				
4	Elaborar el diseño de la	25		✓				

	base de datos							
5	Diseñar el prototipo del sistema con sus especificaciones	11		✓	✓			
6	Programación del sistema web	90			✓	✓	✓	
7	Realizar la validación	11				✓	✓	
8	Realización de un informe final	13					✓	
9	Análisis de metodologías de socialización y difusión de conocimientos	7					✓	
10	Puesta en práctica la difusión y socialización en el tema de las TIC y entrega de certificados.	5						✓

1.3.7 Resultados esperados

Con el análisis de requerimientos utilizando la norma IEEE 830 aplicado al Colegio Fe y Alegría “Humberto Portocarrero” y con el desarrollo del componente sistema se ayudó a que las inscripciones se lleven a cabo de la manera más correcta utilizando los avances tecnológicos de nuestro medio.

A la culminación del proyecto se capacito a los administradores del sistema de gestión logrando la actualización y comprensión de cada componente del sistema de gestión haciendo ver que es fácil y entendible para el usuario.

Los padres de familia se ven muy contentos ya que el proceso de inscripción se lleva a cabo en menos tiempo a comparación de gestiones pasadas haciendo notar que se encuentran agradecidos ya que se evitan largas filas y mucho tiempo de espera al momento de inscribir a sus hijos.

1.3.8 Transferencia de resultados

a) Medios y estrategias para la transferencia de resultados.

- Presentación del sistema informático en su finalidad a las autoridades universitarias y al director del Colegio Fe y Alegría “Humberto Portocarrero I”
- Socialización del sistema Informático a los usuarios del sistema de gestión.

b) Grupo de beneficiarios de los resultados

El grupo de beneficiarios del proyecto empieza desde la parte administrativa que obtendrán reportes claros, la secretaria se beneficiara en el momento de las inscripciones ya que no se tratara de un trabajo que lleve mucho tiempo ni tampoco será riguroso, los padres de familia de los alumnos pertenecientes a la institución ya no realizaran largas filas para inscribir a sus hijos.

Los participantes y miembros de la institución todos son beneficiados de distintas formas cada uno en su contexto de trabajo.

CAPITULO II

COMPONENTES

1 Capitulo II Componentes

1.1 Componente I: Elaborar, diseñar y desarrollar un sistema informático para el Colegio Fe y Alegría.

1.1.1 Marco teórico

1.1.1.1 Metodología de desarrollo

1.1.1.1.1 Metodología de RUP (Racional Unified Process)

Para el desarrollo del siguiente proyecto se utilizara la siguiente metodología.

El proceso Unificado Racional, es un proceso de desarrollo de software y junto con el lenguaje Unificado de Modelado UML, constituye la metodología estándar mas utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Forma disciplinada de asignar tareas y responsabilidades (quien hace que, cuando y como).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

Fase 1: Preparación inicial (“Incepción”):El objetivo principal es establecer los objetivos para el ciclo de vida del proyecto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber que se cubrirá y delimitar el alcance del proyecto.

- El caso de negocio incluye criterios de éxito, la evaluación de riesgos, 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 ha 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 sus sucesivas iteraciones e incrementos se desarrolla un producto de Software.

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 requeridas.

- Esto incluye:
 - ✓ Operación en paralelo con un sistema anterior que el nuevo sistema este sustituyendo
 - ✓ La conversión de la base de datos operacionales.
 - ✓ Entrenamientos y capacitación de los usuarios y la gente de mantenimiento.

El lenguaje unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas

notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores solo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos business.
- Diagramas de Secuencia para modelar el paso de mensaje entre objetos.
- Diagramas de Colaboración para modelar iteraciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de actividad para modelar el comportamiento de los casos de uso objetos u operaciones.
- Diagrama de clases para modelar la estructura estática de las clases en el sistema.
- Diagrama de componentes para modelar componentes.
- Diagramas de despliegue.

1.1.1.1.2 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.

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.

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.

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.

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).

1.1.1.1.3 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.

1.1.1.1.4 Diagramas de secuencia

Que 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.

1.1.1.1.5 Diagrama de Paquetes

Muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Los Paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. Con estas líneas maestras sobre la mesa, los paquetes son buenos elementos de gestión. Cada paquete puede asignarse a un individuo o a un equipo, y las dependencias entre ellos pueden indicar el orden de desarrollo requerido

1.1.1.1.6 Diagrama de Componentes

Que 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.

1.1.1.1.7 Diagrama de Despliegue

Muestra cómo y dónde se desplegará el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Como los artefactos se ubican en los nodos para modelar el despliegue del sistema, la ubicación es guiada por el uso de las especificaciones de despliegue.

Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

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.

1.1.1.2 Tecnología Utilizada

1.1.1.3 Enterprise Architect

Enterprise Architect combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. Con un gran conjunto de características y un valor sin igual para el dinero, EA puede equipar a su equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos. Verifique el rango completo de las herramientas y características case en detalle. Alta capacidad - Características finales superiores a un precio justo.

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las

etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir

software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad. El manual de usuario está disponible en línea.

Velocidad, estabilidad y buen rendimiento.

El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible. Para una mirada rápida al modelado UML en Enterprise Architect vea nuestro tutorial UML y documentos.

1.1.1.4 Argo UML

Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java.

El Magazine de Desarrollo de Software entrega premios anuales a herramientas de desarrollo de software populares en varias categorías. En 2003 ArgoUML fue una de las finalistas en la categoría "Design and Analysis Tools". ArgoUML recibió un premio "runner-up"(revelación), derrotando a muchas herramientas comerciales.

Sin embargo, desde la versión 0.20, ArgoUML está incompleto. No es conforme completamente a los estándares UML.

1.1.1.5 Postgresql

es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyados por organizaciones comerciales. Dicha comunidad es denominada

ventajas

Seguridad en términos generales -Integridad en BD: restricciones en el dominio - Integridad referencial - Afirmaciones (Assertions) -Disparadores (Tiggers) - Autorizaciones -Conexión a DBMS -Transacciones y respaldos

1.1.1.6 Navicat

Navicat es una de las herramientas más fiables y rápidas para la administración de bases de datos, que te permitirán simplificar la gestión de las bases de datos y reducir los costes de administración.

Diseñado para satisfacer las necesidades de los administradores de bases de datos, desarrolladores y pequeñas y medianas empresas.

Navicat dispone de una interfaz gráfica muy intuitiva, que le permitirá crear, organizar, acceder y compartir información de forma fácil y segura.

Navicat es muy conocido, de confianza, y se usan a diario en todo el mundo por las empresas globales, organismos gubernamentales e instituciones educativas. Desde comienzos de 2001, Navicat ha sido descargado más de 2.000.000 de veces en todo el mundo y tiene una base de clientes de más de 50.000 usuarios.

Ademas está ahora disponible en 7 idiomas Esta disponible para MySQL, Oracle y PostgreSQL, para administración/desarrollo local y remoto.

2.1.1.4. Tomcat

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pagés (JSP) de Sun Microsystems Podemos dividir los contenedores de Servlets en:

Contenedores de Servlets Stand-alone (Independientes) Estos son una parte integral del servidor web. Este es el caso cuando usando un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por iPlanet). Éste el modo por defecto usado por Tomcat.

Sin embargo, la mayoría de los servidores, no están basados en Java, lo que nos lleva a los dos siguientes tipos de contenedores:

Contenedores de Servales dentro-de-Proceso

El contenedor Servales es una combinación de un plumín para el servidor web y una implementación de contenedor Java. El plumín del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un serles, el plumín toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un solo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad.

Contenedores de Servlets fuera-de-proceso El contenedor Servlets es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP).

Si una cierta petición debería ejecutar un servlets, el plugin toma el control sobre la petición y lo pasa al contenedor Java. El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

1.1.1.6.1 HTML5

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como

HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).^{1 2} Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas. Al no

ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

1.1.1.6.2 Prototype

Prototype es un framework escrito en JavaScript que se orienta al desarrollo sencillo y dinámico de aplicaciones web. Es una herramienta que implementa las técnicas AJAX y su potencial es aprovechado al máximo cuando se desarrolla con Ruby On Rails. Con la Web 2.0 las técnicas de desarrollo de páginas web necesitaban dar un gran salto. Con esto en mente nació la técnica AJAX, que gracias a Prototype permite el desarrollo ágil y sencillo de páginas Web, esto en relación al desarrollador, y provee al cliente una manera más rápida de acceder al servicio que solicita. Prototype es un Framework basado en JavaScript orientado a proporcionar al desarrollador de técnicas AJAX listas para ser usadas. El potencial de Prototype es aprovechado al máximo si se desarrolla con Ruby On Rails, esto no quiere decir que no se puede usar desde otro lenguaje, solamente que demandara un “mayor esfuerzo” en el desarrollo.

Técnica AJAX.

Antes de comenzar a desarrollar Prototype, es necesario presentar el concepto AJAX, debido a que Prototype emplea AJAX.

AJAX proviene de Ashynchronous JavaScript And XML. En pocas palabras AJAX es una técnica de desarrollo Web que incorpora varias tecnologías, como son el JavaScript y XML, consiguiendo de esta manera una forma de navegar rápida, ágil y dinámica.

1.1.1.6.3 Eclipse

Es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización.

Asimismo, a través de "plugins" libremente disponibles es posible añadir control de versiones con Subversion. e integración con Hibernate.

1.1.1.6.4 JQuery

jQuery es una biblioteca o framework 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 tecnología AJAX a páginas web.

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. jQuery usa las licencias MIT y GPL permitiendo su uso en proyectos libres y privativos.

1.1.1.6.5 Modelo Vista Controlador (MVC)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Para el diseño de aplicaciones con sofisticados interfaces se utiliza el patrón de diseño Modelo-Vista-Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error.

A todo esto, se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Los elementos del patrón son tres:

Modelo: datos y reglas de negocio

Vista: muestra la información del modelo al usuario

Controlador: gestiona las entradas del usuario

1.1.1.7 Base de Datos

Componentes principales de una Base de Datos Los principales componentes de una base de datos son: 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 y el Administrador de la Base de Datos quien se encarga del control general del Sistema de Base de Datos.

1.1.1.7.1 Herramienta PostgreSQL 9.1 para el manejo de la BD

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. PostgreSQL es una derivación libre (OpenSource), y utiliza el lenguaje SQL92/SQL99, así como otras características. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Algunas de las características para hacer uso de este gestor se mencionan a continuación:

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

Soporte para una amplia variedad de tipos nativos. PostgreSQL provee nativamente soporte para: Números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, Arrays. Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son

los tipos de datos GIS creados por el proyecto PostGIS. Consta con un lenguaje propio llamado [PL/PgSQL (similar al PL/SQL de oracle), pero también soporta los lenguajes C, C++, Java PL/Java web, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, L/Scheme.

PostgreSQL es un magnífico gestor de bases de datos. Tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa GPL para el presente proyecto.

Postgres cuenta con Herramientas de Administración como es PgAdmin3 y PgAccess:

Entorno de escritorio visual, PhpPgAdmin y psql que es un Cliente de consola.

VENTAJAS

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Footprint bajo de memoria, bastante poderoso con una configuración adecuada.

- Multiplataforma.
- Capacidades de replicación de datos.
- Soporte empresarial disponible.

DESVENTAJAS

- Sin experticia, configurar llega a ser un caos.
- Es fácil de vulnerar sin protección adecuada.
- El motor MyISAM es instalado por defecto y carece de capacidades de integridad relacional.
- InnoDB genera mucho footprint en memoria al indexar.
- El toolset empresarial tiene un costo adicional por suscripción anual.
- Realizar revisiones llegar a ser una labor manual y tediosa para el DBA.
- Reducida cantidad de tipos de datos.

1.1.2 Plan de desarrollo del software

1.1.2.1 Introducción

El presente documento es una especificación de requisitos de software (ERS) para el sistema de gestión “Mejoramiento registro y control de inscripciones en el Colegio Fe y Alegría Humberto Portocarrero I”, este documento está elaborado cuidadosamente en base a normas especificadas por la IEEE.

El documento está realizado a base de la colaboración de los grupos involucrados con la institución “Colegio Fe y Alegría (Humberto Portocarrero I)”

La documentación es una fuente de requerimientos los cuales están divididos en dos partes que son los requisitos funcionales y los requisitos no funcionales.

Específicamente la documentación presentada está basada en la norma IEEE 830 Recommend Practice for Software Requirements Specifications. ANSI / IEEE Std 830-1998.

1.1.2.1.1 Propósito

El objetivo del sistema de gestión es “Automatizar el registro y control de las inscripciones de estudiantes del colegio Fe y Alegría”, esto por medio de un sistema informático de control y registro para los estudiantes.

Con el sistema se pretende que sea el comunicador entre estudiantes y miembros administrativos de la institución.

1.1.2.1.2 Alcance

El sistema debe registrar los datos de los alumnos que desean pertenecer al colegio, de la misma forma dar de baja o a la vez realizar modificaciones a los registros que se ingresó al sistema.

El sistema debe registrar los datos de los turnos así también el de los periodos de cada turno, pudiendo estos datos ser modificados o en su defecto dado de baja.

El sistema debe permitir el registro de las profesiones de cada profesor, pudiendo estos datos tener la oportunidad de ser modificados o dados de baja si así lo requiere el usuario.

El sistema debe registrar los datos de el/los apoderado del alumno inscrito en el colegio, pudiendo modificar dichos datos o dar de baja al apoderado.

El sistema debe de registrar los datos de los profesores miembros del colegio pudiendo dar de baja o modificar los datos de los mismos.

El sistema debe permitir el control de la cantidad de alumnos en cada uno de los grupos al momento de asignar un alumno a un grupo, permitiendo la modificación de alumnos en un grupo.

El sistema deberá registrar los datos de las aulas pertenecientes al colegio pudiendo hacer una modificación o dar de baja a un aula.

El sistema deberá permitir la búsqueda de datos de los alumnos pertenecientes al colegio.

El sistema deberá realizar un historial de los alumnos que pertenecen al colegio. Así también el sistema deberá permitir gestionar los niveles de estudio.

El sistema debe permitir la asignación de materias a los profesores, previa asignación de profesiones a los profesores, pudiendo ser estos datos modificados o dados de baja, una vez realizada esta asignación debe permitir la primer asignación ya mencionada teniendo la opción de ser modificada dicha asignación así como también dada de baja dicha asignación.

El sistema permitirá asignación de aulas y grupos a los profesores como así también a los estudiantes permitiendo modificar dichas asignaciones.

El sistema permitirá sacar reportes de inscritos en el colegio.

El sistema deberá permitir la asignación de materias a los grados, dando la oportunidad de modificar y dar de baja las asignaciones registradas.

El sistema debe mostrar reportes de diferentes índoles como lo requiera el usuario, reportes como:

- Lista de alumnos por grupos
- Lista de los alumnos según el sexo por grupos.
- Lista de los alumnos reinscritos
- Lista de los alumnos inscritos por primera vez en el colegio
- Lista de los alumnos según los servicios con los que cuenta el alumno
- Lista de los alumnos según los medios de transportes que utilizan para asistir al colegio

- Lista de los alumnos según los medios de comunicación con los que cuentan los alumnos
- Lista de los alumnos según la asistencia a los centros de salud
- Lista de los alumnos que cuenten o no cuenten con alguna discapacidad
- Lista de los alumnos inscritos en todos los niveles
- Lista de los alumnos que cuenta cada materia
- Lista de los profesores que trabajan en el colegio
- Lista de los profesores según los grados, niveles y grupos
- Lista de los profesores según las materias dictadas en los grupos
- Lista de los profesores según sus profesiones estudiadas
- Lista de las materias que se dictan en cada grados
- Lista de los alumnos registrados en los grupos según gestiones

1.1.2.2 Vista General del Proyecto

1.1.2.2.1 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 Colegio Fe y Alegría.

1.1.2.2.1.1 Propósito

La presente propuesta de sistema tiene por objeto hacer más eficiente las actividades dentro del proceso de inscripción, registro de Alumnos, Registro de Profesores,

Solicitudes Generación de Reportes cuando se lo requiera, además de otorgar a los usuarios un registro adecuado de la información: teniendo toda información almacenada en una Base de Datos a la que solo podrá acceder personal autorizado para consultar los datos que más concierne a la institución y con los cuales contara el sistema informático.

1.1.2.2.1.2 Objetivos

1.1.2.2.1.2.1 Objetivo General

Mejoramiento en el proceso de Administración de las inscripciones del Colegio Fe y Alegría “Humberto Portocarrero I”.

1.1.2.2.1.2.2 Objetivos Específicos

Según el estudio previamente realizado para esta institución nos planteamos los siguientes objetivos:

- Otorgar a los usuarios total seguridad en el resguardo de su información, permitiendo el acceso sólo a usuarios autorizados.
- Obtener un sistema que permita generar Reportes de manera Rápida y Sencilla.
- Crear una base de datos de acuerdo a las necesidades de la institución.
- Diseñar una interfaz de usuario de fácil manejo.
- Desarrollar Una Aplicación web Escalable.
- Desarrollar Casos de Prueba que garanticen la calidad del Software.

1.1.2.2.1.3 Alcance

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.

Opciones que permitan el registro adecuado de los estudiantes, de los profesores, las asignaciones que requiere el Colegio. Así como también no hay que olvidarse que se requiere de datos estadísticos que son los reportes los cuales necesita el colegio y por lo cual lleva una gran deficiencia.

Desarrollar un manual de usuario y de instalación del sistema.

1.1.2.2.2 Suposiciones y Restricciones

1.1.2.2.2.1 Suposiciones

Suponemos que el Colegio Fe y Alegría cuenta con:

- La disponibilidad de otorgar toda la información necesaria para la creación del Software.
- Equipo de computación. En el que incluye un computador, impresora.
- Interés de la institución para utilizar el Sistema
- Personal con conocimientos básicos de computación.
- Cuenta con equipos para implementar el sistema.

1.1.2.2.2.2 Restricciones

Establecemos las siguientes restricciones para el 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.
- Deberá contar como mínimo con un sistema operativo Windows XP / Linux.
- El sistema deberá ejecutarse mínimamente en navegadores Mozilla 14 o superior, Chrome 9 o superior, Opera 9 o superior.

- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- No se incluirá ningún reporte contable.
- El sistema no abarca el control del Personal de la Institución.

1.1.2.2.3 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.

1.1.2.2.3.1 Plan de desarrollo del Software.

El plan de desarrollo del Software se describirá paso a paso en los siguientes puntos del proyecto en Base a la Metodología RUP.

1.1.2.2.3.2 Diagrama de Clases

1.1.2.2.3.3 Modelo de Casos de Uso del Sistema

1.1.2.2.3.3.1 Casos de Usos del Sistema General



Figura nº 1 Casos de Usos del Sistema General

Caso de uso de Acceder al sistema

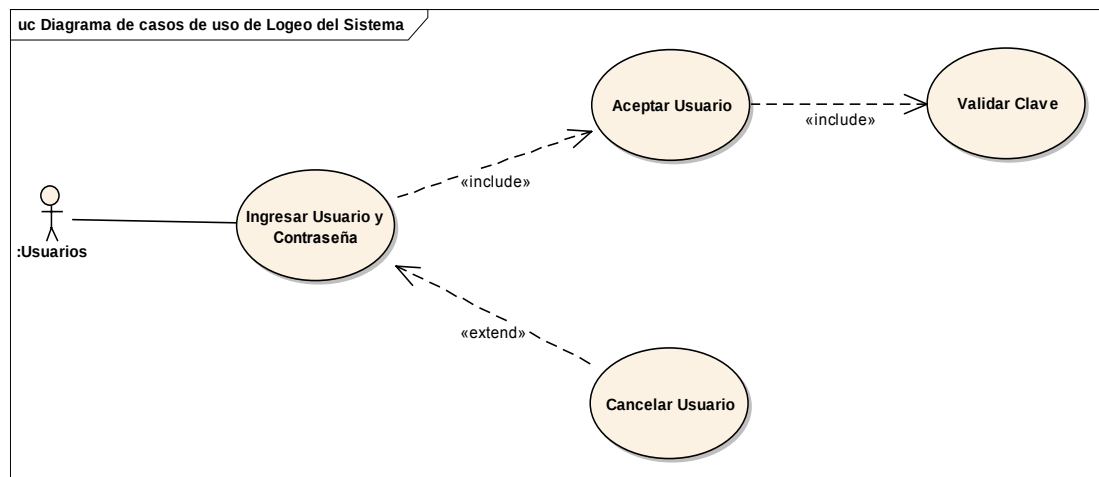


Figura nº 2 Caso de uso de Acceder al sistema

Caso de uso Registrar Usuarios

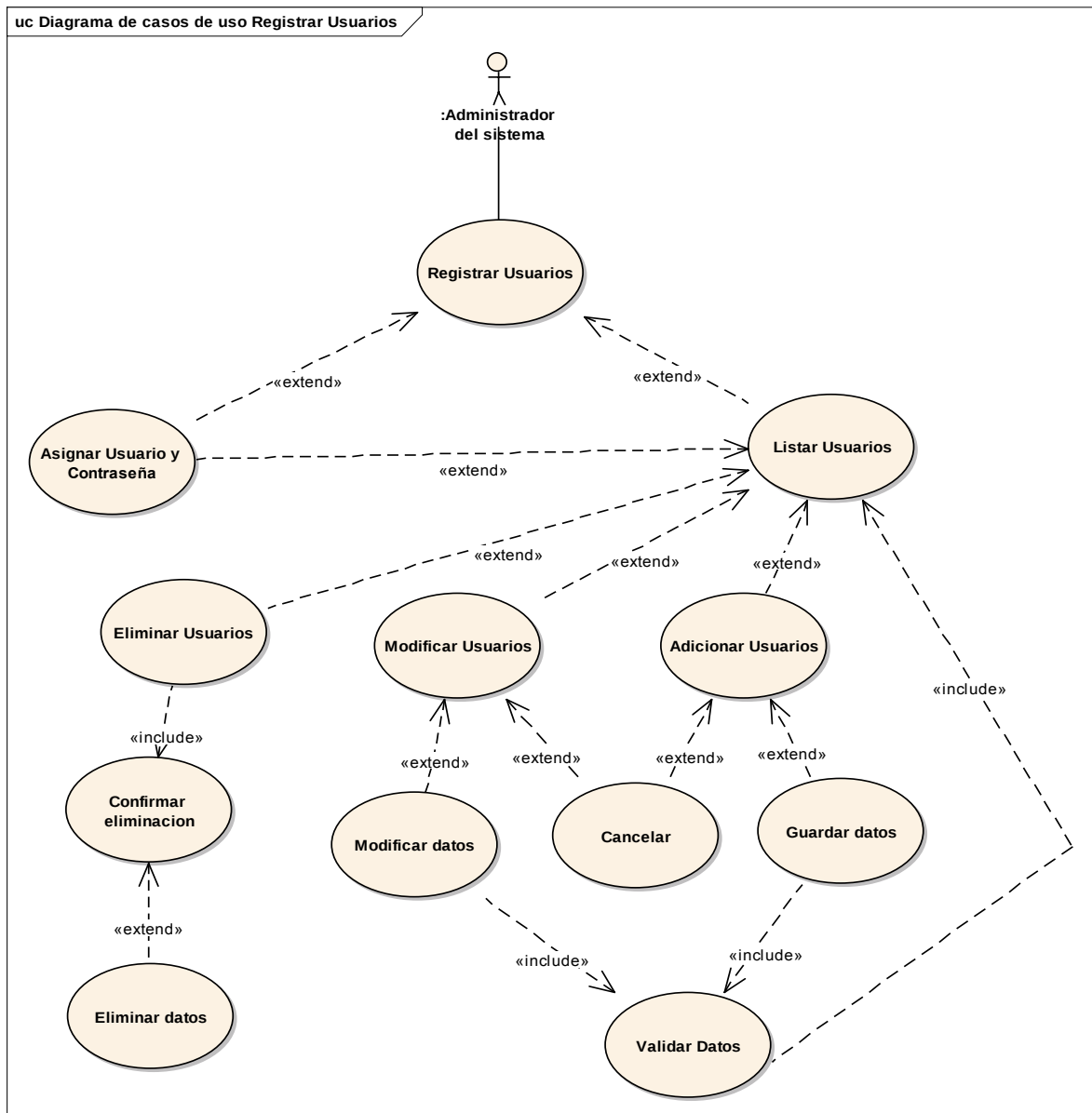


Figura nº 3 Caso de uso Registrar Usuarios

Caso de uso Registrar Alumnos

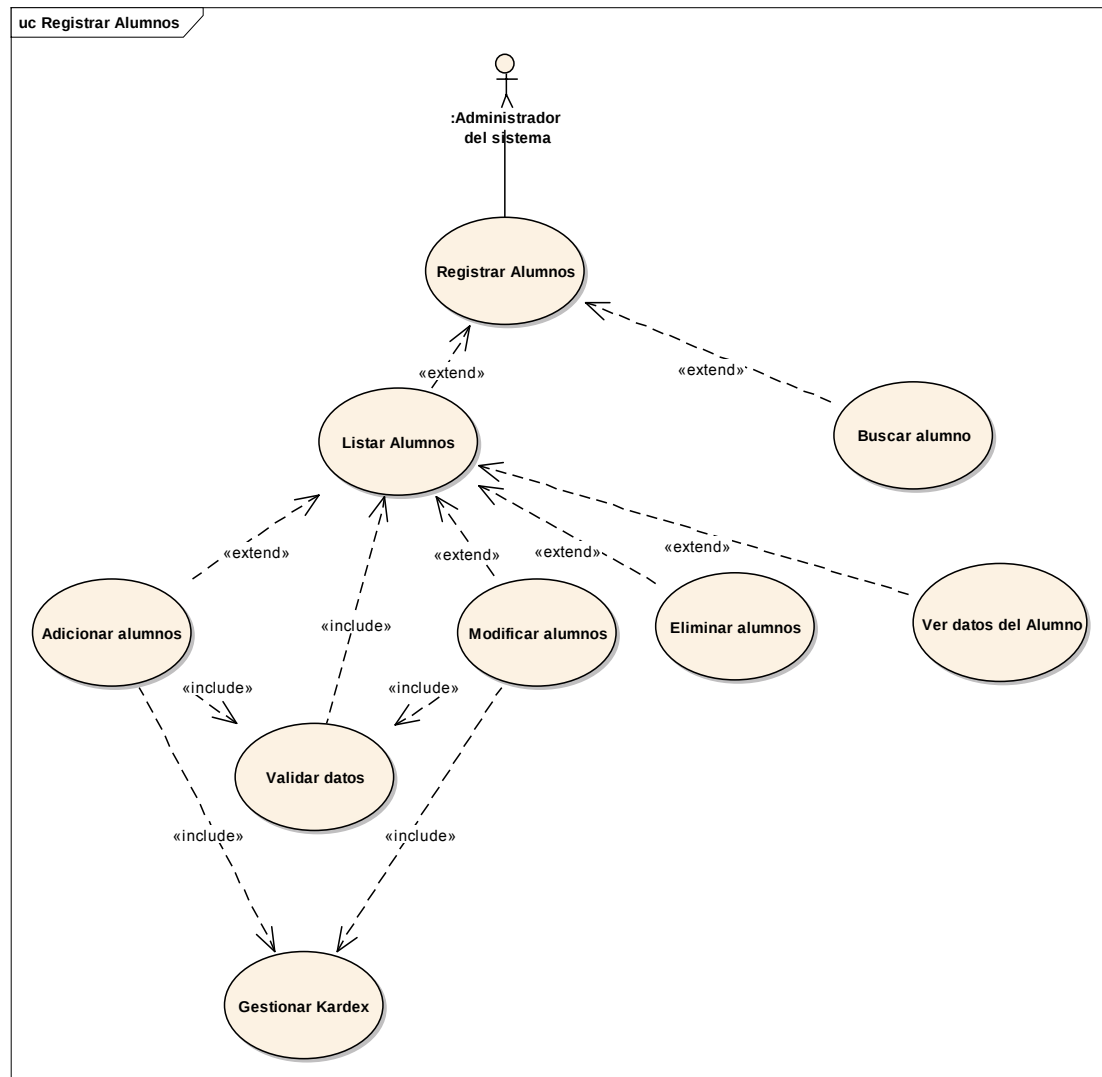


Figura nº 4 Caso de uso Registrar Alumnos

Caso de Uso Gestionar Grupos

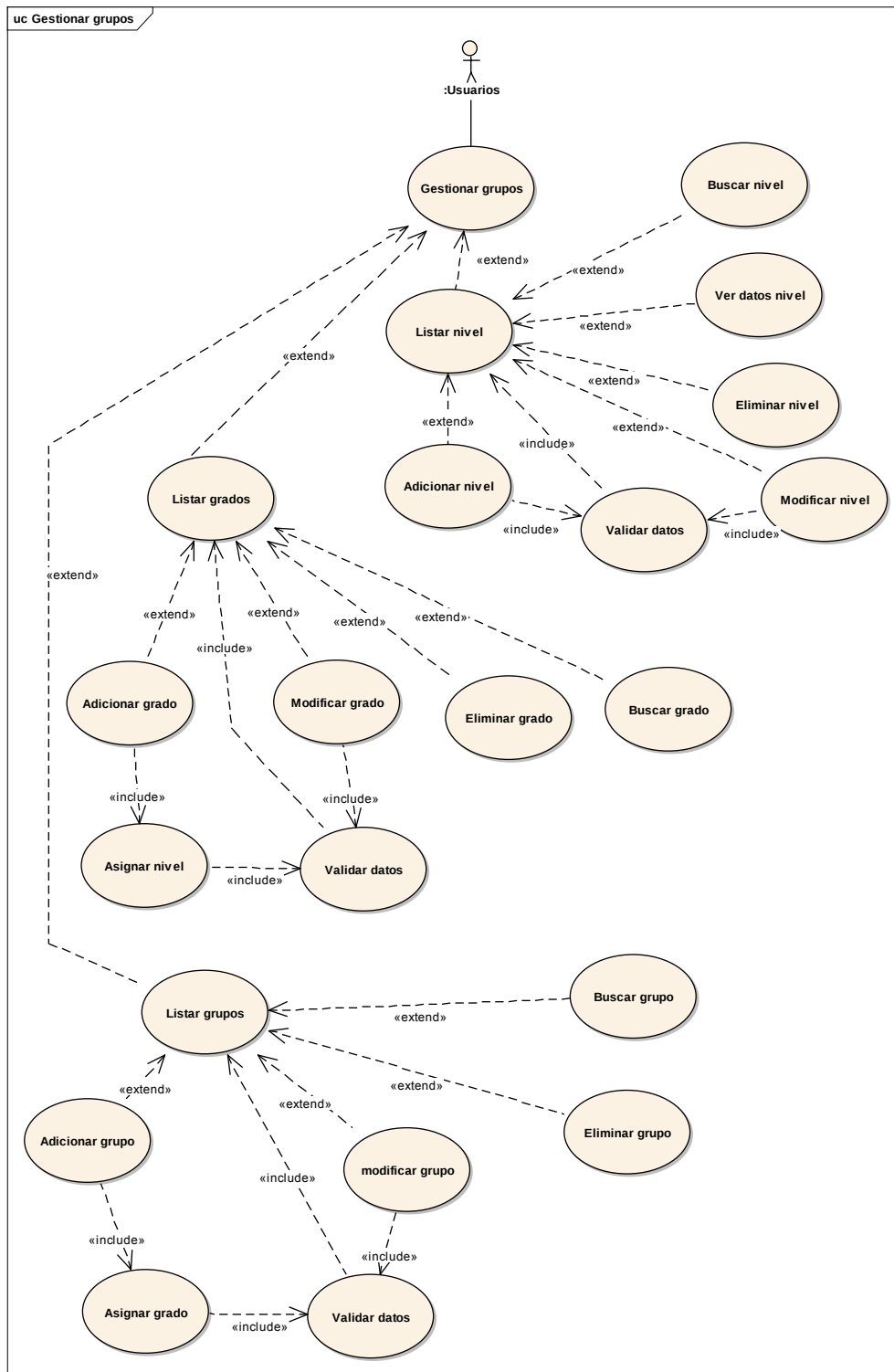


Figura nº 5 Caso de uso Gestionar grupos

Caso de uso Gestionar profesores

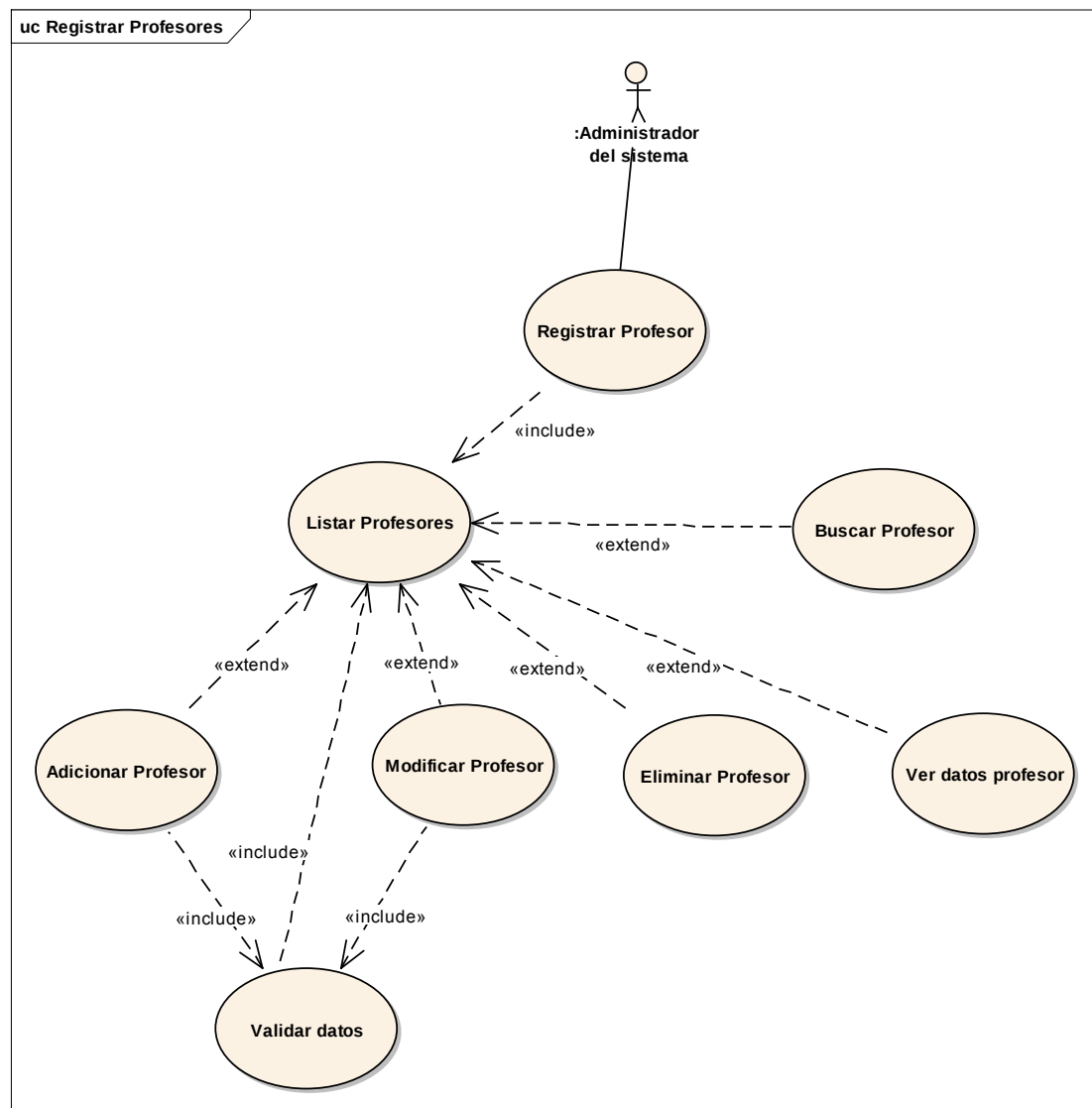


Figura nº 6 Caso de uso Gestionar profesores

Caso de uso gestionar materias

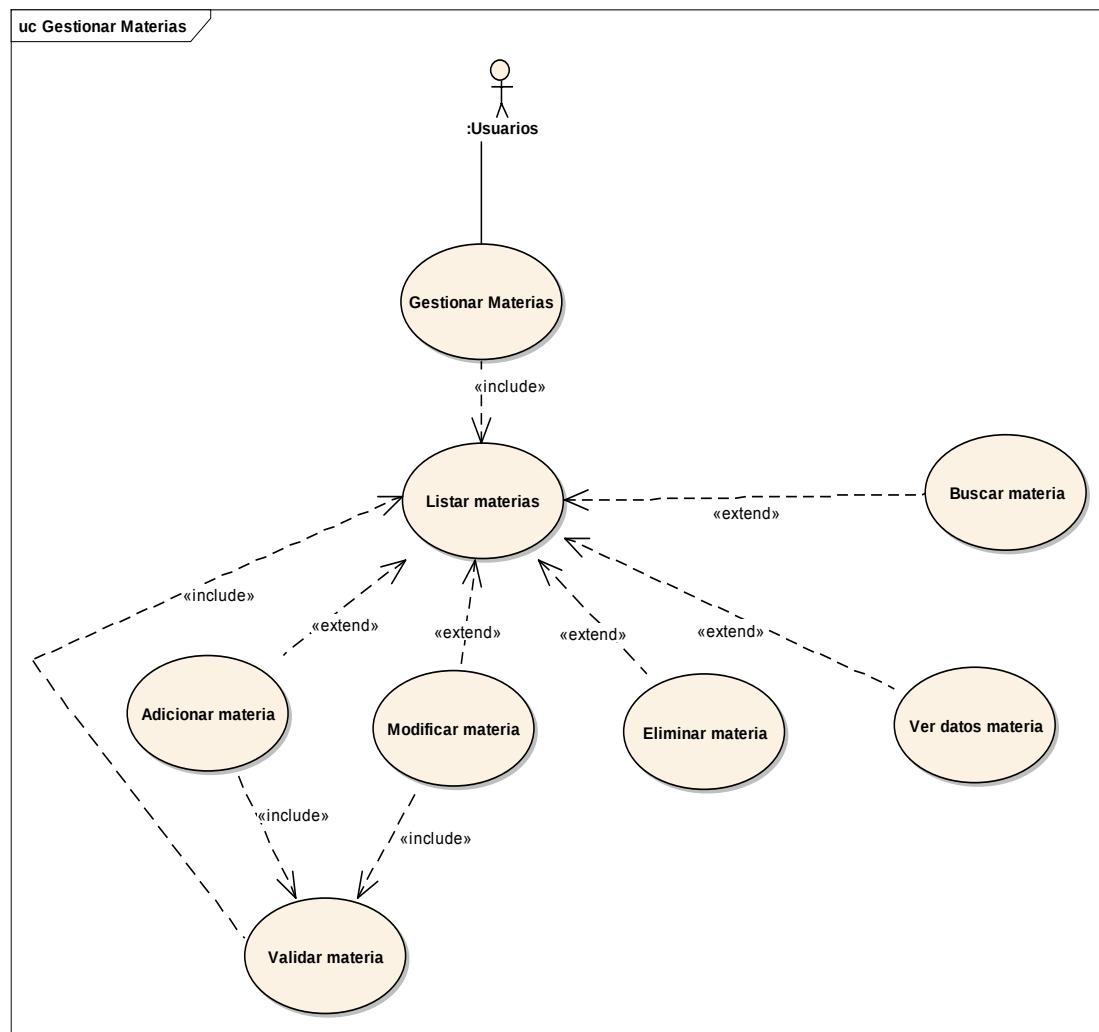


Figura nº 7 Caso de uso gestionar materias

Caso de uso asignar materia a los grados

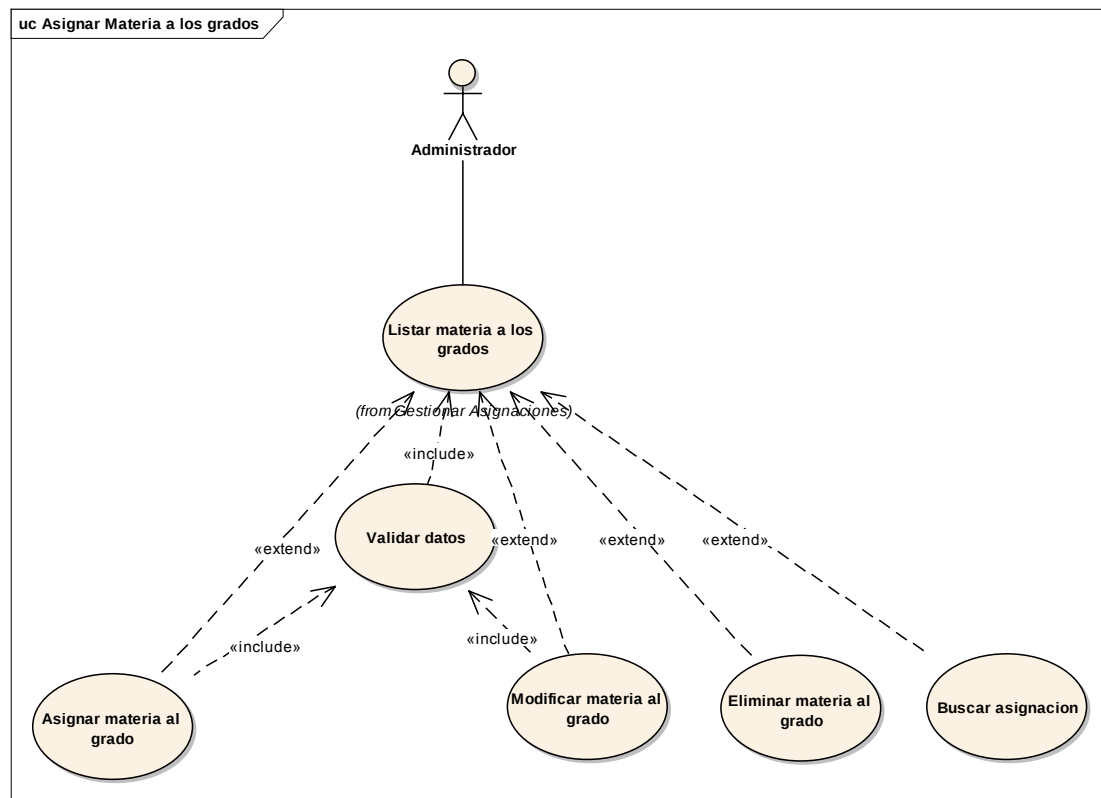


Figura nº 8 Asignar materia a los grados

Caso de uso asignar materia a profesor

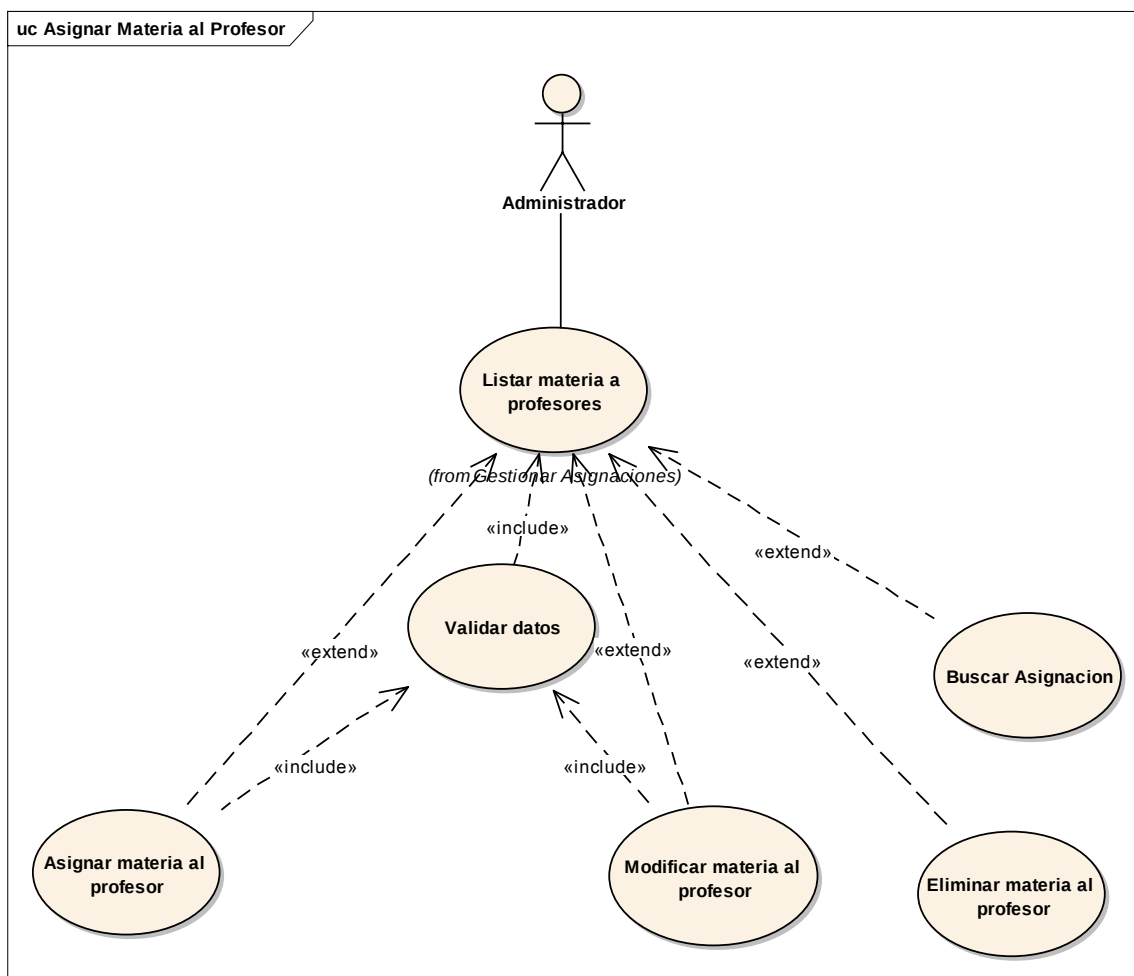


Figura nº 9 Asignar materia a profesor

Caso de uso asignar profesor a grupo

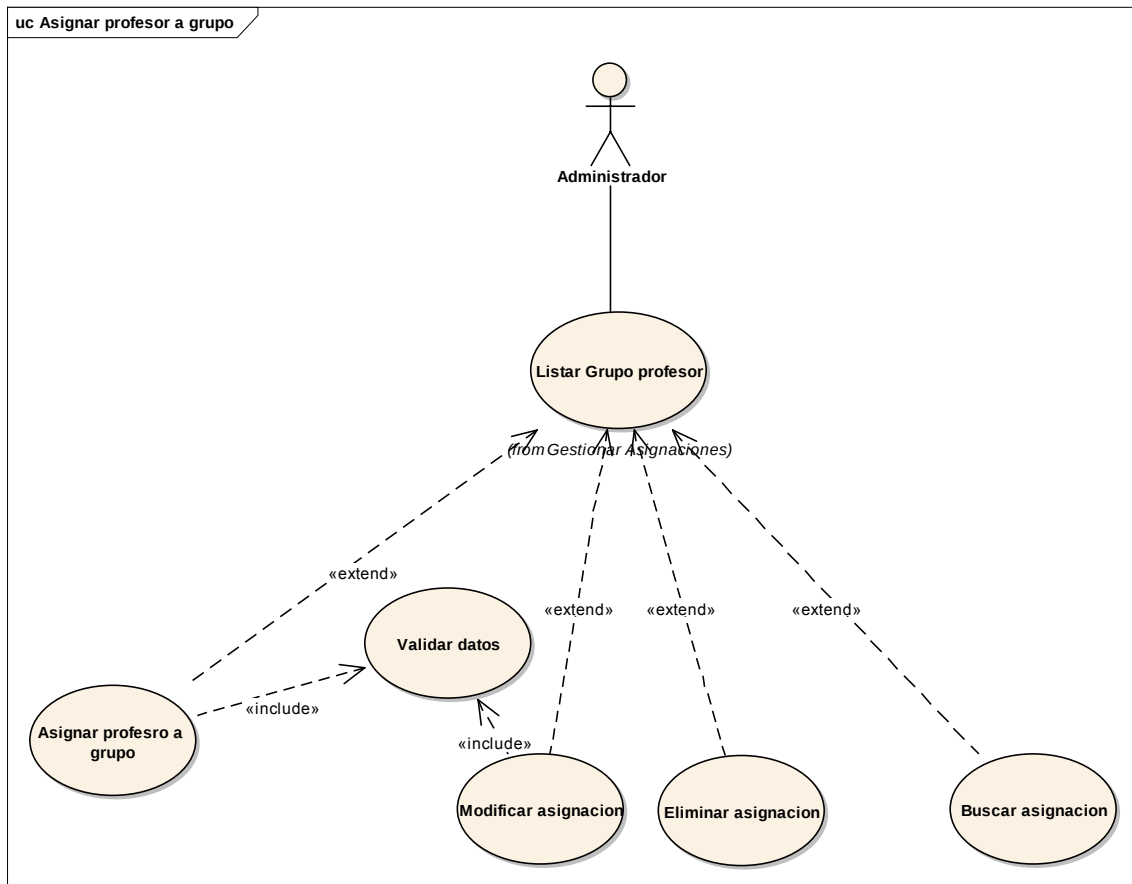


Figura nº 10 Asignar procesos a rol

Caso de uso Gestionar reportes

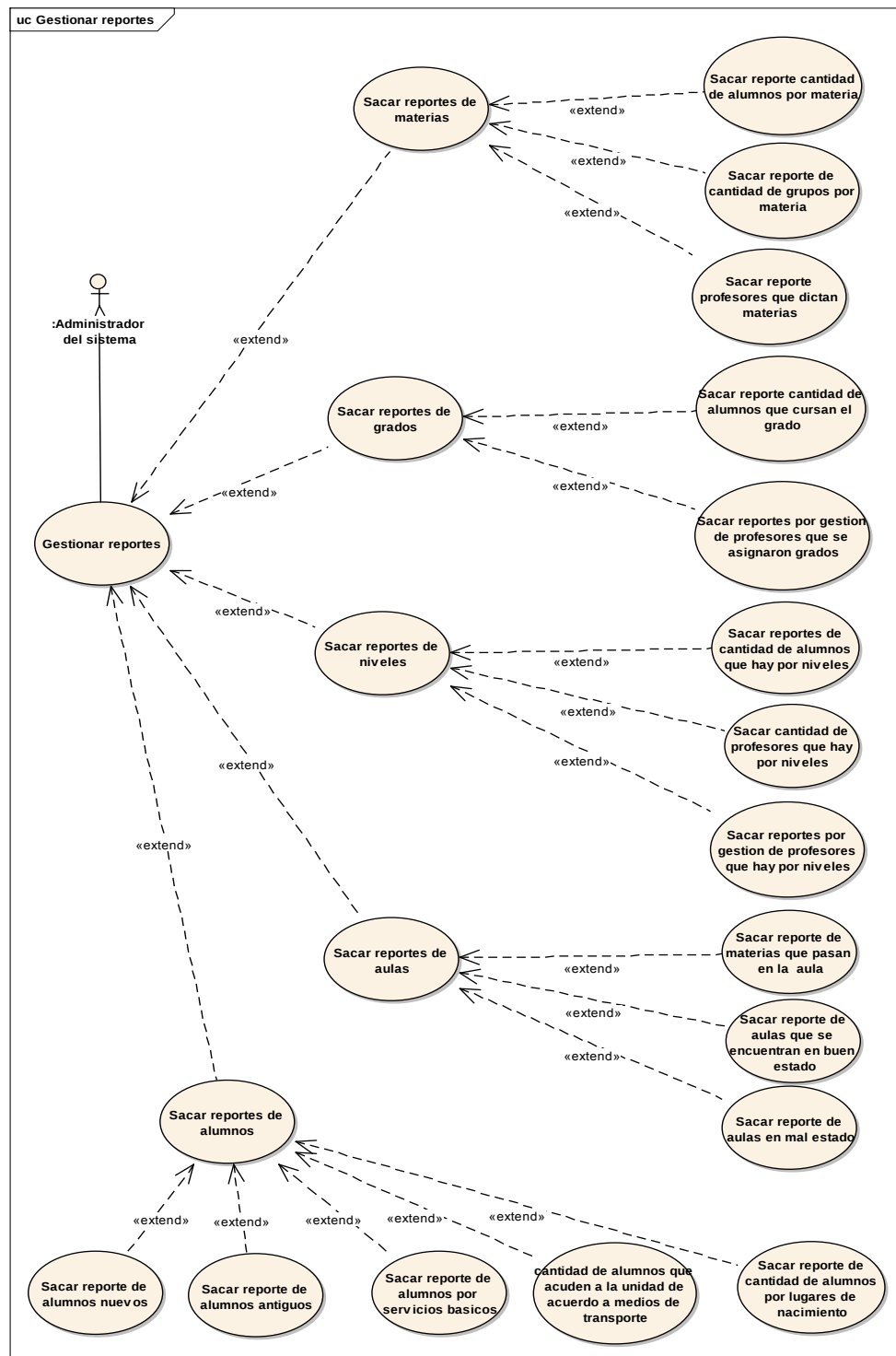


Figura nº 11 Caso de uso Gestionar reportes

1.1.2.2.3.14 Especificación de Casos de Uso

Especificación del caso de uso Acceder al sistema

Descripción:El sistema “SISTEMFYA” deberá permitir el ingreso al menú, a cualquier

usuario que ingrese correctamente sus códigos de usuario y clave en la pantalla inicial.
Actores • Usuario de “SISTEMFYA” (Administrador, Secretaria).
Precondiciones El usuario debe haber ingresado a la pantalla principal.
Flujo Normal 1. El usuario ingresa los códigos de usuario y clave. 2. El usuario presiona el botón Aceptar. 3. El “SISTEMFYA” valida datos ingresados verificando que ninguno de los dos falte. 4. El “SISTEMFYA” busca la lista de usuarios habilitados cuyos códigos de usuario y clave deben coincidir con los códigos ingresados. 5. El “SISTEMFYA” sesiona al usuario identificado en la búsqueda.
Flujo Alternativo 1. El usuario decide cerrar el sistema. 2. El “SISTEMFYA” encuentra datos faltantes y pide el ingreso de estos. 3. El “SISTEMFYA” no encuentra al usuario o este está deshabilitado, entonces visualiza un mensaje.
Pos condiciones El “SISTEMFYA” visualiza el menú según el usuario encontrado en la búsqueda.

Tabla 1 Especificación del caso de uso Logeo del sistema

Especificación del Caso de Uso Registrar Usuarios

Descripción:El sistema “SISTEMFYA” deberá permitir al administrador el registrar Usuarios.

Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones El usuario debe haber ingresado a la pantalla principal.
Flujo Normal 1. El usuario tendrá que ingresar a registrar usuarios. 2. El usuario en ingresar a registrar usuarios tendrá las opciones a: 2.1 Asignar usuario y contraseña 2.2 Listar usuarios, en esta parte el usuario también podrá asignar usuario y contraseña como también las opciones de : a) Adicionar Usuario con las opciones de Cancelar y Guardar datos para luego validarlos. b) Modificar Usuario con las opciones de Eliminar y Modificar datos, para luego hacer la validación de los mismos . c) Eliminar Usuario en esta parte se deberá confirmar la eliminación y tener la opción de eliminar los datos.
Flujo Alternativo 1. El usuario decide cerrar el sistema. 3. El “SISTEMFYA” no encuentra al usuario o este está deshabilitado, entonces visualiza un mensaje.
Pos condiciones El “SISTEMFYA” visualiza el menú según el usuario encontrado en la búsqueda.

Tabla 2 Especificación del Caso de Uso Registrar Usuarios

Especificación del Caso de Uso: Registrar Alumnos

Descripción:El sistema “SISTEMFYA” deberá permitir al usuario o administrador registrar Alumnos.

Actores ☐ Usuario del “SISTEMFYA”(Administrador)
Precondiciones El usuario debe haberse sesionado en “SISTEMFYA”. Seleccionar Registrar Alumnos en el menú principal de “SISTEMFYA”.
Flujo Normal 1. El usuario tendrá dos opciones. ☐ Ver el listado de los Alumnos . Buscar Alumnos 2. Cuando se selecciona listado de alumnos el usuario tendrá las opciones de: a) Adicionar Alumno. b) Modificar Alumno. c) Eliminar Alumno d) Ver datos del Alumno 3. El usuario una vez adicionado el Alumno tendrá que Gestionar el Kardex y hacer la validación de datos. 4. El usuario una vez modificando los datos del Alumno tendrá que Gestionar el Kardex y hacer la validación de datos. 5. El usuario deberá validar los datos.
Flujo Alternativo 1. El usuario elige opción salir (sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 3 Especificación del Caso de Uso: Registrar Alumnos

Especificación del Caso de Uso: Gestionar Grupos

Descripción: El “SISTEMFYA” deberá permitir al usuario gestionar Grupos
Actores ☐ Usuario de “SISTEMFYA” (Administrador).
Precondiciones ☐ El usuario debe haberse sesionado ☐ El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar asignaciones.
Flujo Normal El usuario tendrá las opciones de : 1.Listar nivel , el usuario tendrá las opciones de: a) Adicionar nivel, seguidamente validar los datos. b) Buscar nivel c) Modificar nivel, seguidamente validar los datos. d) Eliminar nivel e) Ver datos nivel 2. Listar grados, el usuario tendrá las opciones de: a) Adicionar grado y luego tendrá que validar los datos. b) Modificar grado y luego tendrá que validar los datos modificados. c) Buscar grado d) Eliminar grado 3. Listar grupos , el usuario tendrá las opciones de: a) Adicionar grupo, como también podrá asignar grado, luego el usuario tendrá que validar los datos.

b) Modificar grupo, el usuario tendrá que validar los datos modificados. c) Buscar grupo d) Eliminar grupo
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 4 Especificación del Caso de Uso: Gestionar Grupos

Especificación del Caso de Uso: Registrar Profesores

Descripción: El sistema “SISTEMFYA” deberá permitir al usuario o administrador registrar Profesores.
Actores ☐ Usuario del “SISTEMFYA”(Administrador)
Precondiciones El usuario debe haberse sesionado en “SISTEMFYA”. Seleccionar Registrar Profesores en el menú principal de “SISTEMFYA”.
Flujo Normal 1. El usuario realiza lo siguiente. ☐ Ver el listado de los Profesores . Validar datos 2. El usuario tiene las siguientes opciones.

a) Adicionar Profesor b) Modificar Profesor. c) eliminar Profesor d) Ver datos Profesor c) Buscar Profesor 3. El usuario una vez adicionado el profesor tendrá que realizar la validación de datos. 4. El usuario una vez modificando los datos del profesor tendrá que validar los datos.
Flujo Alternativo 1. El usuario elige opción salir (sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones.

Tabla 5 Especificación del Caso de Uso: Registrar Profesores

Especificación del Caso de Uso: Gestionar Materias

Descripción: El “SISTEMFYA” deberá permitir al usuario gestionar materias
Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones • El usuario debe haberse sesionado • El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar materias.

Flujo Normal 1. El usuario tendrá que : • Listar Materias. 2. El usuario en listar Materias tendrá las opciones de: a) Adicionar Materia, luego validamos los datos de materia. b) Modificar materia, luego validamos los datos de materia. c) Eliminar materia d) Ver datos materia e) Buscar materia 3) El usuario deberá validar los datos de materia.
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 6 Especificación del Caso de Uso: Gestionar Materias

Especificación del Caso de Uso: Gestionar Asignación Materias a los grados

Descripción:El “SISTEMFYA” deberá permitir al usuario gestionar asignaciones de materias a los grados
Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones

• El usuario debe haberse sesionado • El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar asignaciones
Flujo Normal El usuario tendrá las opciones de : 1. Listar asignar materias a los profesores. 2. Adicionar asignación, seguidamente el usuario aceptara la asignación para la validación de los datos. 3. Modificar asignación, seguidamente el usuario aceptara la modificacion para la validación de los datos. 4. Eliminar asignación, el usuario tendrá que aceptar eliminación de la asignación para luego validar. 5. Buscar asignación
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 7Especificación del Caso de Uso: Gestionar Asignación Materias a los grados

Especificación del Caso de Uso: Gestionar Asignaciones materias a los profesores

Descripción:El “SISTEMFYA” deberá permitir al usuario gestionar asignaciones de materias a los profesores.

Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones • El usuario debe haberse sesionado • El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar asignaciones
Flujo Normal El usuario tendrá las opciones de : 1. Listar asignaciones materias a los profesores. 2. Adicionar asignación, seguidamente el usuario aceptara la asignación para la validación de los datos. 3. Modificar asignación, seguidamente el usuario aceptara la modificación para la validación de los datos. 4. Eliminar asignación, el usuario tendrá que aceptar eliminación de la asignación para luego validar. 5. Buscar asignación
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 8Especificación del Caso de Uso: Gestionar Asignaciones materias a los profesores

Especificación del Caso de Uso: Gestionar Asignación de profesores a los grupos

Descripción:El “SISTEMFYA” deberá permitir al usuario gestionar asignaciones de profesores a los grupos.
Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones • El usuario debe haberse sesionado • El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar asignaciones
Flujo Normal El usuario tendrá las opciones de : 1. Listar asignaciones materias a los grupos. 2. Adicionar asignación, seguidamente el usuario aceptara la asignación para la validación de los datos. 4. Eliminar asignación, el usuario tendrá que aceptar eliminación de la asignación para luego validar. 5. Buscar asignación
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 9Especificación del Caso de Uso: Gestionar Asignación de profesores a los grupos

Especificación del Caso de Uso: Gestionar Asignaciones

Descripción: El “SISTEMFYA” deberá permitir al usuario gestionar Asignaciones
Actores • Usuario de “SISTEMFYA” (Administrador).
Precondiciones • El usuario debe haberse sesionado • El usuario debe de encontrarse en la pantalla principal para luego seleccionar Gestionar asignaciones
Flujo Normal El usuario tendrá las opciones de : 1. Listar materias a profesores. 2. Listar materias a los grados. 3. Listar procesos a rol. 4. Listar Profesor a grupos
Flujo Alternativo 1. El usuario elige opción salir(sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones actualizando los datos.

Tabla 10 Especificación del Caso de Uso: Gestionar Asignaciones

Especificación del Caso de Uso: Gestionar Reportes

Descripción:El sistema “SISTEMFYA” deberá permitir al usuario Gestionar Reportes
Actores □ Usuario del “SISTEMFYA”(Administrador)
Precondiciones El usuario debe haberse sesionado en “SISTEMFYA”. Seleccionar Registrar Alumnos en el menú principal de “SISTEMFYA”.
Flujo Normal 1. El usuario podrá gestionar reportes. 2. el usuario al gestionar reportes tendrá las opciones de a) Sacar reportes de materias, donde el usuario tendrá las opciones de: -Sacar reporte cantidad de alumnos por materia. -Sacar reporte de cantidad de grupos por materia. -Sacar reporte de profesores que dictan materias. b) Sacar reportes de grados, donde el usuario tendrá las opciones de: -Sacar reportes de cantidad de alumnos que cursan el grado. -Sacar reportes por gestión de profesores que se asignaron a grados. c) Sacar reportes de niveles, donde el usuario tendrá las opciones de: -Sacar reportes de cantidad de alumnos que hay por niveles. -Sacar reportes de cantidad de profesores que hay por niveles. -Sacar reportes por gestión de profesores que hay por niveles. d) Sacar reportes de aulas, donde el usuario tiene las opciones de: -Sacar reportes de materias que pasan en el aula. -Sacar reporte de aulas que se encuentran en buen estado. -Sacar reportes de aulas en mal estado.

e) Sacar reportes de alumnos, donde el usuario tiene las opciones de: -Sacar reporte de alumnos nuevos -Sacar reportes de alumnos reinscritos -Sacar reportes de alumnos por servicios básicos. -Sacar reportes de cantidad de alumnos que acuden a la unidad de acuerdo a medios de transporte. -Sacar reportes de cantidad de alumnos por lugares de nacimiento.
Flujo Alternativo 1. El usuario elige opción salir (sale del sistema) 2. “SISTEMFYA” no valida los datos (el sistema visualiza un mensaje de Error)
Pos condiciones “SISTEMFYA” recarga la pantalla con el respectivo menú de opciones.

Tabla 11Especificación del Caso de Uso: Gestionar Reportes

1.1.2.2.3.4 Prototipos de Interface de Usuario.

Introducción: Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel dibujos con alguna herramienta grafica o prototipos ejecutables interactivos, siguiendo ese orden ese orden de acuerdo al avance del proyecto. Solo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

Proposito

- Comprender la idea de cómo será el sistema más adelante.
- Identificar posibles mejoras.

Alcance

- Describir las pantallas para conocer su navegación.
- Identificar y definir las Pantallas del sistema según los objetivos del sistema deseado y aprobado por la organización.

1.1.2.2.3.5 Pantalla de Acceder al sistema



Figura n° 12 Pantalla de Acceder al Sistema

1.2.2.3.9 Pantalla principal

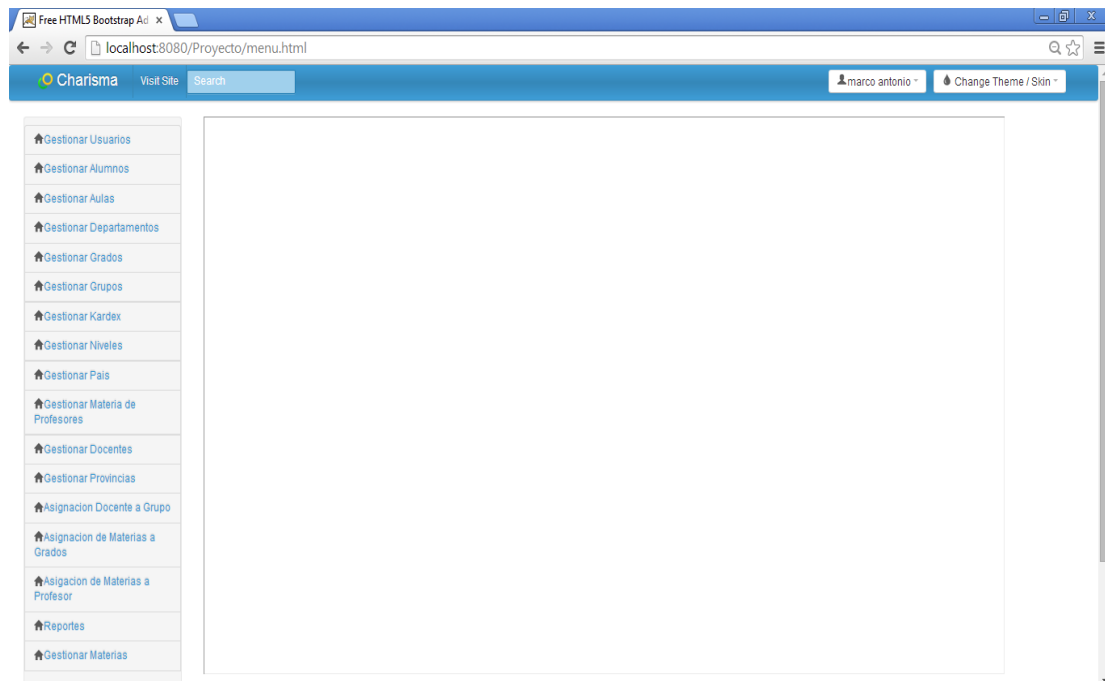


Figura n° 13 Pantalla principal

1.2.2.3.10 Pantalla Gestionar Usuarios

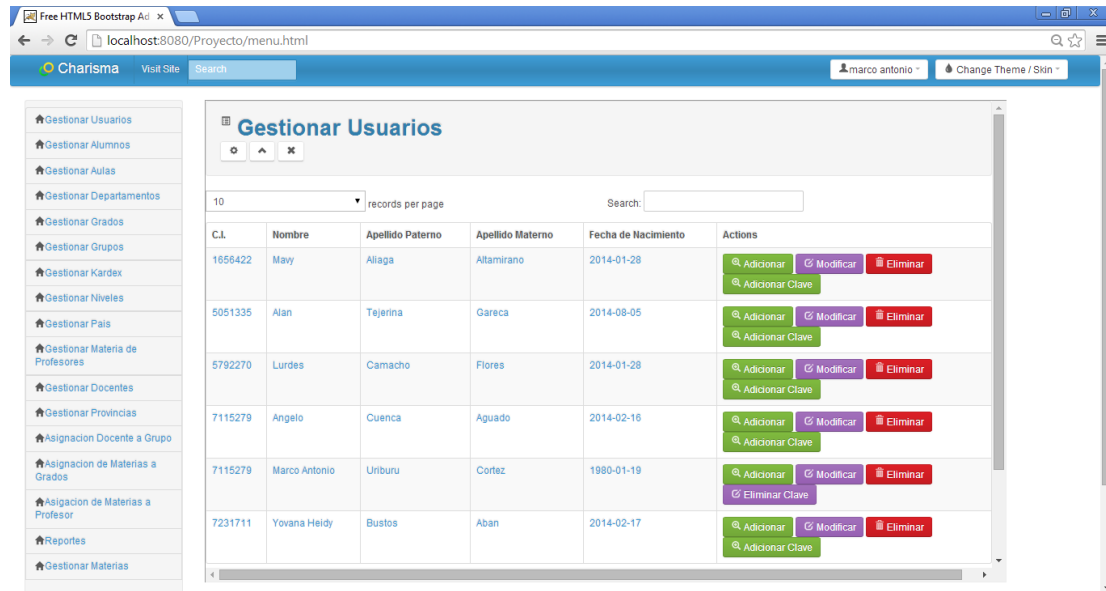


Figura n° 14 Pantalla Gestionar Usuarios

1.2.2.3.10.1 Pantalla Adicionar Usuarios

Formulario Adicionar Usuarios

Ingrese el C.I. del Usuario :

Ingrese el nombre del Usuario :

Ingrese el apellido Paterno :

Ingrese el apellido Materno :

Seleccione el sexo :

Seleccione el estado civil :

Lengua natural con la que se comunica :

Ingrese la fecha de nacimiento :

Dirección del domicilio :

Figura n° 15 Pantalla Adicionar Usuarios

1.2.2.3.10.2 Pantalla Modificar Usuarios

Figura n° 16 Pantalla Modificar Usuarios

1.2.2.3.11 Pantalla Adicionar Clave

Figura n° 17 Pantalla Adicionar Clave

1.2.2.3.12 Pantalla Listar Alumnos

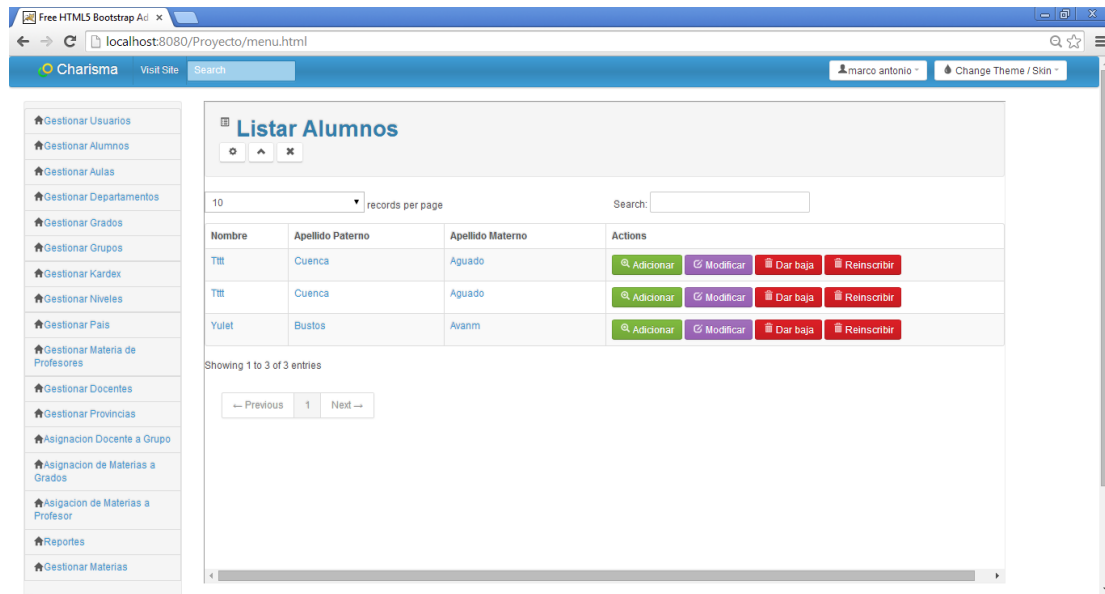


Figura n° 18 Pantalla Listar alumnos

1.2.2.3.12.1 Pantalla Adicionar Alumno

The screenshot shows the 'Formulario RUDE Adicionar Alumno' page. The form is divided into two main sections:

I Datos de la unidad educativa

1.1 Dependencia de la unidad educativa: ☒ Publica, ☐ Comunitaria, ☐ De Convenio, ☐ Privada

1.2 Nombre de la unidad educativa:

1.3 Gestion Escolar:

1.3 Distrito educativo:

II Datos del estudiante

2.1 Apellido(s) y Nombre(s):

Apellido Paterno:

Apellido Materno:

Nombre(s):

2.3 Codigo Estudiantil:

Figura n° 19 Pantalla Adicionar Alumno

1.2.2.3.12.2 Pantalla Modificar Alumno

Figura nº 20 Pantalla Modificar Alumno

1.2.2.3.12.3 Pantalla Reinscribir Alumno

Figura nº 21 Pantalla Reinscribir Alumno

1.2.2.3.13 Pantalla Gestionar Aulas

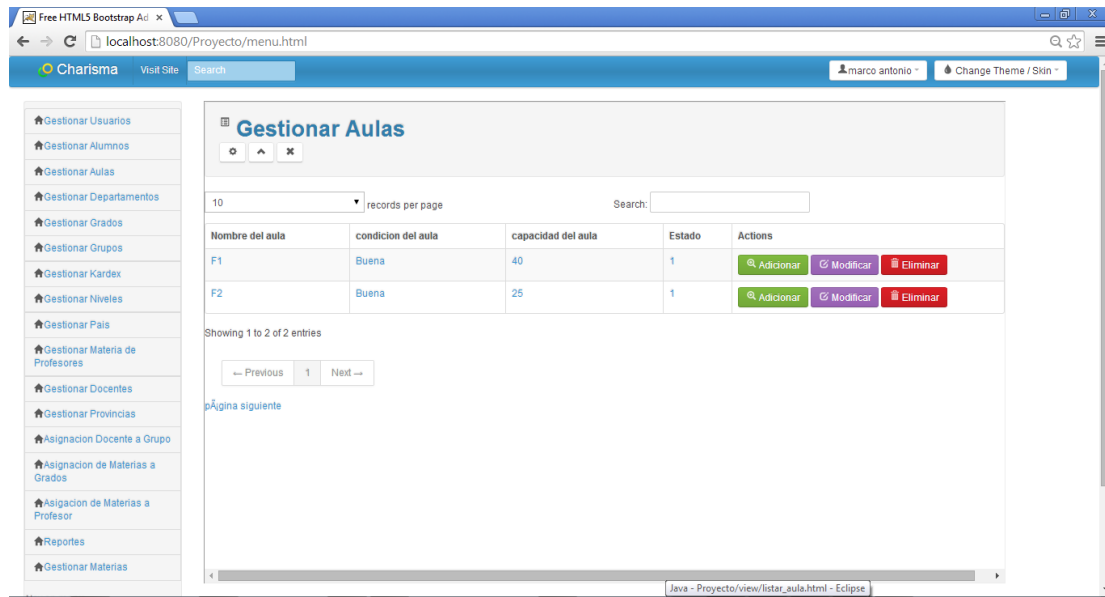


Figura nº 22 Pantalla Gestionar Aulas

1.2.2.3.13.1 Pantalla Adicionar Aula

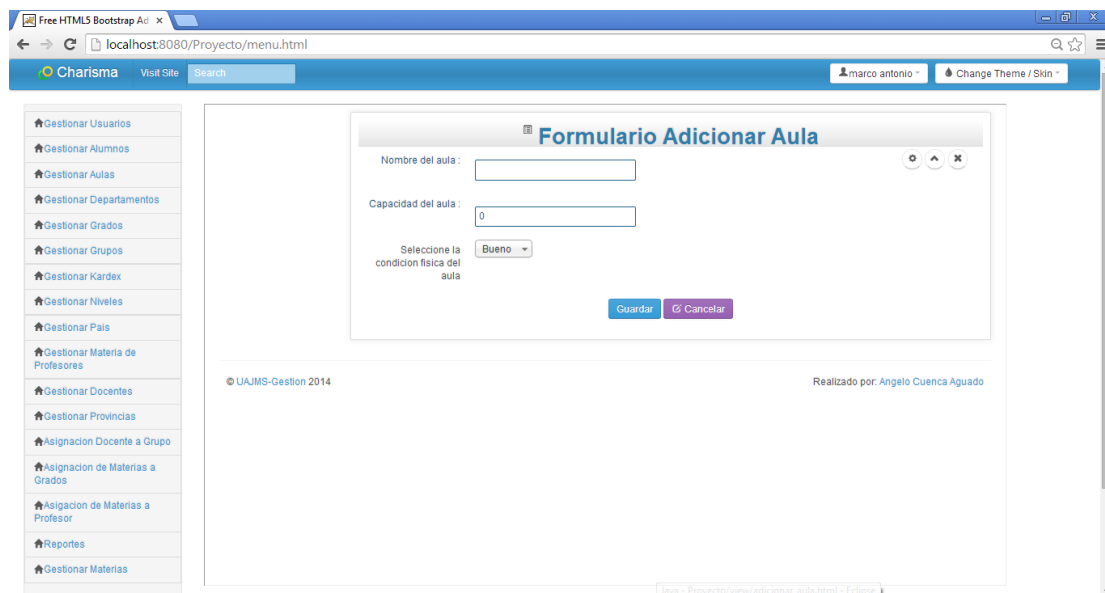


Figura nº 23 Pantalla Adicionar Aula

1.2.2.3.13.2 Pantalla Modificar Aula

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Formulario Modificar Aula

Nombre del aula : F2

Capacidad del aula : 25

Seleccione la condicion fisica del aula: Bueno

Modificar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura n° 24 Pantalla Modificar Aula

1.2.2.3.14 Pantalla Gestionar País

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Gestionar Pais

10 records per page Search:

Nombre del pais	detalle del pais	Actions
Argentina	Sur America	Adicionar Modificar Eliminar
Bolivia	Centro America	Adicionar Modificar Eliminar

Showing 1 to 2 of 2 entries

Previous 1 Next

Figura n° 25Pantalla Gestionar País

1.2.2.3.14.1 Pantalla Adicionar País

The screenshot displays the 'Formulario Adicionar Pais' (Add Country Form) within the Charisma system. The sidebar on the left contains a list of navigation items, including 'Gestionar Usuarios', 'Gestionar Alumnos', 'Gestionar Aulas', 'Gestionar Departamentos', 'Gestionar Grados', 'Gestionar Grupos', 'Gestionar Kardex', 'Gestionar Niveles', 'Gestionar Pais', 'Gestionar Materia de Profesores', 'Gestionar Docentes', 'Gestionar Provincias', 'Asignacion Docente a Grupo', 'Asignacion de Materias a Grados', 'Asignacion de Materias a Profesor', 'Reportes', and 'Gestionar Materias'. The main content area features a form with two input fields: 'Nombre del Pais' (Country Name) and 'Detalle' (Detail). Below the fields are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons. The footer indicates '© UAJMS-Gestion 2014' and 'Realizado por: Angelo Cuenca Aguado'.

Figura n° 26 Pantalla Adicionar País

1.2.2.3.14.2 Pantalla Modificar País

The screenshot displays the 'Formulario Modificar Pais' (Modify Country Form) within the Charisma system. The sidebar and footer are identical to the previous figure. The main content area features a form with two input fields: 'Nombre del Pais' (Country Name) and 'Detalle' (Detail). The 'Nombre del Pais' field is pre-filled with 'argentino' and the 'Detalle' field is pre-filled with 'sur america'. Below the fields are 'Modificar' (Modify) and 'Cancelar' (Cancel) buttons. The footer indicates '© UAJMS-Gestion 2014' and 'Realizado por: Angelo Cuenca Aguado'.

Figura n° 27 Pantalla Modificar País

1.2.2.3.15 Pantalla Gestionar Departamentos

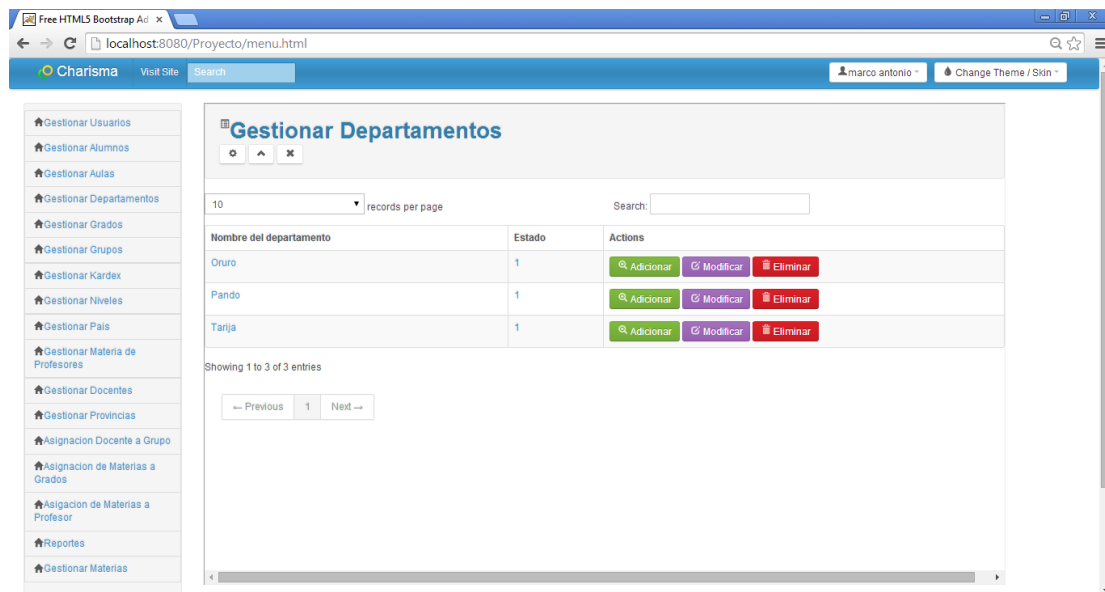


Figura nº 28 Pantalla Gestionar Departamentos

1.2.2.3.15.1 Pantalla Adicionar Departamento

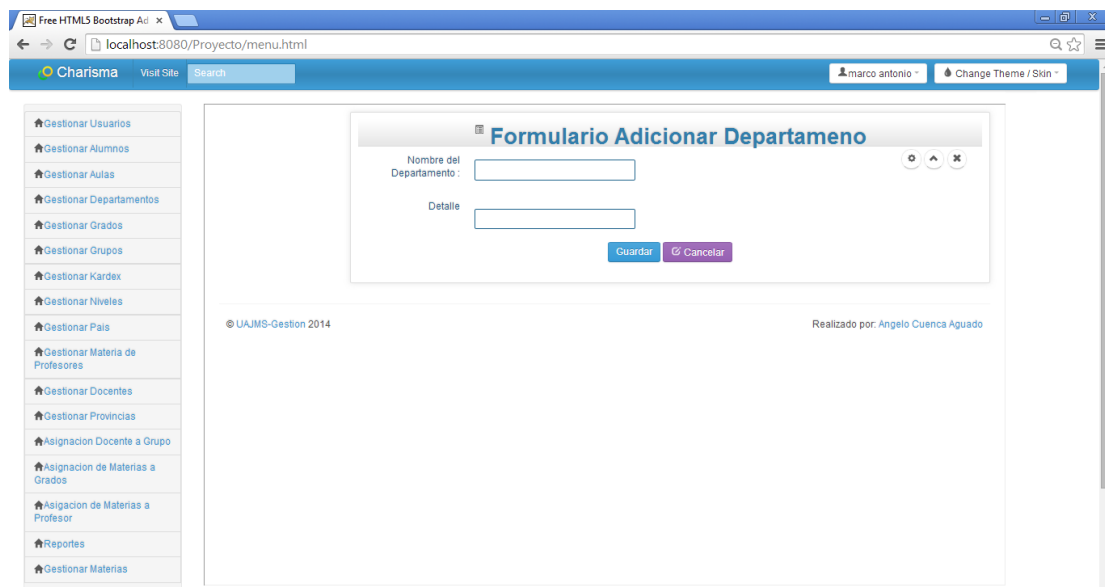


Figura nº 29 Pantalla Adicionar Departamento

1.2.2.3.15.2 Pantalla Modificar Departamento

The screenshot shows a web browser window displaying the 'Formulario Modificar Departamento' page. The browser's address bar shows 'localhost:8080/Proyecto/menu.html'. The page has a blue header with the 'Charisma' logo and a search bar. A sidebar on the left contains a list of navigation links, including 'Gestionar Departamentos'. The main content area features a form titled 'Formulario Modificar Departamento' with two input fields: 'Nombre del Departamento' (containing 'onuro') and 'Detalle' (containing 'onuro'). Below the fields are 'Modificar' and 'Cancelar' buttons. At the bottom of the page, it says '© UAJMS-Gestion 2014' and 'Realizado por: Angelo Cuenca Aguado'.

Figura n° 30 Pantalla Modificar Departamento

1.2.2.3.16 Pantalla Gestionar Provincias

The screenshot shows a web browser window displaying the 'Gestionar Provincias' page. The browser's address bar shows 'localhost:8080/Proyecto/menu.html'. The page has a blue header with the 'Charisma' logo and a search bar. A sidebar on the left contains a list of navigation links, including 'Gestionar Provincias'. The main content area features a table titled 'Gestionar Provincias' with columns: 'Nombre de la provincia', 'pertenece al departamento', 'Estado', and 'Actions'. The table contains one entry: 'Cercado' belonging to 'Oruro' with '1' in the 'Estado' column. Below the table, it says 'Showing 1 to 1 of 1 entries' and 'Previous 1 Next'. At the bottom of the page, it says '© UAJMS-Gestion 2014' and 'Realizado por: Angelo Cuenca Aguado'.

Nombre de la provincia	pertenece al departamento	Estado	Actions
Cercado	Oruro	1	Adicionar Modificar Eliminar

Figura n° 31 Pantalla Gestionar Provincias

1.2.2.3.16.1 Pantalla Adicionar Provincias

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Gestionar Usuarios
Gestionar Alumnos
Gestionar Aulas
Gestionar Departamentos
Gestionar Grados
Gestionar Grupos
Gestionar Kardex
Gestionar Niveles
Gestionar Pais
Gestionar Materia de Profesores
Gestionar Docentes
Gestionar Provincias
Asignacion Docente a Grupo
Asignacion de Materias a Grados
Asignacion de Materias a Profesor
Reportes
Gestionar Materias

Formulario Adicionar Provincia

Nombre de la Provincia:

Seleccione el departamento al que pertenece:

Guardar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura n° 32 Pantalla Adicionar Provincia

1.2.2.3.16.2 Pantalla Modificar Provincia

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Gestionar Usuarios
Gestionar Alumnos
Gestionar Aulas
Gestionar Departamentos
Gestionar Grados
Gestionar Grupos
Gestionar Kardex
Gestionar Niveles
Gestionar Pais
Gestionar Materia de Profesores
Gestionar Docentes
Gestionar Provincias
Asignacion Docente a Grupo
Asignacion de Materias a Grados
Asignacion de Materias a Profesor
Reportes
Gestionar Materias

Formulario Modificar Provincia

Nombre de la Provincia:

Seleccione el departamento al que pertenece:

Modificar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

localhost:8080/Proyecto/listar_provincia.html

Figura n° 33 Pantalla Modificar Provincia

1.2.2.3.17 Pantalla Gestionar Nivel

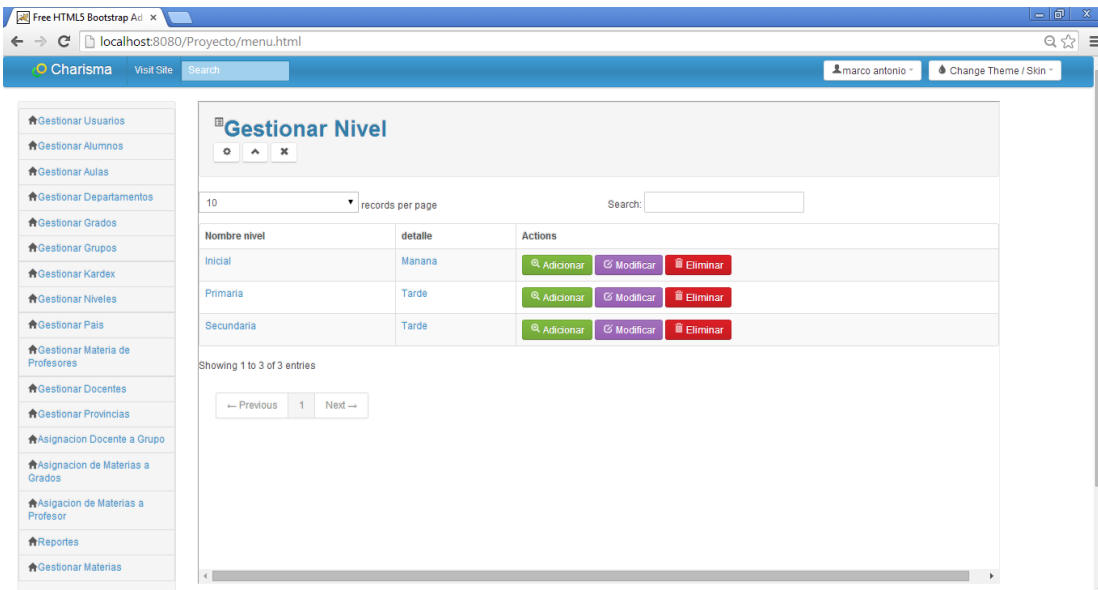


Figura n° 34 Pantalla Gestionar Nivel

1.2.2.3.17.1 Pantalla Adicionar Nivel

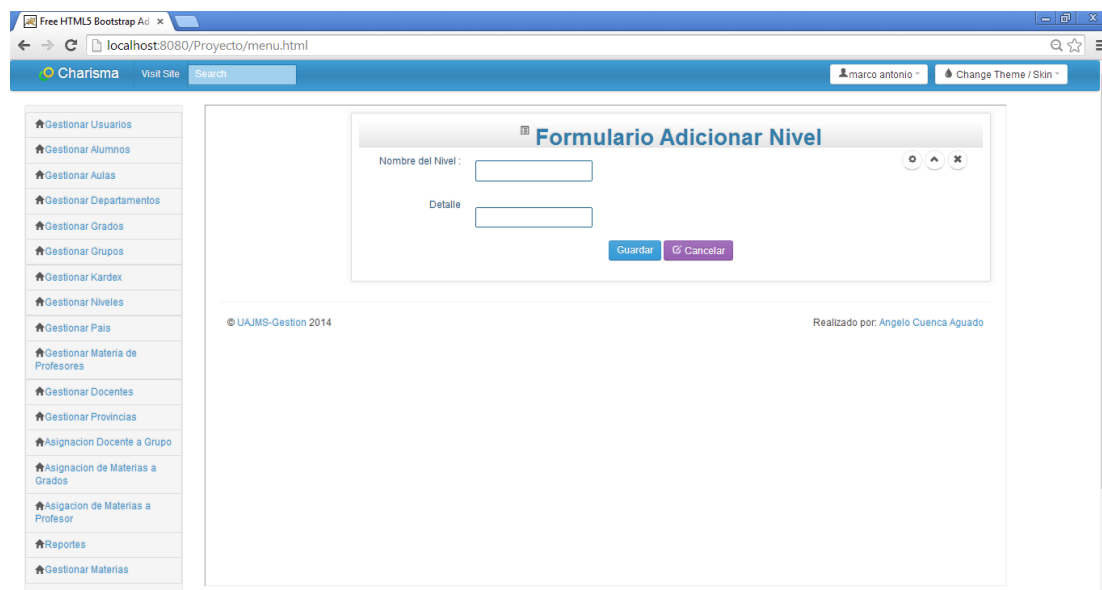


Figura n° 35 Pantalla Adicionar Nivel

1.2.2.3.17.2 Pantalla Modificar Nivel



Figura nº 36 Pantalla Modificar Nivel

1.2.2.3.18 Pantalla Gestionar Grados

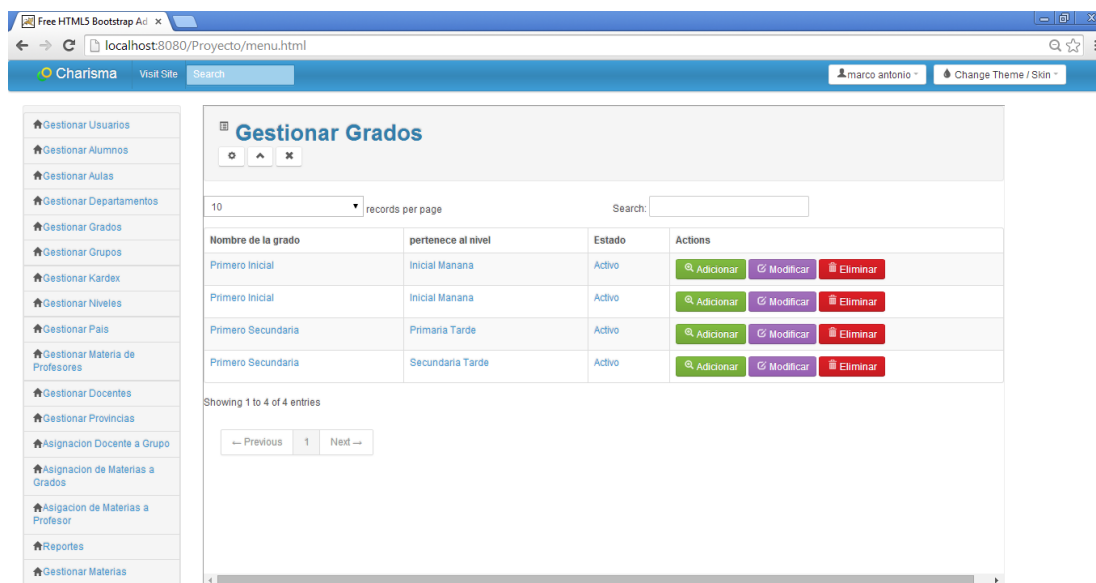


Figura nº 37Pantalla Gestionar Grados

1.2.2.3.18.1 Pantalla Adicionar Grados

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Formulario Adicionar Grado

Nombre del Grado:

Detalle:

Seleccione el nivel al que pertenece:

Guardar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

- Gestionar Usuarios
- Gestionar Alumnos
- Gestionar Aulas
- Gestionar Departamentos
- Gestionar Grados
- Gestionar Grupos
- Gestionar Kardex
- Gestionar Niveles
- Gestionar Pais
- Gestionar Materia de Profesores
- Gestionar Docentes
- Gestionar Provincias
- Asignacion Docente a Grupo
- Asignacion de Materias a Grados
- Asignacion de Materias a Profesor
- Reportes
- Gestionar Materias

Figura n° 38 Pantalla Adicionar Grados

1.2.2.3.18.2 Pantalla Modificar Grados

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio Change Theme / Skin

Formulario Modificar Grado

Nombre del Grado:

Detalle:

Seleccione el nivel al que pertenece:

Modificar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

- Gestionar Usuarios
- Gestionar Alumnos
- Gestionar Aulas
- Gestionar Departamentos
- Gestionar Grados
- Gestionar Grupos
- Gestionar Kardex
- Gestionar Niveles
- Gestionar Pais
- Gestionar Materia de Profesores
- Gestionar Docentes
- Gestionar Provincias
- Asignacion Docente a Grupo
- Asignacion de Materias a Grados
- Asignacion de Materias a Profesor
- Reportes
- Gestionar Materias

Figura n° 39 Pantalla Modificar Grado

1.2.2.3.19 Pantalla Gestionar Grupos

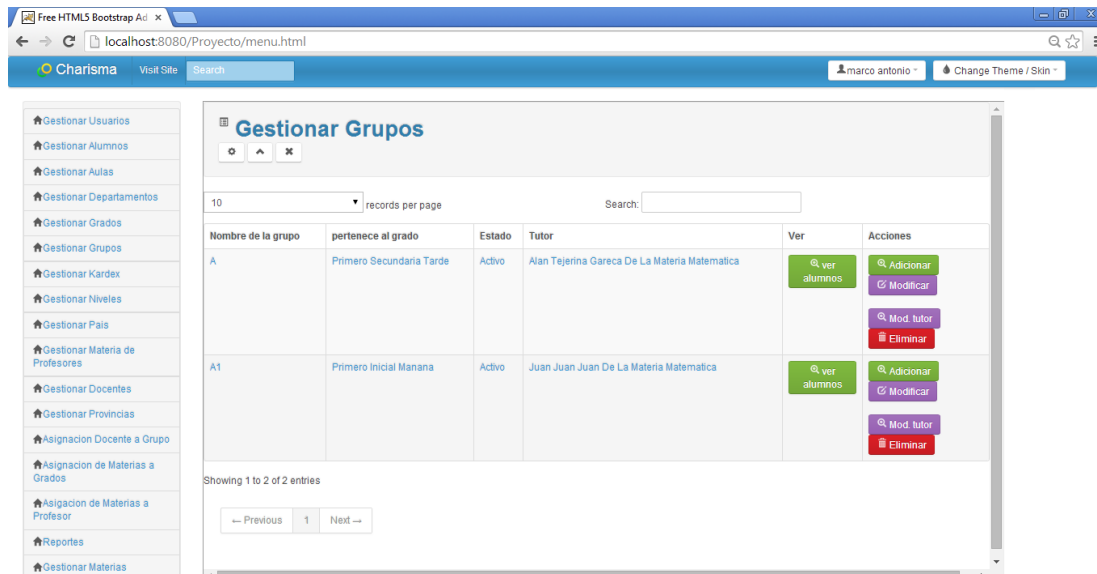


Figura nº 40 Pantalla Gestionar Grupos

1.2.2.3.19.1 Pantalla Adicionar Grupo

Formulario Adicionar Grupo

Nombre del Grupo:

Detalle:

Seleccione el grado al que pertenece:

Seleccione el aula que se asignara al grupo:

[Guardar](#) [Cancelar](#)

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura nº 41 Pantalla Adicionar Grupo

1.2.2.3.19.2 Pantalla Modificar Grupo

Figura nº 42 Pantalla Modificar Grupo

1.2.2.3.20 Pantalla Asignar Tutor

Figura nº 43 Pantalla Asignar Tutor

1.2.2.3.20.1 Pantalla Modificar Tutor

Formulario Modificar Tutor

Actual tutor del grupo : alan tejerina gareca de la materia de matematica

Seleccione el profesor:

☐ mayvy Allaga de la materia sociales

☒ alan tejerina de la materia matematica

Consultas :

Días de consultas :

[Modificar](#) [Cancelar](#)

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura n° 44 Pantalla Modificar Tutor

1.2.2.3.21 Pantalla Ver Alumnos por Grupo

Lista de Alumnos del Grupo A1

10 records per page Search:

Nombre	Apellido Paterno	Apellido Materno	Actions
Juan	Pelaes	Salgado	añadir nota ver libreta
Titi	Cuenca	Aguado	añadir nota ver libreta
Titi	Cuenca	Aguado	añadir nota ver libreta
Yulet	Bustos	Avanm	añadir nota ver libreta

Showing 1 to 4 of 4 entries

[← Previous](#) [1](#) [Next →](#)

Figura n° 45 Pantalla Ver Alumnos por Grupo

1.2.2.3.21 Pantalla Listar Materia por Alumno

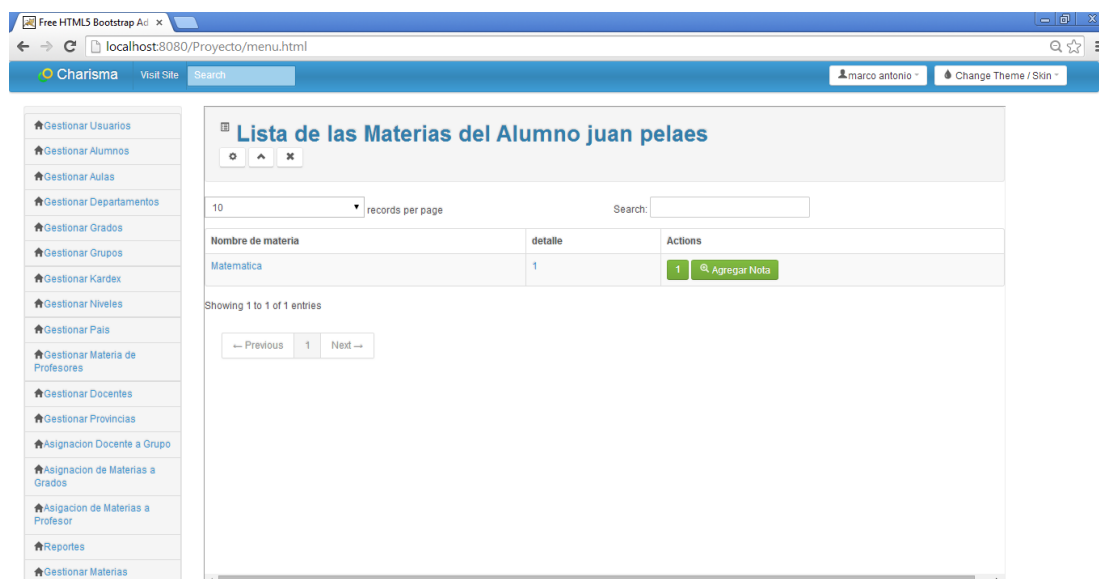


Figura nº 46 Pantalla Listar Materia por Grupo

1.2.2.3.22 Pantalla Mostrar Nota

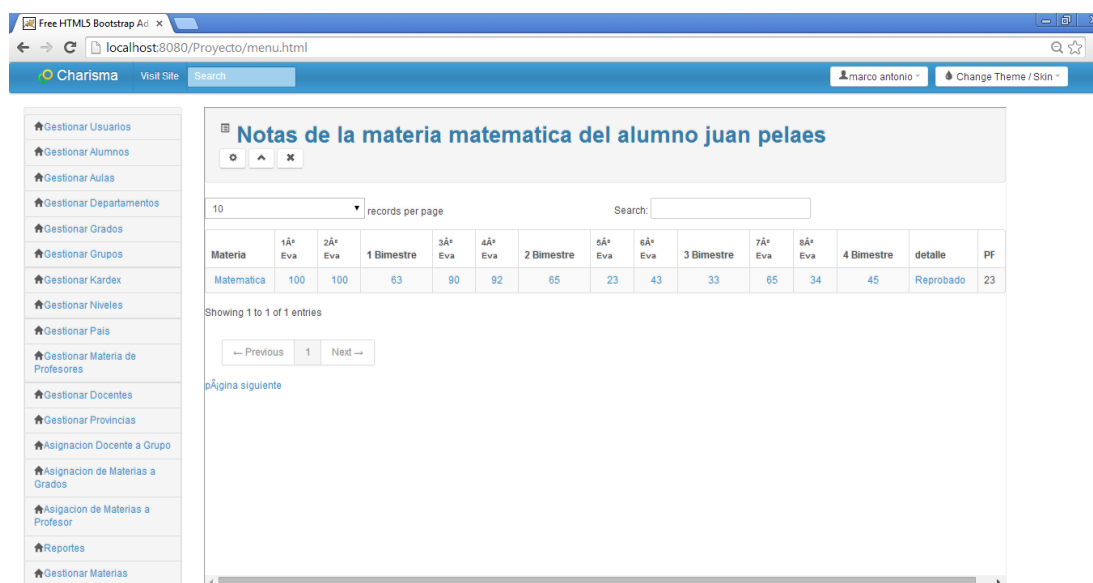


Figura nº 47Pantalla Mostrar Nota

1.2.2.3.23 Pantalla Ver Libreta

Libreta

Fe y Alegria
Humberto Portocarrero

Alumno(a): juan pelaes salgado
Codigo: 1
Curso: Inicial primero A1
Nivel Inicial
Titular: juan juan juan

CAMPOS SABERES Y CONOCIMIENTOS AREAS CURRICULARES	1A° Eva.	2A° Eva.	1A° BIM.	3A° Eva.	4A° Eva.	2A° BIM.	5A° Eva.	6A° Eva.	3A° BIM.	7A° Eva.	8A° Eva.	4A° BIM.	PF
matematica 1	100	100	63	90	92	65	23	43	33	65	34	45	23

OBSERVACIONES Y RECOMENDACIONES

RENDIMIENTO

1 a 50 En desarrollo
51 a 68 Desarrollo aceptable
69 a 84 Desarrollo optimo
85 a 100 Desarrollo pleno

SIGLAS

PF: promedio final

Figura n° 48 Pantalla Ver Libreta

1.2.2.3.24 Pantalla Gestionar Materia Estudiada

Gestionar Materias Est.

10 records per page Search:

Nombre de la materia	detalle de la materia	Actions
Ciencias Sociales	Sociales Para Primaria	Adicionar Modificar Eliminar
Historia	Historia Para Secundaria	Adicionar Modificar Eliminar
Lenguaje	Lenguaje Para Primaria	Adicionar Modificar Eliminar
Lenguaje	Lenguaje Para Secundaria	Adicionar Modificar Eliminar
Matematica	Matematica Para Primaria	Adicionar Modificar Eliminar
Matematica	Matematica Para Secundaria	Adicionar Modificar Eliminar

Showing 1 to 6 of 6 entries

← Previous 1 Next →

Figura n° 49 Pantalla Gestionar Materia Estudiada

1.2.2.3.24.1 Pantalla Adicionar Materia Estudiada

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio - Change Theme / Skin

Gestionar Usuarios
 Gestionar Alumnos
 Gestionar Aulas
 Gestionar Departamentos
 Gestionar Grados
 Gestionar Grupos
 Gestionar Kardex
 Gestionar Niveles
 Gestionar Pais
 Gestionar Materia de Profesores
 Gestionar Docentes
 Gestionar Provincias
 Asignacion Docente a Grupo
 Asignacion de Materias a Grados
 Asignacion de Materias a Profesor
 Reportes
 Gestionar Materias

Formulario Adicionar Materia Est.

Nombre de la Materia Estudiada:

Detalle:

Guardar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura nº 50 Pantalla Adicionar Materia Estudiada

1.2.2.3.24.2 Pantalla Modificar Materia Estudiada

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio - Change Theme / Skin

Gestionar Usuarios
 Gestionar Alumnos
 Gestionar Aulas
 Gestionar Departamentos
 Gestionar Grados
 Gestionar Grupos
 Gestionar Kardex
 Gestionar Niveles
 Gestionar Pais
 Gestionar Materia de Profesores
 Gestionar Docentes
 Gestionar Provincias
 Asignacion Docente a Grupo
 Asignacion de Materias a Grados
 Asignacion de Materias a Profesor
 Reportes
 Gestionar Materias

Formulario Modificar Materia Est.

Nombre de la Materia Estudiada:

Detalle:

Modificar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura nº 51 Pantalla Modificar Materia Estudiada

1.2.2.3.25 Pantalla Gestionar Profesores

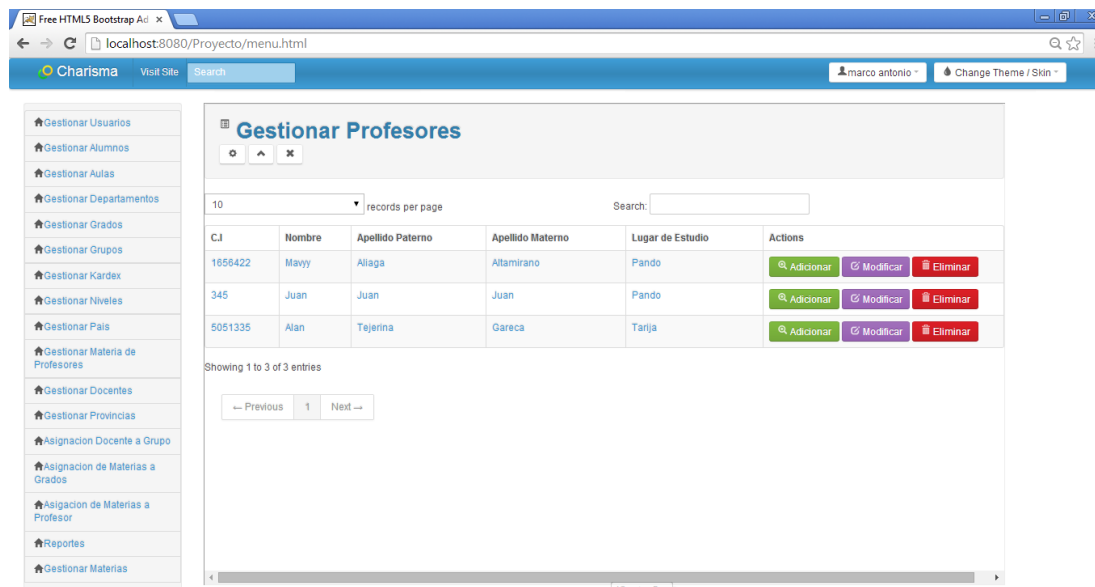


Figura n° 52 Pantalla Gestionar Profesores

1.2.2.3.25.1 Pantalla Adicionar Profesores

The screenshot shows the 'Formulario Adicionar Profesor' page. The form includes the following fields and controls:

- Ingrese el C.I. del Profesor:
- Ingrese el nombre del Profesor:
- Ingrese el apellido Paterno:
- Ingrese el apellido Materno:
- Seleccione el sexo:
- Seleccione el estado civil:
- Lengua natural con la que se comunica:
- Otra lengua con la que se comunica:
- Direccion del domicilio:
- Seleccione el departamento donde:

Figura n° 53 Pantalla Adicionar Profesor

1.2.2.3.25.2 Pantalla Modificar Profesores

Formulario Modificar Profesor

Ingrese el C.I. del Profesor:

Ingrese el nombre del Profesor:

Ingrese el apellido Paterno:

Ingrese el apellido Materno:

Seleccione el sexo:

Seleccione el estado civil:

Lengua natural con la que se comunica:

Otra lengua con la que se comunica:

Dirección del domicilio:

Seleccione el país:

Figura nº 54 Pantalla Modificar Profesor

1.2.2.3.26 Pantalla Gestionar Materias

Gestionar Materias

10 records per page Search:

Nombre de la materia	Detalle	Estado	Actions
Ciencias Naturales	Primaria	Activo	Adicionar Modificar Eliminar
Historia	Secundaria	Activo	Adicionar Modificar Eliminar
Matematica	Primaria	Activo	Adicionar Modificar Eliminar
Psicologia	Secundaria	Activo	Adicionar Modificar Eliminar
Sociales	Primaria	Activo	Adicionar Modificar Eliminar

Showing 1 to 5 of 5 entries

1

[Página siguiente](#)

Figura nº 55 Pantalla Gestionar Materias

1.2.2.3.26.1 Pantalla Adicionar Materias

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio - Change Theme / Skin -

Gestionar Usuarios
Gestionar Alumnos
Gestionar Aulas
Gestionar Departamentos
Gestionar Grados
Gestionar Grupos
Gestionar Kardex
Gestionar Niveles
Gestionar Pais
Gestionar Materia de Profesores
Gestionar Docentes
Gestionar Provincias
Asignacion Docente a Grupo
Asignacion de Materias a Grados
Asignacion de Materias a Profesor
Reportes
Gestionar Materias

Formulario Adicionar Materia

Nombre de la materia :

Detalle :

Guardar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura nº 56 Pantalla Adicionar Materias

1.2.2.3.26.2 Pantalla Modificar Materias

Free HTML5 Bootstrap Ad x

localhost:8080/Proyecto/menu.html

Charisma Visit Site Search

marco antonio - Change Theme / Skin -

Gestionar Usuarios
Gestionar Alumnos
Gestionar Aulas
Gestionar Departamentos
Gestionar Grados
Gestionar Grupos
Gestionar Kardex
Gestionar Niveles
Gestionar Pais
Gestionar Materia de Profesores
Gestionar Docentes
Gestionar Provincias
Asignacion Docente a Grupo
Asignacion de Materias a Grados
Asignacion de Materias a Profesor
Reportes
Gestionar Materias

Formulario Modificar Materia

Nombre de la materia :

Detalle :

Modificar Cancelar

© UAJMS-Gestion 2014 Realizado por: Angelo Cuenca Aguado

Figura nº 57 Pantalla Modificar Materia

1.2.2.3.27 Pantalla Gestionar Materia a Grado



Figura nº 58 Pantalla Gestionar Materia a Grado

1.2.2.3.28 Pantalla Asignar Materia grado



Figura nº 59 Pantalla Asignar Materia a Grado

1.2.2.3.29 Pantalla Ver Materias Asignadas al Grado



Figura n° 60 Pantalla Ver Materias Asignadas al Grado

1.2.2.3.30 Pantalla Gestionar Asignación Materias a Profesor

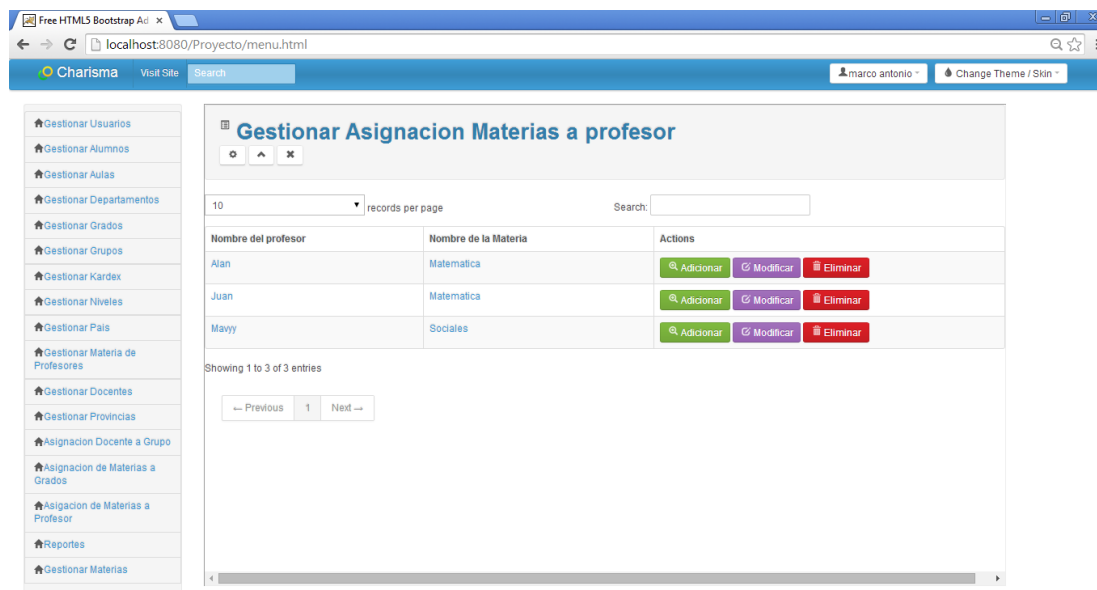


Figura n° 61 Pantalla Gestionar Materias a Profesor

1.2.2.3.31 Pantalla Asignar Materia a Profesor

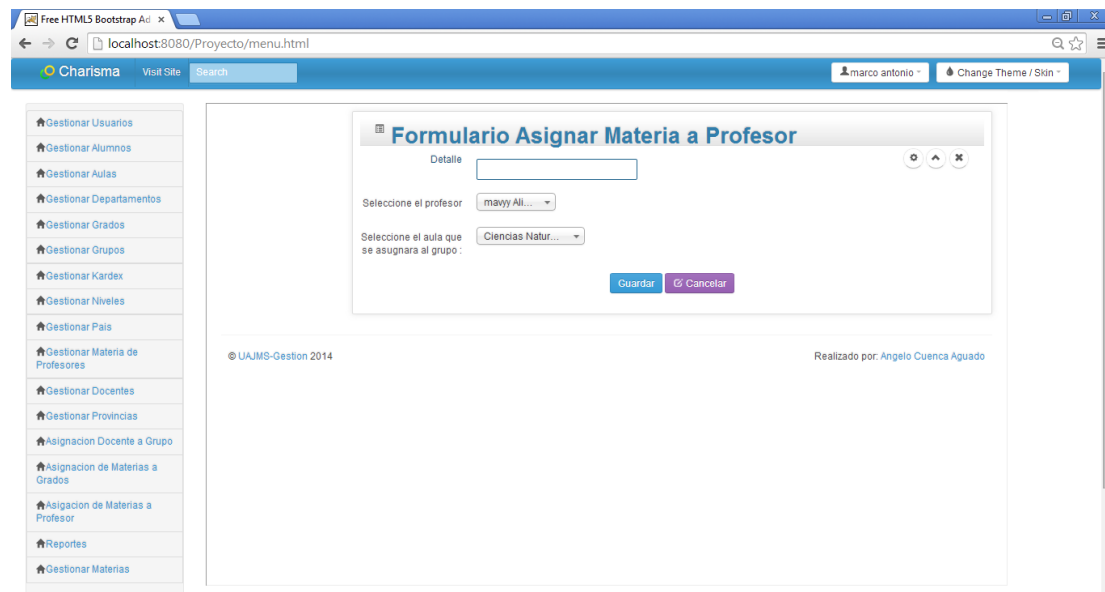


Figura nº 62 Pantalla Asignar Materia a Profesor

1.2.2.3.32 Pantalla Modificar Asignación materia a profesor



Figura nº 63 Pantalla Modificar Asignación Materia a Profesor

1.2.2.3.33 Pantalla Gestionar asignación de profesor a grupo

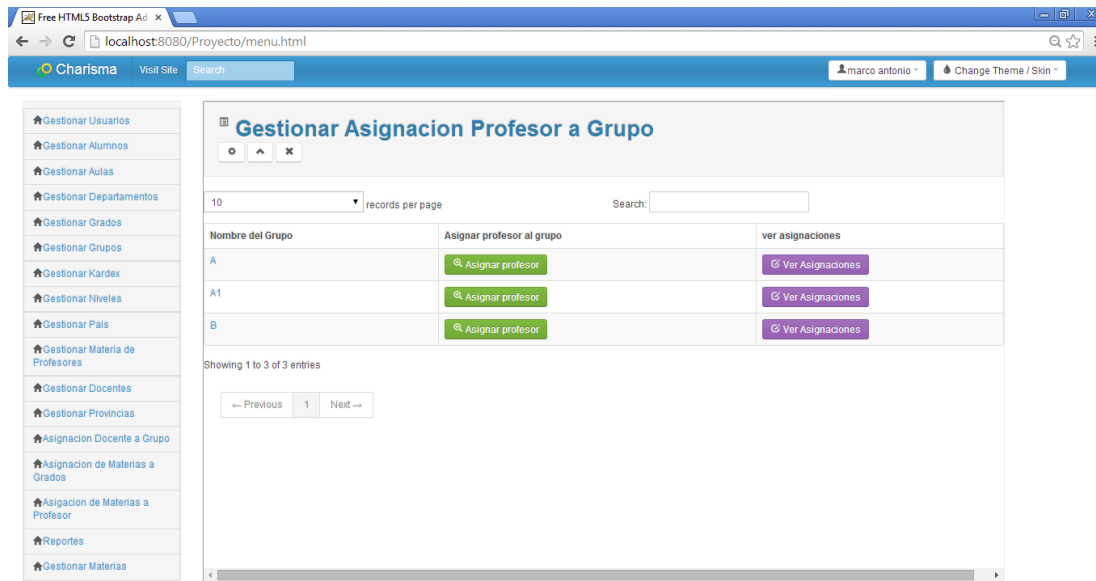


Figura nº 64 Pantalla Gestionar asignación de profesor a grupo

1.2.2.3.34 Pantalla Gestionar asignación de profesor a grupo

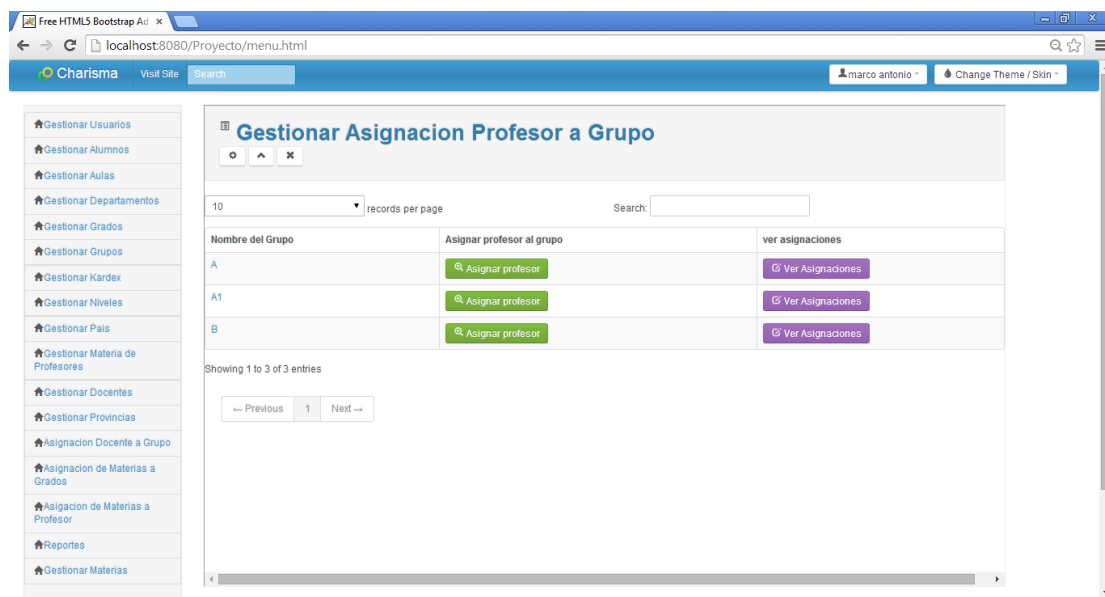


Figura nº 65 Pantalla Gestionar asignación de profesor a grupo

1.2.2.3.35 Pantalla Ver Materias del Grupo

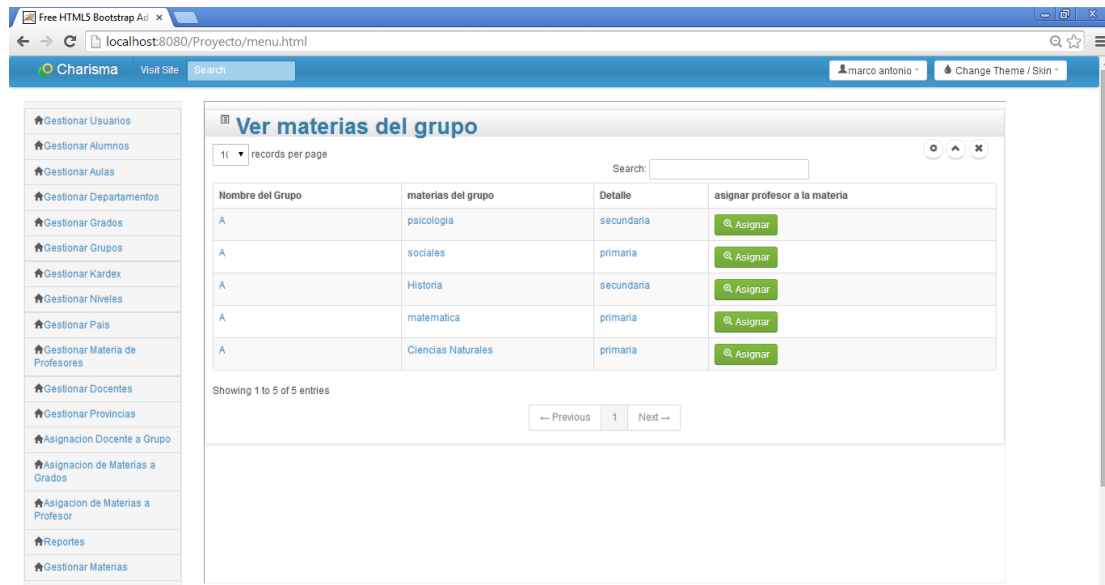


Figura n° 66 Pantalla Ver Materias del Grupo

1.2.2.3.36 Pantalla Asignar Profesor a Grupo

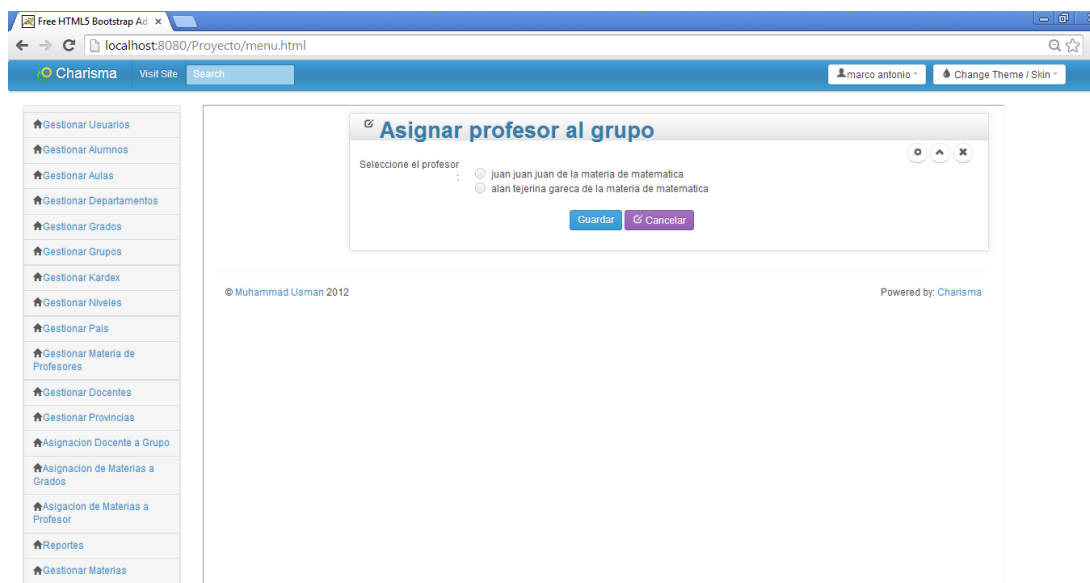


Figura n° 67 Pantalla Asignar Profesor a Grupo

1.2.2.3.37 Pantalla Ver Asignación del Grupo

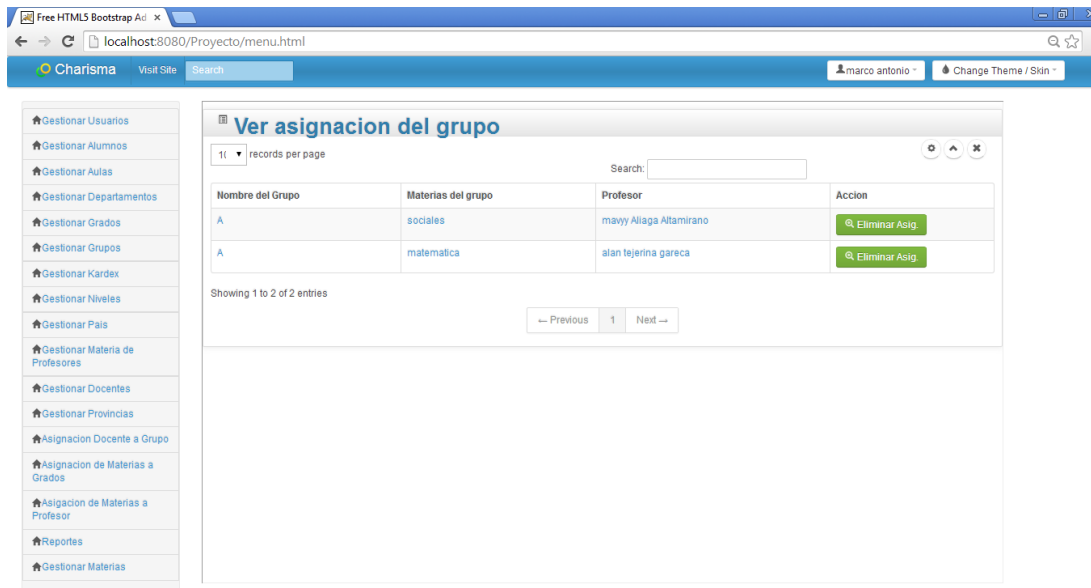


Figura n° 68 Pantalla Ver Asignación del Grupo

1.2.2.3.38 Pantalla Gestionar Tipo Reporte



Figura n° 69 Pantalla Gestionar Tipo Reporte

1.2.2.3.39 Pantalla Reporte Alumno por Aula

Reporte de alumno por aula

marco antonio
Mon Sep 29 07:54:25 CEST 2014

10 records per page Search:

Nombre aula	condicion	capacidad	Nombre del alumno
F1	Buena	40	Ttt Cuenca Aguado
F1	Buena	40	Ttt Cuenca Aguado
F1	Buena	40	Yulet Bustos Avann
F1	Buena	40	Juan Pelaez Salgado

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Figura nº 70 Pantalla Reporte Alumno por Aula

1.2.2.3.40 Pantalla Reportes Alumnos Nivel

Reporte de nivel alumno

marco antonio
Mon Sep 29 07:57:58 CEST 2014

10 records per page Search:

Nombre nivel	detalle	estado	Nombre del alumno	sexo	gestion kardex
Inicial	Manana	Activo	Ttt Cuenca Aguado	Masculino	2015
Inicial	Manana	Activo	Ttt Cuenca Aguado	Masculino	2016
Inicial	Manana	Activo	Yulet Bustos Avann	Femenino	2013
Inicial	Manana	Activo	Juan Pelaez Salgado	Masculino	2014

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Figura nº 71 Pantalla Reportes Alumnos Nivel

1.2.2.3.41 Pantalla Reportes Alumnos Grado

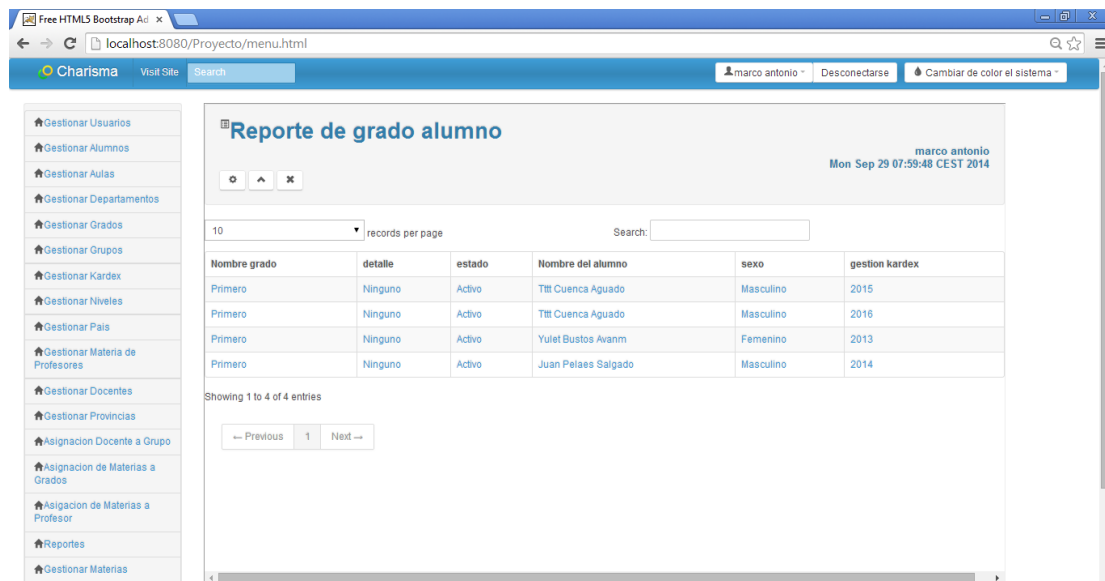


Figura n° 72 Pantalla Reportes Alumnos Grado

1.2.2.3.42 Pantalla Reportes de alumnos por grupo y sexo

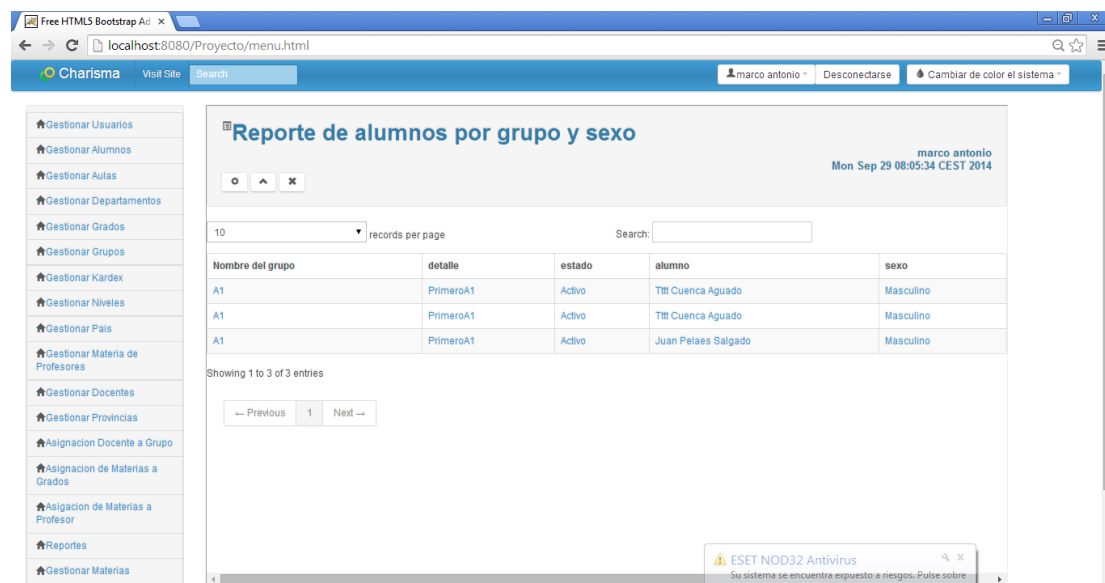


Figura n° 73 Pantalla Reportes de alumnos por grupo y sexo

1.2.2.3.43 Pantalla Reportes de los Grados

Reporte de Grados

marco antonio
Mon Sep 29 08:06:21 CEST 2014

10 records per page Search:

Nombre del grado	detalle	estado
Asd	Awer	Activo
Primero	Secundaria	Activo
Primero	Secundaria	Activo
Primero	Inicial	Activo
Primero	Inicial	Activo
Primero	Ninguno	Activo
Segundo	Secundaria	Activo

Showing 1 to 7 of 7 entries

← Previous 1 Next →

Figura nº 74 Pantalla Reportes de los Grados

1.2.2.3.44 Pantalla Reportes de grupo y tutor

Reporte de Tutor y grupo

marco antonio
Mon Sep 29 08:07:39 CEST 2014

10 records per page Search:

Nombre del grupo	detalle	estado	tutor	consultas
A1	PrimeroA1	Activo	Juan Juan Juan	De 8:00 A 9:30 Lunes, Martes, Viernes
A1	PrimeroA1	Activo	Juan Juan Juan	De 8:00 A 9:30 Lunes, Martes, Viernes
A1	PrimeroA1	Activo	Juan Juan Juan	De 8:00 A 9:30 Lunes, Martes, Viernes
A1	PrimeroA1	Activo	Juan Juan Juan	De 8:00 A 9:30 Lunes, Martes, Viernes

Showing 1 to 4 of 4 entries

← Previous 1 Next →

Figura nº 75 Pantalla Reportes de grupo y tutor

2.1.2.2.4 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.

1.2.2.4.1 Diagramas de Actividades

Introducción: Mediante el uso de los diagramas de actividades podemos mejorar el flujo de control entre actividades del sistema. La idea es generar una especie de diagrama Pert, en el que puede ver el flujo de actividades que tiene lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. Gráficamente es un conjunto de arcos y nodos. Desde un punto de vista conceptual, el diagrama de actividades muestra como fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecuencia de un proceso más complejo. Por este motivo aparecerá acciones y actividades correspondientes a distintas cases, colaborando todas ellas para conseguir un mismo fin.

Propósito:

- Comprender la Estructura y la dinámica del sistema deseado para la Organización.
- Identificar posibles mejoras.

Alcance:

- Describe 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 actividades para cada caso de uso del sistema.

Diagrama de Actividad del C.U.: Ingresar al sistema

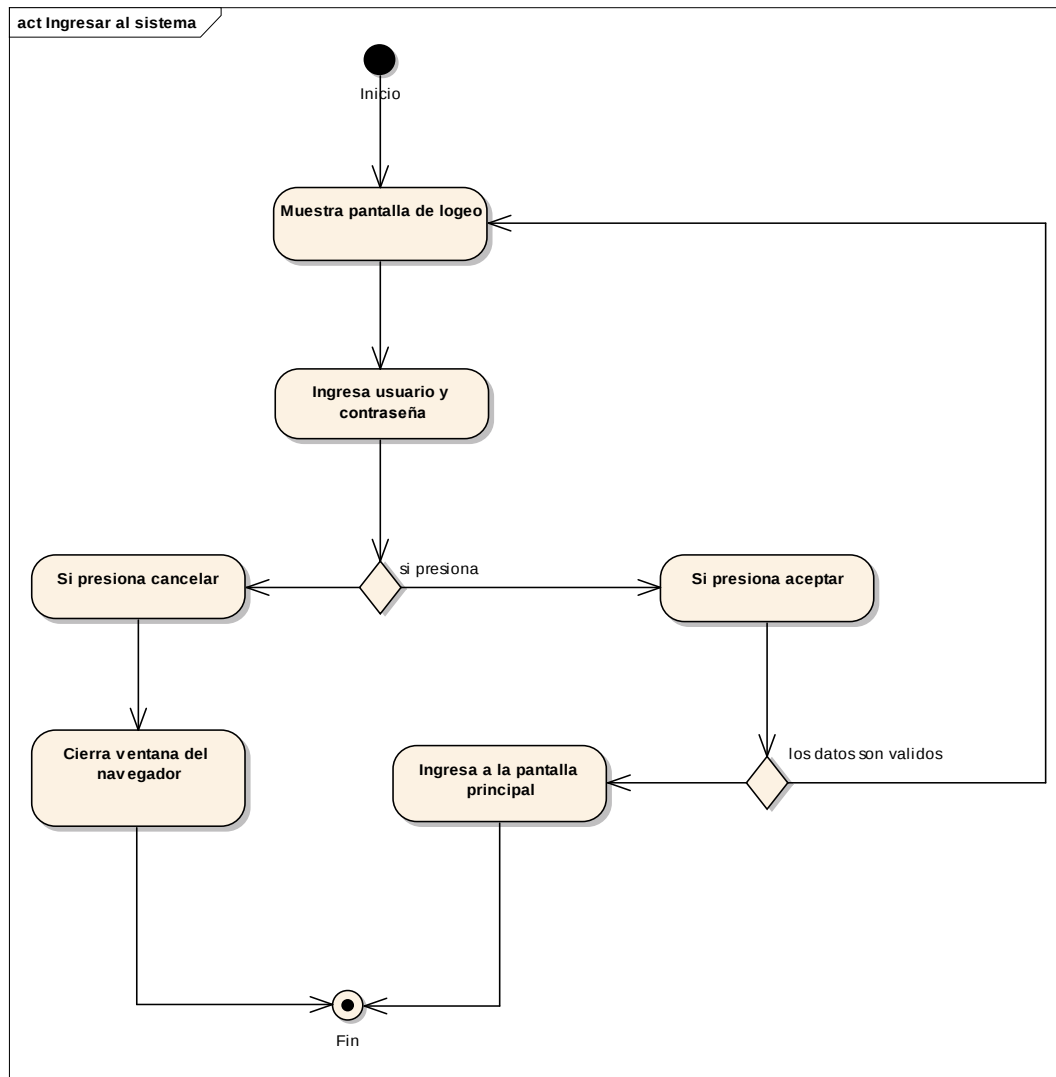


Figura nº 76 Diagrama de Actividad del C.U.: Ingresar al sistema

Diagrama de Actividad del C.U.: Gestionar usuario

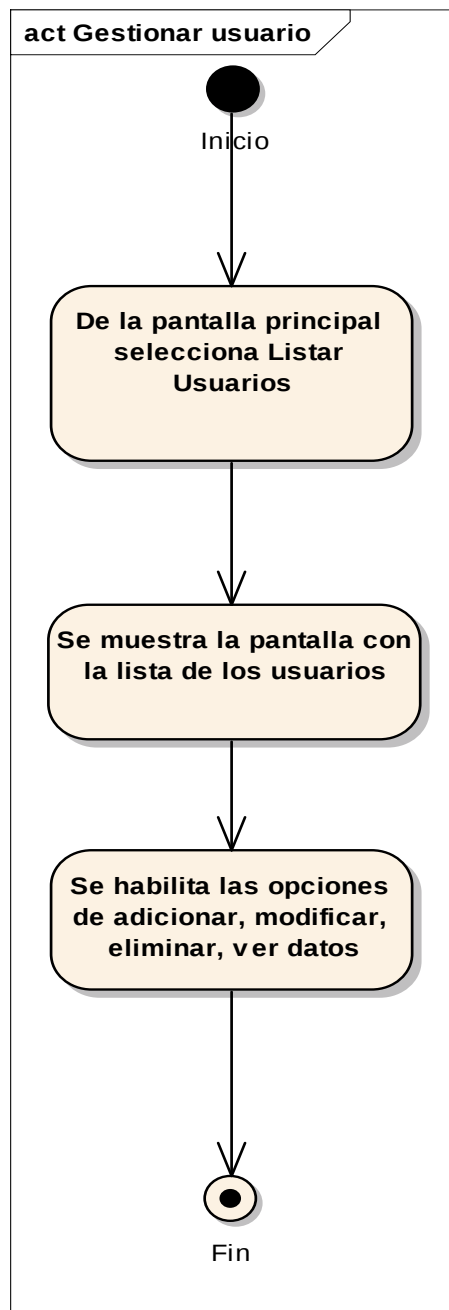


Figura nº 77 Diagrama de Actividad del C.U.: Gestionar usuario

Diagrama de Actividad del C.U.: Adicionar usuario

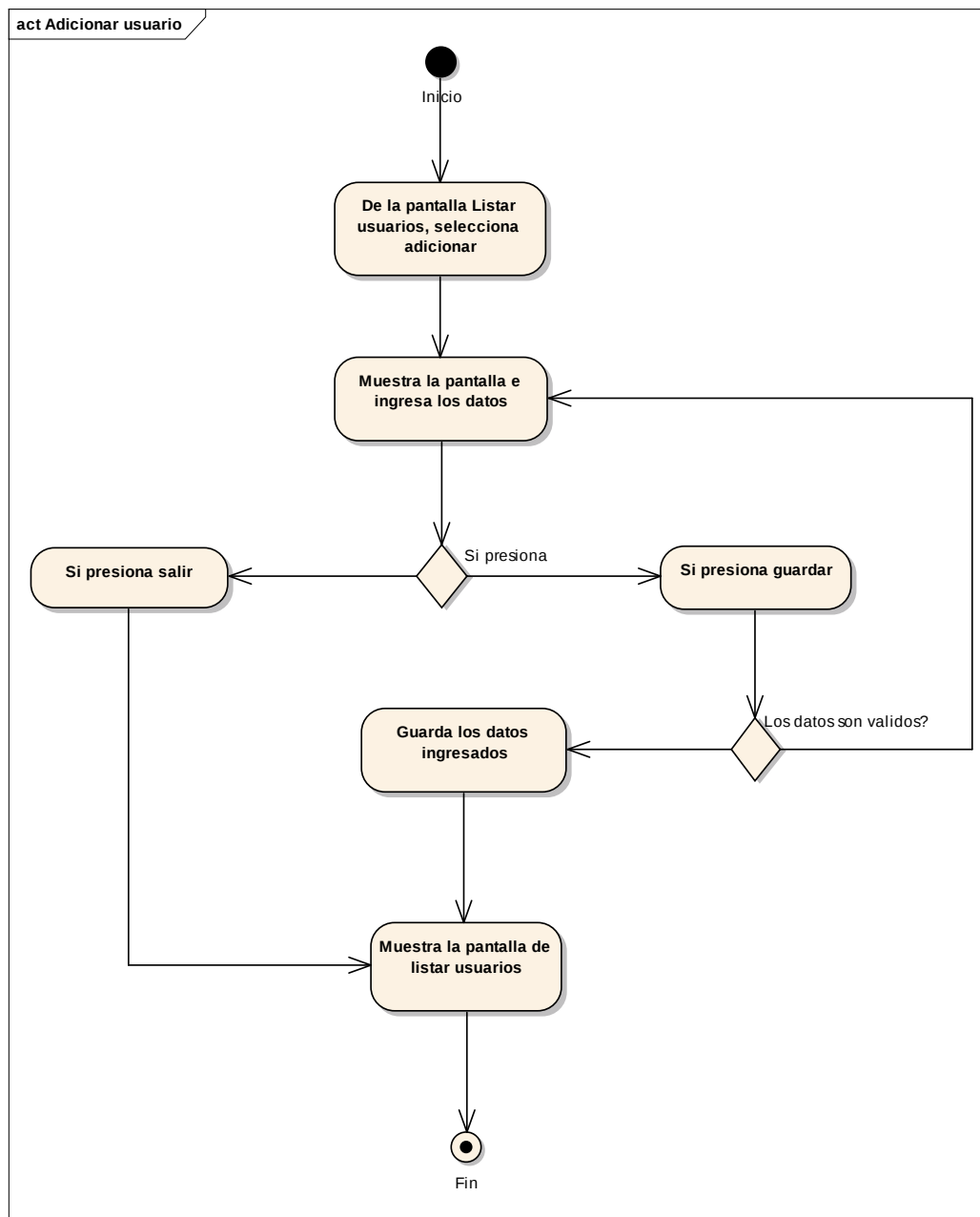


Figura n° 78 Diagrama de Actividad del C.U.: Adicionar usuario

Diagrama de Actividad del C.U.. Modificar usuarios

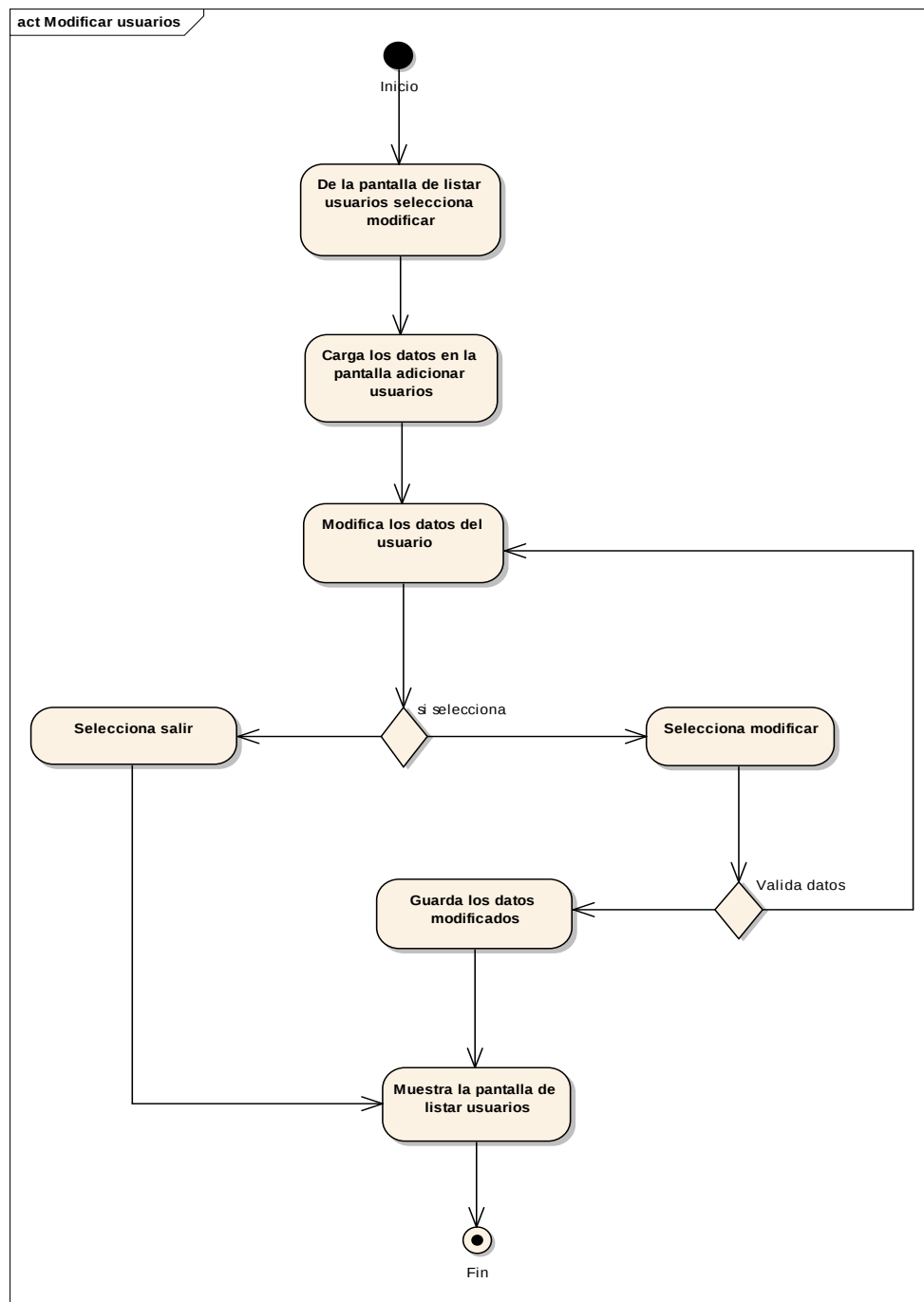


Figura nº 79 Diagrama de Actividad del C.U. Modificar usuarios

Diagrama de Actividad del C.U.: Adicionar Alumno

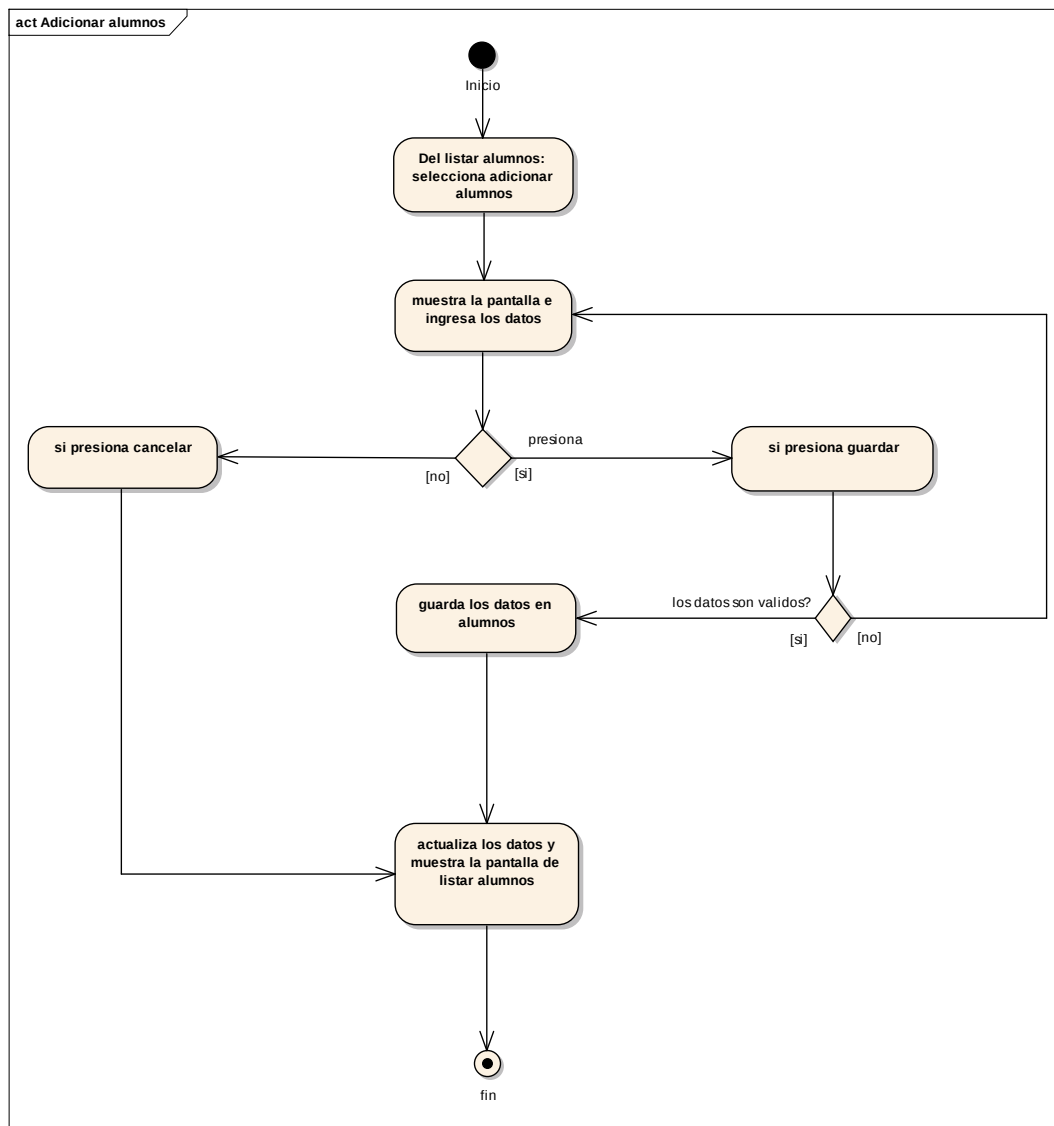


Figura nº 80 Diagrama de Actividad del C.U.: Adicionar Alumno

Diagrama de Actividad del C.U.: Modificar Alumno

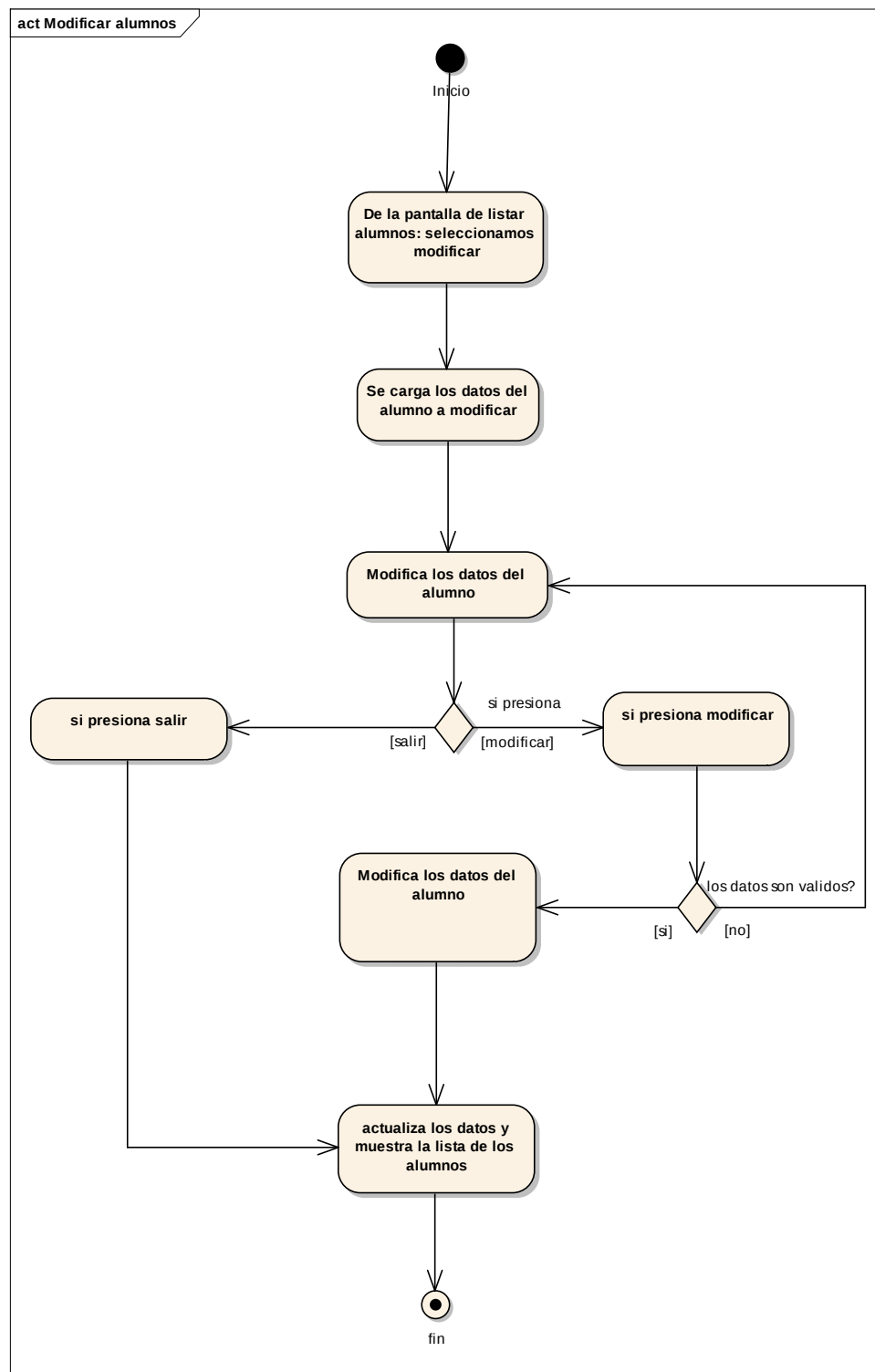


Figura nº 81 Diagrama de Actividad del C.U.: Modificar Alumno

Diagrama de Actividad del C.U.: Gestionar Departamento

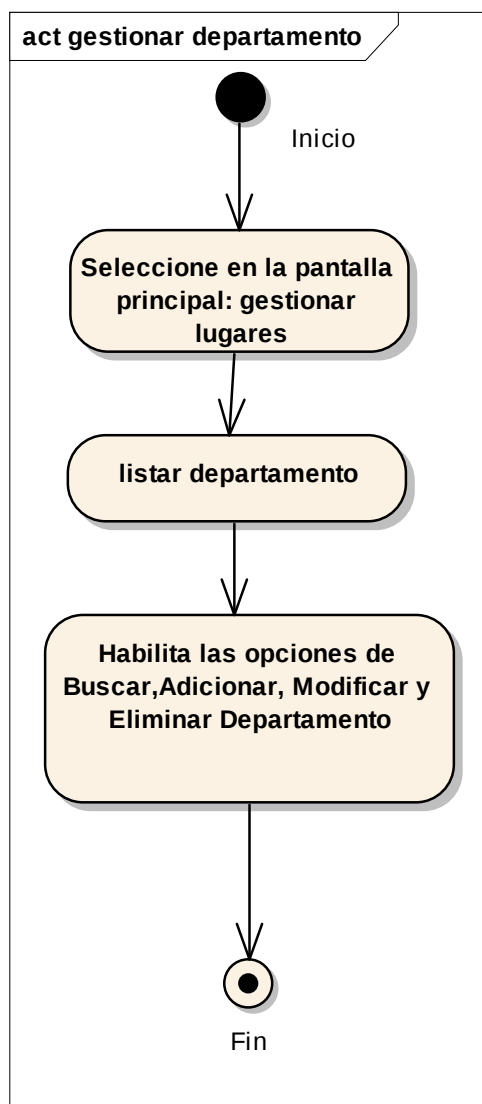


Figura nº 82 Diagrama de Actividad del C.U.: Gestionar Departamento

Diagrama de Actividad del C.U.: Adicionar Departamento

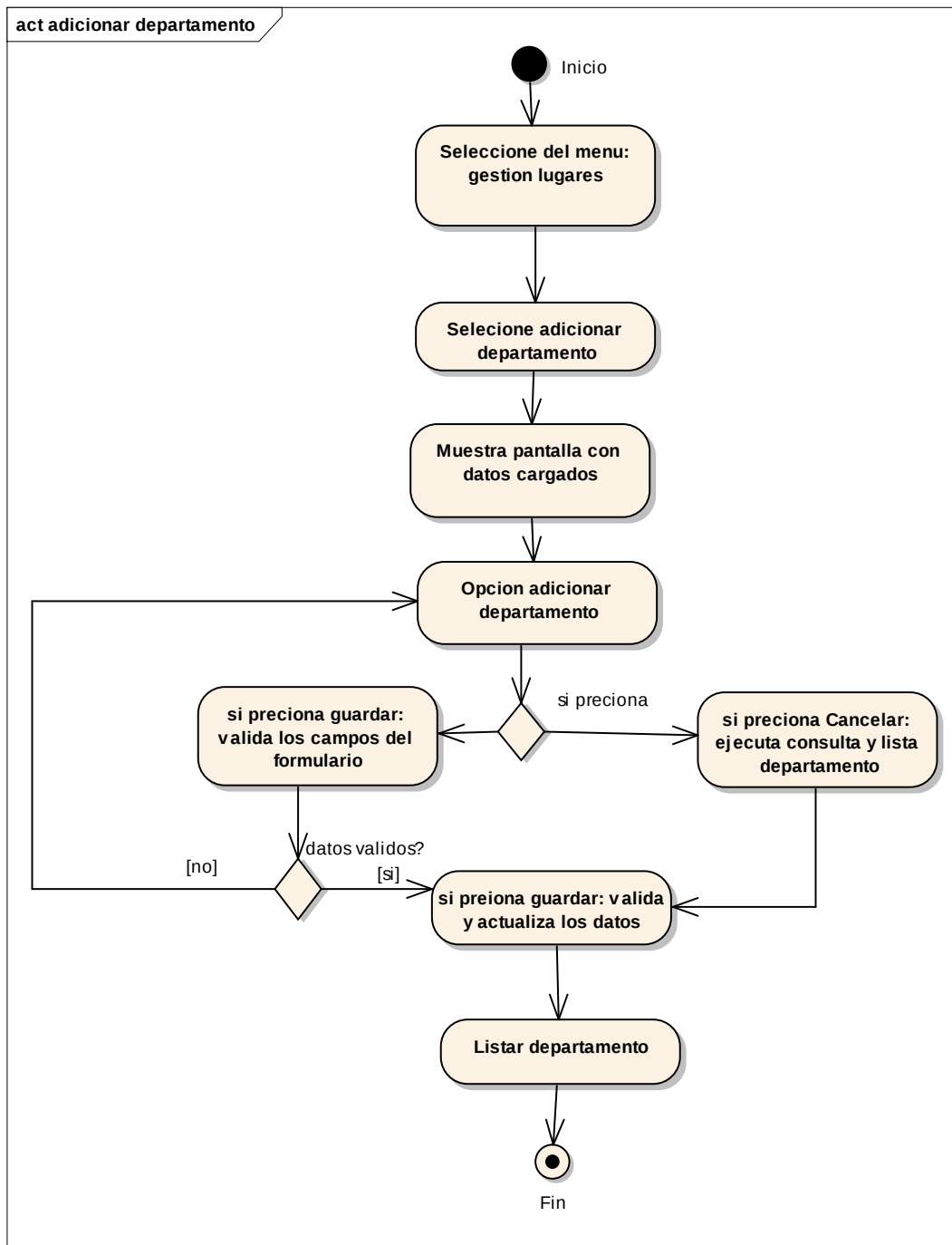


Figura n° 83 Diagrama de Actividad del C.U.: Adicionar Departamento

Diagrama de Actividad del C.U.: Modificar Departamento

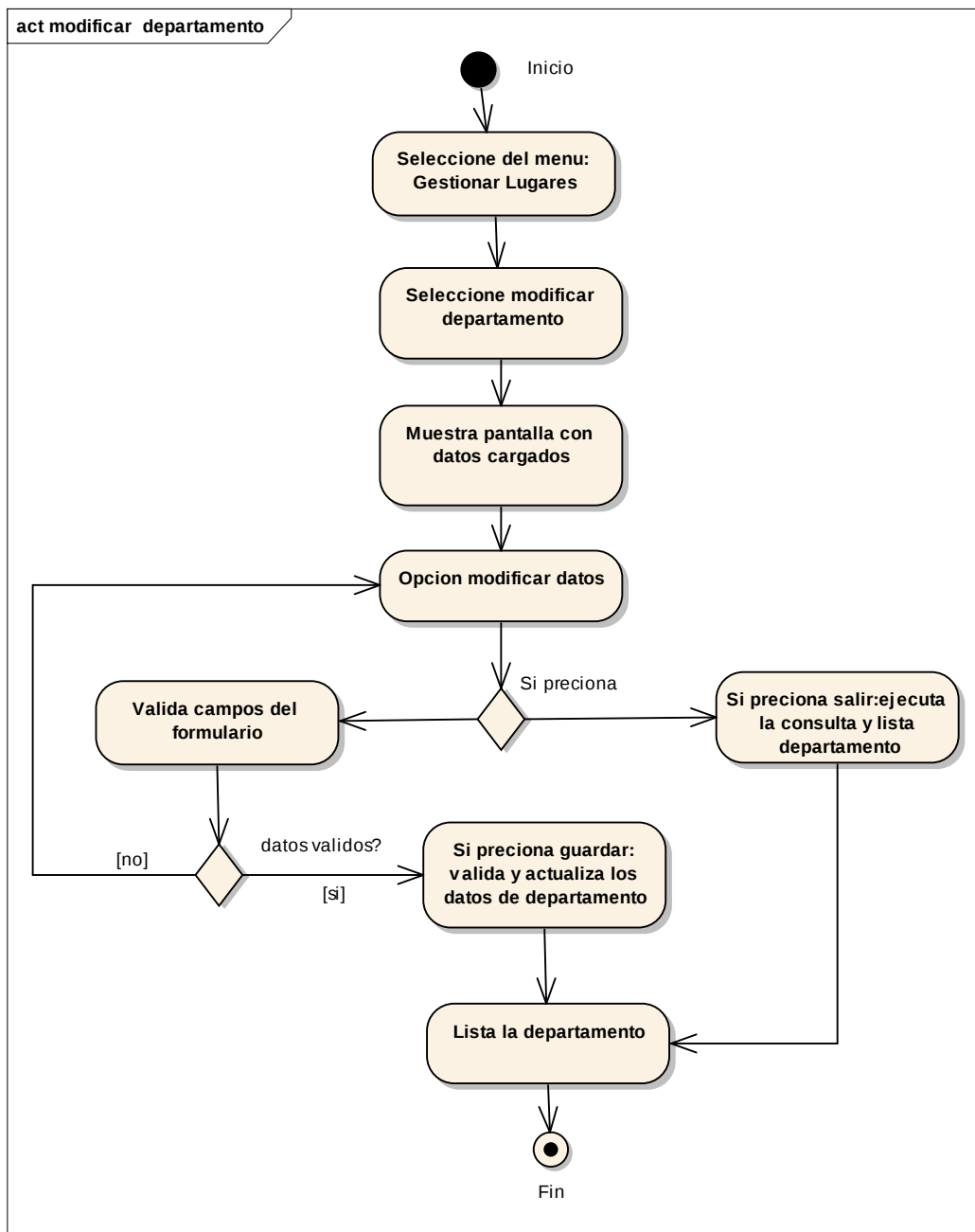


Figura nº 84 Diagrama de Actividad del C.U.: Modificar Departamento

Diagrama de Actividad del C.U.: Gestionar Provincias

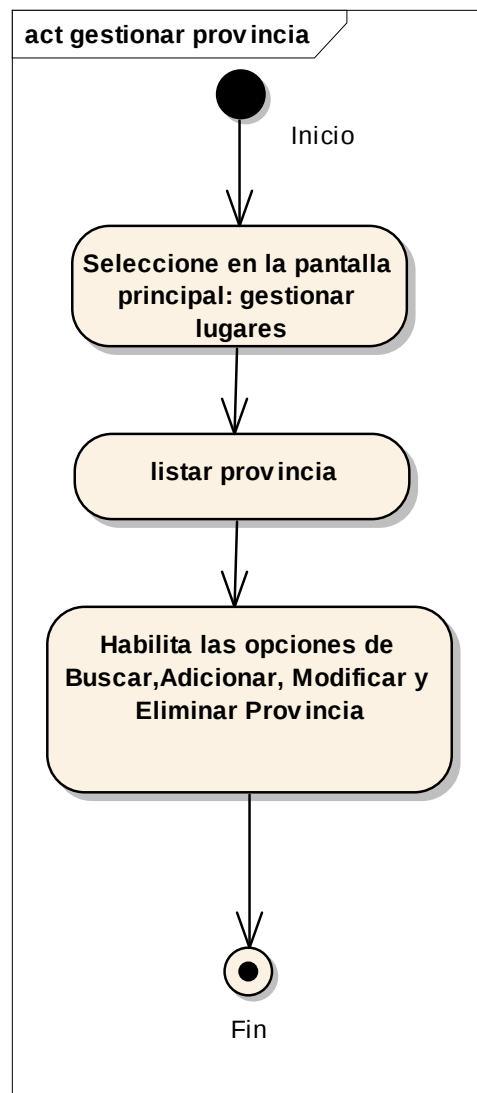


Figura nº 85 Diagrama de Actividad del C.U.: Gestionar Provincias

Diagrama de Actividad del C.U.: Adicionar Provincias

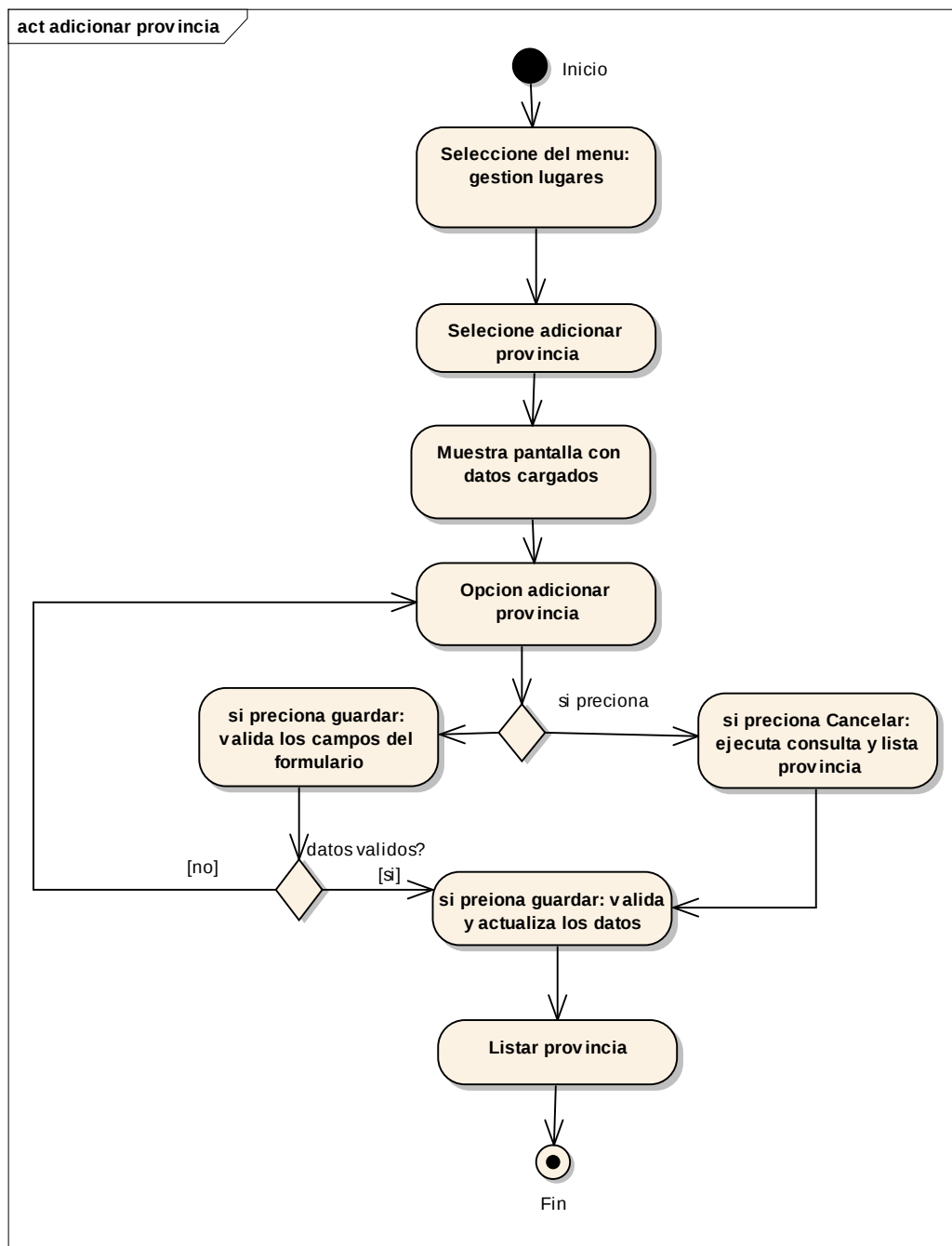


Figura nº 86 Diagrama de Actividad del C.U.: Adicionar Provincias

Diagrama de Actividad del C.U.: Modificar Provincia

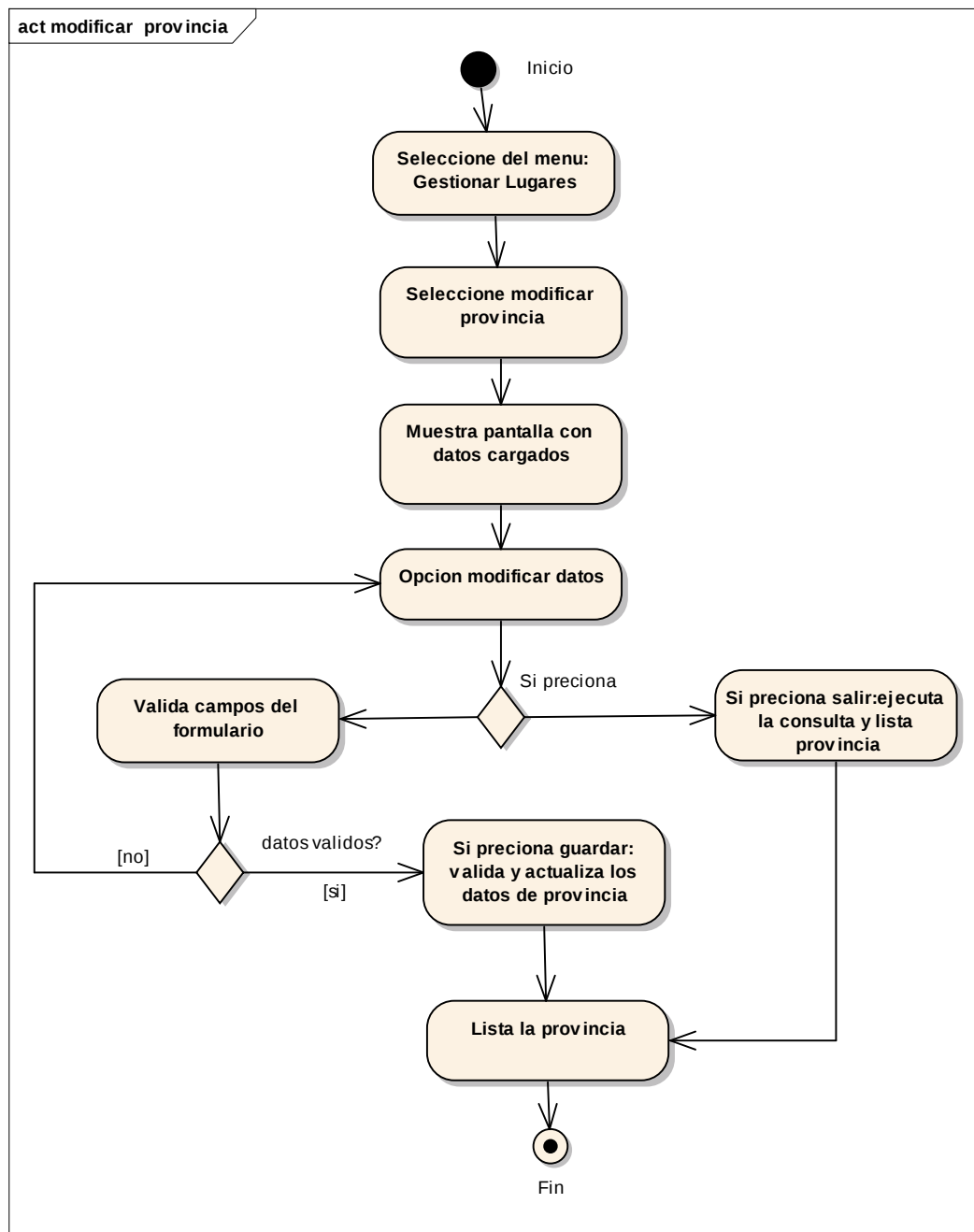


Figura nº 87 Diagrama de Actividad del C.U.: Modificar Provincia

Diagrama de Actividad del C.U.: Gestionar nivel

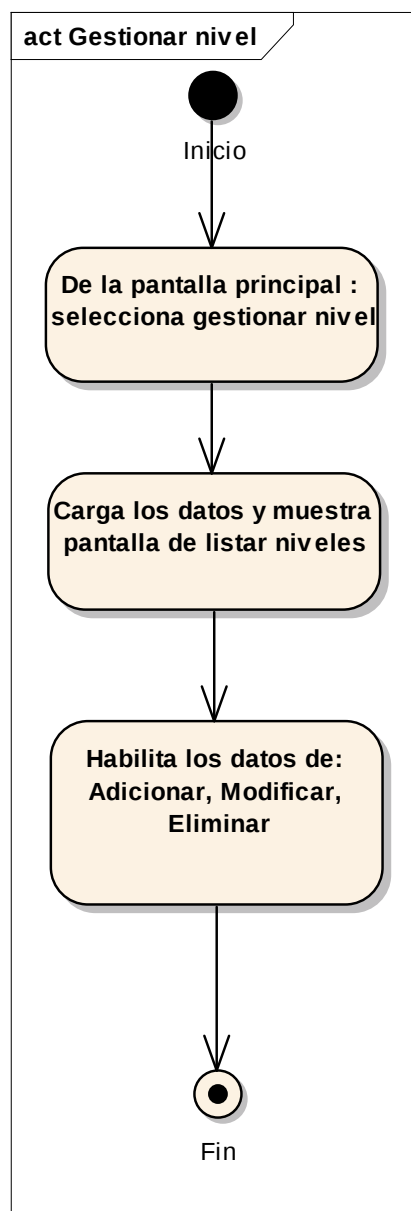


Figura N° 88 Diagrama de Actividad del C.U.: Gestionar nivel

Diagrama de Actividad del C.U.: Adiciona nivel

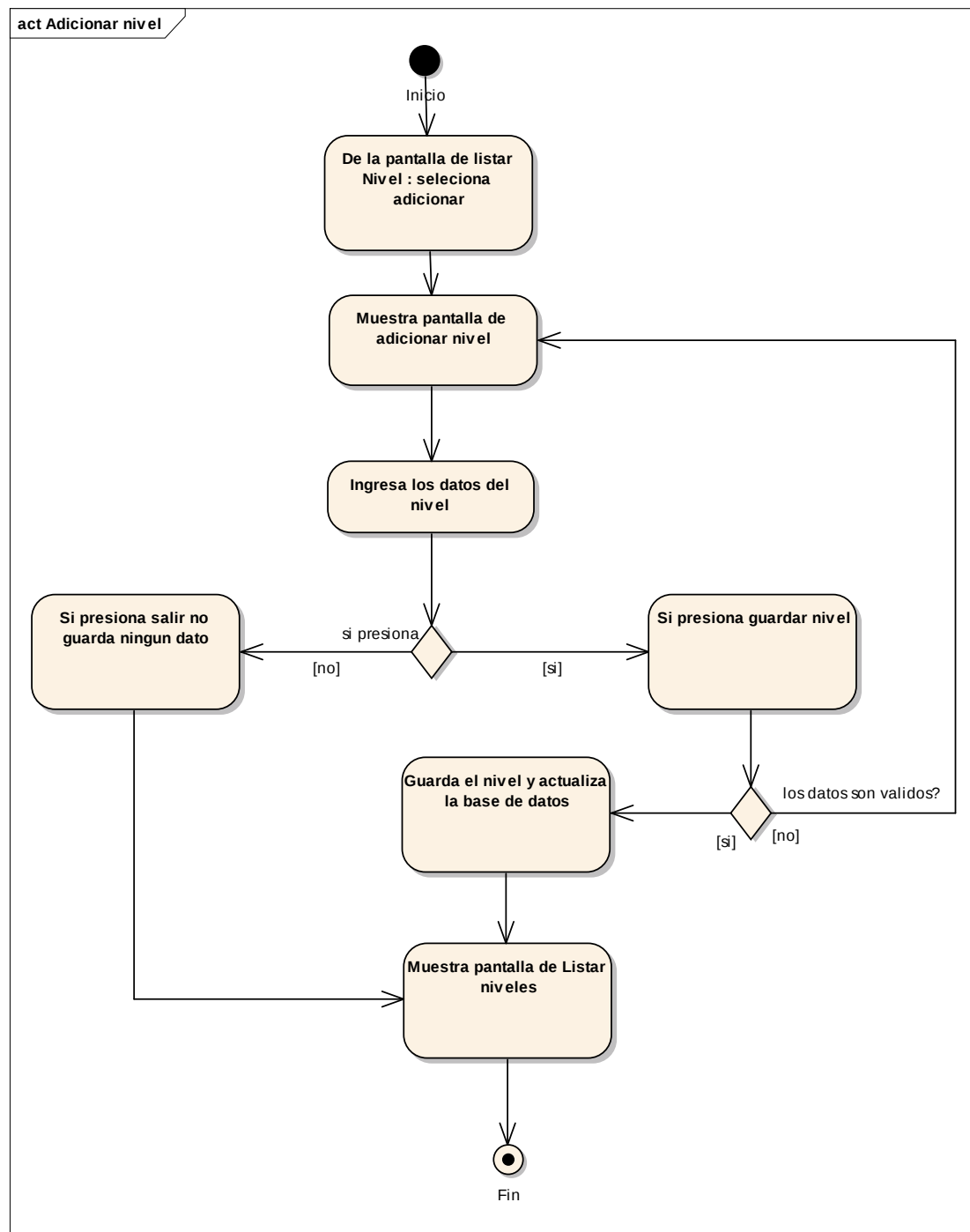


Figura N° 89 Diagrama de Actividad del C.U.: Adiciona nivel

Diagrama de Actividad del C.U.: Modificar nivel

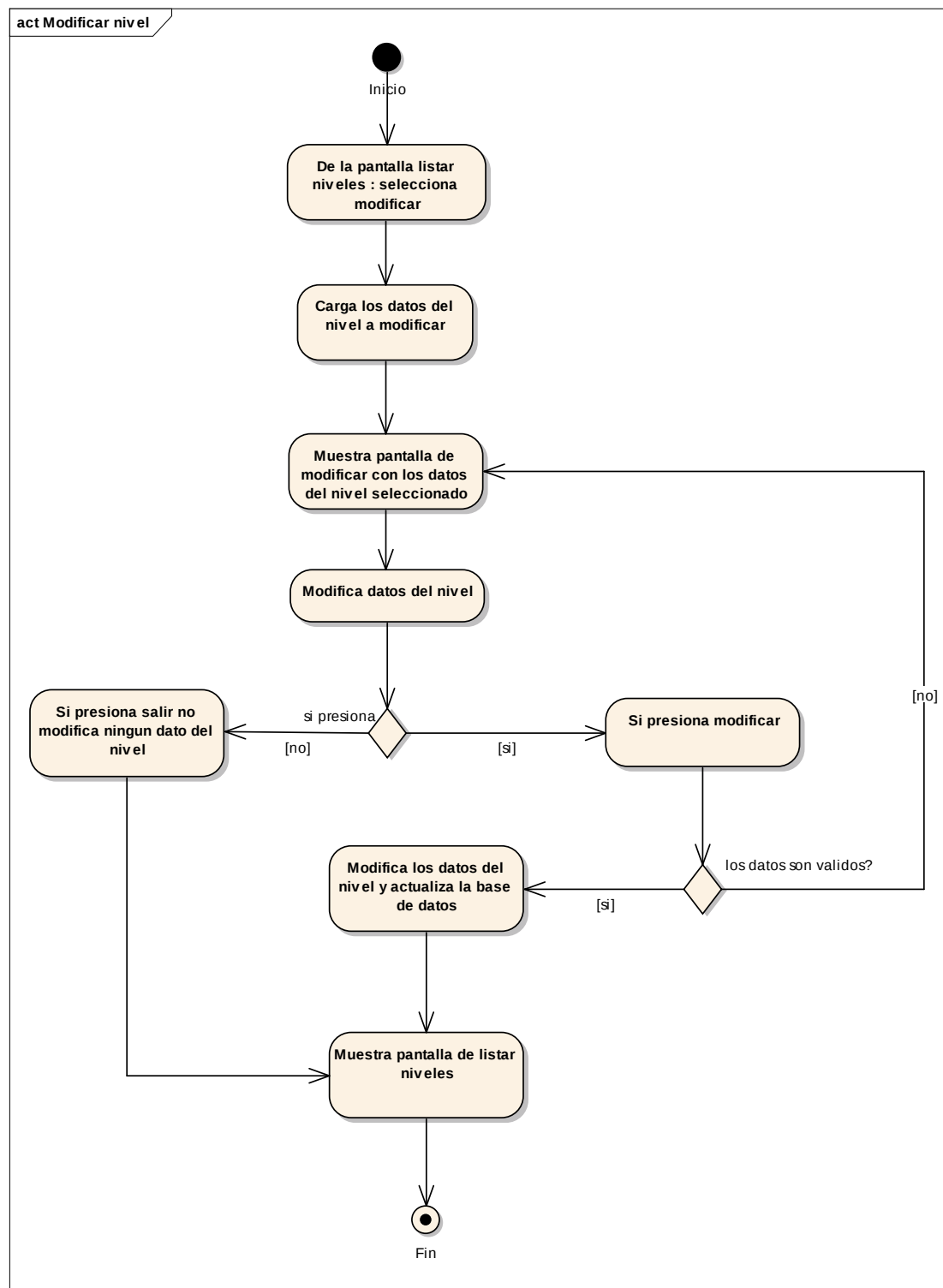


Figura N° 90 Diagrama de Actividad del C.U.: Modificar nivel

Diagrama de Actividad del C.U.: Gestionar grados

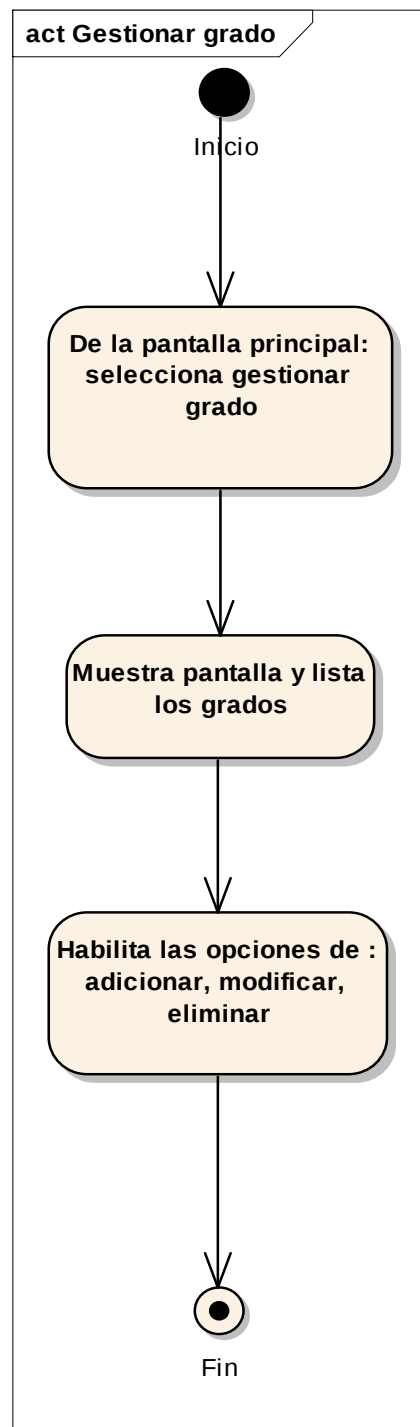


Figura N° 91 Diagrama de Actividad del C.U.: Gestionar grado

Diagrama de Actividad del C.U.: Adicionar grado

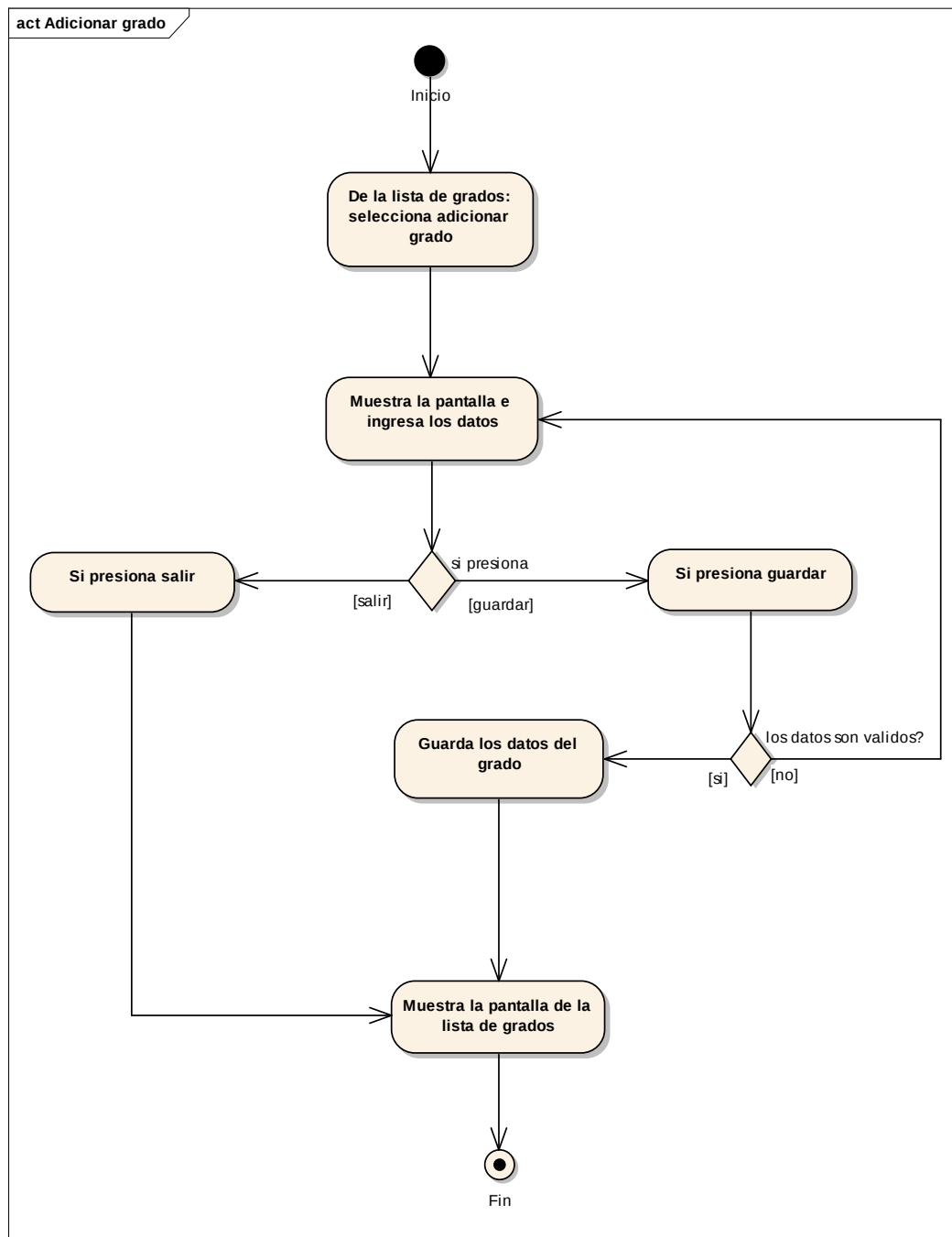


Figura N° 92 Diagrama de Actividad del C.U.: Adicionar grado

Diagrama de Actividad del C.U.: Modificar grado

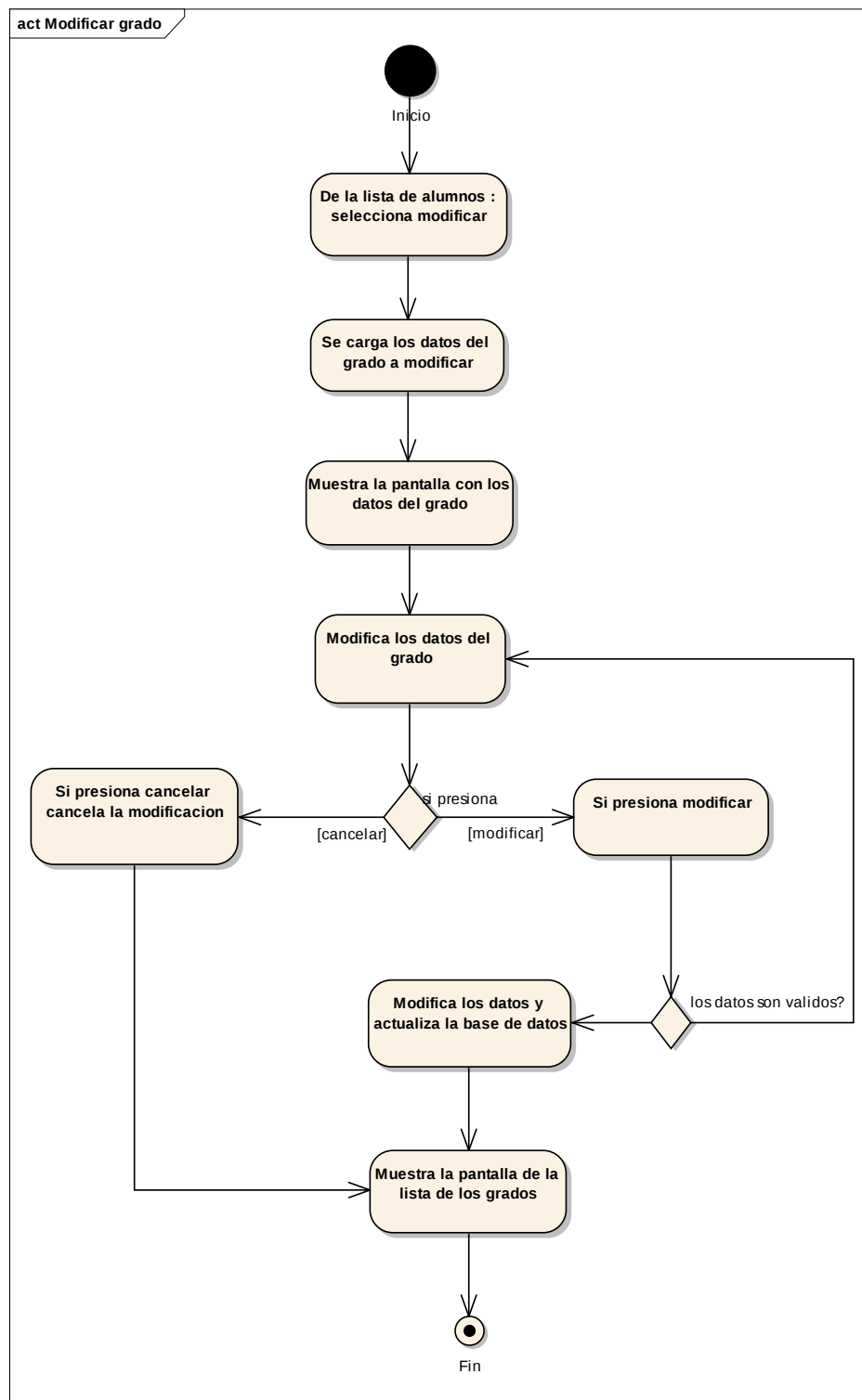


Figura N° 93 Diagrama de Actividad del C.U.: Modificar grado

Diagrama de Actividad del C.U.: Gestionar grupo

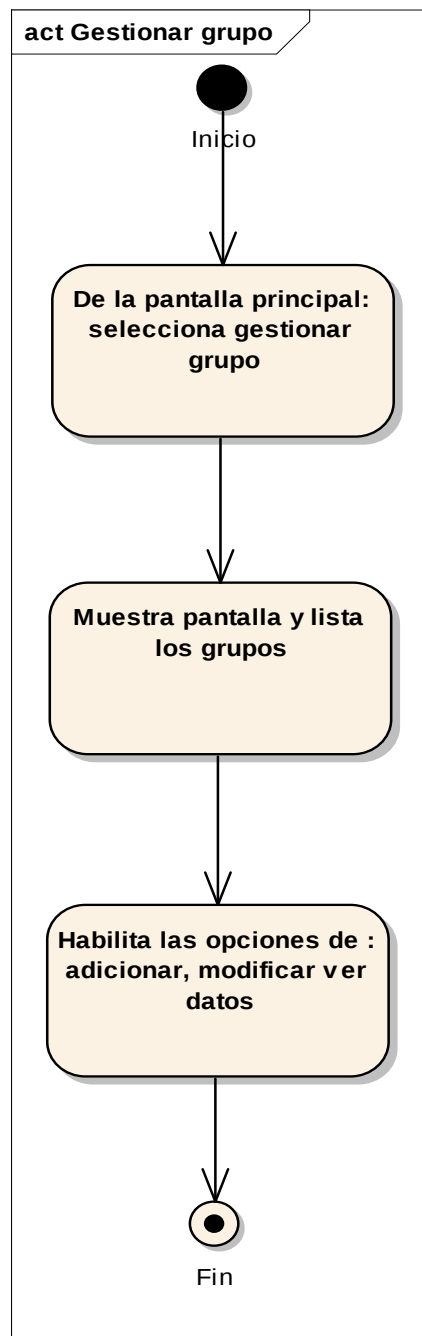


Figura N° 94 Diagrama de Actividad del C.U.: Gestionar grupo

Diagrama de Actividad del C.U.: Adicionar grupo

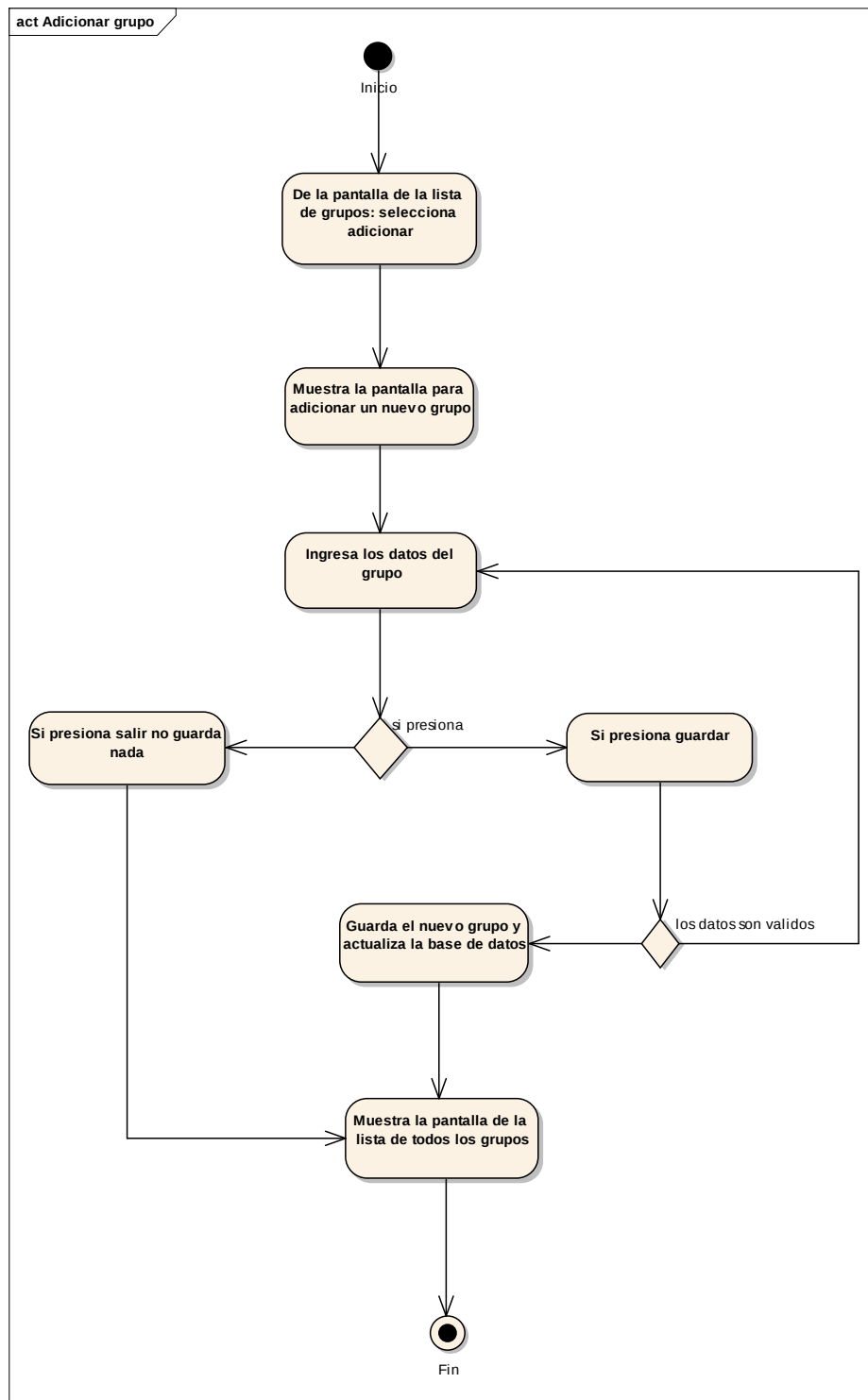


Figura N° 95 Diagrama de Actividad del C.U.: Adicionar grupo

Diagrama de Actividad del C.U.: Modificar grupo

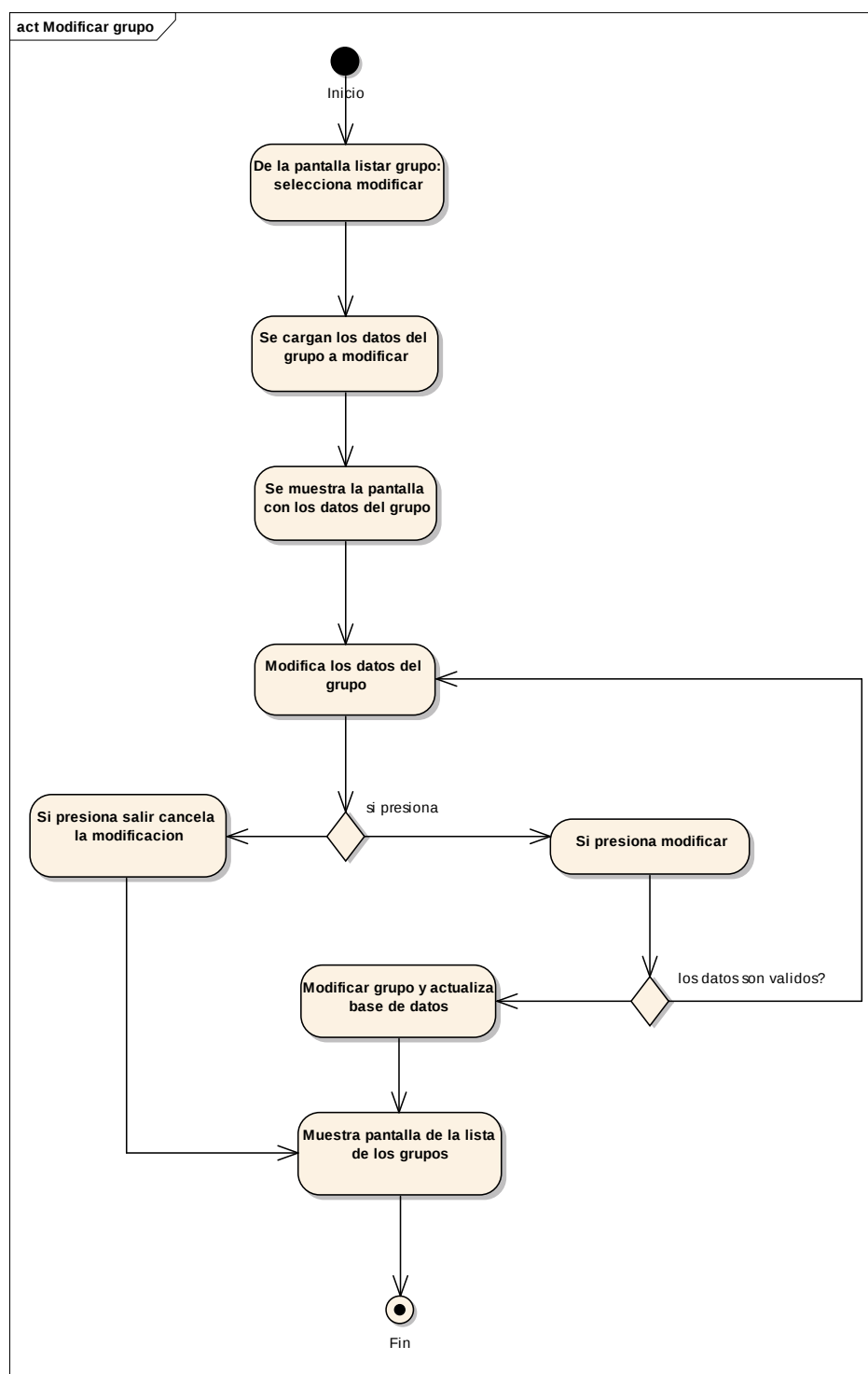


Figura N° 96 Diagrama de Actividad del C.U.: Modificar grupo

Diagrama de Actividad del C.U.: Gestionar profesor

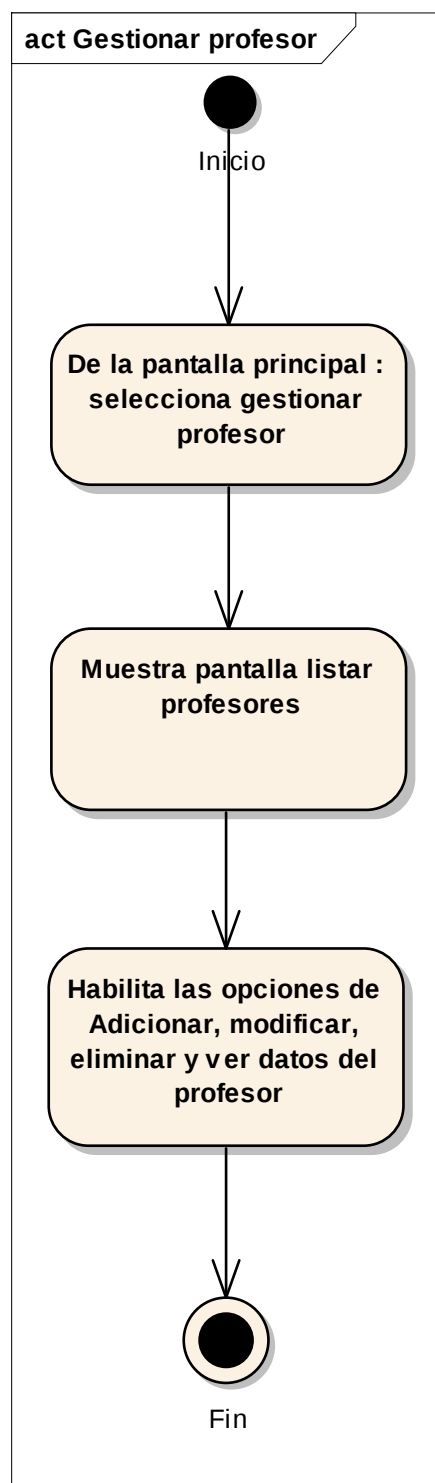


Figura N° 97 Diagrama de Actividad del C.U.: Gestionar profesor

Diagrama de Actividad del C.U.: Adicionar Profesor

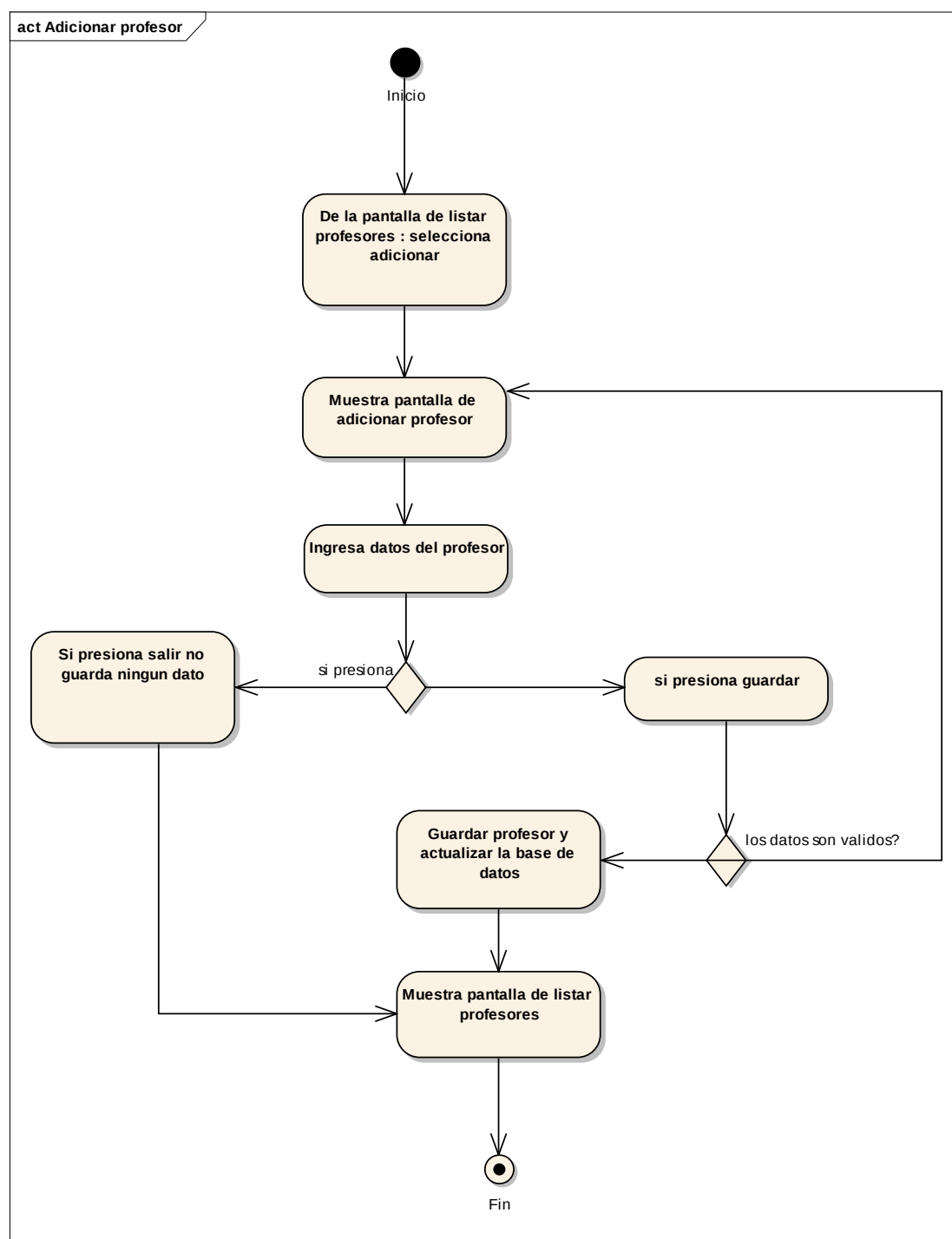


Figura N° 98 Diagrama de Actividad del C.U.: Adicionar Profesor

Diagrama de Actividad del C.U.: Modificar profesor

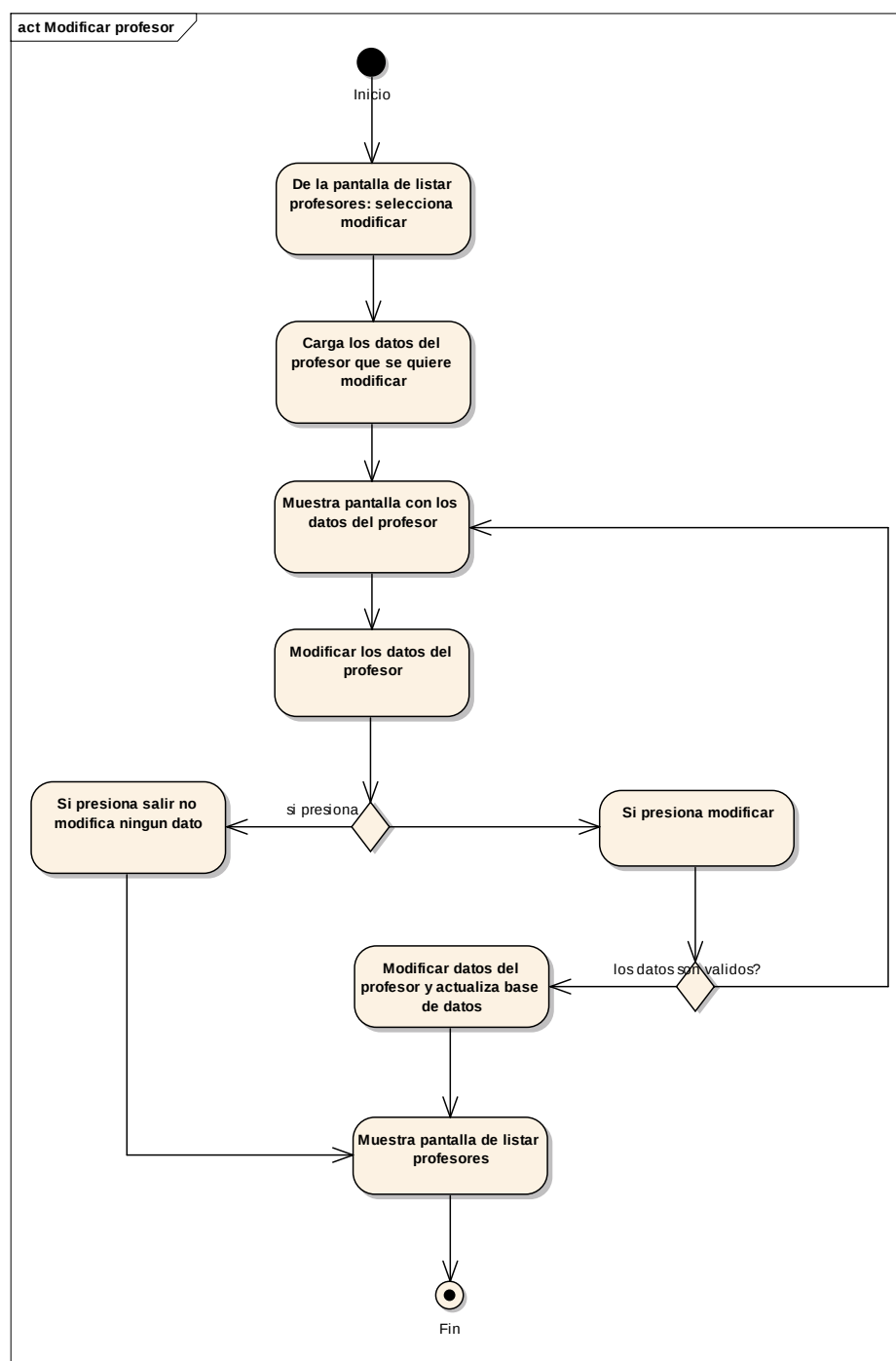


Figura N° 99 Diagrama de Actividad del C.U.: Modificar profesor

Diagrama de Actividad del C.U.: Gestionar materia

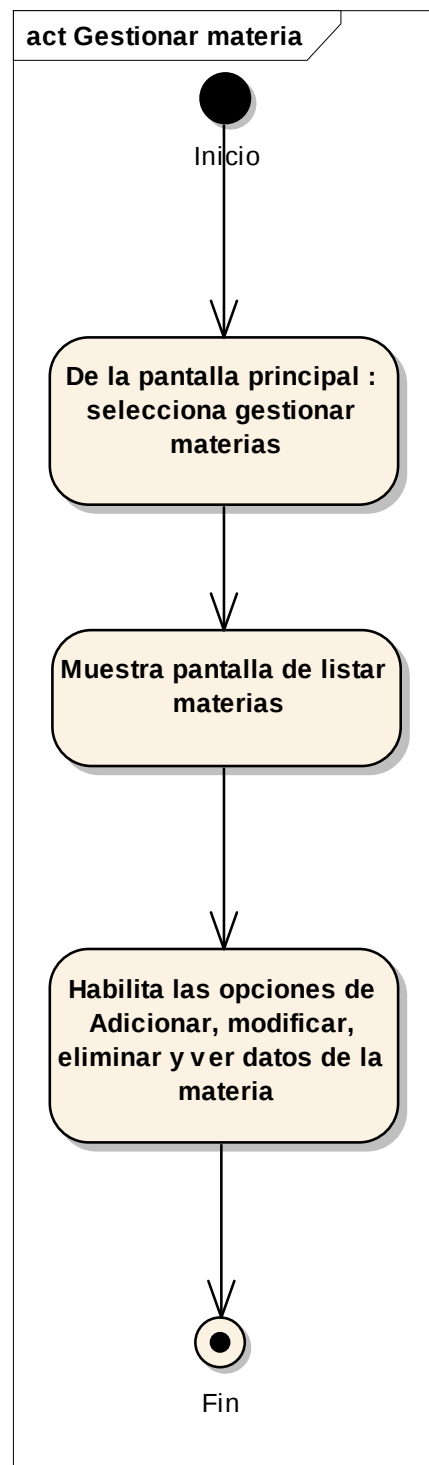


Figura N° 100 Diagrama de Actividad del C.U.: Gestionar materia

Diagrama de Actividad del C.U.: Adicionar materias

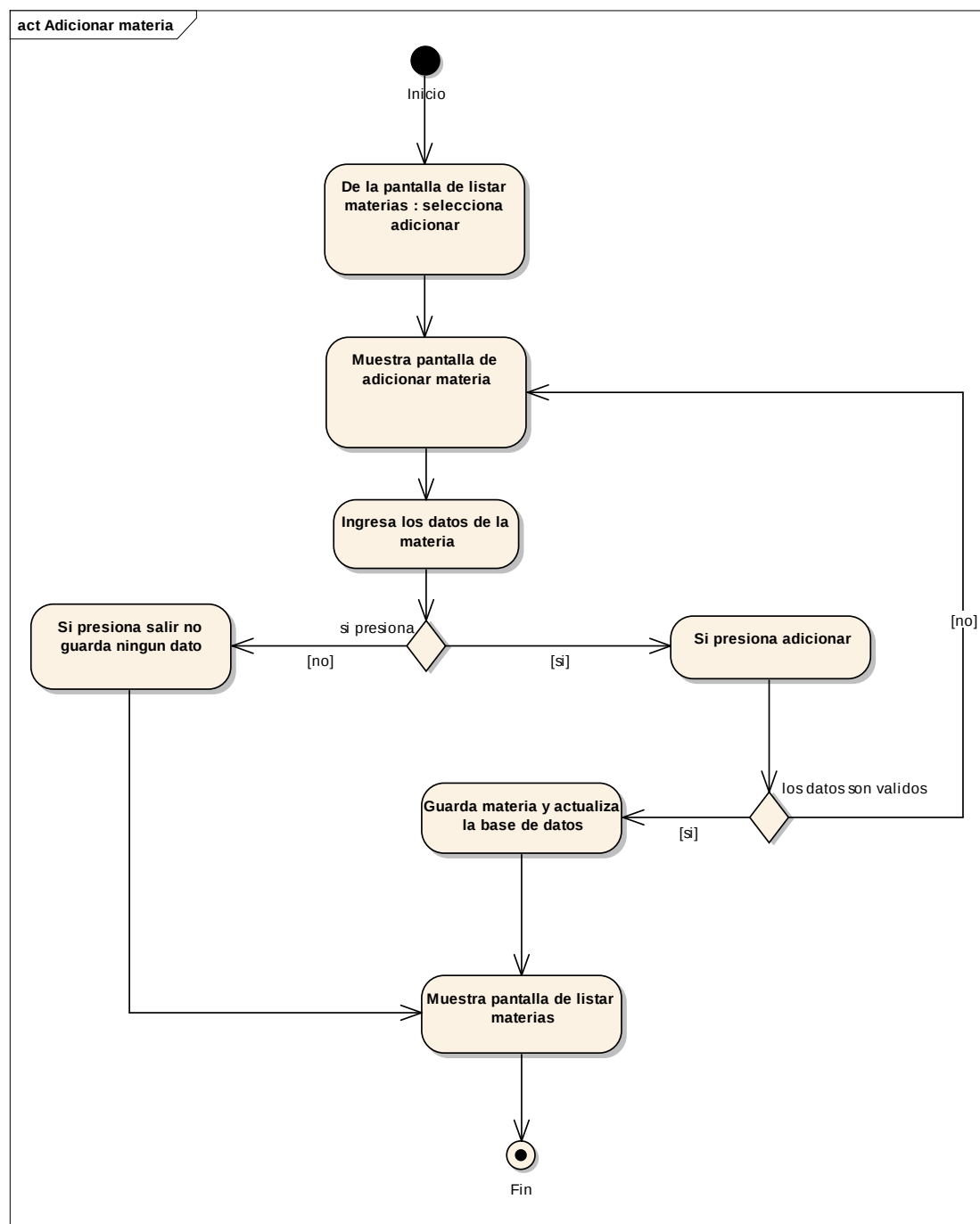


Figura N° 101 Diagrama de Actividad del C.U.: Adicionar materias

Diagrama de Actividad del C.U.: Modificar materia

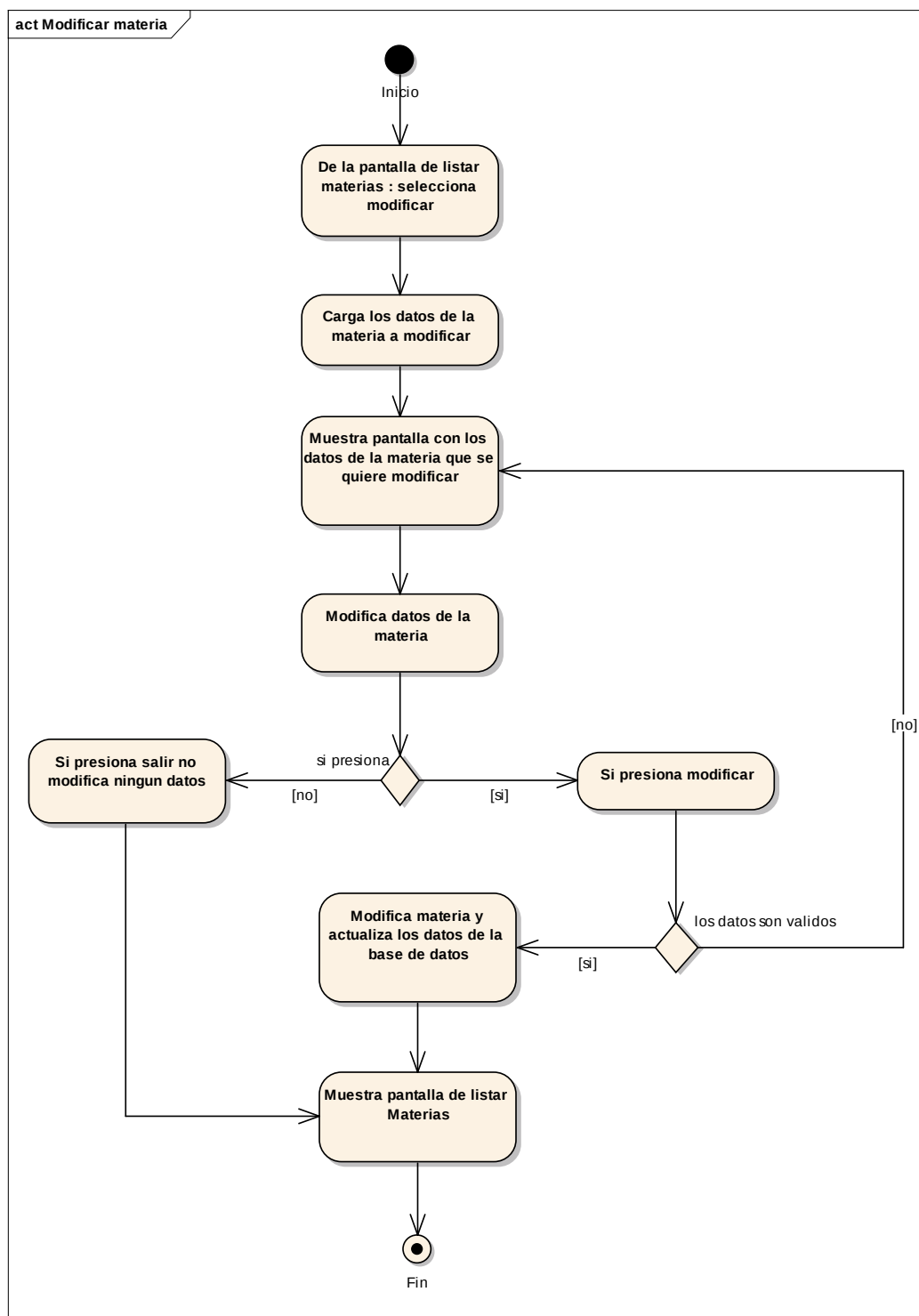


Figura N° 102 Diagrama de Actividad del C.U.: Modificar materia

Diagrama de Actividad del C.U.: Gestionar materia a grado

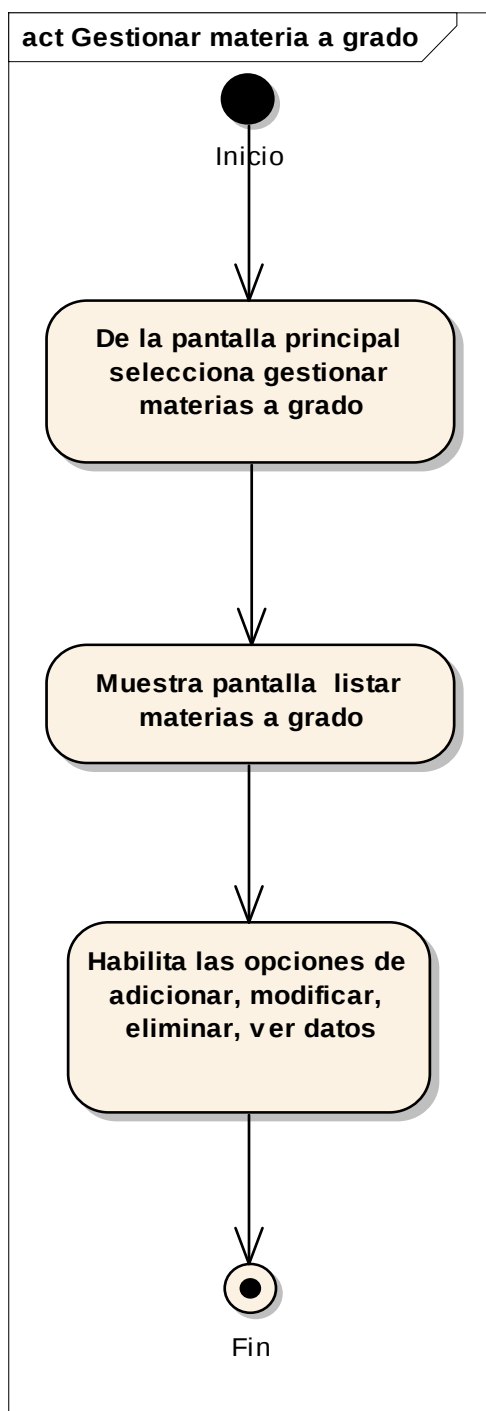


Figura nº 103 Diagrama de Actividad del C.U.: Gestionar materia a grado

Diagrama de Actividad del C.U.: Adicionar materias a los grados

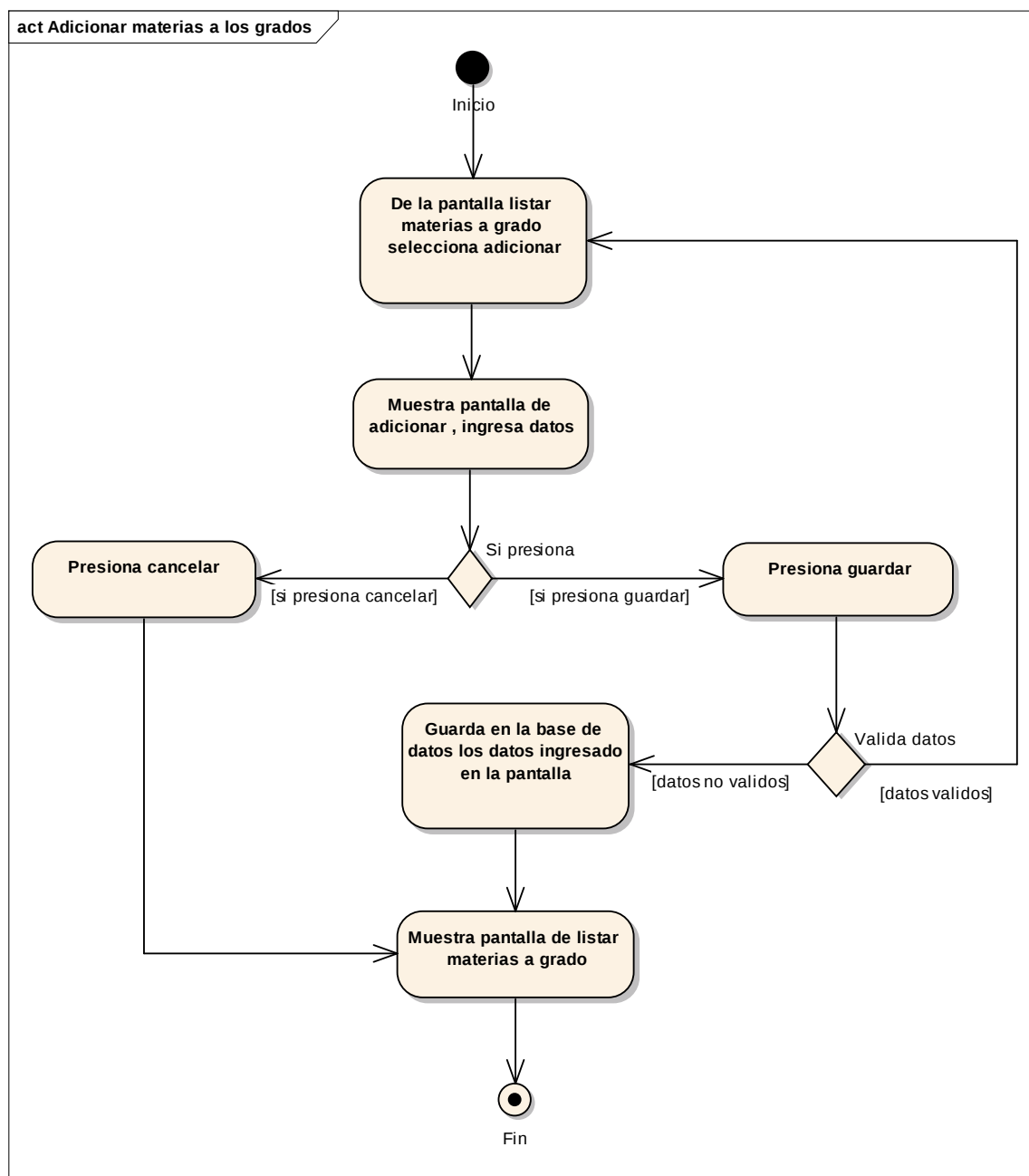


Figura nº 104 Diagrama de Actividad del C.U.: Adicionar materias a los grados

Diagrama de Actividad del C.U.: Modificar materias a grado

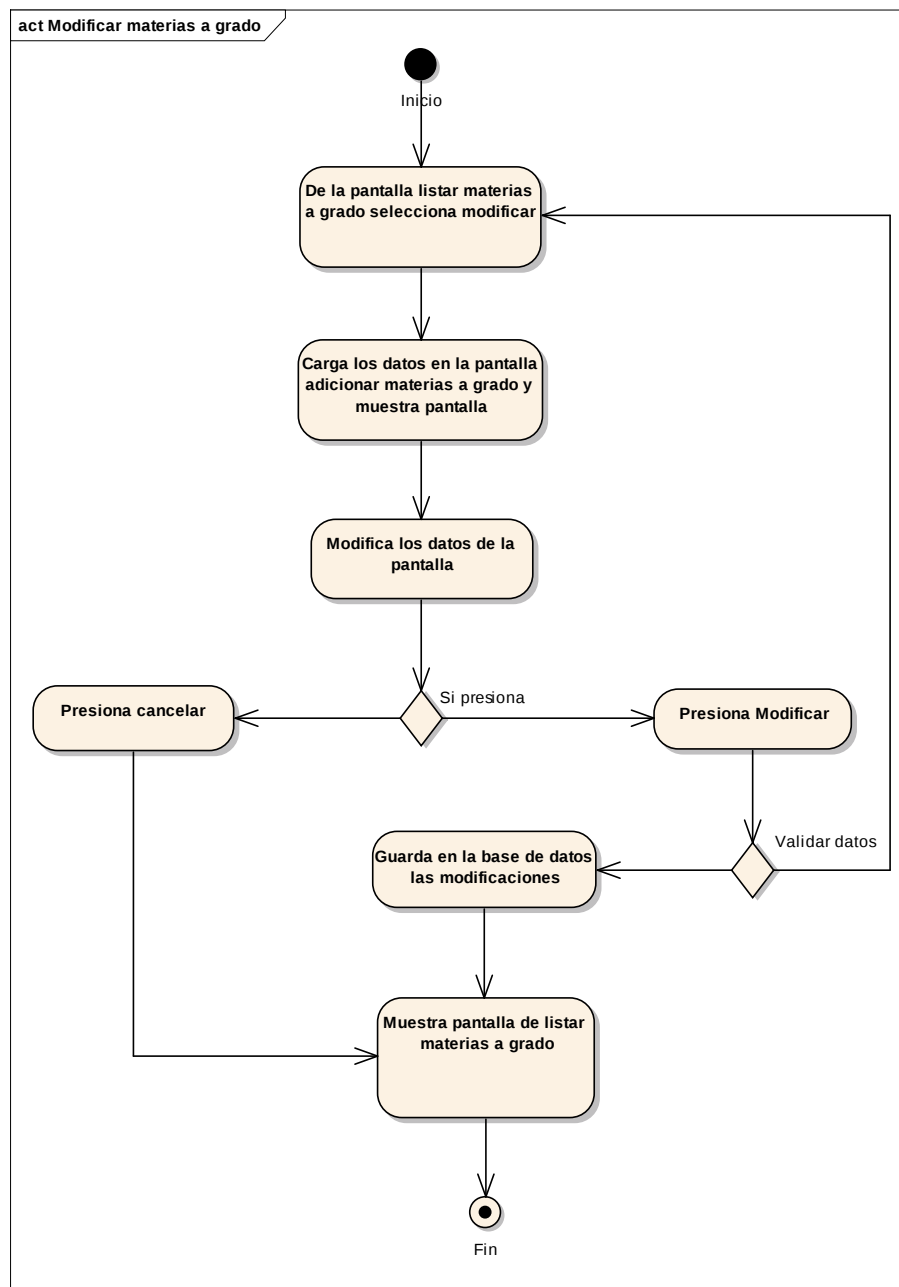


Figura nº 105 Diagrama de Actividad del C.U.: Modificar materias a grado

Diagrama de Actividad del C.U.: Asignar materias a profesor

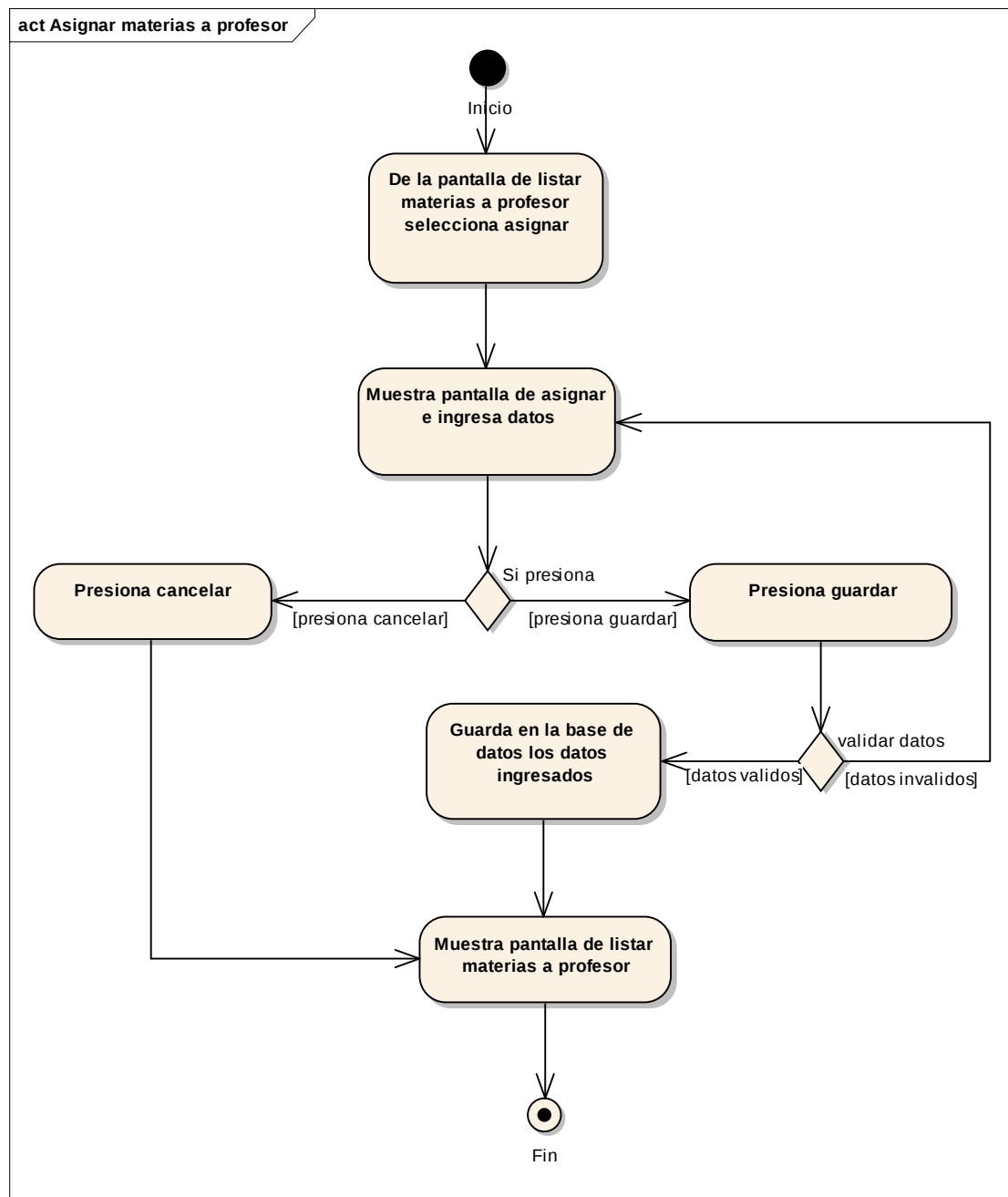


Figura n° 106 Diagrama de Actividad del C.U.:Asignar materias a profesor

Diagrama de Actividad del C.U.: Gestionar materias a los profesores

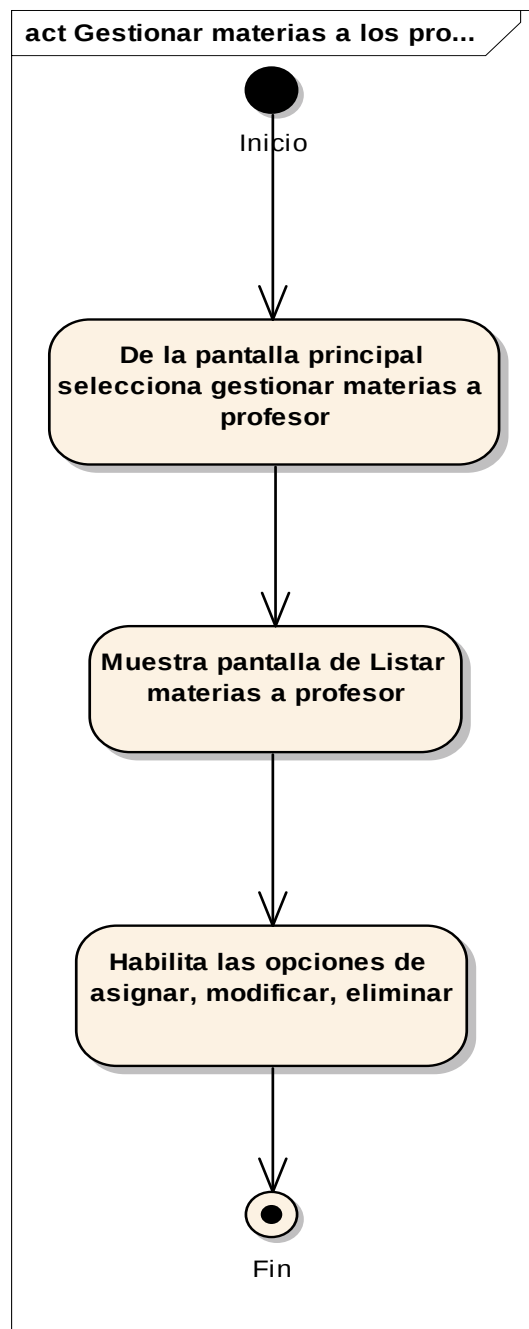


Figura nº 107 Diagrama de Actividad del C.U.: Gestionar materias a los profesores

Diagrama de Actividad del C.U.: Gestionar materias a los profesores

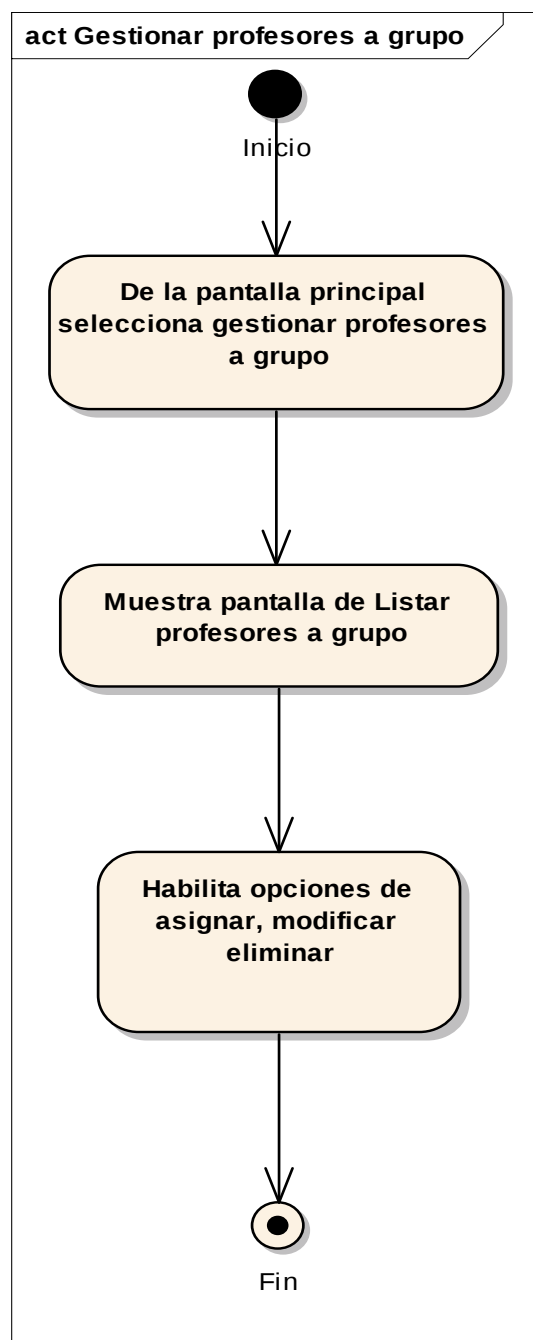


Figura nº 108 Diagrama de Actividad del C.U.: Gestionar materias a los profesores

Diagrama de Actividad del C.U.: Asignar profesores a grupo

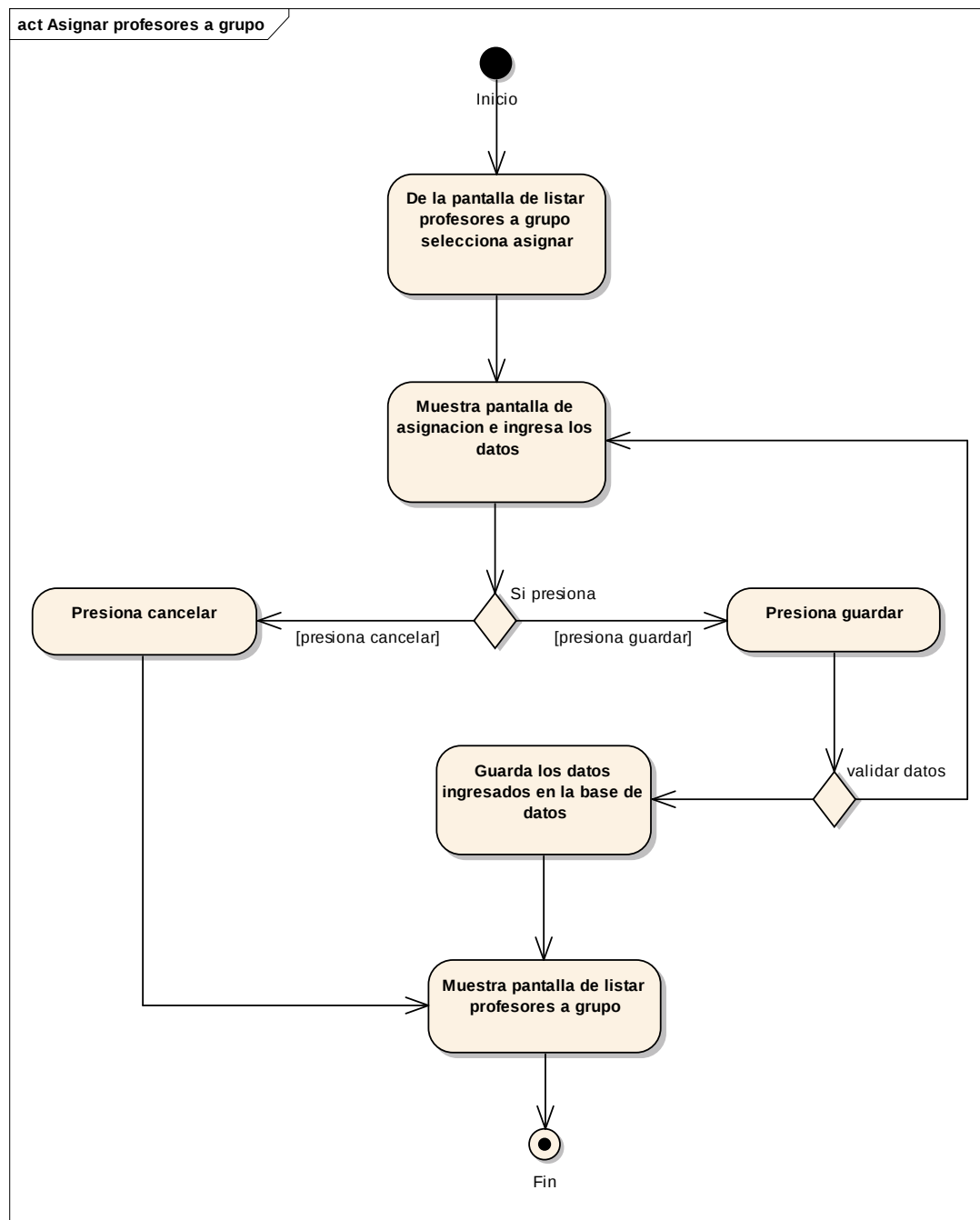


Figura n° 109 Diagrama de Actividad del C.U.: Asignar profesores a grupo

Diagrama de Actividad del C.U.: Gestionar Reportes

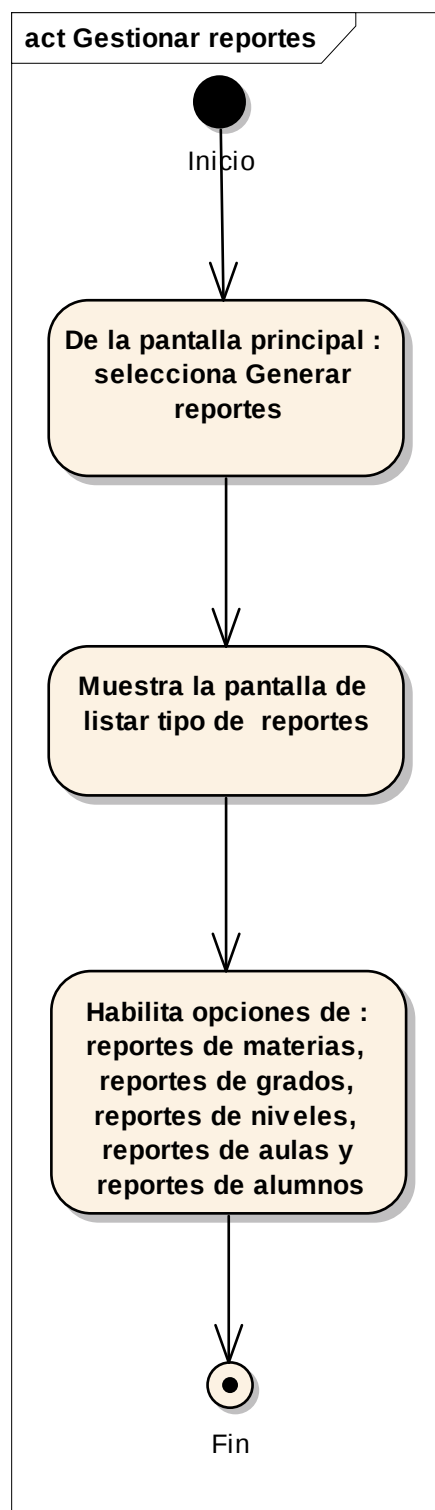


Figura N° 110 Diagrama de Actividad del C.U.: Gestionar Reportes

Diagrama de Actividad del C.U.: Reportes de aulas

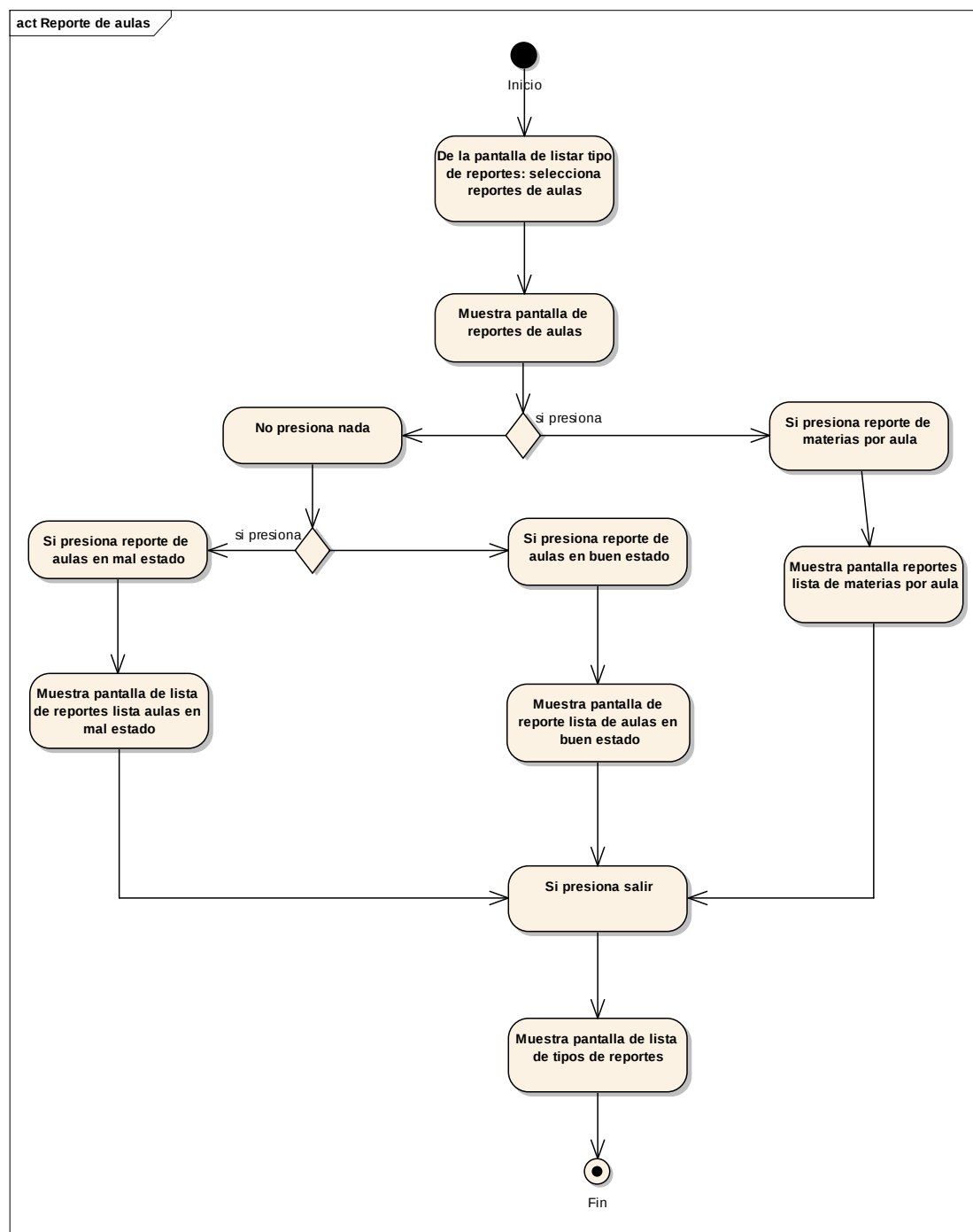


Figura N° 111 Diagrama de Actividad del C.U.: Reportes de aulas

Diagrama de Actividad del C.U.: Reportes de niveles

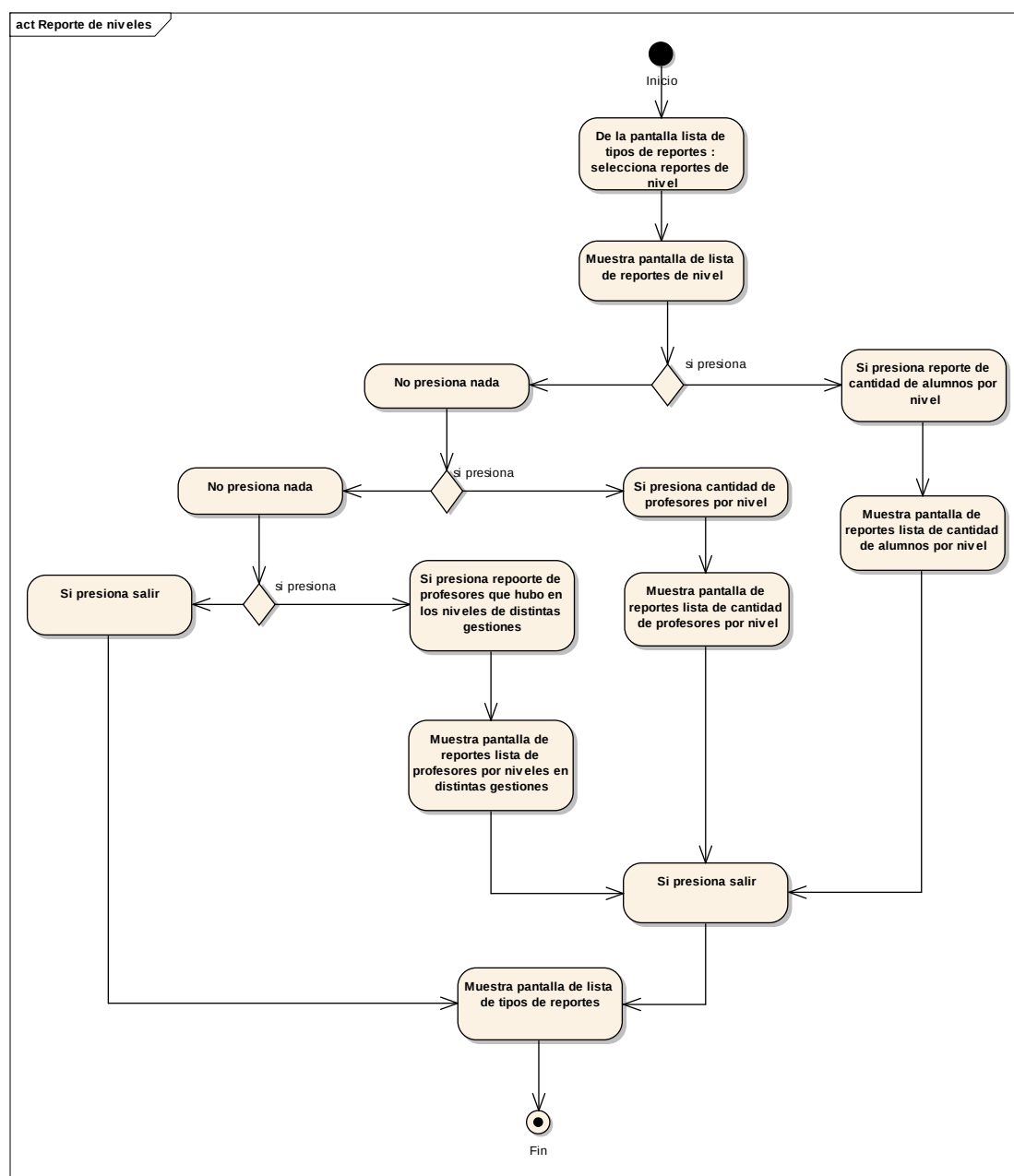


Figura N° 112 Diagrama de Actividad del C.U.: Reportes de niveles

Diagrama de Actividad del C.U.: Reportes de alumnos

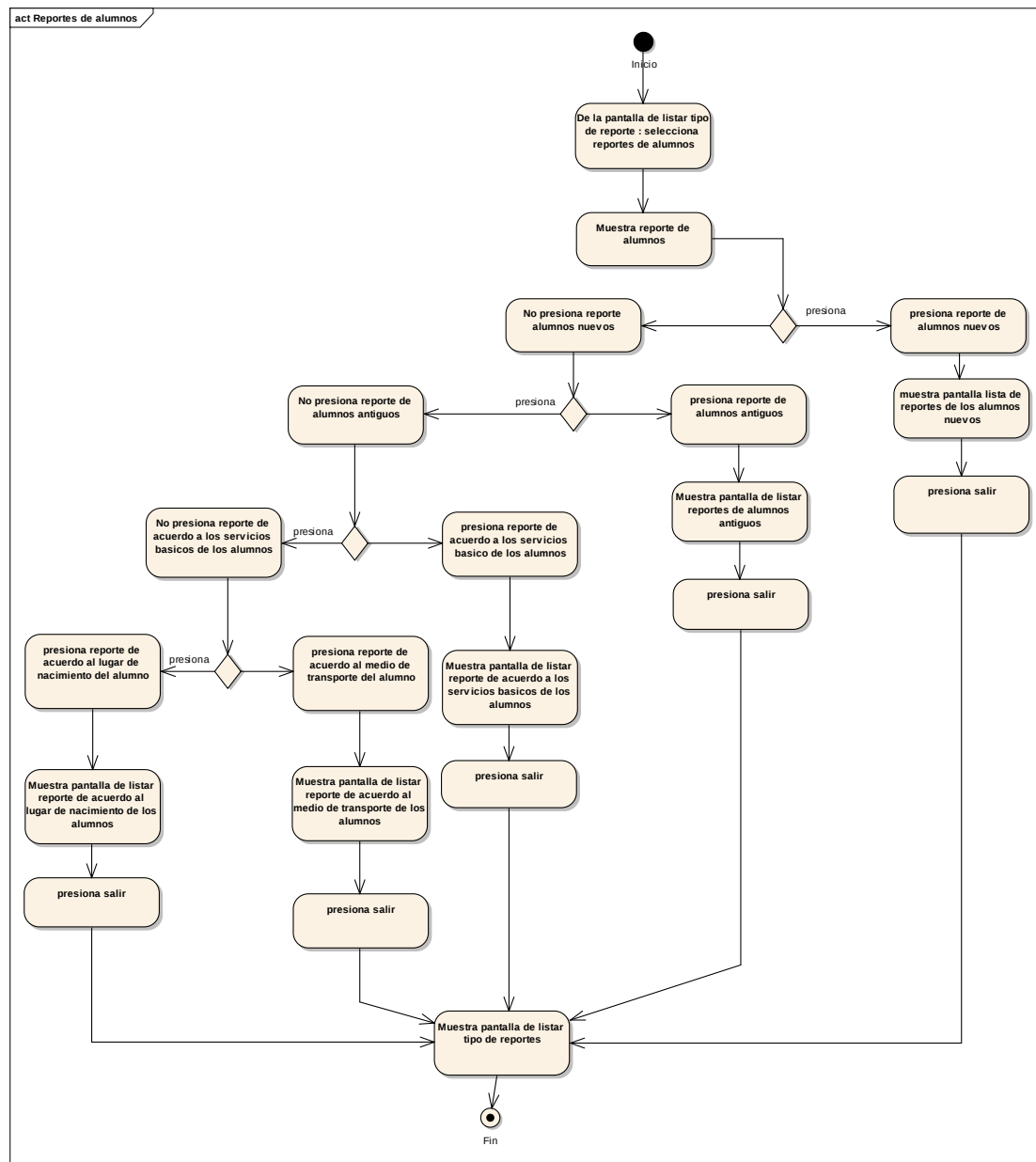


Figura N° 113 Diagrama de Actividad del C.U.: Reportes de alumnos

Diagrama de Actividad del C.U.: Reportes de grado

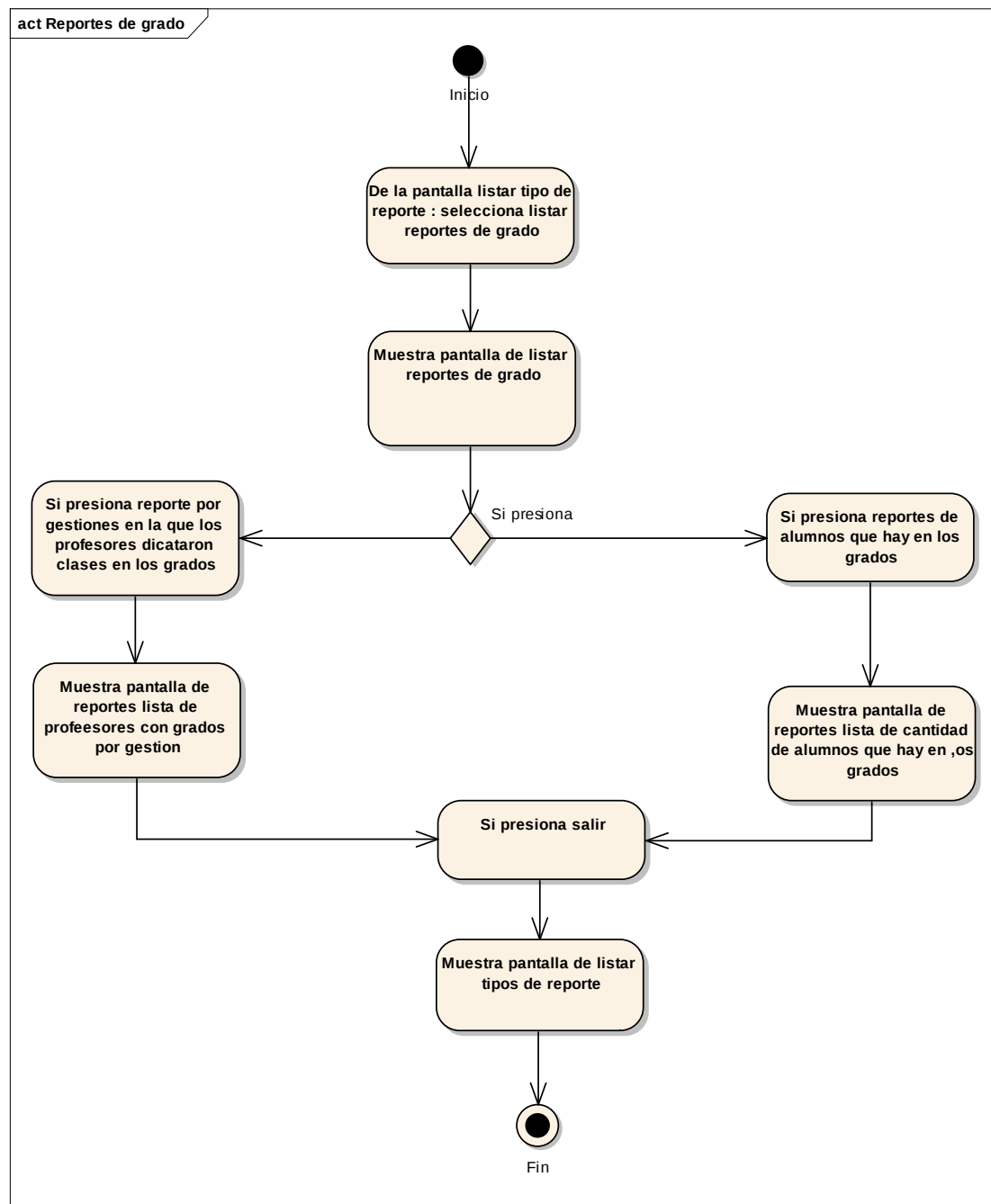


Figura N° 114 Diagrama de Actividad del C.U.: Reportes de grado

Diagrama de Actividad del C.U.: Reportes de materias

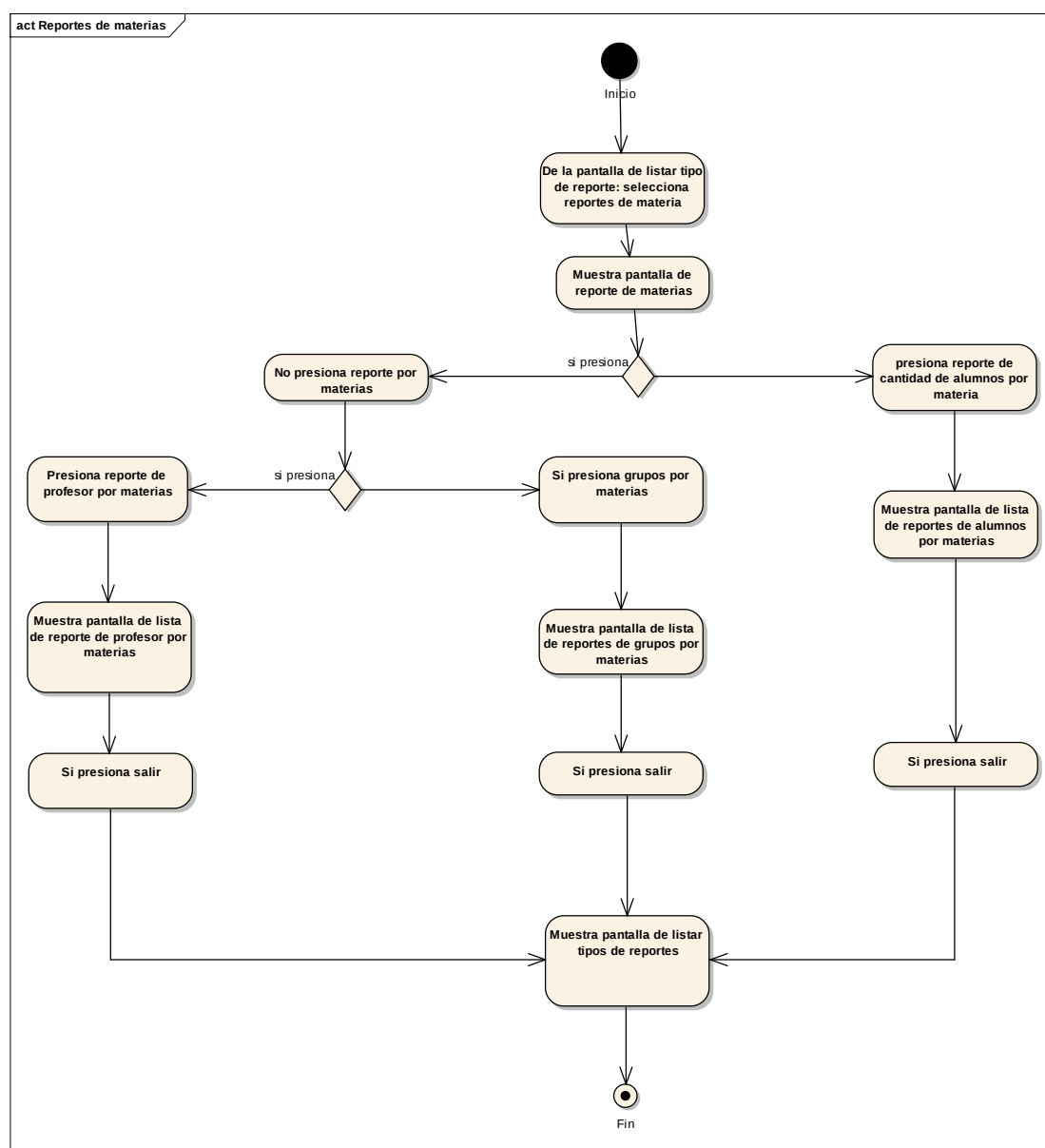


Figura N° 115 Diagrama de Actividad del C.U.: Reportes de materias

1.1.2.2.4 Diagramas de Secuencia

Diagrama de secuencia: C.U. Acceder al sistema

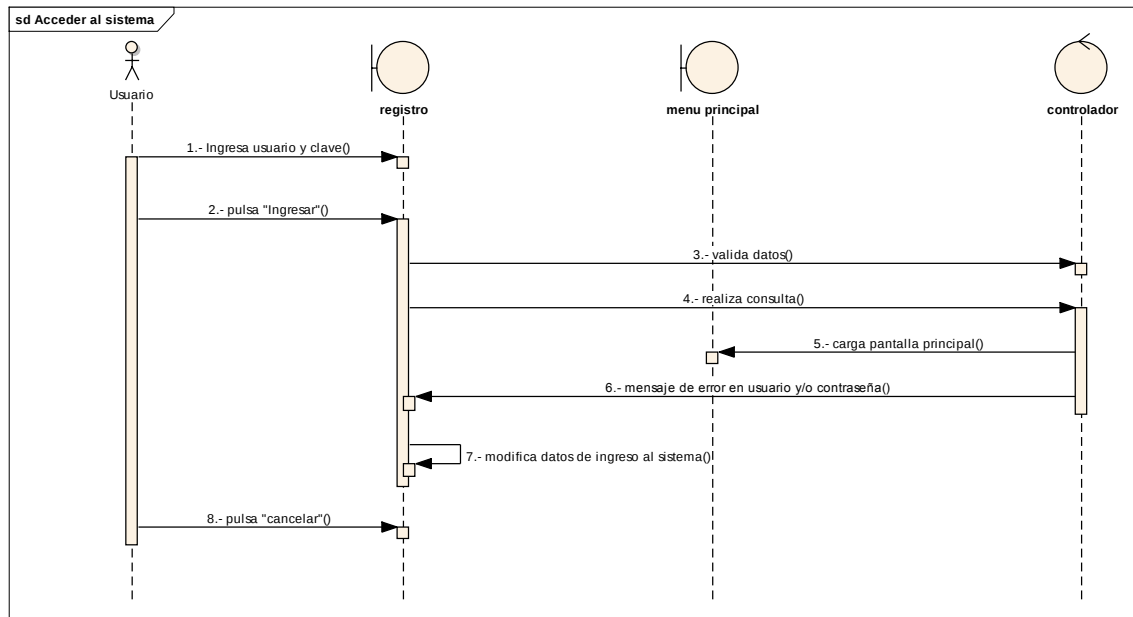


Figura nº 116 Diagrama de secuencia: C.U. Acceder al sistema

Diagrama de secuencia: C.U. Gestionar Usuarios

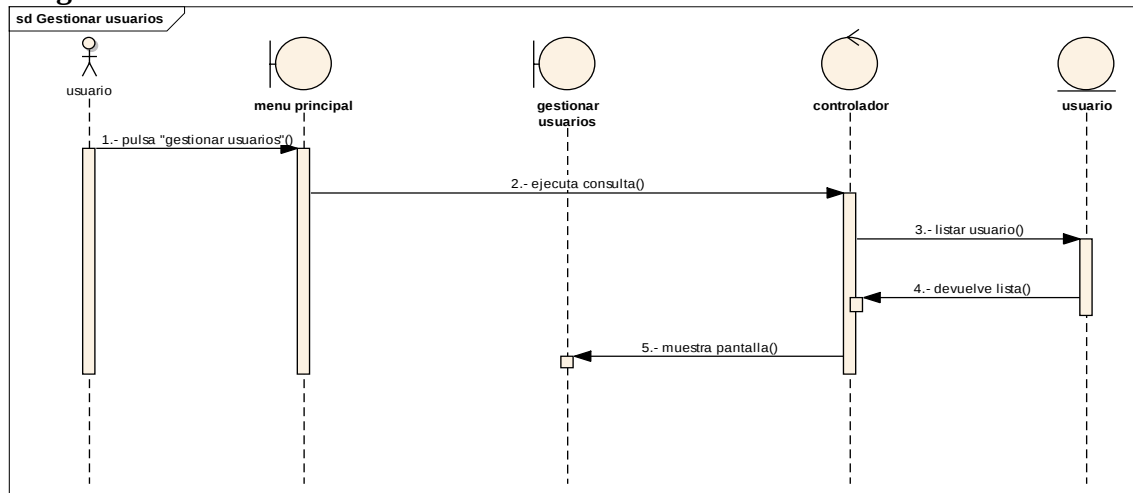


Figura nº 117 Diagrama de secuencia: C.U. Gestionar Usuarios

Diagrama de secuencia: C.U. Adicionar Usuarios

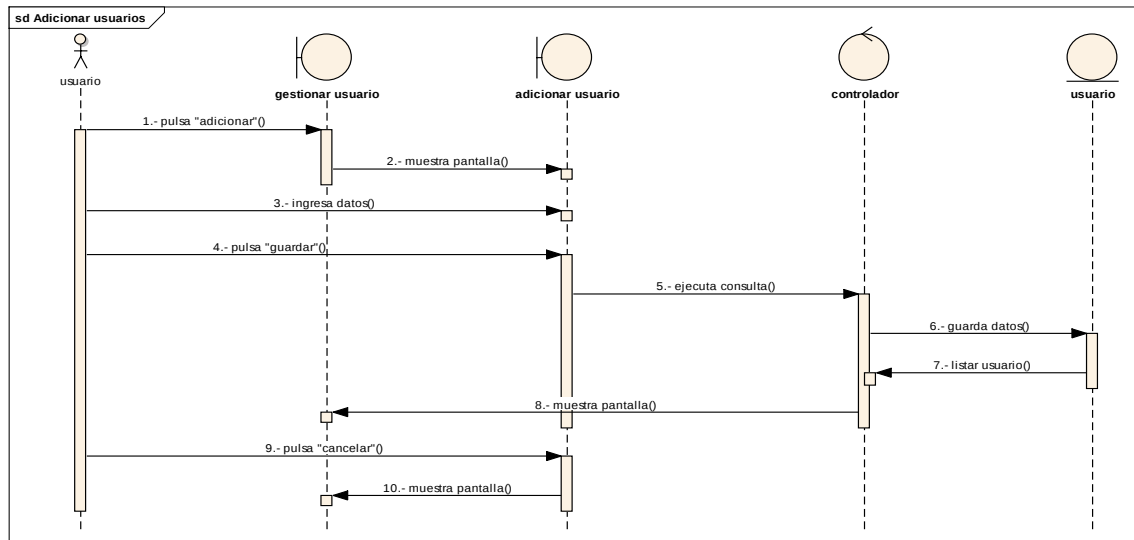


Figura nº 118 Diagrama de secuencia: C.U. Adicionar Usuarios

Diagrama de secuencia: C.U Modificar Usuarios

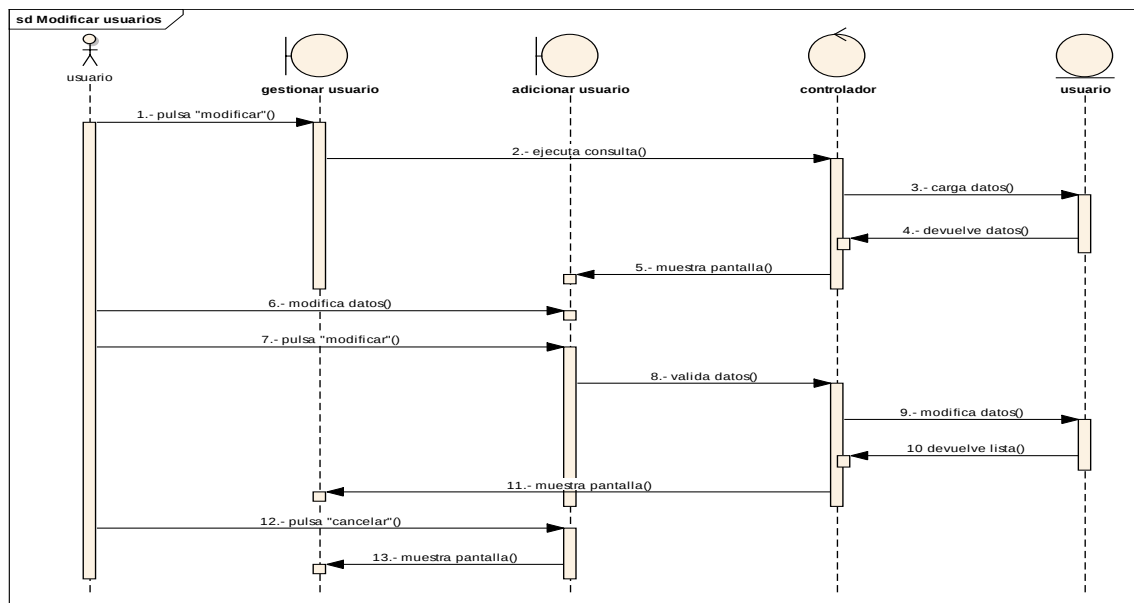


Figura nº 119 Diagrama de secuencia: C.U Modificar Usuarios

Diagrama de secuencia: C.U Eliminar Usuarios

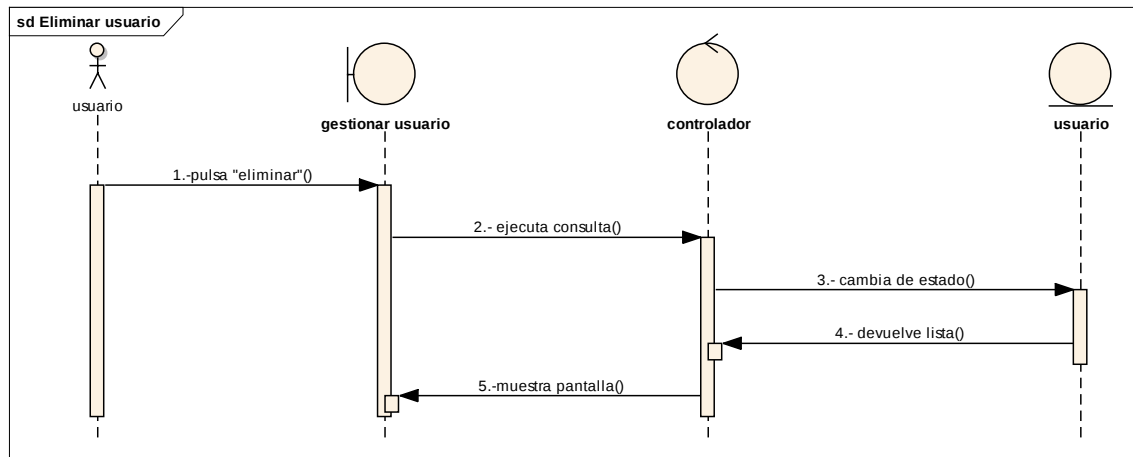


Figura nº 120 Diagrama de secuencia: C.U Eliminar Usuarios

Diagrama de secuencia: C.U Adicionar Clave

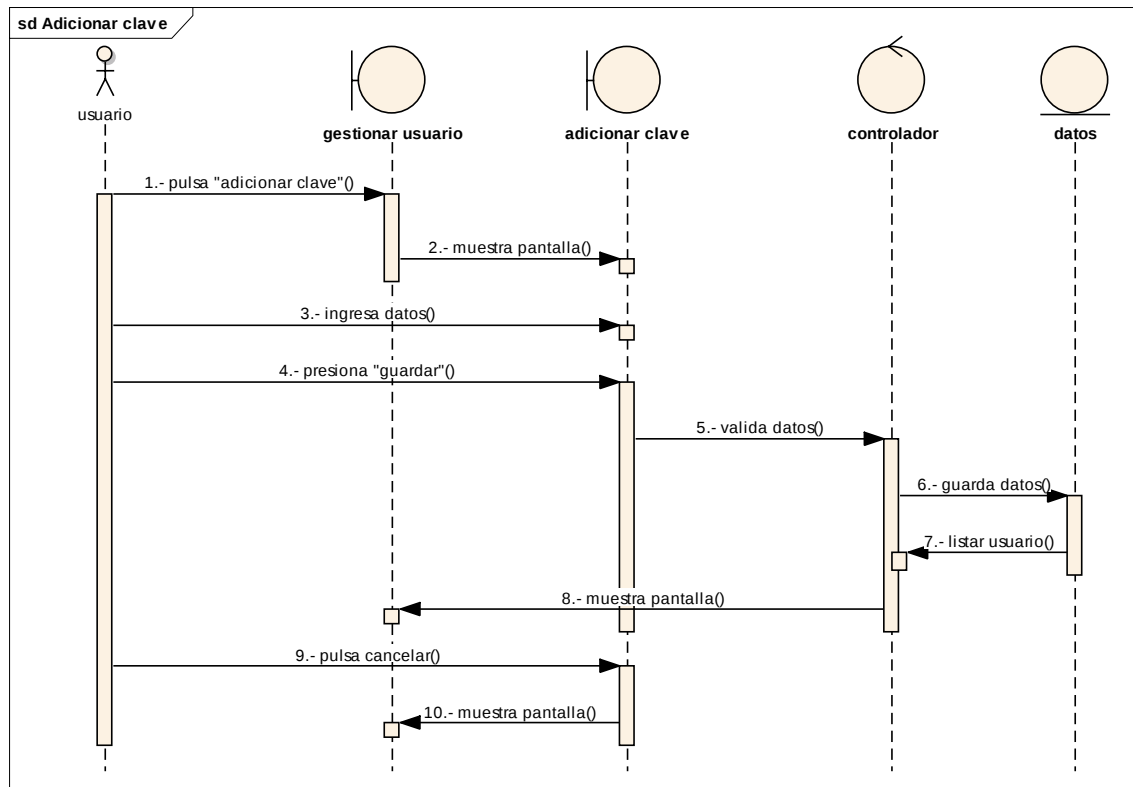


Figura nº 121 Diagrama de secuencia: C.U Adicionar Clave

Diagrama de secuencia: C.U Listar Alumnos

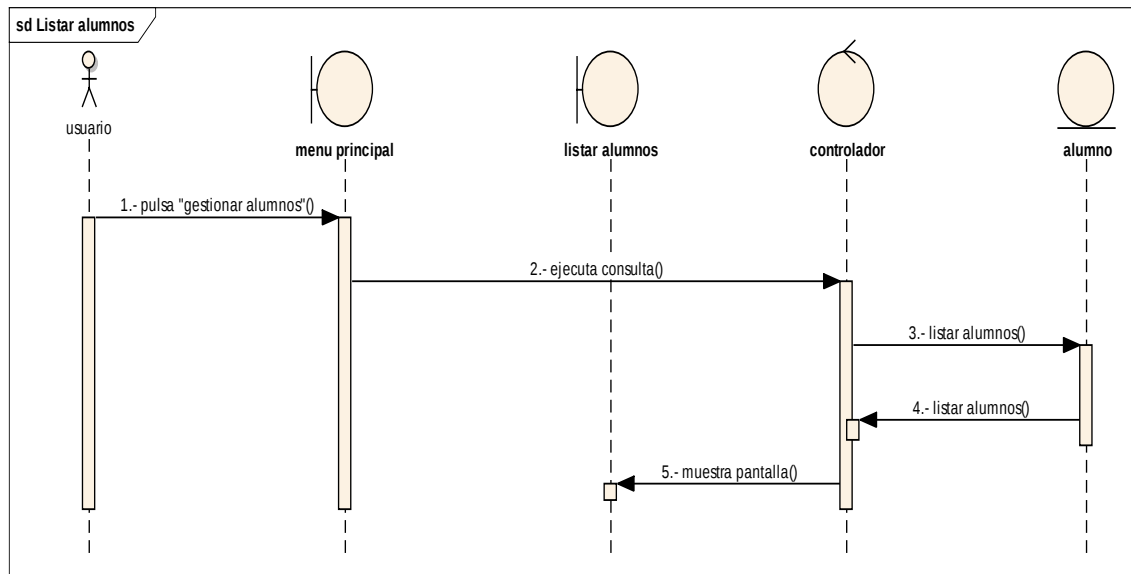


Figura nº 122 Diagrama de secuencia: C.UListar Alumnos

Diagrama de secuencia: C.U Ver Alumnos

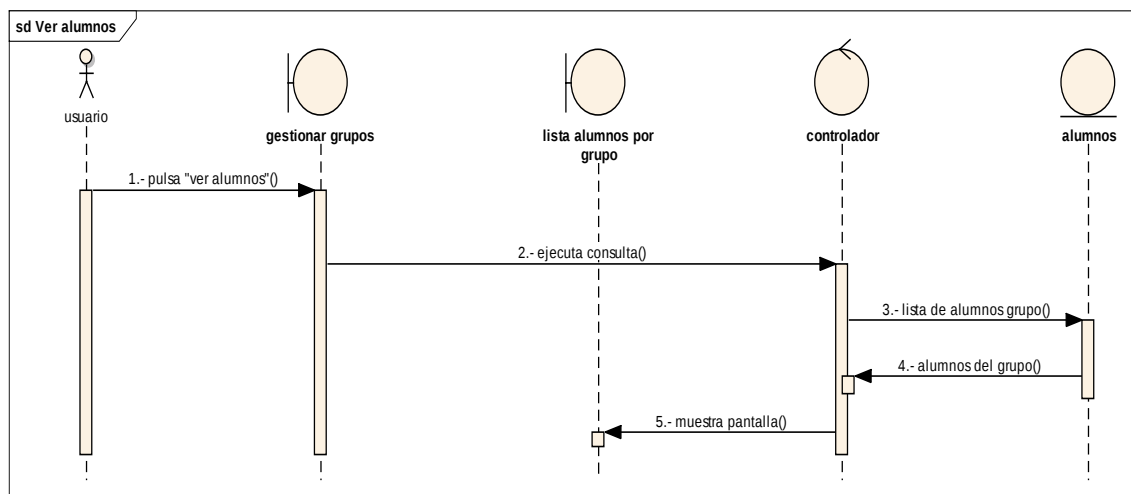


Figura nº 123 Diagrama de secuencia:C.UVer Alumnos

Diagrama de secuencia: C.U Adicionar Alumno

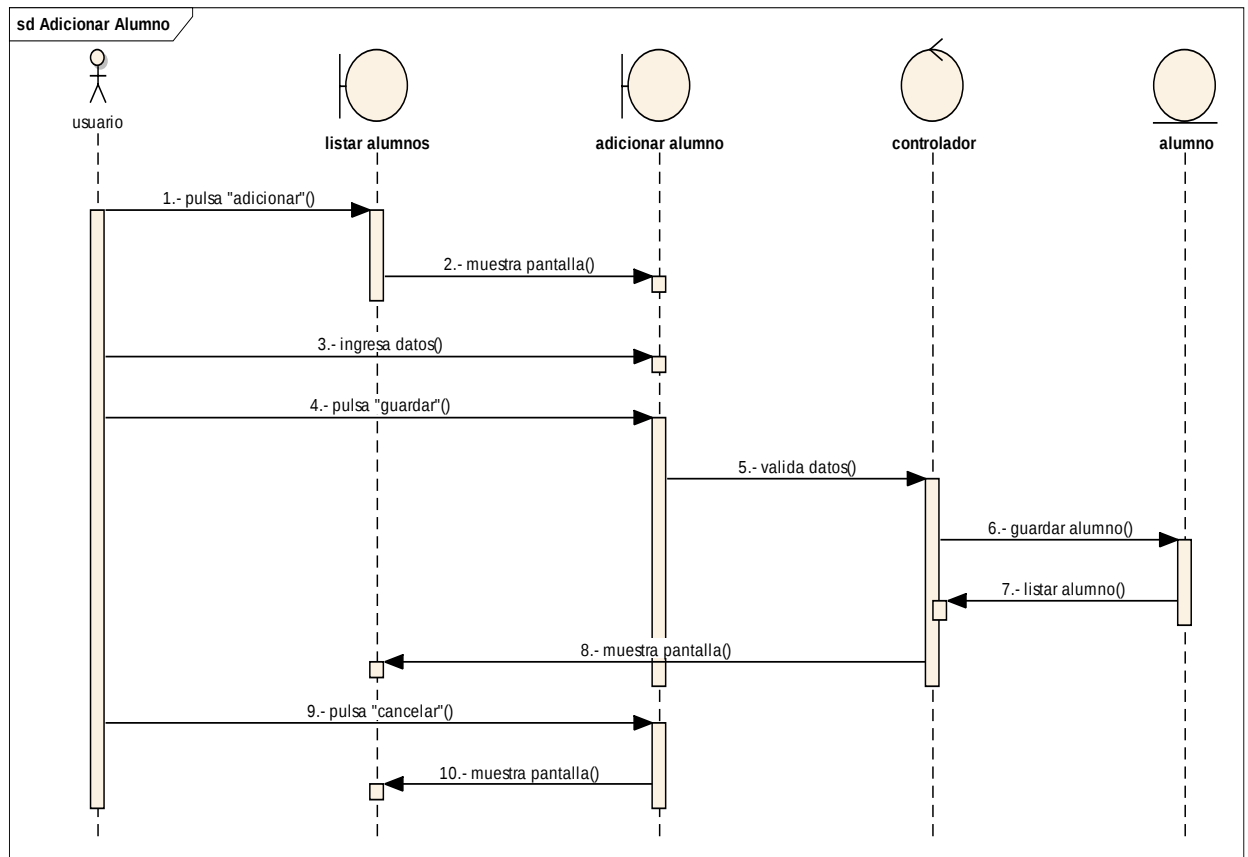


Figura nº 124 Diagrama de secuencia:C.UAdicionar Alumno

Diagrama de secuencia: C.U Modificar Alumno

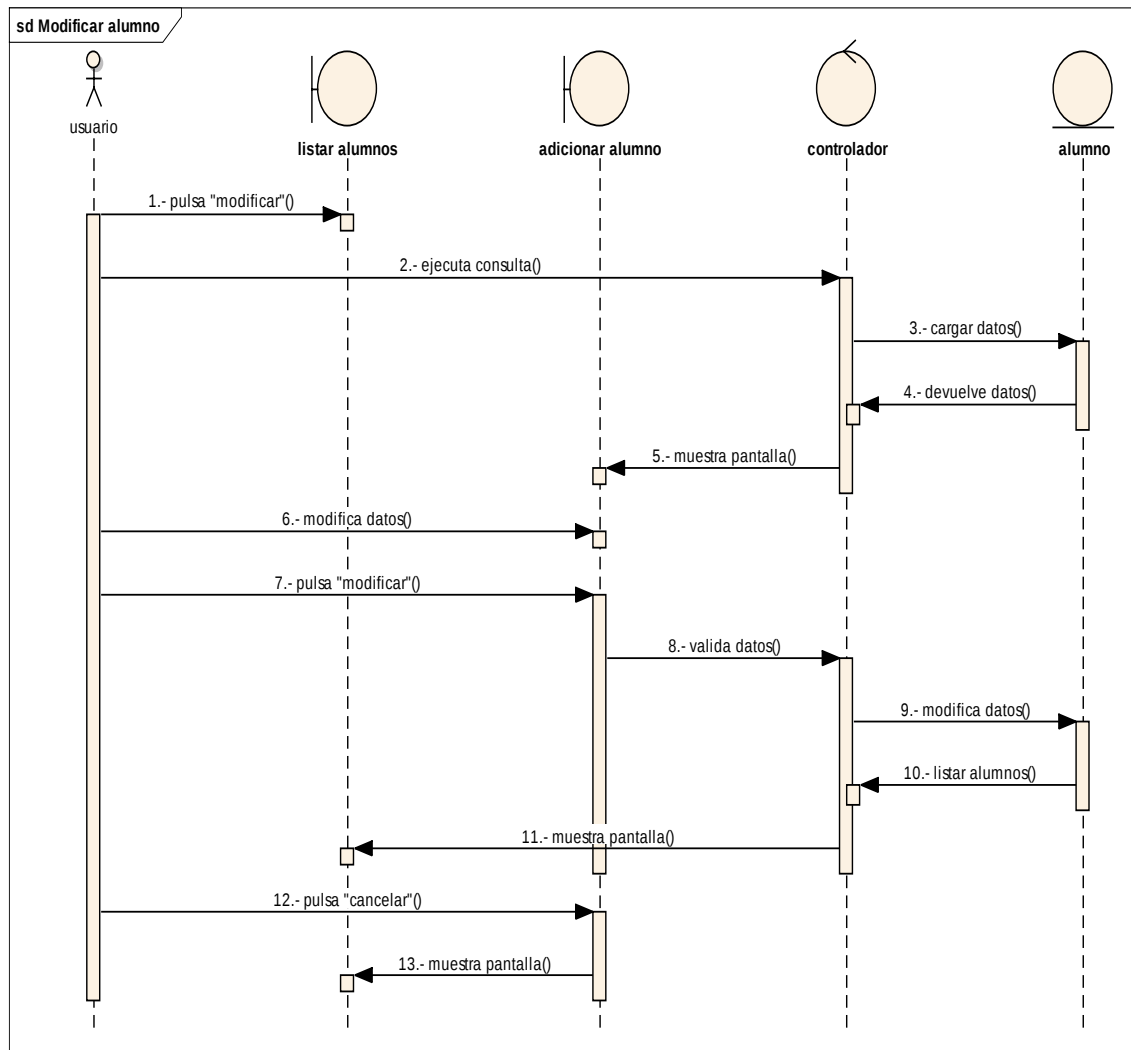


Figura nº 125 Diagrama de secuencia: C.UModificar Alumno

Diagrama de secuencia: C.U Reinscribir Alumno

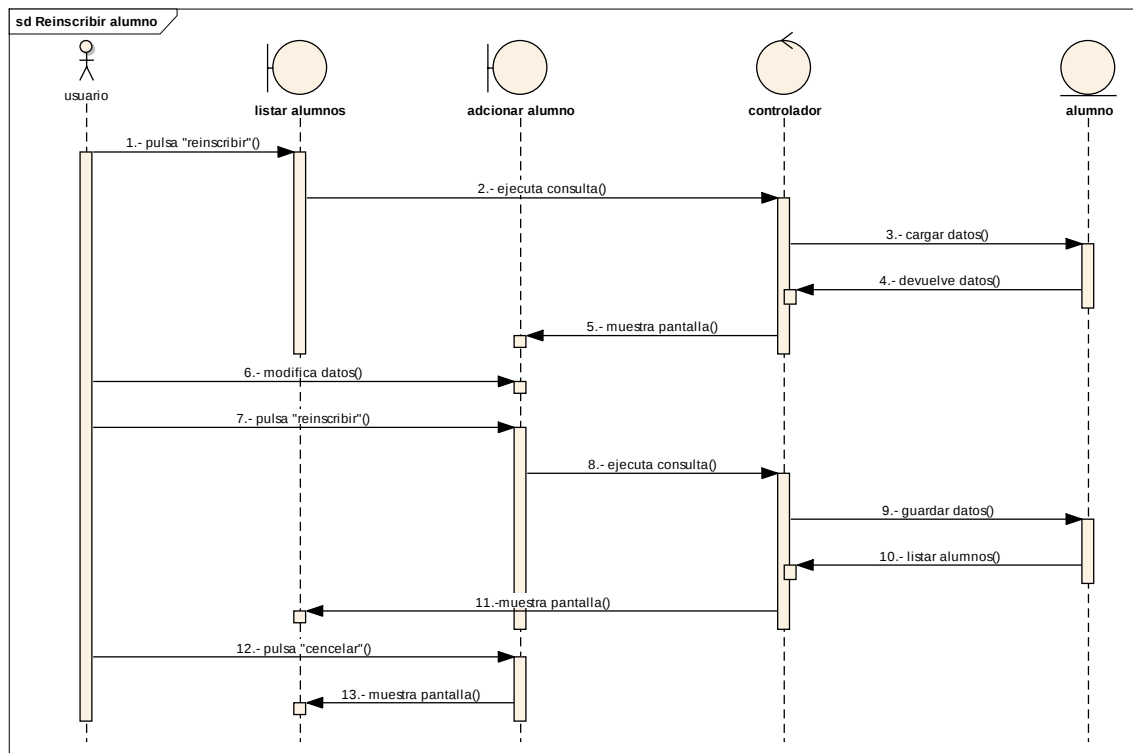


Figura nº 126 Diagrama de secuencia: C.UReinscribir Alumno

Diagrama de secuencia: C.U Gestionar Aulas

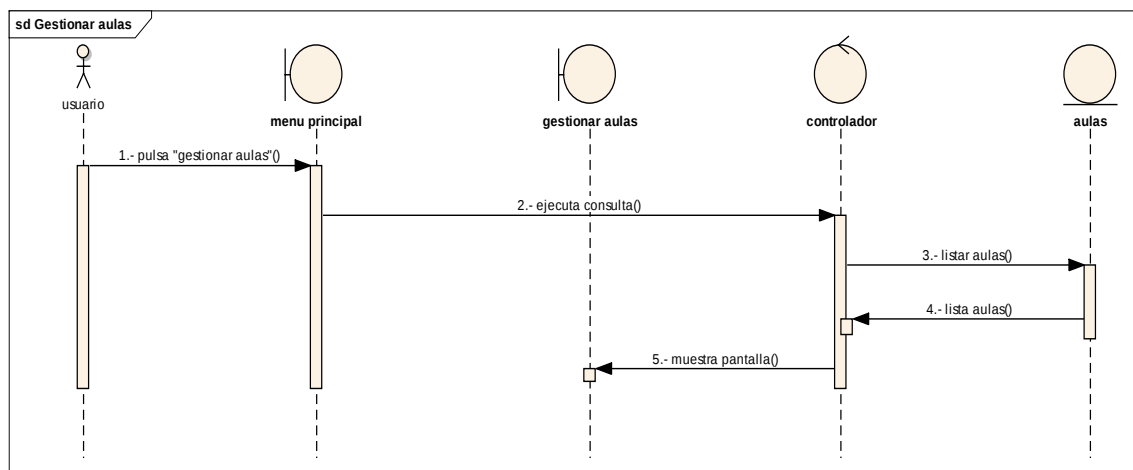


Figura nº 127 Diagrama de secuencia: C.UGestionar Aulas

Diagrama de secuencia: C.U Adicionar Aula

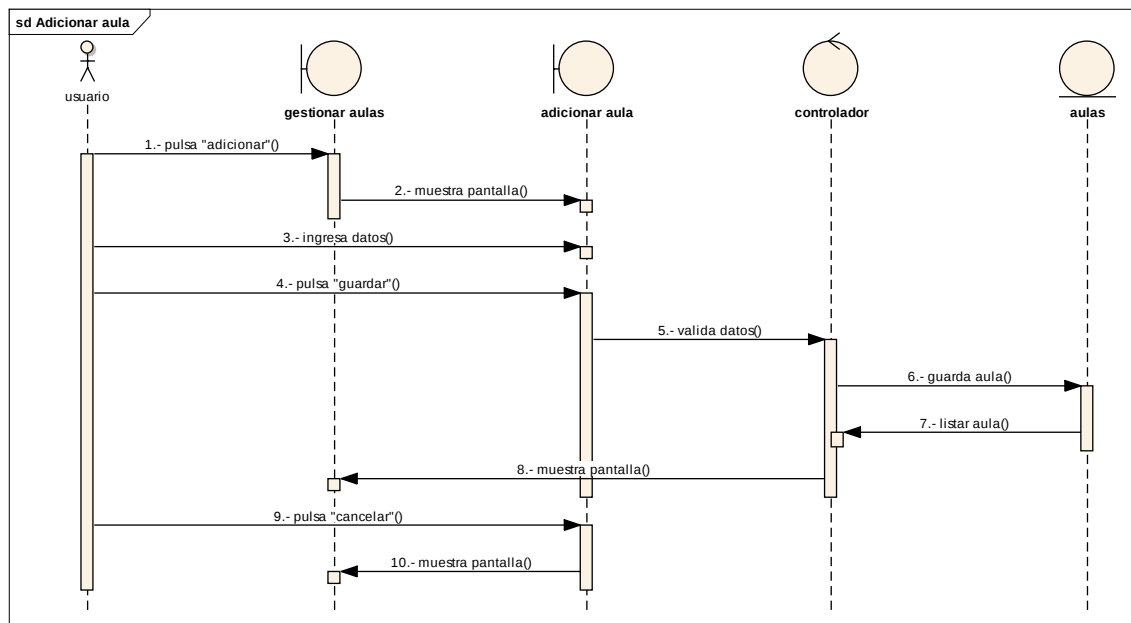


Figura nº 128 Diagrama de secuencia: C.UAdicionar Aula

Diagrama de secuencia: C.U Modificar Aula

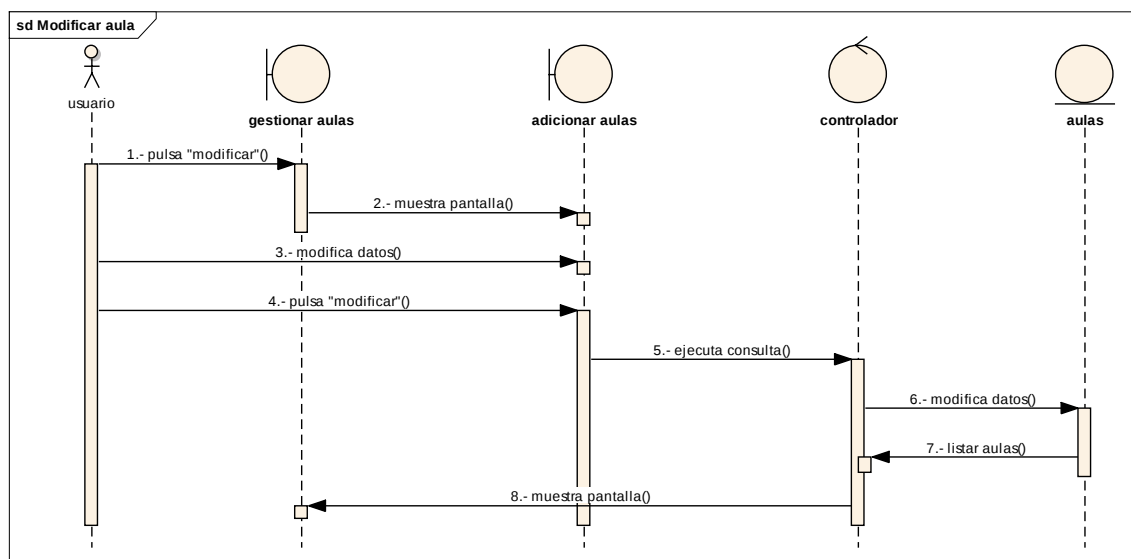


Figura nº 129 Diagrama de secuencia: C.UModificar Aula

Diagrama de secuencia: C.U Eliminar Aula

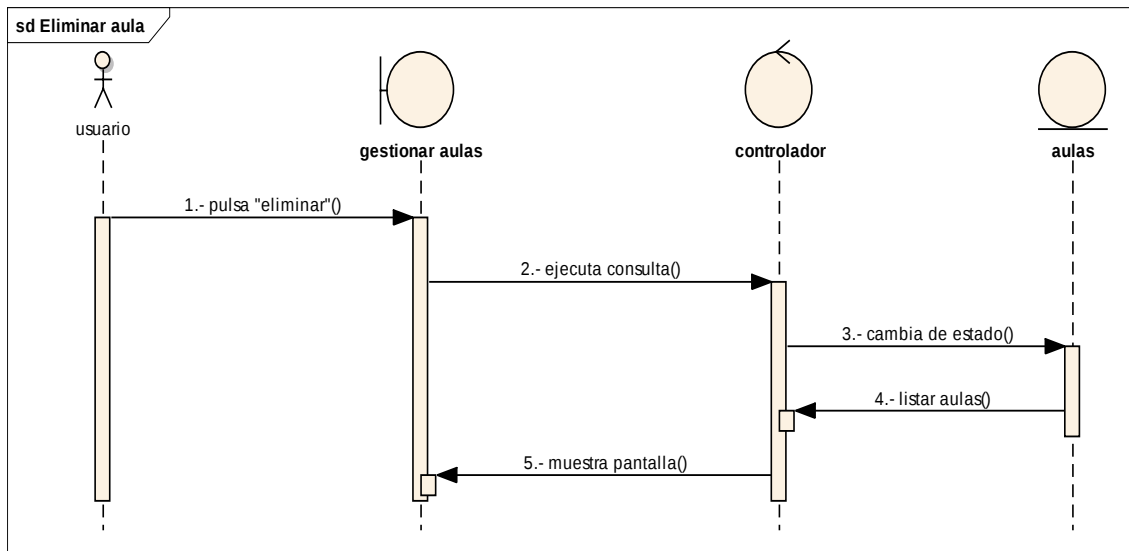


Figura nº 130 Diagrama de secuencia:C.UEliminar Aula

Diagrama de secuencia: C.U Gestionar Departamento

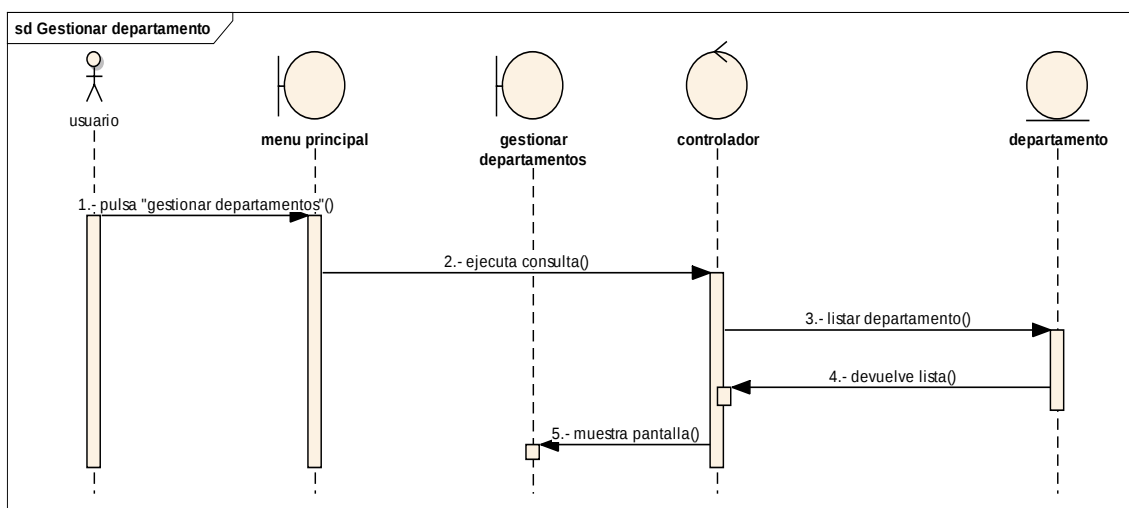


Figura nº 131 Diagrama de secuencia:C.UGestionar Departamento

Diagrama de secuencia: C.U Adicionar Departamento

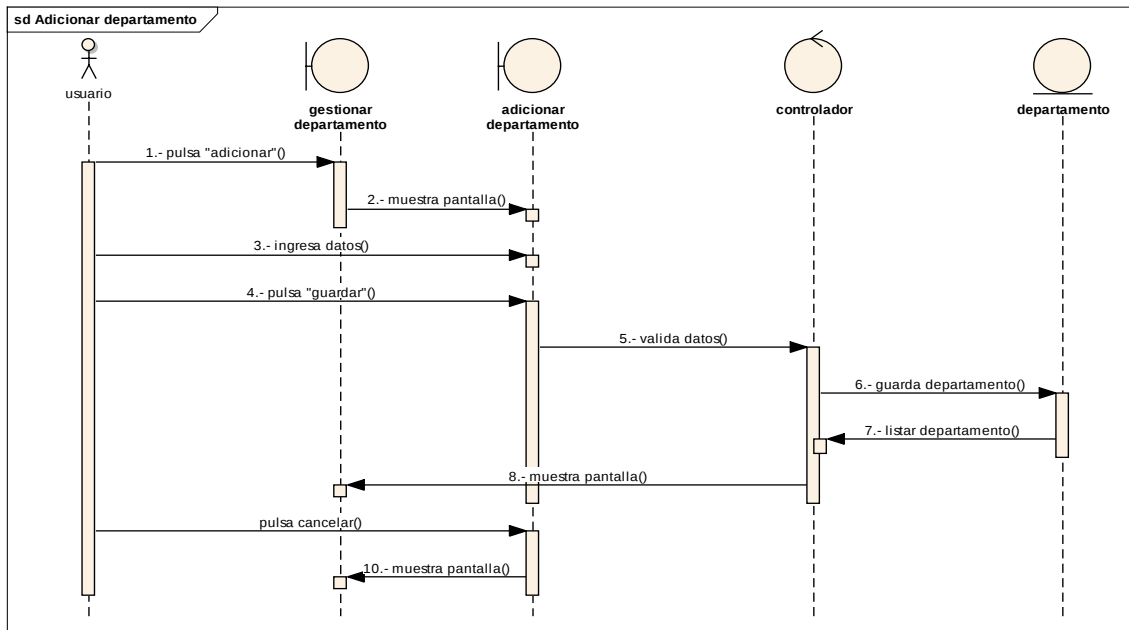


Figura nº 132 Diagrama de secuencia: C.UAdicionar Departamento

Diagrama de secuencia: C.U Modificar Departamento

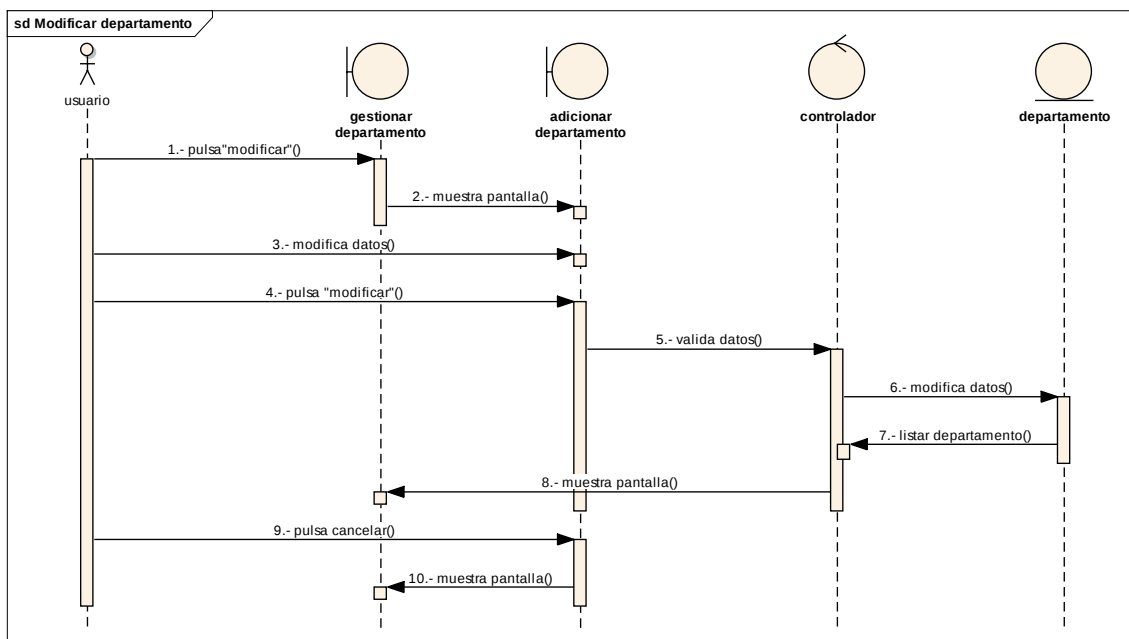


Figura nº 133 Diagrama de secuencia: C.UModificar Departamento

Diagrama de secuencia: C.U Eliminar Departamento

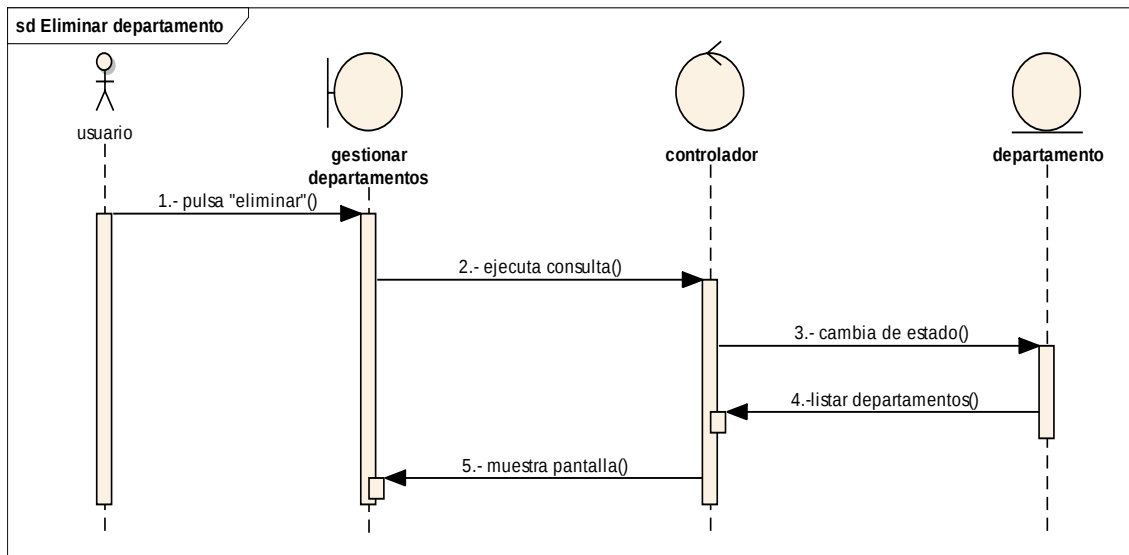


Figura nº 134 Diagrama de secuencia:C.U Eliminar Departamento

Diagrama de secuencia: C.U Gestionar Nivel

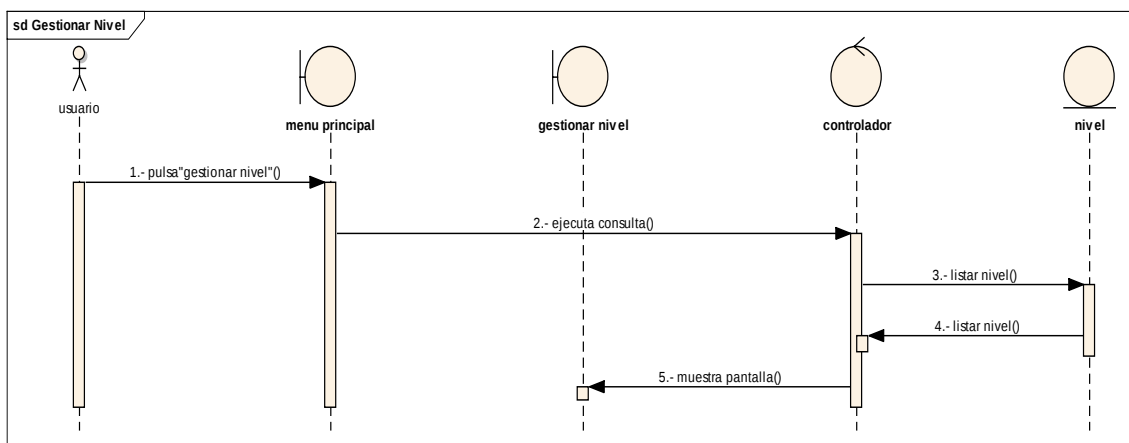


Figura nº 135 Diagrama de secuencia: C.UGestionar Nivel

Diagrama de secuencia: C.U Adicionar Nivel

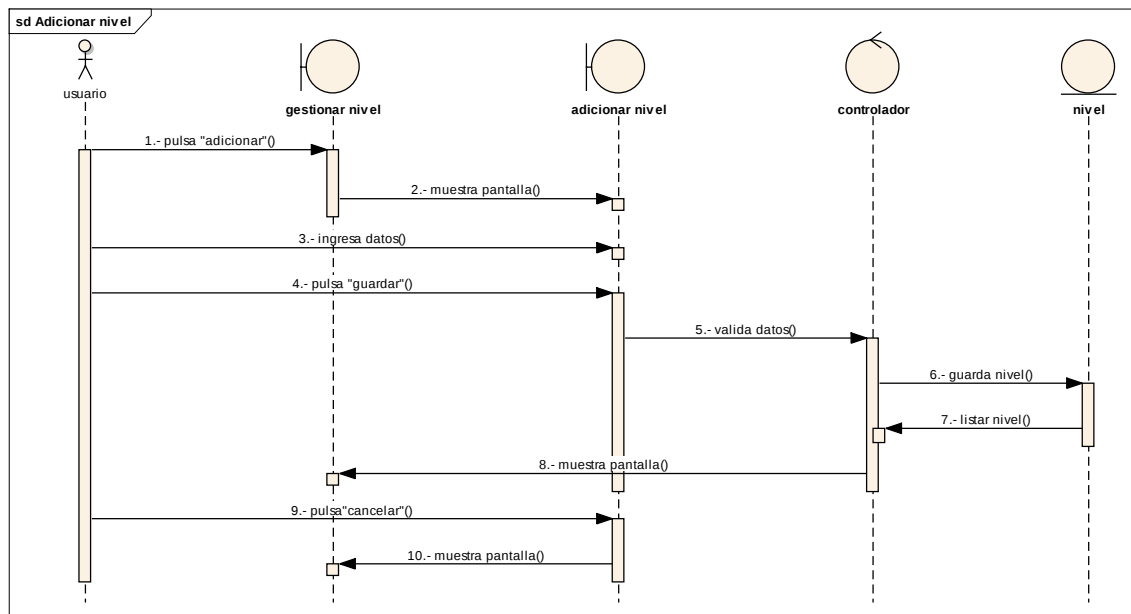


Figura nº 136 Diagrama de secuencia: C.UAdicionar Nivel

Diagrama de secuencia: C.U Modificar Nivel

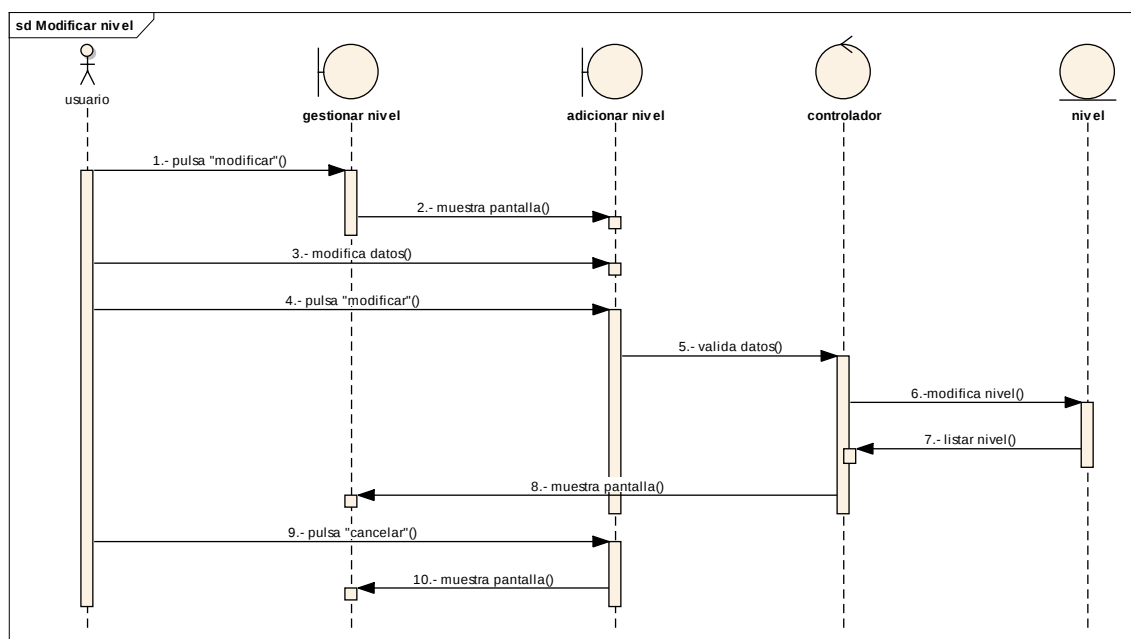


Figura nº 137 Diagrama de secuencia:C.UModificar Nivel

Diagrama de secuencia: C.U Gestionar Grados

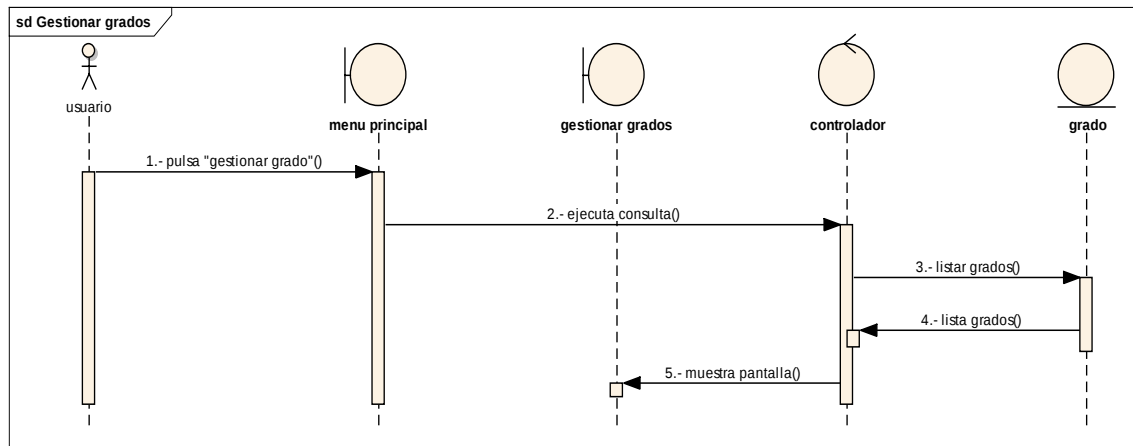


Figura nº 138 Diagrama de secuencia: C.U Gestionar Grados

Diagrama de secuencia: C.U Adicionar Grado

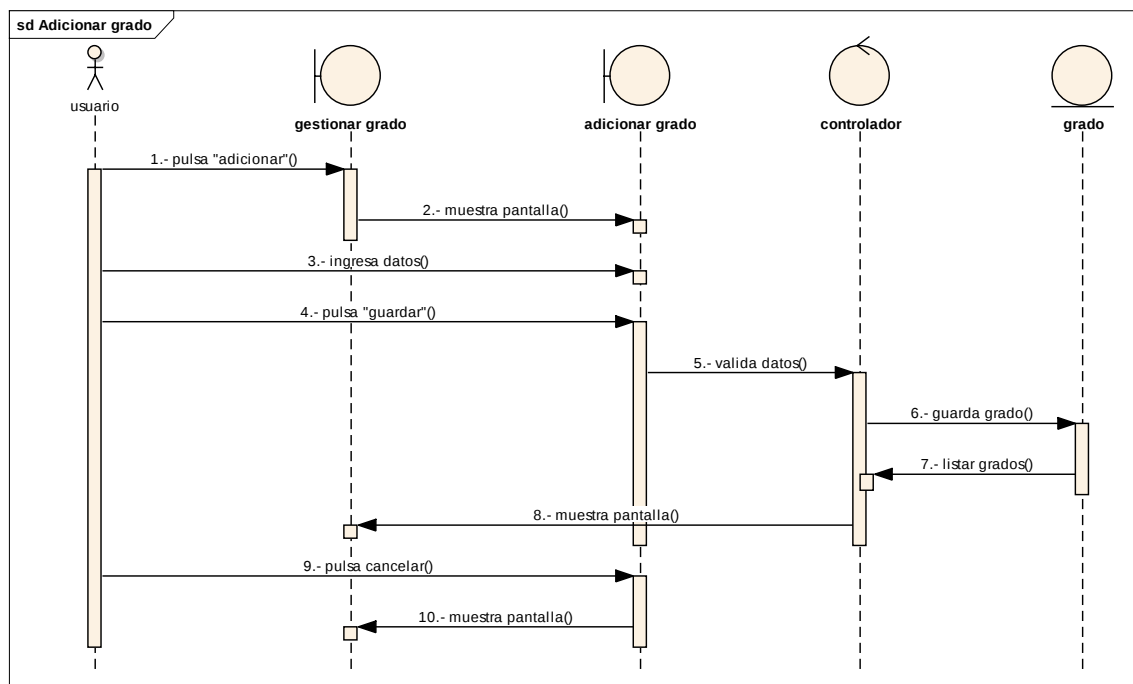


Figura nº 139 Diagrama de secuencia: C.U Adicionar Grado

Diagrama de secuencia: C.U Modificar Grado

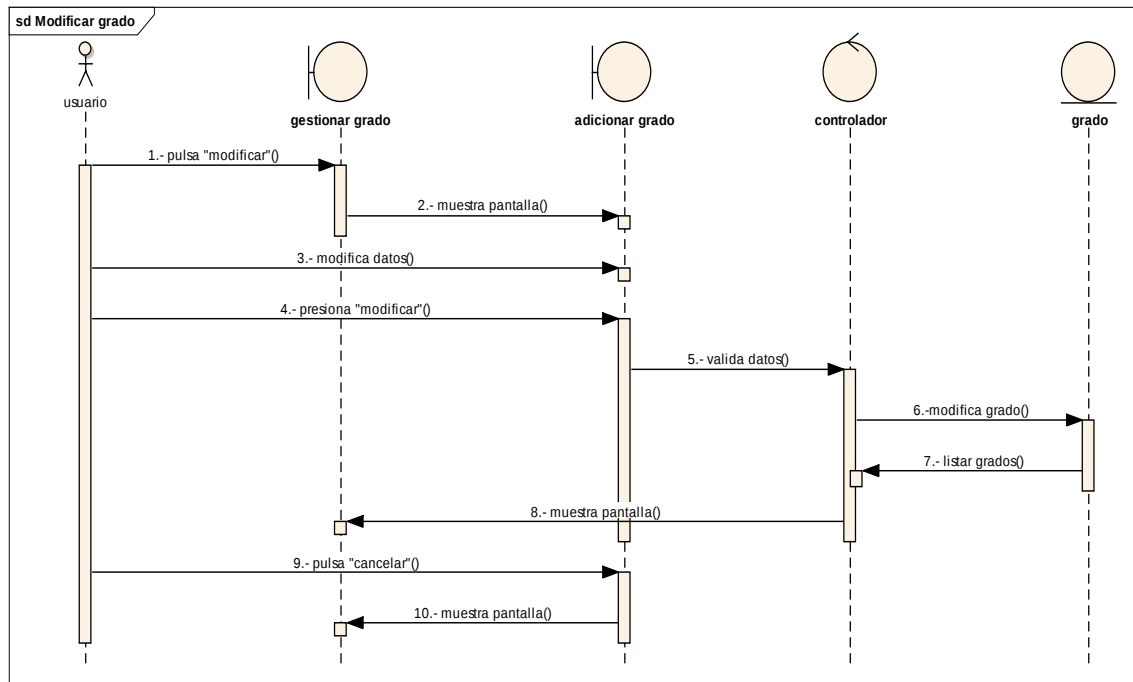


Figura nº 140 Diagrama de secuencia:C.U Modificar Grado

Diagrama de secuencia: C.U Eliminar Grado

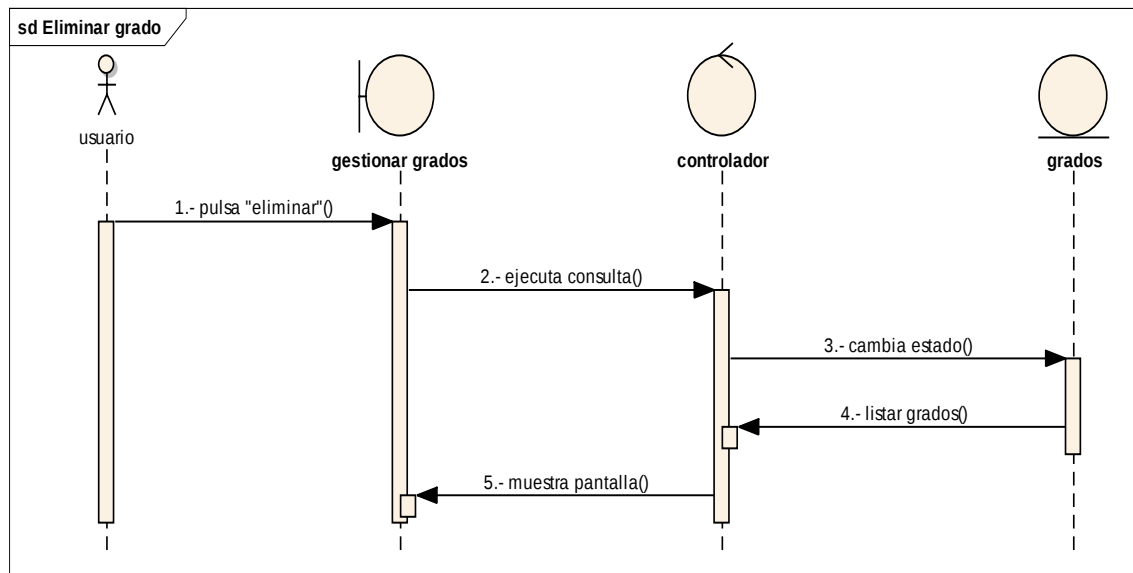


Figura nº 141 Diagrama de secuencia: C.U Eliminar Grado

Diagrama de secuencia: C.U Gestionar Grupos

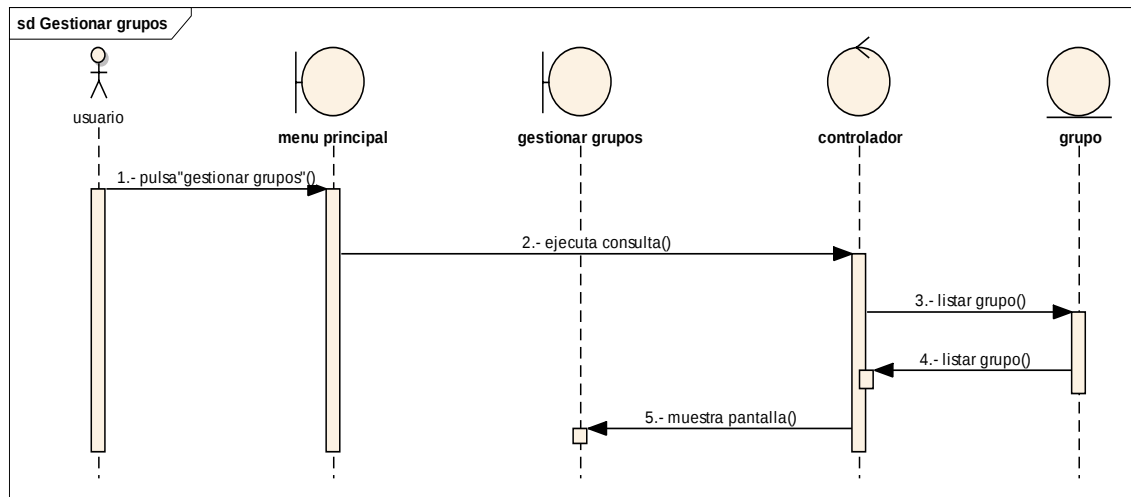


Figura nº 142 Diagrama de secuencia: C.UGestionar Grupos

Diagrama de secuencia: C.U Adicionar Grupo

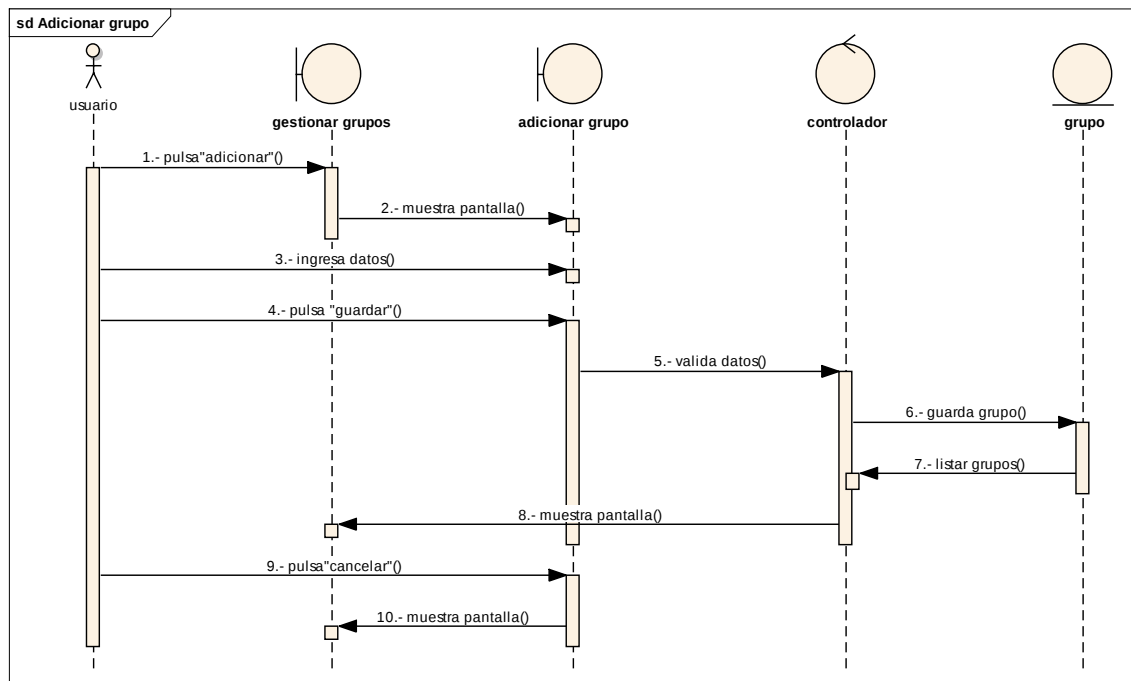


Figura nº 143 Diagrama de secuencia: C.U Adicionar Grupo

Diagrama de secuencia: C.U Modificar Grupo

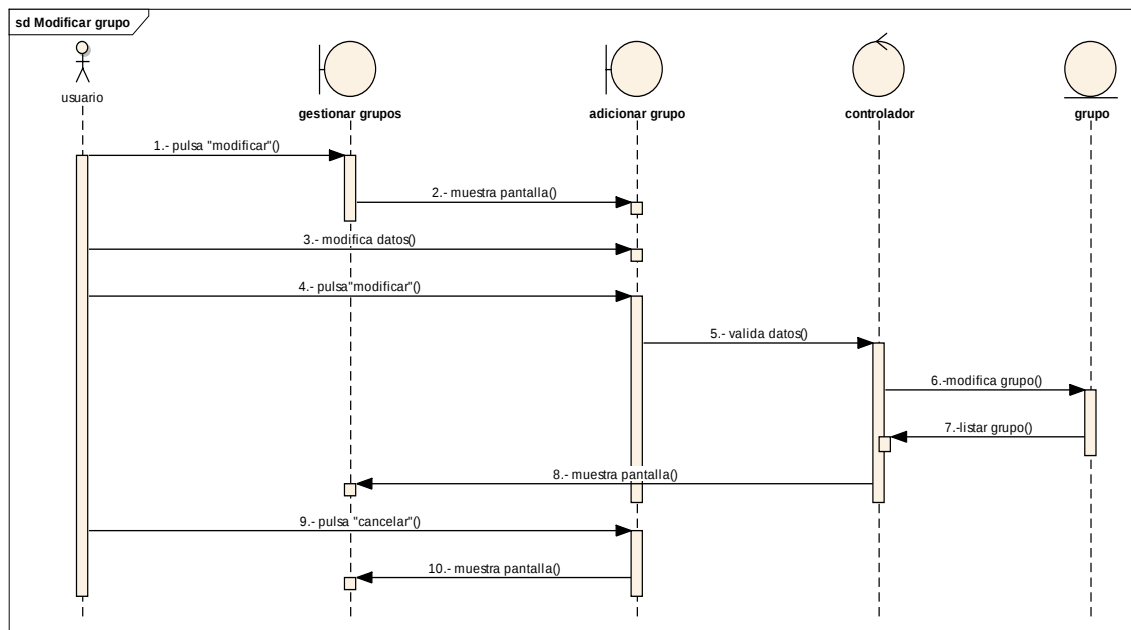


Figura nº 144 Diagrama de secuencia: C.U Modificar Grupo

Diagrama de secuencia: C.U Eliminar Grupo

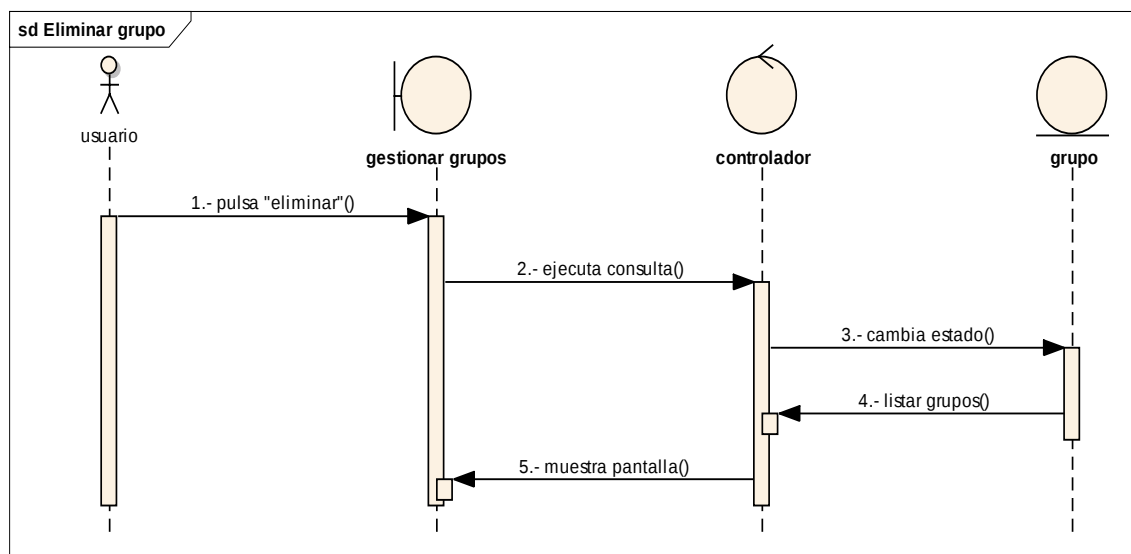


Figura nº 145 Diagrama de secuencia: C.U Eliminar Grupo

Diagrama de secuencia: C.U Asignar Tutor

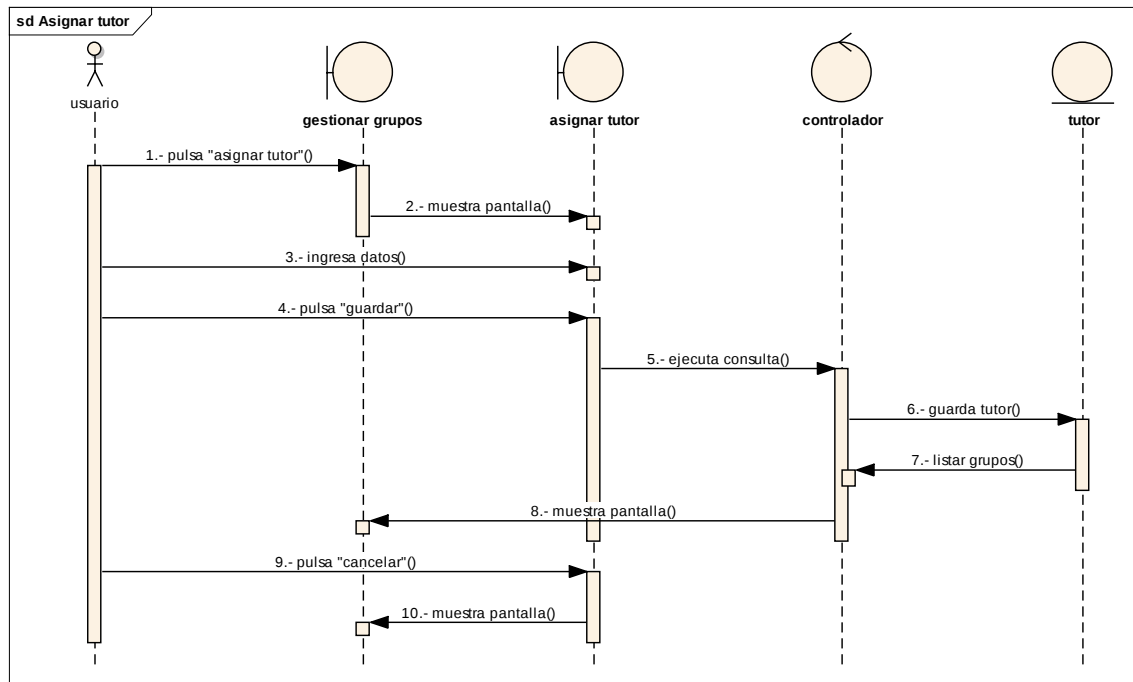


Figura nº 146 Diagrama de secuencia: C.UAsignar Tutor

Diagrama de secuencia: C.U Modificar Tutor

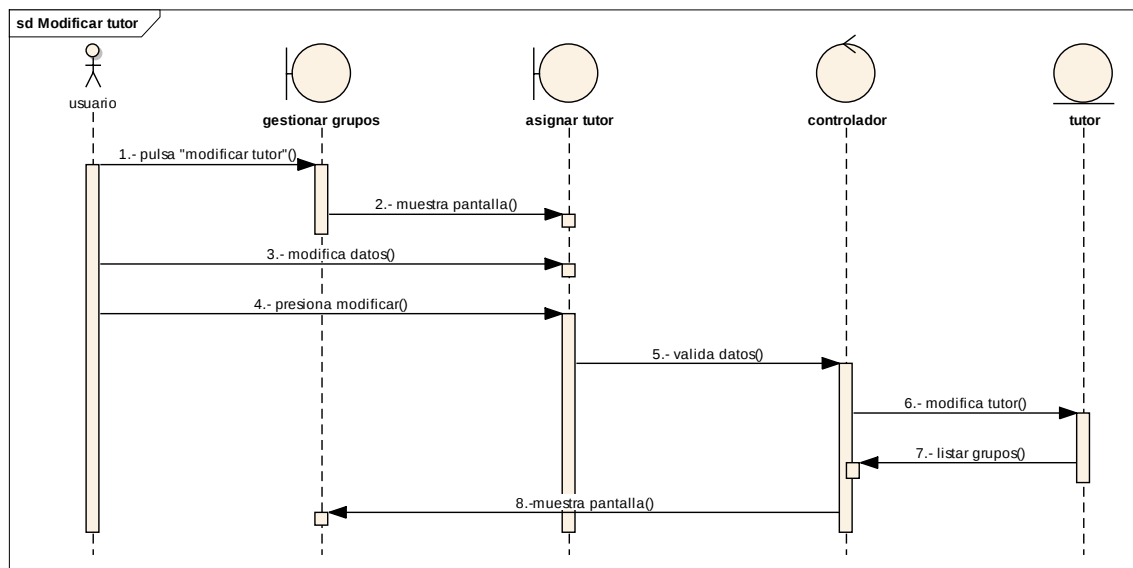


Figura nº 147 Diagrama de secuencia: C.UModificar Tutor

Diagrama de secuencia: C.U Listar Materia por Alumno

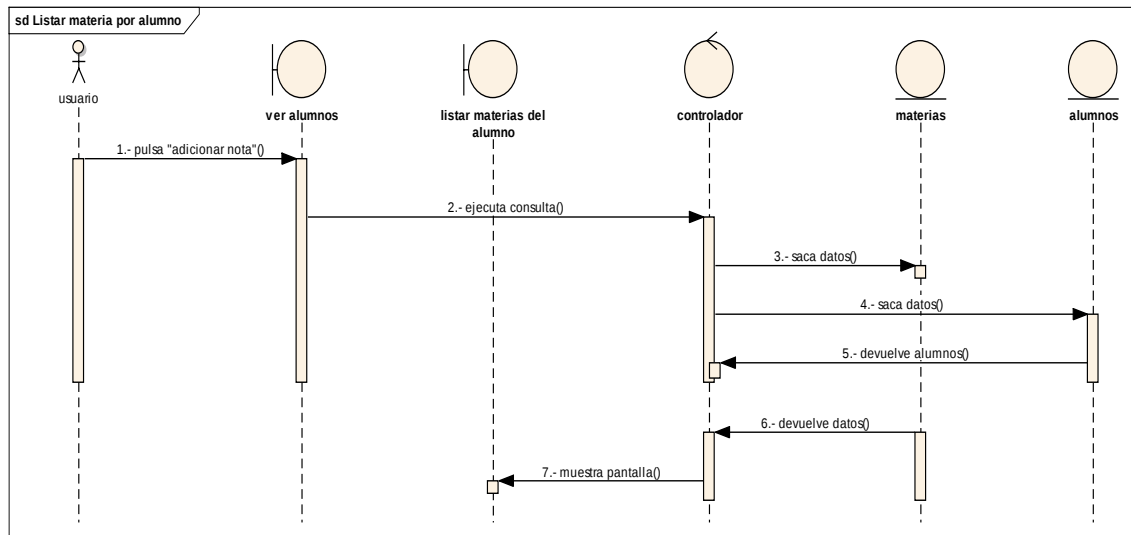


Figura nº 148 Diagrama de secuencia: C.U Listar Materia por Alumno

Diagrama de secuencia: C.U Mostrar Nota

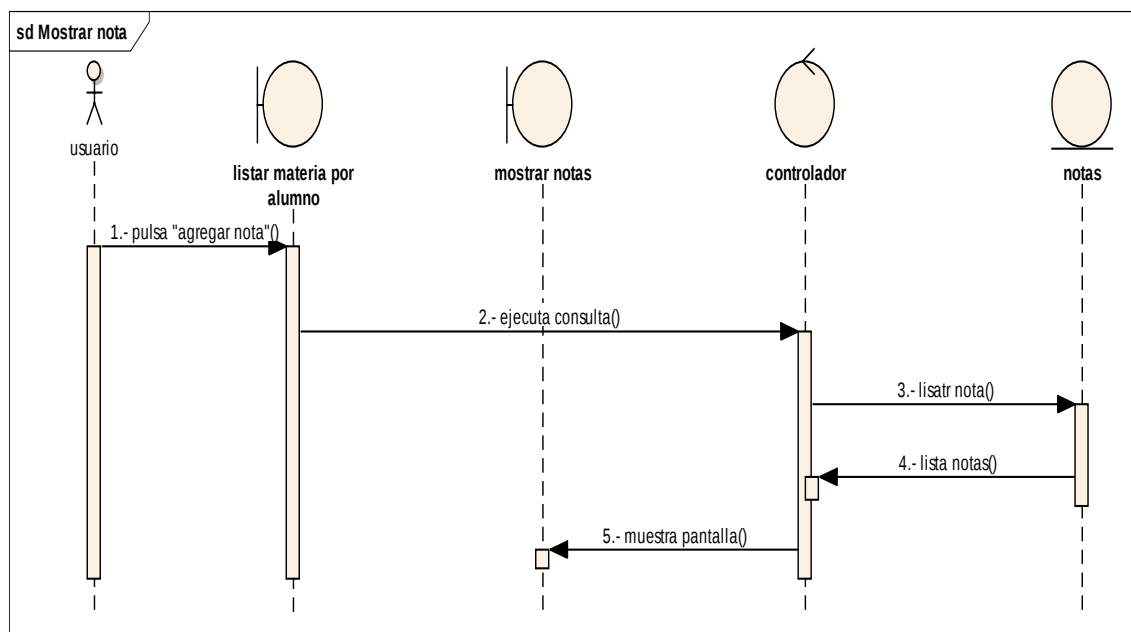


Figura nº 149 Diagrama de secuencia: C.U Mostrar Nota

Diagrama de secuencia: C.U Ver Libreta

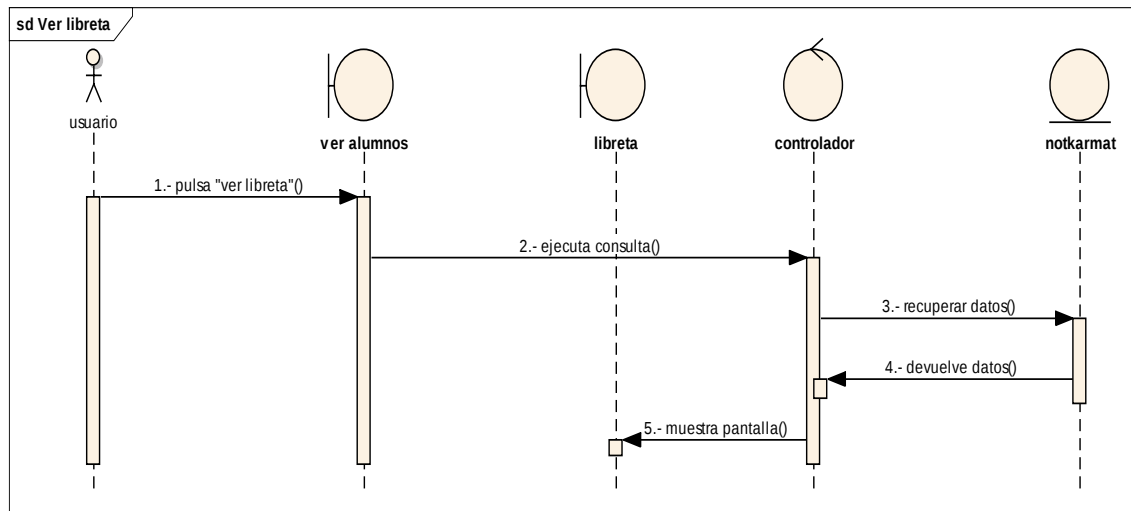


Figura nº 150 Diagrama de secuencia: C.UVer Libreta

Diagrama de secuencia: C.U Gestionar Materia Estudiada

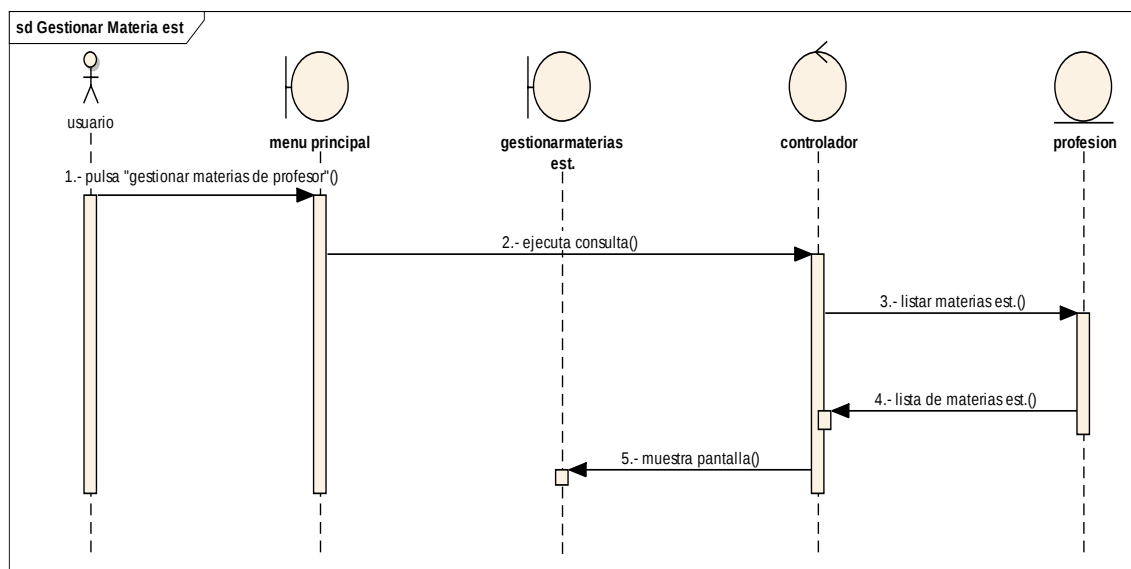


Figura nº 151 Diagrama de secuencia: C.UGestionar Materia Estudiada

Diagrama de secuencia: C.U Adicionar Materia Estudiada

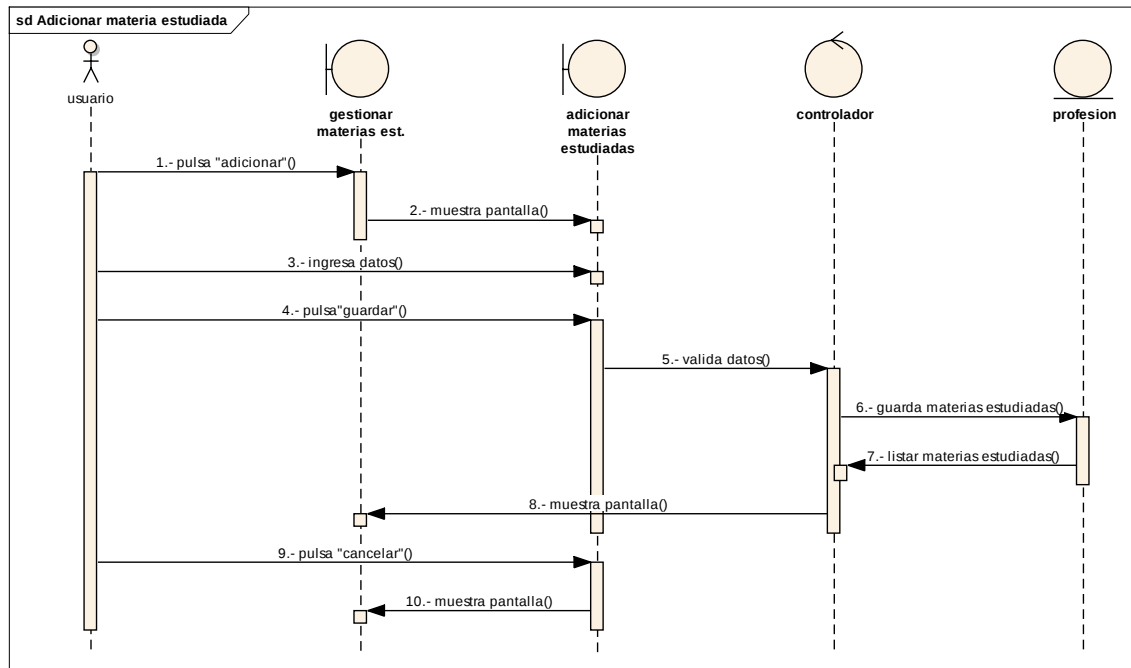


Figura nº 152 Diagrama de secuencia: C.UAdicionar Materia Estudiada

Diagrama de secuencia: C.U Modificar Materia Estudiada

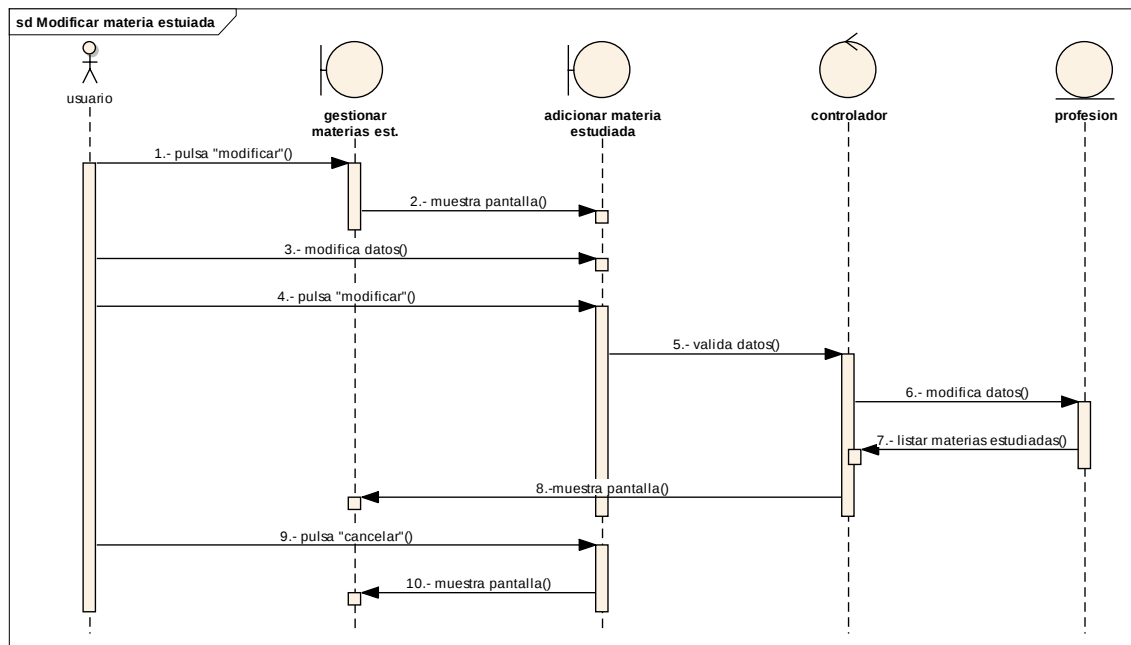


Figura nº 153 Diagrama de secuencia: C.UModificar Materia Estudiada

Diagrama de secuencia: C.U Gestionar Profesores

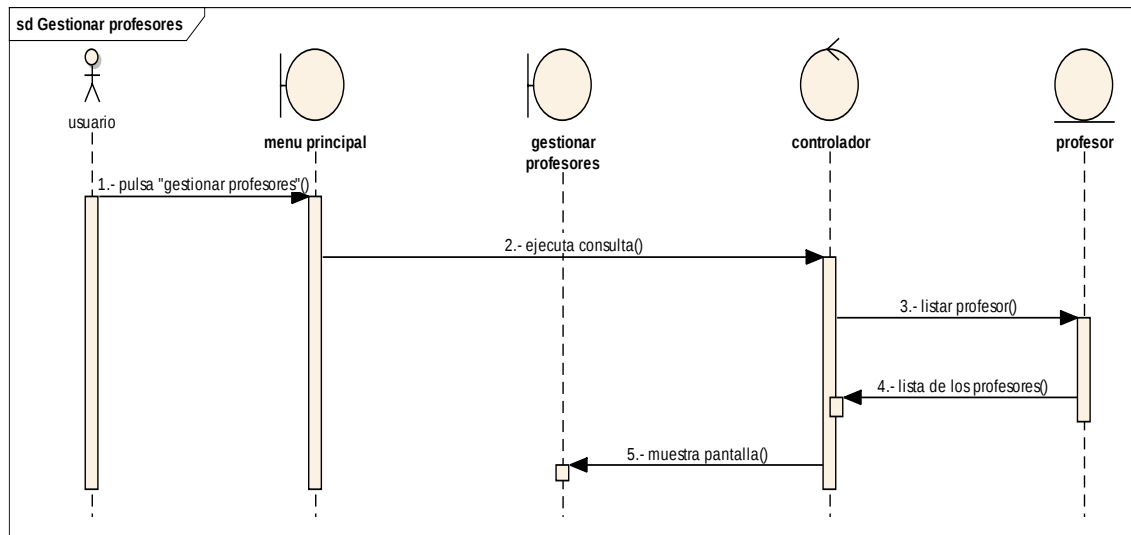


Figura nº 154 Diagrama de secuencia: C.UGestionar Profesores

Diagrama de secuencia: C.U Adicionar Profesor

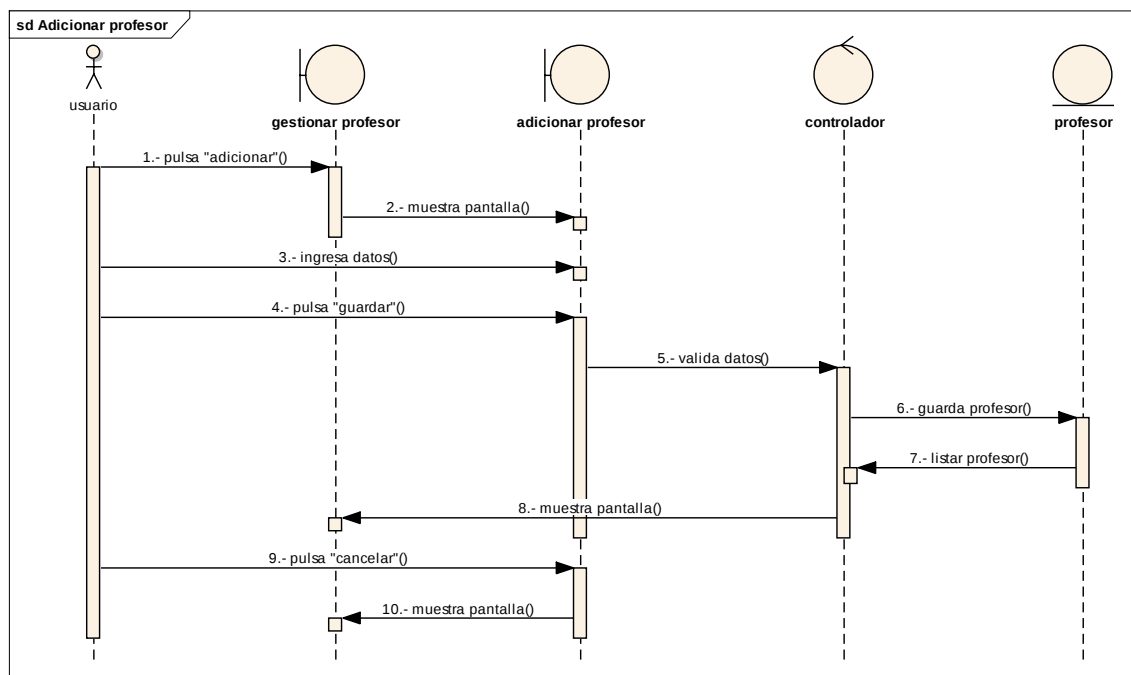


Figura nº 155 Diagrama de secuencia:C.UAdicionar Profesor

Diagrama de secuencia: C.U Modificar Profesores

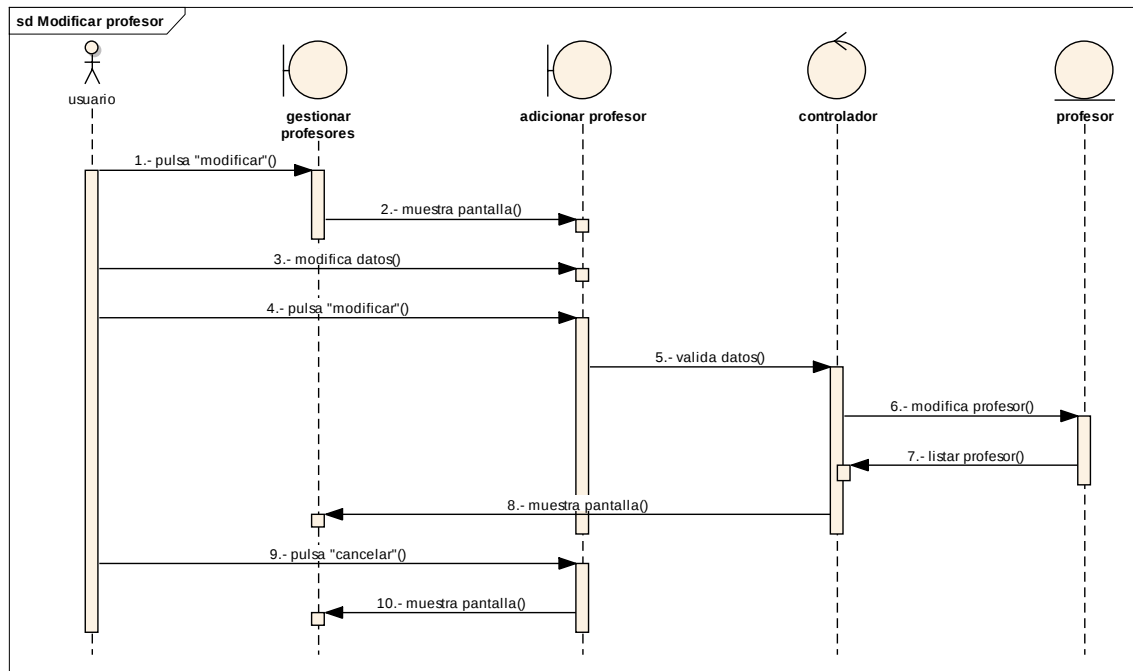


Figura nº 156 Diagrama de secuencia: C.UModificar Profesores

Diagrama de secuencia: C.U Gestionar Materias

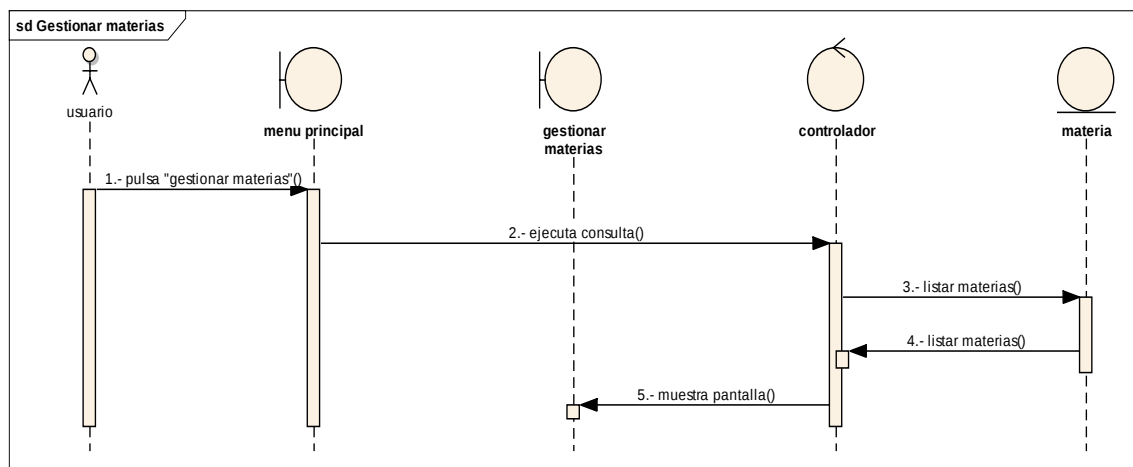


Figura nº 157 Diagrama de secuencia: C.UGestionar Materias

Diagrama de secuencia: C.U Adicionar Materia

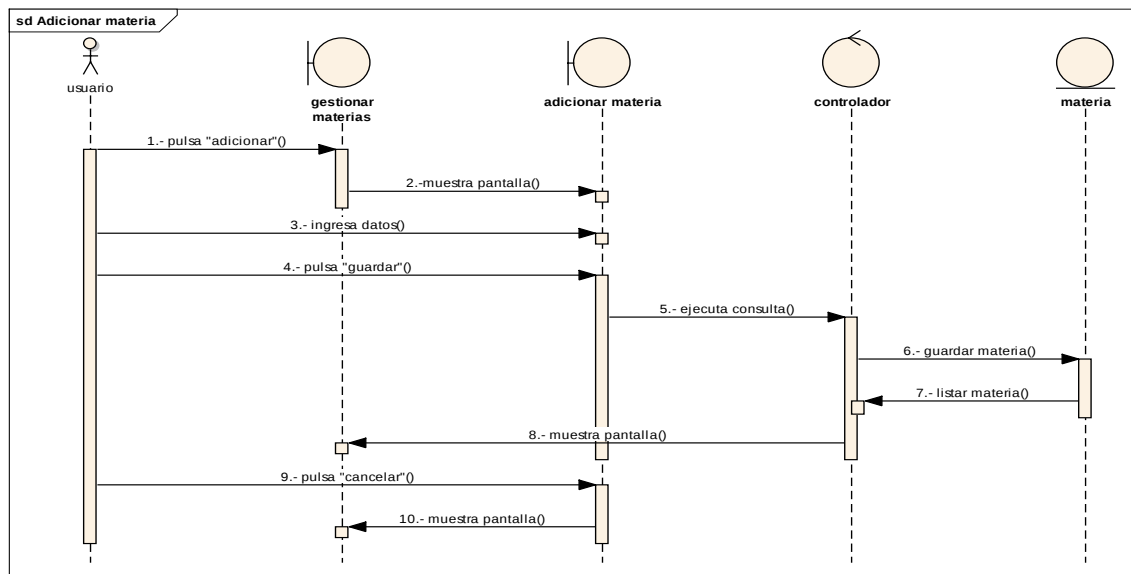


Figura nº 158 Diagrama de secuencia: C.U Adicionar Materia

Diagrama de secuencia: C.U Modificar Materia

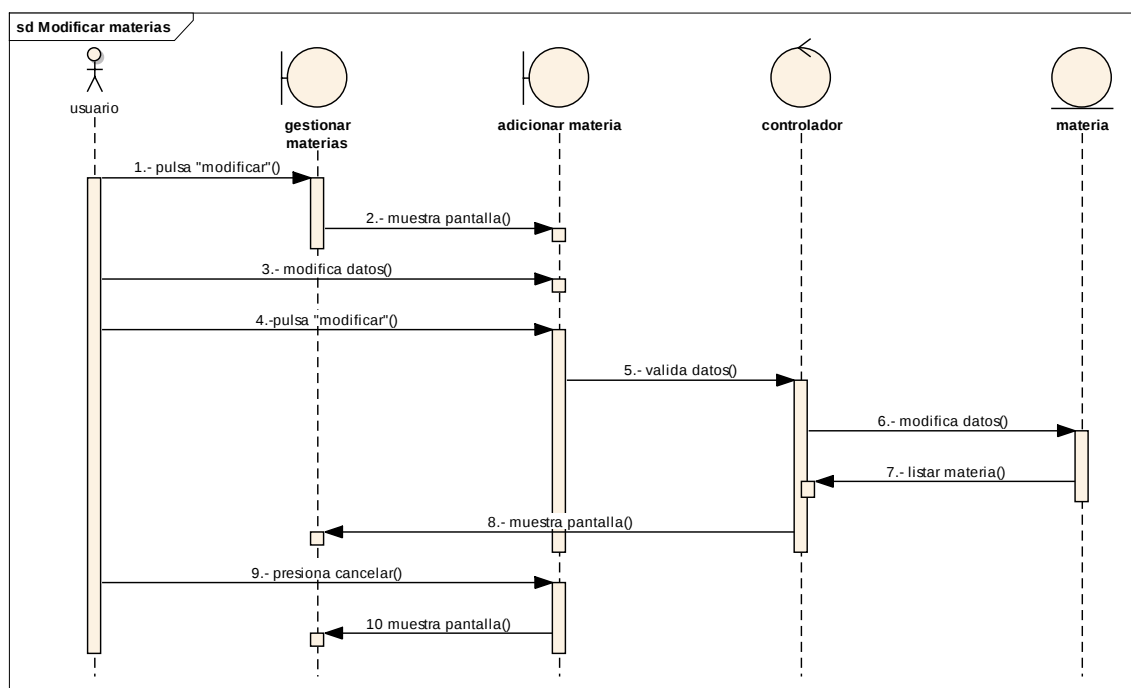


Figura nº 159 Diagrama de secuencia: C.U Modificar Materia

Diagrama de secuencia: C.U Gestionar Materia a Grado

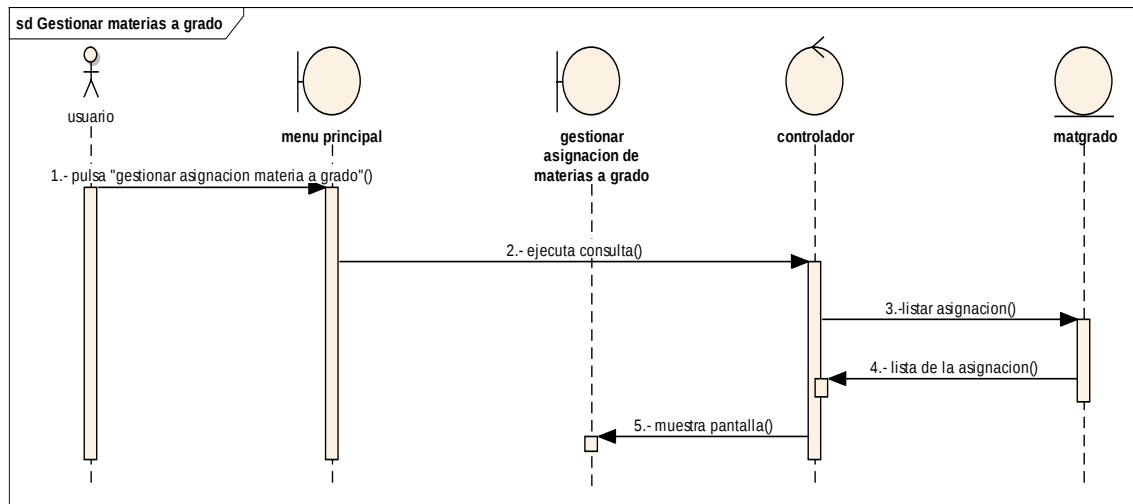


Figura nº 160 Diagrama de secuencia: C.UGestionar Materia a Grado

Diagrama de secuencia: C.U Asignar Materia grado

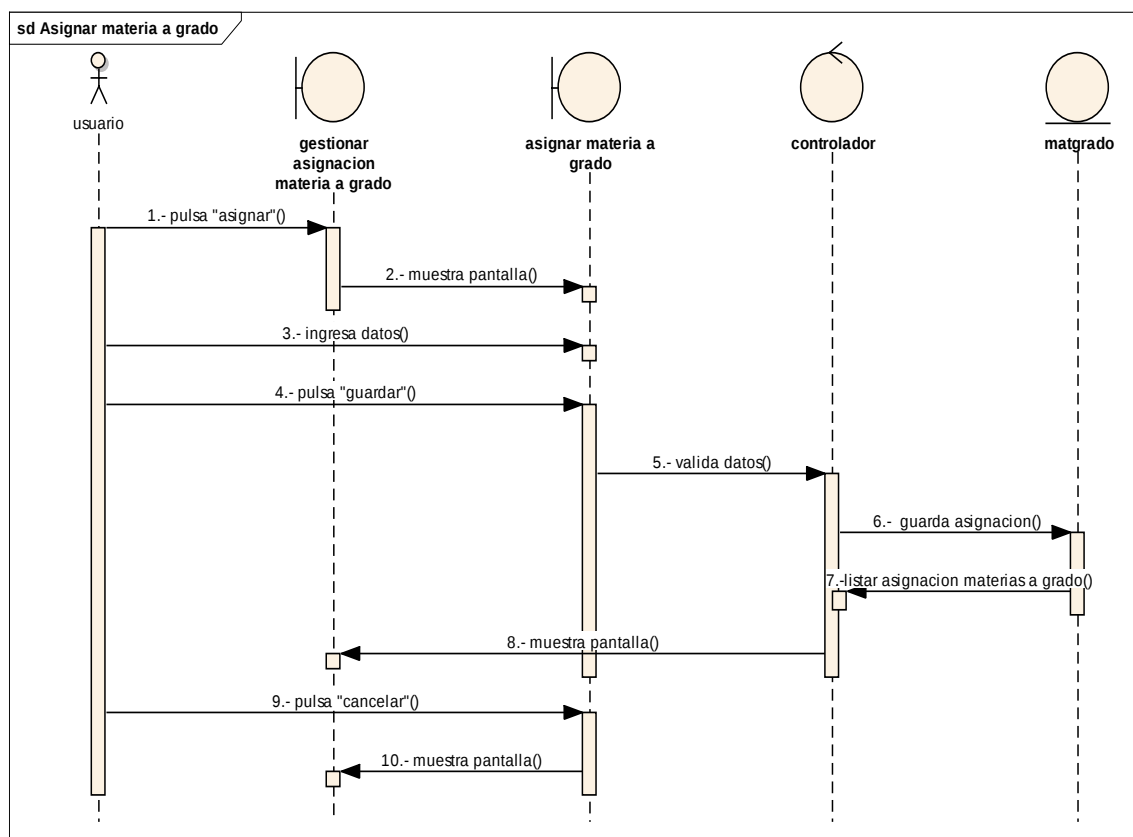


Figura nº 161 Diagrama de secuencia: C.UAsignar Materia grado

Diagrama de secuencia: C.U Ver Materias Asignadas al Grado

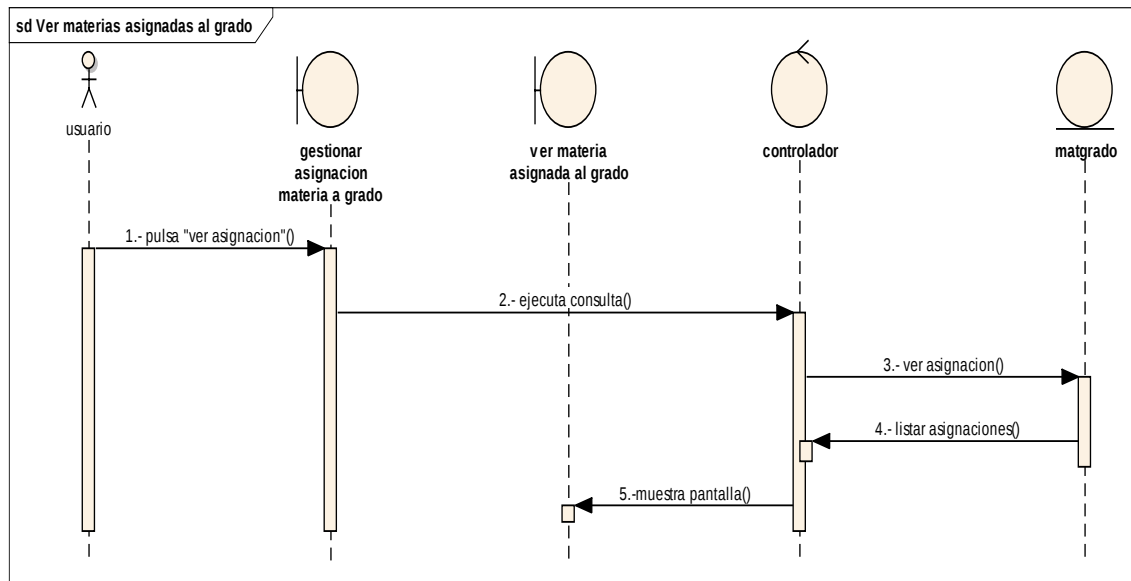


Figura n° 162 Diagrama de secuencia: C.UVer Materias Asignadas al Grado

Diagrama de secuencia: C.U Gestionar Asignación Materias a Profesor

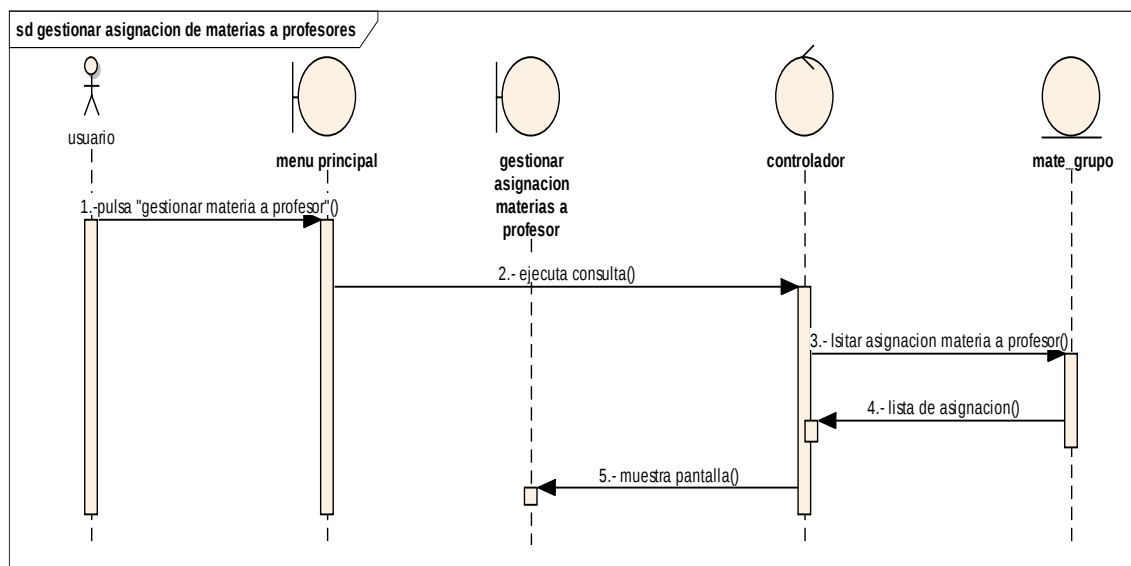


Figura n° 163 Diagrama de secuencia: C.UGestionar Asignación Materias a Profesor

Diagrama de secuencia: C.U Asignar Materia a Profesor

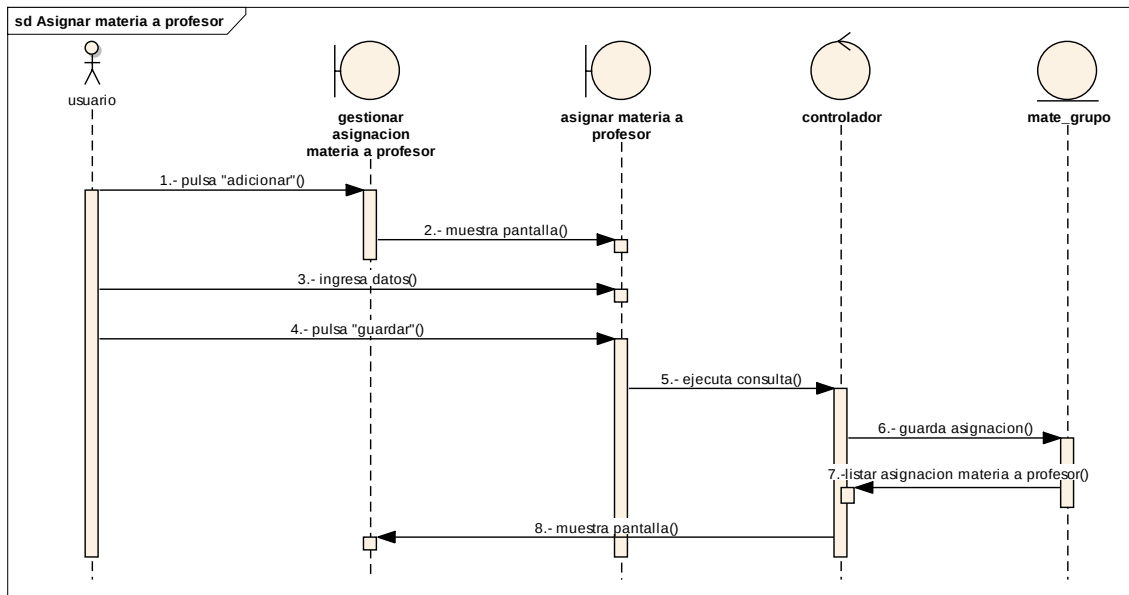


Figura nº 164 Diagrama de secuencia: C.UAsignar Materia a Profesor

Diagrama de secuencia: C.UModificar Asignación materia a profesor

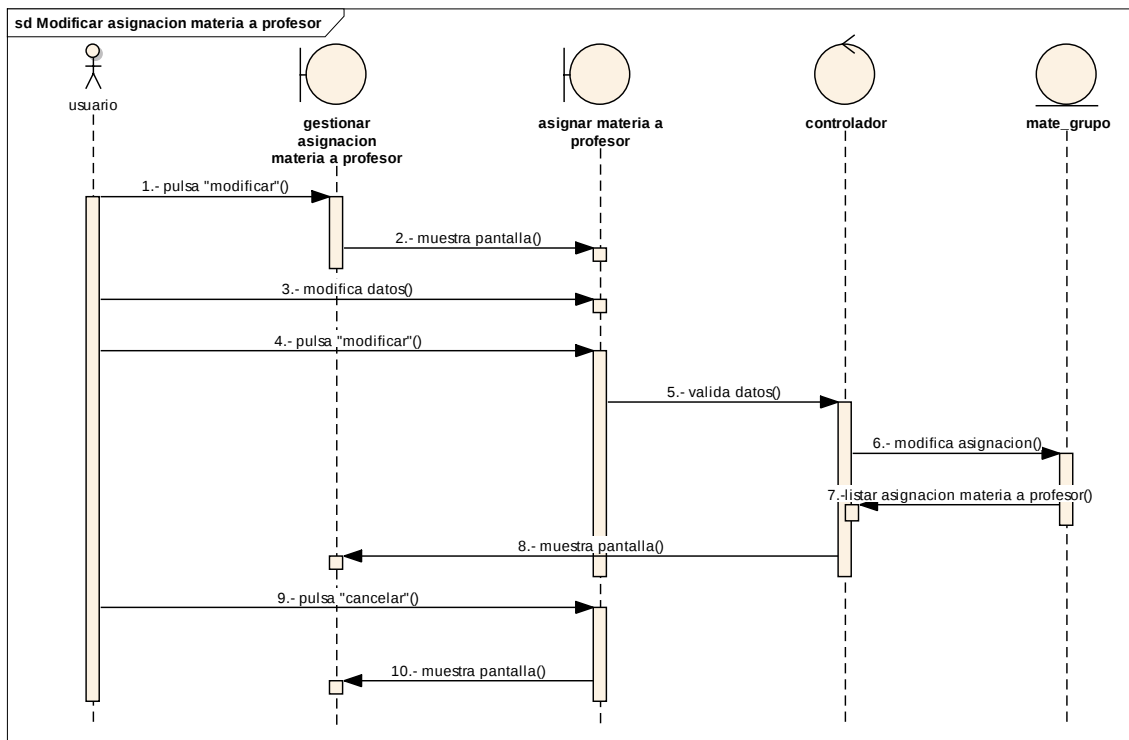


Figura nº 165 Diagrama de secuencia: C.UModificar Asignación materia a profesor

Diagrama de secuencia: C.UGestionar asignación de profesor a grupo

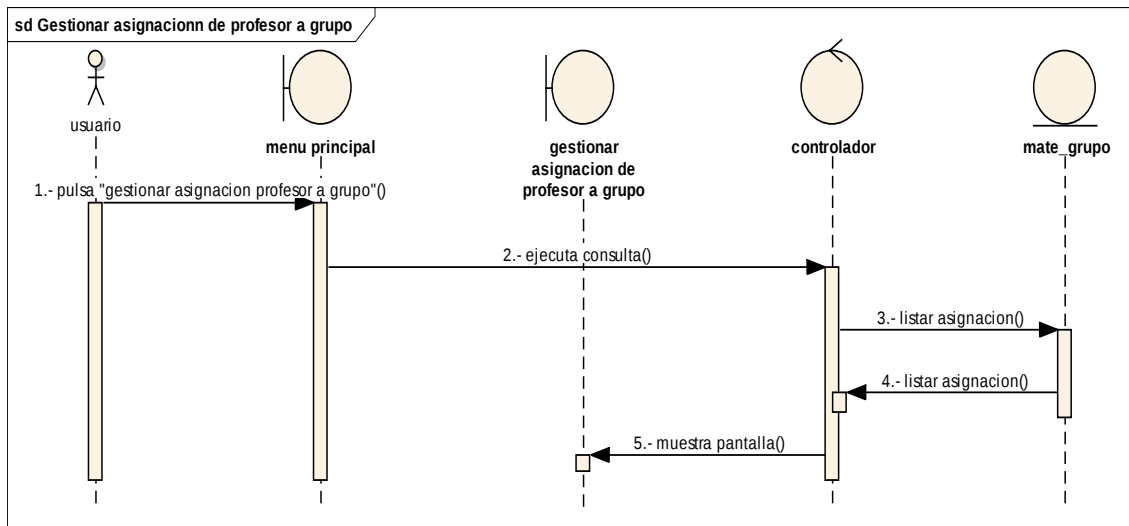


Figura nº 166 Diagrama de secuencia: C.U Gestionar asignación de profesor a grupo

Diagrama de secuencia: C.U Ver Materias del Grupo

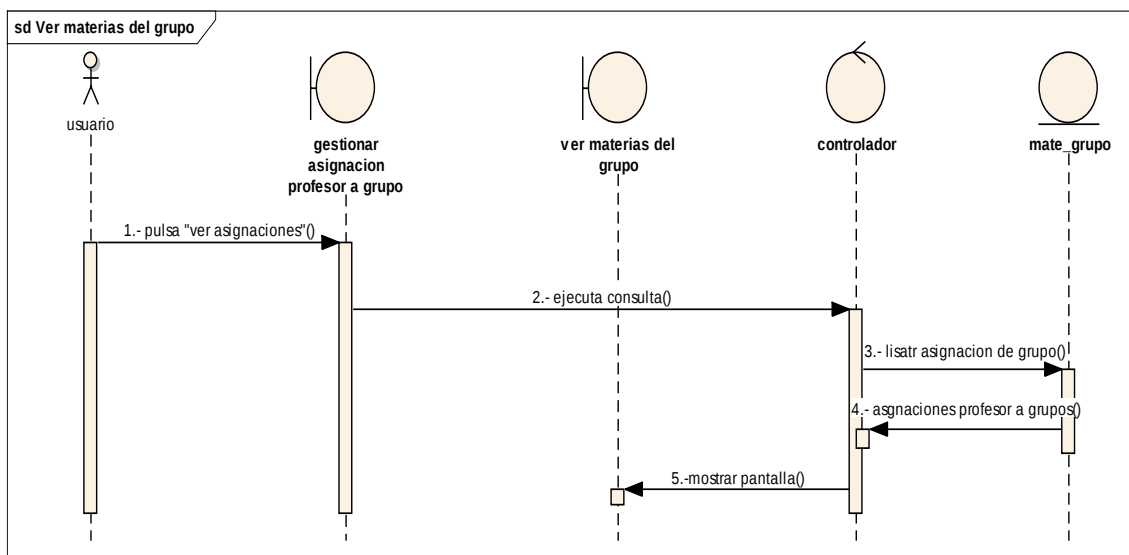


Figura nº 167 Diagrama de secuencia: C.U Ver Materias del Grupo

Diagrama de secuencia: C.U Asignar Profesor a Grupo

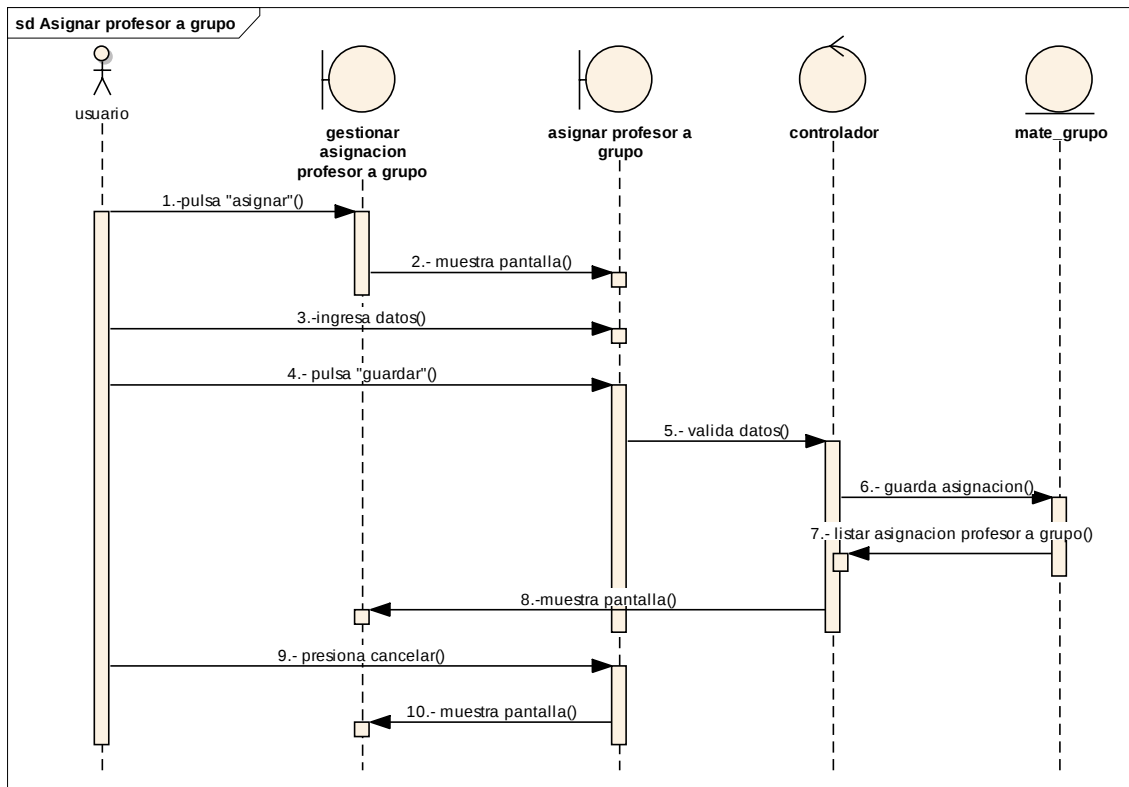


Figura nº 168 Diagrama de secuencia: C.UAsignar Profesor a Grupo

Diagrama de secuencia: Tipos de Reportes

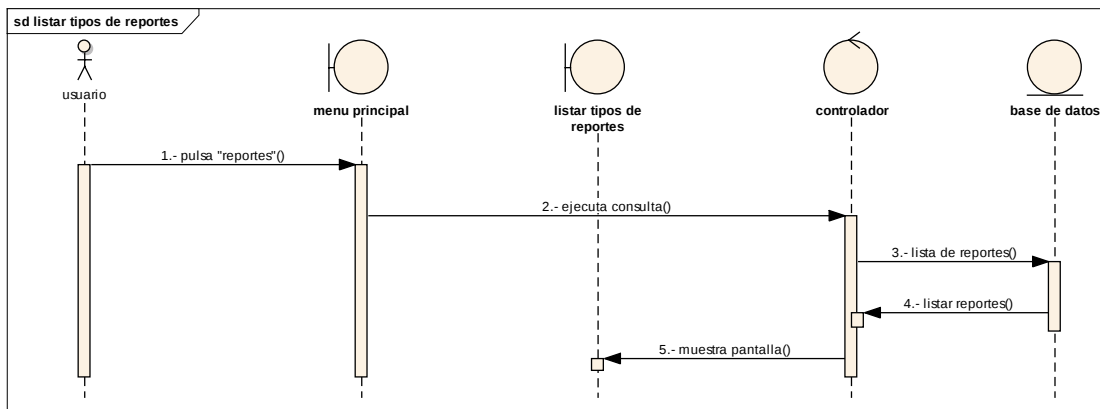


Figura nº 169 Diagrama de secuencia: Tipos de Reportes

Diagrama de secuencia: Ver Reportes

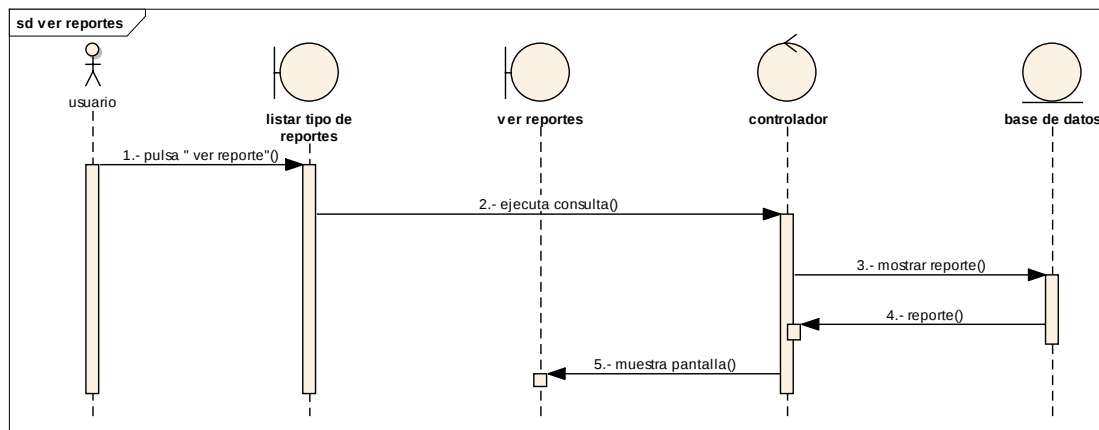


Figura nº 170 Diagrama de secuencia: Ver Reportes

2.1.2.2.5 Modelo de datos

Introducción: Previendo que la información del sistema será soportados por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un pro file UML para el Modelado de Datos, para conseguir la representación de tablas, clave, etc.)

Los Diagramas de Clases son diagramas de estructura estática que muestra las clases del sistema y sus interrelaciones (incluye herencia, agregación, asociación, etc.). Los diagramas de Clases son el pilar fundamental del modelo con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo

Propósito

- Comprende la estructura del sistema deseado para la Organización.
- Identificar posibles mejoras.

Alcance

- Describir las tablas de diseño del sistema en su segunda iteración.
- Identificar y definir las relaciones entre tablas según los objetivos del

sistema deseado aprobado por la Organización.

1.1.2.2.5 Diagrama de Base de Datos

2.1.2.2.7 Traducción de las clases implementadas al SGBD postgresql

Tabla: Alumn_famil

```
CREATE TABLE alumn_famil
(
  codfam integer,
  codparentesco integer,
  estado integer,
  codalumfa integer NOT NULL,
  CONSTRAINT alumn_famil_pkey PRIMARY KEY (codalumfa),
  CONSTRAINT alumn_famil_codfam_fkey FOREIGN KEY (codfam)
```

```

REFERENCES familiar (codfam) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT alumn_famil_parentesco_fkey FOREIGN KEY (codparentesco)
REFERENCES parentesco (codparentesco) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE alumn_famil OWNER TO postgres;

```

Tabla: Alumno

```

CREATE TABLE alumno
(
  codalum integer NOT NULL,
  nombre character varying,
  ap character varying,
  am character varying,
  f_nac date,
  sexo integer,
  num_herm integer,
  telefono integer,
  lengua_materna character varying,
  estado integer,
  otra_lengua character varying,
  identificacion character varying,
  trabajo character varying,
  docidentificacion character varying,
  nroidentificacion integer,
  codprov integer,
  codsalud integer,
  codcomunicacion integer,
  codservicio integer,
  codcertificado integer,
  codalumfa integer,
  coddireccion integer,
  gestion integer,
  CONSTRAINT alumno_pkey PRIMARY KEY (codalum),
  CONSTRAINT alumno_codalumfa_fkey FOREIGN KEY (codalumfa)
REFERENCES alumn_famil (codalumfa) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT alumno_codcertificado_fkey FOREIGN KEY (codcertificado)
REFERENCES certificado_nac (codcertificado) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT alumno_codcomunicacion_fkey FOREIGN KEY (codcomunicacion)
REFERENCES comunicaciontransporte (codcomunicacion) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,

```

```

CONSTRAINT alumno_coddireccion_fkey FOREIGN KEY (coddireccion)
  REFERENCES direccion (coddireccion) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT alumno_codprov_fkey FOREIGN KEY (codprov)
  REFERENCES provincia (codprov) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT alumno_cods salud_fkey FOREIGN KEY (codsalud)
  REFERENCES salud (codsalud) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT alumno_codservicio_fkey FOREIGN KEY (codservicio)
  REFERENCES servicios (codservicio) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE alumno OWNER TO postgres;

```

Tabla: Aula

```

CREATE TABLE aula
(
  codaula integer NOT NULL,
  nombreaula character varying,
  condicion character varying,
  capacidad integer,
  estado integer,
  CONSTRAINT aula_pkey PRIMARY KEY (codaula)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE aula OWNER TO postgres;

```

Tabla: Certificado_nac

```

CREATE TABLE certificado_nac
(
  oficialia integer,
  libro integer,
  partida integer,
  folio integer,
  codcertificado integer NOT NULL,
  CONSTRAINT certificado_nac_pkey PRIMARY KEY (codcertificado)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE certificado_nac OWNER TO postgres;

```

Tabla: Comunicaciontransporte

```

CREATE TABLE comunicaciontransporte
(
codcomunicacion integer NOT NULL,
radio character varying,
televisor character varying,
telefono character varying,
celular character varying,
computadora character varying,
internet character varying,
otro character varying,
transporte character varying,
otro_transporte character varying,
recorrido_metros integer,
demora_horas character varying,
demora_minutos character varying,
recorrido_km integer,
CONSTRAINT comunicaciontransporte_pkey PRIMARY KEY (codcomunicacion)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE comunicaciontransporte OWNER TO postgres;

```

Tabla: Datos

```

CREATE TABLE datos
(
  codda integer NOT NULL,
codusu integer,
  usuario character varying(10),
clave character varying(10),
  f_asig date,
estado integer,
CONSTRAINT datos_pkey PRIMARY KEY (codda),
CONSTRAINT datos_codusu_fkey FOREIGN KEY (codusu)
  REFERENCES usuario (codusu) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE datos OWNER TO postgres;

```

Tabla: Departamento

```

CREATE TABLE departamento
(
coddpto integer NOT NULL,
nombredpto character varying(30),
detalle character varying(30),
estado integer,
codpais integer,
CONSTRAINT departamento_pkey PRIMARY KEY (coddpto),
CONSTRAINT departamento_codpais_fkey FOREIGN KEY (codpais)
REFERENCES pais (codpais) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE departamento OWNER TO postgres;

```

Tabla: Direccion

```

CREATE TABLE direccion
(
coddireccion integer NOT NULL,
seccion_municipio character varying,
canton character varying,
localidad_comunidad character varying,
zona_villa character varying,
avenida_calle character varying,
nro_vivienda integer,
telefono_celular integer,
codprov integer,
CONSTRAINT direccion_pkey PRIMARY KEY (coddireccion),
CONSTRAINT direccion_codprov_fkey FOREIGN KEY (codprov)
REFERENCES provincia (codprov) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE direccion OWNER TO postgres;

```

Tabla: Familiar

```

CREATE TABLE familiar
(
codfam integer NOT NULL,
nombre character varying(30),
ap character varying(30),
am character varying(30),
sexo integer,

```

```

ocupacion character varying(30),
telefono integer,
ginstruccion_max character varying,
idioma character varying,
CONSTRAINT familiar_pkey PRIMARY KEY (codfam)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE familiar OWNER TO postgres;

```

Tabla: Gestion

```

CREATE TABLE gestion
(
  desde date,
  hasta date,
  estado smallint,
  gestion integer NOT NULL,
  CONSTRAINT gestion_pkey PRIMARY KEY (gestion)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE gestion OWNER TO postgres;

```

Tabla: Grado

```

CREATE TABLE grado
(
  codgra integer NOT NULL,
  nombregra character varying(30),
  detalle character varying(30),
  codniv integer,
  estado integer,
  CONSTRAINT grado_pkey PRIMARY KEY (codgra),
  CONSTRAINT grado_codniv_fkey FOREIGN KEY (codniv)
    REFERENCES nivel (codniv) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE grado OWNER TO postgres;

```

Tabla: Grupo

```

CREATE TABLE grupo
(
codgru integer NOT NULL,
nombregru character varying(30),
detalle character varying(30),
codaula integer,
codgra integer,
estado integer,
codtutor integer,
CONSTRAINT grupo_pkey PRIMARY KEY (codgru),
CONSTRAINT grupo_codaula_fkey FOREIGN KEY (codaula)
REFERENCES aula (codaula) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT grupo_codgra_fkey FOREIGN KEY (codgra)
REFERENCES grado (codgra) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT grupo_codtutor_fkey FOREIGN KEY (codtutor)
REFERENCES tutor (codtutor) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE grupo OWNER TO postgres;

```

Tabla: Kardex

```

CREATE TABLE kardex
(
codkardex integer NOT NULL,
f_ingreso date,
conducta_refe character varying,
colegio_anterior character varying,
f_llenado date,
estado integer,
codgru integer,
codusu integer,
gestion integer,
codalum integer,
CONSTRAINT kardex_pkey PRIMARY KEY (codkardex),
CONSTRAINT kardex_codalum_fkey FOREIGN KEY (codalum)
REFERENCES alumno (codalum) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT kardex_codgru_fkey FOREIGN KEY (codgru)
REFERENCES grupo (codgru) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT kardex_codusu_fkey FOREIGN KEY (codusu)
REFERENCES usuario (codusu) MATCH SIMPLE

```

```

        ON UPDATE NO ACTION ON DELETE NO ACTION
    )
    WITH (
        OIDS=FALSE
    );
ALTER TABLE kardex OWNER TO postgres;

```

Tabla: Materia

```

CREATE TABLE materia
(
    codmat integer NOT NULL,
    nombremate character varying(30),
    detalle character varying(30),
    estado character varying(30),
    CONSTRAINT materia_pkey PRIMARY KEY (codmat)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE materia OWNER TO postgres;

```

Tabla: Materia_grupo

```

CREATE TABLE materia_grupo
(
    codmategru integer NOT NULL,
    codgru integer,
    detalle character varying(30),
    codperiodo integer,
    gestion integer,
    codprofe integer,
    codmat integer,
    estado integer,
    CONSTRAINT materia_grupo_pkey PRIMARY KEY (codmategru),
    CONSTRAINT materia_grupo_codgru_fkey FOREIGN KEY (codgru)
        REFERENCES grupo (codgru) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT materia_grupo_codmat_fkey FOREIGN KEY (codmat)
        REFERENCES materia (codmat) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT materia_grupo_codperiodo_fkey FOREIGN KEY (codperiodo)
        REFERENCES periodo (codperiodo) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT materia_grupo_codprofe_fkey FOREIGN KEY (codprofe)
        REFERENCES profesor (codprofe) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT materia_grupo_gestion_fkey FOREIGN KEY (gestion)

```

```

REFERENCES gestion (gestion) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE materia_grupo OWNER TO postgres;

```

Tabla: Matgrado

```

CREATE TABLE matgrado
(
  codmatgra integer NOT NULL,
  codmat integer,
  codgra integer,
  estado integer,
  CONSTRAINT matgrado_pkey PRIMARY KEY (codmatgra),
  CONSTRAINT matgrado_codgra_fkey FOREIGN KEY (codgra)
    REFERENCES grado (codgra) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT matgrado_codmat_fkey FOREIGN KEY (codmat)
    REFERENCES materia (codmat) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE matgrado OWNER TO postgres;

```

Tabla: Menu

```

CREATE TABLE menu
(
  codmenu integer NOT NULL,
  nombremenu character varying,
  url character varying,
  CONSTRAINT menu_pkey PRIMARY KEY (codmenu)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE menu OWNER TO postgres;

```

Tabla: Nivel

```

CREATE TABLE nivel
(

```

```

codniv integer NOT NULL,
nombrenivel character varying(30),
detalle character varying(30),
estado integer,
codturno integer,
CONSTRAINT nivel_pkey PRIMARY KEY (codniv),
CONSTRAINT nivel_codturno_fkey FOREIGN KEY (codturno)
    REFERENCES turno (codturno) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE nivel OWNER TO postgres;

```

Tabla: Notas

```

CREATE TABLE notas
(
    codnotas integer NOT NULL,
    nota integer,
    detalle character varying,
    nota2 integer,
    nota3 integer,
    nota4 integer,
    nota5 integer,
    nota6 integer,
    nota7 integer,
    nota8 integer,
    bimestre1 integer,
    bimestre2 integer,
    bimestre3 integer,
    bimestre4 integer,
    notafinal integer,
    CONSTRAINT notas_pkey PRIMARY KEY (codnotas)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE notas OWNER TO postgres;

```

Tabla: Notkarmat

```

CREATE TABLE notkarmat
(
    codnotas integer,
    codkardex integer,
    codmat integer,
    codnotkarmat integer NOT NULL,
    CONSTRAINT notkarmat_pkey PRIMARY KEY (codnotkarmat),

```

```

CONSTRAINT notkarmat_codkardex_fkey FOREIGN KEY (codkardex)
  REFERENCES kardex (codkardex) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT notkarmat_codmat_fkey FOREIGN KEY (codmat)
  REFERENCES materia (codmat) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT notkarmat_codnotas_fkey FOREIGN KEY (codnotas)
  REFERENCES notas (codnotas) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE notkarmat OWNER TO postgres;

```

Tabla: Pais

```

CREATE TABLE pais
(
  codpais integer NOT NULL,
  nombrepais character varying,
  estado integer,
  detalle character varying,
  CONSTRAINT pais_pkey PRIMARY KEY (codpais)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE pais OWNER TO postgres;

```

Tabla: Parentesco

```

CREATE TABLE parentesco
(
  codparentesco integer NOT NULL,
  nombreparentesco character varying(30),
  detalle character varying(30),
  estado integer,
  CONSTRAINT parentesco_pkey PRIMARY KEY (codparentesco)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE parentesco OWNER TO postgres;

```

Tabla: Periodo

```

CREATE TABLE periodo
(

```

```

codperiodo integer NOT NULL,
nombreperiodo character varying(30),
detalle character varying(30),
horaini character varying(10),
horafin character varying(30),
estado integer,
CONSTRAINT periodo_pkey PRIMARY KEY (codperiodo)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE periodo OWNER TO postgres;

```

Tabla: Profe_materia

```

CREATE TABLE profe_materia
(
  codprofemate integer NOT NULL,
  codmat integer,
  codprofe integer,
  detalle character varying(50),
  f_asignacion date,
  estado integer,
  CONSTRAINT profe_materia_pkey PRIMARY KEY (codprofemate),
  CONSTRAINT profe_materia_codmat_fkey FOREIGN KEY (codmat)
    REFERENCES materia (codmat) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT profe_materia_codprofe_fkey FOREIGN KEY (codprofe)
    REFERENCES profesor (codprofe) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE profe_materia OWNER TO postgres;

```

Tabla: Profe_profesion

```

CREATE TABLE profe_profesion
(
  codpp integer NOT NULL,
  codprofe integer,
  codprofesion integer,
  detalle character varying(50),
  "año graduacion" date,
  "años_experiencia" integer,
  curriculum character varying(50),
  estado integer,
  coddpto integer,

```

```

CONSTRAINT profe_profesion_pkey PRIMARY KEY (codpp),
CONSTRAINT profe_profesion_coddpto_fkey FOREIGN KEY (coddpto)
  REFERENCES departamento (coddpto) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profe_profesion_codprofe_fkey FOREIGN KEY (codprofe)
  REFERENCES profesor (codprofe) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT profe_profesion_codprofesion_fkey FOREIGN KEY (codprofesion)
  REFERENCES profesion (codprofesion) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE profe_profesion OWNER TO postgres;

```

Tabla: Profesion

```

CREATE TABLE profesion
(
  codprofesion integer NOT NULL,
  nomprofesion character varying(30),
  detalle character varying(30),
  estado integer,
  CONSTRAINT profesion_pkey PRIMARY KEY (codprofesion)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE profesion OWNER TO postgres;

```

Tabla: Profesor

```

CREATE TABLE profesor
(
  codprofe integer NOT NULL,
  ci integer,
  nombre character varying(30),
  ap character varying(30),
  am character varying(30),
  sexo integer,
  ecivil character varying(20),
  lengua_materna character varying(30),
  otra_lengua character varying(30),
  codprov integer,
  direccion character varying(50),
  estado integer,
  CONSTRAINT profesor_pkey PRIMARY KEY (codprofe),
  CONSTRAINT profesor_codprov_fkey FOREIGN KEY (codprov)

```

```

REFERENCES provincia (codprov) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE profesor OWNER TO postgres;

```

Tabla: Provincia

```

CREATE TABLE provincia
(
  codprov integer NOT NULL,
  nombreprov character varying(30),
  coddpto integer,
  estado integer,
  CONSTRAINT provincia_pkey PRIMARY KEY (codprov),
  CONSTRAINT provincia_coddpto_fkey FOREIGN KEY (coddpto)
    REFERENCES departamento (coddpto) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE provincia OWNER TO postgres;

```

Tabla: Rol

```

CREATE TABLE rol
(
  codrol integer NOT NULL,
  estado integer,
  detalle character varying,
  nomrol character varying,
  CONSTRAINT rol_pkey PRIMARY KEY (codrol)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE rol OWNER TO postgres;

```

Tabla: Rol menu

```

CREATE TABLE rol_menu
(
  codrol integer,
  f_asig date,
  estado integer,
  codmenu integer,

```

```

codrolmenu integer NOT NULL,
CONSTRAINT rol_menu_pkey PRIMARY KEY (codrolmenu),
CONSTRAINT rol_menu_codmenu_fkey FOREIGN KEY (codmenu)
REFERENCES menu (codmenu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT rol_menu_codrol_fkey FOREIGN KEY (codrol)
REFERENCES rol (codrol) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE rol_menu OWNER TO postgres;

```

Tabla: Salud

```

CREATE TABLE salud
(
  codsalud integer NOT NULL,
  seguro character varying,
  acude_casos character varying,
  grupo_sangre character varying,
  discapacidad character varying,
  su_discapacidad character varying,
  CONSTRAINT "Salud_pkey" PRIMARY KEY (codsalud)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE salud OWNER TO postgres;

```

Tabla: Servicios

```

CREATE TABLE servicios
(
  codservicio integer NOT NULL,
  agua character varying,
  electricidad character varying,
  gas character varying,
  alcantarillado character varying,
  otro character varying,
  CONSTRAINT servicios_pkey PRIMARY KEY (codservicio)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE servicios OWNER TO postgres;

```

Tabla: Turno

```

CREATE TABLE turno
(
    codturno integer NOT NULL,
    nombretorno character varying(30),
    detalle character varying(30),
    estado integer,
    CONSTRAINT turno_pkey PRIMARY KEY (codturno)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE turno OWNER TO postgres;

```

Tabla: Tutor

```

CREATE TABLE tutor
(
    codmategru integer,
    consultas character varying(400),
    dia character varying(400),
    fechaasignacion date,
    codtutor integer NOT NULL,
    CONSTRAINT tutor_pkey PRIMARY KEY (codtutor),
    CONSTRAINT tutor_codmategru_fkey FOREIGN KEY (codmategru)
    REFERENCES materia_grupo (codmategru) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE tutor OWNER TO postgres;

```

Tabla: Usuario

```

CREATE TABLE usuario
(
    codusu integer NOT NULL,
    ci integer,
    nombre character varying(30),
    ap character varying(30),
    am character varying(30),
    sexo integer,
    ecivil character varying(30),
    f_nac date,
    lengua_materna character varying(30),
    direccion character varying(30),
    estado integer,

```

```

codrol integer,
aux integer,
CONSTRAINT usuario_pkey PRIMARY KEY (codusu),
CONSTRAINT usuario_codrol_fkey FOREIGN KEY (codrol)
    REFERENCES rol (codrol) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE usuario OWNER TO postgres;

```

2.1.2.2.8 Casos de Prueba

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

dependiendo del tipo de prueba dicho proceso podrá ser automatizable mediante un script de prueba.

2.1.2.2.8.1 Plan de Pruebas

2.1.2.2.8.1.1 Descripción de Aspectos Generales

Esta sección establece el alcance y el objetivo del plan de pruebas. Es aquí donde se describen los aspectos fundamentales del esfuerzo que se hará para probar cada uno de los módulos que conforman el sistema descrito en este plan de desarrollo de software, independiente las características y el tamaño que esta puede tener.

2.1.2.2.8.1.1.1 Objetivo

Este plan de pruebas fue desarrollado con el fin de guiar el proceso de pruebas al proyecto SISTEMFYA que fue desarrollado por nuestro equipo de trabajo, esto con el fin de asegurar una excelente calidad de software desarrollado, encontrando para ello errores que puedan perjudicar en el funcionamiento de este, los cuales en una vez corregidos, se podrán elaborar un documento que presente las evaluaciones correspondientes en el plazo mas corto posible.

III. Capítulo 3: conclusiones y recomendaciones

1.1.1.1 Resultados Esperados

Personal del colegio Fe y Alegría designados para el manejo del sistema informático capacitados en el manejo adecuado del sistema.

1.1.1.2 Medios de Verificación

- ☐ Carta del Lic. Marco Antonio Uriburo Director del Colegio Fe y Alegría “Humberto Portocarrero I” citando la Conformidad con el Proyecto Realizado
- ☐ Listas firmadas, fotos de la capacitación al personal designado por el colegio.

1.1.1.3 Conclusiones

Se espera que la capacitación se lleve dentro del marco ya establecido con anterioridad donde es establecido un cronograma de cómo se va a llevar a cabo la capacitación.

Se espera que las personas que asistan a la capacitación tengan el interés necesario para poder aprender lo que se capacitara y que salgan de la capacitación con una mente positiva sobre el proyecto.