

CAPÍTULO I

PROYECTO

I.1 PRESENTACIÓN DEL PROYECTO

I.1.1 TÍTULO

“Optimización en el Uso del Sistema Tariquia en Dispositivos Móviles de los Estudiantes de la Universidad Autónoma Juan Misael Saracho a través de una Aplicación Android”.

I.1.2 ÁREA DEL PROYECTO

Software (SOF).

I.1.3 DIRECTOR DEL PROYECTO

Ramirez Apellido Paterno	Villa Apellido Materno	Luis Gonzalo Nombres	5807917 C.I.
Grupo 1 Grupo de Taller III	Ingeniería Informática Carrera	Facultad de Ciencias y Tecnología	
66 – 31891 Teléfono	70234629 Celular	gonzalo.ramirez.villa@gmail.com Correo electrónico	Firma

TABLA 1: DIRECTOR DEL PROYECTO

I.1.4 INSTITUCIÓN COOPERANTE

Nombre: Universidad Autónoma Juan Misael Saracho			
Dirección: Av. Víctor Paz Nro. 149 Casilla 51		Telef. Oficina: 6645097-6643232-6643117-6643113-6645096	
Nombre y Apellidos	Cargo	C.I.	Firma
Andrés Abel Mamani Quiquinta	Ing. Dpto. Tecnologías de Información y Comunicación	3961444	

TABLA 2: INSTITUCIÓN COOPERANTE

I.1.5 ACTIVIDADES

Responsable	Actividades
Director	<p>Definir y concretar los objetivos del proyecto.</p> <p>Planificar y dirigir el proyecto, identificando las actividades a realizar, los plazos y costes previstos.</p> <p>Recopilación de la información requerida para la elaboración del sistema.</p> <p>Análisis y Diseño del Sistema.</p> <p>Desarrollo del Software.</p> <p>Fase de Pruebas del Sistema.</p> <p>Diseño y Documentación del manual de usuario.</p> <p>Diseño de la socialización del Sistema.</p>
Tutor	<p>Coordinar el proceso de evaluación.</p> <p>Orientar y asesorar al estudiante sobre sus posibilidades contempladas en el proyecto.</p> <p>Supervisar el desarrollo del proyecto, de acuerdo a una planificación calendarizada.</p> <p>Facilitar la integración de información recabada para llevar a cabo el buen desarrollo.</p>

TABLA 3: ACTIVIDADES

I.2 DESCRIPCIÓN DEL PROYECTO

I.2.1 DESCRIPCIÓN Y FUNDAMENTACIÓN DEL PROYECTO

El presente proyecto propone soluciones a las necesidades que presentan los estudiantes de la Universidad Autónoma Juan Misael Saracho, que después de un estudio respaldado por metodologías de obtención de datos a los afectados de esta institución, se llegó a la conclusión de diseñar y elaborar una aplicación móvil basada en el sistema operativo Android para móviles que permita facilitar y mejorar el acceso a algunos de los datos más esenciales para los estudiantes como ser: mostrar horarios de las materias asignadas, ver sus correspondientes notas y ver su ficha académica de estudiante, toda esta información obtenida de su cuenta en el actual sistema Tariquia.

La Universidad Autónoma Juan Misael Saracho en la actualidad cuenta con un sistema académico web que permite visualizar información a un estudiante matriculado en la Universidad, entre ellas: ver notas, horarios de las materias designadas, docentes asignados, programación de materias, aulas virtuales, obtener reportes entre otros, todo esto en un navegador web.

Actualmente el avance de la tecnología en los teléfonos móviles nos permite tener acceso a internet para poder enviar y recibir información de manera más rápida, en cualquier momento y lugar que uno desee, pero la universidad no aprovecha las tecnologías que existen actualmente en el mercado originando una desventaja competitiva en el ámbito de aprovechamiento de tecnología, por ende pierde prestigio frente a la competencia.

De los datos recolectados en las encuestas y entrevistas se evidenció que el 73,7% de los encuestados presenta quejas por la adaptabilidad del sistema al momento de navegar en su celular. Para obtener esta información necesitan entrar a un navegador web e ingresar al sistema Tariquia.

Entre los datos recolectados también se pudo observar que el 83,51% de estudiantes cuenta con un dispositivo móvil inteligente, el 73,7% basado en Android, y el 73,1% de los estudiantes usa más su móvil que un ordenador para ingresar al sistema Tariquia para adquirir información.

Basados en estos datos y dado el hecho que el sistema Tariquia no cuenta con una alternativa de solución para móviles, es por esto que se presenta este proyecto para tratar de brindar una solución a este problema en la institución.

Como otro componente del proyecto se pretende llevar a cabo una estrategia de socialización del proyecto que resalte los beneficios de la implementación de este proyecto en los estudiantes de la Universidad Autónoma Juan Misael Saracho.

I.2.1.1 ANÁLISIS DE CAUSAS DEL PROBLEMA

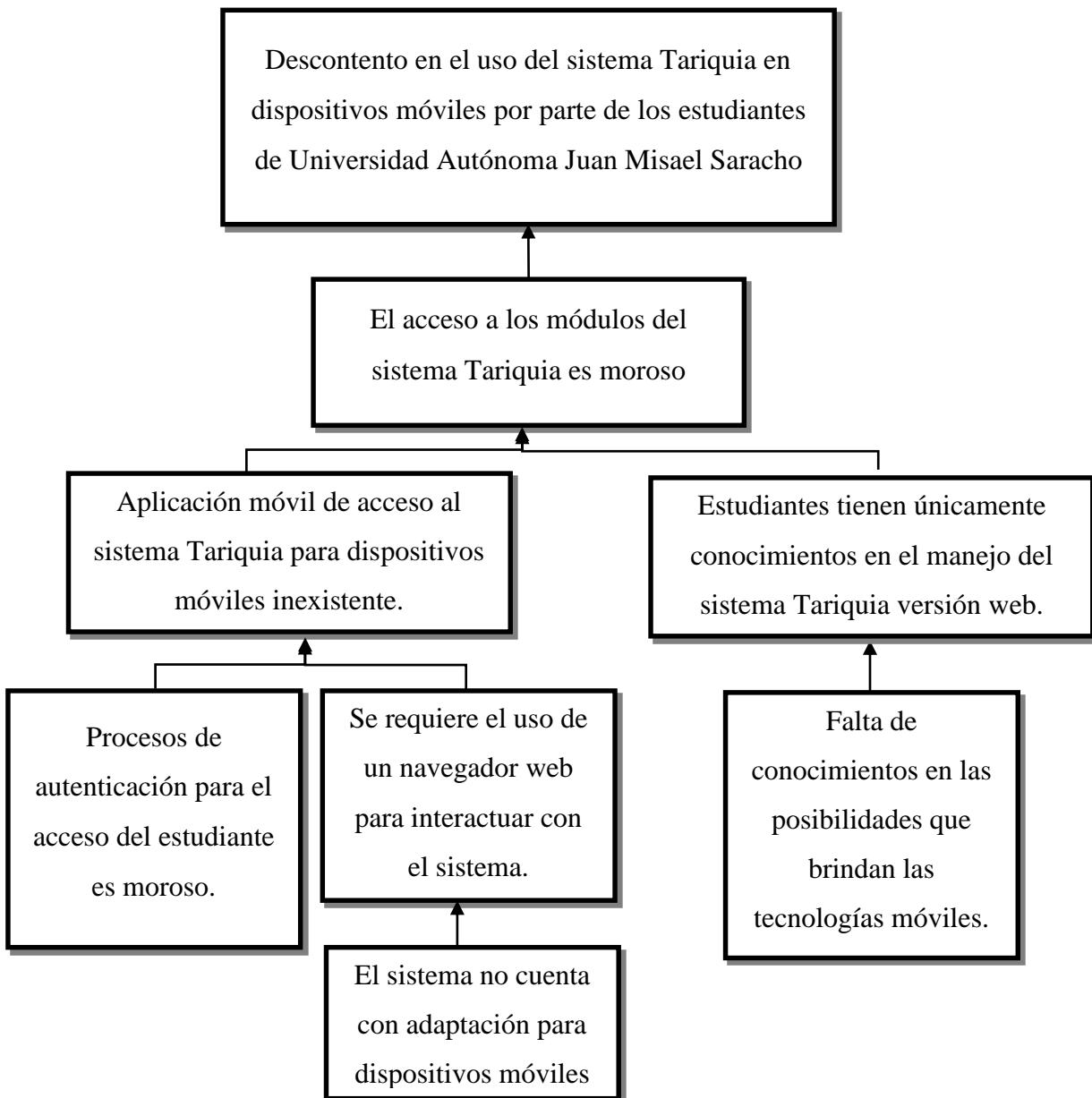


FIGURA 1: ÁRBOL DE PROBLEMAS

I.2.1.2 ANÁLISIS DE OBJETIVOS

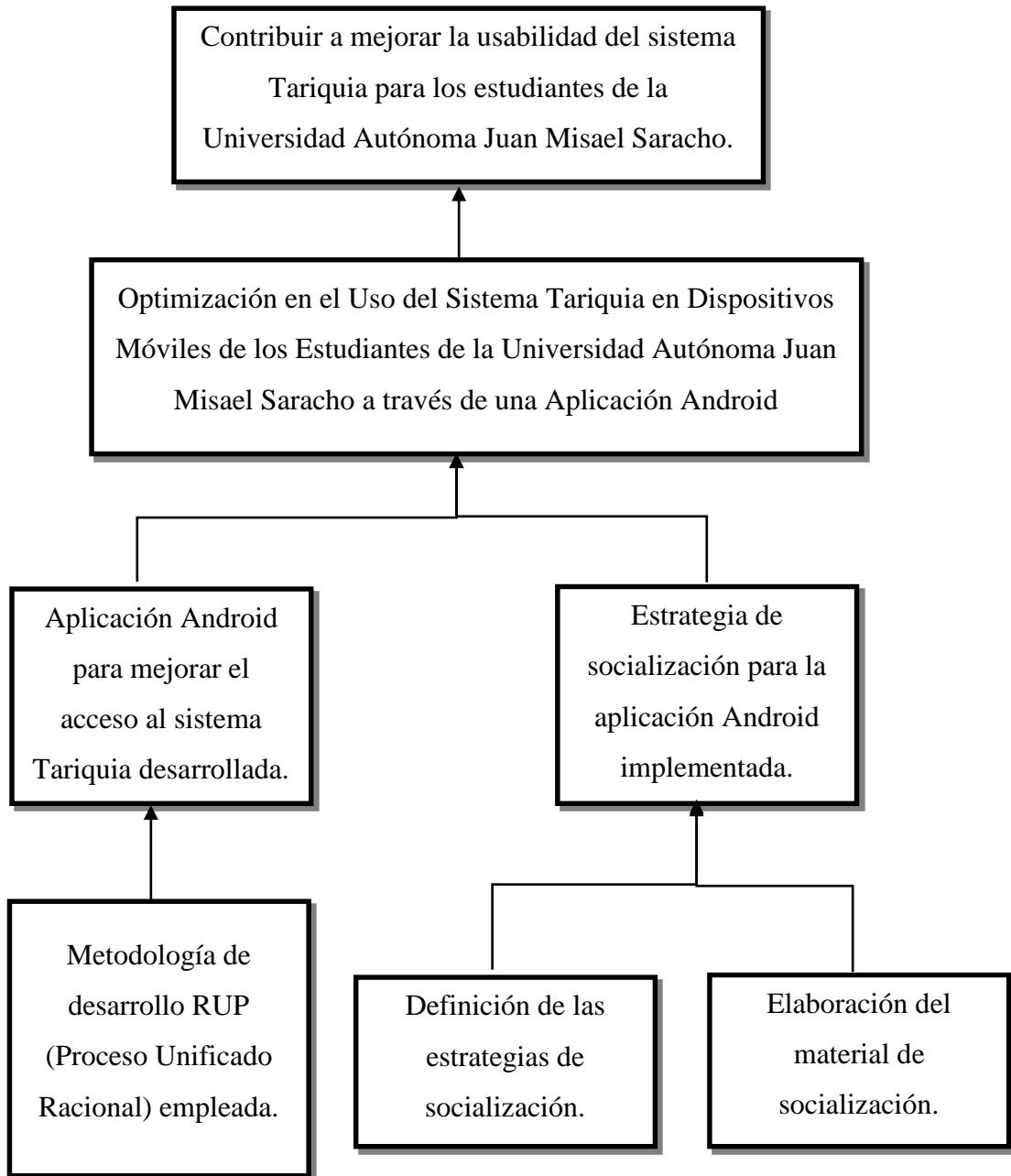


FIGURA 2: ÁRBOL DE OBJETIVOS

I.2.1.3 FLUJO DE TRABAJO CON Y SIN PROYECTO

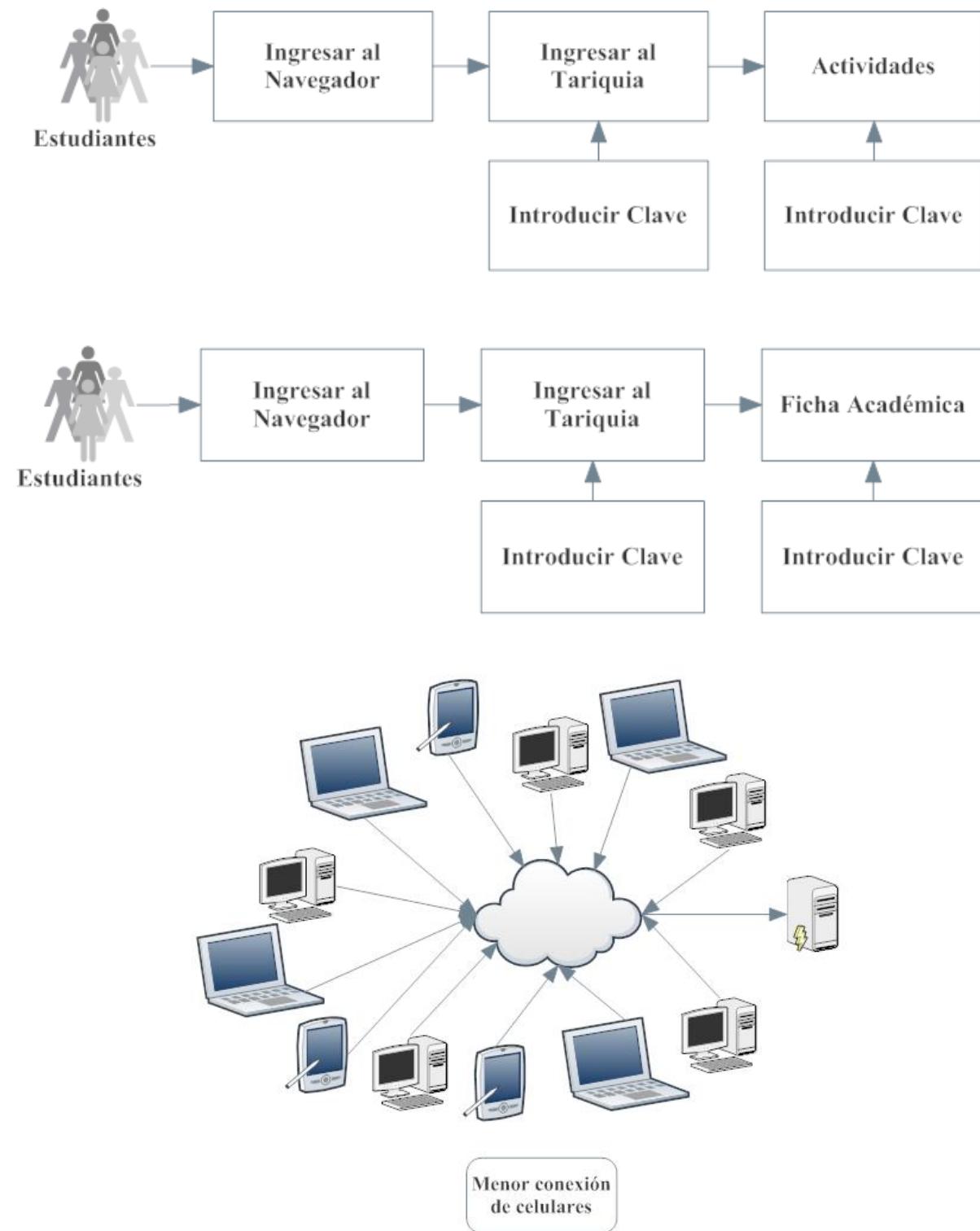


FIGURA 3: FLUJO DE TRABAJO (ANTES DE LA APP)

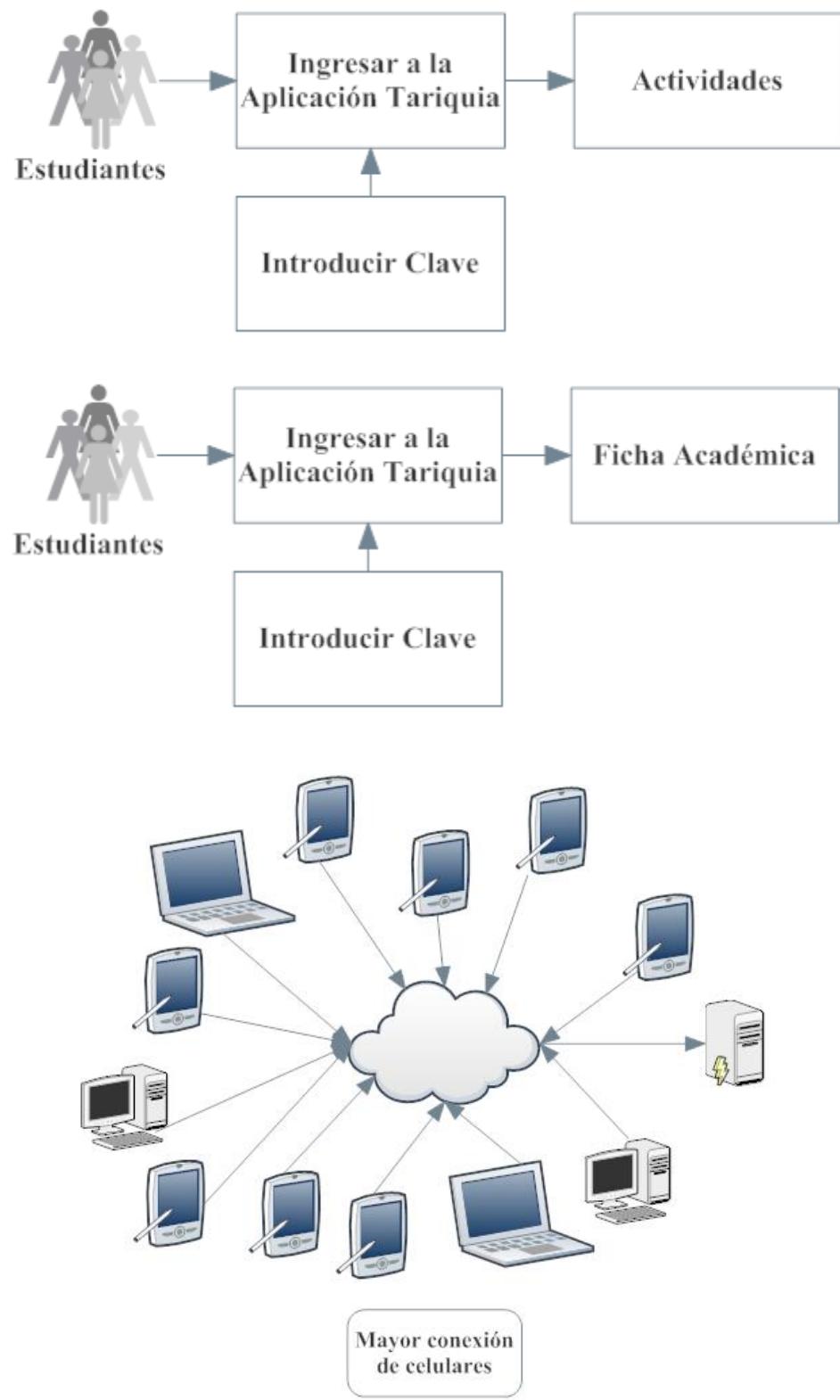


FIGURA 4: FLUJO DE TRABAJO (APLICACIÓN PROPUESTA)

I.2.1.4 SITUACIÓN PLANTEADA CON Y SIN PROYECTO

SITUACIÓN SIN PROYECTO	SITUACIÓN CON PROYECTO
<p>Los estudiantes no pueden contar con información importante en momentos oportunos.</p> <p>La obtención de información es moroso.</p> <p>Procesos de autenticación para el acceso del estudiante es tedioso.</p> <p>Navegabilidad en el sistema es compleja.</p> <p>Se requiere el uso de un navegador web para interactuar con el sistema.</p> <p>El sistema no cuenta con adaptación para dispositivos móviles.</p>	<p>Los estudiantes pueden contar con información importante en momentos oportunos.</p> <p>La obtención de información es oportuna.</p> <p>Procesos de autenticación para el acceso del estudiante optimizado.</p> <p>Navegabilidad en el sistema mejorada.</p> <p>El sistema cuenta con adaptación para dispositivos móviles.</p>

TABLA 4: SITUACIÓN PLANTEADA

I.2.2 OBJETIVOS

I.2.2.1 OBJETIVO GENERAL

Mejorar el acceso al sistema Tariquia en dispositivos móviles, incorporando una aplicación Android.

I.2.2.2 OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación Android para mejorar el acceso al sistema Tariquia a través de la metodología RUP.
- Implementar estrategias de socialización para la aplicación Android desarrollada.

I.2.3 MARCO LÓGICO DEL PROYECTO

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a A un año de Reporte de las La UAJMS mantiene mejorar la implementada descargas realizadas las condiciones usabilidad del aplicación se logra que un en la tienda de tecnológicas para un sistema Tariquia 50% de los estudiantes aplicaciones de uso efectivo de la para los estudiantes con dispositivos móviles Google (Google de la Universidad Android descarguen la Play). Autónoma Juan Misael Saracho. $F = (\text{Est. con App} * 100\%) / \text{Total de Estudiantes con Android.}$ Fórmula=			

	$(8922*100\%)/17832=$ 50,03%		
Objetivo General (Propósito)	Optimización en el Uso del Sistema Tariquia en Dispositivos Móviles de los Estudiantes de la Universidad Autónoma Juan Misael Saracho a través de una Aplicación Android	Al finalizar el proyecto se espera reducir los procesos para la obtención de información del estudiante en un 60% $F = (\text{Cantidad de Procesos App} / \text{Cantidad de Procesos Convencional}) * 100\%$ Fórmula=(3/5)*100%	Certificado de Aprobación del Desarrollo del Proyecto por parte del encargado de sistemas de información del DTIC de la UAJMS.
Objetivos Específicos (Componentes)	1.- Aplicación Android para mejorar el acceso al sistema Tariquia desarrollada.	1.- Al finalizar el proyecto se ha completado con el 100% del desarrollo de la aplicación Android para la UAJMS de acuerdo a los requerimientos expresados bajo la Norma IEEE 830.	1.- Documentación de la aplicación Android completa. Manual del usuario y Guía de Instalación para la aplicación. Se cuenta con los recursos de hardware y software necesarios

<p>2.- Estrategia de socialización para la aplicación Android implementada.</p>	<p>2.- Al finalizar con el 100% del proyecto se ha desarrollado una estrategia de socialización en el uso de la aplicación, logrando capacitar de forma gradual a los estudiantes de la Universidad Autónoma Juan Misael Saracho con ayuda de un video de capacitación con una duración de 14 minutos.</p> <p>$F = (\text{Capacitados} * 100\%) / \text{Total de Estudiantes.}$</p>	<p>2.- Material de Socialización desarrollado</p> <p>Video capacitación realizado.</p> <p>Modelo navegacional de la aplicación.</p> <p>Folleto navegacional de la aplicación.</p>	<p>para el desarrollo de la Aplicación Android.</p> <p>2.- Se cuenta con los recursos económicos necesarios para la campaña de socialización.</p>
<p>Actividades</p> <p>Componente 1:</p> <p>1.1 Etapa de Inicio</p> <p>En esta fase se realizará la recolección de información por medio de técnicas</p>	<p>La fase de inicio comenzó el 20/03/2015 y hasta la fecha se tiene completado el 100%.</p>	<p>Informe presupuestario del proyecto, avalado por el Ing. Abel Mamani, encargado de la unidad de sistemas de información del DTIC - UAJMS.</p>	<p>Los desembolsos se realizan en el tiempo previsto</p>

<p>como: entrevistas, encuestas.</p>			
<p>1.2 Etapa de Elaboración</p> <p>En esta fase se pondrá mayor atención al diseño o arquitectura del sistema, además de elaborar la documentación que respalde el proyecto.</p>	<p>La fase de elaboración comenzó el 09/04/2015 y hasta la fecha se tiene completado el 100%.</p>		
<p>1.3 Etapa de Construcción</p> <p>En esta fase se realizará la programación de acuerdo al documento y diseño elaborado.</p>	<p>La fase de construcción comenzó el 06/06/2015 y hasta la fecha se tiene completado el 100%.</p>		
<p>1.4 Etapa de Transición</p> <p>Finalmente se obtendrá el producto y si es aprobado por el</p>			

cliente, se lo pondrá en funcionamiento.															
<p>Componente 2:</p> <p>2.1 Definición de las estrategias de socialización</p> <p>2.2 Elaboración del material de socialización</p> <p>Ver presupuesto detallado en punto 3.</p> <p>Resumen presupuesto</p> <table> <tbody> <tr> <td>10000</td> <td>Servicios personales</td> <td>40,000.-</td> </tr> <tr> <td>20000</td> <td>Servicios No personales</td> <td>8,800.-</td> </tr> <tr> <td>30000</td> <td>Materiales y Suministros</td> <td>300.-</td> </tr> <tr> <td>Total:</td> <td></td> <td>Bs.- 49,100.-</td> </tr> </tbody> </table>	10000	Servicios personales	40,000.-	20000	Servicios No personales	8,800.-	30000	Materiales y Suministros	300.-	Total:		Bs.- 49,100.-	<p>La definición de las estrategias de socialización comenzó el 02/02/2016 para llevar acabo la elaboración del material de socialización y hasta la fecha contar con el 100% de los materiales para la socialización.</p>	<p>Material de socialización de la Aplicación Android. Video de capacitación realizado.</p> <p>Modelo navegacional de la aplicación.</p> <p>Folleto navegacional de la aplicación.</p>	<p>La información requerida para la socialización se encuentra a tiempo.</p>
10000	Servicios personales	40,000.-													
20000	Servicios No personales	8,800.-													
30000	Materiales y Suministros	300.-													
Total:		Bs.- 49,100.-													

TABLA 5: MATRIZ DE MARCO LÓGICO

I.2.4 METODOLOGÍA DE TRABAJO

La metodología a utilizar para el desarrollo del sistema informático hace uso del proceso unificado RUP y de la notación del lenguaje UML.

El proceso Unificado de Rational (RUP) es un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Este proceso cuenta con las fases detalladas a continuación:

- Inicio: En esta fase se establece los requisitos de negocio que cubrirá el sistema, se obtendrá la especificación de requerimientos.
- Elaboración: En esta segunda fase el problema se analiza y comprende desde el punto de vista del equipo de desarrollo. Al final de la fase se tiene definida la arquitectura y el modelo de requisitos del sistema.
- Construcción: En esta tercera fase se profundiza en el diseño de los componentes del sistema y de manera iterativa se van añadiendo las funcionalidades al software a medida que se construyen y prueban, permitiendo a la vez se puedan ir incorporando cambios. Al final de esta fase se obtiene un sistema completamente operativo y la documentación para entregar a los usuarios.
- Transición: No se aplica al proyecto.

I.2.4.1 DESCRIPCIÓN DE LAS ACTIVIDADES

De acuerdo a las fases implicadas tenemos las siguientes actividades a llevar a cabo:

I.2.4.1.1 FASE DE INICIO

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

Modelado del Negocio

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos.

Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.

Entender el problema actual en la organización objetivo e identificar potenciales mejoras.

Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

Requisitos

Los requisitos definirán lo que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

Establecer y mantener un acuerdo entre clientes sobre lo que el sistema podría hacer.

Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.

Definir el ámbito del sistema.

Definir una interfaz de usuarios para el sistema (metáfora), enfocada a las necesidades y metas del usuario.

I.2.4.1.2 FASE DE ELABORACIÓN

En la fase de elaboración, las iteraciones se orientan al desarrollo de la base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la base de la arquitectura.

Análisis y Diseño

En esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema.

Transformar los requisitos al diseño del sistema.

Desarrollar una arquitectura para el sistema. Adaptar el diseño para que sea consistente con el entorno de implementación.

I.2.4.1.3 FASE DE CONSTRUCCIÓN

Implementación

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables. El resultado final es un sistema ejecutable.

Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración. Cada implementador decide en qué orden implementa los elementos del subsistema.

Si encuentra errores de diseño, los notifica.

Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo sino que debe ir integrado en todo el ciclo de vida. Encontrar y documentar defectos en la calidad del software.

Verificar las funciones del producto de software según lo diseñado.

El lenguaje unificado de modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

UML ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema y aspectos concretos como esquemas de bases de datos y componentes reutilizables.

I.2.4.2 DESCRIPCIÓN Y RELACIÓN DE LAS ESTRATEGIAS CON LOS OBJETIVOS

Estrategias	Objetivos Específicos
<p>Se seguirán las fases de RUP, que implica elaborar un cronograma de actividades, realizar la especificación de requerimientos y el modelado del negocio. Posteriormente se procede a diseñar la base de datos, realizar los diagramas en UML y crear los prototipos de interfaces de usuarios. Posteriormente se realiza la programación y documentación del código fuente así como la elaboración de las pruebas de la aplicación y la aplicación de las mismas.</p>	<p>Desarrollar una aplicación Android para mejorar el acceso a los estudiantes de la UAJMS al Sistema Tariquia.</p>
<p>Se definirá una estrategia de socialización con una actividad central, se elaborará el material que será usado en la socialización y culminará con la posterior realización de la actividad central de la socialización.</p>	<p>Implementar estrategias de socialización para la aplicación Android desarrollada.</p>

TABLA 6: RELACIÓN DE ESTRATEGIAS CON OBJETIVOS

I.2.4.3 CRONOGRAMA DE ACTIVIDADES

Nº	Actividad	Nº días	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
1.	Sistema Informático	202											
1.1	Determinación de Requerimientos	20	X										
1.2	Desarrollo del Análisis	42		X	X								
1.3	Desarrollo del Diseño	15				X	X						
1.4	Desarrollo de la Programación	90						X	X	X			
1.5	Diseño y Documentación (Manuales)	15									X		
1.6	Realización de Pruebas del Sistema	20										X	
2	Estrategia de Socialización	55											
2.1	Definición de la Estrategia de Socialización	20								X			
2.2	Elaboración del Material de Socialización	30									X		
2.3	Realización de la Actividad Central de Socialización	5										X	

TABLA 7: CRONOGRAMA DE ACTIVIDADES

I.2.5 RESULTADOS ESPERADOS

I.2.5.1 APLICACIÓN ANDROID PARA OPTIMIZAR EL USO DEL SISTEMA TARIQUIA DE LA UAJMS

Con el desarrollo del proyecto se pretende optimizar el acceso al sistema Tariquia para los estudiantes de la Universidad Autónoma Juan Misael Saracho en dispositivos móviles Android, permitiendo a éstos el acceso a módulos como: horarios, materias, notas y actividades.

La Aplicación Android se encuentra desarrollada de acuerdo a los requerimientos expresados bajo la norma IEEE 830, orientado a web y cumpliendo los criterios de calidad en cuanto a la funcionalidad, usabilidad, entre los más relevantes.

I.2.5.2 ESTRATEGIAS DE SOCIALIZACIÓN PARA LA APLICACIÓN ANDROID

Este componente se refiere a la realización de:

Una capacitación a distancia usando elementos multimedia como ser un video subido a YouTube y un folleto de navegación.

Con la estrategia de socialización se pretende que los beneficiarios sean capaces de instalar, hacer uso de la aplicación y conocer los beneficios que aporta esta aplicación.

I.2.6 TRANSFERENCIA DE RESULTADOS

I.2.6.1 MEDIOS Y ESTRATEGIAS PARA LA TRANSFERENCIA DE RESULTADOS

La transferencia de resultados se realizará mediante un convenio establecido entre la Universidad y los directos beneficiarios; en este convenio también se definirán los medios a ser utilizados.

I.2.6.2 GRUPO DE BENEFICIARIOS DE LOS RESULTADOS

Los beneficiarios con el desarrollo del proyecto son los estudiantes de la Universidad Autónoma Juan Misael Saracho.

I.3 PRESUPUESTO

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
10000	SERVICIOS PERSONALES			
	12000 Empleados no Permanentes	40,000.-		40,000.-
	Sub total rubro			40,000.-
20000	SERVICIOS NO PERSONALES			
	21000 Servicios Básicos	4,800.-		4,800.-
	22000 Servicios de transporte			
	23000 Alquileres			
	24000 Mantenimiento y reparación			
	25000 Servicios Profesionales y Comerciales	4,000.-		4,000.-
	Sub total rubro			8,800.-
30000	MATERIALES Y SUMINISTROS			
	31000 Alimentos y Productos Forestales			
	32000 Productos de Papel, Cartón e Impresos	300.-		300.-
	33000 Textiles y Vestuario			
	34000 Productos Químicos, Combustibles y Lubricantes			
	39000 Productos Varios			
	Sub total rubro			300.-
40000	ACTIVOS REALES			
	43000 Maquinaria y Equipo			
	46000 Descripción de estudios y proyectos para inversión			
	49000 Otros Activos			
	Sub total rubro			0
	TOTAL			49,100.-

TABLA 8: RUBROS

1) GRUPO 10000. SERVICIOS PERSONALES

a) SUB GRUPO 12000. Empleados no Permanentes

Partida	Personal*	Remuneración	Tiempo/meses	Total
12100	Personal Eventual			
	Ingeniero Junior Área Informática	Bs. 5,000.-	8	40,000.-
Total				0

* Se refiere a gastos para remunerar a personas sujetas a contrato dependientes según la necesidad de cada entidad

TABLA 9: EMPLEADOS NO PERMANENTES

2) GRUPO 20000. SERVICIOS NO PERSONALES

b) SUB GRUPO 21000. Descripción de los gastos de servicios básicos

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación	400	8	3,200.-
21200	Energía Eléctrica	200	8	1,600.-
21300	Agua			
21400	Servicios Telefónicos			
Total				4,800.-

* Se refiere principalmente a los gastos por servicios; como: servicio de correo, radiogramas, servicio telefónico, fax, Internet.

TABLA 10: DESCRIPCIÓN DE LOS GASTOS DE SERVICIOS BÁSICOS

c) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

Partida	Personal	Lugar	Nº de viajes	Costo unitario*	Costo total
22100	Pasajes				
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
Total					
Total sub grupo 22000					0

* En el caso de pasajes debe indicarse el costo de ida y vuelta (costo unitario), indicando el número de viajes.

* En el caso de los viáticos, debe considerarse la escala establecida por la UAJMS.

TABLA 11: DESCRIPCIÓN DE LOS GASTOS DE VIAJES Y TRANSPORTE DE PERSONAL

d) SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria			
23300	Alquiler de Tierras y Terrenos			
Total				0

* Se refiere principalmente a los gastos por el uso de edificios, equipos y maquinaria en general

TABLA 12: DESCRIPCIÓN DE LOS GASTOS POR CONCEPTO DE ALQUILERES DE EQUIPOS Y MAQUINARIAS

e) SUB GRUPO 24000. Descripción mantenimiento y reparación

Partida	Mantenimiento y reparación de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
24100	Mantenimiento y Reparación de Edificios y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación			
Total				0

* Se refiere principalmente a los gastos por el mantenimiento y reparación de edificios, equipos y maquinaria en general

TABLA 13: DESCRIPCIÓN DE MANTENIMIENTO Y REPARACIÓN

f) SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales

Partida	Tipo de servicio profesional y comercial *	Cantidad	Costo unitario	Tiempo mes	Costo total
25200	Estudios e Investigaciones				
25500	Publicidad				
25600	Imprenta	1000	2		2,000
25700	Capacitación de Personal				2,000
25800	Estudios e Investigaciones Para Proyectos de Inversión				
Total					4,000

* Se refiere a gastos por servicios profesionales de asesoramiento especializado, se incluyen estudios, investigaciones, publicidad, imprenta, fotocopias, capacitación de personal y otros ejecutados por terceros.

TABLA 14: DESCRIPCIÓN DE LOS GASTOS EN SERVICIOS PROFESIONALES Y COMERCIALES

3) GRUPO 30000. MATERIALES Y SUMINISTROS

g) SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
31110	Refrigerios y Gastos Administrativos			
31200	Alimento para Animales			
31300	Productos Agroforestales y Pecuarios			
Total				0

* Se refiere a la adquisición de materiales y bienes como: alimentos y productos agroforestales, alimentos y bebidas para personas (indicar el total de refrigerios), alimentos para animales, productos pecuarios.

TABLA 15: DESCRIPCIÓN DE LOS GASTOS ALIMENTOS Y PRODUCTOS AGROFORESTALES

h) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	5 resmas	50	250
32200	Productos de Artes Gráficas, Papel y Cartón	1 banner	50	50
32300	Libros y Revistas			
32400	Textos de Enseñanza			
32500	Periódicos			
Total				300

* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases, impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

TABLA 16: DESCRIPCIÓN DEL GASTO DE PRODUCTOS, PAPEL, CARTÓN E IMPRESOS

i) SUB GRUPO 33000. Descripción del gasto en textiles y vestuario

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
33100	Hilados y Telas			
33200	Confecciones Textiles			
33300	Prendas de vestir			
33400	Calzados			
Total				0

* Se refiere principalmente a los gastos por vestuario uniformes y ropa de trabajo

TABLA 17: DESCRIPCIÓN DEL GASTO EN TEXTILES Y VESTUARIO

j) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total
34110	Combustibles y Lubricantes para Consumo			
34200	Productos químicos y Farmacéuticos			
34400	Productos de Cuero y Caucho			
34500	Productos de Minerales no Metálicos y Plásticos			
34600	Productos Metálicos			
34700	Minerales			
34800	Herramientas Menores			
Total				0

* Se refiere a gastos de combustibles, químicos, productos farmacéuticos, llantas, etc.

TABLA 18: COMBUSTIBLES, PRODUCTOS QUÍMICOS, FARMACÉUTICOS Y OTROS

k) SUB GRUPO 39000. Descripción del gasto en productos varios

Partida	Productos de cuero y caucho	Cantidad	Costo/Unitario	Total
39100	Material de Limpieza			
39400	Instrumental Menor Médico - Quirúrgico			
39500	Útiles de Escritorio y de Oficina			
39700	Útiles y Materiales Eléctricos			
39800	Otros Repuestos y Accesorios			
Total				0

*Se refiere principalmente a los gastos por productos de limpieza, todo lo referente a la funcionamiento de la oficina en material de escritorio.

TABLA 19: DESCRIPCIÓN DEL GASTO EN PRODUCTOS VARIOS

4) GRUPO 40000. ACTIVOS REALES

l) SUB GRUPO 43000. Descripción del gasto de Maquinaria y Equipo

Partida	Tipos de productos	Cantidad	Costo/Unitario	Total
43100	Equipo de Oficina y Muebles			
43200	Maquinaria y Equipo de Producción			
43300	Equipos de Transporte, Tracción y Elevación			
43400	Equipo Médico y de Laboratorio			
43600	Equipo Educacional y Recreativo			
43700	Otra Maquinaria y Equipo			
Total				0

* Se refiere principalmente a los gastos por muebles y enseres, equipo de oficina, comunicación, equipamiento.

TABLA 20: DESCRIPCIÓN DEL GASTO DE MAQUINARIA Y EQUIPO

m) SUB GRUPO 46000. Descripción de estudios y proyectos para inversión

Partida	Productos textiles y vestuarios	Cantidad	Costo/Unitario	Total
46100	Para Construcción de Bienes de Dominio Privado			
	Total			0

* Se refiere principalmente a los gastos por servicios de terceros para la realización de investigaciones y otras actividades técnico – Profesionales necesarias para la construcción y mejoramiento de bienes.

TABLA 21: DESCRIPCIÓN DE ESTUDIOS Y PROYECTOS PARA INVERSIÓN

n) SUB GRUPO 49000. Descripción del gasto de Otros Activos

Partida	Tipos de productos *	Cantidad	Costo/Unitario	Total
49100	Activos Intangibles			
49200	Compra de Bienes Muebles Existentes (Usados)			
49300	Semovientes y otros Animales			
49900	Otros Activos			
	Total			0

* Se refiere a los gastos en la compra de software, licencias.

TABLA 22: DESCRIPCIÓN DEL GASTO DE OTROS ACTIVOS

CAPÍTULO II

COMPONENTES

II.1 COMPONENTE 1: “APLICACIÓN ANDROID”

II.1.1 PLAN DE DESARROLLO DEL SOFTWARE

II.1.1.1 INTRODUCCIÓN

El Plan de Desarrollo de esta Aplicación Android, es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto. En esta parte del documento se especifica de forma detallada las características de la aplicación Android.

Para el desarrollo de la aplicación utilizamos la metodología RUP y el lenguaje de modelado UML; en este documento también se definen los alcances y limitaciones de la aplicación propuesta, haciendo énfasis en que este es una aplicación de optimización en el uso del sistema Tariquia de la UAJMS.

Dentro de cada negocio la información con que se pueda contar, servirá para futuras proyecciones y emprendimientos empresariales. Las aplicaciones Android cubren la necesidad del acceso a la información a través de celulares inteligentes de forma eficiente.

Identificada esta necesidad de contar con información precisa, la administración de la universidad, la utilizará como medio para lograr mejor rendimiento del uso en el sistema académico, logrando de esta manera la optimización del uso de la información.

Lo que se propone es desarrollar una aplicación Android para optimizar el uso del sistema Tariquia por parte de los estudiantes de la UAJMS.

II.1.1.2 PROPÓSITO

El propósito del Plan de Desarrollo de Software es proporcionar la información y descripción de las necesidades en cuanto a funciones, restricciones y limitaciones necesarias para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

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

II.1.1.3 ALCANCE DEL DOCUMENTO

En este documento describimos los procedimientos realizados para el desarrollo del proyecto “Optimización en el Uso del Sistema Tariquia en Dispositivos Móviles de los Estudiantes de la Universidad Autónoma Juan Misael Saracho a través de una Aplicación Android” con un componente importante que es la Aplicación Android desarrollada en el que se realizará la obtención de información de los estudiantes activos en el Sistema Tariquia.

II.1.1.4 OBJETIVOS

II.1.1.4.1 OBJETIVO GENERAL

Desarrollar una aplicación Android para el procesamiento de información, a través de una aplicación que se conecte al Sistema Tariquia para obtener la información, que funcione en el Sistema Operativo Android, utilizando la metodología RUP para su desarrollo.

II.1.1.4.2 OBJETIVOS ESPECÍFICOS

- Diseñar una aplicación Android de acuerdo a los requerimientos del cliente que permita mejorar los procesos dentro de la organización.
- Desarrollar Interfaces de fácil manejo para el Usuario.
- Realizar la planeación del proyecto basándonos en el paradigma de desarrollo RUP (Rational Unified Process).
- Recolectar y seleccionar toda la información necesaria para conocer los problemas y necesidades de la universidad.
- Realizar el análisis y diseño, modelando el sistema con UML.

- Proporcionar Reportes rápidos y formales.
- Trabajar utilizando el lenguaje de programación java.
- Trabajar utilizando Android Studio como IDE.

II.1.1.5 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.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

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

3) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

4) Especificación de los Casos de Uso

Para los casos de usos que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento donde se incluye: pre-condiciones, post-condiciones y flujo de eventos.

5) 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.

6) Glosario

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

Usuarios: Son los que interactúan directamente con el sistema.

Estudiante: Persona que requiere servicios de la universidad.

Docente: Persona que brinda servicios de docencia en la universidad.

Materia: Forman parte de una carrera o un plan de estudios, y que se dictan durante el semestre.

Ficha Académica: Registro general que engloba las nota de un estudiante, de acuerdo a su carrera y materias cursadas.

Programación: Detalle de materia, docente y horario asignado al estudiante.

7) Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada 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.

8) Modelo Navegacional de la Aplicación

Este modelo permite visualizar todas las interfaces de la aplicación con su respectivo orden o camino para poder visualizar que pantalla lleva a que pantalla o que opciones se encuentran disponibles en la aplicación.

9) Diseño de Interfaces de Usuario

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 gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto será desecharido en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

10) Modelado de Componentes

El Modelado de Componentes ilustra los componentes de software que se usarán para construir la aplicación.

11) Modelo de Despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar la disposición física de los artefactos software en nodos.

12) 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 llevará

asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

13) Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación de la aplicación Android.

14) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo:
Manual de Usuario.

15) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada iteración.

II.1.1.6 ALCANCE Y LIMITACIONES DEL COMPONENTE

Debido a la falta de una Aplicación Android acorde a las necesidades de los estudiantes de la institución, pensamos en una aplicación que se adapte al Sistema Operativo comúnmente utilizado.

La Aplicación Android, permite al usuario hacer consultas de sus notas, horarios, ver actividades de sus materias y mantener un control de su rendimiento académico.

El sistema presenta una pantalla principal con un mensaje de bienvenida. Este acceso se da por medio de la inserción de un usuario previamente especificado y una clave, los mismos que deben ser validados. Una vez registrado el usuario y después de haberse validado el registro y la contraseña de usuario, se pueden seleccionar para visualizar la información las siguientes actividades:

- Ficha académica
- Rendimiento Académico
- Notas
- Horarios
- Actividades
- Calendario

El producto final o software será diseñado para funcionar en el Sistema Operativo Android, el cual presenta las siguientes características:

Obtención de las notas de materias programadas visualizadas en la ficha académica. Obtención de las notas de las materias que actualmente cursa el estudiante. Acceso al horario de clases del estudiante. Así también acceso al horario de mesas del estudiante. Opción de revisar las actividades o tareas que tiene en las materias que está cursando. Visualizar el rendimiento académico del estudiante. En adición cuenta con la opción de un calendario virtual para poder agregar sus eventos y administrar su tiempo.

Al ser una aplicación dependiente del Sistema Tariquia se mantiene la seguridad desarrollada en el Tariquia, como ser el tiempo de inactividad, la duplicidad de logueo.

Módulos:

- **Ficha Académica**
- **Notas de Materias Programadas**
- **Horarios**
- **Actividades**
- **Gestionar Calendario**

- **Rendimiento Académico**

Limitaciones:

- La aplicación será desarrollado para ejecutarse en Android, controlando el ingreso de usuarios, con un usuario y clave.
- La aplicación solo contempla el acceso a estudiantes.
- La aplicación depende del Sistema Tariquia.

II.1.2 MARCO METODOLÓGICO

II.1.2.1 MODELO DE PROGRAMACIÓN

Se utilizará una programación orientada a objetos, este modelo usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990.

II.1.2.2 LENGUAJES DE PROGRAMACIÓN

Se utilizarán los siguientes lenguajes de programación:

- **XML:** Son las siglas del Lenguaje de Etiquetado Extensible. La expresión se forma a partir del acrónimo de la expresión inglesa eXtensible Markup Language. Se trata también de un lenguaje estándar que posee una Recomendación del World Wide Web.¹ Utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.
- **Java:** Es un lenguaje orientado a objetos que será usado en los controladores del proyecto. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más

¹ Fuente: <http://www.hipertexto.info/documentos/xml.htm>

populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

II.1.2.3 MOTOR DE BASE DE DATOS

- **PostgreSQL:** Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Algunas de sus principales características son: Llaves ajenas o llaves foráneas (foreignkeys), disparadores (triggers), vistas, integridad transaccional, acceso concurrente multiversión (no se bloquean las tablas ni las filas cuando un proceso escribe), capacidad de albergar programas en el servidor en varios lenguajes, herencia de tablas, tipos de datos y operaciones geométricas. Tiene mejor soporte que los proveedores comerciales.²

II.1.2.4 SOFTWARE PARA LA PROGRAMACIÓN Y DISEÑO DEL SISTEMA

- **Android Studio:** Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014. Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

a) Características.-

- Renderización en tiempo real.
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.

² Fuente: <http://es.scribd.com/doc/96414869/Postgresql>

- Soporte para la construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicación para Android Wear.

b) Requisitos del sistema

- 2 GB de RAM (4 GB recomendados)
- 400 MB de espacio en disco
- 1 GB para Android SDK
- Monitor de 1280x800
- Java Development Kit 7

c) Plataformas soportadas

Android Studio está disponible para Windows 2003, Vista, 7, 8 y GNU/Linux, tanto plataformas de 32 como de 64 bits, Linux con GNOME o KDE y 2 GB RAM mínimo y Mac OS X, desde 10.8.5 en adelante.

- **Eclipse Índigo:** Eclipse es un entorno de desarrollo integrado, de código abierto y Multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que

encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar.

a) **Características.**- Eclipse dispone de un Editor de texto con un analizador sintáctico.

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, test, etc., y refactorización.

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

En el proyecto se trabajó con Eclipse añadiendo Plugins como PMD para realizar las pruebas de caja blanca.

- **JSoup:** Es una librería JAVA para trabajar con código HTML en tiempo real. Es usado como Web Scraping (o Scraping en Java). Provee una API muy conveniente para extraer y manipular información, usando lo mejor de DOM, CSS, y métodos Jquery. Esta librería implementa la especificación de HTML5, y analiza gramáticamente el HTML formando la estructura DOM como los navegadores modernos lo hacen.

a) **Características:**

- Limpia y analiza gramáticamente código HTML desde una URL, archivo o cadena.
- Busca y encuentra información, usando un recorrido a través de la estructura del DOM o selectores CSS.
- Manipula elementos HTML, atributos y texto.
- Limpia el contenido enviado por el usuario contra una segura lista de entidades que puede obtener algún beneficio en particular o reconocimiento, para evitar ataques XSS.
- Tiene como salida un pequeño HTML.

Jsoup está diseñado para trabajar con una variedad de HTML; desde uno código antiguo o con validaciones, hasta código HTML que se encuentra sintáctica o estructuralmente incorrecto; Jsoup va a crear un árbol de estructura sensato para analizar gramáticamente.

II.1.2.5 HERRAMIENTAS SOFTWARE PARA EL PROCESO DE MODELADO

A continuación se enuncia el software utilizado para el proceso de modelado, creación del diagrama de clases, casos de uso, base de datos, etc.

- **Enterprise Architect 7.5:** Es una herramienta ComputerAided Software Engineering (CASE) para diseñar y construir sistemas software en el proceso de modelado. Está basado en la especificación de UML 2.1.³ Ofrece trazabilidad completa desde mapas mentales, pasando por los requerimientos y hasta el diseño y la distribución del software, con el nivel de eficiencia, robustez, herramientas de colaboración y seguridad requerida para sacar adelante proyectos altamente demandantes y de cualquier tamaño.

Enterprise Architect es una aplicación completa para la elaboración de proyectos de Ingeniería. Está diseñado especialmente para el enfoque empresarial, y junto a ello, se especializa en la realización de diagramas UML de todo tipo: Componentes, Paquetes, Clases y Bases de Datos. Soporta el trabajo sobre varios lenguajes de programación como Java, C++, PHP, C#, entre otros.

a) Características.-

- Completa herramienta de análisis y diseño de UML.
- Modelado avanzado para negocios, software y sistemas.
- Completa trazabilidad desde los requerimientos hasta la distribución.
- Ingeniería de código en más de diez lenguajes.
- Altamente escalable, repositorios basados en el equipo de trabajo.

³ Fuente: http://es.wikipedia.org/wiki/Herramienta_CASE

- Mapas mentales, BPMN, arquitectura empresarial, etc.
- **Dia:** Dia es una aplicación informática de propósito general para la creación de diagramas, desarrollada como parte del proyecto GNOME. Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

Dia está diseñado como un sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas. Actualmente se incluyen diagramas entidad-relación, diagramas UML, diagramas de flujo, diagramas de redes, diagramas de circuitos eléctricos, etc. Nuevas formas pueden ser fácilmente agregadas, dibujándolas con un subconjunto de SVG e incluyéndolas en un archivo XML.

El formato para leer y almacenar gráficos es XML (comprimido con gzip, para ahorrar espacio). Puede producir salida en los formatos EPS, SVG y PNG.

También conviene recordar que Dia, gracias al paquete dia2code, puede generar el esqueleto del código a escribir, si utilizáramos con tal fin un UML.

II.1.2.6 WEB SCRAPING

Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.

El web scraping está muy relacionado con la indexación de la web, la cual indexa la información de la web utilizando un robot y es una técnica universal adoptada por la mayoría de los motores de búsqueda. Sin embargo, el web scraping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. El término web scraping también está relacionado con la automatización de tareas en la Web, la cual simula la navegación de un humano utilizando un software de computadora. Alguno de los usos del web scraping son la comparación de precios

en tiendas, la monitorización de datos relacionados con el clima de cierta región, la detección de cambios en sitios webs y la integración de datos en sitios webs.

Técnicas

Web scraping es el proceso de recopilar información de forma automática de la Web. Es un campo con desarrollos activos, compartiendo un propósito en común con la visión de la Web semántica. Utiliza soluciones prácticas basadas en tecnologías existentes que son comúnmente ad hoc. Existen distintos niveles de automatización que las existentes tecnologías de Web Scraping pueden brindar:

- Copiar y pegar humano: algunas veces incluso las mejores técnicas de web scraping no pueden reemplazar la examinación manual de un humano, y a veces esta puede ser la única vía de solución cuando el sitio que tenemos en mente pone ciertas barreras para prevenir que se cree software para realizar tareas automáticas en este.
- Uso de expresiones regulares: una posible vía para extraer información de páginas webs pueden ser las expresiones regulares, aunque comúnmente no se recomienda utilizarlas para parsear el formato HTML.
- Protocolo HTTP: páginas webs estáticas y dinámicas pueden ser obtenidas haciendo peticiones HTTP al servidor remoto utilizando sockets, etc.
- Algoritmos de minería de datos: muchos sitios webs tienen grandes colecciones de páginas generadas dinámicamente a partir de una base de datos. Datos de la misma categoría aparecen usualmente en páginas similares mediante un script o una plantilla. En la minería de datos, un programa detecta estas plantillas en un contexto específico y extrae su contenido.
- Parsers de HTML: Algunos lenguajes, como XQuery y HTQL pueden ser utilizados para parsear documentos, recuperar y transformar el contenido de documentos HTML.
- Aplicaciones para web scraping: existen muchas aplicaciones disponibles que pueden ser utilizadas para personalizar soluciones de Web Scraping. Estas aplicaciones pudieran

reconocer automáticamente la estructura de cierta página o brindar una interfaz al usuario donde este pudiera seleccionar los campos que son de interés dentro del documento. De esta forma no es necesario escribir manualmente código para realizar estas tareas.

- Reconocimiento de información semántica: las páginas que son analizadas podrían incluir metadatos o cierta información semántica como anotaciones o comentarios, los cuales pueden ser usados comúnmente. Si estas anotaciones están en las mismas páginas, como sucede con los microformatos, estas podrían ser de utilidad cuando parseamos el DOM del documento. En otro caso, las anotaciones, organizadas en una capa semántica, son almacenadas y manejadas de forma separada desde otras páginas, por lo que los scrapers pueden recuperar estos esquemas y las instrucciones desde esta capa antes de analizar los documentos.

Medidas para detener a los scrapers

El administrador de un sitio web puede utilizar varias técnicas para detener o disminuir los pedidos de los scrapers. Algunas técnicas incluyen:

- Añadir entradas al fichero robots.txt. Google y otros bots pueden ser detenidos de esta forma.
- Bloquear la dirección IP. Esto también bloqueará todos los accesos desde esa misma IP, por lo que los usuarios no podrán navegar por el sitio web si acceden desde ésta.
- Deshabilitar cualquier interfaz de programación de aplicaciones que el sitio web pudiera estar brindando.
- Los bots o scrapers algunas veces declaran quienes son, y gracias a esto pueden ser bloqueados. Googlebot es un ejemplo. Algunos scrapers no hacen distinción entre ellos y un navegador común.
- Monitorear el exceso de tráfico proveniente de cierta IP.

- Añadir un captcha u otro sistema de verificación manual al sitio web. No se garantiza el completo bloqueo de los scrapers, pero mediante esta técnica se dificulta el acceso de los mismos a los sitios webs.
- Servicios comerciales antibots: algunas empresas ofrecen servicios antibots y antiscraping.
- Incrementar el uso de JavaScript y AJAX. De esta forma es más difícil para los scrapers simular las peticiones como si fueran un navegador común.

II.1.2.6 SISTEMAS OPERATIVOS DE MÓVILES

Un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil al igual que las computadoras utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Características básicas

- **Kernel**

El núcleo o kernel proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece distintos servicios a las superiores capas como son los controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.

- **Middleware**

El middleware es el conjunto de módulos que hacen posible la propia existencia de aplicaciones para móviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajería y comunicaciones, códigos multimedia, intérpretes de páginas web, gestión del dispositivo y seguridad.

- **Entorno de ejecución de aplicaciones**

El entorno de ejecución de aplicaciones consiste en un gestor de aplicaciones y un conjunto de interfaces programables abiertas y programables por parte de los desarrolladores para facilitar la creación de software.

- **Interfaz de usuario**

Las interfaces de usuario facilitan la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas, etc.) y el del marco de interacción.

II.1.2.6.1 SISTEMAS OPERATIVOS MÁS UTILIZADOS

- **ANDROID:** Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets o teléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS.

La versión básica de Android es conocida como Android Open Source Project (AOSP).

El sistema operativo Android se usa en teléfonos inteligentes, ordenadores portátiles, netbooks, tabletas, Google TV, relojes de pulsera, auriculares y otros dispositivos, siendo este sistema operativo accesible desde terminales de menos de \$100 hasta terminales que superen los \$1000.

La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para x86 en el proyecto Android-x86, y Google TV utiliza una versión especial de Android x86.

Android es considerado como uno de los modelos de negocio más exitosos, pues su desarrollo estratégico contempla los factores que más se tienen en cuenta dentro de las herramientas y metodologías desarrollados por expertos en negocios. Este sistema operativo se ha convertido en un modelo a seguir por desarrolladores de tendencias y negocios de alto impacto.

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas todavía no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

En un principio, Android era eminentemente un sistema operativo pensado para usar con teclado, y gracias a un cursor poder navegar entre las aplicaciones. Desde su comienzo, Android ha sido altamente personalizable. Poco después, antes del lanzamiento del primer teléfono Android, esta filosofía cambió para convertirse en eminentemente táctil, y poder competir contra el iPhone, presentado 1 año y 9 meses antes.

- **iOS:** (Anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone siendo después usado en el iPod Touch y en el iPad. No permite la instalación de iOS en hardware de terceros. Tenía el 26% de cuota de mercado de sistemas operativos móviles vendidos en el último cuatrimestre de 2010, detrás de Android y Windows Phone. Actualmente su sistema operativo se encuentra en

la novena versión, mejor conocida como iOS 9. Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee una interfaz fluida. La interacción con el sistema operativo incluye gestos como deslices, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal). iOS se deriva de OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Tipo Unix. iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de "Servicios Principales", la capa de "Medios" y la capa de "Cocoa Touch".

- **MAEMO:** Basado en Linux Debían (casi todo en código abierto) y desarrollado por Nokia para Smartphone y tablets.
- **PALM WEB:** Aprovecha tecnología web como XHTML, Java Script Y CSS pertenece a HP.
- **RIM:** Este sistema operativo ha sido pionero en la gestión del correo electrónico y está muy orientado a un uso profesional.
- **SYMBIAN:** Es el sistema más extendido, nacido de la alianza de varias compañías de móviles. El actual propietario es Nokia.
- **WINDOWS:** Windows Phone (abreviado WP) es un sistema operativo móvil desarrollado por Microsoft, como sucesor de Windows Mobile. A diferencia de su predecesor está enfocado en el mercado de consumo en lugar de en el mercado empresarial. Con Windows Phone; Microsoft ofrece una nueva interfaz de usuario que integra varios de sus servicios propios como OneDrive, Skype y Xbox Live en el sistema operativo. Compite directamente contra Android de Google e iOS de Apple. Su última versión disponible y definitiva es Windows Phone 8.1, lanzado el 14 de abril de 2014. La mayor ventaja de este sistema es la de cualquier producto de Microsoft compatibilidad, facilidad de uso, integración con los sistemas Windows

II.1.2.7 METODOLOGÍA PARA EL ANÁLISIS

II.1.2.7.1 NIVEL DE INVESTIGACIÓN

El nivel de la investigación es descriptivo ya que analizaremos como objeto de estudio el sistema actual de información académica, señalando sus características y propiedades e identificando sus ventajas y desventajas. Además se indagará sobre la aceptación del proyecto propuesto.

II.1.2.7.2 POBLACIÓN Y MUESTRA

De los más de 17 mil estudiantes, de la Universidad Autónoma Juan Misael Saracho, la población tomada en cuenta será determinada por un muestreo estratificado de los estudiantes de la Ciudad de Tarija, de los cuales se obtendrán los datos necesarios para realizar nuestras encuestas.

Para el cálculo del tamaño de la muestra utilizaremos la siguiente fórmula:

$$n = \frac{z^2 * N * p * q}{e^2(N - 1) + z^2 * p * q}$$

Donde:

n = Tamaño de la muestra.

N = Tamaño de la población.

p y q = Son valores de la Desviación Estándar de la población, que generalmente cuando no se tienen su valor, suele utilizarse un valor constante de 0.5, ya que la distribución normal es una curva simétrica, y el valor de 0.5 es tomada por que es en ese punto donde se centra el valor de la media.

Z = Nivel de confianza, valor constante que si no tiene su valor, se lo toma en relación al 95% de confianza que equivale a 1.96 o en relación al 99% de confianza que equivale a 2.58, valor que queda a criterio del investigador.

Para nuestra investigación tomaremos el nivel de confianza del 95% ya que vamos a trabajar con los datos obtenidos de las opiniones de las personas y estas no siempre responden con seriedad las preguntas.

e = Error muestral, que generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% y 5%, valor que queda a criterio del investigador. Para nuestro caso tomaremos el valor de 5% de error y esto depende del valor del nivel de confianza.

Calculando el tamaño de la muestra de una población de 17843 de la universidad con un nivel de confianza del 95% se tiene.

Para calcular el tamaño de la muestra tomamos los siguientes valores:

$$N = 17843$$

$$Z = 1,96$$

$$p = 0,5$$

$$q = 0,5$$

$$e = 5\% = 0,05$$

Fórmula

$$n = \frac{1,96^2 * 17843 * 0,5 * 0,5}{0,05^2(17843 - 1) + 1,96^2 * 0,5 * 0,5}$$

Que dan como resultado:

n = 376 estudiantes.

Por lo tanto tendremos una muestra de 376 estudiantes a los cuales se les realizará las encuestas.

II.1.2.7.3 TÉCNICAS DE RECOLECCIÓN DE DATOS

La recaudación de datos será a través de las encuestas, y para ello haremos uso de herramientas tradicionales y además haremos uso a la tecnología para realizar las tabulaciones respectivas y

obtener los gráficos para una mejor interpretación, con la Herramienta SPSS. Las encuestas estarán basadas en tipo de S.O. del dispositivo móvil que maneja y la interacción con el Sistema Tariquia.

II.1.3 MARCO TEÓRICO

II.1.3.1 MODELO VISTA CONTROLADOR

II.1.3.1.1 INTRODUCCIÓN

El Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), 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.

II.1.3.1.2 DESCRIPCIÓN DEL PATRÓN

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. El modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca peticiones al modelo y probablemente a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos,

algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo el usuario hace clic en un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

II.1.3.2 METODOLOGÍA DE DESARROLLO

II.1.3.2.1 METODOLOGÍA DE DESARROLLO UNIFICADO DE SOFTWARE (RUP)

Es de suma importancia elegir la metodología adecuada así como las herramientas de implementación adecuadas, es por ello que la metodología RUP basada en UML nos proporciona todas las bases para llevar al éxito la elaboración del software.

RUP 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.

Rational Unified Process (Proceso Unificado de Rational), metodología del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios.

II.1.3.2.2 UML – LENGUAJE DE MODELADO UNIFICADO

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está respaldado por el OMG (Object Management Group).⁴

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

UML ofrece un estándar para describir un “plano” del sistema (modelo) incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

⁴ Fuente: http://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

II.1.3.2.2.1 OBJETIVOS DE UML

UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

II.1.3.2.2.2 DIAGRAMAS DE UML

En UML existen dos tipos de diagramas: los diagramas que nos dan una vista estática y dinámica.

II.1.3.2.2.2.1 DIAGRAMAS ESTÁTICOS

Estos diagramas nos describen el comportamiento del sistema en el tiempo.

- **Diagramas de Casos de Uso:** Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.
- **Diagrama de Clases:** El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clases incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.
- **Diagrama de Componentes:** Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las

relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

- **Diagramas de Despliegue:** Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución de instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo.

II.1.3.2.2.2 DIAGRAMAS DINÁMICOS

- **Diagramas de Secuencia:** Diagrama que muestra las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Representa una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente por orden temporal. A diferencia de los diagramas de colaboración, los diagramas de secuencia incluyen secuencia temporales pero no incluyen las relaciones entre objetos. Pueden existir de forma de descriptor (describiendo todos los posibles escenarios) y en forma de instancia (describiendo un escenario real).

- **Diagramas de Colaboración:** Diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de colaboración no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes.
- **Diagramas de Actividades:** Un estado de actividad representa una actividad, un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente

actividad. Un diagrama de actividad es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

- **Diagramas de Estados:** Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro. Los diagramas de estados representan autómatas de estados finitos desde el punto de vista de los estados y las transiciones. Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El estado está caracterizado parcialmente por los valores, algunos de los atributos del objeto. El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase.
- **Diagrama de Paquetes:** Cualquier sistema grande se debe dividir en unidades más pequeñas de modo que las personas puedan trabajar con una cantidad de información limitada, a la vez y de modo que los equipos de trabajo no interfieran con el trabajo de los otros.

Un paquete es una parte de un modelo, cada parte del modelo debe permanecer en un paquete. Pero para ser funcional, la asignación debe seguir un cierto principio racional, tal como funcionalidad común, implementación relacionada y punto de vista común. UML no impone una regla para componer los paquetes.

II.1.3.3 ANTECEDENTES DEL PROYECTO

Desde el surgimiento de la organización, la función esencial de la información en las organizaciones se ha acentuado. Una organización es un sistema compuesto por tres elementos: personas, recursos e información. Los sistemas de información, por su parte, surgen como sistemas complejos, abiertos que interactúan con otros sistemas y subsistemas como parte de su actuación.

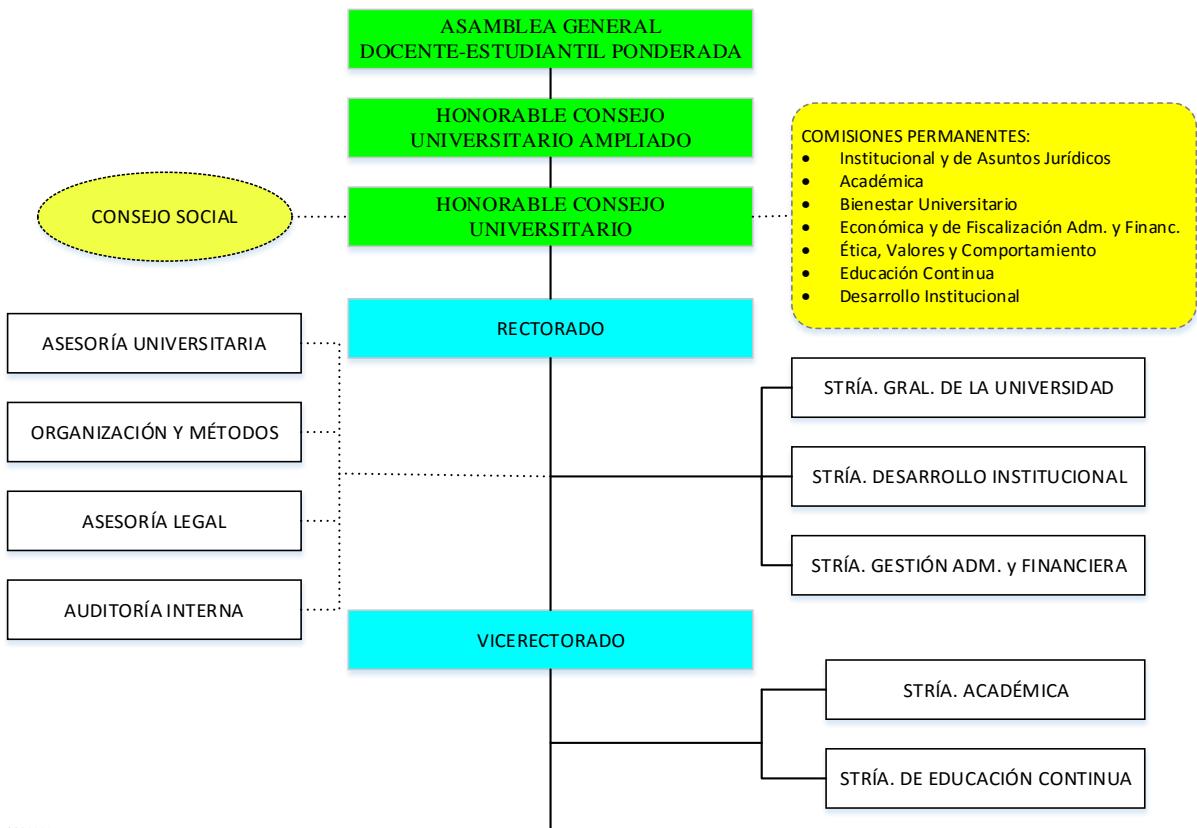
El uso de ciertos conceptos tomados de la teoría de sistemas y del campo de la informática llevó a un alto grado de desarrollo entre los sistemas de información. Aunque existen diversas

definiciones, hechas desde diferentes enfoques, sobre los sistemas de información, en su gran mayoría tienen puntos en común.

Una aplicación móvil exige la compatibilidad con el Sistema Operativo en el que será ejecutado, es por ello que de acuerdo al gran porcentaje de estudiantes que cuentan con dispositivos móviles que cuentan con Android, se optó por desarrollar para este sistema una aplicación que facilite el acceso al Sistema Tariquia Vigente.

II.1.3.4 ESTRUCTURA ORGANIZATIVA DE LA UAJMS

Nivel Universitario Institucional



Nivel Universitario Facultativo

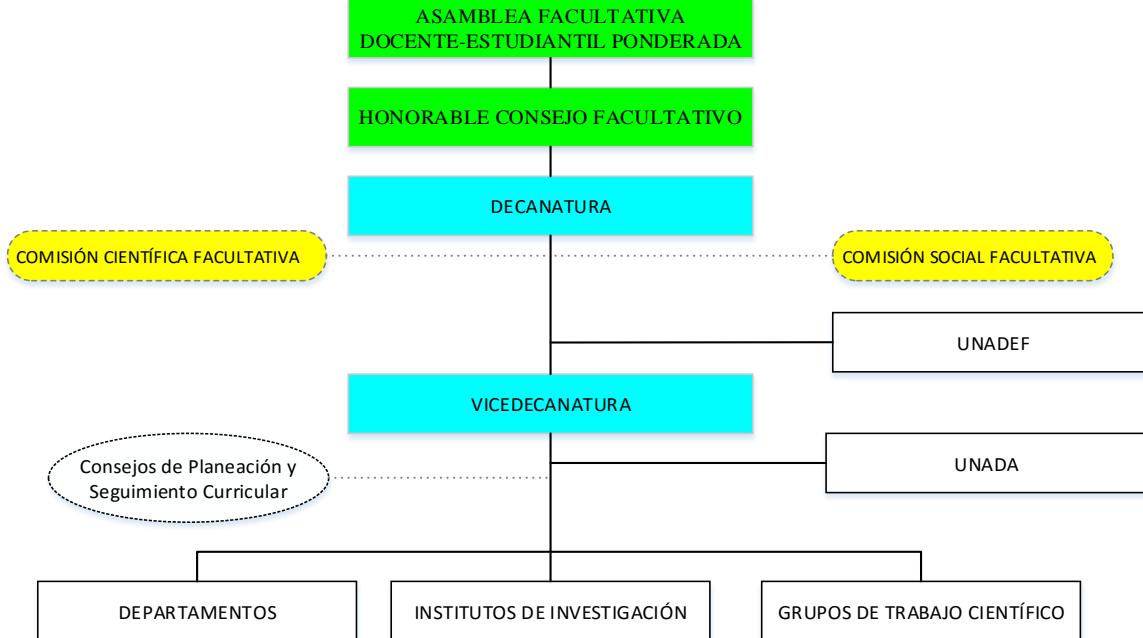


FIGURA 5: ESTRUCTURA ORGANIZATIVA

II.1.3.5 ESTRUCTURA ADMINISTRATIVA DE LA UAJMS

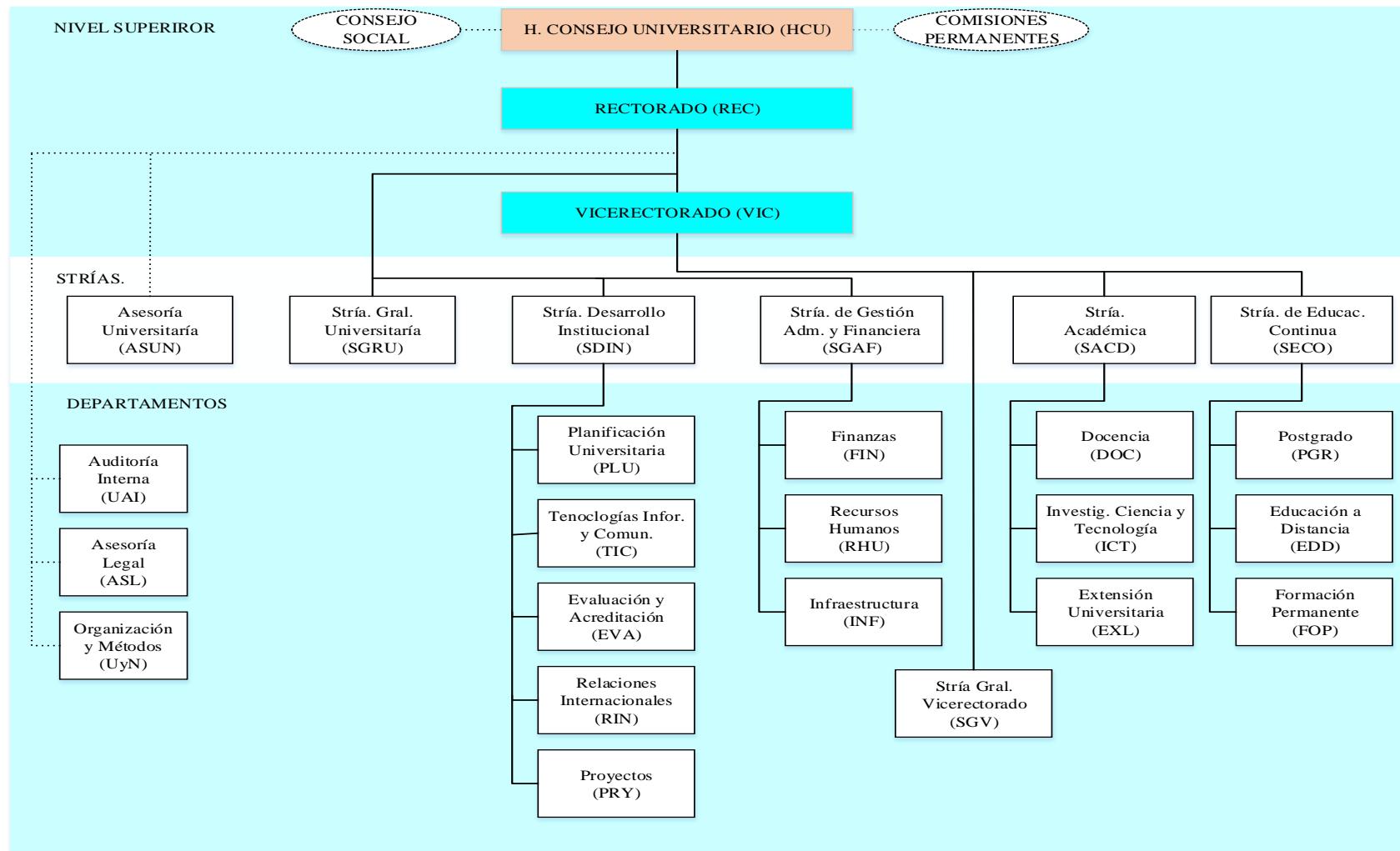


FIGURA 6: ESTRUCTURA ADMINISTRATIVA

II.1.3.6 ESTRUCTURA ACADÉMICA DE LA UAJMS

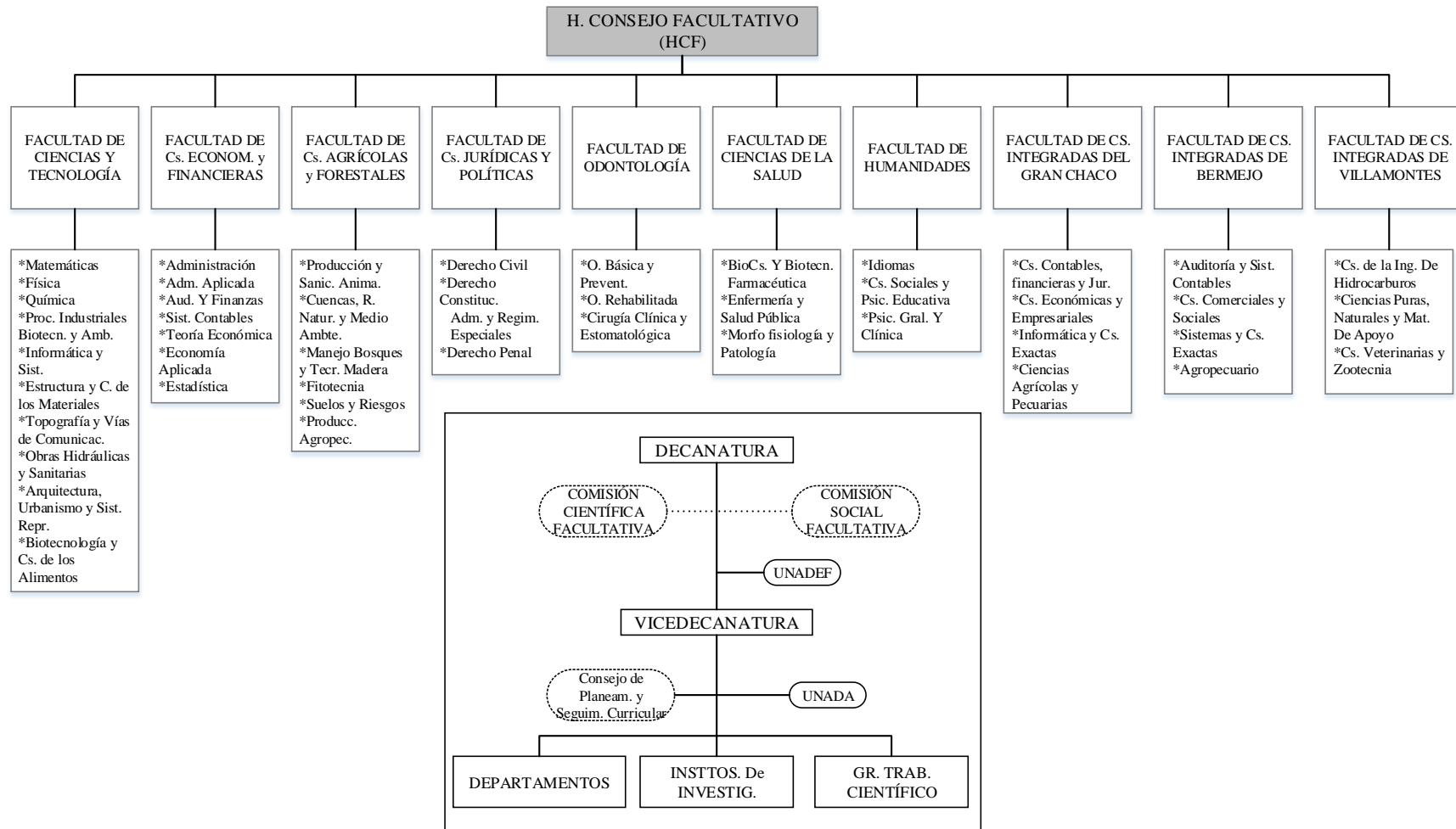


FIGURA 7: ESTRUCTURA ACADÉMICA

II.1.4 ANÁLISIS DE REQUERIMIENTOS

II.1.4.1 INTRODUCCIÓN

La etapa de Análisis de Requerimientos es la primera etapa en el desarrollo de un Sistema Informático. En esta etapa debemos responder a una pregunta fundamental: ¿Qué es lo que quiere el cliente? Y para ello, debemos diagnosticar la situación actual, recopilar los requerimientos del cliente, tanto en relación al sistema como generales respecto del área informática; es decir la situación ideal, para así poder definir alternativas de solución, según las cuales podremos avanzar desde lo que hoy se posee, hacia el punto que se pretende llegar.

Como parte de nuestro trabajo, debemos señalar cuál de las alternativas es la más conveniente y justificarlo en la propuesta.

La “Ingeniería de Requerimientos” se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto determinado. El uso del término “ingeniería” implica que se deben utilizar técnicas sistemáticas y repetibles para asegurar que los requerimientos del sistema estén completos y sean consistentes y relevantes.

Los requerimientos son la pieza fundamental en un proyecto de desarrollo de software, en ellos se basan muchos participantes del proyecto para:

- Planear el proyecto y los recursos que se usarán en él. Se utilizarán los requerimientos como una base para la estimación del esfuerzo necesario en un proyecto.
- Especificar el tipo de verificaciones que habrán de realizar al sistema. Por ejemplo: cuando se está tratando de alinear a cierta norma oficial o estándar.
- Planear la estrategia de prueba a la que habrá de ser sometido el sistema. Los requerimientos son la base sobre la cual se decide si un caso de prueba fue ejecutado exitosamente por el sistema o no.

Son el fundamento del ciclo de vida del proyecto. Los requerimientos documentados son la base para crear la documentación del sistema. De ahí su importancia y la importancia por la que deban de ser definidos y manejados de la forma más adecuada posible.

II.1.4.2 MODELO DE REQUISITOS

II.1.4.2.1 REQUERIMIENTOS FUNCIONALES

- Autenticación del Estudiante, con usuario y clave.
- Visualizar Ficha Académica
- Visualizar las Notas de las materias programadas.
- Visualizar Horarios de Materias cursadas y de las Mesas respectivas.
- Acceder a las actividades o tareas de cada materia.
- Generar gráfica de rendimiento académico.

II.1.4.2.2 REQUERIMIENTOS NO FUNCIONALES

- Dispositivo Móvil (RAM 512).
- Sistema Operativo Android (Min. SDK API 16).
- Conexión estable a Internet.
- Base de datos PostgreSQL.
- Java

Modelo de Casos de Uso del Negocio

II.1.5 MODELO DE LOS CASOS DE USO DEL NEGOCIO

II.1.5.1 INTRODUCCIÓN

El Modelo de Casos de Uso del Negocio es un artefacto que brinda el conocimiento sobre el funcionamiento del negocio o empresa.

II.1.5.2 PROPÓSITO

- Comprender la estructura y la dinámica de la Universidad Autónoma Juan Misael Saracho.
- Comprender los problemas actuales e identificar posibles mejoras.
- Comprender los procesos del negocio de la Universidad Autónoma Juan Misael Saracho.

II.1.5.3 ALCANCE

- Describir los procesos de negocio y los usuarios.
- Identificar y definir los procesos del negocio según los objetivos de la organización.
- Definir un modelo del negocio que pueda mostrar el contexto y sus limitantes.

II.1.5.4 DESCRIPCIÓN DE LOS ACTORES DEL NEGOCIO

Se identificaron los siguientes actores, los cuales interactúan con los procesos principales que se llevan a cabo en la Universidad Autónoma Juan Misael Saracho.

Definición de Actores

- Administrativos
- Autoridades
- Docentes
- Auxiliares de Docentes
- Estudiantes

II.1.5.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

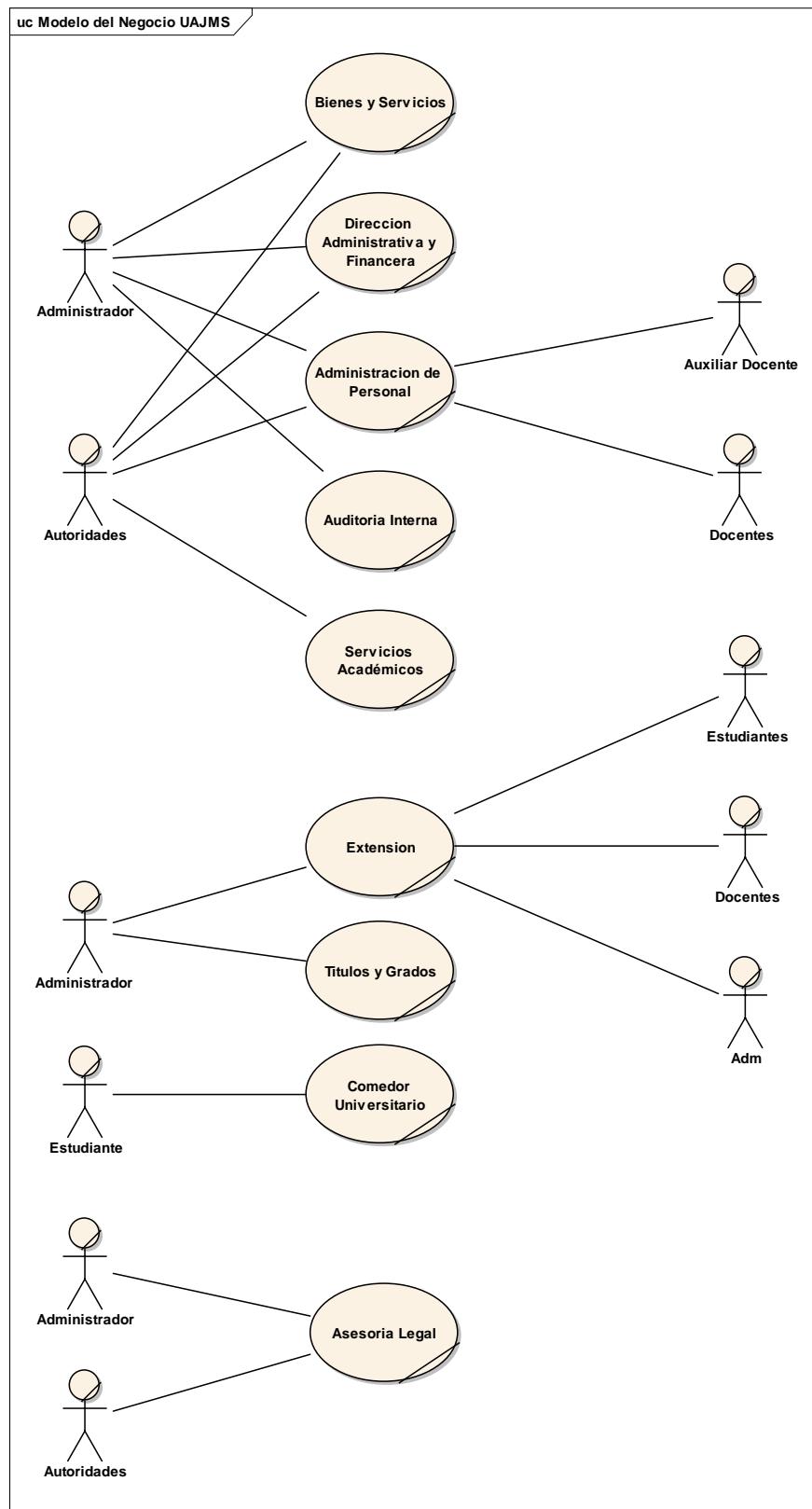


FIGURA 8: DIAGRAMA DE CASOS DE USO DEL NEGOCIO

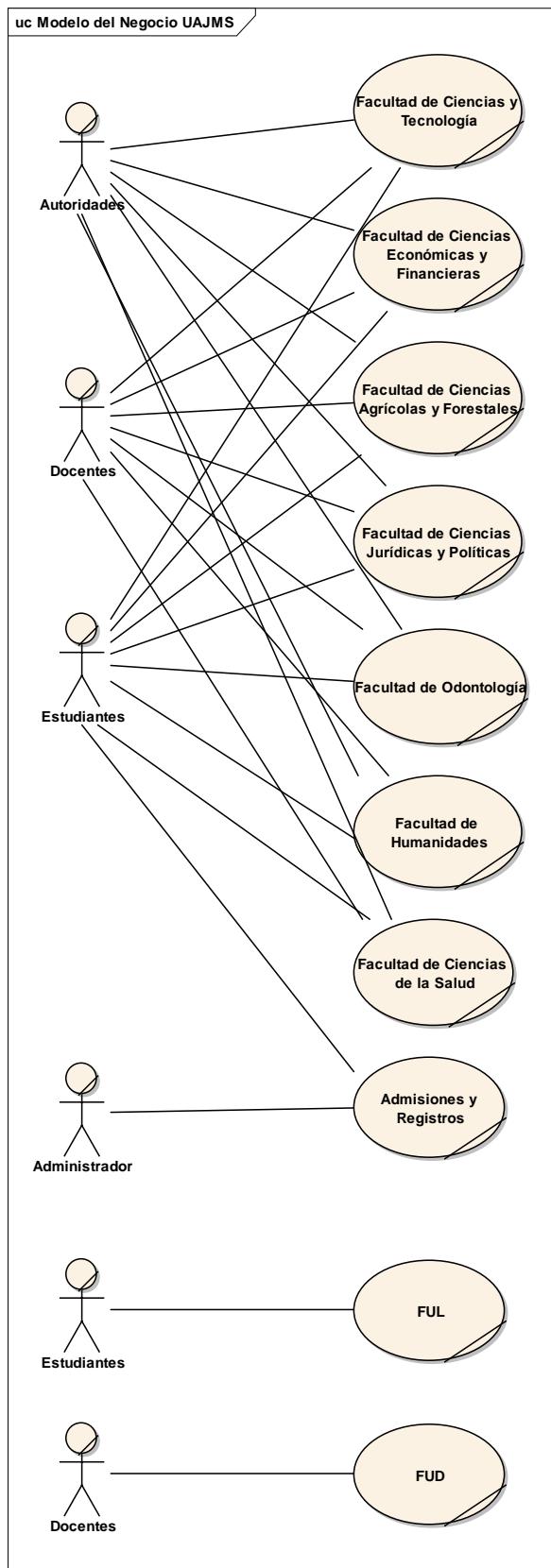


FIGURA 9: DIAGRAMA DE CASOS DE USO DEL NEGOCIO

Modelo de Casos de Uso

II.1.6 MODELO DE CASOS DE USO

II.1.6.1 INTRODUCCIÓN

Los casos de uso nos sirven para representar la manera de cómo el cliente interactúa con el sistema que se desarrollará.

II.1.6.2 PROPÓSITO

- Modelar el contexto de la aplicación.
- Modelar los requerimientos de la aplicación.
- Identificar los procesos de la aplicación.
- Estimular a que los usuarios potenciales hablen de la aplicación desde su propio punto de vista.
- Involucrar a los usuarios en las etapas iniciales del análisis y diseño de la aplicación
- Obtener los requerimientos desde el punto de vista del usuario.

II.1.6.3 ALCANCE

- Describir lo que la aplicación realizará dentro del negocio.
- Describir los alcances de la aplicación.
- Describir los procesos de la aplicación y del cliente.

II.1.6.4 DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN

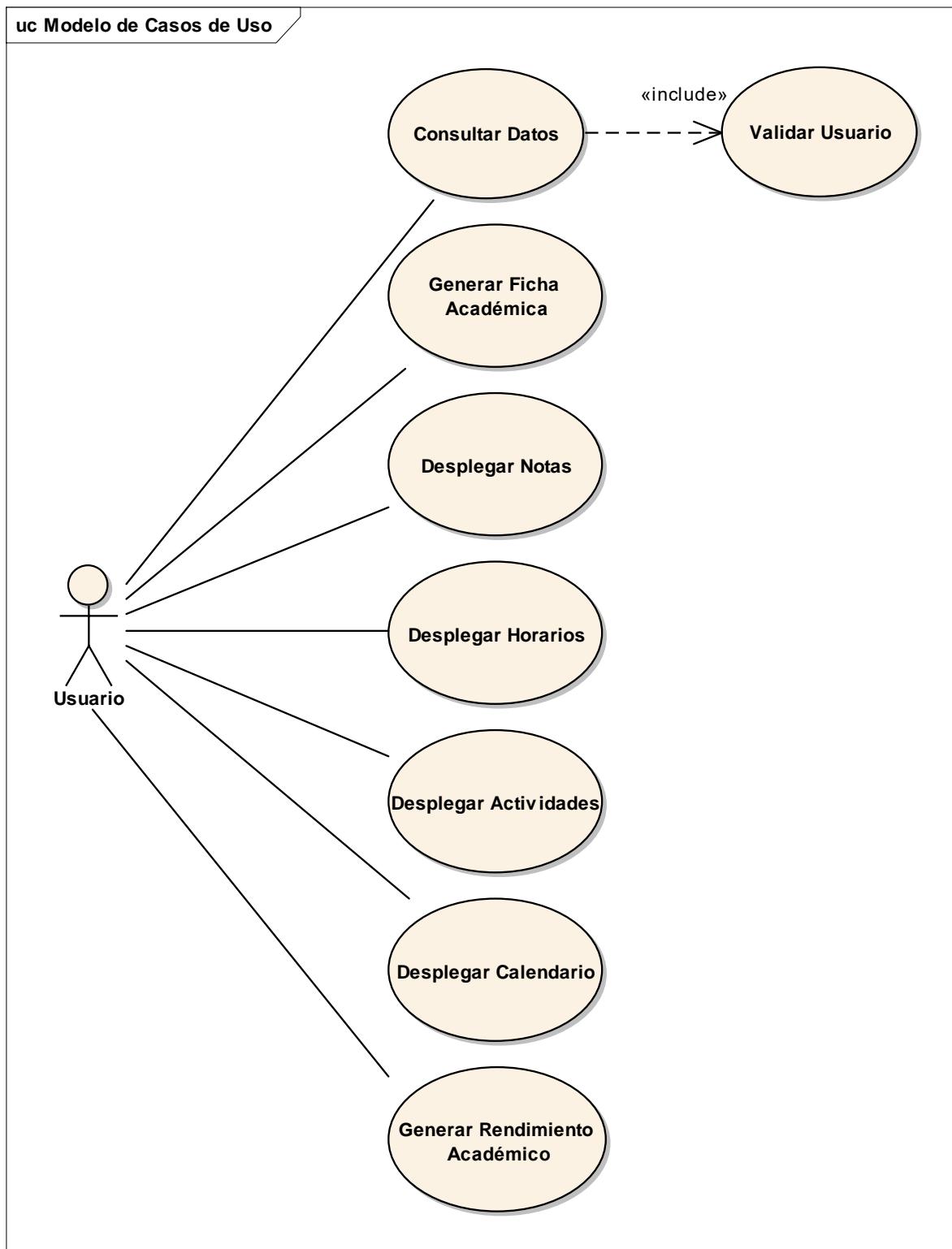


FIGURA 10: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN

II.1.6.4.1 CONSULTAR DATOS

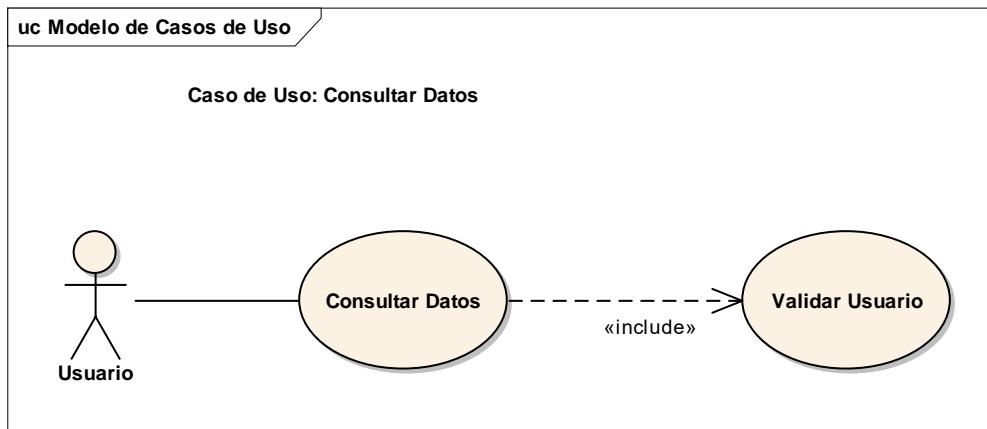


FIGURA 11: DIAGRAMA CONSULTAR DATOS

II.1.6.4.2 GENERAR FICHA ACADÉMICA

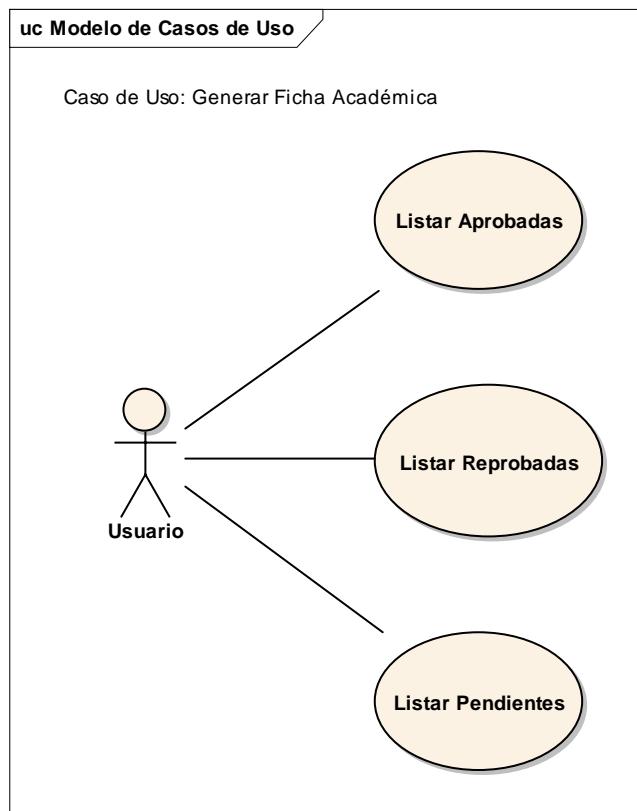


FIGURA 12: DIAGRAMA GENERAR FICHA ACADÉMICA

II.1.6.4.3 DESPLEGAR HORARIOS

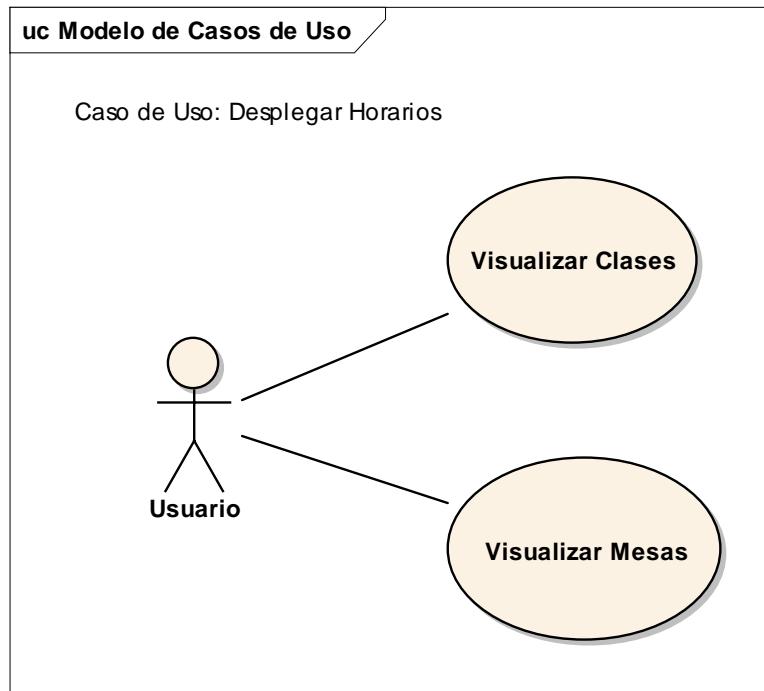


FIGURA 13: DIAGRAMA DESPLEGAR HORARIOS

II.1.6.4.4 GENERAR RENDIMIENTO ACADÉMICO

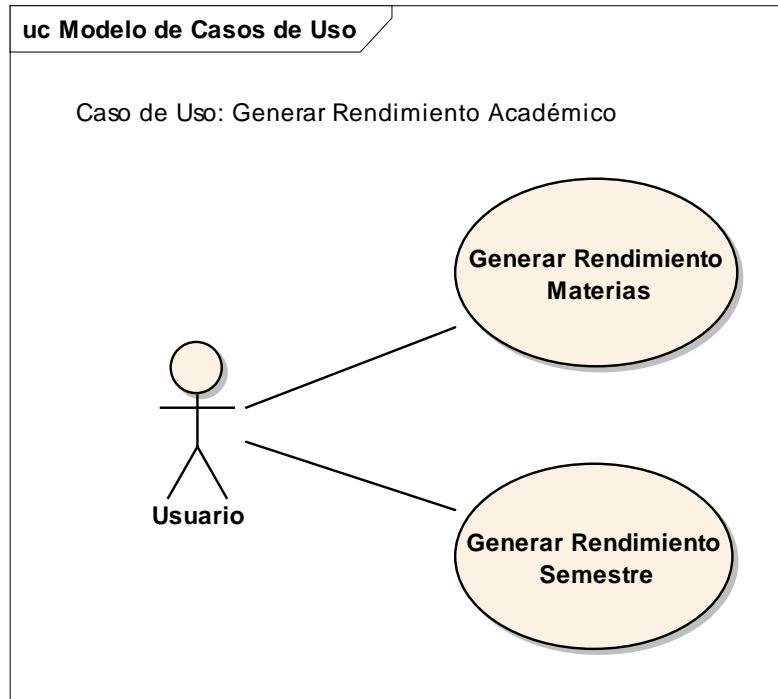


FIGURA 14: DIAGRAMA RENDIMIENTO ACADÉMICO

Especificaciones de Casos de Uso

II.1.7 ESPECIFICACIÓN DE CASOS DE USO

II.1.7.1 INTRODUCCIÓN

Las especificaciones de los Casos de Uso es una descripción detallada de los casos de uso identificados en el análisis.

II.1.7.2 PROPÓSITO

- Comprender el funcionamiento de los Casos de Uso de la Aplicación.
- Describir específicamente cada Caso de Uso.

II.1.7.3 ALCANCE

- Describir los procesos internos de la aplicación en cada Caso de Uso.
- Detallar los flujos de cada Caso de Uso según lo establecido por la organización.

II.1.7.4 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

Se identificó como actores a los estudiantes, los cuales interactúan con el sistema.

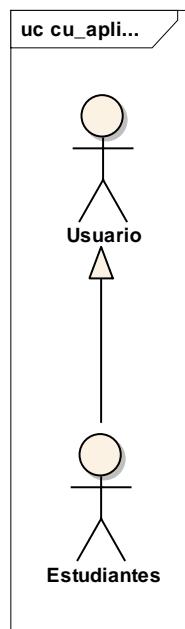


FIGURA 15: ACTORES DEL SISTEMA

Nombre:	Estudiantes
Tipo:	Primario
Descripción:	Una persona que puede visualizar la información de sus materias, notas, horarios

TABLA 23: ESTUDIANTES

II.1.7.5 ESPECIFICACIÓN DE CASOS DE USO

II.1.7.5.1 CONSULTAR DATOS

Descripción Caso de Uso	
Caso de Uso:	Consultar Datos
Actor:	Estudiante
Tipo:	Básico
Propósito:	Permite establecer una sesión válida para el estudiante.
Resumen:	Es una función para el estudiante, de esta manera puede ingresar al sistema y obtener las funciones dentro de éste.
Flujo Principal	
Usuario	Aplicación
1. El estudiante debe proporcionar su usuario y contraseña para establecer una sesión.	2. En caso de que los datos proporcionados sean correctos, la aplicación mostrará una pantalla con las opciones que tiene un estudiante dentro de la aplicación. 3. Caso contrario desplegará un mensaje de usuario y/o contraseña incorrectos.
Subflujo:	Los subflujos son: s-1 Inicio de Sesión Correctamente y muestra pantalla menú.
Excepciones	
Los datos introducidos son incorrectos.	
Usuario	Aplicación
1. El usuario proporciona datos incorrectos.	2. La aplicación mostrará un mensaje de: "Nombre de Usuario y/o contraseña incorrectos".
El estudiante se encuentra conectado en otro navegador.	
Usuario	Aplicación
	2. La aplicación mostrara un mensaje de: "Usted ya inicio sesión en otro navegador"

1. El usuario se encuentra conectado desde un navegador.	
----------------------------------------------------------	--

TABLA 24: CONSULTAR DATOS**II.1.7.5.2 VALIDAR USUARIO**

Descripción Caso de Uso	
Caso de Uso:	Validar Usuario
Actor:	Estudiante
Tipo:	Include
Propósito:	Validar el ingreso al sistema.
Resumen:	El usuario introducirá sus datos Usuario y Contraseña para que sea validado por el sistema
Flujo Principal	
<i>Usuario</i>	<i>Aplicación</i>
1. Ingresa usuario y contraseña, escoge la opción de “Conectar”	2. El sistema valida los datos introducidos. 3. Se ingresa visualizando el menú con las opciones disponibles.
Subflujos:	Ninguno.
Excepciones	
Los datos introducidos son incompletos.	
<i>Usuario</i>	<i>Aplicación</i>
1. El usuario proporciona datos incompletos.	2. La aplicación mostrará un mensaje de: “Debe llenar todos los campos”.
El estudiante ingresa datos erróneos	
<i>Usuario</i>	<i>Aplicación</i>
1. El usuario ingresa datos erróneos	2. La aplicación mostrara un mensaje de: “Nombre de Usuario y/o contraseña incorrectos”

TABLA 25: VALIDAR USUARIO

Una vez que los datos son verificados, la aplicación abre una ventana donde muestra los menús que fueron asignados para el estudiante.

II.1.7.5.3 GENERAR FICHA ACADÉMICA

Descripción Caso de Uso	
Caso de Uso:	Generar Ficha Académica
Actores:	Estudiante
Tipo:	Básico
Propósito:	Permite generar la información de la ficha académica del estudiante.
Resumen:	El estudiante puede visualizar información de su ficha académica como estudiante dentro de la universidad. Tanto materias aprobadas, reprobadas y pendientes.
Precondición:	El estudiante debe haber iniciado sesión previamente.
Postcondición:	El estudiante tiene la posibilidad de ver las materias aprobadas, reprobadas y pendientes.
Flujo Principal:	
<i>Usuario</i>	<i>Sistema</i>
1. El usuario selecciona el botón Ficha Académica.	2. La aplicación carga la página con las opciones de materias Aprobadas, Reprobadas, Pendientes.
Subflujos:	s-1 El estudiante selecciona el botón Aprobadas. A continuación la aplicación listará las materias aprobadas por el estudiante. s-2 El estudiante selecciona el botón Reprobadas. A continuación la aplicación listará las materias reprobadas por el estudiante. En caso de no tener materias reprobadas, se despliega el siguiente mensaje: "No existen materias reprobadas." s-3 El estudiante selecciona el botón Pendientes. A continuación la aplicación listará las materias pendientes del estudiante.
Excepciones	
No hay excepciones.	

TABLA 26: GENERAR FICHA ACADÉMICA

II.1.7.5.3.1 DESPLEGAR NOTAS

Descripción Caso de Uso	
Caso de Uso:	Desplegar Notas
Actores:	Estudiante
Tipo:	Básico
Propósito:	Generar información sobre las notas de las materias del estudiante.
Resumen:	El estudiante puede visualizar las notas de las materias programadas.
Precondición:	El estudiante debe haber iniciado sesión previamente.
Postcondición:	El estudiante puede visualizar las notas de las materias por gestión.
Flujo Principal:	
<i>Usuario</i>	<i>Sistema</i>
1. El usuario selecciona el botón Notas.	2. La aplicación carga la página con las materias que está cursando en el presente año.
3. El usuario selecciona la materia de la cual quiere saber sus notas.	4. La aplicación despliega las notas disponibles de la materia seleccionada.
Subflujos:	s-1 El usuario puede seleccionar la gestión.
Excepciones	
No hay excepciones.	

TABLA 27: DESPLEGAR NOTAS

II.1.7.5.3.2 DESPLEGAR HORARIOS

Descripción Caso de Uso	
Caso de Uso:	Desplegar Horarios
Actores:	Estudiante
Tipo:	Básico
Propósito:	Desplegar información sobre los horarios del estudiante.
Resumen:	El estudiante puede visualizar los horarios de sus clases y de mesas.
Precondición:	El estudiante debe haber iniciado sesión previamente.
Postcondición:	El estudiante puede visualizar el horario tanto de clases como de mesas.
Flujo Principal:	
Usuario	Sistema
1. El usuario selecciona el botón Horarios.	2. La aplicación carga la página con dos opciones: Horarios de Clases y Horarios de Mesas.
3. El usuario selecciona una de las opciones.	4. La aplicación despliega el horario seleccionado.
Subflujos:	s-1 El usuario selecciona el botón Clases, a continuación se carga un formulario para seleccionar las opciones a la hora de desplegar el horario de clases. s-2 El usuario selecciona el botón Mesas, a continuación se carga los horarios de mesa del año seleccionado.
Excepciones	
No hay excepciones.	

TABLA 28: DESPLEGAR HORARIOS

II.1.7.5.3.3 DESPLEGAR ACTIVIDADES

Descripción Caso de Uso		
Caso de Uso:	Desplegar Actividades	
Actores:	Estudiante	
Tipo:	Básico	
Propósito:	Desplegar información sobre las actividades (tareas) de las materias asignadas al estudiante.	
Resumen:	El estudiante puede visualizar las actividades (tareas) de las materias asignadas en las diferentes gestiones. Además de contar con la posibilidad de descargar para revisarlas desde su dispositivo inteligente.	
Precondición:	El estudiante debe haber iniciado sesión previamente.	
Postcondición:	El estudiante puede visualizar las actividades de la actual gestión.	
Flujo Principal:		
<p>Usuario</p> <p>1. El usuario selecciona el botón Actividades.</p> <p>3. El usuario selecciona la materia de la cual quiere saber sus tareas.</p>	<p>Sistema</p> <p>2. La aplicación carga la página con las materias que está cursando.</p> <p>4. La aplicación despliega las actividades disponibles de la materia seleccionada con la opción de descarga.</p>	
Subflujos:	s-1 El usuario puede seleccionar la gestión.	
Excepciones		
No hay excepciones.		

TABLA 29: DESPLEGAR ACTIVIDADES

II.1.7.5.3.4 DESPLEGAR CALENDARIO

Descripción Caso de Uso	
Caso de Uso:	Desplegar Calendario
Actores:	Estudiante
Tipo:	Básico
Propósito:	Gestionar actividades en calendario.
Resumen:	El estudiante puede crear, modificar, borrar y ver eventos en un calendario online.
Precondición:	El estudiante debe haber iniciado sesión previamente.
Postcondición:	El estudiante visualiza un calendario online con eventos que el estudiante puede gestionar.
Flujo Principal:	
<i>Usuario</i>	<i>Sistema</i>
1. El usuario selecciona el botón Calendario.	2. La aplicación carga una página con un calendario online.
3. El usuario puede gestionar eventos dentro de este calendario.	4. La aplicación gestiona los eventos del estudiante.
Subflujos:	No existe.
Excepciones	
No hay excepciones.	

TABLA 30: DESPLEGAR CALENDARIO

II.1.7.5.3.5 GENERAR RENDIMIENTO ACADÉMICO

Descripción Caso de Uso	
Caso de Uso:	Generar Rendimiento Académico
Actores:	Estudiante
Tipo:	Básico
Propósito:	Generar un gráfico de rendimiento académico.
Resumen:	El estudiante puede visualizar gráficos de rendimiento por materias y semestre.
Precondición:	El estudiante debe haber iniciado sesión previamente.
Postcondición:	El estudiante puede visualizar el horario tanto de clases como de mesas.
Flujo Principal:	
<i>Usuario</i>	<i>Sistema</i>
1. El usuario selecciona el botón Rendimiento Académico.	2. La aplicación carga la página con dos opciones: Rendimiento por Materia y Rendimiento por Semestre.
3. El usuario selecciona una de las opciones.	4. La aplicación genera una gráfica de barras
Subflujos:	s-1 El usuario selecciona el botón Materia, a continuación se carga una gráfica de acuerdo a todas las materias cursadas. s-2 El usuario selecciona el botón Semestre, a continuación se carga una gráfica por el rendimiento en promedio de cada semestre.
Excepciones	
No hay excepciones.	

TABLA 31: GENERAR RENDIMIENTO ACADÉMICO

Modelo De Análisis Y Diseño

II.1.8 MODELO DE ANÁLISIS Y DISEÑO

II.1.8.1 MODELADO DE DIAGRAMA DE CLASES

II.1.8.1.1 INTRODUCCIÓN

El modelado de Diagrama de Clases es un artefacto de la disciplina de Análisis/Diseño en la Metodología RUP que se está implementando.

II.1.8.1.2 PROPÓSITO

- Comprender la estructura del sistema, identificar clases de Análisis y Diseño.

II.1.8.1.3 ALCANCE

- Describir las clases y objetos de diseño del sistema, identificar y definir los objetos del sistema según los objetos del sistema deseado.

II.1.8.1.4. DESCRIPCIÓN DEL DIAGRAMA DE CLASES

Clase Usuario:

- Esta clase permite mostrar a todos los usuarios habilitados en el sistema.
- Permite la búsqueda de usuarios en el sistema.
- Permite adicionar, modificar, eliminar lógicamente usuarios del sistema.
- Permite la asignación de usuarios y claves para el ingreso al sistema

Clase Persona:

- Esta clase permite listar a las personas registradas en el sistema.

Clase Docente:

- Esta clase permite listar a los docentes registrados en el sistema.
- Por otro lado permite la adición, modificar, eliminar y mostrar a un docente especificado.

Clase Estudiante:

- Esta clase permite listar a los estudiantes registrados en el sistema.
- Por otro lado permite la adición, modificar, eliminar y mostrar a un estudiante especificado.

Clase Matrícula:

- Esta clase permite listar a las matrículas de los estudiantes.
- Permite administrar la gestión de cobros de todos los estudiantes.

Clase Planes

- Permite visualizar el plan de estudio.

Clase Carrera

- Permite visualizar las carreras habilitadas.
- Listado de estudiantes por carrera.

Clase Materia

- Permite visualizar las materias que podrán ser programadas por los estudiantes de acuerdo al plan de estudios.

Clase Grupo

- Son los grupos habilitados de acuerdo a las materias, fijando un horario.

Clase Horario

- Permite los ABM's de los horarios, registrado la hora de inicio y fin.

Clase Aula

- Permite tener un registro de las diferentes aulas del Campus Universitario.

Clase Edificio

- Permite tener un registro de los diferentes edificios del Campus Universitario.
- Permite la asignación de aulas a los edificios.

Clase Libreta

- Permite los ABM's de las calificaciones de los estudiantes.

II.1.8.1.4.1 DIAGRAMA DE CLASES

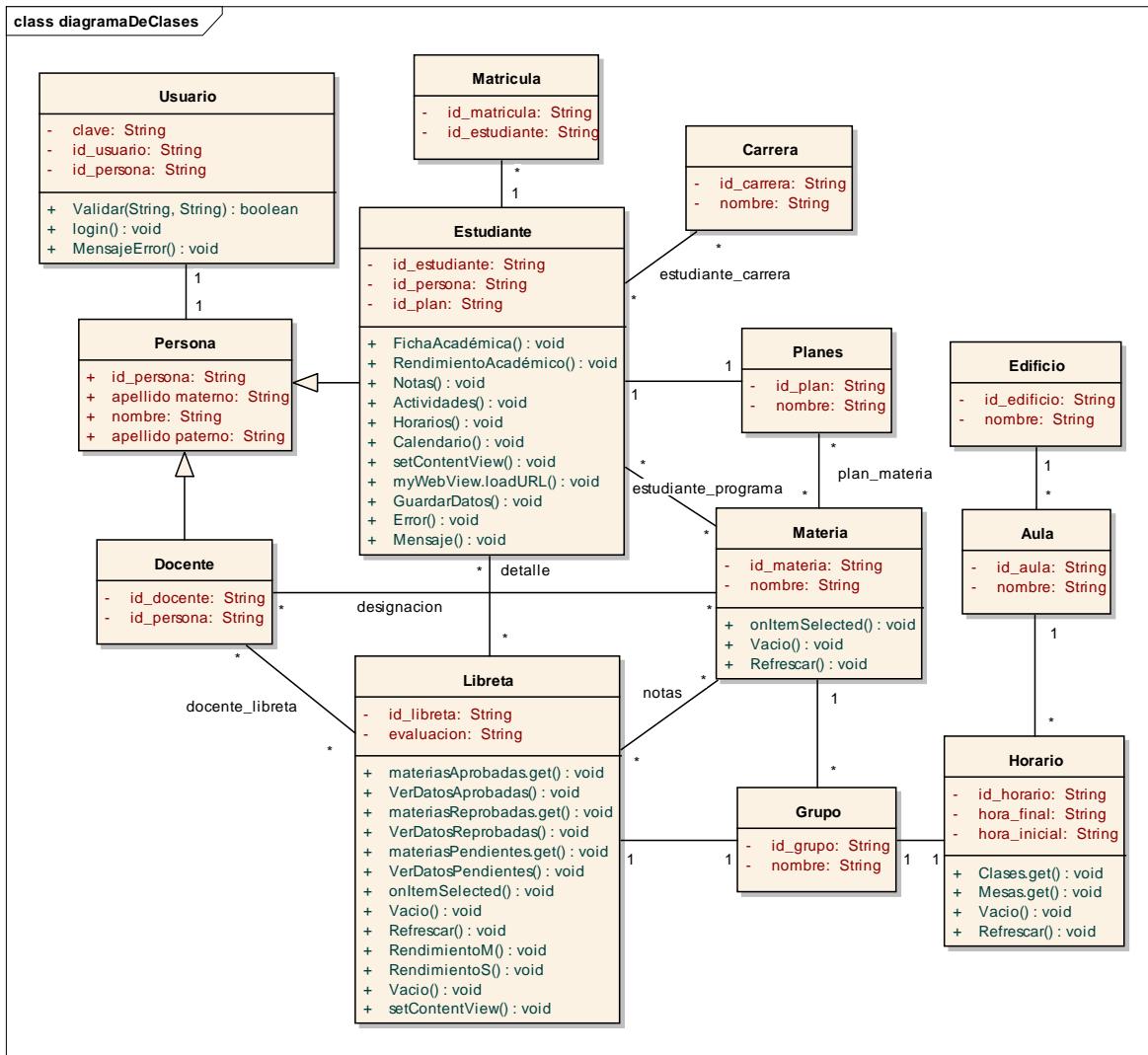


FIGURA 16: DIAGRAMA DE CLASES

II.1.8.2 DIAGRAMAS DE SECUENCIA DE LA APLICACIÓN

II.1.8.2.1 VALIDAR USUARIO

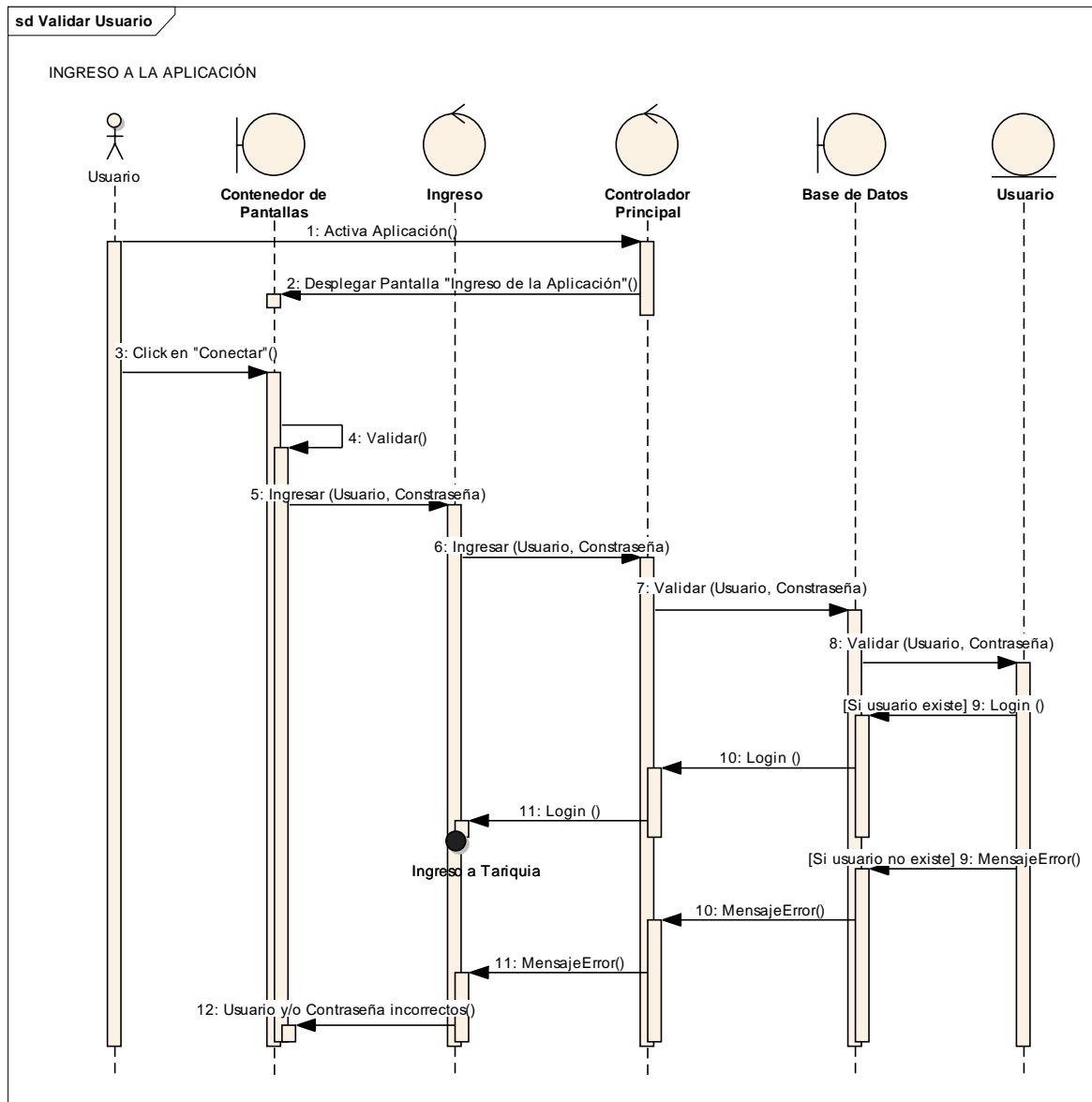


FIGURA 17: VALIDAR USUARIO

II.1.8.2.2 GENERAR FICHA ACADÉMICA

II.1.8.2.2.1 GENERAR FICHA ACADÉMICA APROBADAS

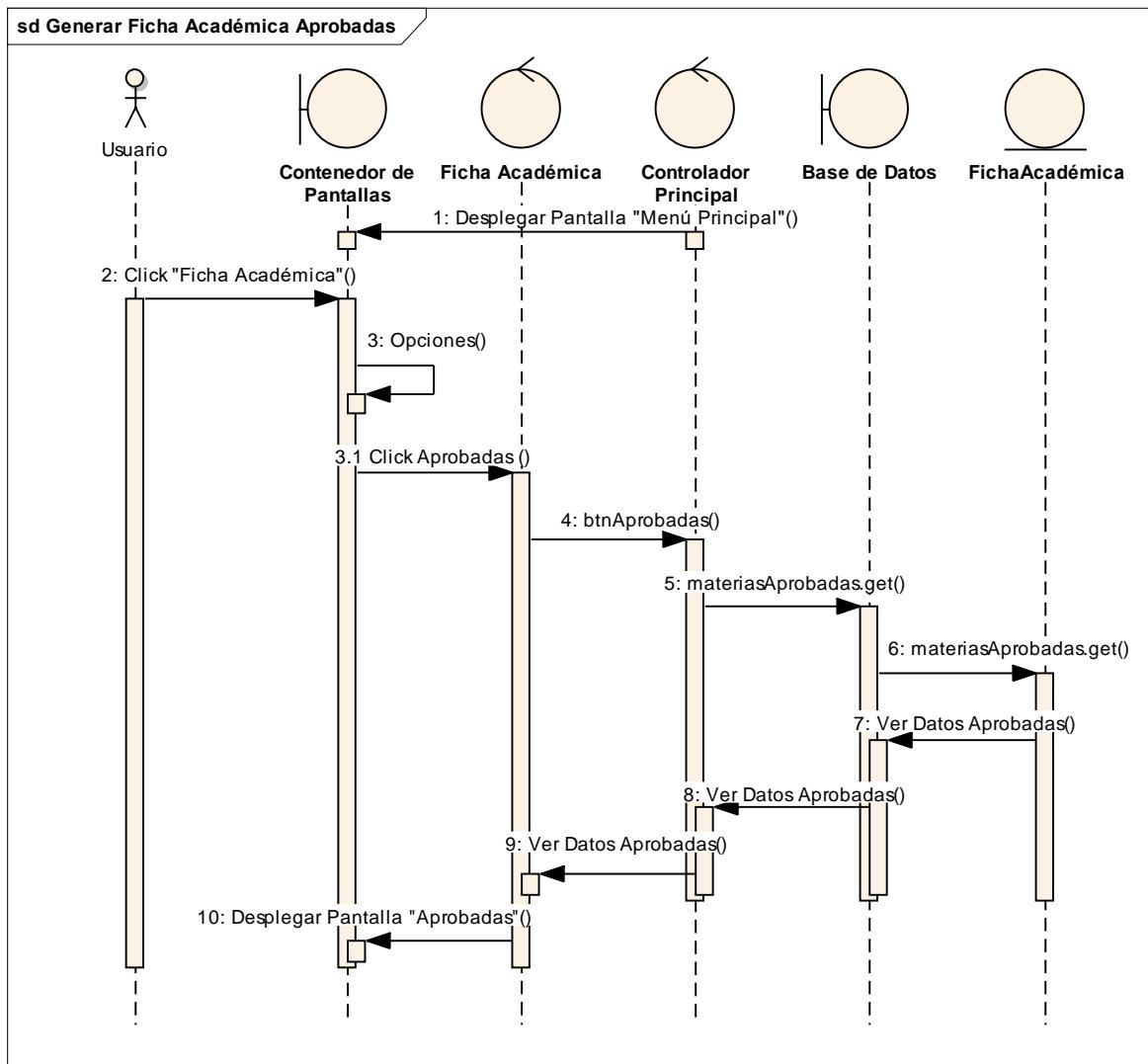


FIGURA 18: GENERAR FICHA ACADÉMICA – APROBADAS

II.1.8.2.2.2 GENERAR FICHA ACADÉMICA REPROBADAS

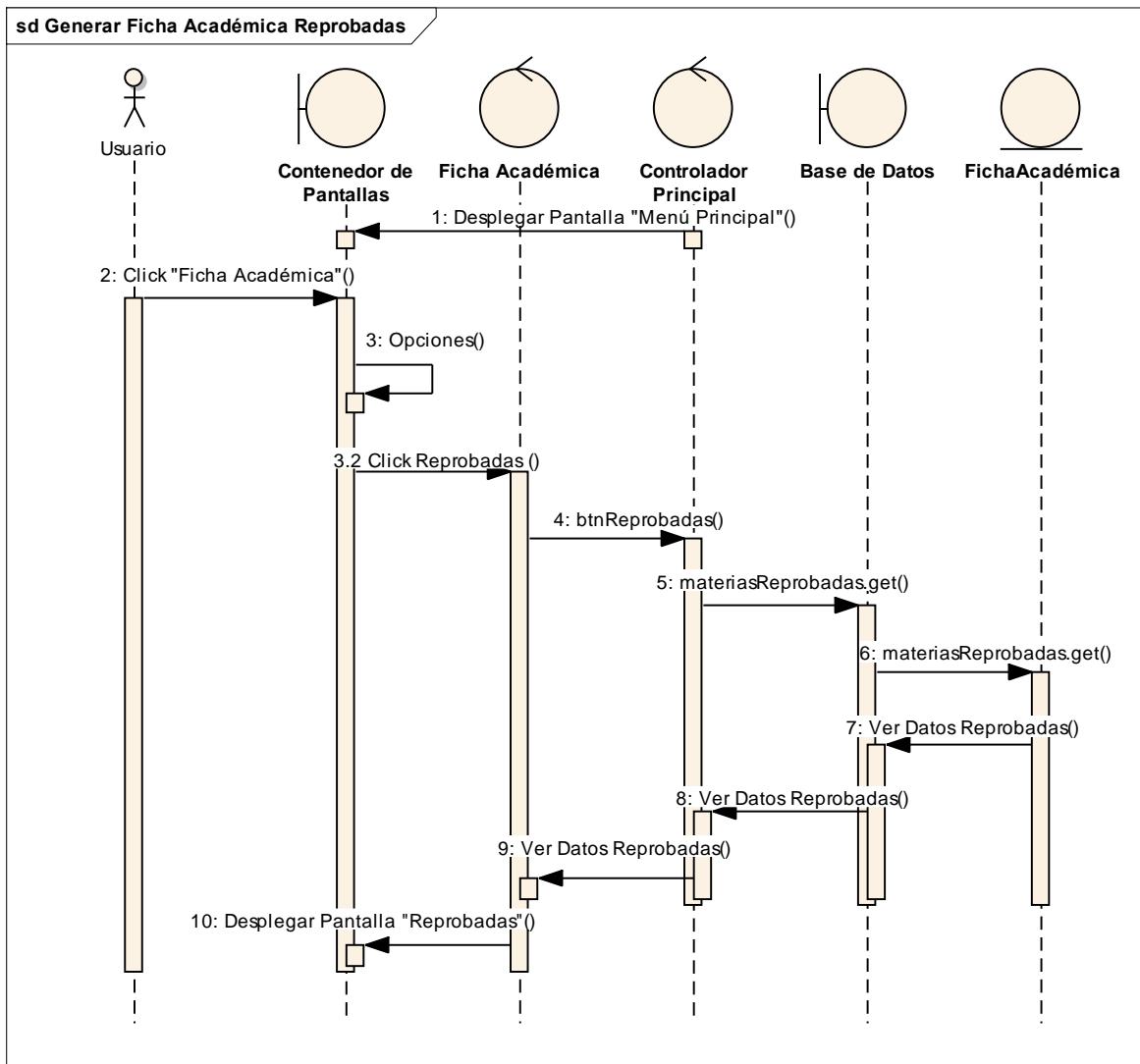


FIGURA 19: GENERAR FICHA ACADÉMICA – REPROBADAS

II.1.8.2.2.3 GENERAR FICHA ACADÉMICA PENDIENTES

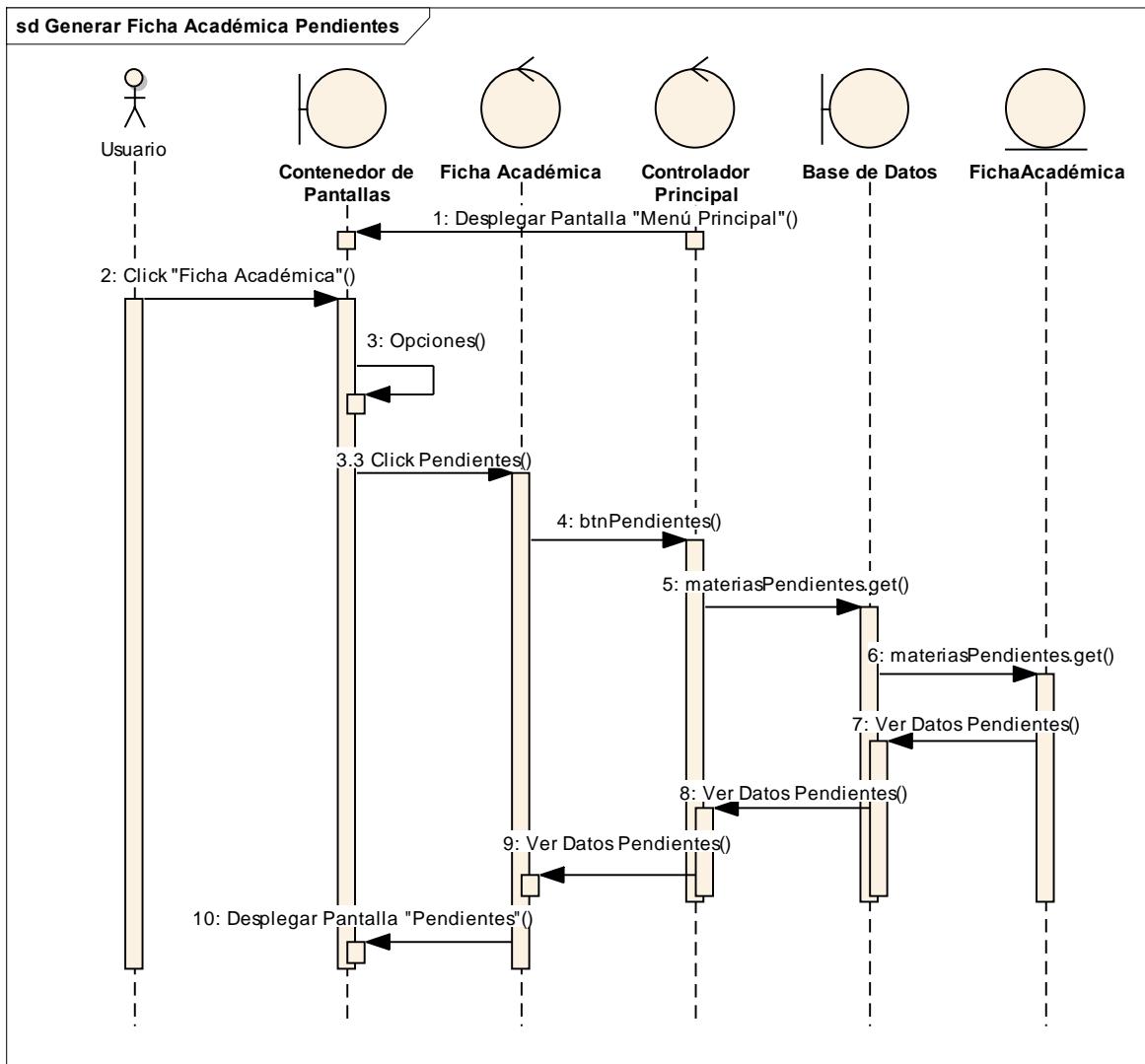


FIGURA 20: GENERAR FICHA ACADÉMICA - PENDIENTES

II.1.8.2.3 DESPLEGAR NOTAS

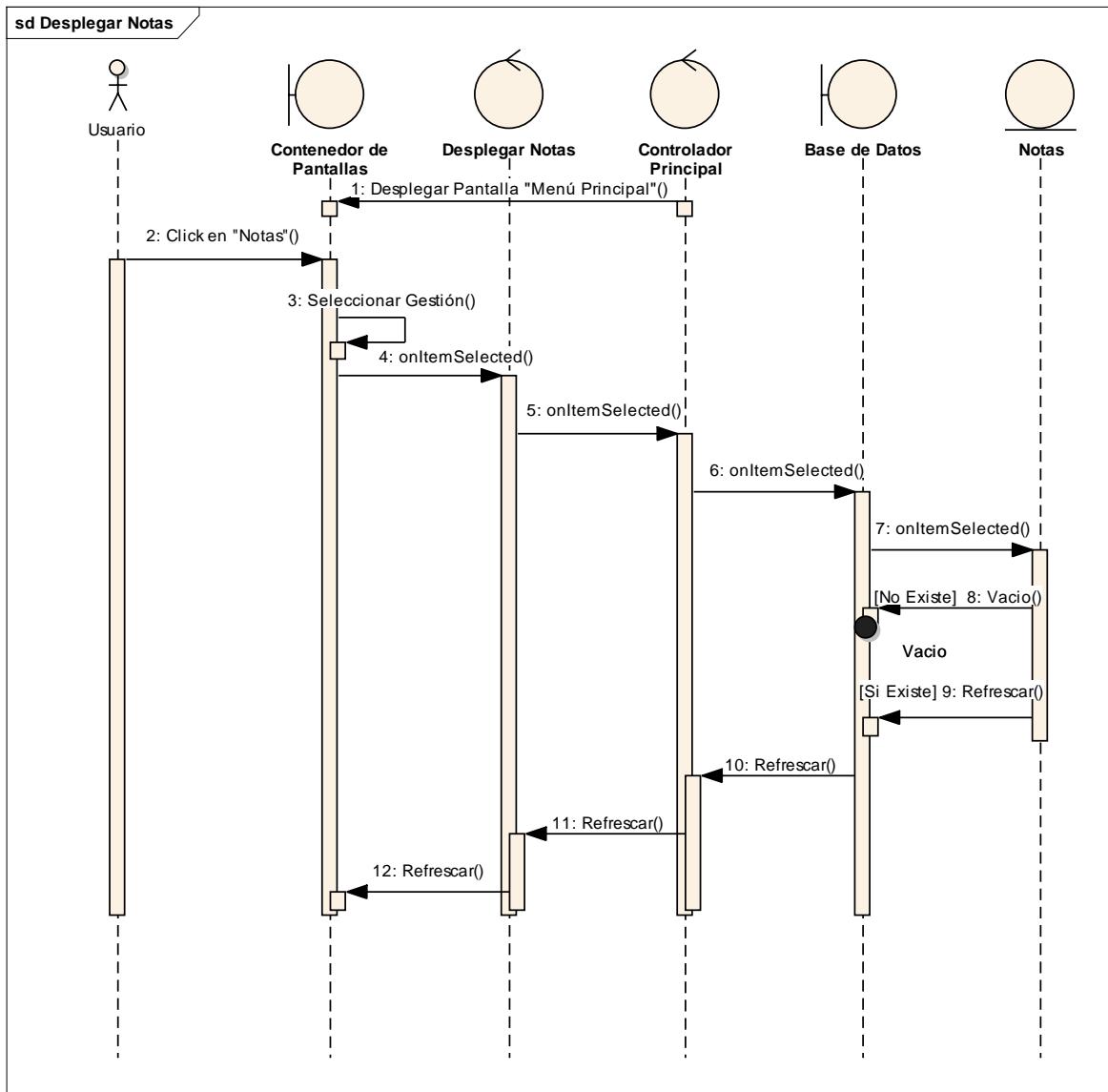


FIGURA 21: DESPLEGAR NOTAS

II.1.8.2.4 DESPLEGAR HORARIOS

II.1.8.2.4.1 DESPLEGAR HORARIOS – CLASES

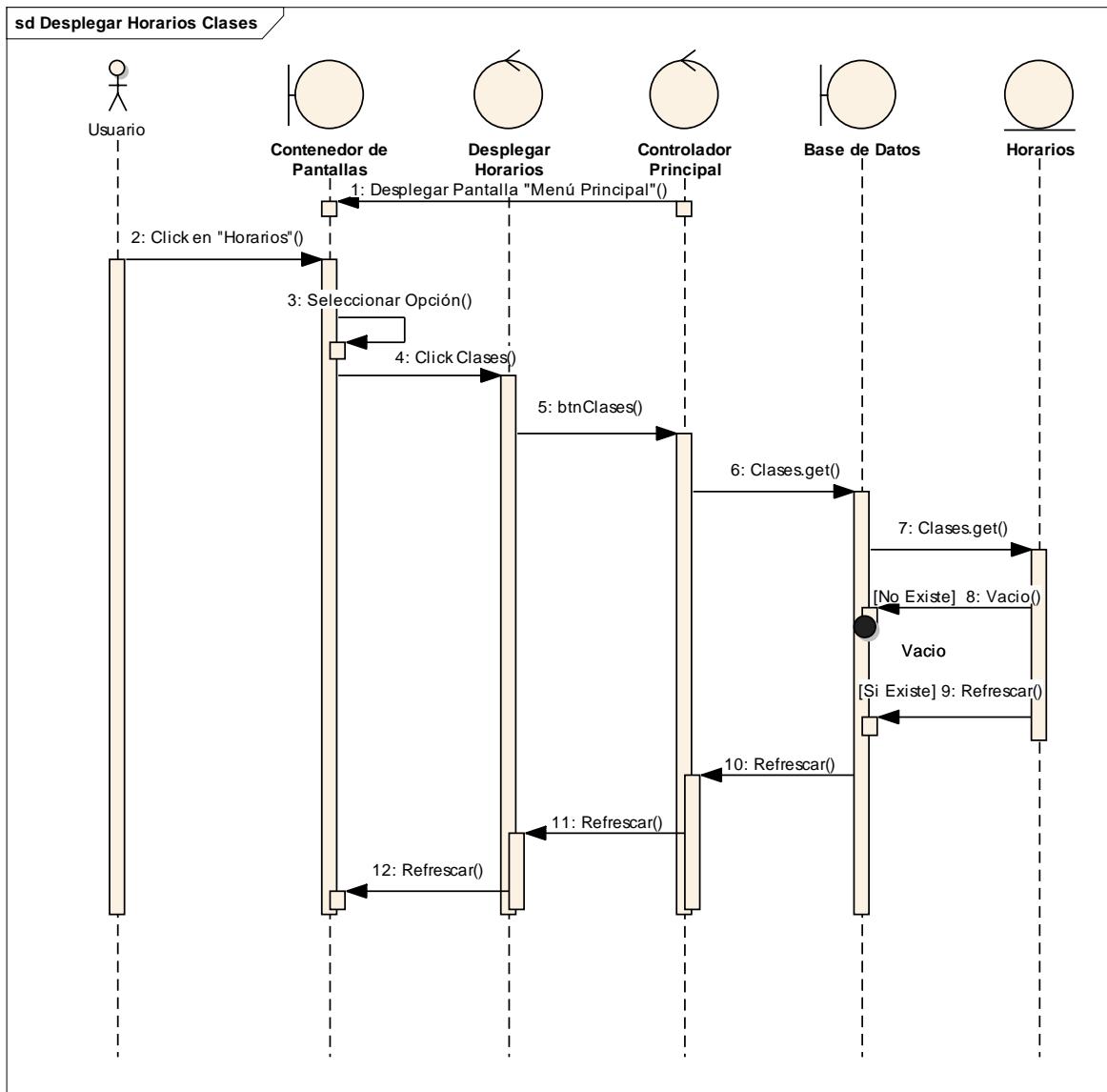


FIGURA 22: DESPLEGAR HORARIOS – CLASES

II.1.8.2.4.1 DESPLEGAR HORARIOS – MESAS

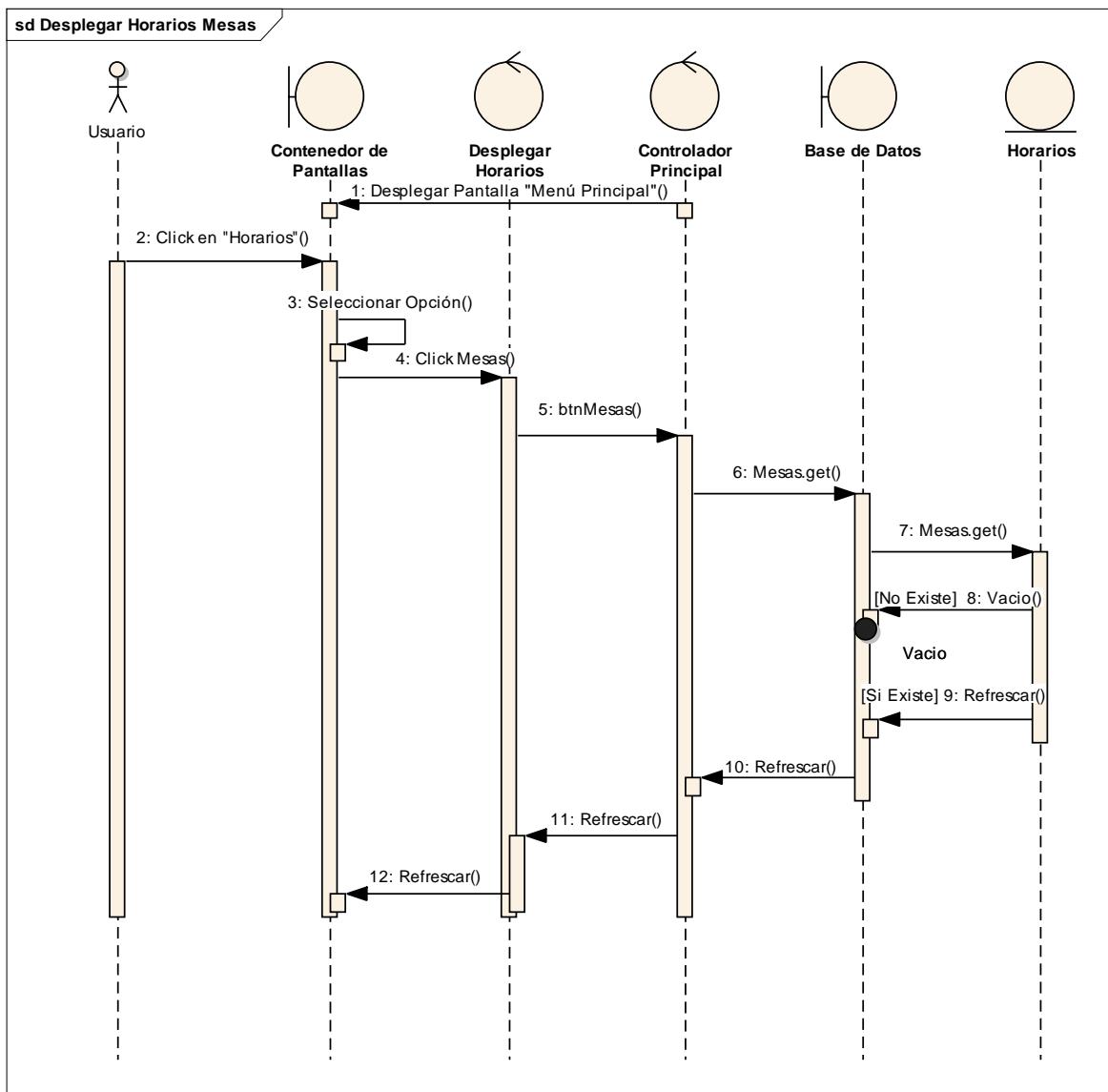


FIGURA 23: DESPLEGAR HORARIOS – MESAS

II.1.8.2.5 DESPLEGAR ACTIVIDADES

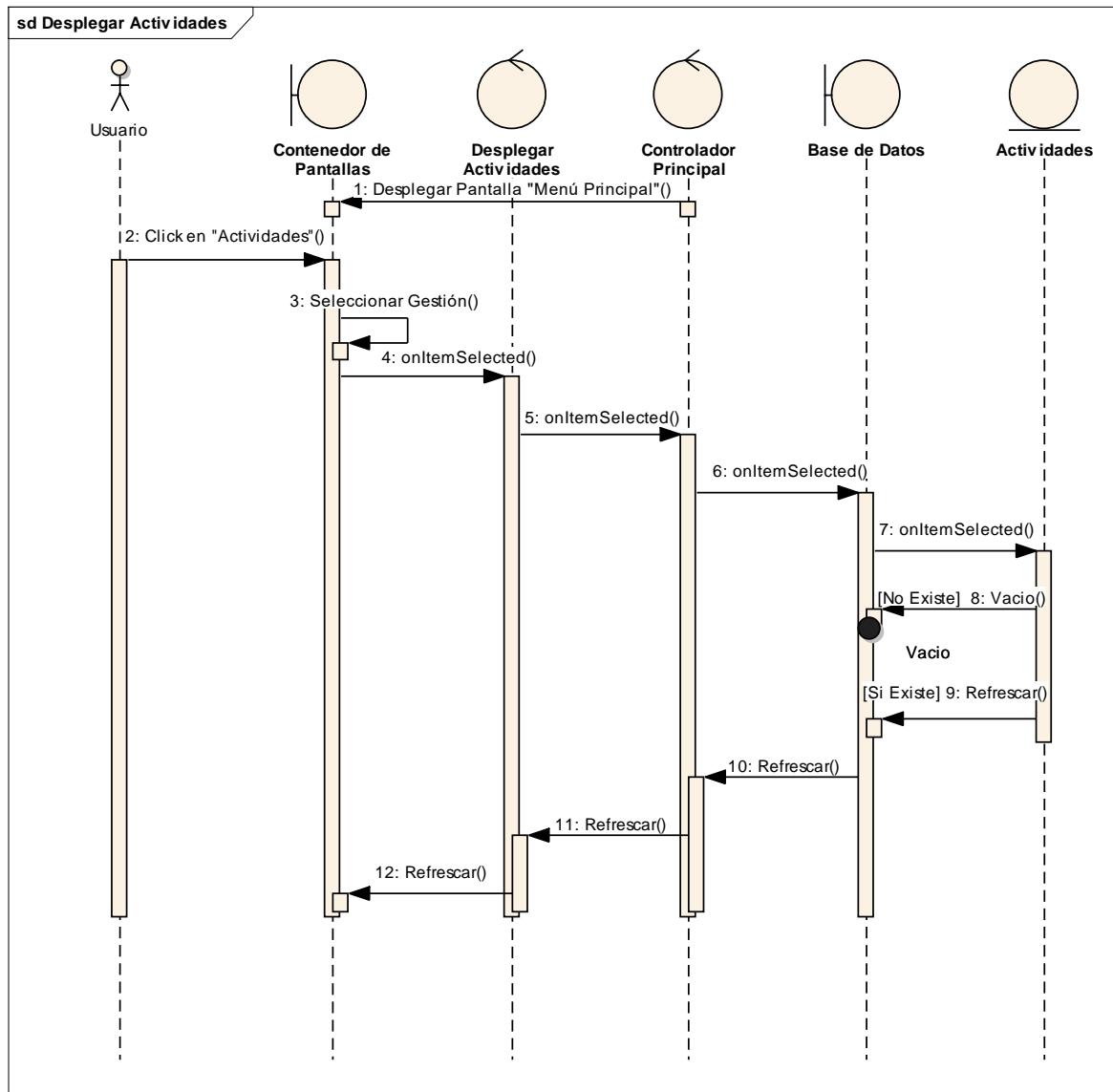


FIGURA 24: DESPLEGAR ACTIVIDADES

II.1.8.2.6 DESPLEGAR CALENDARIO

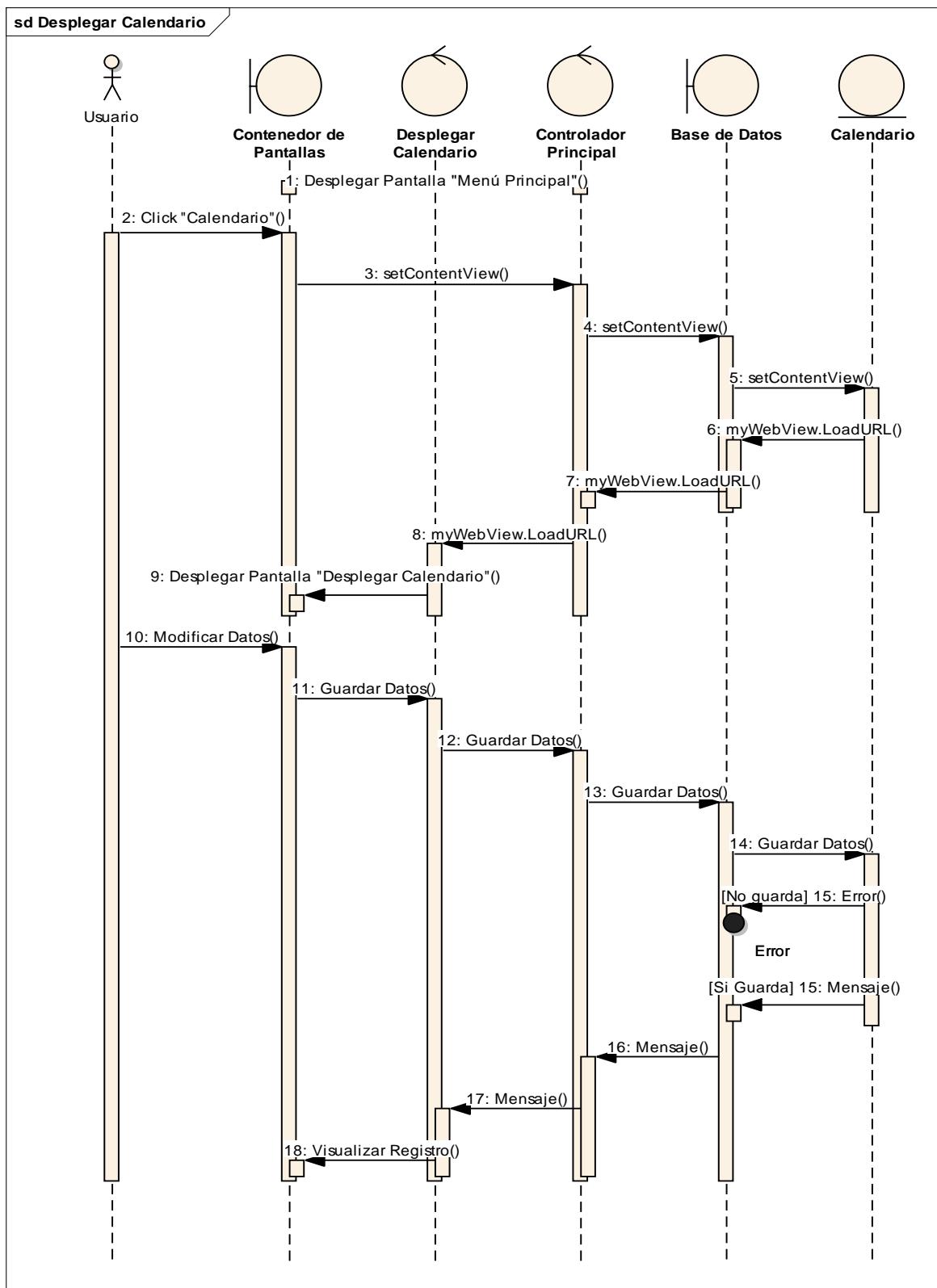


FIGURA 25: DESPLEGAR CALENDARIO

II.1.8.2.7 GENERAR RENDIMIENTO ACADÉMICO

II.1.8.2.7.1 GENERAR RENDIMIENTO ACADÉMICO – MATERIAS

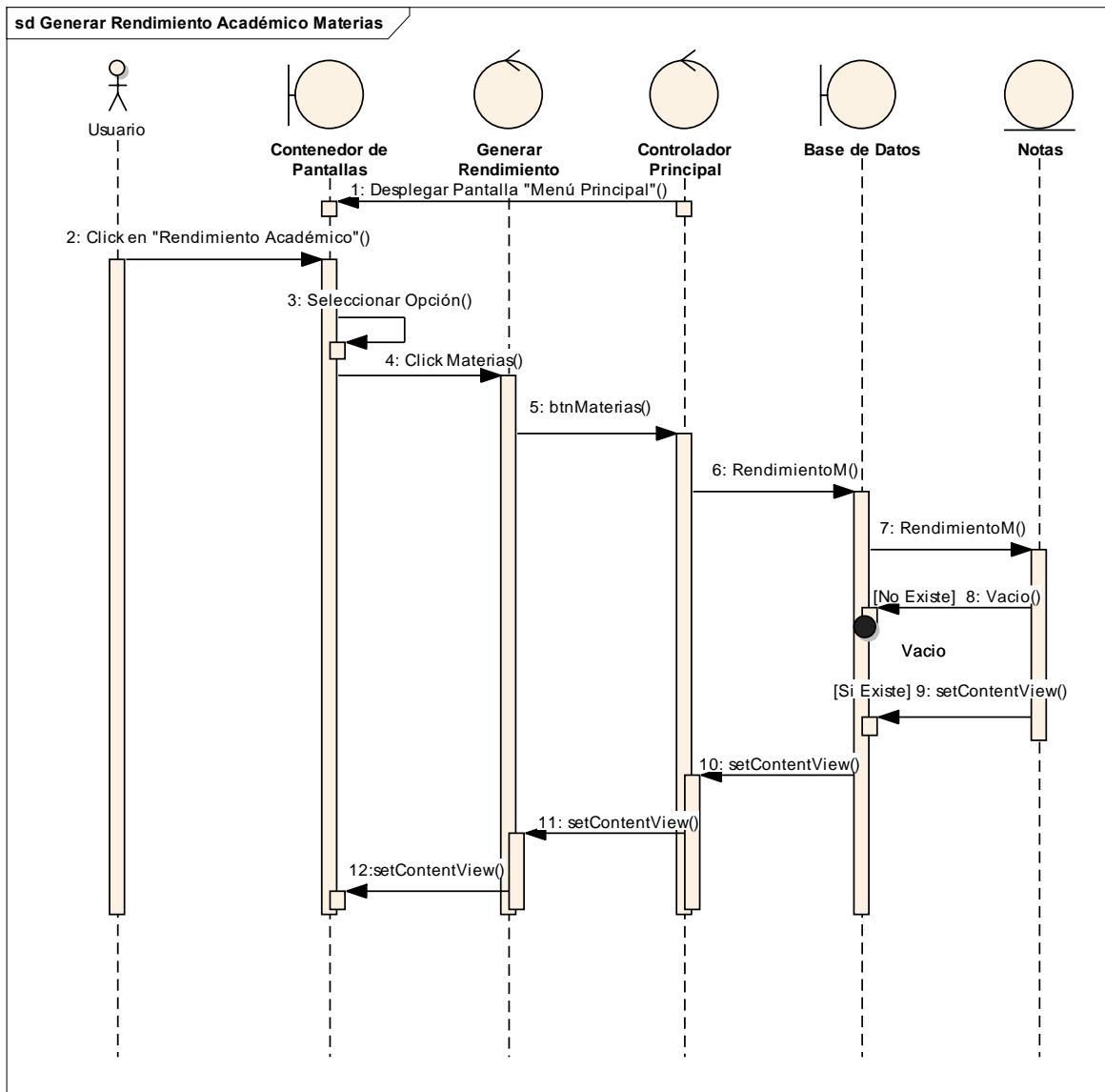


FIGURA 26: GENERAR RENDIMIENTO ACADÉMICO – MATERIAS

II.1.8.2.7.2 GENERAR RENDIMIENTO ACADÉMICO – SEMESTRE

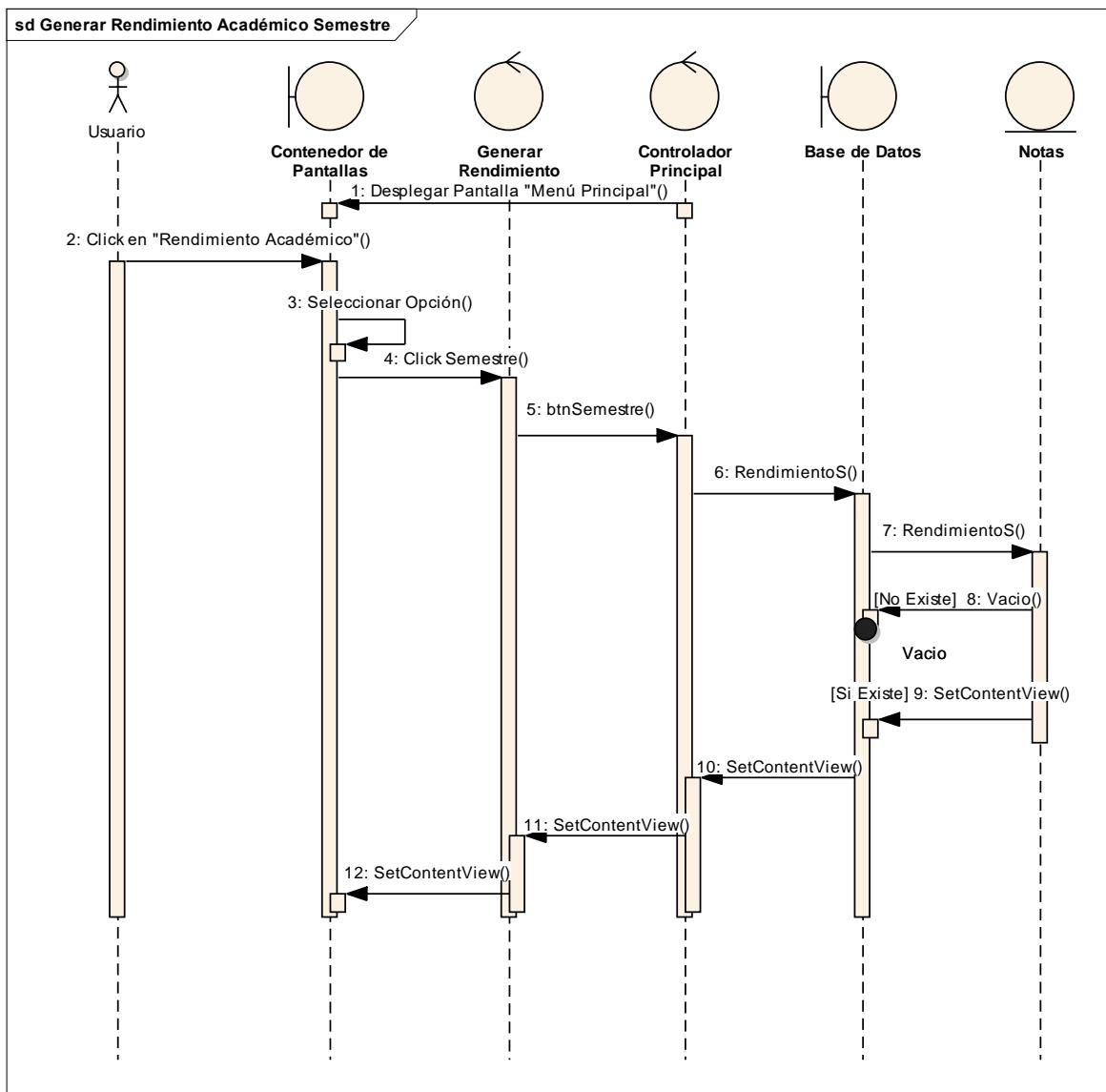


FIGURA 27: GENERAR RENDIMIENTO ACADÉMICO – SEMESTRE

II.1.8.3 DIAGRAMAS DE ACTIVIDADES DE LA APLICACIÓN

II.1.8.3.1 VALIDAR USUARIO

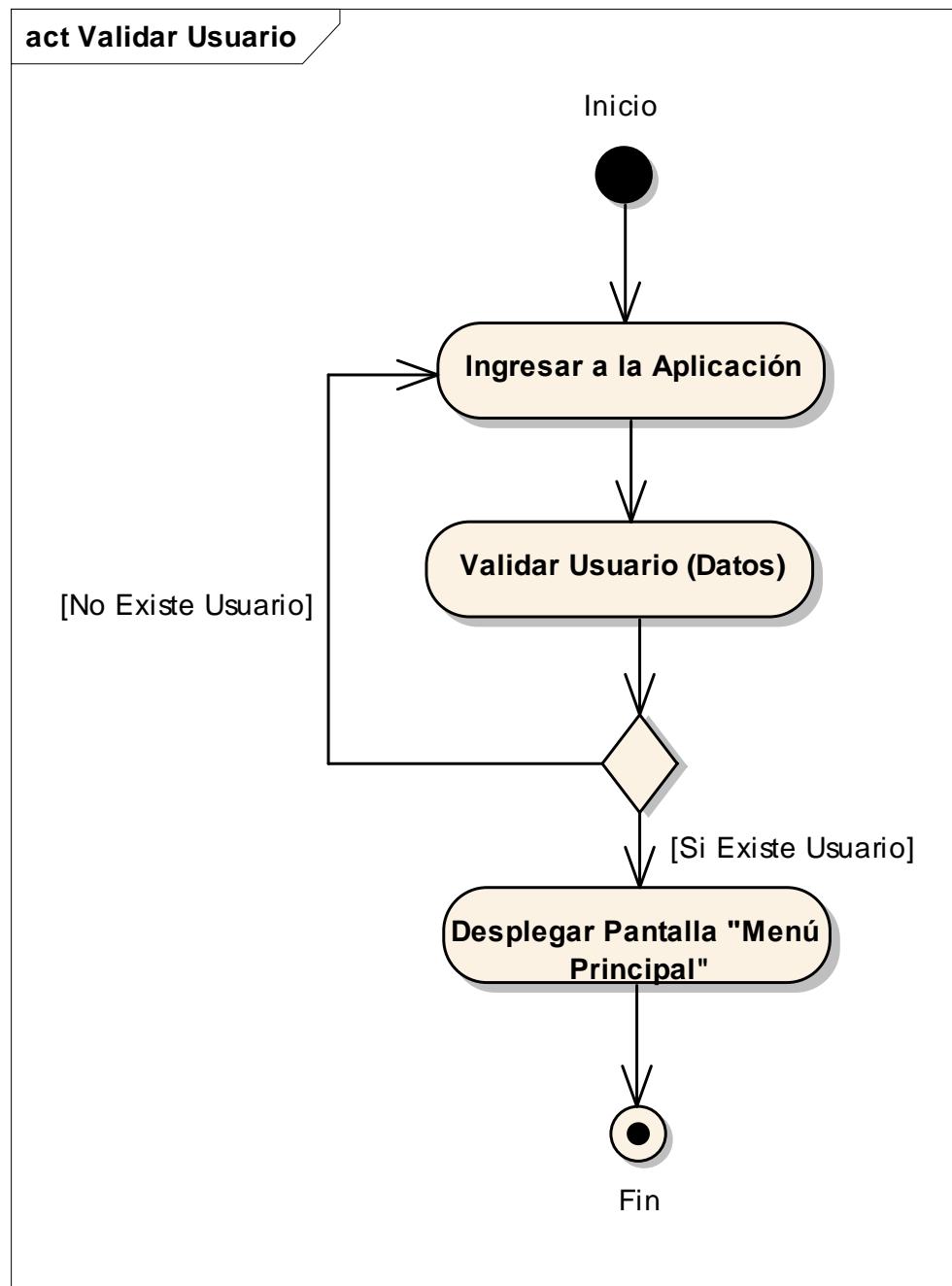


FIGURA 28: VALIDAR USUARIO

II.1.8.3.2 GENERAR FICHA ACADÉMICA

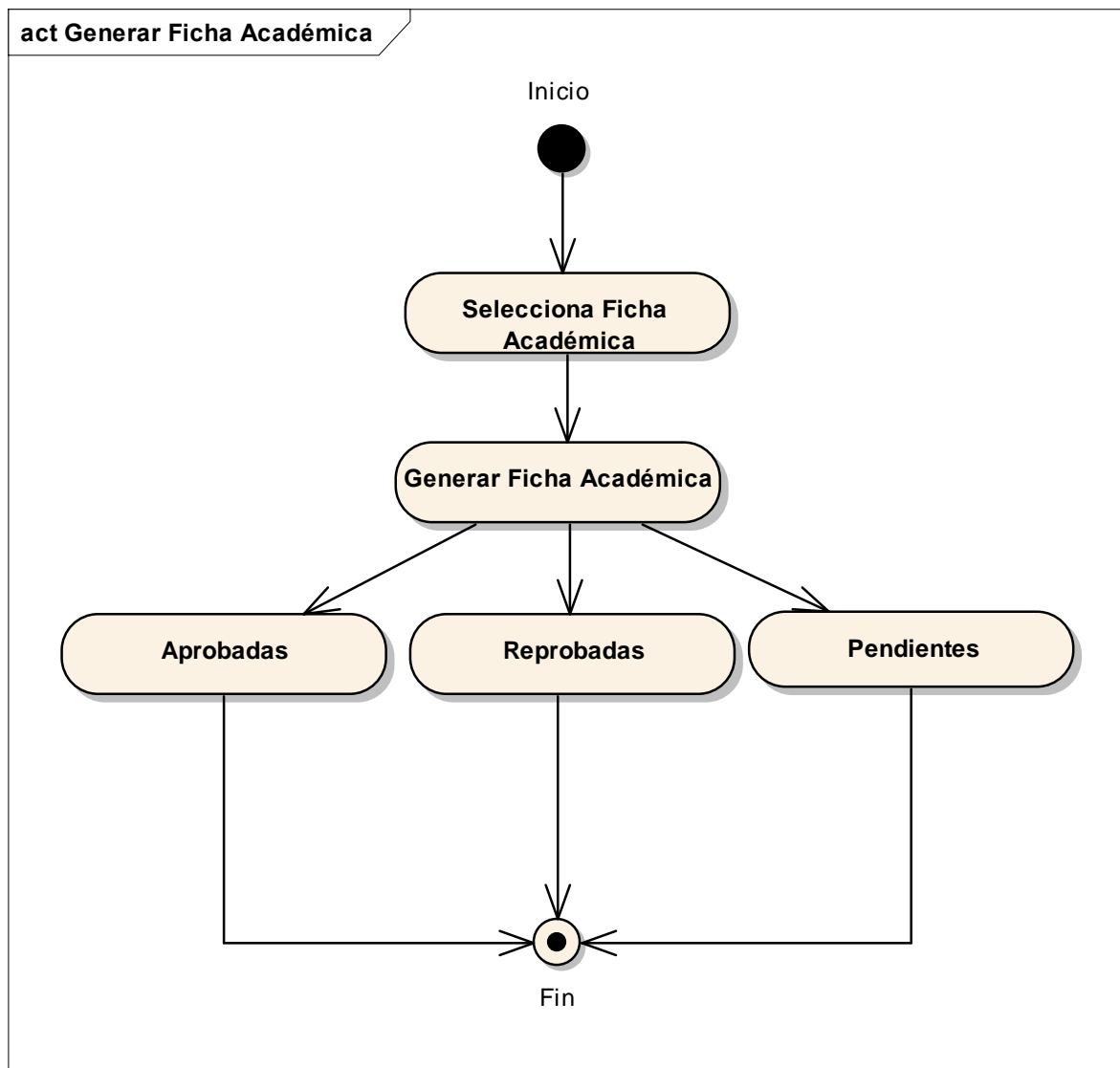


FIGURA 29: GENERAR FICHA ACADÉMICA

II.1.8.3.2.1 GENERAR FICHA ACADÉMICA – APROBADAS

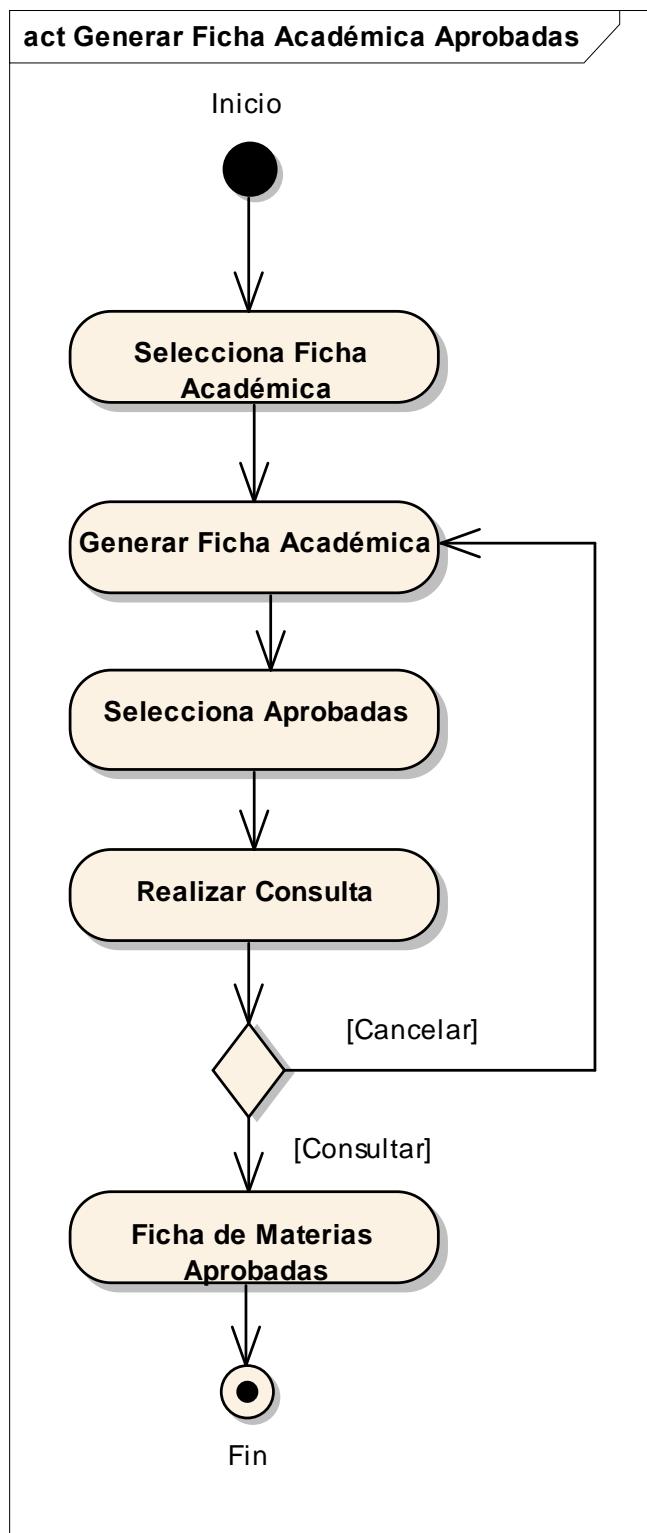


FIGURA 30: GENERAR FICHA ACADÉMICA – APROBADAS

II.1.8.3.2.2 GENERAR FICHA ACADÉMICA – REPROBADAS

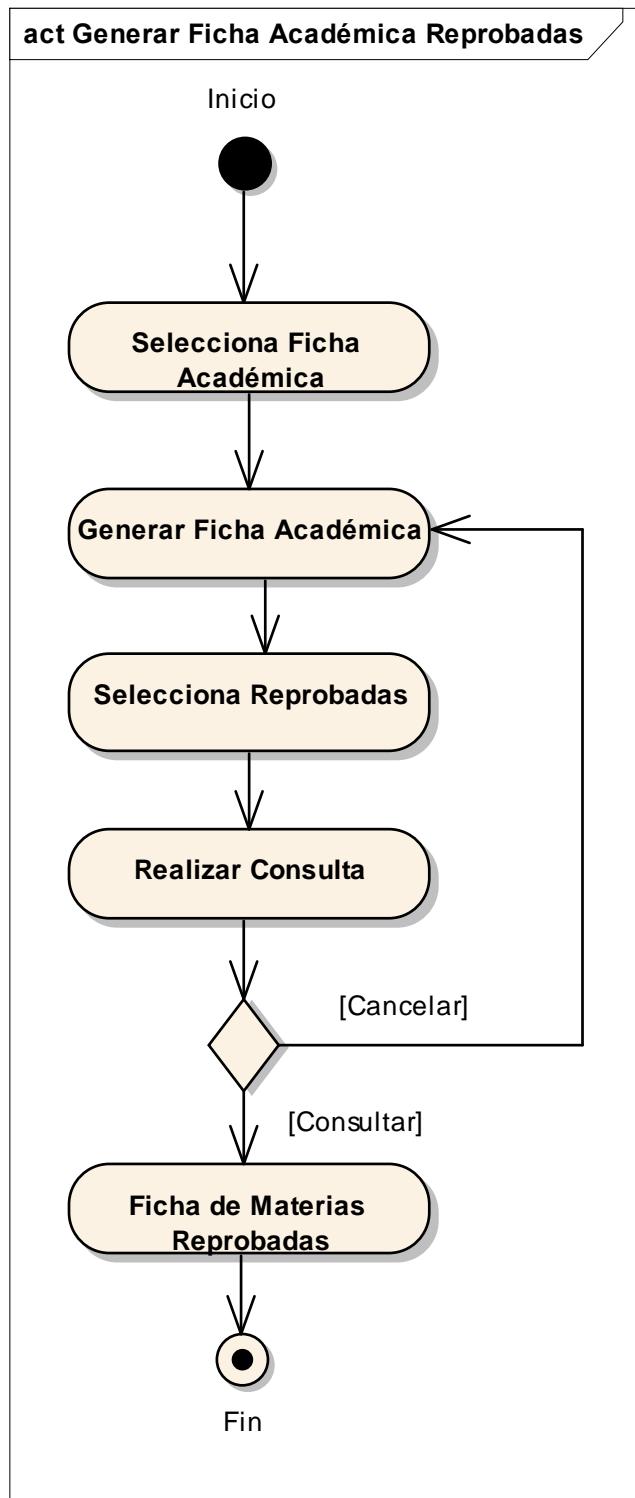


FIGURA 31: GENERAR FICHA ACADÉMICA – REPROBADAS

II.1.8.3.2.3 GENERAR FICHA ACADÉMICA – PENDIENTES

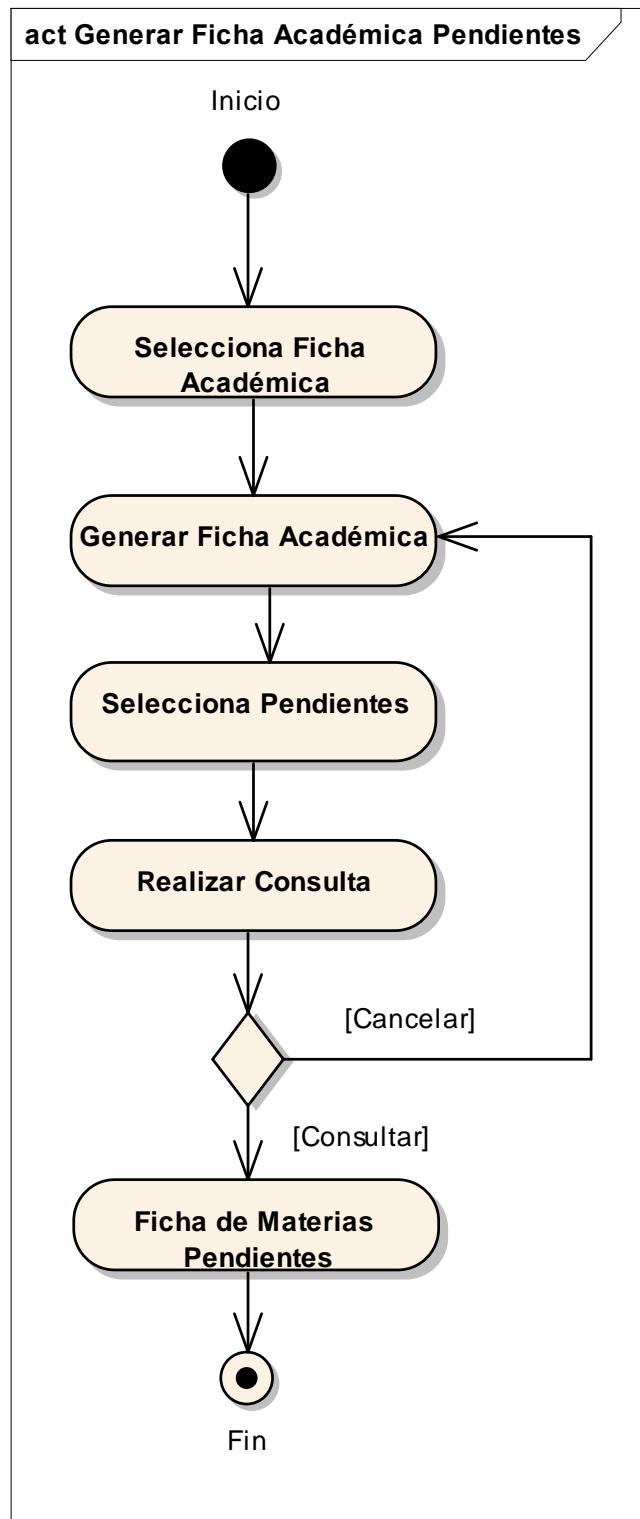


FIGURA 32: GENERAR FICHA ACADÉMICA – PENDIENTES

II.1.8.5.2.1 DESPLEGAR NOTAS

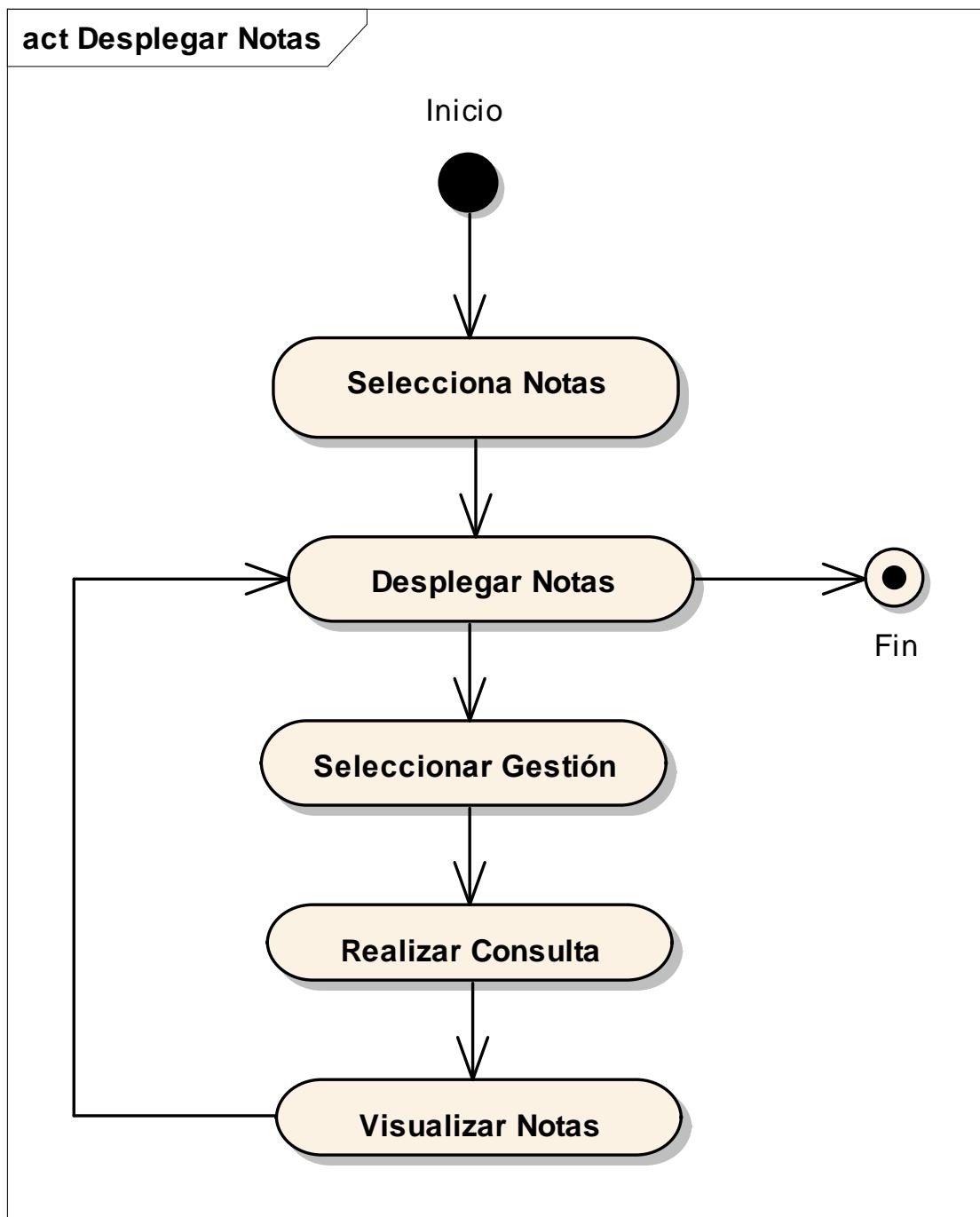


FIGURA 33: DESPLEGAR NOTAS

II.1.8.5.2.2 DESPLEGAR HORARIOS

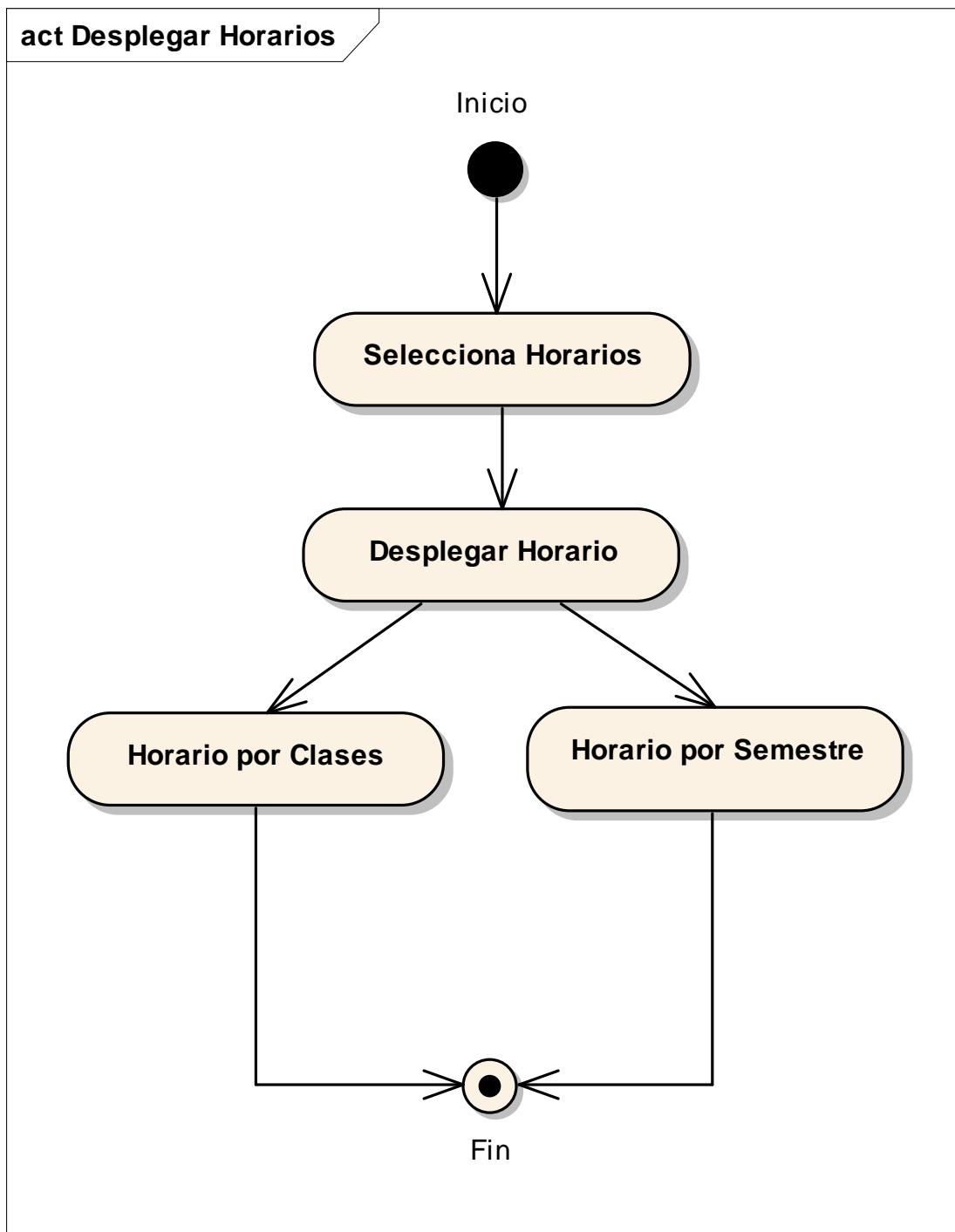


FIGURA 34: DESPLEGAR HORARIOS

II.1.8.5.2.2.1 DESPLEGAR HORARIOS – CLASES

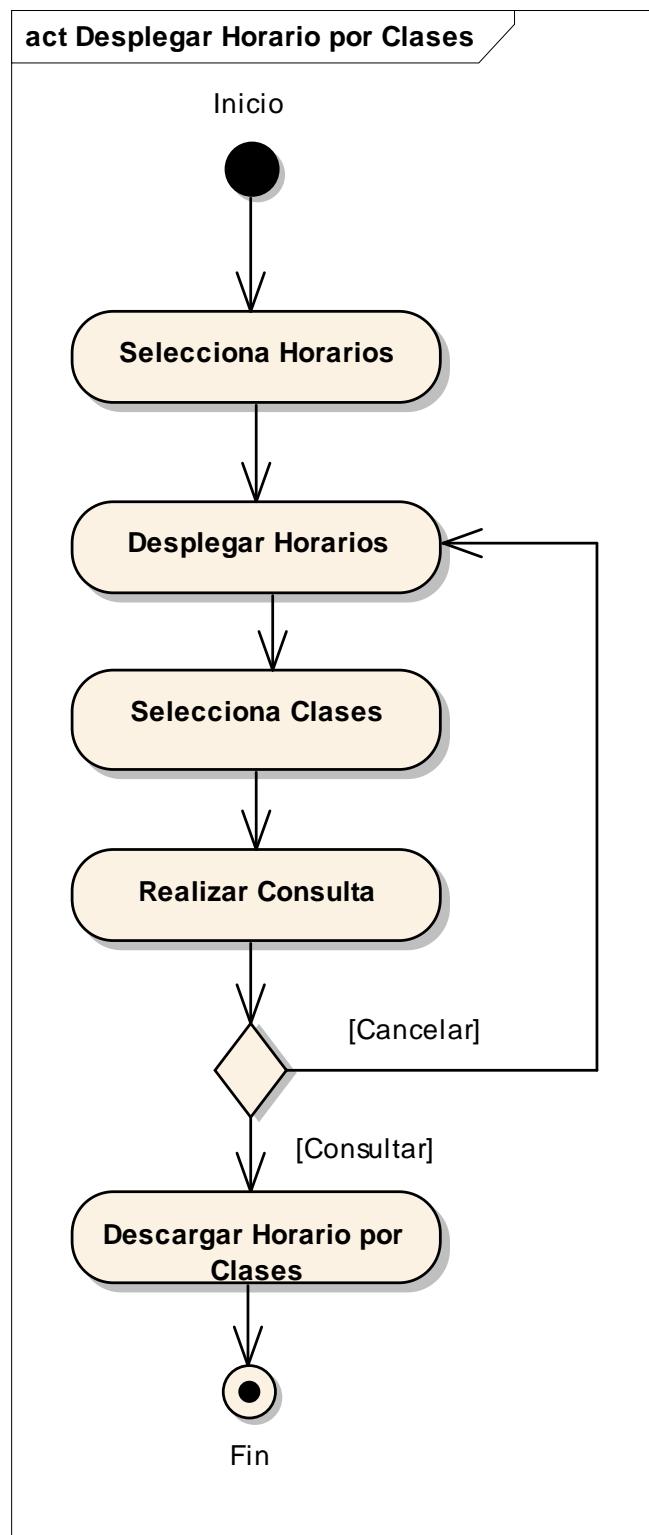


FIGURA 35: DESPLEGAR HORARIOS – CLASES

II.1.8.5.2.2 DESPLEGAR HORARIOS – MESAS

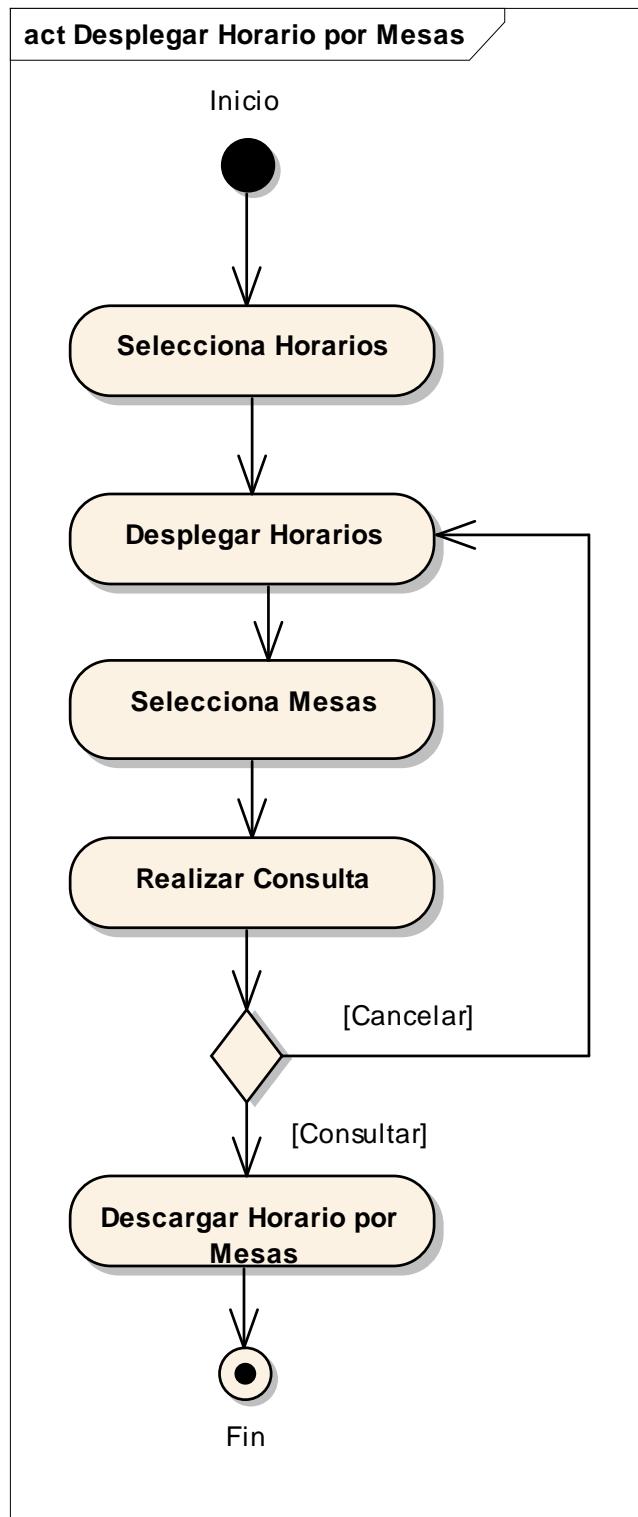


FIGURA 36: DESPLEGAR HORARIOS – MESAS

II.1.8.5.2.3 DESPLEGAR ACTIVIDADES

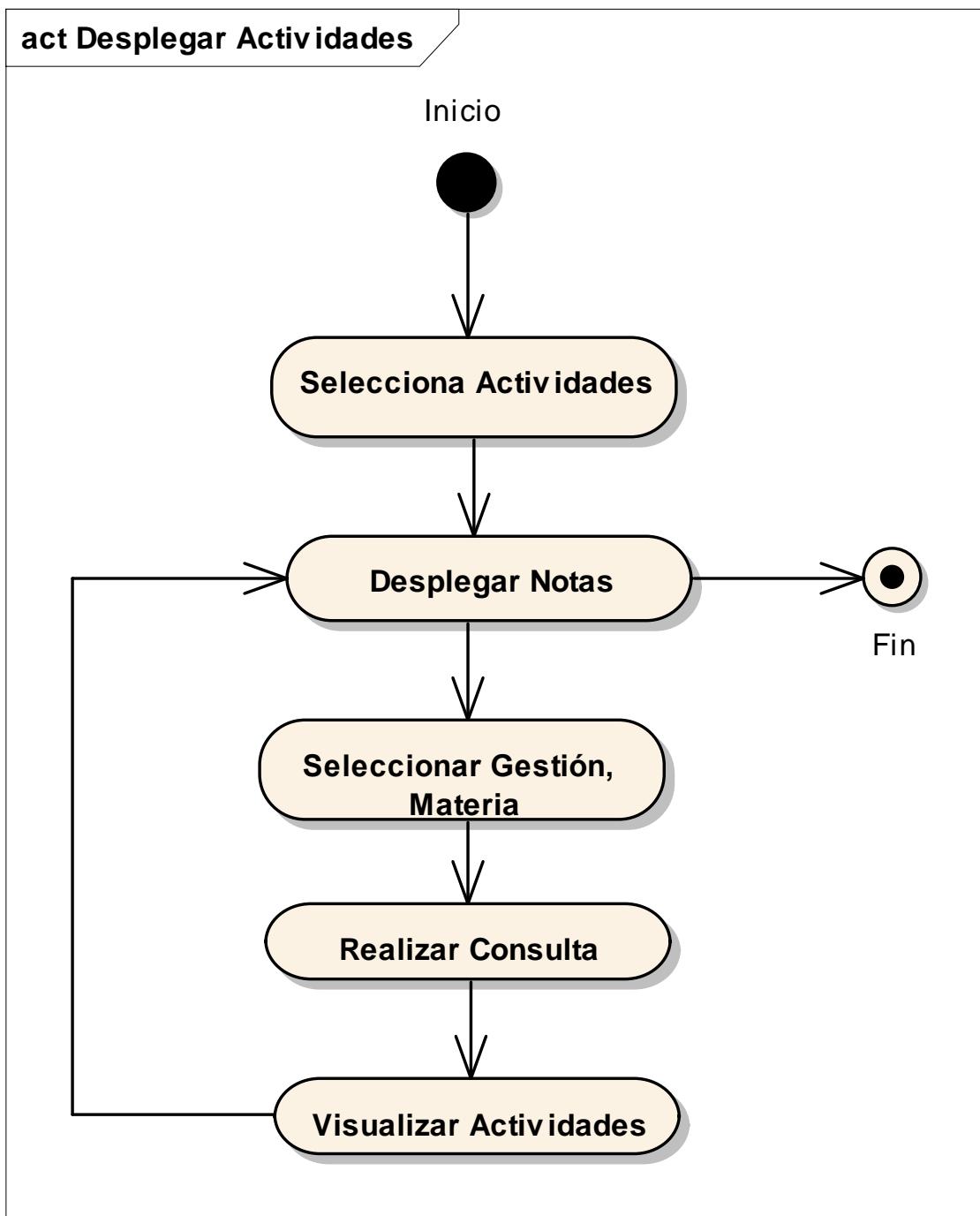
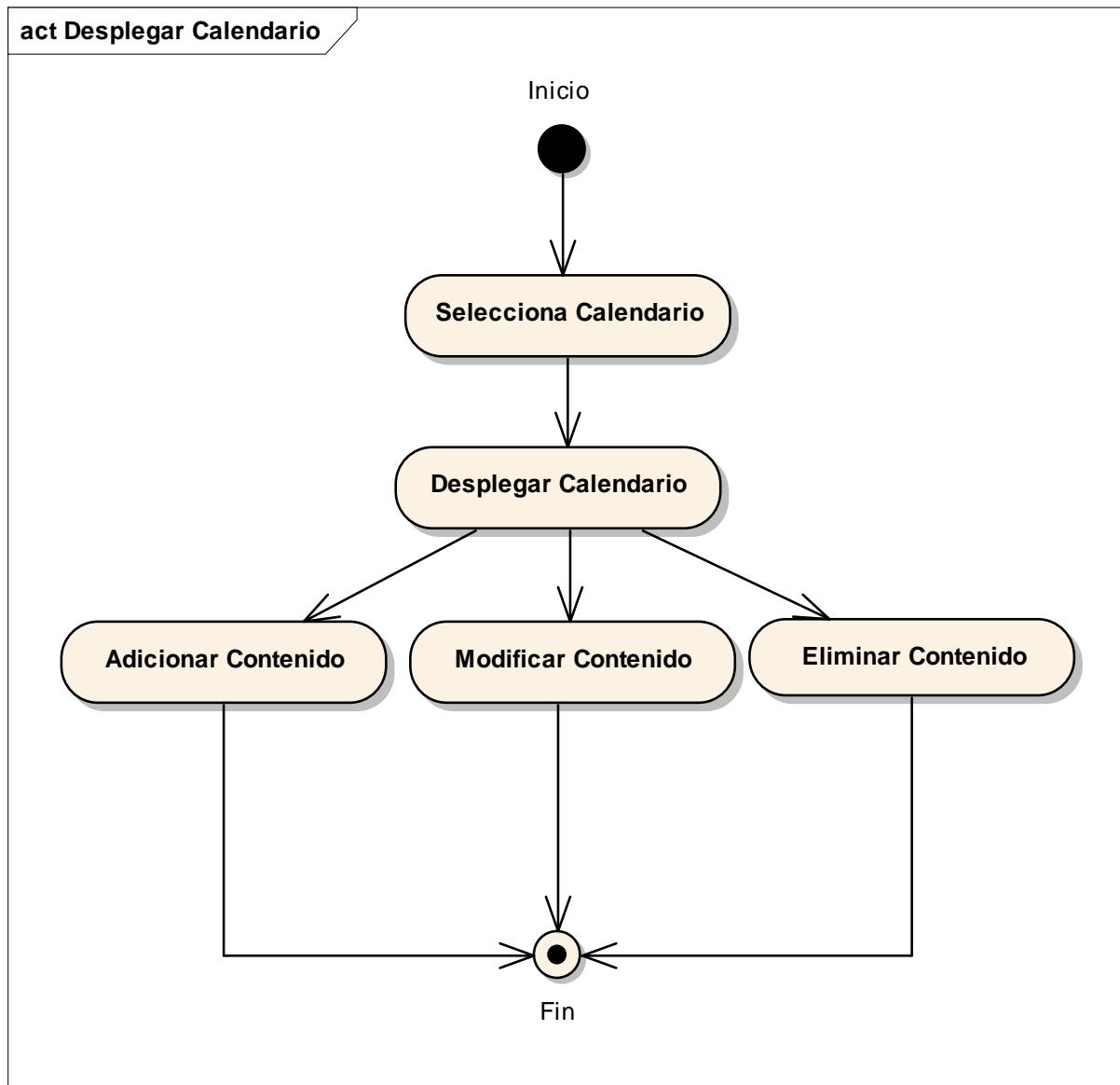


FIGURA 37: DESPLEGAR ACTIVIDADES

II.1.8.5.2.4 DESPLEGAR CALENDARIO**FIGURA 38: DESPLEGAR CALENDARIO**

II.1.8.5.3 GENERAR RENDIMIENTO ACADÉMICO

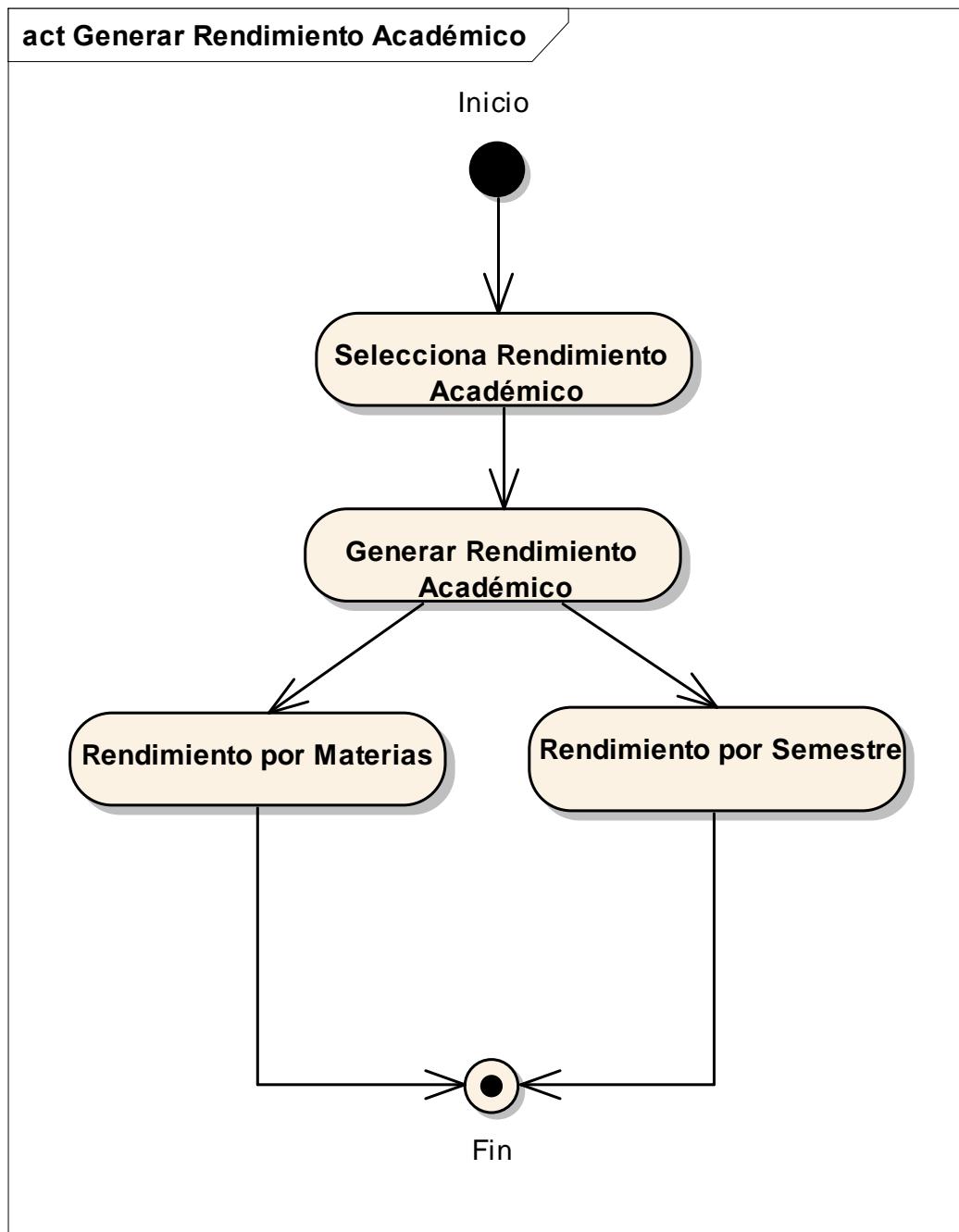
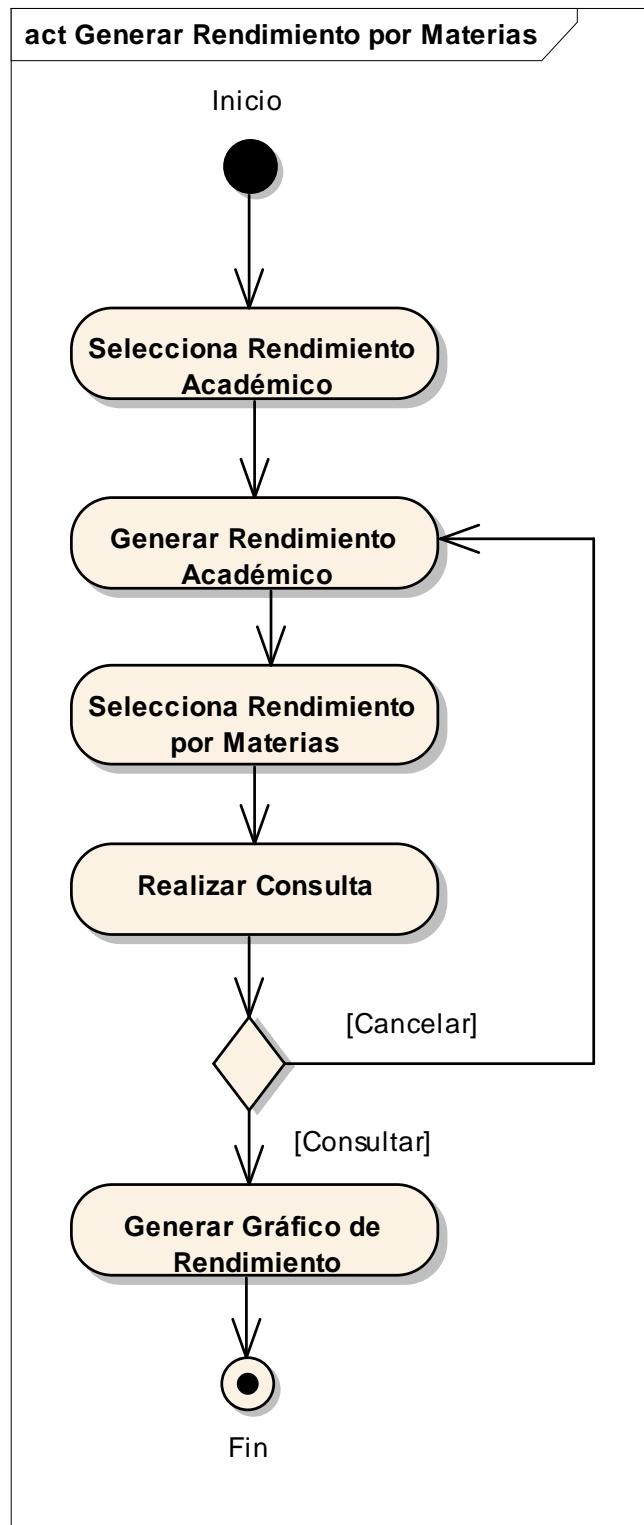


FIGURA 39: GENERAR RENDIMIENTO ACADÉMICO

II.1.8.5.3.1 GENERAR RENDIMIENTO ACADÉMICO – MATERIAS**FIGURA 40: GENERAR RENDIMIENTO ACADÉMICO – MATERIAS**

II.1.8.5.3.2 GENERAR RENDIMIENTO ACADÉMICO – SEMESTRE

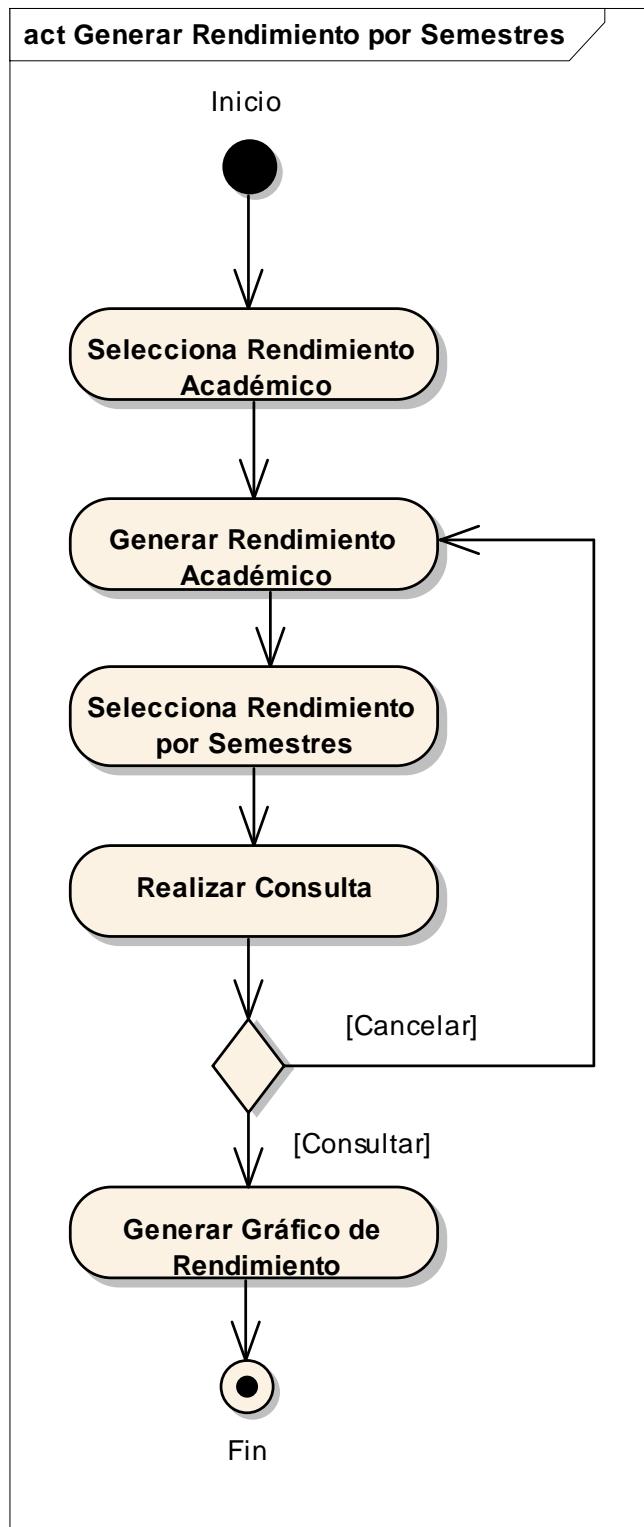


FIGURA 41: GENERAR RENDIMIENTO ACADÉMICO – SEMESTRE

Modelo Navegacional De La Aplicación

II.1.9 DIAGRAMA NAVEGACIONAL DE LA APLICACIÓN

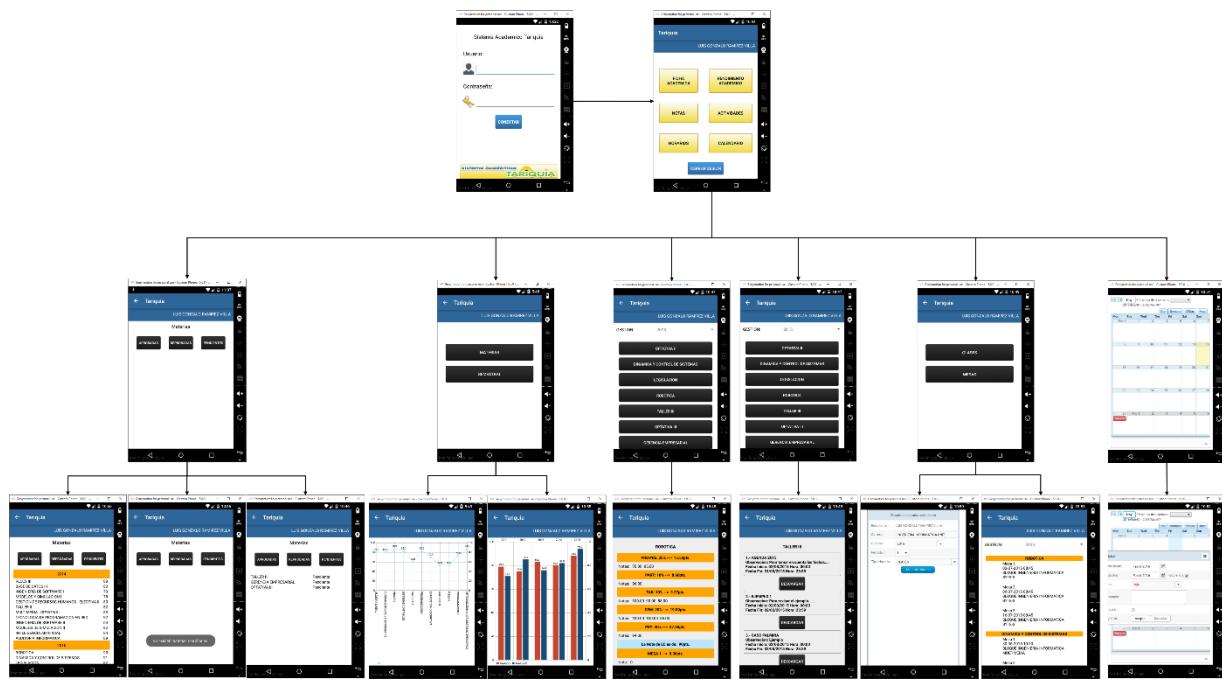


FIGURA 42: DIAGRAMA NAVEGACIONAL

Diseño De Interfaces De Usuario

II.1.10 DISEÑO DE PANTALLAS DE USUARIO

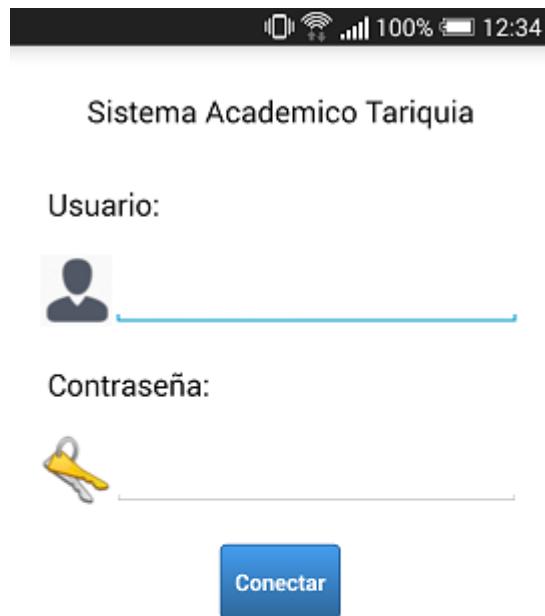


FIGURA 43: PANTALLA INGRESO DE LA APLICACIÓN

Esta es la pantalla de Ingreso de la Aplicación, en ella se debe ingresar los datos de usuario y contraseña.

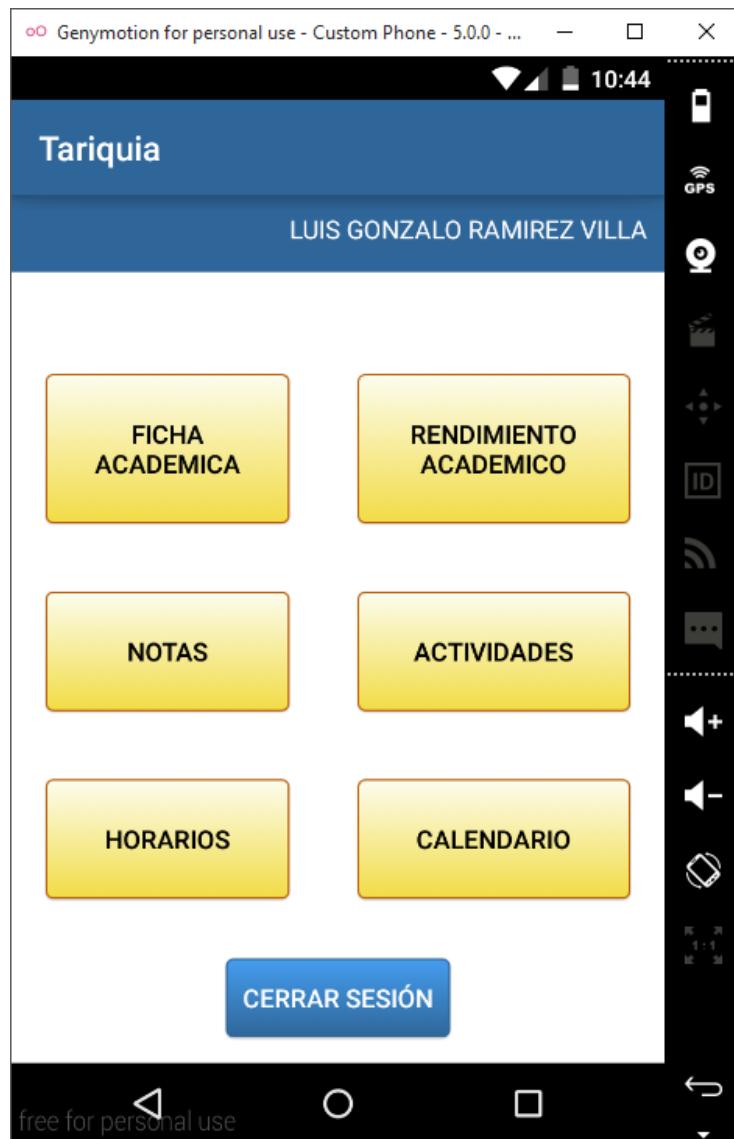


FIGURA 44: PANTALLA MENÚ PRINCIPAL

Esta pantalla es el contenedor de todas las pantallas siguientes, en la parte superior se presenta el nombre del estudiante que ingresa.

II.1.10.1 GENERAR FICHA ACADÉMICA

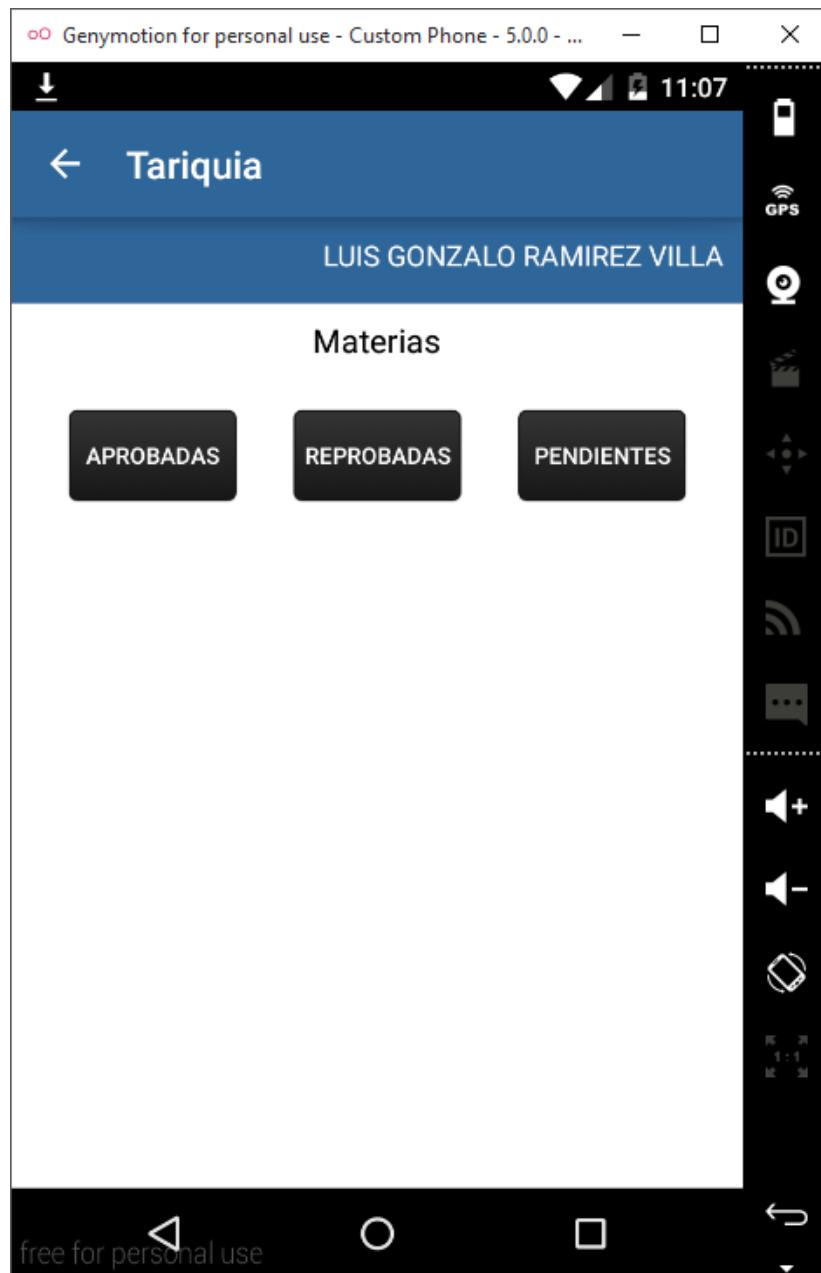


FIGURA 45: PANTALLA FICHA ACADÉMICA

Al hacer clic en “Ficha Académica”, se mostrará el listado de las materias de acuerdo a tres filtros, al seleccionar: Aprobadas podrá visualizar las materias aprobadas, Reprobadas visualizará sólo materias que hayan sido reprobadas, y en caso de Pendientes las materias que debe vencer el estudiante para poder graduarse.



FIGURA 46: PANTALLA MATERIAS APROBADAS

En esta pantalla se observan las materias aprobadas por el estudiante, organizadas por años.

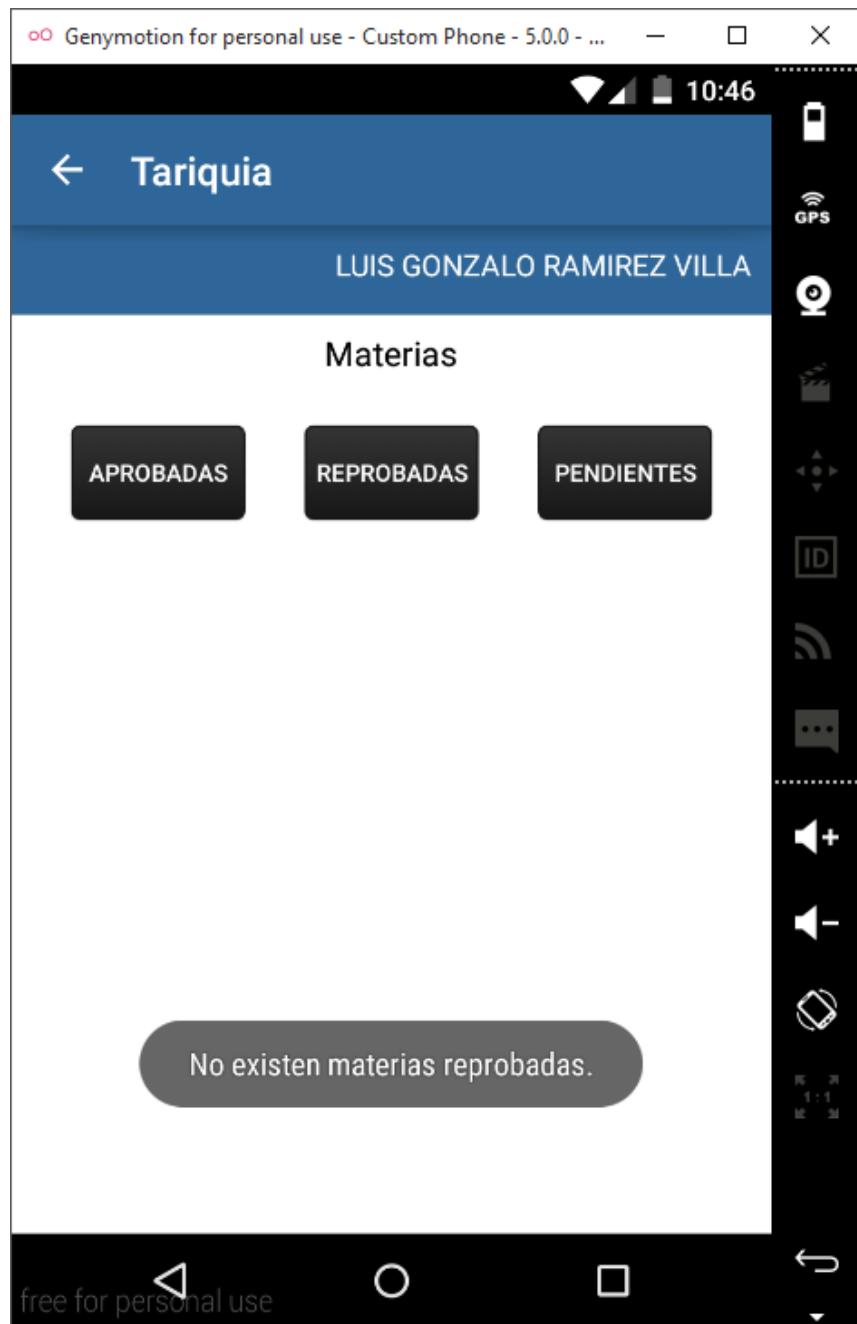


FIGURA 47: PANTALLA MATERIAS REPROBADAS

En esta pantalla se observan las materias reprobadas por el estudiante.

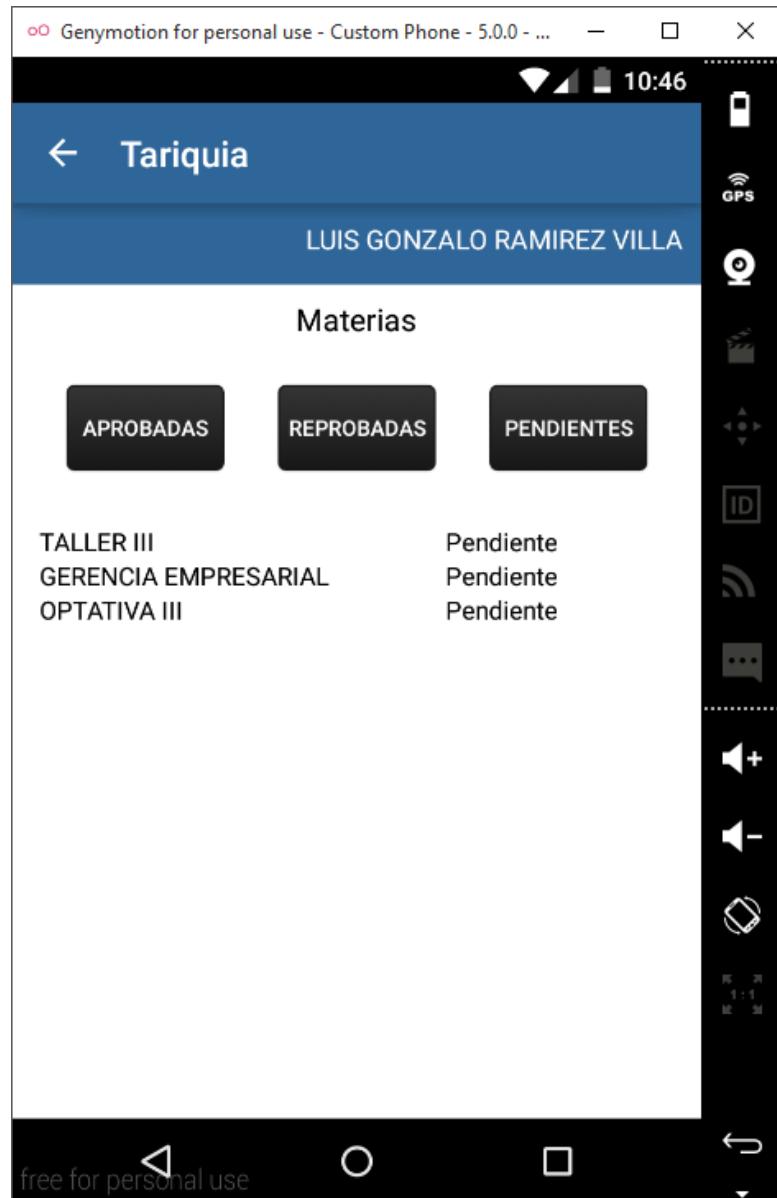


FIGURA 48: PANTALLA MATERIAS PENDIENTES

En esta pantalla se observan las materias que el estudiante aún debe aprobar.

II.1.10.2 DESPLEGAR NOTAS



FIGURA 49: PANTALLA NOTAS POR MATERIA

Al hacer clic en “Notas”, se mostrará el listado de materias que está cursando en la gestión que escoja, de acuerdo a la programación realizada.



FIGURA 50: PANTALLA NOTAS DE MATERIA

Al seleccionar alguna materia de la Pantalla “Notas por Materia” se procede a visualizar las notas de la materia determinada en ponderación Regular y Mesas.

II.1.10.3 DESPLEGAR HORARIO

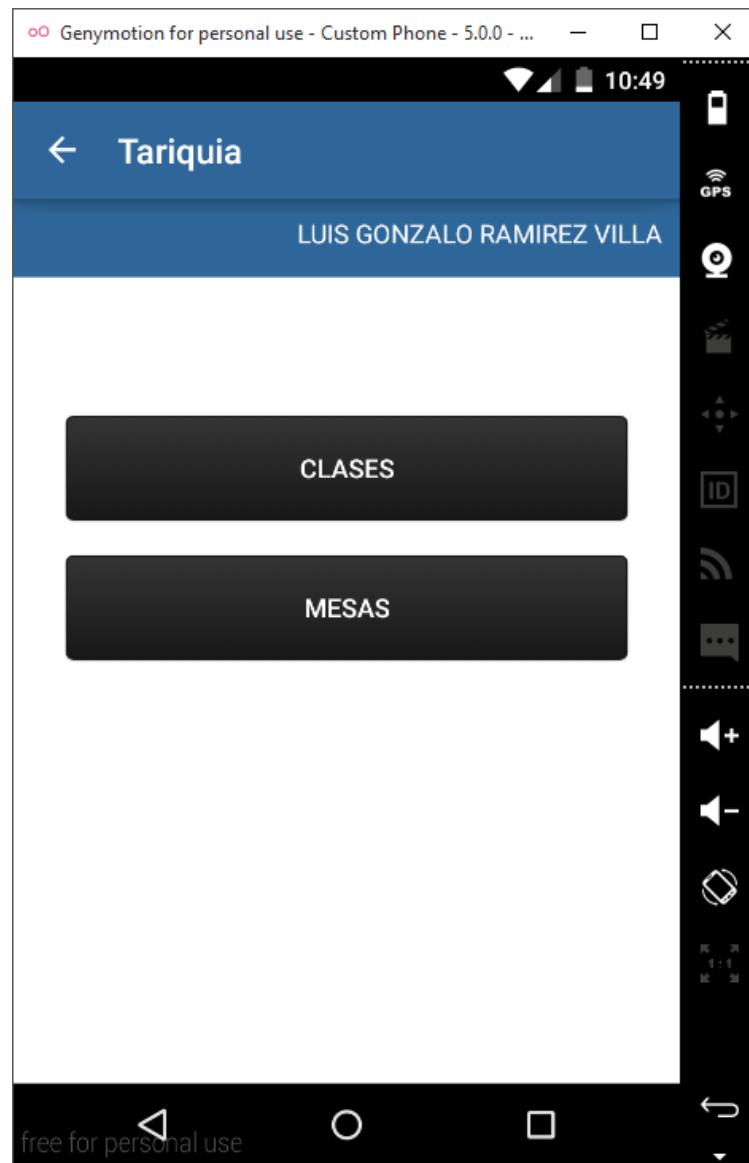


FIGURA 51: PANTALLA DESPLEGAR HORARIO

Al hacer clic en “Horarios”, se mostrará las opciones de “Horario de Clases”, “Horario de Mesas”.

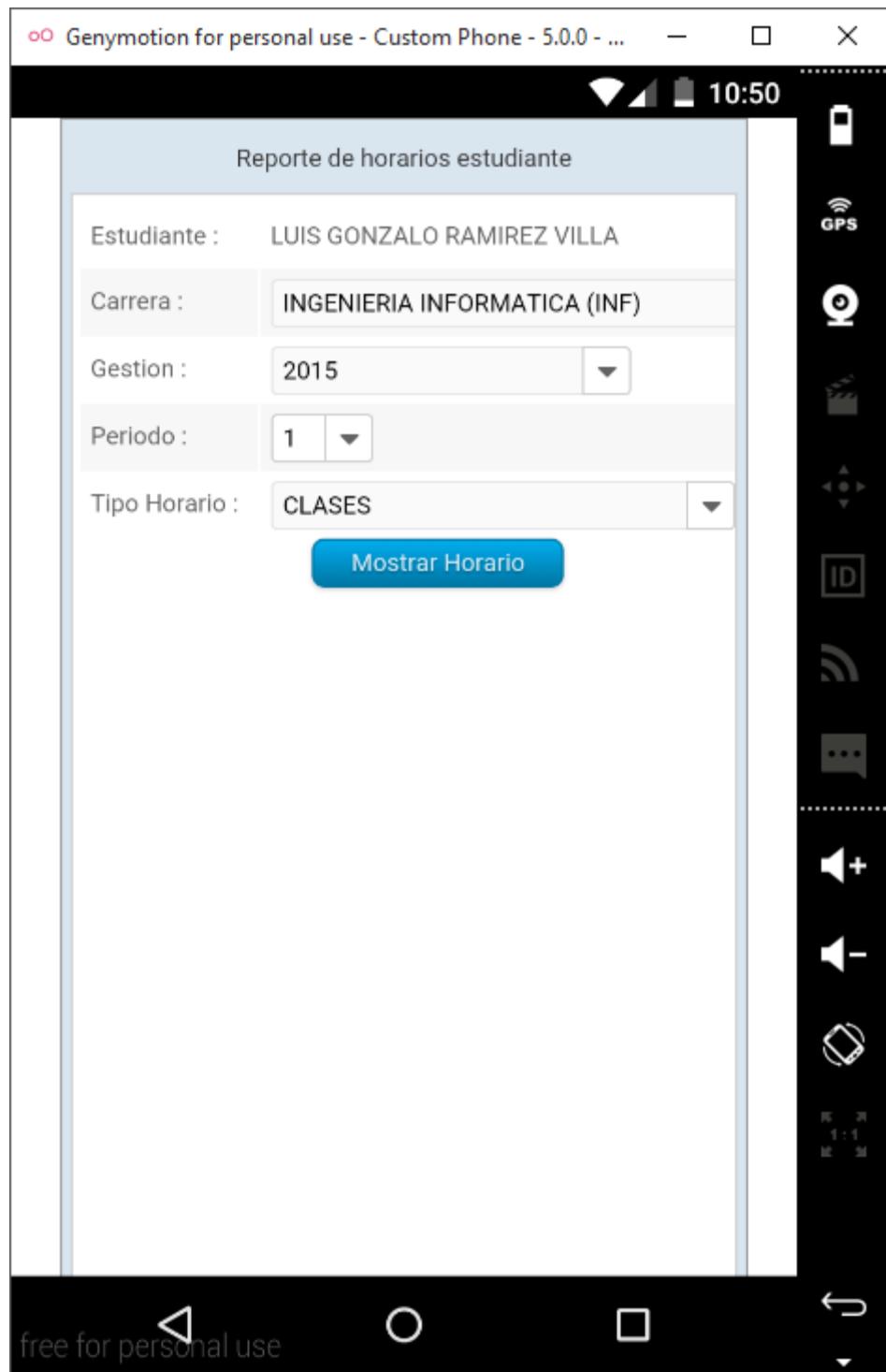


FIGURA 52: PANTALLA HORARIO DE CLASES

En esta pantalla se procede a seleccionar la Gestión, el Período y el Tipo de Horario para descargar el horario, al hacer clic en “Mostrar Horario”.

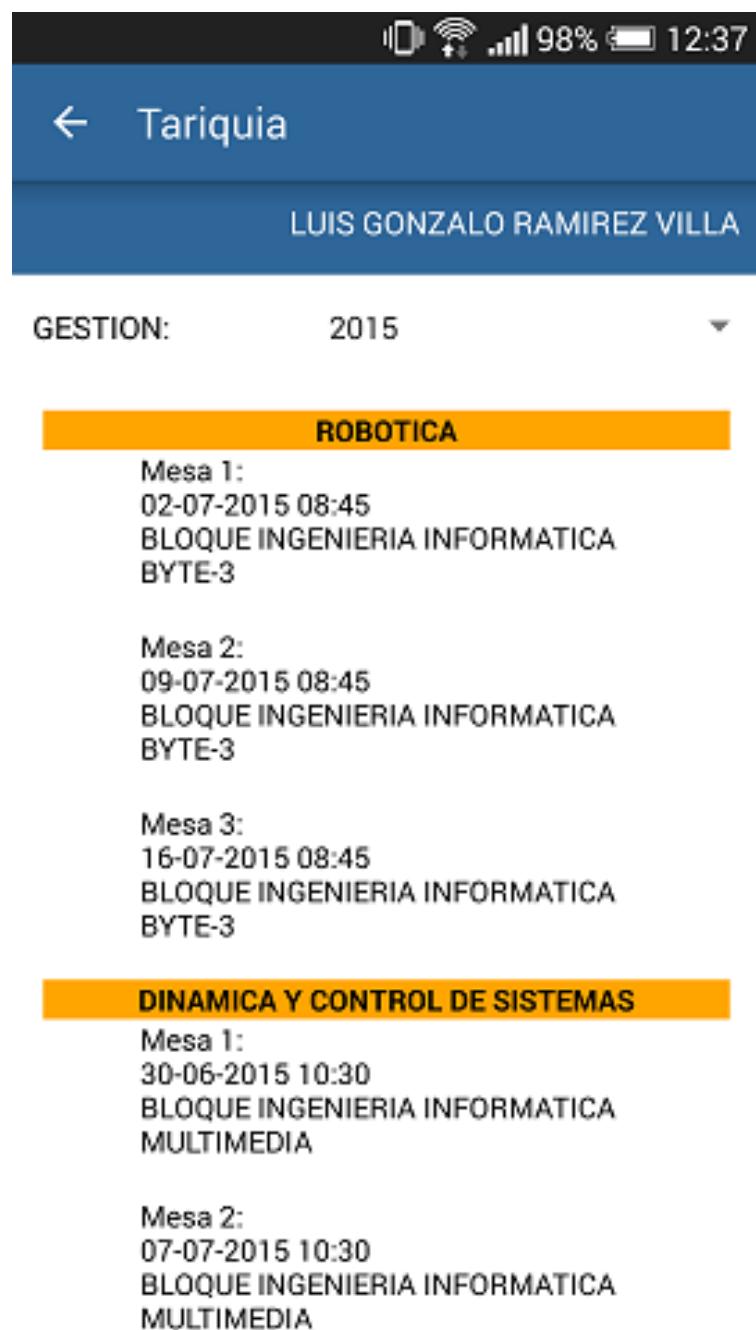


FIGURA 53: PANTALLA HORARIO DE MESAS

Se cuenta con un filtro de Gestión para visualizar las mesas, donde se obtienen los datos de fecha, hora y lugar de las diferentes materias programadas en la gestión.

II.1.10.4 DESPLEGAR ACTIVIDADES

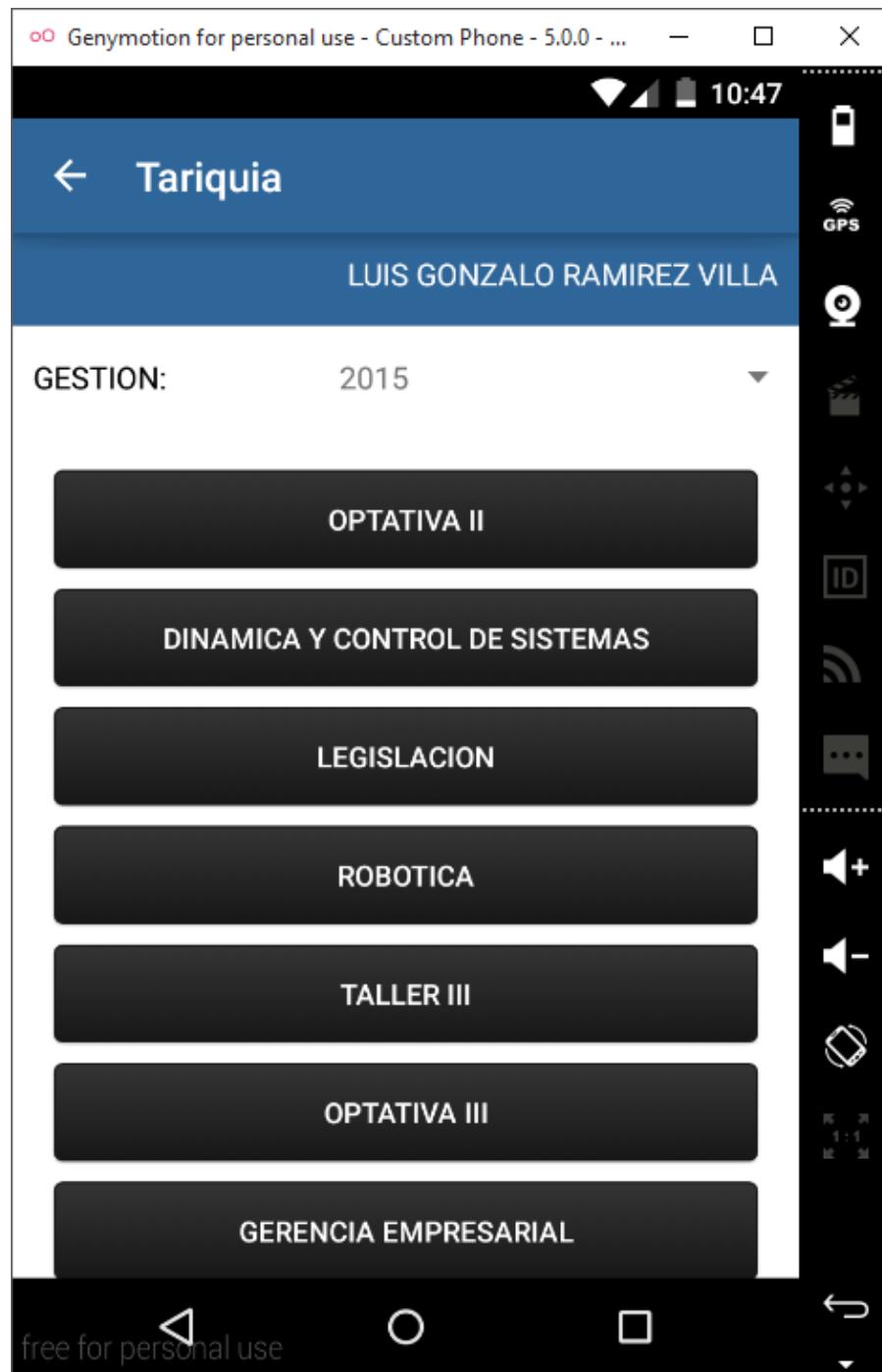


FIGURA 54: PANTALLA ACTIVIDADES

Se cuenta con un filtro de Gestión para visualizar las materias programadas, al seleccionar la materia accederá a las diferentes actividades cargadas por el docente.

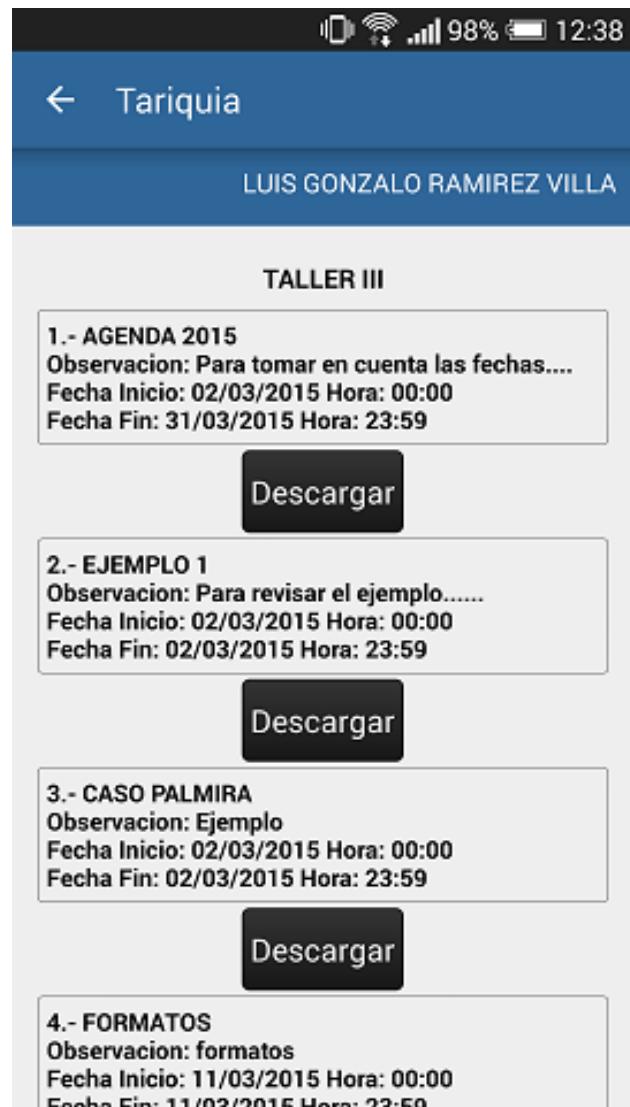


FIGURA 55: PANTALLA ACTIVIDADES POR MATERIA

En esta pantalla se puede observar las diferentes actividades que el docente habilitó por materia.

II.1.10.5 DESPLEGAR CALENDARIO

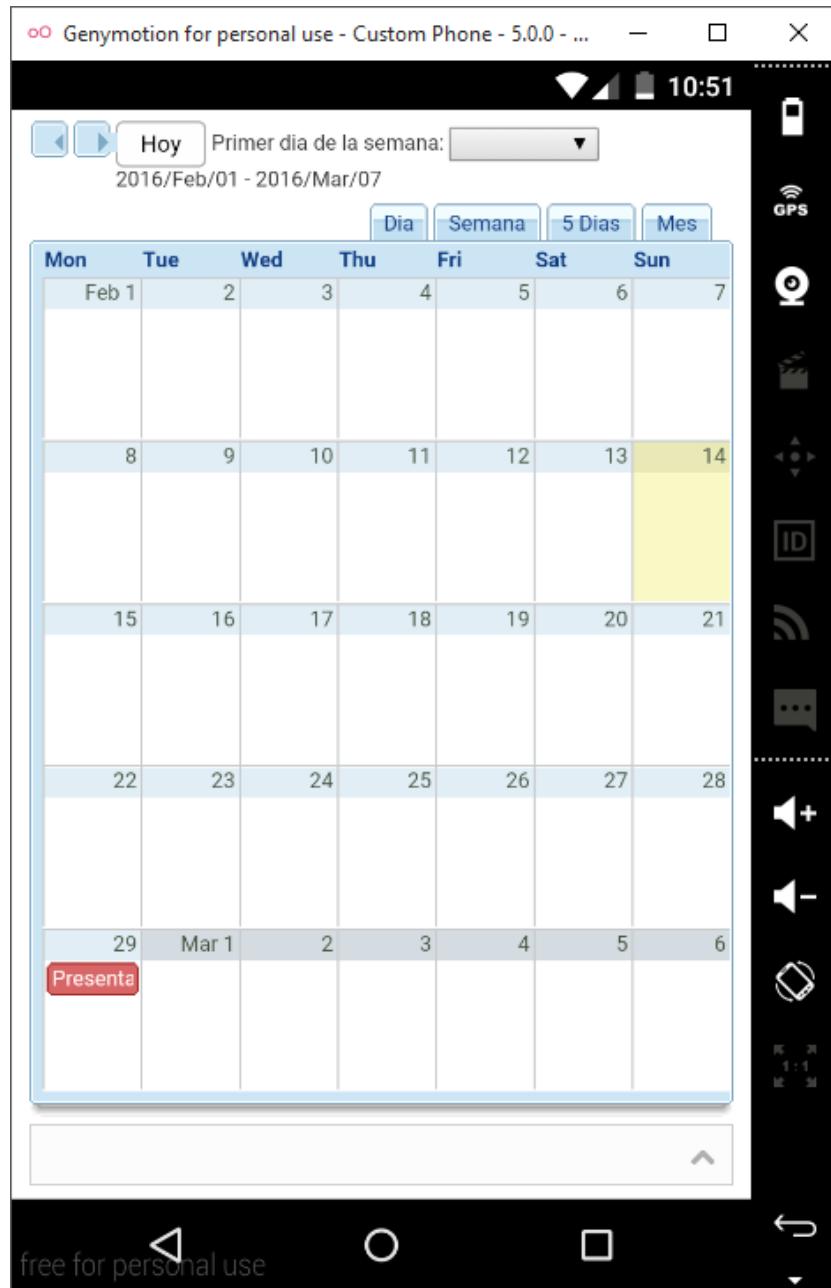


FIGURA 56: PANTALLA CALENDARIO

Al hacer clic en “Calendario”, se mostrará un calendario donde el estudiante podrá ingresar contenido sobre una actividad que realizará.

II.1.10.6 GENERAR RENDIMIENTO ACADÉMICO

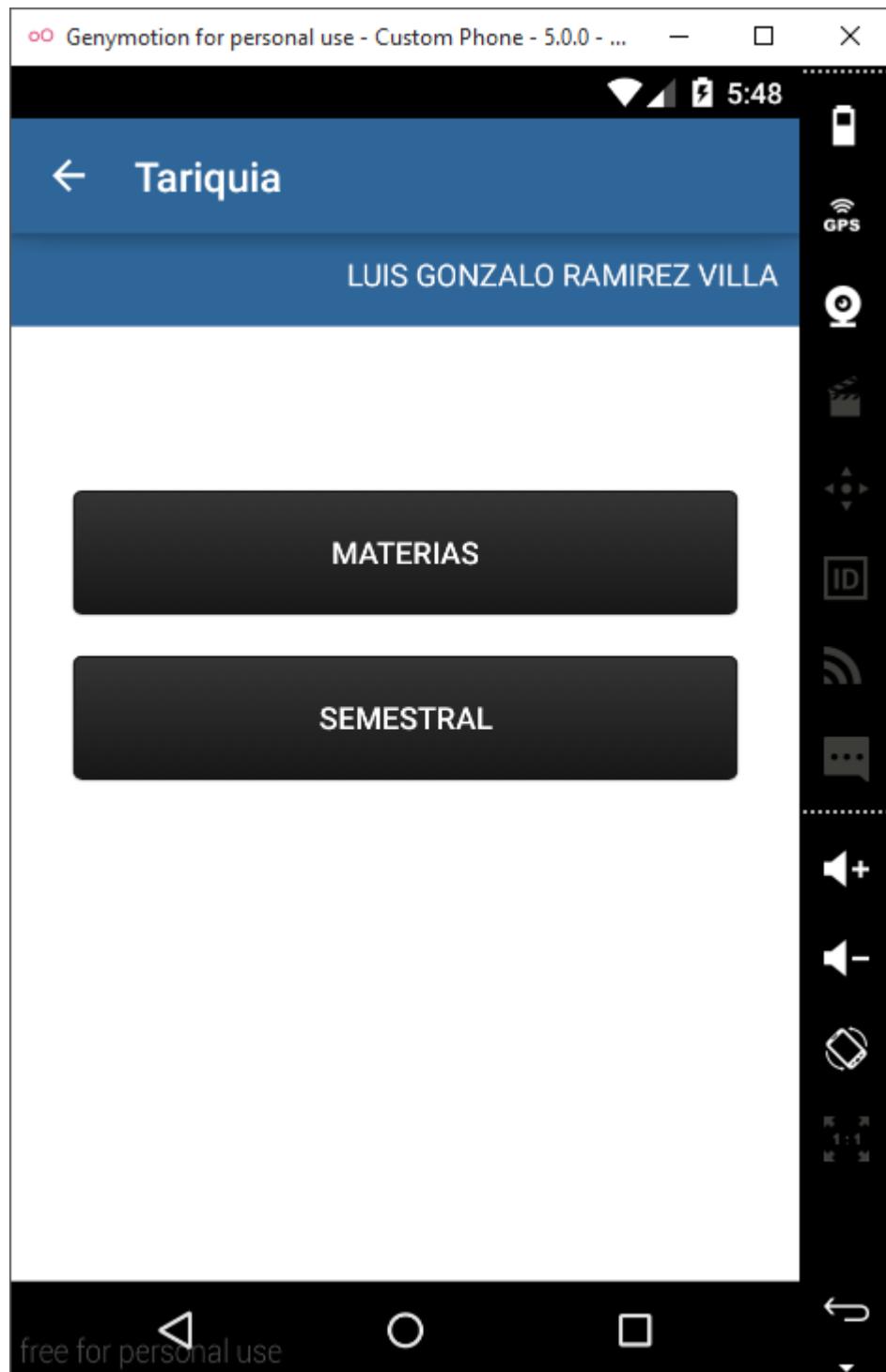


FIGURA 57: PANTALLA RENDIMIENTO ACADÉMICO

Al hacer clic en “Rendimiento Académico”, se mostrará las opciones de “Rendimiento por Semestre”, “Rendimiento por Materias”.

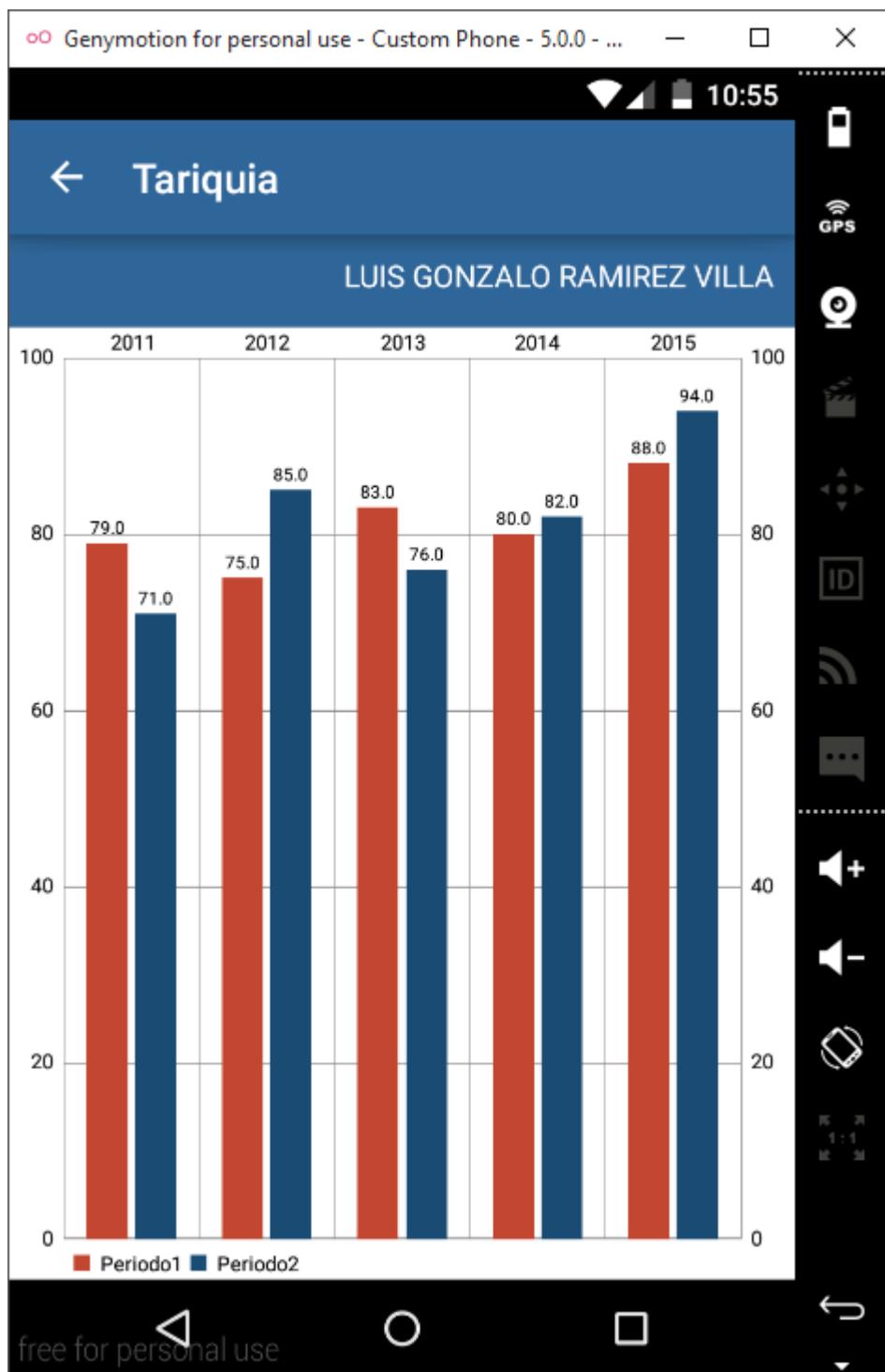


FIGURA 58: PANTALLA RENDIMIENTO POR SEMESTRE

En esta pantalla se observa un gráfico del rendimiento académico por semestre.

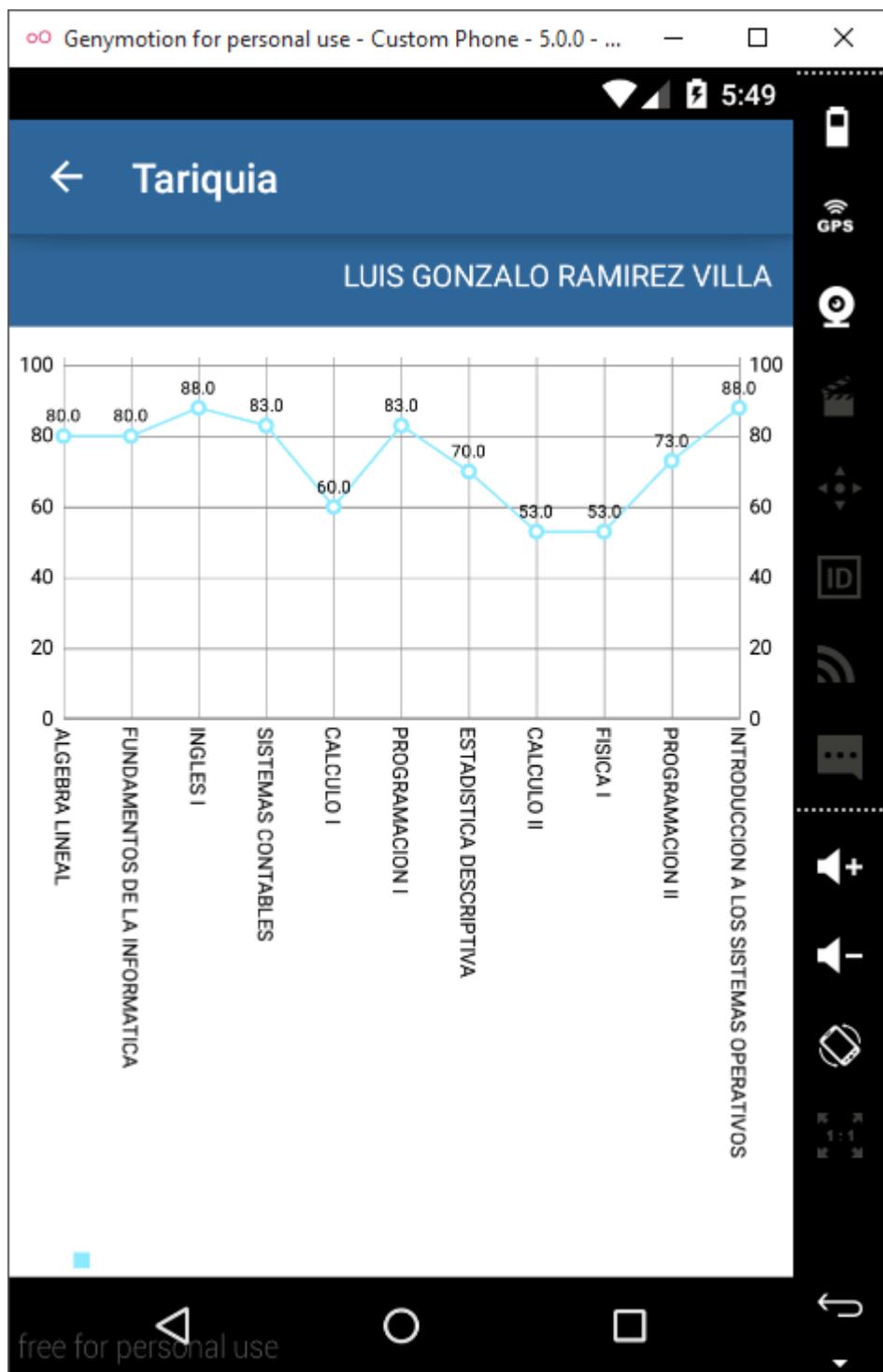


FIGURA 59: PANTALLA RENDIMIENTO POR MATERIAS

En esta pantalla se observa un gráfico del rendimiento académico por materias.

Modelado De Componentes

II.1.11 MODELADO DE COMPONENTES

II.1.11.1 INTRODUCCIÓN

El Modelado de Componentes ilustra los componentes de software que se usarán para construir la aplicación.

II.1.11.2 PROPÓSITO

- Comprender los componentes de la aplicación.
- Identificar los diferentes componentes que integrarán a la aplicación.

II.1.11.3 ALCANCE

- Identificar y definir relaciones entre los diferentes componentes.
- Visualizar de manera gráfica los componentes de la aplicación.

II.1.11.4 MODELO DE COMPONENTES

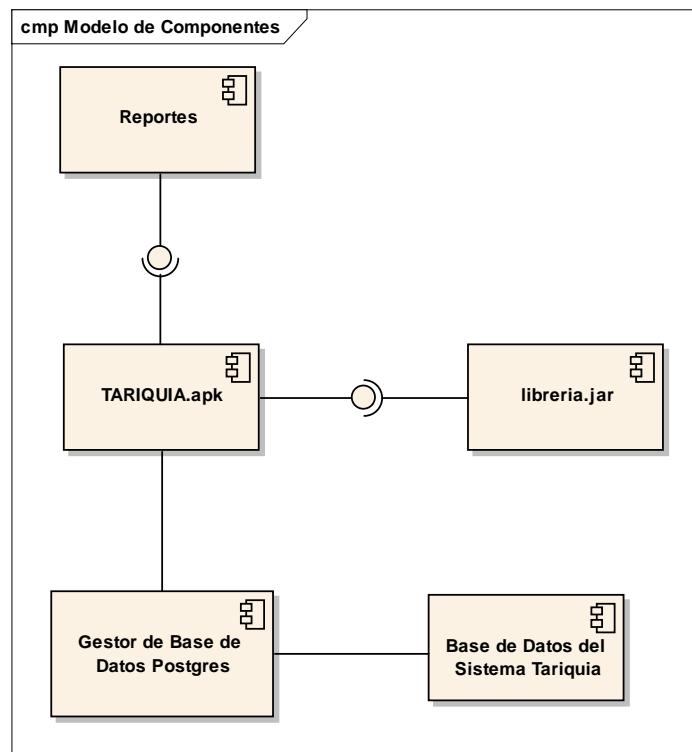


FIGURA 60: MODELO DE COMPONENTES

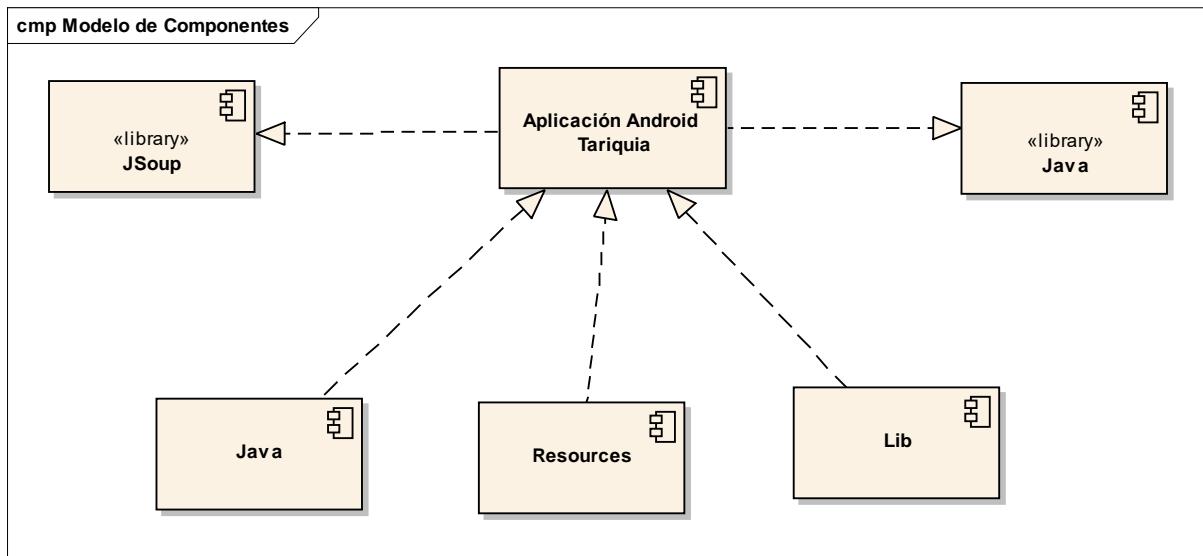


FIGURA 61: COMPONENTES DE LA APLICACIÓN

Modelo de Despliegue

II.1.12 MODELO DE DESPLIEGUE

II.1.12.1 INTRODUCCIÓN

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar la disposición física de los artefactos software en nodos.

II.1.12.2 PROPÓSITO

- Comprender las estructura de los componentes que utilizará la aplicación.
- Identificar los diferentes artefactos involucrados en la aplicación.

II.1.12.3 ALCANCE

- Describir la arquitectura a nivel de especificación de la aplicación.
- Visualizar la gráfica de los artefactos que se encuentran relacionados para el funcionamiento de la aplicación.

II.1.12.4 DIAGRAMA DE DESPLIEGUE

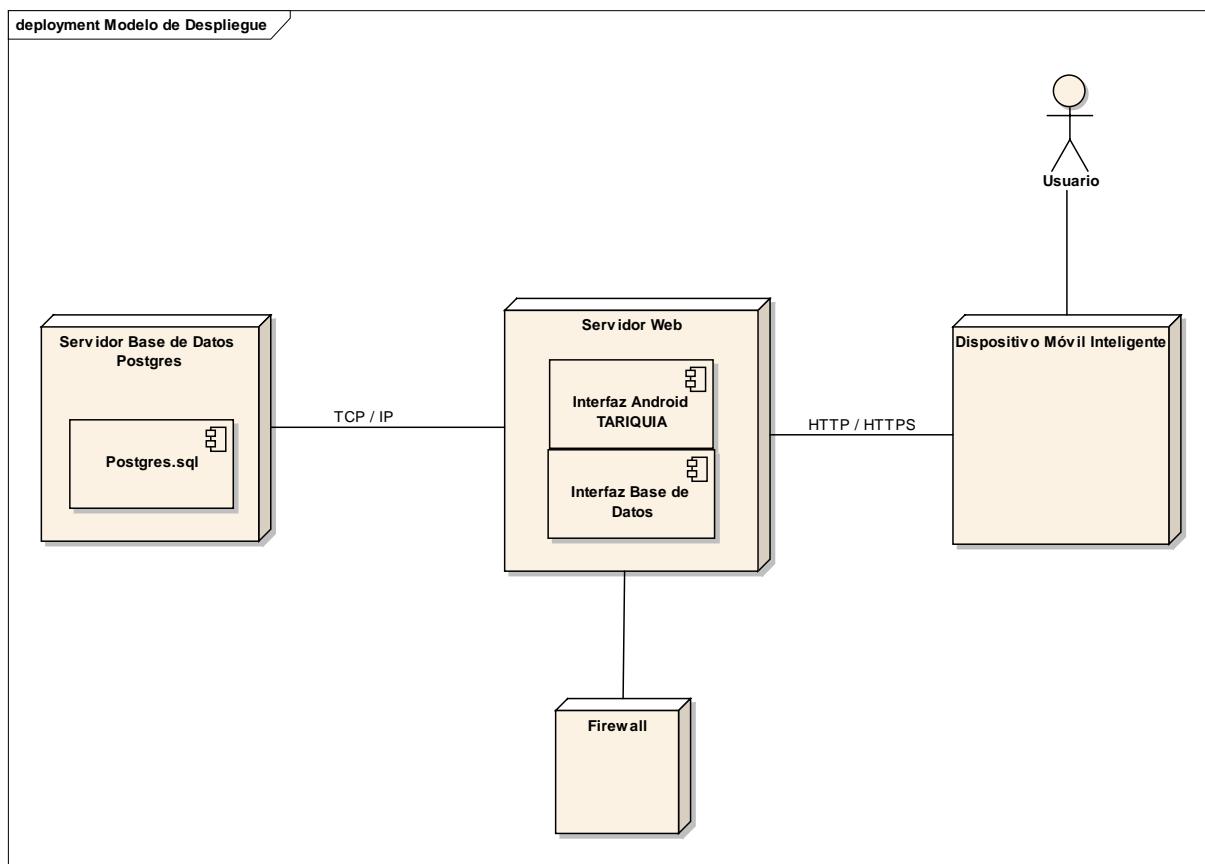


FIGURA 62: DIAGRAMA DE DESPLIEGUE

Casos de Prueba

II.1.13 CASOS DE PRUEBA DE LA APLICACIÓN

Para las pruebas de la aplicación, se empleó las pruebas de caja blanca y caja negra; en el caso de las pruebas de caja blanca se utilizó la herramienta de complemento PMD, la misma que facilita la obtención del grafo de flujo. Por otra parte en las pruebas de caja negra se utilizó la clase de equivalencias válidas y no válidas.

II.1.13.1 PRUEBAS DE CAJA BLANCA

II.1.13.1.1 NOTICIAS

noticias.java – método main

a) CÓDIGO

```
package jsoupPractice;
import java.io.IOException;
import org.jsoup.Connection;
import org.jsoup.Connection.Method;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

public class noticias {
    public static void main(String[] args) {
        String urlLogin="https://tariquia.uajms.edu.bo/tariquia/login.do";
        String
urlFichaAcademica="https://tariquia.uajms.edu.bo/tariquia/buscarProgramasEstudiante.do";
        Document doc;
        String usuario="";
        try {
            Connection.Response res =
Jsoup.connect("http://www.uajms.edu.bo/feed/")
```

```

.userAgent("Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101
Firefox/39.0")

.method(Method.GET)

.execute();

    doc=res.parse();

    System.out.println(doc.select("title"));

    System.out.println(doc.select("description"));

} catch (IOException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

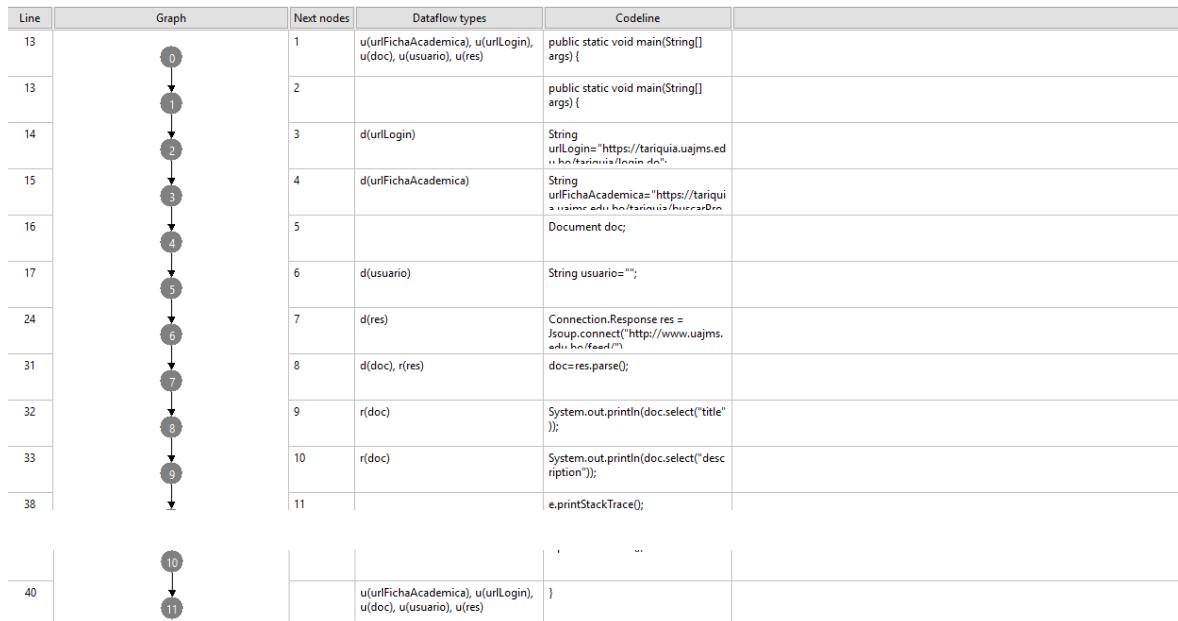
}

}

}

```

b) GRAFO DE FLUJO



**FIGURA 63: GRAFO DE FLUJO NOTICIAS
COMPLEJIDAD CICLOMÁTICA**

$$V(G) = C + 1 = 0 + 1 = 1$$

CAMINOS INDEPENDIENTES

Camino 1.- {0-1-2-3-4-5-6-7-8-9-10-11}

II.1.13.1.2 FICHA ACADÉMICA

fichacedemica.java – método main

a) CÓDIGO

```
package jsoupPractice;
import java.io.IOException;
import org.jsoup.Connection;
import org.jsoup.Connection.Method;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
public class fichacedemica {
    public static void main(String[] args) {
        String urlLogin="https://tariquia.uajms.edu.bo/tariquia/login.do";
        String
urlFichaAcademica="https://tariquia.uajms.edu.bo/tariquia/buscarProgramasEstudiante.do";
        Document doc;
        String usuario="";
        try {
/*doc=Jsoup.connect("https://tariquia.uajms.edu.bo/tariquia/principal.do")
.validateTLCertificates(false)
.get();
System.out.println(doc.title());*/
        Connection.Response res = Jsoup.connect(urlLogin)
```

```

//.data("id_usuario", id_usuario.getText().toString(), "clave",
clave.getText().toString())
    .data("id_usuario", "e69956", "clave", "JRcJ/704")
//.data("id_usuario", "e69835", "clave", "NUAz+466")
//.data("id_usuario", "e70359", "clave", "ftuy*523")
.userAgent("Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101
Firefox/39.0")
.validateTLCertificates(false)
.method(Method.POST)
.execute();

String sessionId = res.cookie("JSESSIONID");
System.out.println(sessionId);
System.out.println(res.url());
doc=res.parse();
    System.out.println(doc.select("div.usuario").text());
    System.out.println(res.statusCode());

/*doc=Jsoup.connect(urlFichaAcademica)
.cookie("JSESSIONID", sessionId)
.cookie("JSESSIONID", "D87E6419F2527AAB7FA6730664D95092.tomcat1")
.validateTLCertificates(false)
.get();
Elements divUsuario=doc.select("div.titulo");

for (Element divUsu : divUsuario) {
    System.out.println("titulo: "+divUsu.text());
}*/

//PARA SACAR EL HTML DE LA FICHA ACADEMICA
System.out.println("Obteniendo el historial academico");

```

```

Connection.Response res0=Jsoup.connect(urlFichaAcademica)
    .data("clave","JRcJ/704")
    .userAgent("Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0)
Gecko/20100101 Firefox/39.0")
    .validateTLCertificates(false)
    .cookie("JSESSIONID", sessionId)
    .method(Method.POST)
    .execute();

System.out.println(res0.statusCode());
System.out.println(res0.url());

doc=Jsoup.connect(res0.url().toString())
    .cookie("JSESSIONID", sessionId)
    .validateTLCertificates(false)
    .get();
Element divUsuario=doc.select("b h3 b").first();
System.out.println(divUsuario.text());
System.out.println("cerrando sesion");
Connection.Response
res1=Jsoup.connect("https://tariquia.uajms.edu.bo/tariquia/logout.do")
    .userAgent("Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0)
Gecko/20100101 Firefox/39.0")
    .validateTLCertificates(false)
    .cookie("JSESSIONID", sessionId)
    .method(Method.GET)
    .execute();

System.out.println(res1.statusCode());
res1.removeCookie("JSESSIONID");
System.out.println("Sesion cerrada");

} catch (IOException e) {

```

```

    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

```

b) GRAFO DE FLUJO

Line	Graph	Next nodes	Dataflow types	Codeline
13		1	u(res), u(divUsuario), u(res1), u(res0), u(sessionId), u(urlFichaAcademica), u(doc), u(doc1)	public static void main(String[] args) {
13		2		public static void main(String[] args) {
14		3	d(urlLogin)	String urlLogin = "https://tariquia.uajms.edu/tarquinia/admin/";
15		4	d(urlFichaAcademica)	String urlFichaAcademica = "https://tariquia.uajms.edu/tarquinia/admin/fichaAcademica";
16		5		Document doc;
17		6	d(usuario)	String usuario = "";
24		7	d(res), r(urlLogin)	Connection.Response res = Jsoup.connect(urlLogin)
35		8	r(res), d(sessionId)	String sessionId = res.cookie("JSESSIONID");
36		9	r(sessionId)	System.out.println(sessionId);
37		10	r(res)	System.out.println(res.url());
38		11	r(res), d(doc)	doc = res.parse();
39		12	r(doc)	System.out.println(doc.select("div.usuario").text());
40		13	r(res)	System.out.println(res.statusCode());
54		14		System.out.println("Obteniendo el historial académico");
55		15	d(res0), r(sessionId), r(urlFichaAcademica)	Connection.Response res0 = Jsoup.connect(urlFichaAcademica);
62		16	r(res0)	System.out.println(res0.statusCode());
63		17	r(res0)	System.out.println(res0.url());
65		18	r(res0), r(sessionId), d(doc)	doc = Jsoup.connect(res0.url().toString())
69		19	d(divUsuario), r(doc)	Element divUsuario = doc.select("b h3 b").first();
70		20	r(divUsuario)	System.out.println(divUsuario.text());
73		21		System.out.println("cerrando sesión");
74		22	d(res1), r(sessionId)	Connection.Response res1 = Jsoup.connect("https://tariquia.uajms.edu/tarquinia/admin/logout");
81		23	r(res1)	System.out.println(res1.statusCode());
82		24	r(res1)	res1.removeCookie("JSESSIONID");
83		25		System.out.println("Sesión cerrada");
87		26		e.printStackTrace();
89			u(res), u(divUsuario), u(res1), u(res0), u(sessionId), u(urlFichaAcademica), u(doc), u(doc1)	}

FIGURA 64: GRAFO DE FLUJO FICHA ACADÉMICA

COMPLEJIDAD CICLOMÁTICA

$$V(G) = C + 1 = 0 + 1 = 1$$

CAMINOS INDEPENDIENTES

Camino 1.- {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26}

II.1.13.2 PRUEBAS DE CAJA NEGRA

II.1.13.2.1 INGRESO A LA APLICACIÓN

Pantalla: “Ingreso de la aplicación”

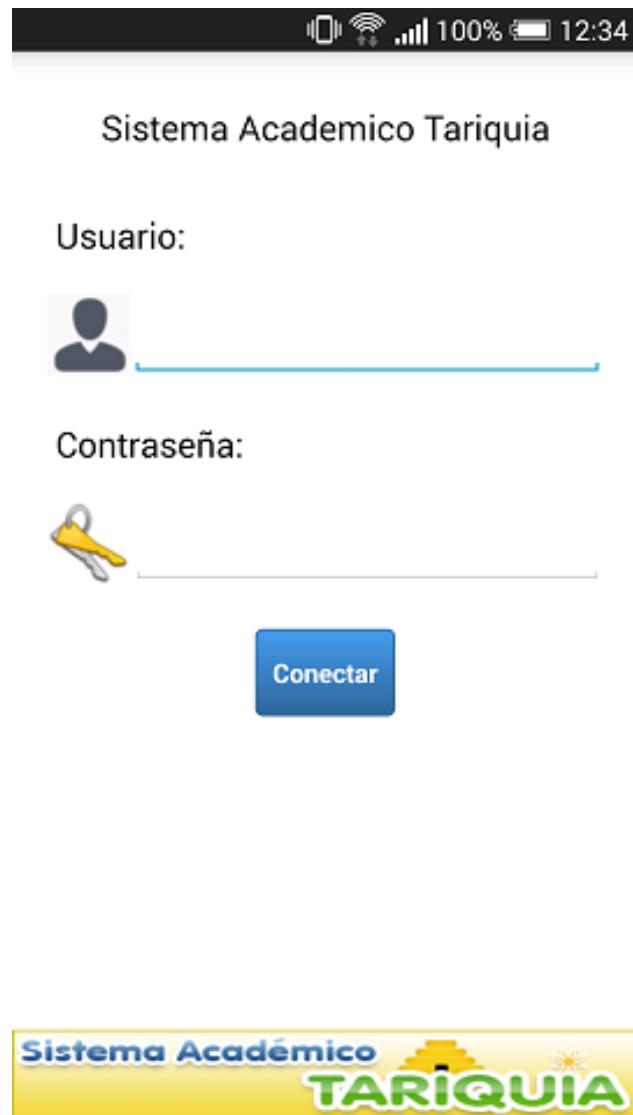


FIGURA 65: PANTALLA INGRESO DE LA APLICACIÓN

Clase de Equivalencias Válidas e Inválidas

Condición de entrada	Tipo	Clases de Equivalencia Válidas	Clases de Equivalencia Inválida
Usuario	Valor	1.- Cadena con valores alfanuméricos	2.- En blanco 3.- Cadena no registrada en la base
Contraseña	Valor	4.- Cadena con valores alfanuméricos	5.- En blanco 6.- Cadena no registrada en la base

TABLA 32: CLASES DE EQUIVALENCIAS VÁLIDAS E INVÁLIDAS – INGRESO DE LA APLICACIÓN

Casos de Prueba - Pantalla: “Ingreso de la Aplicación”

Nº	Clase de Equivalencia	Propósito del Caso	Dato de Prueba	Mensaje
1	Cadena con valores alfanuméricos	Función correcta	abc123	
2	En blanco	Función incorrecta	””	Debe llenar los campos!!!
3	Cadena con valores caracteres especiales	Función incorrecta	abc%123	Usuario y/o Contraseña Incorrectos
4	Cadena con valores alfanuméricos	Función correcta	123a	
5	En blanco	Función incorrecta	””	Debe llenar los campos!!!
6	Cadena con valores caracteres especiales	Función incorrecta	321%\$%123	Usuario y/o Contraseña Incorrectos

TABLA 33: CASOS DE PRUEBA – INGRESO DE LA APLICACIÓN

II.1.12.2.2 NOTAS POR MATERIA

Pantalla: “Notas por Materia”



FIGURA 66: PANTALLA NOTAS POR MATERIA

Clase de Equivalencias Válidas e Inválidas

Condición de entrada	Tipo	Clases de Equivalencia Válidas	Clases de Equivalencia Inválida
Gestión	Conjunto	1.- Selecciona la gestión a visualizar (Ej. 2015)	2.- Gestión = Sin selección

TABLA 34: CLASES DE EQUIVALENCIAS VÁLIDAS E INVÁLIDAS – NOTAS POR MATERIA

Casos de Prueba - Pantalla: “Notas por Materia”

Nº	Clase de Equivalencia	Propósito del Caso	Dato de Prueba	Mensaje
4	Selecciona la gestión a visualizar (Ej. 2015)	Función correcta	2015 = Visualizar materias programadas sólo en la gestión 2015	
5	Gestión = Sin selección	Función incorrecta		Sin Filtro, no visualiza

TABLA 35: CASO DE PRUEBA – NOTAS POR MATERIA

Pantalla: “Notas de Materia”



FIGURA 67: PANTALLA NOTAS DE MATERIA

Se visualiza los datos de la materia de acuerdo a los parciales y mesas.

II.1.14 MEDIOS DE VERIFICACIÓN COMPONENTE I

En este componente se tienen los siguientes medios de verificación:

- La Documentación del Proyecto con la mayor completitud posible.
- Certificado de Corrección del Proyecto.
- Certificado del Tutor Guía del Proyecto.
- Certificado de Aprobación del Desarrollo del Proyecto y presupuesto por parte del encargado de sistemas de información del DTIC de la UAJMS.
- Resumen de la entrevista realizada al personal de la DTIC (Anexo 3).
- El Modelo de Cuestionario (Anexo 4).

II.2 COMPONENTE 2: SOCIALIZACIÓN DEL PROYECTO

II.2.1 CAPACITACIÓN AL PERSONAL ENCARGADO DE LA ADMINISTRACIÓN Y MANTENIMIENTO DE LA APLICACIÓN

II.2.1.1 INTRODUCCIÓN

De acuerdo a las necesidades actuales del mundo laboral y profesional, los términos de Capacitación y Sistemas de Información están modificando la forma de trabajo de las empresas o instituciones; los sistemas de información ayudan a acelerar los procesos, por tanto, las organizaciones que los adoptan en sus funciones logran ventajas competitivas respecto a las demás.

La aplicación de las nuevas tecnologías informáticas en la educación de todas las áreas del conocimiento ha producido cambios notables en el proceso de enseñanza. En este contexto, la Educación a Distancia (e-learning) a través de Internet es en la actualidad una herramienta de formación y capacitación sumamente reconocida en todo el mundo.

La capacitación se refiere a las metodologías que se usan para proporcionar a las personas dentro de la institución, las habilidades precisas que necesitan para realizar su trabajo de una manera más eficaz, esto contempla desde pequeños cursos que le permitan al usuario entender el funcionamiento básico de la aplicación hasta la capacitación más profunda y avanzada a bases de prácticas y material didáctico entre otros elementos.

II.2.1.1.1 PROPÓSITO

Proporcionar a los usuarios (estudiantes) de la Universidad Autónoma “Juan Misael Saracho” una guía importante en el uso de la aplicación móvil, la importancia de su utilización y el uso correcto de la misma.

II.2.1.1.2 OBJETIVO

Capacitar a los estudiantes en el manejo de la aplicación y exponer funcionalidades que ésta ofrece.

II.2.1.2 ALCANCES Y LIMITACIONES

II.2.1.2.1 ALCANCES

La Capacitación se realizará de forma virtual, estará dirigida a estudiantes de la UAJMS, donde se realizará una explicación de las funciones de la aplicación. Logrando abarcar un mayor número de estudiantes.

II.2.1.2.2 LIMITACIONES

Sólo podrán capacitarse los estudiantes que accedan al video alojado en el sitio web para subir y compartir videos conocido como Youtube.

II.2.1.3 PROPUESTA DE CAPACITACIÓN

II.2.1.3.1 MÉTODOS DE CAPACITACIÓN UTILIZADOS

Considerando que los estudiantes cuentan con poco tiempo de disponibilidad y tienen algunos conocimientos en el manejo de las TIC's, en la realización de la capacitación acerca del uso de la Aplicación se empleará la técnica conocida como *capacitación a distancia*. Esta técnica contempla que una persona aprenda sobre un tema, de acuerdo a su disponibilidad de tiempo y logrando repetir las veces que desee el tema hasta poder asimilarlo.

La capacitación a distancia⁵ es una alternativa de enseñanza para todos aquellos que no pueden participar del sistema educativo tradicional permiten reunir personas en diferentes localidades para participar, simultáneamente, en un mismo programa. Son transmitidas de un punto para múltiples puntos. Este método de enseñanza se lleva a cabo a través de las telecomunicaciones requiere de dos aspectos importantes:

- El aspecto tecnológico: medios de transmisión de información adecuados para cubrir distancias.

⁵ Fuente: <http://capacitacionalpersonal.blogspot.com/2010/03/capacitacion-distancia.html>

- El aspecto del aprendizaje: medios atinados de coordinación e interacción.

Las principales ventajas⁶ de la capacitación a distancia son:

- Garantiza la transmisión del mensaje adecuado utilizando el mejor recurso humano disponible.
- Lleva el mensaje a un mayor número de personas en menos tiempo y sin necesidad de movilizarles de su sitio de trabajo.
- Permite la integración de recursos audiovisuales efectivos y modernos.
- Agiliza la comunicación efectiva proveyendo un medio para mantener a los usuarios informados sobre lo último que acontece.
- Ahorran tiempo y esfuerzo en la difusión de los conocimientos más actualizados.
- Si se utilizan adecuadamente pueden proveer una educación y capacitación adecuadas.

II.2.1.3.2 PLAN DE CAPACITACIÓN

El contenido de la capacitación estará enfocado en el estudiante.

Usuario: Estudiantes

Se explica paso a paso las opciones a las que puede acceder el estudiante, en cuanto a las materias, horarios de clases, actividades y su rendimiento académico.

II.2.1.3.3 MATERIAL DE CAPACITACIÓN

Los materiales didácticos que se utilizarán son los siguientes:

- Manual de Usuario.

⁶ Fuente: <http://www.mediosmedios.com.ar/Ventajas.html>

- Computadora.
- Video (Subido a YouTube).
- Folleto de Navegabilidad de la Aplicación (Compartido en Dropbox)

II.2.1.3.4 CONCLUSIONES

Con esta capacitación se busca que los estudiantes puedan ser capacitados de forma correcta de manera que sea posible cumplir los objetivos del proyecto. Se pretende socializar el proyecto con el propósito de que puedan apreciar las funcionalidades de la aplicación móvil.

II.2.2 MEDIOS DE VERIFICACIÓN COMPONENTE II

Este componente tiene los siguientes medios de verificación:

- El contenido de la capacitación que se ha planteado.
- Vídeo de la Capacitación (YouTube).
- Folleto de Navegabilidad de la Aplicación (DropBox).
- Manual del Usuario, indicando el uso de la aplicación.

CAPÍTULO III

CONCLUSIONES

Y

RECOMENDACIONES

III.1 CONCLUSIONES Y RECOMENDACIONES

III.1.1 CONCLUSIONES

- De acuerdo a las encuestas realizadas a los estudiantes de la Universidad Autónoma Juan Misael Saracho, se diagnosticó que existían diversas dificultades en la forma de interactuar con el sistema Tariquia web en sus dispositivos móviles.
- Referente al objetivo, se concluye que se obtuvo la información necesaria para realizar el desarrollo e implementación de la solución que esté acorde a las necesidades de los estudiantes de la Universidad Autónoma Juan Misael Saracho acordes a la metodología empleada.
- Se desarrolló una aplicación móvil bajo tecnología Android, aprovechando las últimas tendencias de la sociedad actual, teniendo como base los requisitos analizados con la metodología RUP empleada, brindando así una solución a los problemas de adaptabilidad y uso para los dispositivos móviles inteligentes de los estudiantes. Además de incluir un apartado innovador que son los reportes de rendimiento académicos que facilitan el control del aprovechamiento de los estudiantes.
- En la socialización del proyecto se plantea el uso de material multimedia que facilitará el aprendizaje de las distintas funcionalidades de la aplicación.

III.1.2 RECOMENDACIONES

A la culminación del presente proyecto se presentan las siguientes recomendaciones:

- Debido a las tendencias sociales actuales en el uso de dispositivos móviles, se recomienda hacer énfasis en potenciar el uso de los dispositivos móviles en la universidad, ya que cada día que pasa los estudiantes prefieren usar más sus dispositivos móviles para consultar y manejar su información, que una computadora de escritorio como lo era años atrás. Para esto se debe tomar en cuenta la adaptabilidad del contenido para estos dispositivos o aplicaciones dedicadas a estas tecnologías para aprovechar sus características de portabilidad y conexión a redes.

- Se recomienda la creación de un equipo de desarrollo encargado de administrar los contenidos y servicios disponibles para los dispositivos móviles.
- Así mismo también se recomienda contar con equipos de hardware dedicados a brindar los servicios para dispositivos móviles de esta manera esta será una plataforma totalmente independiente de los actuales equipos utilizados para el sistema web.
- Se recomienda realizar el mantenimiento y actualización de la aplicación Android y de los dispositivos de hardware en los que se ejecutará el sistema para garantizar un buen funcionamiento.
- Se recomienda contemplar la posibilidad de expandir las funcionalidades de la aplicación.
- Se recomienda capacitar a todo el personal en el uso de tecnología móvil con una orientación a IoT (Internet de las cosas).