

UNIVERSIDAD AUTÓNOMA JUAN MISAEL SARACHO
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA INFORMÁTICA

**“MEJORAMIENTO DEL SERVICIO DE DONACIONES MÓVILES EN EL
BANCO DE SANGRE - TARIJA”**

Por:

Ximena Catari Sarmiento

Trabajo de Grado presentado a consideración de la **“UNIVERSIDAD AUTÓNOMA
JUAN MISAEL SARACHO”**, como requisito para optar el grado académico de
Licenciatura en Ingeniería Informática.

Diciembre de 2012

TARIJA – BOLIVIA

V° B°

.....
Msc. Ing. Luis Alberto Yurquina

DECANO

.....
Msc. Lic. Gustavo Succi

VICEDECANO

APROBADO POR:

TRIBUNAL:

.....
Lic. Silvana Paz

TRIBUNAL ASIGNADO

.....
Lic. Roberto Salinas

TRIBUNAL ASIGNADO

.....
Ing. Efraín Torrejón

TRIBUNAL ASIGNADO

El Tribunal no se solidariza con la forma, términos, modos, y expresiones vertidas en el trabajo, siendo las mismas únicamente de responsabilidad del autor.

DEDICATORIA:

Este trabajo va dedicado a:

Dios, mis padres, familia, amigos que me brindaron su apoyo incondicional en la realización del mismo.

AGRADECIMIENTOS:

Expresar mis sinceros agradecimientos al personal del Banco de Sangre por el apoyo incondicional que me prestaron para la elaboración de este trabajo, como también a mis Docentes Guías por su tiempo y dedicación prestada.

RESUMEN

En la actualidad los sistemas de información y comunicación vía Web ocupan un sitio importante en las instituciones, empresas, etc. Los cuales facilitan el proceso de difusión de la información, que apoya a la toma de decisiones y las actividades realizadas por los usuarios.

El Banco de Sangre realiza actividades de recolección de sangre dentro y fuera de la institución, el caso de estudio serán las donaciones móviles que no cuentan con un medio o herramienta automatizada de información para las actividades que realizan. Se genera problemas en la difusión de la información y no contar con información organizada y segura lo cual dificulta hacer reportes, modificaciones, actualizaciones a los registros de donación.

Actualmente en países más desarrollados, el uso de las TIC para las necesidades de Bancos de Sangre es muy difundido y exitoso, por ello surge la necesidad de integrarse a esa corriente tecnológica para poder gozar de sus beneficios.

Por lo que el proyecto pretende es dar solución a los problemas, desarrollando un sistema de información para el departamento de Promoción y extensión social BSRDT.

Con el desarrollo del componente Sistema se busca la sencillez, la usabilidad y la rapidez en la gestión de la información, de tal forma que el uso del mismo permita al usuario mantener una gestión ordenada y accesible en todo momento.

Además se complementará con la capacitación al personal de salud en el uso y manejo del componente sistema del proyecto.

PRESENTACION DEL PROYECTO

I.1.1. TÍTULO

“Mejoramiento del Servicio de Donaciones Móviles en el banco de sangre - Tarija”.

I.1.2. ÁREA DEL PROYECTO

Tecnologías de la información y comunicación.

I.1.3. RESPONSABLE DEL PROYECTO

Carrera de Ingeniería Informática – Taller III.

I.1.4. ENTIDADES ASOCIADAS

Banco de Sangre de Referencia Departamental de Tarija – BSRDT.

Universidad Autónoma Juan Misael Saracho.

I.1.5. GRUPO RESPONSABLE DEL PROYECTO

Grupo 1 Taller III:

Universitaria:

Ximena Catari Sarmiento.

I.1.6. DURACIÓN (Meses)

La realización del proyecto tendrá una duración de 8 meses.

I.1.7. ACTIVIDADES PREVISTAS PARA INTEGRANTES DEL EQUIPO

Responsable	Actividades
<p>Director</p> <p>Ximena Catari Sarmiento</p>	<p>En cuanto al seguimiento y control del Proyecto</p> <ul style="list-style-type: none"> - Planificación y control del cronograma del proyecto. - Asignar y gestionar recursos y prioridades a los distintos componentes y actividades del proyecto. - Coordinar las iteraciones entre el equipo de trabajo y los usuarios del proyecto. - Mantener al equipo del proyecto enfocado en los objetivos. - Supervisar el desarrollo del proyecto. <p>En cuanto al desarrollo de los componentes</p> <p>Capturar la especificación y validación de requisitos.</p> <ul style="list-style-type: none"> - Elaboración del Análisis, Diseño y Desarrollo del Sistema. - Elaboración del Modelo de datos. - Delineación de los Diagramas UML. - Diseño de la Base de Datos del Sistema. - Programación del Sistema Web. - Preparación de pruebas Funcionales del Sistema - Planificar el contenido temático de los cursos de capacitación. - Formar al personal en el uso de las TIC para el manejo del producto final. - Llevar a cabo la capacitación del producto final.

Tabla 1 Actividades del integrantes del equipo

I.2. Descripción Del Proyecto

I.2.1. Resumen Ejecutivo Del Proyecto

En la actualidad los sistemas de información y comunicación vía Web ocupan un sitio importante en las instituciones, empresas, etc. Los cuales facilitan el proceso de difusión de la información, que apoya a la toma de decisiones y las actividades realizadas por los usuarios.

El Banco de Sangre realiza actividades de recolección de sangre dentro y fuera de la institución, el caso de estudio serán las donaciones móviles que no cuentan con un medio o herramienta automatizada de información para las actividades que realizan. Se genera problemas en la difusión de la información y no contar con información organizada y segura lo cual dificulta hacer reportes, modificaciones, actualizaciones a los registros de donación.

Actualmente en países más desarrollados, el uso de las TIC para las necesidades de Bancos de Sangre es muy difundido y exitoso, por ello surge la necesidad de integrarse a esa corriente tecnológica para poder gozar de sus beneficios.

Por lo que el proyecto pretende es dar solución a los problemas, desarrollando un sistema de información para el departamento de Promoción y extensión social BSRDT.

Con el desarrollo del componente Sistema se busca la sencillez, la usabilidad y la rapidez en la gestión de la información, de tal forma que el uso del mismo permita al usuario mantener una gestión ordenada y accesible en todo momento.

Además se complementará con la capacitación al personal de salud en el uso y manejo del componente sistema del proyecto

I.2.2. Descripción, Fundamentación y Justificación del Proyecto

El proyecto tiene como propósito el “Mejoramiento del Servicio de Donaciones Móviles en el banco de sangre - Tarija” tiene como finalidad mejorar y/o facilitar las diferentes tareas diarias o funciones que realizan los encargados promoción y brindar una herramienta o un medio que pueda mejorar los procesos.

En base a encuestas realizadas en el banco de sangre, respecto a la información que se maneja y se genera de los donantes, se han identificado los siguientes problemas.

- Dificultad para la toma de decisiones.

Ejemplos de decisiones:

- Ubicar a los donantes cuando se requiera su tipo de sangre
- Para la nueva donación de donantes habituales después de 3 meses de su última donación
- Deficiencia en el servicio de donaciones móviles
- Bajo conocimiento en el uso de las tics.
- No se cuenta con medios o mecanismos seguros y responsables que proporcionen información de donantes móviles, como consecuencias tenemos

Cuando quieren consultar los resultados de enfermedades, como están registrados en papel hay demora. (Ejemplo: en colectas.)

Realizar consultas o reportes a corto o a largo plazo con lleva tiempo.

Es necesaria la obtención de reportes a largo plazo, como estadísticas anuales para ver la proyección de avances de donaciones.

Existe Perdidas de registros de donaciones móviles; esto puede ocurrir en el transcurso de viajes para colectar sangre o bien surgen pérdidas de información cuando el donante se lleva los formularios del mismo.

- Registros de Información de los donantes desordenados, como las donaciones fuera del banco de sangre, no existe orden por tipo de sangre u otro aspecto, la atención esto es por llegada del donante.
- Demora en la Transcripción de resultados e información en los cuadernos. Como existen muchos registros de donaciones de diferentes instituciones en diferente tiempo se deben tabularse para su almacenamiento.
- Desperdicio de hojas por errores de transcripción de resultados de los análisis; un error de transcripción se debe rehacer el registro.

Es por este motivo, la institución requiere un medio eficiente y apropiado para satisfacer de manera más adecuada los requerimientos del personal de las áreas que están relacionadas con donaciones móviles.

Por lo general se debe contar con información segura y responsable que ayude a la toma de decisiones y de esta manera anular el gran problema que no solo acomete a esta institución sino a muchas que colapsan por la falta de este instrumento adecuado.

ÁRBOL DE PROBLEMAS

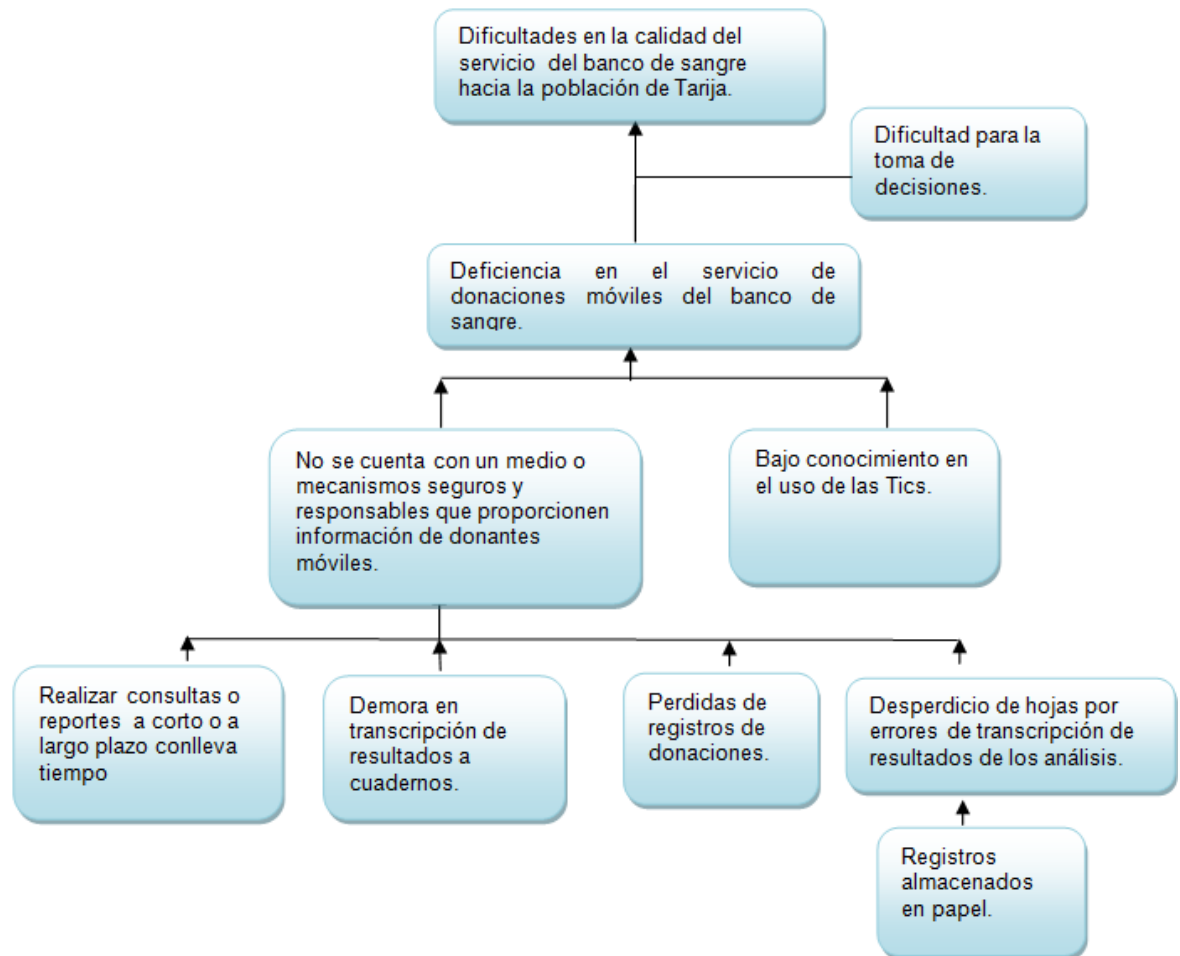


Fig. 1 Arbol de Problemas

ÁRBOL DE OBJETIVOS

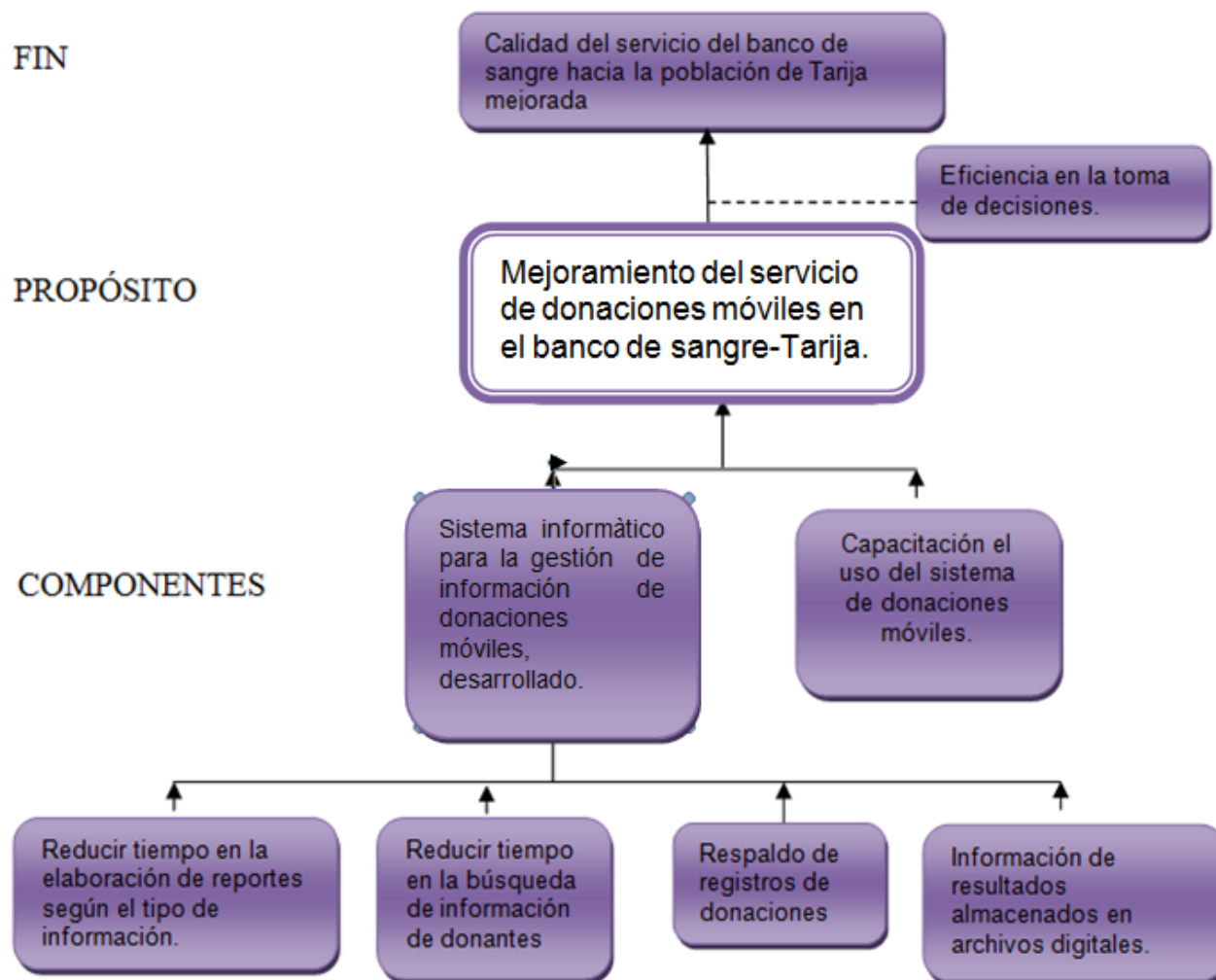


Fig. 2 Arbol de Objetivos

I.2.3. Objetivos

I.2.3.1. Objetivo General

Mejoramiento del Servicio de Donaciones Móviles en el banco de sangre - Tarija.

I.2.3.2. Objetivos Específicos

- Sistema informático para la gestión de información de donaciones móviles, desarrollado.
- Capacitación en el uso del sistema de donaciones móviles.

I.2.4. Metodología

Metodología para el componente1: sistema.

La metodología que se aplicará en el componente sistema será el siguiente:

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

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

El flujo de trabajo fundamental tiene los siguientes pasos:

- Requerimientos: necesidades del negocio trasladadas a un sistema automatizado.
- Análisis y Diseño: requerimientos dentro de la arquitectura de software.

- Programación e Implementación: software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Requerimientos: en base a las entrevistas se obtendrá la información que refleje las necesidades de los involucrados para la determinación de requerimientos.

Análisis y diseño: Se estructurará las diferentes vistas (Diagramas, Base de Datos, Pantallas) de la aplicación.

Programación e Implementación: la programación será modular y orientada a objetos, se utilizarán tecnologías de punta, creando la aplicación informática.

Pruebas y Validación: antes de desarrollar las pruebas se procederá a la introducción de datos. La fase de pruebas de desarrollo que serán mediante casos de prueba tomados de cada módulo y se realizarán los ajustes necesarios para una correcta validación.

Metodología para el componente2: capacitación.

.....

.....

hacere

I.2.5. Bibliografía Consultada

- PRESSMAN Ingeniería de software.
- www.desarrolloweb.com Sitio de análisis y diseño de sistemas.

I.2.6. Resultados Esperados

Sistema automatizado denominado: “Sistema de donaciones móviles” desarrollado, probado y validado, que integre de manera eficaz la lógica de procesamiento de información de acorde a la sección involucrada.

Un sistema que ayude en la gestión de información de donantes externos y contribuir a la mejora de la calidad del servicio de donaciones móviles de la institución “banco de sangre de referencia departamental Tarija”.

Reducción del trabajo manual del personal, evitando pérdidas de información u otros inconvenientes.

Se espera que el sistema sea desarrollado de acuerdo a la especificación de requerimientos y sea aprobada por los docentes de taller III según la norma IEEE830.

Capacitación del personal de un 80% (doctores, enfermeras y bioquímicos) pertenecientes al departamento de promoción y extensión social, del banco de sangre, en el uso del sistema y sobre recursos TIC.

I.2.7. Transferencia de resultados

a) Medios y estrategias para la transferencia de resultados.

Presentación final del sistema informático a la división de calidad y a la dirección ejecutiva del banco de sangre de referencia departamental Tarija.

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

b) Grupo de beneficiarios de los resultados

El jefe de la división de calidad.

El jefe y funcionarios del departamento de promoción y extensión social del banco de sangre.

I.2.8. Cronograma de Actividades

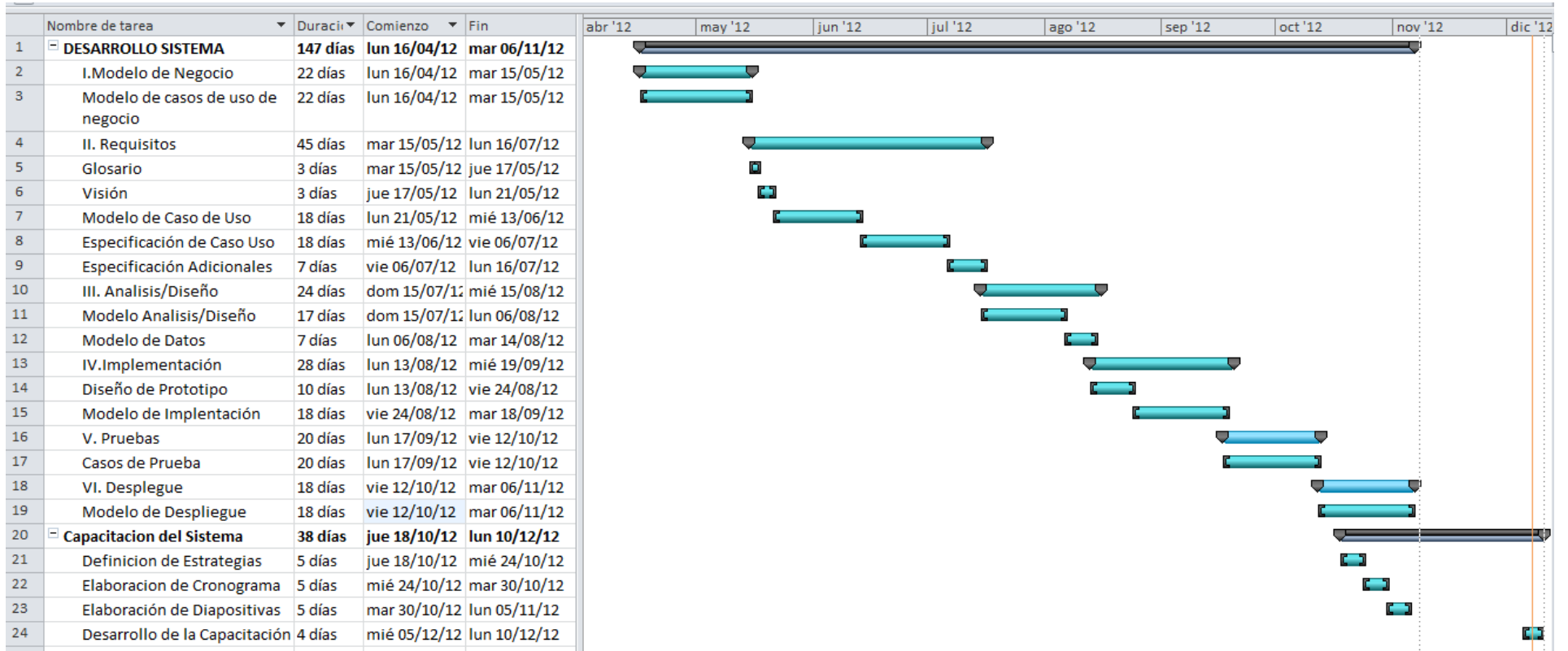


Fig. 3 Cronograma de Actividades

I.2.9. Marco Lógico del Proyecto

Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
<p>Fin</p> <p>Calidad del servicio del banco de sangre hacia la población de Tarija mejorada.</p>	<p>A partir del segundo año de ejecución del proyecto el 30% de los beneficiarios expresan un grado alto de satisfacción con relación al año base de ejecución del proyecto 2012.</p>	<p>Resultados de las encuestas realizadas a la población de beneficiarios del departamento de promoción y extensión social, ejecutadas por la dirección del Banco de Sangre.</p>	<p>El funcionamiento del Banco de Sangre sea normal.</p> <p>Infraestructura adecuada para el desarrollo de sus actividades del proyecto.</p> <p>El banco de sangre incrementa el presupuesto para apoyar estrategias de mejora en el servicio a la población.</p>
<p>Objetivo General (Propósito)</p> <p>Mejoramiento del Servicio de Donaciones Móviles en el banco de sangre - Tarija</p>	<p>A finales del 2012 se ha agilizado la obtención de información de donaciones móviles del banco de sangre con un porcentaje del 60% respecto al año base 2011.</p>	<p>Informe del “jefe de calidad” o jefe de promoción avalando la veracidad de la información de donaciones móviles.</p>	<p>Personal del departamento de promoción y extencion socia se muestran interesados y apoyan el proyecto.</p> <p>La información acerca de las donaciones móviles se encuentra disponible.</p>

<p>Objetivos Específicos</p> <p>(Componentes)</p> <p>1. Sistema informatico para la gestión de registros de donaciones móviles, desarrollado.</p>	<p>El sistema será desarrollado de acuerdo a la especificación de requerimientos aprobada por los docentes de taller III según la norma IEEE830.</p> <p>El sistema informatico estará terminado el 6 de noviembre del 2012.</p>	<p>Sistema automatizado documentada.</p> <p>Informe de jefe de promoción y dirección respecto a la conformidad del sistema.</p> <p>Sistema aprobado por parte de los docentes de la materia de Taller III.</p>	<p>Información de requerimientos recopilada “oportunamente” y confiable.</p>
<p>2. Capacitación en el uso del sistema de donaciones móviles.</p>	<p>Al finalizar el proyecto en diciembre de 2012, se capacito al 65% del personal (doctores, enfermeras y bioquímicos) involucrados en el departamento de promoción y extensión social del banco de sangre, en el uso del sistema y sobre recursos TIC.</p>	<p>Lista de firmas del personal asistido a las capacitaciones.</p> <p>Informe obtenida del jefe de promoción del banco de Sangre con respecto a la capacitación del uso del sistema.</p> <p>Fotografías tomadas durante la realización de la capacitación.</p>	<p>Interés y disponibilidad de tiempo del personal del BSRDT, para la realización de la capacitación del sistema.</p> <p>Las personas involucradas directamente con el sistema cumplen con la citación de la capacitación.</p>

<p>Actividades 1. componente I</p> <p>I. Modelo del Negocio I.1 Modelo de casos de uso de negocio y modelo de objetos del negocio</p> <p>II.Requisitos II.1 Glosario II.2 Visión II.3 Modelo de caso de uso II.4 Especificación de caso de uso. II.5 Especificación adicionales</p> <p>III.Analisis/diseño III.1 Modelo Análisis/diseño III.2 Modelo de datos.</p> <p>IV.Implementacion IV.1 Prototipos de interfaces de usuario. IV.2 Modelo de Implementacion.</p> <p>V.Pruebas V.1 Casos de Prueba.</p> <p>VI.Despliegue V.1 Modelo de Despliegue.</p>	<p>250\$</p> <p>400\$</p> <p>250\$</p> <p>600\$</p> <p>200\$</p> <p>200\$</p> <p>Total 1900\$</p>	<p>Documentación e información recogida en las entrevistas hechas al personal (fotocopias de registros de donantes de puesto móvil).</p> <p>Documentación obtenida por el análisis del sistema y prototipos del mismo.</p> <p>Documento de los casos de pruebas del sistema.</p>	<p>Disponibilidad de brindar información oportuna por parte del departamento de promociones y extensión social. Coordinación con los cooperantes del trabajo del proyecto.</p>
--	--	--	--

2. Componente II			
2.1. Definición de estrategias para la capacitación.	50\$		Asistencia del personal a la convocatoria de capacitación.
2.2. Elaboración del cronograma de actividades	50\$	Documento del plan de capacitación.	
2.3. Elaboración de diapositivas	50\$	Carta de conformidad con el desarrollo de la capacitación	
2.4. Desarrollo de la capacitación.	100\$	de parte de la directora del BANCO DE SANGRE.	
Total	250\$	Lista de capacitados, mas sus firmas.	

Tabla 2 Marco Lógico del Proyecto

I.2.10. Presupuesto /Justificación

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
10000	SERVICIOS PERSONALES			
	12000 Empleados no Permanentes			10020
	Sub total rubro			10020
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos			1720
	22000. Servicios de transporte			325
	23000. Alquileres			
	24000. Mantenimiento y reparación			150
	25000. Servicios Profesionales y Comerciales			
	Sub total rubro			3005
30000	MATERIALES Y SUMINISTROS			
	32000. Productos de Papel, Cartón e Impresos			1400
	Sub total rubro			1400
40000	ACTIVOS REALES			
	43000. Maquinaria y Equipo.			800
	46000. Descripción de estudios y proyectos para inversión			
	Sub total rubro			800
	TOTAL			15225
	TOTAL + 40% Incentivo			21315

Tabla 3 Presupuesto /Justificación

1) GRUPO 10000. SERVICIOS PERSONALES

a) SUB GRUPO 12000. Empleados no Permanentes

Partida	Personal	Remuneración	Tiempo/meses	Total
12100	Personal Eventual		8	10020
Total				10020

Tabla 4 SUB GRUPO 12000. 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- internet	95	8	760
21200	Energía Eléctrica	50	8	400
21300	Agua	20	8	160
21400	Servicios Telefónicos	50	8	400
Total				1720

Tabla 5 SUB GRUPO 21000. Descripción de los gastos de servicios básicos

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

Partida	Personal	Lugar	N° de viajes	Costo unitario*	Costo total
22100	Pasajes	Ciudad de Tarija (cercado).	50	3.5	175
			100	1.5	150
Total					325

Tabla 6 SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

Partida	Personal	Lugar	Duración (días)	Costo unitario*	Costo total
22200	Viáticos				0
22300	Fletes y Almacенamientos				0
Total					0
Total sub grupo 22000					325

e) SUB GRUPO 24000. Descripción mantenimiento y reparación

Partida	Mantenimiento y reparación de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
24300	Otros Gastos por Mantenimiento y Reparación	150	1	150
Total				150

Tabla 7 SUB GRUPO 24000. Descripción 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		200	2	400
25500	Publicidad			0	0
25600	Imprenta		200	1	200
25700	Capacitación de Personal		50	1	50
25800	Estudios e Investigaciones Para Proyectos de Inversión		160	1	160
Total					810

Tabla 8 SUB GRUPO 25000. 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			
Total				0

Tabla 9 SUB GRUPO 31000. 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	1	200	200
32200	Productos de Artes Gráficas, Papel y Cartón	6	200	1200
Total				1400

Tabla 10 SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

II.1.1. Componente 1.

II.1.1.1. Introducción

Los sistemas de gestión de información se refieren a un conjunto de todos los componentes necesarios que se interrelacionan, con el objetivo de gestionar y facilitar la información sobre el tema de interés.

II.1.1.2. Antecedentes

Banco de sangre: Es servicio especializado con registro y licencia de funcionamiento de la Secretaría Nacional de Salud, encargado de la recolección, extracción, procesamiento, almacenamiento, conservación, fraccionamiento, control de calidad y distribución de Sangre Humana destinada a transfusiones o investigaciones en forma total, o de sus componentes separados, sin fines de lucro, a centros de transfusión o investigación públicos o privados.

- El Banco de Sangre Inicia actividades el 15 de julio de 2004 con un Plan Piloto
- Tiene Autorización de funcionamiento SEDES-R.A. 04/09 del 7 de enero de 2009
- El B.S.R.D.T. es un centro Público desconcentrado del Servicio Departamental de Salud con patrimonio propio, autonomía administrativa, financiera, técnica y con competencia de ámbito departamental.

La Legislación aplicable al banco de Sangre:

1. Constitución Política del Estado
2. Decreto Ley N° 12760 (Código Civil)
3. Ley de la Medicina Transfusional y Bancos de Sangre

Objetivos Institucionales

- Planifica y promueve el incremento de la donación voluntaria de sangre.

- Planifica la cobertura del requerimiento de sangre de los servicios de transfusión, seguros sociales y pacientes privados.
- Planifica y revisa continuamente los resultados de sus procesos.
- Realiza el control de calidad establecido para los hemocomponentes.
- Responde a los requisitos y sugerencias de los donantes y servicios de transfusión.
- Capacita y actualiza a sus recursos humanos.
- Implementa, mejora y actualiza nuevas técnicas en inmunohematología

II.1.1.3. Organigrama del Banco de Sangre de Referencia Departamental Tarija

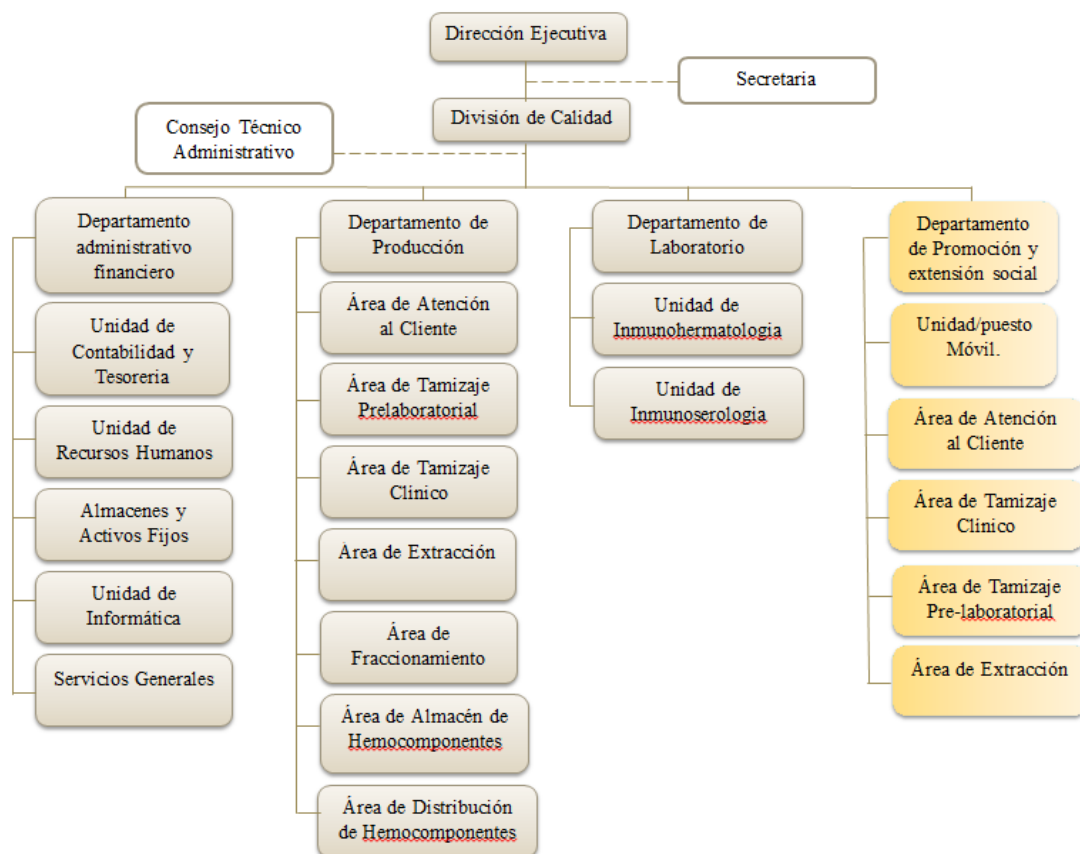


Fig. 4 Organigrama del Banco de Sangre

II.1.1.4. Metodologías de Desarrollo

II.1.1.4.1. Metodología RUP

RUP: es un proceso de desarrollo de software. Junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

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

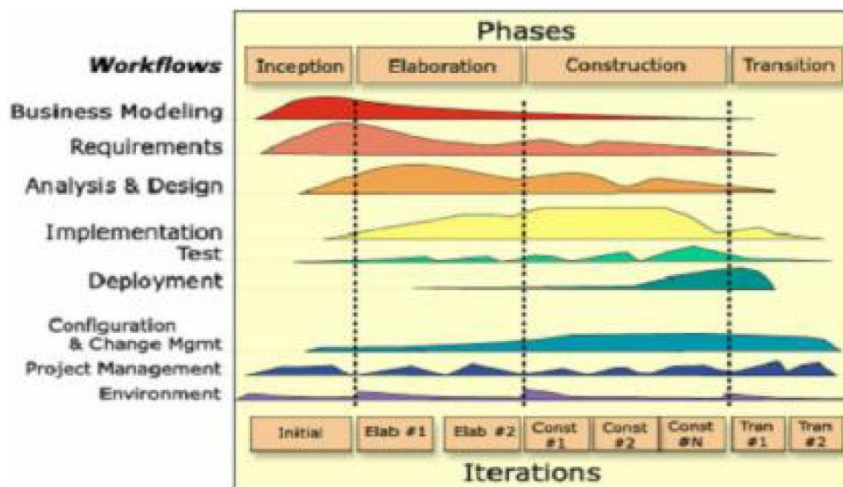
Nos permite realizar un levantamiento exhaustivo de requerimientos.

Principales características

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

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Fases en el Ciclo de Desarrollo.- Existen cuatro fases, la figura muestra las fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica, donde se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.



· Fase 1: Preparación Inicial (“Incepción”)

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

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

· Fase 2: Preparación Detallada (“Elaboración”)

En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio. El resultado de la fase de elaboración es: Un modelo de Caso de Uso (por lo menos 80% completo) - todos los Casos de Uso y actores deben haber sido identificados-, y se han desarrollado la mayoría de las descripciones de Casos de Uso. Requerimientos suplementarios que capturan los requerimientos no funcionales.

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

El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.

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

El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

II.1.1.4.2. UML (Lenguaje Unificado de Modelado)

UML: Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

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

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

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

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

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

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos).

d) UML ofrece notación y semántica estándar:

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

II.1.1.4.2.1. Tipos de Diagramas Utilizados**Diagramas de Clases**

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

Diagramas de Casos de Uso

Que captura las interacciones de los Casos de Uso y los actores. Describe los requisitos funcionales del sistema, la forma en la que las cosas externas (actores) interactúan a través del límite del sistema y la respuesta del sistema.

Diagrama de Actividades

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

Diagrama de Secuencias

Es una representación estructurada del comportamiento como una serie de pasos secuenciales a lo largo del tiempo. Se usa para representar el flujo de trabajo, el paso de mensajes y cómo los elementos en general cooperan a lo largo del tiempo para lograr un resultado.

II.1.1.5. Herramienta de Construcción de Software

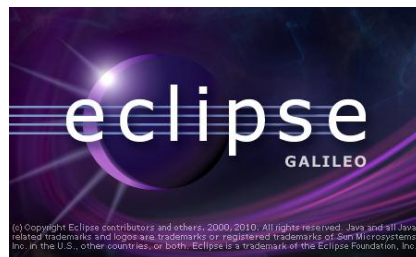
II.1.1.5.1. Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

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

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad **de código abierto** y un conjunto de productos complementarios, capacidades y servicios.



II.1.1.5.2. Visual Paradigm for UML 8.2 Enterprise Edition

Visual Paradigm for UML (VP-UML) es un UML CASE UML herramienta de apoyo 2, SysML y Business Process Modeling Notation (BPMN) del Object Management Group (OMG) . Además del apoyo de modelado, que proporciona capacidades de generación de informes y códigos de ingeniería incluyendo la generación de código . Se puede invertir ingeniero diagramas de código y proporcionar ida y vuelta ingeniería para diversos lenguajes de programación .



II.1.1.5.3. Tomcat

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.³

Podemos dividir los contenedores de Servlets en:

Contenedores de Servlets Stand-alone (Independientes)

Estos son una parte integral del servidor web. Este es el caso cuando usando un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer.

Contenedores de Servales dentro-de-Proceso

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

Contenedores de Servlets fuera-de-proceso

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



II.1.1.5.4. PgAdmin III PosgreSQL Tools

PgAdmin es un proyecto de software libre publicado bajo la licencia de PostgreSQL. El software está disponible en fuente y el formato binario de la red de servidores espejos de PostgreSQL. PgAdmin 3 es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados (EnterpriseDB Postgres Plus Advanced Server y Greenplum Database).

Incluye:

- Interfaz administrativa gráfica
- Herramienta de consulta SQL (con un EXPLAIN gráfico)
- Editor de código procedural
- Agente de planificación SQL/shell/batch
- Administración de Slony-I

pgAdmin se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.



II.1.1.6. Técnica.

II.1.1.6.1. Html5

HTML5 (*HyperText Markup Language*, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como *HTML5* y una variante XHTML conocida como sintaxis *XHTML5* que deberá ser servida como XML (XHTML) (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, lo cual indica la misma W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

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



II.1.1.6.2. CSS

Una hoja de estilo es un conjunto de instrucciones que definen la apariencia de diversos elementos de un documento HTML. En otras palabras una hoja de estilo nos permite indicar por ejemplo el tamaño de la fuente, color y estilo de cierto párrafo que nosotros indiquemos, mediante la definición de estilos y aplicación de los mismos.

Las hojas de estilo se usan porque tienen muchas ventajas sobre los tags tradicionales, ya que por ejemplo es posible crear una sola hoja de estilo que compartan muchos documentos, y al hacer un cambio en la hoja de estilo todos los documentos que la usan tendrán la apariencia deseada.

También se puede tener control sobre ciertos aspectos que antes no se tenía, por ejemplo se pueden definir los márgenes de un documento o párrafo, o definir el espacio entre caracteres.

Las hojas de estilo son la innovación más importante al HTML (también se usan en otros lenguajes como XML y SGML), ya que le dan capacidades que nunca tuvo.

El uso de diversas unidades de medición pixeles, puntos, picas, mm, en los principales elementos del HTML, como son tablas, fluentes, bordes y en general los elementos que tenían atributos como “size “height” width”.

El poder cambiar las características de una hoja de estilo mediante el uso de lenguajes de programación Web como “Javascript” o “VB script”.

II.1.1.6.3. JQuery

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

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos.¹ jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

II.1.1.6.4. JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

II.1.1.6.5. Java.

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Java script. Si bien su uso se

destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet). Java es un lenguaje de objetos, independiente de la plataforma.

Java posee ciertas características que hoy día se consideran estándares en los lenguajes OO:

- Objetos.
- Clases.
- Métodos.
- Subclases.
- Herencia simple.
- Enlace dinámico.
- Encapsulamiento.

Java es un lenguaje que ha sido diseñado para producir software:

- Confiable: minimiza los errores que se escapan a la fase de prueba.
- Multiplataforma: los mismos binarios funcionan correctamente en Windows/95/NT/XP/VISTA, Linux, Unix/Motif y Power/Mac.
- Seguro: Applets recuperados por medio de la red no pueden causar daño a los usuarios.
- Orientado a objetos: beneficioso tanto para el proveedor de bibliotecas de clases como para el programador de aplicaciones.
- Robusto: los errores se detectan en el momento de producirse, lo que facilita la depuración.

II.1.1.6. 6.Arquitectura y diseño: Modelo Vista Controlador

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

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

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

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

II.1.1.6.6.1. Componentes de Patrón MVC

Patrón MVC: A alto nivel, el patrón MVC permite separar los distintos aspectos de una aplicación (Inputs Lógicos, Lógica de Negocio, Lógica de UI), otorgando un bajo acoplamiento entre estos elementos. Esto permite que al momento de construir una aplicación, nos focalicemos sólo en un aspecto, como por ejemplo en una vista, sin tener dependencia de la lógica del negocio.

Los componentes son:

Modelo: corresponde a la parte de la aplicación que implementa la lógica del Dominio de

Datos de la aplicación, además de incorporar la persistencia de datos. Frecuentemente, los objetos del modelo recuperan y almacenan los estados del

modelo en la base de datos. Aquí se hace el levantamiento de los objetos que el sistema deba utilizar y es el proveedor de los recursos al Controlador.

Vista: es el componente que despliega la interfaz de usuario de aplicación. Normalmente, esta UI es creada a partir de los datos del modelo (Un ejemplo puede ser una vista de edición de una tabla de Productos que muestra cuadros de texto, listas desplegables y casillas de verificación basado en el estado actual de los objetos Producto). Esta puede ser una web HTML, un XML, un archivo binario, etc.

Controlador: el controlador maneja la interacción del usuario, trabaja con el modelo, y por último selecciona la vista a renderizar en la interfaz de usuario. En una aplicación MVC, la vista sólo despliega información; el controlador maneja y responde los inputs e interacción del usuario. Éste escucha los cambios a la vista y se los envía al modelo

II.1.1.6.7. Servlet

Los servlets son objetos que corren dentro del contexto de un contenedor de servlets (ej.: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones (ej.: OC4J Oracle), que, además de contenedor para servlet, tendrá contenedor para objetos más avanzados, como son los EJB (Tomcat sólo es un contenedor de servlets).

La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor.

El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

Un servlet implementa la interfaz `javax.servlet.Servlet` o hereda alguna de las clases más convenientes para un protocolo específico (ej: `javax.servlet.HttpServlet`). Al implementar esta interfaz el servlet es capaz de interpretar los objetos de tipo `HttpServletRequest` y `HttpServletResponse` quienes contienen la información de la página que invocó al servlet.

II.1.1.6.8. Spring

Spring es un framework de aplicaciones Java/J2EE desarrollado usando licencia de OpenSource. Es un conjunto de módulos, de los cuales podemos utilizar los que queramos.

En el proyecto utilizaremos el módulo Spring Web MVC para el desarrollo de aplicaciones Java basadas en Web construido sobre el núcleo de Spring. Dos de los objetivos más importantes de Spring MVC es permitir que el desarrollo se concentre en la lógica del negocio y que se haga empleando buenos principios de diseño orientado a objetos.

Para lograrlo se utiliza un concepto muy interesante llamado Inversión del Control (IoC), también conocido como el principio Hollywood. Esto permite que el código escrito por los desarrolladores para la lógica principal del sistema no tenga dependencias sobre las clases del *framework*; lo cual redundo en un código mucho más limpio y con la posibilidad de utilizar todas las ventajas de la programación orientada a objetos (específicamente la herencia).

II.1.1.7. Base de Datos

II.1.1.7.1. Definición de Base de Datos

Una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Entre las principales características de los sistemas de Base de Datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Los Sistemas de Gestión de Base de Datos (en inglés DataBase Management System) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

II.1.1.7.2. Herramienta PostgreSQL como Base de Datos

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley.

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

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

- *Alta concurrencia.* Mediante un sistema denominado MVCC, PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- *Soporte para una amplia variedad de tipos nativos.* PostgreSQL provee nativamente soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas, direcciones IP (IPv4 e IPv6), bloques de direcciones estilo CIDR, direcciones MAC, Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Consta con un lenguaje propio llamado [PL/PgSQL (similar al PL/SQL de oracle), pero también soporta los lenguajes C, C++, Java PL/Java web, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme.

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

Postgres cuenta con Herramientas de Administración como es PgAdmin3 y PgAccess: Entorno de escritorio visual, PhpPgAdmin y psql que es un Cliente de consola.

II.1.2. Plan de Desarrollo de Software

II.1.2.1. Introducción

El presente documento es un plan de desarrollo del software, será la base para el desarrollo de proyecto. Este plan esta basado en la metodología RUP, incluyendo sus fases de inicio y consecutivamente para dar una visión global del proceso.

Este documento a su vez es uno de los artefactos del RUP.

II.1.2.1.1. Propósito

El propósito es proporcionar información necesaria para controlar el proyecto y representar la visión de desarrollo del software.

II. 1.2.1.2. Alcance

El plan de desarrollo de software coadyuvará al cumplimiento de nuestros objetivos en el tiempo propuesto gracias al cronograma de actividades establecido.

II.1.2.1.3. Vista General del Proyecto

Con el proyecto se pretende contribuir a las donaciones móviles del banco de sangre de referencia departamental tarija.

El plan de desarrollo del proyecto de software describe el plan global usado para el desarrollo de “sistema de donaciones móviles”, un sistema informatico que esta comprendido por las siguientes áreas o módulos:

- Modulo Administrador
- Modulo donante
- Modulo Pruebas
- Modulo reportes

Llegando a la conclusión que la implementación del sistema automatizado dara beneficios claramente notorios, el sistema proyectara solución a mediano plazo que

beneficiara a dicha institución, pero lo mas importante es que el sistema actual mejorara.

II.1.2.1.3.1. Objetivos.

II.1.2.1.3.1.1. Objetivo General

Contribuir a la mejora del servicio de donaciones móviles del banco de sangre de referendia departamental Tarija.

II.1.2.1.3.1.2. Objetivos Específicos

- Llegar a brindar un sistema rápido y eficiente para la manipulación de registros
- Conseguir mayor organización en los registros.
- Obtener un sistema para que permita generar reportes de forma rápida y sencilla.
- Crear una base de datos de acuerdo a sus necesidades de la institución
- Diseñar interfaces de usuario de fácil manejo.

II.1.2.1.4. Suposiciones y Restricciones

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

II.1.2.1.4.1. Suposiciones

- Equipo de computación e impresora
- Presupuesto suficiente para la implementación del sistema
- Interés de la institución para utilizar el sistema
- La formación del personal encargado se lleve en la fecha determinada.

II.1.2.1.4.2. Restricciones

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

- El sistema será restringido, sólo usuarios privilegiados podrán acceder al sistema.
 - Para la manipulación de la base de datos, sólo podrá acceder el personal autorizado.
 - Deberá contar como mínimo con un sistema operativo Windows XP/Vista/Seven ·
- El administrador deberá contar con el conocimiento necesario para el manejo del sistema.
- . El sistema no abarca el control de la asistencia de los empleados de la institución.
 - No contar con un Servidor Web que presente las características necesarias para la ejecución del Sistema.

II.1.2.1.4.3 Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es necesario 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.

II.1.3. MODELO DE CASOS DE USO

II.1.3.1. Introducción

El Modelo de Casos de Uso forma parte de los diagramas requeridos en la metodología RUP la cual estamos implementando.

II.1.3.1.1. Propósito

- Comprender la estructura y la dinámica del sistema deseado para la organización.
- Identificar posibles mejoras.

II.1.3.1.2. Alcance

- Describe los procesos de sistema y los usuarios.
- Identificar y definir los *procesos del sistema* según los *objetivos* de la organización.
- Definir un *caso de uso* para cada proceso del sistema (el *diagrama de casos de uso* puede mostrar el contexto y los límites de la organización).

II.1.3.2. Diagrama de Casos de Uso

Casos de Uso General del Sistema

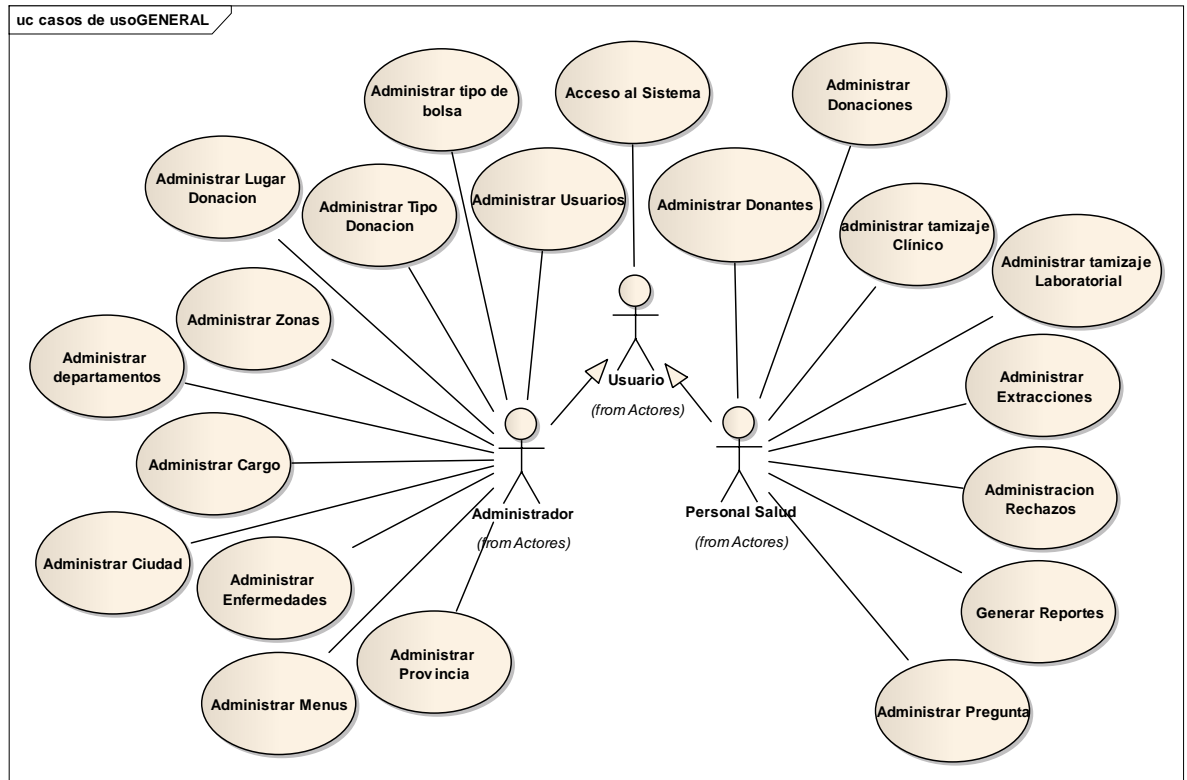


Fig. 5 Caso de Uso General del Sistema

Acceso al sistema.

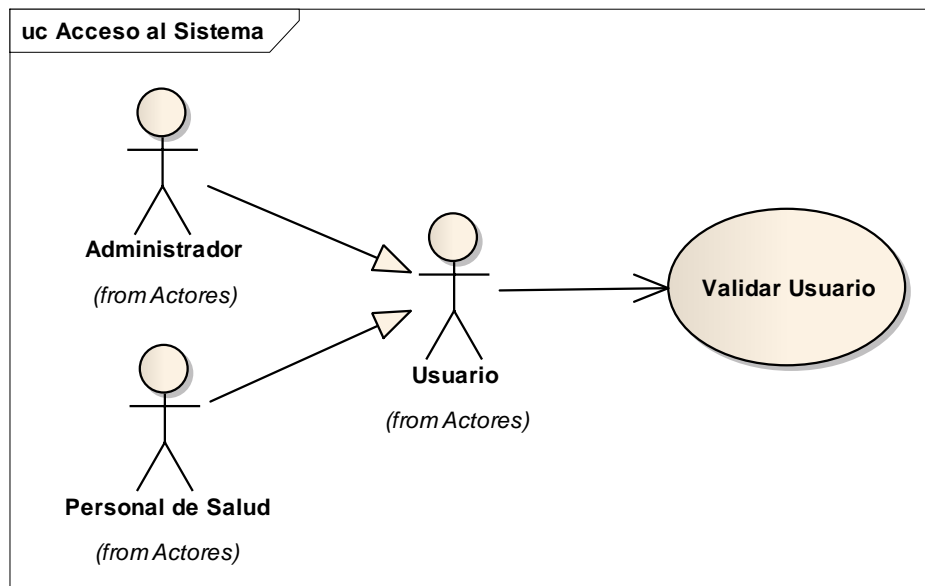


Fig. 6 Acceso al sistema.

Administrar Donantes

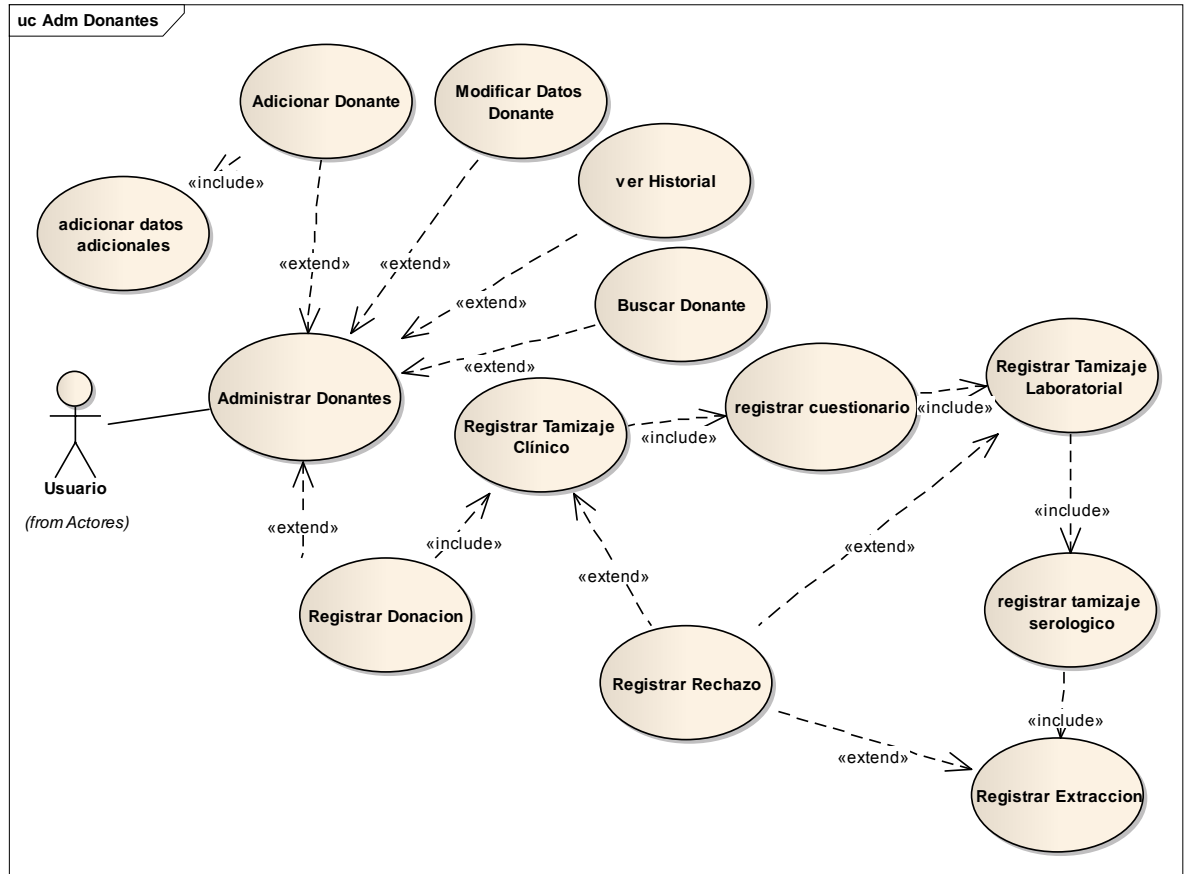


Fig. 7 Administrar Donantes

Administrar Tamizaje Clínico

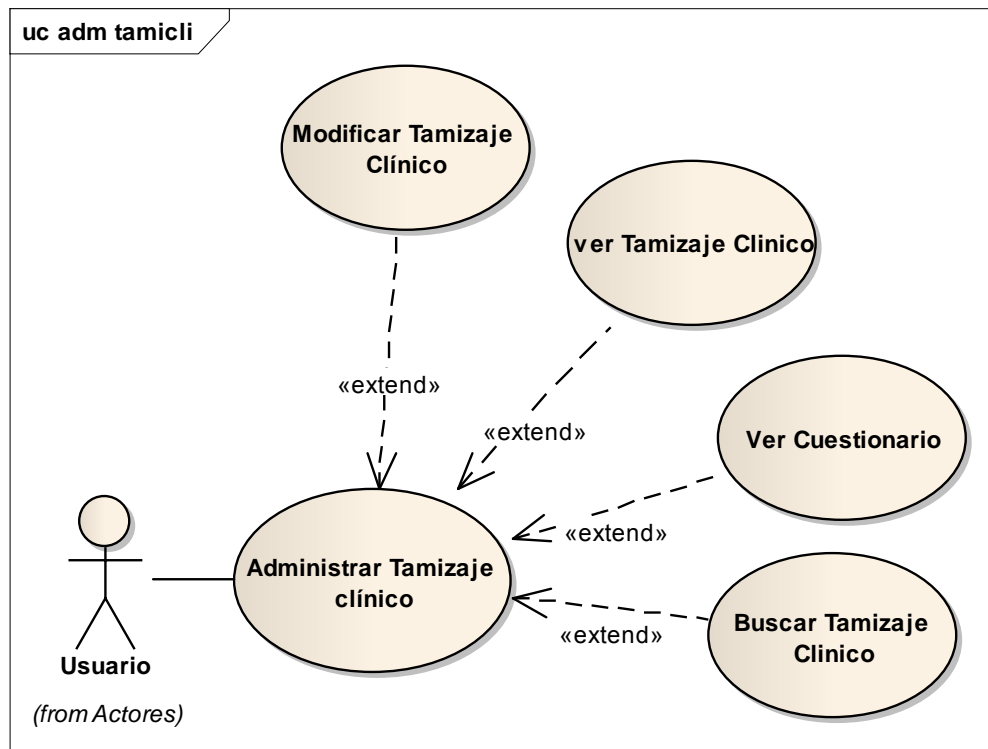


Fig. 8 Administrar Tamizaje Clínico

Administrar Tamizaje Laboratorial

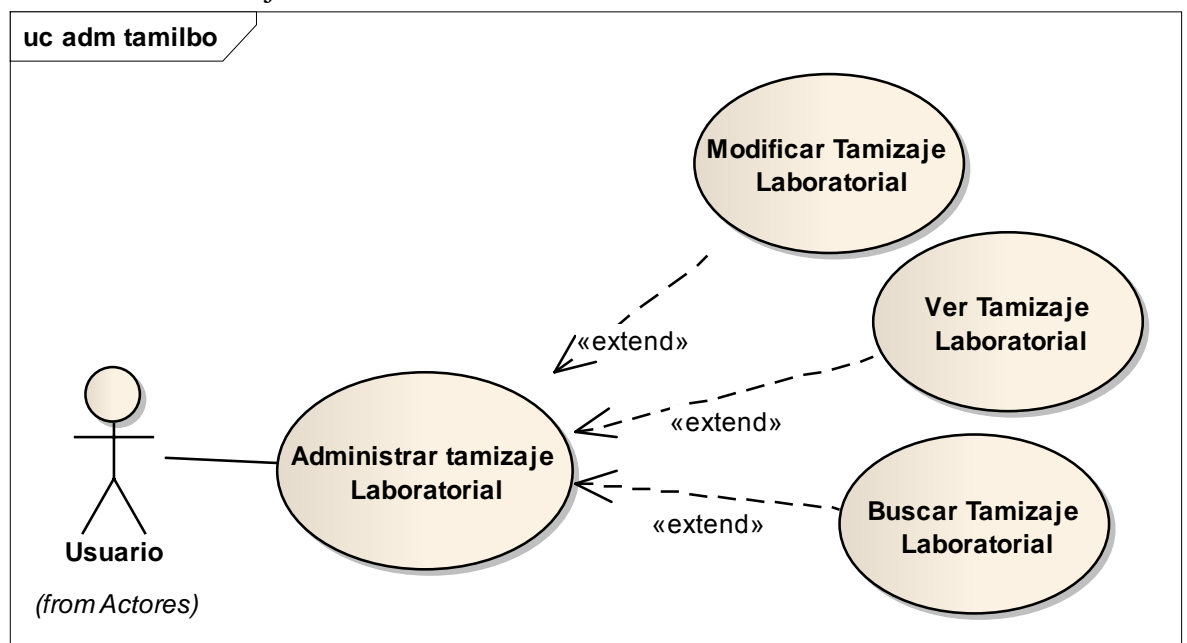


Fig. 9 Administrar Tamizaje Laboratorial

Administrar Extracción

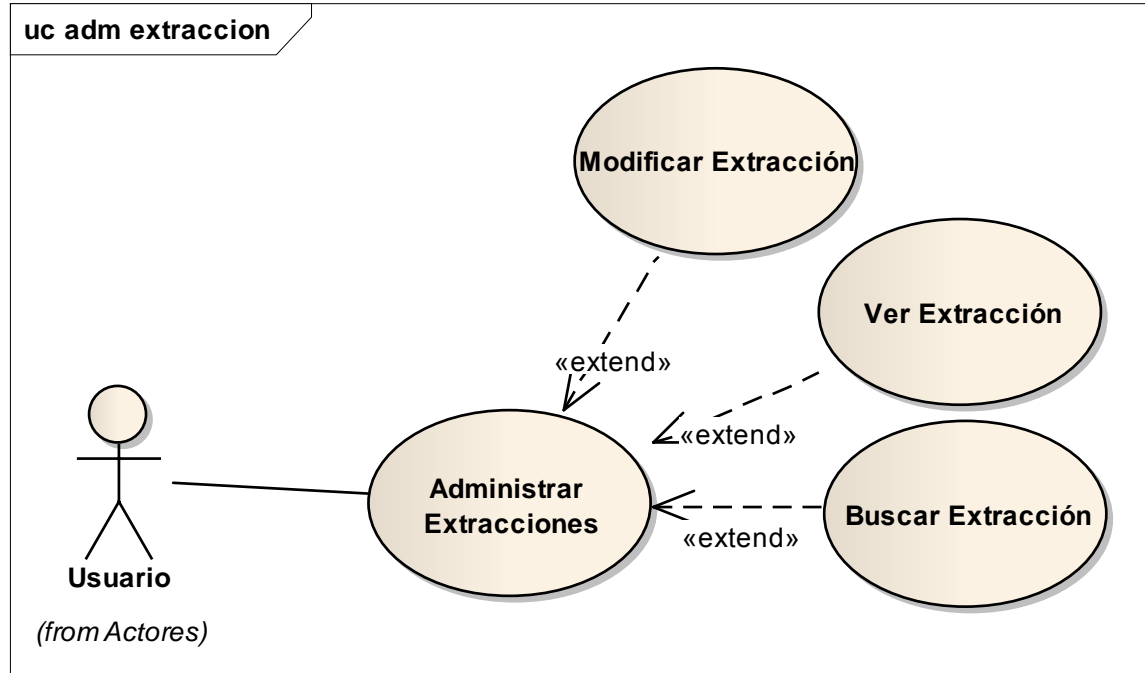


Fig. 10 Administrar Extracción

Administrar Donaciones

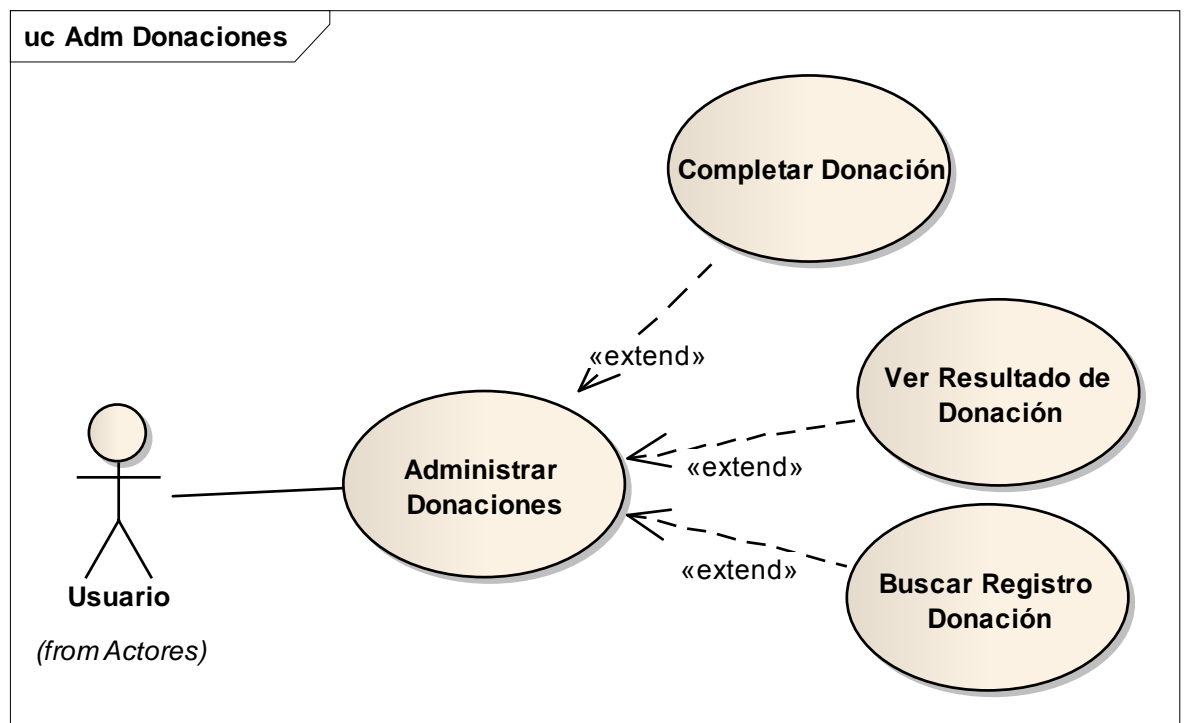


Fig. 11 Administrar Donaciones

Administrar Rechazo

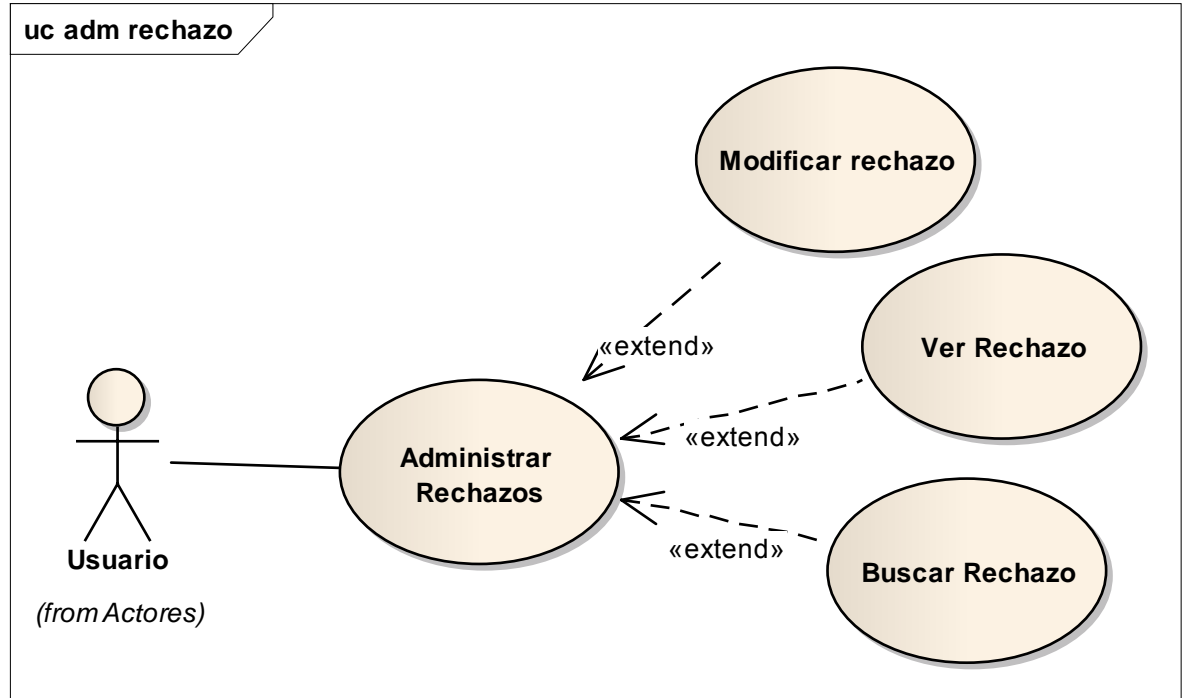


Fig. 12 Administrar Rechazo

Administrar Pregunta.

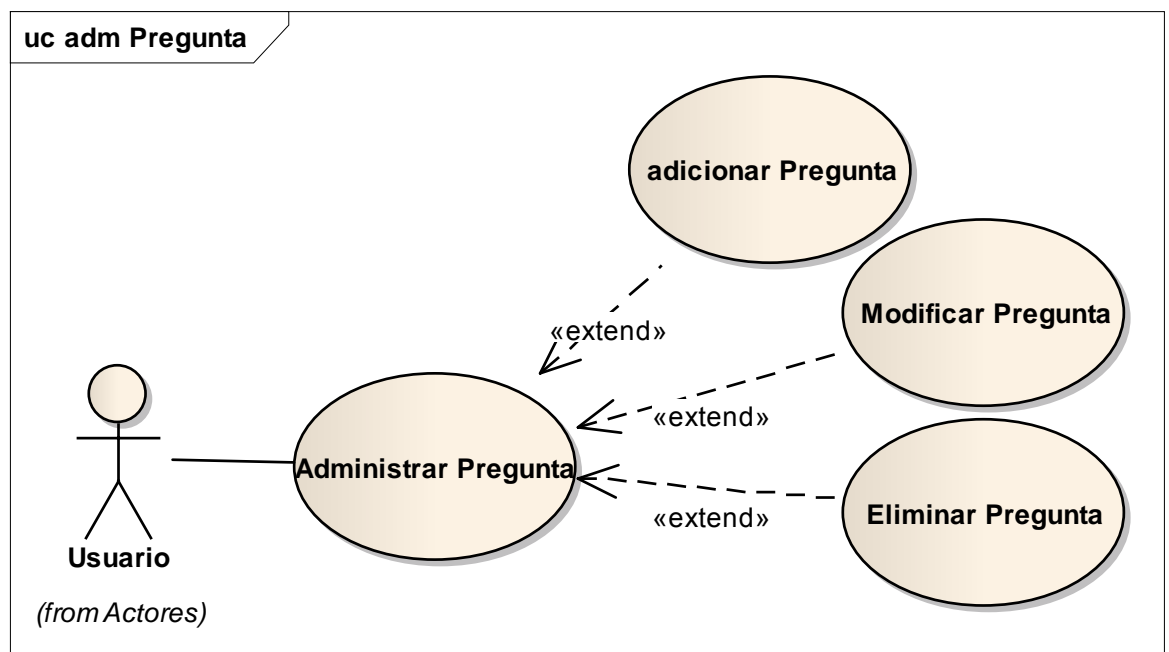


Fig. 13 Administrar Pregunta.

Administrar Tipo Bolsa

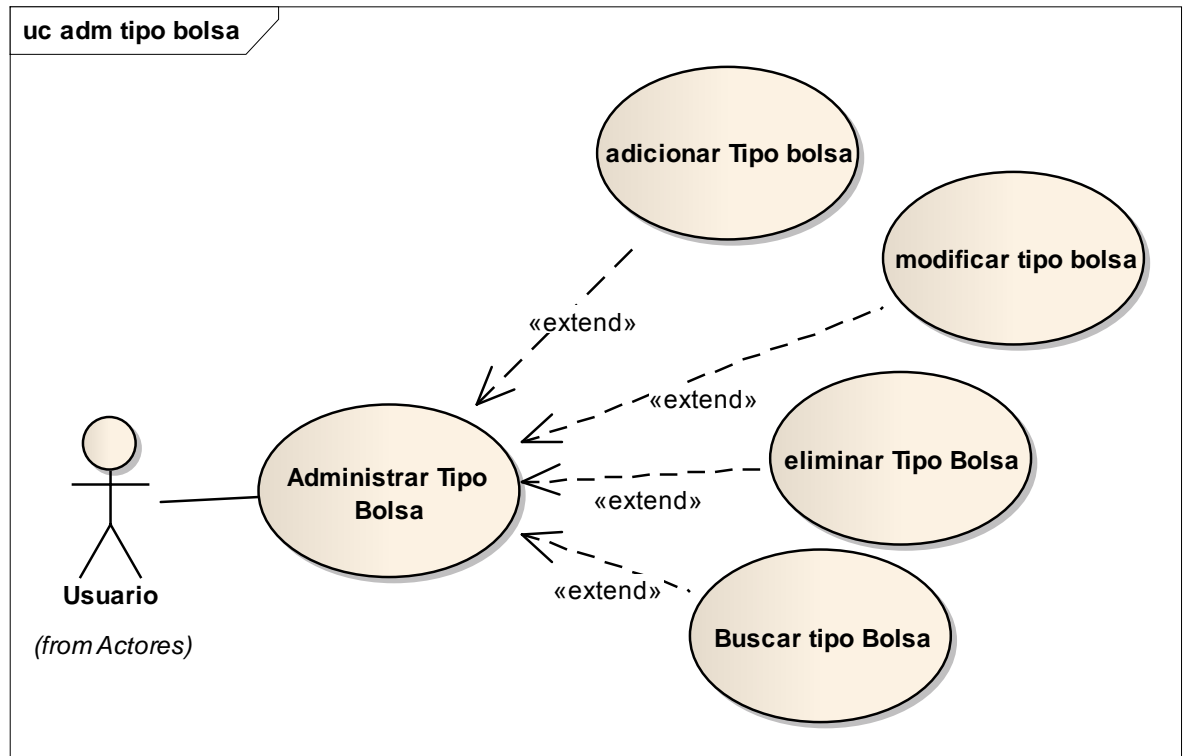


Fig. 14 Administrar Tipo Bolsa

Administrar Tipo Donación

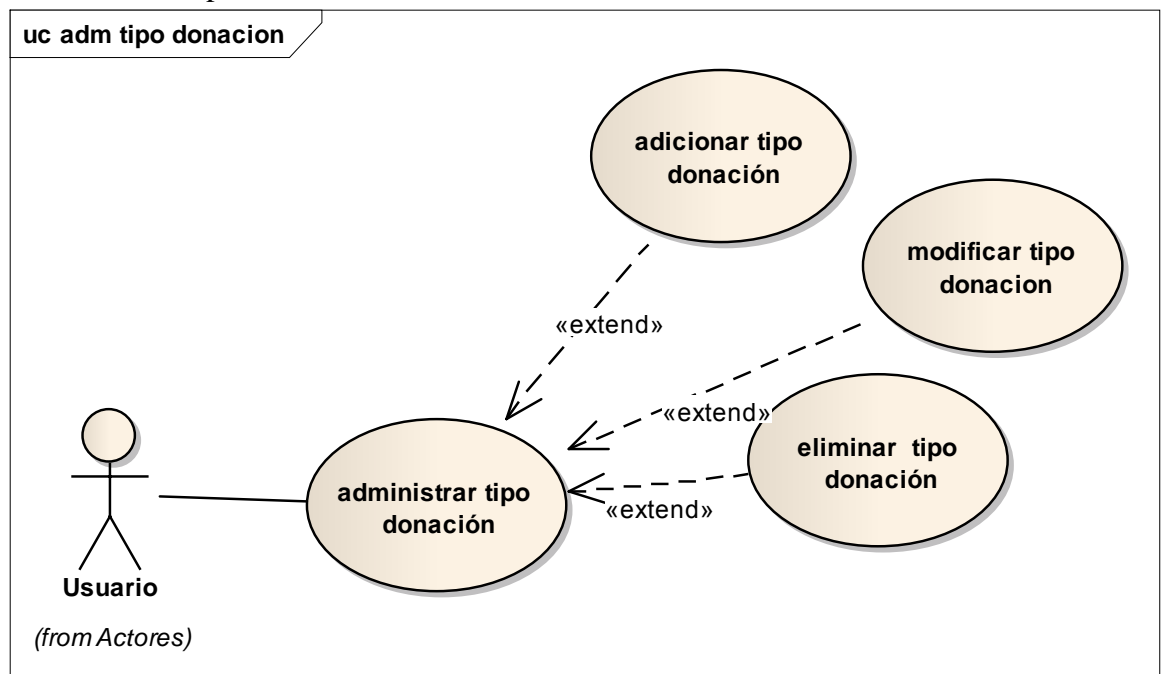


Fig. 15 Administrar Tipo Donación

Administrar Lugar de Donación

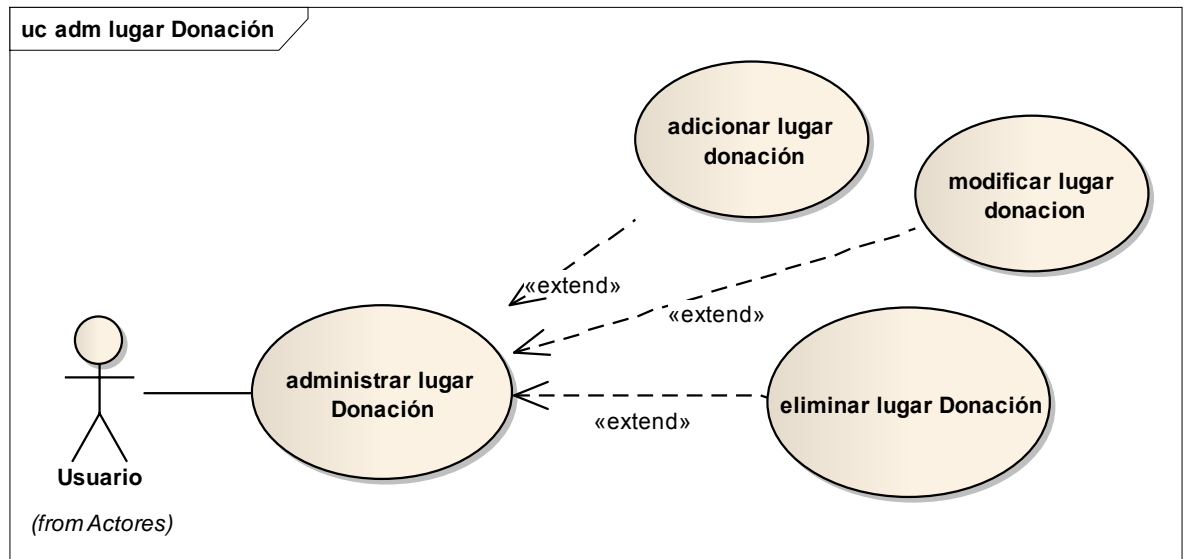


Fig. 16 Administrar Lugar de Donación

Administrar Departamentos

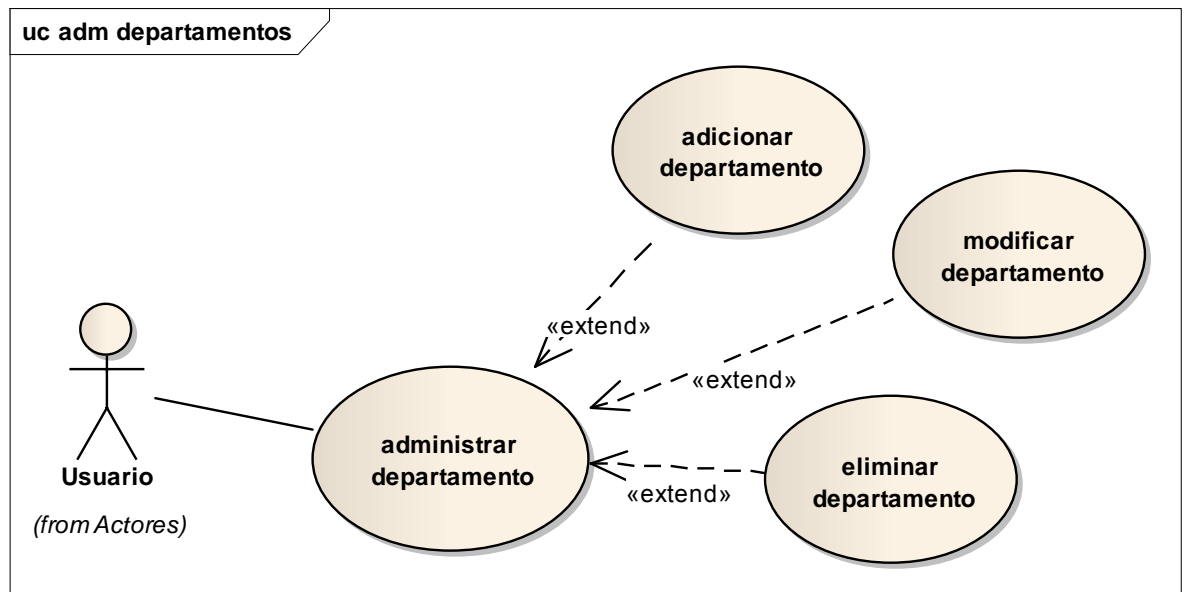


Fig. 17 Administrar Departamentos

Administrar Provincia

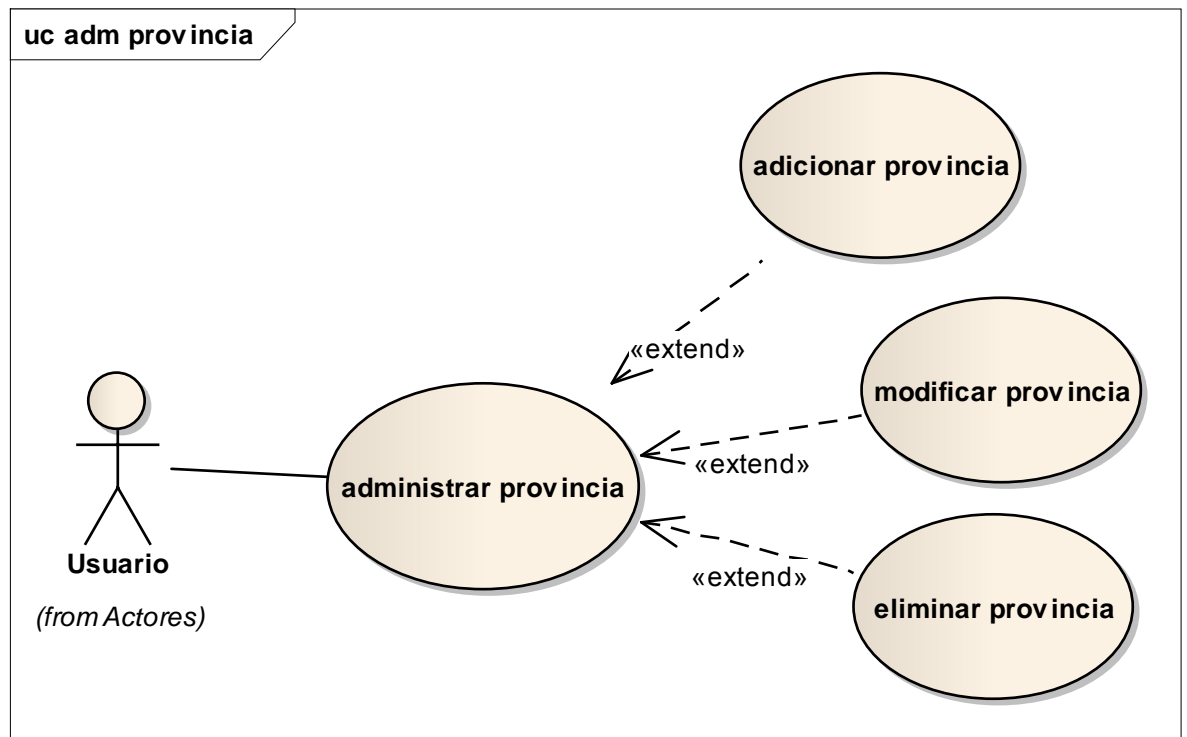


Fig. 18 Administrar Provincia

Administrar ciudad

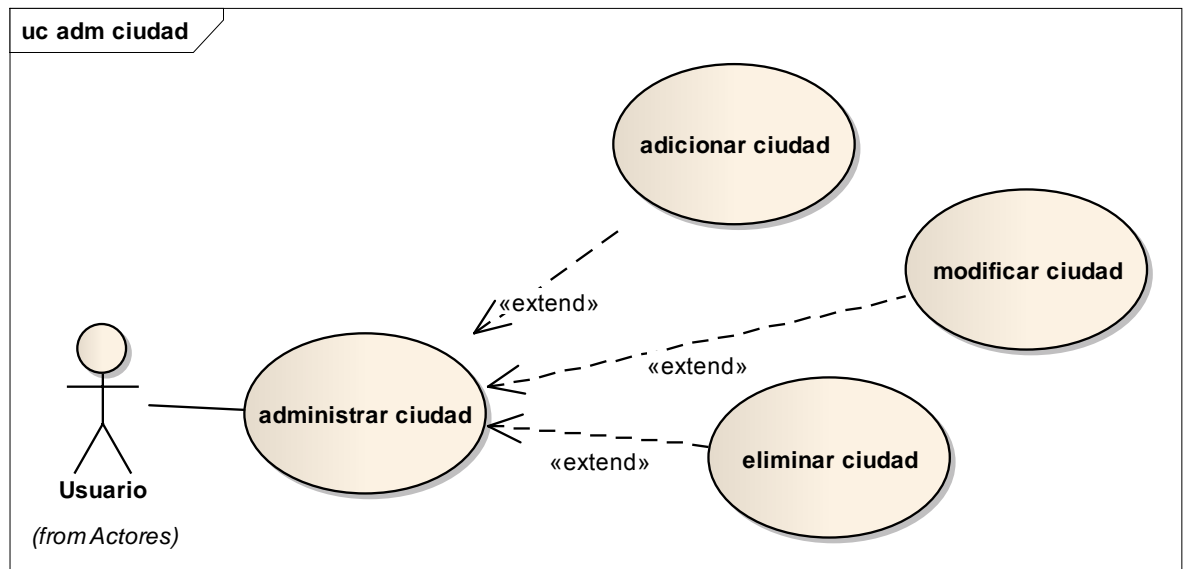


Fig. 19 Administrar ciudad

Administrar Zonas

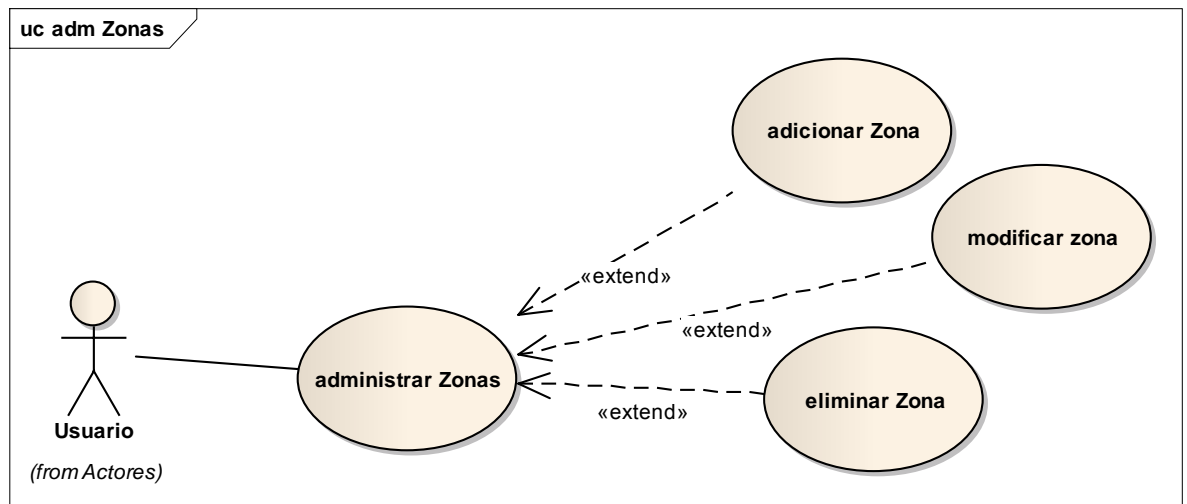


Fig. 20 Administrar Zonas

Administrar Enfermedades

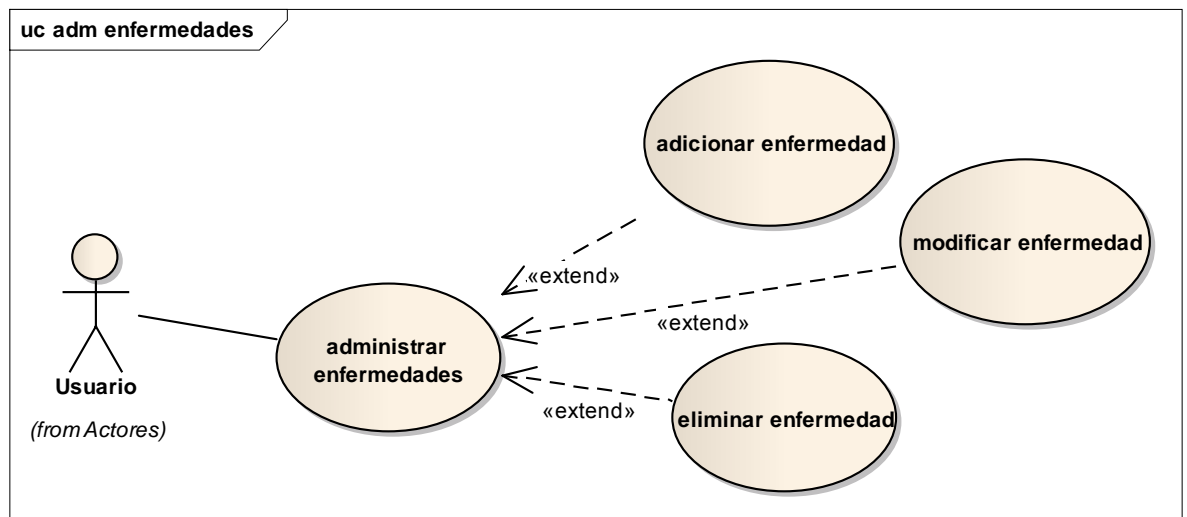


Fig. 21 Administrar Enfermedades

Administrar Usuarios

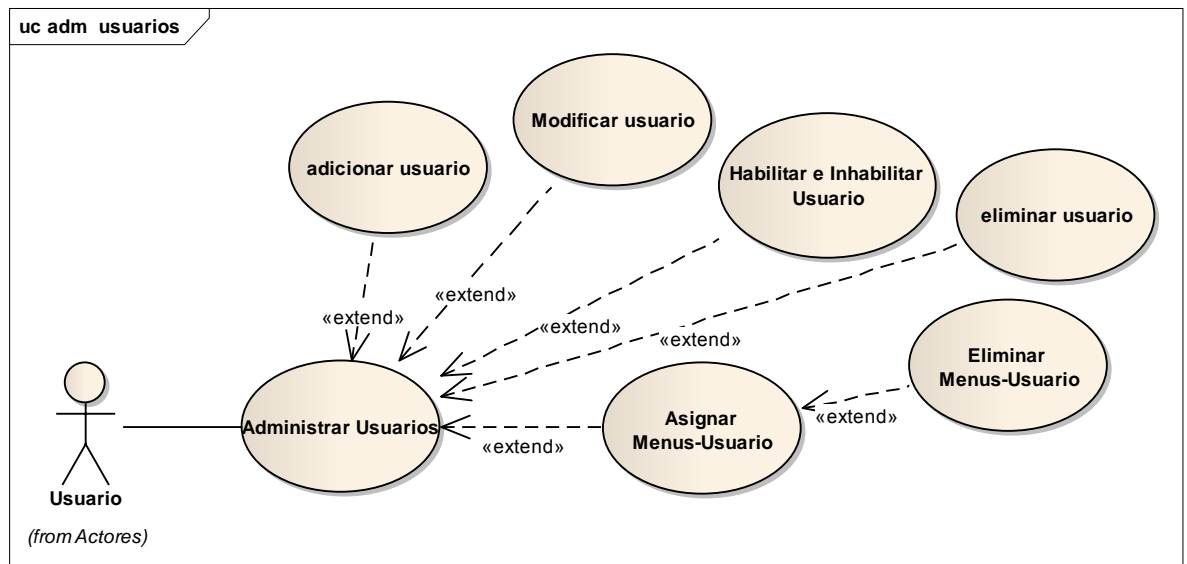


Fig. 22 Administrar Usuarios

Administrar Menús

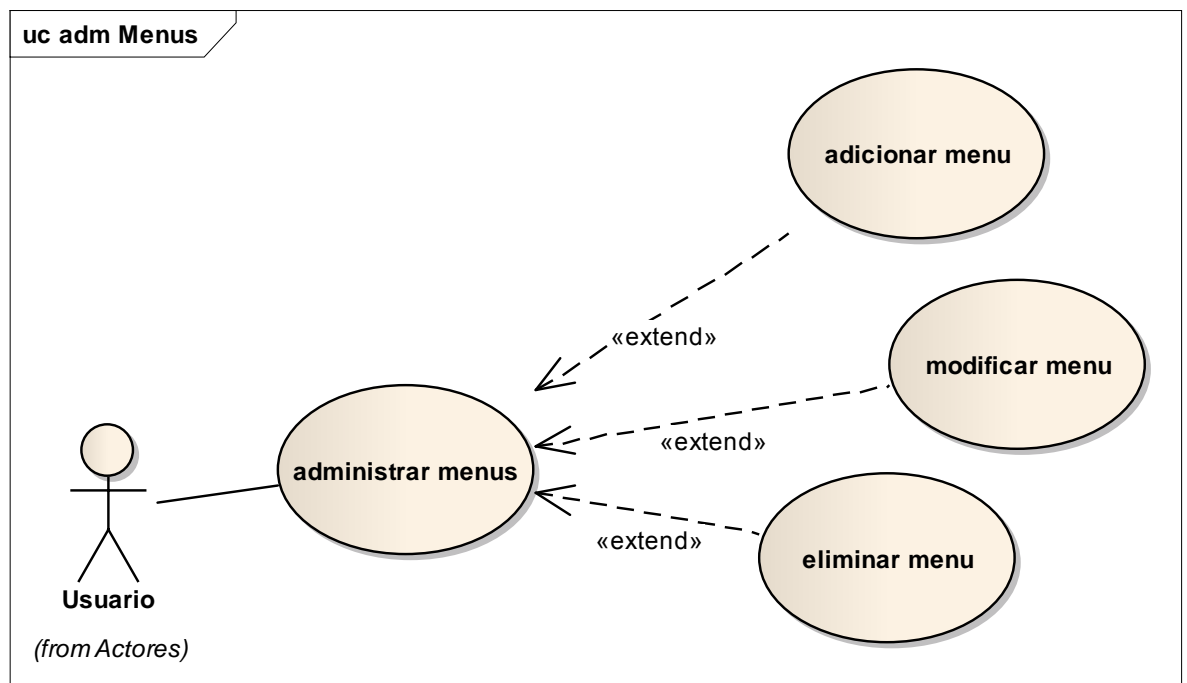


Fig. 23 Administrar Menús

Administrar Cargos

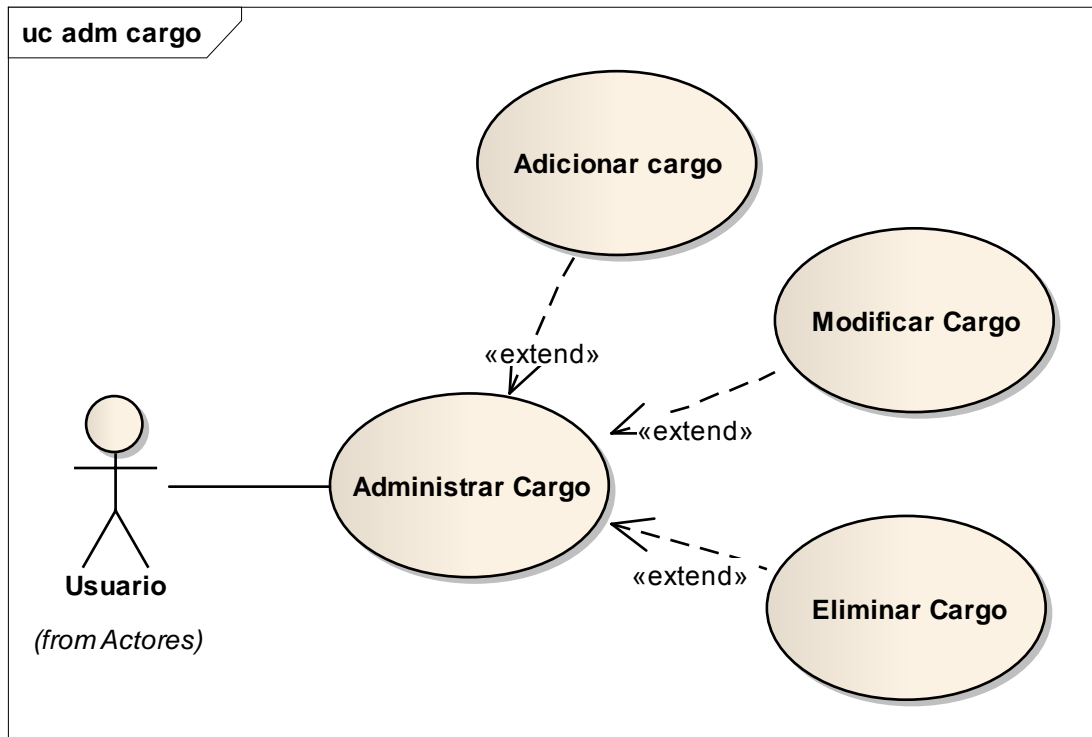


Fig. 24 Administrar Cargos

Generar Reportes

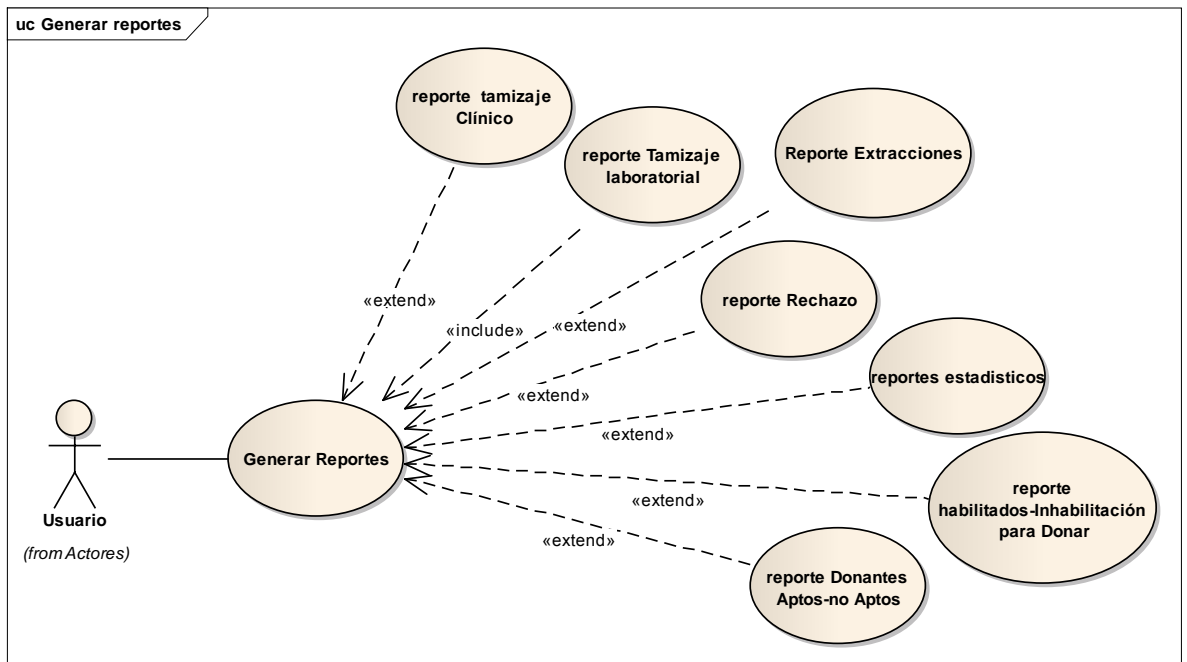


Fig. 25 Generar Reportes

II.1.3.3. Especificación de Casos de Uso

Para los Casos de Uso se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

II.1.3.3.1. Introducción

Las Especificaciones de los Casos de Uso es una descripción detallada de los Casos de Uso del Sistema.

II.1.3.3.1.1. Propósito

Interpretar y describir los Casos de Uso.

II.1.3.3.1.2. Alcance

Describe los procesos internos de los Casos de Uso.

Detalla los flujos de los Casos de Uso según lo establecido por la organización.

Especificación de casos de uso

Caso de uso:	Administrar donantes
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, ver, buscar y registrar una donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar donante. 2. El sistema muestra la lista de donantes y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

II.1.3.3.2. Especificación de los Casos de Uso

Especificación de Caso de Uso Acceso al sistema.

Caso de uso:	Acceso al sistema
Actores:	Administrador, Jefe del departamento, encargado de tamizajes o pruebas.
Tipo:	Básico.
Resumen :	En este Caso de uso el usuario introducirá su Login y clave válidos para ingresar al sistema.
Precondiciones:	ingresó al sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El usuario introduce su Login y clave. 2. El sistema valida los datos introducidos. 3. Si los datos son correctos muestra la pantalla principal 4. Fin de caso de uso.
Subflujos:	2. Mensaje Usuario Incorrecto.
Excepciones:	

Tabla 11 Especificación de Caso de Uso Acceso al sistema.

Especificación de Caso de Uso Administrar Donantes

Caso de uso:	Administrar donantes
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, ver, buscar y registrar una donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 5. El actor hace clic en administrar donante. 6. El sistema muestra la lista de donantes y sus opciones 7. El actor selecciona la opción que requiere en la fila de un donante 8. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 12 Especificación de Caso de Uso Administrar Donantes

Adicionar Donante

Caso de uso:	Adicionar Donante
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar un nuevo donante
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar donante
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos personales del donante 4. El actor hace clic en el botón siguiente 5. Se valida datos del formulario que son obligatorios 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 6.1 Mensaje de Error, Datos introducidos incorrectos. 7. el actor cancela la operación. 7.1. mostrar la pantalla Administrar donante
Excepciones:	

Tabla 13 Adicionar Donante

Adicionar Datos Adicionales

Caso de uso:	Adicionar datos adicionales
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar datos adicionales del donante
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Para ingresar en esta pantalla, antes se deberá registrar el donante y presionar siguiente, para recién aparecer está
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un formulario a llenar 2. El actor ingresa los datos adicionales del donante 3. El actor hace clic en el botón guardar 4. Se valida datos del formulario que son obligatorios 5. El sistema guarda los datos 6. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 3. El actor cancela la operación. 4. Mensaje de Error, Datos introducidos incorrectos. 5. Se muestra un mensaje, se guardó correctamente.
Excepciones:	

Tabla 14 Adicionar Datos Adicionales

Modificar Datos Donante

Caso de uso:	Modificar datos donante
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos del donante
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El donante debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar donante 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar donante 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 15 Modificar Datos Donante

Ver Historial

Caso de uso:	Ver historial
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el historial
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar donante
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en ver de una fila. 2. El sistema muestra el registro 3. El actor hace clic en regresar 4. El sistema muestra la pantalla administrar donante 5. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 16 Ver Historial

Buscar Donante

Caso de uso:	Buscar donante
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar un donante a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar donante
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce el ci o nombre o apellido o código del donante a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 17 Buscar Donante

Registrar Donación

Caso de uso:	Registrar donación
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar nueva donación para un donante
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar donante
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción registrar donación 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos 4. El actor hace clic en el botón siguiente 5. Se valida datos del formulario que son obligatorios 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. 4.1. mostrar la pantalla Administrar donante 5. Mensaje de Error, Datos introducidos incorrectos.
Excepciones:	No se permitirá registrar una nueva donación al donante que haya donado recientemente.

Tabla 18 Registrar Donación

Registrar Tamizaje Clínico

Caso de uso:	Registrar tamizaje clínico.
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar el tamizaje clínico después que se haya creado el registro de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe tener el registro de donación, para pasar recién a registrarse en el tamizaje clínico.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un formulario a llenar 2. El actor ingresa los datos 3. El actor hace clic en el botón siguiente 4. Se valida datos del formulario que son obligatorios 5. El sistema guarda los datos 6. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 3. el actor cancela la operación. 3.1. mostrar la pantalla Administrar donante 4. Mensaje de Error, Datos introducidos incorrectos o nulos.
Excepciones:	

Tabla 19 Registrar Tamizaje Clínico

Registrar Cuestionario

Caso de uso:	Registrar cuestionario
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar el cuestionario después que se haya registrado el tamizaje clínico.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el tamizaje clínico, para pasar recién a registrar el cuestionario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un cuestionario 2. El actor ingresa los respuestas 3. El actor hace clic en el botón siguiente 4. Se valida datos del formulario que son obligatorios 5. El sistema guarda los datos 6. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. Mensaje de Error, Datos introducidos incorrectos o nulos.
Excepciones:	

Tabla 20 Registrar Cuestionario

Registrar Tamizaje Laboratorial

Caso de uso:	Registrar tamizaje laboratorial
Actores:	Jefe del departamento, encargado de pruebas o exámenes
Tipo:	Básico.
Resumen :	Permite registrar el tamizaje laboratorial de la donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe tener registrado el tamizaje clínico y su cuestionario, para pasar recién a registrarse en el tamizaje laboratorial.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un formulario a llenar 2. El actor ingresa los datos 3. El actor hace clic en el botón siguiente 4. Se valida datos del formulario que son obligatorios 5. El sistema guarda los datos 6. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 3. el actor cancela la operación. 3.1. mostrar la pantalla Administrar donante 4. Mensaje de Error, Datos introducidos incorrectos o nulos.
Excepciones:	

Tabla 21 Registrar Tamizaje Laboratorial

Registrar Tamizaje Serológico

Caso de uso:	Registrar tamizaje serológico
Actores:	Jefe del departamento, encargado de pruebas o exámenes
Tipo:	Básico.
Resumen :	Permite registrar el tamizaje serológico
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe tener registrado el tamizaje laboratorial, para pasar recién a registrarse en el tamizaje serológico.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un formulario 2. El actor ingresa los datos 3. El actor hace clic en el botón siguiente 4. El sistema guarda los datos 5. Fin de caso de uso.
Subflujos:	
Excepciones:	

Tabla 22 Registrar Tamizaje Serológico

Registrar Extracción

Caso de uso:	Registrar extracción
Actores:	Jefe del departamento, encargado de pruebas o exámenes
Tipo:	Básico.
Resumen :	Permite registrarla extracción de una donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe tener registrado el tamizaje serológico, para pasar recién a registrar la extracción.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra un formulario a llenar 2. El actor ingresa los datos 3. El actor hace clic en el botón guardar 4. Se valida datos del formulario que son obligatorios 5. El sistema guarda los datos 6. El sistema regresa a la pantalla administrar donante. 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 3. el actor cancela la operación. 3.1. mostrar la pantalla Administrar donante 4. Mensaje de Error, Datos introducidos incorrectos o nulos.

Tabla 23 Registrar Extracción

Registrar Rechazo

Caso de uso:	Registrar rechazo
Actores:	Jefe del departamento, encargado de pruebas o exámenes
Tipo:	Básico.
Resumen :	Permite registrarrechazo de la donación en cualquier nivel de las pruebas.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrando una donación, es decir en el nivel de tamizaje clínico o laboratorio o extracción para recién ser rechazado.
Flujo Principal:	<ol style="list-style-type: none"> 1. Selecciona rechazar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario que son obligatorios 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. 4.1. mostrar la pantalla del nivel de prueba 5. Mensaje de Error, Datos introducidos incorrectos o nulo
Excepciones:	no cumple con los requisitos de una donación

Tabla 24 Registrar Rechazo

Especificación de Caso de Uso Administrar Tamizaje Clínico

Caso de uso:	Administrar Tamizaje Clínico
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, modificar, ver, buscar tamizaje clínico y ver cuestionario.
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar tamizaje clínico. 2. El sistema muestra la lista de los donantes que tienen tamizaje clínico y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 25 Especificación de Caso de Uso Administrar Tamizaje Clínico

Modificar Tamizaje Clínico

Caso de uso:	Modificar Tamizaje Clínico
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos del Tamizaje Clínico
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El Tamizaje Clínico debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar Tamizaje Clínico 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar Tamizaje Clínico 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 26 Modificar Tamizaje Clínico

Ver Tamizaje Clínico

Caso de uso:	Ver tamizaje clínico
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el registro de tamizaje clínico
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tamizaje clínico
Flujo Principal:	<ol style="list-style-type: none"> 5. El actor hace clic en ver de una fila. 6. El sistema muestra el registro 7. El actor hace clic en regresar 8. El sistema muestra la pantalla administrar tamizaje clínico 9. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 27 Ver Tamizaje Clínico

Ver Cuestionario

Caso de uso:	Ver cuestionario
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el cuestionario
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tamizaje clínico
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en ver de una fila. 2. El sistema muestra el registro 3. El actor hace clic en regresar 4. El sistema muestra la pantalla administrar tamizaje clínico 5. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 28 Ver Cuestionario

Buscar Tamizaje Clínico

Caso de uso:	Buscar tamizaje clínico
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar un tamizaje clínico a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tamizaje clínico
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce algún dato del tamizaje clínico a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 29 Buscar Tamizaje Clínico

Especificación de Caso de Uso Administrar Tamizaje Laboratorial

Caso de uso:	Administrar Tamizaje Laboratorial
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, modificar, ver, buscar tamizaje laboratorial.
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar tamizaje laboratorial. 2. El sistema muestra la lista de los donantes que tienen tamizaje laboratorial y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 30 Especificación de Caso de Uso Administrar Tamizaje Laboratorial

Modificar Tamizaje Laboratorial

Caso de uso:	Modificar Tamizaje Laboratorial
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos del Tamizaje Laboratorial
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El Tamizaje Laboratorial debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar Tamizaje Laboratorial 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar Tamizaje Laboratorial 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 31 Modificar Tamizaje Laboratorial

Ver Tamizaje Laboratorial

Caso de uso:	Ver tamizaje laboratorial
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el registro del tamizaje laboratorial
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tamizaje laboratorial
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en ver de una fila. 2. El sistema muestra el registro 3. El actor hace clic en regresar 4. El sistema muestra la pantalla administrar tamizaje laboratorial 5. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 32 Ver Tamizaje Laboratorial

Buscar Tamizaje Laboratorial

Caso de uso:	Buscar tamizaje laboratorial
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar un tamizaje laboratorial a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tamizaje laboratorial
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce algún dato del tamizaje laboratorial a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 33 Buscar Tamizaje Laboratorial

Especificación de Caso de Uso Administrar Extracción

Caso de uso:	Administrar Extracción
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, modificar, ver, buscar tamizaje clínico.
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar extracción. 2. El sistema muestra la lista de los donantes que tienen extracción y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 34 Especificación de Caso de Uso Administrar Extracción

Modificar Extracción

Caso de uso:	Modificar extracción
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos de la extracción
Precondiciones:	El actor debe estar autenticado y validado en el sistema. La extracción debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar extracción 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar extracción 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 35 Modificar Extracción

Ver Extracción

Caso de uso:	Ver extracción
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el registro de extracción
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar extracción
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en ver de una fila. 2. El sistema muestra el registro 3. El actor hace clic en regresar 4. El sistema muestra la pantalla administrar extracción 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor hace clic en imprimir
Excepciones:	

Tabla 36 Ver Extracción

Buscar Extracción

Caso de uso:	Buscar extracción
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar una extracción a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar extracción
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce algún dato de la extracción a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 37 Buscar Extracción

Especificación de Caso de Uso Administrar Donaciones

Caso de uso:	Administrar Donaciones
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar los registros de donación, completar, ver resultados, buscar donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar donaciones. 2. El sistema muestra la lista de los donantes que tienen registros de donaciones recientes y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 38 Especificación de Caso de Uso Administrar Donaciones

Completar Donación

Caso de uso:	Completar donación
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite completar las pruebas para una donación completa
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe existir un registro de donación.
Flujo Principal:	<ol style="list-style-type: none"> 1. El sistema muestra lista de las pruebas que no se realizaron 2. El actor hace clic en la prueba 3. El sistema muestra el formulario de la prueba y consecutivamente las demás pruebas 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 39 Completar Donación

Ver Resultado de Donación

Caso de uso:	Ver resultado de donación
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el registro de resultado de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar donaciones
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en ver de una fila. 2. El sistema muestra el registro 3. El actor hace clic en regresar 4. El sistema muestra la pantalla administrar donaciones 5. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 40 Ver Resultado de Donación

Buscar Registro Donación

Caso de uso:	Buscar registro donación
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar un registro donación a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar registro donación
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce algún dato del registro donación a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 41 Buscar Registro Donación

Especificación de Caso de Uso Administrar Rechazo

Caso de uso:	Administrar Rechazo
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, modificar, ver, buscar el rechazo
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar rechazo. 2. El sistema muestra la lista de los donantes que tienen algún rechazo y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 42 Especificación de Caso de Uso Administrar Rechazo

Modificar Rechazo

Caso de uso:	Modificar rechazo
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos del rechazo
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El rechazo debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar rechazo 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar rechazo 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 43 Modificar Rechazo**Ver Rechazo**

Caso de uso:	Ver rechazo
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite ver el registro de rechazo
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar rechazo
Flujo Principal:	<ol style="list-style-type: none"> 6. El actor hace clic en ver de una fila. 7. El sistema muestra el registro 8. El actor hace clic en regresar 9. El sistema muestra la pantalla administrar rechazo 10. Fin del caso de uso
Subflujos:	3. el actor hace clic en imprimir
Excepciones:	

Tabla 44 Ver Rechazo

Buscar Rechazo

Caso de uso:	Buscar rechazo
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite buscar un rechazo a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar rechazo
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor introduce algún dato del rechazo a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 45 Buscar Rechazo

Especificación de Caso de Uso Administrar Pregunta.

Caso de uso:	Administrar pregunta
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar el pregunta
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar pregunta. 2. El sistema muestra la lista de las preguntas y sus opciones 3. El actor selecciona la opción que requiere en la fila de un donante 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 46 Especificación de Caso de Uso Administrar Pregunta

Adicionar Pregunta

Caso de uso:	Adicionarpregunta
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite registrar una nuevapregunta
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar pregunta
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos de la pregunta. 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar donante 5. Mensaje de Error, Datos introducidos incorrectos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 47 Adicionar Pregunta

Modificar Pregunta

Caso de uso:	Modificar pregunta
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite modificar los datos de la pregunta
Precondiciones:	El actor debe estar autenticado y validado en el sistema. La pregunta debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar pregunta 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar pregunta 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 48 Modificar Pregunta

Eliminar Pregunta

Caso de uso:	Eliminar pregunta
Actores:	Jefe del departamento, encargado de pruebas
Tipo:	Básico.
Resumen :	Permite eliminar del sistema la pregunta
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema la pregunta Debe estar en la pantalla administrar pregunta
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar en la fila de una pregunta 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar pregunta Fin del caso de uso
Subflujos:	3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 49 Eliminar Pregunta

Especificación de Caso de Uso Administrar Tipo Bolsa

Caso de uso:	Administrar tipo bolsa
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar y buscar tipo bolsa
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar tipo bolsa. 2. El sistema muestra la lista de los tipos de bolsa y sus opciones 3. El actor selecciona la opción que requiere en la fila de un tipo de bolsa 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 50 Especificación de Caso de Uso Administrar Tipo Bolsa

Adicionar Tipo Bolsa

Caso de uso:	Adicionartipo bolsa
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar tipos de bolsa
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar tipo de bolsa
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del tipo de bolsa. 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar tipo de bolsa 5.Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente

Tabla 51 Adicionar Tipo Bolsa

Modificar Tipo Bolsa

Caso de uso:	Modificar tipo bolsa
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del tipo de bolsa
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El tipo de bolsa debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar tipo bolsa 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar tipo bolsa 6. Mensaje, los datos se modificaron correctamente.

Tabla 52 Modificar Tipo Bolsa

Eliminar Tipo Bolsa

Caso de uso:	Eliminar tipobolsa
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el tipobolsa
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el tipo debolsa Debe estar en la pantalla administrar tipobolsa
Flujo Principal:	<ol style="list-style-type: none"> 5. El actorselecciona la opción eliminar en la fila de un tipo de bolsa 6. El sistema muestra mensaje de confirmación 7. El actor selecciona si para la eliminación. 8. El sistema regresa a la pantalla administrar tipo bolsa 9. Fin del caso de uso
Subflujos:	3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 53 Eliminar Tipo Bolsa**Buscar Tipo Bolsa**

Caso de uso:	Buscar tipo bolsa
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite buscar un tipo bolsa a la vez
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la pantalla administrar tipo bolsa
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorintroduce algún dato del tipo de bolsa a buscar 2. El sistema muestra los datos 3. El actor selecciona la opción ver todos 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 54 Buscar Tipo Bolsa

Especificación de Caso de Uso Administrar Tipo Donación

Caso de uso:	Administrar tipo donación
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar tipo donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar tipo donación. 2. El sistema muestra la lista de los tipos de donación y sus opciones. 3. El actor selecciona la opción que requiere en la fila de un tipo de donación. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 55 Especificación de Caso de Uso Administrar Tipo Donación

Adicionar Tipo Donación

Caso de uso:	Adicionar tipo donación.
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar tipos de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar tipo donación
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del tipo de donación. 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. <ol style="list-style-type: none"> 4.1. mostrar la pantalla Administrar tipo de donación 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 56 Adicionar Tipo Donación

Modificar Tipo Donación

Caso de uso:	Modificar tipo donación
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del tipo de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El tipo de donación debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar tipo donación 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar tipo donación 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 57 Modificar Tipo Donación

Eliminar Tipo Donación

Caso de uso:	Eliminar tipo Donación
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el tipo Donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el tipo de Donación Debe estar en la pantalla administrar tipo Donación
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de un tipo de donación 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar tipo Donación 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.

Tabla 58 Eliminar Tipo Donación

Especificación de Caso de Uso Administrar Lugar de Donación

Caso de uso:	Administrar lugar de donación
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar lugar de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar lugar de donación. 2. El sistema muestra la lista de los lugares de donación y sus opciones. 3. El actor selecciona la opción que requiere, en la fila de un lugar de donación. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 59 Especificación de Caso de Uso Administrar Lugar de Donación

Adicionar Lugar Donación

Caso de uso:	Adicionar lugar donación
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar lugar de donación.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar lugar de donación
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del lugar de donación 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. 4.1. mostrar la pantalla Administrar lugar de donación 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 60 Adicionar Lugar Donación

Modificar Lugar Donación

Caso de uso:	Modificar lugar donación
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del lugar de donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El lugar de donación debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar lugar donación 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar lugar donación 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 61 Modificar Lugar Donación

Eliminar Lugar Donación

Caso de uso:	Eliminar Lugar Donación
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el Lugar Donación
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el Lugar Donación Debe estar en la pantalla administrar Lugar Donación
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de un Lugar 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar Lugar Donación 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.

Tabla 62 Eliminar Lugar Donación

Especificación de Caso de Uso Administrar Departamentos

Caso de uso:	Administrar departamentos
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar departamentos
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar departamentos. 2. El sistema muestra la lista de los departamentos junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de un departamento. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 63 Especificación de Caso de Uso Administrar Departamentos

Adicionar Departamento

Caso de uso:	Adicionar departamento
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar departamentos
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar departamento
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del departamento 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. <ol style="list-style-type: none"> 4.1. mostrar la pantalla Administrar departamento 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 64 Adicionar Departamento

Modificar Departamento

Caso de uso:	Modificar departamento
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del departamento
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El departamento debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar departamento 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar departamento 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 65 Modificar Departamento

Eliminar Departamento

Caso de uso:	Eliminar departamento
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el departamento
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el departamento Debe estar en la pantalla administrar departamento
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de la departamento 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar departamento 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.

Tabla 66 Eliminar Departamento

Especificación de Caso de Uso Administrar Provincia

Caso de uso:	Administrar provincia
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar provincia
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar provincia. 2. El sistema muestra la lista de las provincias junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de una provincia. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 67 Especificación de Caso de Uso Administrar Provincia

Adicionar Provincia

Caso de uso:	Adicionarprovincia
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar provincia
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar provincia.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del provincia 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar provincia 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 68 Adicionar Provincia

Modificar Provincia

Caso de uso:	Modificar provincia
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos de la provincia
Precondiciones:	El actor debe estar autenticado y validado en el sistema. La provincia debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar provincia 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar provincia 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 69 Modificar Provincia**Eliminar Provincia**

Caso de uso:	Eliminar provincia
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema la provincia
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema la provincia Debe estar en la pantalla administrar provincia
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de la provincia 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar provincia 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 70 Eliminar Provincia

Especificación de Caso de Uso Administrar ciudad

Caso de uso:	Administrar ciudad
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar ciudad
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar ciudad. 2. El sistema muestra la lista de las ciudades junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de una ciudad. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 71 Especificación de Caso de Uso Administrar ciudad

Adicionar Ciudad

Caso de uso:	Adicionarciudad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar ciudad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar ciudad.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos de la ciudad 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar ciudad 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 72 Adicionar Ciudad

Modificar Ciudad

Caso de uso:	Modificar ciudad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos de la ciudad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. La ciudad debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar ciudad 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar ciudad 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 73 Modificar Ciudad

Eliminar Ciudad

Caso de uso:	Eliminar ciudad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema la ciudad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema la ciudad Debe estar en la pantalla administrar ciudad
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de la ciudad 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar ciudad 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.

Tabla 74 Eliminar Ciudad

Especificación de Caso de Uso Administrar Zonas

Caso de uso:	Administrar zonas
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar zona
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar zona. 2. El sistema muestra la lista de las zonas junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de una provincia. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 75 Especificación de Caso de Uso Administrar Zonas

Adicionar Zona

Caso de uso:	Adicionar ciudad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar ciudad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar ciudad
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos de la ciudad 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. <ol style="list-style-type: none"> 4.1. mostrar la pantalla Administrar ciudad 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 76 Adicionar Zona

Modificar Zona

Caso de uso:	Modificar zona
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos de la zona
Precondiciones:	El actor debe estar autenticado y validado en el sistema. La zona debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar zona 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar zona 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 77 Modificar Zona**Eliminar Zona**

Caso de uso:	Eliminar zona
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema la zona
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema la zona Debe estar en la pantalla administrar zona
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar en la fila de la zona 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar zona 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.

Tabla 78 Eliminar Zona

Especificación de Caso de Uso Administrar Enfermedades

Caso de uso:	Administrar enfermedades
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar enfermedades
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar enfermedades. 2. El sistema muestra la lista de las enfermedades junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de una enfermedad. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 79 Especificación de Caso de Uso Administrar Enfermedades

Adicionar Enfermedad

Caso de uso:	Adicionar enfermedad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar enfermedad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar enfermedad
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos de la enfermedad 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4. el actor cancela la operación. <ol style="list-style-type: none"> 4.1. mostrar la pantalla Administrar enfermedad 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 80 Adicionar Enfermedad

Modificar Enfermedad

Caso de uso:	Modificar enfermedad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos de la enfermedad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El enfermedad debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar enfermedad 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar enfermedad 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 81 Modificar Enfermedad

Eliminar Enfermedad

Caso de uso:	Eliminar enfermedad
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema la enfermedad
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema la enfermedad Debe estar en la pantalla administrar enfermedad
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar de la fila del enfermedad 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar enfermedad 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, no realiza ninguna operación.

Tabla 82 Eliminar Enfermedad

Especificación de Caso de Uso Administrar Usuarios

Caso de uso:	Administrar usuarios
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar, habilitar e inhabilitar usuario y asignar menú
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar usuarios. 2. El sistema muestra la lista de los usuarios junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de un usuario. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 83 Especificación de Caso de Uso Administrar Usuarios

Adicionar Usuario

Caso de uso:	Adicionarusuario
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar nuevos usuarios
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar usuario
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del usuario 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar usuario 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente
Excepciones:	

Tabla 84 Adicionar Usuario

Modificar Usuario

Caso de uso:	Modificar usuario
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del usuario
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El usuario debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar usuario 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar usuario 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 85 Modificar Usuario

Eliminar Usuario

Caso de uso:	Eliminar usuario
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema al usuario
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el usuario Debe estar en la pantalla administrar usuario
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción eliminar de la fila del usuario 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar usuario 5. Fin del caso de uso
Subflujos:	<ol style="list-style-type: none"> 3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	No se puede eliminar el mismo usuario que esta sesionado

Tabla 86 Eliminar Usuario

Habilitar e Inhabilitar Usuario

Caso de uso:	Habilitar e Inhabilitar Usuario
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite Habilitar e Inhabilitar al Usuario.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el usuario Debe estar en la pantalla administrar usuario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción de Habilitar o Inhabilita en la fila del usuario. 2. El sistema cambiara de habilitado a Inhabilitado y lo contrario. 3. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 87 Habilitar e Inhabilitar Usuario

Asignar Menús – Usuario

Caso de uso:	Asignar menú - usuario.
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite asignar menús al usuario.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el menú. Debe estar en la pantalla administrar usuario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actorselecciona la opción de asignar menú en la fila del usuario. 2. El sistema muestra formulario 3. Selecciona botón guardar. 4. El sistema guarda la asignación. 5. El sistema muestra mensaje de confirmación 6. El sistema regresa a la pantalla administrar usuario. 7. Fin del caso de uso
Subflujos:	3. el actor selecciona botón cancelar, se regresa a la pantalla administrar usuario.
Excepciones:	

Tabla 88 Asignar Menús – Usuario

Eliminar Menús – Usuario

Caso de uso:	Eliminar menú del usuario.
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el menú del usuario.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el menú con el usuario. Debe estar en la pantalla asignar menús-usuario.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar en la fila del menú. 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla asignar menú usuario. 5. Fin del caso de uso
Subflujos:	3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 89 Eliminar Menús – Usuario

Especificación de Caso de Uso Administrar Menús

Caso de uso:	Administrar menú
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar menú.
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar menú. 2. El sistema muestra la lista de los menús junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de un menú. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 90 Especificación de Caso de Uso Administrar Menús

Adicionar Menú

Caso de uso:	Adicionar menú
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar menús
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar menú
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del menú 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar menú 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente

Tabla 91 Adicionar Menú

Modificar Menú

Caso de uso:	Modificar menú
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del menú
Precondiciones:	El actor debe estar autenticado y validado en el sistema. El menú debe estar registrado en el sistema
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar menú 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar menú 6. Mensaje, los datos se modificaron correctamente 6. Se muestra el mensaje: se guardó correctamente

Tabla 92 Modificar Menú

Eliminar Menú

Caso de uso:	Eliminar menú.
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el menú.
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el menú. Debe estar en la pantalla administrar menú.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar de la fila del menú. 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar menú. 5. Fin del caso de uso
Subflujos:	3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 93 Eliminar Menú

Especificación de Caso de Uso Administrar Cargos

Caso de uso:	Administrar cargos
Actores:	Administrador
Tipo:	Básico.
Resumen :	Permite listar, adicionar, modificar, eliminar cargos
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en administrar cargos. 2. El sistema muestra la lista de los cargos junto a sus opciones. 3. El actor selecciona la opción que requiere en la fila de un cargo. 4. Fin del caso de uso
Subflujos:	
Excepciones:	

Tabla 94 Especificación de Caso de Uso Administrar Cargos

Adicionar Cargo

Caso de uso:	Adicionar cargo
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite registrar cargo
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar en la interfaz administrar cargo
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción adicionar 2. El sistema muestra un formulario a llenar 3. El actor ingresa los datos del cargo 4. El actor hace clic en el botón guardar 5. Se valida datos del formulario 6. El sistema guarda los datos 7. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar cargo 5. Mensaje de Error, Datos introducidos incorrectos o por nulos. 6. Se muestra el mensaje: se guardó correctamente

Tabla 95 Adicionar Cargo

Modificar Cargo

Caso de uso:	Modificar cargo
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite modificar los datos del cargo
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el cargo
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor hace clic en la opción modificar 2. El sistema muestra un formulario con datos 3. El actor realiza los cambios de datos 4. El actor hace clic en el botón modificar 5. Se valida datos del formulario 6. El sistema actualiza los datos del formulario 7. El sistema regresa a la pantalla administrar cargo 8. Fin de caso de uso.
Subflujos:	<ol style="list-style-type: none"> 4.el actor cancela la operación. 4.1. mostrar la pantalla Administrar cargo 6.Mensaje, los datos se modificaron correctamente
Excepciones:	

Tabla 96 Modificar Cargo

Eliminar Cargo

Caso de uso:	Eliminar cargo
Actores:	administrador
Tipo:	Básico.
Resumen :	Permite eliminar del sistema el cargo
Precondiciones:	El actor debe estar autenticado y validado en el sistema. Debe estar registrado en el sistema el cargo. Debe estar en la pantalla administrar cargos
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor selecciona la opción eliminar de la fila del cargo 2. El sistema muestra mensaje de confirmación 3. El actor selecciona si para la eliminación. 4. El sistema regresa a la pantalla administrar cargos 5. Fin del caso de uso
Subflujos:	3. el actor selecciona no, el sistema no realiza ninguna operación.
Excepciones:	

Tabla 97 Eliminar Cargo

Especificación de Caso de Uso Generar Reportes

Reporte Tamizaje Clínico

Caso de uso:	Reporte Tamizaje Clínico
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de todos los tamizajes clínico dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte tamizaje clínico 2. El sistema muestra el reporte 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 98 Reporte Tamizaje Clínico

Reporte Tamizaje Laboratorial

Caso de uso:	Reporte Tamizaje laboratorial
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de todos los tamizajes de laboratorial dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte tamizaje laboratorial 2. El sistema muestra el reporte 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 99 Reporte Tamizaje Laboratorial

Reporte Extracciones

Caso de uso:	Reporte extracciones
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de todas las extracciones dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte extracciones 2. El sistema muestra el reporte 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 100 Reporte Extracciones

Reporte Rechazo

Caso de uso:	Reporte rechazo
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de todos los rechazos dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte rechazo 2. El sistema muestra el reporte 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 101 Reporte Rechazo**Reporte Estadísticos**

Caso de uso:	Reporte estadísticos
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte estadísticos o gráficos dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte estadísticos. 2. El sistema muestra el reporte 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema y el tipo de dato.
Excepciones:	

Tabla 102 Reporte Estadísticos

Reporte Habilitados- Inhabilitados para Donar

Caso de uso:	Reporte Habilitados- Inhabilitados para Donar
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de Habilitados para donar nuevamente e Inhabilitados para Donar, dentro de rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte Habilitados- Inhabilitados para Donar. 2. El sistema muestra el reporte. 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 103 Reporte Habilitados- Inhabilitados para Donar

Reporte Donantes Aptos – No Aptos

Caso de uso:	Reporte Donantes Aptos – No Aptos.
Actores:	Jefe del departamento, encargado de pruebas.
Tipo:	Básico.
Resumen :	Permite visualizar reporte de Donantes Aptos – No Aptos, dentro del rango de fechas actuales o se puede consultar según requiera
Precondiciones:	El actor debe estar autenticado y validado en el sistema.
Flujo Principal:	<ol style="list-style-type: none"> 1. El actor presiona en reporte Donantes Aptos – No Aptos 2. El sistema muestra el reporte. 3. El actor elige la opción de imprimir 4. Fin del caso de uso
Subflujos:	2. el actor selecciona las fechas que requiera el reporte y lo muestra el sistema
Excepciones:	

Tabla 104 Reporte Donantes Aptos – No Aptos

II.1.4. MODELADO DE DIAGRAMA DE ACTIVIDADES

II.1.4.1. Introducción

El Diagrama de Actividades es uno más de los diagramas requeridos por la metodología RUP la cual estamos implementando.

Los diagramas de actividad se utilizan para modelar los aspectos dinámicos de un sistema, esto implica modelar los pasos secuenciales de un proceso.

II.1.4.1.1. Propósito

- Comprender la estructura y la dinámica del sistema deseado para la organización
- Identificar posibles mejoras

II.1.4.1.2. Alcance

- Describir los procesos de sistema.
- Identificar y definir los *procesos de los casos de uso* según los *objetivos* de la organización.
- Definir un *diagrama de actividad* para cada caso de uso del sistema.

II.1.4.2. Diagramas de Actividades

Diagrama de Actividades Caso de uso Acceso al sistema.

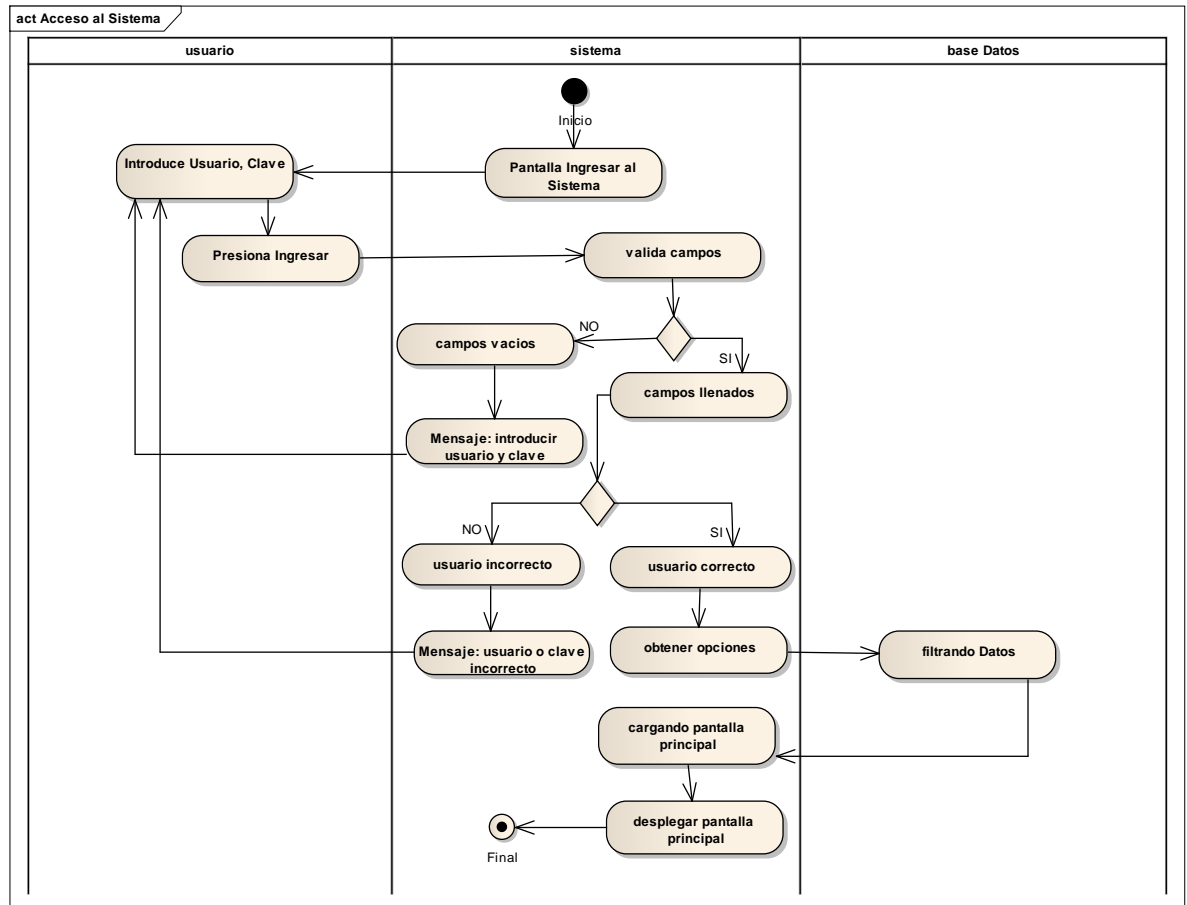


Fig. 26 Diagrama de Actividades Caso de uso Acceso al sistema.

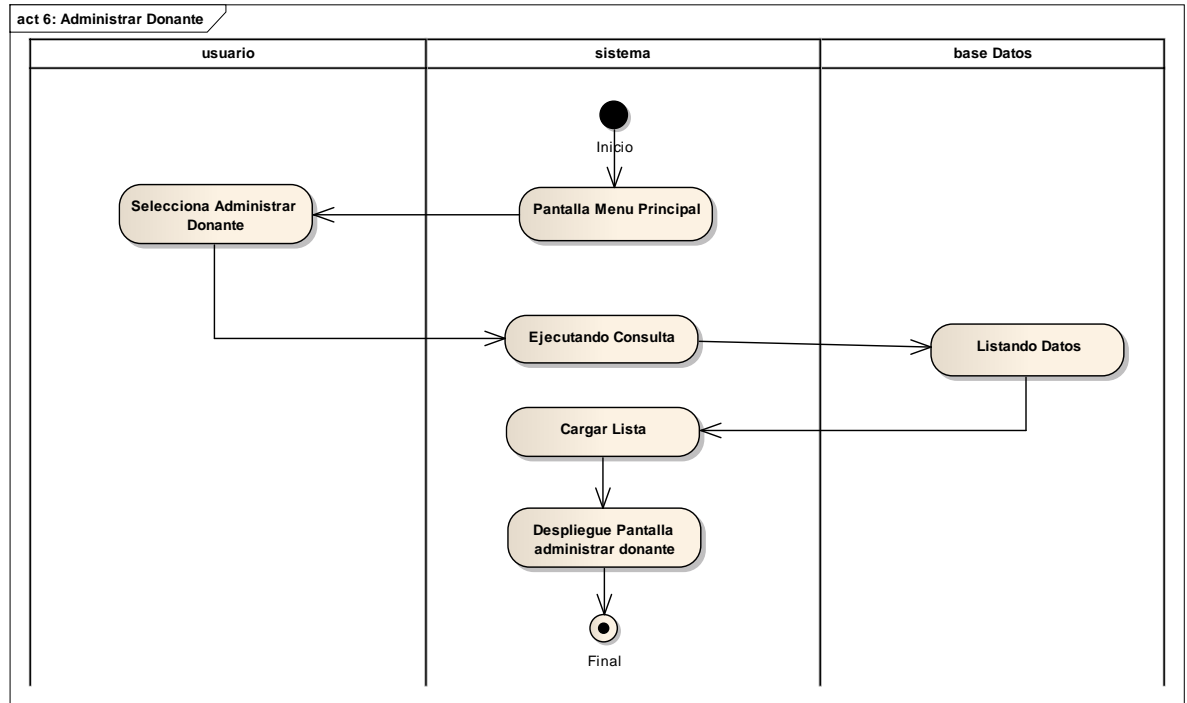
Diagrama de Actividades Caso de uso Administrar Donantes**Fig. 27 Diagrama de Actividades Caso de uso Administrar Donantes**

Diagrama de Actividades Caso de uso Adicionar Donante

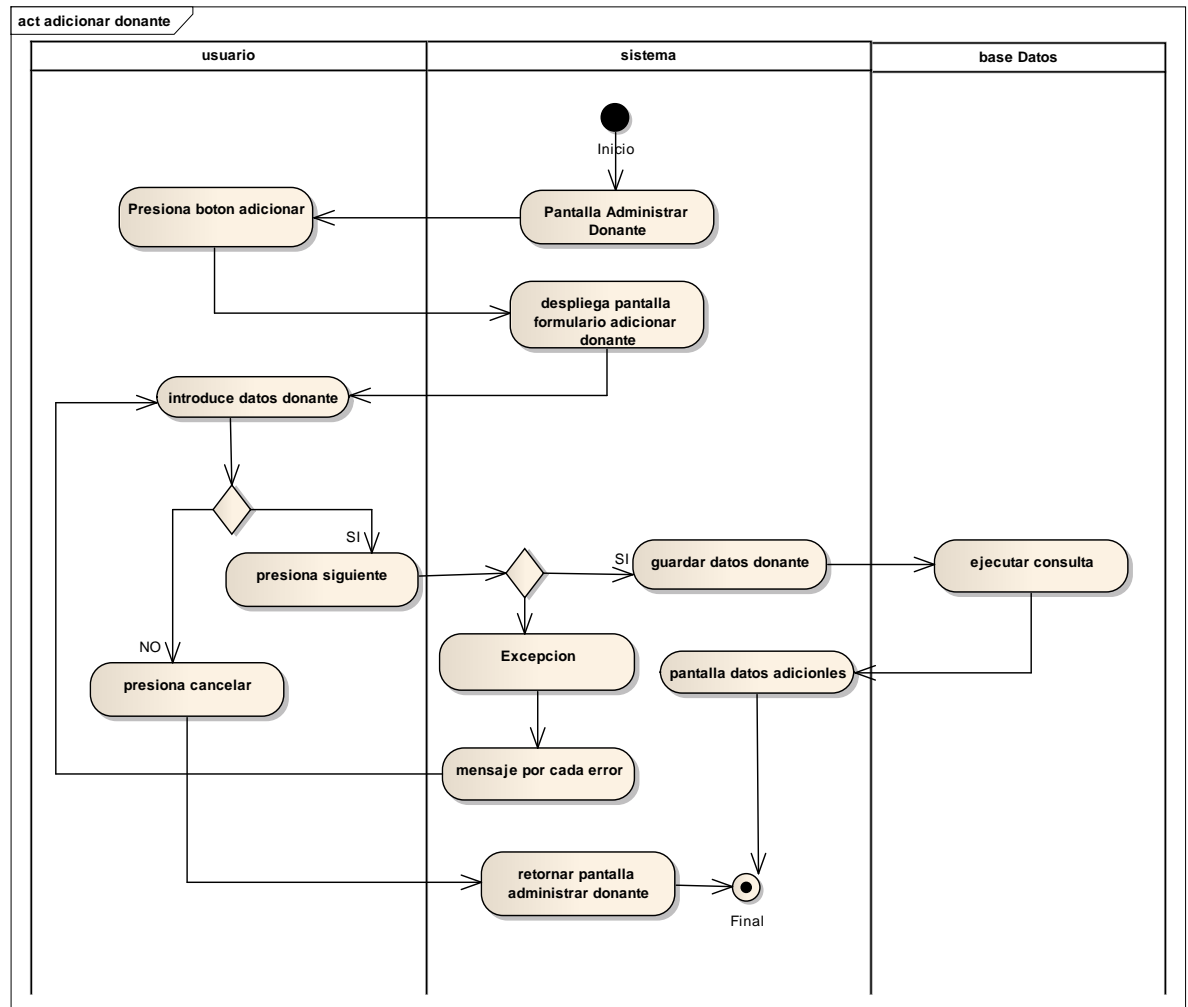


Fig. 28 Diagrama de Actividades Caso de uso Adicionar Donante

Diagrama de Actividades Caso de uso Adicionar Datos Adicionales

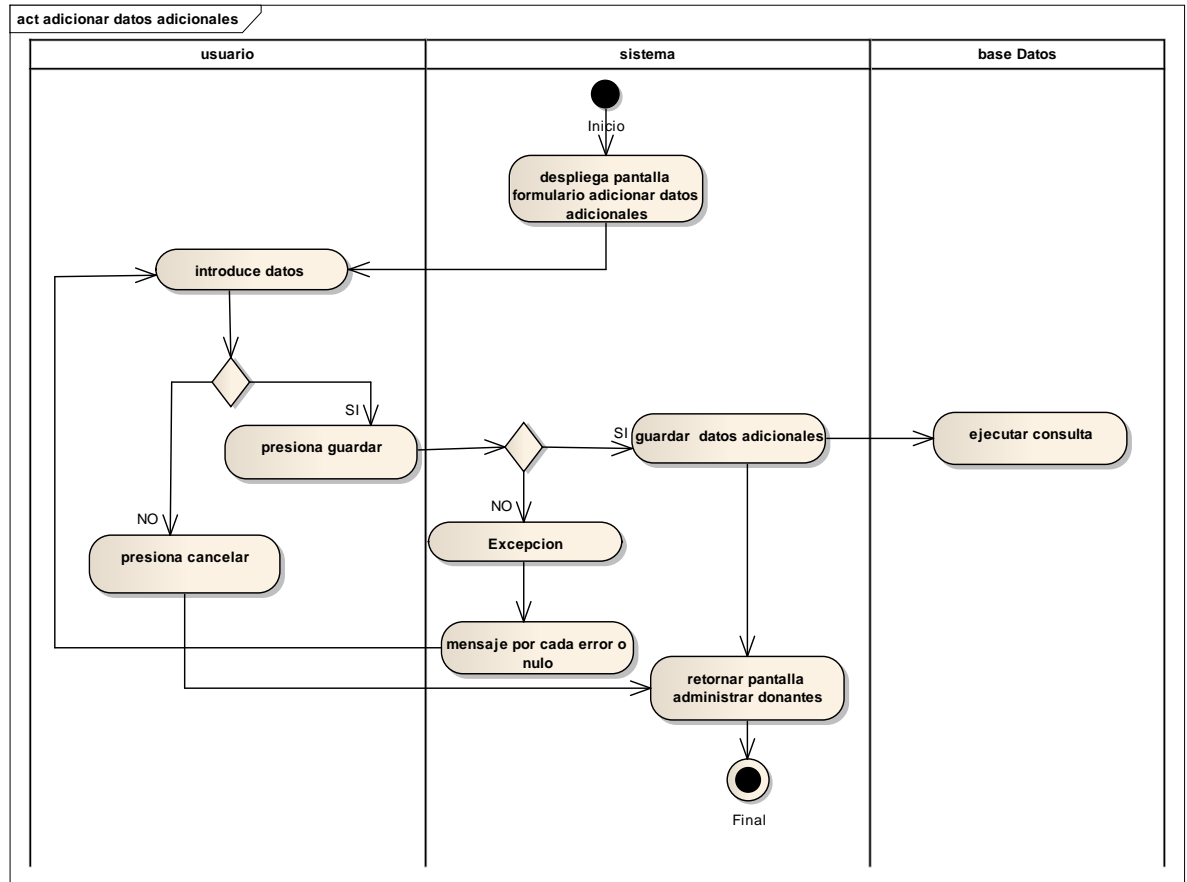


Fig. 29 Diagrama de Actividades Caso de uso Adicionar Datos Adicionales

Diagrama de Actividades Caso de uso Modificar Datos Donante

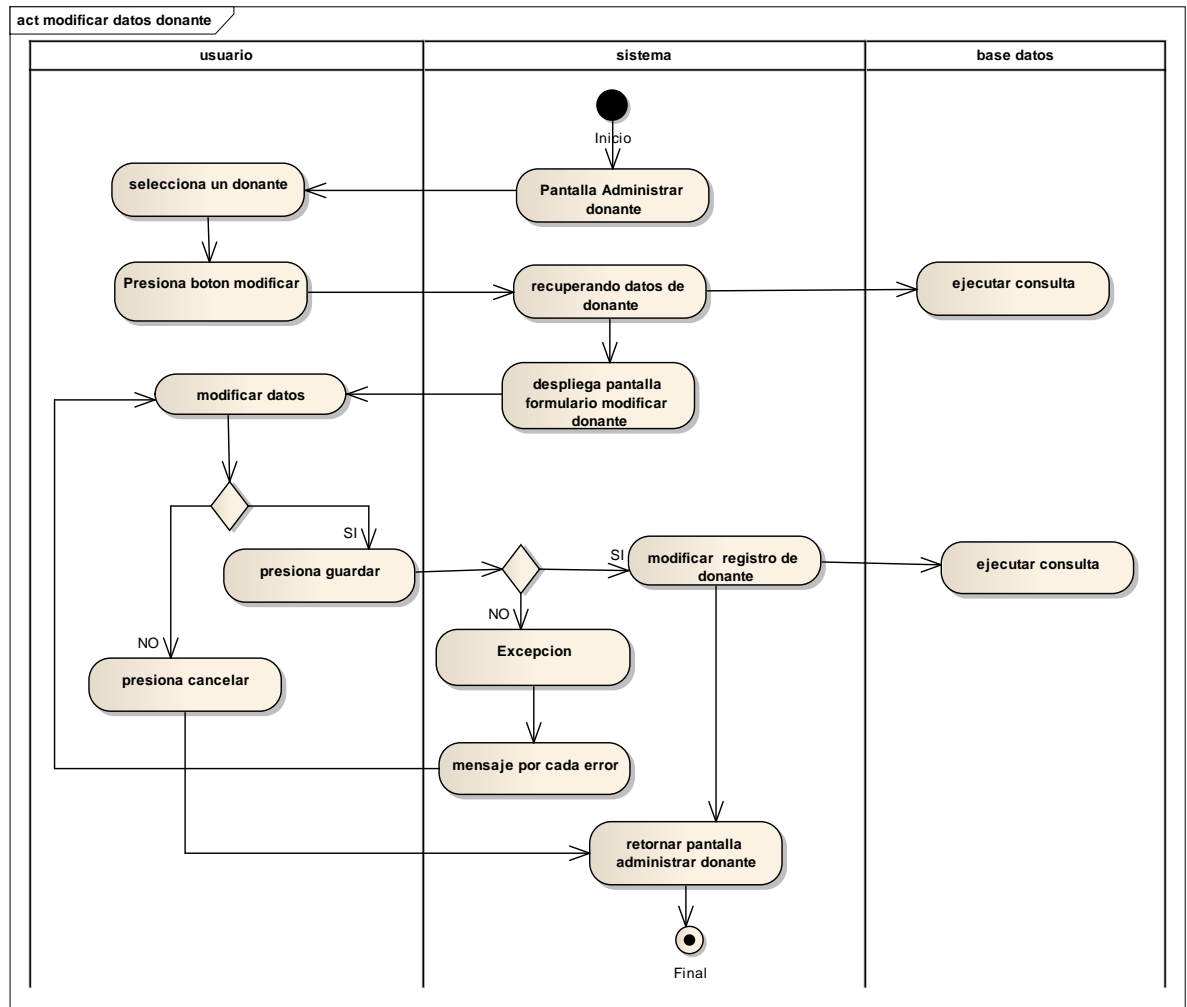


Fig. 30 Diagrama de Actividades Caso de uso Modificar Datos Donante

Diagrama de Actividades Caso de uso Ver Historial

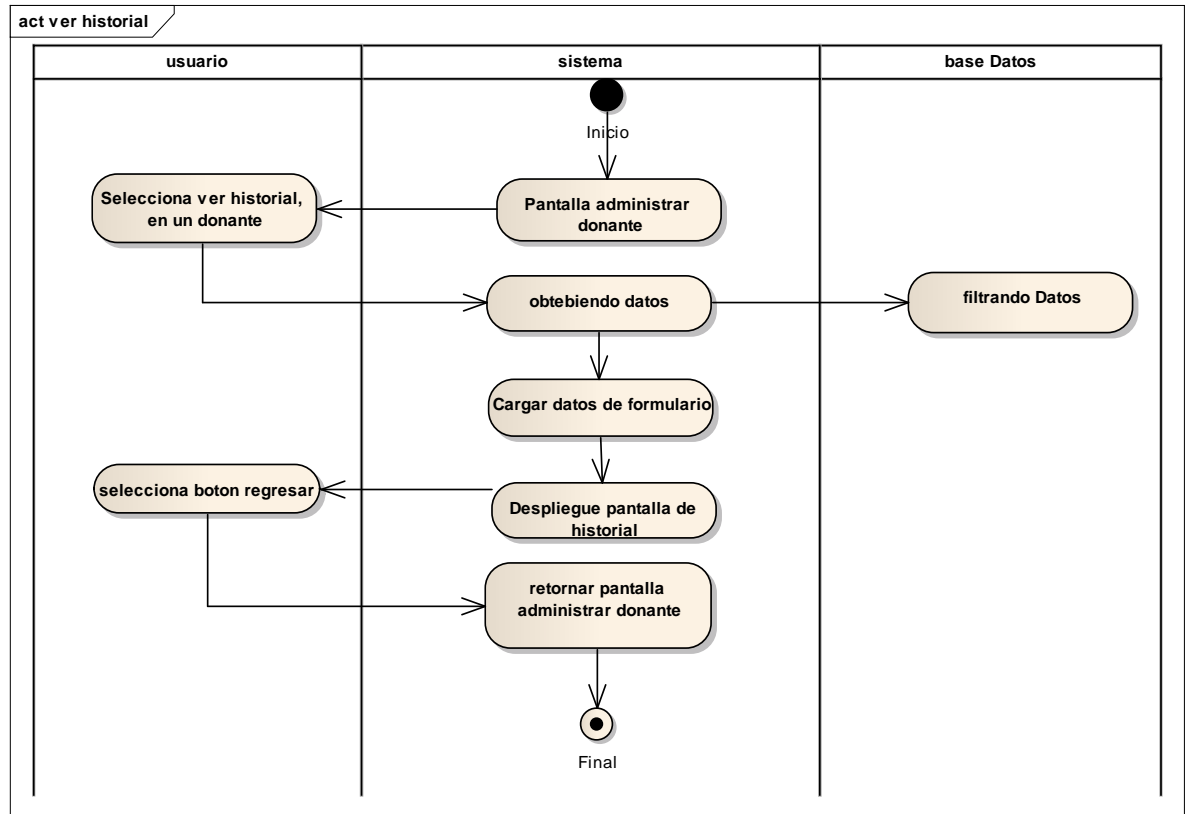


Fig. 31 Diagrama de Actividades Caso de uso Ver Historial

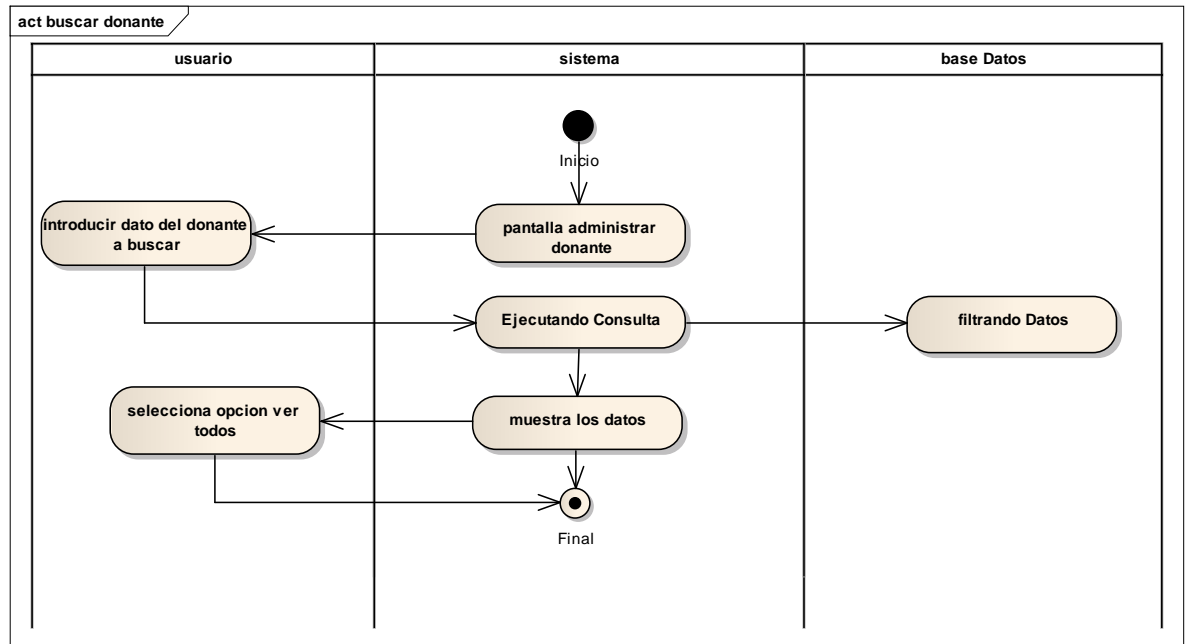
Diagrama de Actividades Caso de uso Buscar Donante**Fig. 32 Diagrama de Actividades Caso de uso Buscar Donante**

Diagrama de Actividades Caso de uso Registrar Donación

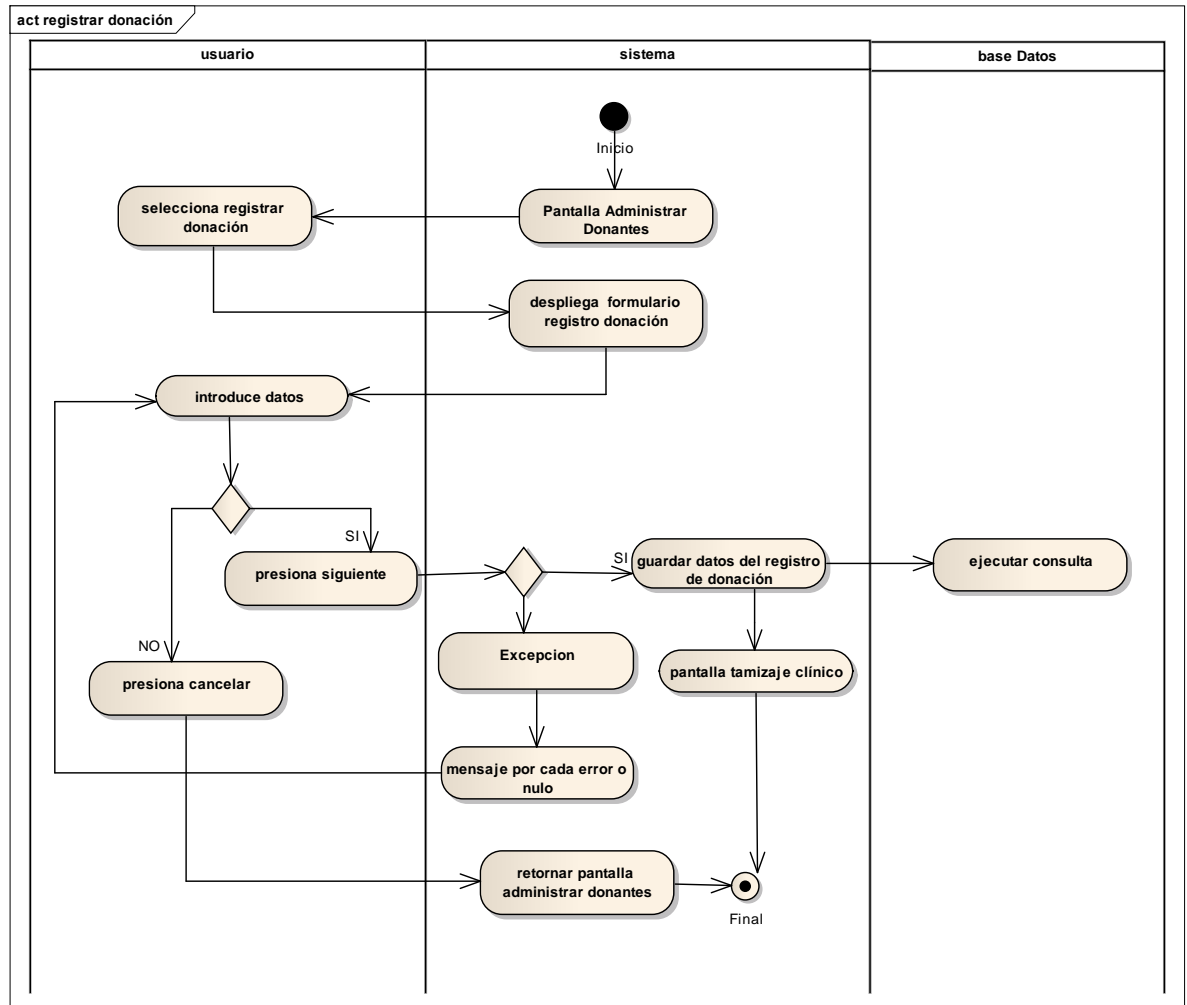


Fig. 33 Diagrama de Actividades Caso de uso Registrar Donación

Diagrama de Actividades Caso de uso Registrar Tamizaje Clínico

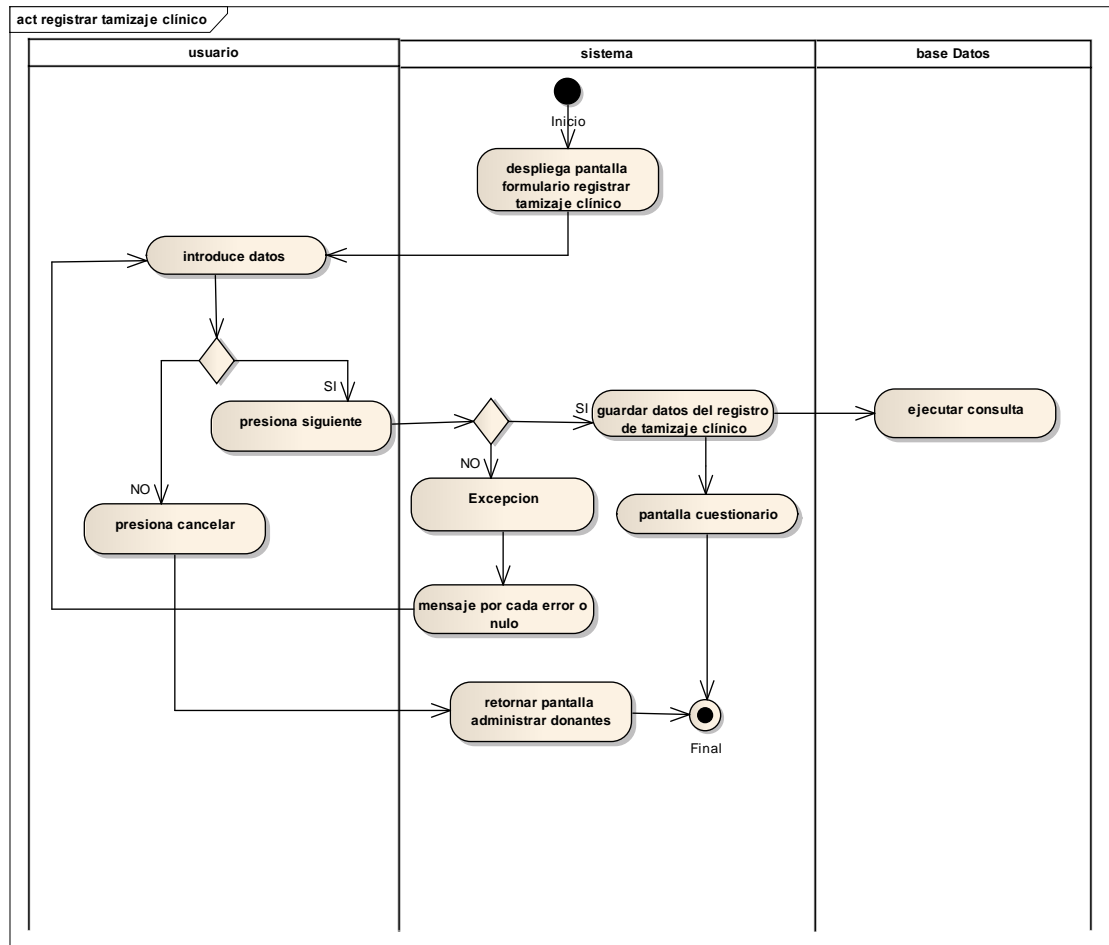


Fig. 34 Diagrama de Actividades Caso de uso Registrar Tamizaje Clínico

Diagrama de Actividades Caso de uso Registrar Cuestionario

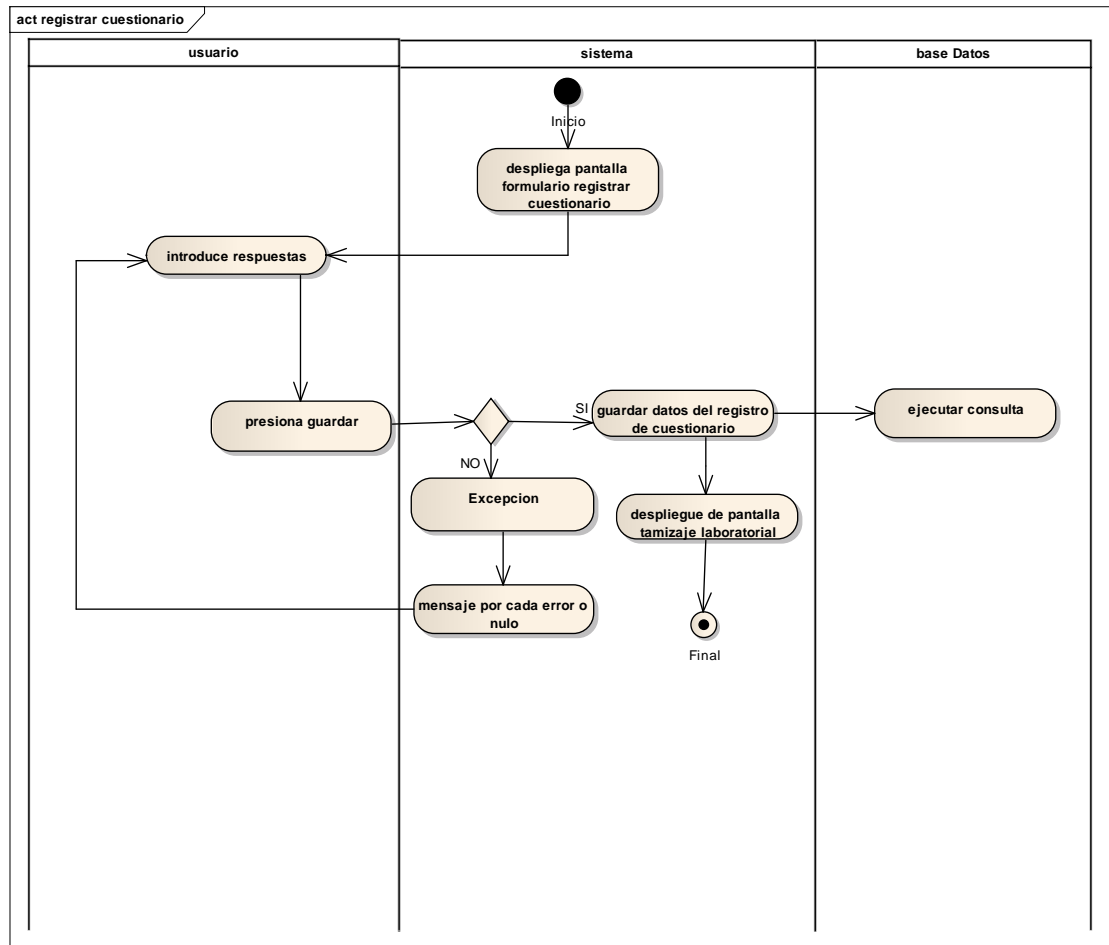


Fig. 35 Diagrama de Actividades Caso de uso Registrar Cuestionario

Diagrama de Actividades Caso de uso Registrar Tamizaje Laboratorial

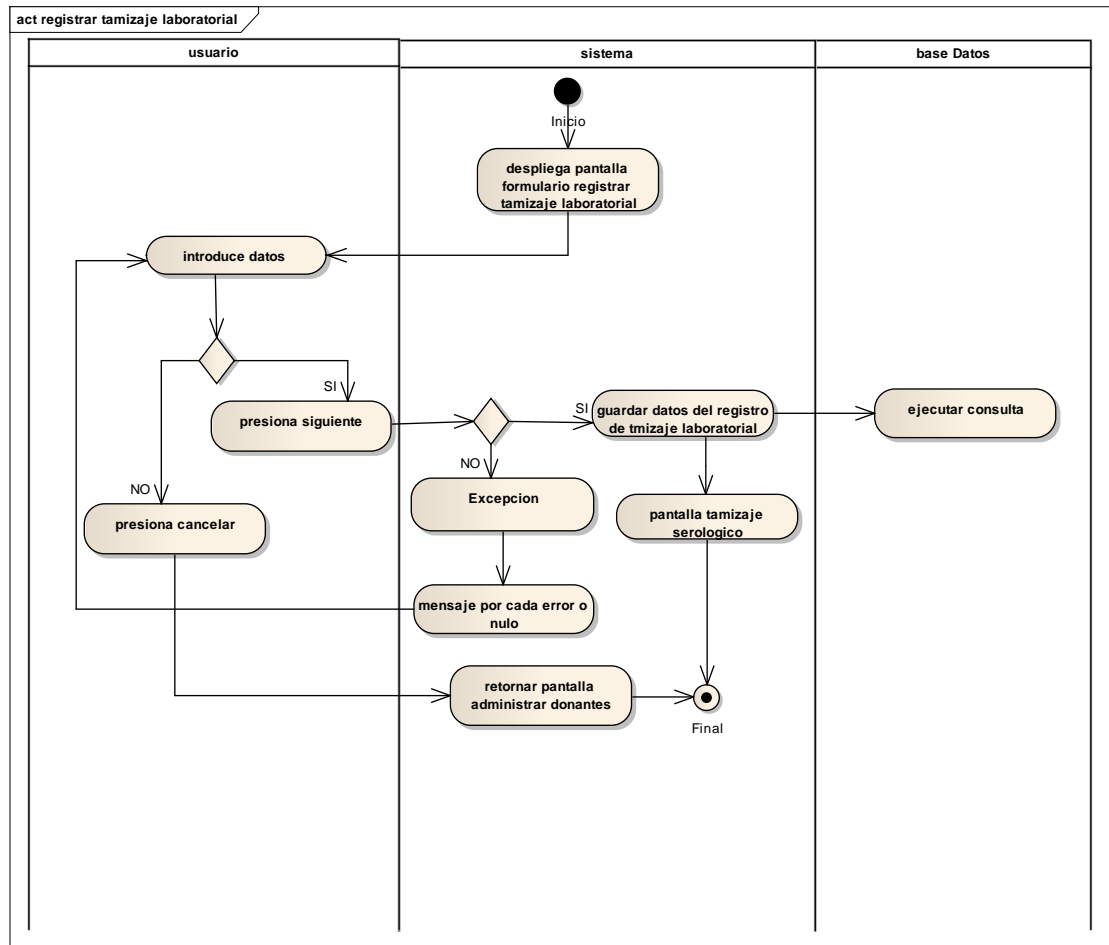


Fig. 36 Diagrama de Actividades Caso de uso Registrar Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Registrar Tamizaje Serológico

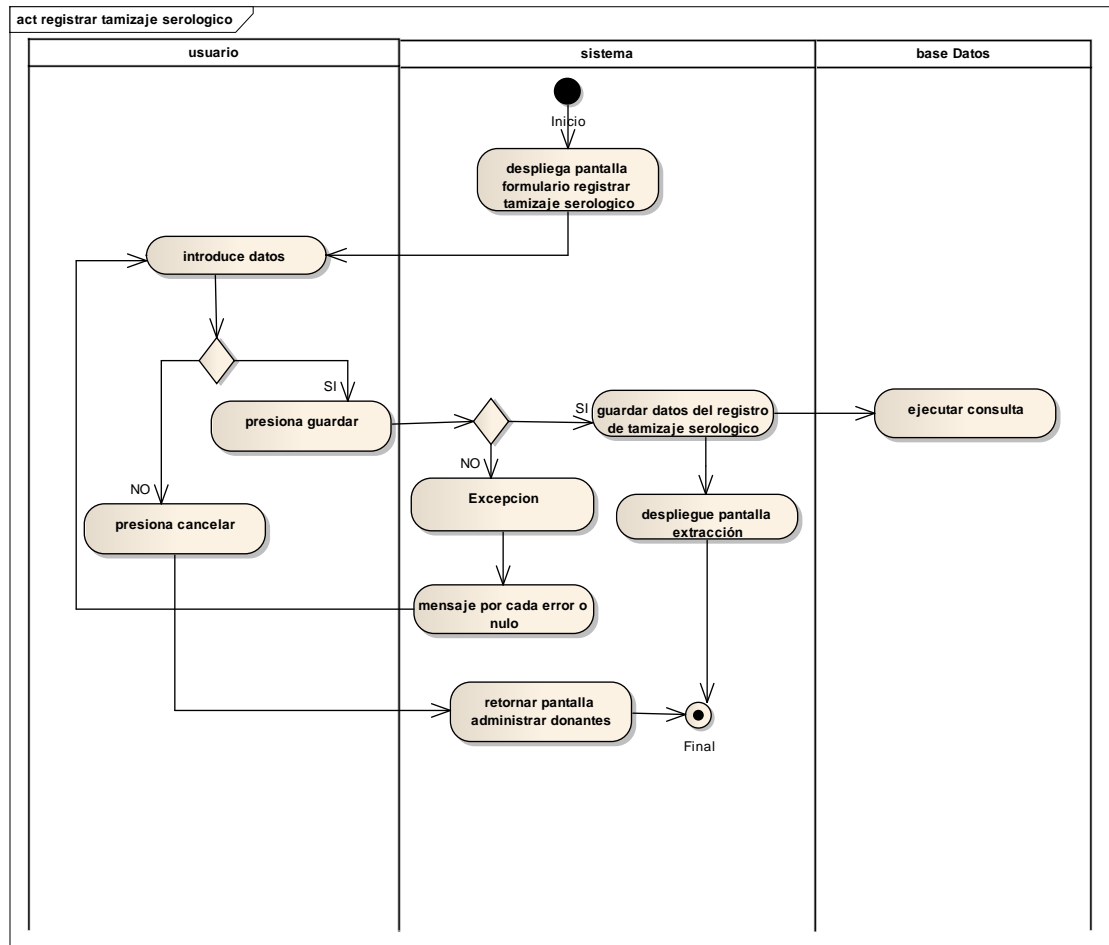


Fig. 37 Diagrama de Actividades Caso de uso Registrar Tamizaje Serológico

Diagrama de Actividades Caso de uso Registrar Extracción

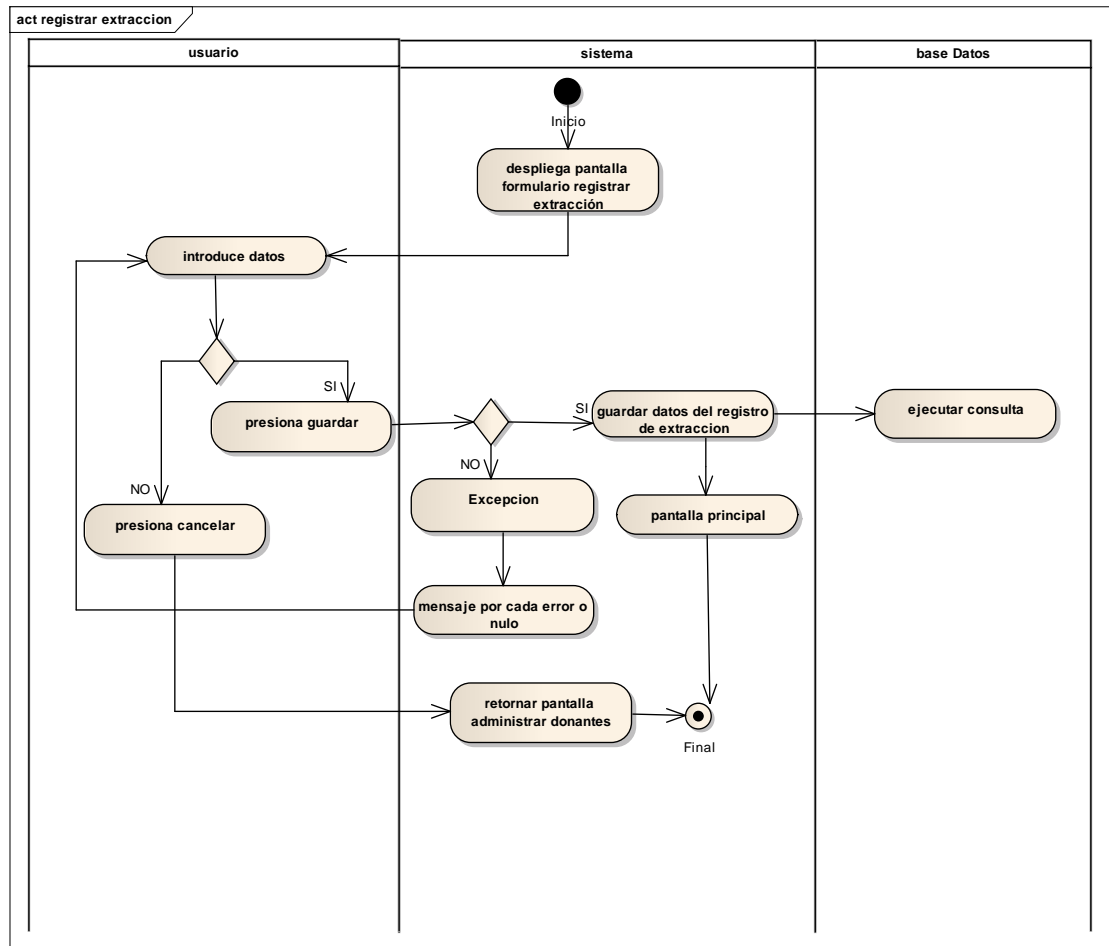


Fig. 38 Diagrama de Actividades Caso de uso Registrar Extracción

Diagrama de Actividades Caso de uso Registrar Rechazo

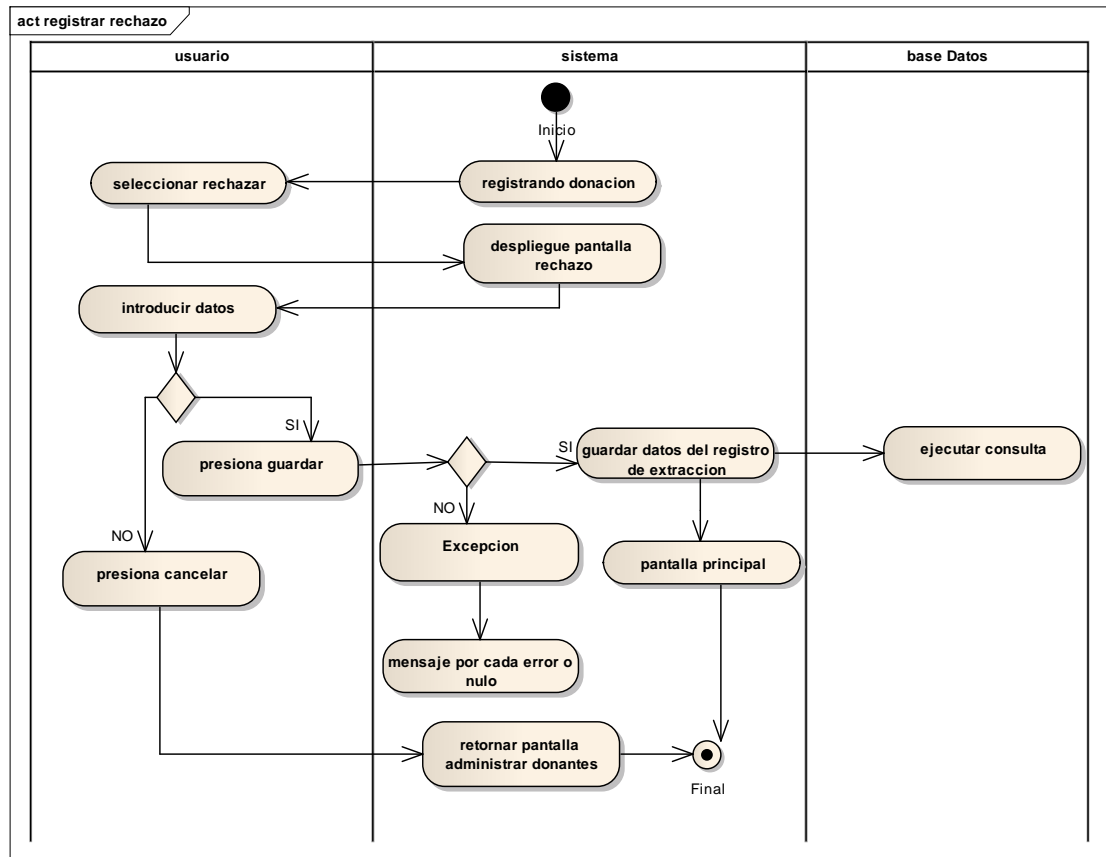


Fig. 39 Diagrama de Actividades Caso de uso Registrar Rechazo

Diagrama de Actividades Caso de uso Administrar Tamizaje Clínico

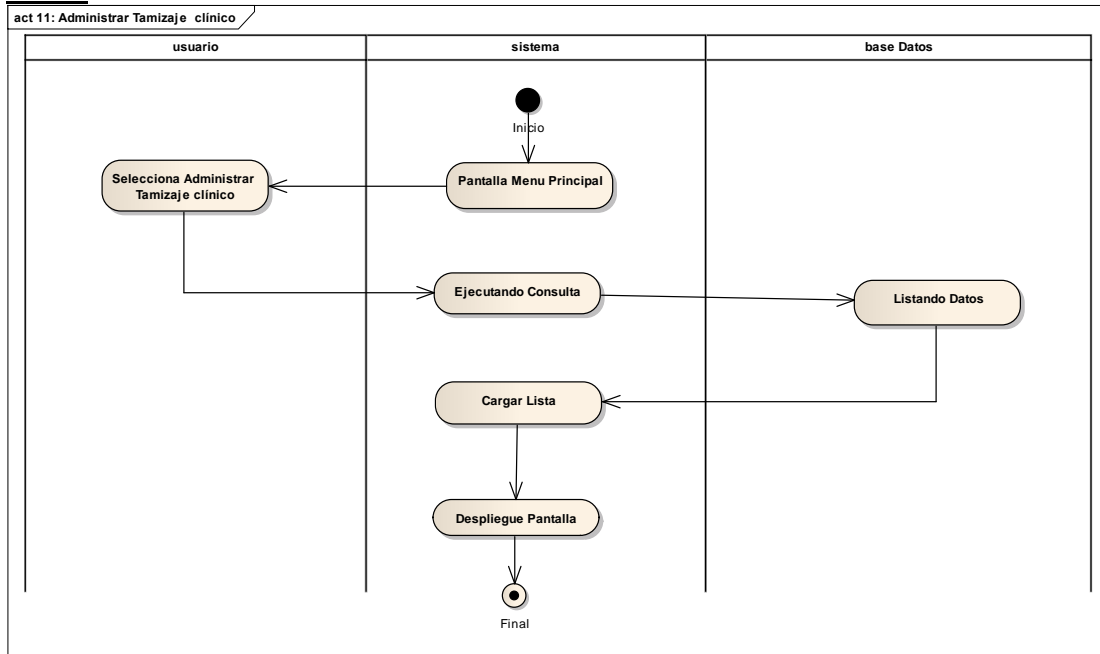


Fig. 40 Diagrama de Actividades Caso de uso Administrar Tamizaje Clínico

Diagrama de Actividades Caso de uso Modificar Tamizaje Clínico

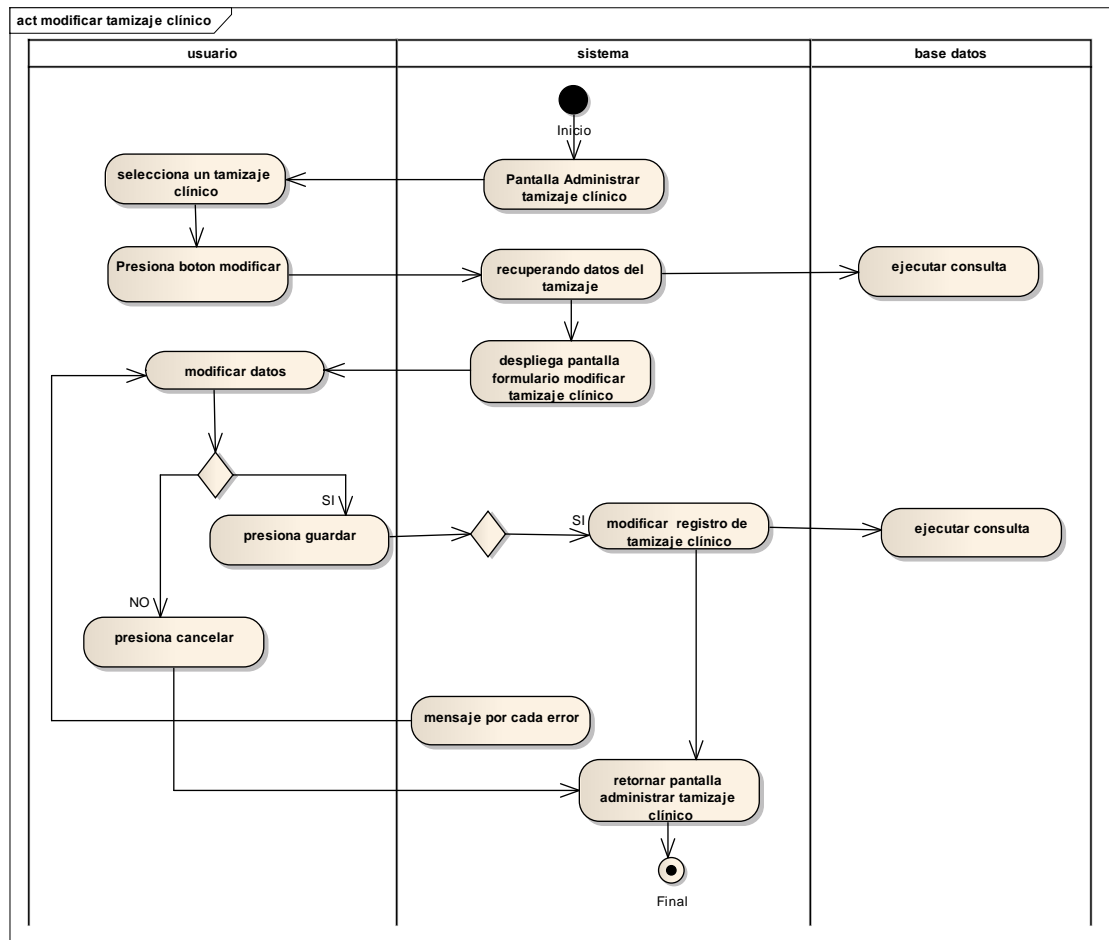


Fig. 41 Diagrama de Actividades Caso de uso Modificar Tamizaje Clínico

Diagrama de Actividades Caso de uso Ver Tamizaje Clínico

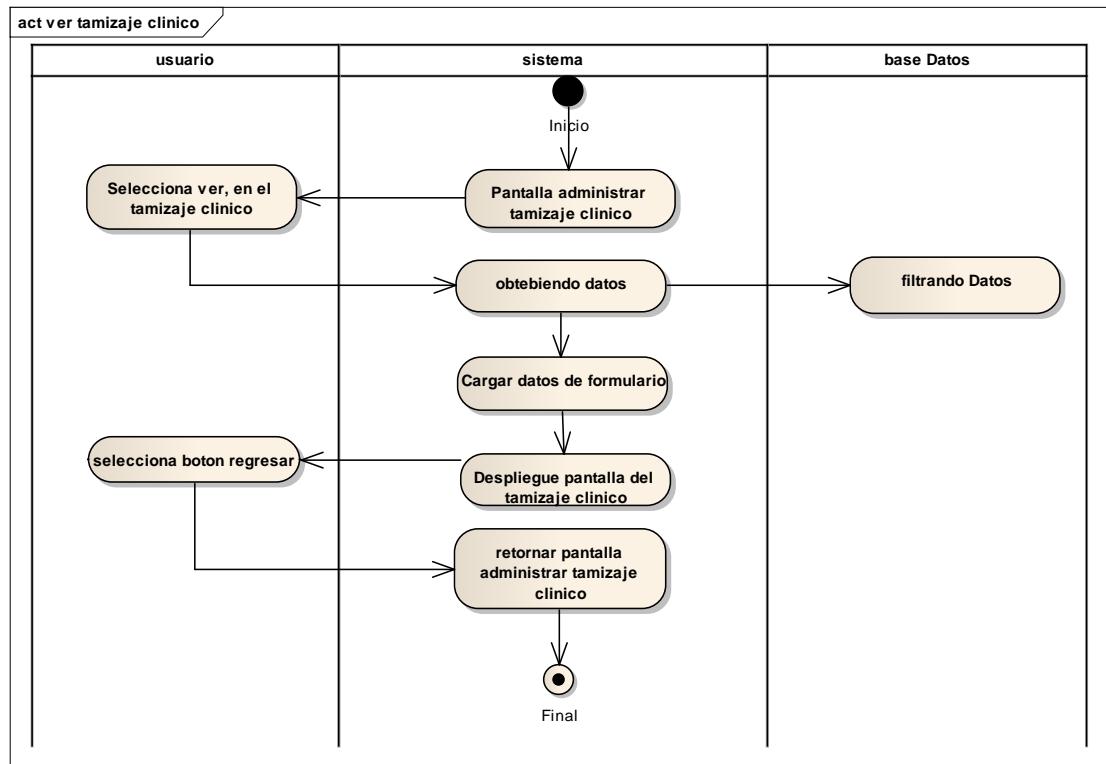


Fig. 42 Diagrama de Actividades Caso de uso Ver Tamizaje Clínico

Diagrama de Actividades Caso de uso Ver Cuestionario

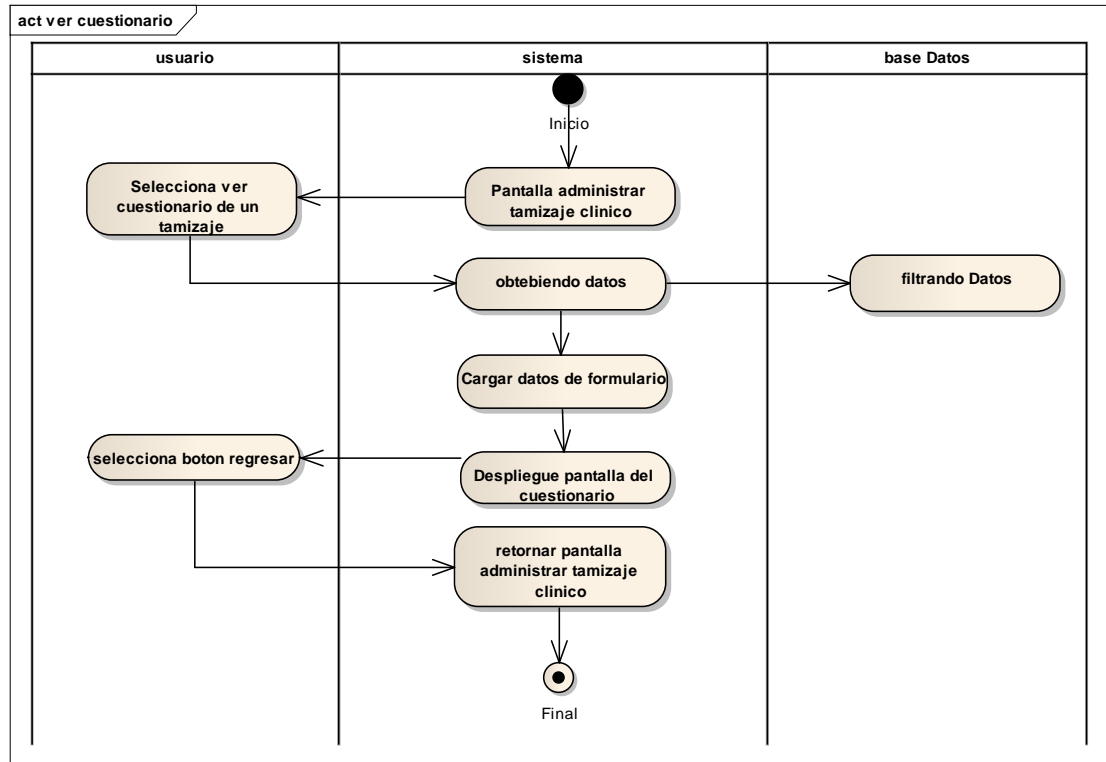


Fig. 43 Diagrama de Actividades Caso de uso Ver Cuestionario

Diagrama de Actividades Caso de uso Buscar Tamizaje Clínico

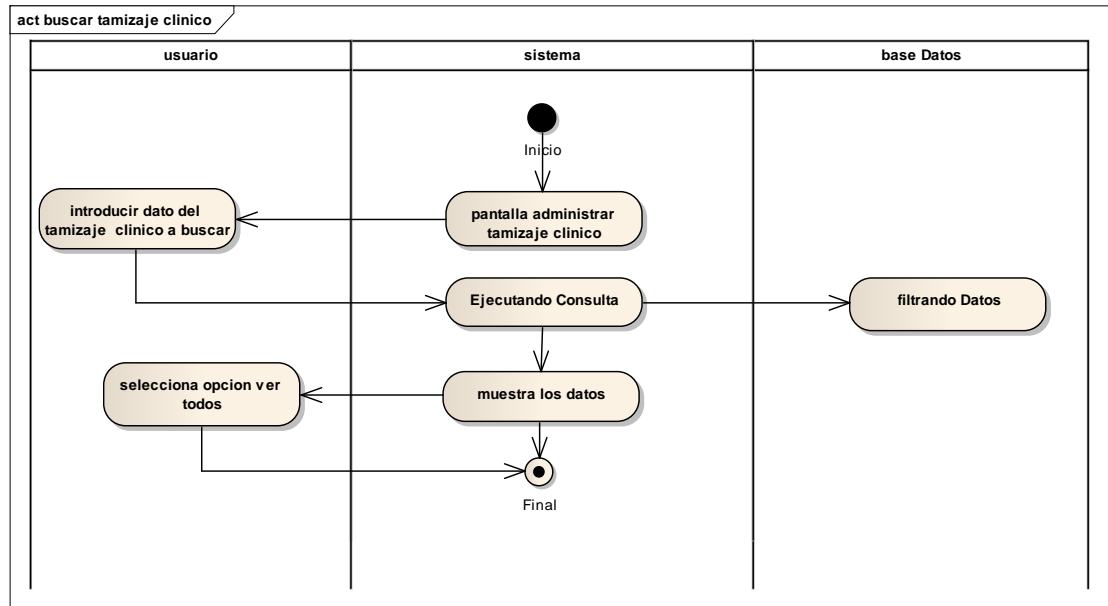


Fig. 44 Diagrama de Actividades Caso de uso Buscar Tamizaje Clínico

Diagrama de Actividades Caso de uso Administrar Tamizaje Laboratorial

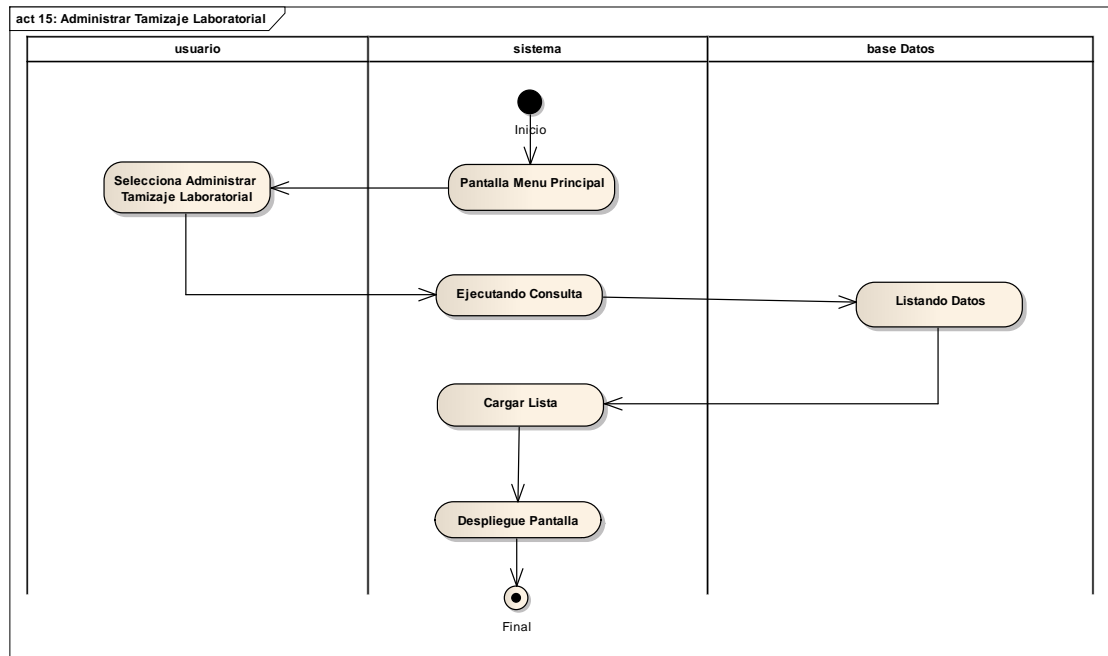


Fig. 45 Diagrama de Actividades Caso de uso Administrar Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Modificar Tamizaje Laboratorial

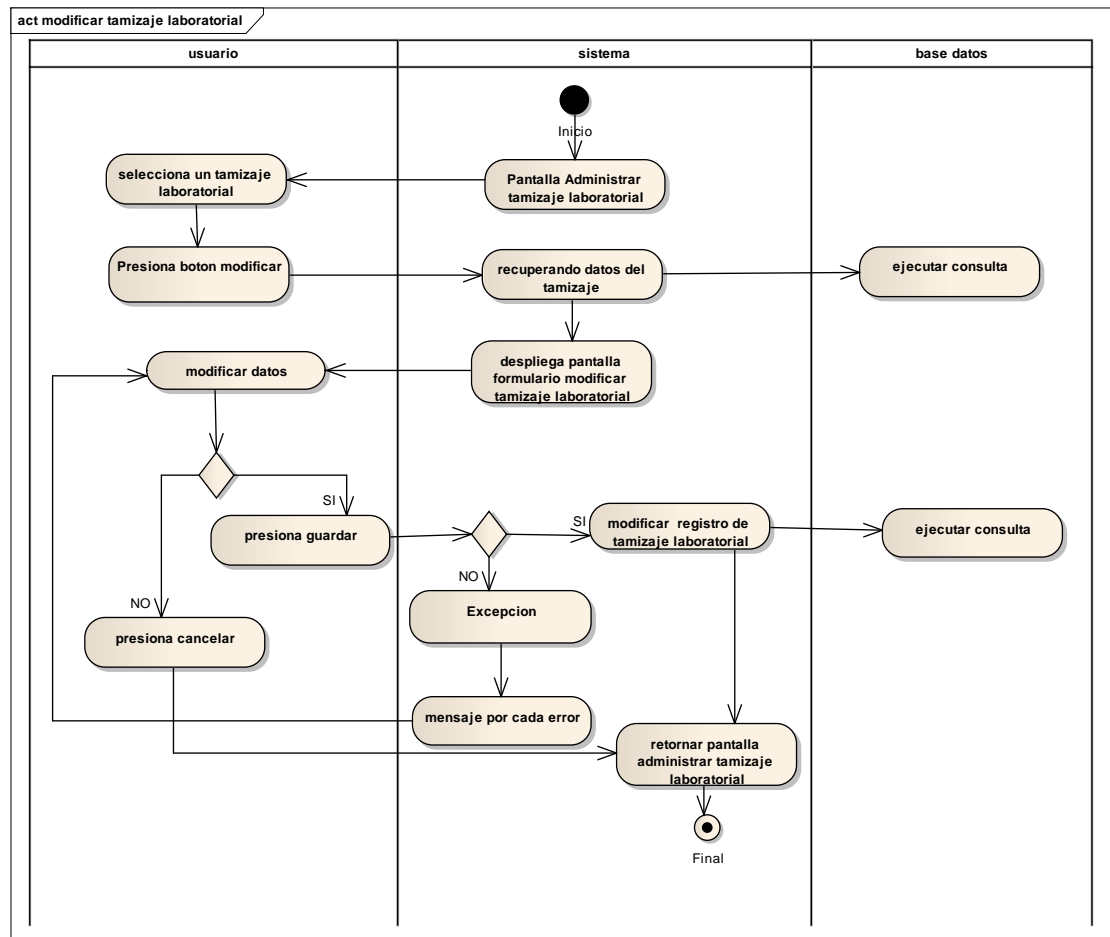


Fig. 46 Diagrama de Actividades Caso de uso Modificar Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Ver Tamizaje Laboratorial

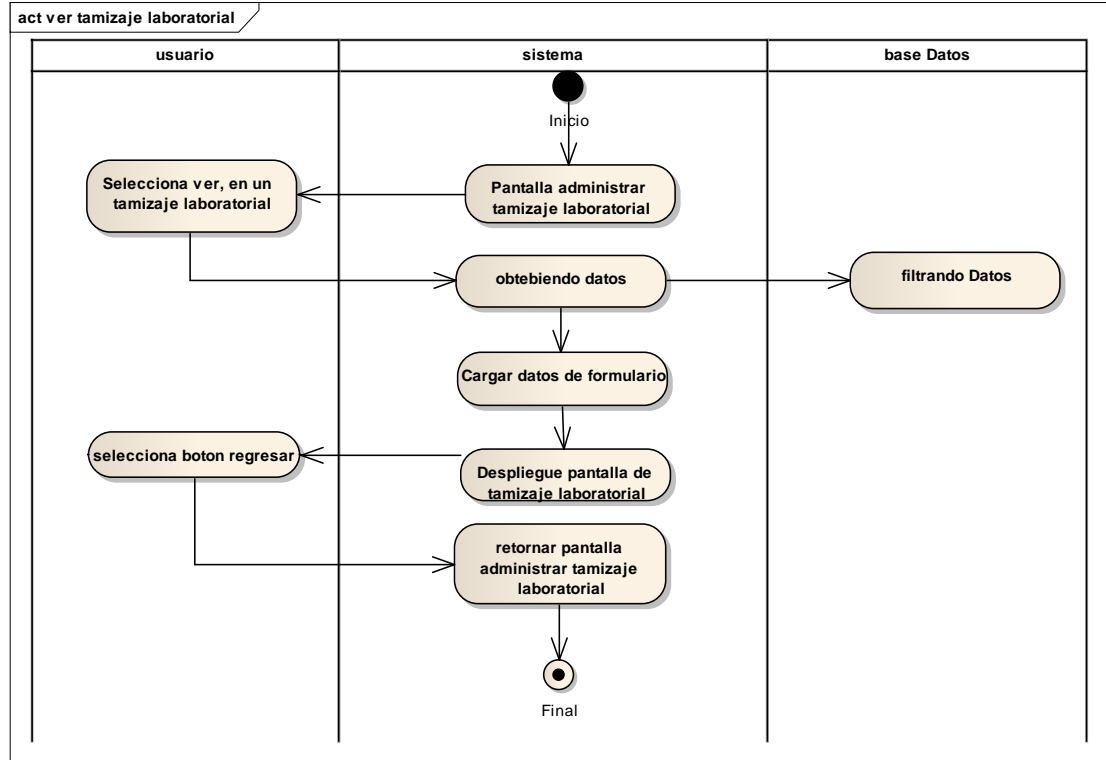


Fig. 47 Diagrama de Actividades Caso de uso Ver Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Buscar Tamizaje Laboratorial

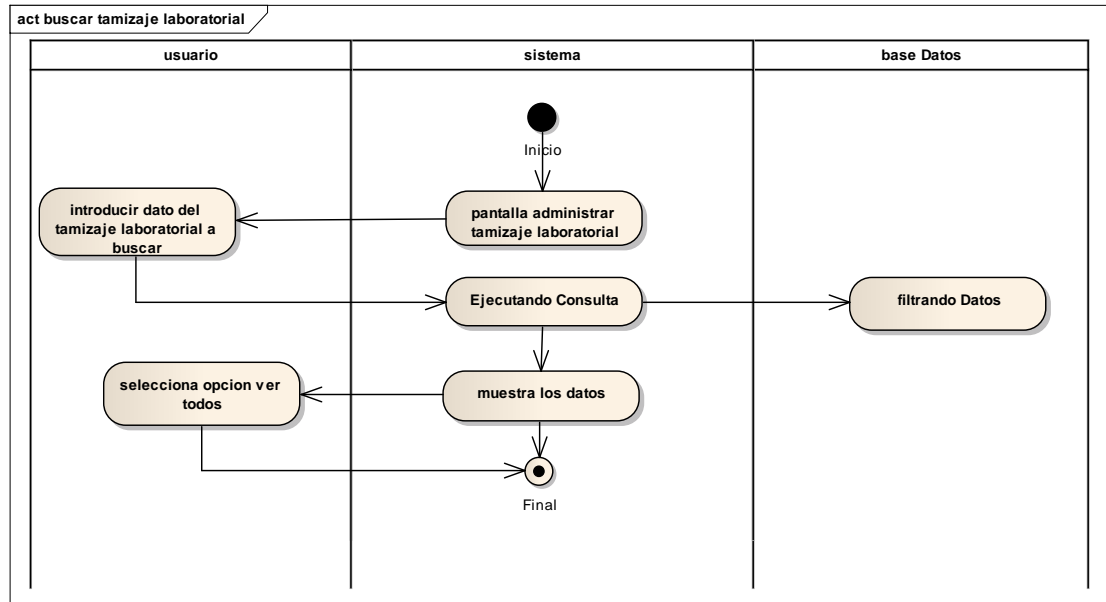


Fig. 48 Diagrama de Actividades Caso de uso Buscar Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Administrar Extracción

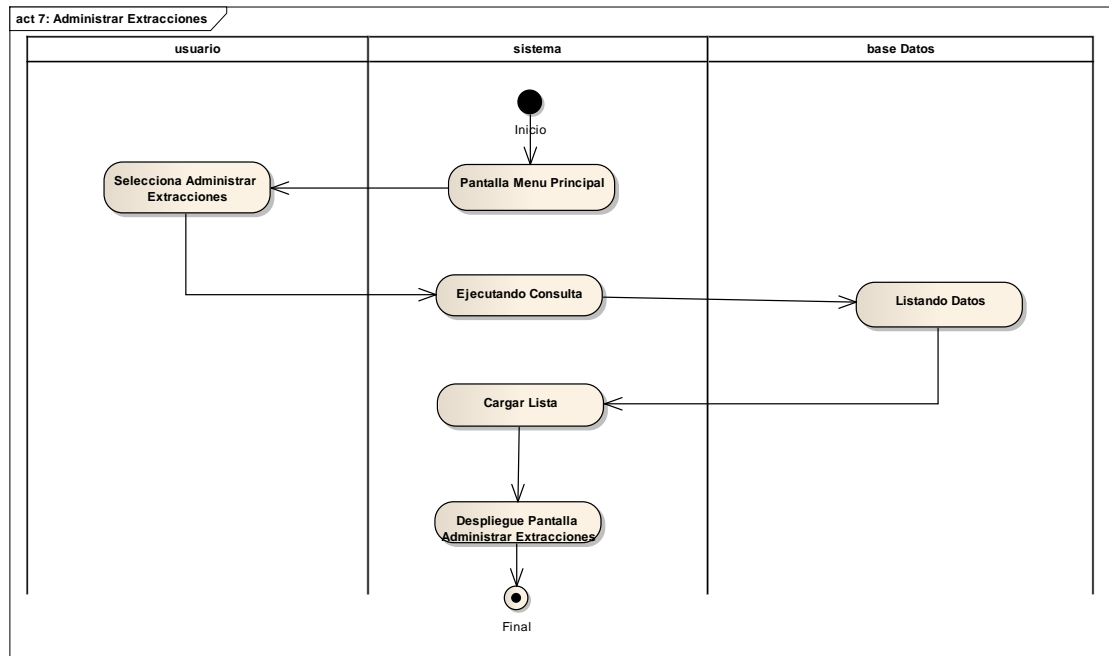


Fig. 49 Diagrama de Actividades Caso de uso Administrar Extracción

Diagrama de Actividades Caso de uso Modificar Extracción

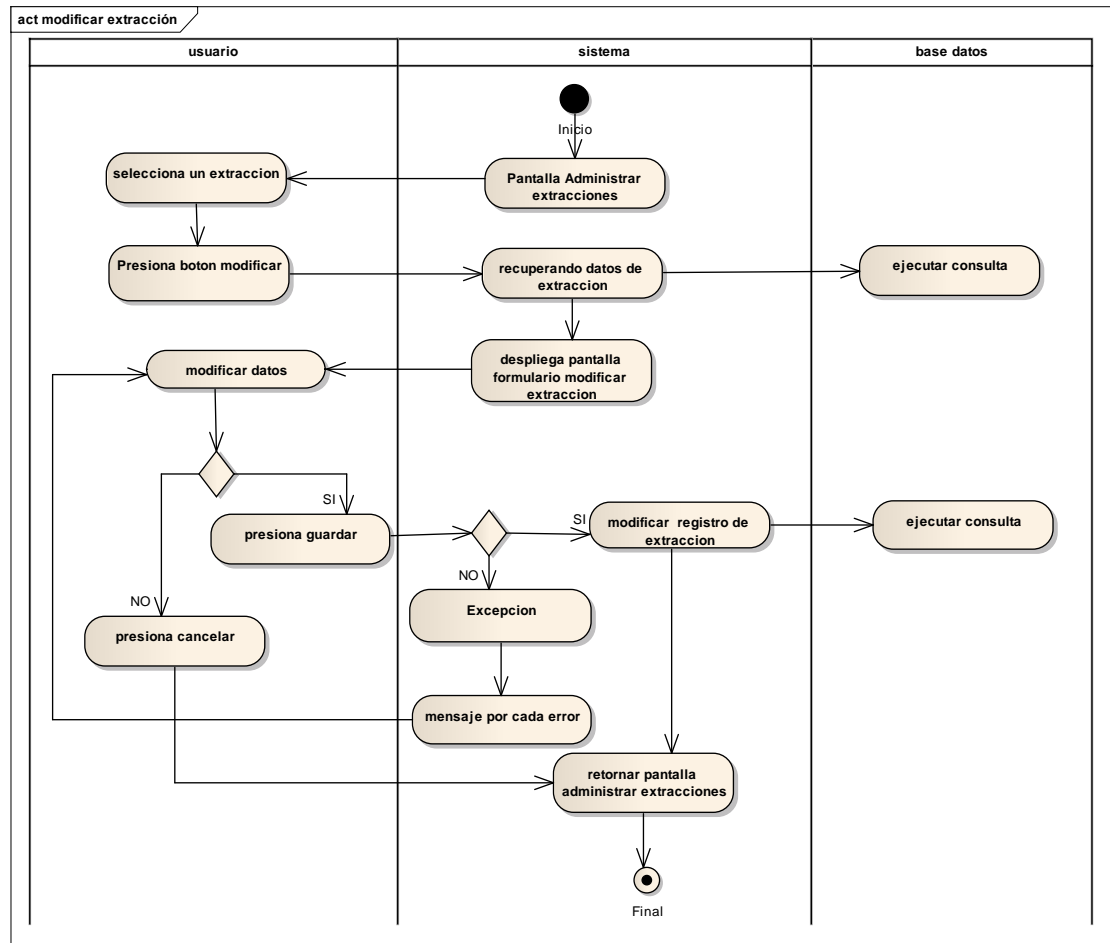


Fig. 50 Diagrama de Actividades Caso de uso Modificar Extracción

Diagrama de Actividades Caso de uso Ver Extracción

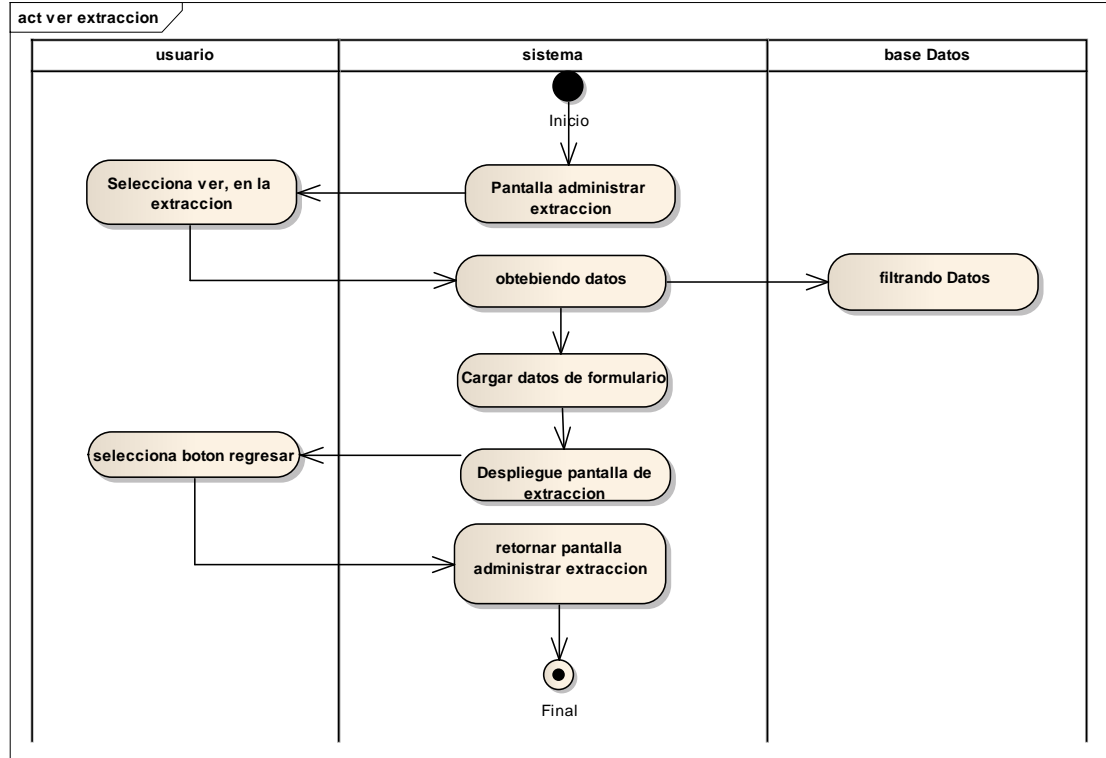


Fig. 51 Diagrama de Actividades Caso de uso Ver Extracción

Diagrama de Actividades Caso de uso Buscar Extracción

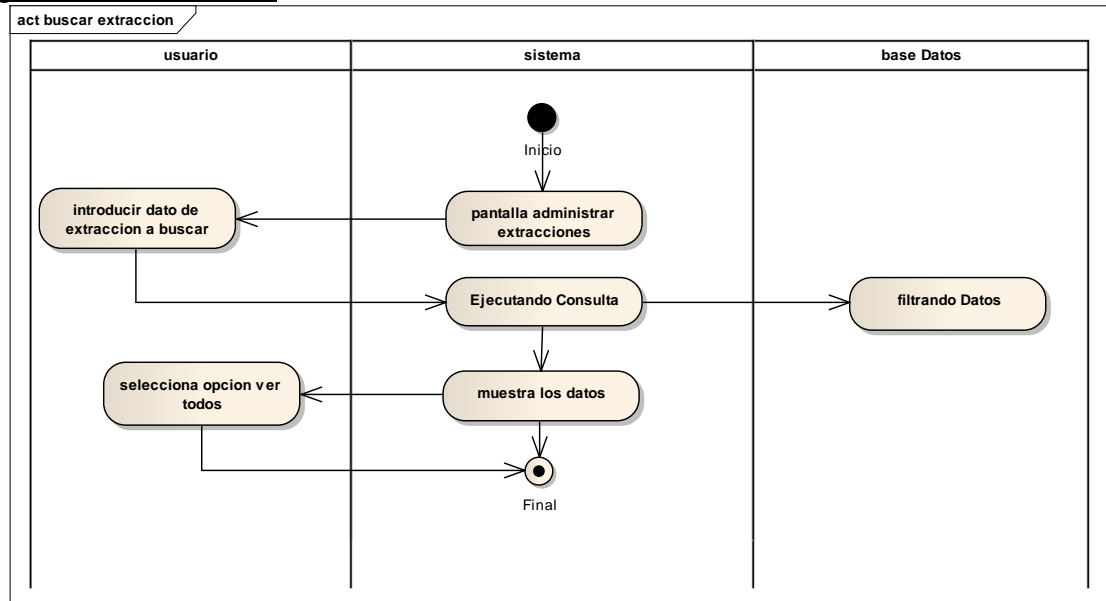


Fig. 52 Diagrama de Actividades Caso de uso Buscar Extracción

Diagrama de Actividades Caso de uso Administrar Donaciones

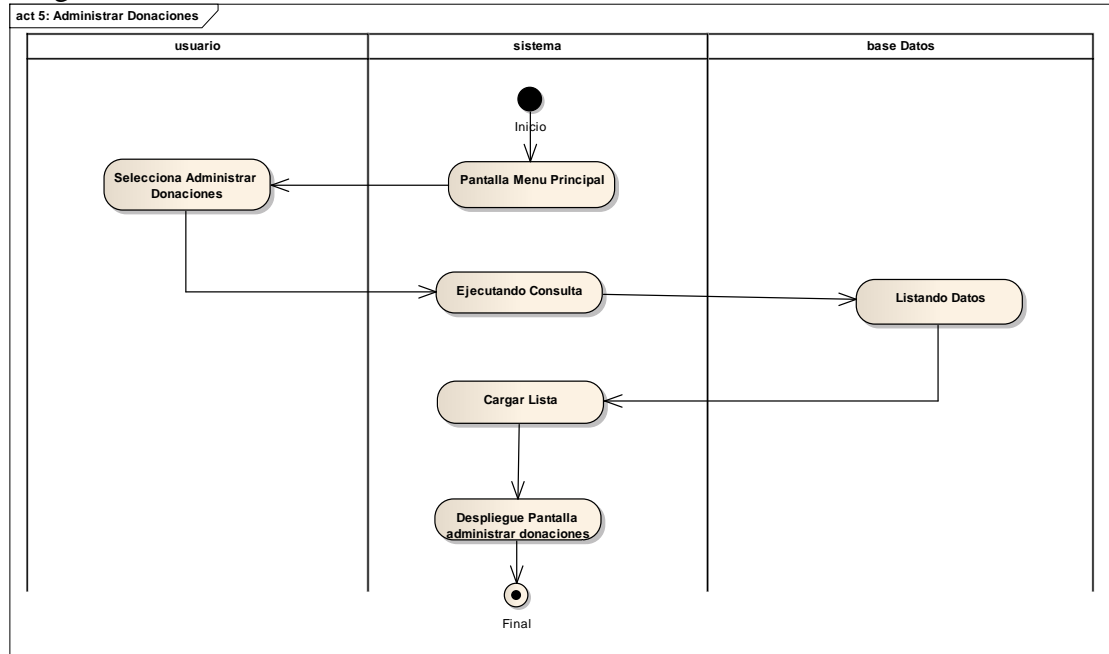


Fig. 53 Diagrama de Actividades Caso de uso Administrar Donaciones

Diagrama de Actividades Caso de uso Completar Donación

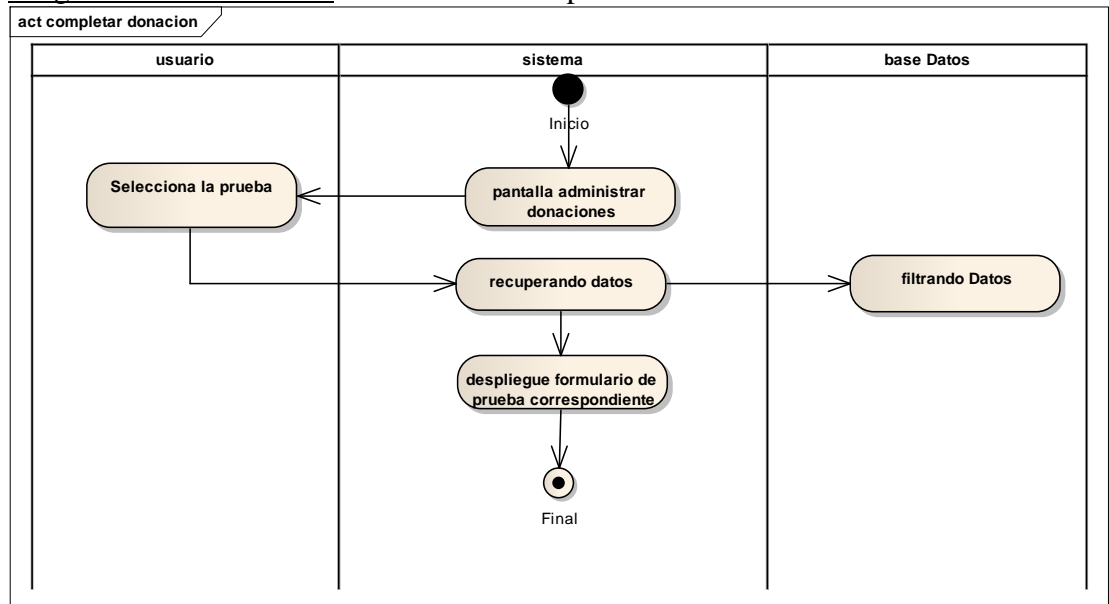


Fig. 54 Diagrama de Actividades Caso de uso Completar Donación

Diagrama de Actividades Caso de uso Ver Resultado de Donación

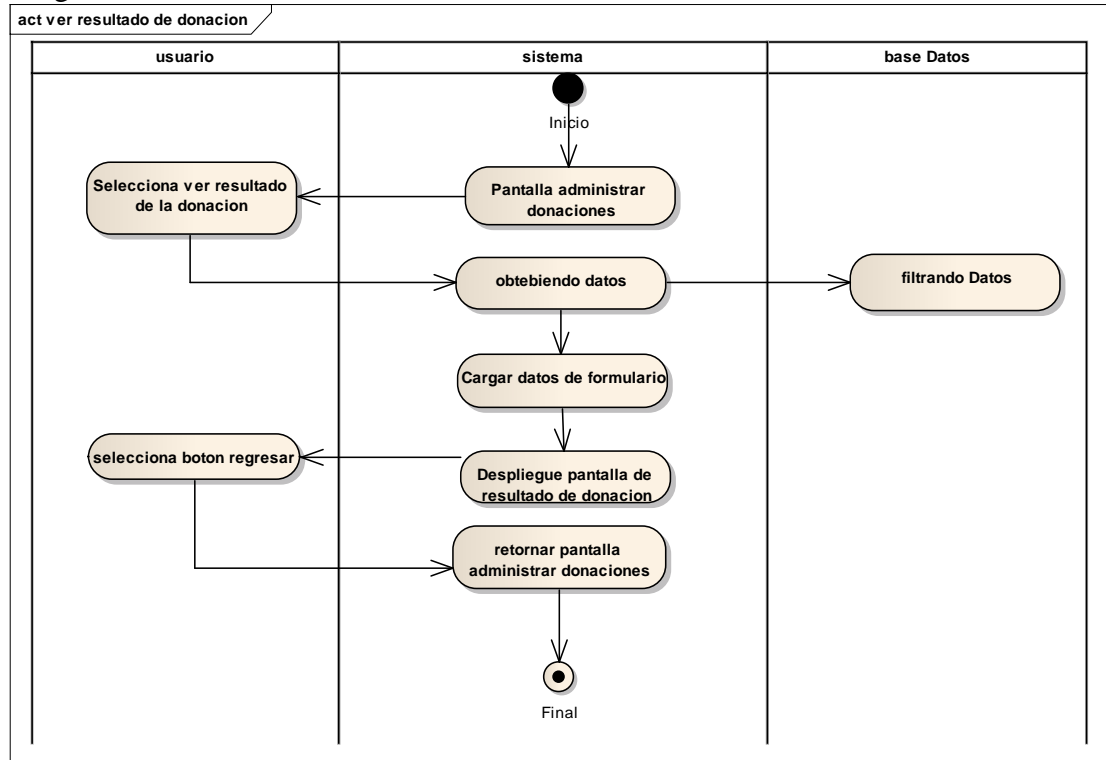


Fig. 55 Diagrama de Actividades Caso de uso Ver Resultado de Donación

Diagrama de Actividades Caso de uso Buscar Registro Donación

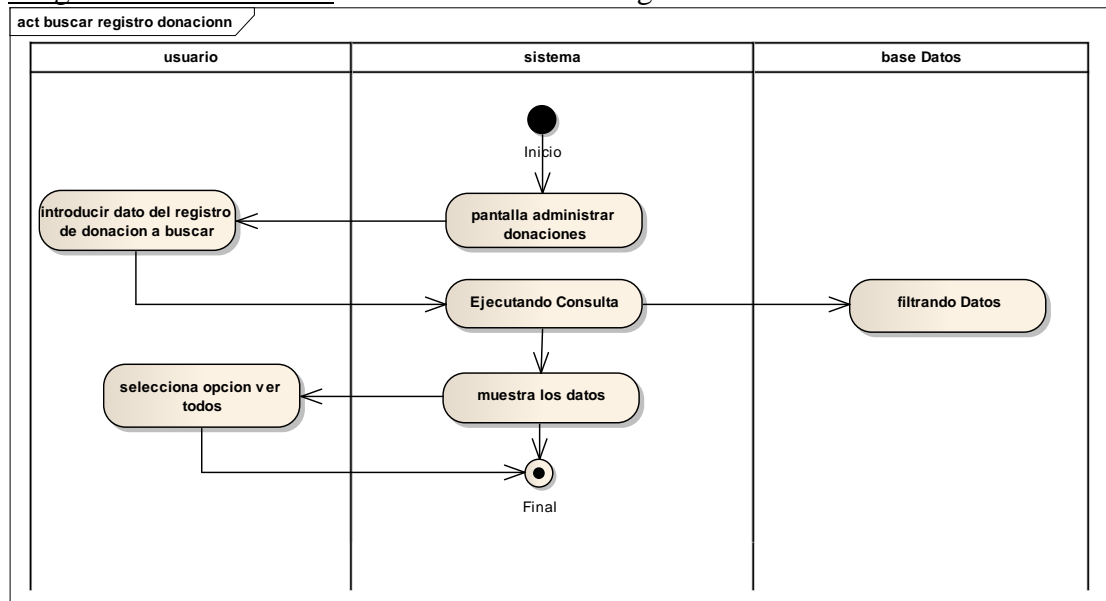


Fig. 56 Diagrama de Actividades Caso de uso Buscar Registro Donación

Diagrama de Actividades Caso de uso Administrar Rechazo

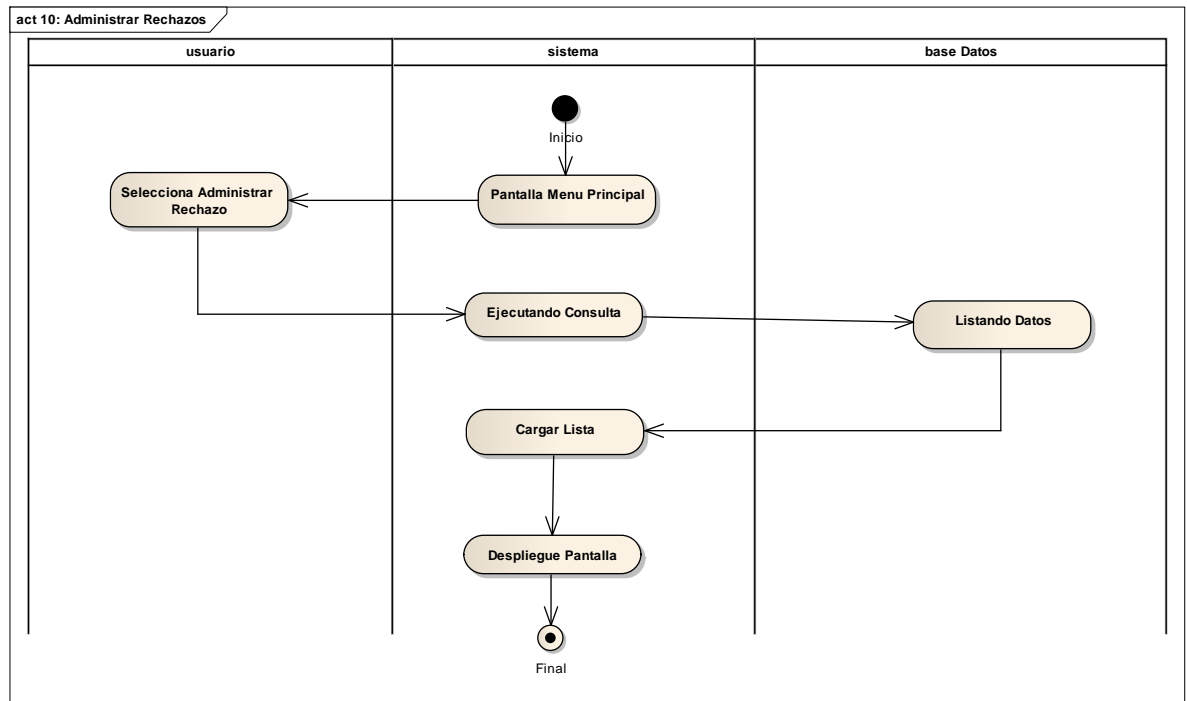


Fig. 57 Diagrama de Actividades Caso de uso Administrar Rechazo

Diagrama de Actividades Caso de uso Modificar Rechazo

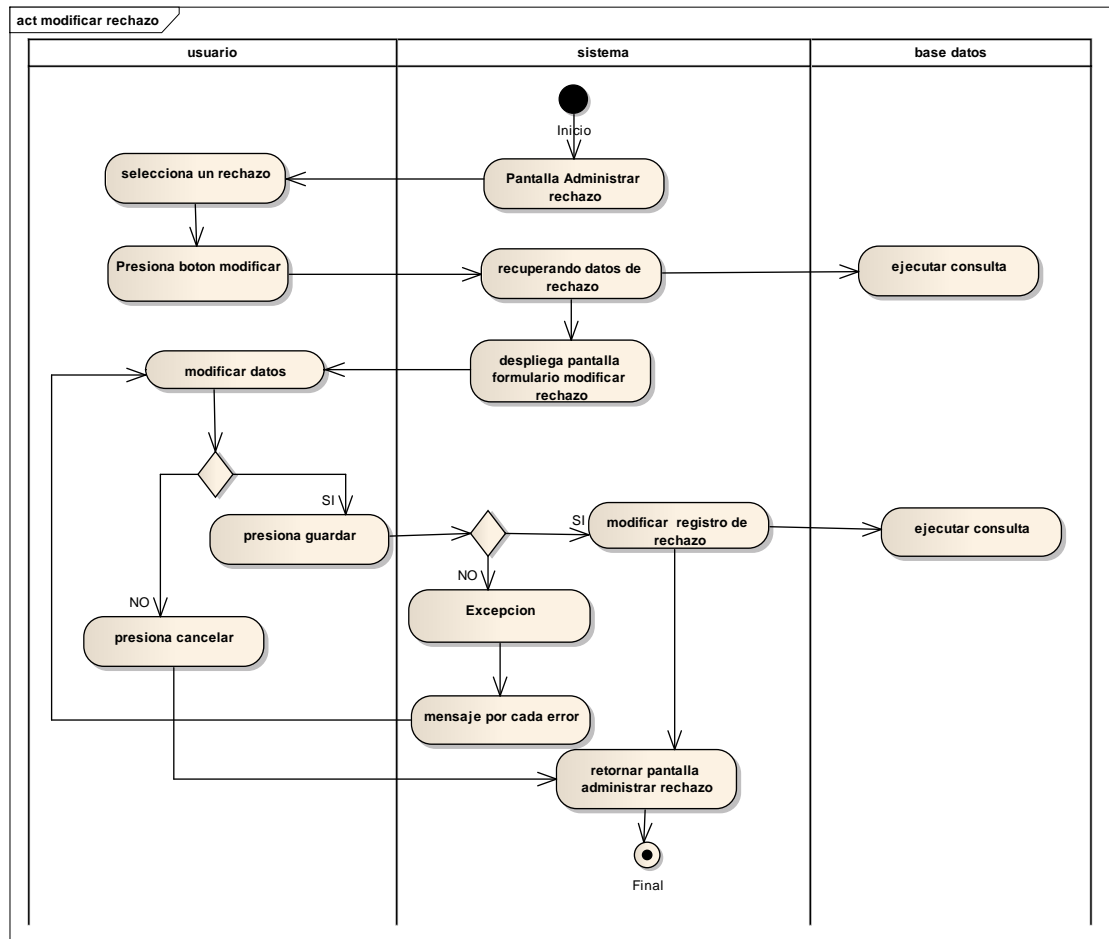


Fig. 58 Diagrama de Actividades Caso de uso Modificar Rechazo

Diagrama de Actividades Caso de uso Ver Rechazo

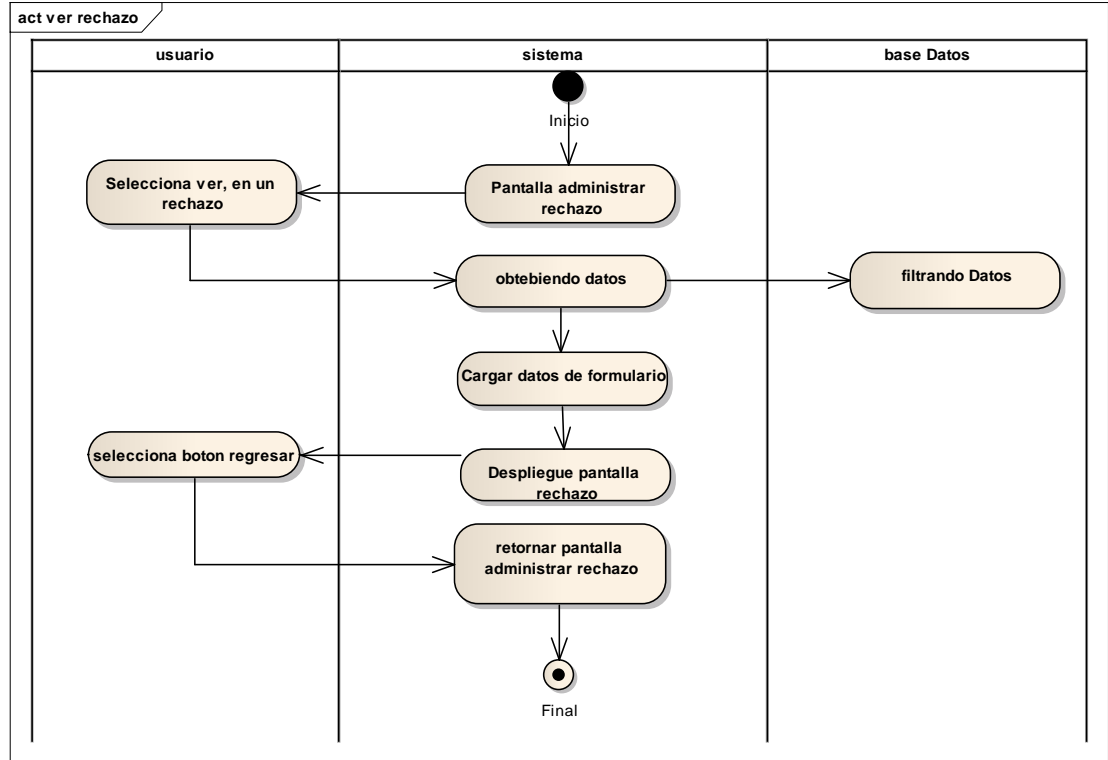


Fig. 59 Diagrama de Actividades Caso de uso Ver Rechazo

Diagrama de Actividades Caso de uso Buscar Rechazo

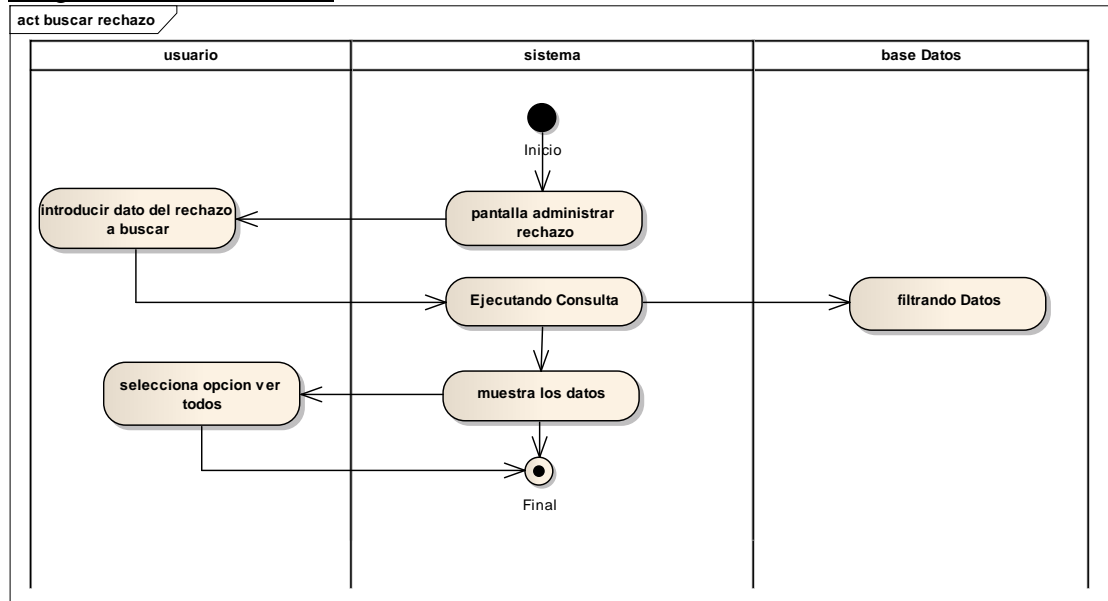


Fig. 60 Diagrama de Actividades Caso de uso Buscar Rechazo

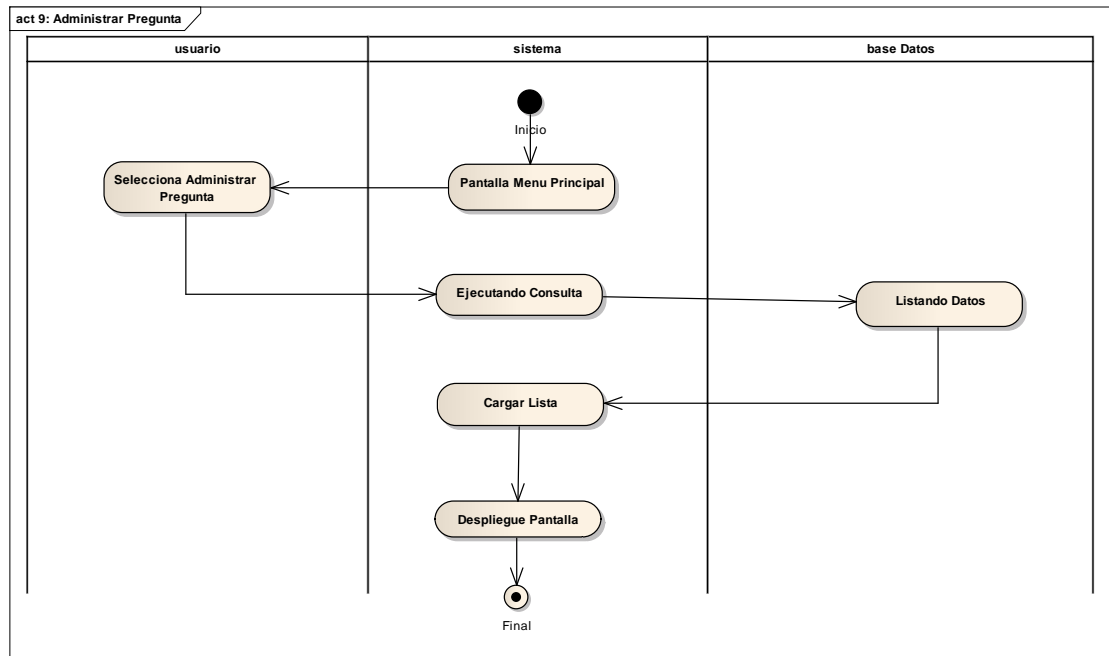
Diagrama de Actividades Caso de uso Administrar Pregunta.**Fig. 61 Diagrama de Actividades Caso de uso Administrar Pregunta.**

Diagrama de Actividades Caso de uso Adicionar Pregunta

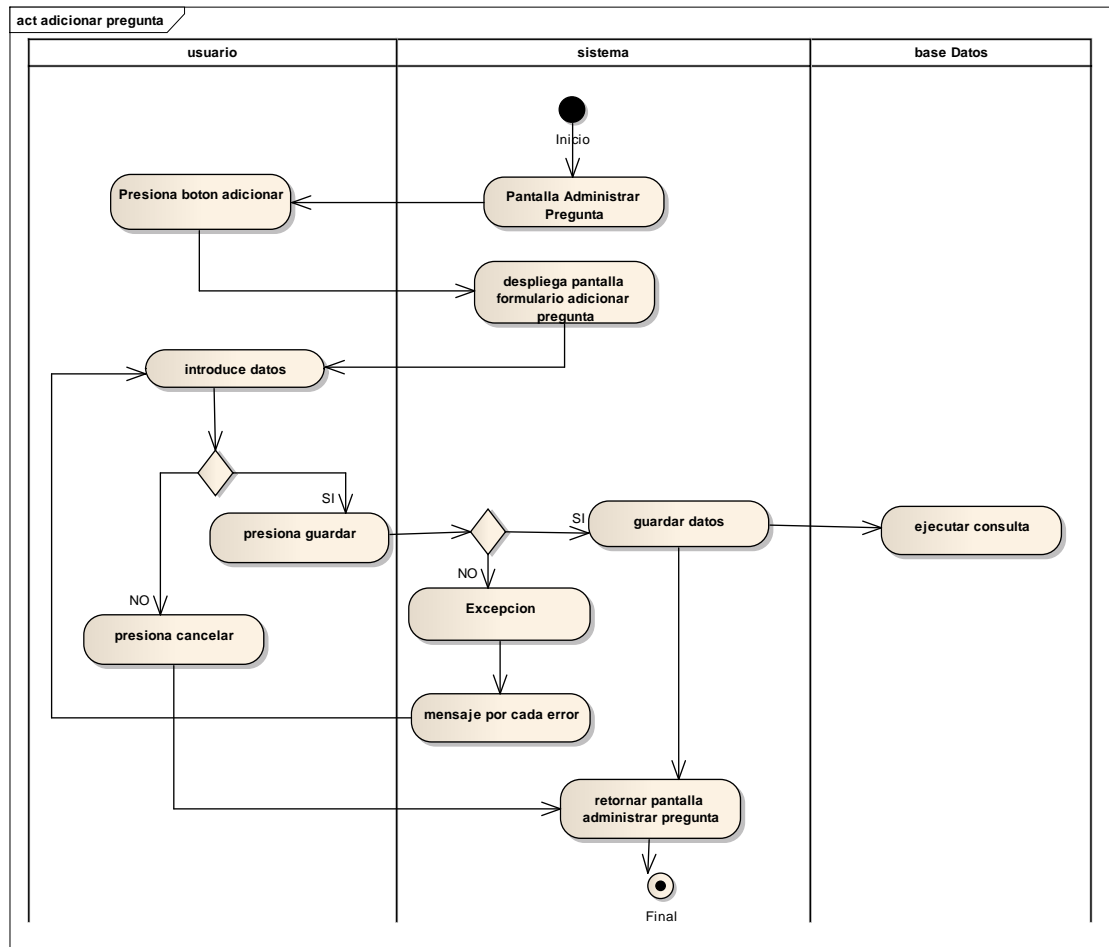


Fig. 62 Diagrama de Actividades Caso de uso Adicionar Pregunta

Diagrama de Actividades Caso de uso Modificar Pregunta

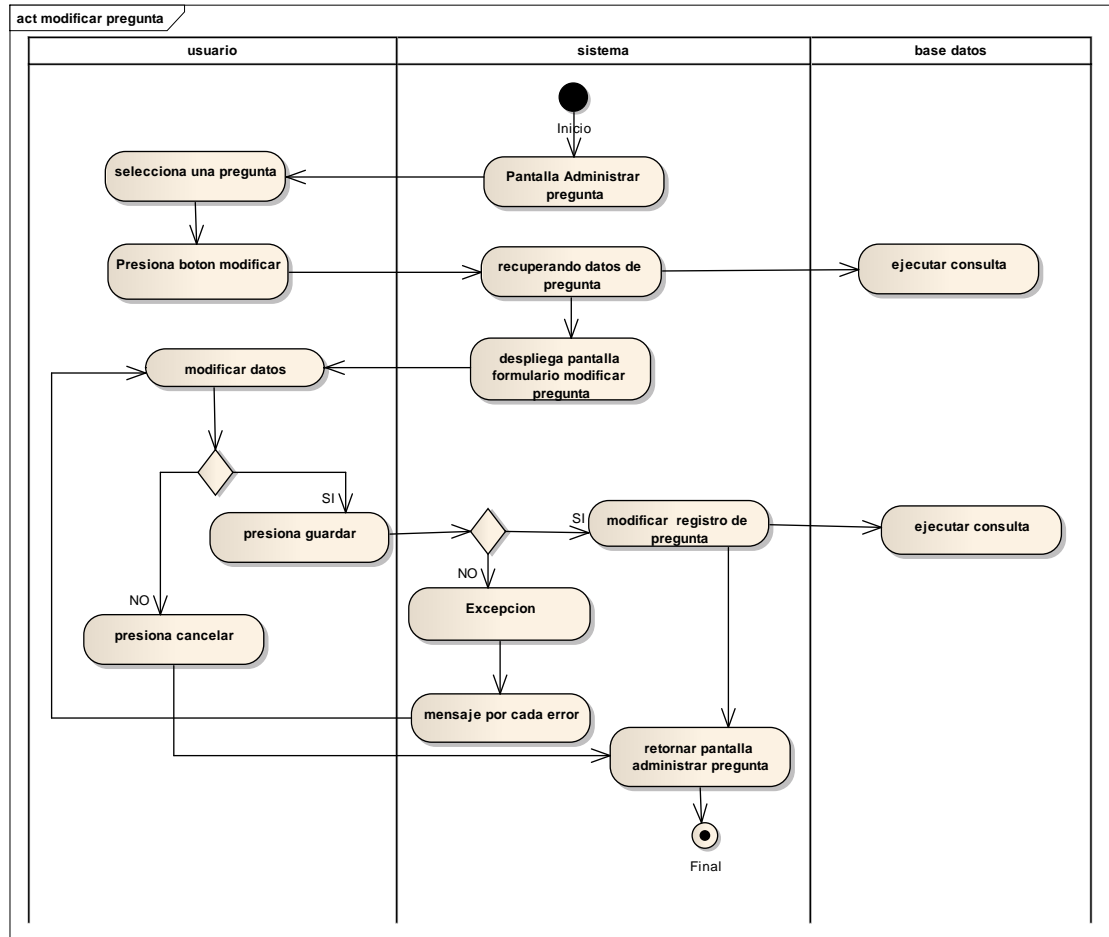


Fig. 63 Diagrama de Actividades Caso de uso Modificar Pregunta

Diagrama de Actividades Caso de uso Eliminar Pregunta

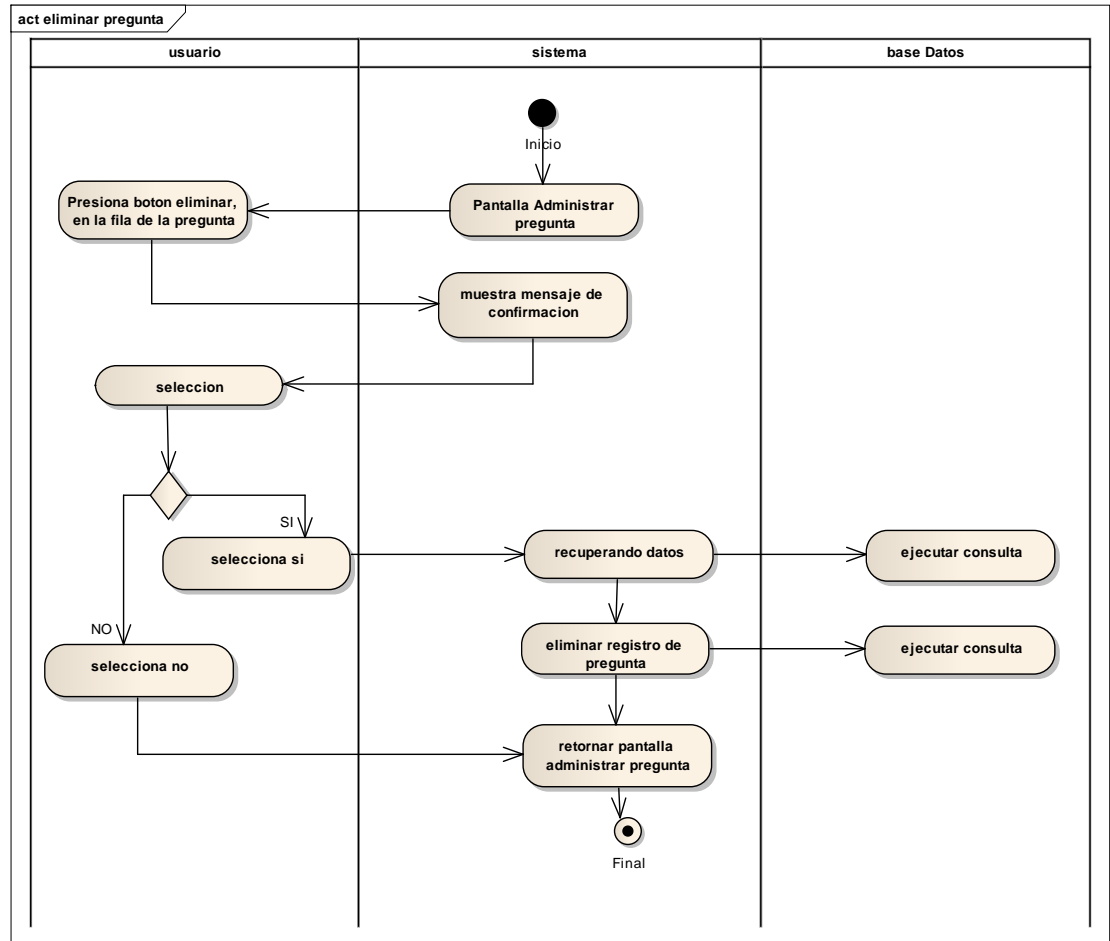


Fig. 64 Diagrama de Actividades Caso de uso Eliminar Pregunta

Diagrama de Actividades Caso de uso Administrar Tipo Bolsa

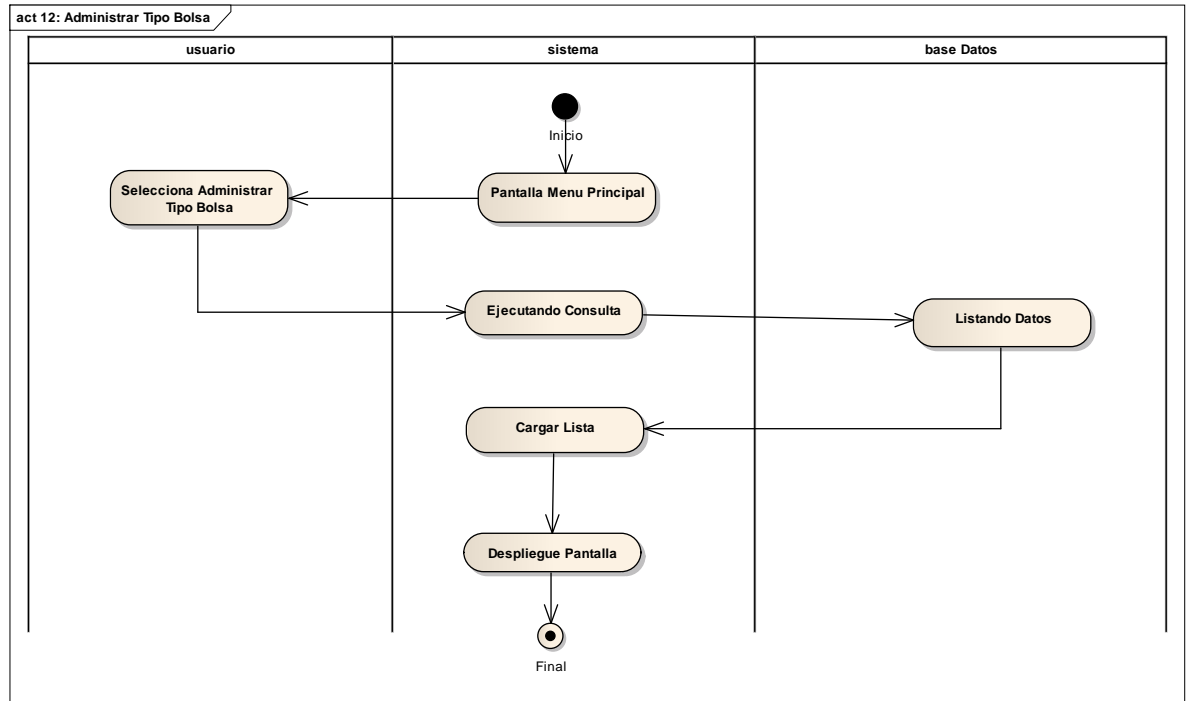


Fig. 65 Diagrama de Actividades Caso de uso Administrar Tipo Bolsa

Diagrama de Actividades Caso de uso Adicionar Tipo Bolsa

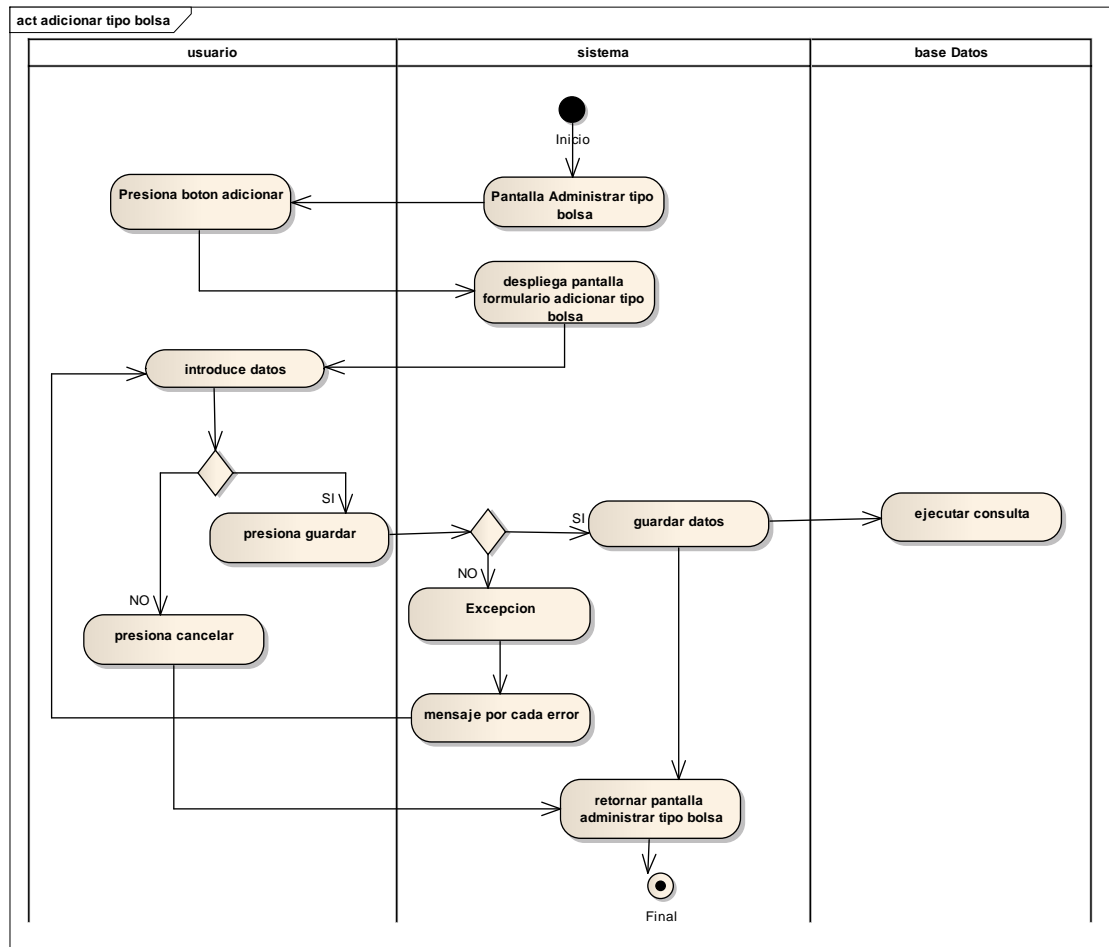


Fig. 66 Diagrama de Actividades Caso de uso Adicionar Tipo Bolsa

Diagrama de Actividades Caso de uso Modificar Tipo Bolsa

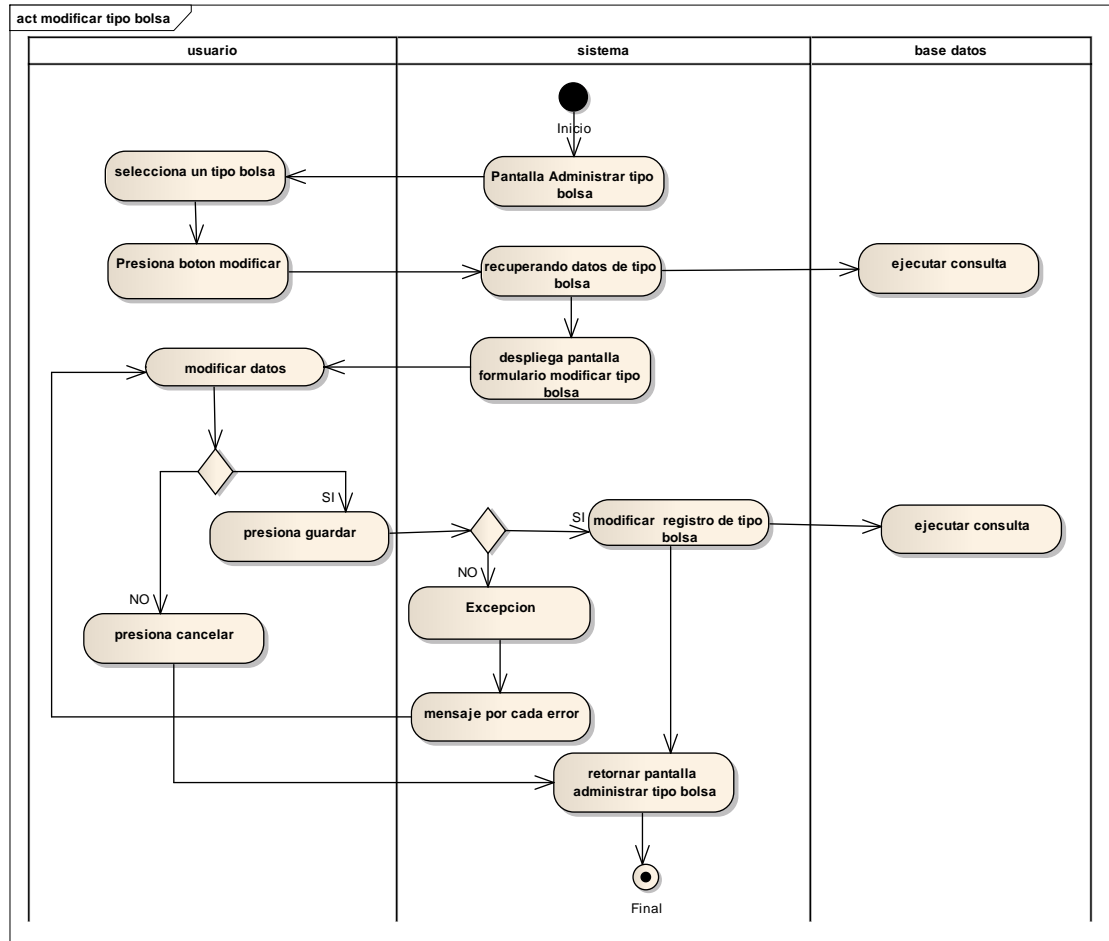


Fig. 67 Diagrama de Actividades Caso de uso Modificar Tipo Bolsa

Diagrama de Actividades Caso de uso Eliminar Tipo Bolsa

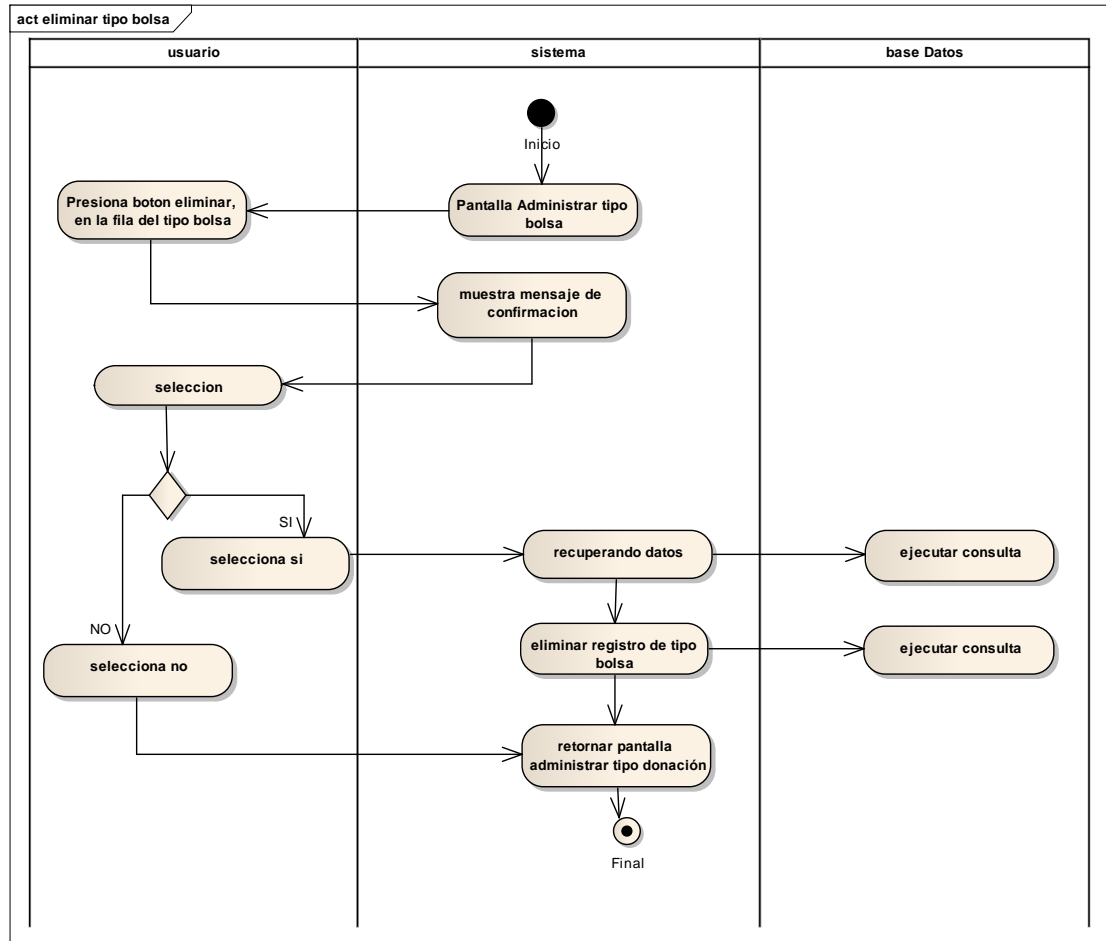


Fig. 68 Diagrama de Actividades Caso de uso Eliminar Tipo Bolsa

Diagrama de Actividades Caso de uso Buscar Tipo Bolsa

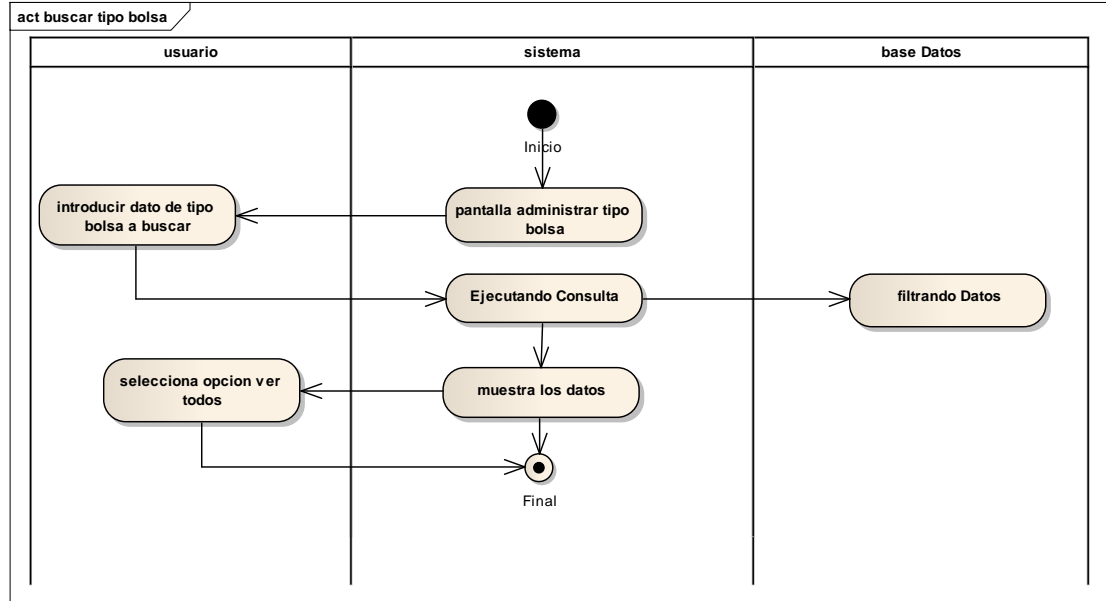


Fig. 69 Diagrama de Actividades Caso de uso Buscar Tipo Bolsa

Diagrama de Actividades Caso de uso Administrar Tipo Donación

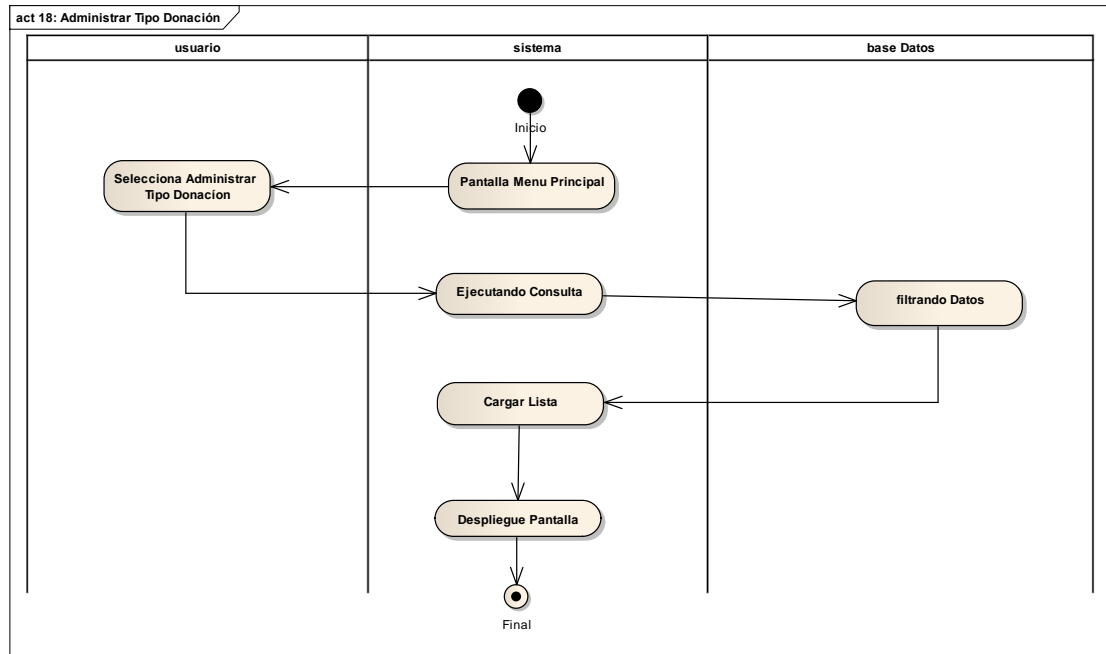


Fig. 70 Diagrama de Actividades Caso de uso Administrar Tipo Donación

Diagrama de Actividades Caso de uso Adicionar Tipo Donación

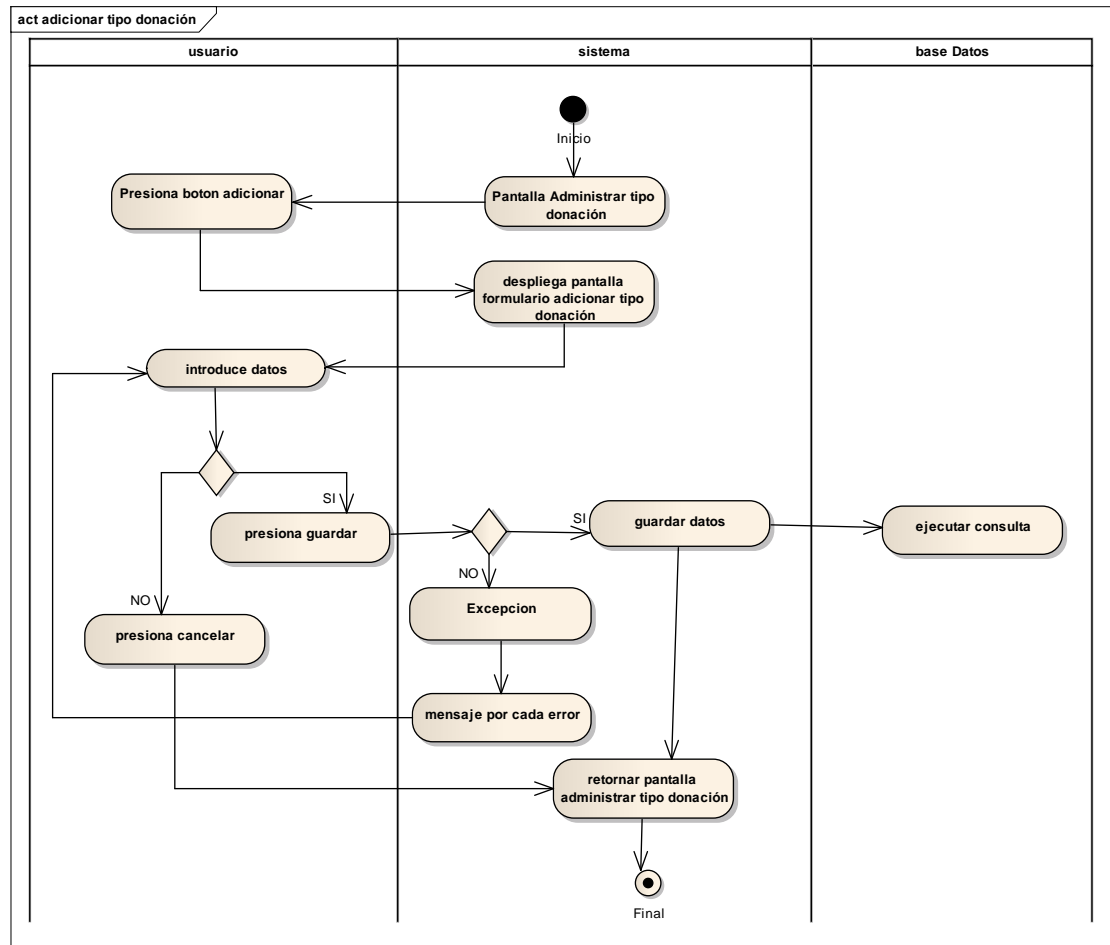


Fig. 71 Diagrama de Actividades Caso de uso Adicionar Tipo Donación

Diagrama de Actividades Caso de uso Modificar Tipo Donación

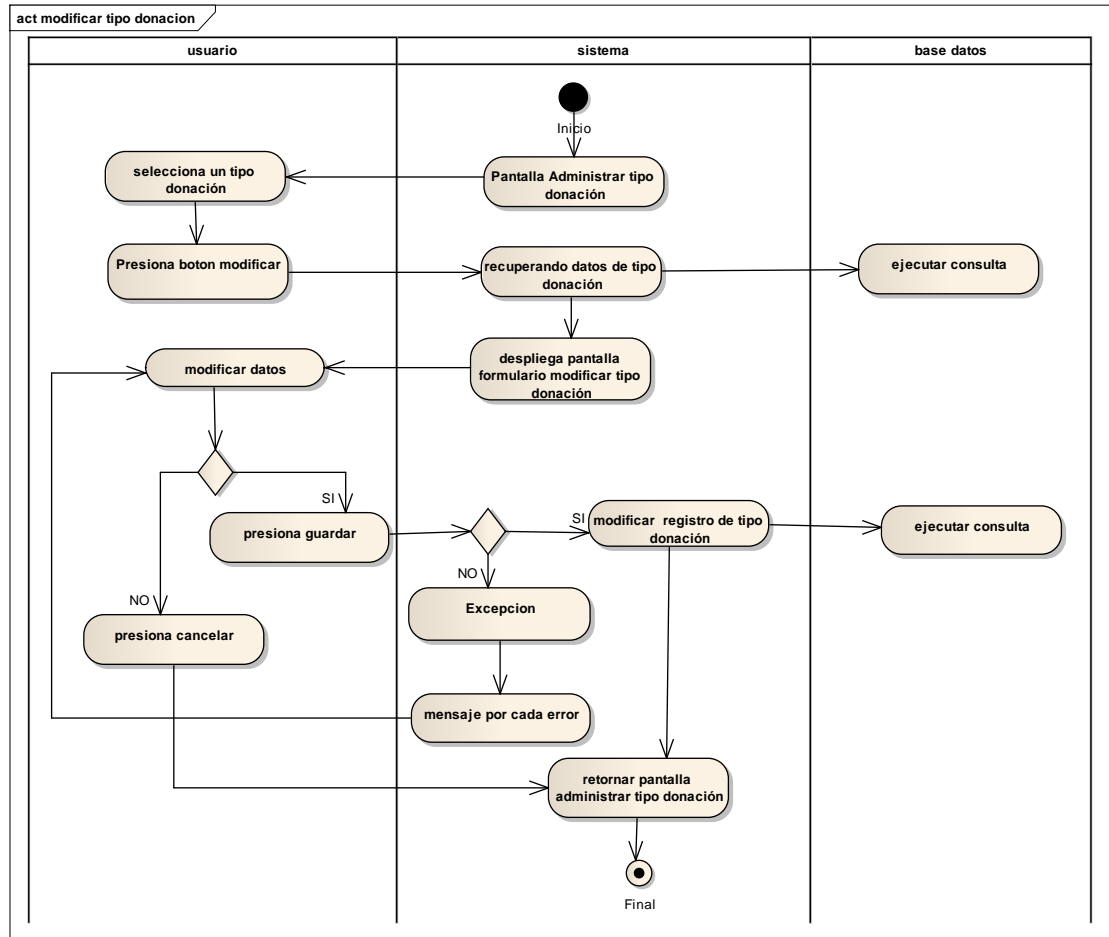


Fig. 72 Diagrama de Actividades Caso de uso Modificar Tipo Donación

Diagrama de Actividades Caso de uso Eliminar Tipo Donación

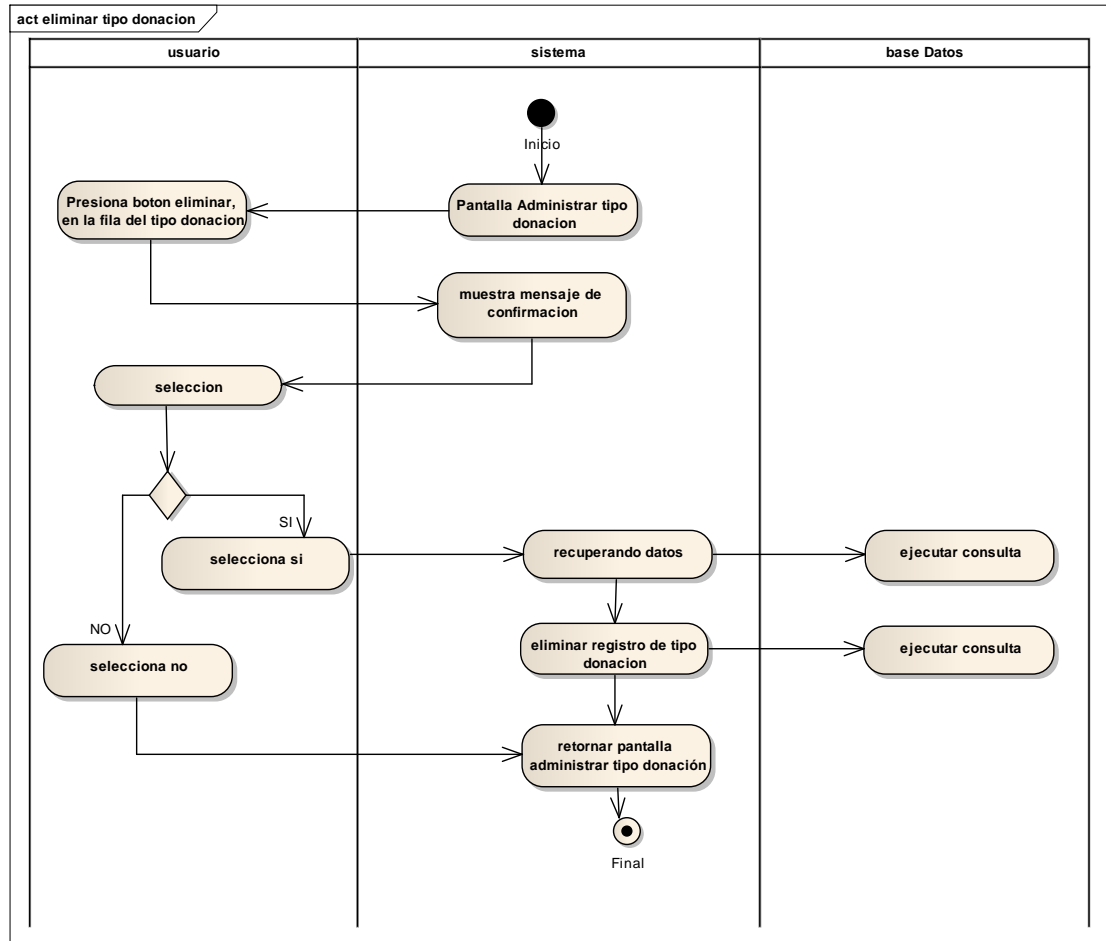


Fig. 73 Diagrama de Actividades Caso de uso Eliminar Tipo Donación

Diagrama de Actividades Caso de uso Administrar Lugar de Donación

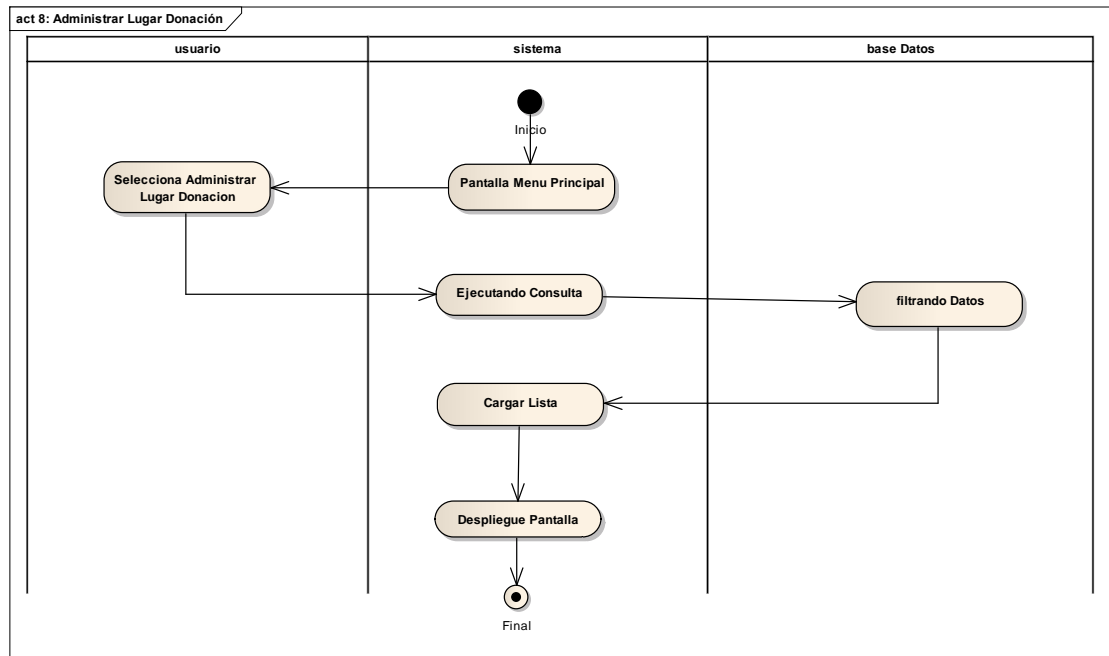


Fig. 74 Diagrama de Actividades Caso de uso Administrar Lugar de Donación

Diagrama de Actividades Caso de uso Adicionar Lugar Donación

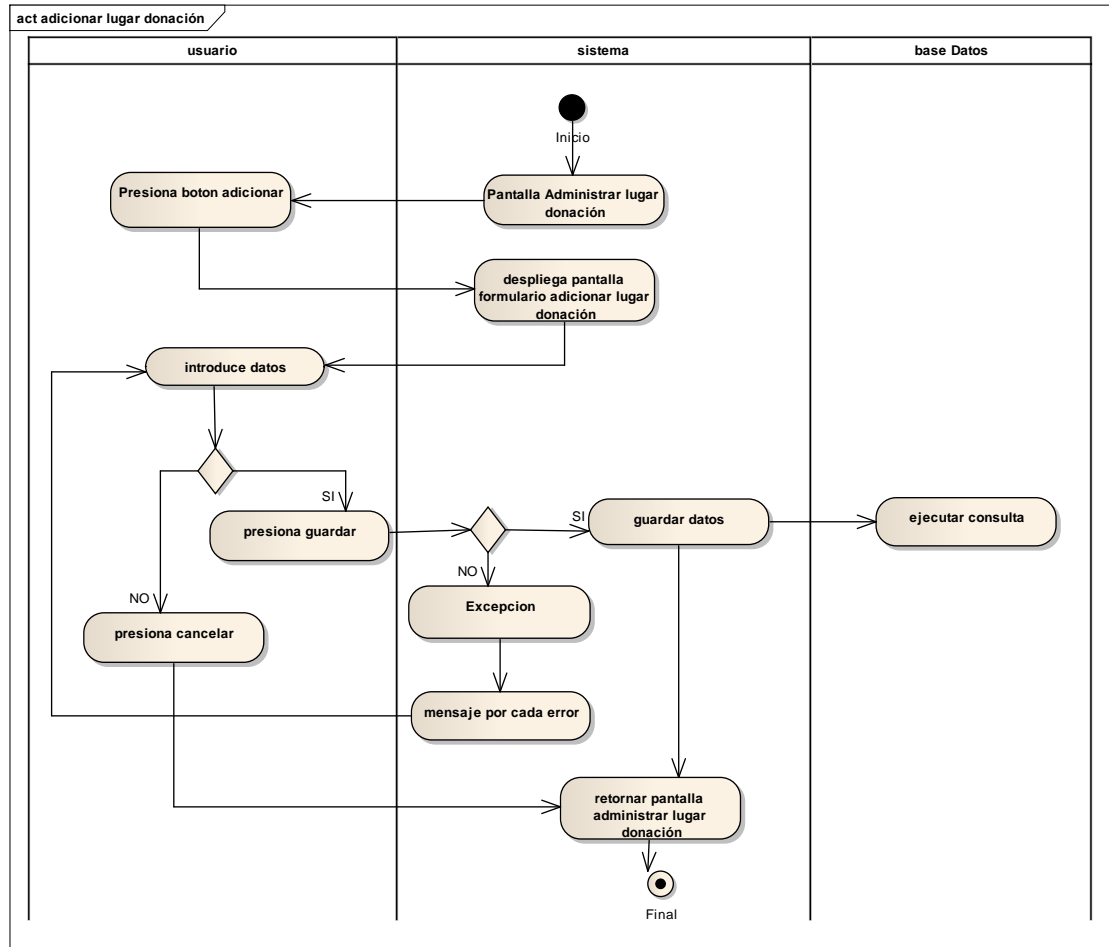


Fig. 75 Diagrama de Actividades Caso de uso Adicionar Lugar Donación

Diagrama de Actividades Caso de uso Modificar Lugar Donación

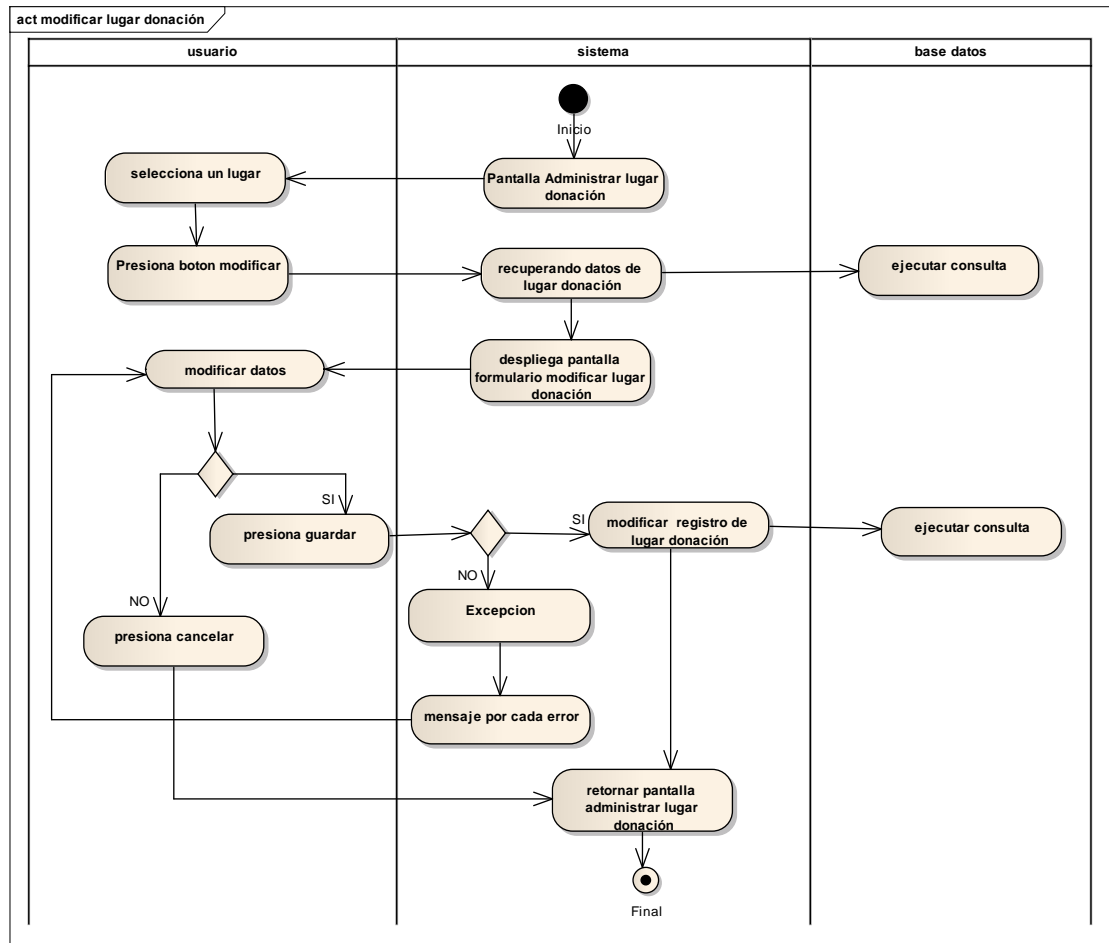


Fig. 76 Diagrama de Actividades Caso de uso Modificar Lugar Donación

Diagrama de Actividades Caso de uso Eliminar Lugar Donación

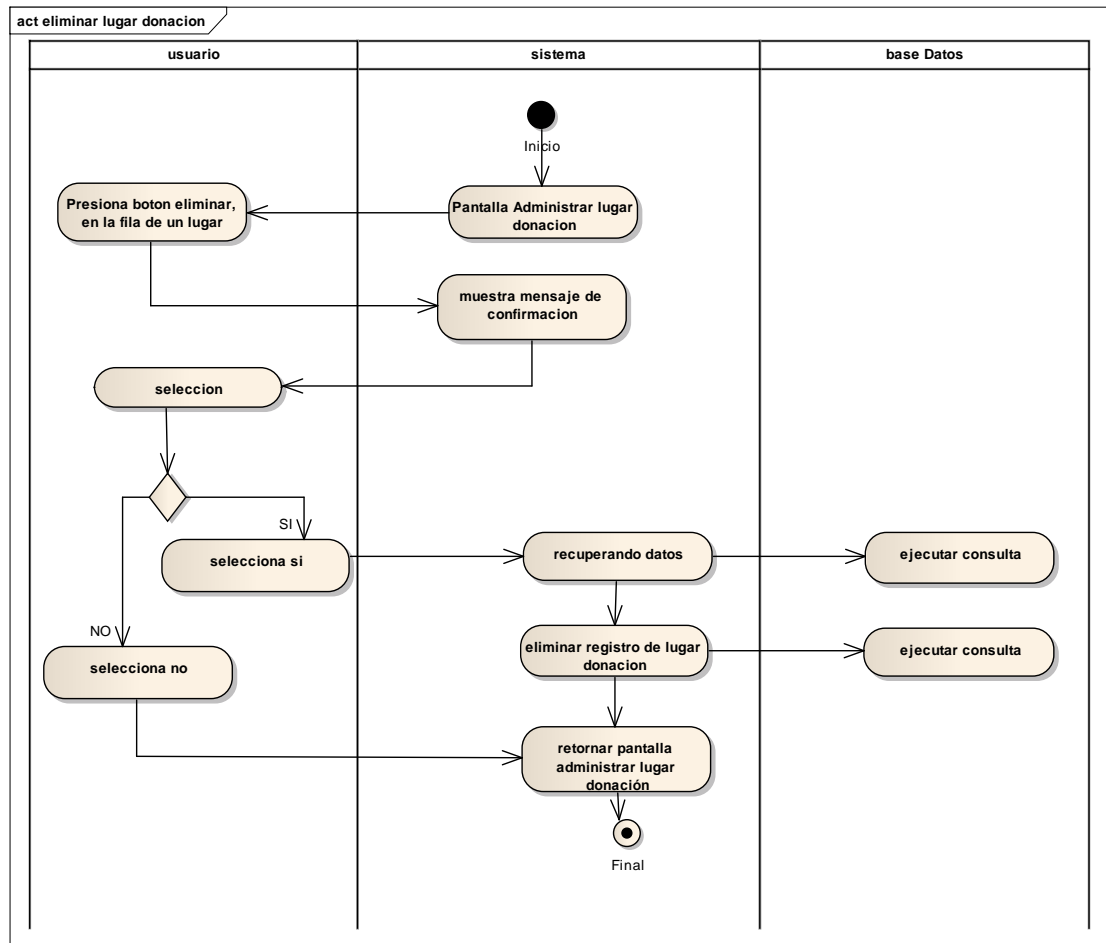


Fig. 77 Diagrama de Actividades Caso de uso Eliminar Lugar Donación

Diagrama de Actividades Caso de uso Administrar Departamentos

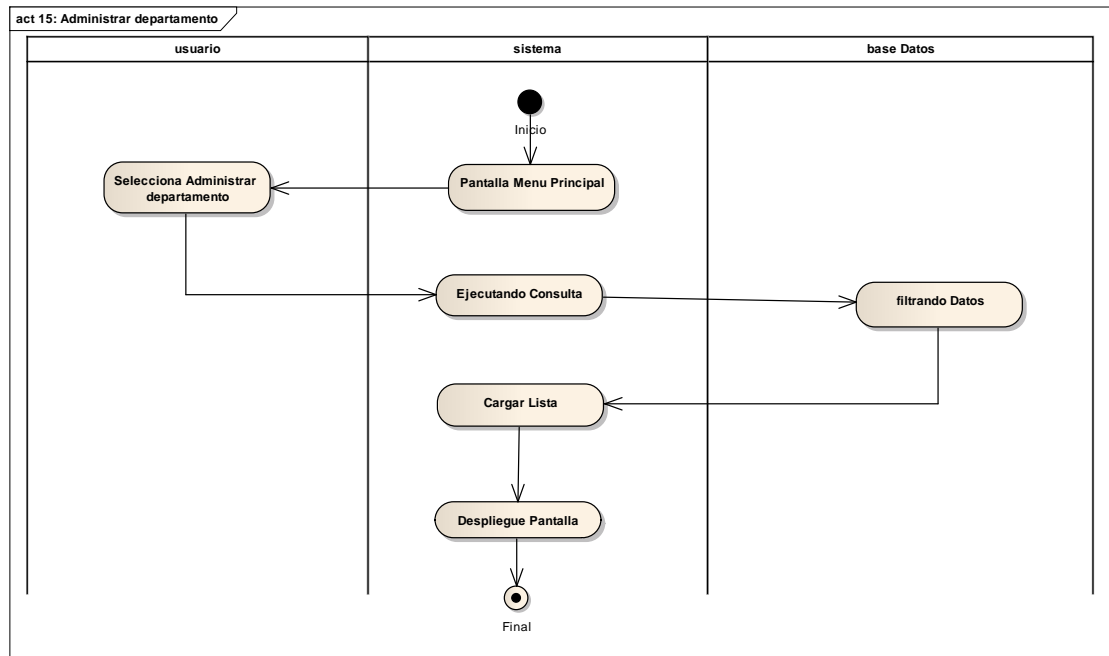


Fig. 78 Diagrama de Actividades Caso de uso Administrar Departamentos

Diagrama de Actividades Caso de uso Adicionar Departamento

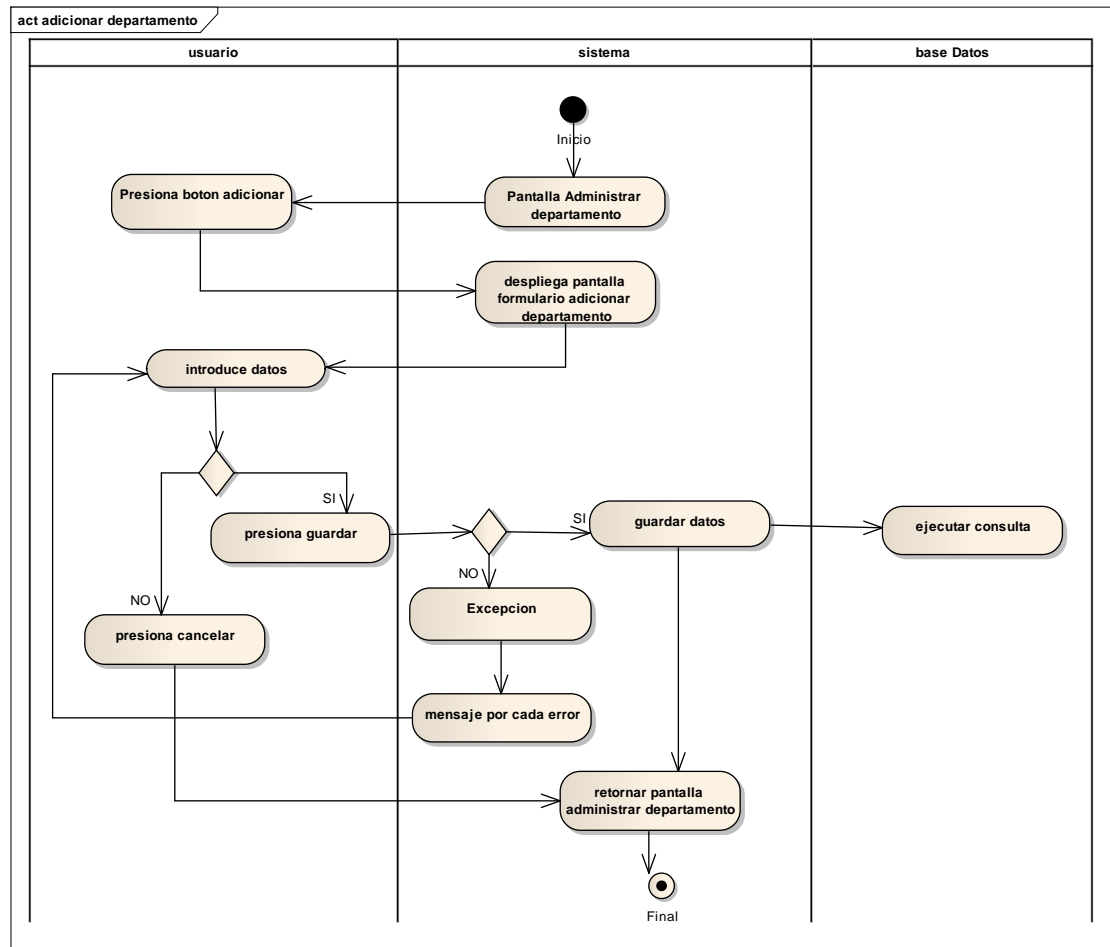


Fig. 79 Diagrama de Actividades Caso de uso Adicionar Departamento

Diagrama de Actividades Caso de uso Modificar Departamento

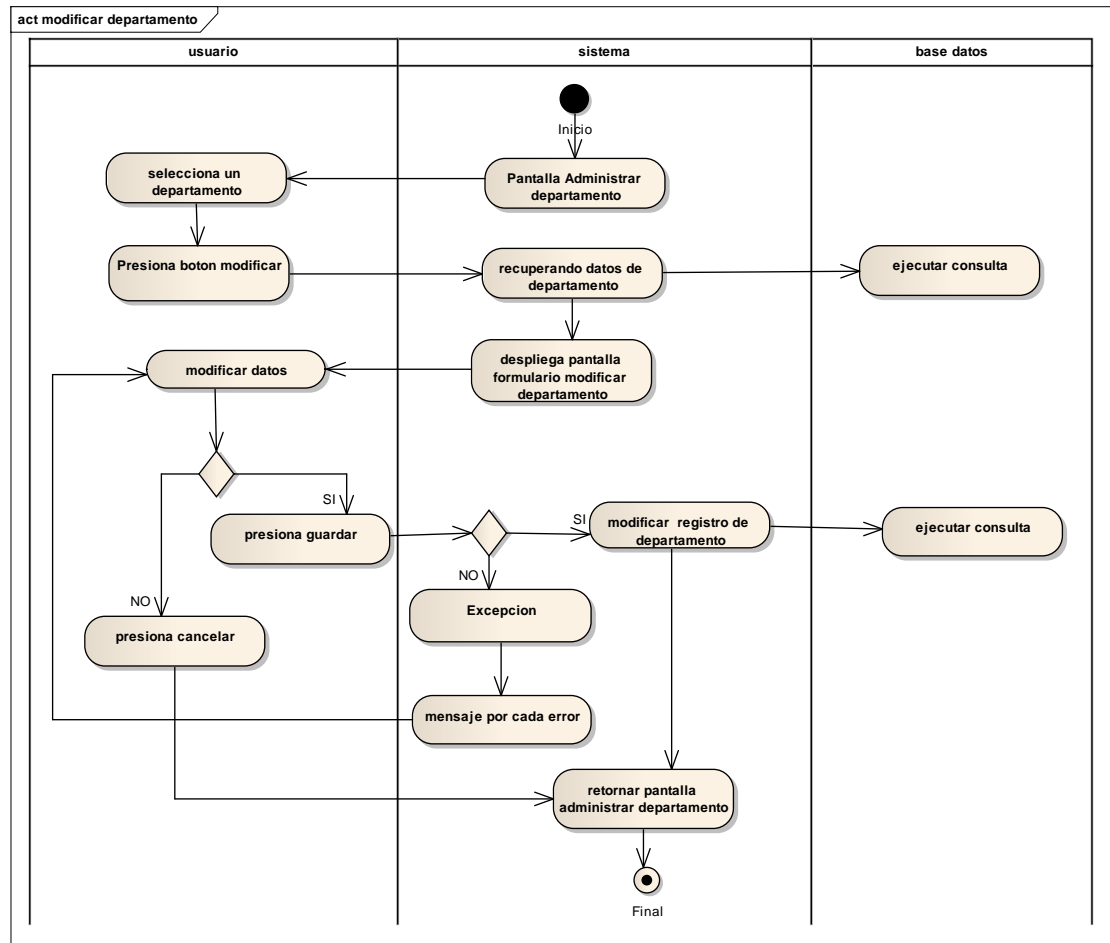


Fig. 80 Diagrama de Actividades Caso de uso Modificar Departamento

Diagrama de Actividades Caso de uso Eliminar Departamento

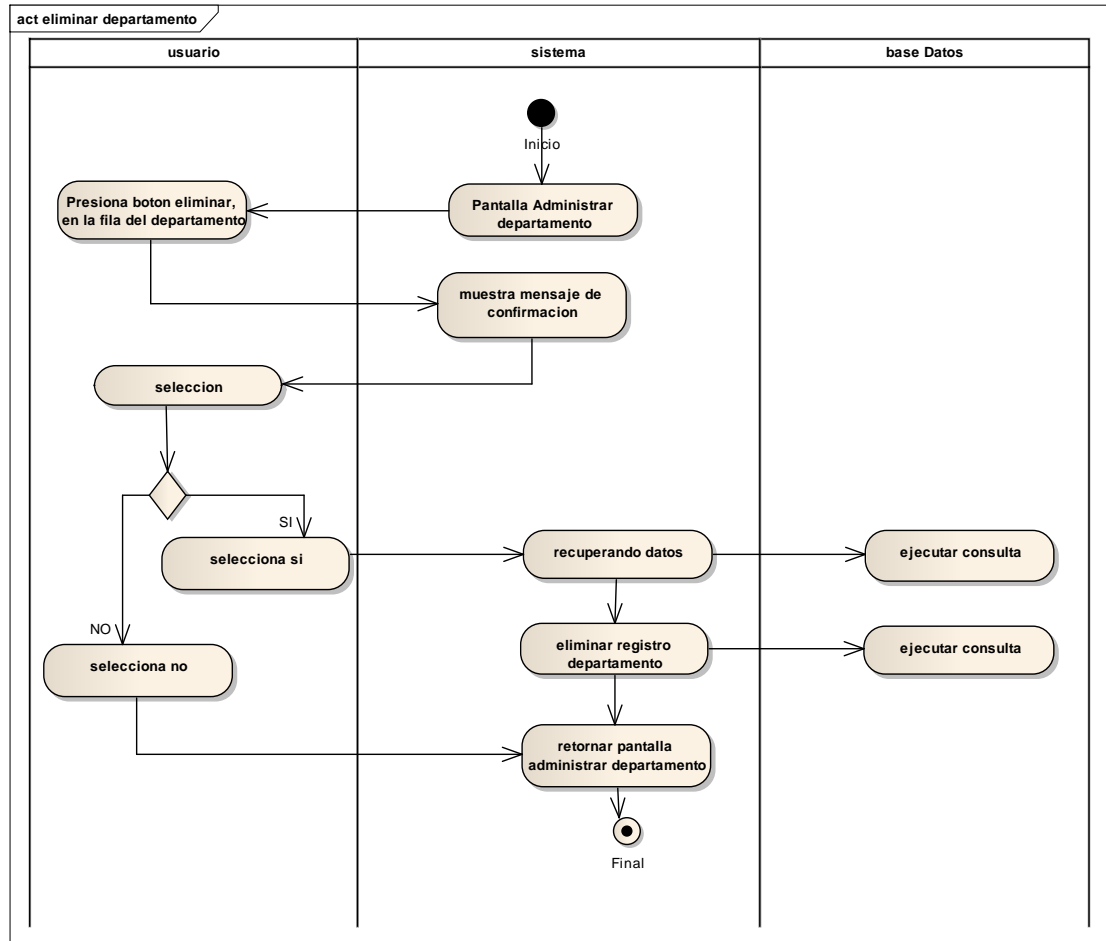


Fig. 81 Diagrama de Actividades Caso de uso Eliminar Departamento

Diagrama de Actividades Caso de uso Administrar Provincia

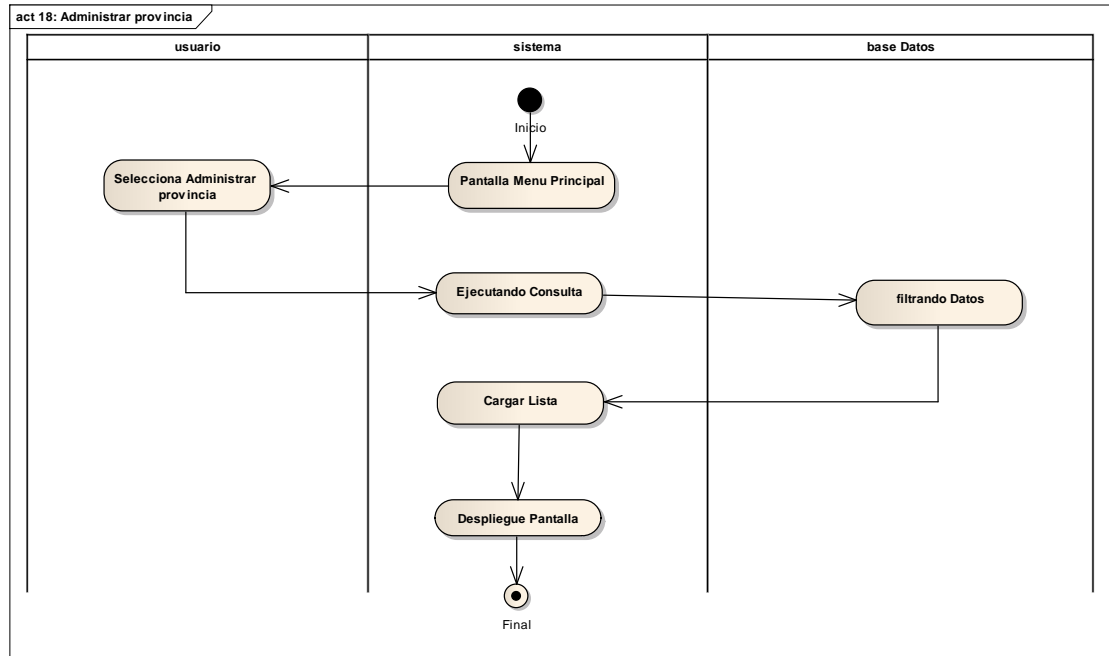


Fig. 82 Diagrama de Actividades Caso de uso Administrar Provincia

Diagrama de Actividades Caso de uso Adicionar Provincia

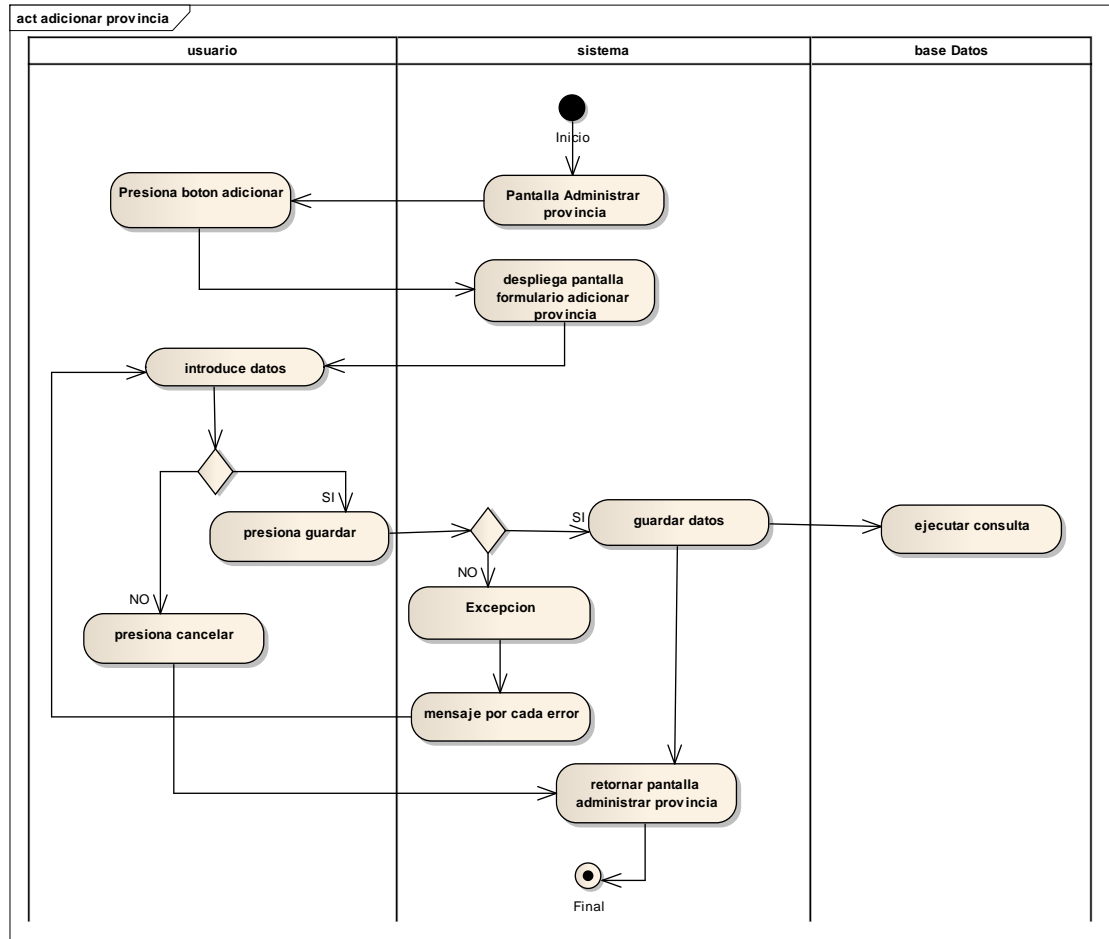


Fig. 83 Diagrama de Actividades Caso de uso Adicionar Provincia

Diagrama de Actividades Caso de uso Modificar Provincia

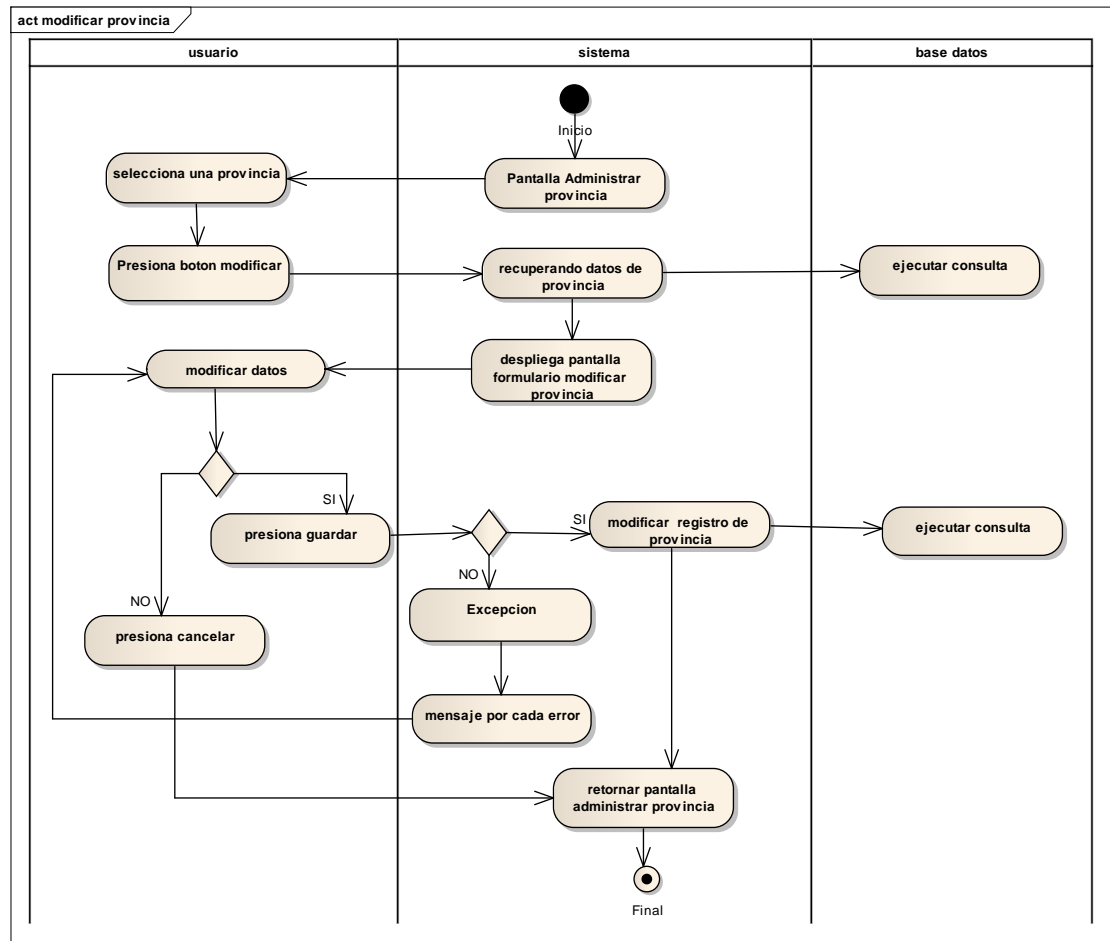


Fig. 84 Diagrama de Actividades Caso de uso Modificar Provincia

Diagrama de Actividades Caso de uso Eliminar Provincia

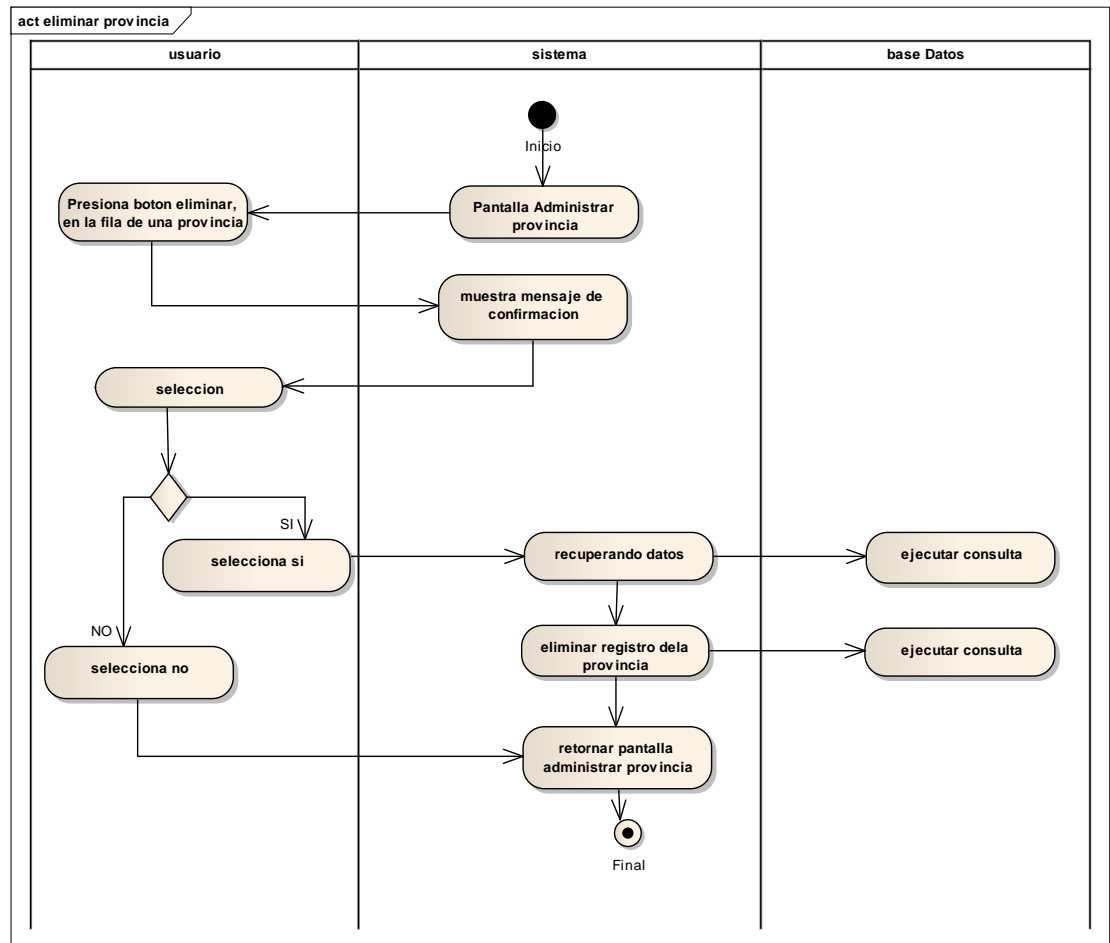


Fig. 85 Diagrama de Actividades Caso de uso Eliminar Provincia

Diagrama de Actividades Caso de uso Administrar ciudad

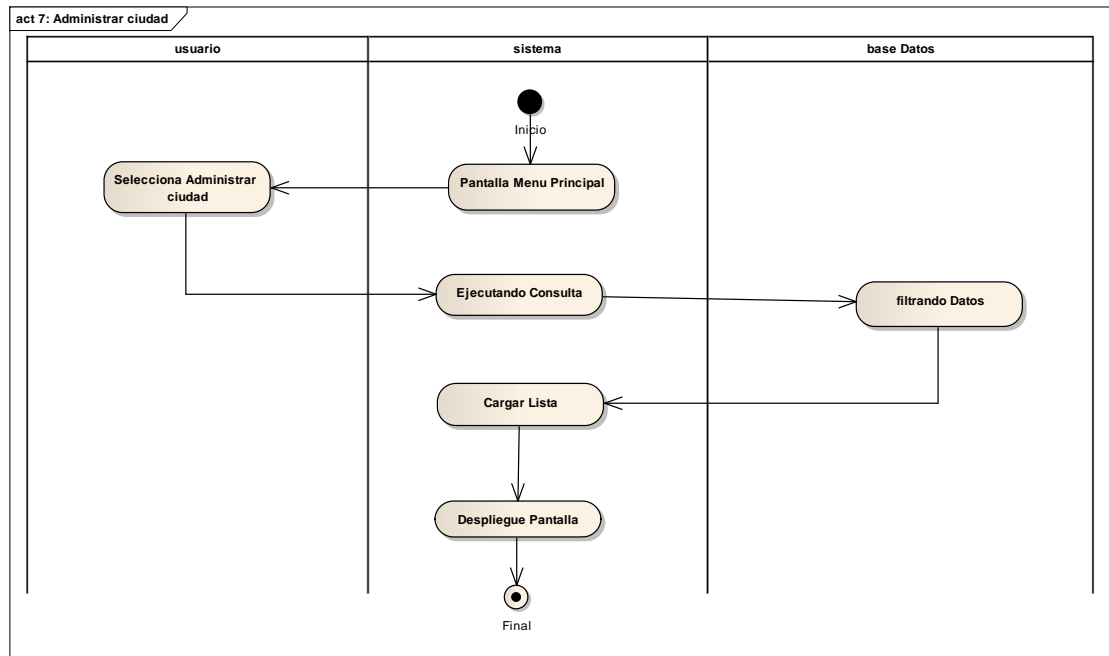


Fig. 86 Diagrama de Actividades Caso de uso Administrar ciudad

Diagrama de Actividades Caso de uso Adicionar Ciudad

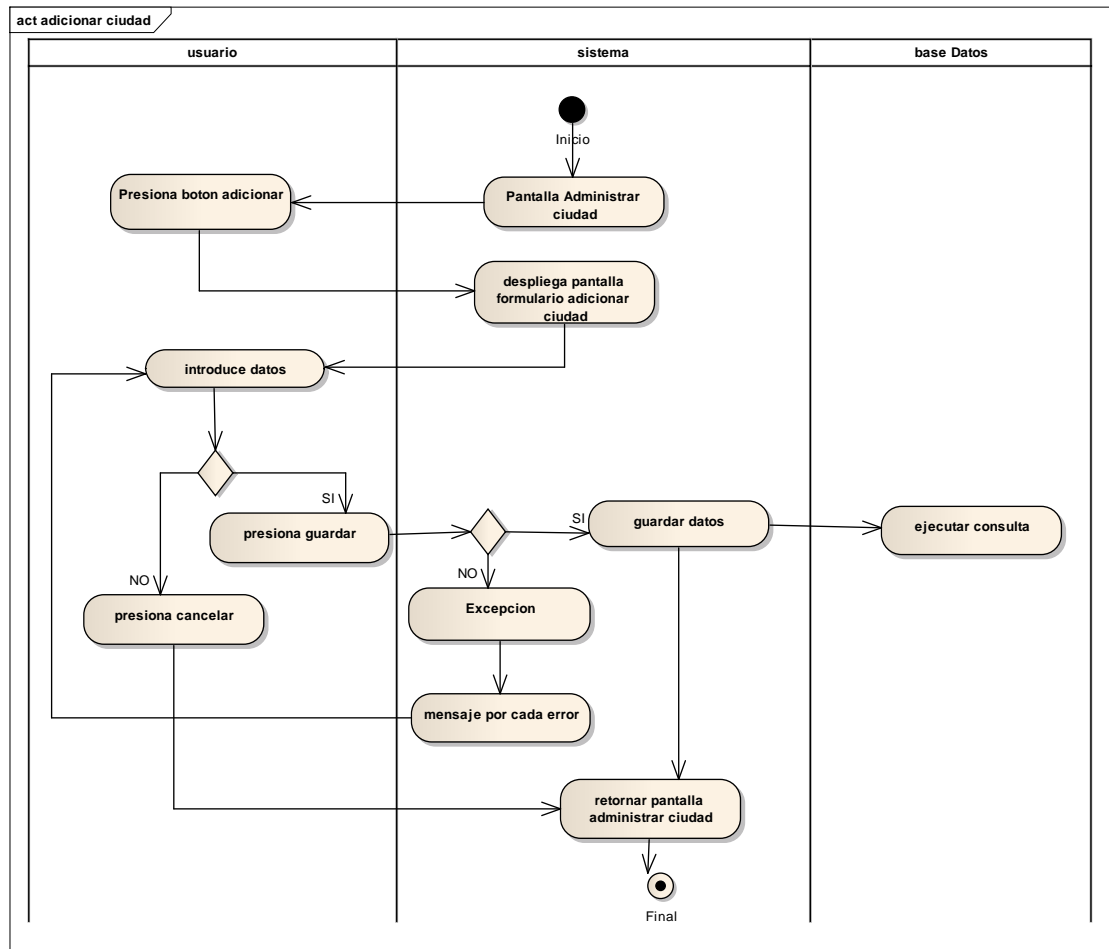


Fig. 87 Diagrama de Actividades Caso de uso Adicionar Ciudad

Diagrama de Actividades Caso de uso Modificar Ciudad

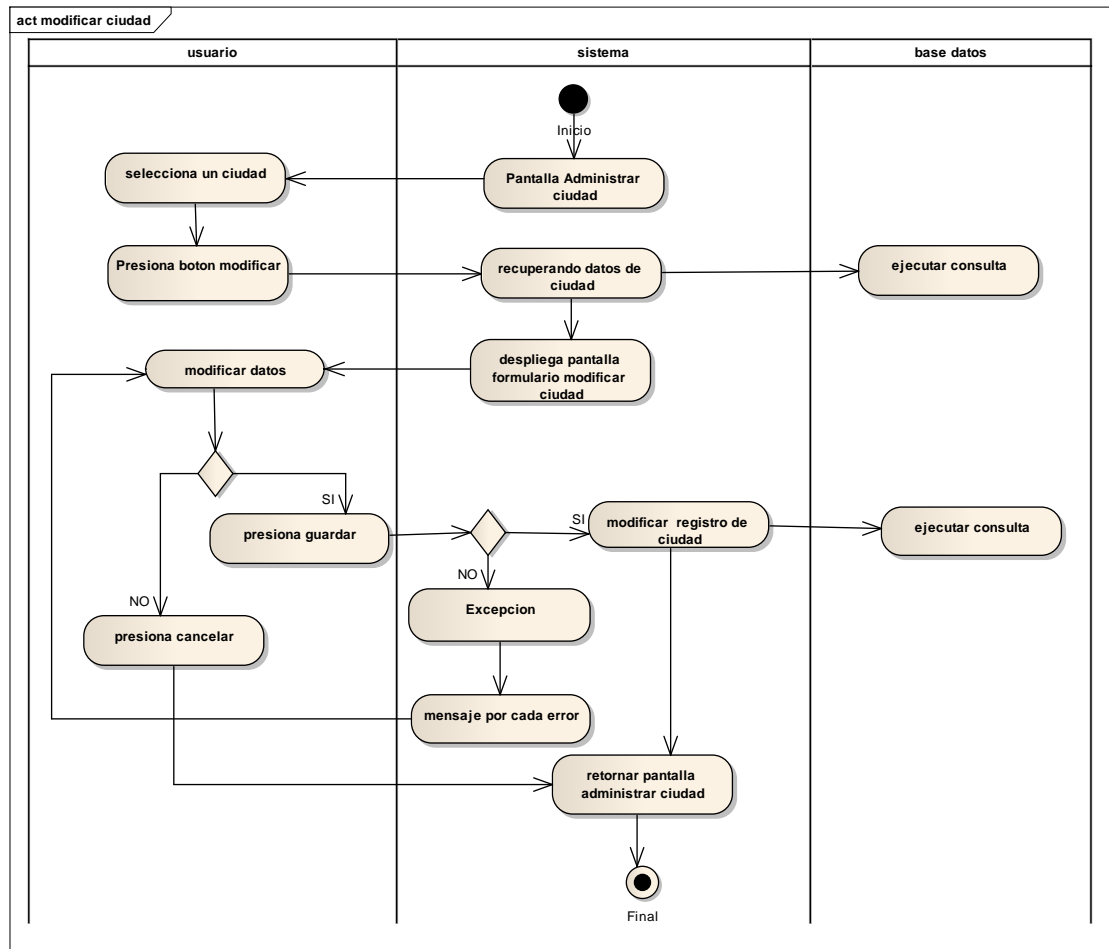


Fig. 88 Diagrama de Actividades Caso de uso Modificar Ciudad

Diagrama de Actividades Caso de uso Eliminar Ciudad

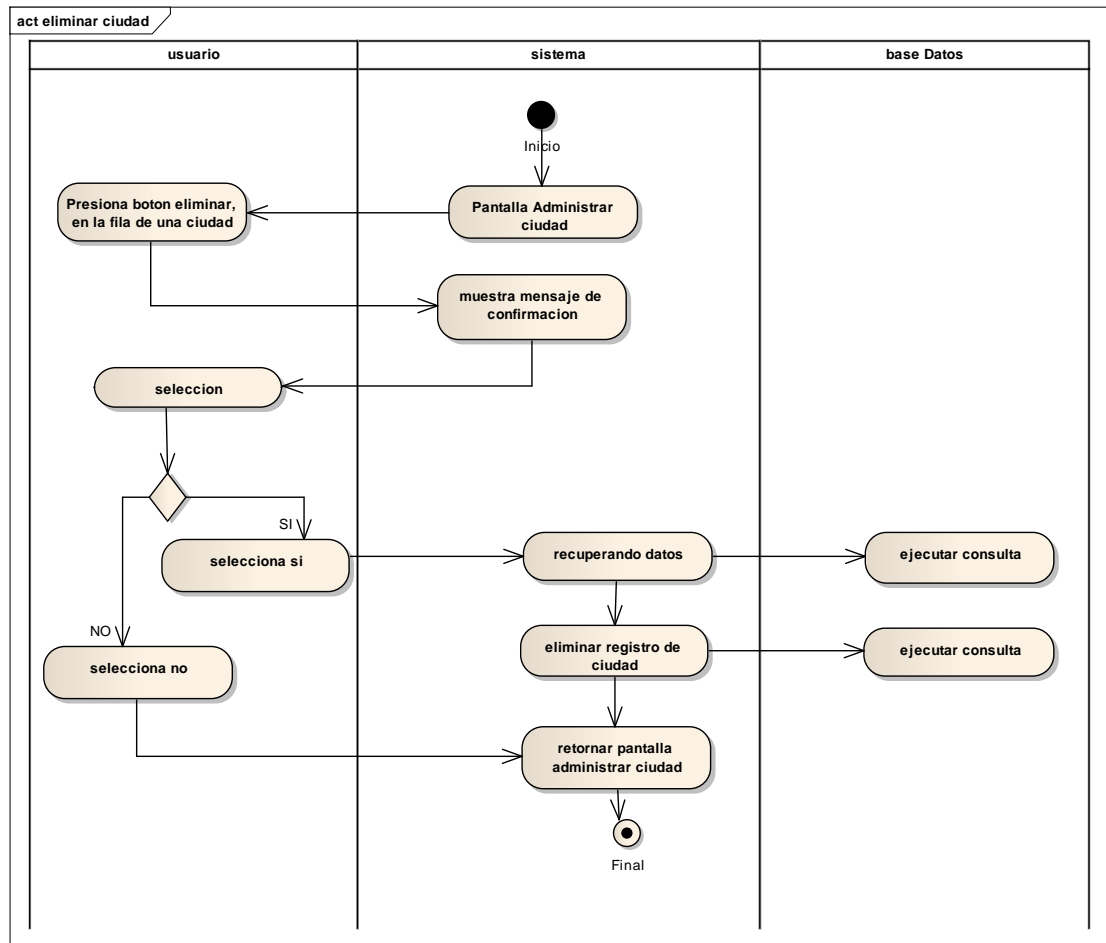


Fig. 89 Diagrama de Actividades Caso de uso Eliminar Ciudad

Diagrama de Actividades Caso de uso Administrar Zonas

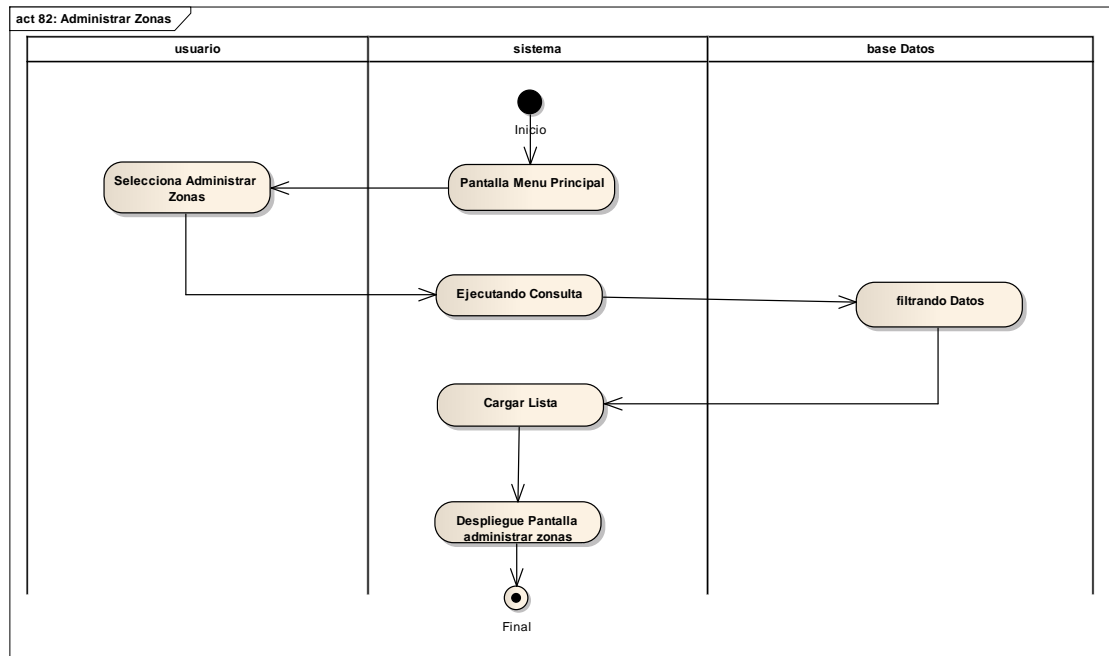


Fig. 90 Diagrama de Actividades Caso de uso Administrar Zonas

Diagrama de Actividades Caso de uso Adicionar Zona

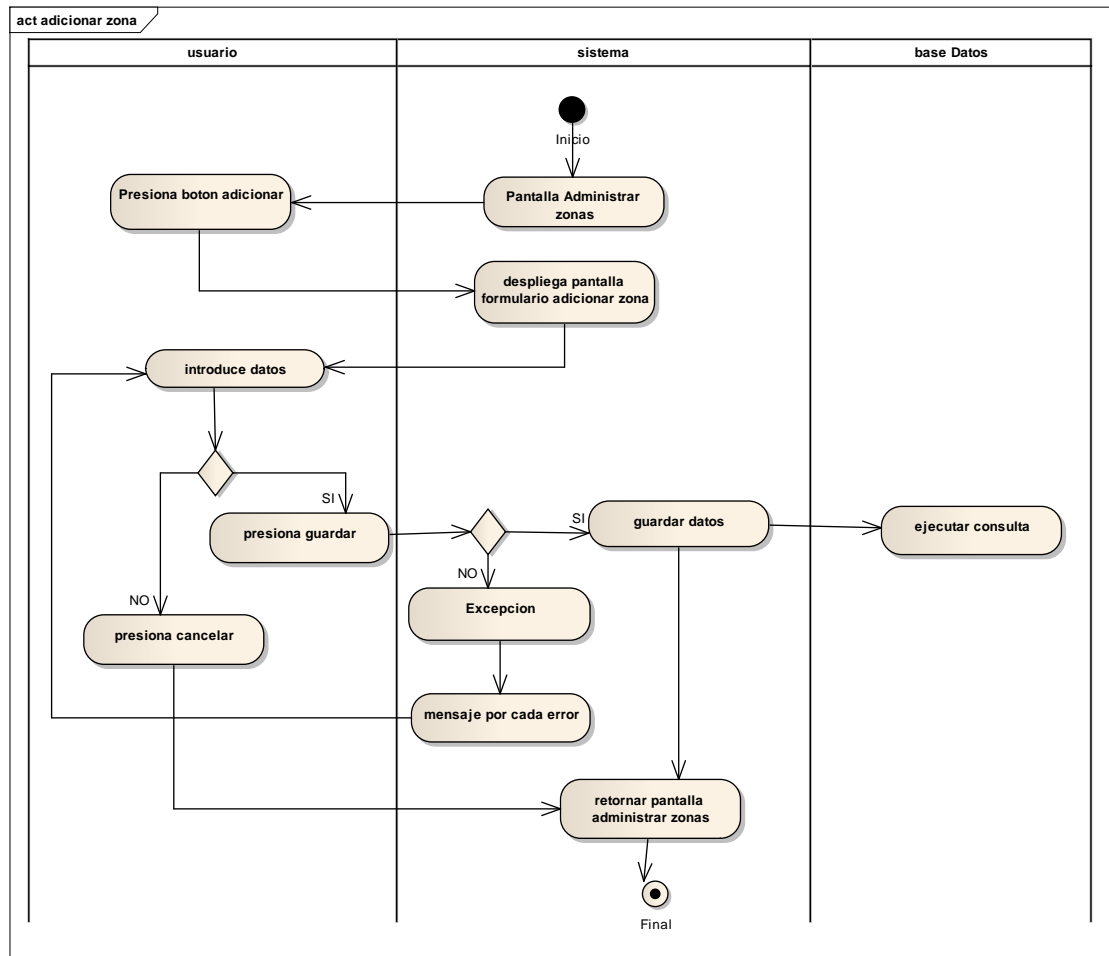


Fig. 91 Diagrama de Actividades Caso de uso Adicionar Zona

Diagrama de Actividades Caso de uso Modificar Zona

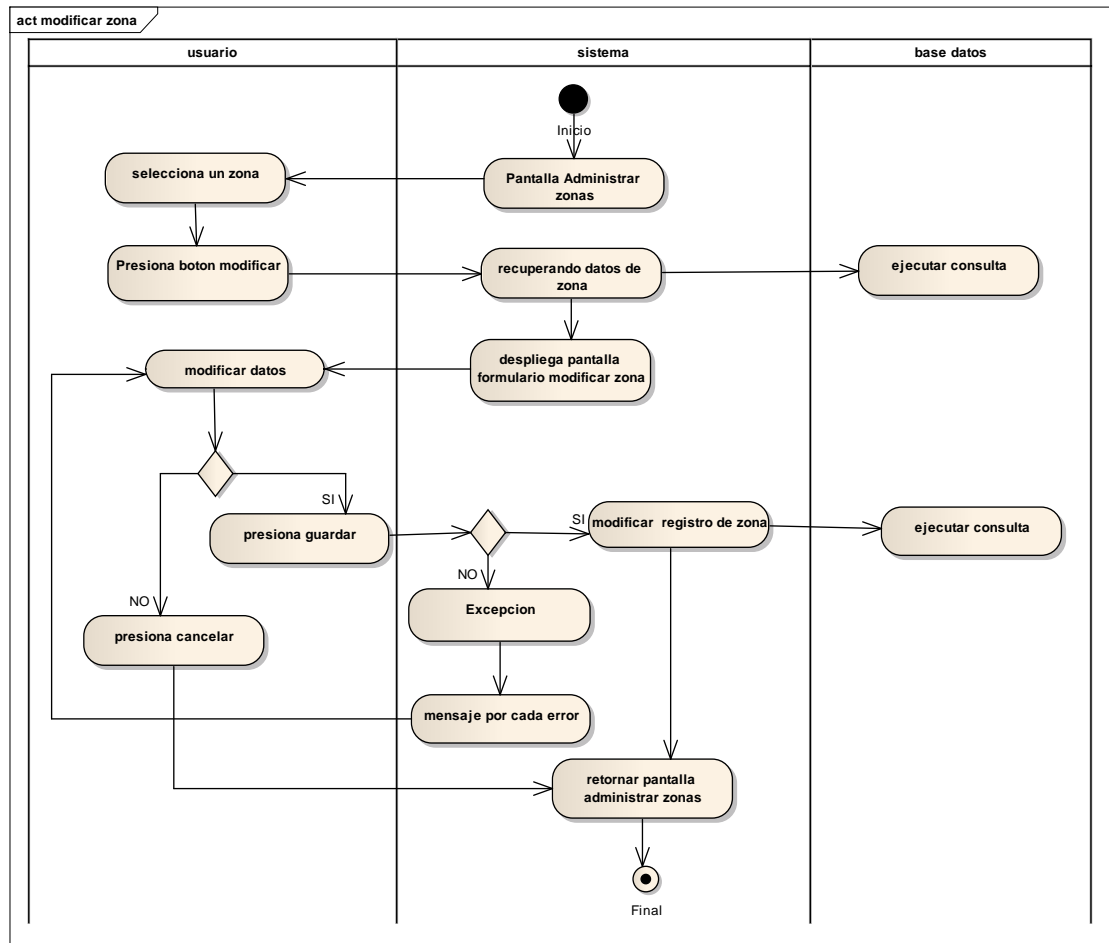


Fig. 92 Diagrama de Actividades Caso de uso Modificar Zona

Diagrama de Actividades Caso de uso Eliminar Zona

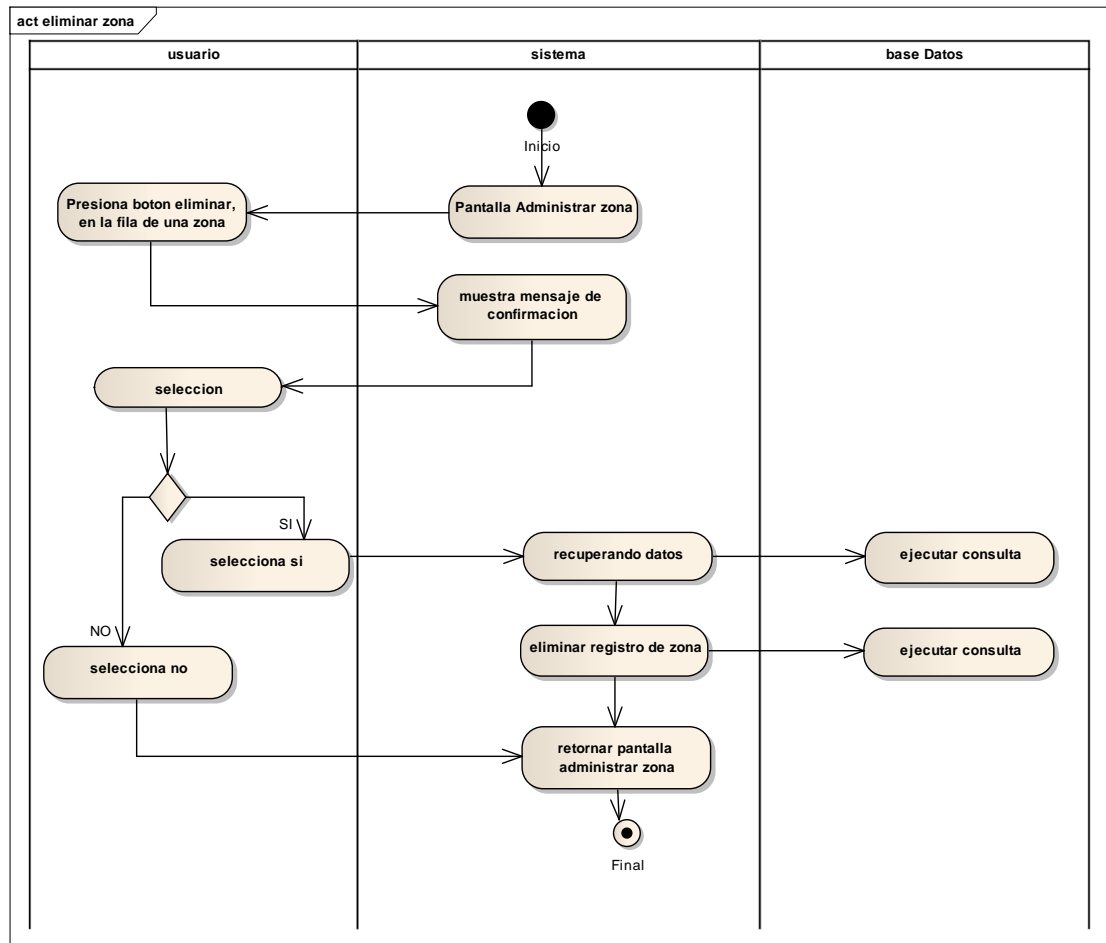


Fig. 93 Diagrama de Actividades Caso de uso Eliminar Zona

Diagrama de Actividades Caso de uso Administrar Enfermedades

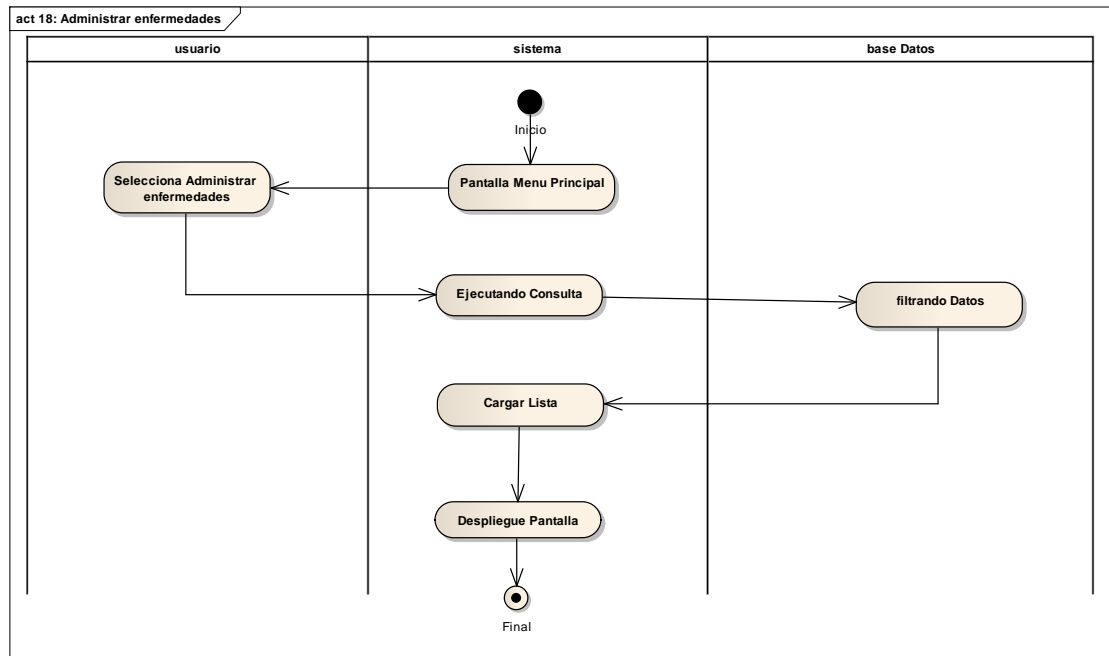


Fig. 94 Diagrama de Actividades Caso de uso Administrar Enfermedades

Diagrama de Actividades Caso de uso Adicionar Enfermedad

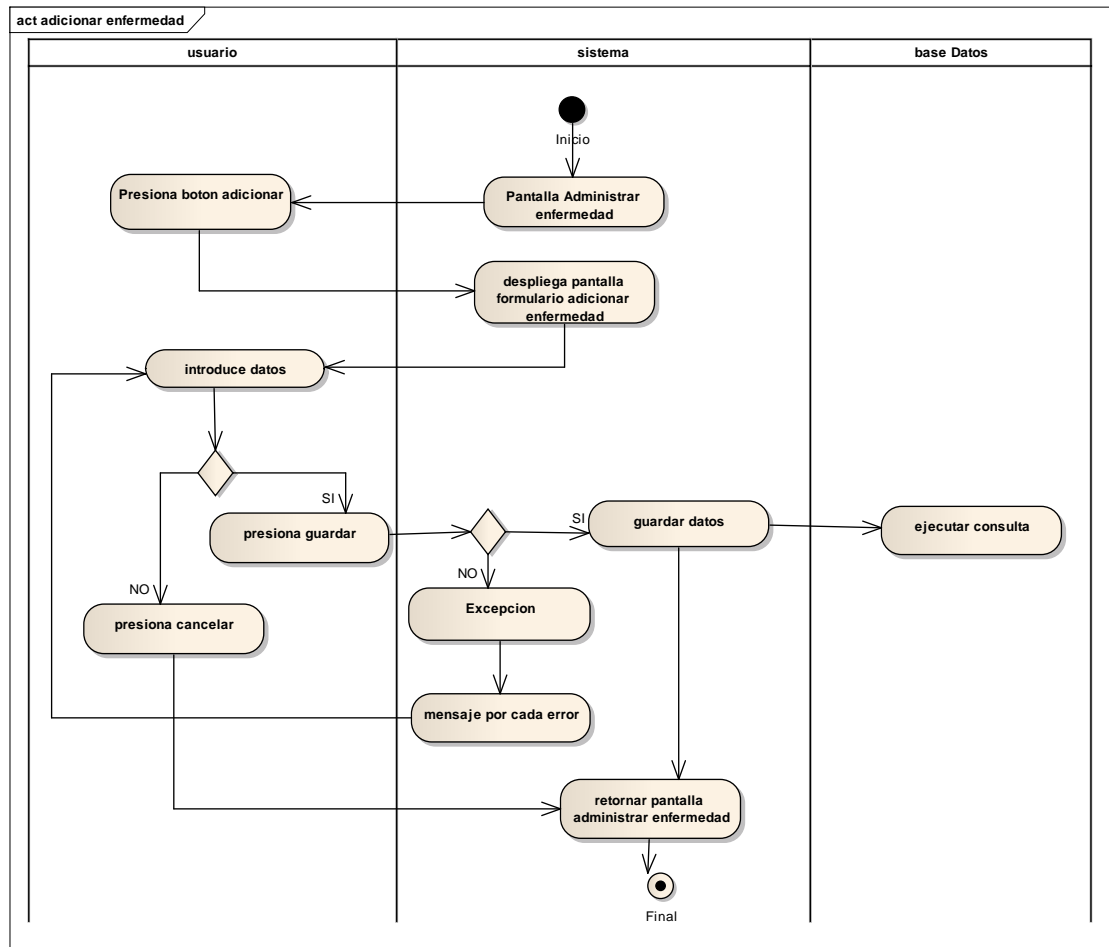


Fig. 95 Diagrama de Actividades Caso de uso Adicionar Enfermedad

Diagrama de Actividades Caso de uso Modificar Enfermedad

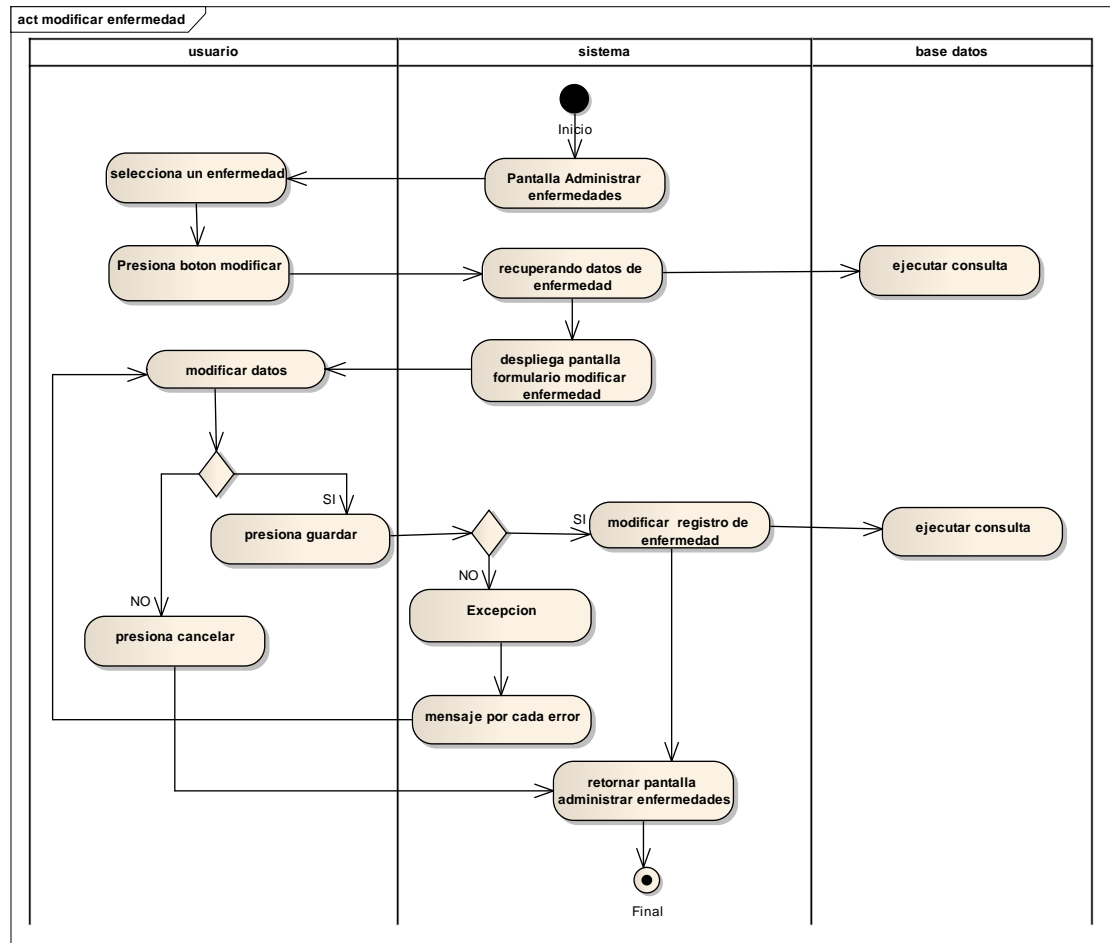


Fig. 96 Diagrama de Actividades Caso de uso Modificar Enfermedad

Diagrama de Actividades Caso de uso Eliminar Enfermedad

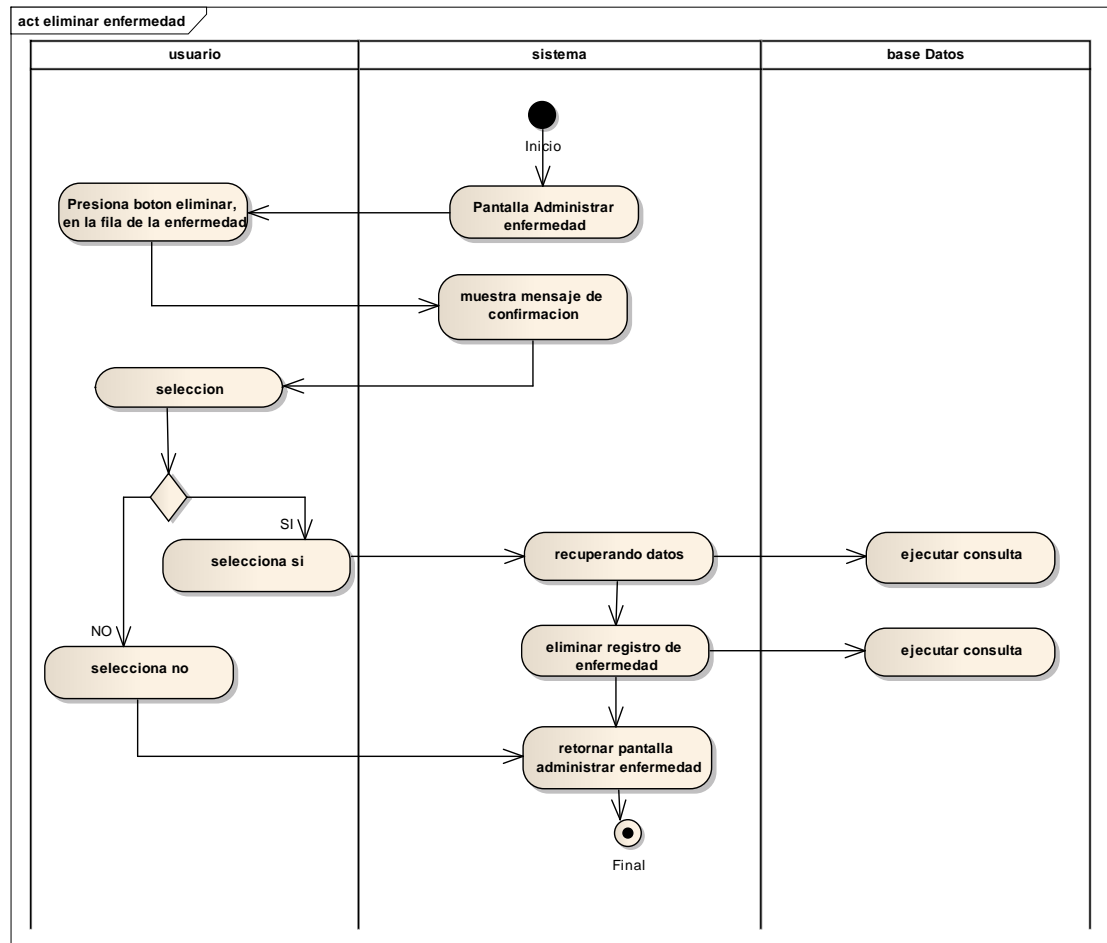


Fig. 97 Diagrama de Actividades Caso de uso Eliminar Enfermedad

Diagrama de Actividades Caso de uso Administrar Usuarios

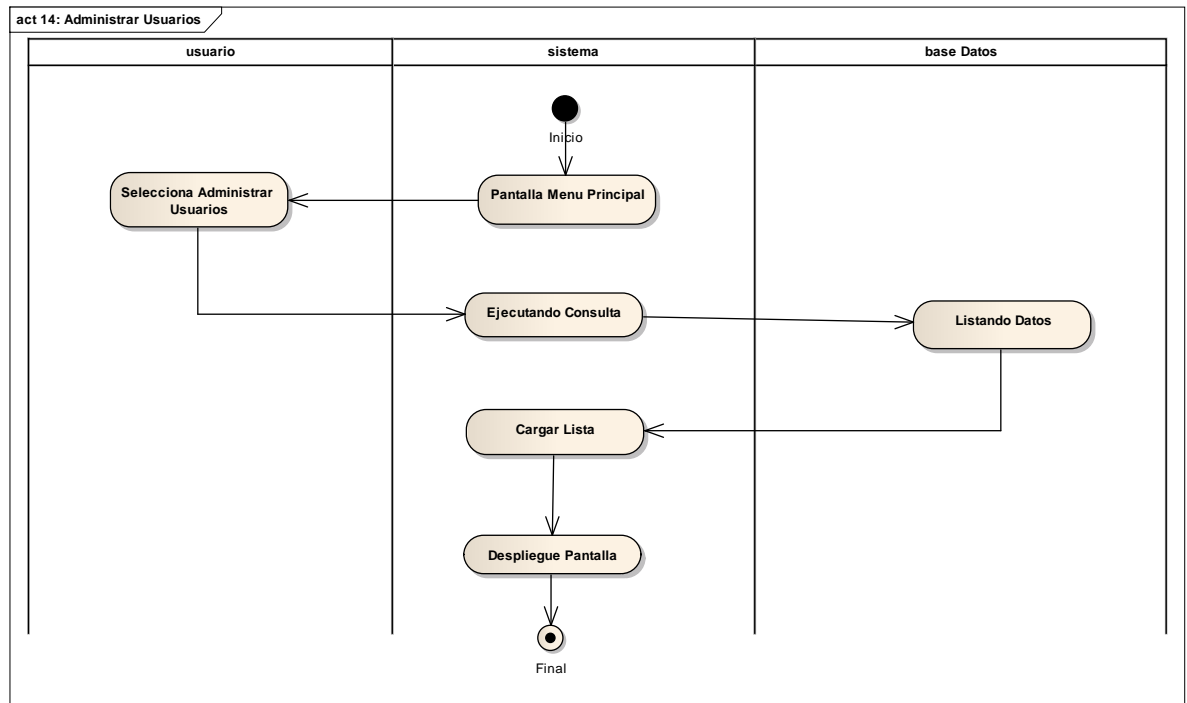


Fig. 98 Diagrama de Actividades Caso de uso Administrar Usuarios

Diagrama de Actividades Caso de uso Adicionar Usuario

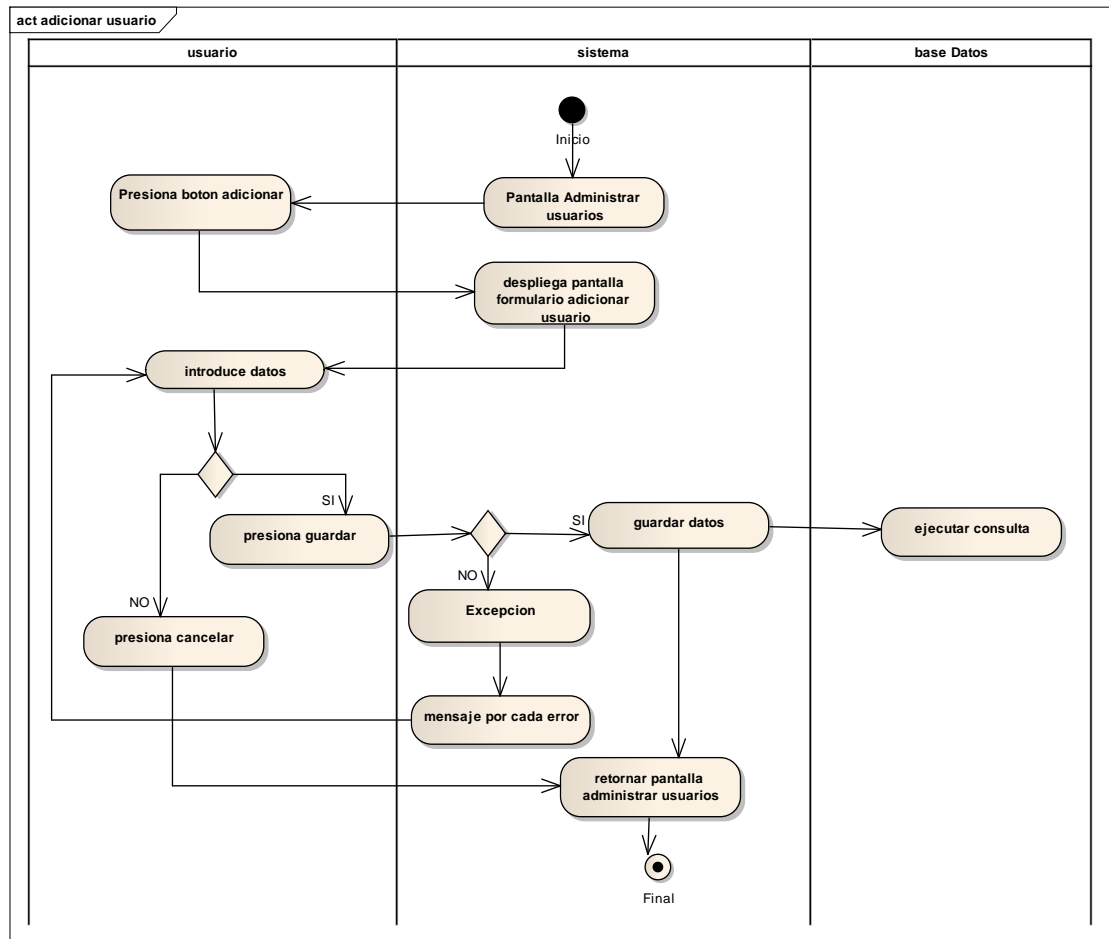


Fig. 99 Diagrama de Actividades Caso de uso Adicionar Usuario

Diagrama de Actividades Caso de uso Modificar Usuario

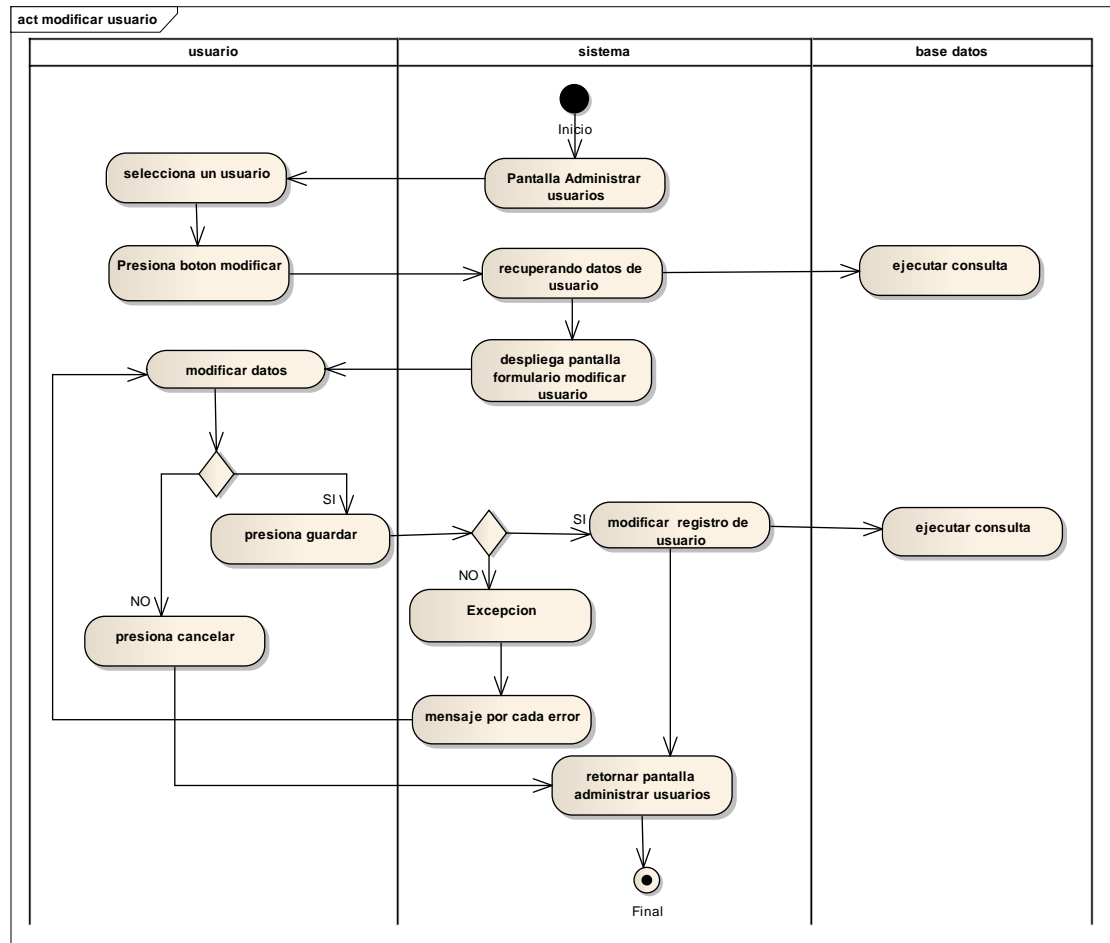


Fig. 100 Diagrama de Actividades Caso de uso Modificar Usuario

Diagrama de Actividades Caso de uso Eliminar Usuario

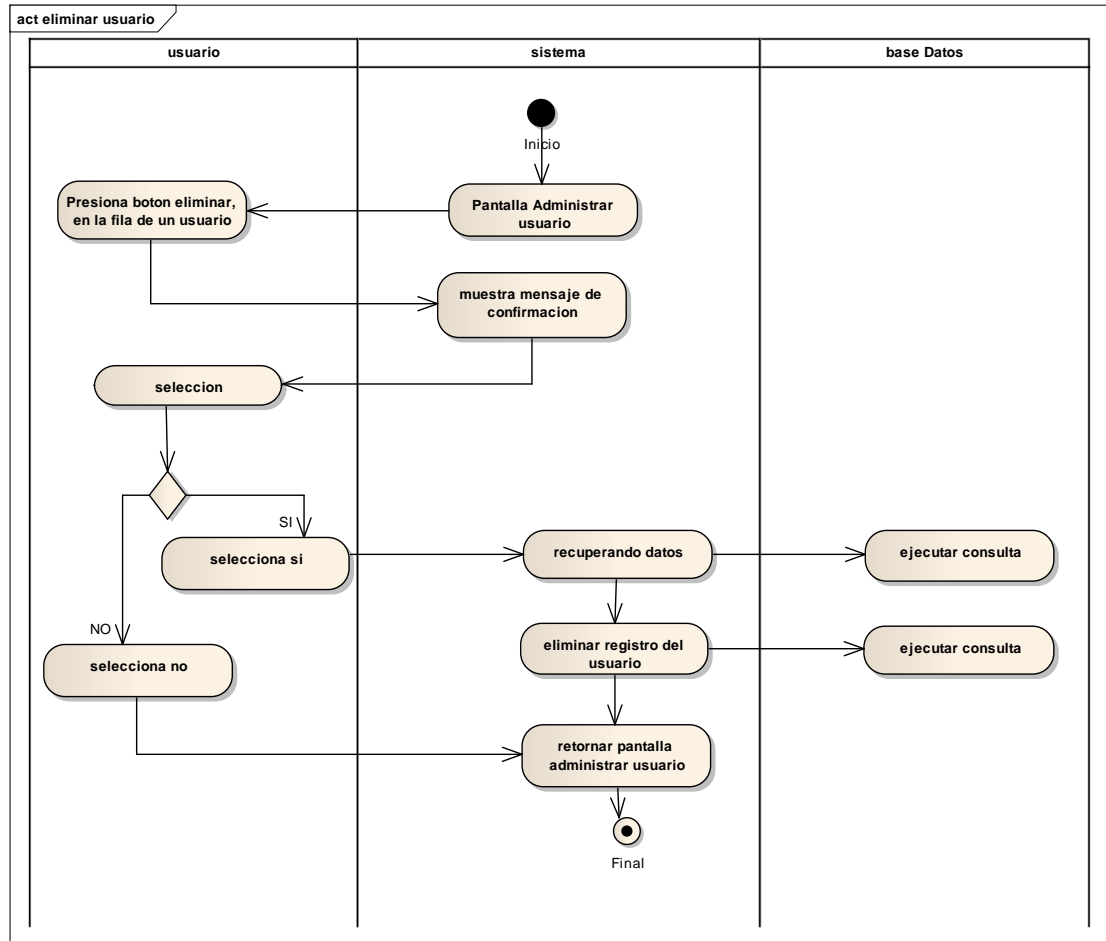


Fig. 101 Diagrama de Actividades Caso de uso Eliminar Usuario

Diagrama de Actividades Caso de uso Habilitar e Inhabilitar Usuario

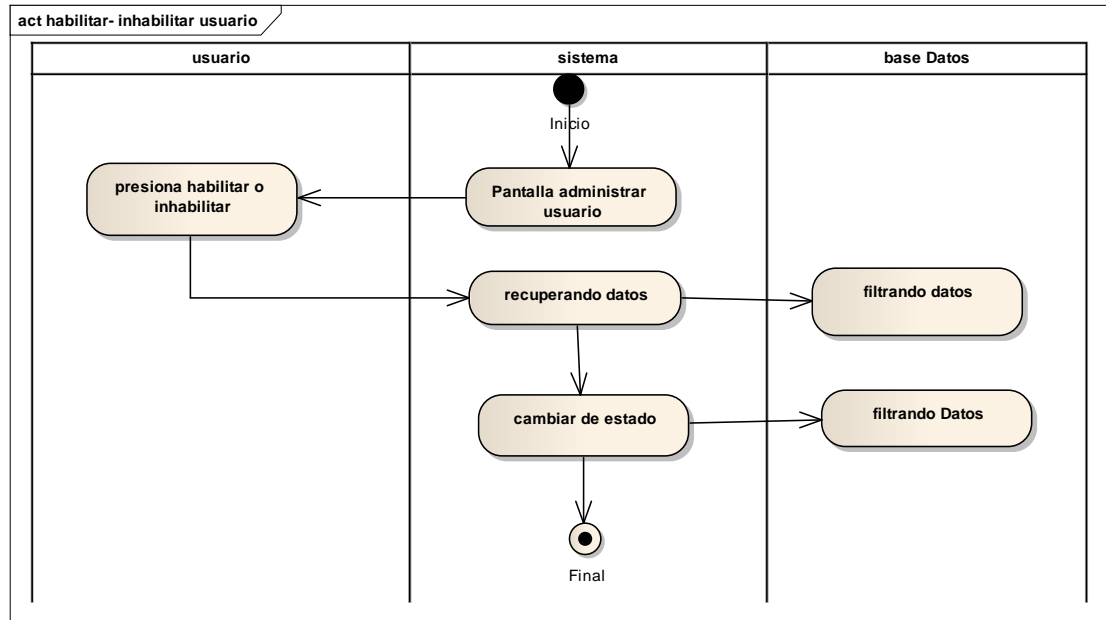


Fig. 102 Diagrama de Actividades Caso de uso Habilitar e Inhabilitar Usuario

Diagrama de Actividades Caso de uso Asignar Menús – Usuario

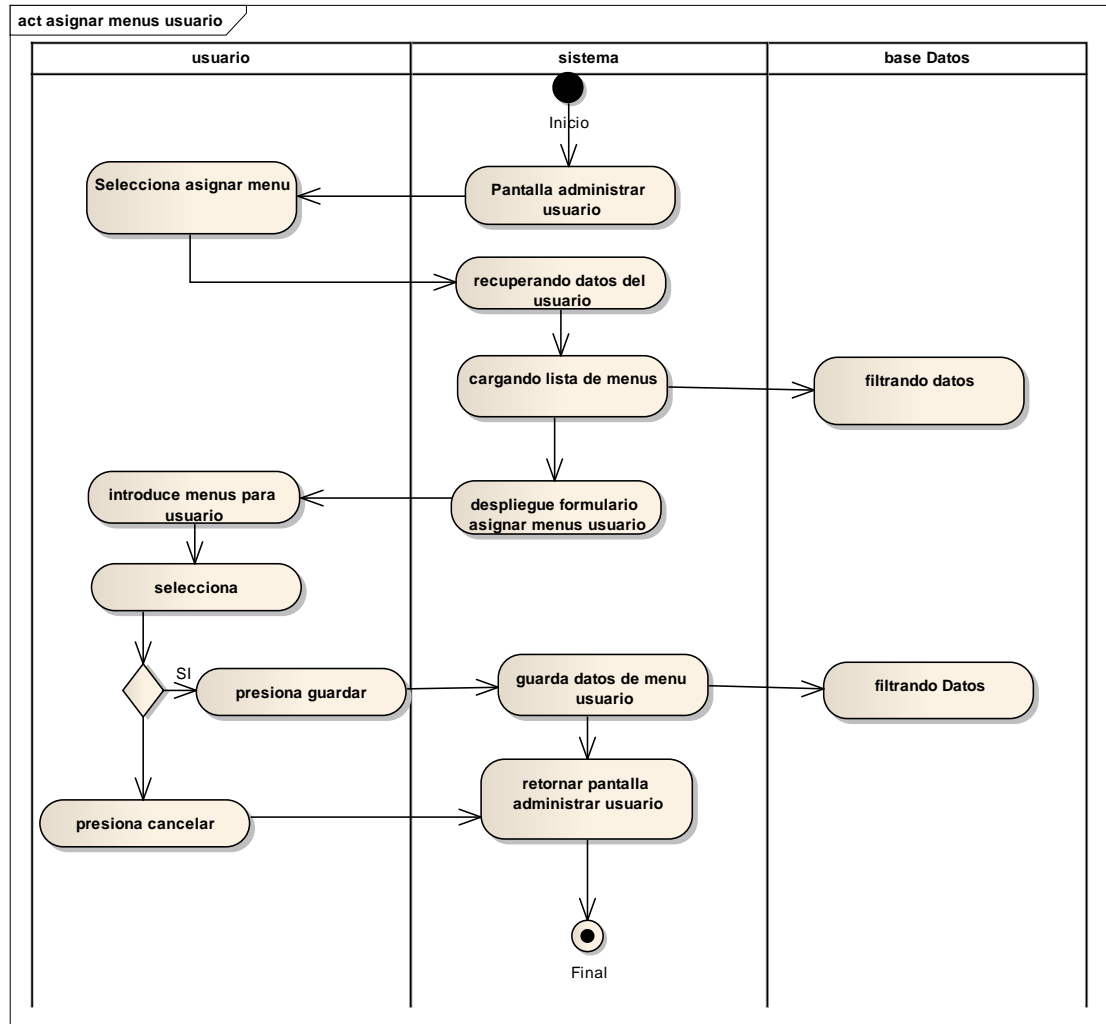


Fig. 103 Diagrama de Actividades Caso de uso Asignar Menús – Usuario

Diagrama de Actividades Caso de uso Eliminar Menús – Usuario

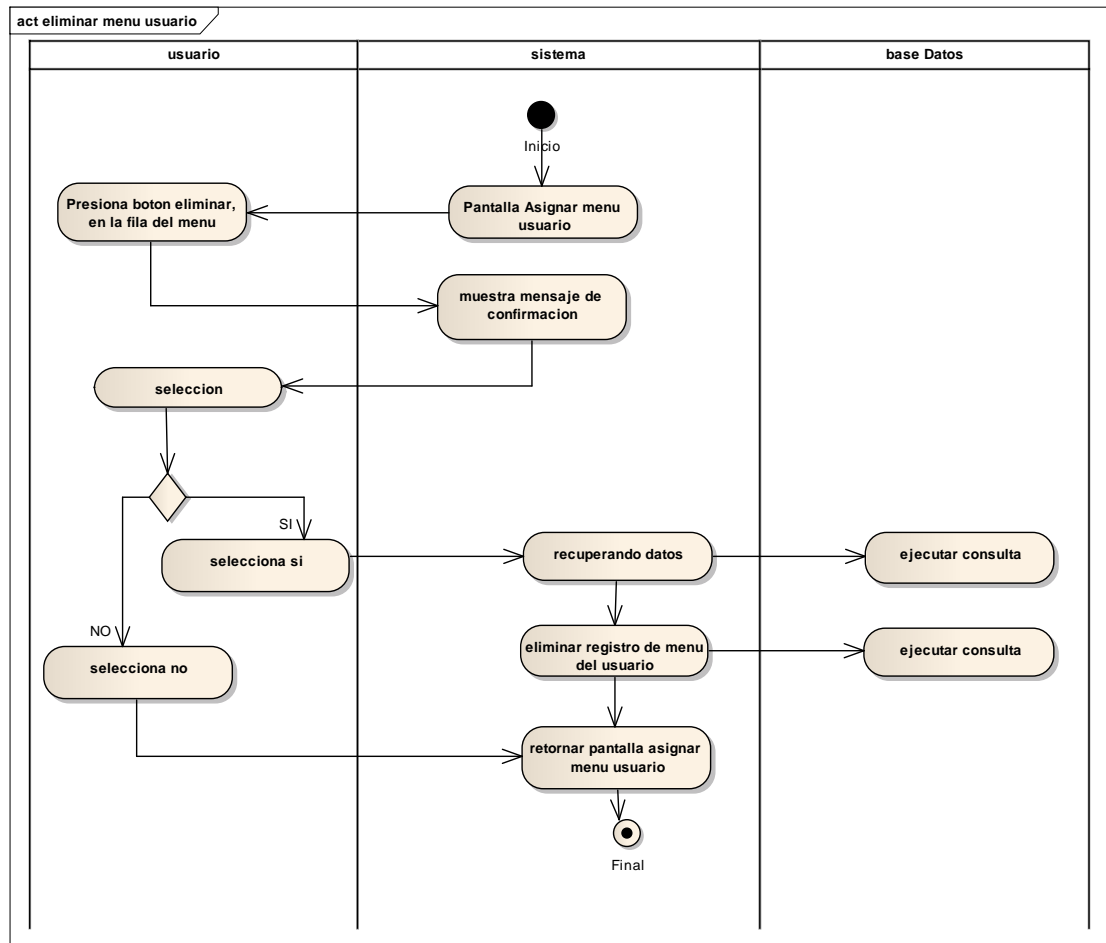


Fig. 104 Diagrama de Actividades Caso de uso Eliminar Menús – Usuario

Diagrama de Actividades Caso de uso Administrar Menús

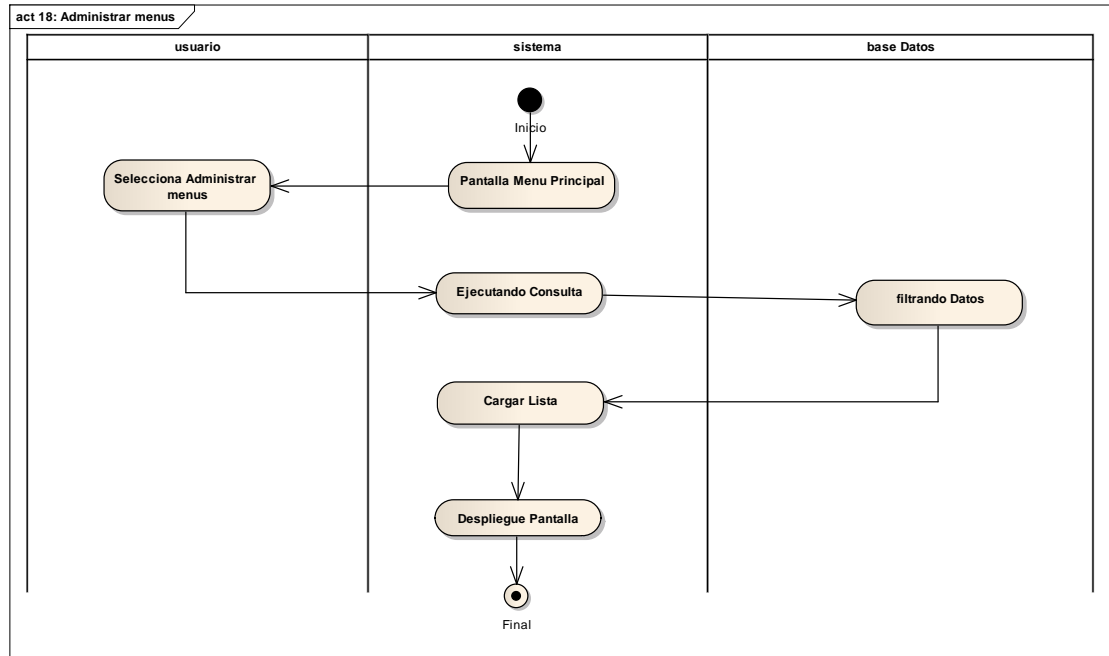


Fig. 105 Diagrama de Actividades Caso de uso Administrar Menús

Diagrama de Actividades Caso de uso Adicionar Menú

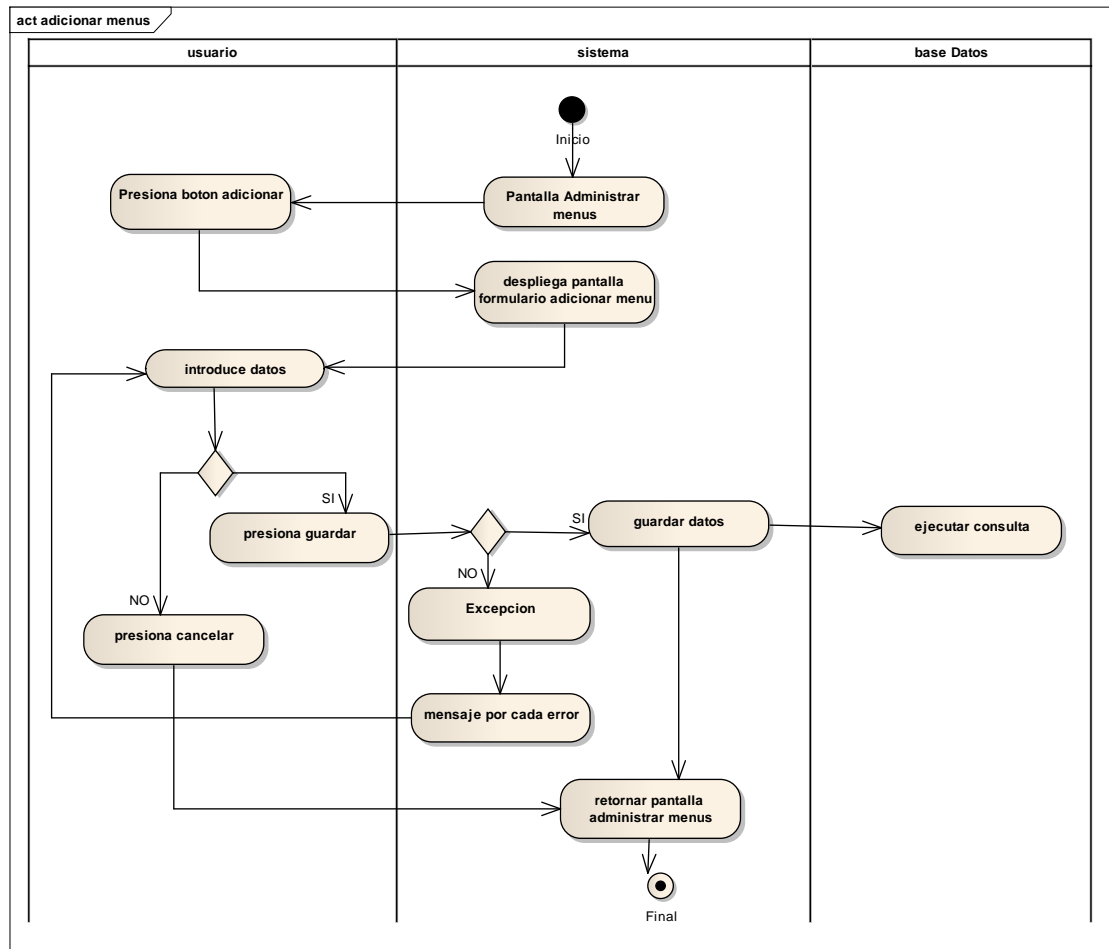


Fig. 106 Diagrama de Actividades Caso de uso Adicionar Menú

Diagrama de Actividades Caso de uso Modificar Menú

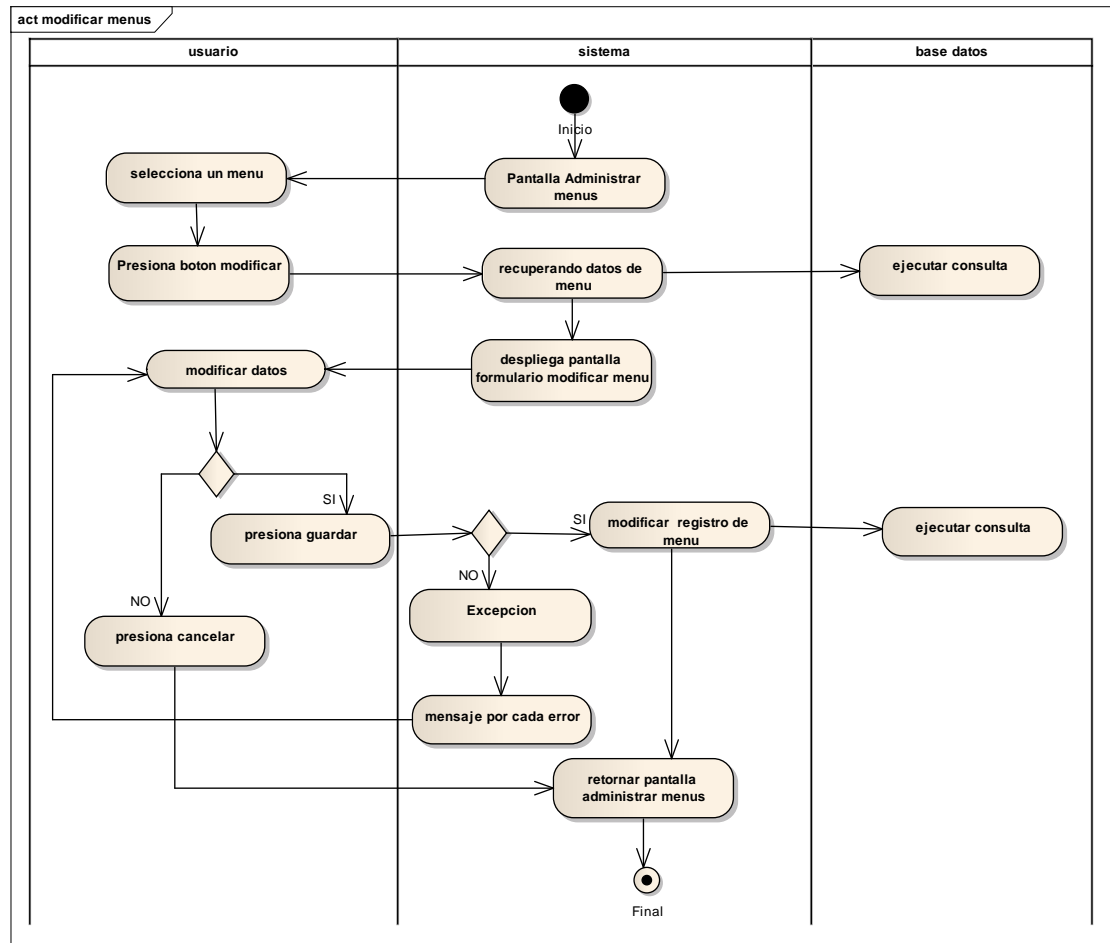


Fig. 107 Diagrama de Actividades Caso de uso Modificar Menú

Diagrama de Actividades Caso de uso Eliminar Menú

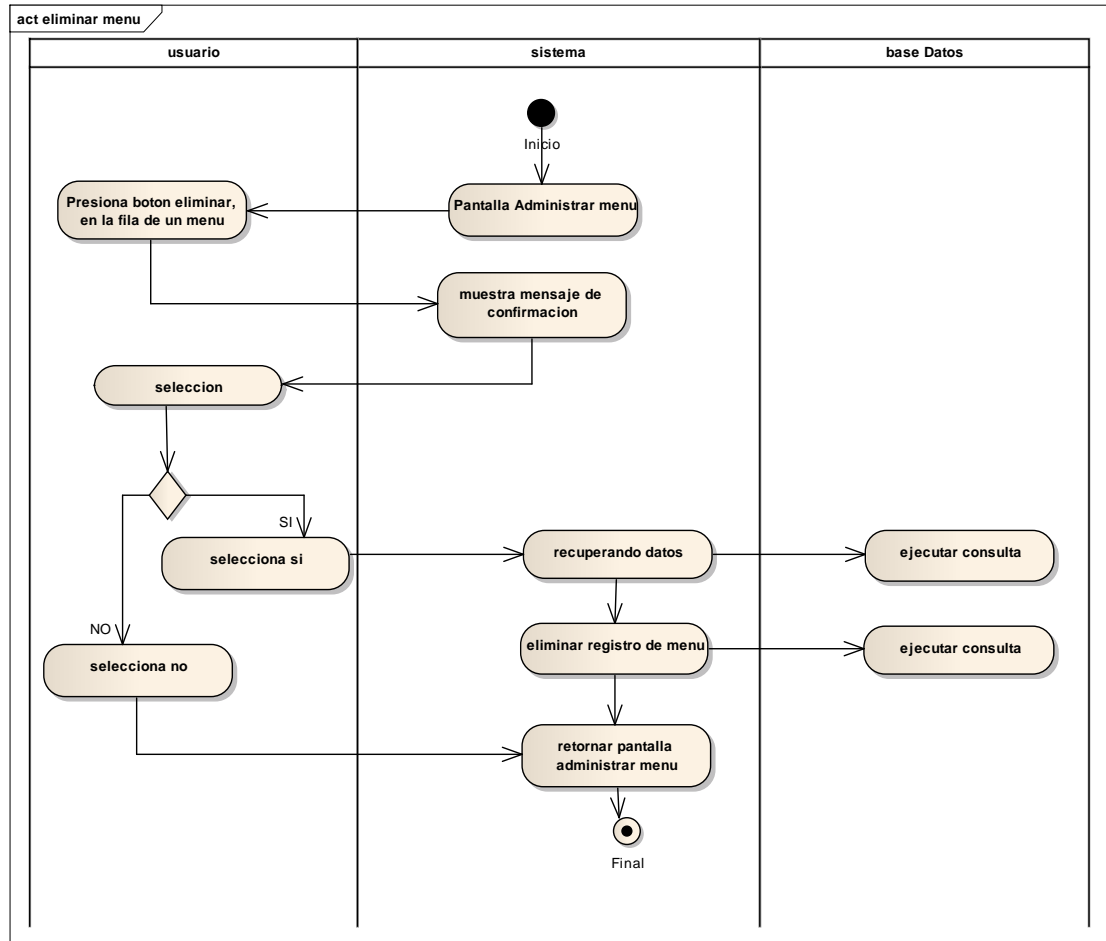


Fig. 108 Diagrama de Actividades Caso de uso Eliminar Menú

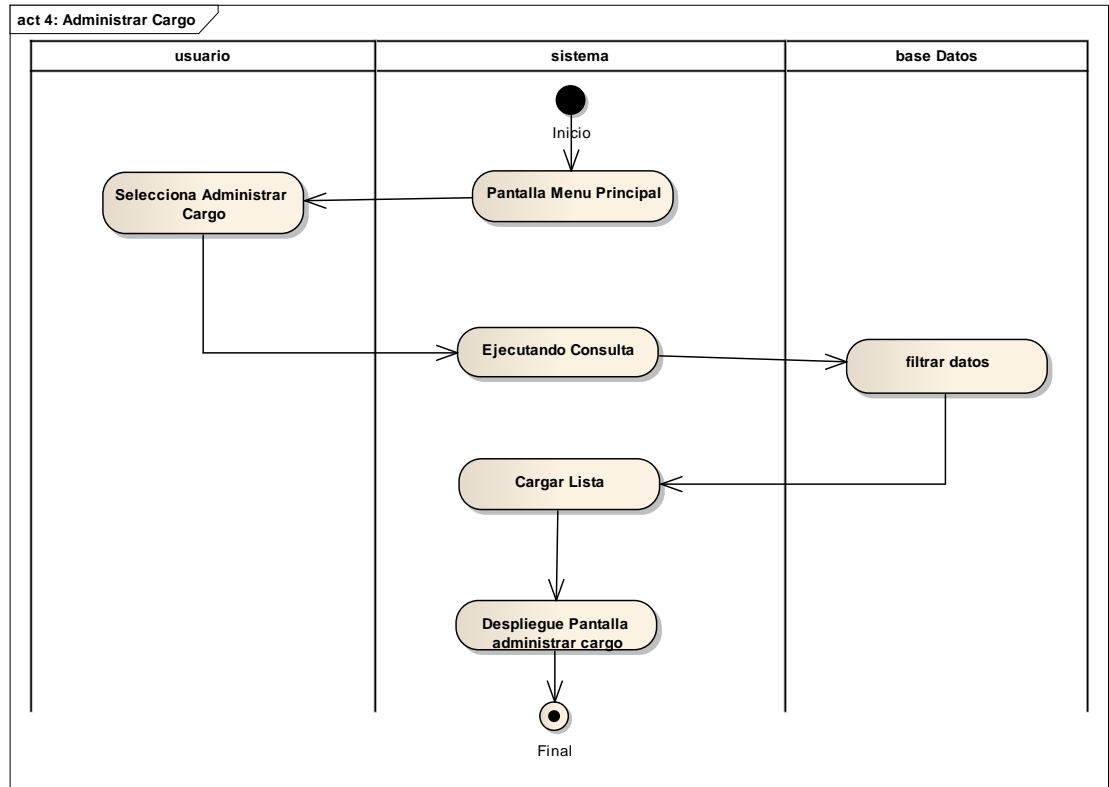
Diagrama de Actividades Caso de uso Administrar Cargos

Fig. 109 Diagrama de Actividades Caso de uso Administrar Cargos

Diagrama de Actividades Caso de uso Adicionar Cargo

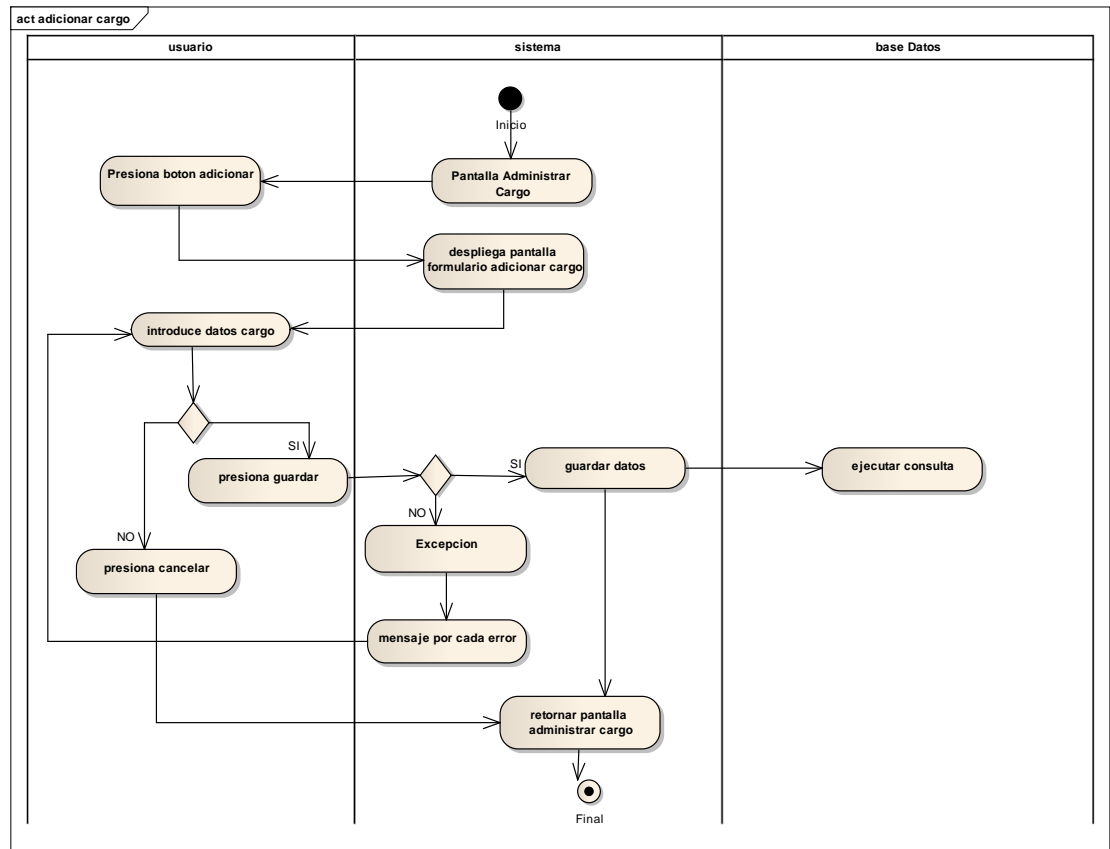


Fig. 110 Diagrama de Actividades Caso de uso Adicionar Cargo

Diagrama de Actividades Caso de uso Modificar Cargo

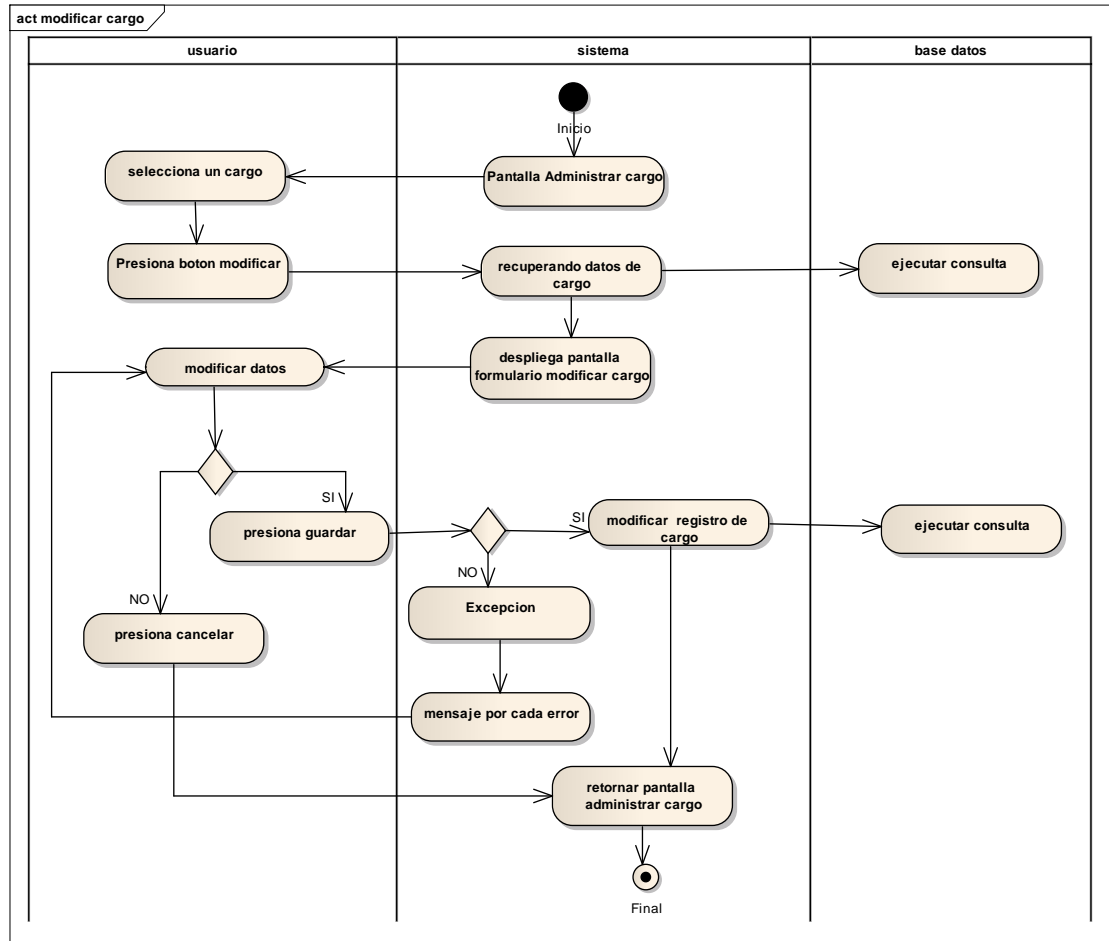


Fig. 111 Diagrama de Actividades Caso de uso Modificar Cargo

Diagrama de Actividades Caso de uso Eliminar Cargo

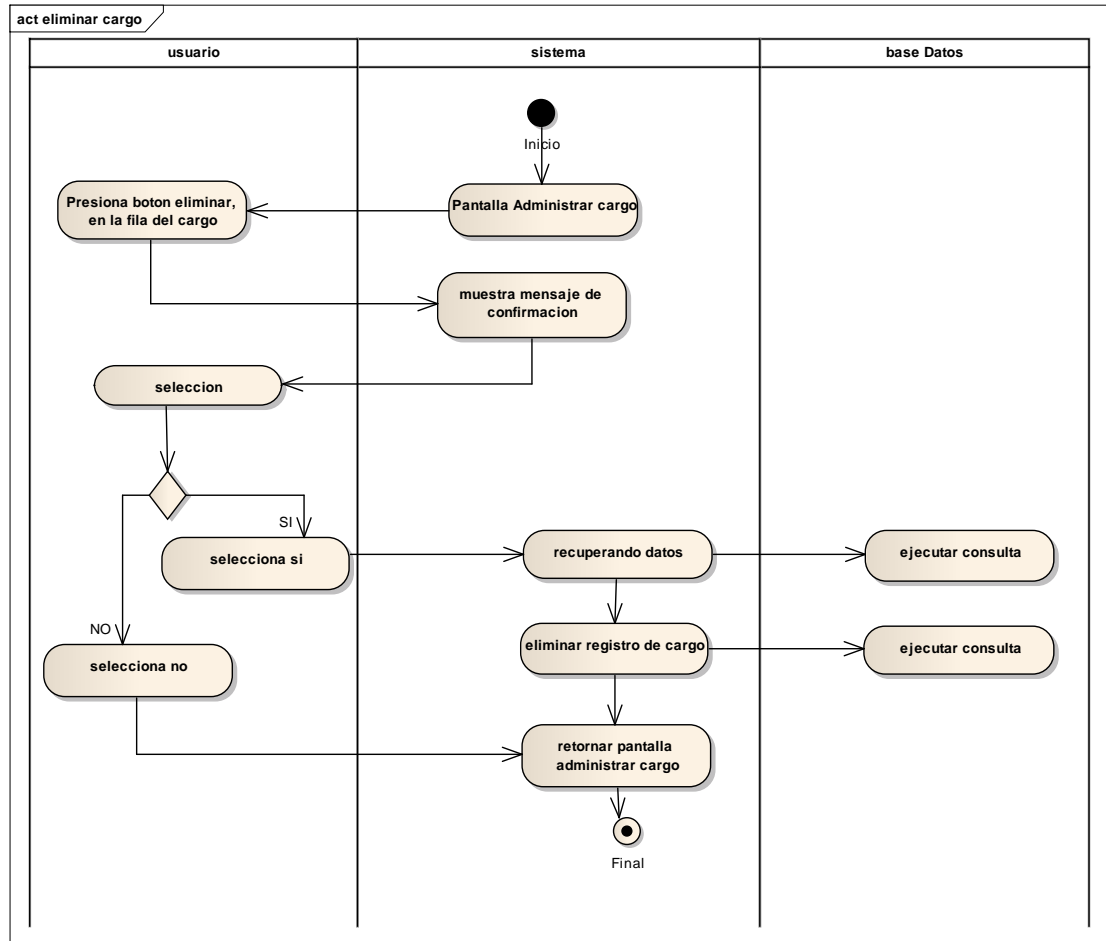


Fig. 112 Diagrama de Actividades Caso de uso Eliminar Cargo

Diagrama de Actividades Caso de uso Generar Reportes

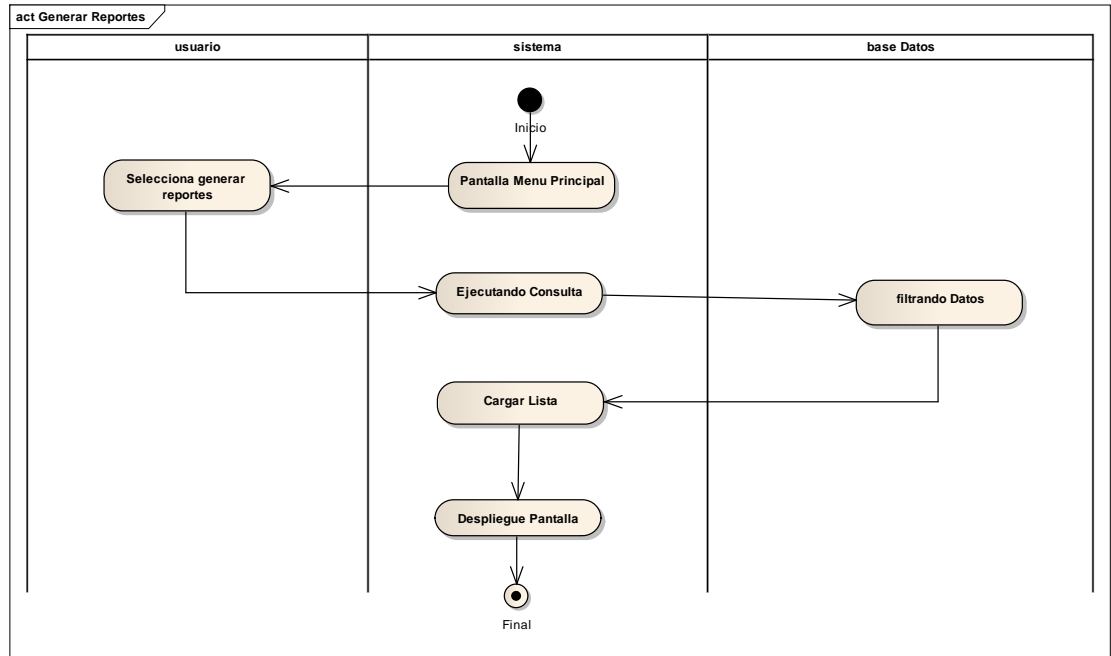


Fig. 113 Diagrama de Actividades Caso de uso Generar Reportes

Diagrama de Actividades Caso de uso Reporte Tamizaje Clínico

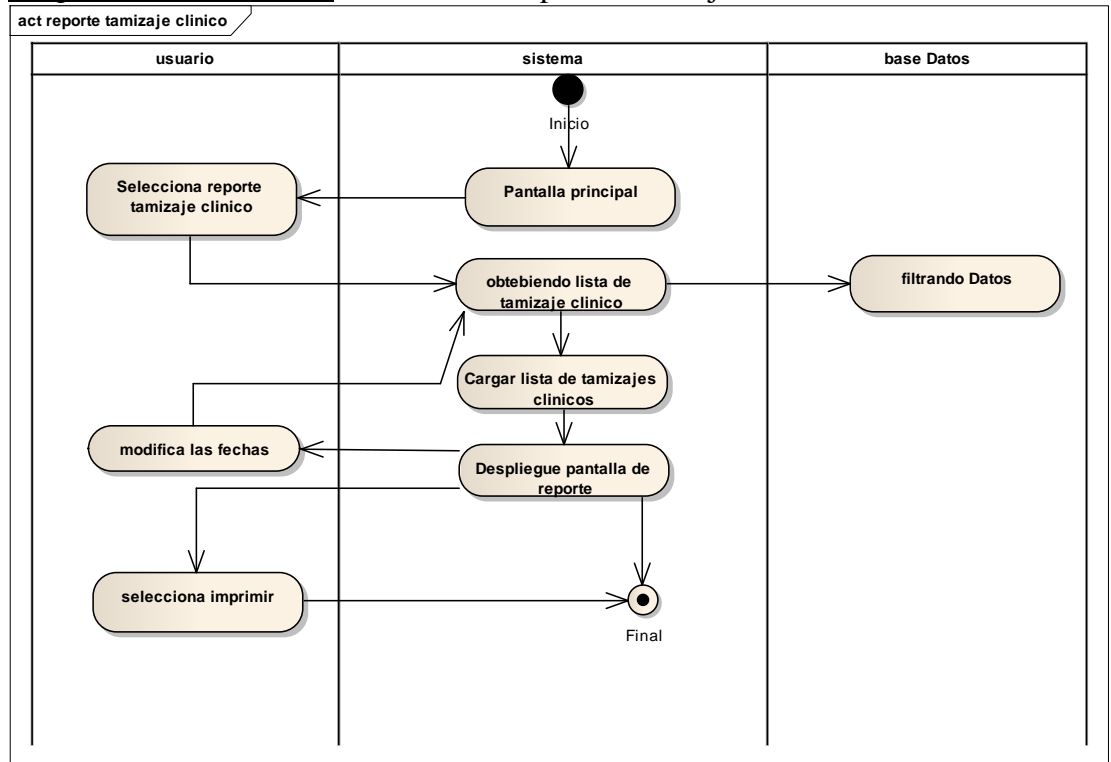


Fig. 114 Diagrama de Actividades Caso de uso Reporte Tamizaje Clínico

Diagrama de Actividades Caso de uso Reporte Tamizaje Laboratorial

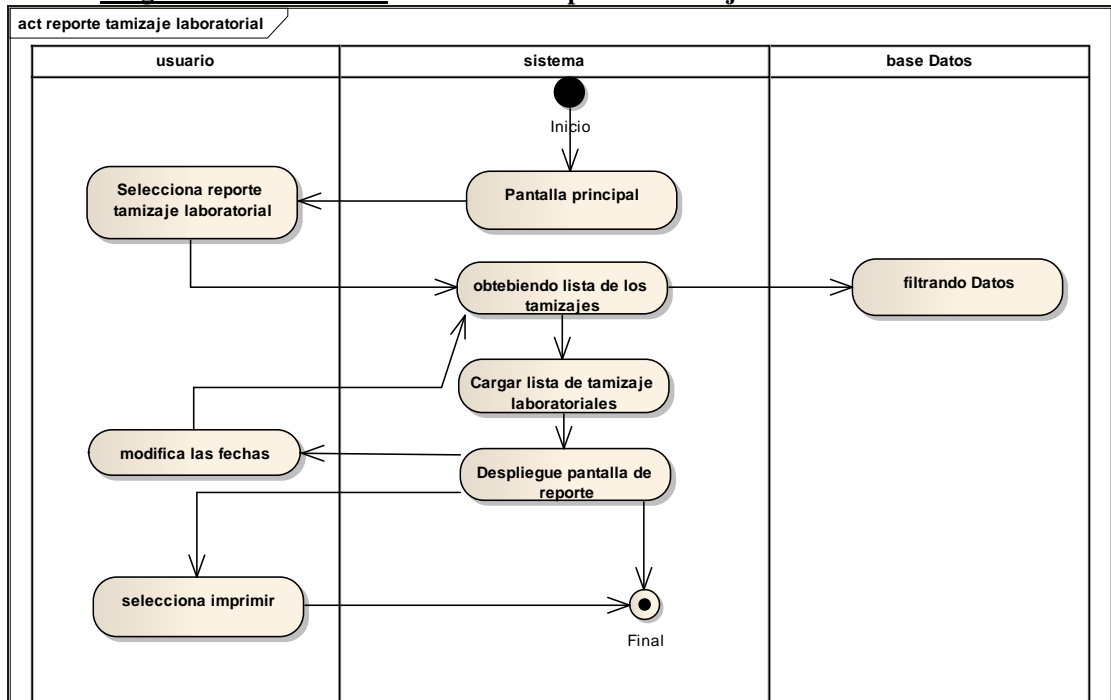


Fig. 115 Diagrama de Actividades Caso de uso Reporte Tamizaje Laboratorial

Diagrama de Actividades Caso de uso Reporte Extracciones

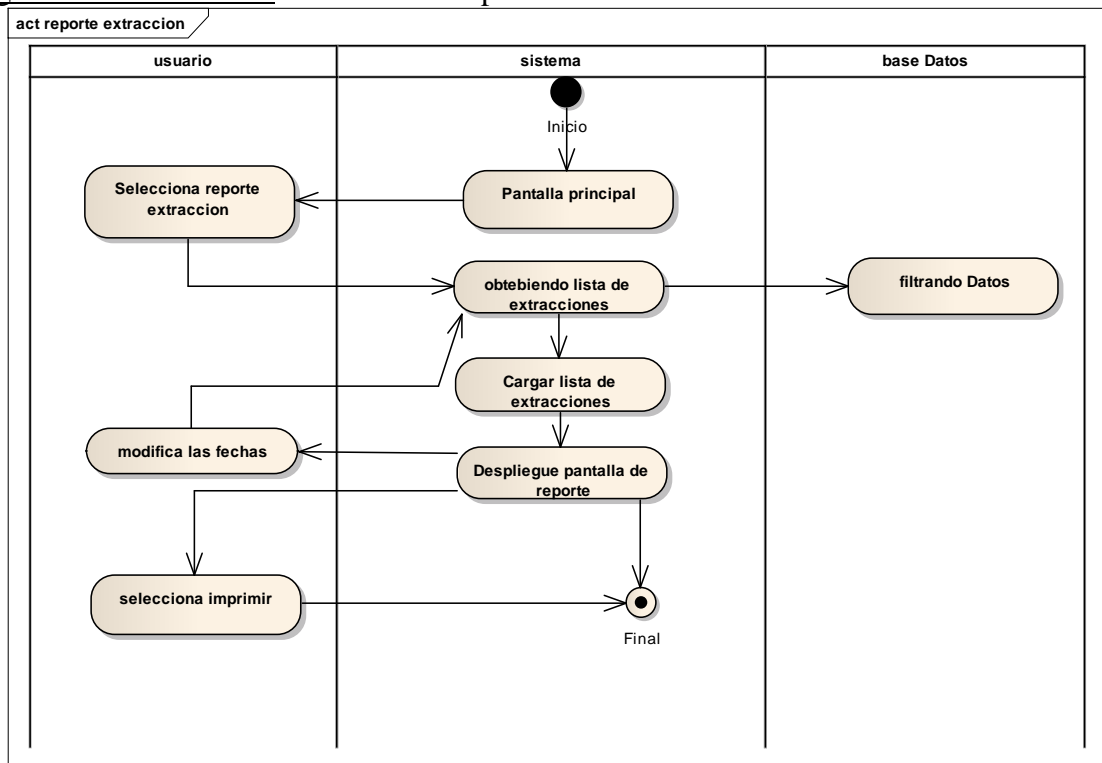


Fig. 116 Diagrama de Actividades Caso de uso Reporte Extracciones

Diagrama de Actividades Caso de uso Reporte Rechazo

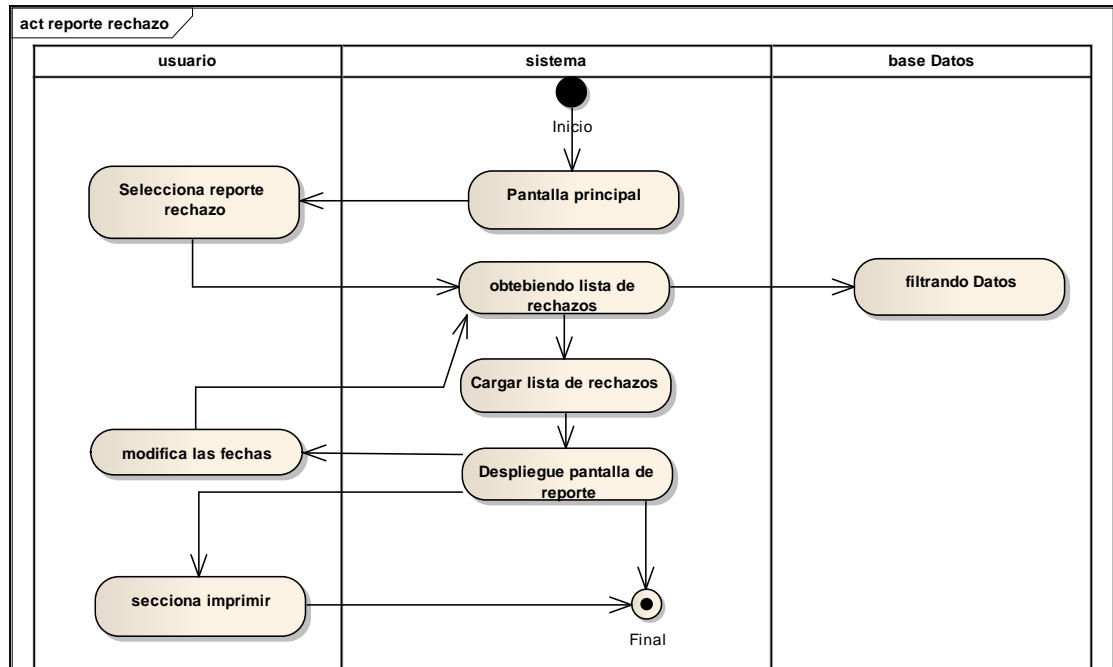


Fig. 117 Diagrama de Actividades Caso de uso Reporte Rechazo

Diagrama de Actividades Caso de uso Reporte Estadísticos

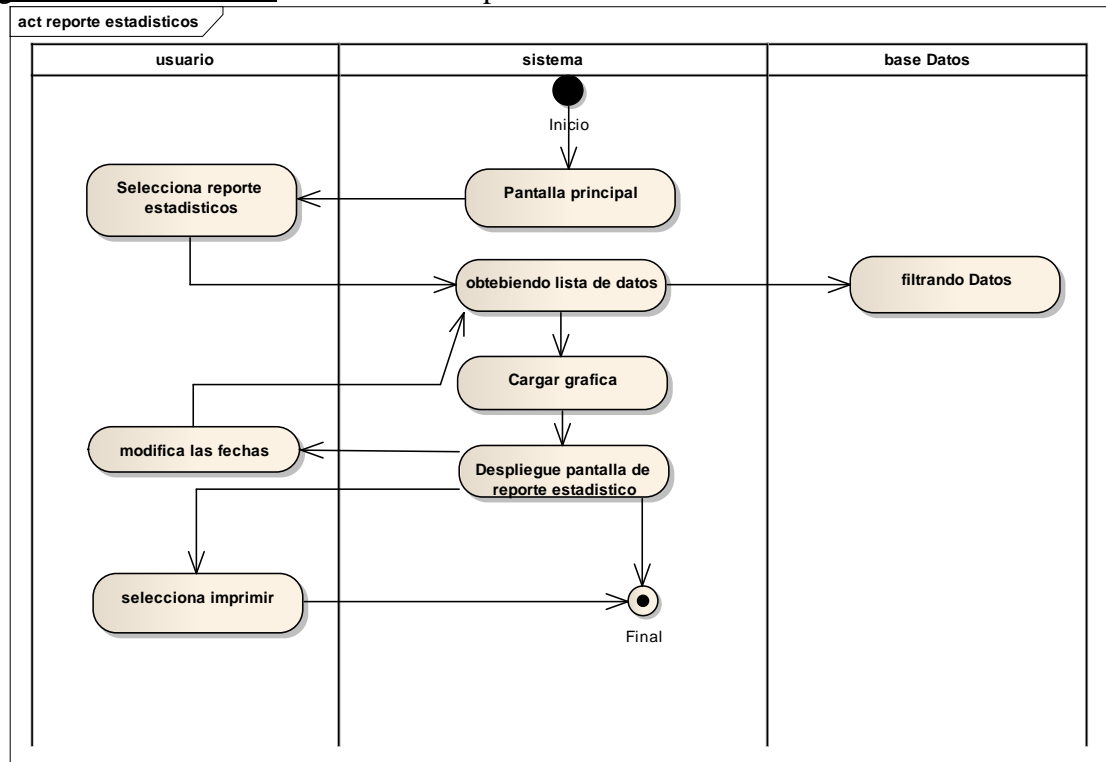


Fig. 118 Diagrama de Actividades Caso de uso Reporte Estadísticos

Diagrama de Actividades Caso de uso Reporte Habilitados- Inhabilitados para Donar

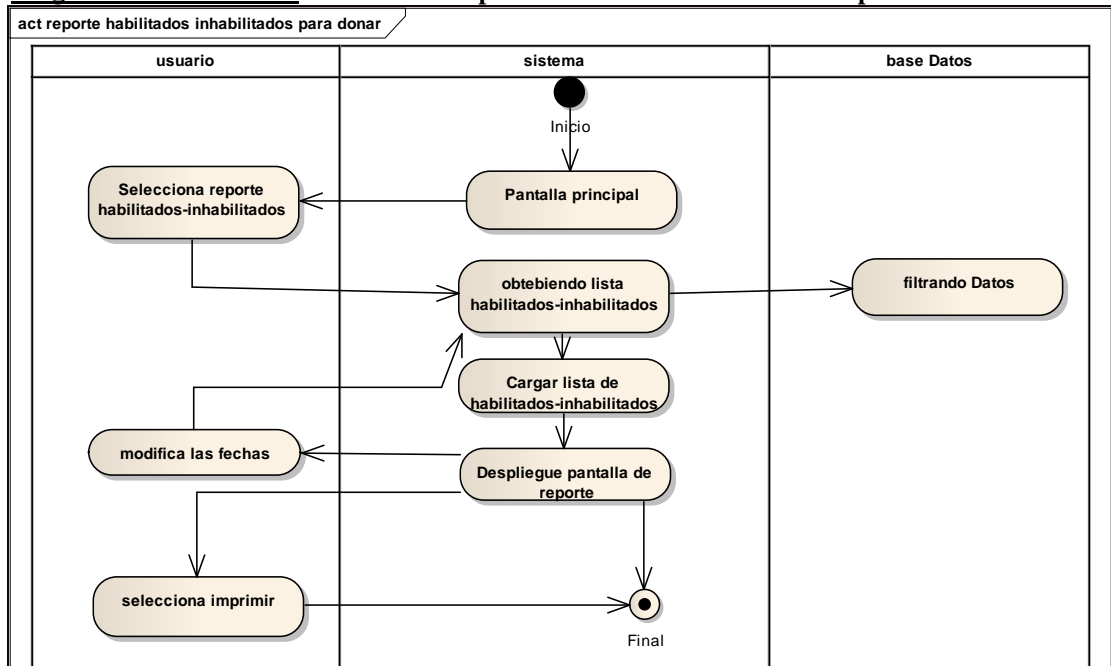


Fig. 119 Diagrama de Actividades Caso de uso Reporte Habilitados- Inhabilitados para Donar

Diagrama de Actividades Caso de uso Reporte Donantes Aptos – No Aptos

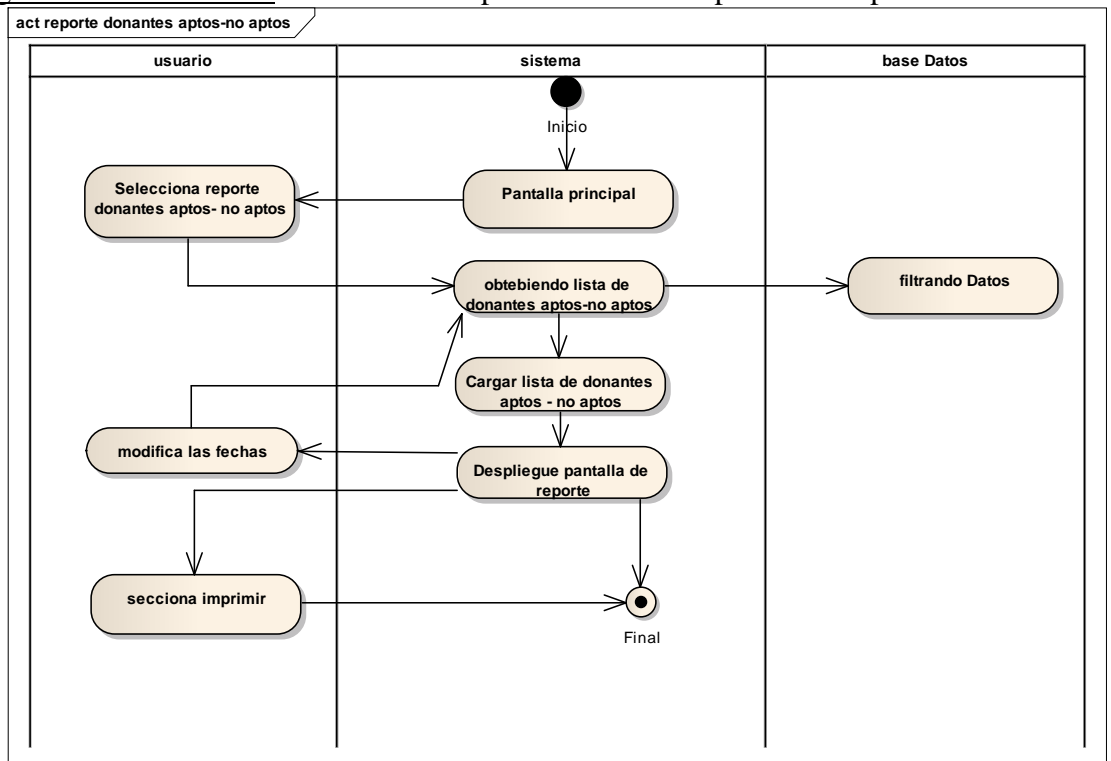


Fig. 120 Diagrama de Actividades Caso de uso Reporte Donantes Aptos – No Aptos

II.1.5. MODELADO DE DIAGRAMA DE INTERACCIÓN

II.1.5.1. Introducción

Los diagramas de interacción consisten en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

II.1.5.1.1. Propósito

- Comprender la dinámica del sistema deseado para la organización
- Identificar clases de análisis y diseño

II.1.5.1.2. Alcance

- Describir la dinámica de sistema en el tiempo de vida de las clases u objetos
- Definir un *diagrama de secuencia* para cada caso de uso del usuario.

II.1.5.2. Diagramas de Secuencias

Diagrama de Secuencias Acceso al sistema.

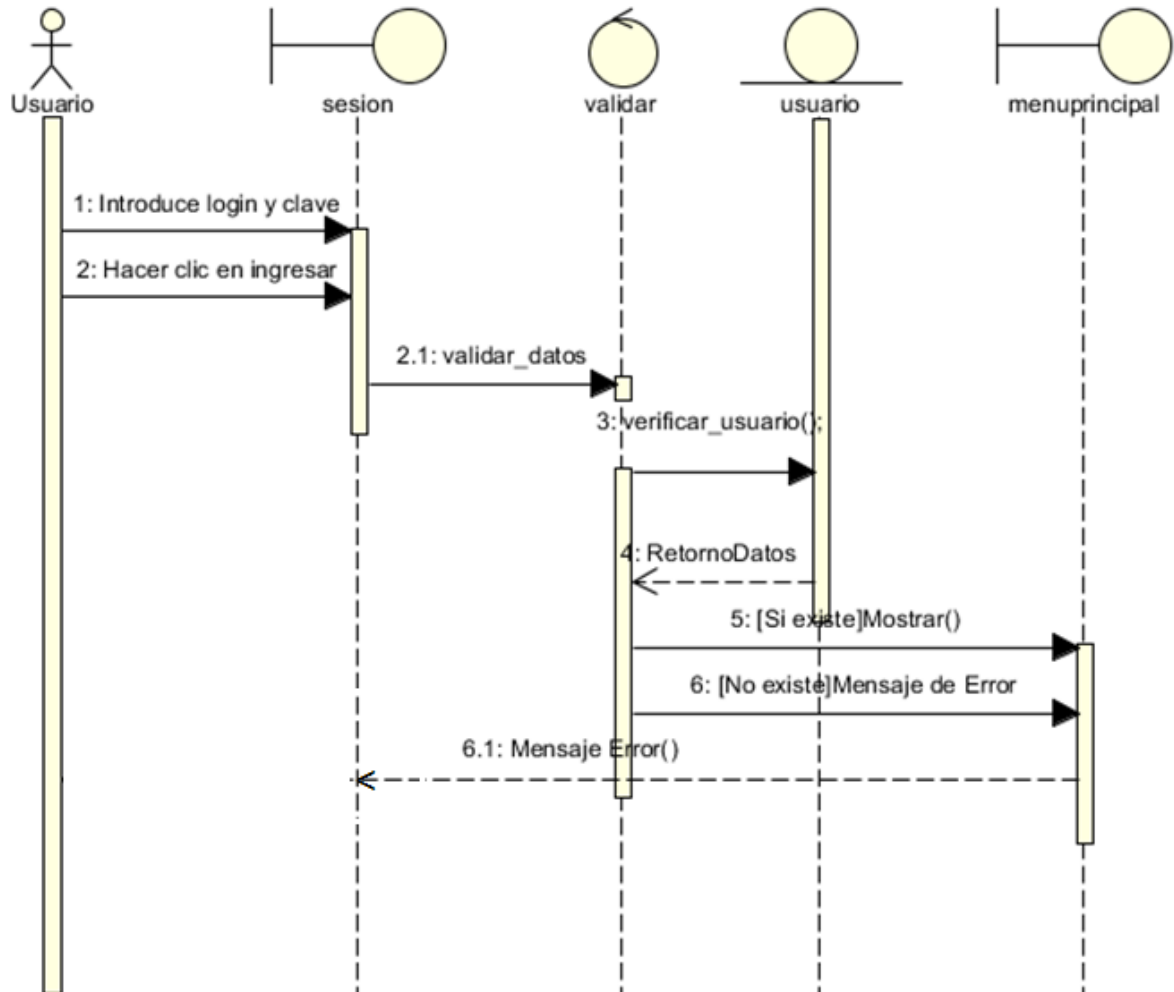


Fig. 121 Diagrama de Secuencias Acceso al sistema.

Diagrama de Secuencias Administrar Donantes

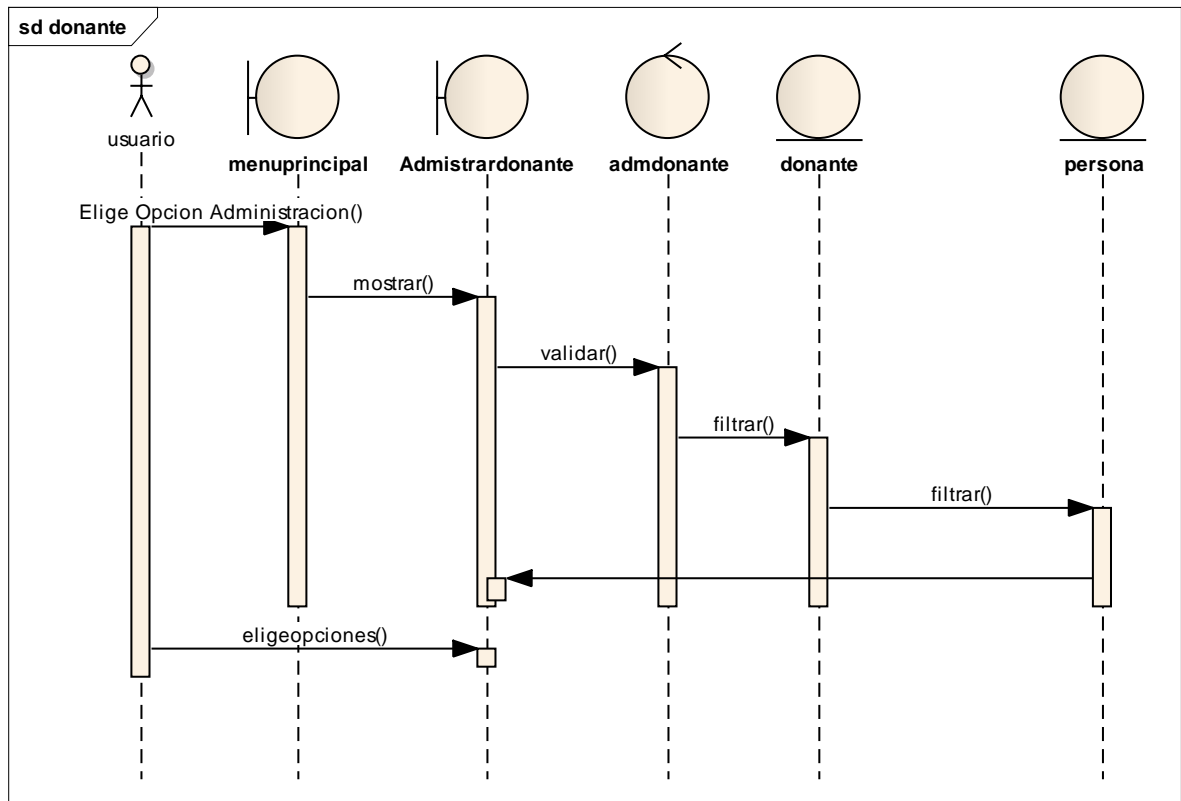


Fig. 122 Diagrama de Secuencias Administrar Donantes

Diagrama de Secuencias Adicionar Donante

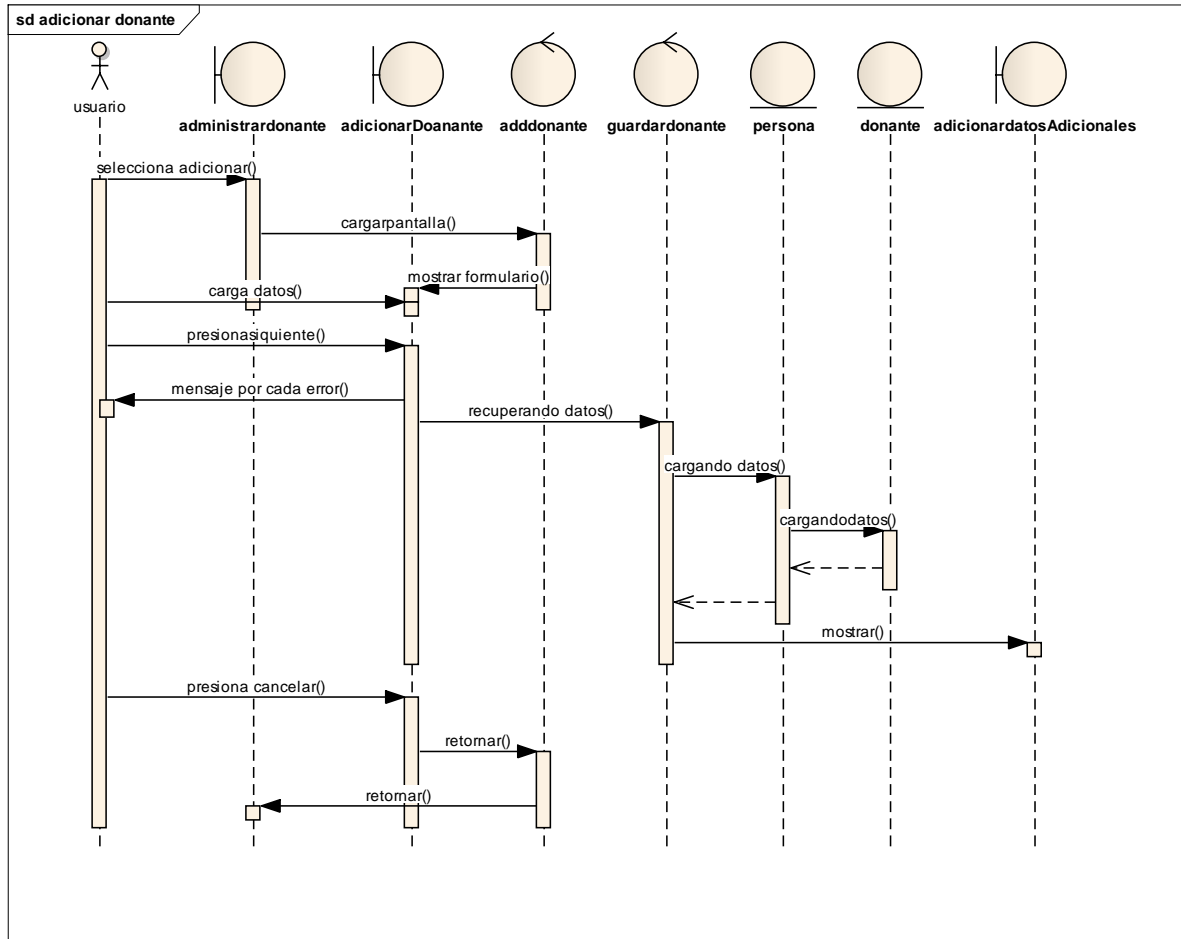


Fig. 123 Diagrama de Secuencias Adicionar Donante

Diagrama de Secuencias Adicionar Datos Adicionales

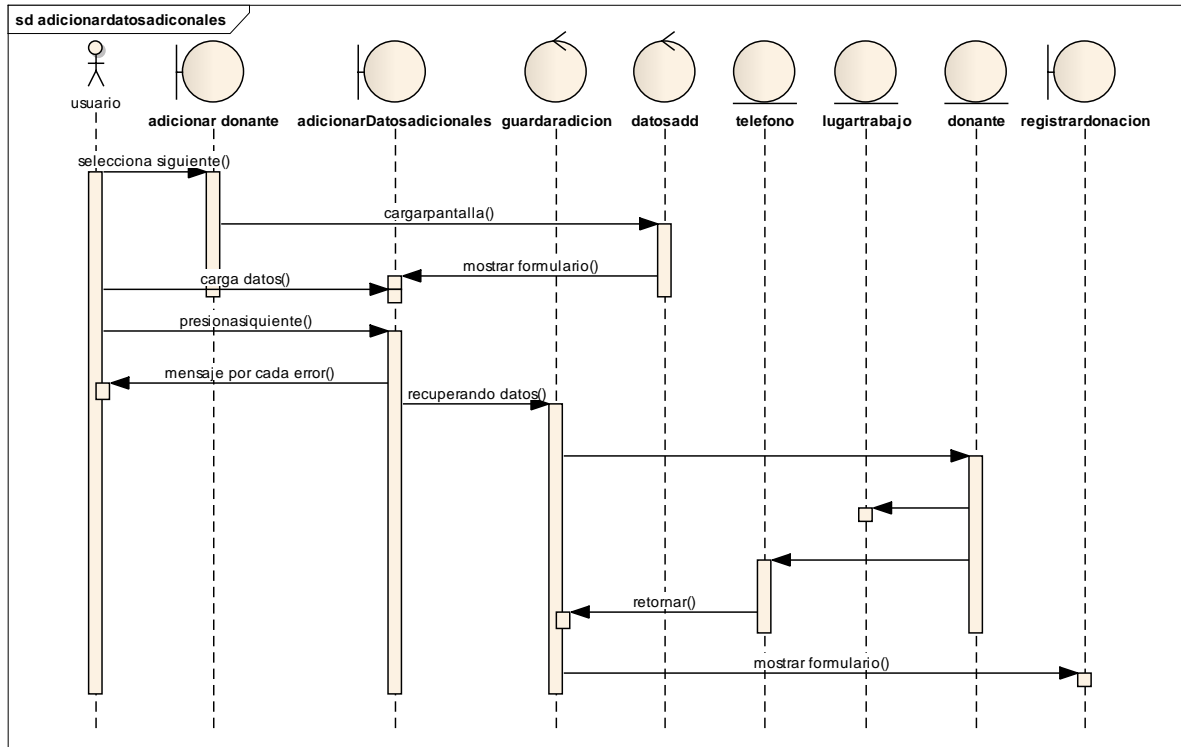


Fig. 124 Diagrama de Secuencias Adicionar Datos Adicionales

Diagrama de Secuencias Registrar Donación

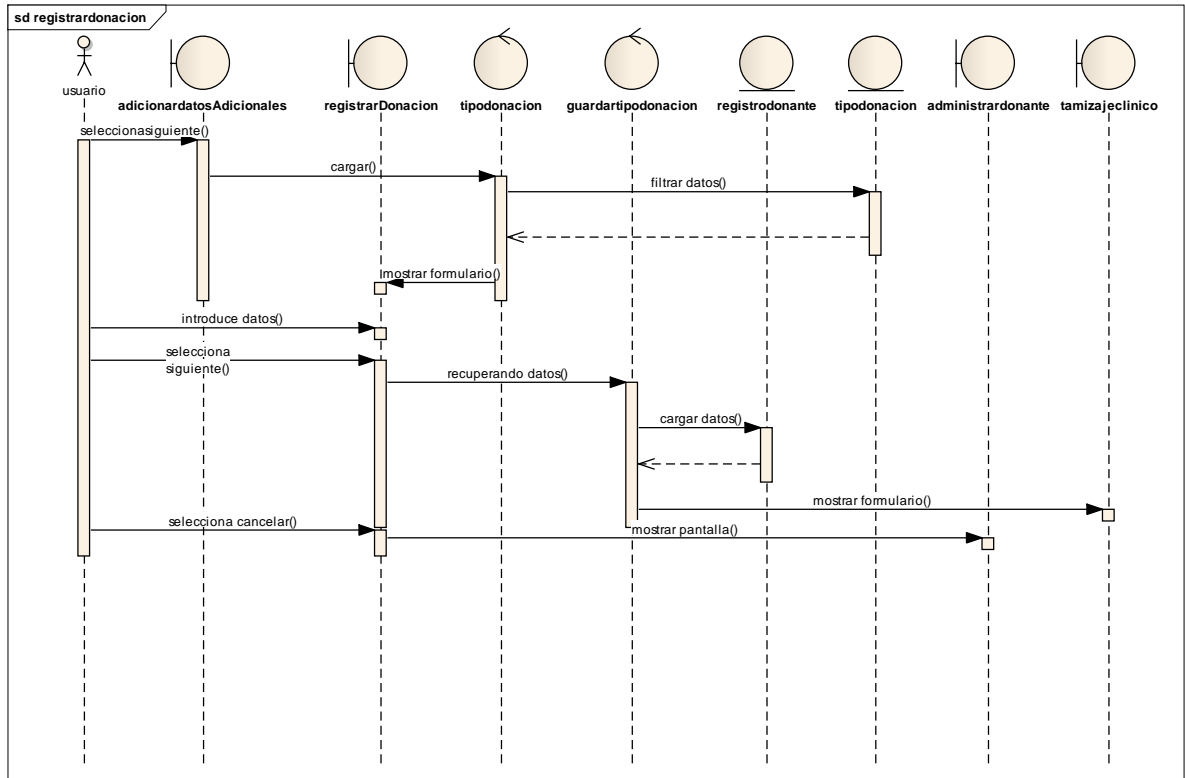


Fig. 125 Diagrama de Secuencias Registrar Donación

Diagrama de Secuencias Registrar Tamizaje Clínico

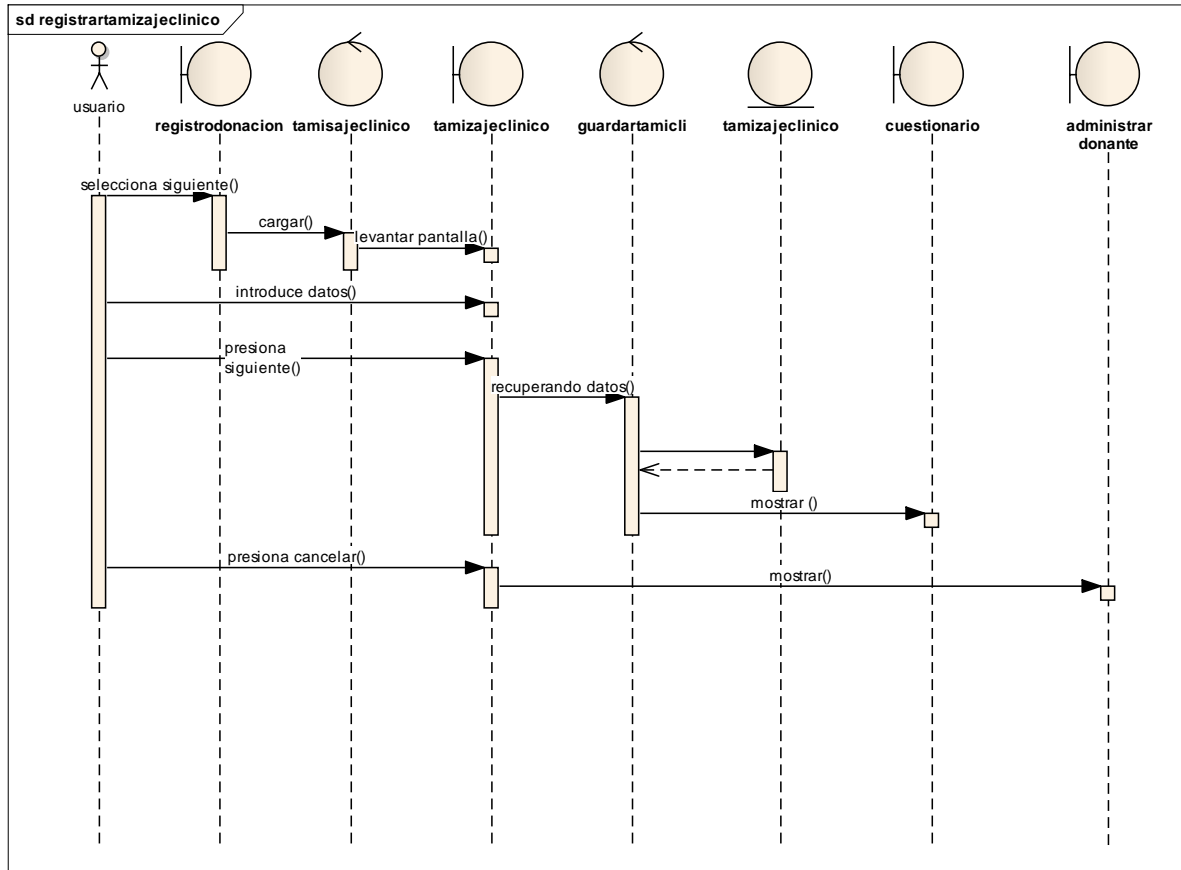


Fig. 126 Diagrama de Secuencias Registrar Tamizaje Clínico

Diagrama de Secuencias Administrar Tamizaje Clínico

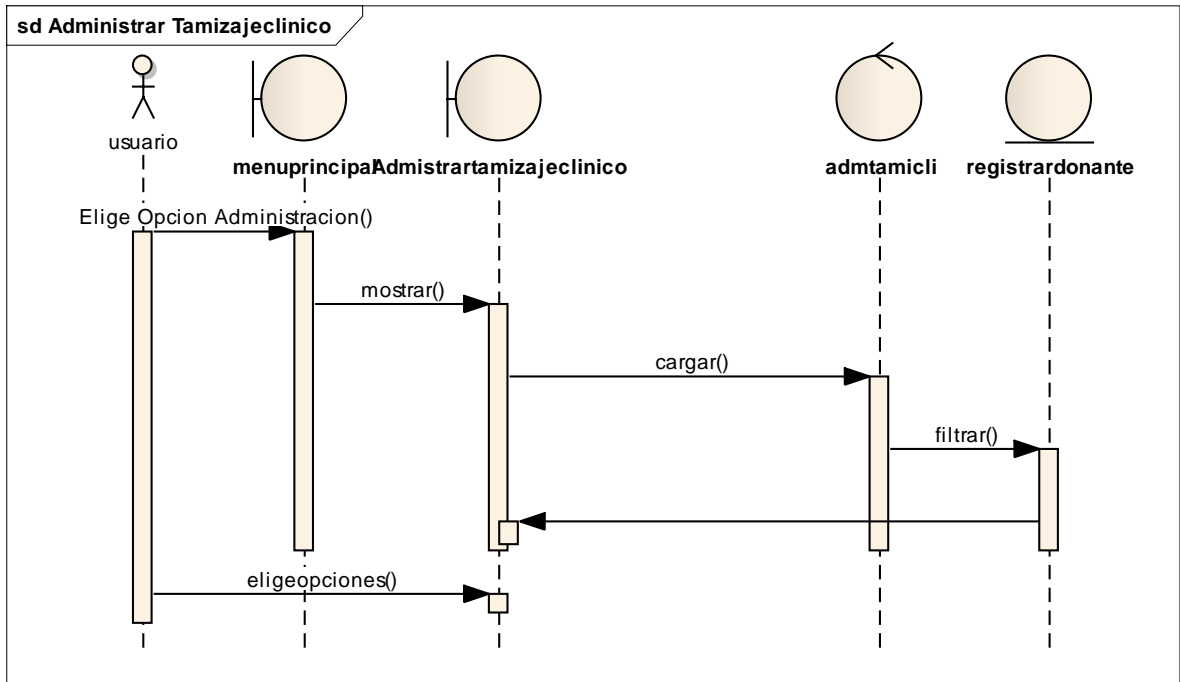


Fig. 127 Diagrama de Secuencias Administrar Tamizaje Clínico

Diagrama de Secuencias Administrar Tamizaje Laboratorial

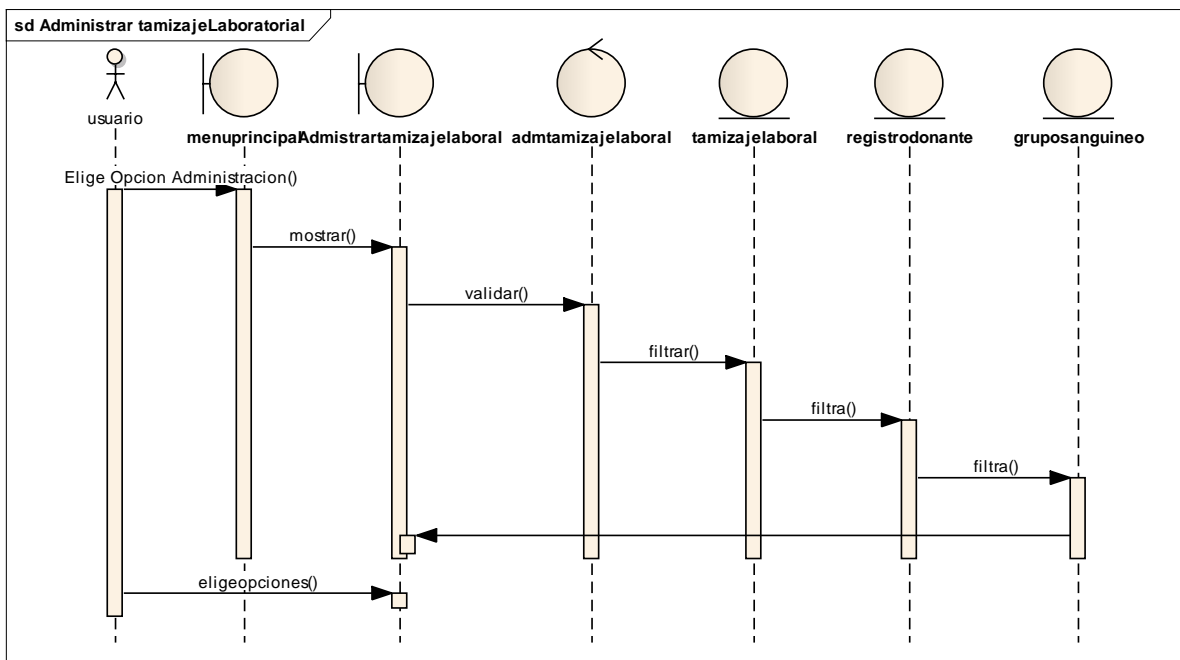


Fig. 128 Diagrama de Secuencias Administrar Tamizaje Laboratorial

Diagrama de Secuencias Administrar Extracción

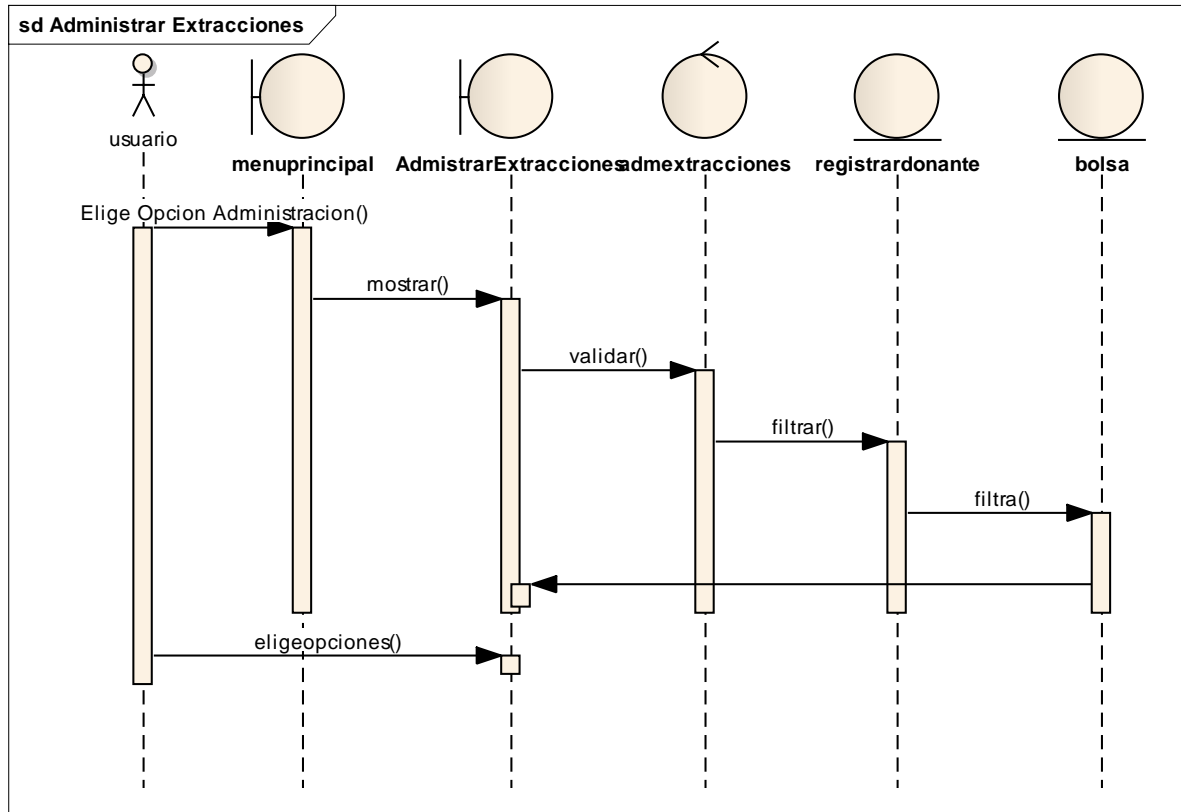


Fig. 129 Diagrama de Secuencias Administrar Extracción

Diagrama de Secuencias Administrar Donaciones

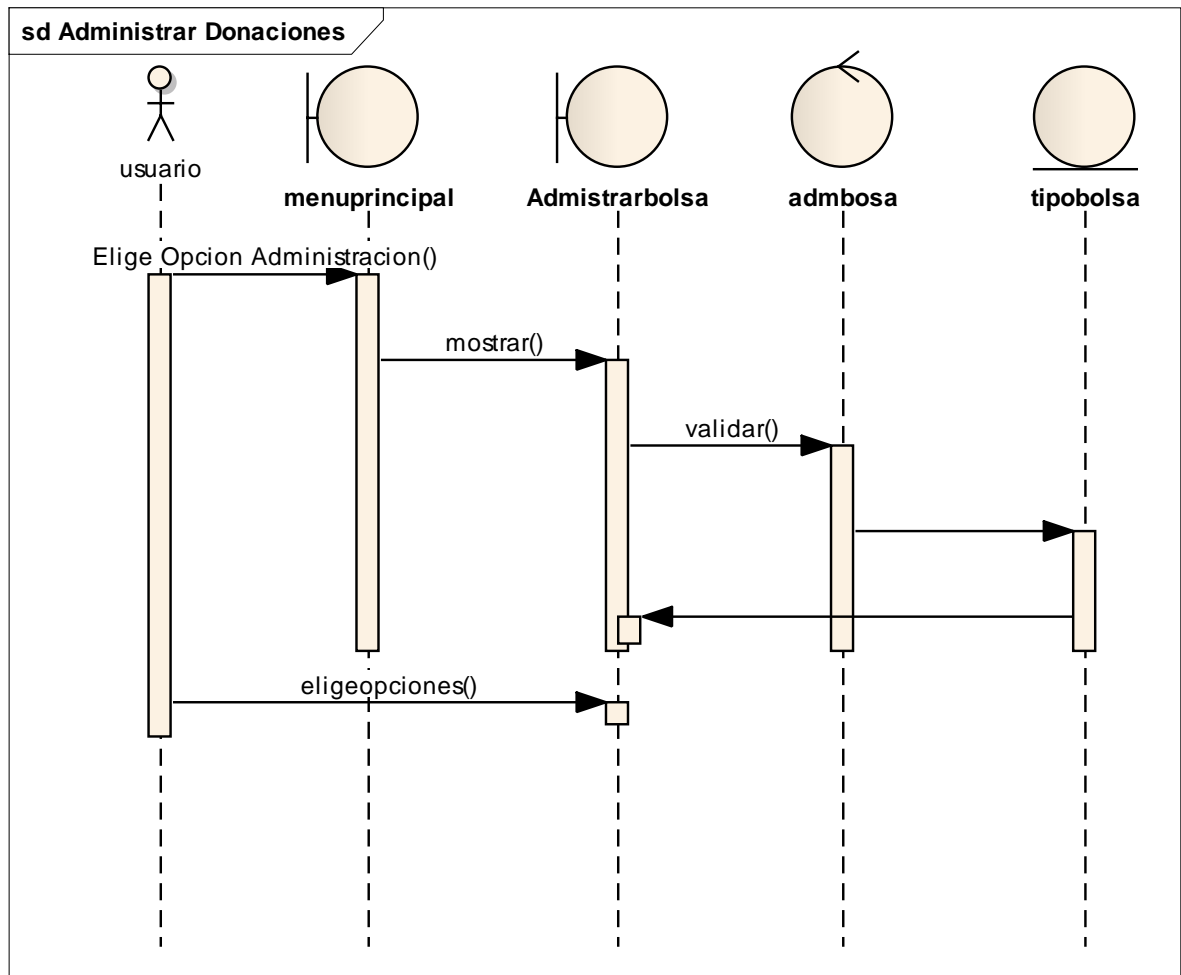


Fig. 130 Diagrama de Secuencias Administrar Donaciones

Diagrama de Secuencias Administrar Rechazo

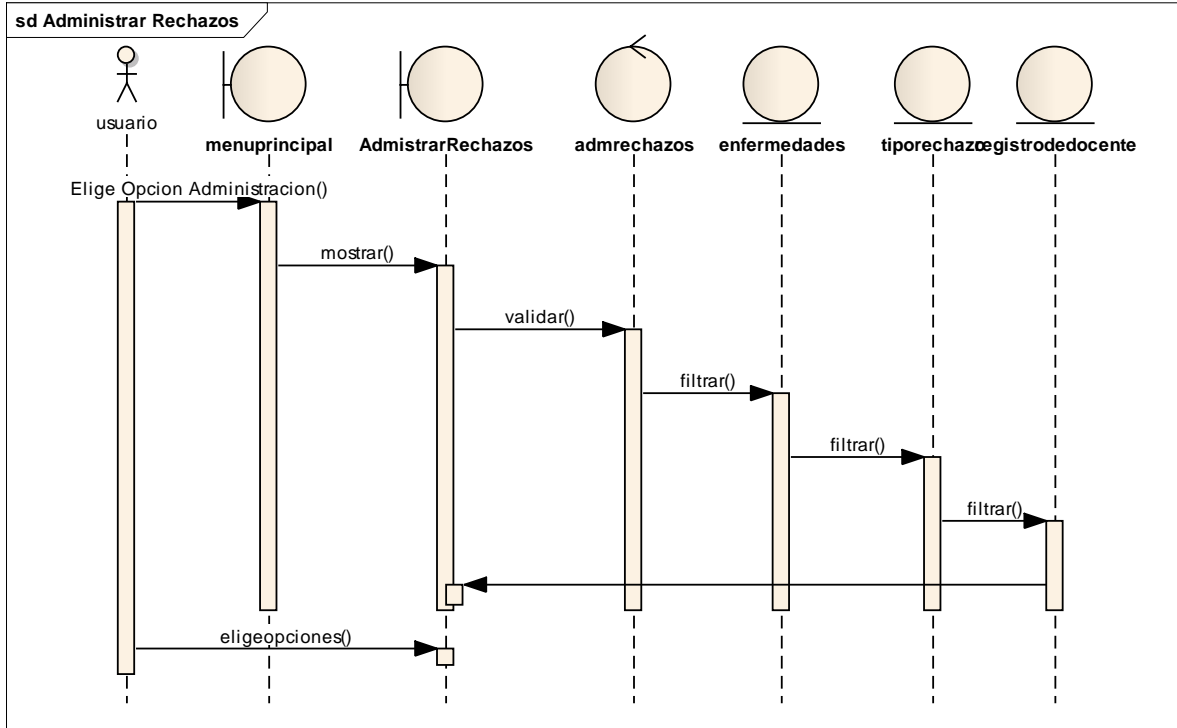


Fig. 131 Diagrama de Secuencias Administrar Rechazo

Diagrama de Secuencias Administrar Pregunta

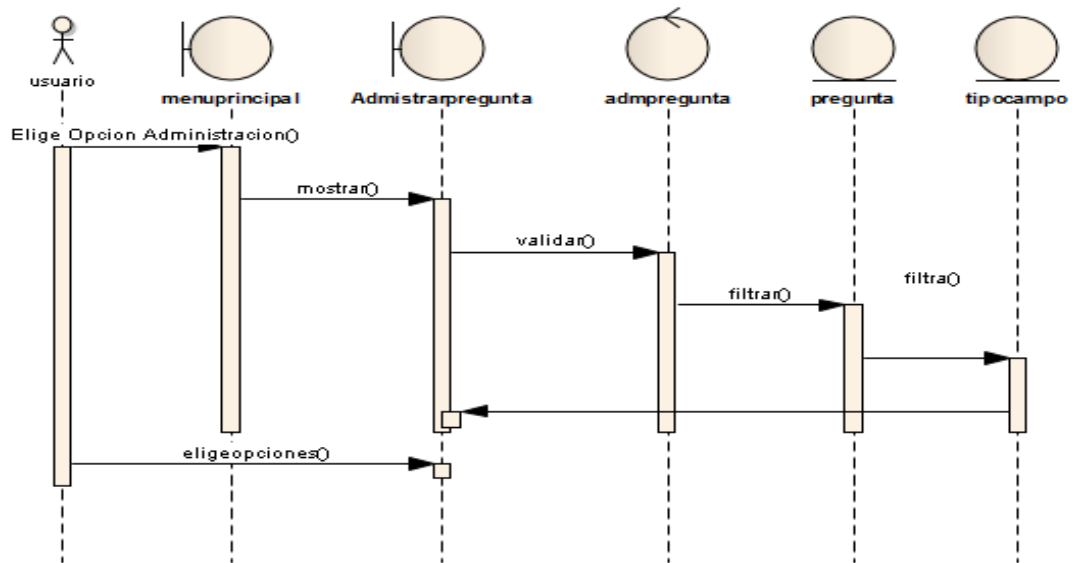


Fig. 132 Diagrama de Secuencias Administrar Pregunta

Diagrama de Secuencias Administrar Tipo Bolsa

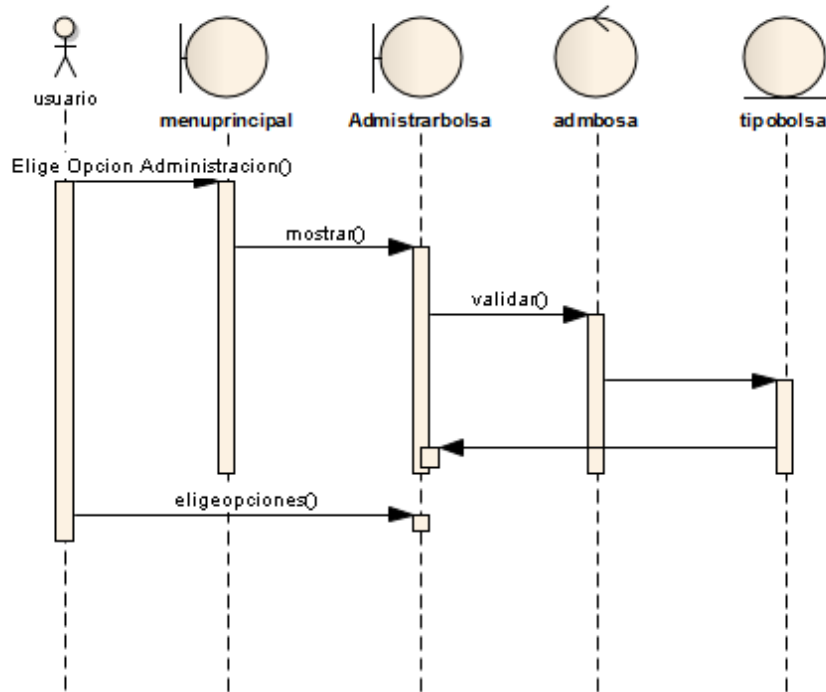


Fig. 133 Diagrama de Secuencias Administrar Tipo Bolsa

Diagrama de Secuencias Administrar Tipo Donación

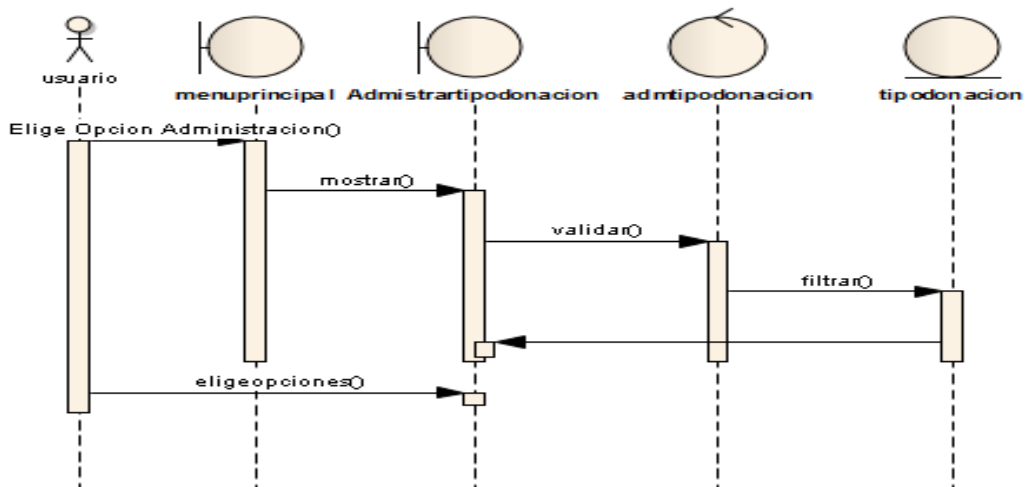


Fig. 134 Diagrama de Secuencias Administrar Tipo Donación

Diagrama de Secuencias Administrar Lugar de Donación

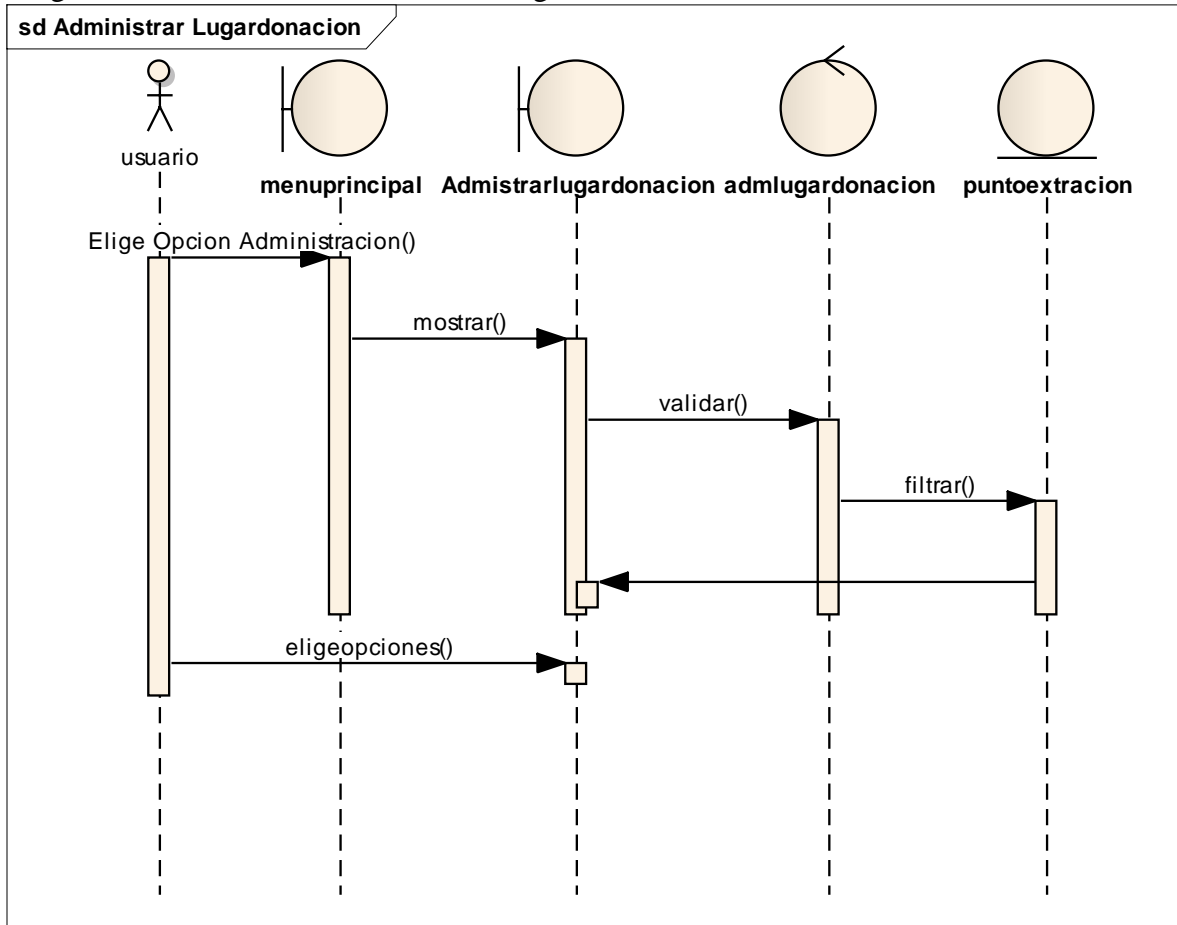


Fig. 135 Diagrama de Secuencias Administrar Lugar de Donación

Diagrama de Secuencias Administrar Departamentos

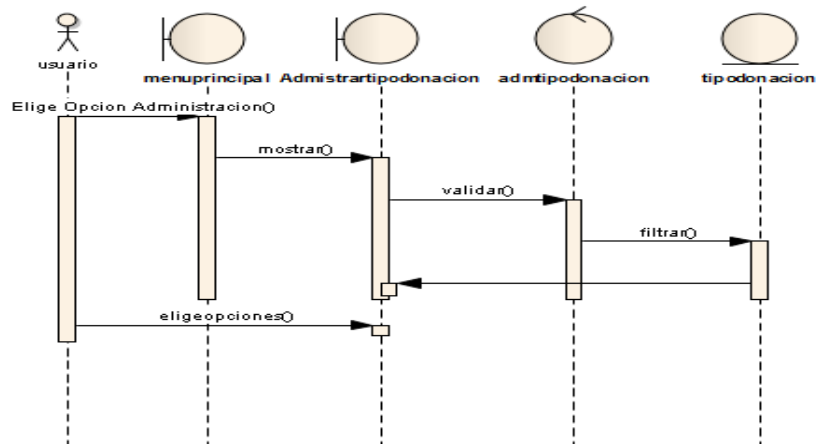


Fig. 136 Diagrama de Secuencias Administrar Departamentos

Diagrama de Secuencias Administrar Provincia

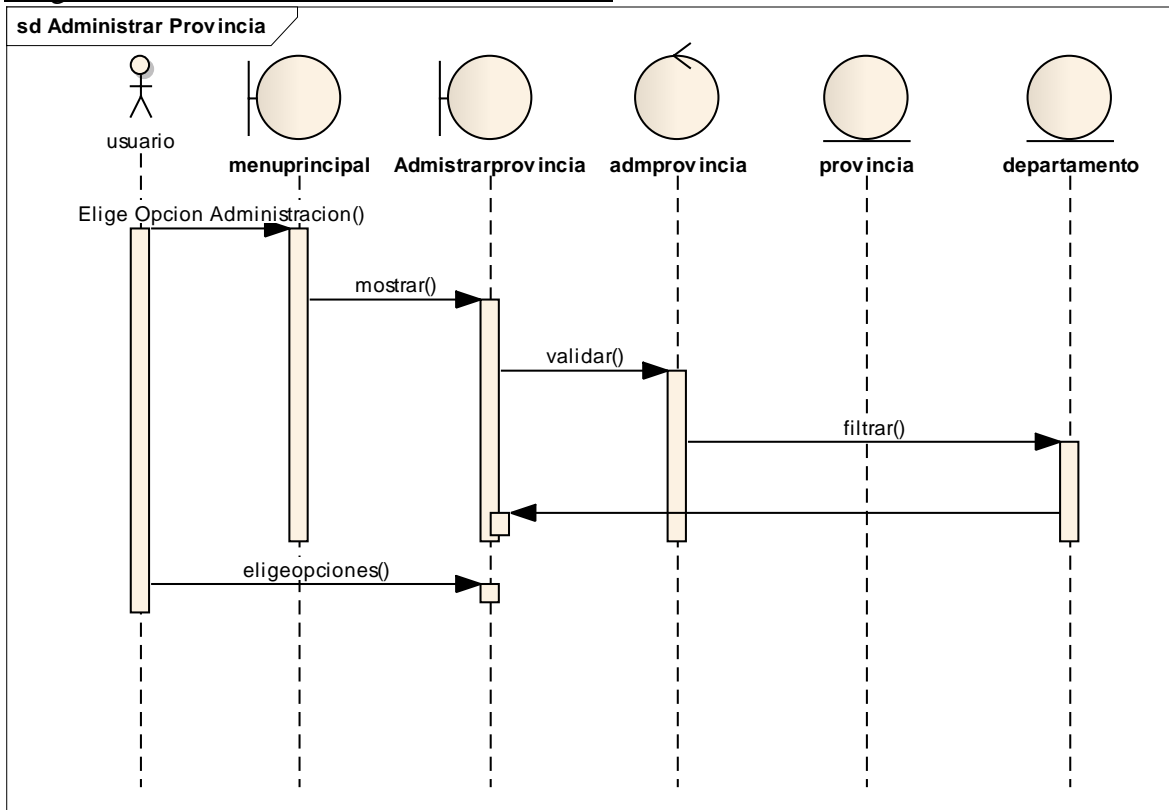


Fig. 137 Diagrama de Secuencias Administrar Provincia

Diagrama de Secuencias Administrar ciudad

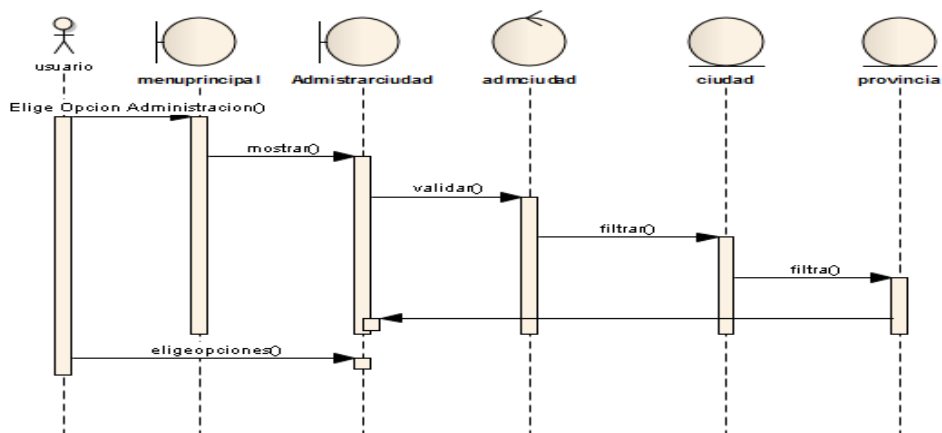


Fig. 138 Diagrama de Secuencias Administrar ciudad

Diagrama de Secuencias Administrar Zonas

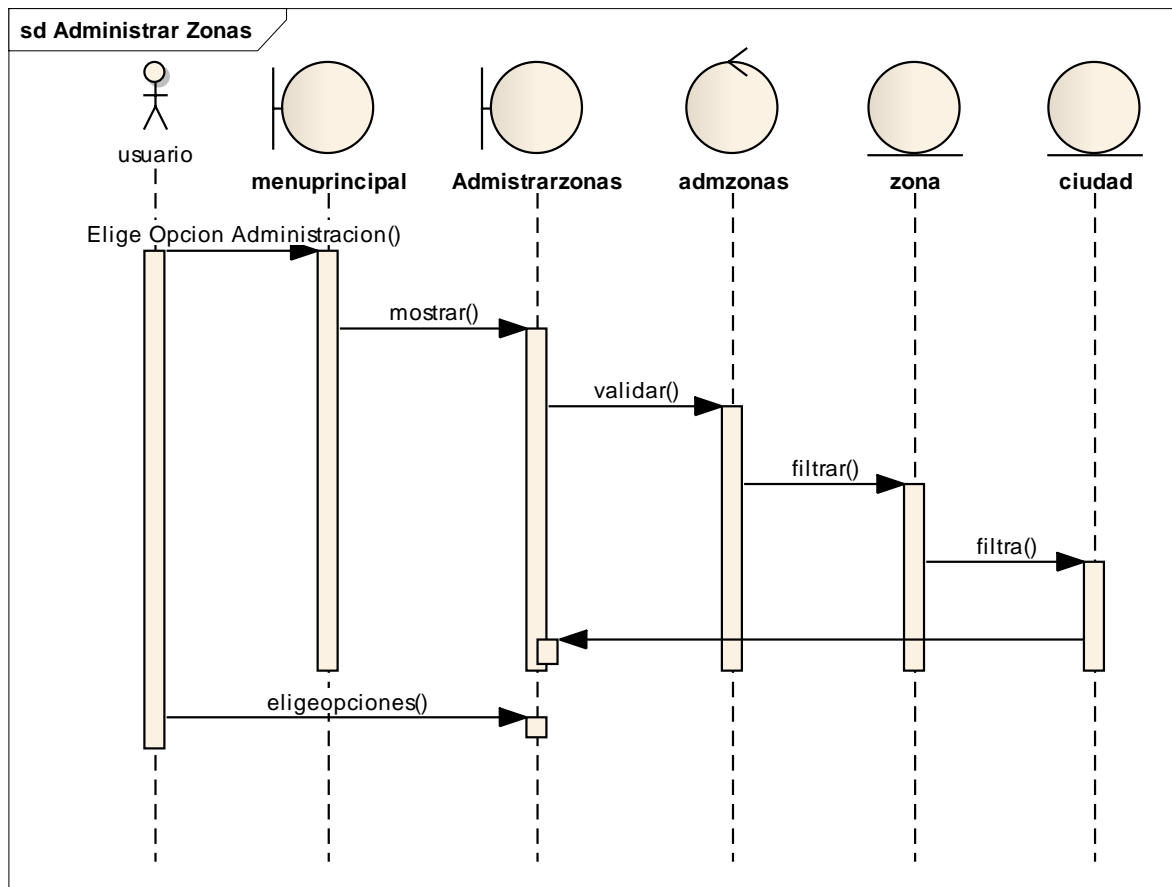


Fig. 139 Diagrama de Secuencias Administrar Zonas

Diagrama de Secuencias Administrar Enfermedades

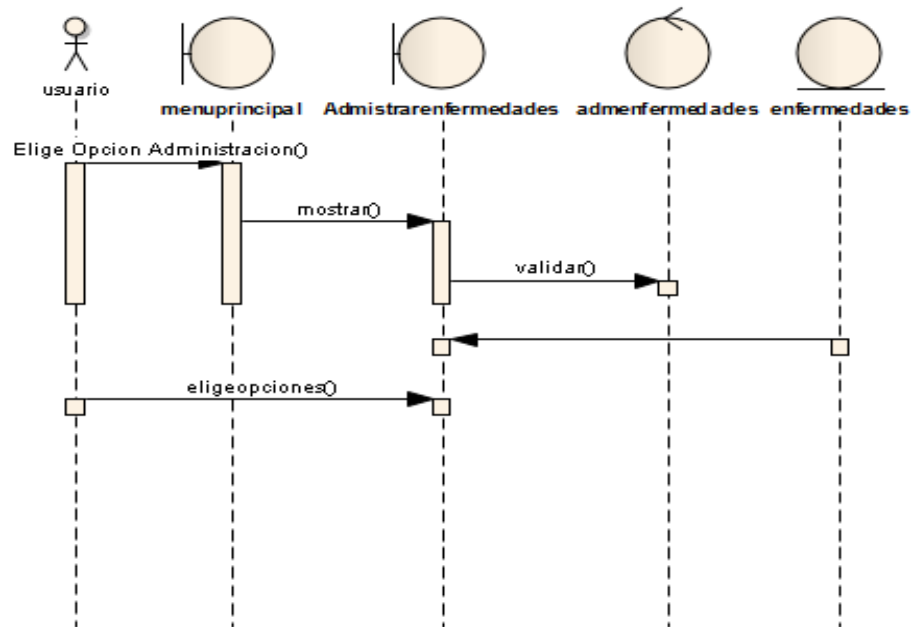


Fig. 140 Diagrama de Secuencias Administrar Enfermedades

Diagrama de Secuencias Administrar Usuarios

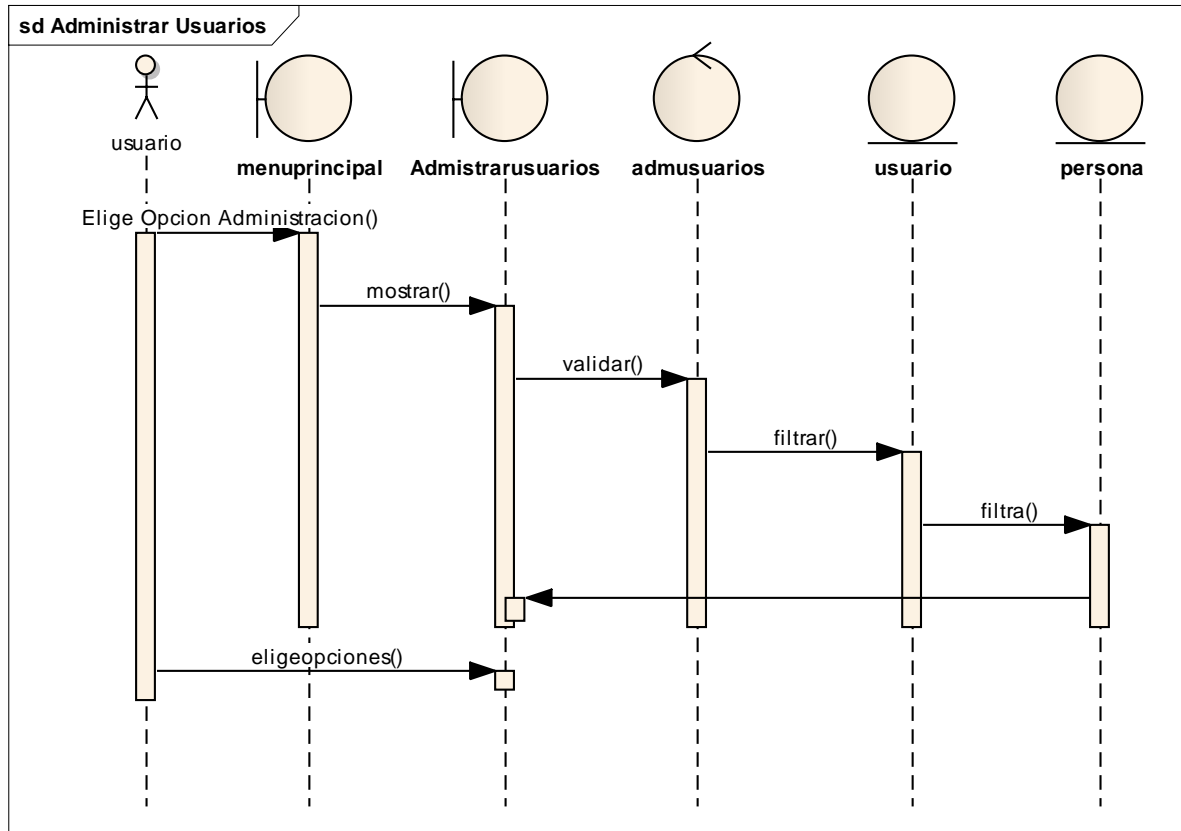


Fig. 141 Diagrama de Secuencias Administrar Usuarios

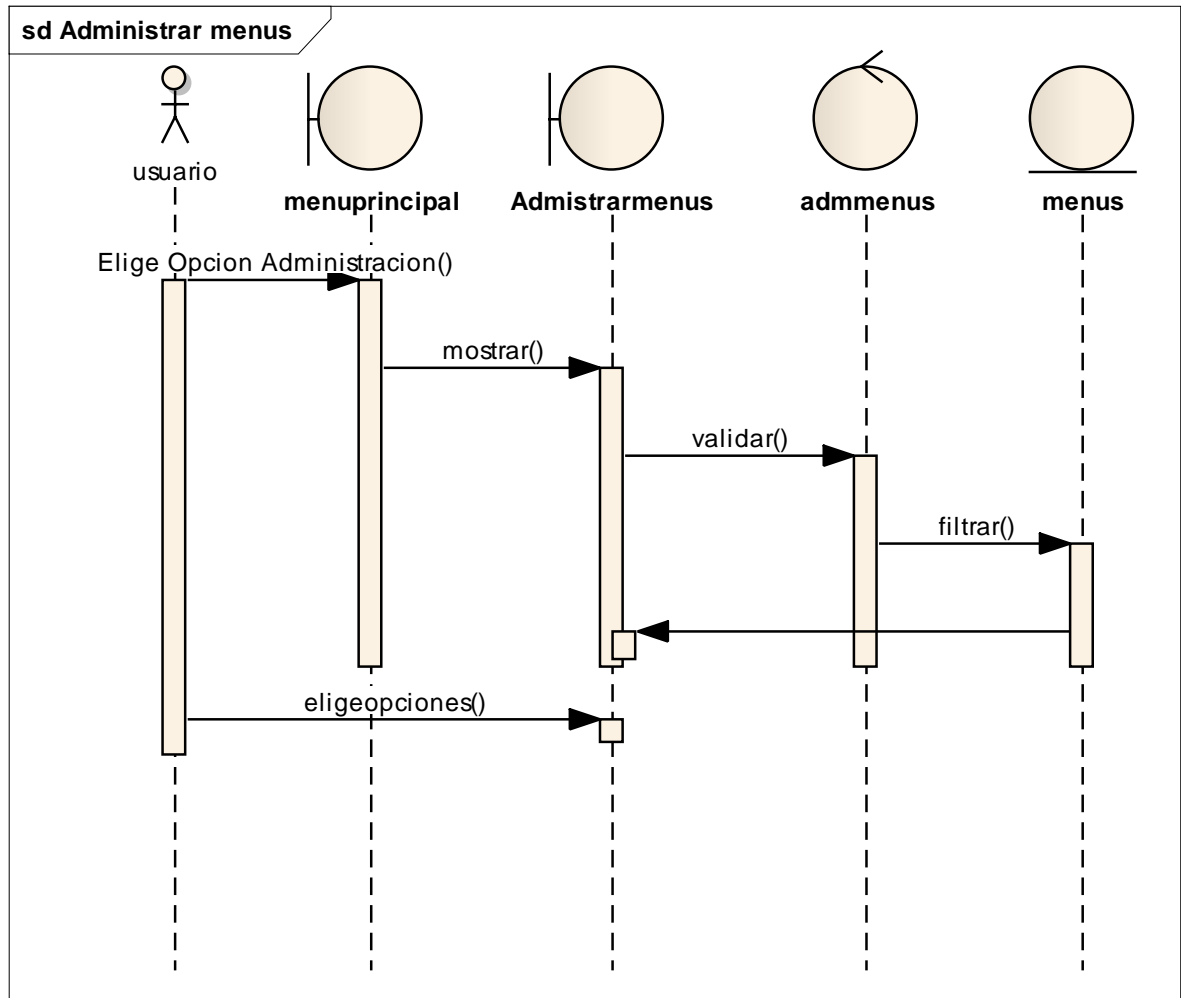
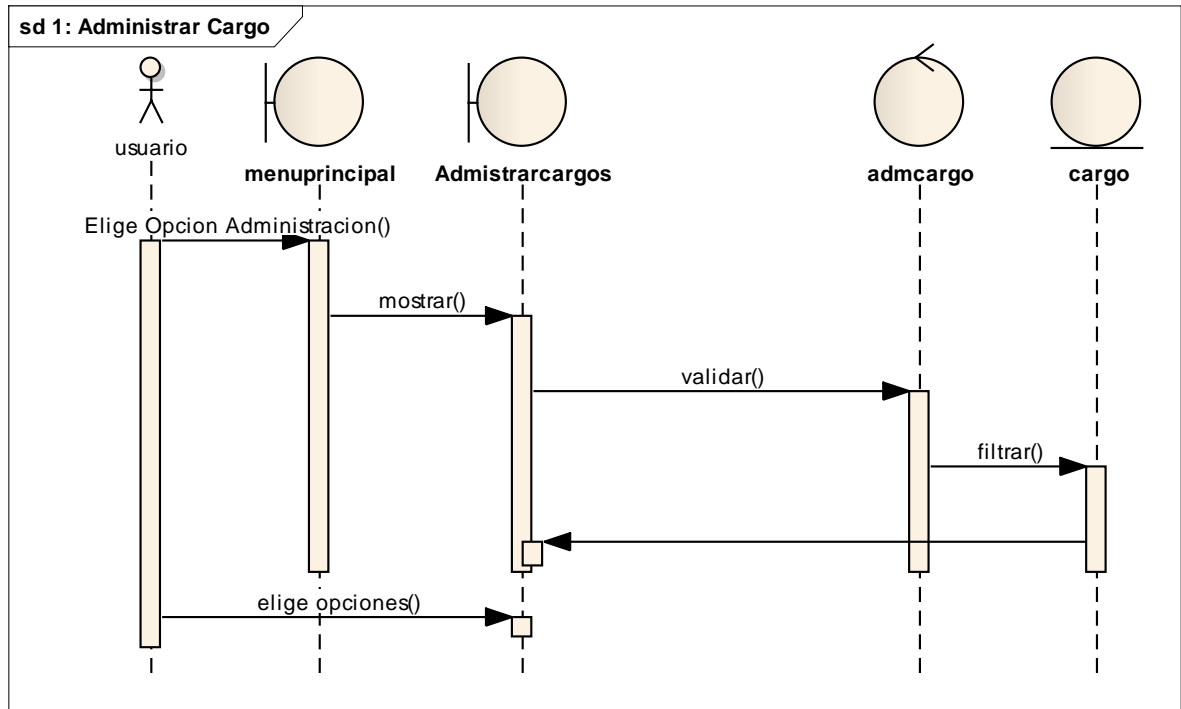
Diagrama de Secuencias Administrar Menús

Fig. 142 Diagrama de Secuencias Administrar Menús

Diagrama de Secuencias Administrar Cargos**Fig. 143 Diagrama de Secuencias Administrar Cargos**

II.1.6. MODELADO DE DIAGRAMA DE CLASES

II.1.6.1. Introducción

El Modelado de Diagrama de Clases es uno más de los diagramas requeridos en la fase de Análisis/Diseño de la metodología RUP la cual estamos implementando.

II.1.6.1. 1. Propósito

- Comprender la estructura del sistema deseado para la organización
- Identificar clases de análisis y diseño

II.1.6.1.2. Alcance

- Describir las clases y objetos de diseño del sistema en su segunda iteración
- Identificar y definir los *objetos del sistema* según los *objetivos* del sistema deseado aprobado por la organización.

II.1.6.2. Diagrama de Clases

II.1.7. Modelo de Datos - Tablas

II.1.7.1. Modelo Entidad – Relación

II.1.7.2. Tablas, Scripts

Tabla bolsa

	codbolsa [PK] character	nolote character vai	notub character vai	codtipo character vai	capacidad character vai
1	B1	870090	009865	02	430
2	B10	4546547	68769680	02	430
3	B2	123	543546	02	430

```
CREATE TABLE bolsa(
codbolsa character varying(20) NOT NULL,
nolote character varying(50),
notub character varying(50),
codtipo character varying(20),
capacidad character varying(50),
CONSTRAINT bolsa_pkey PRIMARY KEY (codbolsa),
CONSTRAINT fkbolsa642077 FOREIGN KEY (codtipo)
REFERENCES tipobolsa (codtipo) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION)
WITH (
OIDS=FALSE
);
ALTER TABLE bolsa OWNER TO postgres;
```

Tabla Cargo

	codcargo [PK] character	nombre character varying(50)	descripcion character vai	estado character vai
1	100	ADMINISTRADOR	ADM	1
2	123	encargado de tamizaje	adm tamizaj	1

```
CREATE TABLE cargo(
codcargo character varying(20) NOT NULL,
nombre character varying(50),
descripcion character varying(200),
estado character varying(20),
CONSTRAINT cargo_pkey PRIMARY KEY (codcargo))
WITH (
OIDS=FALSE
);
ALTER TABLE cargo OWNER TO postgres;
```

Tabla area

	codarea [PK] caracte	nombre character vai	coddepb character vai
*	cl	extraccion	pro

```

CREATE TABLE area
(
codarea character varying(20) NOT NULL,
nombre character varying(50),
coddepb character varying(20),
CONSTRAINT area_pkey PRIMARY KEY (codarea),
CONSTRAINT fkarea251687 FOREIGN KEY (coddepb)
REFERENCES departamentobanco (coddepb) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
oids=false
);
ALTER TABLE area OWNER TO postgres;

```

Tabla ciudad

	codciudad [PK] caracte	nombre character vai	codprovincia character vai
1	tarija	TARIJA	cer
*			

```

CREATE TABLE ciudad
(
codciudad character varying(20) NOT NULL,
nombre character varying(50),
codprovincia character varying(20),
CONSTRAINT ciudad_pkey PRIMARY KEY (codciudad),
CONSTRAINT fkciudad349853 FOREIGN KEY (codprovincia)
REFERENCES provincia (codprovincia) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
oids=false
);
ALTER TABLE ciudad OWNER TO postgres;

```


Tabla clubdonación

	codclub [PK] caracte	nombre character vai	descripcion character vai	estado character vai	codciudad character vai
1	club1	BANCO DE SA	SANGRE DISQ	1	tarija
*					

```

CREATE TABLE clubdonacion
(
codclub character varying(20) NOT NULL,
nombre character varying(50),
descripcion character varying(200),
estado character varying(20),
codciudad character varying(20),
CONSTRAINT clubdonacion_pkey PRIMARY KEY (codclub),
CONSTRAINT fkclubdonaci409692 FOREIGN KEY (codciudad)
REFERENCES ciudad (codciudad) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE clubdonacion OWNER TO postgres;

```

Tabla contador

	codcontador [PK] caracte	personas integer	donantes integer	telefonos integer	registrodo integer	tamizajec integer	tamizajel integer	tamizajesero integer	grupo integer	ext int
1	1	17	26	24	110	49	35	22	0	10
*										

extraccion integer	bolsa integer	tipobolsa integer	respuesta integer	tipopre integer	alias character vai	gestion character vai	lugar integer	usuarios integer	personal integer	cargos integer	rechazo integer
10	10	0	386	0	/12	2012	9	2	2	0	3

```

CREATE TABLE contador
(
codcontador character varying(50) NOT NULL,
personas integer DEFAULT 0,
donantes integer DEFAULT 0,
telefonos integer DEFAULT 0,
registrodo integer DEFAULT 0,
tamizajec integer DEFAULT 0,
tamizajel integer DEFAULT 0,
tamizajesero integer DEFAULT 0,
grupo integer DEFAULT 0,

```

```

extraccion integer DEFAULT 0,
bolsa integer DEFAULT 0,
tipobolsa integer DEFAULT 0,
respuesta integer DEFAULT 0,
tipopre integer DEFAULT 0,
alias character varying(250),
gestion character varying(200),
lugar integer DEFAULT 0,
usuarios integer DEFAULT 0,
personal integer DEFAULT 0,
cargos integer DEFAULT 0,
rechazo integer DEFAULT 0,
    CONSTRAINT contador_pkey PRIMARY KEY (codcontador)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE contador OWNER TO postgres;

```

Tabla departamento banco

	coddepb [PK] caracte	nombre character vai	codclub character vai
*	db1	promocion	club1

```

CREATE TABLE departamentobanco
(
coddepb character varying(20) NOT NULL,
nombre character varying(100),
codclub character varying(20),
    CONSTRAINT departamentobanco_pkey PRIMARY KEY (coddepb),
    CONSTRAINT fkdepartamen411289 FOREIGN KEY (codclub)
        REFERENCES clubdonacion (codclub) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE departamentobanco OWNER TO postgres;

```

Tabla donante

	coddon [PK] character	profesion character va	ocupacion character va	estado character va	codpersona character va	domicilio character va
1	CODD01/12	ING. INFORMA	estudiante	1	PER1	AEROPUERTO
2	CODD02/12	informatico	limpiadora	1	PER2	san geronim
3	CODD021/12	ING. INFORMA	estudiante	1	PER12	ASD

```

CREATE TABLE donante
(
coddon character varying(20) NOT NULL,
profesion character varying(100),
ocupacion character varying(100),
estado character varying(20) DEFAULT 1,
codpersona character varying(20) NOT NULL,
domicilio character varying(200),
CONSTRAINT donante_pkey PRIMARY KEY (coddon),
CONSTRAINT fkdonante70084 FOREIGN KEY (codpersona)
REFERENCES persona (codpersona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE donante OWNER TO postgres;

```

Tabla lugar de trabajo del donante

	coddon [PK] character	codlugar [PK] character	estado character va
1	CODD01/12	LUG1	1
2	CODD022/12	LUG7	1
3	CODD023/12	LUG8	1
4	CODD025/12	LUG9	1
5	CODD03/12	LUG2	1

```

CREATE TABLE donante_lugartrabajo
(
coddon character varying(20) NOT NULL,
codlugar character varying(20) NOT NULL,
estado character varying(20),
CONSTRAINT donante_lugartrabajo_pkey PRIMARY KEY (coddon, codlugar),
CONSTRAINT fkdonante_lu316595 FOREIGN KEY (coddon)
REFERENCES donante (coddon) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkdonante_lu417981 FOREIGN KEY (codlugar)
REFERENCES lugartrabajo (codlugar) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

WITH (
OIDS=FALSE
);
ALTER TABLE donante_lugartrabajo OWNER TO postgres;

```

Tabla enfermedades

	codenfe [PK] character	nombre character vai
1	e1	Sida
2	e2	herpes
3	e3	resfrio
4	e4	pauladismo
*		

```

CREATE TABLE enfermedades
(
codenfe character varying(20) NOT NULL,
nombre character varying(50),
CONSTRAINT enfermedades_pkey PRIMARY KEY (codenfe)
)
WITH (
OIDS=FALSE
);
ALTER TABLE enfermedades OWNER TO postgres;

```

Tabla extracción

	codex [PK] character	fuga character vai	emp character vai	horainicio character vai	horafin character vai	volumen character vai	peso character vai	venopunsion character vai	reaccion character vai	tiporeaccion character vai
1	EXT1	no	si	11:00	12:000	430	45	no	no	vaso vaginal
2	EXT10	no	no	7:10:29	7:10:51	430	46	si	no	nausea
3	EXT2	si	si	11:00	12:000	430	45	no	no	no

```

CREATE TABLE extraccion
(
codex character varying(20) NOT NULL,
fuga character varying(10),
emp character varying(20),
horainicio character varying(10),
horafin character varying(10),
volumen character varying(20),
peso character varying(10),
venopunsion character varying(10),
reaccion character varying(10),
tiporeaccion character varying(50),
primerosauxilios character varying(10),
estado character varying(10),

```

```

nivel character varying(50),
codbolsa character varying(20),
codregis character varying(20),
fechareg date,
responsable character varying(50),
CONSTRAINT extraccion_pkey PRIMARY KEY (codex),
CONSTRAINT fkextraccion374665 FOREIGN KEY (codregis)
REFERENCES registrodonante (codregis) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkextraccion910959 FOREIGN KEY (codbolsa)
REFERENCES bolsa (codbolsa) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE extraccion OWNER TO postgres;

```

Tabla grupo sanguíneo

	codgs [PK] character	grupo character vai	factor character vai
1	1	A	+
2	2	B	+
3	3	AB	+
4	4	O	+
5	5	A	-

```

CREATE TABLE gruposanguineo
(
codgs character varying(20) NOT NULL,
grupo character varying(20),
factor character varying(20),
CONSTRAINT gruposanguineo_pkey PRIMARY KEY (codgs)
)
WITH (
OIDS=FALSE
);
ALTER TABLE gruposanguineo OWNER TO postgres;

```

Tabla lugar de trabajo

	codlugar [PK] character	nombre character vai	direccion character vai	estado character vai	zona character vai	numero character vai	fax character vai	email character vai	casilla character vai	tele character vai
1	LUG1	PIL TJA	AEROPUERTO	1	INDUSTRIAL	234234	234234	F@	123123	77896840
2	LUG2	univ	tejar	1	fatima	689	''	''	56	76787658
3	LUG3	pil	america	1	famitima	76	7889	ytyt@hotmail	34	348

```

CREATE TABLE lugartrabajo
(
codlugar character varying(20) NOT NULL,
nombre character varying(100),
direccion character varying(200),
estado character varying(20),
zona character varying(100),
numero character varying(20),
fax character varying(20),
email character varying(50),
casilla character varying(20),
tele character varying(20),
CONSTRAINT lugartrabajo_pkey PRIMARY KEY (codlugar)
)
WITH (
OIDS=FALSE
);
ALTER TABLE lugartrabajo OWNER TO postgres;

```

Tabla menús

	codmenu [PK] character	nombre character vai	descripcion character vai	estado character(2)
1	1	Administrac hu		1
2	2	Control Don o		1

```

CREATE TABLE menus
(
codmenu character varying(20) NOT NULL,
nombre character varying(50),
descripcion character varying(200),
estado character(20),
CONSTRAINT menus_pkey PRIMARY KEY (codmenu)
)
WITH (
OIDS=FALSE
);
ALTER TABLE menus OWNER TO postgres;

```

Tabla menú procesos

	codmenu [PK] caracte	codproceso [PK] caracte	estado character vai
1	1	13	1
2	1	15	1
3	2	12	1
4	2	16	1

```

CREATE TABLE menus_procesos
(
codmenu character varying(20) NOT NULL,
codproceso character varying(20) NOT NULL,
estado character varying(20),
CONSTRAINT menus_procesos_pkey PRIMARY KEY (codmenu, codproceso),
CONSTRAINT fkmenus_proc793684 FOREIGN KEY (codproceso)
REFERENCES procesos (codproceso) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkmenus_proc820296 FOREIGN KEY (codmenu)
REFERENCES menus (codmenu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE menus_procesos OWNER TO postgres;

```

Tabla país

	codpais [PK] caracte	nombre character vai
1	bol	bolivia
*		

```

CREATE TABLE pais
(
codpais character varying(20) NOT NULL,
nombre character varying(50),
CONSTRAINT pais_pkey PRIMARY KEY (codpais)
)
WITH (
OIDS=FALSE
);
ALTER TABLE pais OWNER TO postgres;

```

Tabla persona

	codpersona [PK] character	ci character va	nombre character va	ap character va	am character va	fechanaci date	sexo character va	edad integer	estadocivil character va	estado character va
1	1	7164806	ximena	catari	sarmiento	1990-02-08	f	22	c	1
2	PER1	7158800	VICTOR	FIGUEROA	FERNANDEZ	2012-09-15	M	24	C	1
3	PER10	7173718	adriana	rodriguez	fern	1990-02-17	M	36	s	1

```

CREATE TABLE persona
(
codpersona character varying(20) NOT NULL,
ci character varying(10),
nombre character varying(50),
ap character varying(50),
am character varying(50),
fechanaci date,
sexo character varying(1),
edad integer,
estadocivil character varying(10),
estado character varying(2),
CONSTRAINT persona_pkey PRIMARY KEY (codpersona)
)
WITH (
OIDS=FALSE
);
ALTER TABLE persona OWNER TO postgres;

```

Tabla personal

	codpersonal [PK] caracte	especialidad character va	estado character va	codpersona character va
1	TRA1	ing.informa	1	PER8
2	TRA2	ing.informa	1	PER9

```

CREATE TABLE personal
(
codpersonal character varying(20) NOT NULL,
especialidad character varying(50),
estado character varying(20),
codpersona character varying(20) NOT NULL,
CONSTRAINT personal_pkey PRIMARY KEY (codpersonal),
CONSTRAINT fkpersonal231439 FOREIGN KEY (codpersona)
REFERENCES persona (codpersona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION)
WITH ( OIDS=FALSE);
ALTER TABLE personal OWNER TO postgres;

```


Tabla personal area

	codpersonal [PK] caracte	codarea [PK] caracte	fecha date
*	per1	a2	07/07/2012

```

CREATE TABLE personal_area
(
codpersonal character varying(20) NOT NULL,
codarea character varying(20) NOT NULL,
fecha date,
CONSTRAINT personal_area_pkey PRIMARY KEY (codpersonal, codarea),
CONSTRAINT fkpersonal_a759459 FOREIGN KEY (codarea)
REFERENCES area (codarea) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkpersonal_a766992 FOREIGN KEY (codpersonal)
REFERENCES personal (codpersonal) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE personal_area OWNER TO postgres;

```

Tabla cargo

	codpersonal [PK] caracte	codcargo [PK] caracte	estado character vai	gestion character vai
1	TRA1	100	1	2012
2	TRA2	100	1	2012
*				

```

CREATE TABLE personal_cargo
(
codpersonal character varying(20) NOT NULL,
codcargo character varying(20) NOT NULL,
estado character varying(20),
gestion character varying(20),
CONSTRAINT personal_cargo_pkey PRIMARY KEY (codpersonal, codcargo),
CONSTRAINT fkpersonal_c223167 FOREIGN KEY (codcargo)
REFERENCES cargo (codcargo) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkpersonal_c294251 FOREIGN KEY (codpersonal)
REFERENCES personal (codpersonal) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

WITH (
  OIDS=FALSE
);
ALTER TABLE personal_cargo OWNER TO postgres;

```

Tabla pregunta

	codpre [PK] character	pregunta character varying(200)	descripcion character vai	codtipocamp character vai
1	1	¿Ha donado sangre alguna vez?	si o no	1
2	10	¿Tiene gripe, resfriado, dolor de garganta, fiebre o dificultad para respirar hoy?	q	1
3	11	¿Ha estado embarazada en los últimos 6 meses?	1	1
4	12	¿Alguna vez ha tenido ictericia (se puso amarillo), hepatitis, trastorno hepático?	1	1

```

CREATE TABLE pregunta
(
  codpre character varying(20) NOT NULL,
  pregunta character varying(200),
  descripcion character varying(100),
  codtipocampo character varying(20),
  CONSTRAINT pregunta_pkey PRIMARY KEY (codpre),
  CONSTRAINT fkpregunta77158 FOREIGN KEY (codtipocampo)
    REFERENCES tipocampo (codtipocampo) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE pregunta OWNER TO postgres;

```

Tabla procesos

	codproceso [PK] character	nombre character varying(100)	descripcion character vai	link character varying(50)
1	12	adicionar enfermedades	add enferme	a_enfermedades.htm
2	13	buscar donante	buscar dona	ci.html
3	14	donante nuevo	registrar n	datosdo.html

```

CREATE TABLE procesos
(
  codproceso character varying(20) NOT NULL,
  nombre character varying(100),
  descripcion character varying(100),
  link character varying(50),
  CONSTRAINT procesos_pkey PRIMARY KEY (codproceso)
)
WITH (
  OIDS=FALSE

```

```
);
ALTER TABLE procesos OWNER TO postgres;
```

Tabla provincia

	codprovincia [PK] caracte	nombre character vai	coddep character vai
1	cer	cercado	tar
2	gr	gran chaco	tar
*			

```
CREATE TABLE provincia
(
codprovincia character varying(20) NOT NULL,
nombre character varying(50),
coddep character varying(20),
CONSTRAINT provincia_pkey PRIMARY KEY (codprovincia),
CONSTRAINT fkprovincia346793 FOREIGN KEY (coddep)
REFERENCES departamentos (coddep) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE provincia OWNER TO postgres;
```

Tabla punto de extracción

	codpunto [PK] caracte	nombre character vai
1	1	UNIV
2	2	COLEGIO
3	3	PLAZA PRINC

```
CREATE TABLE puntoextraccion
(
codpunto character varying(20) NOT NULL,
nombre character varying(50),
CONSTRAINT puntoextraccion_pkey PRIMARY KEY (codpunto)
)
WITH (
OIDS=FALSE
);
ALTER TABLE puntoextraccion OWNER TO postgres;
```

Tabla rechazo

	codrechazo [PK] character	nivel character vai	motivo character vai	codtipor character vai	codregis character vai	codenfe character vai
1	R1	TAMIZAJE CL	resfrio	r1	REG94/12	e1
2	R2	"TAMIZAJE C	baja presio	r1	REG97/12	e1
3	R3	"TAMIZAJE C	de resfrio	r1	REG101/12	e1

```

CREATE TABLE rechazo
(
codrechazo character varying(20) NOT NULL,
nivel character varying(50),
motivo character varying(50),
codtipor character varying(20),
codregis character varying(20),
codenfe character varying(20),
CONSTRAINT rechazo_pkey PRIMARY KEY (codrechazo),
CONSTRAINT fkrechazo24255 FOREIGN KEY (codenfe)
REFERENCES enfermedades (codenfe) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkrechazo267794 FOREIGN KEY (codtipor)
REFERENCES tiporechazo (codtipor) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkrechazo695546 FOREIGN KEY (codregis)
REFERENCES registrodonante (codregis) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE rechazo OWNER TO postgres;

```

Tabla registro donante

	codregis [PK] character	fecha date	hora char	estado character vai	gestion character vai	coddon character vai	codtipodona character vai	codpunto character vai	observacion character vai	codusu character vai
1	REG1/12	2012-09-12	-	1	2012	CODD01/12	1	1	---	1
2	REG10/12	2011-09-16	-	1	2012	CODD03/12	3	3	---	1
3	REG100/12	2012-10-09	-	1	2012	CODD09/12	1	2	---	1

```

CREATE TABLE registrodonante
(
codregis character varying(20) NOT NULL,
fecha date,
hora character varying(15),
estado character varying(20),
gestion character varying(20),
coddon character varying(20) NOT NULL,

```

```

codtipodona character varying(20),
codpunto character varying(20),
observacion character varying(200),
codusu character varying(20) NOT NULL,
CONSTRAINT registrodonante_pkey PRIMARY KEY (codregis),
CONSTRAINT fkregistrodo269195 FOREIGN KEY (codtipodona)
REFERENCES tipodonacion (codtipodona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkregistrodo551028 FOREIGN KEY (coddon)
REFERENCES donante (coddon) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkregistrodo937993 FOREIGN KEY (codpunto)
REFERENCES puntoextraccion (codpunto) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkregistrodo964244 FOREIGN KEY (codusu)
REFERENCES usuario (codusu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE registrodonante OWNER TO postgres;

```

Tabla tamizaje pregunta

	codcu [PK] caracte	codpre [PK] caracte	respuesta character vai	codtc character vai	responsable character vai	fehareg date
1	RESP1	1	no	TCLI24		
2	RESP10	14	no	TCLI24		
3	RESP100	3	si	TCLT44		

```

CREATE TABLE tamizaje_pregunta
(
codcu character varying(20) NOT NULL,
codpre character varying(20) NOT NULL,
respuesta character varying(100),
codtc character varying(20),
responsable character varying(50),
fehareg date,
CONSTRAINT tamizaje_pregunta_pkey PRIMARY KEY (codcu, codpre),
CONSTRAINT fktamizaje_p402085 FOREIGN KEY (codpre)
REFERENCES pregunta (codpre) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fktamizaje_p651774 FOREIGN KEY (codtc)
REFERENCES tamizajeclinico (codtc) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

)
WITH (
  OIDS=FALSE
);
ALTER TABLE tamizaje_pregunta OWNER TO postgres;

```

Tabla tamizaje clínico.

	codtc [PK] character	tension character vai	pulso character vai	talla character vai	peso character vai	temperatura character vai	codregis character vai	estado character vai	nivel character vai	horafin character vai	horainicio character vai	fehareg date	responsable character vai
1	TCLI1	098	345	345	45	45	REG29/12	1	TAMIZAJE CL	11:00	00		
2	TCLI10	90	90	90	90	90	REG44/12	1	TAMIZAJE CL	13:25:10	13:24:51		

```

CREATE TABLE tamizajeclinico
(
  codtc character varying(20) NOT NULL,
  tension character varying(100),
  pulso character varying(100),
  talla character varying(100),
  peso character varying(100),
  temperatura character varying(100),
  codregis character varying(20),
  estado character varying(100),
  nivel character varying(100),
  horafin character varying(100),
  horainicio character varying(100),
  fehareg date,
  responsable character varying(50),
  CONSTRAINT tamizajeclinico_pkey PRIMARY KEY (codtc),
  CONSTRAINT fktamizajecl436473 FOREIGN KEY (codregis)
    REFERENCES registrodonante (codregis) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE tamizajeclinico OWNER TO postgres;

```

Tabla tamizaje laboratorial

	codtl [PK] character	hematocrito character vai	hemoglobina character vai	codregis character vai	codgs character vai	estado character vai	nivel character vai	horainicio character vai	horafin character vai	fehareg date	responsable character vai
28	TLAB34	80%	88	REG108/12	1	1	TAMIZAJE LA	21:54:16	21:56:20	2012-11-04	1
29	TLAB35	e	e	REG109/12	1	1	TAMIZAJE LA	7:9:52	7:10:17	2012-11-12	1
30	TLAB4	jkj	jk	REG14/12	1	1	TAMIZAJE LA	00:00			

```

CREATE TABLE tamizajelaboratorial
(

```

```

codtl character varying(20) NOT NULL,
hematocrito character varying(20),
hemoglobina character varying(20),
codregis character varying(20),
codgs character varying(20),
estado character varying(20),
nivel character varying(100),
horainicio character varying(50),
horafin character varying(50),
fechareg date,
responsable character varying(50),
CONSTRAINT tamizajelaboratorial_pkey PRIMARY KEY (codtl),
CONSTRAINT fktamizajela683259 FOREIGN KEY (codregis)
REFERENCES registrodonante (codregis) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fktamizajela752776 FOREIGN KEY (codgs)
REFERENCES gruposanguineo (codgs) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE tamizajelaboratorial OWNER TO postgres;

```

Tabla serológico

codts [PK] character	codregis character vai	chagas character vai	estado character	nivel character varying(100)	fechareg date	responsable character vai	
14	TSERO21	REG108/12	M	1	TAMIZAJE SEROLOGICO	2012-11-04	1
15	TSERO22	REG109/12	M	1	TAMIZAJE SEROLOGICO	2012-11-12	1

```

CREATE TABLE tamizajeserologico
(
codts character varying(20) NOT NULL,
codregis character varying(20),
chagas character varying(100),
estado character varying(20),
nivel character varying(100),
fechareg date,
responsable character varying(50),
CONSTRAINT tamizajeserologico_pkey PRIMARY KEY (codts),
CONSTRAINT fktamizajese242834 FOREIGN KEY (codregis)
REFERENCES registrodonante (codregis) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

WITH (
  OIDS=FALSE
);
ALTER TABLE tamizajeserologico OWNER TO postgres;

```

Tabla telefono del donante

	codteldo [PK] character va	fono character vai	cel character vai	estado character vai	coddon character vai	tipo character vai
1	TEL1	54646456	345345	1	CODD01/12	PROPIO
2	TEL10		8998	1	CODD06/12	propio
3	TEL11	98245u98	98	1	CODD06/12	Familiar/Ve

```

CREATE TABLE teledonante
(
  codteldo character varying(20) NOT NULL,
  fono character varying(20),
  cel character varying(20),
  estado character varying(20) DEFAULT 1,
  coddon character varying(20) NOT NULL,
  tipo character varying(200),
  CONSTRAINT teledonante_pkey PRIMARY KEY (codteldo),
  CONSTRAINT fkteledonant554993 FOREIGN KEY (coddon)
  REFERENCES donante (coddon) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE teledonante OWNER TO postgres;

```

Tabla teléfono

	codtel [PK] character	fono character vai	cel character vai	estado character vai	codlugar character vai
*	t1	6830915	79980908	1	12

```

CREATE TABLE telefono
(
  codtel character varying(20) NOT NULL,
  fono character varying(20),
  cel character varying(20),
  estado character varying(20),
  codlugar character varying(20),
  CONSTRAINT telefono_pkey PRIMARY KEY (codtel),
  CONSTRAINT fktelefono525923 FOREIGN KEY (codlugar)

```



```

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

```

Tabla tipo de bolsa

	codtipo [PK] caracte	nombre character vai	letra character vai
1	01	doble	D
2	02	triple	T

```

CREATE TABLE tipobolsa
(
  codtipo character varying(20) NOT NULL,
  nombre character varying(50),
  letra character varying(10),
  CONSTRAINT tipobolsa_pkey PRIMARY KEY (codtipo)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE tipobolsa OWNER TO postgres;

```

Tabla tipo de campo

	codtipocampo [PK] caracte	nombre character vai
1	1	select
2	2	text
3	3	checkbox

```

CREATE TABLE tipocampo
(
  codtipocampo character varying(20) NOT NULL,
  nombre character varying(50),
  CONSTRAINT tipocampo_pkey PRIMARY KEY (codtipocampo)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE tipocampo OWNER TO postgres;

```

Tabla tipo de donación

	codtipodona [PK] character	nombre character vai
1	1	Voluntario
2	2	Reposicion
3	3	Fidelizado

```

CREATE TABLE tipodonacion
(
codtipodona character varying(20) NOT NULL,
nombre character varying(50),
CONSTRAINT tipodonacion_pkey PRIMARY KEY (codtipodona)
)
WITH (
OIDS=FALSE
);
ALTER TABLE tipodonacion OWNER TO postgres;

```

Tabla tipo de rechazo

	codtipor [PK] character	nombre character vai	descripcion character vai	fechainicio date	fechafin date
1	r1	temporal	rechazo tem		
2	r2	permanente	no puede do		

```

CREATE TABLE tiporechazo
(
codtipor character varying(20) NOT NULL,
nombre character varying(50),
descripcion character varying(100),
fechainicio date,
fechafin date,
CONSTRAINT tiporechazo_pkey PRIMARY KEY (codtipor)
)
WITH (
OIDS=FALSE
);
ALTER TABLE tiporechazo OWNER TO postgres;

```

Tabla usuario

	codusu [PK] character	login character vai	clave character vai	codpersona character vai
1	1	123	123	1
2	USU1	1234	1234	PER8
3	USU2	111	111	PER9

```

CREATE TABLE usuario
(
codusu character varying(20) NOT NULL,
"login" character varying(10) NOT NULL,
clave character varying(10) NOT NULL,
codpersona character varying(20) NOT NULL,
CONSTRAINT usuario_pkey PRIMARY KEY (codusu),
CONSTRAINT fkusuario577692 FOREIGN KEY (codpersona)
REFERENCES persona (codpersona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);
ALTER TABLE usuario OWNER TO postgres;

```

Tabla usuario menú

	codusu [PK] caracte	codmenu [PK] caracte	estado character vai
1	1	1	1
2	1	2	1
3	1	3	1

```

CREATE TABLE usuario_menus
(
codusu character varying(20) NOT NULL,
codmenu character varying(20) NOT NULL,
estado character varying(20),
CONSTRAINT usuario_menus_pkey PRIMARY KEY (codusu, codmenu),
CONSTRAINT fkusuario_me342258 FOREIGN KEY (codmenu)
REFERENCES menus (codmenu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkusuario_me377374 FOREIGN KEY (codusu)
REFERENCES usuario (codusu) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```
WITH (
  OIDS=FALSE);
ALTER TABLE usuario_menus OWNER TO postgres;
```

Tabla zona

	codzona [PK] caracte	nombre character vai	procedencia character vai	estado character vai	codciudad character vai
1	1	aeropuerto	urbano	1	tarija
2	2	la pampa	urbano	1	tarija
3	3	iv centenar	urbano	1	tarija
*					

```
CREATE TABLE zona(
codzona character varying(20) NOT NULL,
nombre character varying(100),
procedencia character varying(50),
estado character varying(1),
codciudad character varying(20),
CONSTRAINT zona_pkey PRIMARY KEY (codzona),
CONSTRAINT fkzona548588 FOREIGN KEY (codciudad)
REFERENCES ciudad (codciudad) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION)
WITH ( OIDS=FALSE );
ALTER TABLE zona OWNER TO postgres;
```

Tabla zona donante

codzona [PK] caracte	coddon [PK] caracte	estado character vai	gestion character vai
1	cod3	1	2012

```
CREATE TABLE zona_donante(
codzona character varying(20) NOT NULL,
coddon character varying(20) NOT NULL,
estado character varying(20),
gestion character varying(20),
CONSTRAINT zona_donante_pkey PRIMARY KEY (codzona, coddon),
CONSTRAINT fkzona_donan40310 FOREIGN KEY (coddon)
REFERENCES donante (coddon) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT fkzona_donan509911 FOREIGN KEY (codzona)
REFERENCES zona (codzona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION)
WITH ( OIDS=FALSE);
ALTER TABLE zona_donante OWNER TO postgres;
```

II.1.7.3.Especificación de Métodos

sesion

```
public org.springframework.web.servlet.ModelAndView
sesion(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

validar

```
public org.springframework.web.servlet.ModelAndView
validar(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

menuprincipal

```
public org.springframework.web.servlet.ModelAndView
menuprincipal(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

datosdo

```
public org.springframework.web.servlet.ModelAndView
datosdo(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

datosadd

```
public org.springframework.web.servlet.ModelAndView
datosadd(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

tamilabo

```
public org.springframework.web.servlet.ModelAndView
tamilabo(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

tamicli

```
public org.springframework.web.servlet.ModelAndView
tamicli(javax.servlet.http.HttpServletRequest request,
```

```
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

questionario

```
public org.springframework.web.servlet.ModelAndView  
questionario(javax.servlet.http.HttpServletRequest request,  
t,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

tamisero

```
public org.springframework.web.servlet.ModelAndView  
tamisero(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

extra

```
public org.springframework.web.servlet.ModelAndView  
extra(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

rechazo

```
public org.springframework.web.servlet.ModelAndView  
rechazo(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

tipodonacion

```
public org.springframework.web.servlet.ModelAndView  
tipodonacion(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,  
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

ci

```
public org.springframework.web.servlet.ModelAndView  
ci(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception  
javax.servlet.ServletException
```

a_cargo

```
public org.springframework.web.servlet.ModelAndView  
a_cargo(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_clubdonacion

```
public org.springframework.web.servlet.ModelAndView  
a_clubdonacion(javax.servlet.http.HttpServletRequest requ  
est,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_departamentobanco

```
public org.springframework.web.servlet.ModelAndView  
a_departamentobanco(javax.servlet.http.HttpServletRequest  
request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_enfermedades

```
public org.springframework.web.servlet.ModelAndView  
a_enfermedades(javax.servlet.http.HttpServletRequest request,  
                javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
        javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

a_menus_procesos

```
public org.springframework.web.servlet.ModelAndView  
a_menus_procesos(javax.servlet.http.HttpServletRequest request,  
                  javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
        javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

a_menus

```
public org.springframework.web.servlet.ModelAndView  
a_menus(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
        javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

a_personal_area

```
public org.springframework.web.servlet.ModelAndView  
a_personal_area(javax.servlet.http.HttpServletRequest req  
uest,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_personal_cargo

```
public org.springframework.web.servlet.ModelAndView  
a_personal_cargo(javax.servlet.http.HttpServletRequest re  
quest,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_proceso

```
public org.springframework.web.servlet.ModelAndView  
a_proceso(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

a_puntoextraccion

```
public org.springframework.web.servlet.ModelAndView  
a_puntoextraccion(javax.servlet.http.HttpServletRequest request,  
    javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

a_tipobolsa

```
public org.springframework.web.servlet.ModelAndView  
a_tipobolsa(javax.servlet.http.HttpServletRequest request  
    ,  
    javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

a_tipodonacion

```
public org.springframework.web.servlet.ModelAndView  
a_tipodonacion(javax.servlet.http.HttpServletRequest request,  
    javax.servlet.http.HttpServletResponse response)  
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

admcargo

```
public org.springframework.web.servlet.ModelAndView  
admcargo(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

admtipoB

```
public org.springframework.web.servlet.ModelAndView  
admtipoB(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

admtipoR

```
public org.springframework.web.servlet.ModelAndView  
admtipoR(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

admtipoD

```
public org.springframework.web.servlet.ModelAndView  
admtipoD(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)
```

```
throws java.lang.Exception,  
  
javax.servlet.ServletException  
    Throws:  
        java.lang.Exception  
        javax.servlet.ServletException
```

listalabo

```
public org.springframework.web.servlet.ModelAndView  
listalabo(javax.servlet.http.HttpServletRequest request,  
           javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,  
  
javax.servlet.ServletException  
    Throws:  
        java.lang.Exception  
        javax.servlet.ServletException
```

listacli

```
public org.springframework.web.servlet.ModelAndView  
listacli(javax.servlet.http.HttpServletRequest request,  
          javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,  
  
javax.servlet.ServletException  
    Throws:  
        java.lang.Exception  
        javax.servlet.ServletException
```

listaextra

```
public org.springframework.web.servlet.ModelAndView  
listaextra(javax.servlet.http.HttpServletRequest request,  
            javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,  
  
javax.servlet.ServletException  
    Throws:  
        java.lang.Exception  
        javax.servlet.ServletException
```

listarechazo

```
public org.springframework.web.servlet.ModelAndView
listarechazo(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
    javax.servlet.ServletException
```

listacu

```
public org.springframework.web.servlet.ModelAndView
listacu(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
    javax.servlet.ServletException
```

listapendientes

```
public org.springframework.web.servlet.ModelAndView
listapendientes(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
    javax.servlet.ServletException
```

a_usuario_menus

```
public org.springframework.web.servlet.ModelAndView  
a_usuario_menus(javax.servlet.http.HttpServletRequest req  
uest,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

buscardonantes

```
public org.springframework.web.servlet.ModelAndView  
buscardonantes(javax.servlet.http.HttpServletRequest requ  
est,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardardonante

```
public org.springframework.web.servlet.ModelAndView  
guardardonante(javax.servlet.http.HttpServletRequest requ  
est,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardaradicion

```
public org.springframework.web.servlet.ModelAndView
guardaradicion(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardartamilabo

```
public org.springframework.web.servlet.ModelAndView
guardartamilabo(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardartamicli

```
public org.springframework.web.servlet.ModelAndView
guardartamicli(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardarcu

```
public org.springframework.web.servlet.ModelAndView  
guardarcu(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardarextra

```
public org.springframework.web.servlet.ModelAndView  
guardarextra(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardarrechazo

```
public org.springframework.web.servlet.ModelAndView  
guardarrechazo(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,  
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardartipodonacion

```
public org.springframework.web.servlet.ModelAndView
guardartipodonacion(javax.servlet.http.HttpServletRequest
    request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

guardartamisero

```
public org.springframework.web.servlet.ModelAndView
guardartamisero(javax.servlet.http.HttpServletRequest req
    uest,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

guardara_cargo

```
public org.springframework.web.servlet.ModelAndView
guardara_cargo(javax.servlet.http.HttpServletRequest requ
    est,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

guardara_clubdonacion

```
public org.springframework.web.servlet.ModelAndView
guardara_clubdonacion(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_depanco

```
public org.springframework.web.servlet.ModelAndView
guardara_depanco(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_enfer

```
public org.springframework.web.servlet.ModelAndView
guardara_enfer(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_menuspro

```
public org.springframework.web.servlet.ModelAndView  
guardara_menuspro(javax.servlet.http.HttpServletRequest re  
quest,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_menus

```
public org.springframework.web.servlet.ModelAndView  
guardara_menus(javax.servlet.http.HttpServletRequest requ  
est,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_perarea

```
public org.springframework.web.servlet.ModelAndView  
guardara_perarea(javax.servlet.http.HttpServletRequest re  
quest,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_percargo

```
public org.springframework.web.servlet.ModelAndView
guardara_percargo(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_proceso

```
public org.springframework.web.servlet.ModelAndView
guardara_proceso(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_punextra

```
public org.springframework.web.servlet.ModelAndView
guardara_punextra(javax.servlet.http.HttpServletRequest request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_tibolsa

```
public org.springframework.web.servlet.ModelAndView
guardara_tibolsa(javax.servlet.http.HttpServletRequest re
quest,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_tidonacion

```
public org.springframework.web.servlet.ModelAndView
guardara_tidonacion(javax.servlet.http.HttpServletRequest
request,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardara_usumenu

```
public org.springframework.web.servlet.ModelAndView
guardara_usumenu(javax.servlet.http.HttpServletRequest re
quest,
javax.servlet.http.HttpServletResponse response)
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

addusuario

```
public org.springframework.web.servlet.ModelAndView  
addusuario(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

guardarusuario

```
public org.springframework.web.servlet.ModelAndView  
guardarusuario(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

listadonantes

```
public org.springframework.web.servlet.ModelAndView  
listadonantes(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

listausuarios

```
public org.springframework.web.servlet.ModelAndView
listausuarios(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

pregunta

```
public org.springframework.web.servlet.ModelAndView
pregunta(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
```

```
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

guardarpregunta

```
public org.springframework.web.servlet.ModelAndView
guardarpregunta(javax.servlet.http.HttpServletRequest request,
    javax.servlet.http.HttpServletResponse response)
    throws java.lang.Exception,
    javax.servlet.ServletException
```

Throws:

```
    java.lang.Exception
```

```
    javax.servlet.ServletException
```

loading

```
public org.springframework.web.servlet.ModelAndView  
loading(javax.servlet.http.HttpServletRequest request,  
javax.servlet.http.HttpServletResponse response)  
throws java.lang.Exception,
```

```
javax.servlet.ServletException
```

Throws:

```
java.lang.Exception
```

```
javax.servlet.ServletException
```

II.1.8. PROTOTIPOS DE INTERFAZ DE USUARIO

II.1.8.1. Introducción

Se trata de prototipos que permiten al usuario hacerse una idea sobre las interfaces que proveerá el sistema.

II.1.8.1.1. Propósito

Presentar los prototipos de Pantallas para que el usuario tenga una idea de la interfaz que presentara el sistema.

II.1.8.1.2. Alcance

- Mostrar los Prototipos de Pantallas, solamente el diseño que adoptarán todas.

II.1.8.2. Prototipos de Pantallas

Pantalla Acceso al Sistema

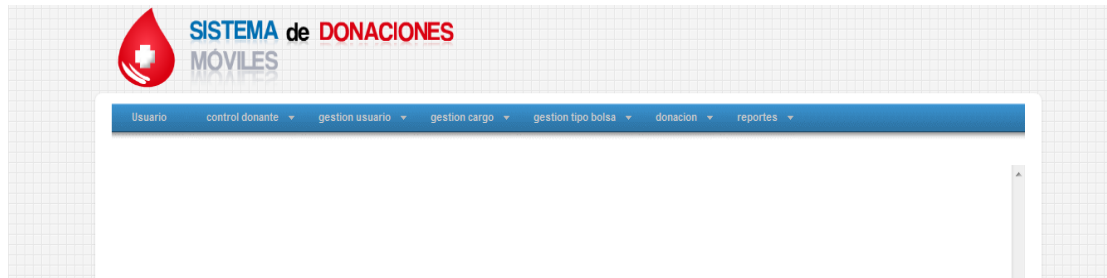
Esta pantalla se muestra al iniciar el sistema, es para acceder al sistema.



Fig. 236

Pantalla Principal

Esta pantalla contiene el menú del sistema, el cual contiene los enlaces a los distintos sectores del mismo que nos facilitaran el trabajo de administrar y gestionar.



Pantalla Administrar Donantes

Esta pantalla contiene una lista de todos los donantes, aquí se permite buscar al donante,

Administración de Donantes

Mostrar Filas Buscar

CI	NOMBRE	A.PATERNO	A.MATERNO	FECHANACI	EDAD	CODDONANTE	DOMICILIO	REG.Donacion
7158800	VICTOR	FIGUEROA	FERNANDEZ	2012-09-15	24	CODDO1/12	AEROPUERTO	Reg.Donacion
879889	wilmo	figueroa	j	1990-02-17	24	CODD022/12	aero	Reg.Donacion

Mostrando 1 hasta 2 de 2 datos (filtrados de 15 total datos)

← Anterior 1 Siguiente →

Pantalla Adicionar Donante

DATOS PERSONALES

C.I.:	<input type="text"/>
Nombres :	<input type="text"/>
Ap.Paterno:	<input type="text"/>
Ap.Materno:	<input type="text"/>
Sexo :	<input checked="" type="radio"/> M <input type="radio"/> F
Fecha Nac:	<input type="text"/>
Pais-Dep-Ciudad-Zona ::	Aeropuerto <input type="text"/>
Estado Civil :	Soltero <input type="text"/>
Domicilio:	<input type="text"/>
Telefono:	<input type="text"/>
Celular:	<input type="text"/>

Pantalla Adicionar Datos Adicionales

DATOS ADICIONALES

coddonante	CODDO25/12
Telefono (Vecino/Familiar):	<input type="text"/>
Celular (Vecino/Familiar):	<input type="text"/>
Lugar Trabajo:	<input type="text"/>
Direccion Trabajo:	
Calle/Av :	<input type="text"/>
Numero:	<input type="text"/>
Zona :	<input type="text"/>
Telefono:	<input type="text"/>
Fax:	<input type="text"/>
Casilla:	<input type="text"/>
E-mail:	<input type="text"/>

Pantalla Modificar Datos Donante

MODIFICAR DATOS PERSONALES

C.I.:

Nombres :

Ap. Paterno:

Ap. Materno:

Sexo : M
 F

Fecha Nac:

Pais-Dep-Ciudad-Zona ::

Estado Civil :

Domicilio:

Telefono:

Celular:

Profesion:

Ocupacion:

Pantalla Registrar Donación

Registrar Donacion

coddonante: CODDO1/12

fecha Registro ::

Tipo Donacion:

Lugar Extraccion:

Pantalla Registrar Tamizaje Clínico

TAMIZAJE MEDICO

Cod.Donante::CODO1/12 Cod.Registro:REG109/12
 Fecha-hora::12/11/2012 - 7:8:13
 Apariencia Gral: Buena Regular Mala
 Ingesta:

Tensión Arterial:

Pulso:

Pul./min:

Temperatura(C°):

Peso (kg):

Talla:

Refrigerio : Si No

Observaciones :

Pantalla Registrar Cuestionario

CUESTIONARIO

coddonante::CODO1/12 Cod.Registro:REG109/12
 Fecha-hora::12/11/2012 - 7:9:13
 Responder adecuadamente el siguiente cuestionario

¿Ha donado sangre alguna vez? Si :: No ::

¿Cuando fue la ultima vez?

¿Ha sido rechazado alguna vez para donar sangre? Si :: No ::

¿Se siente bien hoy? Si :: No ::

¿Alguna Vez ha tenido dolor de pecho,falta de aire,infarto o enfermedad del corazon? Si :: No ::

¿Ha padecido convulsiones? Si :: No ::

¿Ha tenido o tiene alguna enfermedad con tratamiento prolongado? Si :: No ::

¿Ha tenido enfermedades de sangre o Si :: No ::

Pantalla Registrar Tamizaje Laboratorial.

TAMIZAJE LABORATORIAL

Cod.Donante:CODO1/12 Cod.Registro:REG109/12
 Fecha - hora::12/11/2012 - 7:9:52

Grupo Sanguineo:

Hematocrito :

Hemoglobina :

Sulfato de Cobre : SI NO

Pantalla Registrar Tamizaje Serológico

TAMIZAJE SEROLOGICO

Cod.Donante:CODDO1/12 Cod.Registro:REG109/12
fecha y hora ::12/11/2012 - 7:10:17

Chagas: NR R I

Pantalla Registrar Extracción

EXTRACCIÓN

Cod.Donante:CODDO1/12 Cod.Registro:REG109/12

Duración
Hora inicio :: 7:10:29

Brazo: Izquierdo Derecho

Resultado:

Tipo Bolsa:

Peso Unidad:

Volumen:

Tipo Anti:

Reaccion Adversa: No Si

Tipo Reacción:

Nº Tubuladura:

Lote:

Fuga: No Si

Venopunsion Satisfactoria: No Si

Pantalla Registrar Rechazo

REGISTRAR RECHAZO

Cod.Registro : REG110/12
 Cod.Donante : CODDO1/12
 Nivel : "TAMIZAJE CLINICO"
 Tipo Rechazo:
 Enfermedad:
 Motivo :
 Duracion Rechazo Temporal
 Inicio:
 Fin:

Pantalla Administrar Tamizaje Clínico

Administrar Tamizaje Clínico

Mostrar Filas Buscar

COD TAMIZAJE CLINICO	FECHA	COD-DONANTE	NOMBRE COMPLETO	MODIFICAR	TAMIZAJE	CUESTIONARIO
CODTC1	11/11/2012	CD1	patricia fernandez	Modificar	ver	ver
CODTC2	11/11/2012	CD2	ximena duran	Modificar	ver	ver

Mostrando 1 hasta 2 de 2 datos

Pantalla Ver Cuestionario

CUESTIONARIO DE TAMIZAJE CLINICO

fecha: 2012-11-04
 coddo: CODDO4/12
 nombre completo: hugo vaca jj

PREGUNTA	RESPUESTA
¿Ha donado sangre alguna vez?	no
¿Cuándo fue la última vez?	nnnn
¿Ha sido rechazado alguna vez para donar sangre?	no
¿Se siente bien hoy?	no
¿Alguna Vez ha tenido dolor de pecho, falta de aire, infarto o enfermedad del corazón?	no
¿Ha padecido convulsiones?	no
¿Ha tenido o tiene alguna enfermedad con tratamiento prolongado?	no
¿Ha tenido enfermedades de sangre o sangrados anormales?	no
¿Tiene gripe, resfriado, dolor de garganta, fiebre o dificultad para respirar hoy?	no
¿Ha estado embarazada en los últimos 6 meses?	no
¿Alguna vez ha tenido ictericia (se puso amarillo), hepatitis, trastorno hepático o ha estado en contacto con personas con estos padecimientos?	no
¿Padece o está con un proceso alérgico?	no
¿Tiene Asma?	no
¿Conoce la vinchuca o recuerda haber sido picado por este insecto?	no
¿Se ha realizado alguna vez un examen para la enfermedad del chagas?	no
¿Algun familiar padece de la enfermedad de chagas?	no

Pantalla Administrar Tamizaje Laboratorial

Administrar Tamizaje Laboratorial
▲

Mostrar Filas Buscar

COD TAMIZAJE LABORATORIAL	FECHA	COD.DONANTE	NOMBRE COMPLETO	MODIFICAR	TAMIZAJE
CODTL1	11/11/2012	CD1	patricia fernandez	Modificar	ver
CODTL2	11/11/2012	CD2	ximena duran	Modificar	ver

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

Pantalla Administrar Extracción

Administrar Extraccion
▲

Mostrar Filas Buscar

COD EXTRACCION	FECHA	COD.DONANTE	NOMBRE COMPLETO	MODIFICAR	TAMIZAJE
CODEX1	11/11/2012	CD1	patricia fernandez	Modificar	ver
CODEX2	11/11/2012	CD2	ximena duran	Modificar	ver

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

Pantalla Administrar Donaciones

Administrar Donaciones
▲

Mostrar Filas Buscar

REG. DONACION	FECHA	COD.DONANTE	NOMBRE COMPLETO	COMPLETAR	RESULTADO
reg1	11/11/2012	CD1	patricia fernandez	tamizaje laboratorial	ver
reg2	11/11/2012	CD2	ximena duran	extraccion	ver

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

Pantalla Administrar Rechazo

Administrar Rechazos
▲

Mostrar Filas Buscar

COD RECHAZO	FECHA	COD.DONANTE	NOMBRE COMPLETO	MODIFICAR	RECHAZO
R1	11/11/2012	CD1	patricia fernandez	modificar	ver
R2	11/11/2012	CD2	ximena duran	modificar	ver

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

Pantalla Administrar Pregunta.

Administrar Preguntas

Mostrar Filas Buscar

COD PREGUNTA	PREGUNTA	MODIFICAR	BORRAR
p1	¿Tiene gripe, resfriado, dolor de garganta, fiebre o dificultad para respirar hoy?	modificar	borrar
p2	¿Padece o esta con un proceso alergico?	modificar	borrar
p3	¿Ha donado sangre alguna vez?	modificar	borrar

Mostrando 1 hasta 3 de 3 datos

← Anterior 1 Siguiente →

Pantalla Adicionar Pregunta

REGISTRAR PREGUNTAS

Cod Pregunta ::

Pregunta ::

Descripcion ::

Tipo Campo :: ▼

Pantalla Administrar Tipo Bolsa

Administrar Tipo Bolsa

Mostrar Filas Buscar

COD TIPO	NOMBRE	LETRA	MODIFICAR	BORRAR
CTB1	TRIPLE	T	modificar	borrar
CTB2	DOBLE	D	modificar	borrar

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

PantallaAdicionar Tipo Bolsa

ADICIONAR TIPO BOLSA

Cod_tipo:

Nombre:

Letra:

PantallaAdministrar Tipo Donación

Administrador Tipo Donacion

Mostrar Filas Buscar

COD TIPO	NOMBRE	MODIFICAR	BORRAR
CTD1	voluntario	modificar	borrar
CTD2	fidelizado	modificar	borrar

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

PantallaAdicionar Tipo Donación

ADICIONAR TIPO DONACION

Cod_tipo:

Nombre:

PantallaAdministrar Lugar de Donación

Administrador Lugar Donacion

Mostrar Filas Buscar

COD LUGAR	NOMBRE	MODIFICAR	BORRAR
CL1	universidad juan misael saracho	modificar	borrar
CL2	cruz roja	modificar	borrar
CL3	plaza principal	modificar	borrar

Mostrando 1 hasta 3 de 3 datos

← Anterior 1 Siguiente →

PantallaAdicionar Lugar Donación

**REGISTRAR
PUNTO EXTRACCION**

Cod_punto:

Nombre :

PantallaAdministrar Departamentos

Administrar Departamentos

+

COD-DEP	NOMBRE	PAIS	MODIFICAR	BORRAR
CDE1	tarija	bolivia	modificar	borrar
CDE2	pando	bolivia	modificar	borrar
CDE3	santa cruz	bolivia	modificar	borrar
CDE4	sucre	bolivia	modificar	borrar

Mostrando 1 hasta 4 de 4 datos

PantallaAdministrar Provincia

Administrar Provincias

+

COD.PROV	NOMBRE	DEPARTAMENTO	MODIFICAR	BORRAR
CPR01	Gran chaco	Tarija	modificar	borrar

Mostrando 1 hasta 1 de 1 datos

PantallaAdministrar ciudad

Administrar Ciudad

+

COD-CIUDAD	NOMBRE	PROVINCIA	MODIFICAR	BORRAR
CCIU1	Yacuiba	Gran chaco	modificar	borrar

Mostrando 1 hasta 1 de 1 datos

Administrar Zonas

Administrar Zonas

Mostrar Filas Buscar

COD-ZONA	NOMBRE	CIUDAD	MODIFICAR	BORRAR
CZONA1	villa fatima	Cercado	modificar	borrar
CZONA2	el tejar	Cercado	modificar	borrar

Mostrando 1 hasta 2 de 2 datos

← Anterior 1 Siguiente →

Pantalla Administrar Enfermedades

Administrar Enfermedades

Mostrar Filas Buscar

COD-ENFE	NOMBRE	MODIFICAR	BORRAR
CENF1	sida	modificar	borrar
CENF2	herpes	modificar	borrar
CENF3	Chagas	modificar	borrar

Mostrando 1 hasta 3 de 3 datos

← Anterior 1 Siguiente →

Pantalla Adicionar Enfermedad

ADICIONAR ENFERMEDADES

Cod_enfermedad:	
Nombre:	
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>	

Pantalla Administrar Usuarios

Aministracion de Usuarios

Mostrar Filas Buscar

CI	NOMBRE	A.PATerno	A.MATerno	FECHANACI	EDAD	CODUSUARIO	LOGIN	MODIFICAR
345345	adriana	figueroa	fernandez	0012-09-16	33	USU2	111	Modificar
441231	victor	figueroa	fernandez	0012-09-16	34	USU1	1234	Modificar
7164806	ximena	catari	sarmiento	1990-02-08	22	1	123	Modificar

Mostrando 1 hasta 3 de 3 datos

← Anterior 1 Siguiente →

PantallaAsignar Menús – Usuario

ASIGNAR USUARIO-MENU	
Cod_usuario:	
ximena <input type="text"/>	
Cod_menu:	Nombre Menu
6	donacion <input type="checkbox"/>
5	reportes <input type="checkbox"/>
1	Administracion <input type="checkbox"/>
3	Exámenes <input type="checkbox"/>
4	Reportes <input type="checkbox"/>
2	Control Donante <input type="checkbox"/>
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>	

PantallaAdministrar Menús

Administrar Menus				
<div style="display: flex; justify-content: space-between; align-items: center;"> + Mostrar <input type="text" value="10"/> Filas Buscar <input type="text"/> </div>				
COD-MENU	NOMBRE	ESTADO	MODIFICAR	BORRAR
CMENU1	Reportes	1	modificar	borrar
CMENU2	Donantes	1	modificar	borrar
Mostrando 1 hasta 2 de 2 datos				
<input type="button" value="← Anterior"/> <input type="button" value="1"/> <input type="button" value="Siguiete →"/>				

PantallaAdicionar Menú

ADICIONAR MENU	
Cod_menu:	<input type="text"/>
Nombre:	<input type="text"/>
Descripcion:	<input type="text"/>
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>	

Pantalla Asignar Menú proceso

ASIGNAR MENU-PROCESO	
Cod_menu:	
<input type="text" value="Seleccione un Menu"/>	
Cod_proceso:	Nombre
<input type="button" value="Cancelar"/> <input type="button" value="Guardar"/>	

Pantalla Administrar Cargos

Administración de Cargos					
C- Cargo	NOMBRE	DESCRIPCIÓN	ESTADO	B	M
100	ADMINISTRADOR	ADM	1	B	M
67	jefe	tods	1	B	M
68	direc	dr	1	B	M
123	encargado de tamizaje clinico	edm tamizaje clinico	1	B	M
54	jefe de promocion	u	1	B	M

Pantalla Adicionar Cargo

ADICIONAR CARGO

Cod_Cargo:

Nombre Cargo:

Descripcion:

Generar Reportes

Pantalla Reporte Tamizaje Clínico

Lista Tamizaje CLINICO									
fechainicio ::	<input type="text" value="12/11/2012"/>	fechafin ::	<input type="text" value="12/11/2012"/>	<input type="button" value="Consultar"/>					
COD TAMIZAJE CLINICO	FECHA	COD-DONANTE	NOMBRE COMPLETO	PESO	TALLA	LUGAR DE DONACION	HORA INICIO	HORA FIN	CUESTIONARIO
TCL149	2012-11-12	CODD01/12	VICTOR FIGUEROA FERNANDEZ	hh	h	UNIV	7:8-13	7:9-13	ver
TOTAL DE TAMIZAJES CLINICOS::									1
<input type="button" value="Imprimir"/>									

Pantalla Reporte Tamizaje Laboratorial

Lista Tamizaje Laboratorial					
fechainicio ::	<input type="text" value="10/10/2012"/>	fechafin ::	<input type="text" value="19/10/2012"/>	<input type="button" value="Consultar"/>	
COD TAMIZAJE LABORATORIAL	FECHA LLAB	COD-DONANTE	NOMBRE COMPLETO	GRUPO SANGUINEO	LUGAR DE DONACION
TLAB29	2012-10-10	CODD06/12	isaura duran fernandez	A+	UNIV
TLAB30	2012-10-11	CODD01/12	VICTOR FIGUEROA FERNANDEZ	B-	PLAZA PRINCIPAL
TLAB31	2012-10-11	CODD01/12	VICTOR FIGUEROA FERNANDEZ	A+	UNIV
TLAB32	2012-10-11	CODD01/12	VICTOR FIGUEROA FERNANDEZ	A+	UNIV
TOTAL DE TAMIZAJES LABORATORIAL ::					4
<input type="button" value="Imprimir"/>					

PantallaReporte Extracciones

☰ Lista de Extracciones ▲

fechainicio :: 02/05/2012 fechafin :: 22/11/2012 [Consultar](#)

COD EXTRACCION	FECHA EXT.	COD-DONANTE	NOMBRE COMPLETO	LUGAR DE DONACION	TIPO DONACION	GRUPO SANGUINEO	TIPO BOLSA	CAPACIDAD	HORA INICIO	HORA FIN
EXT2	2012-10-08	CODDO21/12	adri cespedes fern	UNIV	Voluntario	O-	triple	430	11:00	12:00
EXT1	2012-10-08	CODDO21/12	adri cespedes fern	UNIV	Voluntario	O-	triple	430	11:00	12:00
EXT3	2012-10-08	CODDO22/12	wilmo figueroa j	UNIV	Voluntario	A+	triple	capacidad	17:7:32	17:7:55
EXT4	2012-10-09	CODDO22/12	wilmo figueroa j	UNIV	Voluntario	O+	doble	430	10:3:19	10:4:40
EXT5	2012-10-10	CODDO6/12	isaura duran fernandez	UNIV	Voluntario	A+	triple	430	2:52:38	2:53:45
EXT6	2012-10-11	CODDO1/12	VICTOR FIGUEROA FERNANDEZ	PLAZA PRINCIPAL	Voluntario	B-	triple	430	10:54:2	10:54:55
EXT7	2012-10-11	CODDO1/12	VICTOR FIGUEROA FERNANDEZ	UNIV	Voluntario	A+	triple	430	11:1:19	11:1:42
EXT8	2012-10-11	CODDO1/12	VICTOR FIGUEROA FERNANDEZ	UNIV	Voluntario	A+	triple	430	12:13:37	12:13:55
EXT9	2012-11-04	CODDO4/12	hugo vaca jj	UNIV	Voluntario	A+	triple	430	21:56:23	21:57:2
EXT10	2012-11-12	CODDO1/12	VICTOR FIGUEROA FERNANDEZ	UNIV	Voluntario	A+	triple	430	7:10:29	7:10:51
TOTAL DE EXTRACCIONES ::										10

[Imprimir](#)

PantallaReporte Rechazo

☰ Lista de Rechazo ▲

fechainicio :: 09/10/2012 fechafin :: 12/11/2012 [Consultar](#)

COD RECHAZO	FECHA RECHAZO	COD-DONANTE	NOMBRE COMPLETO	MOTIVO	NIVEL	LUGAR DE DONACION
R3	2012-10-09	CODDO9/12	pamela duran roca	de resfrio	"TAMIZAJE CLINICO"	COLEGIO
TOTAL DE RECHAZO ::						1

[Imprimir](#)

II.2.1. Introducción

El componente II se denomina como la “Capacitación en el uso del sistema de donaciones móviles”, donde el jefe y el personal del departamento de promoción y extensión social del banco de sangre de Tarija asistentes son capacitados.

La capacitación es un proceso que lleva a la mejora continua de la producción y con esto implantar nuevas formas de trabajo, como este caso el manejo adecuado del sistema informático desarrollado y conocimientos en el área de la tics.

II.2.2. Propósito

El propósito del componente “Capacitación en el uso del sistema de donaciones móviles”, es dar a conocer el sistema por medio de la capacitación al personal del departamento del banco de sangre, en especial a aquellos que frecuentan a realizar donaciones móviles, quienes nos cooperaron con la información necesaria para la realización de éste proyecto.

Ahora bien, a través de este componente se pretende alcanzar los siguientes Objetivos:

- Preparar al personal que frecuentan a obtener cargos en el departamento de promoción y extensión social del banco de sangre.

II.2.3. “Capacitación en el uso del sistema de donaciones móviles”

II.2.3.1. Técnicas y Métodos De Enseñanza

II.2.3.1.1. Principios Didácticos

Los métodos y técnicas de la enseñanza, independiente de las teorías que los originen deben sujetarse a algunos principios comunes, teniendo en cuenta el desarrollo y madurez pedagógica.

- **Principio de proximidad:** Integrar la enseñanza lo más cerca posible en la vida cotidiana del educando.
- **Principio de dirección:** Tornar claros y precisos los objetivos a alcanzar.
- **Principio de marcha propia y continúa:** Procura respetar las diferencias individuales, no exigiendo la misma realización de todos los educandos.
- **Principio de ordenamiento:** Con el establecimiento de un orden se busca facilitar la tarea de aprendizaje.
- **Principio de adecuación:** Es necesario que las tareas y objetivos de la enseñanza sean acordes con las necesidades del educando.
- **Principio de eficiencia:** El ideal: mínimo esfuerzo máxima eficiencia en el aprendizaje.
- **Principio de realidad psicológica:** Previene que no se debe perder de vista la edad evolutiva de los alumnos, así como tampoco sus diferencias individuales.
- **Principio de dificultad o esfuerzo:** Es preciso tener el cuidado de no colocar al educando ante situaciones de las que tenga Posibilidades de salir bien. Pues el fracaso continuado es peor veneno para la criatura humana.
- **Principio de participación:** El educando es parte activa y dinámica del proceso.
- **Principio de espontaneidad:** Cualquier proceso emprendido debe favorecer las manifestaciones naturales del educando.
- **Principio de transparencia:** El conocimiento aprendido debe replicarse en otras situaciones de la vida diaria.

- **Principio de evaluación:** Con un proceso continuo de evaluación, el docente podrá identificar a tiempo dificultades en el proceso de aprendizaje.
- **Principio reflexión:** Inducir al pensamiento reflexivo en el alumno como parte integral de actuar del ser humano.
- **Principio de responsabilidad:** Encaminar todo el proceso de enseñanza de modo que el educando madure en cuanto a comportamiento responsable.

II.2.3.2. Métodos de Enseñanza

II.2.3.2.1. Métodos de Enseñanza individualizada

Su objetivo máximo es ofrecer oportunidades de desenvolvimiento individual más eficiente, y llevar al educando a un completo desarrollo de sus posibilidades personales.

Método de Proyectos

Lleva al alumno a la realización efectiva de algo, es activo y lo lleva para que realice, actúe es en suma determinar una tarea y que el alumno la realice. Ofrece pasos para solucionar problemas con la solución para la realización que da experiencia al alumno.

- ✓ Solución de problemas por realización.
- ✓ Carácter general o global abarca conjunto de disciplinas.
- ✓ Restringido abarca una o dos disciplinas.
- ✓ Desenvuelve espíritu de iniciativa, responsabilidad, solidaridad y libertad.
- ✓ Cadena organizada de actividades para realizar algo.

Plan Dalton

Este plan se basa en la actividad individual y la libertad. El objetivo consiste en desenvolver la actividad individual e incentiva la iniciativa al dejar al alumno escoger los trabajos y el momento de realizarlos.

El alumno hace contratos semanales en lo que está interesado, trabajando inmediatamente a su ritmo y posibilidades.

Este método tiene como particularidades que se dan conferencias del profesor con alumnos para organizar trabajos, también un boletín mural donde el profesor consigna instrucciones y la hoja de tareas como documento esencial del plan que contiene explicaciones sobre trabajos, ejercicios e indicaciones de fuentes de consulta.

Algunos inconvenientes que se presentan con este método es que acentúa exageradamente la individualidad. Tiene un carácter esencialmente individual.

Técnica Winnetka

Busca conjugar las ventajas del trabajo individual con las del trabajo colectivo, sin perder las diferencias individuales.

Enseñanza por unidades o Plan Morrison

Las fases del plan empiezan por la exploración, donde se sondean los conocimientos, luego una presentación donde se expone el tema, sigue la fase de asimilación, en la cual el alumno amplía su conocimiento y continua con una fase de organización, que es el trabajo de integración para culminar con una recitación, que es la presentación oral de la unidad.

Enseñanza programada

Es el método más reciente para individualizar y permitir que cada alumno trabaje según su propio ritmo y posibilidades.

Enseñanza personalizada

Defiende la tesis que cada educando debe desarrollarse y estudiar a su propio ritmo de aprendizaje.

II.2.3.2.2. Métodos de Enseñanza Socializada

Tiene por objeto la integración social, sin descuidar la individualización.

El estudio en grupo

Se basa en los siguientes principios :

- Crear un ambiente de estudio
- Creación de liderazgo distribuido
- Formulación de objetivos
- Permitir el consenso
- Facilitar la comprensión del proceso
- Asegurar una evaluación permanente

Método de la discusión

Sirve de orientación a la clase para realizar de forma cooperativa el estudio de una unidad o tema. Se designan un coordinador y un secretario y el resto de grupo de clase.

Método de la asamblea

Toma la misma forma de una discusión ampliada pero con la diferencia como si fuera un cuerpo colegiado gubernamental.

Método de panel

Es una reunión de especialistas para la discusión general de un tema determinado, el cual es el área de dominio de los participantes. Hay tres formas básicas, panel simple, simple con alternativa y el panel de interrogadores.

II.2.3.3. Técnicas de Enseñanza

Hay muchas técnicas para hacer llegar nuestro conocimiento y lograr un aprendizaje apropiado:

Técnica expositiva

Consiste en la exposición oral, por parte del profesor; esta debe estimular la participación del alumno en los trabajos de la clase, requiere una buena motivación para atraer la atención de los educandos. Esta técnica favorece el desenvolvimiento del autodomínio, y el lenguaje.

Técnica del dictado

Consiste en que el profesor hable pausadamente en tanto los alumnos van tomando nota de lo que él dice.

Este constituye una marcada pérdida de tiempo, ya que mientras el alumno escribe no puede reflexionar sobre lo que registra en sus notas

Técnica biográfica

Consiste en exponer los hechos o problemas a través del relato de las vidas que participan en ellos o que contribuyen para su estudio. Es más común en la historia, filosofía y la literatura.

Técnica exegética

Consiste en la lectura comentada de textos relacionados con el asunto en estudio, requiere la consulta de obras de autores.

Su finalidad consiste en acostumbrar a leer las obras representativas de un autor, de un tema o una disciplina.

Técnica cronológica

Esta técnica consiste en presentar o desenvolver los hechos en el orden y la secuencia de su aparición en el tiempo.

Esta técnica puede ser progresiva o regresiva-progresiva cuando los hechos son abordados partiendo desde el pasado hasta llegar al presente.

Regresiva cuando esos mismos hechos parten desde el presente en sentido inverso hacia el pasado.

Técnica del interrogatorio

Uno de los mejores instrumentos del campo didáctico como auxiliar en la acción de educar, este permite conocer al alumno y resaltar sus aspectos positivos. Puede ser empleado para:

- Motivación de la clase.
- Estimulo para la reflexión.
- Recapitulación y síntesis de lo aprendido.

Técnica del diálogo

El gran objetivo del diálogo es el de orientar al alumno para que reflexione, piense y se convenza que puede investigar valiéndose del razonamiento.

Técnica de la discusión

Exige el máximo de participación de los alumnos en la elaboración de conceptos y en la elaboración misma de la clase.

Consiste en la discusión de un tema, por parte de los alumnos, bajo la dirección del profesor y requiere preparación anticipada.

Técnica del debate

Puede versar sobre:

- Temas que hayan provocado divergencias durante el desarrollo de una clase
- Tópicos del programa
- Dudas surgidas y no aclaradas
- Temas de actualidad social.

Técnica del seminario

El seminario es una técnica más amplia que la discusión o el debate, pudiéndose incluir ambas en su desarrollo.

- El profesor expone lo fundamental del tema.
- Los estudiantes exponen los resultados de sus estudios, donde los llevan al debate.
- Cuando no se queda aclarado el profesor presta ayuda en el tema.
- Al final son coordinadas las conclusiones, con el auxilio del profesor

- Para un seminario eficiente todos los estudiantes deben prepararse para dicho tema.

Técnica del estudio de casos

Consiste en la presentación de un caso o problema para que la clase sugiera o presente soluciones.

- El profesor es orientador
- La presentación de un caso es presentado por el profesor, un alumno, o una autoridad.
- La participación puede llevarse: las opiniones pueden ser dadas individualmente, por los alumnos el tema es subdividido en subtemas que serán dados a grupos para estudiarlos.

Técnica de problemas

Estudia los problemas de las diversas disciplinas en el orden en que van surgiendo y evolucionando, tiene por objeto desarrollar el razonamiento del alumno, a fin de prepararlo para enfrentar situaciones problemáticas que la vida puede presentarle a cada instante.

Técnica de la demostración

Es el procedimiento más deductivo y puede asociarse a cualquier otra técnica de enseñanza cuando sea necesario comprobar afirmaciones no muy evidentes o ver cómo funciona, en la práctica, lo que fue estudiado teóricamente.

Esta técnica tiene por objetivos:

- ✓ confirmar explicaciones orales o escritas
- ✓ ilustrar lo que fue expuesto teóricamente
- ✓ iniciar teóricamente una técnica para evitar errores

- ✓ propiciar un esquema de acción correcto para la ejecución de una tarea.
- ✓ •convencer racionalmente en cuanto al a veracidad de proposiciones abstractas.

Técnica de la experiencia

La experiencia es un procedimiento eminentemente activo y que procura:

- repetir un fenómeno ya conocido
- explicar un fenómeno que no es conocido
- comprobar, con razones lo que va a suceder, partiendo de experiencias
- conferir confianza para actuar en el terreno de la realidad de manera lógica
- convencer a cerca de la veracidad de la ley de causa y efecto
- Fortalecer la confianza en si mismo
- Formar la mentalidad científica
- Orientar para solucionar problemas
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

II.2.4.2. Propuesta del Proyecto referente a la Capacitación

II.2.4.2.1. Descripción del curso

Este curso está diseñado para preparar a los/las participantes en el conocimiento y utilización de las TIC, esto con el fin de que puedan manejar y administrar el sistema web desarrollado en el proyecto de manera individual y sin depender de algún otro experto.

Este curso está dirigido con preferencia al personal que realiza donaciones móviles.

II.2.4.2.2. Metas y objetivos generales del Curso

Influenciar positivamente las actitudes de los/as participantes respecto a los beneficios a utilizar en el sistema de web desarrollado en el proyecto y el uso y manejo de las TIC como medio alternativo para incrementar su competitividad.

Proporcionarle al participante los conocimientos y las habilidades necesarios para manejar y administrar correctamente el software de gestión.

II.2.4.2.3. Objetivos del Aprendizaje del/la participante

Después de asistir a los cursos de capacitación sobre las TIC y manejo del sistema de web desarrollado en el proyecto, el participante estará en capacidad de administrar el sistema de gestión de donaciones móviles.

II.2.4.2.4. Descripción de los métodos de Capacitación

El proyecto plantea llevar a cabo para la capacitación una:

- ✓ **Metodología de Enseñanza Socializada**

El cual pretende ejecutar una metodología de enseñanza socializada, dirigida al jefe y a los encargados de donaciones móviles, con lo que se busca una integración social sin descuidar la individualización.

Para llevarlas a cabo se desarrollará el curso usando las siguientes técnicas:

✓ **Técnica expositiva**

Consiste en la exposición oral, por parte del capacitador, en la que se estimulará la participación del personal en los trabajos que se realicen, este requiere una buena motivación para atraer la atención de los participantes.

✓ **Técnica de la experiencia**

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para realizar alguna acción.
- Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Convencer a cerca de la veracidad de la ley de causa y efecto.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquecer el caudal de informaciones, que mejor contribuyan a interpretar la realidad.

Demostración: Pruebas, ensayos del Sistema de web.

Para la Capacitación se empleará Manuales de usuarios (Administrador y Operador) que contienen toda la información detallada sobre el manejo de sistema de Escritorio y llevaremos un registro del personal que asiste a la capacitación.

II.2.4.2.5. Descripción de los materiales de Capacitación

Materiales impresos: Guías del curso o manuales de usuario.

Materiales audiovisuales: Equipo de computación, proyector, parlantes

II.2.4.2.6. Criterios para la selección de los/las participantes

Los/as participantes de este curso deben pertenecer al departamento de promociones y extensión social.. También, de acuerdo a la decisión de la dirección ejecutiva, pueden ser participantes del curso, los integrantes del departamento de producción y laboratorio del banco de sangre.

II.2.4.2.7. Métodos para la evaluación del curso

Cuestionario precurso: para evaluar los conocimientos precurso acerca de computación.

Guías de aprendizaje: los/as participantes evalúan sus propias habilidades

Verificación de lo aprendido: el/la capacitador/a evalúa las habilidades de los/as participantes en el manejo del sistema de gestión y su administración.

II.2.4.2.8. Duración del curso

El curso tendrá una duración de 25 días, cada clase tendrá una duración aproximada de 2 horas por día.

II.2.4.2.9. Número sugerido de participantes

Preferiblemente se recomienda no sobrepasar la cantidad de 20 participantes en el curso, esto debido a la falta de instalaciones para acoger a más personas y para favorecer al propio ambiente de aprendizaje.

II.2.4.2.10. Fechas del curso

De acuerdo al cronograma del proyecto, el curso se realizará a partir del 15 de noviembre de 2012 hasta el 19 de diciembre de 2012, salvo algún contratiempo no planificado.

II.2.4.2.11. Organizador del curso de capacitación

El curso será organizado por el director del proyecto, bajo asesoramiento de los docentes de la materia de Taller III y de la directora del banco de sangre Dr. Giovanna Martínez Flores.

II.2.4.2.12. Contenido del Curso sobre “Las TIC”

- **Que son Tic**

Las Tecnologías de Información y Comunicación (Tics), se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como medio de sistema informático.

- Hardware
- Software
- Telecomunicaciones

- **Globalización**

La globalización es el proceso por el que la creciente comunicación e interdependencia entre los distintos países del mundo unifica mercados, sociedades y culturas, a través de una serie de transformaciones sociales, económicas y políticas que les dan un carácter global. Así, los modos de producción y de movimientos de capital se configuran a escala planetaria,

mientras los gobiernos van perdiendo atribuciones ante lo que se ha denominado la sociedad en red.

En este contexto el desarrollo tecnológico, y refiriéndome en sí a la incorporación de las TIC en cada nación, hará que el desarrollo en esta llegue sin tardar demasiado, la competitividad de un país en un sistema globalizado depende en gran medida de la tecnología que adquiera, por eso es tan estrecha la relación entre TIC y globalización. Esperemos que nuestros países comiencen a incorporar las TIC en sus procesos productivos y así reducir la dependencia de potencias mundiales.

- **Las Tic y la Competitividad**

La utilización de las TIC abre nuevas oportunidades de negocios que redundan en una serie de beneficios económicos y sociales, los cuales van desde el mejoramiento y la facilitación de la comunicación entre las empresas o instituciones, hasta el manejo más eficiente de los recursos. En tal sentido, las tecnologías deben ser utilizadas por el sector en las diferentes etapas de la cadena de valor.

- Calidad
- Productividad
- Servicio
- Imagen

Los procesos centrados en la producción: En esta etapa, las TICs pueden ser utilizadas para diseñar y probar nuevos productos, procesos de pagos, sistemas de gestión automática de stocks, diferentes tipos de links electrónicos con proveedores, sistemas de control y procesos más relacionados con la producción, entre otros.

Los procesos internos: En lo referente a administración de personal, entrenamiento, reclutamiento interno, compartir y diseminar información de la compañía vía electrónica, entre otros.

Los procesos de compra on-line: Acceso a vendedores y catálogos de productos, compras y pagos electrónicos, utilización de marketplaces electrónicos, administración de inventarios, etc.

Los procesos centrados en el cliente: Marketing, solicitudes y sistemas de pagos, seguimiento y atención al cliente. El comercio electrónico es parte fundamental de esta categoría, esencial para la venta y la post-venta. Este proceso abre nuevas posibilidades de ganancia ampliando el acceso, la promoción y las ventas, mediante la reducción de las barreras geográficas. Surge también la posibilidad de enviar las mercaderías en forma electrónica, en el caso de productos digitales (tales como software, música, libros, asesoría, servicios digitales, entre otros).

- **Software**

Las soluciones a implementar deben tener influencia sobre la productividad y la rentabilidad en muchos eslabones de la cadena productiva. La cartera de soluciones para una empresa debería estar formada al menos por:

- Datos Históricos de la institución para el análisis.
- Soluciones para gestionar la planificación, producción, comercialización, así como para el cumplimiento de las normas y estándares.
- Herramientas para la ingeniería de sistemas (Capacidad de análisis y evolución de los sistemas en el tiempo).
- Sistemas de Recursos Humanos.
- Controladores de Procesos Productivos.

- Optimización de la vida útil de los procesos (Revisión de procesos).

Que debemos hacer cuando un cliente nos hace un pedido: analizar y evaluar.

- La disponibilidad de inventarios de materias primas
- La capacidad para fabricar el producto solicitado y en el tiempo establecido.
- La disponibilidad de recursos de producción y de materiales para su fabricación.
- La adecuada logística para entregar el producto (ya sea propia o de terceros)
- La rentabilidad del producto.

El Software para las instituciones vital debido a:

- Necesidad de optimizar dinámicamente los procesos
- Optimización de la cadena de suministro en la empresa y entre empresas.

- **Conclusiones**

El uso de las TIC por parte del banco de sangre está directamente asociado a la innovación, necesaria para contribuir en la mejora de servicios de donaciones.

El uso de las computadoras está relacionado con la simplificación de labores administrativas y el mejor manejo de los procesos de planificación, producción, ventas, reportes.

El uso correcto de las TIC en las instituciones, conlleva mejores controles administrativos y operativos, por lo tanto damos de hecho mejorar los niveles de administración de donaciones móviles.

II.2.4.3. Medios de Verificación

II.2.4.3. 1. Registro de los participantes de los cursos de capacitación realizados por los desarrolladores del proyecto.

El registro de asistencia se lo presenta en ANEXOS.

III.1. Conclusiones

A la conclusión de éste proyecto se puede afirmar que se llegó a alcanzar los objetivos propuestos por el mismo, los cuales son:

- Sistema informático para la gestión de información de donaciones móviles, desarrollado.
- Capacitación en el uso del sistema de donaciones móviles.

Hoy en día las instituciones son cada vez más dependientes de la información, para mantener sus actividades diarias. Por esto las instituciones hacen uso de TIC's para la gestión y automatización de sus procesos administrativos y están incrementándose.

Se puede afirmar también que gracias a la aplicación del campo informático en el área de donaciones móviles del banco de sangre de referencia departamental Tarija, es posible el acceso rápido y fácil a la información de donaciones, por parte de los diferentes usuarios de manera sencilla y eficiente; coadyuvando de este modo a mejorar la toma de decisiones dentro de las mismas.

El marco lógico sirve para la presentación de resultados finales y hace más eficiente el proceso de preparación del proyecto.

Al implementar el proyecto en el banco de sangre se agilizo la obtención de información de donaciones móviles.

Respecto al sistema antiguo se ha mejorado la realización de reportes y búsqueda de información de colectas o donantes, se tiene respaldo de registros de donaciones y la información es almacenada digitalmente y no existe pérdida de registros hechos manualmente.

Los supuestos son riesgos para el proyecto que son expresados como objetivos y estos deben cumplirse por que es vital para lograr el propósito del proyecto.

Es importante resaltar que para la realización de proyectos como éste, el uso adecuado de las metodologías, garantizan la consecución y cobertura de los componentes del mismo y que éstos hayan sido alcanzados en su totalidad de acuerdo a la solución planteada.

III.2. Recomendaciones

Las recomendaciones que podemos expresar sobre la realización de éste trabajo son las siguientes:

- Para el desarrollo de futuros proyectos con características similares a éste Se recomienda hacer uso de las metodologías utilizadas como (UML, RUP, etc.), ya que éstas son estándares para el desarrollo de éste tipo de aplicaciones, ya que permitieron llegar a la conclusión del Proyecto.
- El sistema de web está destinado para aquellos bancos de sangre que realizan donaciones móviles o puesto móvil. Por tanto éste sistema no es recomendable para otros departamentos como producción o laboratorios y otros. Esto debido a que el sistema que presenta el proyecto no cumple con todos los requerimientos necesarios para ser tomada en cuenta en otros departamentos y áreas del banco de sangre.
- Los componentes aquí implementados no resultan ser un estándar ya que se pueden adicionar otros que ayuden a obtener mejores resultados en futuros proyectos.
- Se recomienda para la implementación de red con fibra óptica, cableado de datos y un servidor propio bajo el cumplimiento de estándares de calidad.

BIBLIOGRAFIA

- 1. BOOCH GRADY, RUMBAUGH JAMES** El Lenguaje Unificado de Modelado
Manual de Referencia.
Pearson Educación

- 2. KENDAL & KENDAL** Análisis y Diseño de Sistemas.
CPM(Tercera Edición)

- 3. KORTH HENRRY,
SILBERSCHATZ ABRAHAN** Fundamentos de Bases de Datos
España, Mc Graw Hill, 1993

- 4. PRESSMAN ROGER S.** Ingeniería de Software
(Segunda Edición)
1999

- 5. SENN, JAMES** Análisis y Diseño de sistemas
de Información.
CPM(Segunda Edición)

INDICE DE FIGURAS

Figura Nro.	Página
Fig. 1 Arbol de Problemas	7
Fig. 2 Arbol de Objetivos	8
Fig. 3 Cronograma de Actividades	12
Fig. 4 Organigrama del Banco de Sangre	21
Fig. 5 Caso de Uso General del Sistema	43
Fig. 6 Acceso al sistema.	43
Fig. 7 Administrar Donantes	44
Fig. 8 Administrar Tamizaje Clínico	45
Fig. 9 Administrar Tamizaje Laboratorial	45
Fig. 10 Administrar Extracción	46
Fig. 11 Administrar Donaciones	46
Fig. 12 Administrar Rechazo	47
Fig. 13 Administrar Pregunta.	47
Fig. 14 Administrar Tipo Bolsa	48
Fig. 15 Administrar Tipo Donación	48
Fig. 16 Administrar Lugar de Donación	49
Fig. 17 Administrar Departamentos	49
Fig. 18 Administrar Provincia	50
Fig. 19 Administrar ciudad	50
Fig. 20 Administrar Zonas	51
Fig. 21 Administrar Enfermedades	51
Fig. 22 Administrar Usuarios	52
Fig. 23 Administrar Menús	52
Fig. 24 Administrar Cargos	53
Fig. 25 Generar Reportes	53
Fig. 26 Diagrama de Actividades Caso de uso Acceso al sistema.	104
Fig. 27 Diagrama de Actividades Caso de uso Administrar Donantes	105
Fig. 28 Diagrama de Actividades Caso de uso Adicionar Donante	106
Fig. 29 Diagrama de Actividades Caso de uso Adicionar Datos Adicionales	107
Fig. 30 Diagrama de Actividades Caso de uso Modificar Datos Donante	108
Fig. 31 Diagrama de Actividades Caso de uso Ver Historial	109
Fig. 32 Diagrama de Actividades Caso de uso Buscar Donante	110
Fig. 33 Diagrama de Actividades Caso de uso Registrar Donación	111
Fig. 34 Diagrama de Actividades Caso de uso Registrar Tamizaje Clínico	112
Fig. 35 Diagrama de Actividades Caso de uso Registrar Cuestionario	113
Fig. 36 Diagrama de Actividades Caso de uso Registrar Tamizaje Laboratorial	114
Fig. 37 Diagrama de Actividades Caso de uso Registrar Tamizaje Serológico	115
Fig. 38 Diagrama de Actividades Caso de uso Registrar Extracción	116
Fig. 39 Diagrama de Actividades Caso de uso Registrar Rechazo	117
Fig. 40 Diagrama de Actividades Caso de uso Administrar Tamizaje Clínico	118

Fig. 41 Diagrama de Actividades Caso de uso Modificar Tamizaje Clínico	119
Fig. 42 Diagrama de Actividades Caso de uso Ver Tamizaje Clínico	120
Fig. 43 Diagrama de Actividades Caso de uso Ver Cuestionario	121
Fig. 44 Diagrama de Actividades Caso de uso Buscar Tamizaje Clínico	122
Fig. 45 Diagrama de Actividades Caso de uso Administrar Tamizaje Laboratorial	122
Fig. 46 Diagrama de Actividades Caso de uso Modificar Tamizaje Laboratorial	123
Fig. 47 Diagrama de Actividades Caso de uso Ver Tamizaje Laboratorial	124
Fig. 48 Diagrama de Actividades Caso de uso Buscar Tamizaje Laboratorial	124
Fig. 49 Diagrama de Actividades Caso de uso Administrar Extracción	125
Fig. 50 Diagrama de Actividades Caso de uso Modificar Extracción	126
Fig. 51 Diagrama de Actividades Caso de uso Ver Extracción	127
Fig. 52 Diagrama de Actividades Caso de uso Buscar Extracción	127
Fig. 53 Diagrama de Actividades Caso de uso Administrar Donaciones	128
Fig. 54 Diagrama de Actividades Caso de uso Completar Donación	128
Fig. 55 Diagrama de Actividades Caso de uso Ver Resultado de Donación	129
Fig. 56 Diagrama de Actividades Caso de uso Buscar Registro Donación	129
Fig. 57 Diagrama de Actividades Caso de uso Administrar Rechazo	130
Fig. 58 Diagrama de Actividades Caso de uso Modificar Rechazo	131
Fig. 59 Diagrama de Actividades Caso de uso Ver Rechazo	132
Fig. 60 Diagrama de Actividades Caso de uso Buscar Rechazo	132
Fig. 61 Diagrama de Actividades Caso de uso Administrar Pregunta.	133
Fig. 62 Diagrama de Actividades Caso de uso Adicionar Pregunta	134
Fig. 63 Diagrama de Actividades Caso de uso Modificar Pregunta	135
Fig. 64 Diagrama de Actividades Caso de uso Eliminar Pregunta	136
Fig. 65 Diagrama de Actividades Caso de uso Administrar Tipo Bolsa	137
Fig. 66 Diagrama de Actividades Caso de uso Adicionar Tipo Bolsa	138
Fig. 67 Diagrama de Actividades Caso de uso Modificar Tipo Bolsa	139
Fig. 68 Diagrama de Actividades Caso de uso Eliminar Tipo Bolsa	140
Fig. 69 Diagrama de Actividades Caso de uso Buscar Tipo Bolsa	141
Fig. 70 Diagrama de Actividades Caso de uso Administrar Tipo Donación	141
Fig. 71 Diagrama de Actividades Caso de uso Adicionar Tipo Donación	142
Fig. 72 Diagrama de Actividades Caso de uso Modificar Tipo Donación	143
Fig. 73 Diagrama de Actividades Caso de uso Eliminar Tipo Donación	144
Fig. 74 Diagrama de Actividades Caso de uso Administrar Lugar de Donación	145
Fig. 75 Diagrama de Actividades Caso de uso Adicionar Lugar Donación	146
Fig. 76 Diagrama de Actividades Caso de uso Modificar Lugar Donación	147
Fig. 77 Diagrama de Actividades Caso de uso Eliminar Lugar Donación	148
Fig. 78 Diagrama de Actividades Caso de uso Administrar Departamentos	149
Fig. 79 Diagrama de Actividades Caso de uso Adicionar Departamento	150
Fig. 80 Diagrama de Actividades Caso de uso Modificar Departamento	151
Fig. 81 Diagrama de Actividades Caso de uso Eliminar Departamento	152
Fig. 82 Diagrama de Actividades Caso de uso Administrar Provincia	153
Fig. 83 Diagrama de Actividades Caso de uso Adicionar Provincia	154
Fig. 84 Diagrama de Actividades Caso de uso Modificar Provincia	155
Fig. 85 Diagrama de Actividades Caso de uso Eliminar Provincia	156

Fig. 86 Diagrama de Actividades Caso de uso Administrar ciudad	157
Fig. 87 Diagrama de Actividades Caso de uso Adicionar Ciudad	158
Fig. 88 Diagrama de Actividades Caso de uso Modificar Ciudad	159
Fig. 89 Diagrama de Actividades Caso de uso Eliminar Ciudad	160
Fig. 90 Diagrama de Actividades Caso de uso Administrar Zonas	161
Fig. 91 Diagrama de Actividades Caso de uso Adicionar Zona	162
Fig. 92 Diagrama de Actividades Caso de uso Modificar Zona	163
Fig. 93 Diagrama de Actividades Caso de uso Eliminar Zona	164
Fig. 94 Diagrama de Actividades Caso de uso Administrar Enfermedades	165
Fig. 95 Diagrama de Actividades Caso de uso Adicionar Enfermedad	166
Fig. 96 Diagrama de Actividades Caso de uso Modificar Enfermedad	167
Fig. 97 Diagrama de Actividades Caso de uso Eliminar Enfermedad	168
Fig. 98 Diagrama de Actividades Caso de uso Administrar Usuarios	169
Fig. 99 Diagrama de Actividades Caso de uso Adicionar Usuario	170
Fig. 100 Diagrama de Actividades Caso de uso Modificar Usuario	171
Fig. 101 Diagrama de Actividades Caso de uso Eliminar Usuario	172
Fig. 102 Diagrama de Actividades Caso de uso Habilitar e Inhabilitar Usuario	173
Fig. 103 Diagrama de Actividades Caso de uso Asignar Menús – Usuario	174
Fig. 104 Diagrama de Actividades Caso de uso Eliminar Menús – Usuario	175
Fig. 105 Diagrama de Actividades Caso de uso Administrar Menús	176
Fig. 106 Diagrama de Actividades Caso de uso Adicionar Menú	177
Fig. 107 Diagrama de Actividades Caso de uso Modificar Menú	178
Fig. 108 Diagrama de Actividades Caso de uso Eliminar Menú	179
Fig. 109 Diagrama de Actividades Caso de uso Administrar Cargos	180
Fig. 110 Diagrama de Actividades Caso de uso Adicionar Cargo	181
Fig. 111 Diagrama de Actividades Caso de uso Modificar Cargo	182
Fig. 112 Diagrama de Actividades Caso de uso Eliminar Cargo	183
Fig. 113 Diagrama de Actividades Caso de uso Generar Reportes	184
Fig. 114 Diagrama de Actividades Caso de uso Reporte Tamizaje Clínico	184
Fig. 115 Diagrama de Actividades Caso de uso Reporte Tamizaje Laboratorial	185
Fig. 116 Diagrama de Actividades Caso de uso Reporte Extracciones	185
Fig. 117 Diagrama de Actividades Caso de uso Reporte Rechazo	186
Fig. 118 Diagrama de Actividades Caso de uso Reporte Estadísticos	186
Fig. 119 Diagrama de Actividades Caso de uso Reporte Habilitados- Inhabilitados para Donar	187
Fig. 120 Diagrama de Actividades Caso de uso Reporte Donantes Aptos – No Aptos	187
Fig. 121 Diagrama de Secuencias Acceso al sistema.	189
Fig. 122 Diagrama de Secuencias Administrar Donantes	190
Fig. 123 Diagrama de Secuencias Adicionar Donante	191
Fig. 124 Diagrama de Secuencias Adicionar Datos Adicionales	192
Fig. 125 Diagrama de Secuencias Registrar Donación	193
Fig. 126 Diagrama de Secuencias Registrar Tamizaje Clínico	194
Fig. 127 Diagrama de Secuencias Administrar Tamizaje Clínico	195
Fig. 128 Diagrama de Secuencias Administrar Tamizaje Laboratorial	195

Fig. 129 Diagrama de Secuencias	Administrar Extracción	196
Fig. 130 Diagrama de Secuencias	Administrar Donaciones	197
Fig. 131 Diagrama de Secuencias	Administrar Rechazo	198
Fig. 132 Diagrama de Secuencias	Administrar Pregunta	198
Fig. 133 Diagrama de Secuencias	Administrar Tipo Bolsa	199
Fig. 134 Diagrama de Secuencias	Administrar Tipo Donación	199
Fig. 135 Diagrama de Secuencias	Administrar Lugar de Donación	200
Fig. 136 Diagrama de Secuencias	Administrar Departamentos	200
Fig. 137 Diagrama de Secuencias	Administrar Provincia	201
Fig. 138 Diagrama de Secuencias	Administrar ciudad	201
Fig. 139 Diagrama de Secuencias	Administrar Zonas	202
Fig. 140 Diagrama de Secuencias	Administrar Enfermedades	203
Fig. 141 Diagrama de Secuencias	Administrar Usuarios	204
Fig. 142 Diagrama de Secuencias	Administrar Menús	205
Fig. 143 Diagrama de Secuencias	Administrar Cargos	206

INDICE DE TABLAS

Tabla Nro	Página
Tabla 1 Actividades del integrantes del equipo	3
Tabla 2 Marco Lógico del Proyecto	16
Tabla 3 Presupuesto /Justificación	17
Tabla 4 SUB GRUPO 12000. Empleados no Permanentes	18
Tabla 5 SUB GRUPO 21000. Descripción de los gastos de servicios básicos	18
Tabla 6 SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal	18
Tabla 7 SUB GRUPO 24000. Descripción mantenimiento y reparación	18
Tabla 8 SUB GRUPO 25000. Descripción de los gastos en servicios profesionales y comerciales	19
Tabla 9 SUB GRUPO 31000. Descripción de los gastos Alimentos y Productos Agroforestales	19
Tabla 10 SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos	19
Tabla 11 Especificación de Caso de Uso Acceso al sistema.	56
Tabla 12 Especificación de Caso de Uso Administrar Donantes	56
Tabla 13 Adicionar Donante	57
Tabla 14 Adicionar Datos Adicionales	57
Tabla 15 Modificar Datos Donante	58
Tabla 16 Ver Historial	58
Tabla 17 Buscar Donante	59
Tabla 18 Registrar Donación	59
Tabla 19 Registrar Tamizaje Clínico	60
Tabla 20 Registrar Cuestionario	60
Tabla 21 Registrar Tamizaje Laboratorial	61
Tabla 22 Registrar Tamizaje Serológico	61
Tabla 23 Registrar Extracción	62
Tabla 24 Registrar Rechazo	62
Tabla 25 Especificación de Caso de Uso Administrar Tamizaje Clínico	63
Tabla 26 Modificar Tamizaje Clínico	63
Tabla 27 Ver Tamizaje Clínico	64
Tabla 28 Ver Cuestionario	64
Tabla 29 Buscar Tamizaje Clínico	65
Tabla 30 Especificación de Caso de Uso Administrar Tamizaje Laboratorial	65
Tabla 31 Modificar Tamizaje Laboratorial	66
Tabla 32 Ver Tamizaje Laboratorial	66
Tabla 33 Buscar Tamizaje Laboratorial	67
Tabla 34 Especificación de Caso de Uso Administrar Extracción	67
Tabla 35 Modificar Extracción	68
Tabla 36 Ver Extracción	68
Tabla 37 Buscar Extracción	69
Tabla 38 Especificación de Caso de Uso Administrar Donaciones	69

Tabla 39 Completar Donación	70
Tabla 40 Ver Resultado de Donación	70
Tabla 41 Buscar Registro Donación	71
Tabla 42 Especificación de Caso de Uso Administrar Rechazo	71
Tabla 43 Modificar Rechazo	72
Tabla 44 Ver Rechazo	72
Tabla 45 Buscar Rechazo	73
Tabla 46 Especificación de Caso de Uso Administrar Pregunta	73
Tabla 47 Adicionar Pregunta	74
Tabla 48 Modificar Pregunta	74
Tabla 49 Eliminar Pregunta	75
Tabla 50 Especificación de Caso de Uso Administrar Tipo Bolsa	75
Tabla 51 Adicionar Tipo Bolsa	76
Tabla 52 Modificar Tipo Bolsa	76
Tabla 53 Eliminar Tipo Bolsa	77
Tabla 54 Buscar Tipo Bolsa	77
Tabla 55 Especificación de Caso de Uso Administrar Tipo Donación	78
Tabla 56 Adicionar Tipo Donación	78
Tabla 57 Modificar Tipo Donación	79
Tabla 58 Eliminar Tipo Donación	79
Tabla 59 Especificación de Caso de Uso Administrar Lugar de Donación	80
Tabla 60 Adicionar Lugar Donación	80
Tabla 61 Modificar Lugar Donación	81
Tabla 62 Eliminar Lugar Donación	81
Tabla 63 Especificación de Caso de Uso Administrar Departamentos	82
Tabla 64 Adicionar Departamento	82
Tabla 65 Modificar Departamento	83
Tabla 66 Eliminar Departamento	83
Tabla 67 Especificación de Caso de Uso Administrar Provincia	84
Tabla 68 Adicionar Provincia	84
Tabla 69 Modificar Provincia	85
Tabla 70 Eliminar Provincia	85
Tabla 71 Especificación de Caso de Uso Administrar ciudad	86
Tabla 72 Adicionar Ciudad	86
Tabla 73 Modificar Ciudad	87
Tabla 74 Eliminar Ciudad	87
Tabla 75 Especificación de Caso de Uso Administrar Zonas	88
Tabla 76 Adicionar Zona	88
Tabla 77 Modificar Zona	89
Tabla 78 Eliminar Zona	89
Tabla 79 Especificación de Caso de Uso Administrar Enfermedades	90
Tabla 80 Adicionar Enfermedad	90
Tabla 81 Modificar Enfermedad	91
Tabla 82 Eliminar Enfermedad	91
Tabla 83 Especificación de Caso de Uso Administrar Usuarios	92

Tabla 84 Adicionar Usuario	92
Tabla 85 Modificar Usuario	93
Tabla 86 Eliminar Usuario	93
Tabla 87 Habilitar e Inhabilitar Usuario	94
Tabla 88 Asignar Menús – Usuario	94
Tabla 89 Eliminar Menús – Usuario	95
Tabla 90 Especificación de Caso de Uso Administrar Menús	95
Tabla 91 Adicionar Menú	96
Tabla 92 Modificar Menú	96
Tabla 93 Eliminar Menú	97
Tabla 94 Especificación de Caso de Uso Administrar Cargos	97
Tabla 95 Adicionar Cargo	98
Tabla 96 Modificar Cargo	98
Tabla 97 Eliminar Cargo	99
Tabla 98 Reporte Tamizaje Clínico	99
Tabla 99 Reporte Tamizaje Laboratorial	100
Tabla 100 Reporte Extracciones	100
Tabla 101 Reporte Rechazo	101
Tabla 102 Reporte Estadísticos	101
Tabla 103 Reporte Habilitados- Inhabilitados para Donar	102
Tabla 104 Reporte Donantes Aptos – No Aptos	102